

Instituto de Ciências Matemáticas e de Computação

ISSN - 0103-2569

DEDALUS - Acervo - ICMSC



**Documentação do sistema SASHE recuperada
por meio de engenharia reversa**

**Valéria Delisandra Feltrim
Renata Pontin de Mattos Fortes**

Nº 90



RELATÓRIOS TÉCNICOS DO ICMC

**São Carlos
Jul./1999**

Documentação do sistema SASHE recuperada por meio de Engenharia Reversa¹

Valéria Delisandra Feltrim
Renata Pontin de Mattos Fortes
{vfeltrim, renata}@icmc.sc.usp.br

Departamento de Ciências da Computação e Estatística
Instituto de Ciências Matemáticas e de Computação
Universidade de São Paulo

Resumo

Este relatório apresenta o resultado de um processo de engenharia reversa que teve como sistema alvo o sistema hipermídia SASHE. Todos os documentos recuperados estão integralmente registrados, juntamente com uma visão geral da semântica de cada documento.

Palavras-chave: Engenharia Reversa de Software, SASHE, Fusion-RE/I.

1 Introdução

A engenharia reversa de software têm por objetivo recuperar informações sobre um software, sejam essas informações de análise, de projeto ou de implementação. Não é um processo de reprodução, mas um estudo investigativo, a fim de recuperar informações úteis perdidas durante fases de desenvolvimento do software (Chikofsky 90).

Dessa forma, a engenharia reversa pode ser dividida em duas categorias, de acordo com a abrangência da investigação: redocumentação e recuperação de projeto. A primeira categoria, a *redocumentação*, visa recuperar aspectos de implementação não documentados durante a escrita do software. Compreende a criação ou revisão de uma representação semanticamente equivalente dentro de um mesmo nível de abstração. Esse processo visa criar novas visões do sistema através da análise do código fonte, com o objetivo de melhorar a compreensão do sistema. No entanto, as informações recuperadas com esse tipo de análise fornecem apenas visões estruturais, de forma que informações como a função e os propósitos do sistema exigem um maior esforço de entendimento (Costa 97).

¹ Este trabalho tem o apoio financeiro da FAPESP, #97/12999-8.

Na engenharia reversa categorizada como *recuperação de projeto*, o domínio de conhecimento, informação externa e dedução são adicionados às observações sobre o sistema para identificar abstrações de mais alto nível que sejam significativas, além daquelas obtidas diretamente pelo exame do sistema em si (Biggerstaff 89). Recuperação de projeto é a forma mais crítica de Engenharia Reversa porque tenta recuperar não só a funcionalidade do sistema, mas o processo no qual ele foi desenvolvido. Esse tipo de análise recupera mais do que visões estruturais, obtendo visões funcionais e até mesmo de domínio.

Visando a recuperação de projeto, nos apoiamos no método Fusion-RE/I (*Fusion – Reverse Engineering / Interface*) (Costa 97), que baseia-se nos conceitos e idéias da engenharia reversa orientada a objetos, e fornece mecanismos para abstrair visões funcionais (modelos de análise de sistemas de acordo com o método FUSION (Coleman 96)) e visões estruturais a partir, inicialmente, dos aspectos operacionais e de dados disponíveis na interface usuário-computador.

Este relatório apresenta os resultados obtidos pela aplicação do método de engenharia reversa Fusion-RE/I ao sistema hipermídia *SASHE - Sistema de Apoio e Suporte Hipermídia ao Ensino* (Nunes 96, Nunes 97), com o objetivo de recuperarmos a documentação de análise e projeto do sistema, inexistente até então.

Na próxima seção é apresentado o sistema alvo do processo de engenharia reversa, o SASHE. Na Seção 3 são apresentados todos os produtos resultantes da engenharia reversa realizada, que juntos compõem o conjunto de modelos propostos pelo método Fusion-RE/I. Finalmente, na Seção 4, as conclusões deste trabalho são apresentadas.

2 O Sistema SASHE

O sistema SASHE é um sistema para autoria e navegação de documentos hipermídia, cujo domínio de aplicação é o ensino. Baseado no Modelo de Contextos Aninhados (MCA) (Casanova 91), o SASHE tem como característica permitir que um autor de hiperdocumentos possa qualificar e organizar os elementos constituintes (nós) do documento, de modo que o sistema resultante contenha funções adicionais à navegação tradicional. Tais funções incluem recursos para a localização contextual do leitor, bem como estratégias instrucionais dependentes dos valores dos atributos dos nós (Nunes 97).

Funcionalmente, podem ser identificados uma hiperbase, um módulo de autoria e um módulo de navegação. Toda a funcionalidade do sistema foi documentada através

dos modelos propostos pelo método Fusion-RE/I para a etapa de recuperação funcional: o modelo de ciclo de vida, o modelo de operações e modelos de objetos. Esses modelos foram recuperados exclusivamente através da interação com a interface, sem nenhuma visão do código-fonte.

Em nível de implementação, o sistema é composto pela hiperbase, um gerenciador da hiperbase e uma interface. A análise do código fonte do software, caracterizou o sistema SASHE com cerca de dez mil linhas de código C++, implementando o gerenciador da hiperbase (71,5% do código), e cerca de quatro mil linhas de código Delphi, implementando a interface do sistema (28,5% do código). A interface entre o gerenciador da hiperbase e a interface do sistema se dá através de uma DLL, a qual exporta as funções de manipulação da hiperbase para a interface.

No total, foram analisadas quarenta e sete (47) classes, quinhentas e vinte e nove (529) funções (descartando-se os protótipos), codificadas em oitenta e quatro (84) arquivos de código C++. Parte dessa análise foi auxiliada pelo uso da ferramenta *Understand for C/C++* (STI 99), da qual foi utilizada uma cópia *demo*, disponível via Internet. Quanto aos procedimentos em Delphi, foram analisados, no total, cento e sessenta e cinco (165) procedimentos, definidos em vinte e um (21) arquivos.

3 Documentação do SASHE Recuperada

Nesta seção é apresentado o conjunto de documentos recuperados através da aplicação do método Fusion-RE/I ao sistema hipermídia SASHE. Cada uma das seguintes subseções apresenta um dos modelos propostos pelo método Fusion-RE/I, sendo que é apresentado uma visão geral do documento e o próprio documento recuperado. Os documentos mostrados nas Subseções 3.1, 3.2 e 3.3 compõem a visão funcional do software, e as subseções restantes a visão estrutural.

3.1 Modelo de Ciclo de Vida

A partir do uso intensivo do sistema, do estudo da documentação existente e das entrevistas com os usuários pôde-se definir a sequência de operações permitidas e os eventos de entrada e de saída que o sistema aceita.

O modelo de ciclo de vida é composto por expressões regulares que definem a sequência de eventos a que o sistema pode interagir durante toda a sua vida. Ele descreve o comportamento completo de como o sistema se comunica com o ambiente, desde sua criação até o seu término (Masiero 95, Penteado 96).

O modelo de ciclo de vida construído para o SASHE contém cerca de cinquenta e cinco (55) sentenças compondo as seqüências de operações permitidas no sistema, incluindo ambos os módulos que compõem o software (Professor e Estudante).

O modelo de ciclo de vida possui a seguinte sintaxe geral (Masiero 95):

life cycle [Nome:] Expressão-Regular
(nome-local = Expressão-Regular)*

onde Nome especifica o nome do sistema, e uma vez que a expressão regular seja composta por “nome-local”, são fornecidas na seqüência as expressões de sua formação.

A seguir é apresentado o modelo de ciclo de vida recuperado para o SASHE.

```
life cycle SASHE = ( PROFESSOR | ESTUDANTE | Saída )

PROFESSOR = ( HIPERBASE | JANELAS | AJUDA | b_Criar_hiperbase |
              b_ABRIR_HIPERBASE | b_SALVAR_HIPERBASE | b_ROTIEIRO |
              b_GLOSSÁRIO | ELOS | NÓS )+

HIPERBASE = ( ((Criar | ABRIR) . SALVAR) | Sair )

JANELAS = ( Lado_a_lado | Cascata | Organizar ícones | 1_Nós_e_Elos |
            Ativar_Janela_n )

AJUDA = ( Conteúdo )

ABRIR = ( (Nome_do_arquivo . (Pastas | Unidades_de_disco |
    Listar_arquivos_do_tipo | Som._Leitura | b_REDE | b_AJUDA)* .
    (b_Ok | b_Cancelar) . (
        ((#msg_não_é_possível_encontrar_este_arquivo...). b_Ok . ABRIR) |
        ((#msg_este_nome_de_arquivo_tem_muitas_letras) . b_Ok . ABRIR) )*)
    | b_Cancelar )

b_ABRIR_HIPERBASE = ( (Nome_do_arquivo . (Pastas | Unidades_de_disco |
    Listar_arquivos_do_tipo | Som._Leitura | b_REDE |
    b_AJUDA)* . (b_Ok | b_Cancelar) . (
        ((#msg_não_é_possível_encontrar_este_arquivo...). b_Ok
        . b_ABRIR_HIPERBASE) |
        ((#msg_este_nome_de_arquivo_tem_muitas_letras) . b_Ok
        . b_ABRIR_HIPERBASE) )*) ) | b_Cancelar )

b_SALVAR_HIPERBASE = ( (Nome_do_arquivo . (Pastas | Unidades_de_disco |
    Salvar_como_tipo | Som._leitura | b_REDE | b_AJUDA)*
    . (b_Ok | b_Cancelar)) . (
        ((#msg_arquivo_já_existente..._sobrepôr?) . (b_Sim |
        b_Não)) |
        ((#msg_este_nome_de_arquivo_tem_muitas_letras) . b_Ok
        . b_SALVAR_HIPERBASE))* ) | b_Cancelar )

b_REDE = ( (Letra_da_unidade . (Caminho | Reconectar_ao_iniciar)* . ((b_Ok |
    b_Cancelar) | (b_Ok . (#msg_erro_na_conexão) . b_Ok . b_Rede))) |
    b_Cancelar )

b_ROTIEIRO = ( (Assunto . (FUNÇÃO_DIDÁTICA | DIFICULDADE | PALAVRAS_CHAVE)* .
    b_Selecionar_nós . (ESCOLHER_TRILHA)+ . (Nome_do_rotieiro)* .
    b_Criar_rotieiro) | (EDITAR_ROTIEIRO)* | Fechar )

ESCOLHER_TRILHA = ( (Selecionar_nó)+ . (INSERIR_NO_ROTIEIRO |
    Visualizar_BROWSER_GRÁFICO) )
```

```

INSERIR_NO_ROTEIRO = ( (Selecionar_nó_liberdade)+ . b_Ok .
                        (#msg_inserindo_um_nó_de_contexto...) . (b_Sim |
                        b_Não) )

EDITAR_ROTEIRO = ( ((Selecionar_nó_pertencente_a_trilha)+ . ( Limpar_roteiro
                    | Apagar_nó | MUDAR_NÓ_LIBERDADE )) | Limpar_roteiro )

MUDAR_NÓ_LIBERDADE = ( (Selecionar_nó_liberdade)+ . b_Ok )

FUNÇÃO_DIDÁTICA = ( (Ajuda | Bibliografia | Definição | Exemplo | Exercício
                    | Introdução | Motivação | Resumo | Todas) |
                    Edição_do_campo )

DIFICULDADE = ( (Fácil | Regular | Difícil | Todas)|Edição_do_campo )

PALAVRAS_CHAVES = ( (Digitar_palavra)+ | Edição_do_campo | EDITAR_CAIXA_PC )

EDITAR_CAIXA_PC = ( Selecionar_palavra . Remover_palavra )*

ELOS = ((b_CRIAR_ELO | Selecionar_elo) . (b_ALTERAR_ELO | b_Eliminar_elo))*

b_CRIAR_ELO = ( (Selecionar_nó_terminal_origem . Selecionar_âncora_origem .
                SELECIONAR_NÓ_DESTINO . Selecionar_âncora_destino . ((b_Ok .
                (#msg_erro_ao_criar_o_elo) . b_Ok) | (b_Ok .
                (#msg_o_elo_foi_criado_no_contexto...) . b_Ok) | b_Cancelar))
                | b_Cancelar )

b_ALTERAR_ELO = ( ((Selecionar_nó_terminal_origem | Selecionar_âncora_origem
                    | SELECIONAR_NÓ_DESTINO | Selecionar_âncora_destino)* .
                    ((b_Ok | b_Cancelar) | (b_Ok . (#msg_erro_ao_editar_o_elo)
                    . b_Ok)) ) | b_Cancelar )

SELECIONAR_NÓ_DESTINO = ( (Selecionar_nó_de_contexto . OPÇÕES_C) |
                          (Selecionar_nó_terminal . OPÇÕES_T) )

OPÇÕES_C = ( Apresenta_nó_destino_juntamente_com_o_nó_origem |
              Apresentar_os_nós_terminals_pertencentes_ao_nó_destino)*

OPÇÕES_T = ( Apresenta_nó_destino_juntamente_com_o_nó_origem )*

NÓS = ( b_CRIAR_NÓ | b_BROWSER_ESTRUTURAL | b_BROWSER_GRÁFICO |
        SELECIONAR_NÓ )*

SELECIONAR_NÓ = ( SELECIONAR_NÓ_C | SELECIONAR_NÓ_T )

SELECIONAR_NÓ_C = ( Selecionar_nó_de_contexto .
                    ((b_Visualizar_contexto_selecionado . (VOLTAR_CONTEXTO))*
                    | b_CRIAR_NÓ | b_EDITAR_NÓ_C | b_RETIRAR_NÓ_DO_CONTEXTO |
                    b_ELIMINAR_NÓ_DA_BASE |
                    b_Abrir_nova_janela_com_o_contexto_selecionado |
                    b_BROWSER_ESTRUTURAL | b_BROWSER_GRÁFICO) )

VOLTAR_CONTEXTO = ( NÓS | b_Voltar_para_o_contexto_anterior )

SELECIONAR_NÓ_T = ( Selecionar_nó_terminal . (b_CRIAR_NÓ | b_EDITAR_NÓ_T |
                    (b_RETIRAR_NÓ_DO_CONTEXTO | (b_Visualizar_conteúdo_do_nó
                    . ((#msg_arquivo_inválido |
                    #msg_não_existe_o_arquivo_especificado) . b_Ok) |
                    (b_Visualizar_conteúdo_do_nó . NAVEGAR) |
                    b_ELIMINAR_NÓ_DA_BASE | b_BROWSER_ESTRUTURAL |
                    b_BROWSER_GRÁFICO) )

b_RETIRAR_NÓ_DO_CONTEXTO = ( (#msg_retira_nó_selecionado_do_contexto) .
                             (b_Sim | b_Não) |
                             (#msg_Erro_ao_retirar_da_hiperbase . b_Ok) )

```

```

b_ELIMINAR_NÓ_DA_BASE = ( (#msg_elimina_nó_selecionado_da_hiperbase...) .
                           (b_Sim | b_Não) )

b_CRIAR_NÓ = ( ((Assunto | Edição_do_campo) . (b_Ok | b_Cancelar) .
                ((#msg_já_existe_nó_com_este_nome) . b_Ok)* ) | NÓ_TERMINAL) |
              (b_Ok . (#msg_nome_inválido_para_o_nó) . b_Ok) | b_Cancelar )

NÓ_TERMINAL = ( ((Texto | Áudio | Vídeo | Imagem | Execução) . (Nome |
                      Edição_do_campo) . (Arquivo | Edição_do_campo |
                      b_LER_ARQUIVO) . (FUNÇÃO_DIDÁTICA | DIFICULDADE |
                      PALAVRAS_CHAVE)* . (b_Ok | b_Cancelar) .
                ((#msg_já_existe_nó_com_este_nome) . b_Ok)* ) | (b_Ok .
                (#msg_nome_inválido_para_o_nó) . b_Ok . NÓ_TERMINAL) |
                b_Cancelar )

b_EDITAR_NÓ_C = ( ((Assunto | Edição_do_campo) . (b_Ok | b_Cancelar) .
                  ((#msg_nome_inválido_para_o_nó) . b_Ok . b_EDITAR_NÓ_C)* )
                | (b_Ok | b_Cancelar) )

b_EDITAR_NÓ_T = ( (((Nome | Edição_do_campo) | (Arquivo | Edição_do_campo) |
                    b_LER_ARQUIVO | FUNÇÃO_DIDÁTICA | DIFICULDADE |
                    PALAVRAS_CHAVE | b_EDITAR_CONTEÚDO) . (b_Ok | b_Cancelar)
                  . ((#msg_nome_inválido_para_o_nó) . b_Ok .
                    b_EDITAR_NÓ_T)* ) | (b_Ok | b_Cancelar) )

b_LER_ARQUIVO = ( ((Nome_do_arquivo . (Pastas | Unidades_de_disco |
                    Listar_arquivos_do_tipo | Som._Leitura | b_REDE )* . (b_Ok
                    | b_Cancelar) .
                    ((#msg_este_nome_de_arquivo_tem_muitas_letras) . b_Ok .
                    b_LER_ARQUIVO)* ) | b_Cancelar )

b_EDITAR_CONTEÚDO = ( EDITAR_ARQUIVO_TXT | Editar_arquivo_bmp |
                      EDITAR_NÓ_DE_AJUDA)

EDITAR_ARQUIVO_TXT = ( Editar_o_conteúdo | CRIAR_ÂNCORAS )

CRIAR_ÂNCORAS = ( (Selecionar_o_texto .
                  b_Criar_âncora_no_texto_selecionado)* . b_Criar_âncoras .
                  (#msg_âncoras_criadas) . b_Ok )

EDITAR_NÓ_DE_AJUDA = ( ((Inserir_dúvidas | Inserir_respostas)* . (b_Inserir
                        | b_Gravar)* ) | Fechar )

b_BROWSER_ESTRUTURAL = ( ((Selecionar_nó_terminal .
                          b_Visualizar_conteúdo_do_nó .
                          ((#msg_não_existe_o_arquivo_especificado) . b_Ok)*
                          . NAVEGAR) | (Selecionar_nó_de_contexto .
                          (b_Abrir_nova_janela_com_o_contexto_selecionado)* )
                        | Fechar )

b_BROWSER_GRÁFICO = ( (Lista_de_contextos_disponíveis | (b_Zoom_out .
                  (b_BROWSER_GRÁFICO)* . b_Zoom_in) | (b_AJUDA_HIPGRAFO
                  . b_BROWSER_GRÁFICO) | (Explorar_nó .
                  ((#msg_erro:_arquivo_não_encontrado) . b_Ok)* .
                  (b_BROWSER_GRÁFICO . b_Voltar_nó)* ) | (Explorar_nó .
                  Fechar) | Fechar )

b_AJUDA_HIPGRAFO = ( (#msg_sobre_o_HipGrafo) . b_Ok )

NAVEGAR = ( Selecionar_âncora . ((#msg_esta_âncora_não_está_disponível) .
                                b_Ok)* )

ESTUDANTE = ( ((Nome_do_estudante)* . (Diretório)* . Hiperbase . ((b_Ok .
                  (#msg_não_existe_o_arquivo_especificado) . b_Ok .
                  MÓDULO_ESTUDANTE_VAZIO) | (b_Ok . MÓDULO_ESTUDANTE)) |
              b_Cancelar )

MÓDULO_ESTUDANTE_VAZIO = ( CONTROLES | SASHE | JANELAS_E | AJUDA )+

```

```

MÓDULO_ESTUDANTE = (CONTROLES | NAVEGAR | SASHE | JANELAS_E | AJUDA)+

SASHE = ( Sair )

JANELAS_E = ( Lado_a_lado | Cascata | Organizar_ícones | (Janela_1 |
Janela_n)* )

CONTROLES = ( (b_Está_fácil . ((#msg não há nada mais difícil! Avance aos
tópicos seguintes) . b_Ok)* ) | (b_Está_difícil . ((#msg não
há nada mais fácil! Retorne aos tópicos anteriores) . b_Ok)*
) | (b_Ajuda . ((#msg não existe nenhum_nó_ajuda) . b_Ok)* ) |
b_MAIS_INFORMAÇÕES | (b_Exercícios .
((#msg não existe nenhum_nó_exercícios) . b_Ok)* ) |
b_Glossário | (b_Bibliografia .
((#msg não existe nenhum_nó_bibliografia) . b_Ok)* ) |
b_CONTEXTO | b_HISTÓRICO | b_Onde_estou? | TÓPICOS | HISTÓRICO
)

b_MAIS_INFORMAÇÕES = ( Selecionar_nó . NAVEGAR ) *

b_CONTEXTO = ( b_BROWSER_GRÁFICO )

TÓPICOS = ( b_Avança_tópico | b_Retrocede_tópico ) +

HISTÓRICO = ( b_Avança_histórico | b_Retrocede_histórico ) +

b_HISTÓRICO = (Selecionar_nó . NAVEGAR ) *

```

Observação: O que está em *itálico* não é constante da interface do sistema, ou seja, não aparece na interface. Ou são ações do usuário, como selecionar elementos que aparecem na hiperbase, ou são recursos de abstração para facilitar a escrita e o entendimento do ciclo de vida, como por exemplo "escolher_trilha".

3.2 Modelo de Operações

O modelo de operações é decorrente do modelo de ciclo de vida. Esse modelo especifica o comportamento de uma operação de forma declarativa em termos das mudanças de estado do sistema e eventual geração de eventos de saída (Masiero 95). Cada operação identificada no modelo de ciclo de vida é descrita individualmente através de um formulário textual, contendo as seguintes informações: nome da operação, descrição informal, dados de entrada, objetos acessados e modificados, eventos de saída e os agentes que os recebem, pré-condições e resultados da operação, descritos como pós-condições (Penteado 96). Dessa forma, o modelo de operações é expresso por uma série de esquemas, sendo que existirá, no mínimo, um esquema para cada operação do sistema (Coleman 96). Foram identificadas no ciclo de vida construído para o sistema SASHE sessenta e duas (62) operações, sendo que cada uma delas foi estudada exaustivamente para que pudesse ser descrita de acordo com os requisitos do esquema proposto pelo método. Todos os esquemas juntos compõem o modelo de operações, que é mostrado a seguir, ordenados pelo nome da operação.

Operação:	1_Nós_e_Elos
Descrição:	Ativa a janela Nós e Elos do módulo professor
Lê:	
Modifica:	
Envia:	
Assume:	
Resultado:	Janela Nós e Elos ativada

Operação:	Abrir
Descrição:	Abre uma hiperbase já existente
Lê:	Nome do arquivo; diretório; drive; somente leitura; botão ok; botão cancelar
Modifica:	
Envia:	Agente externo:{#msg não é possível encontrar este arquivo}; Agente externo:{#msg este nome de arquivo tem muitas letras}
Assume:	
Resultado:	Se botão cancela, operação é cancelada Se botão ok, Se arquivo existente, a hiperbase é aberta Se arquivo não encontrado, mensagem de arquivo não encontrado é enviada, e continua na operação Abrir hiperbase Se nome de arquivo tem mais de 8 letras, mensagem de nome com muitas letras é enviada, e continua na operação Abrir hiperbase

Operação:	Apagar_nó (participante de b_Roteiro)
Descrição:	Apaga o nó selecionado do roteiro. Disponível através do botão direito do mouse clicado sobre a área de Nós Pertencentes a Trilha
Lê:	
Modifica:	Nó Trilha da hiperbase em memória
Envia:	
Assume:	Existe um nó participante do roteiro selecionado
Resultado:	Se apagar nó então o nó selecionado é apagado do roteiro Esta operação só será efetivamente realizada após a operação Criar Roteiro

Operação:	Ativar_Janela_n (Janela_1 ou Janela_n)
Descrição:	Ativa uma das janelas listadas no menu Janelas. Esta operação consiste em rolar o cursor e escolher uma das janelas disponíveis no menu
Lê:	
Modifica:	
Envia:	
Assume:	
Resultado:	A janela n estará ativa

Operação:	b_Abrir_hiperbase
Descrição:	Abre uma hiperbase já existente
Lê:	Nome do arquivo; diretório; drive; somente leitura; botão ok; botão cancelar
Modifica:	
Envia:	Agente externo:{#msg não é possível encontrar o arquivo}; Agente externo:{#msg este nome de arquivo tem muitas letras}
Assume:	
Resultado:	Se botão cancela, operação é cancelada Se botão ok, Se arquivo existente, a hiperbase é aberta Se arquivo não encontrado, mensagem de arquivo não encontrado é enviada, e continua na operação Abrir hiperbase Se nome de arquivo tem mais de 8 letras, mensagem de nome com muitas letras é enviada, e continua na operação Abrir hiperbase

Operação:	b_Abrir_nova_janela_com_o_contexto_selecionado
Descrição:	Abre o contexto selecionado em uma nova janela. Dessa forma, o contexto atual passa a ser o contexto que estiver com a janela ativa
Lê:	
Modifica:	
Envia:	
Assume:	Existe um contexto selecionado
Resultado:	É aberta uma nova janela Nós e Elos com o contexto selecionado. Enquanto esta nova janela estiver ativa, este contexto será o atual

Operação:	b_Ajuda (Controle de Navegação)
Descrição:	Abre um nó dentro do contexto navegado cujo atributo função didática seja ajuda. A busca pelo nó com tal atributo se dá no menor contexto que contém o nó atual; caso não encontre, a busca é feita até o nó liberdade do nó atual
Lê:	
Modifica:	
Envia:	Agente Externo:{#msg_não_existe_nenhum_nó_ajuda}
Assume:	Os nós da hiperbase estão qualificados pelo menos através do atributo função didática
Resultado:	Se existir no contexto um nó cuja função didática seja ajuda então esse nó será aberto Se não existir tal nó então a mensagem de que não existe um nó de ajuda será enviada e o sistema retorna ao estado anterior a operação

Operação:	b_Ajuda_hipgrafo (Participante de Browser Gráfico)
Descrição:	Abre uma janela com informações sobre o hipgrafo, que implementa o browser gráfico
Lê:	botão Ok
Modifica:	
Envia:	Agente Externo:{#msg_sobre_o_HipGrafo}
Assume:	
Resultado:	Se botão Ok então retorna ao Browser Gráfico

Operação:	b Alterar elo
Descrição:	Edita um elo já existente, podendo alterar qualquer atributo componente do mesmo
Lê:	botão Ok; botão Cancelar
Modifica:	Lista de Elos da hiperbase em memória
Envia:	Agente Externo:{#msg erro ao editar o elo}
Assume:	Existe um elo selecionado
Resultado:	Se botão Cancelar então a operação é cancelada Se botão Ok então Se âncoras origem e destino foram alteradas incorretamente, ou nó origem igual ao nó destino então mensagem de erro ao editar o elo é enviada e a operação é cancelada Senão o elo é alterado

Operação:	b_Avança_histórico (Controle de Navegação)
Descrição:	Avança na trilha de nós percorrida historicamente pelo leitor
Lê:	
Modifica:	
Envia:	
Assume:	
Resultado:	O próximo nó na trilha de nós (histórico) é aberto e o atual é fechado

Operação:	b_Avança_tópico (Controle de Navegação)
Descrição:	Avança no roteiro de navegação
Lê:	
Modifica:	
Envia:	
Assume:	
Resultado:	Se botão Avança Tópico então o próximo nó terminal ou contexto do roteiro é aberto e o que estava sendo apresentado é fechado

Operação:	b_Bibliografia (Controle de Navegação)
Descrição:	Abre um nó dentro do roteiro navegado cujo atributo função didática seja bibliografia. A busca pelo nó com tal atributo se dá no menor contexto que contém o nó atual; caso não encontre a busca continua até o nó liberdade do nó atual
Lê:	
Modifica:	
Envia:	Agente Externo:{#msg_não_existe_nenhum_nó_bibliografia}
Assume:	Os nós da hiperbase estão qualificados pelo menos através do atributo função didática
Resultado:	Se existir no contexto um nó cuja função didática seja bibliografia então esse nó será aberto Se não existir tal nó, então a mensagem de que não existe um nó bibliografia será enviada e o sistema retorna ao estado anterior a operação

Operação:	b_Browser_Estrutural
Descrição:	Abre a janela do Browser Estrutural com o contexto atual
Lê:	
Modifica:	
Envia:	
Assume:	
Resultado:	É aberta a janela do browser estrutural onde são mostrados de forma estrutural os nós pertencentes ao contexto atual. Se o contexto for vazio a janela será aberta assim mesmo.

Operação:	b_Contexto (Controle de Navegação)
Descrição:	Abre o browser gráfico com o contexto ao qual pertence o nó atual
Lê:	
Modifica:	
Envia:	
Assume:	Existe um nó sendo apresentado
Resultado:	A janela do browser gráfico passa a ser a janela ativa e pode ser navegada normalmente

Operação:	b_Criar_âncora_no_texto_selecionado
Descrição:	Coloca o sinal de marcação que define uma âncora no texto selecionado
Lê:	
Modifica:	Arquivo .txt em memória
Envia:	
Assume:	Existe um texto selecionado, mesmo que seja um espaço em branco
Resultado:	É colocada a marca de âncora delimitando o texto que estava selecionado e poderá ser criada uma âncora para esse texto

Operação:	b_Criar_âncoras
Descrição:	Cria âncoras no texto de um nó terminal em edição
Lê:	
Modifica:	Lista de âncoras do nó em edição na hiperbase em memória
Envia:	Agente Externo:{#msg_âncoras_criadas}
Assume:	Existem âncoras marcadas no texto
Resultado:	Se existem âncoras marcadas no texto então as âncoras serão criadas e a mensagem de âncoras criadas será enviada Se não existem âncoras marcadas a mensagem é enviada mesmo assim, porém nenhuma âncora é criada

Operação:	b_Criar_Elo
Descrição:	Abre uma janela para a criação de um elo (link) entre dois nós da hiperbase em uso. Para que os nós (origem e destino) sejam efetivamente selecionados é necessário clicar com o botão direito do mouse sobre a área com os nós selecionados
Lê:	nó_terminal_origem; âncora_origem; nó_destino; âncora_destino; Apresenta_nó_destino_juntamente_com_o_nó_origem; Apresentar_os_nós_terminais_pertencentes_ao_nó_destino; botão Ok; botão Cancelar
Modifica:	Lista de Elos na hiperbase em memória
Envia:	Agente Externo:{#msg_erro_ao_criar_o_elo} Agente Externo:{ #msg_o_elo_foi_criado_no_contexto...}
Assume:	Existe pelo menos um nó terminal com âncoras criadas Existe pelo menos um outro nó, seja de contexto ou terminal O elo não é criado no contexto geral (HBP)
Resultado:	Se botão Cancelar então a operação é cancelada Se botão Ok então Se âncoras origem ou destino não foram selecionadas, ou nó origem igual ao nó destino então mensagem de erro ao criar o elo é enviada e a operação é cancelada Se o elo é criado então a mensagem de que o elo foi criado no contexto atual é enviada

Operação:	b_Criar_hiperbase
Descrição:	Cria uma nova hiperbase vazia
Lê:	
Modifica:	
Envia:	
Assume:	
Resultado:	Se existe uma hiperbase aberta, então é criada uma nova área para edição de hiperbase e a hiperbase que estava aberta é fechada Alterações não salvas na hiperbase que estava aberta estarão perdidas

Operação:	b_Criar_nó
Descrição:	Cria um nó terminal ou de contexto no contexto atual. O default é a criação de um nó de contexto e apenas os atributos para a criação de um nó de contexto estão habilitados. No caso da criação de um nó terminal outros atributos serão habilitados na interface
Lê:	Nome; Tipo do nó (terminal ou contexto); Tipo do nó terminal; Arquivo; Função_didática; Dificuldade; Palavras_chave; botão Ok; botão Cancelar
Modifica:	Hiperbase em memória
Envia:	Agente Externo:{#msg_já_existe_nó_com_este_nome} Agente Externo:{#msg_nome_inválido_para_o_nó}
Assume:	
Resultado:	Se b_Cancelar então a operação é cancelada Se b_Ok então Se já existe um nó com o nome especificado então é enviada a mensagem de nome já existente e a operação é cancelada Se o nome dado ao nó é inválido (vazio) então é enviada a mensagem de nome inválido e a operação fica aguardando novo nome

Operação:	b_Criar_Roteiro (b_Roteiro)
Descrição:	Cria efetivamente o roteiro editado
Lê:	
Modifica:	New nó_trilha na hiperbase em memória
Envia:	
Assume:	Existe pelo menos um nó no roteiro
Resultado:	Se botão Criar_Roteiro, então o nó de roteiro é criado no contexto da hiperbase Se já existe um nó de roteiro, então será substituído pelo novo nó criado

Operação:	b_Editar_Conteúdo (Participante de Editar_nó para nó terminal)
Descrição:	Abre o arquivo componente do nó terminal para edição. Esta operação só é válida para arquivos .txt e .bmp
Lê:	
Modifica:	Arquivo.txt; arquivo.bmp
Envia:	Agente Externo: {#msg_âncoras_criadas}
Assume:	O nó terminal é do tipo texto (.txt) ou imagem (.bmp)
Resultado:	Se botão Editar_conteúdo então Se o nó é do tipo texto então é aberta uma janela para a edição do conteúdo do nó Se não existe um arquivo .txt especificado é aberta uma janela para abertura de arquivo e depois é retomada à edição do mesmo Se for feita qualquer edição no texto e houver âncoras especificadas, ao fechar a janela de edição a mensagem de âncoras criadas será enviada e retornará a janela de edição dos atributos do nó Se o nó é do tipo imagem então o arquivo .bmp será aberto no Paint do Windows e após a edição retornará a janela de edição de atributos do nó

Operação:	b_Editar_nó_T
Descrição:	Abre uma janela para edição dos atributos do nó selecionado. Se o nó selecionado for terminal, serão habilitados para a edição atributos de nó terminal
Lê:	Nome; Arquivo; b_ler_arquivo; Função_didática; Dificuldade; Palavras_chave; b_editar_conteúdo; botão Ok; botão Cancelar
Modifica:	Nó na hiperbase em memória
Envia:	Agente Externo: {#msg_nome_inválido_para_o_nó}
Assume:	Existe um nó selecionado, de contexto ou terminal
Resultado:	Se botão Cancelar então a operação será cancelada Se botão Ok então Se nome do nó for apagado a mensagem de nome inválido para o nó será enviada e a janela de edição permanece, tanto para nó terminal como de contexto

Operação:	b_Editar_nó_C
Descrição:	Abre uma janela para edição dos atributos do nó selecionado. Se o nó selecionado for de contexto, serão habilitados para edição atributos de nó de contexto
Lê:	Assunto; botão Ok; botão Cancelar
Modifica:	Nó na hiperbase em memória
Envia:	Agente Externo:{#msg_nome_inválido_para_o_nó}
Assume:	Existe um nó de contexto selecionado
Resultado:	Se botão Cancelar então a operação será cancelada Se botão Ok então Se nome do nó for apagado a mensagem de nome inválido para o nó será enviada e a janela de edição permanece

Operação:	b_Eliminar_elo
Descrição:	Apaga da hiperbase o elo selecionado
Lê:	
Modifica:	Lista de Elos na hiperbase em memória
Envia:	Agente Externo:{#msg_Elimina_o_elo_selecionado}
Assume:	Existe um elo selecionado
Resultado:	Se Eliminar_elo então a mensagem de eliminação de elo é enviada Se botão Sim então o elo é eliminado da hiperbase Se botão Não então a operação é cancelada

Operação:	b_Eliminar_Nó_da_Base
Descrição:	Elimina o nó da hiperbase. Se houver mais de um nó com o mesmo nome, então todos serão eliminados
Lê:	
Modifica:	Hiperbase em memória
Envia:	Agente Externo:{#msg_elimina_nó_selecionado_da_hiperbase...}
Assume:	Existe um nó selecionado
Resultado:	Se botão Eliminar nó da base então a mensagem de eliminação é enviada Se botão Sim, então o nó é eliminado Se botão Não, então a operação é cancelada

Operação:	b_Está_difícil (Controle de Navegação)
Descrição:	Na navegação de uma hiperbase, o botão Está_difícil buscará por um nó que esteja no roteiro e no mesmo contexto do nó atual que tenha o atributo dificuldade inferior ao do nó atual. Caso não encontre, buscará por um nó cuja a função didática seja anterior a do nó atual
Lê:	
Modifica:	
Envia:	Agente Externo:{#msg não há nada mais fácil! Retorne aos tópicos anteriores}
Assume:	Os nós da hiperbase estão qualificados através dos atributos dificuldade e função didática
Resultado:	Se botão Está_difícil então será aberto um nó dentro do mesmo contexto que tenha o atributo dificuldade inferior ao do nó atual Se não houver tal nó, será aberto um nó cuja função didática seja anterior a do nó atual Se não houver um nó com função didática anterior então a mensagem de que não há nada mais fácil será enviada e o sistema retorna ao estado anterior a operação

Operação:	b_Está_fácil (Controle de Navegação)
Descrição:	Na navegação de uma hiperbase, o botão Está_fácil buscará por um nó que esteja no roteiro e no mesmo contexto que tenha o atributo dificuldade superior ao do nó atual. Caso não encontre, buscará por um nó cuja a função didática seja posterior a atual
Lê:	
Modifica:	
Envia:	Agente Externo:{#msg não há nada mais difícil! Avance aos tópicos seguintes}
Assume:	Os nós da hiperbase estão qualificados através dos atributos dificuldade e função didática
Resultado:	Se botão Está_fácil então será aberto um nó dentro do mesmo contexto que tenha o atributo dificuldade superior Se não houver tal nó, será aberto um nó cuja função didática seja posterior ao do nó atual Se também não houver um nó com função didática posterior então a mensagem de que não há nada mais difícil será enviada e o sistema retornará ao estado anterior a operação

Operação:	b_Exercícios (Controle de Navegação)
Descrição:	Abre um nó dentro do roteiro navegado cujo atributo função didática seja exercício. A busca pelo nó exercício se dá no menor contexto que contém o nó atual; caso não encontre, a busca é feita até o nó liberdade definido para o nó atual
Lê:	
Modifica:	
Envia:	Agente Externo:{#msg não existe nenhum nó exercícios}
Assume:	Os nós da hiperbase estão qualificados pelo menos através do atributo função didática
Resultado:	Se existir no roteiro um nó cuja função didática seja exercícios então será aberto Se não existir tal nó, então a mensagem de que não existe um nó de exercícios será enviada e a operação será cancelada

Operação:	b_Histórico (Controle de Navegação)
Descrição:	Abre uma lista de nós situando o nó atual em relação aos nós já navegados. Os nós podem ser navegados a partir dessa lista
Lê:	
Modifica:	
Envia:	
Assume:	
Resultado:	A janela de Histórico passa a ser a janela ativa. Caso um nó seja navegado a partir dessa janela ela permanecerá aberta até que seja explicitamente fechada

Operação:	b Ler arquivo (Participante de Criar nó)
Descrição:	Abre uma janela de abertura de arquivo para a escolha de um arquivo para compor o nó terminal
Lê:	Nome_do_arquivo; Pastas; Unidades_de_disco; Listar_arquivos_do_tipo; Som._Leitura; b_Rede; b_Ok; b_Cancelar
Modifica:	
Envia:	Agente Externo: {#msg_este_nome_de_arquivo_tem_muitas_letras}
Assume:	
Resultado:	Se b_Cancelar então a operação é cancelada e retorna a janela de Criação de nó terminal Se b_Ok então Se nome do arquivo estiver correto retorna a janela de Criação de nó terminal Se nome do arquivo tiver mais de 8 letras então a mensagem de nome de arquivo com muitas letras é enviada e permanece na janela de abertura de arquivo

Operação:	b Mais informações (Controle de Navegação)
Descrição:	Dispara uma busca por uma lista, em todo hiperdocumento, de nós que possuem palavras-chave em comum com o nó atual, sendo que qualquer um dos nós selecionados poderão ser visualizados pelo leitor. Porém não está funcionando para os primeiros nós abertos na inicialização da hiperbase
Lê:	
Modifica:	
Envia:	
Assume:	Os nós da hiperbase estão qualificados pelo menos através do atributo palavras-chave
Resultado:	Se não existirem nós que contenham palavras-chave iguais as do nó atual então a operação será cancelada Se existirem nós que contenham palavras-chave do nó atual então será aberta uma nova janela com a lista de nós encontrados

Operação:	b Onde estou? (Controle de Navegação)
Descrição:	Situa, graficamente, o nó atual no roteiro que está sendo percorrido, isto é, mostra a posição do nó atual em relação ao roteiro programado pelo professor
Lê:	
Modifica:	
Envia:	
Assume:	
Resultado:	É aberta uma janela gráfica mostrando os nós do roteiro programado e a posição atual do estudante

Operação:	b Rede
Descrição:	Botão que aparece em janelas de abertura e salvamento de arquivos que permite direcionar um caminho de rede
Lê:	Letra_da_unidade; Caminho; Reconectar_ao_iniciar; botão Ok; botão Cancelar
Modifica:	
Envia:	Agente Externo: {#msg de erro na conexão}
Assume:	
Resultado:	Se botão Ok, então Se o caminho especificado existe, então o caminho de rede é disponibilizado. Senão é enviada mensagem de erro na conexão e permanece esperando um caminho válido Se botão Cancelar então a operação é cancelada

Operação:	b_Retirar_Nó_do_Contexto
Descrição:	Retira o nó do contexto atual, ou seja, o nó permanece no arquivo da hiperbase, porém não aparece mais no contexto do qual foi retirado. Cópias existentes em outros contextos não são alteradas
Lê:	
Modifica:	Hiperbase em memória
Envia:	Agente Externo: {#msg_retira_nó_selecionado_do_contexto}
Assume:	Existe um nó selecionado
Resultado:	Se botão Retirar Nó do Contexto então a mensagem de retirada é enviada Se botão Sim, então o nó é retirado do contexto Se botão Não, então a operação é cancelada

Operação:	b_Retrocede_histórico (Controle de Navegação)
Descrição:	Retrocede na trilha de nós percorrida historicamente pelo leitor
Lê:	
Modifica:	
Envia:	
Assume:	
Resultado:	O nó anterior na trilha de nós (histórico) é aberto e o atual é fechado

Operação:	b_Retrocede_tópico (Controle de Navegação)
Descrição:	Retrocede no roteiro de navegação
Lê:	
Modifica:	
Envia:	
Assume:	
Resultado:	Se botão Retrocede Tópico então o nó terminal ou de contexto anterior do roteiro é aberto e o atual é fechado

Operação:	b_Roteiro
Descrição:	Abre a janela para criação de um nó roteiro na hiperbase em uso. Essa janela é composta de três módulos: consulta, que é feita de acordo com os atributos definidos, a lista de nós selecionados pela consulta e visualização gráfica do roteiro. É permitido apenas um nó de roteiro para cada hiperbase
Lê:	Assunto(nome do contexto); Função_Didática; Dificuldade; Palavras_Chave; botão Selecionar_nós; Nós inseridos no roteiro; Nome_do_roteiro; botão Criar_roteiro
Modifica:	
Envia:	
Assume:	
Resultado:	A janela de criação de roteiro passa a ser a janela ativa até que o roteiro seja criado (b_Criar_Roteiro) ou a janela seja fechada (Fechar)

Operação:	b_Salvar_hiperbase
Descrição:	Salva a hiperbase construída em arquivo
Lê:	Nome; diretório; drive; somente leitura; salvar como tipo; botão ok; botão cancelar.
Modifica:	New hiperbase.hip; new hiperbase.hi~
Envia:	Agente externo:{#msg_existente_sobrepôr} Agente externo:{#msg_nome com muitas letras}
Assume:	
Resultado:	Se botão cancelar, a operação é cancelada Se botão ok Se arquivo não existente, salva a hiperbase Se arquivo existente, a mensagem arquivo_existente_sobrepôr é enviada Se botão sim é escolhido, o arquivo é sobreposto e uma cópia com a versão anterior é criada no .hi~ Se botão não é escolhido, a operação retorna ao início Se nome do arquivo tiver mais de 8 letras então a mensagem de nome com muitas letras é enviada e a operação retorna ao início

Operação:	b_Selecionar_Nós (b_Roteiro)
Descrição:	Mostra na área de Nós Selecionados os nós pertencentes ao contexto que está em Assunto, para que possam ser escolhidos os nós que farão parte do roteiro
Lê:	
Modifica:	
Envia:	
Assume:	Existe um contexto em Assunto
Resultado:	Os nós pertencentes ao contexto referenciado em Assuntos serão apresentados na área de Nós Selecionados

Operação:	b_Visualizar_conteúdo_do_nó
Descrição:	Apresenta o conteúdo do nó terminal selecionado em uma nova janela
Lê:	
Modifica:	
Envia:	Agente Externo:{#msg_arquivo_inválido} Agente Externo:{#msg_não_existe_o_arquivo_especificado}
Assume:	Existe um nó terminal selecionado
Resultado:	Se houver um arquivo válido especificado para o nó seu conteúdo será apresentado em uma janela e suas âncoras poderão ser navegadas Se não houver um arquivo especificado para o nó a mensagem de arquivo não especificado é enviada e a operação é cancelada Se o arquivo especificado for inválido a mensagem de arquivo inválido é enviada e a operação é cancelada

Operação:	b_Visualizar_contexto_selecionado
Descrição:	Mostra o conteúdo do contexto selecionado dentro da janela Nós e Elos
Lê:	
Modifica:	
Envia:	
Assume:	Existe um nó de contexto selecionado
Resultado:	O contexto atual passa a ser o mostrado na janela Nós e Elos

Operação:	b_Voltar_nó (Participante de Browser Gráfico)
Descrição:	Retorna ao estado anterior a execução da operação Explorar_nó
Lê:	
Modifica:	
Envia:	
Assume:	
Resultado:	Se no browser estiver sendo apresentado um contexto, retorna a apresentação do contexto anterior Se estiver sendo apresentado o conteúdo de um nó retorna a apresentação do contexto do nó em apresentação

Operação:	b_Voltar_para_o_contexto_anterior
Descrição:	Mostra o conteúdo do contexto anterior (pai do contexto atual) na janela Nós e Elos
Lê:	
Modifica:	
Envia:	
Assume:	Existe um contexto anterior, ou seja, a operação Visualizar_Contexto_Selecionado foi realizada pelo menos uma vez
Resultado:	O contexto anterior (pai do contexto atual) passa a ser o contexto atual e a ser mostrado na janela Nós e Elos

Operação:	b_Zoom_in (Participante de Browser Gráfico)
Descrição:	Retoma o tamanho normal dos ícones dos nós apresentados no browser gráfico
Lê:	
Modifica:	
Envia:	
Assume:	A operação b_Zoom_Out foi realizada anteriormente
Resultado:	Os ícones deixam de ser representados por quadrados numerados e voltam a ser apresentados como desenhos

Operação:	b_Zoom_out (Participante de Browser Gráfico)
Descrição:	Diminui o tamanho dos ícones dos nós apresentados no browser gráfico
Lê:	
Modifica:	
Envia:	
Assume:	
Resultado:	Os ícones deixam de ser apresentados como desenhos e passam a ser representados por quadrados numerados

Operação:	Cascata
Descrição:	Organiza as janelas abertas em cascata
Lê:	
Modifica:	
Envia:	
Assume:	
Resultado:	Janelas abertas redimensionadas e organizadas em cascata

Operação:	Copiar
Descrição:	Copia um nó de contexto ou terminal de um contexto para outro. Esta operação não aparece explicitamente na interface. Está disponível através do arrasto do mouse juntamente com a tecla CTRL pressionada. De fato, o nó continua tendo apenas uma instância física, mas passa a ser referenciado em outro contexto também
Lê:	
Modifica:	O arquivo da hiperbase na memória
Envia:	
Assume:	Existem pelo menos dois contextos abertos simultaneamente
Resultado:	O mesmo nó aparecerá em mais de um contexto

Operação:	Criar
Descrição:	Cria uma nova hiperbase vazia
Lê:	
Modifica:	
Envia:	
Assume:	
Resultado:	Se existe uma hiperbase aberta, então é criada uma nova área para edição de hiperbase e a hiperbase que estava aberta é fechada Alterações não salvas na hiperbase que estava aberta estarão perdidas

Operação:	Editar_Nó_de_Ajuda (Participante de b_Editar_Conteúdo)
Descrição:	Esta operação inicia-se no botão Editar Conteúdo para nós de função didática Ajuda. Esta operação não está funcionando corretamente
Lê:	Dúvidas; Respostas; botão inserir; botão gravar
Modifica:	New arquivo.txt; New arquivo.res
Envia:	
Assume:	
Resultado:	Se botão inserir, então é inserida uma linha separando a pergunta da resposta Se botão gravar, então as perguntas são gravadas no arquivo .txt e as respostas são gravadas no arquivo .res

Operação:	Estudante (Entrar no módulo estudante)
Descrição:	Para entrar no módulo estudante é preciso abrir uma das hiperbases que estejam disponíveis
Lê:	Nome_do_estudante; Diretório; Hiperbase; botão Ok; botão Cancelar
Modifica:	
Envia:	Agente Externo: {#msg_não_existe_o_arquivo_especificado}
Assume:	
Resultado:	Se botão Cancelar então o SASHE é finalizado Se botão Ok então Se o arquivo especificado como nó terminal da hiperbase escolhida não for encontrado então a mensagem de arquivo não existente é enviada e o módulo estudante do SASHE é inicializado sem nenhuma hiperbase aberta

Operação:	Explorar_nó (Participante do Browser Gráfico)
Descrição:	Esta operação está disponível através do clique com o botão direito do mouse sobre um nó apresentado no browser gráfico. Se for um nó de contexto, passarão a serem mostrados no browser os nós pertencentes aquele contexto. Se for um nó terminal, o conteúdo do nó será apresentado
Lê:	
Modifica:	
Envia:	Agente Externo: {#msg_erro:_arquivo_não_encontrado}
Assume:	Pelo menos um nó está sendo apresentado no browser
Resultado:	Se o nó clicado for um nó de contexto, o novo contexto será aberto no browser gráfico Se o nó clicado for um nó terminal, seu conteúdo será apresentado Se o arquivo referente ao nó terminal não existir então a mensagem de arquivo não encontrado será enviada

Operação:	Fechar
Descrição:	Fecha a janela atual. É o "x" do canto direito da janela, padrão seguido pelas janelas do Windows
Lê:	
Modifica:	
Envia:	
Assume:	
Resultado:	Se Fechar então a janela atual é fechada e retorna a janela Nós e Elos

Operação:	Inserir_no_roteiro (Participante de b_Roteiro)
Descrição:	Esta operação insere o nó selecionado no roteiro em edição. Torna-se disponível na interface através do clique com o botão direito do mouse sobre a área de nós selecionados
Lê:	Nó_liberdade selecionado; botão Ok
Modifica:	
Envia:	Agente Externo: {#msg_inserindo_um_nó_de_contexto...}
Assume:	
Resultado:	Se inserir_no_roteiro então selecionar nó liberdade Se botão Ok então mensagem de inserir nó no roteiro é enviada Se botão Sim, então o nó é inserido no roteiro e continua edição Se botão Não, então o nó não é inserido e continua edição

Operação:	Lado_a_lado
Descrição:	Organiza as janelas abertas lado a lado
Lê:	
Modifica:	
Envia:	
Assume:	
Resultado:	Janelas abertas redimensionadas e organizadas lado a lado

Operação:	Limpar_roteiro (participante b_roteiro)
Descrição:	Limpa o roteiro, eliminando do roteiro todos os nós que pertenciam a ele. Esta operação torna-se disponível na interface através do clique com o botão direito do mouse sobre a área Nós Pertencentes a Trilha
Lê:	
Modifica:	Nó_trilha na hiperbase em memória
Envia:	
Assume:	Existe um roteiro
Resultado:	Se limpar_roteiro, então o roteiro estará vazio e pronto para edição do novo roteiro Se limpar_roteiro e fechar janela de edição de roteiro, então o roteiro permanecerá como estava antes da execução da operação

Operação:	Mover
Descrição:	Move um nó de contexto ou terminal de um contexto para outro. Esta operação não aparece explicitamente na interface. Está disponível através do arrasto do mouse
Lê:	
Modifica:	O arquivo da hiperbase em memória
Envia:	
Assume:	Existem pelo menos dois contextos abertos simultaneamente
Resultado:	O nó deixa de existir em um contexto e passa a existir em outro

Operação:	Mudar_nó_liberdade (Participante do b_Roteiro)
Descrição:	Abre a janela para a mudança do nó liberdade do nó selecionado no roteiro. Esta operação está disponível através do clique com o botão direito do mouse sobre a área Nós Pertencentes a Trilha
Lê:	botão Ok
Modifica:	Nó Trilha da hiperbase em memória
Envia:	
Assume:	Existe um nó participante do roteiro selecionado
Resultado:	Se Mudar_nó_liberdade então abre janela para a mudança do nó liberdade Se botão Ok, nó liberdade é mudado

Operação:	Navegar
Descrição:	É a navegação efetiva através das âncoras do hiperdocumento. Isto é feito através do clique sobre a âncora
Lê:	
Modifica:	
Envia:	Agente Externo:{#msg esta âncora não está disponível}
Assume:	Existem âncoras a serem navegadas
Resultado:	Se âncora disponível então a navegação para o documento especificado é realizada abrindo uma nova janela com o novo documento Se âncora não disponível então a mensagem de âncora não disponível é enviada e a operação é cancelada

Operação:	Salvar
Descrição:	Salva a hiperbase construída em arquivo e está disponível em menu, mas nunca é habilitada na versão atual
Lê:	
Modifica:	
Envia:	
Assume:	
Resultado:	

Operação:	Saída
Descrição:	Finaliza o SASHE antes de entrar em um dos módulos disponíveis, Professor ou Estudante
Lê:	
Modifica:	
Envia:	
Assume:	
Resultado:	O SASHE é finalizado

Operação:	Sair
Descrição:	Finaliza o SASHE, tanto no módulo professor como estudante
Lê:	
Modifica:	
Envia:	
Assume:	
Resultado:	Se módulo professor e hiperbase não estiver salva após a operação Sair a hiperbase será perdida e SASHE finalizado Se módulo estudante e Sair então SASHE finalizado

Operação:	Visualizar Browser Gráfico ou b Browser Gráfico
Descrição:	Abre a janela do browser gráfico com os nós presentes no contexto atual da janela Nós e Elos
Lê:	
Modifica:	
Envia:	
Assume:	
Resultado:	É inicializado o browser gráfico. Se o contexto estiver vazio, a janela será aberta assim mesmo, vazia. Senão, são mostrados graficamente os nós, tanto terminais como de contexto, pertencentes ao contexto em que a base estava quando foi invocado o browser.

3.3 Modelo de Objetos

Para a elaboração desse modelo no Fusion-RE/I primeiramente definem-se os assuntos relacionados com a funcionalidade do sistema. Esses assuntos são denominados *Temas* (Costa 97). Tendo definido os temas, é feito um agrupamento das operações de acordo com os temas a que se referem. Assim, ao final tem-se uma lista de temas e as operações dos temas. Finalmente, para cada um dos temas definidos, é construído um modelo de objetos.

Para que se pudesse definir os temas foi necessária uma análise das informações recuperadas na primeira etapa (Obtenção de Informações Existentes Sobre o Sistema) e também das abstrações vindas dos modelos de ciclo de vida e de operações. Essa é uma das tarefas mais subjetivas do método Fusion-RE/I e é de fundamental importância para a aplicação do método (Costa 1997). De fato, constatamos a subjetividade da definição dos temas, o que nos levou a várias tentativas de possíveis temas, antes de decidirmos quais seriam os temas definitivos do sistema.

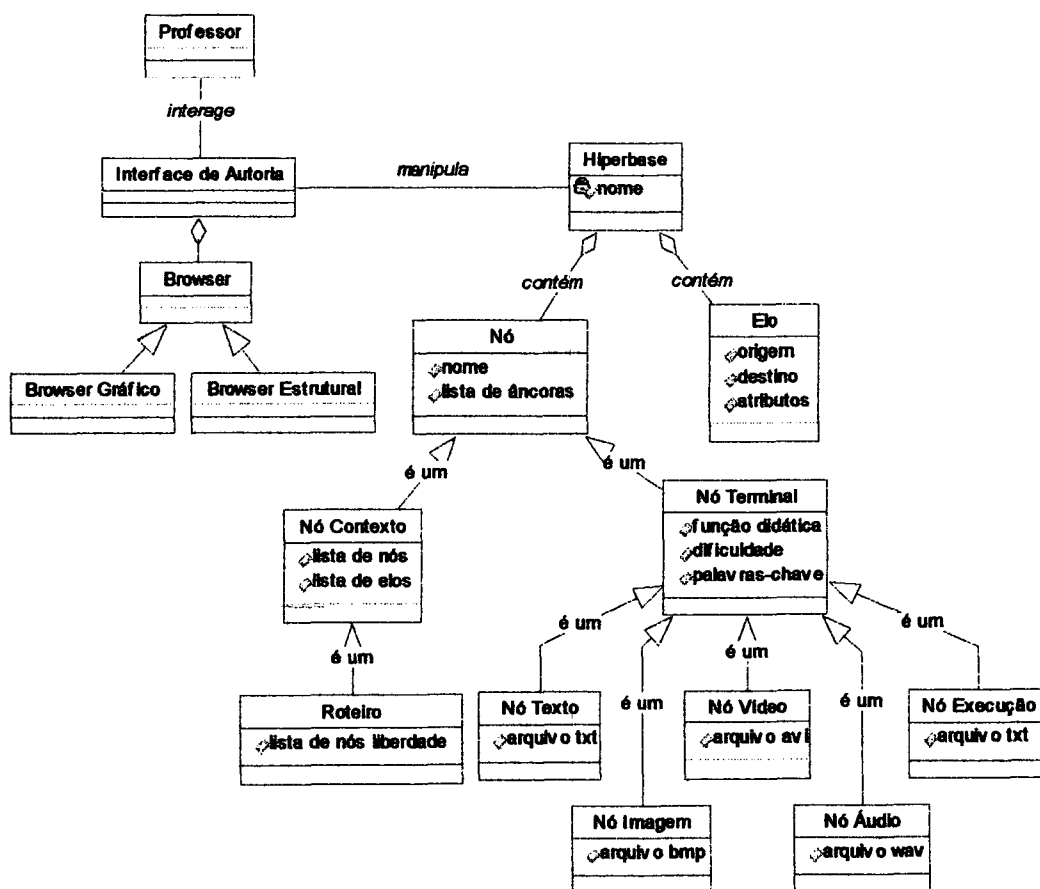


Figura 1. Modelo de objetos para o tema *Autoria*

Após uma primeira tentativa de fazer o agrupamento das operações identificadas nos temas propostos, percebeu-se que os temas definidos ainda não eram adequados, pois algumas operações não encaixavam-se em nenhum dos temas, e alguns temas não continham nenhuma operação. Além disso, os temas definidos não evoluíam para os modelos de objetos. Sendo assim, analisando cuidadosamente as operações e seus relacionamentos propomos dois temas para o sistema: *Autoria* e *Navegação*, o que nos pareceu adequado. Foi feito então o agrupamento de todas as operações identificadas no modelo de operações em um dos dois temas. Após esse agrupamento, as operações foram novamente analisadas, buscando identificar componentes que constituíssem o modelo de objetos, ou seja, classes, relacionamentos, atributos, e possíveis agregações, especializações e generalizações. Dessa forma foram construídos dois modelos de objetos, um para o tema *Autoria* (Figura 1) e um para o tema *Navegação* (Figura 2).

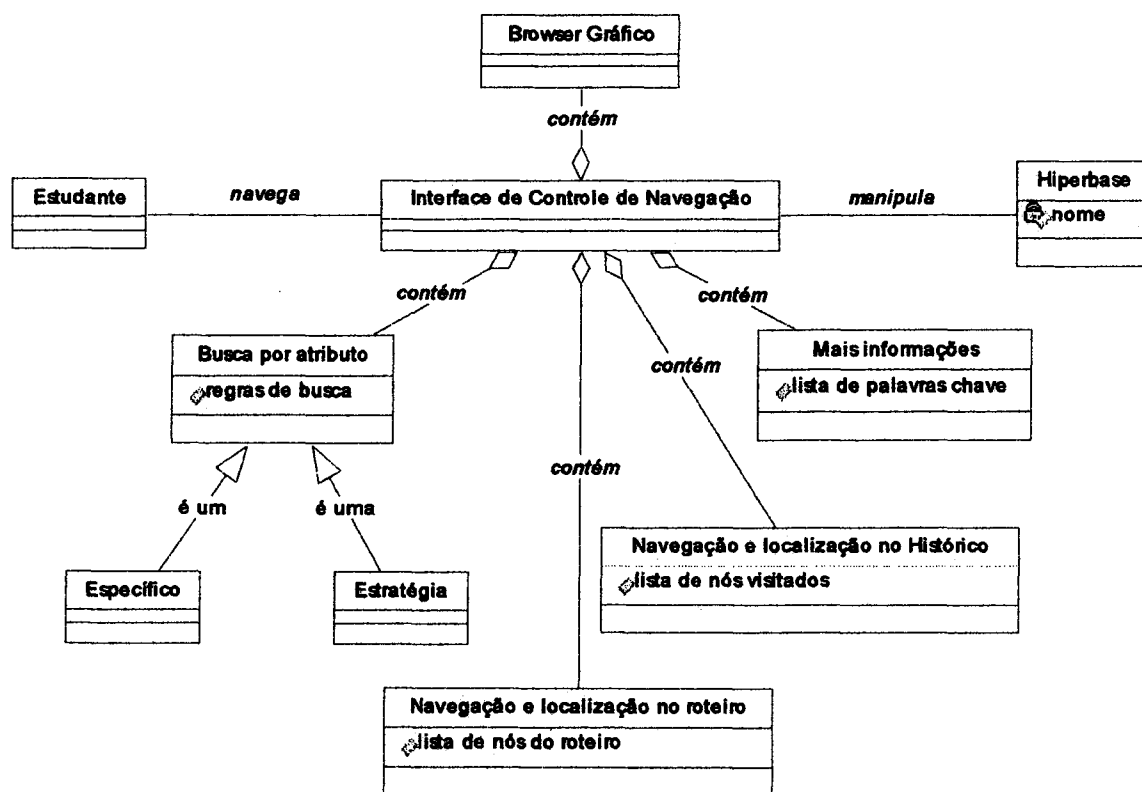


Figura 2. Modelo de objetos para o tema *Navegação*

É importante ressaltar que os modelos foram propostos baseados apenas na visão funcional do software, conseguida através da interação exaustiva com o software. Embora não conste no método Fusion-RE/I, é provável que após a recuperação das informações estruturais do software esses modelos de objetos sofram algumas alterações, para que se tornem consistentes com a estrutura do software, porém

espera-se que essas alterações sejam pequenas. Os modelos de objetos foram construídos na ferramenta Rational Rose (Rational 99), seguindo a notação UML.

Com o término da recuperação do modelo de ciclo de vida, do modelo de operações e dos modelos de objetos, encerra-se a primeira etapa do método Fusion-RE/I, que recupera visões funcionais do software unicamente através da interação com a interface do sistema. Na próxima etapa são recuperadas informações estruturais através da análise do código-fonte do software. Os documentos resultantes dessa nova etapa são apresentados nas próximas subseções.

3.4 Quadro de Chamadas

Esse quadro foi elaborado para cada arquivo de código fonte do sistema, apresentando os procedimentos contidos no arquivo, suas respectivas funcionalidades e os procedimentos utilizados (chamados) e utilizadores (chamado por).

Parte das informações de descrição (funcionalidade) dos procedimentos foram conseguidas a partir de comentários no código fonte. Os procedimentos chamados por cada procedimento foram obtidos pela análise do próprio código, enquanto que os procedimentos utilizadores (chamados por) foram obtidos através dos próprios quadros de chamadas elaborados e da análise do código fonte.

A seguir, são mostrados os quadros de chamadas elaborados para cada arquivo do sistema SASHE.

Sistema: SASHE Arquivo: Sashe\Hyperprop\anc_aud.cpp	
Procedimento: (15) ancora_audio::ancora_audio(int ind, int ide) : ancora(ind, ide) Diretório/Arquivo: anc_aud.cpp	Descrição: Construtor da ancora audio Chama: (15) ancora(ind, ide) Chamado por: No_audio.cpp\ (38) int no_audio::cria ancora_audio(int ide)
Procedimento: (22) ancora_audio::ancora_audio(int ind): ancora(ind) Diretório/Arquivo: anc_aud.cpp	Descrição: Construtor usado quando a ancora audio é recuperada do arquivo. Chama: (22) ancora(ind) Chamado por: List_anc.cpp\ (118) int lista_ancoras :: recupera(No *n, FILE *fp)

Sistema: SASHE Arquivo: Sashe\Hyperprop\anc_aud.h	
Procedimento: (22) int ancora_audio::eh_ancora_audio() Diretório/Arquivo: anc_aud.h	Descrição: Chama: Chamado por: List_anc.cpp\ (29) lista_ancoras::~lista_ancoras()


```
List_anc.cpp\ (71) int lista_ancoras ::
grava(FILE *fp)
```

Sistema: SASHE	
Arquivo: Sashe\Hyperprop\anc_cont.cpp	
Procedimento: (15) ancora_contexto::ancora_contexto(in t ind,int ide,int id_no) : ancora(ind, ide) Diretório/Arquivo: anc_cont.cpp	Descrição: Construtor da ancora contexto Chama: (15) ancora(ind, ide) Chamado por: contexto.cpp\ (43) int contexto::cria_ancora_lambda(int ide) contexto.cpp\ (73) ancora_contexto *contexto::cria_ancora_contexto(int ide, int id no)
Procedimento: (23) ancora_contexto :: ancora_contexto(int ind) : ancora(ind) Diretório/Arquivo: anc_cont.cpp	Descrição: Construtor usado quando a ancora é recuperada de arquivo. Chama: (23) ancora(ind) Chamado por: contvers.cpp\ (54) void contextoversoes::CriaAncoraCV(No *n) list_anc.cpp\ (118) int lista_ancoras :: recupera(No *n, FILE *fp)
Procedimento: (30) void ancora_contexto::muda_conteudo(int ide, int id_no) Diretório/Arquivo: anc_cont.cpp	Descrição: muda o conteudo de uma ancora Chama: (32) troca_ident_externo(ide) Chamado por: Contexto.cpp\ (102) void contexto::muda_ancora_contexto(ancora *p, int ide, int id no)
Procedimento: (38) int ancora_contexto::grava_ancora(FILE *fp) Diretório/Arquivo: anc_cont.cpp	Descrição: Grava uma ancora do tipo contexto em arquivo Chama: (40) ancora::grava_ancora(fp) Chamado por: List_anc.cpp\ (71) int lista_ancoras :: grava(FILE *fp)
Procedimento: (47) int ancora_contexto::recupera_ancora(FI LE *fp) Diretório/Arquivo: anc_cont.cpp	Descrição: Chama: (49) ancora::recupera_ancora(fp) (50) fscanf(fp, "\t\tâncora(%d)\n", &conteudo) Chamado por: List_anc.cpp\ (118) int lista_ancoras :: recupera(No *n, FILE *fp)

Sistema: SASHE**Arquivo:** Sashe\Hyperprop\anc_cont.h

Procedimento: (27) int	Descrição:
----------------------------------	-------------------

ancora_contexto::eh_ancora_contexto () Diretório/Arquivo: anc_cont.h	
	Chama: Chamado por: List_anc.cpp\ (29) lista_ancoras::~lista_ancoras() List_anc.cpp\ (71) int lista_ancoras :: grava(FILE *fp)
Procedimento: (26) int ancora_contexto::obtem_ancora() Diretório/Arquivo: anc_cont.h	Descrição: Chama: Chamado por: Contexto.cpp\ (89) ancora_contexto *contexto::contem_ancora(int id_no) Hipger.cpp\ (1311) extern "C" int FAR PASCAL _export recupera_nome_ancora(char *nome, int id_ancora, char *nome_ancora) Hipger.cpp\ (1372) extern "C" int FAR PASCAL _export primeira_ancora(char *nome, char *nome_ancora) Hipger.cpp\ (1425) extern "C" int FAR PASCAL _export proxima_ancora(char *nome, char *nome_ancora) Trilha.cpp\ (251) int trilha::primeiro_no_trilha(char *nome, char *nome_liberdade) Trilha.cpp\ (278) int trilha::proximo_no_trilha(char *nome, char *nome_liberdade) Trilha.cpp\ (305) int trilha::anterior_no_trilha(char *nome, char *nome_liberdade) Trilha.cpp\ (332) int trilha::ultimo_no_trilha(char *nome, char *nome_liberdade)1

Sistema: SASHE Arquivo: Sashe\Hyperprop\anc_exec.cpp	
Procedimento: (15) ancora_executavel::ancora_executave l(int ind, int ide) : ancora(ind, ide) Diretório/Arquivo: anc_exec.cpp	Descrição: Construtor da ancora executavel Chama: (15) ancora(ind, ide) Chamado por: No_exec.cpp\ (38) int no_executavel::cria_ancora_executavel(in t ide)
Procedimento: (22) ancora_executavel::ancora_executave l(int ind): ancora(ind) Diretório/Arquivo: anc_exec.cpp	Descrição: Construtor usado quando a ancora executavel é recuperada do arquivo. Chama: (22) ancora(ind) Chamado por:

Sistema: SASHE Arquivo: Sashe\Hyperprop\anc_exec.h	
Procedimento: (22) int ancora_executavel::eh_ancora_execut avel() Diretório/Arquivo: anc_exec.h	Descrição: Chama: Chamado por:

Sistema: SASHE	
Arquivo: Sashe\Hyperprop\anc_ima.cpp	
Procedimento: (15) ancora_imagem::ancora_imagem(int ind, int ide) : ancora(ind, ide) Diretório/Arquivo: anc_ima.cpp	Descrição: Construtor da ancora imagem Chama: (15) ancora(ind, ide) Chamado por: No_image.cpp\ (39) int no_imagem::cria_ancora_imagem(int ide)
Procedimento: (22) ancora_imagem::ancora_imagem(int ind) : ancora(ind) Diretório/Arquivo: anc_ima.cpp	Descrição: Construtor usado quando a ancora imagem é recuperada do arquivo. Chama: (22) ancora(ind) Chamado por: List_anc.pp\ (118) int lista_ancoras :: recupera(No *n, FILE *fp)

Sistema: SASHE	
Arquivo: Sashe\Hyperprop\anc_ima.h	
Procedimento: (22) int ancora_imagem::eh_ancora_imagem() Diretório/Arquivo: anc_ima.h	Descrição: Chama: Chamado por: List_anc.cpp\ (29) lista_ancoras::~~lista_ancoras() List_anc.cpp\ (71) int lista_ancoras :: grava(FILE *fp)

Sistema: SASHE	
Arquivo: Sashe\Hyperprop\anc_tab.cpp	
Procedimento: (15) ancora_tabela::ancora_tabela(int ind, int ide) : ancora(ind, ide) Diretório/Arquivo: anc_tab.cpp	Descrição: Construtor da ancora tabela Chama: (15) ancora(ind, ide) Chamado por: No_tab.cpp\ (38) int no_tabela::cria_ancora_tabela(int ide)
Procedimento: (22) ancora_tabela::ancora_tabela(int ind) : ancora(ind) Diretório/Arquivo: anc_tab.cpp	Descrição: Construtor usado quando a ancora tabela é recuperada do arquivo. Chama: (22) ancora(ind) Chamado por:

Sistema: SASHE	
Arquivo: Sashe\Hyperprop\anc_tab.h	
Procedimento: (22) int ancora_tabela::eh_ancora_tabela() Diretório/Arquivo: anc_tab.h	Descrição: Chama: Chamado por:

Sistema: SASHE	
Arquivo: Sashe\Hyperprop\anc_tex.cpp	
Procedimento: (16) ancora_texto::ancora_texto(int	Descrição: cria uma instancia de uma ancora texto

ind, int ide, char *pal, int inicio, int fim) : ancora(ind, ide) Diretório/Arquivo: anc_tex.cpp	Chama: Chamado por: No_texto.cpp\ (42) int no_texto::cria_ancora_texto(int ide, char *pal, int inicio, int fim)
Procedimento: (28) ancora_texto::ancora_texto(int ind): ancora(ind) Diretório/Arquivo: anc_tex.cpp	Descrição: Construtor quando a ancora eh lida do arquivo Chama: (28) ancora(ind) Chamado por: List_anc.cpp\ (118) int lista_ancoras :: recupera(No *n, FILE *fp)
Procedimento: (38) ancora_texto::~~ancora_texto() Diretório/Arquivo: anc_tex.cpp	Descrição: Destrutor da classe ancora Chama: Chamado por:
Procedimento: (45) void ancora_texto::muda_conteudo(int ide, char *pal, int inicio, int fim) Diretório/Arquivo: anc_tex.cpp	Descrição: Chama: (47) troca_ident_externo(ide) Chamado por: No_texto.cpp\ (51) void no_texto::muda_ancora_texto(ancora *p, int ide, char *pal, int inicio, int fim)
Procedimento: (61) int ancora_texto::grava_ancora(FILE *fp) Diretório/Arquivo: anc_tex.cpp	Descrição: Grava uma ancora texto em arquivo Chama: (63) ancora::grava_ancora(fp) Chamado por: List_anc.cpp\ (71) int lista_ancoras :: grava(FILE *fp)
Procedimento: (71) int ancora_texto::recupera_ancora(FILE *fp) Diretório/Arquivo: anc_tex.cpp	Descrição: Chama: (75) ancora::recupera_ancora(fp) (77) fscanf(fp, "\t\tÃncora(%s)\n", string) (82) fscanf(fp, "\t\tPosição(%d,", &Inicio) (83) fscanf(fp, "%d)\n", &Fim) Chamado por: List_anc.cpp\ (118) int lista_ancoras :: recupera(No *n, FILE *fp)

Sistema: SASHE**Arquivo: Sashe\Hyperprop\ anc_tex.h**

Procedimento: (30) int ancora_texto::obtem_inicio_ancora() Diretório/Arquivo: anc_tex.h	Descrição: retorna o endereco inicial da palavra ancorada Chama: Chamado por: No_texto.cpp\ (60) void no_texto::copia_ancoras_texto(lista_ancoras *ancoras)
Procedimento: (32) int ancora_texto::obtem_fim_ancora()	Descrição: correspondente ao inicio e fim de uma ancora

Diretório/Arquivo: anc_tex.h	Chama: Chamado por: No_texto.cpp\ (60) void no_texto::copia_ancoras_texto(lista_ancoras *ancoras)
Procedimento: (34) char ancora_texto::*obtem_ancora() Diretório/Arquivo: anc_tex.h	Descrição: retorna a palavra onde estah a ancora Chama: Chamado por: Hipger.cpp\ (1311) extern "C" int FAR PASCAL _export recupera_nome_ancora(char *nome, int id_ancora, char *nome_ancora) Hipger.cpp\ (1372) extern "C" int FAR PASCAL _export primeira_ancora(char *nome, char *nome_ancora) Hipger.cpp\ (1425) extern "C" int FAR PASCAL _export proxima_ancora(char *nome, char *nome_ancora) Hipger.cpp\ (1489) extern "C" int FAR PASCAL _export ultima_ancora(char *nome, char *nome_ancora) No_texto.cpp\ (60) void no_texto::copia_ancoras_texto(lista_ancoras *ancoras)
Procedimento: (37) int ancora_texto::eh_ancora_texto() Diretório/Arquivo: anc_tex.h	Descrição: Chama: Chamado por: List_anc.cpp\ (29) lista_ancoras::~~lista_ancoras() List_anc.cpp\ (71) int lista_ancoras :: grava(FILE *fp)

Sistema: SASHE Arquivo: Sashe\Hyperprop\anc_tri.cpp	
Procedimento: (15) ancora_trilha::ancora_trilha(int ind,int ide) : ancora(ind, ide) Diretório/Arquivo: anc_tri.cpp	Descrição: Construtor da ancora trilha Chama: (15) ancora(ind, ide) Chamado por: Trilha.cpp\ (38) int trilha::cria_ancora_lambda(int ide)
Procedimento: (22) ancora_trilha :: ancora_trilha(int ind) : ancora(ind) Diretório/Arquivo: anc_tri.cpp	Descrição: Construtor usado quando a ancora trilha é recuperada do arquivo. Chama: (22) ancora(ind) Chamado por: List_anc.pp\ (118) int lista_ancoras :: recupera(No *n, FILE *fp)
Procedimento: (29) void ancora_trilha::muda_conteudo(int ide) Diretório/Arquivo: anc_tri.cpp	Descrição: Chama: (31)troca_identificador(ide) Chamado por:

Sistema: SASHE Arquivo: Sashe\Hyperprop\anc_tri.h	
Procedimento: (19) int	Descrição:

ancora_trilha::eh_ancora_trilha()
Diretório/Arquivo: anc_tri.h

Chama:
Chamado por:
List_anc.cpp\ (71) int lista_ancoras ::
grava(FILE *fp)

Sistema: SASHE

Arquivo: Sashe\Hyperprop\anc_vid.cpp

Procedimento:
(12) ancora_video::ancora_video(int
ind, int ide) : ancora(ind, ide)
Diretório/Arquivo: anc_vid.cpp

Descrição:
Construtor da ancora video

Chama:
(12) ancora(ind, ide)
Chamado por:
no_video.cpp\ (37) int
no_video::cria ancora video(int ide)

Procedimento:
(19) ancora_video::ancora_video(int
ind): ancora(ind)
Diretório/Arquivo: anc_vid.cpp

Descrição:
Construtor usado quando a ancora video é
recuperada do arquivo.

Chama:
(19) ancora(ind)
Chamado por:
List_anc.cpp\ (118) int lista_ancoras ::
recupera(No *n, FILE *fp)

Sistema: SASHE

Arquivo: Sashe\Hyperprop\anc_vid.h

Procedimento:
(19) int
ancora_video::eh_ancora_video()
Diretório/Arquivo: anc_vid.h

Descrição:

Chama:
Chamado por:
List_anc.cpp\ (29)
lista_ancoras::~~lista_ancoras()
List_anc.cpp\ (71) int lista_ancoras ::
grava(FILE *fp)

Sistema: SASHE

Arquivo: Sashe\Hyperprop\ancora.cpp

Procedimento:
(12) ancora::ancora(int Ind, int
ide): entidade(Ind)
Diretório/Arquivo: ancora.cpp

Descrição:
Construtor da ancora

Chama:
(12) entidade(Ind)
Chamado por:
Anc_aud.cpp\ (12)
ancora_audio::ancora_audio(int ind, int
ide) : ancora(ind, ide)
Anc_cont.cpp\ (12)
ancora_contexto::ancora_contexto(int
ind,int ide,int id_no) : ancora(ind,
ide)
Anc_exec.cpp\ (12)
ancora_executavel::ancora_executavel(int
ind, int ide) : ancora(ind, ide)
Anc_ima.cpp\ (12)
ancora_imagem::ancora_imagem(int ind,
int ide) : ancora(ind, ide)
Anc_tab.cpp\ (12)
ancora_tabela::ancora_tabela(int ind,
int ide) : ancora(ind, ide)
Anc_tex.cpp\ (14)
ancora_texto::ancora_texto(int ind, int
ide, char *pal, int inicio, int fim) :
ancora(ind, ide)
Anc_tri.cpp\ (12)
ancora_trilha::ancora_trilha(int ind,int

	<pre> ide) : ancora(ind, ide) Anc_vid.cpp\ (12) ancora_video::ancora_video(int ind, int ide) : ancora(ind, ide) </pre>
Procedimento: (20) ancora::ancora(int Ind): entidade(Ind) Diretório/Arquivo: ancora.cpp	Descrição: Construtor usado quando a ancora é recuperada do arquivo. Chama: (20) entidade(Ind) Chamado por: Anc_aud.cpp\ (19) ancora_audio::ancora_audio(int ind): ancora(ind) Anc_cont.cpp\ (20) ancora_contexto :: ancora_contexto(int ind) : ancora(ind) Anc_exec.cpp\ (19) ancora_executavel::ancora_executavel(int ind): ancora(ind) Anc_ima.cpp\ (19) ancora_imagem::ancora_imagem(int ind): ancora(ind) Anc_tab.cpp\ (19) ancora_tabela::ancora_tabela(int ind): ancora(ind) Anc_tex.cpp\ (24) ancora_texto::ancora_texto(int ind): ancora(ind) Anc_tri.cpp\ (19) ancora_trilha :: ancora_trilha(int ind) : ancora(ind) Anc_vid.cpp\ (19) ancora_video::ancora_video(int ind): ancora(ind)
Procedimento: (28) int ancora::grava_ancora(FILE *fp) Diretório/Arquivo: ancora.cpp	Descrição: Salva os atributos da âncora em arquivo. Chama: (30) entidade::grava(fp) Chamado por: anc_cont.cpp\ (38) int ancora_contexto::grava_ancora(FILE *fp) anc_text.cpp\ (61) int ancora_texto::grava_ancora(FILE *fp) list_anc.cpp\ (71) int lista_ancoras :: grava(FILE *fp)
Procedimento: (38) int ancora::recupera_ancora(FILE *fp) Diretório/Arquivo: ancora.cpp	Descrição: Recupera os atributos da âncora do arquivo. Chama: (40) fscanf(fp, "%d\n", &ident) Chamado por: anc_cont.cpp\ (47) int ancora_contexto::recupera_ancora(FILE *fp) anc_text.cpp\ (71) int ancora_texto::recupera_ancora(FILE *fp) list_anc.cpp\ (118) int lista_ancoras :: recupera(No *n, FILE *fp)
Sistema: SASHE Arquivo: Sashe\Hyperprop\ancora.h	
Procedimento: (21) virtual int ancora::eh_ancora_contexto() Diretório/Arquivo: ancora.h	Descrição: Chama: Chamado por: List_anc.cpp\ (29) lista_ancoras::~~lista_ancoras()

	List_anc.cpp\ (71) int lista_ancoras :: grava(FILE *fp)
Procedimento: (22) virtual int ancora::eh_ancora_trilha() Diretório/Arquivo: ancora.h	Descrição: Chama: Chamado por: List_anc.cpp\ (71) int lista_ancoras :: grava(FILE *fp)
Procedimento: (23) virtual int ancora::eh_ancora_audio Diretório/Arquivo: ancora.h	Descrição: Chama: Chamado por: List_anc.cpp\ (29) lista_ancoras::~~lista_ancoras() List_anc.cpp\ (71) int lista_ancoras :: grava(FILE *fp)
Procedimento: (24) virtual int ancora::eh_ancora_imagem() Diretório/Arquivo: ancora.h	Descrição: Chama: Chamado por: List_anc.cpp\ (29) lista_ancoras::~~lista_ancoras() List_anc.cpp\ (71) int lista_ancoras :: grava(FILE *fp)
Procedimento: (25) virtual int ancora::eh_ancora_texto() Diretório/Arquivo: ancora.h	Descrição: Chama: Chamado por: List_anc.cpp\ (29) lista_ancoras::~~lista_ancoras() List_anc.cpp\ (71) int lista_ancoras :: grava(FILE *fp)
Procedimento: (26) virtual int ancora::eh_ancora_video() Diretório/Arquivo: ancora.h	Descrição: Chama: Chamado por: List_anc.cpp\ (29) lista_ancoras::~~lista_ancoras() List_anc.cpp\ (71) int lista_ancoras :: grava(FILE *fp)
Procedimento: (27) virtual int ancora::eh_ancora_executavel() Diretório/Arquivo: ancora.h	Descrição: Chama: Chamado por:
Procedimento: (28) virtual int ancora::eh_ancora_tabela() Diretório/Arquivo: ancora.h	Descrição: Chama: Chamado por:
Procedimento: (29) int ancora::obtem_ident_externo() Diretório/Arquivo: ancora.h	Descrição: Chama: Chamado por: no.cpp\ (120) int No::obtem_indice(int ide)
Procedimento: (30) void troca_ident_externo(int id) Diretório/Arquivo: ancora.h	Descrição: Chama:

Chamado por: anc_cont.cpp\ (30) void ancora_contexto::muda_conteudo(int ide, int id_no) anc_tex.cpp\ (45) void ancora_texto::muda_conteudo(int ide, char *pal, int inicio, int fim)	
Sistema: SASHE Arquivo: Sashe\Hyperprop\ basepriv.cpp	
Procedimento: (23) baseprivada::baseprivada(char *nome) : contexto(nome) Diretório/Arquivo: basepriv.cpp	Descrição: Construtor da classe Base Privada. Chama: (23) contexto(nome) Chamado por: Hip.cpp\ (120) int hipwindow::recupera_hiperbase(char *nome) Hipger.cpp\ (66) extern "C" int FAR PASCAL _export cria_base_privada(char *nome)
Procedimento: (30) baseprivada::baseprivada(int id) : contexto(id) Diretório/Arquivo: basepriv.cpp	Descrição: Construtor utilizado quando a Base Privada é lida de arquivo. Chama: (30) Contexto(id) Chamado por:
Procedimento: (39) int baseprivada::insere_na_base_privada (No *pno) Diretório/Arquivo: basepriv.cpp	Descrição: Insere o nó na base privada: retorna 0 se conseguiu inserir retorna 1 se falta memória para a lista Chama: (41) nos.insere(pno) Chamado por:
Procedimento: (50) int baseprivada::retira_da_base_privada (No *pno) Diretório/Arquivo: basepriv.cpp	Descrição: Retira o nó da base privada retorna 0 se conseguiu retirar o nó da lista retorna 1 se a lista estiver vazia. retorna 2 se não encontrou o nó na base privada Chama: (52) nos.retira(pno) Chamado por:
Procedimento: (57) No * baseprivada::RetornaNo(int Id) Diretório/Arquivo: basepriv.cpp	Descrição: Chama: (59) primeiro_contexto_pai() (63) obtem_identificador() (64) proximo_contexto_pai() Chamado por:
Procedimento: (71) No * baseprivada::Retorna(char *nome) Diretório/Arquivo: basepriv.cpp	Descrição: Chama: (73) primeiro_contexto_pai() (77) ObtemNome() (78) proximo_contexto_pai() Chamado por: Hip.cpp\ (40) No * hipwindow::RetornaBase(char *nome)
Procedimento: (85) int baseprivada::RetornaIdent(char	Descrição:

*nome) Diretório/Arquivo: basepriv.cpp	Chama: (87) primeiro_contexto_pai() (91) ObtemNome() (91) obtem_identificador() (92) proximo_contexto_pai() Chamado por:
Procedimento: (103) int baseprivada::grava(FILE *fp) Diretório/Arquivo: basepriv.cpp	Descrição: grava a base privada em arquivo Retorno: 1: Erro de escrita no arquivo 0: Ok. Chama: (107) num_contextos() (108) obtem_identificador() (109) composicao::grava(fp) (110) primeiro_contexto_pai() (113) grava(fp) (115) proximo_contexto_pai() (117) grava_no(fp) (118) primeiro_contexto_pai() (121) grava_no(fp) (122) proximo_contexto_pai() (124) obtem_identificador() Chamado por: Hip.cpp \ (97) int hipwindow::grava_hiperbase(char *nome)
Procedimento: (133) int baseprivada::recupera(FILE *fp) Diretório/Arquivo: basepriv.cpp	Descrição: recupera a base privada do arquivo 1: Erro de leitura do arquivo 0: Ok. Chama: (138) fscanf(fp, "%d){\n", &contador) (139) fscanf(fp, "No(%d){\n", &id) (140) troca_identificador(id) (141) composicao::recupera(fp) (146) fscanf(fp, "No(%d,", &ident) (146) fscanf(fp, "%s){\n", tipo) (150) no_audio(ident) (152) no_executavel(ident) (154) no_imagem(ident) (156) no_texto(ident) (158) no_video(ident) (160) contextousuario(ident) (162) contextoversoes(ident) (164) trilha(ident) (170) recupera(fp) (172) recupera_no (fp) (173) primeiro_contexto_pai() (176) recupera_no(fp) (177) proximo_contexto_pai() (179) fscanf(fp, "}Fim_Base(%d)\n", &id) (180) obtem_identificador() Chamado por: Hip.cpp\ (120) int hipwindow::recupera_hiperbase(char *nome)

Sistema: SASHE Arquivo: Sashe\Hyperprop\ basepriv.h	
Procedimento: (27) virtual int eh_base_privada() Diretório/Arquivo: basepriv.h	Descrição: Chama: Chamado por:

Sistema: SASHE Arquivo: Sashe\Hyperprop\composic.cpp	
---	--

Procedimento: (23) composicao::composicao(char *nome_no) : No(nome_no) Diretório/Arquivo: composic.cpp	Descrição: Este construtor eh chamado pela interface quando se cria um nó composição. Chama: (23) No(nome_no) (25) nos.inicializar(0) Chamado por: Contexto.cpp\ (16) contexto::contexto(char *nome_no) : composicao(nome_no) Trilha.cpp\ (16) trilha::trilha(char *nome_no) : composicao(nome_no)
Procedimento: (33) composicao::composicao(int id) : No(id) Diretório/Arquivo: composic.cpp	Descrição: Este construtor é utilizado quando o nó composição é lido do arquivo. Chama: (33) No(id) (35) nos.inicializar(0) Chamado por: Contexto.cpp\ (23) contexto::contexto(int id) : composicao(id) Trilha.cpp\ (27) trilha::trilha(int id) : composicao(id)
Procedimento: (46) int composicao::insere_no(No *pno) Diretório/Arquivo: composic.cpp	Descrição: Insere o nó no nó composição retorna 0 se conseguiu inserir retorna 1 se falta memória para a lista retorna 2 se o nó composição nao for temporário Chama: (48) ObtemEstado() (49) insere_contexto(this) (50) return nos.insere(pno) Chamado por: Composic.cpp\ (207) void composicao::copia nos(lista nos *lista)
Procedimento: (55) int composicao::muda_assunto(char *string) Diretório/Arquivo: composic.cpp	Descrição: Chama: Chamado por: Hipger.cpp\ (964) extern "C" int FAR PASCAL _export muda_assunto(char *nome_no, char *assunto)
Procedimento: (70) int composicao::retira_no(No *pno) Diretório/Arquivo: composic.cpp	Descrição: Retira o nó do nó composição retorna 0 se conseguiu retirar o nó da lista retorna 1 se a lista estiver vazia. retorna 2 se não encontrou o nó na lista do nó composição retorna 3 se não encontrou o nó composição na lista do nó contido retorna 4 se conseguiu retirar o nó da lista e o nó não pertence a nenhum outro contexto Chama: (72) retira_contexto(this) (74) num_contextos() (77) nos.retira(pno) (81) nos.retira(pno) Chamado por: Hipger.cpp\ (339) extern "C" int FAR

	<pre>PASCAL _export retira_do_contexto(char *nome_contexto, char *nome_no) Hipger.cpp\ (731) extern "C" int FAR PASCAL _export retira_da_hiperbase(char *nome_no)</pre>
Procedimento: <pre>(89) int composicao::commit()</pre> Diretório/Arquivo: composic.cpp	Descrição: Este método verifica se as condições são satisfeitas para tornar o nó composição um nó permanente. Chama: <pre>(91) eh_contexto_versoes() (92) nos.primeiro() (97) ObtemEstado() (98) nos.proximo() (100) No::commit()</pre> Chamado por:
Procedimento: <pre>(107) int composicao::contem(No *no)</pre> Diretório/Arquivo: composic.cpp	Descrição: Método utilizado para verificar se o nó composição contém um determinado nó. Retorna 1 se caso afirmativo e 0 se caso contrário. Chama: <pre>(109) nos.primeiro() (113) nos.proximo()</pre> Chamado por: <pre>Hipger.cpp\ (263) extern "C" int FAR PASCAL _export insere_no_contexto(char *nome_cont, char *nome_no) Hipger.cpp\ (339) extern "C" int FAR PASCAL _export retira_do_contexto(char *nome_contexto, char *nome_no) Hipger.cpp\ (359) extern "C" int FAR PASCAL _export retira_da_trilha(char *nome_trilha, char *nome_no liberdade) Hipger.cpp\ (731) extern "C" int FAR PASCAL _export retira_da_hiperbase(char *nome_no) Hipger.cpp\ (1154) extern "C" int FAR PASCAL _export muda_vcorrente(char *nome_cv, char *nome_no) Hipger.cpp\ (1203) extern "C" int FAR PASCAL _export cria_ancora_contexto(char *nome_no, int ident, char *no_ancorado) Hipger.cpp\ (1279) extern "C" int FAR PASCAL _export muda_ancora_contexto(char *nome, int velho_ide, int novo_ide, char *no_ancorado) Hipger.cpp\ (1425) extern "C" int FAR PASCAL _export proxima_ancora(char *nome, char *nome_ancora) Tutor.cpp\ (39) int tutor::primeira_estrategia(contexto *pai) Trilha.cpp\ (251) int trilha::primeiro_no_trilha(char *nome, char *nome liberdade) Trilha.cpp\ (305) int trilha::anterior_no_trilha(char *nome, char *nome liberdade) Trilha.cpp\ (332) int trilha::ultimo_no_trilha(char *nome, char *nome liberdade)</pre>
Procedimento: <pre>(123) int composicao::primeiro_no(char *nome)</pre>	Descrição: Método utilizado para retornar o primeiro nó contido na lista do nó composição. Se o nó existir, este método retorna

Diretório/Arquivo: composic.cpp	o nome do primeiro nó e o valor 0, senão, retorna 1. Chama: (125) nos.primeiro() (128) ObtemNome() Chamado por: Hipger.cpp\ (442) extern "C" int FAR PASCAL _export primeiro_no_contexto(char *nome cont, char *nome_no)
Procedimento: (136) int composicao::proximo_no(char *nome) Diretório/Arquivo: composic.cpp	Descrição: Método utilizado para retornar o próximo nó contido na lista do nó composição. Retorna o nome do próximo nó e o valor 0 ou retorna apenas o valor 1. Chama: (138) nos.proximo() (141) ObtemNome() Chamado por: Hipger.cpp\ (458) extern "C" int FAR PASCAL _export proximo_no_contexto(char *nome cont, char *nome_no)
Procedimento: (150) No *composicao::primeiro_no() Diretório/Arquivo: composic.cpp	Descrição: Método utilizado para retornar o primeiro nó contido na lista do nó composição. Se o nó existir, este método retorna o ponteiro do primeiro nó, senão, retorna NULL. Chama: (152) nos.primeiro() Chamado por: Hip.cpp\ (177) void hipwindow::atualiza base ativa(No *pno)
Procedimento: (159) No *composicao::proximo_no() Diretório/Arquivo: composic.cpp	Descrição: Método utilizado para retornar o próximo nó contido na lista do nó composição. Retorna o ponteiro do próximo nó ou retorna apenas o valor NULL. Chama: (161) nos.proximo() Chamado por: Hip.cpp\ (177) void hipwindow::atualiza base ativa(No *pno)
Procedimento: (169) int composicao::primeiro_elo(int *id) Diretório/Arquivo: composic.cpp	Descrição: Método utilizado para retornar o primeiro elo contido na lista do nó composição. Se o nó existir, este método retorna o identificador do primeiro elo e o valor 0, senão, retorna 1. Chama: (171) elos.primeiro() (173) obtem_identificador() Chamado por: Hipger.cpp\ (1672) extern "C" int FAR PASCAL _export primeiro_elo(char *nome, int *id)
Procedimento: (181) int composicao::proximo_elo(int *id)	Descrição: Método utilizado para retornar o próximo elo contido na lista do nó composição. Retorna o identificador do próximo elo e o valor 0 ou retorna apenas o

Diretório/Arquivo: composic.cpp	valor 1. Chama: (185) elos.proximo() (186) prox->obtem_identificador() Chamado por: Hipger.cpp\ (1687) extern "C" int FAR PASCAL _export proximo_elo(char *nome, int *id)
Procedimento: (193) elo * composicao::retorna_elo(int i) Diretório/Arquivo: composic.cpp	Descrição: Dado o identificador de um elo, esta rotina retorna o ponteiro para este elo ou ponteiro NULL se o elo não existe. Chama: (195) elos.topo() (197) pelo->obtem_identificador() Chamado por: Hipger.cpp\ (1560) extern "C" int FAR PASCAL _export muda_elo(char *nome_contexto, int id_elo, char *origem, int anc_origem, char *destino, int anc_destino, int simultaneo, int mostra_conteudo) Hipger.cpp\ (1608) extern "C" int FAR PASCAL _export retira_elo(char *nome_contexto, int id_elo) Hipger.cpp\ (1647) extern "C" int FAR PASCAL _export recupera_elo(char *nome_contexto, int id_elo, char *id_no_orig, int *id_anc_orig, char *id_no_dest, int *id_anc_dest, int *simultaneo, int *mostra_conteudo)
Procedimento: (207) void composicao::copia_nos(lista_nos *lista) Diretório/Arquivo: composic.cpp	Descrição: Faz uma copia dos nós quando se cria uma versao do composicao. Chama: (209) lista->primeiro() (212) insere_no(pno) (213) lista->proximo(pno) Chamado por: Contexto.cpp\ (30) void contexto::copia_contexto(contexto *nc) Trilha.cpp\ (48) void trilha::copia_trilha(trilha *nc)
Procedimento: (219) int composicao::grava(FILE *fp) Diretório/Arquivo: composic.cpp	Descrição: Chama: (221) No::grava(fp) Chamado por: Contuser.cpp\ (37) int contextousuario::grava(FILE *fp) Contvers.cpp\ (69) int contextoversoes::grava(FILE *fp) Trilha.cpp\ (362) int trilha::grava(FILE *fp) Basepriv.cpp\ (103) int baseprivada::grava(FILE *fp)
Procedimento: (227) int composicao::recupera(FILE *fp) Diretório/Arquivo: composic.cpp	Descrição: Chama: (229) No::recupera(fp) Chamado por: Contuser.cpp\ (49) int contextousuario::recupera(FILE *fp) Contvers.cpp\ (81) int

	contexto::recupera(FILE *fp) Trilha.cpp\ (374) int trilha::recupera(FILE *fp) Basepriv.cpp\ (133) int baseprivada::recupera(FILE *fp)
Procedimento: (238) int composicao::grava_no(FILE *fp) Diretório/Arquivo: composic.cpp	Descrição: Retorna: 1: Erro de escrita 0: Ok. Chama: (240) No::grava_no(fp) (243) nos.grava(fp) (244) elos.grava(fp) Chamado por: Contexto.cpp\ (113) int contexto::grava_no(FILE *fp)
Procedimento: (253) int composicao::recupera_no(FILE *fp) Diretório/Arquivo: composic.cpp	Descrição: Retorna: 1: Erro de leitura 0: Ok. Chama: (255) No::recupera_no(fp) (256) fscanf(fp, "\tAssunto(%s)\n", assunto) (257) nos.recupera(fp) (258) elos.recupera(this,fp) Chamado por: Contexto.cpp\ (139) int contexto::recupera_no(FILE *fp) Trilha.cpp\ (385) int trilha::recupera_no(FILE *fp)

Sistema: SASHE**Arquivo:** Sashe\Hyperprop\composic.h

Procedimento: (27) char *recupera_assunto() Diretório/Arquivo: Composic.h	Descrição: Retorna o endereço do atributo assunto Chama: Chamado por: Hipger.cpp\ (979) extern "C" int FAR PASCAL_export recupera_assunto(char *nome_no, char *assunto)
Procedimento: (32) int insere_elo(elo *e) Diretório/Arquivo: Composic.h	Descrição: Insere na lista_elos um ponteiro para um elo Chama: (32) elos.insere(e) Chamado por: Hipger.cpp\ (1517) extern "C" int FAR PASCAL_export cria_elo(char *nome_contexto, int id_elo, char *origem, int anc_origem, char *destino, int anc_destino, int simultaneo, int mostra_conteudo) List_elo.cpp\ (61) int lista_elos::recupera(No *pno, FILE *fp)
Procedimento: (33) int retira_elo(elo *e) Diretório/Arquivo: Composic.h	Descrição: Retira de elos o ponteiro para o elo e Chama: (33) elos.retira(e) Chamado por: Hipger.cpp\ (731) extern "C" int FAR PASCAL_export retira_da_hiperbase(char *nome_no) Hipger.cpp\ (1608) extern "C" int FAR PASCAL_export retira_elo(char *nome_contexto, int id_elo)

Sistema: SASHE Arquivo: Sashe\Hyperprop\contexto.cpp	
Procedimento: (17) contexto::contexto(char *nome_no) : composicao(nome_no) Diretório/Arquivo: contexto.cpp	Descrição: Este construtor eh chamado pela interface quando se cria um nó contexto. Chama: (17) composicao(nome_no) Chamado por: Basepriv.cpp\ (23)baseprivada::baseprivada(char *nome) : contexto(nome) Contuser.cpp\ (18) contextousuario::contextousuario(char *nome_no) : contexto(nome_no) Contvers.cpp\ (17) contextoversoes::contextoversoes(No *n1,No *n2,char *nome_no) : contexto(nome_no) No.cpp\ (142) int No::criaversao(char *nome_no)
Procedimento: (23) contexto::contexto(int id) : composicao(id) Diretório/Arquivo: contexto.cpp	Descrição: Este construtor é utilizado quando o nó contexto é lido do arquivo. Chama: (23) composicao(id) Chamado por: Basepriv.cpp\ (30) baseprivada::baseprivada(int id) : contexto(id) Contuser.cpp\ (27) contextousuario::contextousuario(int id) : contexto(id) Contvers.cpp\ (31) contextoversoes::contextoversoes(int id) : contexto(id)
Procedimento: (30) void contexto::copia_contexto(contexto *nc) Diretório/Arquivo: contexto.cpp	Descrição: Este contexto é uma copia do contexto apontado por nc. Chama: (32) copia_nos(&nc->nos) (33) cria_ancora_lambda(0) Chamado por: No.cpp\ (142) int No::criaversao(char *nome_no)
Procedimento: (43) int contexto::cria_ancora_lambda(int ide) Diretório/Arquivo: contexto.cpp	Descrição: Todo nó contexto tem pelo menos uma âncora lambda. Uma âncora lambda é ancorada em todo o nó. Retorna: 1: Se erro na alocação de memória. 2: Se já existe uma âncora. Chama: (45) contem_ancora(obtem_identificador()) (48) ancora_contexto(retorna_indice(), ide, obtem_identificador()) (50) insere_ancora(panc) Chamado por: Hipger.cpp\ (161) extern "C" int FAR PASCAL_export cria_contexto(char *nome) Contexto.cpp\ (30) void contexto::copia_contexto(contexto *nc)
Procedimento: (73) ancora_contexto *contexto::cria_ancora_contexto(int ide, int id no)	Descrição: Retorna: NULL: Se não foi possível criar a âncora.

Diretório/Arquivo: contexto.cpp	ponteiro para a âncora: Ok. Chama: <pre>(75) contam_ancora(id_no) (78) ancora_contexto(retorna_indice(), ide, id_no) (80) insere_ancora(panc)</pre> Chamado por: <pre>Hipger.cpp\ (1203) extern "C" int FAR PASCAL _export cria_ancora_contexto(char *nome_no, int ident, char *no_ancorado) Trilha.cpp\ (62) int trilha::insere_no(No *pno, contexto *pno liberdade)</pre>
Procedimento: <pre>(89) ancora_contexto *contexto::contem_ancora(int id_no)</pre> Diretório/Arquivo: contexto.cpp	Descrição: Atraves do identificador do nó passado por parâmetro, esta rotina verifica se existe uma âncora cujo conteúdo é este nó. Chama: <pre>(91) (ancora_contexto *)primeira_ancora() (94) panc->obtem_ancora() (95) (ancora_contexto *)proxima_ancora()</pre> Chamado por: <pre>Contexto.cpp\ (43) int contexto::cria_ancora_lambda(int ide) Contexto.cpp\ (73) ancora_contexto *contexto::cria_ancora_contexto(int ide, int id_no)</pre>
Procedimento: <pre>(102) void contexto::muda_ancora_contexto(anco ra *p, int ide, int id_no)</pre> Diretório/Arquivo: contexto.cpp	Descrição: Chama: <pre>(105) panc->muda_conteudo(ide, id_no)</pre> Chamado por: <pre>Hipger.cpp\ (1279) extern "C" int FAR PASCAL _export muda_ancora_contexto(char *nome, int velho_ide, int novo_ide, char *no_ancorado)</pre>
Procedimento: <pre>(113) int contexto::grava_no(FILE *fp)</pre> Diretório/Arquivo: contexto.cpp	Descrição: Retorno: 1: Erro de gravação 0: Ok. Chama: <pre>(115) composicao::grava_no(fp)</pre> Chamado por: <pre>Contvers.cpp\ (92) int contextoversoes::grava_no(FILE *fp) Basepriv.cpp\ (103) int baseprivada::grava(FILE *fp)</pre>
Procedimento: <pre>(139) int contexto::recupera_no(FILE *fp)</pre> Diretório/Arquivo: contexto.cpp	Descrição: Retorno: 1: Erro de leitura 0: Ok. Chama: <pre>(141) composicao::recupera_no(fp)</pre> Chamado por: <pre>Contvers.cpp\ (103) int contextoversoes::recupera_no(FILE *fp) Basepriv.cpp\ (133) int baseprivada::recupera(FILE *fp)</pre>
Sistema: SASHE Arquivo: Sashe\Hyperprop\ contexto.h	
Procedimento: <pre>(23) virtual int</pre>	Descrição:

contexto::eh_contexto()
Diretório/Arquivo: contexto.h

Chama:
Chamado por:
Hipger.cpp\ (263) extern "C" int FAR
PASCAL _export insere_no_contexto(char
*nome_cont, char *nome_no)
Hipger.cpp\ (339) extern "C" int FAR
PASCAL _export retira_do_contexto(char
*nome_contexto, char *nome_no)
Hipger.cpp\ (442) extern "C" int FAR
PASCAL _export primeiro_no_contexto(char
*nome_cont, char *nome_no)
Hipger.cpp\ (458) extern "C" int FAR
PASCAL _export proximo_no_contexto(char
*nome_cont, char *nome_no)
Hipger.cpp\ (516) No *FAR PASCAL _export
primeiro_ponteiro_no(char *nome_cont)
Hipger.cpp\ (533) No *FAR PASCAL _export
proximo_ponteiro_no(char *nome_cont)
Hipger.cpp\ (1067) extern "C" int FAR
PASCAL _export regras_tutoriais(char
*no_atual, char *escopo, char
*historico, int dificuldade, char
*no_result, char *lib_result, int
*revisto)
Hipger.cpp\ (1203) extern "C" int FAR
PASCAL _export cria_ancora_contexto(char
*nome_no, int ident, char *no_ancorado)
Hipger.cpp\ (1279) extern "C" int FAR
PASCAL _export muda_ancora_contexto(char
*nome, int velho_ide, int novo_ide, char
*no_ancorado)
Hipger.cpp\ (1517) extern "C" int FAR
PASCAL _export cria_elo(char
*nome_contexto, int id_elo, char
*origem, int anc_origem, char *destino,
int anc_destino, int simultaneo, int
mostra_conteudo)
Hipger.cpp\ (1560) extern "C" int FAR
PASCAL _export muda_elo(char
*nome_contexto, int id_elo, char
*origem, int anc_origem, char *destino,
int anc_destino, int simultaneo, int
mostra_conteudo)
Hipger.cpp\ (1608) extern "C" int FAR
PASCAL _export retira_elo(char
*nome_contexto, int id_elo)
Hipger.cpp\ (1672) extern "C" int FAR
PASCAL _export primeiro_elo(char *nome,
int *id)
Hipger.cpp\ (1687) extern "C" int FAR
PASCAL _export proximo_elo(char *nome,
int *id)
Hipger.cpp\ (1697) elo *FAR PASCAL
_export primeiro_ponteiro_elo(char
*nome_contexto)
Hipger.cpp\ (1714) elo *FAR PASCAL
_export proximo_ponteiro_elo(char
*nome_contexto, elo *pelo)

Sistema: SASHE

Arquivo: Sashe\Hyperprop\ contuser.cpp

Procedimento:

(18)

contextousuario::contextousuario(ch
ar *nome_no) : contexto(nome_no)

Diretório/Arquivo: contuser.cpp

Descrição:

Este construtor eh chamado pela
interface quando se cria um nó contexto
do usuário.

Chama:

	(18) contexto(nome_no) Chamado por: Hipger.cpp\ (161) extern "C" int FAR PASCAL _export cria_contexto(char *nome)
Procedimento: (27) contextousuario::contextousuario(int id) : contexto(id) Diretório/Arquivo: contuser.cpp	Descrição: Este construtor é utilizado quando o nó contexto do usuário é lido do arquivo. Chama: (27) contexto(id) Chamado por: Basepriv.cpp\ (133) int baseprivada::recupera(FILE *fp)
Procedimento: (37) int contextousuario::grava(FILE *fp) Diretório/Arquivo: contuser.cpp	Descrição: grava o contexto do usuario em arquivo 1: Erro de escrita no arquivo 0: Ok Chama: (40) contexto::grava(fp) Chamado por:
Procedimento: (49) int contextousuario::recupera(FILE *fp) Diretório/Arquivo: contuser.cpp	Descrição: recupera o contexto do usuario do arquivo 1: Erro de escrita no arquivo 0: Ok Chama: (51) contexto::recupera(fp) Chamado por:

Sistema: SASHE**Arquivo:** Sashe\Hyperprop\ contuser.h

Procedimento: (17) int contextousuario::eh_contexto_usuario() Diretório/Arquivo: contuser.h	Descrição: Chama: Chamado por: Grafov.cpp\ (542) void CGrafoView::DesenhaNo(No *no) Grafov.cpp\ (1140) void CGrafoView::OnLButtonDown(UINT nFlags, CPoint point) Grafov.cpp\ (1220) void CGrafoView::OnRButtonDown(UINT nFlags, CPoint point) Hipger.cpp\ (731) extern "C" int FAR PASCAL _export retira_da_hiperbase(char *nome_no) Hipger.cpp\ (1647) extern "C" int FAR PASCAL _export recupera_elo(char *nome_contexto, int id_elo, char *id_no_orig, int *id_anc_orig, char *id_no_dest, int *id_anc_dest, int *simultaneo, int *mostra_conteudo) List_anc.cpp\ (118) int lista_ancoras :: recupera(No *n, FILE *fp) List_elo.cpp\ (61) int lista_elos::recupera(No *pno, FILE *fp) List_no.cpp\ (98) lista_nos :: ~lista_nos() No.cpp\ (142) int No::criaversao(char *nome_no)
--	---

Sistema: SASHE**Arquivo:** Sashe\Hyperprop\contvers.cpp

Procedimento: (17)	Descrição: Este construtor eh chamado quando se
-----------------------	--

contextoversoes::contextoversoes(No *n1,No *n2,char *nome_no) : contexto(nome_no) Diretório/Arquivo: contvers.cpp	cria um noh contexto de versoes. Chama: (17) contexto(nome_no) (19) altera_tipo(Contexto_Versoes) (20) insere_no(n1) (21) insere_no(n2) (23) n1->obtem_identificador() (25) CriaAncoraCV(n1) Chamado por: No.cpp\ (142) int No::criaversao(char *nome_no)
Procedimento: (31) contextoversoes::contextoversoes(in t id) : contexto(id) Diretório/Arquivo: contvers.cpp	Descrição: Este construtor é utilizado quando o contexto de versões é lido do arquivo. Chama: (31) contexto(id) (35) ObtemNome() (38) altera_tipo(Contexto_Versoes) (48) UltCV(num) Chamado por: Basepriv.cpp\ (133) int baseprivada::recupera(FILE *fp)
Procedimento: (54) void contextoversoes::CriaAncoraCV(No *n) Diretório/Arquivo: contvers.cpp	Descrição: cria uma instancia da classe ancora_contexto cujo conteudo e'a versao corrente como um todo insere a ancora na lista de ancoras do no' Chama: (58) ancora_contexto *pontancora= new ancora_contexto(0) (60) insere_ancora(pontancora) Chamado por: Contvers.cpp\ (17) contextoversoes::contextoversoes(No *n1,No *n2,char *nome_no) : contexto(nome_no)
Procedimento: (69) int contextoversoes::grava(FILE *fp) Diretório/Arquivo: contvers.cpp	Descrição: grava o contexto de versao em arquivo 1: Erro de escrita no arquivo 0: Ok Chama: (72) contexto::grava(fp) () Chamado por:
Procedimento: (81) int contextoversoes::recupera(FILE *fp) Diretório/Arquivo: contvers.cpp	Descrição: recupera o contexto de versao do arquivo 1: Erro de escrita no arquivo 0: Ok Chama: (83) contexto::recupera(fp) Chamado por:
Procedimento: (92) int contextoversoes::grava_no(FILE *fp) Diretório/Arquivo: contvers.cpp	Descrição: Retorno: 1: Erro de gravação do contexto de versão 0: Ok. Chama: (95) contexto::grava_no(fp) Chamado por:
Procedimento: (103) int contextoversoes::recupera_no(FILE	Descrição: Retorno: 1: Erro de Leitura do contexto de

*fp)	versão
	0: Ok.
Diretório/Arquivo: contvers.cpp	Chama:
	(105) fscanf(fp, "VersãoCorrente(%d)\n",
	&v_corrente)
	(106) contexto::recupera_no(fp)
	Chamado por:

Sistema: SASHE	
Arquivo: Sashe\Hyperprop\contvers.h	
Procedimento:	Descrição:
(22) int	
contextoversoes::eh_contexto_versoe	
s()	
Diretório/Arquivo: contvers.h	Chama:
	Chamado por:
	Hipger.cpp\ (263) extern "C" int FAR
	PASCAL _export insere_no_contexto(char
	*nome_cont, char *nome_no)
	Hipger.cpp\ (339) extern "C" int FAR
	PASCAL _export retira_do_contexto(char
	*nome_contexto, char *nome_no) (359)
	extern "C" int FAR PASCAL _export
	retira_da_trilha(char *nome_trilha, char
	*nome_no_liberdade)
	Hipger.cpp\ (376) extern "C" int FAR
	PASCAL _export limpa_trilha(char
	*nome_trilha)
	Hipger.cpp\ (712) extern "C" int FAR
	PASCAL _export tipo_no(char *nome)
	Hipger.cpp\ (731) extern "C" int FAR
	PASCAL _export retira_da_hiperbase(char
	*nome_no)
	Hipger.cpp\ (1121) extern "C" int FAR
	PASCAL _export cria_versao(char
	*nome_no, char *nome_no_versao)
	Hipger.cpp\ (1138) extern "C" int FAR
	PASCAL _export consulta_vcorrente(char
	*nome_cv, char *nome_no)
	Hipger.cpp\ (1154) extern "C" int FAR
	PASCAL _export muda_vcorrente(char
	*nome_cv, char *nome_no)
	Hipger.cpp\ (1517) extern "C" int FAR
	PASCAL _export cria_elo(char
	*nome_contexto, int id_elo, char
	*origem, int anc_origem, char *destino,
	int anc_destino, int simultaneo, int
	mostra_conteudo)
	Hipger.cpp\ (1560) extern "C" int FAR
	PASCAL _export muda_elo(char
	*nome_contexto, int id_elo, char
	*origem, int anc_origem, char *destino,
	int anc_destino, int simultaneo, int
	mostra_conteudo)
	Hipger.cpp\ (1608) extern "C" int FAR
	PASCAL _export retira_elo(char
	*nome_contexto, int id_elo)
	List_anc.cpp\ (118) int lista_ancoras ::
	recupera(No *n, FILE *fp)
	List_no.cpp\ (98) lista_nos ::
	~lista_nos()
	Composic.cpp\ (89) int
	composicao::commit()
Procedimento:	Descrição:
(24) int	
contextoveersoes::ObtemVersaoCorren	
te()	

Diretório/Arquivo: contvers.h	Chama: Chamado por:
Sistema: SASHE Arquivo: Sashe\Hyperprop\elo.cpp	
Procedimento: (15) elo::elo(int id) : entidade(id) Diretório/Arquivo: elo.cpp	Descrição: Construtor utilizado apenas quando o elo é carregado do arquivo Chama: (15) entidade(id) Chamado por: List_elo.cpp\ (61) int lista_elos::recupera(No *pno, FILE *fp)
Procedimento: (23) elo::elo(int id, No *orig, char *ani_orig, int ind_or, No *dest, char *ani_dest, int ind_dest, int simul, int mostr) : entidade(id) Diretório/Arquivo: elo.cpp	Descrição: Construtor utilizado pela interface Chama: (23) entidade(id) Chamado por: Trilha.cpp\ (62) int trilha::insere_no(No *pno, contexto *pno_liberdade) Trilha.cpp\ (97) int trilha::insere_no(No *pno) Trilha.cpp\ (131) int trilha::insere_no(No *pno, contexto *pno_liberdade, int posicao) Trilha.cpp\ (164) int trilha::insere_no(No *pno, int posicao) Hipger.cpp\ (1517) extern "C" int FAR PASCAL _export cria_elo(char *nome_contexto, int id_elo, char *origem, int anc_origem, char *destino, int anc_destino, int simultaneo, int mostra conteudo)
Procedimento: (41) int elo::grava_elo(FILE *fp) Diretório/Arquivo: elo.cpp	Descrição: Grava o identificador do elo, nome do nó origem, identificador da âncora origem, nome do nó destino e o identificador da âncora destino Chama: (43) obtem_identificador() (44) obtem_origem() (44) obtem_aninhamento_origem() (45) obtem_destino() (45) obtem_aninhamento_destino() Chamado por: List_elo.cpp\ (51) int lista_elos::grava(FILE *fp)
Procedimento: (56) int elo::recupera_elo(FILE *fp) Diretório/Arquivo: elo.cpp	Descrição: O identificador do elo foi recuperada pela lista de elos. Recupera do arquivo o nome do nó origem, identificador da âncora origem, nome do nó destino e o identificador da âncora destino. Chama: (60) fscanf(fp, "\t\tOrigem(%s,\"", pnome) (61) hb.Retorna(pnome) (63) fscanf(fp, "%s\",", aninhamento_origem) (64) fscanf(fp, "%d)\n",

<pre> &ancora_no_origem) (65)fscanf(fp, "\t\tDestino(%s,\"", pnome) (66) hb.Retorna(pnome) (68) fscanf(fp, "%s\",", aninhamento_destino) (69) fscanf(fp, "%d)\n", &ancora_no_destino) (70)fscanf(fp, "\t\tAtributos(%d,", &simultaneo) (71)fscanf(fp, "%d)\n", &mostra_conteudo_contexto) Chamado por: List_elo.cpp\ (61) int lista_elos::recupera(No *pno,FILE *fp) </pre>

Sistema: SASHE**Arquivo:** Sashe\Hyperprop\ elo.h

Procedimento: (23) coordArco::CoordArco() Diretório/Arquivo: elo.h	Descrição: Chama: Chamado por:
Procedimento: (51) elo::No *obtem_no_origem() Diretório/Arquivo: elo.h	Descrição: Chama: Chamado por: Hipger.cpp\ (731) extern "C" int FAR PASCAL _export retira_da_hiperbase(char *nome_no) Trilha.cpp\ (385) int trilha::recupera_no(FILE *fp) Grafov.cpp\ (309) void CGrafoView::DeterminaNivel(No *no, int nivel)
Procedimento: (53) elo::No *obtem_no_destino() Diretório/Arquivo: elo.h	Descrição: Chama: Chamado por: Hipger.cpp\ (731) extern "C" int FAR PASCAL _export retira_da_hiperbase(char *nome_no) Trilha.cpp\ (251) int trilha::primeiro_no_trilha(char *nome, char *nome_liberdade) Trilha.cpp\ (278) int trilha::proximo_no_trilha(char *nome, char *nome_liberdade) Trilha.cpp\ (305) int trilha::anterior_no_trilha(char *nome, char *nome_liberdade) Trilha.cpp\ (332) int trilha::ultimo_no_trilha(char *nome, char *nome_liberdade) Trilha.cpp\ (385) int trilha::recupera_no(FILE *fp) Grafov.cpp\ (309) void CGrafoView::DeterminaNivel(No *no, int nivel)
Procedimento: (54)void elo::muda_ancora_origem(int id) Diretório/Arquivo: elo.h	Descrição: Chama: Chamado por: Hipger.cpp\ (1560) extern "C" int FAR PASCAL _export muda_elo(char *nome_contexto, int id_elo, char *origem, int anc origem, char *destino,

	int anc_destino, int simultaneo, int mostra_conteudo)
Procedimento: (56) elo::ancora *obtem_ancora_origem() Diretório/Arquivo: elo.h	Descrição: Chama: Chamado por:
Procedimento: (57) void elo::muda_ancora_destino(int id) Diretório/Arquivo: elo.h	Descrição: Chama: Chamado por: Hipger.cpp\ (1560) extern "C" int FAR PASCAL _export muda_elo(char *nome_contexto, int id_elo, char *origem, int anc_origem, char *destino, int anc_destino, int simultaneo, int mostra_conteudo)
Procedimento: (59) elo::ancora *obtem_ancora_destino() Diretório/Arquivo: elo.h	Descrição: Chama: Chamado por:
Procedimento: (61) char elo::*obtem_origem() Diretório/Arquivo: elo.h	Descrição: Chama: Chamado por: Elo.cpp\ (41) int elo::grava_elo(FILE *fp) Grafov.w.cpp\ (857) void CGrafoView::DesenhaArco(elo *Elo) Grafov.w.cpp\ (1140) void CGrafoView::OnLButtonDown(UINT nFlags, CPoint point) Grafov.w.cpp\ (1220) void CGrafoView::OnRButtonDown(UINT nFlags, CPoint point)
Procedimento: (63) char elo::*obtem_destino() Diretório/Arquivo: elo.h	Descrição: Chama: (63) no_destino->ObtemNome() Chamado por: Elo.cpp\ (41) int elo::grava_elo(FILE *fp) Grafov.w.cpp\ (1140) void CGrafoView::OnLButtonDown(UINT nFlags, CPoint point) Grafov.w.cpp\ (1220) void CGrafoView::OnRButtonDown(UINT nFlags, CPoint point)
Procedimento: (65) char elo::*obtem_aninhamento_origem() Diretório/Arquivo: elo.h	Descrição: Chama: Chamado por: Hipger.cpp\ (1647) extern "C" int FAR PASCAL _export recupera_elo(char *nome_contexto, int id_elo, char *id_no_orig, int *id_anc_orig, char *id_no_dest, int *id_anc_dest, int *simultaneo, int *mostra_conteudo) Elo.cpp\ (41) int elo::grava_elo(FILE *fp) Grafov.w.cpp\ (25) #define NO_ORIGEM(Elo) UltimoNoAninhado(Elo- >obtem_aninhamento_origem())
Procedimento:	Descrição:

(67) char elo::obtem_aninhamento_destino() Diretório/Arquivo: elo.h	Chama: Chamado por: Hipger.cpp\ (1647) extern "C" int FAR PASCAL _export recupera_elo(char *nome_contexto, int id_elo, char *id_no_orig, int *id_anc_orig, char *id_no_dest, int *id_anc_dest, int *simultaneo, int *mostra_conteudo) Elo.cpp\ (41) int elo::grava_elo(FILE *fp) Grafov.w.cpp\ (26) #define NO_DESTINO(Elo) UltimoNoAninhado(Elo- >obtem_aninhamento_destino())
Procedimento: (68) int elo::obtem_id_ancora_origem() Diretório/Arquivo: elo.h	Descrição: Chama: Chamado por: Hipger.cpp\ (1647) extern "C" int FAR PASCAL _export recupera_elo(char *nome_contexto, int id_elo, char *id_no_orig, int *id_anc_orig, char *id_no_dest, int *id_anc_dest, int *simultaneo, int *mostra_conteudo)
Procedimento: (69) int elo::obtem_id_ancora_destino() Diretório/Arquivo: elo.h	Descrição: Chama: Chamado por: Hipger.cpp\ (1647) extern "C" int FAR PASCAL _export recupera_elo(char *nome_contexto, int id_elo, char *id_no_orig, int *id_anc_orig, char *id_no_dest, int *id_anc_dest, int *simultaneo, int *mostra_conteudo) Trilha.cpp\ (251) int trilha::primeiro_no_trilha(char *nome, char *nome_liberdade) Trilha.cpp\ (278) int trilha::proximo_no_trilha(char *nome, char *nome_liberdade) Trilha.cpp\ (305) int trilha::anterior_no_trilha(char *nome, char *nome_liberdade) Trilha.cpp\ (332) int trilha::ultimo_no_trilha(char *nome, char *nome_liberdade)
Procedimento: (70) int elo::eh_simultaneo() Diretório/Arquivo: elo.h	Descrição: Chama: Chamado por: Hipger.cpp\ (1647) extern "C" int FAR PASCAL _export recupera_elo(char *nome_contexto, int id_elo, char *id_no_orig, int *id_anc_orig, char *id_no_dest, int *id_anc_dest, int *simultaneo, int *mostra_conteudo) Grafov.w.cpp\ (1140) void CGrafoView::OnLButtonDown(UINT nFlags, CPoint point) Grafov.w.cpp\ (1220) void CGrafoView::OnRButtonDown(UINT nFlags, CPoint point)
Procedimento: (72) int elo::eh_para_mostrar()	Descrição:

Diretório/Arquivo: elo.h	Chama: Chamado por: Hipger.cpp\ (1647) extern "C" int FAR PASCAL _export recupera_elo(char *nome_contexto, int id_elo, char *id_no_orig, int *id_anc_orig, char *id_no_dest, int *id_anc_dest, int *simultaneo, int *mostra conteudo)
Procedimento: (73) void elo::muda_mostra_conteudo_contexto(int tipo) Diretório/Arquivo: elo.h	Descrição: Chama: Chamado por: Hipger.cpp\ (1560) extern "C" int FAR PASCAL _export muda_elo(char *nome_contexto, int id_elo, char *origem, int anc_origem, char *destino, int anc_destino, int simultaneo, int mostra conteudo)

Sistema: SASHE Arquivo: Sashe\Hyperprop\entidade.cpp	
Procedimento: (12) entidade::entidade(int id) Diretório/Arquivo: entidade.cpp	Descrição: Chama: Chamado por: No.cpp\ (26) No::No(char *nome_no) : entidade() Ancora.cpp\ (12) ancora::ancora(int Ind, int ide): entidade(Ind) Ancora.cpp\ (20) ancora::ancora(int Ind): entidade(Ind) Elo.cpp\ (15) elo::elo(int id) : entidade(id) Elo.cpp\ (23) elo::elo(int id, No *orig, char *ani_orig, int ind_or, No *dest, char *ani_dest, int ind_dest, int simul, int mostr) : entidade(id)
Procedimento: (19) void entidade::troca_identificador(int id) Diretório/Arquivo: entidade.cpp	Descrição: Chama: Chamado por: No.cpp\ (26) No::No(char *nome_no) : entidade() No.cpp\ (43) No::No(int id) : entidade() No_audio.cpp\ (47) void no_audio::muda_ancora_audio(ancora_audio *p, int ide) No_exec.cpp\ (47) void no_executavel::muda_ancora_executavel(an cora_executavel *p, int ide) No_image.cpp\ (48) void no_imagem::muda_ancora_imagem(ancora_ima gem *p, int ide) No_tab.cpp\ (47) no_tabela::muda_ancora_tabela(ancora_tab ela *p, int ide) No_video.cpp\ (46) void no_video::muda_ancora_video(ancora_video *p, int ide) Anc_tri.cpp\ (29) void ancora_trilha::muda_conteudo(int ide) Basepriv.cpp\ (133) int baseprivada::recupera(FILE *fp)

Procedimento:	Descrição:
(26) int entidade::obtem_identificador() Diretório/Arquivo: entidade.cpp	Chama: Chamado por: GrafoVw.cpp\ (542) void CGrafoView::DesenhaNo(No *no) GrafoVw.cpp\ (857) void CGrafoView::DesenhaArco(elo *Elo) GrafoVw.cpp\ (1494) UINT CGrafoView::GetMouseOverTip(CPoint point) Hip.cpp\ (49) No * hipwindow::RetornaNo(int Id) Hip.cpp\ (63) int hipwindow::RetornaIdent(char *nome) Hip.cpp\ (165) int hipwindow::obtem_identificador_ultimo_no() Hipger.cpp\ (1154) extern "C" int FAR PASCAL _export muda_vcorrente(char *nome_cv, char *nome_no) Hipger.cpp\ (1203) extern "C" int FAR PASCAL _export cria_ancora_contexto(char *nome_no, int ident, char *no_ancorado) Hipger.cpp\ (1279) extern "C" int FAR PASCAL _export muda_ancora_contexto(char *nome, int velho_ide, int novo_ide, char *no_ancorado) Hipger.cpp\ (1311) extern "C" int FAR PASCAL _export recupera_nome_ancora(char *nome, int id_ancora, char *nome_ancora) Hipger.cpp\ (1372) extern "C" int FAR PASCAL _export primeira_ancora(char *nome, char *nome_ancora) Hipger.cpp\ (1425) extern "C" int FAR PASCAL _export proxima_ancora(char *nome, char *nome_ancora) List_no.cpp\ (182) int lista_nos::grava(FILE *fp) No.cpp\ (26) No::No(char *nome_no) : entidade() No.cpp\ (107) ancora * No::retorna_ancora(int ind) No.cpp\ (120) int No::obtem_indice(int ide) No.cpp\ (133) int No::retorna_indice() No.cpp\ (142) int No::criaversao(char *nome_no) No.cpp\ (264) int No :: grava(FILE *fp) No.cpp\ (275) int No :: recupera(FILE *fp) No_audio.cpp\ (55) void no_audio::copia_ancoras_audio(lista_ancoras *ancoras) No_exec.cpp\ (55) void no_executavel::copia_ancoras_executavel(lista_ancoras *ancoras) No_image.cpp\ (56) void no_imagem::copia_ancoras_imagem(lista_ancoras *ancoras) No_tab.cpp\ (55) void no_tabela::copia_ancoras_tabela(lista_ancoras *ancoras) No_texto\ (60) void no_texto::copia_ancoras_texto(lista_ancoras *ancoras)

	<pre> No_video.cpp\ (54) void no_video::copia_ancoras_video(lista_ancoras *ancoras) Trilha.cpp\ (62) int trilha::insere_no(No *pno, contexto *pno_liberdade) Trilha.cpp\ (97) int trilha::insere_no(No *pno) Trilha.cpp\ (131) int trilha::insere_no(No *pno, contexto *pno_liberdade, int posicao) Trilha.cpp\ (164) int trilha::insere_no(No *pno, int posicao) Basepriv.cpp\ (57) No * baseprivada::RetornaNo(int Id) Basepriv.cpp\ (85) int baseprivada::RetornaIdent(char *nome) Basepriv.cpp\ (103) int baseprivada::grava(FILE *fp) Basepriv.cpp\ (133) int baseprivada::recupera(FILE *fp) Composic.cpp\ (169) int composicao::primeiro_elo(int *id) Composic.cpp\ (181) int composicao::proximo_elo(int *id) Composic.cpp\ (193) elo * composicao::retorna_elo(int i) Contexto.cpp\ (43) int contexto::cria_ancora_lambda(int ide) Contvers.cpp\ (17) contextoversoes::contextoversoes(No *n1, No *n2, char *nome_no) : contexto(nome_no) Elo.cpp\ (41) int elo::grava_elo(FILE *fp) </pre>
Procedimento: (33) int entidade::grava(FILE *fp) Diretório/Arquivo: entidade.cpp	Descrição: Chama: Chamado por: Ancora.cpp\ (28) int ancora::grava_ancora(FILE *fp)
Procedimento: (40) int entidade::recupera(FILE *fp) Diretório/Arquivo: entidade.cpp	Descrição: Chama: Chamado por:
Procedimento: (50) int entidade::fscanf(FILE *fp, const char *format, int *item) Diretório/Arquivo: entidade.cpp	Descrição: Chama: Chamado por: No.cpp\ (318) int No::recupera_no(FILE *fp) No.cpp\ (275) int No :: recupera(FILE *fp) Elo.cpp\ (56) int elo::recupera_elo(FILE *fp) Basepriv.cpp\ (133) int baseprivada::recupera(FILE *fp) Anc_tex.cpp\ (71) int ancora_texto::recupera_ancora(FILE *fp) List_no.cpp\ (231) int lista_nos::recupera(FILE *fp) List_elo.cpp\ (61) int lista_elos::recupera(No *pno, FILE *fp) List_anc.cpp\ (118) int lista_ancoras :: recupera(No *n, FILE *fp)

	<pre>Elo.cpp\ (56) int elo::recupera_elo(FILE *fp) Contvers.cpp\ (103) int contextoversoes::recupera_no(FILE *fp) Ancora.cpp\ (38) int ancora::recupera_ancora(FILE *fp) Anc_cont.cpp\ (47) int ancora_contexto::recupera_ancora(FILE *fp)</pre>
Procedimento: <pre>(96) int entidade::fscanf(FILE *fp, const char *format, char *item) Diretório/Arquivo: entidade.cpp</pre>	Descrição: Chama: Chamado por: <pre>Terminal.cpp\ (123) int no_terminal::recupera_no(FILE *fp) No.cpp\ (318) int No::recupera_no(FILE *fp) No.cpp\ (275) int No :: recupera(FILE *fp) Hip.cpp\ (120) int hipwindow::recupera_hiperbase(char *nome) Elo.cpp\ (56) int elo::recupera_elo(FILE *fp) Composic.cpp\ (253) int composicao::recupera_no(FILE *fp) Basepriv.cpp\ (133) int baseprivada::recupera(FILE *fp) Anc_tex.cpp\ (71) int ancora_texto::recupera_ancora(FILE *fp)</pre>

Sistema: SASHE**Arquivo:** Sashe\Hyperprop\entidade.h**Procedimento:**

```
(21) int obtem_identificador()
Diretório/Arquivo: entidade.h
```

Descrição:

Chama:
Chamado por:

```
Basepriv.cpp\ (57) No *
baseprivada::RetornaNo(int Id)
Basepriv.cpp\ (85) int
baseprivada::RetornaIdent(char *nome)
Basepriv.cpp\ (103) int
baseprivada::grava(FILE *fp)
Basepriv.cpp\ (133) int
baseprivada::recupera(FILE *fp)
Composic.cpp\ (169)
composicao::primeiro_elo(int *id)
Composic.cpp\ (181)int
composicao::proximo_elo(int *id)
Composic.cpp\ (193) elo *
composicao::retorna_elo(int i)
Contexto.cpp\ (43) int
contexto::cria_ancora_lambda(int ide)
Contvers.cpp(17)
contextoversoes::contextoversoes(No *n1,No *n2,char *nome_no) :
contexto(nome_no)
Elo.cpp\ (41) int elo::grava_elo(FILE *fp)
Grafov.cpp\ (542) void
CGrafoView::DesenhaNo(No *no)
Grafov.cpp\ (857) void
CGrafoView::DesenhaArco(elo *Elo)
Grafov.cpp\ (1494) UINT
CGrafoView::GetMouseOverTip(CPoint point)
```

```
Hip.cpp\ (49) No *
hipwindow::RetornaNo(int Id)
Hip.cpp\ (63) int
hipwindow::RetornaIdent(char *nome)
Hip.cpp\ (165) int
hipwindow::obtem_identificador_ultimo_no
()
Hipger.cpp\ (1154) extern "C" int FAR
PASCAL _export muda_vcorrente(char
*nome_cv, char *nome_no)
Hipger.cpp\ (1203) extern "C" int FAR
PASCAL _export cria_ancora_contexto(char
*nome_no, int ident, char *no_ancorado)
Hipger.cpp\ (1279) extern "C" int FAR
PASCAL _export muda_ancora_contexto(char
*nome, int velho_ide, int novo_ide, char
*no_ancorado)
Hipger.cpp\ (1311) extern "C" int FAR
PASCAL _export recupera_nome_ancora(char
*nome, int id_ancora, char *nome_ancora)
Hipger.cpp\ (1372) extern "C" int FAR
PASCAL _export primeira_ancora(char
*nome, char *nome_ancora)
Hipger.cpp\ (1425) extern "C" int FAR
PASCAL _export proxima_ancora(char
*nome, char *nome_ancora)
List_no.cpp\ (182) int
lista_nos::grava(FILE *fp) // grava a
lista de nos (os identificadores dos
nos)
No.cpp\ (26) No::No(char *nome_no) :
entidade()
No.cpp\ (107) ancora *
No::retorna_ancora(int ind)
No.cpp\ (120) int No::obtem_indice(int
ide)
No.cpp\ (133) int No::retorna_indice()
No.cpp\ (142) int No::criaversao(char
*nome_no)
No.cpp\ (264) int No :: grava(FILE *fp)
No.cpp\ (275) int No :: recupera(FILE
*fp)
No_audio.cpp\ (55) void
no_audio::copia_ancoras_audio(lista_anco
ras *ancoras)
No_exec.cpp\ (55) void
no_executavel::copia_ancoras_executavel(
lista_ancoras *ancoras)
No_image.cpp\ (56) void
no_imagem::copia_ancoras_imagem(lista_an
coras *ancoras)
No_tab.cpp\ (55) void
no_tabela::copia_ancoras_tabela(lista_an
coras *ancoras)
No_texto.cpp\ (60) void
no_texto::copia_ancoras_texto(lista_anco
ras *ancoras)
No_video.cpp\ (54) void
no_video::copia_ancoras_video(lista_anco
ras *ancoras)
Trilha.cpp\ (62) int
trilha::insere_no(No *pno, contexto
*pno_liberdade)
Trilha.cpp\ (97) int
trilha::insere_no(No *pno)
Trilha.cpp\ (131) int
trilha::insere_no(No *pno, contexto
```

*pno liberdade, int posicao) Trilha.cpp\ (164) int trilha::insere no(No *pno, int posicao)
--

Sistema: SASHE**Arquivo:** Sashe\Hyperprop\ frame.cpp

Procedimento: (51) CFrame::CFrame() Diretório/Arquivo: frame.cpp	Descrição: Chama: Chamado por: Grafo.cpp\ (172) extern "C" BOOL FAR PASCAL _export MostrarJanela(HWND hWndParent)
Procedimento: (56) CFrame::~~CFrame() Diretório/Arquivo: frame.cpp	Descrição: Chama: Chamado por:
Procedimento: (61) void CFrame::CreateFrame(HWND hWndParent) Diretório/Arquivo: frame.cpp	Descrição: Chama: Chamado por:
Procedimento: (80) void CFrame::AssertValid() const Diretório/Arquivo: frame.cpp	Descrição: Chama: (82) CFrameWnd::AssertValid() Chamado por:
Procedimento: (85) void CFrame::Dump(CdumpContext& dc) const Diretório/Arquivo: frame.cpp	Descrição: Chama: (87) CFrameWnd::Dump(dc) Chamado por:
Procedimento: (96) void CFrame::OnFileOpen() Diretório/Arquivo: frame.cpp	Descrição: Chama: Chamado por:
Procedimento: (100) int CFrame::OnCreate(LPCREATESTRUCT lpCreateStruct) Diretório/Arquivo: frame.cpp	Descrição: Chama: (102) CFrameWnd::OnCreate(lpCreateStruct) (105) m_wndToolBar.Create(this) (106) m_wndToolBar.LoadBitmap(IDR_MAINFRAME) (107) m_wndToolBar.SetButtons(buttons, sizeof(b uttons)/sizeof(UINT)) (114) CreateToolBar() Chamado por:
Procedimento: (119) BOOL Cframe::CreateToolBar() Diretório/Arquivo: frame.cpp	Descrição: Chama: (121)m_wndToolBar.Create(this,WS_CHILD W S_VISIBLE CBRS_TOP) (122) m_wndToolBar.LoadBitmap(IDR_MAINFRAME) (123)m_wndToolBar.SetButtons(buttons,siz eof(buttons)/sizeof(UINT)) (131) m_wndToolBar.SetButtonInfo(1, IDW_COMBO, TBBS_SEPARATOR, 150) (133) m_wndToolBar.SetButtonInfo(2, ID_SEPARATOR, TBBS_SEPARATOR, 12)

<pre> (135) m_wndToolBar.GetItemRect(1, &rect) (138) m_wndToolBar.m_comboBox.Create(CBS_DROPD OWNLIST WS_VISIBLE WS_TABSTOP, rect, &m_wndToolBar, IDW_COMBO)) (147) memset(&logFont, 0, sizeof(logFont)) (154) m_wndToolBar.m_font.CreateFontIndirect(& logFont) (157) m_wndToolBar.m_comboBox.SetFont(&m_wndTo olBar.m_font) </pre>	
Chamado por:	
Procedimento: (162) void CFrame::OnDestroy() Diretório/Arquivo: frame.cpp	Descrição:
	Chama: Chamado por:
Procedimento: (170) void CFrame::OnClose() Diretório/Arquivo: frame.cpp	Descrição:
	Chama: Chamado por:
Procedimento: (185) void CFrame::OnSize(UINT nType, int cx, int cy) Diretório/Arquivo: frame.cpp	Descrição:
	Chama: Chamado por:
Procedimento: (193) void CFrame::OnPaint() Diretório/Arquivo: frame.cpp	Descrição:
	Chama: Chamado por:

Sistema: SASHE**Arquivo:** Sashe\Hyperprop\ frame.h

Procedimento: (32) LRESULT ProcJanela(UINT mensagem, WPARAM wParam, LPARAM lParam) Diretório/Arquivo: frame.h	Descrição:
	Chama: (32) WindowProc(mensagem, wParam, lParam) Chamado por:

Sistema: SASHE**Arquivo:** Sashe\Hyperprop\ grafo.cpp

Procedimento: (41) CGrafoApp::CGrafoApp() Diretório/Arquivo: grafo.cpp	Descrição:
	Chama: Chamado por:
Procedimento: (55) BOOL CGrafoApp::InitInstance() Diretório/Arquivo: grafo.cpp	Descrição:
	Chama: Chamado por:
Procedimento: (107) CAboutDlg::CAboutDlg() : CDialog(CAboutDlg::IDD) Diretório/Arquivo: grafo.cpp	Descrição:
	Chama: Chamado por:
Procedimento: (113) void CAboutDlg::DoDataExchange(CDataExch ange* pDX) Diretório/Arquivo: grafo.cpp	Descrição:
	Chama: (115) CDialog::DoDataExchange(pDX)

	Chamado por:
Procedimento: (127) void CGrafoApp::OnAppAbout() Diretório/Arquivo: grafo.cpp	Descrição: Chama: Chamado por:
Procedimento: (133) void CGrafoApp::OnAppExit() Diretório/Arquivo: grafo.cpp	Descrição: Chama: Chamado por:
Procedimento: (149) extern "C" BOOL FAR PASCAL _export InicializarGrafo() Diretório/Arquivo: grafo.cpp	Descrição: Chama: Chamado por:
Procedimento: (172) extern "C" BOOL FAR PASCAL _export MostrarJanela(HWND hWndParent) Diretório/Arquivo: grafo.cpp	Descrição: Chama: (176) new Cframe (177) AfxGetApp()- >m_templateList.GetHead() (181) pTemplate->CreateNewDocument() (187) CWnd::FromHandle(hWndParent) (191) pTemplate-> InitialUpdateFrame(pMainFrame, context.m_pCurrentDoc, 1) (192) AfxGetApp()->m_pMainWnd-> ShowWindow(SW_SHOW) (193) AfxGetApp()->m_pMainWnd-> UpdateWindow() (195) AfxGetApp()->m_pMainWnd-> DestroyWindow() Chamado por: Hip_prin.pas\ (171) To_fo_Principal.Apresentar_Browser_Grafi co(p Contexto: Ptochar)
Procedimento: (199) extern "C" BOOL FAR PASCAL _export EsconderJanela() Diretório/Arquivo: grafo.cpp	Descrição: Chama: Chamado por:
Procedimento: (210) extern "C" void FAR PASCAL _export ApresentaContexto(char *no) Diretório/Arquivo: grafo.cpp	Descrição: Chama: (214) GetActiveDocument() (217) GetActiveDocument() (221) grafo->Hiperbase(pView->contexto) (223) retorna_identificador(no) (225) tipo_no(no) (231) sprintf(erro, "%s não é Nó Contexto ou Trilha.", no) (233) grafo->Hiperbase(pView->contexto) (239) sprintf(erro, "%s não consta na Hiperbase.", no); (241) grafo->Hiperbase(pView->contexto) (244) grafo->SetTitle(pView->contexto) (245) pView->SetDataObject() (246) pView->Refresh() (247) pMainWnd-> m_wndToolBar.m_comboBox.InsertString(0,p View->contexto) (248) pMainWnd-> m_wndToolBar.m_comboBox.SelectString(- 1,pView->contexto) Chamado por:

	Hip_prin.pas\ (171) To_fo_Principal.Apresentar_Browser_Grafi co(p_Contexto: Ptochar)
Procedimento: (256) extern "C" void FAR PASCAL _export WindowResize(int hsize,int vsize) Diretório/Arquivo: grafo.cpp	Descrição: Chama: (259) m_pMainWnd->MoveWindow(0, 0, hsize, vsize, TRUE) Chamado por:
Procedimento: (262) extern "C" BOOL FAR PASCAL _export TranslateMsg(LPMSG lpMsg) Diretório/Arquivo: grafo.cpp	Descrição: Chama: (264) ProcessDlIdle() (265) AfxGetApp()- >PreTranslateMessage(lpMsg) Chamado por: Hip_prin.pas\ (235) To_fo_Principal.AppMessage(var Msg: TMsg; var Handled: Boolean)
Procedimento: (268) extern "C" void FAR PASCAL _export ProcessDlIdle() Diretório/Arquivo: grafo.cpp	Descrição: Chama: Chamado por:

Sistema: SASHE	
Arquivo: Sashe\Hyperprop\grafo.h	
Procedimento: (24) void OpenWindow() Diretório/Arquivo: grafo.h	Descrição: Chama: (24) OnFileNew() Chamado por:

Sistema: SASHE	
Arquivo: Sashe\Hyperprop\ grafoapi.h	
Procedimento: (5) BOOL FAR PASCAL _export InicializarGrafo() Diretório/Arquivo: grafoapi.h	Descrição: Chama: Chamado por:
Procedimento: (6) BOOL FAR PASCAL _export MostrarJanela(HWND hWndParent) Diretório/Arquivo: grafoapi.h	Descrição: Chama: Chamado por:
Procedimento: (7) BOOL FAR PASCAL _export EsconderJanela() Diretório/Arquivo: grafoapi.h	Descrição: Chama: Chamado por:
Procedimento: (8) BOOL FAR PASCAL _export EsconderJanela() Diretório/Arquivo: grafoapi.h	Descrição: Chama: Chamado por:
Procedimento: (9) int FAR PASCAL _export recuperar(char *) Diretório/Arquivo: grafoapi.h	Descrição: Chama: Chamado por:
Procedimento: (10) void FAR PASCAL _export WindowResize(int, int)	Descrição:

Diretório/Arquivo: grafoapi.h	Chama: Chamado por:
Procedimento: (11) BOOL FAR PASCAL _export TranslateMsg(LPMSG lpMsg) Diretório/Arquivo: grafoapi.h	Descrição: Chama: Chamado por:
Procedimento: (12) void FAR PASCAL _export ProcessDllIdle() Diretório/Arquivo: grafoapi.h	Descrição: Chama: Chamado por:

Sistema: SASHE	
Arquivo: Sashe\Hyperprop\ grafoob.cpp	
Procedimento: (31) CGrafoObj::CGrafoObj() Diretório/Arquivo: grafoob.cpp	Descrição: Chama: Chamado por:
Procedimento: (43) CGrafoObj::~~CGrafoObj() Diretório/Arquivo: grafoob.cpp	Descrição: Chama: Chamado por:
Procedimento: (51) BOOL CGrafoObj::OnNewDocument() Diretório/Arquivo: grafoob.cpp	Descrição: Chama: (53) Cdocument::OnNewDocument() Chamado por:
Procedimento: (63) void CGrafoObj::Serialize(CArchive& ar) Diretório/Arquivo: grafoob.cpp	Descrição: Chama: Chamado por:
Procedimento: (79) void CGrafoObj::AssertValid() const Diretório/Arquivo: grafoob.cpp	Descrição: Chama: (81) Cdocument::AssertValid() Chamado por:
Procedimento: (84) void CgrafoObj::Dump(CdumpContext& dc) const Diretório/Arquivo: grafoob.cpp	Descrição: Chama: (86) Cdocument::Dump(dc) Chamado por:

Sistema: SASHE	
Arquivo: Sashe\Hyperprop\ grafoob.h	
Procedimento: (21) CGrafoObj::CSize GetDocSize() Diretório/Arquivo: grafoob.h	Descrição: Chama: Chamado por: Grafov.w.cpp\ (211) void CGrafoView::OnInitialUpdate()
Procedimento: (22) int CGrafoObj::Salvar(char *arg) Diretório/Arquivo: grafoob.h	Descrição: Chama: Chamado por:
Procedimento: (23) int CGrafoObj::Recuperar(char *arg) Diretório/Arquivo: grafoob.h	Descrição: Chama: (23) recuperar(arg)

	Chamado por: Grafov.w.cpp\ (1075) void CGrafoView::OnFileOpen()
Procedimento: (24) int CgrafoObj::LimparHiperbase() Diretório/Arquivo: grafoob.h	Descrição: Chama: (24) limpar_hiperbase() Chamado por: Grafov.w.cpp\ (1049) void CGrafoView::OnFileNew()
Procedimento: (25) int CgrafoObj::Hiperbase(char *nome) Diretório/Arquivo: grafoob.h	Descrição: Chama: (25) primeiro_no_hiperbase(nome) Chamado por: Grafo.cpp\ (210) extern "C" void FAR PASCAL _export ApresentaContexto(char *no) Grafov.w.cpp\ (1075) void CGrafoView::OnFileOpen()
Procedimento: (26) CgrafoObj::No *RetornaPtNo(int id) Diretório/Arquivo: grafoob.h	Descrição: Chama: (26) retorna_ponteiro_no(id) Chamado por: Grafov.w.cpp\ (331) No *CGrafoView::UltimoNoAninhado(char *s)
Procedimento: (27) CgrafoObj::No *PrimeiroNo(char *nome) Diretório/Arquivo: grafoob.h	Descrição: Chama: (27) primeiro_ponteiro_no(nome) Chamado por: Grafov.w.cpp\ (220) void CGrafoView::SetDataObject() Grafov.w.cpp\ (509) void CGrafoView::Coordenada(int deslx, int desly, int width, int height, int espx, int espy) Grafov.w.cpp\ (639) int CGrafoView::Caminho(int x2temp, int y2temp, int comp) Grafov.w.cpp\ (996) void CGrafoView::OnAtualizar() Grafov.w.cpp\ (1032) void CGrafoView::Refresh() Grafov.w.cpp\ (1140) void CGrafoView::OnLButtonDown(UINT nFlags, CPoint point) Grafov.w.cpp\ (1220) void CGrafoView::OnRButtonDown(UINT nFlags, CPoint point) Grafov.w.cpp\ (1494) UINT CGrafoView::GetMouseOverTip(CPoint point)
Procedimento: (28) CgrafoObj::No *ProximoNo(char *nome) Diretório/Arquivo: grafoob.h	Descrição: Chama: (28) proximo_ponteiro_no(nome) Chamado por: Grafov.w.cpp\ (220) void CGrafoView::SetDataObject() Grafov.w.cpp\ (509) void

	<pre> CGrafoView::Coordenada(int deslx, int desly, int width, int height, int espx, int espy) Grafov.cpp\ (639) int CGrafoView::Caminho(int x2temp, int y2temp, int comp) Grafov.cpp\ (996) void CGrafoView::OnAtualizar() Grafov.cpp\ (1032) void CGrafoView::Refresh() Grafov.cpp\ (1140) void CGrafoView::OnLButtonDown(UINT nFlags, CPoint point) Grafov.cpp\ (1220) void CGrafoView::OnRButtonDown(UINT nFlags, CPoint point) Grafov.cpp\ (1494) UINT CGrafoView::GetMouseOverTip(CPoint point) </pre>
Procedimento: <pre> (29) CgrafoObj::elo *PrimeiroElo(char *nome) Diretório/Arquivo: grafoob.h </pre>	Descrição: Chama: <pre> (29) primeiro_ponteiro_elo(nome) </pre> Chamado por: <pre> Grafov.cpp\ (220) void CGrafoView::SetDataObject() Grafov.cpp\ (309) void CGrafoView::DeterminaNivel(No *no, int nivel) Grafov.cpp\ (345) int CGrafoView::PaisNoNivelDeCima(No *no) Grafov.cpp\ (380) int CGrafoView::DeterminaEscopo(No *no) Grafov.cpp\ (412) No *CGrafoView::FilhosNoNivelDeBaixo(No *no, int &i) Grafov.cpp\ (448) int CGrafoView::SomaEscopoEsquerda(No *no, No *filho) Grafov.cpp\ (471) void CGrafoView::Posicao(No *no) Grafov.cpp\ (639) int CGrafoView::Caminho(int x2temp, int y2temp, int comp) Grafov.cpp\ (672) void CGrafoView::Roteamento(elo *Elo) Grafov.cpp\ (1032) void CGrafoView::Refresh() Grafov.cpp\ (1140) void CGrafoView::OnLButtonDown(UINT nFlags, Cpoint point) Grafov.cpp\ (1220) void CGrafoView::OnRButtonDown(UINT nFlags, CPoint point) </pre>
Procedimento: <pre> (30) CgrafoObj::elo *ProximoElo(char *nome, elo *e) Diretório/Arquivo: grafoob.h </pre>	Descrição: Chama: <pre> (30) proximo_ponteiro_elo(nome, e) </pre> Chamado por: <pre> Grafov.cpp\ (220) void CGrafoView::SetDataObject() Grafov.cpp\ (309) void CGrafoView::DeterminaNivel(No *no, int nivel) Grafov.cpp\ (345) int </pre>

CGrafoView::PaisNoNivelDeCima(No *no)
Grafovw.cpp\ (380) int
CGrafoView::DeterminaEscopo(No *no)
Grafovw.cpp\ (412) No
*CGrafoView::FilhosNoNivelDeBaixo(No
*no, int &i)
Grafovw.cpp\ (448) int
CGrafoView::SomaEscopoEsquerda(No *no,
No *filho)
Grafovw.cpp\ (471) void
CGrafoView::Posicao(No *no)
Grafovw.cpp\ (639) int
CGrafoView::Caminho(int x2temp, int
y2temp, int comp)
Grafovw.cpp\ (672) void
CGrafoView::Roteamento(elo *Elo)
Grafovw.cpp\ (1032) void
CGrafoView::Refresh()
Grafovw.cpp\ (1140) void
CGrafoView::OnLButtonDown(UINT nFlags,
Cpoint point)
Grafovw.cpp\ (1220) void
CGrafoView::OnRButtonDown(UINT nFlags,
CPoint point)

Sistema: SASHE	
Arquivo: Sashe\Hyperprop\ grafos.h	
Procedimento: (94) grafo::grafo() Diretório/Arquivo: grafos.h	Descrição: Chama: Chamado por:
Procedimento: (95) grafo::~~grafo() Diretório/Arquivo: grafos.h	Descrição: Chama: Chamado por:
Procedimento: (96) int grafo::CriaItem(char *nome,long tipo,long classe,int classif,char *arq) Diretório/Arquivo: grafos.h	Descrição: Chama: Chamado por:
Procedimento: (97) int grafo::CriaItem(int id, FILE *fp) Diretório/Arquivo: grafos.h	Descrição: Chama: Chamado por:
Procedimento: (98) int grafo::CriaLigacao(int id1, int id2, long tipo) Diretório/Arquivo: grafos.h	Descrição: Chama: Chamado por:
Procedimento: (99) int grafo::CriaLigacao(int id, FILE *fp) Diretório/Arquivo: grafos.h	Descrição: Chama: Chamado por:
Procedimento: (100) int grafo::SalvaGrafo(FILE *fp) Diretório/Arquivo: grafos.h	Descrição: Chama: Chamado por:
Procedimento: (101) int grafo::LeGrafo(FILE *fp) Diretório/Arquivo: grafos.h	Descrição: Chama: Chamado por:
Procedimento: (102) grafo::item *GetItem(int id)	Descrição:

Diretório/Arquivo: grafos.h	Chama: Chamado por:
Procedimento: (103) grafo::ligacao *GetLink(int id)	Descrição:
Diretório/Arquivo: grafos.h	Chama: Chamado por:
Procedimento: (105) int grafo::ExisteItemFilho(int id1, int id2)	Descrição:
Diretório/Arquivo: grafos.h	Chama: Chamado por:
Procedimento: (107) int grafo::DestroiItem(int id)	Descrição:
Diretório/Arquivo: grafos.h	Chama: Chamado por:
Procedimento: (108) int grafo::RetiraItem(item *item1)	Descrição:
Diretório/Arquivo: grafos.h	Chama: Chamado por:
Procedimento: (110) void grafo::AtualizaGrafo()	Descrição:
Diretório/Arquivo: grafos.h	Chama: Chamado por:
Procedimento: (111) void grafo::Atualiza(item *item1, int topico)	Descrição:
Diretório/Arquivo: grafos.h	Chama: Chamado por:
Procedimento: (112) int grafo::RetiraTopico(item *topico)	Descrição:
Diretório/Arquivo: grafos.h	Chama: Chamado por:
Procedimento: (114) int grafo::DestroiLigacao(int id)	Descrição:
Diretório/Arquivo: grafos.h	Chama: Chamado por:
Procedimento: (115) int grafo::RetiraLigacao(ligacao *link)	Descrição:
Diretório/Arquivo: grafos.h	Chama: Chamado por:
Procedimento: (121) int grafo::PrimeiroItemGrafo()	Descrição:
Diretório/Arquivo: grafos.h	Chama: Chamado por:
Procedimento: (128) int grafo::ProximoItemGrafo()	Descrição:
Diretório/Arquivo: grafos.h	Chama: Chamado por:
Procedimento: (138) int grafo::PrimeiroTopicoGrafo()	Descrição:
Diretório/Arquivo: grafos.h	Chama: Chamado por:
Procedimento: (145) int grafo::ProximoTopicoGrafo()	Descrição:
Diretório/Arquivo: grafos.h	Chama: Chamado por:
Procedimento:	Descrição:

(155) int grafo::PrimeiroLinkGrafo() Diretório/Arquivo: grafos.h	Chama: Chamado por:
Procedimento: (162) int grafo::ProximoLinkGrafo() Diretório/Arquivo: grafos.h	Descrição: Chama: Chamado por:
Procedimento: (170) int grafo::NItens() Diretório/Arquivo: grafos.h	Descrição: Chama: Chamado por:
Procedimento: (171) int grafo::NLinks() Diretório/Arquivo: grafos.h	Descrição: Chama: Chamado por:
Procedimento: (172) int grafo::NTopicos() Diretório/Arquivo: grafos.h	Descrição: Chama: Chamado por:
Procedimento: (173) int grafo::NConceitos() Diretório/Arquivo: grafos.h	Descrição: Chama: Chamado por:
Procedimento: (174) int grafo::Nresultados() Diretório/Arquivo: grafos.h	Descrição: Chama: Chamado por:
Procedimento: (175) int grafo::NExercicios() Diretório/Arquivo: grafos.h	Descrição: Chama: Chamado por:
Procedimento: (176) int grafo::NExemplos() Diretório/Arquivo: grafos.h	Descrição: Chama: Chamado por:
Procedimento: (177) int grafo::NBugs() Diretório/Arquivo: grafos.h	Descrição: Chama: Chamado por:
Procedimento: (178) void grafo::MudaArquivo(char *nome) Diretório/Arquivo: grafos.h	Descrição: Chama: Chamado por:
Procedimento: (179) char grafo::*GetArquivo() Diretório/Arquivo: grafos.h	Descrição: Chama: Chamado por:

Sistema: SASHE	
Arquivo: Sashe\Hyperprop\ grafov.w.cpp	
Procedimento: (28) int Pilha::pop(char *string) Diretório/Arquivo: grafov.w.cpp	Descrição: Chama: Chamado por: Grafov.w.cpp\ (996) CGrafoView::OnAtualizar()
Procedimento: (35) int Pilha::push(char *string) Diretório/Arquivo: grafov.w.cpp	Descrição: Chama: Chamado por: Grafov.w.cpp\ (1075) void CGrafoView::OnFileOpen() Grafov.w.cpp\ (1140) void CGrafoView::OnLButtonDown(UINT nFlags,

	Cpoint point) Grafov.w.cpp\ (1220) void CGrafoView::OnRButtonDown(UINT nFlags, Cpoint point)
Procedimento: (70) CGrafoView::CGrafoView() Diretório/Arquivo: grafov.w.cpp	Descrição: Chama: (82) aninhamento.inicializar() Chamado por:
Procedimento: (97) CGrafoView::~CGrafoView() Diretório/Arquivo: grafov.w.cpp	Descrição: Chama: (102) m_pToolTipPopup->DestroyWindow() (107) m_pFont->DeleteObject() Chamado por:
Procedimento: (114) CGrafoView::Create(LPCSTR lpszClassName, LPCSTR lpszWindowName, DWORD dwStyle, const RECT& rect, CWnd* pParentWnd, UINT nID, CcreateContext* pContext) Diretório/Arquivo: grafov.w.cpp	Descrição: Chama: (117) CscrollView::Create(lpszClassName, lpszWindowName, dwStyle, rect, pParentWnd, nID, pContext) (120) CtipTop(this) (131) Cfont() Chamado por:
Procedimento: (154) BOOL CGrafoView::SetFont(const LOGFONT* pLogFont) Diretório/Arquivo: grafov.w.cpp	Descrição: Chama: (160) m_pFont->DeleteObject() (164) m_pFont-> >CreateFontIndirect(pLogFont) Chamado por: Grafov.w.cpp\ (114) CGrafoView::Create(LPCSTR lpszClassName, LPCSTR lpszWindowName, DWORD dwStyle, const RECT& rect, CWnd* pParentWnd, UINT nID, CcreateContext* pContext)
Procedimento: (173) void CGrafoView::OnDraw(CDC* pDC) Diretório/Arquivo: grafov.w.cpp	Descrição: Chama: (175) GetDocument() (179) no_escolhido->eh_terminal() (181) no_escolhido->eh_imagem() (183) mPic->DisplayPicture (pDC) Chamado por:
Procedimento: (194) void CGrafoView::AssertValid() const Diretório/Arquivo: grafov.w.cpp	Descrição: Chama: (196) Cview::AssertValid() Chamado por:
Procedimento: (199) void CGrafoView::Dump(CDumpContext& dc) const Diretório/Arquivo: grafov.w.cpp	Descrição: Chama: (201) Cview::Dump(dc) Chamado por:
Procedimento: (204) CGrafoObj* CGrafoView::GetDocument() Diretório/Arquivo: grafov.w.cpp	Descrição: Chama:

	Chamado por: Grafovw.cpp\ (173) void CGrafoView::OnDraw(CDC* pDC) Grafovw.cpp\ (211) void CGrafoView::OnInitialUpdate() Grafovw.cpp\ (220) void CGrafoView::SetDataObject() Grafovw.cpp\ (1075) void CGrafoView::OnFileOpen()
Procedimento: (211) void CGrafoView::OnInitialUpdate() Diretório/Arquivo: grafov.w.cpp	Descrição: Chama: (213) GetDocument() (214) CScrollView::OnInitialUpdate() Chamado por:
Procedimento: (220) void CGrafoView::SetDataObject() Diretório/Arquivo: grafov.w.cpp	Descrição: Chama: (226) grafo->PrimeiroNo(contexto) (237) grafo->PrimeiroElo(contexto) (255) grafo->ProximoElo(contexto, Elo) (263) grafo->ProximoNo(contexto) (265) DeterminaNivel(temp, 1) (266) DeterminaEscopo(temp) (267) Posicao(temp) (269) grafo->PrimeiroNo(contexto) (274) grafo->ProximoNo(contexto) (283) Coordenada(100, minimo, largura, altura, espacamentoX, espacamentoY) (285) grafo->PrimeiroElo(contexto) (289) Roteamento(Elo) (302) grafo->ProximoElo(contexto, Elo) Chamado por: Grafo.cpp\ (210) extern "C" void FAR PASCAL _export ApresentaContexto(char *no) Grafovw.cpp\ (996) void CGrafoView::OnAtualizar() Grafovw.cpp\ (1140) void CGrafoView::OnLButtonDown(UINT nFlags, CPoint point) Grafovw.cpp\ (1220) void CGrafoView::OnRButtonDown(UINT nFlags, CPoint point) Grafovw.cpp\ (1442) void CGrafoView::OnNomeIdentificador()
Procedimento: (309) void CGrafoView::DeterminaNivel(No *no, int nivel) Diretório/Arquivo: grafov.w.cpp	Descrição: Esta rotina determina o nível de cada no' no grafo. Chama: (311) grafo->PrimeiroElo(contexto) (322) DeterminaNivel(NO_DESTINO(Elo), nivel+1) (323) //Elo->obtem_no_origem() (324) //Elo->obtem_no_destino() (325) grafo->ProximoElo(contexto, Elo) Chamado por: (220) void CGrafoView::SetDataObject() (309) void CGrafoView::DeterminaNivel(No *no, int nivel)
Procedimento: (331) No *CGrafoView::UltimoNoAninhado(char *s)	Descrição:

Diretório/Arquivo: grafov.w.cpp	Chama: <pre>(338) grafo->RetornaPtNo(atoi(ident))</pre> Chamado por: <pre>Grafov.w.cpp\ (25) #define NO_ORIGEM(Elo) UltimoNoAninhado(Elo- >obtem_aninhamento_origem()) Grafov.w.cpp\ (26) #define NO_DESTINO(Elo) UltimoNoAninhado(Elo- >obtem_aninhamento_destino())</pre>
Procedimento: <pre>(345) int CGrafoView::PaisNoNivelDeCima(No *no)</pre> Diretório/Arquivo: grafov.w.cpp	Descrição: Soma todos os pais de um dado no' que se localizam no nível acima de seu nível. Chama: <pre>(349) grafo->PrimeiroElo(contexto) (355) NO_DESTINO(Elo) (356) NO_ORIGEM(Elo) (358) NO_ORIGEM(Elo) (361) grafo->ProximoElo(contexto, Elo)</pre> Chamado por: <pre>Grafov.w.cpp\ (380) int CGrafoView::DeterminaEscopo(No *no) GrafoView.cpp\ (471) void CGrafoView::Posicao(No *no)</pre>
Procedimento: <pre>(380) int CGrafoView::DeterminaEscopo(No *no)</pre> Diretório/Arquivo: grafov.w.cpp	Descrição: Esta rotina determina o escopo de cada no em relação a coordenada y. Chama: <pre>(383) grafo->PrimeiroElo(contexto) (391) NO_DESTINO(Elo)->no.nivel_x (393) DeterminaEscopo(NO_DESTINO(Elo) (397) grafo->ProximoElo(contexto, Elo) (399) aisNoNivelDeCima(no)</pre> Chamado por: <pre>Grafov.w.cpp\ (220) void CGrafoView::SetDataObject() GrafoView.cpp\ (380) int CGrafoView::DeterminaEscopo(No *no)</pre>
Procedimento: <pre>(412) No *CGrafoView::FilhosNoNivelDeBaixo(No *no, int &i)</pre> Diretório/Arquivo: grafov.w.cpp	Descrição: Retorna a quantidade de filhos de um determinado no' no nível abaixo do nível. Chama: <pre>(414) grafo->PrimeiroElo(contexto) (422) NO_ORIGEM(Elo) (425) NO_DESTINO(Elo) (428) grafo->ProximoElo(contexto, Elo)</pre> Chamado por: <pre>Grafov.w.cpp\ (471) void CGrafoView::Posicao(No *no)</pre>
Procedimento: <pre>(448) int CGrafoView::SomaEscopoEsquerda(No *no, No *filho)</pre> Diretório/Arquivo: grafov.w.cpp	Descrição: Soma os escopos dos no's a esquerda do no' no mesmo nível Chama: <pre>(450) grafo->PrimeiroElo(contexto) (456) NO_ORIGEM(Elo) (458) NO_DESTINO(Elo) (461) grafo->ProximoElo(contexto, Elo)</pre> Chamado por: <pre>GrafoView.cpp\ (471) void CGrafoView::Posicao(No *no)</pre>
Procedimento: <pre>(471) void CGrafoView::Posicao(No *no)</pre> Diretório/Arquivo: grafov.w.cpp	Descrição: Calcula a posição sem o deslocamento e o espaçamento da coordenada y. Chama: <pre>(477) FilhosNoNivelDeBaixo(no, i)</pre>

	(482) PaisNoNivelDeCima(filho)) (485) Posicao(filho) (488) grafo->PrimeiroElo(contexto) (491) NO_ORIGEM(Elo) (498) Posicao(NO_DESTINO(Elo)) (500) grafo->ProximoElo(contexto, Elo) Chamado por: Grafov.w.cpp\ (220) void CGrafoView::SetDataObject() Grafov.w.cpp\ (471) void CGrafoView::Posicao(No *no)
Procedimento: (509) void CGrafoView::Coordenada(int deslx, int desly, int width, int height, int espX, int espY) Diretório/Arquivo: grafov.w.cpp	Descrição: Calcula as coordenadas x e y de cada nó no grafo. Chama: (511) grafo->PrimeiroNo(contexto) (532) grafo->ProximoNo(contexto) Chamado por: Grafov.w.cpp\ (220) void CGrafoView::SetDataObject()
Procedimento: (542) void CGrafoView::DesenhaNo(No *no) Diretório/Arquivo: grafov.w.cpp	Descrição: Desenha o retângulo e coloca o identificador de cada nó'. Chama: (549) no->ObtemNome() (550) no->ObtemNome() (553) no->ObtemNome() (558) no->obtem_identificador() (561) no->eh_contexto_usuario() (564) no->eh_texto() (567) no->eh_video() (570) no->eh_imagem() (573) no->eh_audio() (576) no->eh_trilha() Chamado por: Grafov.w.cpp\ (1032) void CGrafoView::Refresh()
Procedimento: (613) void CGrafoView::DestacaNo(No *no) Diretório/Arquivo: grafov.w.cpp	Descrição: Destaca o nó invertendo as cores do desenho. Chama: (625) Crect Rect(no->no.x-3, no-> no.y+altura-20, no->no.x+largura, no-> no.y+altura-5) (628) Crect Rect(no->no.x+7, no->no.y+4, no->no.x+largura+1, no->no.y+altura-2) Chamado por: Grafov.w.cpp\ (1032) void CGrafoView::Refresh() Grafov.w.cpp\ (1140) void CGrafoView::OnLButtonDown(UINT nFlags, CPoint point) Grafov.w.cpp\ (1220) void CGrafoView::OnRButtonDown(UINT nFlags, CPoint point)
Procedimento: (639) int CGrafoView::Caminho(int x2temp, int y2temp, int comp) Diretório/Arquivo: grafov.w.cpp	Descrição: Esta rotina verifica se existe algum nó ou arco no caminho. Retorna: 0: se encontrou algum nó ou arco em seu caminho. 1: caso contrário. Chama: (641) grafo->PrimeiroNo(contexto) (651) grafo->ProximoNo(contexto) (653) grafo->PrimeiroElo(contexto)

	(664) grafo->ProximoElo(contexto, Elo) Chamado por: Grafov.w.cpp\ (672) void CGrafoView::Roteamento(elo *Elo)
Procedimento: (672) void CGrafoView::Roteamento(elo *Elo) Diretório/Arquivo: grafov.w.cpp	Descrição: Esta rotina faz o roteamento dos arcos do grafo. Chama: (695) Caminho(Elo->arco.x1, aux, largura) (700) Caminho(Elo->arco.x1, aux1, largura) (721) grafo->PrimeiroElo(contexto) (734) grafo->ProximoElo(contexto, e) (745) Caminho(Elo->arco.x2, aux, comp) (763) Caminho(Elo->arco.x2, aux, comp) (782) Caminho(Elo->arco.x2, aux1, comp) (787) Caminho(Elo->arco.x2, aux, comp) (805) Caminho(Elo->arco.x1+espacamentoX, Elo->arco.y1, comp) (814) Caminho(Elo->arco.x1+espacamentoX, aux, comp) (824) Caminho(Elo->arco.x1+espacamentoX, aux, comp) (835) Caminho(Elo->arco.x1+espacamentoX, aux, comp) (844) Caminho(Elo->arco.x1+espacamentoX, aux, comp) Chamado por: Grafov.w.cpp\ (220) void CGrafoView::SetDataObject()
Procedimento: (857) void CGrafoView::DesenhaArco(elo *Elo) Diretório/Arquivo: grafov.w.cpp	Descrição: Esta rotina desenha os arcos do grafo. Chama: (865) Elo->obtem_origem() (869) Elo->obtem_identificador() Chamado por: Grafov.w.cpp\ (1032) void CGrafoView::Refresh()
Procedimento: (983) void CGrafoView::DestacaArco(elo *Elo) Diretório/Arquivo: grafov.w.cpp	Descrição: Esta rotina destaca os arcos do grafo. Chama: Chamado por: Grafov.w.cpp\ (1032) void CGrafoView::Refresh() Grafov.w.cpp\ (1140) void CGrafoView::OnLButtonDown(UINT nFlags, CPoint point) Grafov.w.cpp\ (1220) void CGrafoView::OnRButtonDown(UINT nFlags, CPoint point)
Procedimento: (996) void CGrafoView::OnAtualizar() Diretório/Arquivo: grafov.w.cpp	Descrição: Chama: (999) grafo->PrimeiroNo(contexto) (1002) grafo->ProximoNo(contexto) (1005) aninhamento.pop(contexto) (1011) grafo->SetTitle(contexto) Chamado por: Grafov.w.cpp\ (1075) void CGrafoView::OnFileOpen() Grafov.w.cpp\ (1220) void CGrafoView::OnRButtonDown(UINT nFlags, Cpoint point)

	Grafov.w.cpp\ (1515) void CGrafoView::OnUpdateAtualizar(CCmdUI* pCmdUI)
Procedimento: (1032) void CGrafoView::Refresh() Diretório/Arquivo: grafov.w.cpp	Descrição: Chama: (1034) grafo->PrimeiroNo(contexto) (1037) DesenhaNo(no) (1038) grafo->ProximoNo(contexto) (1040) grafo->PrimeiroElo(contexto) (1042) DesenhaArco(Elo) (1043) grafo->ProximoElo(contexto, Elo) (1045) DestacaNo(no_escolhido) (1046) DestacaArco(elo_escolhido) Chamado por: Grafo.cpp\ (210) extern "C" void FAR PASCAL _export ApresentaContexto(char *no) Grafov.w.cpp\ (173) void CGrafoView::OnDraw(CDC* pDC) Grafov.w.cpp\ (996) void CGrafoView::OnAtualizar() Grafov.w.cpp\ (1140) void CGrafoView::OnLButtonDown(UINT nFlags, CPoint point) Grafov.w.cpp\ (1220) void CGrafoView::OnRButtonDown(UINT nFlags, CPoint point) Grafov.w.cpp\ (1442) void CGrafoView::OnNomeIdentificador()
Procedimento: (1049) void CGrafoView::OnFileNew() Diretório/Arquivo: grafov.w.cpp	Descrição: Chama: (1053) grafo->LimparHiperbase() (1054) aninhamento.inicializar() (1055) Invalidate() Chamado por: Grafo.h\ (24) void OpenWindow() Grafov.w.cpp\ (1075) void CGrafoView::OnFileOpen()
Procedimento: (1059) void CGrafoView::OnFileSave() Diretório/Arquivo: grafov.w.cpp	Descrição: Chama: (1066) RevFileDialog.DoModal() (1069) RevFileDialog.GetPathName() (1070) // grafo->Salvar(Arquivo) Chamado por: (52) ON_COMMAND(ID_FILE_SAVE, OnFileSave)
Procedimento: (1075) void CGrafoView::OnFileOpen() Diretório/Arquivo: grafov.w.cpp	Descrição: Chama: (1079) grafo = GetDocument() (1080) OnFileNew() (1083) RevFileDialog.DoModal() (1086) grafo = GetDocument() (1087) RevFileDialog.GetPathName() (1088) grafo->Recuperar(Arquivo) (1090) grafo->Hiperbase(contexto) (1091) aninhamento.push(contexto) (1092) OnAtualizar() Chamado por: (53) ON_COMMAND(ID_FILE_OPEN, OnFileOpen)
Procedimento:	Descrição:

(1098) void CGrafoView::OnMouseMove(UINT nFlags, CPoint point) Diretório/Arquivo: grafov.w.cpp	Chama: (1101) OnPrepareDC(&dc) (1105) CView::OnMouseMove(nFlags, point) (1108) GetMouseOverTip(point) (1114) m_pToolTipPopup->HideCurrentTip() (1124) dc.LPtoDP(&point) (1126) ClientToScreen(&point) (1130) SetTimer(1, m_nWait, NULL) (1137) m_pToolTipPopup->HideCurrentTip() Chamado por:
Procedimento: (1140) void CGrafoView::OnLButtonDown(UINT nFlags, CPoint point) Diretório/Arquivo: grafov.w.cpp	Descrição: Chama: (1143) OnPrepareDC(&dc) (1144) dc.DPtoLP(&point) (1146) grafo->PrimeiroNo(contexto) (1151) DestacaNo(no_escolhido) (1154) DestacaArco(elo_escolhido) (1158) DestacaNo(no) (1159) no->eh_contexto_usuario() (1159) no->eh_trilha (1163) grafo->PrimeiroNo(contexto) (1166) grafo->ProximoNo(contexto) (1168) aninhamento.push(contexto) (1169) no->ObtemNome() (1170) grafo->SetTitle(contexto) (1171) SetDataObject() (1172) Invalidate() (1173) Refresh() (1177) grafo->ProximoNo(contexto) (1179) grafo->PrimeiroElo(contexto) (1186) DestacaArco(elo_escolhido) (1189) DestacaNo(no_escolhido) (1193) DestacaArco(Elo) (1194) Elo->eh_simultaneo() (1195) Elo->obtem_origem() (1195) Elo->obtem_destino(); (1197) Elo->obtem_origem() (1197) Elo->obtem_destino(); (1201) grafo->ProximoElo(contexto, Elo) Chamado por:
Procedimento: (1210) void CGrafoView::OnLButtonUp(UINT nFlags, CPoint point) Diretório/Arquivo: grafov.w.cpp	Descrição: Chama: (1212) CView::OnLButtonUp(nFlags, point) Chamado por:
Procedimento: (1220) void CGrafoView::OnRButtonDown(UINT nFlags, CPoint point) Diretório/Arquivo: grafov.w.cpp	Descrição: Chama: (1223) OnPrepareDC(&dc) (1224) dc.DPtoLP(&point) (1226) grafo->PrimeiroNo(contexto) (1231) DestacaNo(no_escolhido) (1234) DestacaArco(elo_escolhido) (1238) DestacaNo(no) (1239) no->eh_contexto_usuario() (1239) no->eh_trilha

	<pre> (1243) grafo->PrimeiroNo(contexto) (1246) grafo->ProximoNo(contexto) (1248) aninhamento.push(contexto) (1252) grafo->SetTitle(contexto) (1253) no->eh_terminal() (1255) no->eh_imagem() (1259) noi->recupera_nome_arquivo() (1261) new CPics(arquivo) (1268) OnAtualizar() (1280) no->eh_video() (1284) nov->recupera_nome_arquivo() (1287) mAvi.Open (this, arquivo) (1293) OnAtualizar() (1305) mAvi.Start () (1310) OnAtualizar() (1314) no->eh_audio() (1317) CWave::GetNumWaveDevices () (1322) noa->recupera_nome_arquivo() (1323) new CWave (arquivo) (1329) pWave->Play () (1335) OnAtualizar() (1344) OnAtualizar() (1348) no->eh_texto() (1354) not->recupera_nome_arquivo() (1366) OnAtualizar() (1385) OnAtualizar() (1394) OnAtualizar() (1401) SetDataObject() (1402) Refresh() (1404) Invalidate() (1408) grafo->ProximoNo(contexto) (1410) grafo->PrimeiroElo(contexto) (1417) DestacaArco(elo_escolhido) (1420) DestacaNo(no_escolhido) (1424) DestacaArco(Elo) (1425) Elo->eh_simultaneo() (1426) Elo->obtem_origem() (1426) Elo->obtem_destino(); (1428) Elo->obtem_origem() (1428) Elo->obtem_destino(); (1432) grafo->ProximoElo(contexto, Elo) Chamado por: </pre>
Procedimento: (1437) void CGrafoView::OnRButtonUp(UINT nFlags, CPoint point) Diretório/Arquivo: grafov.cpp	Descrição: Chama: (1439) Cview::OnRButtonUp(nFlags, point) Chamado por:
Procedimento: (1442) void CGrafoView::OnNomeIdentificador() Diretório/Arquivo: grafov.cpp	Descrição: Chama: (1460) Invalidate() (1461) SetDataObject() (1462) Refresh() Chamado por: Grafovw.cpp\ (59) ON_COMMAND(ID_EDITAR_NOMEUIDENTIFICADOR , OnNomeIdentificador)
Procedimento: (1465) void CGrafoView::OnTimer(UINT nIDEvent) Diretório/Arquivo: grafov.cpp	Descrição: Chama: (1475) AfxGetMainWnd() (1476) pMain->IsWindowEnabled() (1476) !pMain->IsIconic()

	<pre> (1481) ::GetCursorPos(&ptCur) (1482) ScreenToClient(&ptCur) (1484) m_pToolTipPopup- >ShowTip(m_nShowTip, m_nShowX, m_nShowY, m_szItemName) (1488) KillTimer(nIDEvent) Chamado por: </pre>
Procedimento: (1494) UINT CgrafoView::GetMouseOverTip(CPoint point) Diretório/Arquivo: grafov.w.cpp	Descrição: Get the tip the mouse is over, if any (0 if none) Chama: (1501) grafo->PrimeiroNo(contexto) (1506) no->ObtemNome() (1507) no->obtem_identificador() (1509) grafo->ProximoNo(contexto) Chamado por: Grafov.w.cpp\ (1098) void CgrafoView::OnMouseMove(UINT nFlags, CPoint point)
Procedimento: (1515) void CgrafoView::OnUpdateAtualizar(CCcmdU I* pCmdUI) Diretório/Arquivo: grafov.w.cpp	Descrição: Chama: (1517) aninhamento.number() (1518) AfxGetApp()->m_pMainWnd (1519) pMainWnd- >m_wndToolBar.m_comboBox.GetCurSel() (1520) pMainWnd- >m_wndToolBar.m_comboBox.GetCurSel() (1522) pMainWnd- >m_wndToolBar.m_comboBox.GetCurSel() (1524) OnAtualizar() Chamado por: Grafov.w.cpp\ (62) ON_UPDATE_COMMAND_UI(ID_ATUALIZAR, OnUpdateAtualizar)
Procedimento: (1528) void CgrafoView::OnUpdateEditNomIdent(CC mdUI* pCmdUI) Diretório/Arquivo: grafov.w.cpp	Descrição: Chama: (1530) mAvi.mOpen Chamado por: Grafov.w.cpp\ (63) ON_UPDATE_COMMAND_UI(ID_EDITAR NOME OU IDE NTIFICADOR, OnUpdateEditNomIdent)

Sistema: SASHE**Arquivo:** Sashe\Hyperprop\ grafov.w.h

Procedimento: (21) void pilha::inicializar() Diretório/Arquivo: grafov.w.h	Descrição: Chama: Chamado por: Grafov.w.cpp\ (70) CgrafoView::CgrafoView() Grafov.w.cpp\ (1049) void CgrafoView::OnFileNew()
Procedimento: (24) int pilha::number() Diretório/Arquivo: grafov.w.h	Descrição: Chama: Chamado por: Grafov.w.cpp\ (1515) void CgrafoView::OnUpdateAtualizar(CCcmdUI* pCmdUI)
Procedimento:	Descrição:

(48) CgrafoView::CgrafoObj* GetDocument() Diretório/Arquivo: grafov.w.h	Protótipo da função implementada em Grafov.w.cpp Chama: Chamado por:
Procedimento: (49) void CgrafoView::OnInitialUpdate() Diretório/Arquivo: grafov.w.h	Descrição: Protótipo da função implementada em Grafov.w.cpp Chama: Chamado por:
Procedimento: (73) void CgrafoView::SetDataObject() Diretório/Arquivo: grafov.w.h	Descrição: Protótipo da função implementada em Grafov.w.cpp Chama: Chamado por:
Procedimento: (75) void CgrafoView::Refresh() Diretório/Arquivo: grafov.w.h	Descrição: Protótipo da função implementada em Grafov.w.cpp Chama: Chamado por:
Procedimento: (94) virtual CgrafoView::~~CgrafoView() Diretório/Arquivo: grafov.w.h	Descrição: Protótipo da função implementada em Grafov.w.cpp Chama: Chamado por:
Procedimento: (101) virtual void CgrafoView::AssertValid() const Diretório/Arquivo: grafov.w.h	Descrição: Protótipo da função implementada em Grafov.w.cpp Chama: Chamado por:
Procedimento: (109) void CgrafoView::OnAtualizar() Diretório/Arquivo: grafov.w.h	Descrição: Protótipo da função implementada em Grafov.w.cpp Chama: Chamado por:
Procedimento: (110) void CgrafoView::OnFileSave() Diretório/Arquivo: grafov.w.h	Descrição: Protótipo da função implementada em Grafov.w.cpp Chama: Chamado por:
Procedimento: (111) void CgrafoView::OnFileOpen() Diretório/Arquivo: grafov.w.h	Descrição: Protótipo da função implementada em Grafov.w.cpp Chama: Chamado por:
Procedimento: (117) void CgrafoView::OnNomeIdentificador() Diretório/Arquivo: grafov.w.h	Descrição: Protótipo da função implementada em Grafov.w.cpp Chama: Chamado por:
Procedimento: (118) void CgrafoView::OnFileNew() Diretório/Arquivo: grafov.w.h	Descrição: Protótipo da função implementada em Grafov.w.cpp Chama: Chamado por:
Sistema: SASHE	
Arquivo: Sashe\Hyperprop\ hip.cpp	
Procedimento: (19) hipwindow::hipwindow() Diretório/Arquivo: hip.cpp	Descrição: Chama: Chamado por:
Procedimento: (26) No * hipwindow::Retorna(char *nome) Diretório/Arquivo: hip.cpp	Descrição: Chama: (28) nos hiperbase.primeiro()

```
(32) n->ObtemNome()  
(33) nos_hiperbase.proximo()  
Chamado por:  
Tutor.cpp\ (78) int  
tutor::segunda_estrategia(contexto *pai)  
Hip.cpp\ (120) int  
hipwindow::recupera_hiperbase(char  
*nome)  
Elo.cpp\ (56) int elo::recupera_elo(FILE  
*fp)  
Hipger.cpp\ (66) extern "C" int FAR  
PASCAL _export cria_base_privada(char  
*nome)  
Hipger.cpp\ (161) extern "C" int FAR  
PASCAL _export cria_contexto(char *nome)  
Hipger.cpp\ (174) extern "C" int FAR  
PASCAL _export cria_trilha(char *nome)  
Hipger.cpp\ (187) extern "C" int FAR  
PASCAL _export cria_texto(char *nome)  
Hipger.cpp\ (200) extern "C" int FAR  
PASCAL _export cria_video(char *nome)  
Hipger.cpp\ (213) extern "C" int FAR  
PASCAL _export cria_audio(char *nome)  
Hipger.cpp\ (226) extern "C" int FAR  
PASCAL _export cria_imagem(char *nome)  
Hipger.cpp\ (239) extern "C" int FAR  
PASCAL _export cria_executavel(char  
*nome)  
Hipger.cpp\ (252) extern "C" int FAR  
PASCAL _export cria_tabela(char *nome)  
Hipger.cpp\ (263) extern "C" int FAR  
PASCAL _export insere_no_contexto(char  
*nome_cont, char *nome_no)  
Hipger.cpp\ (288) extern "C" int FAR  
PASCAL _export insere_na_trilha(char  
*nome_trilha, char *nome_no, char  
*no_liberdade)  
Hipger.cpp\ (317) extern "C" int FAR  
PASCAL _export insere_na_posicao(char  
*nome_trilha, char *nome_no, char  
*no_liberdade, int posicao)  
Hipger.cpp\ (339) extern "C" int FAR  
PASCAL _export retira_do_contexto(char  
*nome_contexto, char *nome_no)  
Hipger.cpp\ (359) extern "C" int FAR  
PASCAL _export retira_da_trilha(char  
*nome_trilha, char *nome_no liberdade)  
Hipger.cpp\ (376) extern "C" int FAR  
PASCAL _export limpa_trilha(char  
*nome_trilha)  
Hipger.cpp\ (390) extern "C" int FAR  
PASCAL _export  
quantos_contextos_pertence(char *nome)  
Hipger.cpp\ (400) extern "C" int FAR  
PASCAL _export  
primeiro_contexto_pai(char *nome, char  
*nome_pai)  
Hipger.cpp\ (413) extern "C" int FAR  
PASCAL _export ultimo_contexto_pai(char  
*nome, char *nome_pai)  
Hipger.cpp\ (426) extern "C" int FAR  
PASCAL _export proximo_contexto_pai(char  
*nome, char *nome_pai)  
Hipger.cpp\ (442) extern "C" int FAR  
PASCAL _export primeiro_no_contexto(char  
*nome_cont, char *nome_no)  
Hipger.cpp\ (458) extern "C" int FAR
```

```
PASCAL_export proximo_no_contexto(char
*nome_cont, char *nome_no)
Hipger.cpp\ (472) extern "C" int FAR
PASCAL_export primeiro_no_trilha(char
*nome_trilha, char *nome_no, char
*nome_liberdade)
Hipger.cpp\ (483) extern "C" int FAR
PASCAL_export proximo_no_trilha(char
*nome_trilha, char *nome_no, char
*nome_liberdade)
Hipger.cpp\ (494) extern "C" int FAR
PASCAL_export ultimo_no_trilha(char
*nome_trilha, char *nome_no, char
*nome_liberdade)
Hipger.cpp\ (505) extern "C" int FAR
PASCAL_export anterior_no_trilha(char
*nome_trilha, char *nome_no, char
*nome_liberdade)
Hipger.cpp\ (516) No *FAR PASCAL_export
primeiro_ponteiro_no(char *nome_cont)
Hipger.cpp\ (533) No *FAR PASCAL_export
proximo_ponteiro_no(char *nome_cont)
Hipger.cpp\ (557) extern "C" int FAR
PASCAL_export
recupera_nome_arquivo_video(char
*nome_video, char *arq_video)
Hipger.cpp\ (569) extern "C" int FAR
PASCAL_export
insere_nome_arquivo_video(char
*nome_video, char *arq_video)
Hipger.cpp\ (581) extern "C" int FAR
PASCAL_export
recupera_nome_arquivo_texto(char
*nome_texto, char *arquivo)
Hipger.cpp\ (593) extern "C" int FAR
PASCAL_export
insere_nome_arquivo_texto(char
*nome_texto, char *arquivo)
Hipger.cpp\ (605) extern "C" int FAR
PASCAL_export
recupera_nome_arquivo_audio(char
*nome_audio, char *arquivo)
Hipger.cpp\ (617) extern "C" int FAR
PASCAL_export
insere_nome_arquivo_audio(char
*nome_audio, char *arquivo)
Hipger.cpp\ (629) extern "C" int FAR
PASCAL_export
recupera_nome_arquivo_imagem(char
*nome_imagem, char *arquivo)
Hipger.cpp\ (641) extern "C" int FAR
PASCAL_export
insere_nome_arquivo_imagem(char
*nome_imagem, char *arquivo)
Hipger.cpp\ (654) extern "C" int FAR
PASCAL_export
recupera_nome_arquivo_executavel(char
*nome_executavel, char *arquivo)
Hipger.cpp\ (666) extern "C" int FAR
PASCAL_export
insere_nome_arquivo_executavel(char
*nome_executavel, char *arquivo)
Hipger.cpp\ (678) extern "C" int FAR
PASCAL_export
recupera_nome_arquivo_tabela(char
*nome_tabela, char *arquivo)
Hipger.cpp\ (690) extern "C" int FAR
```

```
PASCAL_export
insere_nome_arquivo_tabela(char
*nome_tabela, char *arquivo)
Hipger.cpp\ (712) extern "C" int FAR
PASCAL_export tipo_no(char *nome)
Hipger.cpp\ (731) extern "C" int FAR
PASCAL_export retira_da_hiperbase(char
*nome_no)
Hipger.cpp\ (866) extern "C" int FAR
PASCAL_export muda_atributos(char
*nome_velho, char *nome_novo, int
num_estado)
Hipger.cpp\ (890) extern "C" int FAR
PASCAL_export muda_tipo(char *nome_no,
char *tipo)
Hipger.cpp\ (908) extern "C" int FAR
PASCAL_export recupera_tipo(char
*nome_no, char *tipo)
Hipger.cpp\ (927) extern "C" int FAR
PASCAL_export muda_funcao_didatica(char
*nome_no, char *categoria)
Hipger.cpp\ (945) extern "C" int FAR
PASCAL_export
recupera_funcao_didatica(char *nome_no,
char *categoria)
Hipger.cpp\ (964) extern "C" int FAR
PASCAL_export muda_assunto(char
*nome_no, char *assunto)
Hipger.cpp\ (979) extern "C" int FAR
PASCAL_export recupera_assunto(char
*nome_no, char *assunto)
Hipger.cpp\ (993) extern "C" int FAR
PASCAL_export muda_dificuldade(char
*nome_no, int nivel)
Hipger.cpp\ (1009) extern "C" int FAR
PASCAL_export recupera_dificuldade(char
*nome_no, int *nivel)
Hipger.cpp\ (1028) extern "C" int FAR
PASCAL_export muda_palavra_chave(char
*nome_no, int pos, char *palavra)
Hipger.cpp\ (1046) extern "C" int FAR
PASCAL_export
recupera_palavra_chave(char *nome_no,
int pos, char *palavra)
Hipger.cpp\ (1067) extern "C" int FAR
PASCAL_export regras_tutoriais(char
*no_atual, char *escopo, char
*historico, int dificuldade, char
*no_result, char *lib_result, int
*revisto)
Hipger.cpp\ (1121) extern "C" int FAR
PASCAL_export cria_versao(char
*nome_no, char *nome_no_versao)
Hipger.cpp\ (1154) extern "C" int FAR
PASCAL_export muda_vcorrente(char
*nome_cv, char *nome_no)
Hipger.cpp\ (1181) extern "C" int FAR
PASCAL_export cria_ancora_texto(char
*nome_no, int ident, char *palavra, int
inicio, int fim)
Hipger.cpp\ (1203) extern "C" int FAR
PASCAL_export cria_ancora_contexto(char
*nome_no, int ident, char *no_ancorado)
Hipger.cpp\ (1227) extern "C" int FAR
PASCAL_export retira_ancora(char *nome,
int ident)
Hipger.cpp\ (1246) extern "C" int FAR
```

	<pre> PASCAL _export recupera_identificador_interno(char *nome, int ide) Hipger.cpp\ (1260) extern "C" int FAR PASCAL _export muda_ancora_texto(char *nome, int velho_ide, int novo_ide, char *palavra, int inicio, int fim) Hipger.cpp\ (1279) extern "C" int FAR PASCAL _export muda_ancora_contexto(char *nome, int velho_ide, int novo_ide, char *no_ancorado) Hipger.cpp\ (1297) extern "C" int FAR PASCAL _export numero_de_ancoras(char *nome) Hipger.cpp\ (1311) extern "C" int FAR PASCAL _export recupera_nome_ancora(char *nome, int id_ancora, char *nome_ancora) Hipger.cpp\ (1372) extern "C" int FAR PASCAL _export primeira_ancora(char *nome, char *nome_ancora) Hipger.cpp\ (1425) extern "C" int FAR PASCAL _export proxima_ancora(char *nome, char *nome_ancora) Hipger.cpp\ (1489) extern "C" int FAR PASCAL _export ultima_ancora(char *nome, char *nome_ancora) Hipger.cpp\ (1517) extern "C" int FAR PASCAL _export cria_elo(char *nome_contexto, int id_elo, char *origem, int anc_origem, char *destino, int anc_destino, int simultaneo, int mostra_conteudo) Hipger.cpp\ (1560) extern "C" int FAR PASCAL _export muda_elo(char *nome_contexto, int id_elo, char *origem, int anc_origem, char *destino, int anc_destino, int simultaneo, int mostra_conteudo) Hipger.cpp\ (1608) extern "C" int FAR PASCAL _export retira_elo(char *nome_contexto, int id_elo) Hipger.cpp\ (1627) extern "C" int FAR PASCAL _export numero_de_elos(char *nome_contexto) Hipger.cpp\ (1647) extern "C" int FAR PASCAL _export recupera_elo(char *nome_contexto, int id_elo, char *id_no_orig, int *id_anc_orig, char *id_no_dest, int *id_anc_dest, int *simultaneo, int *mostra_conteudo) Hipger.cpp\ (1672) extern "C" int FAR PASCAL _export primeiro_elo(char *nome, int *id) Hipger.cpp\ (1687) extern "C" int FAR PASCAL _export proximo_elo(char *nome, int *id) Hipger.cpp\ (1697) elo *FAR PASCAL _export primeiro_ponteiro_elo(char *nome_contexto) Hipger.cpp\ (1714) elo *FAR PASCAL _export proximo_ponteiro_elo(char *nome_contexto, elo *pelo) </pre>
Procedimento: (40) No * hipwindow::RetornaBase(char *nome) Diretório/Arquivo: hip.cpp	Descrição: Chama: (43) base_ativa->Retorna(nome)

	Chamado por: Hipger.cpp\ (288) extern "C" int FAR PASCAL _export insere_na_trilha(char *nome_trilha, char *nome_no, char *no_liberdade) Hipger.cpp\ (317) extern "C" int FAR PASCAL _export insere_na_posicao(char *nome_trilha, char *nome_no, char *no_liberdade, int posicao)
Procedimento: (49) No * hipwindow::RetornaNo(int Id) Diretório/Arquivo: hip.cpp	Descrição: Chama: (51) nos_hiperbase.primeiro() (55) n->obtem_identificador() (56) nos_hiperbase.proximo() Chamado por: Hipger.cpp\ (144) extern "C" int FAR PASCAL _export retorna_nome(int ident, char *nome) Hipger.cpp\ (550) No *FAR PASCAL _export retorna_ponteiro_no(int id) Hipger.cpp\ (731) extern "C" int FAR PASCAL _export retira_da_hiperbase(char *nome_no) Hipger.cpp\ (1311) extern "C" int FAR PASCAL _export recupera_nome_ancora(char *nome, int id_ancora, char *nome_ancora) Hipger.cpp\ (1372) extern "C" int FAR PASCAL _export primeira_ancora(char *nome, char *nome_ancora) Hipger.cpp\ (1425) extern "C" int FAR PASCAL _export proxima_ancora(char *nome, char *nome_ancora) Hipger.cpp\ (1517) extern "C" int FAR PASCAL _export cria_elo(char *nome_contexto, int id_elo, char *origem, int anc_origem, char *destino, int anc_destino, int simultaneo, int mostra_conteudo) Hipger.cpp\ (1560) extern "C" int FAR PASCAL _export muda_elo(char *nome_contexto, int id_elo, char *origem, int anc_origem, char *destino, int anc_destino, int simultaneo, int mostra_conteudo) List_nó.cpp\ (231) int lista_nos::recupera(FILE *fp) No.cpp\ (142) int No::criaversao(char *nome_no) Trilha.cpp\ (251) int trilha::primeiro_no_trilha(char *nome, char *nome_liberdade) Trilha.cpp\ (278) int trilha::proximo_no_trilha(char *nome, char *nome_liberdade) Trilha.cpp\ (305) int trilha::anterior_no_trilha(char *nome, char *nome_liberdade) Trilha.cpp\ (332) int trilha::ultimo_no_trilha(char *nome, char *nome_liberdade)
Procedimento: (63) int hipwindow::RetornaIdent(char *nome) Diretório/Arquivo: hip.cpp	Descrição: Chama: (65) nos_hiperbase.primeiro()

	<pre> (69) n->ObtemNome() (69) n->obtem_identificador() (70) nos_hiperbase.proximo() Chamado por: Hipger.cpp\ (137) extern "C" int FAR PASCAL _export retorna_identificador(char *nome) </pre>
Procedimento: (78) int hipwindow::ativa_hiperbase(char *nome) Diretório/Arquivo: hip.cpp	Descrição: Torna a hipwindow ativa. Chama: (80) nos_hiperbase.primeiro() (84) n->ObtemNome() (84) n->eh_base_privada() (89) nos_hiperbase.proximo() Chamado por: Hipger.cpp\ (66) extern "C" int FAR PASCAL _export cria_base_privada(char *nome) Hipger.cpp\ (78) extern "C" int FAR PASCAL _export ativa_hiperbase(char *nome)
Procedimento: (97) int hipwindow::grava_hiperbase(char *nome) Diretório/Arquivo: hip.cpp	Descrição: 1: Erro na abertura do arquivo. Chama: (103) base_ativa->ObtemNome() (105) base_ativa->grava(fp) Chamado por: Hipger.cpp\ (88) extern "C" int FAR PASCAL _export salvar(char *nome)
Procedimento: (120) int hipwindow::recupera_hiperbase(char *nome) Diretório/Arquivo: hip.cpp	Descrição: 1: Já existe outra base privada com este nome carregado na memória. 2: Erro na leitura do arquivo. 3: Erro de alocação de memória. Chama: (126) fscanf(fp, "Base(%s,", string) (128) Retorna(string) (129) new baseprivada(string) (131) base_ativa->recupera(fp) Chamado por: Hipger.cpp\ (100) extern "C" int FAR PASCAL _export recuperar(char *nome)
Procedimento: (142) int hipwindow::primeiro_no_hiperbase(char *nome) Diretório/Arquivo: hip.cpp	Descrição: recupera o nome do primeiro noh armazenado na lista da hipwindow Chama: (144) nos_hiperbase.primeiro() (146) primeiro->ObtemNome() Chamado por: Grafoob.h\ (25) int Hiperbase(char *nome) Hipger.cpp\ (115) extern "C" int FAR PASCAL _export primeiro_no_hiperbase(char *nome)
Procedimento: (154) int hipwindow::proximo_no_hiperbase(char *nome) Diretório/Arquivo: hip.cpp	Descrição: recupera o nome do proximo noh armazenado na lista da hipwindow Chama: (156) nos_hiperbase.proximo() (158) proximo->ObtemNome() Chamado por:

	Hipger.cpp\ (122) extern "C" int FAR PASCAL _export proximo_no_hiperbase(char *nome)
Procedimento: (165) int hipwindow::obtem_identificador_ulti mo_no() Diretório/Arquivo: hip.cpp	Descrição: recupera o indice do ultimo noh armazenado na lista da hipwindow Chama: (167) nos_hiperbase.ultimo() (169) ultimo->obtem_identificador() Chamado por: No.cpp\ (26) No::No(char *nome_no) : entidade()
Procedimento: (177) void hipwindow::atualiza_base_ativa(No *pno) Diretório/Arquivo: hip.cpp	Descrição: Atualiza a base ativa, ou seja, verifica se o nó jah estah na base ativa. Se ele nao estiver, ele serah inserido. Se ele for um no contexto tambem serah inserido todos os nos pertencentes a ele e assim recursivamente. Chama: (179) pertence_base_ativa(pno) (179) pno->eh_base_privada() (182) insere_base_ativa(pno) (183) pno->primeiro_contexto_pai() (183) pno->proximo_contexto_pai() (184) atualiza_base_ativa(I) (185) pno->eh_contexto() (185) pno->eh_trilha() (188) pno1->primeiro_no() (188) I = pno1->proximo_no() (189) atualiza_base_ativa(I) Chamado por: Hip.cpp\ (177) void hipwindow::atualiza_base_ativa(No *pno) Hipger.cpp\ (263) extern "C" int FAR PASCAL _export insere_no_contexto(char *nome_cont, char *nome_no) Hipger.cpp\ (288) extern "C" int FAR PASCAL _export insere_na_trilha(char *nome_trilha, char *nome_no, char *no liberdade)

Sistema: SASHE	
Arquivo: Sashe\Hyperprop\ hip.h	
Procedimento: (26) int insere_hipwindow(No *pontno) Diretório/Arquivo: hip.h	Descrição: armazena o no na hiperbase Chama: (26) nos_hiperbase.insere(pontno) Chamado por: No.cpp\ (26) No::No(char *nome_no) : entidade() No.cpp\ (43) No::No(int id) : entidade()
Procedimento: (27) int insere_base_ativa(No *pontno) Diretório/Arquivo: hip.h	Descrição: armazena o no na base privada que está ativa Chama: (27) base_ativa->insere_contexto(pontno) Chamado por: Hip.cpp\ (177) void hipwindow::atualiza_base_ativa(No *pno) Hipger.cpp\ (161) extern "C" int FAR PASCAL _export cria_contexto(char *nome) Hipger.cpp\ (174) extern "C" int FAR PASCAL _export cria_trilha(char *nome) Hipger.cpp\ (187) extern "C" int FAR

	PASCAL _export cria_texto(char *nome) Hipger.cpp\ (200) extern "C" int FAR PASCAL _export cria_video(char *nome) Hipger.cpp\ (213) extern "C" int FAR PASCAL _export cria_audio(char *nome) Hipger.cpp\ (226) extern "C" int FAR PASCAL _export cria_imagem(char *nome) Hipger.cpp\ (239) extern "C" int FAR PASCAL _export cria_executavel(char *nome) Hipger.cpp\ (252) extern "C" int FAR PASCAL _export cria_tabela(char *nome) Hipger.cpp\ (317) extern "C" int FAR PASCAL _export insere_na_posicao(char *nome_trilha, char *nome_no, char *no liberdade, int posicao)
Procedimento: (28) int retira_base_ativa(No *pontno) Diretório/Arquivo: hip.h	Descrição: Chama: (28) base_ativa->retira_contexto(pontno) Chamado por:
Procedimento: (29) int pertence_base_ativa(No *pontno) Diretório/Arquivo: hip.h	Descrição: Chama: (29) base_ativa->pertence_contexto(pontno) Chamado por: Hip.cpp\ (177) void hipwindow::atualiza_base_ativa(No *pno) Hipger.cpp\ (317) extern "C" int FAR PASCAL _export insere_na_posicao(char *nome_trilha, char *nome_no, char *no liberdade, int posicao)
Procedimento: (31) No *primeiro() Diretório/Arquivo: hip.h	Descrição: Chama: (31) nos_hiperbase.primeiro() Chamado por: Hipger.cpp\ (731) extern "C" int FAR PASCAL _export retira_da_hiperbase(char *nome no)
Procedimento: (32) No *proximo() Diretório/Arquivo: hip.h	Descrição: Chama: (32)nos_hiperbase.proximo() Chamado por: Hipger.cpp\ (731) extern "C" int FAR PASCAL _export retira_da_hiperbase(char *nome no)
Procedimento: (33) int retira(No *pno) Diretório/Arquivo: hip.h	Descrição: Chama: (33) nos_hiperbase.retira(pno) Chamado por: Hipger.cpp\ (731) extern "C" int FAR PASCAL _export retira_da_hiperbase(char *nome no)
Procedimento: (34) baseprivada *recupera_base_ativa() Diretório/Arquivo: hip.h	Descrição: Chama: Chamado por: List_no.cpp\ (182) int lista nos::grava(FILE *fp)

Sistema: SASHE	
Arquivo: Sashe\Hyperprop\ hipdll.h	
Procedimento: (14) int FAR PASCAL cria_base_privada(char *) Diretório/Arquivo: hipdll.h	Descrição: Protótipo da função de DLL Chama: Chamado por:
Procedimento: (15) int FAR PASCAL ativa_hiperbase(char *) Diretório/Arquivo: hipdll.h	Descrição: Protótipo da função de DLL Chama: Chamado por:
Procedimento: (16) int FAR PASCAL salvar(char *) Diretório/Arquivo: hipdll.h	Descrição: Protótipo da função de DLL Chama: Chamado por:
Procedimento: (17) int FAR PASCAL recuperar(char *) Diretório/Arquivo: hipdll.h	Descrição: Protótipo da função de DLL Chama: Chamado por:
Procedimento: (18) int FAR PASCAL limpar_hiperbase() Diretório/Arquivo: hipdll.h	Descrição: Protótipo da função de DLL Chama: Chamado por:
Procedimento: (19) int FAR PASCAL retorna_identificador(char *) Diretório/Arquivo: hipdll.h	Descrição: Protótipo da função de DLL Chama: Chamado por:
Procedimento: (20) int FAR PASCAL retorna_nome(int, char *) Diretório/Arquivo: hipdll.h	Descrição: Protótipo da função de DLL Chama: Chamado por:
Procedimento: (21) int FAR PASCAL primeiro_no_hiperbase(char *) Diretório/Arquivo: hipdll.h	Descrição: Protótipo da função de DLL Chama: Chamado por:
Procedimento: (22) int FAR PASCAL proximo_no_hiperbase(char *) Diretório/Arquivo: hipdll.h	Descrição: Protótipo da função de DLL Chama: Chamado por:
Procedimento: (23) int FAR PASCAL cria_contexto(char *) Diretório/Arquivo: hipdll.h	Descrição: Protótipo da função de DLL Chama: Chamado por:
Procedimento: (24) int FAR PASCAL cria_trilha(char *) Diretório/Arquivo: hipdll.h	Descrição: Protótipo da função de DLL Chama: Chamado por:
Procedimento: (25) int FAR PASCAL cria_texto(char *) Diretório/Arquivo: hipdll.h	Descrição: Protótipo da função de DLL Chama: Chamado por:
Procedimento: (26) int FAR PASCAL cria_video(char *) Diretório/Arquivo: hipdll.h	Descrição: Protótipo da função de DLL Chama: Chamado por:

Procedimento: (27) int FAR PASCAL cria_audio(char *) Diretório/Arquivo: hipdll.h	Descrição: Protótipo da função de DLL Chama: Chamado por:
Procedimento: (28) int FAR PASCAL cria_imagem(char *) Diretório/Arquivo: hipdll.h	Descrição: Protótipo da função de DLL Chama: Chamado por:
Procedimento: (29) int FAR PASCAL cria_executavel(char *) Diretório/Arquivo: hipdll.h	Descrição: Protótipo da função de DLL Chama: Chamado por:
Procedimento: (30) int FAR PASCAL cria_tabela(char *) Diretório/Arquivo: hipdll.h	Descrição: Protótipo da função de DLL Chama: Chamado por:
Procedimento: (31) int FAR PASCAL insere_no_contexto(char *,char *) Diretório/Arquivo: hipdll.h	Descrição: Protótipo da função de DLL Chama: Chamado por:
Procedimento: (32) int FAR PASCAL insere_na_trilha(char *,char *,char *) Diretório/Arquivo: hipdll.h	Descrição: Protótipo da função de DLL Chama: Chamado por:
Procedimento: (33) int FAR PASCAL insere_na_posicao(char *, char *,char *, int) Diretório/Arquivo: hipdll.h	Descrição: Protótipo da função de DLL Chama: Chamado por:
Procedimento: (34) int FAR PASCAL retira_do_contexto(char *,char *) Diretório/Arquivo: hipdll.h	Descrição: Protótipo da função de DLL Chama: Chamado por:
Procedimento: (35) int FAR PASCAL retira_da_trilha(char *,char *) Diretório/Arquivo: hipdll.h	Descrição: Protótipo da função de DLL Chama: Chamado por:
Procedimento: (36) int FAR PASCAL limpa_trilha(char *) Diretório/Arquivo: hipdll.h	Descrição: Protótipo da função de DLL Chama: Chamado por:
Procedimento: (37) int FAR PASCAL Quantos_contextos_pertence(char *) Diretório/Arquivo: hipdll.h	Descrição: Protótipo da função de DLL Chama: Chamado por:
Procedimento: (38) int FAR PASCAL primeiro_contexto_pai(char *, char *) Diretório/Arquivo: hipdll.h	Descrição: Protótipo da função de DLL Chama: Chamado por:
Procedimento: (39) int FAR PASCAL ultimo_contexto_pai(char *, char *)	Descrição: Protótipo da função de DLL

Diretório/Arquivo: hipdll.h	Chama: Chamado por:
Procedimento: (40) int FAR PASCAL proximo_contexto_pai(char *, char *)	Descrição: Protótipo da função de DLL
Diretório/Arquivo: hipdll.h	Chama: Chamado por:
Procedimento: (41) int FAR PASCAL primeiro_no_contexto(char *, char *)	Descrição: Protótipo da função de DLL
Diretório/Arquivo: hipdll.h	Chama: Chamado por:
Procedimento: (42) int FAR PASCAL proximo_no_contexto(char *, char *)	Descrição: Protótipo da função de DLL
Diretório/Arquivo: hipdll.h	Chama: Chamado por:
Procedimento: (43) int FAR PASCAL primeiro_no_trilha(char *, char *, char *)	Descrição: Protótipo da função de DLL
Diretório/Arquivo: hipdll.h	Chama: Chamado por:
Procedimento: (44) int FAR PASCAL proximo_no_trilha(char *, char *, char *)	Descrição: Protótipo da função de DLL
Diretório/Arquivo: hipdll.h	Chama: Chamado por:
Procedimento: (45) int FAR PASCAL ultimo_no_trilha(char *, char *, char *)	Descrição: Protótipo da função de DLL
Diretório/Arquivo: hipdll.h	Chama: Chamado por:
Procedimento: (46) int FAR PASCAL anterior_no_trilha(char *, char *, char *)	Descrição: Protótipo da função de DLL
Diretório/Arquivo: hipdll.h	Chama: Chamado por:
Procedimento: (47) No *FAR PASCAL primeiro_ponteiro_no(char *)	Descrição: Protótipo da função de DLL
Diretório/Arquivo: hipdll.h	Chama: Chamado por:
Procedimento: (48) No *FAR PASCAL proximo_ponteiro_no(char *)	Descrição: Protótipo da função de DLL
Diretório/Arquivo: hipdll.h	Chama: Chamado por:
Procedimento: (49) No *FAR PASCAL retorna_ponteiro_no(int)	Descrição: Protótipo da função de DLL
Diretório/Arquivo: hipdll.h	Chama: Chamado por:
Procedimento: (50) int FAR PASCAL recupera_nome_arquivo_video(char *, char *)	Descrição: Protótipo da função de DLL
Diretório/Arquivo: hipdll.h	Chama: Chamado por:
Procedimento: (51) int FAR PASCAL	Descrição: Protótipo da função de DLL

insere_nome_arquivo_video(char *, char *) Diretório/Arquivo: hipdll.h	Chama: Chamado por:
Procedimento: (52) int FAR PASCAL recupera_nome_arquivo_texto(char *, char *) Diretório/Arquivo: hipdll.h	Descrição: Protótipo da função de DLL Chama: Chamado por:
Procedimento: (53) int FAR PASCAL insere_nome_arquivo_texto(char *, char *) Diretório/Arquivo: hipdll.h	Descrição: Protótipo da função de DLL Chama: Chamado por:
Procedimento: (54) int FAR PASCAL recupera_nome_arquivo_audio(char *, char *) Diretório/Arquivo: hipdll.h	Descrição: Protótipo da função de DLL Chama: Chamado por:
Procedimento: (55) int FAR PASCAL insere_nome_arquivo_audio(char *, char *) Diretório/Arquivo: hipdll.h	Descrição: Protótipo da função de DLL Chama: Chamado por:
Procedimento: (56) int FAR PASCAL recupera_nome_arquivo_imagem(char *, char *) Diretório/Arquivo: hipdll.h	Descrição: Protótipo da função de DLL Chama: Chamado por:
Procedimento: (57) int FAR PASCAL insere_nome_arquivo_imagem(char *, char *) Diretório/Arquivo: hipdll.h	Descrição: Protótipo da função de DLL Chama: Chamado por:
Procedimento: (58) int FAR PASCAL recupera_nome_arquivo_executavel(ch ar *, char *) Diretório/Arquivo: hipdll.h	Descrição: Protótipo da função de DLL Chama: Chamado por:
Procedimento: (59) int FAR PASCAL insere_nome_arquivo_executavel(char *, char *) Diretório/Arquivo: hipdll.h	Descrição: Protótipo da função de DLL Chama: Chamado por:
Procedimento: (60) int FAR PASCAL recupera_nome_arquivo_tabela(char *, char *) Diretório/Arquivo: hipdll.h	Descrição: Protótipo da função de DLL Chama: Chamado por:
Procedimento: (61) int FAR PASCAL insere_nome_arquivo_tabela(char *, char *) Diretório/Arquivo: hipdll.h	Descrição: Protótipo da função de DLL Chama: Chamado por:
Procedimento: (62) int FAR PASCAL tipo_no(char *) Diretório/Arquivo: hipdll.h	Descrição: Protótipo da função de DLL Chama:

	Chamado por:
Procedimento: (63) int FAR PASCAL retira_da_hiperbase(char *) Diretório/Arquivo: hipdll.h	Descrição: Protótipo da função de DLL Chama: Chamado por:
Procedimento: (64) int FAR PASCAL muda_atributos(char *, char *, int) Diretório/Arquivo: hipdll.h	Descrição: Protótipo da função de DLL Chama: Chamado por:
Procedimento: (65) int FAR PASCAL muda_tipo(char *, char *) Diretório/Arquivo: hipdll.h	Descrição: Protótipo da função de DLL Chama: Chamado por:
Procedimento: (66) int FAR PASCAL recupera_tipo(char *, char *) Diretório/Arquivo: hipdll.h	Descrição: Protótipo da função de DLL Chama: Chamado por:
Procedimento: (67) int FAR PASCAL muda_funcao_didatica(char *, char *) Diretório/Arquivo: hipdll.h	Descrição: Protótipo da função de DLL Chama: Chamado por:
Procedimento: (68) int FAR PASCAL recupera_funcao_didatica(char *, char *) Diretório/Arquivo: hipdll.h	Descrição: Protótipo da função de DLL Chama: Chamado por:
Procedimento: (69) int FAR PASCAL muda_assunto(char *, char *) Diretório/Arquivo: hipdll.h	Descrição: Protótipo da função de DLL Chama: Chamado por:
Procedimento: (70) int FAR PASCAL recupera_assunto(char *, char *) Diretório/Arquivo: hipdll.h	Descrição: Protótipo da função de DLL Chama: Chamado por:
Procedimento: (71) int FAR PASCAL muda_dificuldade(char *, int) Diretório/Arquivo: hipdll.h	Descrição: Protótipo da função de DLL Chama: Chamado por:
Procedimento: (72) int FAR PASCAL recupera_dificuldade(char *, int *) Diretório/Arquivo: hipdll.h	Descrição: Protótipo da função de DLL Chama: Chamado por:
Procedimento: (73) int FAR PASCAL muda_palavra_chave(char *, int, char *) Diretório/Arquivo: hipdll.h	Descrição: Protótipo da função de DLL Chama: Chamado por:
Procedimento: (74) int FAR PASCAL recupera_palavra_chave(char *, int, char *) Diretório/Arquivo: hipdll.h	Descrição: Protótipo da função de DLL Chama: Chamado por:
Procedimento:	Descrição:

(75) int FAR PASCAL regras_tutoriais(char *, char *, char *, int, char *, char *, int *) Diretório/Arquivo: hipdll.h	Protótipo da função de DLL Chama: Chamado por:
Procedimento: (76) int FAR PASCAL cria_ancora_texto(char *, int, char *, int, int) Diretório/Arquivo: hipdll.h	Descrição: Protótipo da função de DLL Chama: Chamado por:
Procedimento: (77) int FAR PASCAL muda_ancora_texto(char *, int, int, char *, int, int) Diretório/Arquivo: hipdll.h	Descrição: Protótipo da função de DLL Chama: Chamado por:
Procedimento: (78) int FAR PASCAL cria_ancora_contexto(char *, int, char *) Diretório/Arquivo: hipdll.h	Descrição: Protótipo da função de DLL Chama: Chamado por:
Procedimento: (79) int FAR PASCAL muda_ancora_contexto(char *, int, int, char *) Diretório/Arquivo: hipdll.h	Descrição: Protótipo da função de DLL Chama: Chamado por:
Procedimento: (80) int FAR PASCAL retira_ancora(char *, int) Diretório/Arquivo: hipdll.h	Descrição: Protótipo da função de DLL Chama: Chamado por:
Procedimento: (81) int FAR PASCAL recupera_identificador_interno(char *, int) Diretório/Arquivo: hipdll.h	Descrição: Protótipo da função de DLL Chama: Chamado por:
Procedimento: (82) int FAR PASCAL numero_de_ancoras(char *) Diretório/Arquivo: hipdll.h	Descrição: Protótipo da função de DLL Chama: Chamado por:
Procedimento: (83) int FAR PASCAL recupera_nome_ancora(char *, int, char *) Diretório/Arquivo: hipdll.h	Descrição: Protótipo da função de DLL Chama: Chamado por:
Procedimento: (84) int FAR PASCAL primeira_ancora(char *, char *) Diretório/Arquivo: hipdll.h	Descrição: Protótipo da função de DLL Chama: Chamado por:
Procedimento: (85) int FAR PASCAL ultima_ancora(char *, char *) Diretório/Arquivo: hipdll.h	Descrição: Protótipo da função de DLL Chama: Chamado por:
Procedimento: (86) int FAR PASCAL proxima_ancora(char *, char *) Diretório/Arquivo: hipdll.h	Descrição: Protótipo da função de DLL Chama: Chamado por:

Procedimento: (87) int FAR PASCAL mostra_primeiro_lixo(char *) Diretório/Arquivo: hipdll.h	Descrição: Protótipo da função de DLL Chama: Chamado por:
Procedimento: (88) int FAR PASCAL mostra_proximo_lixo(char *) Diretório/Arquivo: hipdll.h	Descrição: Protótipo da função de DLL Chama: Chamado por:
Procedimento: (89) int FAR PASCAL cria_versao(char *,char *) Diretório/Arquivo: hipdll.h	Descrição: Protótipo da função de DLL Chama: Chamado por:
Procedimento: (90) int FAR PASCAL muda_vcorrente(char *,char *) Diretório/Arquivo: hipdll.h	Descrição: Protótipo da função de DLL Chama: Chamado por:
Procedimento: (91) int FAR PASCAL cria_elo(char *, int, char *, int, char *, int, int, int) Diretório/Arquivo: hipdll.h	Descrição: Protótipo da função de DLL Chama: Chamado por:
Procedimento: (92) int FAR PASCAL muda_elo(char *, int, char *, int, char *, int, int, int) Diretório/Arquivo: hipdll.h	Descrição: Protótipo da função de DLL Chama: Chamado por:
Procedimento: (93) int FAR PASCAL retira_elo(char *, int) Diretório/Arquivo: hipdll.h	Descrição: Protótipo da função de DLL Chama: Chamado por:
Procedimento: (94) int FAR PASCAL numero_de_elos(char *) Diretório/Arquivo: hipdll.h	Descrição: Protótipo da função de DLL Chama: Chamado por:
Procedimento: (95) int FAR PASCAL recupera_elo(char *, int, char *, int *, char *, int *, int *, int *) Diretório/Arquivo: hipdll.h	Descrição: Protótipo da função de DLL Chama: Chamado por:
Procedimento: (96) int FAR PASCAL primeiro_elo(char *, int *) Diretório/Arquivo: hipdll.h	Descrição: Protótipo da função de DLL Chama: Chamado por:
Procedimento: (97) int FAR PASCAL proximo_elo(char *, int *) Diretório/Arquivo: hipdll.h	Descrição: Protótipo da função de DLL Chama: Chamado por:
Procedimento: (98) elo *FAR PASCAL primeiro_ponteiro_elo(char *) Diretório/Arquivo: hipdll.h	Descrição: Protótipo da função de DLL Chama: Chamado por:
Procedimento: (99) elo *FAR PASCAL proximo_ponteiro_elo(char *, elo *)	Descrição: Protótipo da função de DLL

Diretório/Arquivo: hipdll.h	Chama: Chamado por:
Sistema: SASHE Arquivo: Sashe\Hyperprop\hipger.cpp	
Procedimento: (66) extern "C" int FAR PASCAL _export cria_base_privada(char *nome) Diretório/Arquivo: hipger.cpp	Descrição: Retorna: 1: Já existe uma base privada com este nome carregado na memória. 2: Erro ao ativar hiperbase. 0: Ok. Chama: (68) hb.Retorna(nome) (69) new baseprivada(nome) (70) hb.ativa_hiperbase(nome) Chamado por: Hip_prin.pas\ (256) To_fo_Principal.FormActivate(Sender: TObject) Hip_prin.pas\ (382) To_fo_Principal.o_bu_CriarHiperbaseClick (Sender: TObject)
Procedimento: (78) extern "C" int FAR PASCAL _export ativa_hiperbase(char *nome) Diretório/Arquivo: hipger.cpp	Descrição: Retorna: 1: Erro ao ativar a Hiperbase. 0: Ok. Chama: (80) hb.ativa_hiperbase(nome) Chamado por: Atexto.pas\ (176) To_fo_Texto.o_hv_TextoHotSpotClick(Sende r: TObject) Controle.pas\ (63) To_fo_Controles.o_bu_TopicoAnteriorClick (Sender: TObject) Controle.pas\ (107) To_fo_Controles.o_bu_ProximoTopicoClick (Sender: TObject) Controle.pas\ (273) To_fo_Controles.o_bu_FacilClick (Sender: TObject) Controle.pas\ (592)To_fo_Controles.o_bu_HistoricoClick (Sender: TObject) Hip_prin.pas\ (256) To_fo_Principal.FormActivate(Sender: TObject) Hip_prin.pas\ (317) To_fo_Principal.FormCloseQuery (Sender: TObject; var CanClose: Boolean)
Procedimento: (88) extern "C" int FAR PASCAL _export salvar(char *nome) Diretório/Arquivo: hipger.cpp	Descrição: Salva os atributos da hiperbase em arquivo Retorna: 1: Impossível abrir arquivo principal 2: Erro de escrita do contador 3: Impossível abrir arquivo de cada um dos nós Chama: (90) hb.grava_hiperbase(nome) Chamado por: Grafoob.h\ (22) int Salvar(char *arq) Hip_prin.pas\ (317) To_fo_Principal.FormCloseQuery (Sender: TObject; var CanClose: Boolean) Hip_prin.pas\ (438) To_fo_Principal.o_bu_GravarHiperbaseClic

Procedimento: (100) extern "C" int FAR PASCAL _export recuperar(char *nome) Diretório/Arquivo: hipger.cpp	k (Sender: TObject) Descrição: Recupera os atributos da hiperbase gravados em arquivo Retorna: 1: Não foi possível abrir arquivo 2: Erro de leitura de contador 3: Erro de leitura de cada nó 4: Não foi possível abrir arquivo do nó 5: Falha de posicionamento no arquivo Chama: (110) hb.recupera_hiperbase(nome) Chamado por: Grafoob.h\ (23) int Recuperar(char *arq) Hip_prin.pas\ (256) To_fo_Principal.FormActivate(Sender: TObject) Hip_prin.pas\ (409) To_fo_Principal.o_bu_AbrirHiperbaseClick (Sender: TObject)
Procedimento: (115) extern "C" int FAR PASCAL _export primeiro_no_hiperbase(char *nome) Diretório/Arquivo: hipger.cpp	Descrição: Coloca na variável nome o nome que identifica o primeiro nó na lista de nós da hiperbase Chama: (117) hb.primeiro_no_hiperbase(nome) Chamado por: Grafoob.h\ (25) int Hiperbase(char *nome) Criar_no.pas\ (62) To_fo_CriarNo.FormCreate(Sender: TObject) Roteiros.pas\ (193) To_fo_Roteiro.FormCreate (Sender: TObject) Edit_no.pas\ (60) To_fo_EditarNo.FormCreate(Sender: TObject)
Procedimento: (122) extern "C" int FAR PASCAL _export proximo_no_hiperbase(char *nome) Diretório/Arquivo: hipger.cpp	Descrição: Coloca na variável nome o nome que identifica o próximo nó na lista de nós da hiperbase Chama: (124) hb.proximo_no_hiperbase(nome) Chamado por: Criar_no.pas\ (62) To_fo_CriarNo.FormCreate(Sender: TObject) Roteiros.pas\ (193) To_fo_Roteiro.FormCreate (Sender: TObject) Edit_no.pas\ (60) To_fo_EditarNo.FormCreate(Sender: TObject)
Procedimento: (129) extern "C" int FAR PASCAL _export limpar_hiperbase() Diretório/Arquivo: hipger.cpp	Descrição: Apaga todos os nós, âncoras e elos da hiperbase e libera a memória utilizada pela hiperbase Chama: (131) hb.~hipwindow() Chamado por: Grafoob.h\ (24) int LimparHiperbase()
Procedimento: (137) extern "C" int FAR PASCAL _export retorna_identificador(char *nome) Diretório/Arquivo: hipger.cpp	Descrição: Dado o nome do nó esta rotina retorna o identificador do nó Chama:

	(139) hb.RetornaIdent(nome) Chamado por: Grafo.cpp\ (210) extern "C" void FAR PASCAL _export ApresentaContexto(char *no) Nos.pas\ (930) To_fo_Nos.o_bu_CriarEloClick(Sender: TObject) Nos.pas\ (1309) To_fo_Nos.o_bu_EditarEloClick(Sender: TObject)
Procedimento: (144) extern "C" int FAR PASCAL _export retorna_nome(int ident, char *nome) Diretório/Arquivo: hipger.cpp	Descrição: Dado o identificador do nó esta rotina retorna o nome do nó Chama: (146) hb.RetornaNo(ident) Chamado por: Atexto.pas\ (52) To_fo_Texto.Procurar_Elo(Ancora: string; NomeDoNoLiberdade: string): boolean Atexto.pas\ (176) To_fo_Texto.o_hv_TextoHotSpotClick(Sender: TObject; const SRC: String; var Handled: Boolean) Nos.pas\ (291) To_fo_Nos.Ler_Elos Nos.pas\ (1079) To_fo_Nos.o_lb_ListaElosClick(Sender: TObject) Nos.pas\ (1309) To_fo_Nos.o_bu_EditarEloClick(Sender: TObject)
Procedimento: (161) extern "C" int FAR PASCAL _export cria_contexto(char *nome) Diretório/Arquivo: hipger.cpp	Descrição: Cria um nó contexto com o nome passado como argumento Retorna: 1: Já existe um nó com este nome na hiperbase 0: Ok. Chama: (163) hb.Retorna(nome) (164) new contextousuario(nome) (165) pno->cria_ancora_lambda(0) (166) hb.insere_base_ativa(pno) Chamado por: Nos.pas\ (557) To_fo_Nos.o_pm_CriarNoClick(Sender: TObject)
Procedimento: (174) extern "C" int FAR PASCAL _export cria_trilha(char *nome) Diretório/Arquivo: hipger.cpp	Descrição: Cria um nó trilha com o nome passado como argumento Retorna: 1: Já existe um nó com este nome na hiperbase 0: Ok. Chama: (176) hb.Retorna(nome) (177) new trilha(nome) (178) pno->cria_ancora_lambda(0) (179) hb.insere_base_ativa(pno) Chamado por: Hip_prin.pas\ (256) To_fo_Principal.FormActivate(Sender: TObject) Roteiros.pas\ (383) To_fo_Roteiro.o_b_Criar_RoteiroClick(Sender: TObject)

Procedimento: (187) extern "C" int FAR PASCAL _export cria_texto(char *nome) Diretório/Arquivo: hipger.cpp	Descrição: Cria um nó texto com o nome passado como argumento Retorna: 1: Já existe um nó com este nome na hiperbase 0: Ok. Chama: (189) (hb.Retorna(nome) (190) new no_texto(nome) (191) pno->cria_ancora_texto(0, "lambda", 0, 0) (192) hb.insere_base_ativa(pno) Chamado por: Nos.pas\ (557) To_fo_Nos.o_pm_CriarNoClick(Sender: TObject)
Procedimento: (200) extern "C" int FAR PASCAL _export cria_video(char *nome) Diretório/Arquivo: hipger.cpp	Descrição: Cria um nó vídeo com o nome passado como argumento Retorna: 1: Já existe um nó com este nome na hiperbase 0: Ok. Chama: (202) (hb.Retorna(nome) (203) new no_video(nome) (204) pno->cria_ancora_video(0) (205) hb.insere_base_ativa(pno) Chamado por: Nos.pas\ (557) To_fo_Nos.o_pm_CriarNoClick(Sender: TObject)
Procedimento: (213) extern "C" int FAR PASCAL _export cria_audio(char *nome) Diretório/Arquivo: hipger.cpp	Descrição: Cria um nó áudio com o nome passado como argumento Retorna: 1: Já existe um nó com este nome na hiperbase 0: Ok. Chama: (215) (hb.Retorna(nome) (216) new no_audio(nome) (217) pno->cria_ancora_audio(0) (218) hb.insere_base_ativa(pno) Chamado por: Nos.pas\ (557) To_fo_Nos.o_pm_CriarNoClick(Sender: TObject)
Procedimento: (226) extern "C" int FAR PASCAL _export cria_imagem(char *nome) Diretório/Arquivo: hipger.cpp	Descrição: Cria um nó imagem com o nome passado como argumento Retorna: 1: Já existe um nó com este nome na hiperbase 0: Ok. Chama: (228) (hb.Retorna(nome) (229) new no_imagem(nome) (230) pno->cria_ancora_imagem(0) (231) hb.insere_base_ativa(pno) Chamado por: Nos.pas\ (557) To_fo_Nos.o_pm_CriarNoClick(Sender: TObject)
Procedimento: (239) extern "C" int FAR PASCAL	Descrição: Retorna:

<pre>_export cria_executavel(char *nome)</pre> Diretório/Arquivo: hipger.cpp	1: Já existe um nó com este nome na hiperbase 0: Ok. Chama: <pre>(241) (hb.Retorna(nome) (242) new no_executavel(nome) (243) pno->cria_ancora_executavel(0) (244) hb.insere_base_ativa(pno)</pre> Chamado por:
Procedimento: <pre>(252) extern "C" int FAR PASCAL _export cria_tabela(char *nome)</pre> Diretório/Arquivo: hipger.cpp	Descrição: Retorna: 1: Já existe um nó com este nome na hiperbase 0: Ok. Chama: <pre>(254) (hb.Retorna(nome) (255) new no_tabela(nome) (256) pno->cria_ancora_tabela(0) (257) hb.insere_base_ativa(pno)</pre> Chamado por:
Procedimento: <pre>(263) extern "C" int FAR PASCAL _export insere_no_contexto(char *nome_cont, char *nome_no)</pre> Diretório/Arquivo: hipger.cpp	Descrição: Insere um nó (texto, imagem, audio, video, trilha e contexto) em um nó contexto Chama: <pre>(269) pontno->eh_contexto (270) pontno->eh_contexto_versoes() (271) pontno->ObtemEstado() (273) hb.Retorna(nome_no) (275) pontno->contem(pno1) (276) hb.atualiza_base_ativa(pno1) (277) pontno->insere_no(pno1)</pre> Chamado por: <pre>Hip_prin.pas\ (256) To_fo_Principal.FormActivate(Sender: TObject) Roteiros.pas\ (383) To_fo_Roteiro.o_b_Criar_RoteiroClick (Sender: TObject) Nos.pas\ (557) To_fo_Nos.o_pm_CriarNoClick(Sender: TObject) Nos.pas\ (1224) To_fo_Nos.o_lb_NosDragDrop(Sender, Source: TObject; X, Y: Integer)</pre>
Procedimento: <pre>(288) extern "C" int FAR PASCAL _export insere_na_trilha(char *nome_trilha, char *nome_no, char *no_liberdade)</pre> Diretório/Arquivo: hipger.cpp	Descrição: Insere um nó no final da trilha Retorna: 1: Se o nome da trilha passado como parâmetro não existe 2: Se o nome passado não é um nó do tipo trilha 3: Se o nó trilha não é temporário 5: Se o nome do nó a ser inserido ou o nome do nó liberdade não existirem Chama: <pre>(290) hb.RetornaBase(nome_trilha) (294) pontno->eh_trilha() (295) hb.Retorna(nome_no) (299) hb.Retorna(no_liberdade) (301) hb.atualiza_base_ativa(pno1) (302) hb.atualiza_base_ativa(pno2) (303) pontno->insere_no(pno1, pno2) (305) pontno->insere_no(pno1)</pre> Chamado por: <pre>Atexto.pas\ (176) To_fo_Texto.o_hv_TextoHotSpotClick(Sende</pre>

	r: TObject Controle.pas\ (63) To_fo_Controles.o_bu_TopicoAnteriorClick (Sender: TObject) Controle.pas\ (107) To_fo_Controles.o_bu_ProximoTopicoClick (Sender: TObject) Controle.pas\ (273) To_fo_Controles.o_bu_FacilClick (Sender: TObject) Controle.pas\ (322) To_fo_Controles.o_bu_DificilClick (Sender: TObject) Hip_prin.pas\ (256) To_fo_Principal.FormActivate (Sender: TObject) Roteiros.pas\ (383) To_fo_Roteiro.o_b_Criar_RoteiroClick (Sender: TObject)
Procedimento: (317) extern "C" int FAR PASCAL _export insere_na_posicao(char *nome_trilha, char *nome_no, char *no_liberdade, int posicao)	Descrição: Retorna: 1: O nó trilha não existe. 2: O nó não é trilha. 3: O nó a ser inserido não existe. 4: Houve problema na alocação de memória 5: A posição não é válida 6: A trilha não é temporária 0: Ok. Chama: (319) hb.RetornaBase(nome_trilha) (324) pontno->eh_trilha() (325) hb.Retorna(nome_no) (326) hb.Retorna(no_liberdade) (328) hb.pertence_base_ativa(pno1) (329) hb.insere_base_ativa(pno1) (330) hb.pertence_base_ativa(pno2) (331) hb.insere_base_ativa(pno2) (332) pontno->insere_no(pno1, pno2, posicao) Chamado por:
Diretório/Arquivo: hipger.cpp	
Procedimento: (339) extern "C" int FAR PASCAL _export retira_do_contexto(char *nome_contexto, char *nome_no) Diretório/Arquivo: hipger.cpp	Descrição: Retira do nó contexto um determinado nó Chama: (341) hb.Retorna(nome_contexto) (342) hb.Retorna(nome_no) (345) pontno->eh_contexto (346) pontno->eh_contexto_versoes() (347) pontno->ObtemEstado() (349) pontno->contem(pontno2) (350) pontno->retira_no(pontno2) (352) retira_da_hiperbase(nome_no) Chamado por: Nos.pas\ (845) To_fo_Nos.o_pm_RetirarNoClick (Sender: TObject) Nos.pas\ (1224) To_fo_Nos.o_lb_NosDragDrop (Sender, Source: TObject; X, Y: Integer)
Procedimento: (359) extern "C" int FAR PASCAL _export retira_da_trilha(char *nome_trilha, char *nome_no_liberdade) Diretório/Arquivo: hipger.cpp	Descrição: Chama:

	<pre> (361) hb.Retorna(nome_trilha) (365) pontno->eh_trilha()) return 2 (366) pontno->eh_contexto_versoes() (367) pontno->ObtemEstado() (369) pontno->contem(pontno2) (370) pontno->retira_no(pontno2) Chamado por: Aparece em funcoes.pas e hip_prin.pas, mas apenas como funcao declarada. VERIFICAR!!! </pre>
Procedimento: <pre> (376) extern "C" int FAR PASCAL _export limpa_trilha(char *nome_trilha) Diretório/Arquivo: hipger.cpp </pre>	Descrição: Chama: <pre> (378) hb.Retorna(nome_trilha) (381) pontno->eh_trilha() (382) pontno->eh_contexto_versoes (383) pontno->ObtemEstado() (384) pontno->limpa_trilha() Chamado por: Roteiros.pas\ (383) To_fo_Roteiro.o_b_Criar_RoteiroClick (Sender: TObject) Nos.pas\ (819) To_fo_Nos.o_pm_ApagarNoClick(Sender: TObject) </pre>
Procedimento: <pre> (390) extern "C" int FAR PASCAL _export quantos_contextos_pertence(char *nome) Diretório/Arquivo: hipger.cpp </pre>	Descrição: Chama: <pre> (395) pontno->num_contextos() Chamado por: </pre>
Procedimento: <pre> (400) extern "C" int FAR PASCAL _export primeiro_contexto_pai(char *nome, char *nome_pai) Diretório/Arquivo: hipger.cpp </pre>	Descrição: Chama: <pre> (402) hb.Retorna(nome) (405) pontno->primeiro_contexto_pai() (407) primeiro->ObtemNome() Chamado por: Atexto.pas \ (52) To_fo_Texto.Procurar_Elo(Ancora: string; NomeDoNoLiberdade: string): boolean Controle.pas\ (371)To_fo_Conroles.o_bu_AjudaClick (Sender: TObject) Controle.pas\ (419) To_fo_Conroles.o_bu_BibliografiaClick (Sender: TObject) Controle.pas\ (467) To_fo_Conroles.o_bu_ExerciciosClick (Sender: TObject) Roteiros.pas\ (242) To_fo_Roteiro.o_mp_InserirNoRoteiroClick (Sender: TObject) Roteiros.pas\ (593) To_fo_Roteiro.o_mp_MudarNoLiberdadeClick (Sender: TObject) </pre>
Procedimento: <pre> (413) extern "C" int FAR PASCAL _export ultimo_contexto_pai(char *nome, char *nome_pai) Diretório/Arquivo: hipger.cpp </pre>	Descrição: Chama:

	(415) hb.Retorna(nome) (418) pontno->ultimo_contexto_pai() (420) aux->ObtemNome() Chamado por:
Procedimento: (426) extern "C" int FAR PASCAL _export proximo_contexto_pai(char *nome, char *nome_pai) Diretório/Arquivo: hipger.cpp	Descrição: Chama: (428) hb.Retorna(nome) (431) pontno->proximo_contexto_pai() (433) prox->ObtemNome() Chamado por: Aparece em funcoes.pas e hip_prin.pas, mas apenas como funcao declarada. VERIFICAR!!!
Procedimento: (442) extern "C" int FAR PASCAL _export primeiro_no_contexto(char *nome_cont, char *nome_no) Diretório/Arquivo: hipger.cpp	Descrição: Retorna em nome_no o nome do primeiro nó que pertence a lista do nó contexto Retorna: 1: O nó contexto não existe. 2: O nó não é do tipo contexto. Chama: (444) hb.Retorna(nome_cont) (448) pontno->eh_contexto() (449) pontno->primeiro_no(nome_no) Chamado por: Controle.pas\ (371)To_fo_Controles.o_bu_AjudaClick (Sender: TObject) Controle.pas\ (419) To_fo_Controles.o_bu_BibliografiaClick (Sender: TObject) Controle.pas\ (467) To_fo_Controles.o_bu_ExerciciosClick (Sender: TObject) Hip_prin.pas\ (189) To_fo_Principal.Apresentar_No(p_NomeDoNo : Ptochar; ApresentarNosTerminais: integer) Roteiros.pas\ (94) To_fo_Roteiro.SelecionarNos (p_IndiceInicial: LongInt; p_ContextoInicial: String) Browser.pas\ (47) To_fo_BrowserEstrutural.Ler_Contexto(Ind ice: LongInt; Contexto_Inicial: String) Criarelo.pas\ (61)To_fo_CriarElos.Ler_Contexto_Origem(Indice: LongInt; Contexto_Inicial: String) Criarelo.pas\ (103) To_fo_CriarElos.Ler_Contexto_Destino(Ind ice: LongInt; Contexto_Inicial: String) Maisinfo.pas\ (36) To_fo_MaisInformacoes.Ler_Contexto(Con tex to_Inicial: String) Nos.pas\ (106) To_fo_Nos.Ler_Nos
Procedimento: (458) extern "C" int FAR PASCAL _export proximo_no_contexto(char *nome_cont, char *nome_no) Diretório/Arquivo: hipger.cpp	Descrição: Retorna em nome_no o nome que identifica o próximo nó na lista do nó contexto Retorna: 1: O nó contexto não existe. 2: O nó não é do tipo contexto. Chama: (460) hb.Retorna(nome_cont) (464) pontno->eh_contexto()

	<pre> (465) pontno->proximo_no(nome_no) Chamado por: Controle.pas\ (371)To_fo_Controles.o_bu_AjudaClick (Sender: TObject) Controle.pas\ (419) To_fo_Controles.o_bu_BibliografiaClick (Sender: TObject) Controle.pas\ (467) To_fo_Controles.o_bu_ExerciciosClick (Sender: TObject) Hip_prin.pas\ (189) To_fo_Principal.Apresentar_No(p_NomeDoNo : Ptochar; ApresentarNosTerminais: integer) Roteiros.pas\ (94) To_fo_Roteiro.SelecionarNos (p_IndiceInicial: LongInt; p_ContextoInicial: String) Browser.pas\ (47) To_fo_BrowserEstrutural.Ler_Contexto(Ind ice: LongInt; Contexto_Inicial: String) Criarelo.pas\ (61)To_fo_CriarElos.Ler_Contexto_Origem(Indice: LongInt; Contexto_Inicial: String) Criarelo.pas\ (103) To_fo_CriarElos.Ler_Contexto_Destino(Ind ice: LongInt; Contexto_Inicial: String) Maisinfo.pas\ (36) To_fo_MaisInformacoes.Ler_Contexto(Conte xto_Inicial: String) Nos.pas\ (106) To_fo_Nos.Ler Nos </pre>
Procedimento: <pre> (472) extern "C" int FAR PASCAL _export primeiro_no_trilha(char *nome_trilha, char *nome_no, char *nome_liberdade) Diretório/Arquivo: hipger.cpp </pre>	Descrição: Chama: <pre> (474) hb.Retorna(nome_trilha) (477) pontno->eh_trilha() (478) pontno- >primeiro_no_trilha(nome_no, nome_liberdade) Chamado por: Controle.pas\ (546) To_fo_Controles.o_bu_OndeEstouClick(Send er: TObject) Controle.pas\ (592)To_fo_Controles.o_bu_HistoricoClick (Sender: TObject) Hip_prin.pas\ (256) To_fo_Principal.FormActivate(Sender: TObject) Roteiros.pas\ (193) To_fo_Roteiro.FormCreate (Sender: TObject) Nos.pas\ (362) To_Fo_Nos.ApresentarNoTrilha(NoSeleciona do: Ptochar) </pre>
Procedimento: <pre> (483) extern "C" int FAR PASCAL _export proximo_no_trilha(char *nome_trilha, char *nome_no, char *nome_liberdade) Diretório/Arquivo: hipger.cpp </pre>	Descrição: Chama: <pre> (485) hb.Retorna(nome_trilha) </pre>

	<pre> (488) pontno->eh_trilha() (489) pontno->proximo_no_trilha(nome_no, nome_liberdade) Chamado por: Controle.pas\ (107) To_fo_Controles.o_bu_ProximoTopicoClick (Sender: TObject) Controle.pas\ (546) To_fo_Controles.o_bu_OndeEstouClick(Send er: TObject) Controle.pas\ (592)To_fo_Controles.o_bu_HistoricoClick (Sender: TObject) Roteiros.pas\ (193) To_fo_Roteiro.FormCreate (Sender: TObject) Nos.pas\ (362) To_Fo_Nos.ApresentarNoTrilha(NoSeleciona do: Ptochar) </pre>
Procedimento: <pre> (494) extern "C" int FAR PASCAL _export ultimo_no_trilha(char *nome_trilha, char *nome_no, char *nome_liberdade) Diretório/Arquivo: hipger.cpp </pre>	Descrição: Chama: <pre> (496) hb.Retorna(nome_trilha) (499) pontno->eh_trilha() (500) pontno->ultimo_no_trilha(nome_no, nome_liberdade) </pre> Chamado por:
Procedimento: <pre> (505) extern "C" int FAR PASCAL _export anterior_no_trilha(char *nome_trilha, char *nome_no, char *nome_liberdade) Diretório/Arquivo: hipger.cpp </pre>	Descrição: Chama: <pre> (507) hb.Retorna(nome_trilha) (510) pontno->eh_trilha() (511) pontno- >anterior_no_trilha(nome_no, nome_liberdade) </pre> Chamado por: Controle.pas\ (63) To_fo_Controles.o_bu_TopicoAnteriorClick (Sender: TObject)
Procedimento: <pre> (516) No *FAR PASCAL _export primeiro_ponteiro_no(char *nome_cont) Diretório/Arquivo: hipger.cpp </pre>	Descrição: Chama: <pre> (518) hb.Retorna(nome_cont) (521) pontno->eh_contexto() (522) pontno->nos.primeiro() (523) pontno->eh_trilha() (526) tri->nos.primeiro() </pre> Chamado por: Grafoob.cpp\ (27) No *PrimeiroNo(char *nome)
Procedimento: <pre> (533) No *FAR PASCAL _export proximo_ponteiro_no(char *nome_cont) Diretório/Arquivo: hipger.cpp </pre>	Descrição: Chama: <pre> (535) hb.Retorna(nome_cont) (538) pontno->eh_contexto() (539) pontno->nos.proximo() (540) pontno->eh_trilha() </pre>

	(543) tri->nos.proximo() Chamado por: Grafoob.cpp\ (28) No *ProximoNo(char *nome)
Procedimento: (550) No *FAR PASCAL _export retorna_ponteiro_no(int id) Diretório/Arquivo: hipger.cpp	Descrição: Chama: (552) hb.RetornaNo(id) Chamado por: Grafoob.cpp\ (26) No *RetornaPtNo(int id)
Procedimento: (557) extern "C" int FAR PASCAL _export recupera_nome_arquivo_video(char *nome_video, char *arq_video) Diretório/Arquivo: hipger.cpp	Descrição: Recupera o nome do arquivo que contém o vídeo a ser mostrado Chama: (559) hb.Retorna(nome_video) (562) pontno->eh_video() (563) pontno->recupera_nome_arquivo() Chamado por: Nos.pas\ (135) To_fo_Nos.Visualizar_No_Terminal(Tipo: char; Nome_No: Ptochar) Nos.pas\ (667) To_fo_Nos.o_pm_EditarNoClick(Sender: TObject)
Procedimento: (569) extern "C" int FAR PASCAL _export insere_nome_arquivo_video(char *nome_video, char *arq_video) Diretório/Arquivo: hipger.cpp	Descrição: Atualiza o nome do arquivo que contém o vídeo Chama: (571) hb.Retorna(nome_video) (574) pontno->eh_video() (575) pontno->muda_nome_arquivo(arq_video) Chamado por: Nos.pas\ (557) To_fo_Nos.o_pm_CriarNoClick(Sender: TObject) Nos.pas\ (667) To_fo_Nos.o_pm_EditarNoClick(Sender: TObject)
Procedimento: (581) extern "C" int FAR PASCAL _export recupera_nome_arquivo_texto(char *nome_texto, char *arquivo) Diretório/Arquivo: hipger.cpp	Descrição: Recupera o nome do arquivo que contém o texto a ser mostrado Chama: (583) hb.Retorna(nome_texto) (586) pontno->eh_texto() (587) pontno->recupera_nome_arquivo() Chamado por: Nos.pas\ (135) To_fo_Nos.Visualizar_No_Terminal(Tipo: char; Nome_No: Ptochar) Nos.pas\ (667) To_fo_Nos.o_pm_EditarNoClick(Sender: TObject) Atexto.pas\ (116) To_fo_Texto.Ler_Resposta(Numero_Duvida: integer)
Procedimento: (593) extern "C" int FAR PASCAL _export	Descrição: Atualiza o nome do arquivo que contém o texto

<pre>insere_nome_arquivo_texto(char *nome_texto, char *arquivo) Diretório/Arquivo: hipger.cpp</pre>	Chama: <pre>(595) hb.Retorna(nome_texto) (598) pontno->eh_texto() (599) pontno->muda_nome_arquivo(arquivo)</pre> Chamado por: <pre>Nos.pas\ (557) To_fo_Nos.o_pm_CriarNoClick(Sender: TObject) Nos.pas\ (667) To_fo_Nos.o_pm_EditarNoClick(Sender: TObject)</pre>
Procedimento: <pre>(605) extern "C" int FAR PASCAL _export recupera_nome_arquivo_audio(char *nome_audio, char *arquivo) Diretório/Arquivo: hipger.cpp</pre>	Descrição: Recupera o nome do arquivo que contém o audio a ser tocado Chama: <pre>(607) hb.Retorna(nome_audio) (610) pontno->eh_audio() (611) pontno->recupera_nome_arquivo()</pre> Chamado por: <pre>Nos.pas\ (135) To_fo_Nos.Visualizar_No_Terminal(Tipo: char; Nome_No: Ptochar) Nos.pas\ (667) To_fo_Nos.o_pm_EditarNoClick(Sender: TObject)</pre>
Procedimento: <pre>(617) extern "C" int FAR PASCAL _export insere_nome_arquivo_audio(char *nome_audio, char *arquivo) Diretório/Arquivo: hipger.cpp</pre>	Descrição: Atualiza o nome do arquivo que contém o audio Chama: <pre>(619) hb.Retorna(nome_audio) (622) pontno->eh_audio() (623) pontno->muda_nome_arquivo(arquivo)</pre> Chamado por: <pre>Nos.pas\ (557) To_fo_Nos.o_pm_CriarNoClick(Sender: TObject) Nos.pas\ (667) To_fo_Nos.o_pm_EditarNoClick(Sender: TObject)</pre>
Procedimento: <pre>(629) extern "C" int FAR PASCAL _export recupera_nome_arquivo_imagem(char *nome_imagem, char *arquivo) Diretório/Arquivo: hipger.cpp</pre>	Descrição: Recupera o nome do arquivo que contém a imagem a ser mostrada Chama: <pre>(631) hb.Retorna(nome_imagem) (634) pontno->eh_imagem() (635) pontno->recupera_nome_arquivo()</pre> Chamado por: <pre>Nos.pas\ (135) To_fo_Nos.Visualizar_No_Terminal(Tipo: char; Nome_No: Ptochar) Nos.pas\ (667) To_fo_Nos.o_pm_EditarNoClick(Sender: TObject)</pre>
Procedimento: <pre>(641) extern "C" int FAR PASCAL _export insere_nome_arquivo_imagem(char *nome_imagem, char *arquivo) Diretório/Arquivo: hipger.cpp</pre>	Descrição: Atualiza o nome do arquivo que contém a imagem Chama:

	(643) hb.Retorna(nome_imagem) (646) pontno->eh_imagem() (647) pontno->muda_nome_arquivo(arquivo) Chamado por: Nos.pas\ (557) To_fo_Nos.o_pm_CriarNoClick(Sender: TObject) Nos.pas\ (667) To_fo_Nos.o_pm_EditarNoClick(Sender: TObject)
Procedimento: (654) extern "C" int FAR PASCAL _export recupera_nome_arquivo_executavel(ch ar *nome_executavel, char *arquivo) Diretório/Arquivo: hipger.cpp	Descrição: Chama: (656) hb.Retorna(nome_executavel) (659) pontno->eh_executavel() (660) pontno->recupera_nome_arquivo() Chamado por:
Procedimento: (666) extern "C" int FAR PASCAL _export insere_nome_arquivo_executavel(char *nome_executavel, char *arquivo) Diretório/Arquivo: hipger.cpp	Descrição: Chama: (668) hb.Retorna(nome_executavel) (671) pontno->eh_executavel() (672) pontno->muda_nome_arquivo(arquivo) Chamado por:
Procedimento: (678) extern "C" int FAR PASCAL _export recupera_nome_arquivo_tabela(char *nome_tabela, char *arquivo) Diretório/Arquivo: hipger.cpp	Descrição: Chama: (680) hb.Retorna(nome_tabela) (683) pontno->eh_tabela() (684) pontno->recupera_nome_arquivo() Chamado por:
Procedimento: (690) extern "C" int FAR PASCAL _export insere_nome_arquivo_tabela(char *nome_tabela, char *arquivo) Diretório/Arquivo: hipger.cpp	Descrição: Chama: (692) hb.Retorna(nome_tabela) (695) pontno->eh_tabela() (696) pontno->muda_nome_arquivo(arquivo) Chamado por:
Procedimento: (712) extern "C" int FAR PASCAL _export tipo_no(char *nome) Diretório/Arquivo: hipger.cpp	Descrição: Identifica o tipo do nó Retorna: 0: O nó não consta na hiperbase. 1: Nó do tipo contexto. 2: Nó do tipo contexto de versão. 3: Nó do tipo texto. 4: Nó do tipo vídeo. 5: Nó do tipo áudio. 6: Nó do tipo imagem. 7: Nó do tipo trilha. 8: Nó do tipo executável. 9: Nó do tipo tabela. Chama: (714) hb.Retorna(nome) (717) pontno->eh_contexto() (718) pontno->eh_contexto_versoes()

```

(719) pontno->eh_texto()
(720) pontno->eh_video()
(721) pontno->eh_audio()
(722) pontno->eh_imagem()
(723) pontno->eh_trilha()
(724) pontno->eh_executavel()
(725) pontno->eh_tabela()
Chamado por:
Grafo.cpp\ (210) extern "C" void FAR
PASCAL _export ApresentaContexto(char
*no)
Controle.pas\
(515)To_fo_Conroles.o_bu_MaisInformacoesClick(Sender: TObject)
Controle.pas\ (546)
To_fo_Conroles.o_bu_OndeEstouClick(Sender: TObject);
Controle.pas\
(592)To_fo_Conroles.o_bu_HistoricoClick(Sender: TObject);
Hip_prin.pas\ (189)
To_fo_Principal.Apresentar_No(p_NomeDoNo
: Ptochar; ApresentarNosTerminais:
integer)
Roteiros.pas\ (94)
To_fo_Roteiro.SelecionarNos
(p_IndiceInicial: LongInt;
p_ContextoInicial: String)
Roteiros.pas\ (193)
To_fo_Roteiro.FormCreate (Sender:
TObject)
Roteiros.pas\ (242)
To_fo_Roteiro.o_mp_InserirNoRoteiroClick
(Sender: TObject)
Roteiros.pas\ (303)
To_fo_Roteiro.o_mp_Apresentar_ConteudoClick(Sender: TObject)
Roteiros.pas\ (428)
To_fo_Roteiro.o_o_Nos_SelecionadosClick(Sender: TObject)
Roteiros.pas\ (467)
To_fo_Roteiro.o_t_TemporizadorTimer
(Sender: TObject)
Roteiros.pas\ (525)
To_fo_Roteiro.o_dg_DesenhaRoteiro(Sender:
TObject; Col, Row: Longint; Rect:
TRect; State: TgridDrawState)
Roteiros.pas\ (559)
To_fo_Roteiro.o_dg_NoSelecionado(Sender:
TObject; Col, Row: Longint; var
CanSelect: Boolean)
Browser.pas\ (89)
To_fo_BrowserEstrutural.o_bu_AbrirJanelaClick(Sender: TObject);
Roteiros.pas\ (123)
To_fo_BrowserEstrutural.o_bu_ApresentarConteudoClick(Sender: TObject);
Roteiros.pas\ (142)
To_fo_BrowserEstrutural.o_ti_TemporizadorTimer(Sender: TObject);
Roteiros.pas\ (170)
To_fo_BrowserEstrutural.o_ol_BrowserClick(Sender: TObject);
Criarelo.pas\
(61)To_fo_CriarElos.Ler_Contexto_Origem(
Indice: LongInt; Contexto_Inicial:
String)

```



	Criarelo.pas\ (103) To_fo_CriarElos.Ler_Contexto_Destino(Ind ice: LongInt; Contexto_Inicial: String) Criarelo.pas\ (151) To_fo_CriarElos.No_Origem_Selecionado; Criarelo.pas\ (202) To_fo_CriarElos.No_Destino_Selecionado; Criarelo.pas\ (271) To_fo_CriarElos.o_ol_BrowserOrigemClick(Sender: TObject); Criarelo.pas\ (313) To_fo_CriarElos.o_ol_BrowserDestinoClick (Sender: TObject); Edit_no.pas\ (103) To_fo_EditarNo.o_bu_EditarConteudoClick(Sender: TObject); Maisinfo.pas\ (36) To_fo_MaisInformacoes.Ler_Contexto(Conte xtó_Inicial: String) Maisinfo.pas\ (136) To_fo_MaisInformacoes.o_lb_NosSelecionad osDbClick(Sender: TObject); Nos.pas\ (106) To_fo_Nos.Ler_Nos Nos.pas\ (362) To_Fo_Nos.ApresentarNoTrilha(NoSeleciona do: Ptochar); Nos.pas\ (667) To_fo_Nos.o_pm_EditarNoClick(Sender: TObject); Nos.pas\ (780) To_fo_Nos.o_lb_NosClick(Sender: TObject); Nos.pas\ (819) To_fo_Nos.o_pm_ApagarNoClick(Sender: TObject); Nos.pas\ (862) To_fo_Nos.o_pm_ApresentarConteudoDoNoCli ck(Sender: TObject); Nos.pas\ (930) To_fo_Nos.o_bu_CriarEloClick(Sender: TObject); Nos.pas\ (1079) To_fo_Nos.o_lb_ListaElosClick(Sender: TObject); Nos.pas\ (1180) To_fo_Nos.o_lb_NosMouseDown(Sender: TObject; Button: TmouseButton; Shift: TShiftState; X, Y: Integer); Nos.pas\ (1257) To_fo_Nos.o_ti_Temporizador1Timer(Sender : TObject); (1309) To_fo_Nos.o_bu_EditarEloClick(Sender: TObject);
Procedimento: (731) extern "C" int FAR PASCAL _export retira_da_hiperbase(char *nome_no) Diretório/Arquivo: hipger.cpp	Descrição: Retira um determinado nó da hiperbase Chama: (733) hb.Retorna(nome_no) (737) pontno->eh_contexto_versoes() (738) pontno->ObtemEstado() (741) hb.primeiro() (744) n->eh_contexto() (747) pontc->contem(pontno) (747) pontc->retira_no(pontno) (750) pontc->elos.primeiro()

	<pre> (753) pelo->obtem_no_origem() (754) pelo->obtem_no_destino() (755) pontc->retira_elo(pelo) (756) pontc->elos.proximo() (759) n->eh_base_privada() (760) n->retira_contexto(pontno) (762) n->eh_trilha() (765) pontt->contem(pontno) (766) pontt->retira_no(pontno) (768) n->eh_contexto_versoes() (771) pontcv->contem(pontno) (771) pontcv->retira_no(pontno) (773) pontno->eh_contexto_usuario() (776) pontc->contem(n) (778) n->retira_contexto(pontno) (785) lista_temp.insere(n) (789) pontno->eh_trilha() (792) pontt->contem(n) (794) n->retira_contexto(pontno) (801) lista_temp.insere(n) (805) hb.proximo() (807) hb.retira(pontno) (809) nos eliminados.insere(pontno) (814) hb.RetornaNo(cv) (815) pcv->nos.num_elementos() (815) hb.retira(pcv) (817) lista_temp.primeiro() (817) lista_temp.proximo() (819) retira_da_hiperbase(nos->ObtemNome()) (820) lista_temp.retira(nos) Chamado por: Hipger.cpp\ (339) extern "C" int FAR PASCAL _export retira_do_contexto(char *nome_contexto, char *nome_no) Hipger.cpp\ (731) extern "C" int FAR PASCAL _export retira_da_hiperbase(char *nome_no) Hip_prin.pas\ (464) To_fo_Principal.o im_LixoDragDrop (Sender, Source: TObject; X, Y: Integer) Roteiros.pas\ (383) To_fo_Roteiro.o b_Criar_RoteiroClick (Sender: TObject) Nos.pas\ (819) To_fo_Nos.o pm_ApagarNoClick(Sender: TObject) </pre>
Procedimento: (832) extern "C" int FAR PASCAL _export mostra_primeiro_lixo(char *nome) Diretório/Arquivo: hipger.cpp	Descrição: Recupera o nome do primeiro nó que foi colocado no lixo Retorna: 1: Se a lixeira estiver vazia. 0: Caso contrário. Chama: (834) os eliminados.primeiro() (837) primeiro_lixo->ObtemNome() Chamado por:
Procedimento: (846) extern "C" int FAR PASCAL _export mostra_proximo_lixo(char *nome) Diretório/Arquivo: hipger.cpp	Descrição: Recupera o nome do próximo nó que foi colocado no lixo Retorna: 1: Se a lixeira estiver vazia. 0: Caso contrário. Chama: (848) nos eliminados.proximo() (852) proximo_lixo->ObtemNome() Chamado por:

Procedimento: (866) extern "C" int FAR PASCAL _export muda_atributos(char *nome_velho, char *nome_novo, int num_estado) Diretório/Arquivo: hipger.cpp	Descrição: Altera os atributos do nó, trocando o nome e/ou o estado que se encontra o nó. Os números válidos para trocar de estado são: 1: se o nó for Temporário este passará para Permanente 2: se o nó for Permanente este passará para Obsoleto Retorna: 1: O nome passado como parâmetro não consta na hiperbase. 2: O nome do nó não pode ser alterado. 3: O nó não é Temporário. Tentou passar de Temporário para Permanente, mas o nó não é Temporário. 4: O nó não é Permanente. Tentou passar de Permanente para Obsoleto, mas o nó não é Permanente. 0: Ok. Chama: (867) hb.Retorna(nome_velho) (871) n->MudaNome(nome_novo) (875) n->commit() (878) n->obsoleto() Chamado por: Nos.pas\ (667) To fo_Nos.o_pm_EditarNoClick(Sender: TObject)
Procedimento: (890) extern "C" int FAR PASCAL _export muda_tipo(char *nome_no, char *tipo) Diretório/Arquivo: hipger.cpp	Descrição: Retorna: 1: O nome passado como parâmetro não consta na hiperbase. 2: O nó não é um nó terminal 3: O tipo ultrapassou o tamanho maximo. 0: Ok. Chama: (892) hb.Retorna(nome_no) (895) n->eh_terminal() (897) muda_tipo(tipo) Chamado por: Aparece em funcoes.pas e hip_prin.pas, mas apenas como funcao declarada. VERIFICAR!!!
Procedimento: (908) extern "C" int FAR PASCAL _export recupera_tipo(char *nome_no, char *tipo) Diretório/Arquivo: hipger.cpp	Descrição: Retorna: 1: O nome passado como parâmetro não consta na hiperbase. 2: O nó não é um nó terminal 0: Ok. Chama: (910) hb.Retorna(nome_no) (913) n->eh_terminal() (915) recupera_tipo() Chamado por: Aparece em funcoes.pas e hip_prin.pas, mas apenas como funcao declarada. VERIFICAR!!!
Procedimento: (927) extern "C" int FAR PASCAL _export muda_funcao_didatica(char *nome_no, char *categoria)	Descrição: Retorna: 1: O nome passado como parâmetro não consta na hiperbase. 2: O nó não é um nó terminal 3: A categoria ultrapassou o tamanho maximo. 0: Ok.

Diretório/Arquivo: hipger.cpp	Chama: (929) hb.Retorna(nome_no) (932) n->eh_terminal() (934) muda_funcao_didatica(categoria) Chamado por: Nos.pas\ (557) To_fo_Nos.o_pm_CriarNoClick(Sender: TObject) Nos.pas\ (667) To_fo_Nos.o_pm_EditarNoClick(Sender: TObject)
Procedimento: (945) extern "C" int FAR PASCAL _export recupera_funcao_didatica(char *nome_no, char *categoria) Diretório/Arquivo: hipger.cpp	Descrição: Retorna: 1: O nome passado como parâmetro não consta na hiperbase. 2: O nó não é um nó terminal 0: Ok. Chama: (947) hb.Retorna(nome_no) (950) n->eh_terminal() (952) recupera_funcao_didatica() Chamado por: Atexto.pas\ (176) To_fo_Texto.o_hv_TextoHotSpotClick(Sender: TObject; const SRC: String; var Handled: Boolean) Nos.pas\ (780) To_fo_Nos.o_lb_NosClick(Sender: TObject) Nos.pas\ (667) To_fo_Nos.o_pm_EditarNoClick(Sender: TObject) Nos.pas\ (135) To_fo_Nos.Visualizar_No_Terminal(Tipo: char; Nome_No: Ptochar) Edit_no.pas\ (103) To_fo_EditarNo.o_bu_EditarConteudoClick(Sender: TObject) Criarelo.pas\ (313) To_fo_CriarElos.o_ol_BrowserDestinoClick(Sender: TObject) Criarelo.pas\ (271) To_fo_CriarElos.o_ol_BrowserOrigemClick(Sender: TObject) Browser.pas\ (170) To_fo_BrowserEstrutural.o_ol_BrowserClick(Sender: TObject) Roteiros.pas\ (559) To_fo_Roteiro.o_dg_NoSelecionado(Sender: TObject; Col, Row: Longint; var CanSelect: Boolean) Roteiros.pas\ (428) To_fo_Roteiro.o_o_Nos_SelecionadosClick(Sender: TObject) Roteiros.pas\ (94) To_fo_Roteiro.SelecionarNos(p_IndiceInicial: LongInt; p_ContextoInicial: String) Controle.pas\ (467) To_fo_Conroles.o_bu_ExerciciosClick(Sender: TObject) Controle.pas\ (419) To_fo_Conroles.o_bu_BibliografiaClick(Sender: TObject) Controle.pas\ (371)To_fo_Conroles.o_bu_AjudaClick(Sender: TObject)
Procedimento:	Descrição:

<pre>(964) extern "C" int FAR PASCAL _export muda_assunto(char *nome_no, char *assunto)</pre> Diretório/Arquivo: hipger.cpp	Retorna: 1: O nome passado como parâmetro não consta na hiperbase. 2: O nó não é um nó de composição 3: O assunto ultrapassou o tamanho maximo. 0: Ok. Chama: <pre>(966) hb.Retorna(nome_no) (969) n->eh_contexto() (969) n->eh_trilha() (970)muda_assunto(assunto)</pre> Chamado por: <pre>Nos.pas\ (667) To_fo_Nos.o_pm_EditarNoClick(Sender: TObject)</pre>
Procedimento: <pre>(979) extern "C" int FAR PASCAL _export recupera_assunto(char *nome_no, char *assunto)</pre> Diretório/Arquivo: hipger.cpp	Descrição: Retorna: 1: O nome passado como parâmetro não consta na hiperbase. 2: O nó não é um nó contexto 0: Ok. Chama: <pre>(981) hb.Retorna(nome_no) (984) n->eh_contexto() (985) recupera_assunto()</pre> Chamado por: <pre>Roteiros.pas\ (94) To_fo_Roteiro.SelecionarNos (p_IndiceInicial: LongInt; p_ContextoInicial: String) Roteiros.pas\ (428) To_fo_Roteiro.o_o_Nos_SelecionadosClick(Sender: TObject)</pre>
Procedimento: <pre>(993) extern "C" int FAR PASCAL _export muda_dificuldade(char *nome_no, int nivel)</pre> Diretório/Arquivo: hipger.cpp	Descrição: Retorna: 1: O nome passado como parâmetro não consta na hiperbase. 0: Ok. Chama: <pre>(995) hb.Retorna(nome_no) (999) no->muda_dificuldade(nivel)</pre> Chamado por: <pre>Nos.pas\ (557) To_fo_Nos.o_pm_CriarNoClick(Sender: TObject) Nos.pas\ (667) To_fo_Nos.o_pm_EditarNoClick(Sender: TObject)</pre>
Procedimento: <pre>(1009) extern "C" int FAR PASCAL _export recupera_dificuldade(char *nome_no, int *nivel)</pre> Diretório/Arquivo: hipger.cpp	Descrição: Retorna: 1: O nome passado como parâmetro não consta na hiperbase. 0: Ok. Chama: <pre>(1011) hb.Retorna(nome_no) (1015) no->recupera_dificuldade()</pre> Chamado por: <pre>Roteiros.pas\ (94) To_fo_Roteiro.SelecionarNos (p_IndiceInicial: LongInt; p_ContextoInicial: String) Roteiros.pas\ (428) To_fo_Roteiro.o_o_Nos_SelecionadosClick(Sender: TObject) Roteiros.pas\ (559) To_fo_Roteiro.o dg NoSelecionado(Sender:</pre>

	TObject; Col, Row: Longint; var CanSelect: Boolean) Browser.pas\ (170) To_fo_BrowserEstrutural.o_ol_BrowserClick(Sender: TObject) Criarelo.pas\ (271) To_fo_CriarElos.o_ol_BrowserOrigemClick(Sender: TObject) Criarelo.pas\ (313) To_fo_CriarElos.o_ol_BrowserDestinoClick(Sender: TObject) Nos.pas\ (667) To_fo_Nos.o_pm_EditarNoClick(Sender: TObject) Nos.pas\ (780) To_fo_Nos.o_lb_NosClick(Sender: TObject)
Procedimento: (1028) extern "C" int FAR PASCAL _export muda_palavra_chave(char *nome_no, int pos, char *palavra) Diretório/Arquivo: hipger.cpp	Descrição: Retorna: 1: O nome passado como parâmetro não consta na hiperbase. 2: O nó não é um nó terminal 3: A posicao nao é válida ou palavra excedeu o tamanho máximo. 0: Ok. Chama: (1030) hb.Retorna(nome_no) (1033) n->eh_terminal() (1035) muda_palavras_chaves(pos, palavra) Chamado por: Nos.pas\ (557) To_fo_Nos.o_pm_CriarNoClick(Sender: TObject) Nos.pas\ (667) To_fo_Nos.o_pm_EditarNoClick(Sender: TObject)
Procedimento: (1046) extern "C" int FAR PASCAL _export recupera_palavra_chave(char *nome_no, int pos, char *palavra) Diretório/Arquivo: hipger.cpp	Descrição: Retorna: 1: O nome passado como parâmetro não consta na hiperbase. 2: O nó não é um nó terminal 0: Ok. Chama: (1048) hb.Retorna(nome_no) (1051) n->eh_terminal() (1053) recupera_palavras_chaves(pos) Chamado por: Roteiros.pas\ (94) To_fo_Roteiro.SelecionarNos(p_IndiceInicial: LongInt; p_ContextoInicial: String) Maisinfo.pas\ (36) To_fo_MaisInformacoes.Ler_Contexto(Contexto_Inicial: String) Nos.pas\ (667) To_fo_Nos.o_pm_EditarNoClick(Sender: TObject)
Procedimento: (1067) extern "C" int FAR PASCAL _export regras_tutoriais(char *no_atual, char *escopo, char *historico, int dificuldade, char *no_result, char *lib_result, int *revisto) Diretório/Arquivo: hipger.cpp	Descrição: Retorna: 1: Os nomes dos nós passado como parâmetro não consta na hiperbase. 2: O nó escopo não é um nó contexto. 3: O nó histórico não é um nó trilha. 4: O nível de dificuldade não é válido. 5: Não encontrou o nó resultado. 0: Ok. Chama:

	<pre> (1070) hb.Retorna(no_atual (1072) regras_tutor.muda_no_atual(n) (1073) hb.Retorna(historico) (1075) tri->eh_trilha() (1076) regras_tutor.muda_historico(tri) (1079) regras_tutor.muda_nivel(Botao_Facil) (1081) regras_tutor.muda_nivel(Botao_Dificil) (1085) hb.Retorna(escopo) (1087) cont->eh_contexto() (1088) regras_tutor.primeira_estrategia(cont)) (1090) regras_tutor.recupera_no_atual() (1092) n->ObtemNome() (1094) regras_tutor.jah_foi_visto() (1098) regras_tutor.segunda_estrategia(cont) (1100) regras_tutor.recupera_no_atual() (1102) n->ObtemNome() (1103) cont->ObtemNome() (1104) regras_tutor.jah_foi_visto() Chamado por: Controle.pas\ (273) To_fo_Controles.o_bu_FacilClick (Sender: TObject) Controle.pas\ (322) To_fo_Controles.o_bu_DificilClick (Sender: TObject) </pre>
Procedimento: <pre> (1121) extern "C" int FAR PASCAL _export cria_versao(char *nome_no, char *nome_no_versao) </pre> Diretório/Arquivo: hipger.cpp	Descrição: Retorna: 1: O primeiro nome passado como parâmetro não consta na hiperbase. 2: O primeiro nome é um contexto de versão. 3: O nó não é temporário, portanto, não pode ser versionável. 4: O segundo nome passado como parâmetro já existe na hiperbase. Chama: <pre> (1123) hb.Retorna(nome_no) (1126) pontno->eh_contexto_versoes() (1127) pontno->ObtemEstado() (1128) hb.Retorna(nome_no_versao) (1129) pontno- >criaversao(nome_no_versao) Chamado por: </pre>
Procedimento: <pre> (1138) extern "C" int FAR PASCAL _export consulta_vcorrente(char *nome_cv, char *nome_no) </pre> Diretório/Arquivo: hipger.cpp	Descrição: Retorna: 1: O nome passado como parâmetro não consta na hiperbase. 2: O nó não é um Contexto de Versões. Chama: <pre> (1140) hb.Retorna(nome_cv) (1143) eh_contexto_versoes() </pre>
	Chamado por: NÃO APARECE NEM NOS PROPÓTIPOS DA DLL
Procedimento: <pre> (1154) extern "C" int FAR PASCAL _export muda_vcorrente(char *nome_cv, char *nome_no) </pre> Diretório/Arquivo: hipger.cpp	Descrição: Retorna: 1: O primeiro nome passado como parâmetro não consta na hiperbase. 2: O nó não é um contexto de versões. 3: O segundo nome passado como parâmetro não consta na hiperbase. 4: O nó contexto de versões não contém o nó do segundo parâmetro. Chama:

	<pre>(1156) hb.Retorna(nome_cv) (1160) pcv->eh_contexto_versoes() (1163) pcv->contem(pontno) (1164) pcv->MudaVersaoCorrente(pontno->obtem_identificador()) Chamado por:</pre>
Procedimento: <pre>(1181) extern "C" int FAR PASCAL _export cria_ancora_texto(char *nome_no, int ident, char *palavra, int inicio, int fim)</pre> Diretório/Arquivo: hipger.cpp	Descrição: Cria uma âncora texto em um nó texto Retorna: 1: O nome passado como parâmetro não consta na hiperbase. 2: O nó não é temporário. Portanto, não pode ser alterado. 3: Não se pode criar ancoras em um nó versão. 4: O nó não é do tipo texto. 5: Falta memória para a âncora. 0: Ok. Chama: <pre>(1183) hb.Retorna(nome_no) (1186) (pontno->ObtemEstado()) (1187) pontno->ObtemCV() (1188) pontno->eh_texto() (1189) pontno->cria_ancora_texto(ident, palavra, inicio, fim)</pre> Chamado por: Aparece em funcoes.pas e hip_prin.pas, mas apenas como funcao declarada. VERIFICAR!!!
Procedimento: <pre>(1203) extern "C" int FAR PASCAL _export cria_ancora_contexto(char *nome_no, int ident, char *no_ancorado)</pre> Diretório/Arquivo: hipger.cpp	Descrição: Cria uma âncora contexto em um nó contexto Retorna: 1: Não existe nó contexto com esse nome na hiperbase. 2: Esse nó não pode ser alterado. 3: Não se pode criar ancoras em um nó versão. 4: O nó não é do tipo contexto. 5: Nó não pertence ao contexto ou não existe. 6: Falta memória para criar a âncora. 0: Ok. Chama: <pre>(1206) hb.Retorna(nome_no) (1207) hb.Retorna(no_ancorado) (1209) pontnc->ObtemEstado() (1210) pontnc->ObtemCV() (1211) pontnc->eh_contexto() (1212) pontno==NULL !pontnc->contem(pontno)) (1213) pontnc->cria_ancora_contexto(ident, pontno->obtem_identificador())</pre> Chamado por: Aparece em funcoes.pas e hip_prin.pas, mas apenas como funcao declarada. VERIFICAR!!!
Procedimento: <pre>(1227) extern "C" int FAR PASCAL _export retira_ancora(char *nome, int ident)</pre>	Descrição: Retira uma âncora da lista de um determinado nó (texto ou contexto) Retorna: 1: Não existe nó com esse nome na hiperbase. 2: Não se pode retirar ancoras de um nó versão. 3: Esse nó não pode ser alterado.

Diretório/Arquivo: hipger.cpp	4: Não existe âncora com este identificador. 5: Erro na estrutura da lista de âncoras. 0: Ok. Chama: (1229) hb.Retorna(nome) (1233) pontno->ObtemCV() (1234) pontno->ObtemEstado() (1235) pontno->retorna_ancora(ident) (1237) pontno->retira_ancora(pontanc) Chamado por: Aparece em funcoes.pas e hip_prin.pas, mas apenas como funcao declarada. VERIFICAR!!!
Procedimento: (1246) extern "C" int FAR PASCAL _export recupera_identificador_interno(char *nome, int ide) Diretório/Arquivo: hipger.cpp	Descrição: Dado o nome do nó e o identificador externo da âncora esta função retorna o identificador que é utilizado internamente pelo sistema Retorna: -1 Não existe nó com esse nome na hiperbase. caso contrário: Retorna o identificador interno da âncora. Chama: (1248) hb.Retorna(nome) (1251) pontno->obtem_indice(ide) Chamado por: Nos.pas\ (930) To_fo_Nos.o_bu_CriarEloClick(Sender: TObject) Nos.pas\ (1309) To_fo_Nos.o_bu_EditarEloClick(Sender: TObject)
Procedimento: (1260) extern "C" int FAR PASCAL _export muda_ancora_texto(char *nome, int velho_ide, int novo_ide, char *palavra, int inicio, int fim) Diretório/Arquivo: hipger.cpp	Descrição: Retorna: 1: Não existe nó com esse nome na hiperbase. 2: O nó não é um nó do tipo texto. 3: Não existe âncora com este identificador. 0: Ok. Chama: (1262) hb.Retorna(nome) (1265) pontno->eh_texto() (1266) pontno->retorna_ancora(pontno->obtem_indice(velho_ide) (1268) pontno->muda_ancora_texto(panc, novo_ide, palavra, inicio, fim) Chamado por: Aparece em funcoes.pas e hip_prin.pas, mas apenas como funcao declarada. VERIFICAR!!!
Procedimento: (1279) extern "C" int FAR PASCAL _export muda_ancora_contexto(char *nome, int velho_ide, int novo_ide, char *no_ancorado) Diretório/Arquivo: hipger.cpp	Descrição: Retorna: 1: Não existe nó com esse nome na hiperbase. 2: O nó não é um nó do tipo contexto. 3: Não existe âncora com este identificador. 4: Nó não pertence ao contexto ou não existe. 0: Ok. Chama: (1281) hb.Retorna(nome) (1282) hb.Retorna(no_ancorado)

	<pre>(1285) pontnc->eh_contexto() (1286) pontnc->retorna_ancora(pontno->obtem_indice(velho_ide)) (1288) pontnc->contem(pontno) (1289) pontnc->muda_ancora_contexto(panc, novo_ide, pontno->obtem_identificador())</pre> Chamado por: Aparece em funcoes.pas e hip_prin.pas, mas apenas como funcao declarada. VERIFICAR!!!
Procedimento: <pre>(1297) extern "C" int FAR PASCAL _export numero_de_ancoras(char *nome) Diretório/Arquivo: hipger.cpp</pre>	Descrição: Dado o nome de um nó esta rotina retorna o número total de âncoras que ele contém. Se retornar -1: Ocorreu erro. Chama: <pre>(1299) hb.Retorna(nome) (1302) pontno->ancoras.num_elementos()</pre> Chamado por: Aparece em funcoes.pas e hip_prin.pas, mas apenas como funcao declarada. VERIFICAR!!!
Procedimento: <pre>(1311) extern "C" int FAR PASCAL _export recupera_nome_ancora(char *nome, int id_ancora, char *nome_ancora) Diretório/Arquivo: hipger.cpp</pre>	Descrição: Esta rotina funciona apenas para âncoras textos. Dado o identificador interno de uma âncora, esta rotina retorna o nome marcado pelo usuário como âncora. Retorna: Char *nome_ancora: nome marcado pelo usuário no texto como âncora. 1: Não existe nó com esse nome na hiperbase. 2: Não existe âncora com este identificador. 0: Ok. Chama: <pre>(1313) hb.Retorna(nome) (1316) pontno->ancoras.primeiro() (1319) anc->obtem_identificador() (1321) pontno->eh_texto() (1324) at->obtem_ancora() (1327) pontno->eh_contexto() (1330) hb.RetornaNo(ac->obtem_ancora()) (1334) pontno->eh_audio() (1336) anc->obtem_identificador() (1339) pontno->eh_imagem() (1341) anc->obtem_identificador() (1344) pontno->eh_video() (1346) anc->obtem_identificador() (1349) pontno->eh_executavel() (1351) anc->obtem_identificador() (1354) pontno->eh_tabela() (1356) anc->obtem_identificador() (1360) pontno->ancoras.proximo()</pre> Chamado por: Atexto.pas\ (52) To_fo_Texto.Procurar_Elo(Ancora: string; NomeDoNoLiberdade: string): boolean Atexto.pas\ (176) To_fo_Texto.o_hv_TextoHotSpotClick(Sender: TObject; const SRC: String; var Handled: Boolean) Nos.pas\ (291) To_fo_Nos.Ler_Elos
Procedimento: <pre>(1372) extern "C" int FAR PASCAL _export primeira_ancora(char *nome, char *nome_ancora)</pre>	Descrição: Retorna: Char *nome_ancora: nome marcado pelo usuário no texto como âncora.

Diretório/Arquivo: hipger.cpp	1: Não existe nó com esse nome na hiperbase. 2: O nó não tem âncoras. 0: Ok. Chama: (1374) hb.Retorna(nome) (1378) pontno->primeira_ancora() (1379) pontno->eh_texto() (1381) pontno->primeira_ancora() (1382) primeira->obtem_ancora() (1384) pontno->eh_contexto() (1386) pontno->primeira_ancora() (1387) hb.RetornaNo(primeira->obtem_ancora()) (1389) pno->ObtemNome() (1391) pontno->eh_audio() (1394) pontno->primeira_ancora() (1395) primeira->obtem_identificador() (1401) pontno->eh_video() (1403) pontno->primeira_ancora() (1404) primeira->obtem_identificador() (1406) pontno->eh_executavel() (1408) pontno->primeira_ancora() (1409) primeira->obtem_identificador() (1411) pontno->eh_tabela() (1413) pontno->primeira_ancora() (1414) primeira->obtem_identificador() Chamado por: Criarelo.pas\ (151) To fo CriarElos.No Origem Selecionado Criarelo.pas\ (202) To fo CriarElos.No Destino Selecionado
Procedimento: (1425) extern "C" int FAR PASCAL _export proxima_ancora(char *nome, char *nome_ancora) Diretório/Arquivo: hipger.cpp	Descrição: Retorna: char *nome_ancora: nome marcado pelo usuário no texto como âncora. 1: Não existe nó com esse nome na hiperbase. 2: Fim da lista de âncoras. 0: Ok. Chama: (1427) hb.Retorna(nome) (1430) pontno->eh_texto() (1432) pontno->proxima_ancora() (1434) proxima->obtem_ancora() (1436) pontno->eh_contexto() (1438) pontno->proxima_ancora() (1440) hb.RetornaNo(proxima->obtem_ancora()) (1441) pontno->contem(pno) (1443) pontno->retira_ancora(proxima) (1447) pno->ObtemNome() (1449) pontno->eh_audio() (1451) pontno->proxima_ancora() (1453) proxima->obtem_identificador() (1455) pontno->eh_imagem() (1457) pontno->proxima_ancora() (1459) proxima->obtem_identificador() (1461) pontno->eh_video() (1463) pontno->proxima_ancora() (1465) proxima->obtem_identificador() (1467) pontno->eh_executavel() (1469) pontno->proxima_ancora() (1471) proxima->obtem_identificador() (1473) pontno->eh_tabela() (1475) pontno->proxima_ancora() (1477) proxima->obtem_identificador()

	Chamado por: Criarelo.pas\ (151) To fo CriarElos.No Origem Selecionado Criarelo.pas\ (202) To fo CriarElos.No Destino Selecionado
Procedimento: (1489) extern "C" int FAR PASCAL _export ultima_ancora(char *nome, char *nome_ancora) Diretório/Arquivo: hipger.cpp	Descrição: Retorna: char *nome_ancora: nome marcado pelo usuário no texto como âncora. 1: Não existe nó com esse nome na hiperbase. 2: O nó não é um nó do tipo texto. 3: O nó não tem âncoras. 0: Ok. Chama: (1491) hb.Retorna(nome) (1494) pontno->eh_texto() (1495) pontno->ultima_ancora() (1497) ultima->obtem_ancora() Chamado por:
Procedimento: (1517) extern "C" int FAR PASCAL _export cria_elo(char *nome_contexto, int id_elo, char *origem, int anc_origem, char *destino, int anc_destino, int simultaneo, int mostra_conteudo) Diretório/Arquivo: hipger.cpp	Descrição: Dado o nome do nó contexto e dois nós com âncoras que estão neste contexto, esta rotina é responsável pela criação de um elo entre os dois nós Retorna: 1: Esse contexto não existe na hiperbase. 2: O Nó origem não existe na hiperbase. 3: O Nó destino não existe na hiperbase. 4: Esse nó não é contexto. 5: O contexto de versões é de responsabilidade do sistema. 6: Não conseguiu alocar memória para o elo. 7: O nó origem e o nó destino são os mesmos. 0: Ok. Chama: (1522) hb.Retorna(nome_contexto) (1529) hb.RetornaNo(atoi(ident)) (1534) hb.RetornaNo(atoi(ident)) (1537) pontnoc->eh_contexto() (1538) pontnoc->eh_contexto_versoes() (1542) elo(id_elo, no_origem, origem, anc_origem, no_destino, destino, anc_destino, simultaneo, mostra_conteudo) (1544) pontnoc->insere_elo(E Chamado por: Nos.pas\ (930) To fo Nos.o_bu_CriarEloClick(Sender: TObject)
Procedimento: (1560) extern "C" int FAR PASCAL _export muda_elo(char *nome_contexto, int id_elo, char *origem, int anc_origem, char *destino, int anc_destino, int simultaneo, int mostra_conteudo)	Descrição: Retorna: 1: Esse contexto não existe na hiperbase. 2: O Nó origem não existe na hiperbase. 3: O Nó destino não existe na hiperbase. 4: Esse nó não é contexto. 5: O contexto de versões é de responsabilidade do sistema. 6: Não conseguiu alocar memória para o elo. 7: O nó origem e o nó destino são os mesmos. 0: Ok.

Diretório/Arquivo: hipger.cpp	Chama: <pre>(1565) hb.Retorna(nome_contexto) (1572) hb.RetornaNo(atoi(ident)) (1577) hb.RetornaNo(atoi(ident)) (1580) pontc->eh_contexto() (1581) pontc->eh_contexto_versoes() (1585) pontc->retorna_elo(id_elo) (1587) pelo->muda_no_origem(no_origem) (1588) pelo->muda_no_destino(no_destino) (1589) pelo- >muda_ancora_origem(anc_origem) (1590) pelo- >muda_ancora_destino(anc_destino) (1591) pelo- >muda_aninhamento_origem(origem) (1592) pelo- >muda_aninhamento_destino(destino) (1593) pelo->muda_simultaneo(simultaneo) (1594) pelo- >muda_mostra_conteudo_contexto(mostra_conteudo)</pre> Chamado por: Nos.pas\ (1309) To_fo_Nos.o_bu_EditarEloClick(Sender: TObject)
Procedimento: <pre>(1608) extern "C" int FAR PASCAL _export retira_elo(char *nome_contexto, int id_elo)</pre> Diretório/Arquivo: hipger.cpp	Descrição: Dado o nome do nó contexto e o identificador do elo contido no nó contexto, esta rotina retira o elo da lista de elos do nó contexto Retorna: 1: Não existe nó contexto com esse nome na hiperbase. 2: Esse nó não é contexto. 3: O contexto de versões é de responsabilidade do sistema. 4: Esse nó não pode ser alterado. 5: Não existe esse elo no contexto. 6: Erro na estrutura da lista de elos. 0: Ok. Chama: <pre>(1611) hb.Retorna(nome_contexto) (1613) pontc->eh_contexto() (1614) pontc->eh_contexto_versoes() (1615) pontc->ObtemEstado() (1616) pontc->retorna_elo(id_elo) (1618) pontc->retira_elo(pontelo)</pre> Chamado por: Nos.pas\ (1049) To_fo_Nos.o_bu_ApagarEloClick(Sender: TObject)
Procedimento: <pre>(1627) extern "C" int FAR PASCAL _export numero_de_elos(char *nome_contexto)</pre> Diretório/Arquivo: hipger.cpp	Descrição: Dado o nome de um nó, esta rotina retorna o total de âncoras que ele contém Retorna: Retorna o número total de elos. -1: Ocorreu erro. Chama: <pre>(1629) hb.Retorna(nome_contexto) (1632) pontc->elos.num_elementos()</pre> Chamado por: Atexto.pas\ (52) To_fo_Texto.Procurar_Elo(Ancora: string; NomeDoNoLiberdade: string): boolean Nos.pas\ (291) To_fo_Nos.Ler_Elos Nos.pas\ (930)

Procedimento: <pre>(1647) extern "C" int FAR PASCAL _export recupera_elo(char *nome_contexto, int id_elo, char *id_no_orig, int *id_anc_orig, char *id_no_dest, int *id_anc_dest, int *simultaneo, int *mostra_conteudo)</pre> Diretório/Arquivo: hipger.cpp	To_fo_Nos.o_bu_CriarEloClick(Sender: TObject) Descrição: Dado o nome do nó contexto e o identificador do elo contido no nó contexto, esta rotina retorna a lista dos nós aninhados origem e destino e as âncoras origem e destino Retorna: char *id_no_orig: lista de identificadores dos nós aninhados na origem. int *id_anc_orig: identificador interno da âncora origem. char *id_no_dest: lista de identificadores dos nós aninhados no destino. int *id_anc_dest: identificador interno da âncora destino. int *simultaneo: indica se o nós origem e destinos são mostrados simultaneamente. 1: Não existe nó contexto com esse nome na hiperbase. 2: O nó não é um nó do tipo contexto. 3: Não existe elo com este identificador. 0: Ok.. Chama: <pre>(1649) hb.Retorna(nome_contexto) (1652) pontc->eh_contexto_usuario() (1653) pontc->retorna_elo(id_elo) (1656) pelo->obtem_aninhamento_origem() (1657) pelo->obtem_id_ancora_origem() (1658) pelo->obtem_aninhamento_destino() (1659) pelo->obtem_id_ancora_destino() (1660) pelo->eh_simultaneo() (1661) Pelo->eh_para_mostrar() Chamado por: Atexto.pas\ (52) To_fo_Texto.Procurar_Elo(Ancora: string; NomeDoNoLiberdade: string): boolean Atexto.pas\ (176) To_fo_Texto.o_hv_TextoHotSpotClick(Sende r: TObject; const SRC: String; var Handled: Boolean) Nos.pas\ (291) To_fo_Nos.Ler_Elos Nos.pas\ (1079) To_fo_Nos.o_lb_ListaElosClick(Sender: TObject) Nos.pas\ (1309) To_fo_Nos.o_bu_EditarEloClick(Sender: TObject)</pre>
Procedimento: <pre>(1672) extern "C" int FAR PASCAL _export primeiro_elo(char *nome, int *id)</pre> Diretório/Arquivo: hipger.cpp	Descrição: Retorna: 1: Não existe nó com esse nome na hiperbase. 2: O nó não tem âncoras. 0: Ok. Chama: <pre>(1674) hb.Retorna(nome) (1677) pontno->eh_contexto() (1678) pontno->primeiro_elo(id)</pre> Chamado por: <pre>Nos.pas\ (291) To_fo_Nos.Ler_Elos</pre>
Procedimento: <pre>(1687) extern "C" int FAR PASCAL</pre>	Descrição: Retorna:

<pre>_export proximo_elo(char *nome, int *id) Diretório/Arquivo: hipger.cpp</pre>	<pre>1: Não existe nó com esse nome na hiperbase. 2: Fim da lista de âncoras. 0: Ok. Chama: (1688) hb.Retorna(nome) (1691) pontno->eh_contexto() (1692) pontno->proximo_elo(id) Chamado por: Nos.pas\ (291) To fo Nos.Ler Elos</pre>
<pre>Procedimento: (1697) elo *FAR PASCAL _export primeiro_ponteiro_elo(char *nome_contexto) Diretório/Arquivo: hipger.cpp</pre>	<pre>Descrição: Chama: (1699) hb.Retorna(nome_contexto) (1702) pontno->eh_contexto() (1703) pontno->elos.primeiro() (1704) (pontno->eh_trilha()) (1707) tri->elos.primeiro() Chamado por: Grafoob.cpp\ (29) elo *PrimeiroElo(char *nome)</pre>
<pre>Procedimento: (1714) elo *FAR PASCAL _export proximo_ponteiro_elo(char *nome_contexto, elo *pelo) Diretório/Arquivo: hipger.cpp</pre>	<pre>Descrição: Chama: (1716) hb.Retorna(nome_contexto) (1719) pontno->eh_contexto() (1720) pontno->elos.proximo(pelo) (1721) pontno->eh_trilha() (1724) tri->elos.proximo(pelo) Chamado por: Grafoob.cpp\ (30) elo *ProximoElo(char *nome, elo *e)</pre>

Sistema: SASHE Arquivo: Sashe\Hyperprop\ List_anc.cpp	
<pre>Procedimento: (29) lista_ancoras::~lista_ancoras() Diretório/Arquivo: List_anc.cpp</pre>	<pre>Descrição: destrutor da lista Chama: (35) proxdado->elem (36) anc->eh_ancora_contexto() (41) anc->eh_ancora_texto() (46) anc->eh_ancora_audio() (51) anc->eh_ancora_video() (56) anc->eh_ancora_imagem() (62) proxdado->proximo Chamado por:</pre>
<pre>Procedimento: (71) int lista_ancoras :: grava(FILE *fp) Diretório/Arquivo: List_anc.cpp</pre>	<pre>Descrição: grava a lista de ancoras Chama: (73) num_elementos() (74) primeiro() (80) anc->eh_ancora_contexto() (83) ancc->grava_ancora(fp) (85) anc->eh_ancora_texto() (88) anct->grava_ancora(fp) (90) anc->eh_ancora_audio() (93) anca->grava_ancora(fp) (95) anc->eh_ancora_video() (98) ancv->grava_ancora(fp) (100) anc->eh_ancora_imagem()</pre>

	<pre> (103) anci->grava_ancora(fp) (105) anc->eh_ancora_trilha() (108) anct->grava_ancora(fp) (110) proximo() Chamado por: No.cpp\ (307) int No::grava_no(FILE *fp) </pre>
Procedimento: (118) int lista_ancoras :: recupera(No *n, FILE *fp) Diretório/Arquivo: List_anc.cpp	Descrição: recupera a lista de ancoras Chama: <pre> (122) fscanf(fp, "\tListaAncora(%d)\n", &ne) (126) fscanf(fp, "\t\tIdentificadores(%d,", &indice) (127) n->eh_contexto_usuario() (127) n->eh_contexto_versoes() (130) ancora_contexto(indice) (133) ac->recupera_ancora(fp) (134) nc->insere_ancora(ac) (136) n->eh_texto() (139) ancora_texto(indice) (142) ac->recupera_ancora(fp) (143) nt->insere_ancora(ac) (145) n->eh_audio() (148) ancora_audio(indice) (151) ac->recupera_ancora(fp) (152) na->insere_ancora(ac) (154) n->eh_video() (157) ancora_video(indice) (160) ac->recupera_ancora(fp) (161) nv->insere_ancora(ac) (163) n->eh_imagem() (166) ancora_imagem(indice) (169) ac->recupera_ancora(fp) (170) ni->insere_ancora(ac) (172) n->eh_trilha() (175) ancora_trilha(indice) (179) ac->recupera_ancora(fp) (180) nt->insere_ancora(ac) Chamado por: No.cpp\ (318) int No::recupera_no(FILE *fp) </pre>

Sistema: SASHE**Arquivo:** Sashe\Hyperprop\ List_anc.h

Procedimento: (20) int insere(ancora *pontancora) Diretório/Arquivo: List_anc.h	Descrição: Chama: (20) lista::insere(pontancora) Chamado por: No.cpp\ (77) int insere_ancora(ancora *p)
Procedimento: (21) int retira(ancora *pontancora) Diretório/Arquivo: List_anc.h	Descrição: Chama: (21) lista::retira(pontancora) Chamado por: No.cpp\ (78) int retira_ancora(ancora *p)
Procedimento: (22) ancora * primeiro() Diretório/Arquivo: List_anc.h	Descrição: Chama: (22) lista::primeiro() Chamado por: Hipger.cpp\ (1311) extern "C" int FAR PASCAL _export recupera_nome_ancora(char *nome, int id ancora, char *nome ancora)

	<pre> List_anc.cpp\ (71) int lista_ancoras :: grava(FILE *fp) No.cpp\ (94) int No::contem_ancora(ancora *pnc) No.cpp\ (107) ancora * No::retorna_ancora(int ind) No.cpp\ (120) int No::obtem_indice(int ide) No.cpp\ (243) ancora *No::primeira_ancora() No_audio.cpp\ (55) void no_audio::copia_ancoras_audio(lista_ancoras *ancoras) No_exec.cpp\ (55) void no_executavel::copia_ancoras_executavel(lista_ancoras *ancoras) No_image.cpp\ (56) void no_imagem::copia_ancoras_imagem(lista_ancoras *ancoras) No_tab.cpp\ (55) void no_tabela::copia_ancoras_tabela(lista_ancoras *ancoras) No_texto.cpp\ (60) void no_texto::copia_ancoras_texto(lista_ancoras *ancoras) No_video.cpp\ (54) void no_video::copia_ancoras_video(lista_ancoras *ancoras) </pre>
Procedimento: (23) ancora * ultimo()	Descrição:
Diretório/Arquivo: List_anc.h	Chama: (23) lista::ultimo()
	Chamado por: No.cpp\ (133) int No::retorna_indice() No.cpp\ (250) ancora *No::ultima_ancora()
Procedimento: (24) ancora * proximo() Diretório/Arquivo: List_anc.h	Descrição: Chama: (24) lista::proximo() Chamado por: Hipger.cpp\ (1311) extern "C" int FAR PASCAL _export recupera_nome_ancora(char *nome, int id_ancora, char *nome_ancora) List_anc.cpp\ (71) int lista_ancoras :: grava(FILE *fp) No.cpp\ (94) int No::contem_ancora(ancora *pnc) No.cpp\ (107) ancora * No::retorna_ancora(int ind) No.cpp\ (120) int No::obtem_indice(int ide) No.cpp\ (257) ancora *No::proxima_ancora()
Procedimento: (25) ancora * proximo(ancora * pontancora) Diretório/Arquivo: List_anc.h	Descrição: Chama: (25) lista :: proximo(pontancora) Chamado por: No_audio.cpp\ (55) void no_audio::copia_ancoras_audio(lista_ancoras *ancoras) No_exec.cpp\ (55) void no_executavel::copia_ancoras_executavel(lista_ancoras *ancoras) No_image\ (56) void

	<pre> no_imagem::copia_ancoras_imagem(lista_ancoras *ancoras) No_tab.cpp\ (55) void no_tabela::copia_ancoras_tabela(lista_ancoras *ancoras) No_texto.cpp\ (60) void no_texto::copia_ancoras_texto(lista_ancoras *ancoras) No_video.cpp\ (54) void no_video::copia_ancoras_video(lista_ancoras *ancoras) </pre>
--	--

Sistema: SASHE**Arquivo:** Sashe\Hyperprop\ List_elo.cpp**Procedimento:**

(35) lista_elos :: ~lista_elos()

Diretório/Arquivo: List_elo.cpp

Descrição:

destrutor da lista

Chama:

(41) proxdado->elem

(43) proxdado->proximo

(46) reinicia_lista()

Chamado por:

Procedimento:

(51) int lista_elos::grava(FILE *fp)

Diretório/Arquivo: List_elo.cpp

Descrição:

Chama:

(53) fprintf(fp, "\tListaElo(%d)\n", num_elementos())

(54) elo *e = primeiro()

(54) e = proximo()

(55) e->grava_elo(fp)

Chamado por:

Composic.cpp\ (238) int

composicao::grava_no(FILE *fp)

Procedimento:

(61) int lista_elos::recupera(No *pno, FILE *fp)

Diretório/Arquivo: List_elo.cpp

Descrição:

Chama:

(65) fscanf(fp, "\tListaElo(%d)\n", &ne)

(71) fscanf(fp, "\t\tElo(%d)\n", &id)

(72) new elo(id)

(74) e->recupera_elo(fp)

(75) pno->eh_contexto_usuario()

(78) c->insere_elo(e)

(83) t->insere_elo(e)

Chamado por:

Composic.cpp\ (253) int

composicao::recupera_no(FILE *fp)

Sistema: SASHE**Arquivo:** Sashe\Hyperprop\ List_elo.h**Procedimento:**

(17) int lista_elos::insere(elo *pelo)

Diretório/Arquivo: List_elo.h

Descrição:

Chama:

(17) lista::insere(pelo)

Chamado por:

Composic.h\ (32) int insere_elo(elo *e)

Trilha.cpp\ (62) int

trilha::insere_no(No *pno, contexto *pno_liberdade)

Trilha.cpp\ (97) int

trilha::insere_no(No *pno)

Procedimento:

(18) int insere(elo *elo1, elo *elo2)

Diretório/Arquivo: List_elo.h

Descrição:

Chama:

(18) lista::insere(elo1, elo2)

	Chamado por: Trilha.cpp\ (131) int trilha::insere_no(No *pno, contexto *pno_liberdade, int posicao) Trilha.cpp\ (164) int trilha::insere_no(No *pno, int posicao)
Procedimento: (19) int retira(elo *pelo) Diretório/Arquivo: List_elo.h	Descrição: Chama: (19) lista::retira(pelo) Chamado por: Composic.h\ (33) int retira_elo(elo *e) Trilha.cpp\ (197) int trilha::retira_no(No *pno) Trilha.cpp\ (232) void trilha::limpa_trilha()
Procedimento: (20) elo * topo() Diretório/Arquivo: List_elo.h	Descrição: Chama: (20) lista::topo() Chamado por: Composic.cpp\ (193) elo * composicao::retorna_elo(int i)
Procedimento: (21) elo * primeiro() Diretório/Arquivo: List_elo.h	Descrição: Chama: (21) lista::primeiro() Chamado por: Composic.cpp\ (169) int composicao::primeiro_elo(int *id) Hipger.cpp\ (731) extern "C" int FAR PASCAL _export retira_da_hiperbase(char *nome_no) Hipger.cpp\ (1697) elo *FAR PASCAL _export primeiro_ponteiro_elo(char *nome_contexto) List_elo.cpp\ (51) int lista_elos::grava(FILE *fp) Trilha.cpp\ (131) int trilha::insere_no(No *pno, contexto *pno_liberdade, int posicao) Trilha.cpp\ (164) int trilha::insere_no(No *pno, int posicao) Trilha.cpp\ (197) int trilha::retira_no(No *pno) Trilha.cpp\ (232) void trilha::limpa_trilha() Trilha.cpp\ (251) int trilha::primeiro_no_trilha(char *nome, char *nome_liberdade) Trilha.cpp\ (385) int trilha::recupera_no(FILE *fp)
Procedimento: (22) elo * ultimo() Diretório/Arquivo: List_elo.h	Descrição: Chama: (22) lista::ultimo() Chamado por: Trilha.cpp\ (332) int trilha::ultimo_no_trilha(char *nome, char *nome_liberdade)
Procedimento: (23) elo * proximo() Diretório/Arquivo: List_elo.h	Descrição: Chama: (23) lista::proximo() Chamado por: Composic.cpp\ (181) int composicao::proximo_elo(int *id)

	<pre> Hipger.cpp\ (731) extern "C" int FAR PASCAL _export retira_da_hiperbase(char *nome_no) List_elo.cpp\ (51) int lista_elos::grava(FILE *fp) Trilha.cpp\ (131) int trilha::insere_no(No *pno, contexto *pno_liberdade, int posicao) Trilha.cpp\ (164) int trilha::insere_no(No *pno, int posicao) Trilha.cpp\ (197) int trilha::retira_no(No *pno) Trilha.cpp\ (278) int trilha::proximo_no_trilha(char *nome, char *nome_liberdade) Trilha.cpp\ (385) int trilha::recupera_no(FILE *fp) </pre>
Procedimento: (24) elo * anterior() Diretório/Arquivo: List_elo.h	Descrição: Chama: (24) lista::anterior() Chamado por: Trilha.cpp\ (305) int trilha::anterior_no_trilha(char *nome, char *nome_liberdade)
Procedimento: (25) elo * proximo(elo *pelo) Diretório/Arquivo: List_elo.h	Descrição: Chama: (25) lista::proximo(pelo) Chamado por: Composic.cpp\ (193) elo * composicao::retorna_elo(int i) Hipger.cpp\ (1714) elo *FAR PASCAL _export proximo_ponteiro_elo(char *nome contexto, elo *pelo)
Procedimento: (26) elo * anterior(elo *pelo) Diretório/Arquivo: List_elo.h	Descrição: Chama: (26) lista::anterior(pelo) Chamado por:

Sistema: SASHE**Arquivo:** Sashe\Hyperprop\ List_no.cpp

Procedimento: (67) void lista_nos :: inicializar(int lib_mem) Diretório/Arquivo: List_no.cpp	Descrição: Inicializa os atributos da classe lista de nós. Chama: Chamado por: Composic.cpp\ (23) composicao::composicao(char *nome_no) : No(nome_no) Composic.cpp\ (33) composicao::composicao(int id) : No(id) No.cpp\ (26) No::No(char *nome_no) : entidade() No.cpp\ (43) No::No(int id) : entidade()
Procedimento: (98) lista_nos :: ~lista_nos() Diretório/Arquivo: List_no.cpp	Descrição: Destruir da lista de nós. Chama: (109) dado->eh_texto() (115) dado->eh_video() (120) dado->eh_audio() (125) dado->eh_imagem() (130) dado->eh_contexto_usuario() (135) dado->eh_contexto_versoes() (140) dado->eh_trilha() (145) dado->eh_base_privada()

	(153) reinicia_lista() Chamado por:
Procedimento: (182) int lista_nos::grava(FILE *fp) Diretório/Arquivo: List_no.cpp	Descrição: Grava a estrutura da lista de nós em arquivo. 1: Erro de Escrita da lista de nós 0: Ok. Chama: (184) int ne = num_elementos() (185) No *no = primeiro() (187) fprintf(fp, "\tListaNos(%d)\n", ne) (190) no->eh_base_privada() (192) hb.recupera_base_ativa() (194) fprintf(fp, "\t\tIdentificador(%d)\n", pno->obtem_identificador()) (198) fprintf(fp, "\t\tIdentificador(%d)\n", no->obtem_identificador()) (200) proximo() Chamado por: Composic.cpp\ (238) int composicao::grava_no(FILE *fp) No.cpp\ (307) int No::grava_no(FILE *fp)
Procedimento: (231) int lista_nos::recupera(FILE *fp) Diretório/Arquivo: List_no.cpp	Descrição: Recupera a estrutura da lista de nós do arquivo. 1: Erro de Leitura 0: Ok. Chama: (234) fscanf(fp, "\tListaNos(%d)\n", &ne) (238) fscanf(fp, "\t\tIdentificador(%d)\n", &ident) (239) hb.RetornaNo(ident) (241) insere(pno) Chamado por: Composic.cpp\ (253) int composicao::recupera_no(FILE *fp) No.cpp\ (318) int No::recupera_no(FILE *fp)

Sistema: SASHE Arquivo: Sashe\Hyperprop\ List_no.h	
Procedimento: (21) lista_nos() Diretório/Arquivo: List_no.h	Descrição: Construtor da classe lista_nos Chama: Chamado por:
Procedimento: (24) int insere(No *pno) Diretório/Arquivo: List_no.h	Descrição: Chama: (24) lista::insere(pno) Chamado por: Basepriv.cpp\ (39) int baseprivada::insere_na_base_privada(No *pno) Composic.cpp\ (46) int composicao::insere_no(No *pno) Hip.h\ (26) int insere_hipwindow(No *pontno) Hipger.cpp\ (731) extern "C" int FAR PASCAL _export retira_da_hiperbase(char *nome_no) List_no.cpp\ (231) int lista_nos::recupera(FILE *fp) No.h\ (88) int insere_contexto(No

	*pontno) Trilha.cpp\ (62) int trilha::insere_no(No *pno, contexto *pno_liberdade) Trilha.cpp\ (97) int trilha::insere_no(No *pno) Trilha.cpp\ (131) int trilha::insere_no(No *pno, contexto *pno_liberdade, int posicao) Trilha.cpp\ (164) int trilha::insere_no(No *pno, int posicao)
Procedimento: (25) int insere(No *pant, No *pno) Diretório/Arquivo: List_no.h	Descrição: Chama: (25) lista::insere(pant, pno) Chamado por:
Procedimento: (26) int retira(No *pno) Diretório/Arquivo: List_no.h	Descrição: Chama: (26) lista::retira(pno) Chamado por: Basepriv.cpp\ (50) int baseprivada::retira_da_base_privada(No *pno) Composic.cpp\ (70) int composicao::retira_no(No *pno) Hip.h\ (33) int retira(No *pno) Hipger.cpp\ (731) extern "C" int FAR PASCAL _export retira_da_hiperbase(char *nome_no) No.h\ (90) int retira_contexto(No *pontno) Trilha.cpp\ (197) int trilha::retira_no(No *pno) Trilha.cpp\ (232) void trilha::limpa_trilha()
Procedimento: (27) No * primeiro() Diretório/Arquivo: List_no.h	Descrição: Chama: (27) lista::primeiro() Chamado por: Composic.cpp\ (89) int composicao::commit() Composic.cpp\ (107) int composicao::contem(No *no) Composic.cpp\ (123) int composicao::primeiro_no(char *nome) Composic.cpp\ (150) No *composicao::primeiro_no() Hip.cpp\ (26) No * hipwindow::Retorna(char *nome) Hip.cpp\ (49) No * hipwindow::RetornaNo(int Id) Hip.cpp\ (63) int hipwindow::RetornaIdent(char *nome) Hip.cpp\ (78) int hipwindow::ativa_hiperbase(char *nome) Hip.cpp\ (142) int hipwindow::primeiro_no_hiperbase(char *nome) Hip.h\ (31) No *primeiro() Hipger.cpp\ (516) No *FAR PASCAL _export primeiro_ponteiro_no(char *nome_cont) Hipger.cpp\ (731) extern "C" int FAR PASCAL _export retira_da_hiperbase(char *nome_no) Hipger.cpp\ (832) extern "C" int FAR

	PASCAL _export mostra_primeiro_lixo(char *nome) No.cpp\ (215) int No::pertence_contexto(No *pontno) No.cpp\ (229) No *No::primeiro_contexto_pai() Trilha.cpp\ (232) void trilha::limpa_trilha() Tutor.cpp\ (39) int tutor::primeira_estrategia(contexto *pai) Tutor.cpp\ (78) int tutor::segunda_estrategia(contexto *pai)
Procedimento: (28) No * ultimo() Diretório/Arquivo: List_no.h	Descrição: Chama: (28) lista::ultimo() Chamado por: Hip.cpp\ (165) int hipwindow::obtem_identificador_ultimo_no() No.h\ (95) No *ultimo contexto pai()
Procedimento: (29) No * proximo() Diretório/Arquivo: List_no.h	Descrição: Chama: (29) lista::proximo() Chamado por: Composic.cpp\ (89) int composicao::commit() Composic.cpp\ (107) int composicao::contem(No *no) Composic.cpp\ (136) int composicao::proximo_no(char *nome) Composic.cpp\ (159) No *composicao::proximo_no() Hip.cpp\ (26) No * hipwindow::Retorna(char *nome) Hip.cpp\ (49) No * hipwindow::RetornaNo(int Id) Hip.cpp\ (63) int hipwindow::RetornaIdent(char *nome) Hip.cpp\ (78) int hipwindow::ativa_hiperbase(char *nome) Hip.cpp\ (154) int hipwindow::proximo_no_hiperbase(char *nome) Hip.h\ (32) No *proximo() Hipger.cpp\ (533) No *FAR PASCAL _export proximo_ponteiro_no(char *nome_cont) Hipger.cpp\ (731) extern "C" int FAR PASCAL _export retira_da_hiperbase(char *nome_no) Hipger.cpp\ (846) extern "C" int FAR PASCAL _export mostra_proximo_lixo(char *nome) List_no.cpp\ (182) int lista_nos::grava(FILE *fp) No.cpp\ (215) int No::pertence_contexto(No *pontno) No.cpp\ (236) No *No::proximo_contexto_pai() Tutor.cpp\ (39) int tutor::primeira_estrategia(contexto *pai) Tutor.cpp\ (78) int tutor::segunda_estrategia(contexto *pai)
Procedimento:	Descrição:

(30) No *anterior() Diretório/Arquivo: List_no.h	Chama: (30) lista::anterior() Chamado por:
Procedimento: (31) No * proximo(No *pno) Diretório/Arquivo: List_no.h	Descrição: Chama: (31) lista::proximo(pno) Chamado por:
Procedimento: (32) No *anterior(No *pno) Diretório/Arquivo: List_no.h	Descrição: Chama: (32) lista::anterior(pno) Chamado por:
Procedimento: (35) void insere_topo(No *pno) Diretório/Arquivo: List_no.h	Descrição: Chama: (35) lista::insere_topo(pno) Chamado por:
Procedimento: (36) No * retira_topo() Diretório/Arquivo: List_no.h	Descrição: Chama: (36) lista::retira_topo() Chamado por:
Procedimento: (37) No * topo() Diretório/Arquivo: List_no.h	Descrição: Chama: (37) lista::topo() Chamado por:

Sistema: SASHE Arquivo: Sashe\Hyperprop\ lista.cpp	
Procedimento: (12) lista :: lista() Diretório/Arquivo: lista.cpp	Descrição: construtor da lista Chama: Chamado por:
Procedimento: (19) void lista :: reinicia_lista() Diretório/Arquivo: Lista.cpp	Descrição: reinicializa uma lista Chama: Chamado por: List_no.cpp\ (98) lista_nos :: ~lista_nos() List_elo.cpp\ (35) lista_elos :: ~lista_elos()
Procedimento: (30) qqtipo lista::primeiro() Diretório/Arquivo: Lista.cpp	Descrição: primeiro da lista Chama: Chamado por: Composic.cpp\ (207) void composicao::copia_nos(lista_nos *lista) List_anc.h\ (22) ancora * primeiro() List_elo.h\ (21) elo * primeiro() List_no.h\ (27) No * primeiro()
Procedimento: (43) qqtipo lista::ultimo() Diretório/Arquivo: Lista.cpp	Descrição: ultimo da lista Chama: Chamado por: List_anc.h\ (23) ancora * ultimo() List_elo.ch\ (22) elo * ultimo() List_no.h\ (28) No * ultimo()
Procedimento: (56) qqtipo lista::proximo() Diretório/Arquivo: Lista.cpp	Descrição: proximo da lista Chama: Chamado por: List_anc.h\ (24) ancora * proximo() List_elo.h\ (23) elo * proximo()

Procedimento: (72) qqtipo lista::anterior() Diretório/Arquivo: Lista.cpp	List_no.h\ (29) No * proximo() Descrição: anterior da lista Chama: Chamado por: List_elo.h\ (24) elo * anterior() List_no.h\ (30) No *anterior()
Procedimento: (88) qqtipo lista::proximo(qqtipo alguem) Diretório/Arquivo: Lista.cpp	Descrição: proximo na lista em relacao a alguem Chama: Chamado por: Composic.cpp\ (207) void composicao::copia_nos(lista_nos *lista) List_anc.h\ (25) ancora * proximo(ancora * pontancora) List_elo.h\ (25) elo * proximo(elo *pelo) List_no.h\ (31) No * proximo(No *pno)
Procedimento: (105) qqtipo lista::anterior(qqtipo alguem) Diretório/Arquivo: Lista.cpp	Descrição: anterior na lista em relacao a alguem Chama: Chamado por: List_elo.h\ (26) elo * anterior(elo *pelo) List_no.h\ (32) No *anterior(No *pno)
Procedimento: (124) int lista::insere(qqtipo a) Diretório/Arquivo: Lista.cpp	Descrição: insere um elemento no final da lista retorna 0 se conseguiu inserir o elemento na lista retorna 1 se falta memoria para a lista Chama: (128) new elemento(a, ult, NULL) (134) new elemento(a, NULL, NULL) Chamado por: List_anc.h\ (20) int insere(ancora *pontancora) List_elo.h\ (17) int insere(elo *pelo) List_no.h\ (24) int insere(No *pno)
Procedimento: (147) int lista::insere(qqtipo anterior, qqtipo a) Diretório/Arquivo: Lista.cpp	Descrição: insere um elemento apos o elemento anterior retorna 0 se conseguiu inserir o elemento na lista retorna 1 se falta memoria para a lista retorna 2 se elemento anterior nao foi encontrado Chama: (154) new elemento(a, dado, dado- >proximo) Chamado por: List_elo.h\ (18) int insere(elo *elo1, elo *elo2) List_no.h\ (25) int insere(No *pant, No *pno)
Procedimento: (171) int lista::retira(qqtipo a) Diretório/Arquivo: Lista.cpp	Descrição: retira um elemento da lista. retorna 0 se conseguiu retirar o elemento da lista retorna 1 se a lista estiver vazia. retorna 2 se nao encontrou o elemento na lista Chama: Chamado por: List_anc.h\ (21) int retira(ancora *pontancora)

	List_elo.h\ (19) int retira(elo *pelo) List_no.h\ (26) int retira(No *pno)
Procedimento: (207) void lista::insere_topo(qqtipo a) Diretório/Arquivo: Lista.cpp	Descrição: insere um elemento no inicio da lista Chama: (211) elemento(a, NULL, prim) (216) elemento(a, NULL, NULL) Chamado por: List_no.h\ (35) void insere_topo(No *pno)
Procedimento: (225) qqtipo lista::retira_topo() Diretório/Arquivo: Lista.cpp	Descrição: retira um elemento no topo da lista Chama: Chamado por: List_no.h\ (36) No * retira_topo()

Sistema: SASHE**Arquivo: Sashe\Hyperprop\ Lista.h**

Procedimento: (43) int lista::num_elementos() Diretório/Arquivo: Lista.h	Descrição: Chama: Chamado por: Hipger.cpp\ (731) extern "C" int FAR PASCAL _export retira_da_hiperbase(char *nome_no) Hipger.cpp\ (1297) extern "C" int FAR PASCAL _export numero_de_ancoras(char *nome) Hipger.cpp\ (1627) extern "C" int FAR PASCAL _export numero_de_elos(char *nome_contexto) List_anc.cpp\ (71) int lista_ancoras :: grava(FILE *fp) List_elo.cpp\ (51) int lista_elos::grava(FILE *fp) List_no.cpp\ (182) int lista_nos::grava(FILE *fp) No.h\ (80) int numero_ancoras() No.h\ (93) int num_contextos() Trilha.cpp\ (62) int trilha::insere_no(No *pno, contexto *pno_liberdade) Trilha.cpp\ (97) int trilha::insere_no(No *pno) Trilha.cpp\ (131) int trilha::insere_no(No *pno, contexto *pno_liberdade, int posicao) Trilha.cpp\ (164) int trilha::insere_no(No *pno, int posicao) Trilha.cpp\ (197) int trilha::retira_no(No *pno)
Procedimento: (47) qqtipo lista::topo() Diretório/Arquivo: Lista.h	Descrição: Chama: Chamado por: List_elo.h\ (20) elo * topo() List_no.h\ (37) No * topo()

Sistema: SASHE**Arquivo: Sashe\Hyperprop\ No.cpp**

Procedimento: (26) No::No(char *nome_no) : entidade() Diretório/Arquivo: No.cpp	Descrição: Este construtor é chamado pela interface quando se cria um nó. Chama: (26) entidade() (29) contextos.inicializar(0)
--	--

	(31) troca_identificador(hb.obtem_identificad or_ultimo_no()+1) (34) obtem_identificador() (37) hb.insere_hipwindow(this) Chamado por: Composic.cpp\ (23) composicao::composicao(char *nome_no) : No(nome_no) Terminal.cpp\ (20) no_terminal::no_terminal(char *nome_no) : No(nome_no)
Procedimento: (43) No::No(int id) : entidade() Diretório/Arquivo: No.cpp	Descrição: Este construtor é utilizado quando o nó é lido do arquivo. Chama: (43) entidade() (46) contextos.inicializar(0) (47) troca_identificador(id) (52) hb.insere_hipwindow(this) Chamado por: Composic.cpp\ (33) composicao::composicao(int id) : No(id) Terminal.cpp\ (33) no_terminal::no_terminal(int id) : No(id)
Procedimento: (57) int No::MudaNome(char *novo_nome) Diretório/Arquivo: No.cpp	Descrição: Chama: (60) ObtemEstado() Chamado por: Hipger.cpp\ (866) extern "C" int FAR PASCAL _export muda_atributos(char *nome_velho, char *nome_novo, int num_estado)
Procedimento: (67) int No::commit() Diretório/Arquivo: No.cpp	Descrição: Chama: (70) ObtemEstado() Chamado por: Composic.cpp\ (89) int composicao::commit() Hipger.cpp\ (866) extern "C" int FAR PASCAL _export muda_atributos(char *nome_velho, char *nome_novo, int num_estado)
Procedimento: (80) int No::obsolete() Diretório/Arquivo: No.cpp	Descrição: Chama: (83) ObtemEstado() Chamado por: Hipger.cpp\ (866) extern "C" int FAR PASCAL _export muda_atributos(char *nome_velho, char *nome_novo, int num_estado)
Procedimento: (94) int No::contem_ancora(ancora *panc) Diretório/Arquivo: No.cpp	Descrição: Chama: (96) ancoras.primeiro() (100) ancoras.proximo() Chamado por:
Procedimento: (107) ancora * No::retorna_ancora(int ind) Diretório/Arquivo: No.cpp	Descrição: Chama:

	<pre> (109) ancoras.primeiro() (112) p->obtem_identificador() (113) ancoras.proximo() Chamado por: Elo.h\ (56) ancora *obtem_ancora_origem() (59) ancora *obtem_ancora_destino() Hipger.cpp\ (1227) extern "C" int FAR PASCAL _export retira_ancora(char *nome, int ident) Hipger.cpp\ (1260) extern "C" int FAR PASCAL _export muda_ancora_texto(char *nome, int velho_ide, int novo_ide, char *palavra, int inicio, int fim) Hipger.cpp\ (1279) extern "C" int FAR PASCAL _export muda_ancora_contexto(char *nome, int velho_ide, int novo_ide, char *no_ancorado) Trilha.cpp\ (251) int trilha::primeiro_no_trilha(char *nome, char *nome_liberdade) Trilha.cpp\ (278) int trilha::proximo_no_trilha(char *nome, char *nome_liberdade) Trilha.cpp\ (305) int trilha::anterior_no_trilha(char *nome, char *nome_liberdade) Trilha.cpp\ (332) int trilha::ultimo_no_trilha(char *nome, char *nome_liberdade) </pre>
Procedimento: (120) int No::obtem_indice(int ide) Diretório/Arquivo: No.cpp	Descrição: Chama: <pre> (122) ancoras.primeiro() (125) p->obtem_ident_externo() (125) p->obtem_identificador() (126) ancoras.proximo() </pre> Chamado por: <pre> Hipger.cpp\ (1246) extern "C" int FAR PASCAL _export recupera_identificador_interno(char *nome, int ide) Hipger.cpp\ (1260) extern "C" int FAR PASCAL _export muda_ancora_texto(char *nome, int velho_ide, int novo_ide, char *palavra, int inicio, int fim) Hipger.cpp\ (1279) extern "C" int FAR PASCAL _export muda_ancora_contexto(char *nome, int velho_ide, int novo_ide, char *no_ancorado) </pre>
Procedimento: (133) int No::retorna_indice() Diretório/Arquivo: No.cpp	Descrição: Chama: <pre> (135) ancoras.ultimo() (137) p->obtem_identificador() </pre> Chamado por: <pre> Contexto.cpp\ (43) int contexto::cria_ancora_lambda(int ide) Contexto.cpp\ (61) ancora_contexto *contexto::cria_ancora_contexto(int ide, int id_no) No_audio.cpp\ (38) int no_audio::cria_ancora_audio(int ide) No_exec.cpp\ (38) int no_executavel::cria_ancora_executavel(in t ide) No_image.cpp\ (39) int </pre>

	<pre> no_imagem::cria_ancora_imagem(int ide) No_tab.cpp\ (38) int no_tabela::cria_ancora_tabela(int ide) No_texto.cpp\ (42) int no_texto::cria_ancora_texto(int ide, char *pal, int inicio, int fim) No_video.cpp\ (37) int no_video::cria_ancora_video(int ide) Trilha.cpp\ (38) int trilha::cria_ancora_lambda(int ide) </pre>
Procedimento: (142) int No::criaversao(char *nome_no) Diretório/Arquivo: No.cpp	Descrição: Chama: <pre> (148) ObtemEstado() (152) eh_contexto_usuario() (154) new contexto(nome_no) (155) ncv->copia_contexto((contexto *)this (158) eh_trilha() (160) new trilha(nome_no) (161) ntr->copia_trilha((trilha *)this) (164) eh_texto() (166) new no_texto(nome_no) (167) ntv->copia_texto((no_texto *)this) (170) eh_audio() (172) new no_audio(nome_no) (173) naudv->copia_audio((no_audio *)this) (176) eh_imagem() (178) new no_imagem(nome_no) (179) nimav->copia_imagem((no_imagem *)this) (182) eh_video() (184) new no_video(nome_no) (185) nvidv->copia_video((no_video *)this) (189) ObtemCV() (194) contextoversoes(this,nv,numchar) (196) NCV->obtem_identificador() (199) MudaCV(ICV) (202) ObtemCV() (203) NCV = hb.RetornaNo(ICV) (205) pontncv->insere_no(nv) (207) nv->MudaCV(ICV) </pre> Chamado por: Hipger.cpp\ (1121) extern "C" int FAR PASCAL _export cria_versao(char *nome_no, char *nome_no versao)
Procedimento: (215) int No::pertence_contexto(No *pontno) Diretório/Arquivo: No.cpp	Descrição: Chama: <pre> (217) contextos.primeiro() (221) contextos.proximo() </pre> Chamado por: Hip.h\ (29) int pertence_base_ativa(No *pontno)
Procedimento: (229) No *No::primeiro_contexto_pai() Diretório/Arquivo: No.cpp	Descrição: Chama: <pre> (231) contextos.primeiro() </pre> Chamado por: Basepriv.cpp\ (57) No * baseprivada::RetornaNo(int Id) Basepriv.cpp\ (71) No *

	<pre> baseprivada::Retorna(char *nome) Basepriv.cpp\ (85) int baseprivada::RetornaIdent(char *nome) Basepriv.cpp\ (103) int baseprivada::grava(FILE *fp) Basepriv.cpp\ (133) int baseprivada::recupera(FILE *fp) Hip.cpp\ (177) void hipwindow::atualiza_base_ativa(No *pno) Hipger.cpp\ (400) extern "C" int FAR PASCAL _export primeiro_contexto_pai(char *nome, char *nome_pai) </pre>
Procedimento: (236) No *No::proximo_contexto_pai() Diretório/Arquivo: No.cpp	Descrição: Chama: (238) return contextos.proximo() Chamado por: Basepriv.cpp\ (57) No * baseprivada::RetornaNo(int Id) Basepriv.cpp\ (71) No * baseprivada::Retorna(char *nome) Basepriv.cpp\ (85) int baseprivada::RetornaIdent(char *nome) Basepriv.cpp\ (103) int baseprivada::grava(FILE *fp) Basepriv.cpp\ (133) int baseprivada::recupera(FILE *fp) Hip.cpp\ (177) void hipwindow::atualiza_base_ativa(No *pno) Hipger.cpp\ (426) extern "C" int FAR PASCAL _export proximo_contexto_pai(char *nome, char *nome_pai)
Procedimento: (243) ancora *No::primeira_ancora() Diretório/Arquivo: No.cpp	Descrição: Chama: (245) ancoras.primeiro() Chamado por: Contexto.cpp\ (77) ancora_contexto *contexto::contem_ancora(int id no) Hipger.cpp\ (1372) extern "C" int FAR PASCAL _export primeira_ancora(char *nome, char *nome_ancora)
Procedimento: (250) ancora *No::ultima_ancora() Diretório/Arquivo: No.cpp	Descrição: Chama: (252) ancoras.ultimo() Chamado por: Hipger.cpp\ (1489) extern "C" int FAR PASCAL _export ultima_ancora(char *nome, char *nome_ancora)
Procedimento: (257) ancora *No::proxima_ancora() Diretório/Arquivo: No.cpp	Descrição: Chama: (259) ancoras.proximo() Chamado por: Contexto.cpp\ (77) ancora_contexto *contexto::contem_ancora(int id no) Hipger.cpp\ (1425) extern "C" int FAR PASCAL _export proxima_ancora(char *nome, char *nome_ancora)
Procedimento: (264) int No :: grava(FILE *fp) Diretório/Arquivo: No.cpp	Descrição: Chama: (266) fprintf(fp, "\tNome=%s;\n", nome) (267) fprintf(fp, "\tEstado=%c;\n",

	Estado) (268) fprintf(fp, "\tVersão=%d;\n", ContextoVersoes) (269) fprintf(fp, "}Fim_No(%d)\n", obtem_identificador()) Chamado por: Basepriv.cpp\ (103) int baseprivada::grava(FILE *fp) Composic.cpp\ (219) int composicao::grava(FILE *fp) No_audio.cpp\ (68) int no_audio::grava(FILE *fp) No_exec.cpp\ (68) int no_executavel::grava(FILE *fp) No_image.cpp\ (69) int no_imagem::grava(FILE *fp) No_tab.cpp\ (68) int no_tabela::grava(FILE *fp) No_texto.cpp\ (74) int no_texto::grava(FILE *fp) No_video.cpp\ (67) int no_video::grava(FILE *fp)
Procedimento: (275) int No :: recupera(FILE *fp) Diretório/Arquivo: No.cpp	Descrição: Chama: (279) fscanf(fp, "\tNome=%s;\n", nome) (280) fscanf(fp, "\tEstado=%c;\n", &Estado) (281) fscanf(fp, "\tVersão=%d;\n", &ContextoVersoes) (282) fscanf(fp, "}Fim_No(%d)\n", &ident) (283) obtem_identificador() Chamado por: Basepriv.cpp\ (133) int baseprivada::recupera(FILE *fp) Composic.cpp\ (227) int composicao::recupera(FILE *fp) No_audio.cpp\ (77) int no_audio::recupera(FILE *fp) No_exec.cpp\ (77) int no_executavel::recupera(FILE *fp) No_image.cpp\ (78) int no_imagem::recupera(FILE *fp) No_tab.cpp\ (77) int no_tabela::recupera(FILE *fp) No_texto.cpp\ (83) int no_texto::recupera(FILE *fp) No_video.cpp\ (76) int no_video::recupera(FILE *fp)
Procedimento: (289) int No::UltCV(int ultcv) Diretório/Arquivo: No.cpp	Descrição: Registra ou retorna o numero do ultimo contexto de versoes Chama: Chamado por: No.cpp\ (142) int No::criaversao(char *nome no)
Procedimento: (307) int No::grava_no(FILE *fp) Diretório/Arquivo: No.cpp	Descrição: Chama: (309) fprintf(fp, "Nome(%s)\n", nome) (310) contextos.grava(fp) (311) ancoras.grava(fp) (312) fprintf(fp, "\tDificuldade(%d)\n", dificuldade) Chamado por: Basepriv.cpp\ (103) int baseprivada::grava(FILE *fp)

	Composic.cpp\ (238) int composicao::grava_no(FILE *fp) Terminal.cpp\ (106) int no terminal::grava no(FILE *fp)
Procedimento: (318) int No::recupera_no(FILE *fp) Diretório/Arquivo: No.cpp	Descrição: Chama: (320) fscanf(fp, "Nome(%s)\n", nome) (321) contextos.recupera(fp) (322) ancoras.recupera(this, fp) (323) fscanf(fp, "\tDificuldade(%d)\n", &dificuldade) Chamado por: Basepriv.cpp\ (133) int baseprivada::recupera(FILE *fp) Composic.cpp\ (253) int composicao::recupera_no(FILE *fp) Terminal.cpp\ (123) int no terminal::recupera no(FILE *fp)

Sistema: SASHE Arquivo: Sashe\Hyperprop\ No.h	
Procedimento: (67) char ObtemEstado() Diretório/Arquivo: No.h	Descrição: Retorna o Estado de um no Chama: Chamado por: Composic.cpp\ (46) int composicao::insere_no(No *pno) Composic.cpp\ (89) int composicao::commit() Hipger.cpp\ (263) extern "C" int FAR PASCAL _export insere_no_contexto(char *nome_cont, char *nome_no) Hipger.cpp\ (339) extern "C" int FAR PASCAL _export retira_do_contexto(char *nome_contexto, char *nome_no) Hipger.cpp\ (359) extern "C" int FAR PASCAL _export retira_da_trilha(char *nome_trilha, char *nome_no_liberdade) Hipger.cpp\ (376) extern "C" int FAR PASCAL _export limpa_trilha(char *nome_trilha) Hipger.cpp\ (731) extern "C" int FAR PASCAL _export retira_da_hiperbase(char *nome_no) Hipger.cpp\ (1121) extern "C" int FAR PASCAL _export cria_versao(char *nome_no, char *nome_no_versao) Hipger.cpp\ (1181) extern "C" int FAR PASCAL _export cria_ancora_texto(char *nome_no, int ident, char *palavra, int inicio, int fim) Hipger.cpp\ (1203) extern "C" int FAR PASCAL _export cria_ancora_contexto(char *nome_no, int ident, char *no_ancorado) Hipger.cpp\ (1227) extern "C" int FAR PASCAL _export retira_ancora(char *nome, int ident) Hipger.cpp\ (1608) extern "C" int FAR PASCAL _export retira_elo(char *nome_contexto, int id_elo) No.cpp\ (57) int No::MudaNome(char *novo_nome) No.cpp\ (67) int No::commit() No.cpp\ (80) int No::obsoleto() No.cpp\ (142) int No::criaversao(char *nome_no)

	Trilha.cpp\ (62) int trilha::insere_no(No *pno, contexto *pno_liberdade) Trilha.cpp\ (97) int trilha::insere_no(No *pno) Trilha.cpp\ (131) int trilha::insere_no(No *pno, contexto *pno_liberdade, int posicao) Trilha.cpp\ (164) int trilha::insere_no(No *pno, int posicao) Trilha.cpp\ (197) int trilha::retira_no(No *pno)
Procedimento: (680 char *ObtemNome() Diretório/Arquivo: No.h	Descrição: Retorna o nome para o no Chama: Chamado por: Elo.h\ (61) char *obtem_origem() Elo.h\ (63) char *obtem_destino() Basepriv.cpp\ (71) No * baseprivada::Retorna(char *nome) Basepriv.cpp\ (85) int baseprivada::RetornaIdent(char *nome) Composic.cpp\ (123) int composicao::primeiro_no(char *nome) Composic.cpp\ (136) int composicao::proximo_no(char *nome) Grafov.cpp\ (220) void CGrafoView::SetDataObject() Grafov.cpp\ (542) void CGrafoView::DesenhaNo(No *no) Grafov.cpp\ (1140) void CGrafoView::OnLButtonDown(UINT nFlags, CPoint point) Grafov.cpp\ (1220) void CGrafoView::OnRButtonDown(UINT nFlags, CPoint point) Grafov.cpp\ (1494) UINT CGrafoView::GetMouseOverTip(CPoint point) Hip.cpp\ (26) No * hipwindow::Retorna(char *nome) Hip.cpp\ (63) int hipwindow::RetornaIdent(char *nome) Hip.cpp\ (78) int hipwindow::ativa_hiperbase(char *nome) Hip.cpp\ (97) int hipwindow::grava_hiperbase(char *nome) Hip.cpp\ (142) int hipwindow::primeiro_no_hiperbase(char *nome) Hip.cpp\ (154) int hipwindow::proximo_no_hiperbase(char *nome) Hipger.cpp\ (144) extern "C" int FAR PASCAL _export retorna_nome(int ident, char *nome) Hipger.cpp\ (400) extern "C" int FAR PASCAL _export primeiro_contexto_pai(char *nome, char *nome_pai) Hipger.cpp\ (413) extern "C" int FAR PASCAL _export ultimo_contexto_pai(char *nome, char *nome_pai) Hipger.cpp\ (426) extern "C" int FAR PASCAL _export proximo_contexto_pai(char *nome, char *nome_pai) Hipger.cpp\ (731) extern "C" int FAR

	<pre> PASCAL _export retira_da_hiperbase(char *nome_no) Hipger.cpp\ (832) extern "C" int FAR PASCAL _export mostra_primeiro_lixo(char *nome) Hipger.cpp\ (846) extern "C" int FAR PASCAL _export mostra_proximo_lixo(char *nome) Hipger.cpp\ (1067) extern "C" int FAR PASCAL _export regras_tutoriais(char *no_atual, char *escopo, char *historico, int dificuldade, char *no_result, char *lib_result, int *revisto) Hipger.cpp\ (1311) extern "C" int FAR PASCAL _export recupera_nome_ancora(char *nome, int id_ancora, char *nome_ancora) Hipger.cpp\ (1372) extern "C" int FAR PASCAL _export primeira_ancora(char *nome, char *nome_ancora) Hipger.cpp\ (1425) extern "C" int FAR PASCAL _export proxima_ancora(char *nome, char *nome_ancora) Trilha.cpp\ (251) int trilha::primeiro_no_trilha(char *nome, char *nome_liberdade) Trilha.cpp\ (278) int trilha::proximo_no_trilha(char *nome, char *nome_liberdade) Trilha.cpp\ (305) int trilha::anterior_no_trilha(char *nome, char *nome_liberdade) Trilha.cpp\ (332) int trilha::ultimo_no_trilha(char *nome, char *nome_liberdade) </pre>
Procedimento: (70) int ObtemCV() Diretório/Arquivo: No.h	Descrição: Chama: Chamado por: <pre> Hipger.cpp\ (731) extern "C" int FAR PASCAL _export retira_da_hiperbase(char *nome_no) Hipger.cpp\ (1181) extern "C" int FAR PASCAL _export cria_ancora_texto(char *nome_no, int ident, char *palavra, int inicio, int fim) Hipger.cpp\ (1203) extern "C" int FAR PASCAL _export cria_ancora_contexto(char *nome_no, int ident, char *no_ancorado) Hipger.cpp\ (1227) extern "C" int FAR PASCAL _export retira_ancora(char *nome, int ident) No.cpp\ (142) int No::criaversao(char *nome_no) </pre>
Procedimento: (71) void muda_dificuldade(int nivel) Diretório/Arquivo: No.h	Descrição: Chama: Chamado por: <pre> Hipger.cpp\ (993) extern "C" int FAR PASCAL _export muda_dificuldade(char *nome_no, int nivel) </pre>
Procedimento: (72) int recupera_dificuldade() Diretório/Arquivo: No.h	Descrição: Chama: Chamado por: <pre> Hipger.cpp\ (1009) extern "C" int FAR </pre>

	PASCAL_export recupera_dificuldade(char *nome_no, int *nivel) Tutor.cpp\ (39) int tutor::primeira_estrategia(contexto *pai)
Procedimento: (77) int insere_ancora(ancora *p) Diretório/Arquivo: No.h	Descrição: Insere a ancora na lista de ancoras de um no' Chama: (77) ancoras.insere(p) Chamado por: Contexto.cpp\ (43) int contexto::cria_ancora_lambda(int ide) Contexto.cpp\ (61) ancora_contexto *contexto::cria_ancora_contexto(int ide, int id_no) Contvers.cpp\ (54) void contextoversoes::CriaAncoraCV(No *n) List_anc.cpp\ (118) int lista_ancoras :: recupera(No *n, FILE *fp) No_audio.cpp\ (38) int no_audio::cria_ancora_audio(int ide) No_exec.cpp\ (38) int no_executavel::cria_ancora_executavel(int ide) No_image.cpp\ (39) int no_imagem::cria_ancora_imagem(int ide) No_tab.cpp\ (38) int no_tabela::cria_ancora_tabela(int ide) No_texto.cpp\ (42) int no_texto::cria_ancora_texto(int ide, char *pal, int inicio, int fim) No_video.cpp\ (37) int no_video::cria_ancora_video(int ide) Trilha.cpp\ (38) int trilha::cria_ancora_lambda(int ide)
Procedimento: (78) int retira_ancora(ancora *p) Diretório/Arquivo: No.h	Descrição: Chama: (78) ancoras.retira(p) Chamado por: Hipger.cpp\ (1227) extern "C" int FAR PASCAL_export retira_ancora(char *nome, int ident) Hipger.cpp\ (1425) extern "C" int FAR PASCAL_export proxima_ancora(char *nome, char *nome_ancora)
Procedimento: (79) int contem_ancora(ancora *) Diretório/Arquivo: No.h	Descrição: Protótipo da função implementada em No.cpp Chama: Chamado por:
Procedimento: (80) int numero_ancoras() Diretório/Arquivo: No.h	Descrição: Chama: (80) ancoras.num_elementos() Chamado por:
Procedimento: (81) ancora *retorna_ancora(int) Diretório/Arquivo: No.h	Descrição: Protótipo da função implementada em No.cpp Chama: Chamado por:
Procedimento: (82) int obtem_indice(int) Diretório/Arquivo: No.h	Descrição: Protótipo da função implementada em No.cpp Chama: Chamado por:

Procedimento: (83) int retorna_indice() Diretório/Arquivo: No.h	Descrição: Protótipo da função implementada em No.cpp Chama: Chamado por:
Procedimento: (84) int retira_ancora(int) Diretório/Arquivo: No.h	Descrição: Retira uma ancora da lista de ancoras do no' Chama: Chamado por:
Procedimento: (85) int criaversao(char *nome_versao) Diretório/Arquivo: No.h	Descrição: Protótipo da função implementada em No.cpp Chama: Chamado por:
Procedimento: (86) int UltCV(int) Diretório/Arquivo: No.h	Descrição: Protótipo da função implementada em No.cpp Chama: Chamado por:
Procedimento: (88) int insere_contexto(No *pontno) Diretório/Arquivo: No.h	Descrição: Chama: (88) contextos.insere(pontno) Chamado por: Composic.cpp\ (46) int composicao::insere_no(No *pno) Hip.h\ (27) int insere_base_ativa(No *pontno) Trilha.cpp\ (62) int trilha::insere_no(No *pno, contexto *pno_liberdade) Trilha.cpp\ (97) int trilha::insere_no(No *pno) Trilha.cpp\ (131) int trilha::insere_no(No *pno, contexto *pno_liberdade, int posicao) Trilha.cpp\ (164) int trilha::insere_no(No *pno, int posicao)
Procedimento: (90) int retira_contexto(No *pontno) Diretório/Arquivo: No.h	Descrição: Retira o contexto da lista de no's contexto Chama: (90) contextos.retira(pontno) Chamado por: Composic.cpp\ (70) int composicao::retira_no(No *pno) Hip.h\ (28) int retira_base_ativa(No *pontno) Hipger.cpp\ (731) extern "C" int FAR PASCAL _export retira_da_hiperbase(char *nome_no) Trilha.cpp\ (197) int trilha::retira_no(No *pno) Trilha.cpp\ (232) void trilha::limpa_trilha()
Procedimento: (93) int num_contextos() Diretório/Arquivo: No.h	Descrição: Chama: (93) contextos.num_elementos() Chamado por: Basepriv.cpp\ (103) int baseprivada::grava(FILE *fp) Composic.cpp\ (70) int composicao::retira_no(No *pno)

	Hipger.cpp\ (390) extern "C" int FAR PASCAL _export quantos contextos pertence(char *nome)
Procedimento: (95) No *ultimo_contexto_pai() Diretório/Arquivo: No.h	Descrição: Chama: (95) contextos.ultimo() Chamado por: Hipger.cpp\ (413) extern "C" int FAR PASCAL _export ultimo_contexto_pai(char *nome, char *nome_pai)
Procedimento: (96) No *proximo_contexto_pai() Diretório/Arquivo: No.h	Descrição: Protótipo da função implementada em No.cpp Chama: Chamado por:
Procedimento: (97) ancora *primeira_ancora() Diretório/Arquivo: No.h	Descrição: Protótipo da função implementada em No.cpp Chama: Chamado por:
Procedimento: (98) ancora *ultima_ancora() Diretório/Arquivo: No.h	Descrição: Protótipo da função implementada em No.cpp Chama: Chamado por:
Procedimento: (99) ancora *proxima_ancora() Diretório/Arquivo: No.h	Descrição: Protótipo da função implementada em No.cpp Chama: Chamado por:
Procedimento: (101) virtual int eh_terminal() Diretório/Arquivo: No.h	Descrição: Chama: Chamado por: Grafov.cpp\ (173) void CGrafoView::OnDraw(CDC* pDC) Grafov.cpp\ (1220) void CgravoView::OnRButtonDown(UINT nFlags, CPoint point) Hipger.cpp\ (890) extern "C" int FAR PASCAL _export muda_tipo(char *nome_no, char *tipo) Hipger.cpp\ (908) extern "C" int FAR PASCAL _export recupera_tipo(char *nome_no, char *tipo) Hipger.cpp\ (927) extern "C" int FAR PASCAL _export muda_funcao_didatica(char *nome_no, char *categoria) Hipger.cpp\ (945) extern "C" int FAR PASCAL _export recupera_funcao_didatica(char *nome_no, char *categoria) Hipger.cpp\ (1028) extern "C" int FAR PASCAL _export muda_palavra_chave(char *nome_no, int pos, char *palavra) Hipger.cpp\ (1046) extern "C" int FAR PASCAL _export recupera_palavra_chave(char *nome_no, int pos, char *palavra) Tutor.cpp\ (78) int tutor::segunda_estrategia(contexto *pai)
Procedimento: (102) virtual int eh_texto() Diretório/Arquivo: No.h	Descrição: Chama: Chamado por: Grafov.cpp\ (542) void

	<pre> CGrafoView::DesenhaNo(No *no) Grafov.w.cpp\ (1220) void CGrafoView::OnRButtonDown(UINT nFlags, CPoint point) Hipger.cpp\ (581) extern "C" int FAR PASCAL _export recupera_nome_arquivo_texto(char *nome_texto, char *arquivo) Hipger.cpp\ (593) extern "C" int FAR PASCAL _export insere_nome_arquivo_texto(char *nome_texto, char *arquivo) Hipger.cpp\ (712) extern "C" int FAR PASCAL _export tipo_no(char *nome) Hipger.cpp\ (1181) extern "C" int FAR PASCAL _export cria_ancora_texto(char *nome_no, int ident, char *palavra, int inicio, int fim) Hipger.cpp\ (1260) extern "C" int FAR PASCAL _export muda_ancora_texto(char *nome, int velho_ide, int novo_ide, char *palavra, int inicio, int fim) Hipger.cpp\ (1311) extern "C" int FAR PASCAL _export recupera_nome_ancora(char *nome, int id_ancora, char *nome_ancora) Hipger.cpp\ (1372) extern "C" int FAR PASCAL _export primeira_ancora(char *nome, char *nome_ancora) Hipger.cpp\ (1425) extern "C" int FAR PASCAL _export proxima_ancora(char *nome, char *nome_ancora) Hipger.cpp\ (1489) extern "C" int FAR PASCAL _export ultima_ancora(char *nome, char *nome_ancora) List_anc.cpp\ (118) int lista_ancoras :: recupera(No *n, FILE *fp) List_no.cpp\ (98) lista_nos :: ~lista_nos() No.cpp\ (142) int No::criaversao(char *nome_no) </pre>
Procedimento: (103) virtual int eh_video() Diretório/Arquivo: No.h	Descrição: Chama: Chamado por: Grafov.w.cpp\ (542) d CGrafoView::DesenhaNo(No *no) Grafov.w.cpp\ (1220) void CGrafoView::OnRButtonDown(UINT nFlags, CPoint point) Hipger.cpp\ (557) extern "C" int FAR PASCAL _export recupera_nome_arquivo_video(char *nome_video, char *arq_video) Hipger.cpp\ (569) extern "C" int FAR PASCAL _export insere_nome_arquivo_video(char *nome_video, char *arq_video) Hipger.cpp\ (712) extern "C" int FAR PASCAL _export tipo_no(char *nome) Hipger.cpp\ (1311) extern "C" int FAR PASCAL _export recupera_nome_ancora(char *nome, int id_ancora, char *nome_ancora) Hipger.cpp\ (1372) extern "C" int FAR PASCAL _export primeira_ancora(char *nome, char *nome_ancora) Hipger.cpp\ (1425) extern "C" int FAR PASCAL _export proxima_ancora(char

	<pre> *nome, char *nome_ancora) List_anc.cpp\ (118) int lista_ancoras :: recupera(No *n, FILE *fp) List_no.cpp\ (98) lista_nos :: ~lista_nos() No.cpp\ (142) int No::criaversao(char *nome no) </pre>
Procedimento: <pre>(104) virtual int eh_audio()</pre> Diretório/Arquivo: No.h	Descrição: Chama: Chamado por: <pre> Grafov.cpp\ (542) d CGrafoView::DesenhaNo(No *no) Grafov.cpp\ (1220) void CgrafoView::OnRButtonDown(UINT nFlags, CPoint point) Hipger.cpp\ (605) extern "C" int FAR PASCAL _export recupera_nome_arquivo_audio(char *nome_audio, char *arquivo) Hipger.cpp\ (617) extern "C" int FAR PASCAL _export insere_nome_arquivo_audio(char *nome_audio, char *arquivo) Hipger.cpp\ (712) extern "C" int FAR PASCAL _export tipo_no(char *nome) Hipger.cpp\ (1311) extern "C" int FAR PASCAL _export recupera_nome_ancora(char *nome, int id_ancora, char *nome_ancora) Hipger.cpp\ (1372) extern "C" int FAR PASCAL _export primeira_ancora(char *nome, char *nome_ancora) Hipger.cpp\ (1425) extern "C" int FAR PASCAL _export proxima_ancora(char *nome, char *nome_ancora) List_anc.cpp\ (118) int lista_ancoras :: recupera(No *n, FILE *fp) List_no.cpp\ (98) lista_nos :: ~lista_nos() No.cpp\ (142) int No::criaversao(char *nome no) </pre>
Procedimento: <pre>(105) virtual int eh_imagem()</pre> Diretório/Arquivo: No.h	Descrição: Chama: Chamado por: <pre> Grafov.cpp\ (173) void CGrafoView::OnDraw(CDC* pDC) Grafov.cpp\ (542) d CGrafoView::DesenhaNo(No *no) Grafov.cpp\ (1220) void CgrafoView::OnRButtonDown(UINT nFlags, CPoint point) Hipger.cpp\ (629) extern "C" int FAR PASCAL _export recupera_nome_arquivo_imagem(char *nome_imagem, char *arquivo) Hipger.cpp\ (641) extern "C" int FAR PASCAL _export insere_nome_arquivo_imagem(char *nome_imagem, char *arquivo) Hipger.cpp\ (712) extern "C" int FAR PASCAL _export tipo_no(char *nome) Hipger.cpp\ (1311) extern "C" int FAR PASCAL _export recupera_nome_ancora(char *nome, int id_ancora, char *nome_ancora) Hipger.cpp\ (1372) extern "C" int FAR PASCAL _export primeira_ancora(char </pre>

	<pre> *nome, char *nome_ancora) Hipger.cpp\ (1425) extern "C" int FAR PASCAL _export proxima_ancora(char *nome, char *nome_ancora) List_anc.cpp\ (118) int lista_ancoras :: recupera(No *n, FILE *fp) List_no.cpp\ (98) lista_nos :: ~lista_nos() No.cpp\ (142) int No::criaversao(char *nome no) </pre>
Procedimento: (106) virtual int eh_executavel() Diretório/Arquivo: No.h	Descrição: Chama: Chamado por: <pre> Hipger.cpp\ (654) extern "C" int FAR PASCAL _export recupera_nome_arquivo_executavel(char *nome_executavel, char *arquivo) Hipger.cpp\ (666) extern "C" int FAR PASCAL _export insere_nome_arquivo_executavel(char *nome_executavel, char *arquivo) Hipger.cpp\ (712) extern "C" int FAR PASCAL _export tipo_no(char *nome) Hipger.cpp\ (1311) extern "C" int FAR PASCAL _export recupera_nome_ancora(char *nome, int id_ancora, char *nome_ancora) Hipger.cpp\ (1372) extern "C" int FAR PASCAL _export primeira_ancora(char *nome, char *nome_ancora) Hipger.cpp\ (1425) extern "C" int FAR PASCAL _export proxima_ancora(char *nome, char *nome_ancora) </pre>
Procedimento: (107) virtual int eh_tabela() Diretório/Arquivo: No.h	Descrição: Chama: Chamado por: <pre> Hipger.cpp\ (678) extern "C" int FAR PASCAL _export recupera_nome_arquivo_tabela(char *nome_tabela, char *arquivo) Hipger.cpp\ (690) extern "C" int FAR PASCAL _export insere_nome_arquivo_tabela(char *nome_tabela, char *arquivo) Hipger.cpp\ (712) extern "C" int FAR PASCAL _export tipo_no(char *nome) Hipger.cpp\ (1311) extern "C" int FAR PASCAL _export recupera_nome_ancora(char *nome, int id_ancora, char *nome_ancora) Hipger.cpp\ (1372) extern "C" int FAR PASCAL _export primeira_ancora(char *nome, char *nome_ancora) Hipger.cpp\ (1425) extern "C" int FAR PASCAL _export proxima_ancora(char *nome, char *nome_ancora) </pre>
Procedimento: (108) virtual int eh_contexto() Diretório/Arquivo: No.h	Descrição: Chama: Chamado por: <pre> Trilha.cpp\ (251) int trilha::primeiro_no_trilha(char *nome, char *nome_liberdade) Trilha.cpp\ (278) int trilha::proximo_no_trilha(char *nome, char *nome_liberdade) Trilha.cpp\ (305) int </pre>

	<pre> trilha::anterior_no_trilha(char *nome, char *nome_liberdade) Trilha.cpp\ (332) int trilha::ultimo_no_trilha(char *nome, char *nome_liberdade) Hipger.cpp\ (712) extern "C" int FAR PASCAL _export tipo_no(char *nome) Hipger.cpp\ (731) extern "C" int FAR PASCAL _export retira_da_hiperbase(char *nome_no) Hipger.cpp\ (964) extern "C" int FAR PASCAL _export muda_assunto(char *nome_no, char *assunto) Hipger.cpp\ (979) extern "C" int FAR PASCAL _export recupera_assunto(char *nome_no, char *assunto) Hipger.cpp\ (1311) extern "C" int FAR PASCAL _export recupera_nome_ancora(char *nome, int id_ancora, char *nome_ancora) Hipger.cpp\ (1372) extern "C" int FAR PASCAL _export primeira_ancora(char *nome, char *nome_ancora) Hipger.cpp\ (1425) extern "C" int FAR PASCAL _export proxima_ancora(char *nome, char *nome_ancora) Hip.cpp\ (177) void hipwindow::atualiza_base_ativa(No *pno) </pre>
Procedimento: (109) virtual int eh_trilha() Diretório/Arquivo: No.h	Descrição: Chama: Chamado por: Grafov.cpp\ (542) d CGrafoView::DesenhaNo(No *no) Hip.cpp\ (177) void hipwindow::atualiza_base_ativa(No *pno) Hipger.cpp\ (288) extern "C" int FAR PASCAL _export insere_na_trilha(char *nome_trilha, char *nome_no, char *no_liberdade) Hipger.cpp\ (317) extern "C" int FAR PASCAL _export insere_na_posicao(char *nome_trilha, char *nome_no, char *no_liberdade, int posicao) Hipger.cpp\ (359) extern "C" int FAR PASCAL _export retira_da_trilha(char *nome_trilha, char *nome_no, char *no_liberdade) Hipger.cpp\ (376) extern "C" int FAR PASCAL _export limpa_trilha(char *nome_trilha) Hipger.cpp\ (472) extern "C" int FAR PASCAL _export primeiro_no_trilha(char *nome_trilha, char *nome_no, char *nome_liberdade) Hipger.cpp\ (483) extern "C" int FAR PASCAL _export proximo_no_trilha(char *nome_trilha, char *nome_no, char *nome_liberdade) Hipger.cpp\ (494) extern "C" int FAR PASCAL _export ultimo_no_trilha(char *nome_trilha, char *nome_no, char *nome_liberdade) Hipger.cpp\ (505) extern "C" int FAR PASCAL _export anterior_no_trilha(char *nome_trilha, char *nome_no, char *nome_liberdade) Hipger.cpp\ (516) No *FAR PASCAL _export primeiro_ponteiro_no(char *nome_cont)

	<pre> (533) No *FAR PASCAL _export proximo_ponteiro_no(char *nome_cont) Hipger.cpp\ (712) extern "C" int FAR PASCAL _export tipo_no(char *nome) Hipger.cpp\ (731) extern "C" int FAR PASCAL _export retira_da_hiperbase(char *nome_no) Hipger.cpp\ (964) extern "C" int FAR PASCAL _export muda_assunto(char *nome_no, char *assunto) Hipger.cpp\ (1067) extern "C" int FAR PASCAL _export regras_tutoriais(char *no_atual, char *escopo, char *historico, int dificuldade, char *no_result, char *lib_result, int *revisto) Hipger.cpp\ (1697) elo *FAR PASCAL _export primeiro_ponteiro_elo(char *nome_contexto) Hipger.cpp\ (1714) elo *FAR PASCAL _export proximo_ponteiro_elo(char *nome_contexto, elo *pelo) List_anc.cpp\ (118) int lista_ancoras :: recupera(No *n, FILE *fp) List_no.cpp\ (98) lista_nos :: ~lista_nos() No.cpp\ (142) int No::criaversao(char *nome_no) </pre>
Procedimento: (110) virtual int eh_contexto_usuario() Diretório/Arquivo: No.h	Descrição: Chama: Chamado por: Grafov.w.cpp\ (542) void CGrafoView::DesenhaNo(No *no) Grafov.w.cpp\ (1140) void CGrafoView::OnLButtonDown(UINT nFlags, CPoint point) Grafov.w.cpp\ (1220) void CGrafoView::OnRButtonDown(UINT nFlags, CPoint point) Hipger.cpp\ (731) extern "C" int FAR PASCAL _export retira_da_hiperbase(char *nome_no) Hipger.cpp\ (1647) extern "C" int FAR PASCAL _export recupera_elo(char *nome_contexto, int id_elo, char *id_no_orig, int *id_anc_orig, char *id_no_dest, int *id_anc_dest, int *simultaneo, int *mostra_conteudo) List_anc.cpp\ (118) int lista_ancoras :: recupera(No *n, FILE *fp) List_elo.cpp\ (61) int lista_elos::recupera(No *pno, FILE *fp) List_no.cpp\ (98) lista_nos :: ~lista_nos() No.cpp\ (142) int No::criaversao(char *nome_no)
Procedimento: (111) virtual int eh_contexto_versoes() Diretório/Arquivo: No.h	Descrição: Chama: Chamado por: Chamado por: Hipger.cpp\ (263) extern "C" int FAR PASCAL _export insere_no_contexto(char *nome_cont, char *nome_no)

	<pre> Hipger.cpp\ (339) extern "C" int FAR PASCAL _export retira_do_contexto(char *nome_contexto, char *nome_no) (359) extern "C" int FAR PASCAL _export retira_da_trilha(char *nome_trilha, char *nome_no_liberdade) Hipger.cpp\ (376) extern "C" int FAR PASCAL _export limpa_trilha(char *nome_trilha) Hipger.cpp\ (712) extern "C" int FAR PASCAL _export tipo_no(char *nome) Hipger.cpp\ (731) extern "C" int FAR PASCAL _export retira_da_hiperbase(char *nome_no) Hipger.cpp\ (1121) extern "C" int FAR PASCAL _export cria_versao(char *nome_no, char *nome_no_versao) Hipger.cpp\ (1138) extern "C" int FAR PASCAL _export consulta_vcorrente(char *nome_cv, char *nome_no) Hipger.cpp\ (1154) extern "C" int FAR PASCAL _export muda_vcorrente(char *nome_cv, char *nome_no) Hipger.cpp\ (1517) extern "C" int FAR PASCAL _export cria_elo(char *nome_contexto, int id_elo, char *origem, int anc_origem, char *destino, int anc_destino, int simultaneo, int mostra_conteudo) Hipger.cpp\ (1560) extern "C" int FAR PASCAL _export muda_elo(char *nome_contexto, int id_elo, char *origem, int anc_origem, char *destino, int anc_destino, int simultaneo, int mostra_conteudo) Hipger.cpp\ (1608) extern "C" int FAR PASCAL _export retira_elo(char *nome_contexto, int id_elo) List_anc.cpp\ (118) int lista_ancoras :: recupera(No *n, FILE *fp) List_no.cpp\ (98) lista_nos :: ~lista_nos() Composic.cpp\ (89) int composicao::commit() </pre>
Procedimento: (112) virtual int eh_hiperbase_publica() Diretório/Arquivo: No.h	Descrição: Chama: Chamado por:
Procedimento: (113) virtual int eh_base_privada() Diretório/Arquivo: No.h	Descrição: Chama: Chamado por: Hip.cpp\ (78) int hipwindow::ativa_hiperbase(char *nome) Hip.cpp\ (177) void hipwindow::atualiza_base_ativa(No *pno) Hipger.cpp\ (731) extern "C" int FAR PASCAL _export retira_da_hiperbase(char *nome_no) List_no.cpp\ (98) lista_nos :: ~lista_nos() List_no.cpp\ (182) int lista_nos::grava(FILE *fp)
Procedimento: (114) virtual int eh anotacao()	Descrição:

Diretório/Arquivo: No.h	Chama: Chamado por:
Procedimento: (116) virtual int grava(FILE *)	Descrição: Protótipo da função implementada em No.cpp
Diretório/Arquivo: No.h	Chama: Chamado por:
Procedimento: (117) virtual int recupera(FILE *)	Descrição: Protótipo da função implementada em No.cpp
Diretório/Arquivo: No.h	Chama: Chamado por:
Procedimento: (118) virtual int grava_no(FILE *)	Descrição: Protótipo da função implementada em No.cpp
Diretório/Arquivo: No.h	Chama: Chamado por:
Procedimento: (119) virtual int recupera_no(FILE *)	Descrição: Protótipo da função implementada em No.cpp
Diretório/Arquivo: No.h	Chama: Chamado por:

Sistema: SASHE**Arquivo: Sashe\Hyperprop\ No_audio.cpp**

Procedimento: (15) no_audio::no_audio(char *nome_no) : no_terminal(nome_no)	Descrição: Construtor usado na criacao
Diretório/Arquivo: No_audio.cpp	Chama: (15) no_terminal(nome_no) Chamado por: Hipger.cpp\ (213) extern "C" int FAR PASCAL _export cria_audio(char *nome) No.cpp\ (142) int No::criaversao(char *nome_no)
Procedimento: (22) no_audio::no_audio(int id) : no_terminal(id)	Descrição: Construtor usado na recuperacao do no'
Diretório/Arquivo: No_audio.cpp	Chama: (22) no_terminal(id) Chamado por: Basepriv.cpp\ (133) int baseprivada::recupera(FILE *fp)
Procedimento: (28) void no_audio::copia_audio(no_audio *nt)	Descrição: Faz uma copia para um no vazio
Diretório/Arquivo: No_audio.cpp	Chama: (30) muda_nome_arquivo(nt->recupera_nome_arquivo()) (31) copia_ancoras_audio(&nt->ancoras) Chamado por: No.cpp\ (142) int No::criaversao(char *nome_no)
Procedimento: (38) int no_audio::cria_ancora_audio(int ide)	Descrição: Cria ancora em um no audio e insere-a na lista de ancoras do no' Retorna: 1: se falta memória. 0: se OK.
Diretório/Arquivo: No_audio.cpp	Chama: (40) new ancora_audio(retorna_indice(), ide) (42) insere_ancora(panc) Chamado por: Hipger.cpp\ (213) extern "C" int FAR PASCAL _export cria_audio(char *nome) No_audio.cpp\ (55) void no_audio::copia_ancoras_audio(lista anco

	ras *ancoras)
Procedimento: (47) void no_audio::muda_ancora_audio(ancora_audio *p, int ide) Diretório/Arquivo: No_audio.cpp	Descrição: Muda o conteúdo de uma ancora Chama: (49) p->troca_identificador(ide) Chamado por:
Procedimento: (55) void no_audio::copia_ancoras_audio(lista_ancoras *ancoras) Diretório/Arquivo: No_audio.cpp	Descrição: Faz uma copia das ancoras. Utilizado quando se cria uma versao Chama: (57) ancoras->primeiro() (61) cria_ancora_audio(anc->obtem_identificador()) (62) ancoras->proximo(anc) Chamado por: No_audio.cpp\ (28) void no_audio::copia_audio(no_audio *nt)
Procedimento: (68) int no_audio::grava(FILE *fp) Diretório/Arquivo: No_audio.cpp	Descrição: Chama: (70) fprintf(fp, "áudio){\n") (71) No::grava(fp) Chamado por:
Procedimento: (77) int no_audio::recupera(FILE *fp) Diretório/Arquivo: No_audio.cpp	Descrição: Chama: (79) No::recupera(fp) Chamado por:

Sistema: SASHE**Arquivo:** Sashe\Hyperprop\ No_audio.h

Procedimento: (22) int no_audio::eh_audio() Diretório/Arquivo: No_audio.h	Descrição: Chama: Chamado por: Grafov.w.cpp\ (542) d CGrafoView::DesenhaNo(No *no) Grafov.w.cpp\ (1220) void CgrafoView::OnRButtonDown(UINT nFlags, CPoint point) Hipger.cpp\ (605) extern "C" int FAR PASCAL _export recupera_nome_arquivo_audio(char *nome_audio, char *arquivo) Hipger.cpp\ (617) extern "C" int FAR PASCAL _export insere_nome_arquivo_audio(char *nome_audio, char *arquivo) Hipger.cpp\ (712) extern "C" int FAR PASCAL _export tipo_no(char *nome) Hipger.cpp\ (1311) extern "C" int FAR PASCAL _export recupera_nome_ancora(char *nome, int id_ancora, char *nome_ancora) Hipger.cpp\ (1372) extern "C" int FAR PASCAL _export primeira_ancora(char *nome, char *nome_ancora) Hipger.cpp\ (1425) extern "C" int FAR PASCAL _export proxima_ancora(char *nome, char *nome_ancora) List_anc.cpp\ (118) int lista_ancoras :: recupera(No *n, FILE *fp) List_no.cpp\ (98) lista_nos :: ~lista_nos()
---	---

No.cpp\ (142) int No::criaversao(char *nome_no)	
Sistema: SASHE	
Arquivo: Sashe\Hyperprop\ No_exec.cpp	
Procedimento: (15) no_executavel::no_executavel(char *nome_no) : no_terminal(nome_no) Diretório/Arquivo: No_exec.cpp	Descrição: Construtor usado na criaçao Chama: (15) no_terminal(nome_no) Chamado por: Hipger.cpp\ (239) extern "C" int FAR PASCAL _export cria_executavel(char *nome)
Procedimento: (22) no_executavel::no_executavel(int id) : no_terminal(id) Diretório/Arquivo: No_exec.cpp	Descrição: Construtor usado na recuperacao do no' Chama: (22) no_terminal(id) Chamado por: Basepriv.cpp\ (133) int baseprivada::recupera(FILE *fp)
Procedimento: (28) void no_executavel::copia_executavel(no_executavel *nt) Diretório/Arquivo: No_exec.cpp	Descrição: Faz uma copia para um no vazio Chama: (30) muda_nome_arquivo(nt->recupera_nome_arquivo()) (31) copia_ancoras_executavel(&nt->ancoras) Chamado por:
Procedimento: (38) int no_executavel::cria_ancora_executavel(int ide) Diretório/Arquivo: No_exec.cpp	Descrição: Cria ancora em um no audio e insere-a na lista de ancoras do no' Retorna: 1: se falta memória. 0: OK. Chama: (40) new ancora_executavel(retorna_indice(), ide) (42) insere_ancora(panc) Chamado por: Hipger.cpp\ (239) extern "C" int FAR PASCAL _export cria_executavel(char *nome) No_exec.cpp\ (55) Void no_executavel::copia_ancoras_executavel(lista ancoras *ancoras)
Procedimento: (47) void no_executavel::muda_ancora_executavel(ancora_executavel *p, int ide) Diretório/Arquivo: No_exec.cpp	Descrição: Muda o conteudo de uma ancora Chama: (49) p->troca_identificador(ide) Chamado por:
Procedimento: (55) void no_executavel::copia_ancoras_executavel(lista ancoras *ancoras) Diretório/Arquivo: No_exec.cpp	Descrição: Faz uma copia das ancoras. Utilizado quando se cria uma versao Chama: (57) ancoras->primeiro() (61) cria_ancora_executavel(anc->obtem_identificador())

	(62) ancoras->proximo(anc) Chamado por: No_exec.cpp\ (28) void no_executavel::copia_executavel(no_execu tavel *nt)
Procedimento: (68) int no_executavel::grava(FILE *fp) Diretório/Arquivo: No_exec.cpp	Descrição: Chama: (70) fprintf(fp, "executável){\n") (71) No::grava(fp) Chamado por:
Procedimento: (77) int no_executavel::recupera(FILE *fp) Diretório/Arquivo: No_exec.cpp	Descrição: Chama: (79) No::recupera(fp) Chamado por:

Sistema: SASHE Arquivo: Sashe\Hyperprop\ No_exec.h	
Procedimento: (22) int no_executavel::eh_executavel() Diretório/Arquivo: No_exec.h	Descrição: Chama: Chamado por: Hipger.cpp\ (654) extern "C" int FAR PASCAL _export recupera_nome_arquivo_executavel(char *nome_executavel, char *arquivo) Hipger.cpp\ (666) extern "C" int FAR PASCAL _export insere_nome_arquivo_executavel(char *nome_executavel, char *arquivo) Hipger.cpp\ (712) extern "C" int FAR PASCAL _export tipo_no(char *nome) Hipger.cpp\ (1311) extern "C" int FAR PASCAL _export recupera_nome_ancora(char *nome, int id_ancora, char *nome_ancora) Hipger.cpp\ (1372) extern "C" int FAR PASCAL _export primeira_ancora(char *nome, char *nome_ancora) Hipger.cpp\ (1425) extern "C" int FAR PASCAL _export proxima_ancora(char *nome, char *nome_ancora)

Sistema: SASHE Arquivo: Sashe\Hyperprop\ No_image.cpp	
Procedimento: (16) no_imagem::no_imagem(char *nome_no) : no_terminal(nome_no) Diretório/Arquivo: No_image.cpp	Descrição: Construtor usado na criaçao Chama: (16) no_terminal(nome_no) Chamado por: Hipger.cpp\ (226) extern "C" int FAR PASCAL _export cria_imagem(char *nome) No.cpp\ (142) int No::criaversao(char *nome_no)
Procedimento: (23) no_imagem::no_imagem(int id) : no_terminal(id) Diretório/Arquivo: No_image.cpp	Descrição: Construtor usado na recuperacao do no' Chama: (23) no_terminal(id) Chamado por: Basepriv.cpp\ (133) int baseprivada::recupera(FILE *fp)
Procedimento:	Descrição:

(29) void no_imagem::copia_imagem(no_imagem *nt) Diretório/Arquivo: No_image.cpp	Faz uma copia para um no vazio Chama: (31) muda_nome_arquivo(nt- >recupera_nome_arquivo()) (32) copia_ancoras_imagem(&nt->ancoras) Chamado por: No.cpp\ (142) int No::criaversao(char *nome no)
Procedimento: (39) int no_imagem::cria_ancora_imagem(int ide) Diretório/Arquivo: No_image.cpp	Descrição: Cria ancora em um no imagem e insere-a na lista de ancoras do no' Retorna: 1: se falta memória. 0: se OK. Chama: (41) new ancora_imagem(retorna_indice(), ide) (43) insere_ancora(panc) Chamado por: Hipger.cpp\ (226) extern "C" int FAR PASCAL _export cria_imagem(char *nome) No_image.cpp\ (56) void no_imagem::copia_ancoras_imagem(lista_ancoras *ancoras)
Procedimento: (48) void no_imagem::muda_ancora_imagem(ancora_imagem *p, int ide) Diretório/Arquivo: No_image.cpp	Descrição: Muda o conteudo de uma ancora Chama: (50) p->troca_identificador(ide) Chamado por:
Procedimento: (56) void no_imagem::copia_ancoras_imagem(lista_ancoras *ancoras) Diretório/Arquivo: No_image.cpp	Descrição: Faz uma copia das ancoras. Utilizado quando se cria uma versao Chama: (58) ancoras->primeiro() (62) cria_ancora_imagem(anc->obtem_identificador()) (63) ancoras->proximo(anc) Chamado por: No_image.cpp\ (29) void no_imagem::copia_imagem(no_imagem *nt)
Procedimento: (69) int no_imagem::grava(FILE *fp) Diretório/Arquivo: No_image.cpp	Descrição: Chama: (71) fprintf(fp, "imagem) (\n") (72) No::grava(fp) Chamado por:
Procedimento: (78) int no_imagem::recupera(FILE *fp) Diretório/Arquivo: No_image.cpp	Descrição: Chama: (80) (No::recupera(fp) Chamado por:
Sistema: SASHE Arquivo: Sashe\Hyperprop\ No_image.h	
Procedimento: (22) int no_imagem::eh_imagem() Diretório/Arquivo: No_image.h	Descrição: Chama: Chamado por: Grafov.cpp\ (173) void CGrafoView::OnDraw(CDC* pDC) Grafov.cpp\ (542) d

CGrafoView::DesenhaNo(No *no) Grafovw.cpp\ (1220) void CGrafoView::OnRButtonDown(UINT nFlags, CPoint point) Hipger.cpp\ (629) extern "C" int FAR PASCAL _export recupera_nome_arquivo_imagem(char *nome_imagem, char *arquivo) Hipger.cpp\ (641) extern "C" int FAR PASCAL _export insere_nome_arquivo_imagem(char *nome_imagem, char *arquivo) Hipger.cpp\ (712) extern "C" int FAR PASCAL _export tipo_no(char *nome) Hipger.cpp\ (1311) extern "C" int FAR PASCAL _export recupera_nome_ancora(char *nome, int id_ancora, char *nome_ancora) Hipger.cpp\ (1372) extern "C" int FAR PASCAL _export primeira_ancora(char *nome, char *nome_ancora) Hipger.cpp\ (1425) extern "C" int FAR PASCAL _export proxima_ancora(char *nome, char *nome_ancora) List_anc.cpp\ (118) int lista_ancoras :: recupera(No *n, FILE *fp) List_no.cpp\ (98) lista_nos :: ~lista_nos() No.cpp\ (142) int No::criaversao(char *nome_no)

Sistema: SASHE**Arquivo:** Sashe\Hyperprop\ No_tab.cpp

Procedimento: (15) no_tabela::no_tabela(char *nome_no) : no_terminal(nome_no) Diretório/Arquivo: No_tab.cpp	Descrição: Construtor usado na criaçao Chama: (15) no_terminal(nome_no) Chamado por: Hipger.cpp\ (252) extern "C" int FAR PASCAL _export cria_tabela(char *nome)
Procedimento: (22) no_tabela::no_tabela(int id) : no_terminal(id) Diretório/Arquivo: No_tab.cpp	Descrição: Construtor usado na recuperacao do no' Chama: (22) no_terminal(id) Chamado por:
Procedimento: (28) void no_tabela::copia_tabela(no_tabela *nt) Diretório/Arquivo: No_tab.cpp	Descrição: Faz uma copia para um no vazio Chama: (30) muda_nome_arquivo(nt-> recupera_nome_arquivo()) (31) copia_ancoras_tabela(&nt->ancoras) Chamado por:
Procedimento: (38) int no_tabela::cria_ancora_tabela(int ide) Diretório/Arquivo: No_tab.cpp	Descrição: Cria ancora em um no tabela e insere-a na lista de ancoras do no' Retorna: 1: se falta memória. 0: se OK. Chama: (40) new ancora_tabela(retorna_indice(), ide) (42) insere_ancora(panc) Chamado por: Hipger.cpp\ (252) extern "C" int FAR

	PASCAL_export cria_tabela(char *nome) No_tab.cpp\ (55) void no_tabela::copia_ancoras_tabela(lista_ancoras *ancoras)
Procedimento: (47) no_tabela::muda_ancora_tabela(ancora_tabela *p, int ide) Diretório/Arquivo: No_tab.cpp	Descrição: Muda o conteúdo de uma ancora Chama: (49) p->troca_identificador(ide) Chamado por:
Procedimento: (55) void no_tabela::copia_ancoras_tabela(lista_ancoras *ancoras) Diretório/Arquivo: No_tab.cpp	Descrição: Faz uma copia das ancoras. Utilizado quando se cria uma versao Chama: (57) ancoras->primeiro() (61) cria_ancora_tabela(ancora->obtem_identificador()) (62) ancoras->proximo(anc) Chamado por: No_tab.cpp\ (28) void no_tabela::copia_tabela(no_tabela *nt)
Procedimento: (68) int no_tabela::grava(FILE *fp) Diretório/Arquivo: No_tab.cpp	Descrição: Chama: (70) fprintf(fp, "áudio){\n") (71) No::grava(fp) Chamado por:
Procedimento: (77) int no_tabela::recupera(FILE *fp) Diretório/Arquivo: No_tab.cpp	Descrição: Chama: (79) No::recupera(fp) Chamado por:

Sistema: SASHE**Arquivo:** Sashe\Hyperprop\ No_tab.h

Procedimento: (22) int no_tabela::eh_tabela() Diretório/Arquivo: No_tab.h	Descrição: Chama: Chamado por: Hipger.cpp\ (678) extern "C" int FAR PASCAL_export recupera_nome_arquivo_tabela(char *nome_tabela, char *arquivo) Hipger.cpp\ (690) extern "C" int FAR PASCAL_export insere_nome_arquivo_tabela(char *nome_tabela, char *arquivo) Hipger.cpp\ (712) extern "C" int FAR PASCAL_export tipo_no(char *nome) Hipger.cpp\ (1311) extern "C" int FAR PASCAL_export recupera_nome_ancora(char *nome, int id_ancora, char *nome_ancora) Hipger.cpp\ (1372) extern "C" int FAR PASCAL_export primeira_ancora(char *nome, char *nome_ancora) Hipger.cpp\ (1425) extern "C" int FAR PASCAL_export proxima_ancora(char *nome, char *nome_ancora)
---	---

Sistema: SASHE**Arquivo:** Sashe\Hyperprop\ No_texto.cpp

Procedimento: (18) no_texto::no_texto(char *nome no) : no_terminal(nome no)	Descrição: Construtor usado na criacao
--	---

Diretório/Arquivo: No_texto.cpp	Chama: (18) no_terminal(nome_no) Chamado por: Hipger.cpp\ (187) extern "C" int FAR PASCAL _export cria_texto(char *nome) No.cpp\ (142) int No::criaversao(char *nome_no)
Procedimento: (25) no_texto::no_texto(int id) : no_terminal(id) Diretório/Arquivo: No_texto.cpp	Descrição: Construtor usado na recuperacao do no' Chama: (25) no_terminal(id) Chamado por: Basepriv.cpp\ (133) int baseprivada::recupera(FILE *fp)
Procedimento: (32) void no_texto::copia_texto(no_texto *nt) Diretório/Arquivo: No_texto.cpp	Descrição: Faz uma copia para um no vazio Chama: (34) muda_nome_arquivo(nt->recupera_nome_arquivo()) (35) copia_ancoras_texto(&nt->ancoras) Chamado por: No.cpp\ (142) int No::criaversao(char *nome_no)
Procedimento: (42) int no_texto::cria_ancora_texto(int ide, char *pal, int inicio, int fim) Diretório/Arquivo: No_texto.cpp	Descrição: Cria ancora em um no texto e insere-a na lista de ancoras do no' Retorna: 1: se falta memória. 0: se OK. Chama: (44) new ancora_texto(retorna_indice(), ide, pal, inicio, fim) (46) insere_ancora(panc) Chamado por: Hipger.cpp\ (187) extern "C" int FAR PASCAL _export cria_texto(char *nome) Hipger.cpp\ (1181) extern "C" int FAR PASCAL _export cria_ancora_texto(char *nome_no, int ident, char *palavra, int inicio, int fim) No_texto.cpp\ (60) void no_texto::copia_ancoras_texto(lista_ancoras *ancoras)
Procedimento: (51) void no_texto::muda_ancora_texto(ancora *p, int ide, char *pal, int inicio, int fim) Diretório/Arquivo: No_texto.cpp	Descrição: Muda o conteudo de uma ancora Chama: (54) pt->muda_conteudo(ide, pal, inicio, fim) Chamado por: Hipger.cpp\ (1260) extern "C" int FAR PASCAL _export muda_ancora_texto(char *nome, int velho_ide, int novo_ide, char *palavra, int inicio, int fim)
Procedimento: (60) void no_texto::copia_ancoras_texto(lista_ancoras *ancoras) Diretório/Arquivo: No_texto.cpp	Descrição: Faz uma copia das ancoras. Utilizado quando se cria uma versao Chama: (62) ancoras->primeiro() (66) cria_ancora_texto(anc->obtem_identificador(), anc->obtem_ancora(), anc-

	>obtem_inicio_ancora(), anc- >obtem_fim_ancora() (68) ancoras->proximo(anc) Chamado por: No_texto.cpp\ (32) void no_texto::copia_texto(no_texto *nt)
Procedimento: (74) int no_texto::grava(FILE *fp) Diretório/Arquivo: No_texto.cpp	Descrição: Chama: (76) fprintf(fp, "texto{\n") (77) No::grava(fp) Chamado por:
Procedimento: (83) int no_texto::recupera(FILE *fp) Diretório/Arquivo: No_texto.cpp	Descrição: Chama: (85) No::recupera(fp) Chamado por:

Sistema: SASHE**Arquivo: Sashe\Hyperprop\ No_texto.h**

Procedimento: (25) int no_texto::eh_texto() Diretório/Arquivo: No_texto.h	Descrição: Chama: Chamado por: Grafov.cpp\ (542) void CGrafoView::DesenhaNo(No *no) Grafov.cpp\ (1220) void CGrafoView::OnRButtonDown(UINT nFlags, CPoint point) Hipger.cpp\ (581) extern "C" int FAR PASCAL _export recupera_nome_arquivo_texto(char *nome_texto, char *arquivo) Hipger.cpp\ (593) extern "C" int FAR PASCAL _export insere_nome_arquivo_texto(char *nome_texto, char *arquivo) Hipger.cpp\ (712) extern "C" int FAR PASCAL _export tipo_no(char *nome) Hipger.cpp\ (1181) extern "C" int FAR PASCAL _export cria_ancora_texto(char *nome_no, int ident, char *palavra, int inicio, int fim) Hipger.cpp\ (1260) extern "C" int FAR PASCAL _export muda_ancora_texto(char *nome, int velho_ide, int novo_ide, char *palavra, int inicio, int fim) Hipger.cpp\ (1311) extern "C" int FAR PASCAL _export recupera_nome_ancora(char *nome, int id_ancora, char *nome_ancora) Hipger.cpp\ (1372) extern "C" int FAR PASCAL _export primeira_ancora(char *nome, char *nome_ancora) Hipger.cpp\ (1425) extern "C" int FAR PASCAL _export proxima_ancora(char *nome, char *nome_ancora) Hipger.cpp\ (1489) extern "C" int FAR PASCAL _export ultima_ancora(char *nome, char *nome_ancora) List_anc.cpp\ (118) int lista_ancoras :: recupera(No *n, FILE *fp) List_no.cpp\ (98) lista_nos :: ~lista_nos() No.cpp\ (142) int No::criaversao(char *nome no)
---	--

Sistema: SASHE	
Arquivo: Sashe\Hyperprop\ no_video.cpp	
Procedimento: (14) no_video::no_video(char *nome_no) : no_terminal(nome_no) Diretório/Arquivo: no_video.cpp	Descrição: Construtor usado na criação Chama: (14) no_terminal(nome_no) Chamado por: Hipger.cpp\ (200) extern "C" int FAR PASCAL _export cria_video(char *nome) No.cpp\ (142) int No::criaversao(char *nome no)
Procedimento: (21) no_video::no_video(int id) : no_terminal(id) Diretório/Arquivo: no_video.cpp	Descrição: Construtor usado na recuperação do no' Chama: (21) no_terminal(id) Chamado por: No.cpp\ (142) int No::criaversao(char *nome no)
Procedimento: (27) void no_video::copia_video(no_video *nt) Diretório/Arquivo: no_video.cpp	Descrição: Faz uma cópia para um no vazio Chama: (30) muda_nome_arquivo(nt->recupera_nome_arquivo()) (31) copia_ancoras_video(&nt->ancoras) Chamado por: No.cpp\ (142) int No::criaversao(char *nome no)
Procedimento: (37) int no_video::cria_ancora_video(int ide) Diretório/Arquivo: no_video.cpp	Descrição: Cria ancora em um no video e insere-a na lista de ancoras do no' Retorna: 1: se falta memória. 0: se OK. Chama: (39) new ancora_video(retorna_indice(), ide) (41) insere_ancora(panc) Chamado por: Hipger.cpp\ (200) extern "C" int FAR PASCAL _export cria_video(char *nome) no_video.cpp\ (54) void no_video::copia_ancoras_video(lista_ancoras *ancoras)
Procedimento: (46) void no_video::muda_ancora_video(ancora_video *p, int ide) Diretório/Arquivo: no_video.cpp	Descrição: Muda o conteúdo de uma ancora Chama: (48) p->troca_identificador(ide) Chamado por:
Procedimento: (54) void no_video::copia_ancoras_video(lista_ancoras *ancoras) Diretório/Arquivo: No_video.cpp	Descrição: Faz uma cópia das ancoras. Utilizado quando se cria uma versao Chama: (56) ancoras->primeiro() (60) cria_ancora_video(anc->obtem_identificador()) (61) ancoras->proximo(anc) Chamado por: No_video.cpp\ (27) void no_video::copia_video(no_video *nt)
Procedimento: (67) int no_video::grava(FILE *fp)	Descrição:

Diretório/Arquivo: no_video.cpp	Chama: (69) fprintf(fp, "video")\n") (70) No::grava(fp) Chamado por:
Procedimento: (76) int no_video::recupera(FILE *fp) Diretório/Arquivo: no_video.cpp	Descrição: Chama: (78) No::recupera(fp) Chamado por:

Sistema: SASHE Arquivo: Sashe\Hyperprop\ no_video.h	
Procedimento: (22) int no_video::eh_video() Diretório/Arquivo: no_video.h	Descrição: Chama: Chamado por: Grafov.w.cpp\ (542) d CGrafoView::DesenhaNo(No *no) Grafov.w.cpp\ (1220) void CGrafoView::OnRButtonDown(UINT nFlags, CPoint point) Hipger.cpp\ (557) extern "C" int FAR PASCAL _export recupera_nome_arquivo_video(char *nome_video, char *arq_video) Hipger.cpp\ (569) extern "C" int FAR PASCAL _export insere_nome_arquivo_video(char *nome_video, char *arq_video) Hipger.cpp\ (712) extern "C" int FAR PASCAL _export tipo_no(char *nome) Hipger.cpp\ (1311) extern "C" int FAR PASCAL _export recupera_nome_ancora(char *nome, int id_ancora, char *nome_ancora) Hipger.cpp\ (1372) extern "C" int FAR PASCAL _export primeira_ancora(char *nome, char *nome_ancora) Hipger.cpp\ (1425) extern "C" int FAR PASCAL _export proxima_ancora(char *nome, char *nome_ancora) List_anc.cpp\ (118) int lista_ancoras :: recupera(No *n, FILE *fp) List_no.cpp\ (98) lista_nos :: ~lista_nos() No.cpp\ (142) int No::criaversao(char *nome_no)

Sistema: SASHE Arquivo: Sashe\Hyperprop\ Pics.h	
Procedimento: (26) CPics () Diretório/Arquivo: Pics.h	Descrição: Chama: Chamado por:
Procedimento: (27) CPics (char *) Diretório/Arquivo: Pics.h	Descrição: Chama: Chamado por: Grafov.w.cpp\ (1220) void CGrafoView::OnRButtonDown(UINT nFlags, CPoint point)
Procedimento: (28) ~CPics () Diretório/Arquivo: Pics.h	Descrição: Chama: Chamado por:
Procedimento:	Descrição:

(29) virtual BOOL LoadPicture (CDC *, int Flags = 0) Diretório/Arquivo: Pics.h		Chama: Chamado por:
Procedimento: (30) virtual BOOL SavePicture (char *Filename = NULL, int PictureType = -1) Diretório/Arquivo: Pics.h	Descrição: Chama: Chamado por:	
Procedimento: (31) BOOL DisplayPicture (CDC *, int x = -1, int y = -1) Diretório/Arquivo: Pics.h	Descrição: Chama: Chamado por: Grafov.w.cpp\ (173) void CGrafoView::OnDraw(CDC* pDC)	

Sistema: SASHE		
Arquivo: Sashe\Hyperprop\ Sound.h		
Procedimento: (31) CMidi (CString Name) Diretório/Arquivo: Sound.h	Descrição: Chama: Chamado por:	
Procedimento: (32) ~CMidi () Diretório/Arquivo: Sound.h	Descrição: Chama: Chamado por:	
Procedimento: (43) BOOL Play (void) Diretório/Arquivo: Sound.h	Descrição: Chama: Chamado por:	
Procedimento: (44) BOOL IsPlaying (void) Diretório/Arquivo: Sound.h	Descrição: Chama: Chamado por:	
Procedimento: (45) BOOL Restart (void) Diretório/Arquivo: Sound.h	Descrição: Chama: Chamado por:	
Procedimento: (46) void Stop (void) Diretório/Arquivo: Sound.h	Descrição: Chama: Chamado por:	
Procedimento: (47) CString GetErrorString (void) Diretório/Arquivo: Sound.h	Descrição: Chama: Chamado por:	
Procedimento: (48) static int GetNumMidiDevices (void) Diretório/Arquivo: Sound.h	Descrição: Chama: Chamado por:	
Procedimento: (60) CWave (CString Name, int Type = WT_DISK) Diretório/Arquivo: Sound.h	Descrição: Chama: Chamado por: Grafov.w.cpp\ (1220) void CGrafoView::OnRButtonDown(UINT nFlags, CPoint point)	
Procedimento: (61) Cwave::~~CWave () Diretório/Arquivo: Sound.h	Descrição: Chama: Chamado por:	
Procedimento:	Descrição:	

(75) BOOL Play (void) Diretório/Arquivo: Sound.h	Chama: Chamado por: Grafov.w.cpp\ (1220) void CGrafoView::OnRButtonDown(UINT nFlags, CPoint point)
Procedimento: (76) BOOL PlayAsync (void) Diretório/Arquivo: Sound.h	Descrição: Chama: Chamado por:
Procedimento: (77) BOOL PlayLoop (void) Diretório/Arquivo: Sound.h	Descrição: Chama: Chamado por:
Procedimento: (78) void Stop (void) Diretório/Arquivo: Sound.h	Descrição: Chama: Chamado por:
Procedimento: (79) BOOL CanRecord (void) Diretório/Arquivo: Sound.h	Descrição: Chama: Chamado por:
Procedimento: (80) BOOL StartRecording (void) Diretório/Arquivo: Sound.h	Descrição: Chama: Chamado por:
Procedimento: (81) BOOL StopRecording (void) Diretório/Arquivo: Sound.h	Descrição: Chama: Chamado por:
Procedimento: (82) BOOL SaveRecording (CString) Diretório/Arquivo: Sound.h	Descrição: Chama: Chamado por:
Procedimento: (83) CString GetErrorString (void) Diretório/Arquivo: Sound.h	Descrição: Chama: Chamado por:
Procedimento: (84) static int GetNumWaveDevices (void) Diretório/Arquivo: Sound.h	Descrição: Chama: Chamado por: Grafov.w.cpp\ (1220) void CGrafoView::OnRButtonDown(UINT nFlags, CPoint point)

Sistema: SASHE**Arquivo: Sashe\Hyperprop\ Terminal.cpp**

Procedimento: (20) no_terminal::no_terminal(char *nome_no) : No(nome_no) Diretório/Arquivo: Terminal.cpp	Descrição: Construtor da Classe No Terminal usado na criação Chama: (20) No(nome_no) Chamado por: No_audio.cpp\ (15) no_audio::no_audio(char *nome_no) : no_terminal(nome_no) No_exec.cpp\ (15) no_executavel::no_executavel(char *nome_no) : no_terminal(nome_no) No_image.cpp\ (16) no_imagem::no_imagem(char *nome_no) : no_terminal(nome_no) No_tab.cpp\ (15)
--	--

	<pre> no_tabela::no_tabela(char *nome_no) : no_terminal(nome_no) No_texto.cpp\ (18) no_texto::no_texto(char *nome_no) : no_terminal(nome_no) No_video.cpp\ (14) no_video::no_video(char *nome_no) : no_terminal(nome_no) </pre>
Procedimento: (33) no_terminal::no_terminal(int id) : No(id) Diretório/Arquivo: Terminal.cpp	Descrição: Construtor da Classe No Terminal utilizado quando o nó é lido do arquivo. Chama: (33) No(id) Chamado por: No_audio.cpp\ (22) no_audio::no_audio(int id) : no_terminal(id) No_audio.cpp\ (22) no_executavel::no_executavel(int id) : no_terminal(id) No_image.cpp\ (23) no_imagem::no_imagem(int id) : no_terminal(id) No_tab.cpp\ (22) no_tabela::no_tabela(int id) : no_terminal(id) No_texto.cpp\ (25) no_texto::no_texto(int id) : no_terminal(id) No_video.cpp\ (21) no_video::no_video(int id) : no_terminal(id)
Procedimento: (45) int no_terminal::muda_nome_arquivo(char *nome) Diretório/Arquivo: Terminal.cpp	Descrição: Chama: Chamado por: Hipger.cpp\ (569) extern "C" int FAR PASCAL _export insere_nome_arquivo_video(char *nome_video, char *arq_video) Hipger.cpp\ (593) extern "C" int FAR PASCAL _export insere_nome_arquivo_texto(char *nome_texto, char *arquivo) Hipger.cpp\ (617) extern "C" int FAR PASCAL _export insere_nome_arquivo_audio(char *nome_audio, char *arquivo) Hipger.cpp\ (641) extern "C" int FAR PASCAL _export insere_nome_arquivo_imagem(char *nome_imagem, char *arquivo) Hipger.cpp\ (666) extern "C" int FAR PASCAL _export insere_nome_arquivo_executavel(char *nome_executavel, char *arquivo) Hipger.cpp\ (690) extern "C" int FAR PASCAL _export insere_nome_arquivo_tabela(char *nome_tabela, char *arquivo) No_audio.cpp\ (28) void no_audio::copia_audio(no_audio *nt) No_exec.cpp\ (28) void no_executavel::copia_executavel(no_executavel *nt)

	No_image.cpp\ (29) void no_imagem::copia_imagem(no_imagem *nt) No_tab.cpp\ (28) void no_tabela::copia_tabela(no_tabela *nt) No_texto.cpp\ (32) void no_texto::copia_texto(no_texto *nt) No_video.cpp\ (27) void no_video::copia_video(no_video *nt)
Procedimento: (61) int no_terminal::muda_tipo(char *stipo) Diretório/Arquivo: Terminal.cpp	Descrição: Chama: Chamado por: Hipger.cpp\ (890) extern "C" int FAR PASCAL _export muda_tipo(char *nome_no, char *tipo)
Procedimento: (70) int no_terminal::muda_funcao_didatica(char *string) Diretório/Arquivo: Terminal.cpp	Descrição: Chama: Chamado por: Hipger.cpp\ (927) extern "C" int FAR PASCAL _export muda_funcao_didatica(char *nome no, char *categoria)
Procedimento: (79) int no_terminal::muda_palavras_chaves(int pos, char *chaves) Diretório/Arquivo: Terminal.cpp	Descrição: Chama: Chamado por: Hipger.cpp\ (1028) extern "C" int FAR PASCAL _export muda_palavra_chave(char *nome no, int pos, char *palavra)
Procedimento: (88) char *no_terminal::recupera_nome_arquivo() Diretório/Arquivo: Terminal.cpp	Descrição: Chama: Chamado por: Grafov.cpp\ (1220) void CGrafoView::OnRButtonDown(UINT nFlags, CPoint point) Hipger.cpp\ (557) extern "C" int FAR PASCAL _export recupera_nome_arquivo_video(char *nome_video, char *arq_video) Hipger.cpp\ (581) extern "C" int FAR PASCAL _export recupera_nome_arquivo_texto(char *nome_texto, char *arquivo) Hipger.cpp\ (605) extern "C" int FAR PASCAL _export recupera_nome_arquivo_audio(char *nome_audio, char *arquivo) Hipger.cpp\ (629) extern "C" int FAR PASCAL _export recupera_nome_arquivo_imagem(char *nome_imagem, char *arquivo) Hipger.cpp\ (654) extern "C" int FAR PASCAL _export recupera_nome_arquivo_executavel(char *nome_executavel, char *arquivo) Hipger.cpp\ (678) extern "C" int FAR PASCAL _export recupera_nome_arquivo_tabela(char *nome_tabela, char *arquivo)

	No_audio.cpp\ (28) void no_audio::copia_audio(no_audio *nt) No_exec.cpp\ (28) void no_executavel::copia_executavel(no_executavel *nt) No_image.cpp\ (29) void no_imagem::copia_imagem(no_imagem *nt) No_tab.cpp\ (28) void no_tabela::copia_tabela(no_tabela *nt) No_texto.cpp\ (32) void no_texto::copia_texto(no_texto *nt) No_video.cpp\ (27) void no_video::copia_video(no_video *nt)
Procedimento: (106) int no_terminal::grava_no(FILE *fp) Diretório/Arquivo: Terminal.cpp	Descrição: Retorno: 1: Erro de gravação 0: Ok. Chama: (108) No::grava_no(fp) (110) fprintf(fp, "\tArquivo(%s)\n", arquivo) (111) fprintf(fp, "\tTipo(%s)\n", tipo) (112) fprintf(fp, "\tFunção Didática(%s)\n", funcao_didatica) (114) fprintf(fp, "\tPalavra-Chave(%s)\n", palavras_chaves[i]) Chamado por:
Procedimento: (123) int no_terminal::recupera_no(FILE *fp) Diretório/Arquivo: Terminal.cpp	Descrição: Retorno: 1: Erro de leitura 0: Ok. Chama: (125) No::recupera_no(fp) (126) fscanf(fp, "\tArquivo(%s)\n", arquivo) (135) fscanf(fp, "\tTipo(%s)\n", tipo) (136) fscanf(fp, "\tFunção Didática(%s)\n", funcao_didatica) (138) fscanf(fp, "\tPalavra-Chave(%s)\n", palavras_chaves[i]) Chamado por:

Sistema: SASHE Arquivo: Sashe\Hyperprop\ Terminal.h	
Procedimento: (30) char *recupera_tipo() Diretório/Arquivo: Terminal.h	Descrição: Chama: Chamado por: Hipger.cpp\ (908) extern "C" int FAR PASCAL _export recupera_tipo(char *nome_no, char *tipo)
Procedimento: (31) char *recupera_funcao_didatica() Diretório/Arquivo: Terminal.h	Descrição: Chama: Chamado por: Hipger.cpp\ (945) extern "C" int FAR PASCAL _export recupera_funcao_didatica(char *nome_no, char *categoria) Tutor.cpp\ (78) int tutor::segunda_estrategia(contexto *pai)
Procedimento: (32) char *recupera_palavras_chaves(int pos) Diretório/Arquivo: Terminal.h	Descrição: Chama: Chamado por:

	Hipger.cpp\ (1046) extern "C" int FAR PASCAL _export recupera_palavra_chave(char *nome_no, int pos, char *palavra)
Procedimento: (33) virtual int eh_terminal() Diretório/Arquivo: Terminal.h	Descrição: Chama: Chamado por: Grafov.w.cpp\ (173) void CGrafoView::OnDraw(CDC* pDC) Grafov.w.cpp\ (1220) void CGrafoView::OnRButtonDown(UINT nFlags, CPoint point) Hipger.cpp\ (890) extern "C" int FAR PASCAL _export muda_tipo(char *nome_no, char *tipo) Hipger.cpp\ (908) extern "C" int FAR PASCAL _export recupera_tipo(char *nome_no, char *tipo) Hipger.cpp\ (927) extern "C" int FAR PASCAL _export muda_funcao_didatica(char *nome_no, char *categoria) Hipger.cpp\ (945) extern "C" int FAR PASCAL _export recupera_funcao_didatica(char *nome_no, char *categoria) Hipger.cpp\ (1028) extern "C" int FAR PASCAL _export muda_palavra_chave(char *nome_no, int pos, char *palavra) Hipger.cpp\ (1046) extern "C" int FAR PASCAL _export recupera_palavra_chave(char *nome_no, int pos, char *palavra) Tutor.cpp\ (78) int tutor::segunda_estrategia(contexto *pai)

Sistema: SASHE Arquivo: Sashe\Hyperprop\Tiptop.cpp	
Procedimento: (20) CTipTop::CTipTop(CWnd* pParentWnd) Diretório/Arquivo: Tiptop.cpp	Descrição: Chama: (31) CreateSolidBrush(RGB(255, 255, 128)) (32) AfxRegisterWndClass(CS_SAVEBITS, 0, hBrush, 0) (33) ::DeleteObject(hBrush) (36) CreateEx(0, m_strWndClass, NULL, WS_POPUP WS_CLIPSIBLINGS WS_BORDER, 0, 0, 0, 0, m_pParentWnd->m_hWnd, 0) Chamado por: Grafov.w.cpp\ (114) CGrafoView::Create(LPCSTR lpszClassName, LPCSTR lpszWindowName, DWORD dwStyle, const RECT& rect, CWnd* pParentWnd, UINT nID, CCreateContext* pContext)
Procedimento: (39) CTipTop::~CTipTop() Diretório/Arquivo: Tiptop.cpp	Descrição: Chama: (44) m_pFont->DeleteObject() Chamado por:
Procedimento: (60) int CTipTop::OnCreate(LPCREATESTRUCT lpCreateStruct) Diretório/Arquivo: Tiptop.cpp	Descrição: Chama: (62) CWnd::OnCreate(lpCreateStruct)

Procedimento: (92) void CTipTop::OnPaint() Diretório/Arquivo: Tiptop.cpp	Chamado por: Descrição: Chama: (94) CPaintDC dc(this) (97) dc.SelectObject(m_pFont) (101) GetClientRect(&rect) (104) dc.SetBkMode(TRANSPARENT) (105) dc.SetTextColor(RGB(0, 0, 0)) (106) dc.DrawText(m_strText, m_strText.GetLength(), &rect, DT_CENTER DT_SINGLELINE DT_VCENTER) (109) dc.SelectObject(pOldFont) Chamado por:
Procedimento: (112) BOOL CTipTop::SetFont(const LOGFONT* pLogFont) Diretório/Arquivo: Tiptop.cpp	Descrição: Chama: (118) m_pFont->DeleteObject() (122) m_pFont-> >CreateFontIndirect(pLogFont) Chamado por: Tiptop.cpp\ (60) int CTipTop::OnCreate(LPCREATESTRUCT lpCreateStruct)
Procedimento: (128) void CTipTop::ShowTip(UINT nID, UINT x, UINT y, char *szItemName) Diretório/Arquivo: Tiptop.cpp	Descrição: Chama: (142) CClientDC dc(this) (143) dc.SelectObject(m_pFont) (144) dc.GetTextExtent(szItemName, lstrlen(szItemName)) (147) SetWindowPos(&wndNoTopMost, x, y, size.cx + (2 * m_nClearance), size.cy + (2 * m_nClearance), SWP_DRAWFRAME SWP_NOACTIVATE SWP_SHOWWINDOW) (150) dc.SelectObject(pOldFont) (153) InvalidateRect(NULL) (154) UpdateWindow() Chamado por: Grafov.w.cpp\ (1465) void CGrafoView::OnTimer(UINT nIDEvent)
Procedimento: (157) void CTipTop::HideCurrentTip() Diretório/Arquivo: Tiptop.cpp	Descrição: Hide the current tip (if any) Chama: (159) IsWindowVisible() (161) ShowWindow(SW_HIDE) Chamado por: Grafov.w.cpp\ (1098) void CGrafoView::OnMouseMove(UINT nFlags, CPoint point) Grafov.w.cpp\ (1140) void CGrafoView::OnLButtonDown(UINT nFlags, CPoint point) Grafov.w.cpp\ (1210) void CGrafoView::OnLButtonUp(UINT nFlags, CPoint point)

Sistema: SASHE**Arquivo:** Sashe\Hyperprop\ Trilha.cpp

Procedimento: (16) trilha::trilha(char *nome_no) : composicao(nome_no) Diretório/Arquivo: Trilha.cpp	Descrição: Construtor usado na criacao Chama:
---	---

	<pre> (16) composicao(nome_no) Chamado por: Hipger.cpp\ (174) extern "C" int FAR PASCAL _export cria_trilha(char *nome) No.cpp\ (142) int No::criaversao(char *nome no) </pre>
<pre> Procedimento: (27) trilha::trilha(int id) : composicao(id) Diretório/Arquivo: Trilha.cpp </pre>	<pre> Descrição: Construtor usado na recuperacao Chama: (27) composicao(id) Chamado por: Basepriv.cpp\ (133) int baseprivada::recupera(FILE *fp) </pre>
<pre> Procedimento: (38) int trilha::cria_ancora_lambda(int ide) Diretório/Arquivo: Trilha.cpp </pre>	<pre> Descrição: Chama: (40) ancora_trilha(retorna_indice(), ide) (42) insere_ancora(panc) Chamado por: Hipger.cpp\ (174) extern "C" int FAR PASCAL _export cria_trilha(char *nome) Trilha.cpp\ (48) void trilha::copia_trilha(trilha *nc) </pre>
<pre> Procedimento: (48) void trilha::copia_trilha(trilha *nc) Diretório/Arquivo: Trilha.cpp </pre>	<pre> Descrição: Faz uma copia para um no vazio Chama: (50) copia_nos(&nc->nos) (51) cria_ancora_lambda(0) Chamado por: No.cpp\ (142) int No::criaversao(char *nome no) </pre>
<pre> Procedimento: (62) int trilha::insere_no(No *pno, contexto *pno_liberdade) Diretório/Arquivo: Trilha.cpp </pre>	<pre> Descrição: Insere o nó liberdade no final da lista do nó trilha, cria uma âncora no nó contido no contexto e cria o elo entre o nó anterior (se existir) e o nó. Retorna 0 se conseguiu inserir Retorna 1 se falta memoria para a lista Retorna 2 se o nó trilha nao for temporario Chama: (66) ObtemEstado() (67) pno_liberdade- >cria_ancora_contexto(-1, pno- >obtem_identificador()) (69) pno_liberdade- >obtem_identificador() (70) nos.num_elementos() (72) obtem_identificador() (73) elo(ident, this, aninha1, 0, pno_liberdade, aninha2, panc- >obtem_identificador(), 0) (75) elos.insere(pelo) (80) ultimo->obtem_identificador() (81) elo(ident, ultimo, aninha1, 0, pno_liberdade, aninha2, panc- >obtem_identificador(), 0) (83) elos.insere(pelo) (86) pno_liberdade- >insere_contexto(this) (88) nos.insere(pno_liberdade) Chamado por: Hipger.cpp\ (288) extern "C" int FAR </pre>

	PASCAL_export insere_na_trilha(char *nome_trilha, char *nome_no, char *no liberdade)
Procedimento: (97) int trilha::insere_no(No *pno) Diretório/Arquivo: Trilha.cpp	Descrição: Insere o nó no final da lista do nó trilha Retorna 0 se conseguiu inserir Retorna 1 se falta memoria para a lista Retorna 2 se o nó trilha nao for temporario Chama: (101) ObtemEstado() (102) pno->obtem_identificador() (103) nos.num_elementos() (105) obtem_identificador() (106) elo(ident, this, aninha1, 0, pno, aninha2, 0, 0) (108) elos.insere(pelo) (113) ultimo->obtem_identificador() (114) elo(ident, ultimo, aninha1, 0, pno, aninha2, 0, 0) (116) elos.insere(pelo) (119) pno->insere_contexto(this) (121) nos.insere(pno) Chamado por: Hipger.cpp\ (288) extern "C" int FAR PASCAL_export insere_na_trilha(char *nome_trilha, char *nome_no, char *no liberdade)
Procedimento: (131) int trilha::insere_no(No *pno, contexto *pno_liberdade, int posicao) Diretório/Arquivo: Trilha.cpp	Descrição: Insere o nó na posição determinada. Retorna 0 se conseguiu inserir Retorna 1 se houve problema na alocação de memória Retorna 2 se a posição não é válida Retorna 3 se o nó trilha nao for temporario Chama: (137) nos.num_elementos() (138) ObtemEstado() (139) elos.primeiro() (143) elos.proximo() (146) pno_liberdade- >obtem_identificador() (147) elo(ident, pno, aninha1, 0, elo1- >no_destino, elo1->aninhamento_destino, elo1->ancora_no_destino, 0) (148) pno_liberdade- >cria_ancora_contexto(-1, pno- >obtem_identificador()) (149) elos.insere(elo1, elo2) (151) panc->obtem_identificador() (152) pno->obtem_identificador() (153) pno->insere_contexto(this) (155) nos.insere(pno_liberdade) Chamado por: Hipger.cpp\ (317) extern "C" int FAR PASCAL_export insere_na_posicao(char *nome_trilha, char *nome_no, char *no liberdade, int posicao)
Procedimento: (164) int trilha::insere_no(No *pno, int posicao) Diretório/Arquivo: Trilha.cpp	Descrição: Insere o nó na posição determinada Retorna 0 se conseguiu inserir Retorna 1 se falta memoria para a lista Retorna 2 se o nó trilha nao for temporario Chama:

	<pre> (169) nos.num_elementos() (170) ObtemEstado() (171) elos.primeiro() (175) elos.proximo() (178) pno->obtem_identificador() (179) elo(ident, pno, aninhal, 0, elo1->no_destino, elo1->aninhamento_destino, elo1->ancora_no_destino, 0) (180) elos.insere(elo1, elo2) (183) pno->obtem_identificador() (184) pno->insere_contexto(this) (186) nos.insere(pno) Chamado por: </pre>
Procedimento: <pre> (197) int trilha::retira_no(No *pno) </pre> Diretório/Arquivo: Trilha.cpp	Descrição: Retira o nó da trilha Retorna 0 se conseguiu retirar o nó da lista Retorna 1 se a lista estiver vazia Retorna 2 se não encontrou o nó na lista do nó trilha Retorna 3 se não encontrou o nó trilha na lista do nó contido Retorna 4 se o nó trilha nao for temporario Chama: <pre> (200) ObtemEstado() (201) nos.num_elementos() (202) elos.primeiro() (202) elos.proximo() (205) elos.primeiro() (205) elos.proximo() (213) elos.retira(elo2) (218) elos.retira(elo1) (222) pno->retira_contexto(this) (225) nos.retira(pno) Chamado por: Hipger.cpp\ (359) extern "C" int FAR PASCAL _export retira_da_trilha(char *nome_trilha, char *nome_no_liberdade) Hipger.cpp\ (731) extern "C" int FAR PASCAL _export retira_da_hiperbase(char *nome_no) </pre>
Procedimento: <pre> (232) void trilha::limpa_trilha() </pre> Diretório/Arquivo: Trilha.cpp	Descrição: Retira todos os nós e elos da trilha Chama: <pre> (234) elos.primeiro() (237) elos.retira(pelo) (238) elos.primeiro() (240) nos.primeiro() (243) nos.retira(no) (244) no->retira_contexto(this) (245) nos.primeiro() Chamado por: Hipger.cpp\ (376) extern "C" int FAR PASCAL _export limpa_trilha(char *nome_trilha) </pre>
Procedimento: <pre> (251) int trilha::primeiro_no_trilha(char *nome, char *nome_liberdade) </pre> Diretório/Arquivo: Trilha.cpp	Descrição: Retorna o primeiro nó que pertence ao nó trilha Chama: <pre> (253) elos.primeiro() (256) proximo->obtem_no_destino() (257) atual->eh_contexto() (259) atual->retorna_ancora(proximo->obtem_id_ancora_destino()) (261) hb.RetornaNo(panc->obtem_ancora()) </pre>

	<pre> (264) atual->ObtemNome() (265) temp->ObtemNome() (269) atual->ObtemNome() (270) atual->ObtemNome() Chamado por: Hipger.cpp\ (472) extern "C" int FAR PASCAL _export primeiro_no_trilha(char *nome_trilha, char *nome_no, char *nome_liberdade) </pre>
Procedimento: <pre> (278) int trilha::proximo_no_trilha(char *nome, char *nome_liberdade) Diretório/Arquivo: Trilha.cpp </pre>	Descrição: Retorna o próximo nó que pertence ao nó trilha Chama: <pre> (280) elos.proximo() (283) proximo->obtem_no_destino() (284) atual->eh_contexto() (286) atual->retorna_ancora(proximo- >obtem_id_ancora_destino()) (288) hb.RetornaNo(panc->obtem_ancora()) (291) atual->ObtemNome() (292) temp->ObtemNome() (296) atual->ObtemNome() (297) atual->ObtemNome() Chamado por: Hipger.cpp\ (483) extern "C" int FAR PASCAL _export proximo_no_trilha(char *nome_trilha, char *nome_no, char *nome_liberdade) Tutor.cpp\ (78) int tutor::segunda estrategia(contexto *pai) </pre>
Procedimento: <pre> (305) int trilha::anterior_no_trilha(char *nome, char *nome_liberdade) Diretório/Arquivo: Trilha.cpp </pre>	Descrição: Retorna o nó anterior que pertence ao nó trilha Chama: <pre> (307) elos.anterior() (310) anterior->obtem_no_destino() (311) atual->eh_contexto() (313) atual->retorna_ancora(anterior- >obtem_id_ancora_destino()) (315) hb.RetornaNo(panc->obtem_ancora()) (318) atual->ObtemNome() (319) temp->ObtemNome() (323) atual->ObtemNome() (324) atual->ObtemNome() Chamado por: Hipger.cpp\ (505) extern "C" int FAR PASCAL _export anterior_no_trilha(char *nome_trilha, char *nome_no, char *nome_liberdade) Tutor.cpp\ (78) int tutor::segunda estrategia(contexto *pai) </pre>
Procedimento: <pre> (332) int trilha::ultimo_no_trilha(char *nome, char *nome_liberdade) Diretório/Arquivo: Trilha.cpp </pre>	Descrição: Retorna o último nó que pertence ao nó trilha Chama: <pre> (334) elos.ultimo() (337) ultimo->obtem_no_destino() (338) atual->eh_contexto() (340) atual->retorna_ancora(ultimo- >obtem_id_ancora_destino()) (342) hb.RetornaNo(panc->obtem_ancora()) (345) atual->ObtemNome() (346) temp->ObtemNome() (350) atual->ObtemNome() </pre>

	<pre>(351) temp->ObtemNome() Chamado por: Hipger.cpp\ (494) extern "C" int FAR PASCAL _export ultimo_no_trilha(char *nome_trilha, char *nome_no, char *nome_liberdade)</pre>
Procedimento: <pre>(362) int trilha::grava(FILE *fp)</pre> Diretório/Arquivo: Trilha.cpp	Descrição: Grava a trilha em arquivo 1: Erro de escrita no arquivo 0: Ok Chama: <pre>(364) fprintf(fp, "Trilha){\n") (365) composicao::grava(fp)</pre> Chamado por:
Procedimento: <pre>(374) int trilha::recupera(FILE *fp)</pre> Diretório/Arquivo: Trilha.cpp	Descrição: Recupera a trilha do arquivo 1: Erro de escrita no arquivo 0: Ok Chama: <pre>(376) composicao::recupera(fp)</pre> Chamado por:
Procedimento: <pre>(385) int trilha::recupera_no(FILE *fp)</pre> Diretório/Arquivo: Trilha.cpp	Descrição: Retorna: 1: Erro de leitura 0: Ok. Chama: <pre>(387) composicao::recupera_no(fp) (388) elos.primeiro() (391) e->obtem_no_origem() (391) e->obtem_no_destino() (392) e->obtem_no_destino() (394) elos.proximo()</pre> Chamado por:

Sistema: SASHE**Arquivo:** Sashe\Hyperprop\ Trilha.h

Procedimento: <pre>(35) int eh_trilha()</pre> Diretório/Arquivo: Trilha.h	Descrição: Chama: Chamado por: Grafov.cpp\ (542) d CGrafoView::DesenhaNo(No *no) Hip.cpp\ (177) void hipwindow::atualiza_base_ativa(No *pno) Hipger.cpp\ (288) extern "C" int FAR PASCAL _export insere_na_trilha(char *nome_trilha, char *nome_no, char *no_liberdade) Hipger.cpp\ (317) extern "C" int FAR PASCAL _export insere_na_posicao(char *nome_trilha, char *nome_no, char *no_liberdade, int posicao) Hipger.cpp\ (359) extern "C" int FAR PASCAL _export retira_da_trilha(char *nome_trilha, char *nome_no_liberdade) Hipger.cpp\ (376) extern "C" int FAR PASCAL _export limpa_trilha(char *nome_trilha) Hipger.cpp\ (472) extern "C" int FAR PASCAL _export primeiro_no_trilha(char *nome_trilha, char *nome_no, char *nome_liberdade) Hipger.cpp\ (483) extern "C" int FAR PASCAL _export proximo_no_trilha(char *nome_trilha, char *nome_no, char *nome_liberdade) Hipger.cpp\ (494) extern "C" int FAR
--	---

```

PASCAL _export ultimo_no_trilha(char
*nome_trilha, char *nome_no, char
*nome_liberdade)
Hipger.cpp\ (505) extern "C" int FAR
PASCAL _export anterior_no_trilha(char
*nome_trilha, char *nome_no, char
*nome_liberdade)
Hipger.cpp\ (516) No *FAR PASCAL _export
primeiro_ponteiro_no(char *nome_cont)
Hipger.cpp\ (533) No *FAR PASCAL _export
proximo_ponteiro_no(char *nome_cont)
Hipger.cpp\ (712) extern "C" int FAR
PASCAL _export tipo_no(char *nome)
Hipger.cpp\ (731) extern "C" int FAR
PASCAL _export retira_da_hiperbase(char
*nome_no)
Hipger.cpp\ (964) extern "C" int FAR
PASCAL _export muda_assunto(char
*nome_no, char *assunto)
Hipger.cpp\ (1067) extern "C" int FAR
PASCAL _export regras_tutoriais(char
*no_atual, char *escopo, char
*historico, int dificuldade, char
*no_result, char *lib_result, int
*revisto)
Hipger.cpp\ (1697) elo *FAR PASCAL
_export primeiro_ponteiro_elo(char
*nome_contexto)
Hipger.cpp\ (1714) elo *FAR PASCAL
_export proximo_ponteiro_elo(char
*nome_contexto, elo *pelo)
List_anc.cpp\ (118) int lista_ancoras ::
recupera(No *n, FILE *fp)
List_no.cpp\ (98) lista_nos ::
~lista_nos()
No.cpp\ (142) int No::criaversao(char
*nome_no)

```

Sistema: SASHE

Arquivo: Sashe\Hyperprop\ Tutor.cpp

Procedimento:

(19) tutor::tutor()

Diretório/Arquivo: Tutor.cpp

Descrição:

Construtor da classe tutor

Chama:

Chamado por:

Procedimento:

(39) int

tutor::primeira_estrategia(contexto
*pai)

Descrição:

Dado o escopo (contexto no qual o nó atual pertence) e o nó atual (nó visitado no momento em que o estudante apertou um dos botões difícil ou fácil) este método procura por outros nós no escopo que contenham um nível de dificuldade menor (apertou o botão difícil) ou um nível de dificuldade maior (apertou o botão fácil).

Retorna:

1: se encontrou o nó desejado e ele não pertence ao histórico

2: se encontrou o nó desejado e ele pertence ao histórico

0: caso contrário.

Se retornar 1, o nó desejado será apontado pelo nó atual (no_atual).

Chama:

(41) pai->nos.primeiro()

(47) no->recupera_dificuldade()

(47) no_atual->recupera_dificuldade()

(48) no_atual->recupera_dificuldade()

Diretório/Arquivo: Tutor.cpp

	<pre> (50) historico->contem(no) (55) no->recupera_dificuldade() (55) no_atual->recupera_dificuldade() (55) no_atual->recupera_dificuldade() (58) historico->contem(no) (63) no = pai->nos.proximo() Chamado por: Hipger.cpp\ (1067) extern "C" int FAR PASCAL _export regras_tutoriais(char *no_atual, char *escopo, char *historico, int dificuldade, char *no_result, char *lib_result, int *revisto) </pre>										
Procedimento: <pre> (78) int tutor::segunda_estrategia(contexto *pai) </pre> Diretório/Arquivo: Tutor.cpp	Descrição: Este método é recursivo. Ele procura o nó correspondente a dificuldade desejada no nó atual se não encontrar aplica a estratégia "funcao didatica". Essa estratégia corresponde em fazer buscas através das relações definidas em uma tabela. Retorna: 1: se encontrou o nó desejado. 0: caso contrário. Se o resultado for 1, o nó desejado será o nó atual (no_atual). Chama: <pre> (80) no_atual->eh_terminal() (84) no_atual)- >recupera_funcao_didatica() (89) tabela_dificil(origem, destino) (94) hb.Retorna(destino) (95) pontno->anterior_no_trilha(no, cont) (97) hb.Retorna(no) (98) hb.Retorna(cont) (104) pai->nos.primeiro() (107) no->eh_terminal() (109) no->recupera_funcao_didatica() (115) pai->nos.proximo() (123) tabela_facil(origem, destino) (128) hb.Retorna(destino) (129) pontno->proximo_no_trilha(no, cont) (131) hb.Retorna(no) (132) hb.Retorna(cont) (138) pai->nos.primeiro() (141) no->eh_terminal() (143) no->recupera_funcao_didatica() (149) pai->nos.proximo() Chamado por: Hipger.cpp\ (1067) extern "C" int FAR PASCAL _export regras_tutoriais(char *no_atual, char *escopo, char *historico, int dificuldade, char *no_result, char *lib_result, int *revisto) </pre>										
Procedimento: <pre> (176) int tutor::tabela_dificil(char *origem, char *destino) </pre>	Descrição: A tabela contém a "funcao didatica" que deve ser aplicada pelo tutor. Quando o estudante aperta o botão difícil, a seguinte relação é válida: <table> <tr> <td>ORIGEM</td><td>DESTINO</td></tr> <tr> <td>INTRODUCAO</td><td>MOTIVACAO</td></tr> <tr> <td>EXERCICIO</td><td>EXEMPLO</td></tr> <tr> <td>MOTIVACAO</td><td>TOPICO ANTERIOR(BACK DA</td></tr> <tr> <td>TRILHA)</td><td></td></tr> </table>	ORIGEM	DESTINO	INTRODUCAO	MOTIVACAO	EXERCICIO	EXEMPLO	MOTIVACAO	TOPICO ANTERIOR(BACK DA	TRILHA)	
ORIGEM	DESTINO										
INTRODUCAO	MOTIVACAO										
EXERCICIO	EXEMPLO										
MOTIVACAO	TOPICO ANTERIOR(BACK DA										
TRILHA)											

	DEFINICAO	INTRODUCAO														
	EXEMPLO	DEFINICAO														
	RESUMO	INTRODUCAO														
	Retorna: 1: se a relacao foi aplicada corretamente 0: caso contrário. Se retornar 1, o nó desejado será apontado pelo nó atual (no_atual). Chama: Chamado por: Tutor.cpp\ (78) int tutor::segunda_estrategia(contexto *pai)															
Diretório/Arquivo: Tutor.cpp																
Procedimento: (241) int tutor::tabela_facil(char *origem, char *destino)	Descrição: A tabela contem a "funcao didatica" que deve ser aplicada pelo tutor. Quando o estudante aperta o botao facil, a seguinte relacao eh valida: <table><tr><td>ORIGEM</td><td>DESTINO</td></tr><tr><td>INTRODUCAO</td><td>DEFINICAO</td></tr><tr><td>EXERCICIO</td><td>RESUMO</td></tr><tr><td>MOTIVACAO</td><td>DEFINICAO</td></tr><tr><td>DEFINICAO</td><td>EXEMPLO</td></tr><tr><td>EXEMPLO</td><td>EXERCICIO</td></tr><tr><td>RESUMO</td><td>PROXIMO TOPICO (NEXT DA TRILHA)</td></tr></table> Retorna: 1: se a relacao foi aplicada corretamente 0: caso contrário. Se retornar 1, o nó desejado será apontado pelo nó atual (no_atual). Chama: Chamado por: Tutor.cpp\ (78) int tutor::segunda_estrategia(contexto *pai)		ORIGEM	DESTINO	INTRODUCAO	DEFINICAO	EXERCICIO	RESUMO	MOTIVACAO	DEFINICAO	DEFINICAO	EXEMPLO	EXEMPLO	EXERCICIO	RESUMO	PROXIMO TOPICO (NEXT DA TRILHA)
ORIGEM	DESTINO															
INTRODUCAO	DEFINICAO															
EXERCICIO	RESUMO															
MOTIVACAO	DEFINICAO															
DEFINICAO	EXEMPLO															
EXEMPLO	EXERCICIO															
RESUMO	PROXIMO TOPICO (NEXT DA TRILHA)															
Diretório/Arquivo: Tutor.cpp																

Sistema: SASHE	
Arquivo: Sashe\Hyperprop\ Tutor.h	
Procedimento: (32) void muda_no_atual(No *no)	Descrição:
Diretório/Arquivo: Tutor.h	Chama: Chamado por: Hipger.cpp\ (1067) extern "C" int FAR PASCAL _export regras_tutoriais(char *no_atual, char *escopo, char *historico, int dificuldade, char *no_result, char *lib_result, int *revisto)
Procedimento: (33) void muda_historico(trilha *no)	Descrição:
Diretório/Arquivo: Tutor.h	Chama: Chamado por: Hipger.cpp\ (1067) extern "C" int FAR PASCAL _export regras_tutoriais(char *no_atual, char *escopo, char *historico, int dificuldade, char *no_result, char *lib_result, int *revisto)
Procedimento: (34) void muda_nivel(Botoes_Tutor _nivel)	Descrição:
Diretório/Arquivo: Tutor.h	Chama: Chamado por: Hipger.cpp\ (1067) extern "C" int FAR

	PASCAL_export regras_tutoriais(char *no_atual, char *escopo, char *historico, int dificuldade, char *no_result, char *lib_result, int *revisto)
Procedimento: (35) No * recupera_no_atual() Diretório/Arquivo: Tutor.h	Descrição: Chama: Chamado por: Hipger.cpp\ (1067) extern "C" int FAR PASCAL_export regras_tutoriais(char *no_atual, char *escopo, char *historico, int dificuldade, char *no_result, char *lib_result, int *revisto)
Procedimento: (36) BOOL jah_foi_visto() Diretório/Arquivo: Tutor.h	Descrição: Chama: Chamado por: Hipger.cpp\ (1067) extern "C" int FAR PASCAL_export regras_tutoriais(char *no_atual, char *escopo, char *historico, int dificuldade, char *no_result, char *lib_result, int *revisto)

Sistema: SASHE	
Arquivo: Sashe\Hyperprop\ Video.h	
Procedimento: (14) CAVi () Diretório/Arquivo: Video.h	Descrição: Chama: Chamado por:
Procedimento: (15) ~CAVi () Diretório/Arquivo: Video.h	Descrição: Chama: Chamado por:
Procedimento: (29) BOOL Open (CWnd *, CString) Diretório/Arquivo: Video.h	Descrição: Chama: Chamado por: Grafov.cpp\ (1220) void CGrafoView::OnRButtonDown(UINT nFlags, CPoint point)
Procedimento: (30) void Close (void) Diretório/Arquivo: Video.h	Descrição: Chama: Chamado por: Grafov.cpp\ (996) void CGrafoView::OnAtualizar()
Procedimento: (31) BOOL Start (void) Diretório/Arquivo: Video.h	Descrição: Chama: Chamado por: Grafov.cpp\ (1220) void CGrafoView::OnRButtonDown(UINT nFlags, CPoint point)
Procedimento: (32) BOOL Stop (void) Diretório/Arquivo: Video.h	Descrição: Chama: Chamado por:
Procedimento: (33) BOOL Rewind (void) Diretório/Arquivo: Video.h	Descrição: Chama: Chamado por:
Procedimento: (34) BOOL Forward (void)	Descrição:

Diretório/Arquivo: Video.h	Chama: Chamado por:
Procedimento: (35) BOOL FrameForward (void) Diretório/Arquivo: Video.h	Descrição: Chama: Chamado por:
Procedimento: (36) BOOL FrameBack (void) Diretório/Arquivo: Video.h	Descrição: Chama: Chamado por:
Procedimento: (37) int GetWidth (void) Diretório/Arquivo: Video.h	Descrição: Chama: Chamado por: Grafov.w.cpp\ (1220) void CGrafoView::OnRButtonDown(UINT nFlags, CPoint point)
Procedimento: (38) int GetHeight (void) Diretório/Arquivo: Video.h	Descrição: Chama: Chamado por: Grafov.w.cpp\ (1220) void CGrafoView::OnRButtonDown(UINT nFlags, CPoint point)
Procedimento: (39) CString GetErrorString (void) Diretório/Arquivo: Video.h	Descrição: Chama: Chamado por:
Procedimento: (41) BOOL SetPosition (void) Diretório/Arquivo: Video.h	Descrição: Chama: Chamado por:

Sistema: SASHE	
Arquivo: sashe\ ajuda.pas	
Procedimento: (39) To_fo_Ajuda.FormCreate(Sender: TObject); Diretório/Arquivo: ajuda.pas	Descrição: Construtor da janela de construção de nós de ajuda (Dúvidas e Respostas) Chama: Chamado por:
Procedimento: (50) To_fo_Ajuda.o_bu_InserirClick(Sender: TObject); Diretório/Arquivo: ajuda.pas	Descrição: Adiciona as strings "[Dúvida]" e "[Resposta]" e também os pontos de separação ao texto da janela quando é acionado o botão inserir Chama: (53) o_me_Duvidas.Lines.Add('[Dúvida]') (54) o_me_Duvidas.Lines.Add('') (55) o_me_Duvidas.Lines.Add('::: '::) (58) o_me_Respostas.Lines.Add('[Resposta]') (59) o_me_Respostas.Lines.Add('') (60) o_me_Respostas.Lines.Add('::: '::) Chamado por:
Procedimento: (67) To_fo_Ajuda.o_bu_GravarClick(Sender: TObject); Diretório/Arquivo: ajuda.pas	Descrição: Grava o texto contido na janela de construção de nós de ajuda nos arquivos em variáveis tipo textfile Chama: (69) o_me_Duvidas.Lines.SaveToFile(a_Arquivo_Duvida) (70) o_me_Respostas.Lines.SaveToFile(a_Arquivo_Resposta) Chamado por:

Sistema: SASHE Arquivo: sashe\atexto.pas	
Procedimento: (52) To_fo_Texto.Procurar_Elo(Ancora: string; NomeDoNoLiberdade: string): boolean;	Descrição: Procura pelo Elo que contém a Âncora especificada. Percorre todos os contextos que estão acima do nó atual (a_Nome_No) tentando encontrar um elo que contenha como ancora de origem o valor do parametro passado (Ancora). Chama: (67) primeiro_contexto_pai(a_Nome_No, ContextoPai) (71) numero_de_elos(ContextoPai) (73) recupera_elos(ContextoPai, I, Lista_Origem, Ident_Anc_Origem, Lista_Destino, Ident_Anc_Destino, ApresentarNoSimultaneo, ApresentarNosTerminais) (87) retorna_nome(StrToInt(StrPas(Ident_No_Origem)), Nome_No_Origem) (88) recupera_nome_ancora(Nome_No_Origem, Ident_Anc_Origem, Nome_Ancora) (106) primeiro_contexto_pai(ContextoPai, ContextoPai) Chamado por: atexto.pas\ (176) To_fo_Texto.o_hv_TextoHotSpotClick(Sender: TObject; const SRC: String; var Handled: Boolean)
Procedimento: (116) To_fo_Texto.Ler_Resposta(Numero_Duvida: integer);	Descrição: Le a resposta para a dúvida pressionada no arquivo de ajuda. É chamada somente quando for apresentado um nó de ajuda. Chama: (123) recupera_nome_arquivo_texto(a_Nome_No, Arquivo) (141) Txt_Html(ArqResp) (144) o_fo_Resposta.o_hv_Resposta.LoadFromFile(ArqResp){declarado onde?} Chamado por: atexto.pas\ (176) To_fo_Texto.o_hv_TextoHotSpotClick(Sender: TObject; const SRC: String; var Handled: Boolean)
Procedimento: (158) To_fo_Texto.FormCreate(Sender: TObject);	Descrição: Construtor da janela de apresentação e navegação de um nó texto Chama: Chamado por:
Procedimento: (167) To_fo_Texto.FormClose(Sender: TObject; var Action: TCloseAction);	Descrição: Destrutor da janela de apresentação e navegação de um nó texto Chama: Chamado por:
Procedimento: (176) To_fo_Texto.o_hv_TextoHotSpotClick(Sender: TObject; const SRC: String; var Handled: Boolean);	Descrição: Implementa a navegação, verificando contextos, âncoras e elos



Diretório/Arquivo: atexto.pas	Chama: (187) recupera_funcao_didatica(a_Nome_No, FuncaoDidatica) (194) Ler_Resposta(StrToInt(SRC)) (200) recupera_nome_ancora(a_Nome_No, Numero_Ancora, Nome_Anc_Origem) (202) Procurar_Elo(StrPas(Nome_Anc_Origem), g_NoLiberdadeDoRoteiro) (204) Procurar_Elo(StrPas(Nome_Anc_Origem), g_NomeDaHiperbase) (209) recupera_elo(StrPascalC(_Nome_No_Procura do), _Ident_Elo_Procurado, Lista_Origem, Ident_Anc_Origem, Lista_Destino, Ident_Anc_Destino, ApresentarNoSimultaneo, ApresentarNosTerminais) (209) StrPascalC(_Nome_No_Procurado) (224) retorna_nome(StrToInt(StrPas(Ident_No_De stino)), Nome_No_Destino) (225) recupera_nome_ancora(Nome_No_Destino, Ident_Anc_Destino, Nome_Anc_Destino) (241) Apresentar_No(Nome_No_Destino, ApresentarNosTerminais) (242) g_TrilhaDoEstudante.Add(StrPas(Nome_No_D estino)) (243) g_TrilhaDosNosLiberdade.Add(g_NoLiberdad eDoRoteiro) (246) ativa_hiperbase(StrPascalC('HBE')) (246) StrPascalC('HBE') (247) insere_na_trilha(StrPascalC('Estudante') , Nome_No_Destino, StrPascalC(g_NoLiberdadeDoRoteiro)) (247) StrPascalC('Estudante') (248) StrPascalC(g_NoLiberdadeDoRoteiro)) (249) ativa_hiperbase(StrPascalC(g_NomeDaHiper base)) (249) StrPascalC(g_NomeDaHiperbase) (253) Apresentar_No(Nome_No_Destino, ApresentarNosTerminais) Chamado por:
-------------------------------	---

Sistema: SASHE	
Arquivo: sashe\ atrilha.pas	
Procedimento: (33) To_fo_Trilha.FormCreate(Sender: TObject); Diretório/Arquivo: atrilha.pas	Descrição: Cria a janela onde é apresentado a trilha Chama: Chamado por:
Procedimento: (42)To_fo_Trilha.FormClose(Sender: TObject; var Action: TCloseAction); Diretório/Arquivo: atrilha.pas	Descrição: Destroi a janela onde é apresentada a trilha Chama: Chamado por:

Sistema: SASHE Arquivo: sashe\bgrafo.pas	
Procedimento: (32) To_fo_BrowserGrafico.FormClose(Sender: TObject; var Action: TCloseAction); Diretório/Arquivo: bgrafo.pas	Descrição: Fecha a janela do Browser Gráfico e libera a memória Chama: Chamado por:
Procedimento: (41) To_fo_BrowserGrafico.FormResize(Sender: TObject); Diretório/Arquivo: bgrafo.pas	Descrição: Redesenha a janela do browser gráfico Chama: Chamado por:

Sistema: SASHE Arquivo: sashe\ browser.pas	
Procedimento: (47) To_fo_BrowserEstrutural.Ler_Contexto(Indice: LongInt; Contexto_Inicial: String); Diretório/Arquivo: browser.pas	Descrição: Faz a leitura dos nós pertencentes ao contexto atual e monta a árvore hierárquica do contexto {para apresentação dentro do browser estrutural} Chama: (53) primeiro_no_contexto(StrPascalC(Contexto_Inicial), Nome) (55) o_ol_Browser.AddChild(Indice, StrPas(Nome)) (57) Ler_Contexto(Indice_Auxiliar, StrPas(Nome)) (58) proximo_no_contexto(StrPascalC(Contexto_Inicial), Nome) (60) o_ol_Browser.AddChild(Indice, StrPas(Nome)) (62) Ler_Contexto(Indice_Auxiliar, StrPas(Nome)) Chamado por:
Procedimento: (70) To_fo_BrowserEstrutural.Atualiza_Browser; Diretório/Arquivo: browser.pas	Descrição: Atualiza o browser estrutural, limpando o que havia, e chamando o procedimento Ler_Contexto com um novo contexto Chama: (72) o_ol_Browser.Clear (73) Ler_Contexto(0, a_Contexto_Browser) (74) o_ol_Browser.FullExpand Chamado por: Nos.pas\ (908) To_fo_Nos.o_pm_BrowserClick(Sender: TObject)
Procedimento: (80) To_fo_BrowserEstrutural.FormClose(Sender: TObject; var Action: TCloseAction); Diretório/Arquivo: browser.pas	Descrição: Fecha a janela do browser estrutural e libera a memória Chama: Chamado por:
Procedimento: (89) To_fo_BrowserEstrutural.o_bu_AbrirJanelaClick(Sender: TObject); Diretório/Arquivo: browser.pas	Descrição: Cria uma nova janela "Nós e Elos" com o contexto atual do browser estrutural Chama: (98) tipo_no(StrPascalC(Items[SelectedItem].Text))

	(112) o_fo_Principal.MDIChildren[Numero].Show Chamado por:
Procedimento: (123) To_fo_BrowserEstrutural.o_bu_Aprese ntarConteudoClick(Sender: TObject); Diretório/Arquivo: browser.pas	Descrição: Chama o procedimento para a visualização do conteúdo do nó selecionado Chama: (128) tipo_no(StrPascalC(Items[SelectedItem].T ext)) (130) o_fo_Nos.Visualizar_No_Terminal('t', StrPascalC(Items[SelectedItem].Text)) (131) o_fo_Nos.Visualizar_No_Terminal('v', StrPascalC(Items[SelectedItem].Text)) (132) o_fo_Nos.Visualizar_No_Terminal('a', StrPascalC(Items[SelectedItem].Text)) (133)o_fo_Nos.Visualizar_No_Terminal('i' , StrPascalC(Items[SelectedItem].Text)) (134) o_fo_Nos.ApresentarNoTrilha(StrPascalC(I tems[SelectedItem].Text)) Chamado por:
Procedimento: (142) To_fo_BrowserEstrutural.o_ti_Tempor izadorTimer(Sender: TObject); Diretório/Arquivo: browser.pas	Descrição: É um timer acionado a cada milésimo de segundo que interrompe o controle normal do aplicativo e executa o código especificado no corpo do procedimento Chama: (151)tipo_no(StrPascalC(Items[SelectedIt em].Text)) Chamado por:
Procedimento: (170) To_fo_BrowserEstrutural.o_ol_Browse rClick(Sender: TObject); Diretório/Arquivo: browser.pas	Descrição: Este procedimento seta os valores mostrados nas etiquetas que aparecem quando um nó é selecionado dentro do browser estrutural Chama: (178) recupera_funcao_didatica(StrPascalC(Items [SelectedItem].Text), v_sc_FuncaoDidatica) (179) recupera_dificuldade(StrPascalC(Items[Se lectedItem].Text), v_i_Dificuldade) (185)tipo_no(StrPascalC(Items[SelectedIt em].Text)) Chamado por:
Sistema: SASHE Arquivo: sashe\controle.pas	
Procedimento: (63) To_fo_Controles.o_bu_TopicoAnterior Click(Sender: TObject); Diretório/Arquivo: controle.pas	Descrição: Implementa a funcionalidade do botão Tópico Anterior dos controles de navegação do módulo estudante Chama: (68) anterior_no_Trilha(StrPascalC('Roteiro') , NomeDoNo, NomeDoNoLiberdade) (68) StrPascalC('Roteiro') (80) g_TrilhaDoEstudante.Add(StrPas(NomeDoNo)) (81) g_TrilhaDosNosLiberdade.Add(StrPas(NomeD

	<pre> oNoLiberdade)) (83) ativa_hiperbase(StrPascalC('HBE')) (83) StrPascalC('HBE') (84) insere_na_trilha (StrPascalC('Estudante'), NomeDoNo, NomeDoNoLiberdade) (84) StrPascalC('Estudante') (85) ativa_hiperbase (StrPascalC(g_NomeDaHiperbase)) (85) StrPascalC(g_NomeDaHiperbase) (86) o_fo_Principal.Apresentar_No (NomeDoNo, 1) Chamado por: </pre>
Procedimento: (107) To_fo_Controles.o_bu_ProximoTopicoClick (Sender: TObject); Diretório/Arquivo: controle.pas	Descrição: Implementa a funcionalidade do botão Próximo Tópico dos controles de navegação do módulo estudante Chama: <pre> (112) proximo_no_Trilha (StrPascalC('Roteiro'), NomeDoNo, NomeDoNoLiberdade) (112) StrPascalC('Roteiro') (124) g_TrilhaDoEstudante.Add (StrPas(NomeDoNo)) (125) g_TrilhaDosNosLiberdade.Add (StrPas(NomeDoNoLiberdade)) (127) ativa_hiperbase (StrPascalC('HBE')) (127) StrPascalC('HBE') (128) insere_na_trilha (StrPascalC('Estudante'), NomeDoNo, NomeDoNoLiberdade) (128) StrPascalC('Estudante') (129) ativa_hiperbase (StrPascalC(g_NomeDaHiperbase)) (129) StrPascalC(g_NomeDaHiperbase) (130) o_fo_Principal.Apresentar_No (NomeDoNo, 1) Chamado por: </pre>
Procedimento: (151) To_fo_Controles.o_bu_AnteriorClick (Sender: TObject); Diretório/Arquivo: controle.pas	Descrição: Implementa a funcionalidade do botão Histórico Anterior dos controles de navegação do módulo estudante Chama: <pre> (165) Apresentar_No (StrPascalC(g_NoDoRoteiro), 1) (165) StrPascalC(g_NoDoRoteiro) Chamado por: </pre>
Procedimento: (187) To_fo_Controles.o_bu_ProximoClick (Sender: TObject); Diretório/Arquivo: controle.pas	Descrição: Implementa a funcionalidade do botão Próximo Histórico dos controles de navegação do módulo estudante Chama: <pre> (201) Apresentar_No (StrPascalC(g_NoDoRoteiro), 1) (201) StrPascalC(g_NoDoRoteiro) Chamado por: </pre>
Procedimento: (223) To_fo_Controles.o_bu_ContextoClick (Sender: TObject); Diretório/Arquivo: controle.pas	Descrição: Implementa a funcionalidade do botão Contexto dos controles de navegação do módulo estudante Chama: <pre> (225) o_fo_Principal.Apresentar_No (StrPascalC(g_NoLiberdadeDoRoteiro), 0) (225) StrPascalC(g_NoLiberdadeDoRoteiro) Chamado por: </pre>
Procedimento:	Descrição:

(244) To_fo_Controles.o_bu_GlossarioClick (Sender: TObject); Diretório/Arquivo: controle.pas	Implementa a funcionalidade do botão Glossário dos controles de navegação do módulo estudante Chama: Chamado por:
Procedimento: (273) To_fo_Controles.o_bu_FacilClick (Sender: TObject); Diretório/Arquivo: controle.pas	Descrição: Implementa a funcionalidade do botão Está Fácil! dos controles de navegação do módulo estudante Chama: (278) regras_tutoriais (StrPascalC(g_NoDoRoteiro), StrPascalC(g_NoLiberdadeDoRoteiro), StrPascalC('Estudante'), 1, NoDestino, NoLiberdadeDestino, Revisto) (278) StrPascalC(g_NoDoRoteiro) (279) StrPascalC(g_NoLiberdadeDoRoteiro) (279) StrPascalC('Estudante') (285) g_TrilhaDoEstudante.Add (StrPas(NoDestino)) (286) g_TrilhaDosNosLiberdade.Add (StrPas(NoLiberdadeDestino)) (288) ativa_hiperbase (StrPascalC('HBE')) (288) StrPascalC('HBE') (289)insere_na_trilha (StrPascalC('Estudante'), NoDestino, NoLiberdadeDestino) (289) StrPascalC('Estudante') (290) ativa_hiperbase (StrPascalC(g_NomeDaHiperbase)) (290) StrPascalC(g_NomeDaHiperbase) (296)o_fo_Principal.Apresentar_No (NoDestino, 1) Chamado por:
Procedimento: (322) To_fo_Controles.o_bu_DifícilClick (Sender: TObject); Diretório/Arquivo: controle.pas	Descrição: Implementa a funcionalidade do botão Está Difícil! dos controles de navegação do módulo estudante Chama: (327) regras_tutoriais (StrPascalC(g_NoDoRoteiro), StrPascalC(g_NoLiberdadeDoRoteiro), StrPascalC('Estudante'), 2, NoDestino, NoLiberdadeDestino, Revisto) (327) StrPascalC(g_NoDoRoteiro) (328) StrPascalC(g_NoLiberdadeDoRoteiro) (328) StrPascalC('Estudante') (334) g_TrilhaDoEstudante.Add (StrPas(NoDestino)) (335) g_TrilhaDosNosLiberdade.Add (StrPas(NoLiberdadeDestino)) (337) ativa_hiperbase (StrPascalC('HBE')) (337) StrPascalC('HBE') (338) insere_na_trilha (StrPascalC('Estudante'), NoDestino, NoLiberdadeDestino) (338) StrPascalC('Estudante') (339) ativa_hiperbase(StrPascalC(g_NomeDaHiperbase)) (339) StrPascalC(g_NomeDaHiperbase) (345) o_fo_Principal.Apresentar_No (NoDestino, 1) Chamado por:

Procedimento: (371)To_fo_Controles.o_bu_AjudaClick (Sender: TObject); Diretório/Arquivo: controle.pas	Descrição: Implementa a funcionalidade do botão Ajuda dos controles de navegação do módulo estudante Chama: (378) primeiro_contexto_pai (StrPascalC(g_NoDoRoteiro), ContextoPai) (378) StrPascalC(g_NoDoRoteiro) (380)primeiro_no_contexto (ContextoPai, NomeDoNo) (382)recupera_funcao_didatica (NomeDoNo, FuncaoDidatica) (385) proximo_no_contexto (ContextoPai, NomeDoNo) (391) primeiro_contexto_pai (ContextoPai, ContextoPai) (395) o_fo_Principal.Apresentar_No (NomeDoNo, 0) Chamado por: Descrição: Implementa a funcionalidade do botão Bibliografia dos controles de navegação do módulo estudante Chama: (426) primeiro_contexto_pai (StrPascalC(g_NoDoRoteiro), ContextoPai) (426) StrPascalC(g_NoDoRoteiro) (428) primeiro_no_contexto (ContextoPai, NomeDoNo) (430) recupera_funcao_didatica (NomeDoNo, FuncaoDidatica) (433) proximo_no_contexto (ContextoPai, NomeDoNo) (439) primeiro_contexto_pai (ContextoPai, ContextoPai) (443) o_fo_Principal.Apresentar_No (NomeDoNo, 0) Chamado por: Descrição: Implementa a funcionalidade do botão Exercícios dos controles de navegação do módulo estudante Chama: (474) primeiro_contexto_pai (StrPascalC(g_NoDoRoteiro), ContextoPai) (474) StrPascalC(g_NoDoRoteiro) (476)primeiro_no_contexto (ContextoPai, NomeDoNo) (478) recupera_funcao_didatica (NomeDoNo, FuncaoDidatica) (481) proximo_no_contexto (ContextoPai, NomeDoNo) (487) primeiro_contexto_pai (ContextoPai, ContextoPai) (491) o_fo_Principal.Apresentar_No (NomeDoNo, 0) Chamado por: Descrição: Implementa a funcionalidade do botão Mais Informações! dos controles de navegação do módulo estudante Chama: (517)tipo_no (StrPascalC(g_NoLiberdadeDoRoteiro)) (517) StrPascalC(g_NoLiberdadeDoRoteiro) (518) tipo_no (StrPascalC(g_NoDoRoteiro)) (518) StrPascalC(g_NoDoRoteiro)
Procedimento: (419) To_fo_Controles.o_bu_BibliografiaClick (Sender: TObject); Diretório/Arquivo: controle.pas	Chamado por: Descrição: Implementa a funcionalidade do botão Exercícios dos controles de navegação do módulo estudante Chama: (426) primeiro_contexto_pai (StrPascalC(g_NoDoRoteiro), ContextoPai) (426) StrPascalC(g_NoDoRoteiro) (428) primeiro_no_contexto (ContextoPai, NomeDoNo) (430) recupera_funcao_didatica (NomeDoNo, FuncaoDidatica) (433) proximo_no_contexto (ContextoPai, NomeDoNo) (439) primeiro_contexto_pai (ContextoPai, ContextoPai) (443) o_fo_Principal.Apresentar_No (NomeDoNo, 0) Chamado por: Descrição: Implementa a funcionalidade do botão Exercícios dos controles de navegação do módulo estudante Chama: (474) primeiro_contexto_pai (StrPascalC(g_NoDoRoteiro), ContextoPai) (474) StrPascalC(g_NoDoRoteiro) (476)primeiro_no_contexto (ContextoPai, NomeDoNo) (478) recupera_funcao_didatica (NomeDoNo, FuncaoDidatica) (481) proximo_no_contexto (ContextoPai, NomeDoNo) (487) primeiro_contexto_pai (ContextoPai, ContextoPai) (491) o_fo_Principal.Apresentar_No (NomeDoNo, 0) Chamado por: Descrição: Implementa a funcionalidade do botão Mais Informações! dos controles de navegação do módulo estudante Chama: (517)tipo_no (StrPascalC(g_NoLiberdadeDoRoteiro)) (517) StrPascalC(g_NoLiberdadeDoRoteiro) (518) tipo_no (StrPascalC(g_NoDoRoteiro)) (518) StrPascalC(g_NoDoRoteiro)
Procedimento: (515)To_fo_Controles.o_bu_MaisInformacoesClick(Sender: TObject); Diretório/Arquivo: controle.pas	Chamado por: Descrição: Implementa a funcionalidade do botão Mais Informações! dos controles de navegação do módulo estudante Chama: (517)tipo_no (StrPascalC(g_NoLiberdadeDoRoteiro)) (517) StrPascalC(g_NoLiberdadeDoRoteiro) (518) tipo_no (StrPascalC(g_NoDoRoteiro)) (518) StrPascalC(g_NoDoRoteiro)

Procedimento: (546) To_fo_Controles.o_bu_OndeEstouClick (Sender: TObject); Diretório/Arquivo: controle.pas	(525) Ler_Contexto(g_NoLiberdadeDoRoteiro) Chamado por: Descrição: Implementa a funcionalidade do botão Onde Estou? dos controles de navegação do módulo estudante Chama: (552) primeiro_no_Trilha(StrPascalC('Roteiro')) , NomeDoNo, NomeDoNoLiberdade) (552) StrPascalC('Roteiro') (556) tipo_no (NomeDoNo) (565) proximo_no_Trilha (StrPascalC('Roteiro'), NomeDoNo, NomeDoNoLiberdade) (570) primeiro_no_Trilha(StrPascalC('Roteiro')) , NomeDoNo, NomeDoNoLiberdade) (570) StrPascalC('Roteiro') (572) proximo_no_Trilha(StrPascalC('Roteiro'), NomeDoNo, NomeDoNoLiberdade) (572) StrPascalC('Roteiro') Chamado por: Descrição: Implementa a funcionalidade do botão Histórico dos controles de navegação do módulo estudante Chama: (598) ativa_hiperbase (StrPascalC('HBE')) (598) StrPascalC('HBE') (599) primeiro_no_Trilha(StrPascalC('Estu dante'), NomeDoNo, NomeDoNoLiberdade) (599) StrPascalC('Estudante') (603) tipo_no (NomeDoNo) (612) proximo_no_Trilha (StrPascalC('Estudante'), NomeDoNo, NomeDoNoLiberdade) (612) StrPascalC('Estudante') (617) ativa_hiperbase(StrPascalC(g_NomeDaHiper base)) (617) StrPascalC(g_NomeDaHiperbase) Chamado por:
---	---

Sistema: SASHE**Arquivo:** sashe\criar_no.pas

Procedimento: (62) To_fo_CriarNo.FormCreate(Sender: TObject) Diretório/Arquivo: criar_no.pas	Descrição: Construtor da janela para criação de um novo nó Chama: (67) primeiro_no_hiperbase(NomeDoNo) (68) proximo_no_hiperbase(NomeDoNo) (69) o_cb_NomeDoNo.Items.Add(StrPas(NomeDoNo)) (69) StrPas(NomeDoNo) Chamado por: Descrição: Verifica se o campo Nome do Nó está preenchido antes de fechar a janela de criação de nó Chama: Chamado por: Descrição:
Procedimento: (77) To_fo_CriarNo.FormCloseQuery (Sender: TObject; var CanClose: Boolean); Diretório/Arquivo: criar_no.pas	Chamado por: Descrição:

(90) To_fo_CriarNo.FormActivate (Sender: TObject) Diretório/Arquivo: criar_no.pas	Não é utilizado!
Procedimento: (102) To_fo_CriarNo.o_gb_TipoDoNoClick (Sender: TObject) Diretório/Arquivo: criar_no.pas	Chama: Chamado por: Descrição: Gerencia a área da janela de criação de nós onde se escolhe o tipo do nó, terminal ou de contexto
Procedimento: (129) To_fo_CriarNo.o_gb_TipoDeNosTermina isClick (Sender: TObject) Diretório/Arquivo: criar_no.pas	Chama: Chamado por: Descrição: Gerencia a escolha do tipo de nó terminal na janela de criação de nós
Procedimento: (143) To_fo_CriarNo.o_bu_ProcurarArquivoC lick (Sender: TObject) Diretório/Arquivo: criar_no.pas	Chama: Chamado por: Descrição: Dispara o dialogo para a escolha de um arquivo. É acionado quando o botão ao lado da caixa Arquivo é clicado
Procedimento: (152) To_fo_CriarNo.o_bu_OkClick (Sender: TObject)	Chama: Chamado por: Descrição: Procedimento disparado quando o botão OK é acionado. Verifica se o nome do nó é válido e emite mensagem de erro caso contrário
Diretório/Arquivo: criar_no.pas	Chama: Chamado por:
Procedimento: (165) To_fo_CriarNo.o_cb_PalavraChaveClic k (Sender: TObject) Diretório/Arquivo: criar_no.pas	Descrição: Faz com que a palavra clicada na lista de palavras-chave apareça no campo Palavras-Chave
Procedimento: (171) To_fo_CriarNo.o_cb_PalavraChaveKeyP ress (Sender: TObject; var Key: Char) Diretório/Arquivo: criar_no.pas	Chama: (168) o_lb_PalavraChave.Items.Add(Items.String s[ItemIndex]) Chamado por: Descrição: Adiciona uma palavra chave a lista de palavras-chave depois do <enter>
Procedimento: (183) To_fo_CriarNo.o_mp_RemoverPalavraCl ick(Sender: TObject) Diretório/Arquivo: criar_no.pas	Chama: (175) o_lb_PalavraChave.Items.Add(o_cb_Palavra Chave.Text) Chamado por: Descrição: Remove a palavra clicada da lista de palavras-chave caso seja selecionado o item de menu Remover Palavra
	Chama: (187) o_lb_PalavraChave.Items.Delete(ItemIndex) Chamado por:

Sistema: SASHE**Arquivo: sashe\ criarelo.pas**

Procedimento: (61)To_fo_CriarElos.Ler_Contexto_Or igem(Indice: LongInt; Contexto_Inicial: String); Diretório/Arquivo: criarelo.pas	Descrição: Lê todos os nós abaixo do Contexto Inicial para formar o 'Browser' estrutural 'Origem'. Chama: (68) primeiro_no_contexto(StrPascalC(Contexto
--	---

	<pre> _Inicial), Nome) (72)o_ol_BrowserOrigem.AddChild(Indice, StrPas(Nome)) (73) Ler_Contexto_Origem(Indice_Auxiliar, StrPas(Nome)) (76)o_ol_BrowserOrigem.AddChild(Indice, StrPas(Nome)) (80) proximo_no_contexto(StrPascalC(Contexto_ _Inicial), Nome) (81) tipo_no(Nome) (84)o_ol_BrowserOrigem.AddChild(Indice, StrPas(Nome)) (85) Ler_Contexto_Origem(Indice_Auxiliar, StrPas(Nome)) (88)o_ol_BrowserOrigem.AddChild(Indice, StrPas(Nome)) (93)o_ol_BrowserOrigem.AddChild(Indice, '..') Chamado por: Criarelo.pas\ (61)To_fo_CriarElos.Ler_Contexto_Origem(Indice: LongInt; Contexto_Inicial: String) Criarelo.pas\ (131) To_fo_CriarElos.Inicializa_Browser(Conte xto_Inicial: String) </pre>
Procedimento: (103) To_fo_CriarElos.Ler_Contexto_Destin o(Indice: LongInt; Contexto_Inicial: String); Diretório/Arquivo: criarelo.pas	Descrição: Lê todos os nós abaixo do Contexto Inicial para formar o 'Browser' estrutural 'Destino'. Chama: (110) primeiro_no_contexto(StrPascalC(Contexto _Inicial), Nome) <i>condição do if</i> (111) tipo_no(Nome) (112)o_ol_BrowserDestino.AddChild(Indice , StrPas(Nome)) (114) Ler_Contexto_Destino(Indice_Auxiliar, StrPas(Nome)) (116) proximo_no_contexto(StrPascalC(Contexto_ Inicial), Nome) <i>condição do while</i> (117) tipo_no(Nome) (118)o_ol_BrowserDestino.AddChild(Indice , StrPas(Nome)) (120) Ler_Contexto_Destino(Indice_Auxiliar, StrPas(Nome)) (123)o_ol_BrowserDestino.AddChild(Indice , '..') Chamado por: Criarelo.pas\ (103) To_fo_CriarElos.Ler_Contexto_Destino(Ind ice: LongInt; Contexto_Inicial: String) Criarelo.pas\ (131) To_fo_CriarElos.Inicializa_Browser(Conte xto_Inicial: String)
Procedimento: (131) To_fo_CriarElos.Inicializa_Browser(Contexto_Inicial: String); Diretório/Arquivo: criarelo.pas	Descrição: Inicializa o 'Browser' estrutural 'Origem' e 'Destino' na janela de criação de elos Chama:

	<pre> (133) o_ol_BrowserOrigem.Clear (134) o_ol_BrowserDestino.Clear (135) o_ol_BrowserOrigem.AddChild(0, Contexto_Inicial) (136) o_ol_BrowserDestino.AddChild(0, Contexto_Inicial) (137) Ler_Contexto_Origem(1, Contexto_Inicial) (138) Ler_Contexto_Destino(1, Contexto_Inicial) (139) o_ol_BrowserOrigem.FullExpand (140) o_ol_BrowserDestino.FullExpand Chamado por: Nos.pas\ (930) To_fo_Nos.o_bu_CriarEloClick(Sender: TObject) Nos.pas\ (1309) To_fo_Nos.o_bu_EditarEloClick(Sender: TObject) </pre>
Procedimento: (151) To_fo_CriarElos.No_Origem_Seleciona do; Diretório/Arquivo: criarelo.pas	Descrição: Procedimento que disponibiliza o nó selecionado e suas ancoras Chama: (158) tipo_no(StrPascalC(Items[SelectedItem].T ext)) (162) o_cb_AncoraOrigem.Clear (163) primeira_ancora(StrPascalC(Items[Selecte dItem].Text), Ancora) (164) g_Erro := proxima_ancora(StrPascalC(Items[Selected Item].Text), Ancora) (166) o_cb_AncoraOrigem.Items.Add(StrPas(Ancor a)) (167) g_Erro := proxima_ancora(StrPascalC(Items[Selected Item].Text), Ancora) (175) o_cb_AncoraOrigem.Clear Chamado por: Criarelo.pas\ (189) To_fo_CriarElos.o_ol_BrowserOrigemMouseU p(Sender: TObject; Button: TMouseButton; Shift: TShiftState; X,Y: Integer) Nos.pas\ (1309) To_fo_Nos.o_bu_EditarEloClick(Sender: Tobject)
Procedimento: (189) To_fo_CriarElos.o_ol_BrowserOrigemM ouseUp(Sender: TObject; Button: TMouseButton; Shift: TShiftState; X,Y: Integer); Diretório/Arquivo: criarelo.pas	Descrição: Procedimento do 'Browser' Origem. Se o botão direito do mouse for apertado o nó selecionado no 'Browser' é Colocado como nó origem Chama: (195) To_fo_CriarElos.No_Origem_Selecionado Chamado por:
Procedimento: (202) To_fo_CriarElos.No_Destino_Selecion ado; Diretório/Arquivo: criarelo.pas	Descrição: Procedimento que disponibiliza o nó selecionado e suas ancoras. Chama: (209) tipo_no(StrPascalC(Items[SelectedItem].T

	<pre> ext)) (219) o_cb_AncoraDestino.Clear (220) g_Erro := primeira_ancora(StrPascalC(Items[Selecte dItem].Text), Ancora) (222) o_cb_AncoraDestino.Items.Add(StrPas(Anc ora)) (223) g_Erro := proxima_ancora(StrPascalC(Items[Selected Item].Text), Ancora) Chamado por: Criarelo.pas\ (242) To_fo_CriarElos.o_ol_BrowserDestinoMouse Up(Sender: TObject; Button: TMouseButton; Shift: TShiftState; X,Y: Integer) </pre>
Procedimento: (242) To_fo_CriarElos.o_ol_BrowserDestino MouseUp(Sender: TObject; Button: TMouseButton; Shift: TShiftState; X,Y: Integer); Diretório/Arquivo: criarelo.pas	Descrição: Procedimento do 'Browser' Destino. Se o botão direito do mouse for apertado o nó selecionado no 'Browser' é colocado como nó Destino. Chama: (248) To_fo_CriarElos.No_Destino_Selecionado Chamado por:
Procedimento: (255) To_fo_CriarElos.o_cb_AncoraOrigemCl ick(Sender: TObject); Diretório/Arquivo: criarelo.pas	Descrição: Seta a âncora clicada como Âncora Origem Chama: Chamado por:
Procedimento: (263) To_fo_CriarElos.o_cb_AncoraDestinoC lick(Sender: TObject); Diretório/Arquivo: criarelo.pas	Descrição: Seta a âncora clicada como Âncora Destino Chama: Chamado por:
Procedimento: (271) To_fo_CriarElos.o_ol_BrowserOrigemC lick(Sender: TObject); Diretório/Arquivo: criarelo.pas	Descrição: Este procedimento seta os valores mostrados nas etiquetas que aparecem quando um nó é selecionado dentro do 'browser' Origem na janela de criação de elos Chama: (281) recupera_funcao_didatica(StrPascalC(Items [SelectedItem].Text), v_sc_FuncaoDidatica) (282) recupera_dificuldade(StrPascalC(Items[Se lectedItem].Text), v_i_Dificuldade) (288) tipo_no(StrPascalC(Items[SelectedItem].T ext)) Chamado por:
Procedimento: (313) To_fo_CriarElos.o_ol_BrowserDestino Click(Sender: TObject); Diretório/Arquivo: criarelo.pas	Descrição: Este procedimento seta os valores mostrados nas etiquetas que aparecem quando um nó é selecionado dentro do 'browser' Destino na janela de criação deelos Chama: (323) recupera_funcao_didatica(StrPascalC (Items[SelectedItem].Text), v_sc_FuncaoDidatica)

(324) recupera_dificuldade(StrPascalC(Items[SelectedItem].Text), v_i_Dificuldade) (330) tipo_no(StrPascalC(Items[SelectedItem].Text)) Chamado por:	
Sistema: SASHE Arquivo: sashe\ edit_no.pas	
Procedimento: (60) To_fo_EditarNo.FormCreate(Sender: TObject); Diretório/Arquivo: edit_no.pas	Descrição: Inicializa a janela de edição de nós, incluindo o nome de todos os nós da hiperbase na lista do campo 'nome' Chama: (65) primeiro_no_hiperbase(NomeDoNo) (66) proximo_no_hiperbase(NomeDoNo) (67) o_cb_NomeDoNo.Items.Add(StrPas(NomeDoNo)) Chamado por:
Procedimento: (73) To_fo_EditarNo.FormActivate(Sender: TObject); Diretório/Arquivo: edit_no.pas	Descrição: Ativa a janela de edição, ou seja, passa o controle para essa janela Chama: (75) o_cb_NomeDoNo.SetFocus Chamado por:
Procedimento: (81) To_fo_EditarNo.FormCloseQuery(Sender: TObject; var CanClose: Boolean); Diretório/Arquivo: edit_no.pas	Descrição: Verifica se o campo 'nome' está completo antes de permitir que a janela seja fechada Chama: (87) o_cb_NomeDoNo.Setfocus Chamado por:
Procedimento: (94) To_fo_EditarNo.o_bu_ProcurarArquivo.Click(Sender: TObject); Diretório/Arquivo: edit_no.pas	Descrição: Dispara o dialogo para a escolha de um arquivo. É acionado quando o botão ao lado da caixa Arquivo é clicado Chama: (96) o_od_ProcurarArquivo.Execute condicao do if Chamado por:
Procedimento: (103) To_fo_EditarNo.o_bu_EditarConteudoClick(Sender: TObject); Diretório/Arquivo: edit_no.pas	Descrição: Implementa o botão Editar_Conteúdo. Caso o seja um nó ajuda, é criada a janela de edição de nó ajuda; caso seja texto, cria-se uma janela de edição de texto e caso seja um nó imagem o aplicativo PBRUSH.EXE é ativado Chama: (108) recupera_funcao_didatica(StrPascalC(o_cb_NomeDoNo.Text), FuncaoDidatica) (113) o_fo_Ajuda.Copy(a_Arquivo_Duvida, 1, Length(a_Arquivo_Duvida) - 3) (116) o_me_Duvidas.Lines.LoadFromFile(a_Arquivo_Duvida) (117) o_me_Respostas.Lines.LoadFromFile(a_Arquivo_Resposta) (119) o_fo_Ajuda.ShowModal (122) tipo_no(StrPascalC(o_cb_NomeDoNo.Text)) (126) FileExists(o_ed_Arquivo.Text) condicao do if

	(129) o_me_Texto.Lines.LoadFromFile(o_ed_Arqui vo.Text) (133) o_fo_EditarTexto.Show (139) o_fo_EditarTexto.Close (143) o_od_ProcurarArquivo.Execute (144) o_me_Texto.Lines.SaveToFile(o_od_Procura rArquivo.FileName) (149) o_fo_EditarTexto.Close Chamado por:
Procedimento: (165) To_fo_EditarNo.o_bu_OkClick(Sender: TObject); Diretório/Arquivo: edit_no.pas	Descrição: Verifica se o campo 'nome' está preenchido quando o botão OK é clicado; caso não esteja, exibe uma mensagem de erro Chama: Chamado por:
Procedimento: (178) To_fo_EditarNo.o_cb_PalavraChaveCli ck(Sender: TObject); Diretório/Arquivo: edit_no.pas	Descrição: Faz com que a palavra clicada na lista de palavras-chave apareça no campo Palavras-Chave Chama: (181) o_lb_PalavraChave.Items.Add(Items.String s[ItemIndex]) Chamado por:
Procedimento: (184) To_fo_EditarNo.o_cb_PalavraChaveKey Press(Sender: TObject; var Key: Char); Diretório/Arquivo: edit_no.pas	Descrição: Adiciona uma palavra chave a lista de palavras-chave depois do <enter> Chama: (188) o_lb_PalavraChave.Items.Add(o_cb_Palavra Chave.Text) Chamado por:
Procedimento: (194) To_fo_EditarNo.o_mp_RemoverPalavraC lick(Sender: TObject); Diretório/Arquivo: edit_no.pas	Descrição: Remove a palavra clicada da lista de palavras-chave caso seja selecionado o item de menu Remover Palavra Chama: (198) Items.Delete(ItemIndex) Chamado por:

Sistema: SASHE Arquivo: sashe\ etexto.pas	
Procedimento: (43) procedure To_fo_EditarTexto.FormClose(Sender: TObject); Diretório/Arquivo: etexto.pas	Descrição: Fecha a janela de edição de texto e criação de âncoras. Antes de fechar a janela, este procedimento percorre o texto na procura de âncoras a serem criadas Chama: (47) o_bu_GerarAncorasClick(Sender) Chamado por:
Procedimento: (54) procedure To_fo_EditarTexto.o_bu_CriarAncoraC lick(Sender: TObject); Diretório/Arquivo: etexto.pas	Descrição: Procedimentos do Button Criar Âncora Chama: (69) o_me_Texto.Lines.SaveToFile(a Arquivo) Chamado por:
Procedimento: (78) procedure	Descrição: Procedimentos do Button Gerar Âncoras

To_fo_EditarTexto.o_bu_GerarAncoras Click(Sender: TObject);	
Diretório/Arquivo: etexto.pas	Chama: (87)o_me_Texto.Lines.SaveToFile(a_Arquiv o) (94) numero_de_ancoras(StrPascalC(Nome_No)) (107) recupera_identificador_interno(StrPascal C(Nome_No), Ident_Ext) (108) cria_ancora_texto(StrPascalC(Nome_No), Ident_Ext, StrPascalC(Ancora), Inicio, Fim) (117) muda_ancora_texto(StrPascalC(Nome_No), Ident_Ext - Anc_Novas, Ident_Ext, StrPascalC(Ancora), Inicio, Fim) (130)recupera_identificador_interno(StrP ascalC(Nome_No), I); (131)retira_ancora(StrPascalC(Nome_No), Ident Int);
	Chamado por: Etexto.pas\ (43) procedure To_fo_EditarTexto.FormClose(Sender: TObject;

Sistema: SASHE	
Arquivo: sashe\funcoes.pas	
Procedimento: (163) StrPascalC(Cadeia: string)	Descrição: Converte uma string do formato "Pascal" para o formato "C"
Diretório/Arquivo: funcoes.pas	Chama: Chamado por: Atexto.pas\ (176) To_fo_Texto.o_hv_TextoHotSpotClick(Sende r: TObject; const SRC: String; var Handled: Boolean) Controle.pas\ (63) To_fo_Conroles.o_bu_TopicoAnteriorClick (Sender: TObject) Controle.pas\ (107) To_fo_Conroles.o_bu_ProximoTopicoClick (Sender: TObject) Controle.pas\ (151) To_fo_Conroles.o_bu_AnteriorClick (Sender: TObject) Controle.pas\ (187) To_fo_Conroles.o_bu_ProximoClick (Sender: TObject) Controle.pas\ (223) To_fo_Conroles.o_bu_ContextoClick (Sender: TObject) Controle.pas\ (273) To_fo_Conroles.o_bu_FacilClick (Sender: TObject) Controle.pas\ (322) To_fo_Conroles.o_bu_DifícilClick (Sender: TObject) Controle.pas\ (371)To_fo_Conroles.o_bu_AjudaClick (Sender: TObject) Controle.pas\ (467) To_fo_Conroles.o_bu_ExerciciosClick (Sender: TObject) Controle.pas\ (515)To_fo_Conroles.o_bu_MaisInformacaoe

```
sClick(Sender: TObject)
Controle.pas\ (546)
To_fo_Controles.o_bu_OndeEstouClick(Sender: TObject)
Controle.pas\
(592)To_fo_Controles.o_bu_HistoricoClick
(Sender: TObject)
Hip_prin.pas\ (256)
To_fo_Principal.FormActivate(Sender:
TObject)
Hip_prin.pas\ (317)
To_fo_Principal.FormCloseQuery (Sender:
TObject; var CanClose: Boolean)
Hip_prin.pas\ (382)
To_fo_Principal.o_bu_CriarHiperbaseClick
(Sender: TObject)
Hip_prin.pas\ (409)
To_fo_Principal.o_bu_AbrirHiperbaseClick
(Sender: TObject)
Hip_prin.pas\ (438)
To_fo_Principal.o_bu_GravarHiperbaseClic
k (Sender: TObject)
Hip_prin.pas\ (464)
To_fo_Principal.o_im_LixoDragDrop
(Sender, Source: TObject; X, Y: Integer)
Hip_prin.pas\ (475)
To_fo_Principal.o_bu_BrowserGraficoClick
(Sender: TObject)
Listanos.pas\ (52)
To_fo_ListaDeNos.o_lb_NosDbClick
(Sender: TObject)
Roteiros.pas\ (94)
To_fo_Roteiro.SelecionarNos
(p_IndiceInicial: LongInt;
p_ContextoInicial: String)
Roteiros.pas\ (193)
To_fo_Roteiro.FormCreate (Sender:
TObject)
Roteiros.pas\ (242)
To_fo_Roteiro.o_mp_InserirNoRoteiroClick
(Sender: TObject)
Roteiros.pas\ (303)
To_fo_Roteiro.o_mp_Apresentar_ConteudoCl
ick(Sender: TObject)
Roteiros.pas\ (321)
To_fo_Roteiro.o_mp_Visualizar_GrafoClick
(Sender: TObject)
Roteiros.pas\ (383)
To_fo_Roteiro.o_b_Criar_RoteiroClick
(Sender: TObject)
Roteiros.pas\ (428)
To_fo_Roteiro.o_o_Nos_SelecionadosClick(
Sender: TObject)
Roteiros.pas\ (467)
To_fo_Roteiro.o_t_TemporizadorTimer
(Sender: TObject)
Roteiros.pas\ (525)
To_fo_Roteiro.o_dg_DesenhaRoteiro(Sender
: TObject; Col, Row: Longint; Rect:
TRect; State: TGridDrawState)
Roteiros.pas\ (559)
To_fo_Roteiro.o_dg_NoSelecionado(Sender:
TObject; Col, Row: Longint; var
CanSelect: Boolean)
Roteiros.pas\ (593)
To_fo_Roteiro.o_mp_MudarNoLiberdadeClick
(Sender: TObject)
```

```
Browser.pas\ (47)
To_fo_BrowserEstrutural.Ler_Contexto(Ind
ice: LongInt; Contexto_Inicial: String)
Browser.pas\ (89)
To_fo_BrowserEstrutural.o_bu_AbrirJanela
Click(Sender: TObject)
Browser.pas\ (123)
To_fo_BrowserEstrutural.o_bu_ApresentarC
onteudoClick(Sender: TObject)
Browser.pas\ (142)
To_fo_BrowserEstrutural.o_ti_Temporizado
rTimer(Sender: TObject)
Browser.pas\ (170)
To_fo_BrowserEstrutural.o_ol_BrowserClic
k(Sender: TObject)
Criarelo.pas\
(61)To_fo_CriarElos.Ler_Contexto_Origem(
Indice: LongInt; Contexto_Inicial:
String)
Criarelo.pas\ (103)
To_fo_CriarElos.Ler_Contexto_Destino(Ind
ice: LongInt; Contexto_Inicial: String)
Criarelo.pas\ (151)
To_fo_CriarElos.No_Origem_Selecionado
Criarelo.pas\ (202)
To_fo_CriarElos.No_Destino_Selecionado
Criarelo.pas\ (271)
To_fo_CriarElos.o_ol_BrowserOrigemClick(
Sender: TObject)
Criarelo.pas\ (313)
To_fo_CriarElos.o_ol_BrowserDestinoClick
(Sender: TObject)
Edit_no.pas\ (103)
To_fo_EditarNo.o_bu_EditarConteudoClick(
Sender: TObject)
Maisinfo.pas\ (36)
To_fo_MaisInformacoes.Ler_Contexto(Conte
xto_Inicial: String)
Maisinfo.pas\ (136)
To_fo_MaisInformacoes.o_lb_NosSelecionad
osDblClick(Sender: TObject)
Nos.pas\ (106) To_fo_Nos.Ler_Nos
Nos.pas\ (291) To_fo_Nos.Ler_Elos
Nos.pas\ (557)
To_fo_Nos.o_pm_CriarNoClick(Sender:
TObject)
Nos.pas\ (561) Muda_Atributos;
Nos.pas\ (667)
To_fo_Nos.o_pm_EditarNoClick(Sender:
TObject)
Nos.pas\ (780)
To_fo_Nos.o_lb_NosClick(Sender: TObject)
Nos.pas\ (819)
To_fo_Nos.o_pm_ApagarNoClick(Sender:
TObject)
Nos.pas\ (845)
To_fo_Nos.o_pm_RetirarNoClick(Sender:
TObject)
Nos.pas\ (922)
To_fo_Nos.o_bu_BrowserGraficoClick(Sende
r: TObject)
Nos.pas\ (930)
To_fo_Nos.o_bu_CriarEloClick(Sender:
TObject)
Nos.pas\ (1049)
To_fo_Nos.o_bu_ApagarEloClick(Sender:
TObject)
```

	Nos.pas\ (1079) To_fo_Nos.o_lb_ListaElosClick(Sender: TObject) Nos.pas\ (1180) To_fo_Nos.o_lb_NosMouseDown(Sender: TObject; Button: TMouseButton; Shift: TShiftState; X, Y: Integer) Nos.pas\ (1224) To_fo_Nos.o_lb_NosDragDrop(Sender, Source: TObject; X, Y: Integer) Nos.pas\ (1257) To_fo_Nos.o_ti_Temporizador1Timer(Sender: TObject) Nos.pas\ (1309) To_fo_Nos.o_bu_EditarEloClick(Sender: TObject)
Procedimento: (182) Txt_Html (Nome_Arq: string) Diretório/Arquivo: funcoes.pas	Descrição: Converte um arquivo texto (TXT) para o formato "HTML". Esta função é chamada antes da apresentação de um nó do tipo texto. Chama: Chamado por: Atexto.pas\ (116) To_fo_Texto.Ler_Resposta(Numero_Duvida: integer) Nos.pas\ (135) To_fo_Nos.Visualizar_No_Terminal(Tipo: char; Nome No: Ptochar)
Procedimento: (225) Ler_Ajuda (Nome_Arq: string) Diretório/Arquivo: funcoes.pas	Descrição: Le os arquivos de um nó ajuda e os converte para o formato HTML. Chama: Chamado por: Nos.pas\ (135) To_fo_Nos.Visualizar_No_Terminal(Tipo: char; Nome No: Ptochar)
Procedimento: (270) Ler_ArquivoINI Diretório/Arquivo: funcoes.pas	Descrição: Le o arquivo de inicializacao. Chama: Chamado por:
Procedimento: (290) IntToBool (Valor: integer): boolean Diretório/Arquivo: funcoes.pas	Descrição: Converte um valor integer para boolean Chama: Chamado por: Nos.pas\ (1079) To_fo_Nos.o_lb_ListaElosClick(Sender: TObject) Nos.pas\ (1309) To_fo_Nos.o_bu_EditarEloClick(Sender: TObject)
Procedimento: (302) BoolToInt (Valor: boolean): integer Diretório/Arquivo: funcoes.pas	Descrição: Converte um valor boolean para integer Chama: Chamado por: Nos.pas\ (930) To_fo_Nos.o_bu_CriarEloClick(Sender: TObject) Nos.pas\ (1309) To_fo_Nos.o_bu_EditarEloClick(Sender: TObject)

Sistema: SASHE

Arquivo: sashe\ glossari.pas

Procedimento: (41) To_fo_Glossario.FormCreate(Sender: TObject); Diretório/Arquivo: glossari.pas	Descrição: Cria a janela de busca no glossário e abre a tabela do glossário Chama: (43) o_ta_Glossario.Open Chamado por:
Procedimento: (49) To_fo_Glossario.FormClose(Sender: TObject); Diretório/Arquivo: glossari.pas	Descrição: Fecha a janela do glossário e fecha a tabela glossário Chama: (52) o_ta_Glossario.Close Chamado por:
Procedimento: (58) To_fo_Glossario.o_ed_PalavraABuscarKeyPress(Sender: TObject; var Key: Char); Diretório/Arquivo: glossari.pas	Descrição: Após o <enter> este procedimento procura na tabela glossário pela palavra digitada Chama: (62) o_ta_Glossario.FindKey([o_ed_PalavraABuscar.Text]) <i>condição do if</i> Chamado por:
Procedimento: (74) To_fo_Glossario.o_dbn_ControlesClick(Sender: TObject; Button: TnavigateBtn); Diretório/Arquivo: glossari.pas	Descrição: Passa o controle para o campo Palavra caso o botao pressionado seja Inserção Chama: (78) o_dbe_Palavra.SetFocus Chamado por:

Sistema: SASHE**Arquivo: sashe\hip_prin.pas**

Procedimento: (87) To_fo_Principal.Ler_DLL Diretório/Arquivo: hip_prin.pas	Descrição: Carrega a DLL e importa as funções que serão usadas Chama: Chamado por:
Procedimento: (171) To_fo_Principal.Apresentar_Browser_Grafico(p_Contexto: Ptochar) Diretório/Arquivo: hip_prin.pas	Descrição: Inicializa o browser gráfico dentro do Sashe Chama: (179) StrPas(p_Contexto) (181) MostrarJanela(Handle) (183) ApresentaContexto(p_Contexto) Chamado por: Hip_prin.pas\ (189) To_fo_Principal.Apresentar_No(p_NomeDoNo: Ptochar; ApresentarNosTerminais: integer)
Procedimento: (189) To_fo_Principal.Apresentar_No(p_NomeDoNo: Ptochar; ApresentarNosTerminais: integer) Diretório/Arquivo: hip_prin.pas	Descrição: Apresenta os nós da hiperbase, quando os controles de navegação são acionados, ou o browser gráfico Chama: (196) tipo_no(p_NomeDoNo) (221) Apresentar_Browser_Grafico(p_NomeDoNo) (213) primeiro_no_contexto(p_NomeDoNo, NomeDoNo) (215) tipo_no(NomeDoNo) (217) o fo Nos.Visualizar No Terminal('t',

	NomeDoNo) (218) o_fo_Nos.Visualizar_No_Terminal('v', NomeDoNo) (219) o_fo_Nos.Visualizar_No_Terminal('a', NomeDoNo) (220) o_fo_Nos.Visualizar_No_Terminal('i', NomeDoNo) (222) proximo_no_contexto(p_NomeDoNo, NomeDoNo) (225) o_fo_Nos.Visualizar_No_Terminal('t', NomeDoNo) (226) o_fo_Nos.Visualizar_No_Terminal('v', NomeDoNo) (227) o_fo_Nos.Visualizar_No_Terminal('a', NomeDoNo) (228) o_fo_Nos.Visualizar_No_Terminal('i', NomeDoNo) Chamado por: Atexto.pas\ (176) To_fo_Texto.o_hv_TextoHotSpotClick(Sende r: TObject; const SRC: String; var Handled: Boolean) Controle.pas\ (63) To_fo_Controles.o_bu_TopicoAnteriorClick (Sender: TObject) Controle.pas\ (107) To_fo_Controles.o_bu_ProximoTopicoClick (Sender: TObject) Controle.pas\ (151) To_fo_Controles.o_bu_AnteriorClick (Sender: TObject) Controle.pas\ (187) To_fo_Controles.o_bu_ProximoClick (Sender: TObject) Controle.pas\ (223) To_fo_Controles.o_bu_ContextoClick (Sender: TObject) Controle.pas\ (273) To_fo_Controles.o_bu_FacilClick (Sender: TObject) Controle.pas\ (322) To_fo_Controles.o_bu_DificilClick (Sender: TObject) Controle.pas\ (371)To_fo_Controles.o_bu_AjudaClick (Sender: TObject) Controle.pas\ (419) To_fo_Controles.o_bu_BibliografiaClick (Sender: TObject) Controle.pas\ (467) To_fo_Controles.o_bu_ExerciciosClick (Sender: TObject)
Procedimento: (235) To_fo_Principal.AppMessage(var Msg: TMsg; var Handled: Boolean) Diretório/Arquivo: hip_prin.pas	Descrição: Chama: (238) TranslateMsg(@Msg) Chamado por:
Procedimento:	Descrição:

(244) To_fo_Principal.FormCreate(Sender: TObject) Diretório/Arquivo: hip_prin.pas	Construtor da janela principal Chama: Chamado por:
Procedimento: (256) To_fo_Principal.FormActivate(Sender : TObject) Diretório/Arquivo: hip_prin.pas	Descrição: Se a entrada foi para o módulo do estudante, este procedimento ativa a janela do estudante para navegação. Se a entrada foi para o módulo do professor, é ativado o procedimento o_bu_CriarHiperbaseClick, inicializando a janela para edição de hiperbase Chama: (277) recuperar(StrPascalC(a_ArquivoDaHiperbas e)) (277) StrPascalC(a_ArquivoDaHiperbase) (281) cria_base_privada (StrPascalC('HBE')) (281) StrPascalC('HBE') (282) cria_trilha (StrPascalC('Estudante')) (282) StrPascalC('Estudante') (283) insere_no_contexto (StrPascalC('HBE'), StrPascalC('Estudante')) (283) StrPascalC('HBE') (283) StrPascalC('Estudante') (284) ativa_hiperbase (StrPascalC('HBP')) (284) StrPascalC('HBP') (288) primeiro_no_Trilha (StrPascalC('Roteiro'), NomeDoNo, NomeDoNoLiberdade) (288) StrPascalC('Roteiro') (290) Apresentar_No(NomeDoNo, 1) (292) g_TrilhaDoEstudante.Add(StrPas(Nome DoNo)) (292) StrPas(NomeDoNo) (293) g_TrilhaDosNosLiberdade.Add (StrPas(NomeDoNoLiberdade)) (293) StrPas(NomeDoNoLiberdade) (294) StrPas(NomeDoNo) (295) StrPas(NomeDoNoLiberdade) (299) ativa_hiperbase(StrPascalC('HBE')) (299) StrPascalC('HBE') (300) insere_na_trilha (StrPascalC('Estudante'), NomeDoNo, NomeDoNoLiberdade) (300) StrPascalC('Estudante') (301) ativa_hiperbase(StrPascalC('HBP')) (301) StrPascalC('HBP') (311) o_bu_CriarHiperbaseClick(Sender) Chamado por:
Procedimento: (317) To_fo_Principal.FormCloseQuery (Sender: TObject; var CanClose: Boolean) Diretório/Arquivo: hip_prin.pas	Descrição: Faz os ajustes necessários antes de finalizar a aplicação Chama: (326) ativa_hiperbase(StrPascalC('HBE')) (326) StrPascalC('HBE') (328) salvar(StrPascalC(Arquivo)) (328) StrPascalC(Arquivo) Chamado por:
Procedimento:	Descrição:

(341) To_fo_Principal.o_mn_SairClick (Sender: TObject) Diretório/Arquivo: hip_prin.pas	Fecha a aplicação quando o comando Sair do menu principal é acionado Chama: Chamado por:
Procedimento: (349) To_fo_Principal.o_mn_LadoALadoClick (Sender: TObject) Diretório/Arquivo: hip_prin.pas	Descrição: Implementa o comando Lado a Lado da opção Janela do menu principal Chama: Chamado por:
Procedimento: (358) To_fo_Principal.o_mn_CascataClick (Sender: TObject) Diretório/Arquivo: hip_prin.pas	Descrição: Implementa o comando Cascata da opção Janela do menu principal Chama: Chamado por:
Procedimento: (366) To_fo_Principal.o_mn_OrganizarIcone sClick (Sender: TObject) Diretório/Arquivo: hip_prin.pas	Descrição: Implementa o comando Organizar ícones da opção Janela do menu principal Chama: Chamado por:
Procedimento: (374) To_fo_Principal.o_mn_ConteudoClick (Sender: TObject) Diretório/Arquivo: hip_prin.pas	Descrição: Implementa o comando Conteúdo da opção Ajuda do menu principal Chama: Chamado por:
Procedimento: (382) To_fo_Principal.o_bu_CriarHiperbase Click (Sender: TObject) Diretório/Arquivo: hip_prin.pas	Descrição: Libera toda a memória que estava sendo usada e inicializa uma nova hiperbase privada Chama: (388) Ler_DLL (403) cria_base_privada (StrPascalC(g_NomeDaHiperbase)) (403) StrPascalC(g_NomeDaHiperbase) Chamado por: Hip_prin.pas\ (256) To_fo_Principal.FormActivate(Sender: TObject)
Procedimento: (409) To_fo_Principal.o_bu_AbrirHiperbase Click (Sender: TObject) Diretório/Arquivo: hip_prin.pas	Descrição: Recarrega a DLL, fecha as janelas filhas que estiverem abertas, ativa a procura do arquivo, e recupera o arquivo escolhido Chama: (415) Ler_DLL (423) recuperar (StrPascalC(o_od_ProcurarArquivo.FileName)) (423) StrPascalC(o_od_ProcurarArquivo.FileName) Chamado por:
Procedimento: (438) To_fo_Principal.o_bu_GravarHiperbas eClick (Sender: TObject) Diretório/Arquivo: hip_prin.pas	Descrição: Ativa a janela para o salvamento da hiperbase, salva e cria backup do arquivo da hiperbase aberta. Chama: (447) salvar (StrPascalC(o_sd_ProcurarArquivo.FileName)) (447) StrPascalC(o_sd_ProcurarArquivo.FileName)

Chamado por:	
Procedimento: (455) To_fo_Principal.o_im_LixoDragOver (Sender, Source: TObject; X, Y: Integer; State: TDragState; var Accept: Boolean) Diretório/Arquivo: hip_prin.pas	Descrição: Manipula uma lixeira que não é utilizada na aplicação
Chama: Chamado por:	
Procedimento: (464) To_fo_Principal.o_im_LixoDragDrop (Sender, Source: TObject; X, Y: Integer) Diretório/Arquivo: hip_prin.pas	Descrição: Manipula uma lixeira que não é utilizada na aplicação
Chama: (467) retira_da_hiperbase (StrPascalC(g_NomeDrag)) (467) StrPascalC(g_NomeDrag) (469) g_ListaDrag.Items.Delete (g_LinhaDrag) Chamado por:	
Procedimento: (475) To_fo_Principal.o_bu_BrowserGrafico Click (Sender: TObject) Diretório/Arquivo: hip_prin.pas	Descrição: Apresenta os nós dentro do contexto atual no browser gráfico
Chama: (477) Apresentar_No (StrPascalC(a_ContextoAtual), 0) (477) StrPascalC(a_ContextoAtual) Chamado por:	
Procedimento: (483) To_fo_Principal.o_bu_RoteiroClick(S ender: TObject) Diretório/Arquivo: hip_prin.pas	Descrição: Abre a janela de criação de roteiros
Chama: Chamado por:	
Procedimento: (502) To_fo_Principal.o_bu_GlossarioClick (Sender: TObject) Diretório/Arquivo: hip_prin.pas	Descrição: Abre a janela de glossário
Chama: Chamado por:	

Sistema: SASHE	
Arquivo: sashe\imagem.pas	
Procedimento: (32) To_fo_Imagem.FormCreate(Sender: TObject); Diretório/Arquivo: imagem.pas	Descrição:
Chama: Cria uma janela para a apresentação de uma imagem Chamado por:	
Procedimento: (46) To_fo_Imagem.FormClose(Sender: TObject; var Action: TCloseAction); Diretório/Arquivo: imagem.pas	Descrição: Fecha a janela de apresentação de imagem
Chama: Chamado por:	

Sistema: SASHE	
Arquivo: sashe\listanos.pas	
Procedimento: (34) To_fo_ListaDeNos.FormCreate (Sender: TObject)	Descrição: Cria a janela que apresenta os nós quando os botões Histórico ou Onde Estou? dos comandos de navegação são acionados

Diretório/Arquivo: listanos.pas	Chama: Chamado por:
Procedimento: (43) To_fo_ListaDeNos.FormClose (Sender: TObject; var Action: TCloseAction)	Descrição: Libera a memória alocada para a janela quando a mesma é fechada
Diretório/Arquivo: listanos.pas	Chama: Chamado por:
Procedimento: (52) To_fo_ListaDeNos.o_lb_NosDbClick (Sender: TObject);	Descrição: Implementa a funcionalidade atribuída ao click duplo dentro da janela Histórico, ou seja, apresenta o nó que for clicado duplamente
Diretório/Arquivo: listanos.pas	Chama: (64) o_fo_Principal.Apresentar_No (StrPascalC(Items.Strings[ItemIndex]), 1) (64) StrPascalC(Items.Strings[ItemIndex]) Chamado por:

Sistema: SASHE	
Arquivo: sashe\ maisinfo.pas	
Procedimento: (36) To_fo_MaisInformacoes.Ler_Contexto(Contexto_Inicial: String);	Descrição: Faz a busca dentro do contexto pelos nós que possuam alguma palavra-chave em comum com o nó atual e mostra na janela Mais Informações!
Diretório/Arquivo: maisinfo.pas	Chama: (43) primeiro_no_contexto(StrPascalC(Contexto _Inicial), NomeDoNo) (47) recupera_palavra_chave(StrPascalC(g_NoDo Roteiro), I, PalavraChave) (56) recupera_palavra_chave(NomeDoNo, J, PalavraChave) (65) o_lb_NosSelecionados.Items.AddObject(Str Pas(NomeDoNo), TObject(Makelong(0,4))) (66) o_lb_NosSelecionados.Items.AddObject(Str Pas(NomeDoNo), TObject(Makelong(0,3))) (67) o_lb_NosSelecionados.Items.AddObject(Str Pas(NomeDoNo), TObject(Makelong(0,1))) (68) o_lb_NosSelecionados.Items.AddObject(Str Pas(NomeDoNo), TObject(Makelong(0,2))) (76) Ler_Contexto(StrPas(NomeDoNo)) (77) proximo_no_contexto(StrPascalC(Contexto_ Inicial), NomeDoNo) (78) tipo_no(NomeDoNo) (81) recupera_palavra_chave(StrPascalC(g_NoDo Roteiro), I, PalavraChave) (90) recupera_palavra_chave(NomeDoNo, J, PalavraChave) (99) o_lb_NosSelecionados.Items.AddObject(Str Pas(NomeDoNo), TObject(Makelong(0,4))) (100) o_lb_NosSelecionados.Items.AddObject(Str Pas(NomeDoNo), TObject(Makelong(0,3))) (101) o_lb_NosSelecionados.Items.AddObject(Str

	Pas(NomeDoNo), TObject(Makelong(0,1))) (102) o_lb_NosSelecionados.Items.AddObject(StrPas(NomeDoNo), TObject(Makelong(0,2))) (110) Ler_Contexto(StrPas(NomeDoNo)) Chamado por: Browser.pas\ (47) to_fo_browserestrutural.ler_contexto(indice: longint; contexto_inicial: string) Browser.pas\ (70) To_fo_BrowserEstrutural.Atualiza_BrowserMaisinfo.pas\ (36) To_fo_MaisInformacoes.Ler_Contexto(Contexto_Inicial: String) Controle.pas\ (515)To_fo_Conroles.o_bu_MaisInformacoesClick(Sender: TObject)
Procedimento: (118) To_fo_MaisInformacoes.FormCreate(Sender: TObject); Diretório/Arquivo: maisinfo.pas	Descrição: Cria a janela Mais Informações! Chama: Chamado por:
Procedimento: (127) To_fo_MaisInformacoes.FormClose(Sender: TObject; var Action: TCloseAction); Diretório/Arquivo: maisinfo.pas	Descrição: Fecha a janela e libera a memória Chama: Chamado por:
Procedimento: (136) To_fo_MaisInformacoes.o_lb_NosSelecionadosDblClick(Sender: TObject); Diretório/Arquivo: maisinfo.pas	Descrição: Chama o procedimento responsável pela visualização do nó caso seja dado um clique duplo Chama: (142) tipo_no(StrPascalC(Items.Strings[ItemIndex])) (144) o_fo_Nos.Visualizar_No_Terminal('t', StrPascalC(Items.Strings[ItemIndex])) (145) o_fo_Nos.Visualizar_No_Terminal('v', StrPascalC(Items.Strings[ItemIndex])) (146) o_fo_Nos.Visualizar_No_Terminal('a', StrPascalC(Items.Strings[ItemIndex])) (147) o_fo_Nos.Visualizar_No_Terminal('i', StrPascalC(Items.Strings[ItemIndex])) Chamado por:

Sistema: SASHE**Arquivo:** sashe\nos.pas

Procedimento: (106) To_fo_Nos.Ler_Nos; Diretório/Arquivo: nos.pas	Descrição: Lê e Mostra os Nós do Contexto Atual Chama: (111) o_lb_Nos.Clear (112) primeiro_no_contexto(StrPascalC(a_ContextoAtual), NomeDoNo) (114) tipo_no(NomeDoNo) (117) o_lb_Nos.Items.AddObject(StrPas(NomeDoNo), TObject(Makelong(0,0))) (118) o_lb_Nos.Items.AddObject(StrPas(NomeDoNo)
---	--

	<pre>), TObject(Makelong(0,4)) (119) o_lb_Nos.Items.AddObject(StrPas(NomeDoNo), TObject(Makelong(0,3))) (120) o_lb_Nos.Items.AddObject(StrPas(NomeDoNo), TObject(Makelong(0,1))) (121) o_lb_Nos.Items.AddObject(StrPas(NomeDoNo), TObject(Makelong(0,2))) (122) o_lb_Nos.Items.AddObject(StrPas(NomeDoNo), TObject(Makelong(0,5))) (124) proximo_no_contexto(StrPascalC(a_ContextoAtual), NomeDoNo) Chamado por: Nos.pas\ (488) To_fo_Nos.FormActivate(Sender: TObject) Nos.pas\ (542) To_fo_Nos.o_pm_VisualizarContextoClick(Sender: TObject) Nos.pas\ (667) To_fo_Nos.o_pm_EditarNoClick(Sender: TObject) </pre>
Procedimento: (135) To_fo_Nos.Visualizar_No_Terminal(Tipo: char; Nome_No: Ptochar); Diretório/Arquivo: nos.pas	Descrição: Apresenta o conteúdo de um nó terminal Parâmetros de Entrada: Tipo: Tipo do Nó Nome_No: Nome do Nó a ser apresentado Chama: (144) recupera_nome_arquivo_texto(Nome_No, Arquivo) (159) o_fo_Texto.recupera_funcao_didatica(Nome_No, FuncaoDidatica) (161) o_fo_Texto.Ler_Ajuda(StrPas(Arquivo)) (163) o_fo_Texto.Txt_Html(StrPas(Arquivo)) (165) o_hv_Texto.LoadFromFile(ArquivoTemp) (179) o_fo_Principal.MDIChildren[Numero].BringToFront (184) recupera_nome_arquivo_audio(Nome_No, Arquivo) (207) recupera_nome_arquivo_video(Nome_No, Arquivo) (246) o_fo_Principal.MDIChildren[Numero].BringToFront (251) recupera_nome_arquivo_imagem(Nome_No, Arquivo) (264) o_im_Imagem.Picture.Bitmap.LoadFromFile(StrPas(Arquivo)) (281) o_fo_Principal.MDIChildren[Numero].BringToFront Chamado por: Nos.pas\ (862) To_fo_Nos.o_pm_ApresentarConteudoDoNoCli

	<pre> ck(Sender: TObject) Hip_prin.pas\ (189) To_fo_Principal.Apresentar_No(p_NomeDoNo : Ptochar; ApresentarNosTerminais: integer) Roteiros.pas\ (303) To_fo_Roteiro.o_mp_Apresentar_ConteudoCl ick(Sender: TObject) Browser.pas\ (123) To_fo_BrowserEstrutural.o_bu_ApresentarC onteudoClick(Sender: TObject) Maisinfo.pas\ (136) To_fo_MaisInformacoes.o_lb_NosSelecionad osDbClick(Sender: TObject) </pre>
Procedimento: (291) To_fo_Nos.Ler_Elos; Diretório/Arquivo: nos.pas	Descrição: Lê e apresenta os elos do contexto atual Chama: <pre> (301) o_ol_OrigemDoElo.Clear (302) o_ol_DestinoDoElo.Clear (305) numero_de_elos(StrPascalC(a_ContextoAtua l)) (306) o_lb_ListaElos.Clear (307) primeiro_elo(StrPascalC(a_ContextoAtual) , I) (309) recupera_elo(StrPascalC(a_ContextoAtual) , I, Lista_Origem, Ident_Anc_Origem, Lista_Destino, Ident_Anc_Destino, ApresentarNoSimultaneo , ApresentarNosTerminais) (324) o_lb_ListaElos. retorna_nome(StrToInt(StrPas(Ident_No_Or igem)), Nome_No) (325) o_lb_ListaElos.recupera_nome_ancora(Nome _No, Ident_Anc_Origem, Nome_Ancora) (337) o_lb_ListaElos.retorna_nome(StrToInt(Str Pas(Ident_No_Destino)), Nome_No) (340) o_lb_ListaElos.recupera_nome_ancora(Nome _No, Ident_Anc_Destino, Nome_Ancora) (344) o_lb_ListaElos.Items.Add(Origem + ' -> ' + Destino) (351) proximo_elo(StrPascalC(a_ContextoAtual), I) Chamado por: Nos.pas\ (488) To_fo_Nos.FormActivate(Sender: TObject) Nos.pas\ (1049) To_fo_Nos.o_bu_ApagarEloClick(Sender: TObject) Nos.pas\ (1170) To_fo_Nos.o_tn_PastaChange(Sender: TObject; NewTab: Integer; var AllowChange: Boolean) </pre>
Procedimento: (362) To_Fo_Nos.ApresentarNoTrilha(NoSele cionado: Ptochar); Diretório/Arquivo: nos.pas	Descrição: Abre uma janela para apresentação de uma trilha, caso o nó selecionado ao se selecionar o comando Apresentar Nó seja um nó trilha Chama: (373)

	o_fo_Trilha.primeiro_no_Trilha(NoSelecio nado, NomeDoNo, NomeDoNoLiberdade) (375) tipo_no(NomeDoNo) (463) proximo_no_Trilha(NoSelecio nado, NomeDoNo, NomeDoNoLiberdade) Chamado por: Nos.pas\ (862) To_fo_Nos.o_pm ApresentarConteudoDoNoCli ck(Sender: TObject) Browser.pas\ (123) To_fo_BrowserEstrutural.o_bu ApresentarC onteudoClick(Sender: TObject)
Procedimento: (472) To_fo_Nos.FormCreate(Sender: TObject); Diretório/Arquivo: nos.pas	Descrição: Cria a janela de Nós e Elos e abre o contexto especificado em g_ContextoASerAberto Chama: (478) a_Perspectiva.Add(a_ContextoAtual) Chamado por:
Procedimento: (488) To_fo_Nos.FormActivate(Sender: TObject); Diretório/Arquivo: nos.pas	Descrição: Quando a janela é ativada é mostrado os nós e os elos do contexto Chama: (491) Ler_Nos (492) Ler_Elos Chamado por:
Procedimento: (499) To_fo_Nos.FormDeactivate(Sender: TObject); Diretório/Arquivo: nos.pas	Descrição: Quando o controle passa para outra aplicação esse procedimento é realizado Chama: (502) o_ti_Temporizador1Timer(Sender) Chamado por:
Procedimento: (510) To_fo_Nos.FormClose(Sender: TObject; var Action: TcloseAction); Diretório/Arquivo: nos.pas	Descrição: Fecha a janela Nós e Elos, somente se ela não for a primeira Nós e Elos Chama: (514) a_Perspectiva.Free Chamado por:
Procedimento: (521) To_fo_Nos.o_pm_SubirContextoClick(S ender: TObject); Diretório/Arquivo: nos.pas	Descrição: Permite a navegação entre os contextos, voltando pro contexto anterior, através do comando Retornar ao Contexto Anterior Chama: (523) a_Perspectiva.Delete(a_Perspectiva.Count - 1) Chamado por:
Procedimento: (542) To_fo_Nos.o_pm_VisualizarContextoCl ick(Sender: TObject); Diretório/Arquivo: nos.pas	Descrição: Permite a navegação entre os contextos, visualizando o conteúdo do contexto selecionado através do comando Visualizar Contexto Chama: (548) a_Perspectiva.Add(a_ContextoAtual) (550) Ler_Nos Chamado por:
Procedimento: (557) To_fo_Nos.o_pm_CriarNoClick(Sender: TObject); Diretório/Arquivo: nos.pas	Descrição: Abre a janela para criação de um novo nó, terminal ou de contexto. Caso o nó seja terminal, ele será criado de acordo com o tipo de nó escolhido Chama: (566)muda_funcao_didatica(StrPascalC(o_c b_NomeDoNo.Text), StrPascalC(o_cb_FuncaoDidatica.Text))

```
(569)
muda_dificuldade(StrPascalC(o_cb_NomeDoNo
o.Text), Items.IndexOf(Text))
(571)
muda_palavra_chave(StrPascalC(o_cb_NomeD
oNo.Text), I, StrPascalC(''))
(573)
muda_palavra_chave(StrPascalC(o_cb_NomeD
oNo.Text), I,
StrPascalC(o_lb_PalavraChave.Items.Strin
gs[I]))
(585)
cria_contexto(StrPascalC(o_cb_NomeDoNo.T
ext))
(590)
insere_no_contexto(StrPascalC(a_Contexto
Atual), StrPascalC(o_cb_NomeDoNo.Text))
(593)
Items.AddObject(o_cb_NomeDoNo.Text,
TObject(Makelong(0,0)))
(603)
cria_texto(StrPascalC(o_cb_NomeDoNo.Text
))
(604) Muda_Atributos
(605)
insere_no_contexto(StrPascalC(a_Contexto
Atual), StrPascalC(o_cb_NomeDoNo.Text))
(607)
insere_nome_arquivo_texto(StrPascalC(o_c
b_NomeDoNo.Text),
StrPascalC(o_ed_Arquivo.Text))
(609)
Items.AddObject(o_cb_NomeDoNo.Text,
TObject(Makelong(0,4)))
(617)
cria_audio(StrPascalC(o_cb_NomeDoNo.Text
))
(618) Muda_Atributos
(619)
insere_no_contexto(StrPascalC(a_Contexto
Atual), StrPascalC(o_cb_NomeDoNo.Text))
(621)
insere_nome_arquivo_audio(StrPascalC(o_c
b_NomeDoNo.Text),
StrPascalC(o_ed_Arquivo.Text))
(623)
Items.AddObject(o_cb_NomeDoNo.Text,
TObject(Makelong(0,1)))
(631)
cria_video(StrPascalC(o_cb_NomeDoNo.Text
))
(632) Muda_Atributos
(633)
insere_no_contexto(StrPascalC(a_Contexto
Atual), StrPascalC(o_cb_NomeDoNo.Text))
(635)
insere_nome_arquivo_video(StrPascalC(o_c
b_NomeDoNo.Text),
StrPascalC(o_ed_Arquivo.Text))
(636)
Items.AddObject(o_cb_NomeDoNo.Text,
TObject(Makelong(0,3)))
(645)
cria_imagem(StrPascalC(o_cb_NomeDoNo.Tex
t))
(646) Muda_Atributos
(647)
```

	<pre> insere_no_contexto(StrPascalC(a_Contexto Atual), StrPascalC(o_cb_NomeDoNo.Text)) (649) insere_nome_arquivo_imagem(StrPascalC(o_ cb_NomeDoNo.Text), StrPascalC(o_ed_Arquivo.Text)) (651) Items.AddObject(o_cb_NomeDoNo.Text, TObject(Makelong(0,2))) Chamado por: </pre>
Procedimento: (561) Muda Atributos; Diretório/Arquivo: nos.pas	Descrição: Muda os atributos dos nós terminais Chama: (566) muda_funcao_didatica(StrPascalC(o_cb_Nom eDoNo.Text), StrPascalC(o_cb_FuncaoDidatica.Text)) (569) muda_dificuldade(StrPascalC(o_cb_NomeDoN o.Text), Items.IndexOf(Text)) (571) muda_palavra_chave(StrPascalC(o_cb_NomeD oNo.Text), I, StrPascalC('')) (573) muda_palavra_chave(StrPascalC(o_cb_NomeD oNo.Text), I, StrPascalC(o_lb_PalavraChave.Items.Strin gs[I])) Chamado por: Nos.pas\ (557) To_fo_Nos.o_pm_CriarNoClick(Sender: TObject)
Procedimento: (667) To_fo_Nos.o_pm_EditarNoClick(Sender : TObject); Diretório/Arquivo: nos.pas	Descrição: Faz a chamada para a criação de uma janela de edição de nós, para a edição do nó selecionado Chama: (677) tipo_no(StrPascalC(Nome_atual)) (686) recupera_dificuldade(StrPascalC(o_cb_Nom eDoNo.Text), Atributo2) (689) recupera_funcao_didatica(StrPascalC(o_cb _NomeDoNo.Text), Atributo1) (692) recupera_palavra_chave(StrPascalC(o_cb_N omeDoNo.Text), I, Atributo1) (696) o_lb_PalavraChave.Items.Add(StrPas(Atrib uto1)) (712) recupera_nome_arquivo_texto(StrPascalC(N ome_atual), Arquivo) (721) recupera_nome_arquivo_video(StrPascalC(N ome_atual), Arquivo) (727) recupera_nome_arquivo_audio(StrPascalC(N ome_atual), Arquivo) (733) recupera_nome_arquivo_imagem(StrPascalC(Nome_atual), Arquivo) (741) o_lb_PalavraChave.Clear (745) muda_atributos(StrPascalC(Nome_atual), StrPascalC(o_cb_NomeDoNo.Text), 0) (747)

	<pre> insere_nome_arquivo_texto(StrPascalC(o_c b_NomeDoNo.Text), StrPascalC(o_ed_Arquivo.Text)) (749) insere_nome_arquivo_video(StrPascalC(o_c b_NomeDoNo.Text), StrPascalC(o_ed_Arquivo.Text)) (751) insere_nome_arquivo_audio(StrPascalC(o_c b_NomeDoNo.Text), StrPascalC(o_ed_Arquivo.Text)) (753) insere_nome_arquivo_imagem(StrPascalC(o_ cb_NomeDoNo.Text), StrPascalC(o_ed_Arquivo.Text)) (760) muda_dificuldade(StrPascalC(o_cb_NomeDoN o.Text), Items.IndexOf(Text)) (761) muda_funcao_didatica(StrPascalC(o_cb_Nom eDoNo.Text), StrPascalC(o_cb_FuncaoDidatica.Text)) (765) muda_palavra_chave(StrPascalC(o_cb_NomeD oNo.Text), I, StrPascalC('')) (767) muda_palavra_chave(StrPascalC(o_cb_NomeD oNo.Text), I, StrPascalC(o_lb_PalavraChave.Items.Strin gs[I])) (771) Ler_Nos Chamado por: </pre>
<pre> Procedimento: (780) To_fo_Nos.o_lb_NosClick(Sender: TObject); Diretório/Arquivo: nos.pas </pre>	<pre> Descrição: Prepara o conteúdo das etiquetas que são mostradas quando um nó é selecionado Chama: (788) recupera_funcao_didatica(StrPascalC(Item s[ItemIndex]), v_sc_FuncaoDidatica) (789) recupera_dificuldade(StrPascalC(Items[It emIndex]), v_i_Dificuldade) (795) tipo_no(StrPascalC(Items[ItemIndex])) Chamado por: </pre>
<pre> Procedimento: (819) To_fo_Nos.o_pm_ApagarNoClick(Sender : TObject); Diretório/Arquivo: nos.pas </pre>	<pre> Descrição: Apaga o nó efetivamente, retirando-o da hiperbases. Se for um nó trilha a trilha é limpa Chama: (830) tipo_no(StrPascalC(Items[ItemIndex])) (832) limpa_trilha(StrPascalC(Items[ItemIndex])) (833) retira_da_hiperbase(StrPascalC(Items[Ite mIndex])) (834) Items.Delete(ItemIndex) Chamado por: </pre>
<pre> Procedimento: (845) To_fo_Nos.o_pm_RetirarNoClick(Sende r: TObject); Diretório/Arquivo: nos.pas </pre>	<pre> Descrição: Retira o nó do contexto Chama: (850) </pre>

	retira_do_contexto(StrPascalC(a_ContextoAtual), StrPascalC(o_lb_Nos.Items[o_lb_Nos.ItemIndex])) (852) o_lb_Nos.Items.Delete(o_lb_Nos.ItemIndex) Chamado por:
Procedimento: (862) To_fo_Nos.o_pm_ApresentarConteudoDo NoClick(Sender: TObject); Diretório/Arquivo: nos.pas	Descrição: Apresenta o conteúdo do nó de acordo com seu tipo Chama: (869) tipo_no(NoSelecioneado) (871) Visualizar_No_Terminal('t', NoSelecioneado) (872) Visualizar_No_Terminal('v', NoSelecioneado) (873) Visualizar_No_Terminal('a', NoSelecioneado) (874) Visualizar_No_Terminal('i', NoSelecioneado) (875) ApresentarNoTrilha(NoSelecioneado) Chamado por:
Procedimento: (883) To_fo_Nos.o_pm_AbrirJanelaClick(Sender: TObject); Diretório/Arquivo: nos.pas	Descrição: Abre o contexto selecionado em uma nova janela Nós e Elos Chama: (900) o_fo_Principal.MDIChildren[Numero].BringToFront Chamado por:
Procedimento: (908) To_fo_Nos.o_pm_BrowserClick(Sender: TObject); Diretório/Arquivo: nos.pas	Descrição: Faz a chamada para o browser estrutural Chama: (915) Atualiza_Browser Chamado por:
Procedimento: (922) To_fo_Nos.o_bu_BrowserGraficoClick(Sender: TObject); Diretório/Arquivo: nos.pas	Descrição: Faz a chamada para o browser gráfico Chama: (924) o_fo_Principal.Apresentar_No(StrPascalC(a_ContextoAtual), 0) Chamado por:
Procedimento: (930) To_fo_Nos.o_bu_CriarEloClick(Sender: TObject); Diretório/Arquivo: nos.pas	Descrição: Abre a janela para criação de um novo elo, ativado pelo botão Criar Elo Chama: (941) Inicializa_Browser(a_ContextoAtual) (951) Lista.Add(Pai) (976) Lista.Free (984) retorna_identificador(StrPascalC(Items[a_Ident_No_Origem].Text)) (989) retorna_identificador(StrPascalC(Items[Index_Pai].Text)) (1002) retorna_identificador(StrPascalC(Items[a

	<pre> _Ident_No_Destino].Text)) (1007) retorna_identificador(StrPascalC(Items[Indice_Pai].Text)) (1016) numero_de_elos(StrPascalC(Contexto)) (1017) tipo_no(StrPascalC(o_ol_BrowserDestino.Items[a_Ident_No_Destino].Text)) (1019) recupera_identificador_interno(StrPascalC(o_la_NomeDoNoOrigem.Caption), a_Ident_Externo_Ancora_Origem) (1021) recupera_identificador_interno(StrPascalC(o_la_NomeDoNoDestino.Caption), a_Ident_Externo_Ancora_Destino) (1031) cria_elo(StrPascalC(Contexto), N_Elos, StrPascalC(Lista_Origem), Ident_Anc_Origem, StrPascalC(Lista_Destino), Ident_Anc_Destino, BoolToInt(o_chb_ApresentarSimultaneo.Checked), BoolToInt(o_chb_ApresentarNosTerminaisDoContexto.Checked)) (1041) o_tn_PastaChange(Sender, 1, Aux) Chamado por: </pre>
Procedimento: (1049) To_fo_Nos.o_bu_ApagarEloClick(Sender: TObject); Diretório/Arquivo: nos.pas	Descrição: Elimina o elo da lista de elos, ativado pelo botão Apagar Elo Chama: (1054) retira_elo(StrPascalC(a_ContextoAtual), o_lb_ListaElos.ItemIndex) (1061) Items.Delete(ItemIndex) (1066) Ler_Elos (1068) o_lb_ListaElosClick(Sender) (1071) o_ol_OrigemDoElo.Clear (1072) o_ol_DestinoDoElo.Clear Chamado por:
Procedimento: (1079) To_fo_Nos.o_lb_ListaElosClick(Sender: TObject); Diretório/Arquivo: nos.pas	Descrição: Faz com que, ao se clicar em um dos elos mostrados na janela, o mesmo seja mostrado graficamente Chama: (1094) recupera_elo(StrPascalC(a_ContextoAtual), StrToInt(StrPas(Indice)), Lista_Origem, Ident_Anc_Origem, Lista_Destino, Ident_Anc_Destino, ApresentarNoSimultaneo, ApresentarNosTerminais) (1099) o_ol_OrigemDoElo.Clear (1100) o_ol_DestinoDoElo.Clear (1117) retorna_nome(StrToInt(StrPas(Ident_No_Origem)), Nome_No) (1118) Lista_Auxiliar.Insert(0, StrPas(Nome_No)) (1125) o_ol_OrigemDoElo.AddChild(Cont, Lista_Auxiliar.Strings[Cont]) (1126) o_ol_OrigemDoElo.FullExpand (1130) Lista_Auxiliar.Clear (1141) retorna_nome(StrToInt(StrPas(Ident_No_De

	<pre> stino)), Nome_No) (1142) Lista_Auxiliar.Insert(0, StrPas(Nome_No)) (1149) o_ol_DestinoDoElo.AddChild(Cont, Lista_Auxiliar.Strings[Cont]) (1150) tipo_no(StrPascalC(Lista_Auxiliar.String s[Cont])) (1152) o_ol_DestinoDoElo.AddChild(Cont + 1, '...') (1153) IntToBool(ApresentarNosTerminais) (1155) o_ol_DestinoDoElo.FullExpand (1157) Lista_Auxiliar.Free (1158) IntToBool(ApresentarNoSimultaneo) Chamado por: Nos.pas\ (1049) To_fo_Nos.o_bu_ApagarEloClick(Sender: TObject) </pre>
Procedimento: (1170) To_fo_Nos.o_tn_PastaChange(Sender: TObject; NewTab: Integer; var AllowChange: Boolean); Diretório/Arquivo: nos.pas	Descrição: Faz com mudança de pastas (Nós ou Elos) dentro da janela Nós e Elos Chama: (1173) Ler_Elos Chamado por: Nos.pas\ (930) To_fo_Nos.o_bu_CriarEloClick(Sender: TObject) Nos.pas\ (1309) To_fo_Nos.o_bu_EditarEloClick(Sender: TObject)
Procedimento: (1180) To_fo_Nos.o_lb_NosMouseDown(Sender: TObject; Button: TMouseButton; Shift: TShiftState; X, Y: Integer); Diretório/Arquivo: nos.pas	Descrição: Implementa os comandos Copiar e Mover entre janelas Nós e Elos, controlados pelo mouse Chama: (1194) tipo_no(StrPascalC(Items[ItemIndex])) Chamado por:
Procedimento: (1205) To_fo_Nos.o_lb_NosMouseUp(Sender: TObject; Button: TMouseButton; Shift: TShiftState; X, Y: Integer); Diretório/Arquivo: nos.pas	Descrição: Faz parte da implementação dos comandos Copiar e Mover entre janelas Nós e Elos, controlados pelo mouse Chama: Chamado por:
Procedimento: (1214) , To_fo_Nos.o_lb_NosDragOver(Sender, Source: TObject; X, Y: Integer; State: TDragState; var Accept: Boolean); Diretório/Arquivo: nos.pas	Descrição: Faz parte da implementação dos comandos Copiar e Mover entre janelas Nós e Elos, controlados pelo mouse Chama: Chamado por:
Procedimento: (1224) To_fo_Nos.o_lb_NosDragDrop(Sender, Source: TObject; X, Y: Integer); Diretório/Arquivo: nos.pas	Descrição: Faz parte da implementação dos comandos Copiar e Mover entre janelas Nós e Elos, controlados pelo mouse Chama: (1227) insere_no_contexto(StrPascalC(a_Contexto Atual), StrPascalC(g_NomeDrag)) (1232) Items.AddObject(g_NomeDrag,

	<pre> TObject(Makelong(0,0))) (1233) Items.AddObject(g_NomeDrag, TObject(Makelong(0,4))) (1234) Items.AddObject(g_NomeDrag, TObject(Makelong(0,3))) (1235) Items.AddObject(g_NomeDrag, TObject(Makelong(0,1))) (1236) Items.AddObject(g_NomeDrag, TObject(Makelong(0,2))) (1237) Items.AddObject(g_NomeDrag, TObject(Makelong(0,5))) (1239) retira_do_contexto(StrPascalC(g_Contexto Drag), StrPascalC(g_NomeDrag)) (1241) g_ListaDrag.Items.Delete(g_LinhaDrag) Chamado por: </pre>
Procedimento: (1257) To_fo_Nos.o_ti_Temporizador1Timer(S ender: TObject); Diretório/Arquivo: nos.pas	Descrição: É um timer acionado a cada milisegundo que interrompe o controle normal do aplicativo e executando o código especificado no corpo do procedimento Chama: (1276) tipo_no(StrPascalC(o_lb_Nos.Items[o_lb_N os.ItemIndex])) Chamado por: Nos.pas\ (499) To_fo_Nos.FormDeactivate(Sender: TObject)
Procedimento: (1309) To_fo_Nos.o_bu_EditarEloClick(Sende r: TObject); Diretório/Arquivo: nos.pas	Descrição: Abre uma nova janela de criação de elos para a edição do elo selecionado Chama: (1320) StrPascalC(o_lb_ListaElos.Items[o_lb_Lis taElos.ItemIndex]) (1327) recupera_elo(StrPascalC(a_ContextoAtual) , Id_Elo, Recupera_Lista_Origem, Ident_Anc_Origem, Recupera_Lista_Destino, Ident_Anc_Destino, ApresentarNoSimultaneo, ApresentarNosTerminais) (1333) Inicializa_Browser(a_ContextoAtual) (1335) IntToBool(ApresentarNoSimultaneo) (1336) IntToBool(ApresentarNosTerminais) (1342) retorna_nome(StrToInt(StrPas(Recupera_Li sta_Origem)), Indice) (1343) o_ol_BrowserOrigem.GetTextItem(StrPas(In dice)) (1344) No_Origem_Selecionado (1360) retorna_nome(StrToInt(StrPas(Recupera_Li sta_Destino)), Indice) (1362) o_ol_BrowserDestino.GetTextItem(StrPas(I ndice)) (1374) Lista.Add(Pai) (1407) retorna_identificador(StrPascalC(Items[a _Ident_No_Origem].Text))

```

(1412)
retorna_identificador(StrPascalC(Items[I
ndice_Pai].Text))
(1425)
retorna_identificador(StrPascalC(Items[a
_Ident_No_Destino].Text))
(1430)
retorna_identificador(StrPascalC(Items[I
ndice_Pai].Text))
(1439)
tipo_no(StrPascalC(o_ol_BrowserDestino.I
tems[a_Ident_No_Destino].Text))
(1441)
recupera_identificador_interno(StrPascal
C(o_la_NomeDoNoOrigem.Caption),
a_Ident_Externo_Ancora_Origem)
(1443)
recupera_identificador_interno(StrPascal
C(o_la_NomeDoNoDestino.Caption),
a_Ident_Externo_Ancora_Destino)
(1453) muda_elo(StrPascalC(Contexto),
Id_Elo, StrPascalC(Lista_Origem),
Ident_Anc_Origem,
StrPascalC(Lista_Destino),
Ident_Anc_Destino,
BoolToInt(o_chb_ApresentarSimultaneo.Che
cked),
BoolToInt(o_chb_ApresentarNosTerminaisDo
Contexto.Checked))
(1458) o_tn_PastaChange(Sender, 1, Aux)
Chamado por:

```

Sistema: SASHE**Arquivo:** sashe\roteiros.pas**Procedimento:**

```

(94) To_fo_Roteiro.SelecionarNos
(p_IndiceInicial: LongInt;
p_ContextoInicial: String)
Diretório/Arquivo: roteiros.pas

```

Descrição:

Escolhe os nós que devem ser apresentados na janela Nós Selecionados de acordo com os atributos especificados

Chama:

```

(103) primeiro_no_contexto
(StrPascalC(p_ContextoInicial) ,
NomeDoNo)
(103) StrPascalC(p_ContextoInicial)
(106) tipo_no(NomeDoNo)
(111) recupera_funcao_didatica
(NomeDoNo, FuncaoDidatica)
(112) recupera_dificuldade (NomeDoNo,
Dificuldade)
(119) recupera_palavra_chave (NomeDoNo,
J, PalavraChave)
(120) StrPas (PalavraChave)
(134) AddChild (p_IndiceInicial,
StrPas (NomeDoNo) )
(134) StrPas (NomeDoNo)
(138) StrPas (FuncaoDidatica)
(141) AddChild(p_IndiceInicial,
StrPas (NomeDoNo) )
(141) StrPas (NomeDoNo)
(144) StrPas (FuncaoDidatica)
(146) AddChild(p_IndiceInicial,
StrPas (NomeDoNo) )
(146) StrPas (NomeDoNo)
(151) AddChild(p_IndiceInicial,
StrPas (NomeDoNo) )
(151) StrPas (NomeDoNo)
(155) AddChild(p_IndiceInicial,
StrPas (NomeDoNo) )

```

	<pre> (155) StrPas (NomeDoNo) (158) StrPas (FuncaoDidatica) (159) AddChild (p_IndiceInicial, StrPas (NomeDoNo)) (159) StrPas (NomeDoNo) (163) AddChild (p_IndiceInicial, StrPas (NomeDoNo)) (163) StrPas (NomeDoNo) (169) recupera_assunto (NomeDoNo, Assunto) (171) AddChild (p_IndiceInicial, StrPas (NomeDoNo)) (171) StrPas (NomeDoNo) (173) StrPas (Assunto) (174) AddChild (p_IndiceInicial, StrPas (NomeDoNo)) (174) StrPas (NomeDoNo) (175) SelecionarNos (IndiceAuxiliar, StrPas (NomeDoNo)) (175) StrPas (NomeDoNo) (177) proximo_no_contexto (StrPascalC (p_ContextoInicial), NomeDoNo) (177) StrPascalC (p_ContextoInicial) (180) AddChild (p_IndiceInicial, '...') (184) AddChild (p_IndiceInicial, '...') Chamado por: Roteiros.pas\ (94) To_fo_Roteiro.SelecionarNos (p_IndiceInicial: LongInt; p_ContextoInicial: String) Roteiros.pas\ (361) To_fo_Roteiro.o_b_Selecionar_NosClick (Sender: TObject) </pre>
Procedimento: (193) To_fo_Roteiro.FormCreate (Sender: TObject) Diretório/Arquivo: roteiros.pas	Descrição: Cria a janela para a criação de roteiros e atualiza os atributos necessários Chama: <pre> (207) primeiro_no_hiperbase (NomeDoNo) (208) tipo_no (NomeDoNo) (209) o_cb_Assunto.Items.Add (StrPas (NomeDoNo)) (209) StrPas (NomeDoNo) (210) proximo_no_hiperbase (NomeDoNo) (211) tipo_no (NomeDoNo) (212) o_cb_Assunto.Items.Add (StrPas (NomeDoNo)) (212) StrPas (NomeDoNo) (213) StrPas (NomeDoNo) (215) primeiro_no_trilha (NomeDoNo) (216) tipo_no (NomeDoNo) (217) o_cb_Assunto.Items.Add (StrPas (NomeDoNo)) (217) StrPas (NomeDoNo) (219) primeiro_no_trilha (StrPascalC ('Roteiro') , NomeDoNo, NomeNoLiberdade) (219) StrPascalC ('Roteiro') (220) DesenharRoteiro (StrPas (NomeNoLiberdade), StrPas (NomeDoNo)) (220) StrPas (NomeNoLiberdade) (220) StrPas (NomeDoNo) (221) proximo_no_trilha (StrPascalC ('Roteiro'), NomeDoNo, NomeNoLiberdade) (221) StrPascalC ('Roteiro') </pre>

	(222) DesenharRoteiro(StrPas (NomeNoLiberdade), StrPas (NomeDoNo)) (222) StrPas (NomeNoLiberdade) (222) StrPas (NomeDoNo) Chamado por:
Procedimento: (231) To_fo_Roteiro.FormClose (Sender: TObject; var Action: TCloseAction) Diretório/Arquivo: roteiros.pas	Descrição: Fecha a janela Roteiro e libera a memória Chama: Chamado por:
Procedimento: (242) To_fo_Roteiro.o_mp_InserirNoRoteiro Click (Sender: TObject) Diretório/Arquivo: roteiros.pas	Descrição: Implementa a operação Inserir no Roteiro, ativando a janela para a escolha do nó liberdade antes da inserção Chama: (255) ListaDeNos.Insert (0, Items[SelectedItem].Text) (256) primeiro_contexto_pai (StrPascalC(Items[SelectedItem].Text), Contexto_Pai) (256) StrPascalC(Items[SelectedItem].Text) (258) ListaDeNos.Insert(0, StrPas (Contexto_Pai)) (258) StrPas (Contexto_Pai) (259) primeiro_contexto_pai(Contexto_Pai, Contexto_Pai) (260) StrPas (Contexto_Pai) (263) o_ol_NoLiberdade.AddChild (I, ListaDeNos.Strings[I]) (271) tipo_no (StrPascalC(Items[SelectedItem].Text)) (271) StrPascalC(Items[SelectedItem].Text) (281) DesenharRoteiro (Nome_No_Liberdade, Items[SelectedItem].Text) Chamado por:
Procedimento: (289) To_fo_Roteiro.DesenharRoteiro (p_NoLiberdade, p_NoSelecioneado: String) Diretório/Arquivo: roteiros.pas	Descrição: Insere o nó selecionado e seu nó liberdade na lista de nós da trilha e habilita os comandos de edição da trilha Chama: (291) _Lista_Nos.Add(p_NoLiberdade) (292) _Lista_Nos.Add(p_NoSelecioneado) Chamado por: Roteiros.pas\ (193) To_fo_Roteiro.FormCreate (Sender: TObject) Roteiros.pas\ (242) To_fo_Roteiro.o_mp_InserirNoRoteiroClick (Sender: TObject)
Procedimento: (303) To_fo_Roteiro.o_mp_Apresentar_ConteudoClick(Sender: TObject) Diretório/Arquivo: roteiros.pas	Descrição: Implementa o comando Apresentar Conteúdo do Nó apresentado dentro da janela de edição de roteiros (NUNCA É HABILITADO!) Chama: (308) tipo_no(StrPascalC(Items[SelectedItem].Text)) (308) StrPascalC(Items[SelectedItem].Text) (310) o_fo_Nos.Visualizar No Terminal

	<pre> ('t', StrPascalC(Items[SelectedItem].Text)) (310) StrPascalC(Items[SelectedItem].Text) (311) o_fo_Nos.Visualizar_No_Terminal ('v', StrPascalC(Items[SelectedItem].Text)) (311) StrPascalC(Items[SelectedItem].Text) (312) o_fo_Nos.Visualizar_No_Terminal ('a', StrPascalC(Items[SelectedItem].Text)) (312) StrPascalC(Items[SelectedItem].Text) (313) o_fo_Nos.Visualizar_No_Terminal ('i', StrPascalC(Items[SelectedItem].Text)) (313) StrPascalC(Items[SelectedItem].Text) Chamado por: </pre>
Procedimento: (321) To_fo_Roteiro.o_mp_Visualizar_Grafo Click(Sender: TObject) Diretório/Arquivo: roteiros.pas	Descrição: Implementa o comando Visualizar Browser Gráfico apresentado dentro da janela de edição de roteiros Chama: (324) o_fo_Principal.Apresentar_No(StrPascalC(Items[SelectedItem].Text), 0) (324) StrPascalC(Items[SelectedItem].Text) Chamado por:
Procedimento: (330) To_fo_Roteiro.o_cb_Palavra_ChaveClick(Sender: TObject) Diretório/Arquivo: roteiros.pas	Descrição: Escolhe uma palavra da lista de palavras-chaves do combobox Chama: (333) o_lb_Palavra_Chave.Items.Add(Items.Strings[ItemIndex]) Chamado por:
Procedimento: (339) To_fo_Roteiro.o_cb_Palavra_ChaveKeyPress(Sender: TObject; var Key: Char); Diretório/Arquivo: roteiros.pas	Descrição: Adiciona a palavra a lista de palavras-chave Chama: (343) o_lb_Palavra_Chave.Items.Add(o_cb_Palavra_Chave.Text) Chamado por:
Procedimento: (351) To_fo_Roteiro.o_mp_Remover_PalavraClick(Sender: TObject) Diretório/Arquivo: roteiros.pas	Descrição: Remove a palavra-chave da lista caso o comando remover for selecionado Chama: (355) o_lb_Palavra_Chave.Items.Delete(ItemIndex) Chamado por:
Procedimento: (361) To_fo_Roteiro.o_b_Selecionar_NosClick(Sender: TObject) Diretório/Arquivo: roteiros.pas	Descrição: Implementa a funcionalidade do botão Selecionar nós, setando os atributos selecionados e chamando o procedimento SelecionarNos() Chama: (369) o_cb_Dificuldade.Items.IndexOf(o_cb_Dificuldade.Text) (376) SelecionarNos(0,

	o_cb Assunto.Text) Chamado por:
Procedimento: (383) To_fo_Roteiro.o_b_Criar_RoteiroClick (Sender: TObject) Diretório/Arquivo: roteiros.pas	Descrição: Limpa a trilha (se já existir uma) e cria uma nova trilha, inserindo na trilha todos os nós selecionados Chama: (388) limpa_trilha(StrPascalC('Roteiro')) (388) StrPascalC('Roteiro') (389) retira_da_hiperbase(StrPascalC('Roteiro')) (389) StrPascalC('Roteiro') (391) cria_trilha(StrPascalC(o_e_NomeDoRoteiro.Text)) (391) StrPascalC(o_e_NomeDoRoteiro.Text) (392) insere_no_contexto(StrPascalC(g_NomeDaHiperbase), StrPascalC(o_e_NomeDoRoteiro.Text)) (392) StrPascalC(g_NomeDaHiperbase) (393) StrPascalC(o_e_NomeDoRoteiro.Text) (396) insere_na_trilha (StrPascalC(o_e_NomeDoRoteiro.Text), StrPascalC(Lista_Nos.Strings[I + 1]), StrPascalC(Lista_Nos.Strings[I])) (396) StrPascalC(o_e_NomeDoRoteiro.Text) (397) StrPascalC(Lista_Nos.Strings[I + 1]) (397) StrPascalC(Lista_Nos.Strings[I]) Chamado por:
Procedimento: (411) To_fo_Roteiro.o_mp_Limpar_TrilhaClick (Sender: TObject) Diretório/Arquivo: roteiros.pas	Descrição: Limpa a trilha em edição Chama: Chamado por:
Procedimento: (428) To_fo_Roteiro.o_o_Nos_SelecionadosClick (Sender: TObject) Diretório/Arquivo: roteiros.pas	Descrição: Faz com que ao se clicar um nó apareça a etiqueta com o tipo do nó, a função didática e a dificuldade Chama: (435) recupera_assunto (StrPascalC(Items[SelectedItem].Text), v_sc_Assunto) (435) StrPascalC(Items[SelectedItem].Text) (436) recupera_funcao_didatica (StrPascalC(Items[SelectedItem].Text), v_sc_FuncaoDidatica) (436) StrPascalC(Items[SelectedItem].Text) (437) recupera_dificuldade (StrPascalC(Items[SelectedItem].Text), v_i_Dificuldade) (437) StrPascalC(Items[SelectedItem].Text) (443) tipo_no (StrPascalC(Items[SelectedItem].Text)) (443) StrPascalC(Items[SelectedItem].Text) (446) StrPas (v_sc_Assunto) (448) StrPas(v_sc_FuncaoDidatica) (451) StrPas(v_sc_FuncaoDidatica) (454) StrPas(v_sc_FuncaoDidatica)

	(457) StrPas(v_sc_FuncaoDidatica) Chamado por:
Procedimento: (467) To_fo_Roteiro.o_t_TemporizadorTimer (Sender: TObject) Diretório/Arquivo: roteiros.pas	Descrição: É um timer acionado a cada 100 milisegundos que interrompe o controle normal do aplicativo e executando o código especificado no corpo do procedimento Chama: (475) tipo_no (StrPascalC(Items[SelectedItem].Text)) (475) StrPascalC(Items[SelectedItem].Text) Chamado por:
Procedimento: (498) To_fo_Roteiro.o_cb_Funcao_DidaticaC hange (Sender: TObject) Diretório/Arquivo: roteiros.pas	Descrição: Se nenhuma função didática foi especificada então o atributo é setado com 'TODAS' Chama: Chamado por:
Procedimento: (504) To_fo_Roteiro.o_cb_DificuldadeChang e (Sender: TObject) Diretório/Arquivo: roteiros.pas	Descrição: Se nenhum grau de dificuldade foi especificado então o atributo é setado com 'TODAS' Chama: Chamado por:
Procedimento: (510) To_fo_Roteiro.o_mp_Apagar_NoClick (Sender: TObject) Diretório/Arquivo: roteiros.pas	Descrição: Apaga um nó da trilha Chama: (512) _Lista_Nos.Delete (_NoSelecionado*2) (513) _Lista_Nos.Delete (_NoSelecionado*2) Chamado por:
Procedimento: (525) To_fo_Roteiro.o_dg_DesenhaRoteiro(S ender: TObject; Col, Row: Longint; Rect: TRect; State: TGridDrawState) Diretório/Arquivo: roteiros.pas	Descrição: Desenha na região da trilha os nós que a ela pertencem e seus graus de liberdade Chama: (537) tipo_no(StrPascalC(_Lista_Nos.Strings[I+1])) (537) StrPascalC(_Lista_Nos.Strings[I+1]) (552) o_dg_Trilha.Canvas.CopyRect(Regiao2, o_im_Nos.Canvas, Regiao1) (553) o_dg_Trilha.Canvas.TextOut(Regiao2.Left, 0, IntToStr(J) + ' ') (553) IntToStr(J) Chamado por:
Procedimento: (559) To_fo_Roteiro.o_dg_NoSelecionado(Se nder: TObject; Col, Row: Longint; var CanSelect: Boolean) Diretório/Arquivo: roteiros.pas	Descrição: Seleciona um nó pertencente a trilha Chama: (572) tipo_no(StrPascalC(_Lista_Nos.Strings[I+1])) (572) StrPascalC(_Lista_Nos.Strings[I+1]) (575) recupera funcao didatica

	<pre> (StrPascalC(_Lista_Nos.Strings[I+1]), v_sc_FuncaoDidatica) (575) StrPascalC(_Lista_Nos.Strings[I+1]) (576) recupera_dificuldade (StrPascalC(_Lista_Nos.Strings[I+1]), v_i_Dificuldade) (576) StrPascalC(_Lista_Nos.Strings[I+1]) (584) StrPas(v_sc_FuncaoDidatica) Chamado por: </pre>
Procedimento: (593) To_fo_Roteiro.o_mp_MudarNoLiberdade Click (Sender: TObject) Diretório/Arquivo: roteiros.pas	Descrição: Abre uma janela para a edição do nó liberdade Chama: (604) ListaDeNos.Insert(0, _Lista_Nos.Strings[_NoSelecioneado*2+1]) (605) primeiro_contexto_pai (StrPascalC(_Lista_Nos.Strings[_NoSelecioneado*2+1]), Contexto_Pai) (605) StrPascalC(_Lista_Nos.Strings[_NoSelecioneado*2+1]) (607) ListaDeNos.Insert(0, StrPas(Contexto_Pai)) (607) StrPas(Contexto_Pai) (608) primeiro_contexto_pai(Contexto_Pai, Contexto_Pai) (609) StrPas(Contexto_Pai) (612) o_ol_NoLiberdade.AddChild(I, ListaDeNos.Strings[I]) Chamado por:

Sistema: SASHE Arquivo: sashe\ sashe.pas	
Procedimento: To_fo_Sashe.FormCreate(Sender: TObject); Diretório/Arquivo: sashe.pas	Descrição: Procedimentos do Formulário principal Chama: Chamado por:
Procedimento: To_fo_Sashe.FormShow(Sender: TObject); Diretório/Arquivo: sashe.pas	Descrição: Procedimentos do Formulário principal Chama: Chamado por:
Procedimento: To_fo_Sashe.Image1MouseMove(Sender: TObject; Shift: TShiftState; X, Y: Integer); Diretório/Arquivo: sashe.pas	Descrição: Procedimentos da Imagem Principal Chama: Chamado por:
Procedimento: To_fo_Sashe.o_p_ProfessorClick(Send er: TObject); Diretório/Arquivo: sashe.pas	Descrição: Procedimentos do Picture Professor Chama: Chamado por:
Procedimento: To_fo_Sashe.o_im_RegiaoProfessorMou seMove(Sender: TObject; Shift: TShiftState; X, Y: Integer); Diretório/Arquivo: sashe.pas	Descrição: Procedimentos do Picture Professor Chama: Chamado por:
Procedimento: To_fo_Sashe.o_p_EstudanteClick(Send	Descrição: Procedimentos do Picture Estudante

er: TObject); Diretório/Arquivo: sashe.pas	Chama: Chamado por:
Procedimento: To_fo_Sashe.o_im_RegiaoEstudanteMou seMove(Sender: TObject; Shift: TShiftState; X, Y: Integer); Diretório/Arquivo: sashe.pas	Descrição: Procedimentos do Picture Estudante Chama: Chamado por:
Procedimento: To_fo_Sashe.o_im_SaidaClick(Sender: TObject); Diretório/Arquivo: sashe.pas	Descrição: Procedimentos do Picture Saída Chama: Chamado por:

Sistema: SASHE Arquivo: sashe\ som.pas	
Procedimento: (35) To_fo_Audio.FormClose(Sender: TObject; var Action: TCloseAction); Diretório/Arquivo: som.pas	Descrição: Fecha uma janela de áudio e libera a memória Chama: (37) o_mp_Audio.Close (38) Action := caFree Chamado por:
Procedimento: (46) To_fo_Audio.FormCreate(Sender: TObject); Diretório/Arquivo: som.pas	Descrição: Cria uma janela para a apresentação de um arquivo de áudio Chama: Chamado por:
Procedimento: (61) To_fo_Audio.o_mp_AudioNotify(Sender: TObject); Diretório/Arquivo: som.pas	Descrição: Procedimentos do MediaPlayer Áudio Chama: (64) o_mp_Audio.Previous Chamado por:

Sistema: SASHE Arquivo: sashe\video.pas	
Procedimento: (47) To_fo_Video.FormCreate (Sender: TObject) Diretório/Arquivo: video.pas	Descrição: Cria um form para a apresentação de vídeo Chama: Chamado por:
Procedimento: (61) To_fo_Video.FormClose (Sender: TObject; var Action: TCloseAction) Diretório/Arquivo: video.pas	Descrição: Fecha o form e libera memória Chama: Chamado por:
Procedimento: (70) To_fo_Video.o_mp_VideoClick (Sender: TObject; Button: TMPBtnType; var DoDefault: Boolean) Diretório/Arquivo: video.pas	Descrição: Caso o botão Stop seja clicado o vídeo volta ao início Chama: Chamado por:
Procedimento: (80) To_fo_Video.o_pm_PlayClick (Sender: TObject) Diretório/Arquivo: video.pas	Descrição: Dispara a execução do vídeo Chama: Chamado por:
Procedimento: (88) To_fo_Video.o_pm_StopClick (Sender: TObject) Diretório/Arquivo: video.pas	Descrição: Pára a execução do vídeo Chama: Chamado por:

Procedimento: (96) To_fo_Video.o_pm_PauseClick (Sender: TObject) Diretório/Arquivo: video.pas	Descrição: Dá uma pausa na execução do vídeo Chama: Chamado por:
Procedimento: (104) To_fo_Video.o_pm_NextClick (Sender: TObject) Diretório/Arquivo: video.pas	Descrição: Vai para o fim o vídeo Chama: Chamado por:
Procedimento: (112) To_fo_Video.o_pm_PreviousClick (Sender: TObject) Diretório/Arquivo: video.pas	Descrição: Volta ao início do vídeo Chama: Chamado por:

3.5 Quadro Índice de Procedimentos

Esse quadro apresenta todos os procedimentos da implementação do sistema em ordem alfabética, com as respectivas localizações (arquivo e diretório).

Para a elaboração desse quadro foram utilizados todos os quadros de chamadas obtidos anteriormente.

Os quadros índice de procedimentos obtidos são apresentados a seguir.

Sistema: SASHE - Sistema de Autoria e Suporte Hiperídia ao Ensino OS arquivos referenciados neste índice encontram-se no diretório sashe\	
Procedimentos	Diretório\Arquivo
Ler Ajuda	funcoes.pas
Ler ArquivoINI	funcoes.pas
StrPascalC	funcoes.pas
To fo Ajuda.FormCreate	ajuda.pas
To fo Ajuda.o bu InserirClick	ajuda.pas
To fo Ajuda.o bu GravarClick	ajuda.pas
To fo Audio.FormClose	som.pas
To fo Audio.FormCreate	som.pas
To fo Audio.o mp AudioNotify	som.pas
To fo BrowserEstrutural.Atualiza Browser	browser.pas
To fo BrowserEstrutural.FormClose	browser.pas
To fo BrowserEstrutural.Ler Contexto	browser.pas
To fo BrowserEstrutural.o bu AbrirJanelaClick	browser.pas
To fo BrowserEstrutural.o bu ApresentarConteudoClick	browser.pas
To fo BrowserEstrutural.o ol BrowserClick	browser.pas
To fo BrowserEstrutural.o ti TemporizadorTimer	browser.pas
To fo BrowserGrafico.FormClose	bgrafo.pas
To fo BrowserGrafico.FormResize	bgrafo.pas
To fo Controles.o bu AjudaClick	controle.pas
To fo Controles.o bu AnteriorClick	controle.pas
To fo Controles.o bu BibliografiaClick	controle.pas
To fo Controles.o bu ContextoClick	controle.pas
To fo Controles.o bu DificilClick	controle.pas
To fo Controles.o bu ExerciciosClick	controle.pas
To fo Controles.o bu FacilClick	controle.pas
To fo Controles.o bu GlossarioClick	controle.pas
To fo Controles.o bu HistoricoClick	controle.pas
To fo Controles.o bu MaisInformacoesClick	controle.pas

To fo Controles.o bu OndeEstouClick	controle.pas
To fo Controles.o bu ProximoClick	controle.pas
To fo Controles.o bu ProximoTopicoClick	controle.pas
To fo Controles.o bu TopicoAnteriorClick	controle.pas
To fo CriarElos.Inicializa Browser	criarelo.pas
To fo CriarElos.Ler Contexto Destino	criarelo.pas
To fo CriarElos.Ler Contexto Origem	criarelo.pas
To fo CriarElos.No Origem Selecionado	criarelo.pas
To fo CriarElos.No Destino Selecionado	criarelo.pas
To fo CriarElos.o cb AncoraOrigemClick	criarelo.pas
To fo CriarElos.o cb AncoraDestinoClick	criarelo.pas
To fo CriarElos.o ol BrowserDestinoClick	criarelo.pas
To fo CriarElos.o ol BrowserDestinoMouseUp	criarelo.pas
To fo CriarElos.o ol BrowserOrigemClick	criarelo.pas
To fo CriarElos.o ol BrowserOrigemMouseUp	criarelo.pas
To fo CriarNo.FormActivate	criar no.pas
To fo CriarNo.FormCloseQuery	criar no.pas
To fo CriarNo.FormCreate	criar no.pas
To fo CriarNo.o bu OkClick	criar no.pas
To fo CriarNo.o bu ProcurarArquivoClick	criar no.pas
To fo CriarNo.o cb PalavraChaveClick	criar no.pas
To fo CriarNo.o gb TipoDeNosTerminaisClick	criar no.pas
To fo CriarNo.o gb TipoDoNoClick	criar no.pas
To fo CriarNo.o mp RemoverPalavraClick	criar no.pas
To fo EditarNo.FormCloseQuery	edit no.pas
To fo EditarNo.FormCreate	edit no.pas
To fo EditarNo.o bu EditarConteudoClick	edit no.pas
To fo EditarNo.o bu OkClick	edit no.pas
To fo EditarNo.o bu ProcurarArquivoClick	edit no.pas
To fo EditarNo.o cb PalavraChaveClick	edit no.pas
To fo EditarNo.o cb PalavraChaveKeyPress	edit no.pas
To fo EditarNo.o mp RemoverPalavraClick	edit no.pas
To fo Glossario.FormClose	glossari.pas
To fo Glossario.FormCreate	glossari.pas
To fo Glossario.o dbn ControlesClick	glossari.pas
To fo Glossario.o ed PalavraABuscarKeyPress	glossari.pas
To fo Imagem.FormClose	imagem.pas
To fo Imagem.FormCreate	imagem.pas
To fo ListaDeNos.FormClose	listanos.pas
To fo ListaDeNos.FormCreate	listanos.pas
To fo ListaDeNos.o lb NosDb1Click	listanos.pas
To fo MaisInformacoes.FormCreate	maisinfo.pas
To fo MaisInformacoes.FormClose	maisinfo.pas
To fo MaisInformacoes.Ler Contexto	maisinfo.pas
To fo MaisInformacoes.o lb NosSelecionadosDb1Click	maisinfo.pas
To Fo Nos.ApresentarNoTrilha	nos.pas
To fo Nos.FormActivate	nos.pas
To fo Nos.FormClose	nos.pas
To fo Nos.FormCreate	nos.pas
To fo Nos.FormDeactivate	nos.pas
To fo Nos.Ler Elos	nos.pas
To fo Nos.Ler Nos	nos.pas
To fo Nos.o bu ApagarEloClick	nos.pas
To fo Nos.o bu BrowserGraficoClick	nos.pas
To fo Nos.o bu CriarEloClick	nos.pas
To fo Nos.o bu EditarEloClick	nos.pas
To fo Nos.o lb ListaElosClick	nos.pas
To fo Nos.o lb NosClick	nos.pas
To fo Nos.o lb NosDragDrop	nos.pas
To fo Nos.o lb NosDragOver	nos.pas
To fo Nos.o lb NosMouseDown	nos.pas
To fo Nos.o lb NosMouseUp	nos.pas
To fo Nos.o ti Temporizador1Timer	nos.pas
To fo Nos.o tn PastaChange	nos.pas

To fo Nos.o pm AbrirJanelaClick	nos.pas
To fo Nos.o pm ApagarNoClick	nos.pas
To fo Nos.o pm ApresentarConteudoDoNoClick	nos.pas
To fo Nos.o pm BrowserClick	nos.pas
To fo Nos.o pm CriarNoClick	nos.pas
To fo Nos.o pm EditarNoClick	nos.pas
To fo Nos.o pm RetirarNoClick	nos.pas
To fo Nos.o pm SubirContextoClick	nos.pas
To fo Nos.o pm VisualizarContextoClick	nos.pas
To fo Nos.Visualizar No Terminal	nos.pas
To fo Principal.AppMessage	hip prin.pas
To fo Principal.Apresentar Browser Grafico	hip prin.pas
To fo Principal.Apresentar No	hip prin.pas
To fo Principal.FormActivate	hip prin.pas
To fo Principal.FormCloseQuery	hip prin.pas
To fo Principal.FormCreate	hip prin.pas
To fo Principal.Ler DLL	hip prin.pas
To fo Principal.o bu AbrirHiperbaseClick	hip prin.pas
To fo Principal.o bu BrowserGraficoClick	hip prin.pas
To fo Principal.o bu CriarHiperbaseClick	hip prin.pas
To fo Principal.o bu GlossarioClick	hip prin.pas
To fo Principal.o bu GravarHiperbaseClick	hip prin.pas
To fo Principal.o bu RoteiroClick	hip prin.pas
To fo Principal.o im LixoDragDrop	hip prin.pas
To fo Principal.o im LixoDragOver	hip prin.pas
To fo Principal.o mn CascataClick	hip prin.pas
To fo Principal.o mn ConteudoClick	hip prin.pas
To fo Principal.o mn LadoALadoClick	hip prin.pas
To fo Principal.o mn OrganizarIconesClick	hip prin.pas
To fo Principal.o mn SairClick	hip prin.pas
To fo Roteiro.DesenharRoteiro	roteiros.pas
To fo Roteiro.FormClose	roteiros.pas
To fo Roteiro.FormCreate	roteiros.pas
To fo Roteiro.SelecionarNos	roteiros.pas
To fo Roteiro.o b Criar RoteiroClick	roteiros.pas
To fo Roteiro.o b Selecionar NosClick	roteiros.pas
To fo Roteiro.o cb DificuldadeChange	roteiros.pas
To fo Roteiro.o cb Funcao DidaticaChange	roteiros.pas
To fo Roteiro.o cb Palavra ChaveClick	roteiros.pas
To fo Roteiro.o cb Palavra ChaveKeyPress	roteiros.pas
To fo Roteiro.o dg DesenhaRoteiro	roteiros.pas
To fo Roteiro.o dg NoSelecioneado	roteiros.pas
To fo Roteiro.o mp Apagar NoClick	roteiros.pas
To fo Roteiro.o mp Apresentar ConteudoClick	roteiros.pas
To fo Roteiro.o mp InserirNoRoteiroClick	roteiros.pas
To fo Roteiro.o mp Limpar TrilhaClick	roteiros.pas
To fo Roteiro.o mp MudarNoLiberdadeClick	roteiros.pas
To fo Roteiro.o mp Remover PalavraClick	roteiros.pas
To fo Roteiro.o mp Visualizar GrafoClick	roteiros.pas
To fo Roteiro.o o Nos SeleccionadosClick	roteiros.pas
To fo Roteiro.o t TemporizadorTimer	roteiros.pas
To fo Texto.FormClose	atexto.pas
To fo Texto.FormCreate	atexto.pas
To fo Texto.Ler Resposta	atexto.pas
To fo Texto.o hv TextoHotSpotClick	atexto.pas
To fo Texto.Procurar Elo	atexto.pas
To fo Trilha.FormClose	atrilha.pas
To fo Trilha.FormCreate	atrilha.pas
To fo Video.FormClose	video.pas
To fo Video.FormCreate	video.pas
To fo Video.o mp VideoClick	video.pas
To fo Video.o pm NextClick	video.pas
To fo Video.o pm PauseClick	video.pas
To fo Video.o pm PlayClick	video.pas

To fo Video.o pm PreviousClick	video.pas
To fo Video.o pm StopClick	video.pas
Txt_Html	funcoes.pas

Sistema:

SASHE - Sistema de Autoria e Suporte Hiperímia ao Ensino

OS arquivos referenciados neste índice encontram-se no diretório

sashe\hyperprop

Procedimentos	Diretório\Arquivo
ancora_audio::ancora_audio(int ind):ancora(ind)	anc_aud.cpp
ancora_audio::ancora_audio(int ind, int ide):ancora(ind, ide)	anc_aud.cpp
ancora::ancora(int Ind):entidade(Ind)	ancora.cpp
ancora::ancora(int Ind, int ide):entidade(Ind)	ancora.cpp
ancora_contexto::ancora_contexto(int ind):ancora(ind)	anc_cont.cpp
ancora_contexto::ancora_contexto(int ind,int ide,int id no):ancora(ind, ide)	anc_cont.cpp
ancora_contexto *contexto::contem ancora(int id no)	contexto.cpp
ancora_contexto *contexto::cria_ancora_contexto(int ide, int id no)	contexto.cpp
ancora_executavel::ancora_executavel(int ind):ancora(ind)	anc_exec.cpp
ancora_executavel::ancora_executavel(int ind, int ide):ancora(ind, ide)	anc_exec.cpp
ancora_imagem::ancora_imagem(int ind):ancora(ind)	anc_ima.cpp
ancora_imagem::ancora_imagem(int ind, int ide):ancora(ind, ide)	anc_ima.cpp
ancora *No::primeira ancora()	No.cpp
ancora *No::proxima ancora()	No.cpp
ancora * No::retorna ancora(int ind)	No.cpp
ancora *No::ultima ancora()	No.cpp
ancora_tabela::ancora_tabela(int ind):ancora(ind)	anc_tab.cpp
ancora_tabela::ancora_tabela(int ind, int ide):ancora(ind, ide)	anc_tab.cpp
ancora_texto::ancora_texto(int ind):ancora(ind)	anc_tex.cpp
ancora_texto::~~ancora_texto()	anc_tex.cpp
ancora_texto::ancora_texto(int ind, int ide, char *pal, int inicio, int fim):ancora(ind, ide)	anc_tex.cpp
ancora_trilha::ancora_trilha(int ind):ancora(ind)	anc_tri.cpp
ancora_trilha::ancora_trilha(int ind,int ide):ancora(ind, ide)	anc_tri.cpp
ancora_video::ancora_video(int ind):ancora(ind)	anc_vid.cpp
ancora_video::ancora_video(int ind, int ide):ancora(ind, ide)	anc_vid.cpp
baseprivada::baseprivada(char *nome):contexto(nome)	basepriv.cpp
baseprivada::baseprivada(int id):contexto(id)	basepriv.cpp
BOOL Cframe::CreateToolBar()	frame.cpp
BOOL CGrafoApp::InitInstance()	grafo.cpp
BOOL CGrafoObj::OnNewDocument()	grafoob.cpp
BOOL CGrafoView::SetFont(const LOGFONT* pLogFont)	grafovw.cpp
BOOL CTipTop::SetFont(const LOGFONT* pLogFont)	Tiptop.cpp
BOOL DisplayPicture (CDC *, int x = -*, int y = -*)	Pics.h
BOOL FAR PASCAL export EsconderJanela()	grafoapi.h
BOOL FAR PASCAL export InicializarGrafo()	grafoapi.h
BOOL FAR PASCAL export MostrarJanela(HWND hWndParent)	grafoapi.h
BOOL FAR PASCAL export TranslateMsg(LPMSG lpMsg)	grafoapi.h
CAboutDlg::CAboutDlg():CDialog(CAboutDlg::IDD)	grafo.cpp
Cavi::BOOL FrameBack (void)	Video.h
Cavi::BOOL FrameForward (void)	Video.h
Cavi::BOOL Forward (void)	Video.h
Cavi::BOOL Open (CWnd *, CString)	Video.h
Cavi::BOOL Rewind (void)	Video.h
Cavi::BOOL SetPosition (void)	Video.h

Cavi::BOOL Start (void)	Video.h
Cavi::BOOL Stop (void)	Video.h
CAvi::CAvi ()	Video.h
Cavi::~~CAvi ()	Video.h
Cavi::CString GetErrorString (void)	Video.h
CFrame::CFrame()	frame.cpp
CFrame::~~CFrame()	frame.cpp
CGrafoApp::CGrafoApp()	grafo.cpp
CGrafoObj::CGrafoObj()	grafoob.cpp
CGrafoObj::~~CGrafoObj()	grafoob.cpp
CGrafoObj::CSize GetDocSize()	grafoob.h
CGrafoObj* CgrafoView::GetDocument()	grafovw.cpp
CgrafoObj::elo *PrimeiroElo(char *nome)	grafoob.h
CgrafoObj::elo *ProximoElo(char *nome, elo *e)	grafoob.h
CgrafoObj::No *PrimeiroNo(char *nome)	grafoob.h
CgrafoObj::No *ProximoNo(char *nome)	grafoob.h
CgrafoObj::No *RetornaPtNo(int id)	grafoob.h
CGrafoView::CGrafoView()	grafovw.cpp
CGrafoView::~~CGrafoView()	grafovw.cpp
CGrafoView::CgrafoObj* GetDocument()	grafovw.h
CGrafoView::Create(LPCSTR lpszClassName, LPCSTR lpszWindowName, DWORD dwStyle, const RECT& rect, CWnd* pParentWnd, UINT nID, CcreateContext* pContext)	grafovw.cpp
char ancora texto::*obtem ancora()	anc tex.h
char composicao::*recupera assunto()	Composic.h
char elo::*obtem aninhamento destino()	elo.h
char elo::*obtem aninhamento origem()	elo.h
char elo::*obtem destino()	elo.h
char elo::*obtem origem()	elo.h
char grafo::*GetArquivo()	grafos.h
char no::ObtemEstado()	No.h
char no::*ObtemNome()	No.h
char no terminal::*recupera funcao didatica()	Terminal.h
char *no terminal::recupera nome arquivo()	Terminal.cpp
char no terminal::*recupera palavras chaves(int pos)	Terminal.h
char no terminal::*recupera tipo()	Terminal.h
Cmidi::BOOL IsPlaying (void)	Sound.h
Cmidi::BOOL Play (void)	Sound.h
Cmidi::BOOL Restart (void)	Sound.h
Cmidi::CMidi (CString Name)	Sound.h
Cmidi::~~CMidi ()	Sound.h
Cmidi::CString GetErrorString (void)	Sound.h
composicao::composicao(char *nome no):No(nome no)	composic.cpp
composicao::composicao(int id):No(id)	composic.cpp
contexto::contexto(char *nome no):composicao(nome no)	contexto.cpp
contexto::contexto(int id):composicao(id)	contexto.cpp
contextousuario::contextousuario(char *nome no):contexto(nome no)	contuser.cpp
contextousuario::contextousuario(int id):contexto(id)	contuser.cpp
contextoversoes::contextoversoes(No *n*, No *n*, char *nome no):contexto(nome no)	contvers.cpp
contextoversoes::contextoversoes(int id):contexto(id)	contvers.cpp
coordArco::CoordArco()	elo.h
Cpics::CPics ()	Pics.h
Cpics::~~CPics ()	Pics.h
Cpics::CPics (char *)	Pics.h
Cpics::virtual BOOL LoadPicture (CDC *, int Flags = *)	Pics.h
Cpics::virtual BOOL SavePicture (char *Filename = NULL, int PictureType = -*)	Pics.h
CTipTop::CTipTop(CWnd* pParentWnd)	Tiptop.cpp
CTipTop::~~CTipTop()	Tiptop.cpp
Cwave::CString GetErrorString (void)	Sound.h
Cwave::CWave (CString Name, int Type = WT_DISK)	Sound.h
Cwave::~~CWave ()	Sound.h

Cwave::BOOL CanRecord (void)	Sound.h
Cwave::BOOL PlayAsync (void)	Sound.h
Cwave::BOOL PlayLoop (void)	Sound.h
Cwave::BOOL Play (void)	Sound.h
Cwave::BOOL SaveRecording (CString)	Sound.h
Cwave::BOOL StartRecording (void)	Sound.h
Cwave::BOOL StopRecording (void)	Sound.h
elo::ancora *obtem ancora destino()	elo.h
elo::ancora *obtem ancora origem()	elo.h
elo * composicao::retorna elo(int i)	composic.cpp
elo::elo(int id):entidade(id)	elo.cpp
elo::elo(int id, No *orig, char *ani_orig, int ind_or, No *dest, char *ani_dest, int ind_dest, int simul, int mostr):entidade(id)	elo.cpp
elo *FAR PASCAL _export primeiro_ponteiro_elo(char *nome contexto)	hipger.cpp
elo *FAR PASCAL _export proximo_ponteiro_elo(char *nome contexto, elo *pelo)	hipger.cpp
elo *FAR PASCAL primeiro_ponteiro_elo(char *)	hipdll.h
elo *FAR PASCAL proximo_ponteiro_elo(char *, elo *)	hipdll.h
elo::No *obtem no destino()	elo.h
elo::No *obtem no origem()	elo.h
entidade::entidade(int id)	entidade.cpp
extern "C" BOOL FAR PASCAL _export EsconderJanela()	grafo.cpp
extern "C" BOOL FAR PASCAL _export InicializarGrafo()	grafo.cpp
extern "C" BOOL FAR PASCAL _export MostrarJanela(HWND hWndParent)	grafo.cpp
extern "C" BOOL FAR PASCAL _export TranslateMsg(LPMSG lpMsg)	grafo.cpp
extern "C" int FAR PASCAL _export anterior_no_trilha(char *nome trilha, char *nome no, char *nome liberdade)	hipger.cpp
extern "C" void FAR PASCAL _export ApresentaContexto(char *no)	grafo.cpp
extern "C" int FAR PASCAL _export ativa_hiperbase(char *nome)	hipger.cpp
extern "C" int FAR PASCAL _export cria_ancora_contexto(char *nome no, int ident, char *no ancorado)	hipger.cpp
extern "C" int FAR PASCAL _export cria_ancora_texto(char *nome no, int ident, char *palavra, int inicio, int fim)	hipger.cpp
extern "C" int FAR PASCAL _export cria_audio(char *nome)	hipger.cpp
extern "C" int FAR PASCAL _export cria_base_privada(char *nome)	hipger.cpp
extern "C" int FAR PASCAL _export cria_contexto(char *nome)	hipger.cpp
extern "C" int FAR PASCAL _export cria_elo(char *nome_contexto, int id_elo, char *origem, int anc_origem, char *destino, int anc_destino, int simultaneo, int mostra conteudo)	hipger.cpp
extern "C" int FAR PASCAL _export cria_executavel(char *nome)	hipger.cpp
extern "C" int FAR PASCAL _export cria_imagem(char *nome)	hipger.cpp
extern "C" int FAR PASCAL _export cria_tabela(char *nome)	hipger.cpp
extern "C" int FAR PASCAL _export cria_texto(char *nome)	hipger.cpp
extern "C" int FAR PASCAL _export cria_trilha(char *nome)	hipger.cpp
extern "C" int FAR PASCAL _export cria_versao(char *nome_no, char *nome no versao)	hipger.cpp
extern "C" int FAR PASCAL _export cria_video(char *nome)	hipger.cpp
extern "C" int FAR PASCAL _export insere_na_posicao(char *nome_trilha, char *nome_no, char *no_liberdade, int posicao)	hipger.cpp
extern "C" int FAR PASCAL _export insere_na_trilha(char *nome_trilha, char *nome_no, char *no_liberdade)	hipger.cpp
extern "C" int FAR PASCAL _export insere_no_contexto(char *nome cont, char *nome no)	hipger.cpp
extern "C" int FAR PASCAL _export insere_nome_arquivo_audio(char *nome audio, char *arquivo)	hipger.cpp

extern "C" int FAR PASCAL _export insere_nome_arquivo_executavel(char *nome_executavel, char *arquivo)	hipger.cpp
extern "C" int FAR PASCAL _export insere_nome_arquivo_imagem(char *nome_imagem, char *arquivo)	hipger.cpp
extern "C" int FAR PASCAL _export insere_nome_arquivo_tabela(char *nome_tabela, char *arquivo)	hipger.cpp
extern "C" int FAR PASCAL _export insere_nome_arquivo_texto(char *nome_texto, char *arquivo)	hipger.cpp
extern "C" int FAR PASCAL _export insere_nome_arquivo_video(char *nome_video, char *arq_video)	hipger.cpp
extern "C" int FAR PASCAL _export limpar_hiperbase()	hipger.cpp
extern "C" int FAR PASCAL _export limpa_trilha(char *nome_trilha)	hipger.cpp
extern "C" int FAR PASCAL _export mostra_primeiro_lixo(char *nome)	hipger.cpp
extern "C" int FAR PASCAL _export mostra_proximo_lixo(char *nome)	hipger.cpp
extern "C" int FAR PASCAL _export muda_ancora_contexto(char *nome, int velho_ide, int novo_ide, char *no_ancorado)	hipger.cpp
extern "C" int FAR PASCAL _export muda_ancora_texto(char *nome, int velho_ide, int novo_ide, char *palavra, int inicio, int fim)	hipger.cpp
extern "C" int FAR PASCAL _export muda_assunto(char *nome_no, char *assunto)	hipger.cpp
extern "C" int FAR PASCAL _export muda_atributos(char *nome_velho, char *nome_novo, int num_estado)	hipger.cpp
extern "C" int FAR PASCAL _export muda_dificuldade(char *nome_no, int nivel)	hipger.cpp
extern "C" int FAR PASCAL _export muda_elo(char *nome_contexto, int id_elo, char *origem, int anc_origem, char *destino, int anc_destino, int simultaneo, int mostra_conteudo)	hipger.cpp
extern "C" int FAR PASCAL _export muda_funcao_didatica(char *nome_no, char *categoria)	hipger.cpp
extern "C" int FAR PASCAL _export muda_palavra_chave(char *nome_no, int pos, char *palavra)	hipger.cpp
extern "C" int FAR PASCAL _export muda_tipo(char *nome_no, char *tipo)	hipger.cpp
extern "C" int FAR PASCAL _export muda_vcorrente(char *nome_cv, char *nome_no)	hipger.cpp
extern "C" int FAR PASCAL _export numero_de_ancoras(char *nome)	hipger.cpp
extern "C" int FAR PASCAL _export numero_de_elos(char *nome_contexto)	hipger.cpp
extern "C" int FAR PASCAL _export primeira_ancora(char *nome, char *nome_ancora)	hipger.cpp
extern "C" int FAR PASCAL _export primeiro_contexto_pai(char *nome, char *nome_pai)	hipger.cpp
extern "C" int FAR PASCAL _export primeiro_elo(char *nome, int *id)	hipger.cpp
extern "C" int FAR PASCAL _export primeiro_no_contexto(char *nome_cont, char *nome_no)	hipger.cpp
extern "C" int FAR PASCAL _export primeiro_no_hiperbase(char *nome)	hipger.cpp
extern "C" int FAR PASCAL _export primeiro_no_trilha(char *nome_trilha, char *nome_no, char *nome_liberdade)	hipger.cpp
extern "C" void FAR PASCAL _export ProcessDllIdle()	grafo.cpp
extern "C" int FAR PASCAL _export proxima_ancora(char *nome, char *nome_ancora)	hipger.cpp
extern "C" int FAR PASCAL _export proximo_contexto_pai(char *nome, char *nome_pai)	hipger.cpp
extern "C" int FAR PASCAL _export proximo_elo(char *nome,	hipger.cpp

int *id)	
extern "C" int FAR PASCAL _export proximo_no_contexto(char *nome cont, char *nome no)	hipger.cpp
extern "C" int FAR PASCAL _export proximo_no_hiperbase(char *nome)	hipger.cpp
extern "C" int FAR PASCAL _export proximo_no_trilha(char *nome trilha, char *nome no, char *nome liberdade)	hipger.cpp
extern "C" int FAR PASCAL _export quantos_contextos_pertence(char *nome)	hipger.cpp
extern "C" int FAR PASCAL _export recupera_assunto(char *nome no, char *assunto)	hipger.cpp
extern "C" int FAR PASCAL _export recupera_dificuldade(char *nome no, int *nivel)	hipger.cpp
extern "C" int FAR PASCAL _export recupera_elo(char *nome_contexto, int id_elo, char *id_no_orig, int *id_anc_orig, char *id_no_dest, int *id_anc_dest, int *simultaneo, int *mostra conteudo)	hipger.cpp
extern "C" int FAR PASCAL _export recupera_funcao_didatica(char *nome no, char *categoria)	hipger.cpp
extern "C" int FAR PASCAL _export recupera_identificador_interno(char *nome, int ide)	hipger.cpp
extern "C" int FAR PASCAL _export recupera_nome_ancora(char *nome, int id ancora, char *nome ancora)	hipger.cpp
extern "C" int FAR PASCAL _export recupera_nome_arquivo_audio(char *nome_audio, char *arquivo)	hipger.cpp
extern "C" int FAR PASCAL _export recupera_nome_arquivo_executavel(char *nome_executavel, char *arquivo)	hipger.cpp
extern "C" int FAR PASCAL _export recupera_nome_arquivo_imagem(char *nome_imagem, char *arquivo)	hipger.cpp
extern "C" int FAR PASCAL _export recupera_nome_arquivo_tabela(char *nome_tabela, char *arquivo)	hipger.cpp
extern "C" int FAR PASCAL _export recupera_nome_arquivo_texto(char *nome_texto, char *arquivo)	hipger.cpp
extern "C" int FAR PASCAL _export recupera_nome_arquivo_video(char *nome_video, char *arg video)	hipger.cpp
extern "C" int FAR PASCAL _export recupera_palavra_chave(char *nome_no, int pos, char *palavra)	hipger.cpp
extern "C" int FAR PASCAL _export recupera_tipo(char *nome no, char *tipo)	hipger.cpp
extern "C" int FAR PASCAL _export recuperar(char *nome)	hipger.cpp
extern "C" int FAR PASCAL _export regras_tutoriais(char *no_atual, char *escopo, char *historico, int dificuldade, char *no result, char *lib result, int *revisto)	hipger.cpp
extern "C" int FAR PASCAL _export retira_ancora(char *nome, int ident)	hipger.cpp
extern "C" int FAR PASCAL _export retira_da_hiperbase(char *nome no)	hipger.cpp
extern "C" int FAR PASCAL _export retira_da_trilha(char *nome trilha, char *nome no liberdade)	hipger.cpp
extern "C" int FAR PASCAL _export retira_do_contexto(char *nome contexto, char *nome no)	hipger.cpp
extern "C" int FAR PASCAL _export retira_elo(char *nome contexto, int id elo)	hipger.cpp
extern "C" int FAR PASCAL _export retorna_identificador(char *nome)	hipger.cpp
extern "C" int FAR PASCAL _export retorna_nome(int ident, char *nome)	hipger.cpp
extern "C" int FAR PASCAL _export salvar(char *nome)	hipger.cpp
extern "C" int FAR PASCAL _export tipo_no(char *nome)	hipger.cpp

extern "C" int FAR PASCAL _export ultima_ancora(char *nome, char *nome ancora)	hipger.cpp
extern "C" int FAR PASCAL _export ultimo_contexto_pai(char *nome, char *nome pai)	hipger.cpp
extern "C" int FAR PASCAL _export ultimo_no_trilha(char *nome trilha, char *nome no, char *nome liberdade)	hipger.cpp
extern "C" void FAR PASCAL _export WindowResize(int hsize, int vsize)	grafo.cpp
grafo::~grafo()	grafos.h
grafo::grafo()	grafos.h
grafo::item *GetItem(int id)	grafos.h
grafo::ligacao *GetLink(int id)	grafos.h
hipwindow::baseprivada *recupera base ativa()	hip.h
hipwindow::hipwindow()	hip.cpp
hipwindow::insere hipwindow(No *pontno)	hip.h
hipwindow::insere base ativa(No *pontno)	hip.h
hipwindow::No *primeiro()	hip.h
hipwindow::No *proximo()	hip.h
hipwindow::pertence base ativa(No *pontno)	hip.h
hipwindow::retira base ativa(No *pontno)	hip.h
int ancora audio::eh ancora audio()	anc aud.h
int ancora contexto::eh ancora contexto()	anc cont.h
int ancora contexto::grava ancora(FILE *fp)	anc cont.cpp
int ancora contexto::obtem ancora()	anc cont.h
int ancora contexto::recupera ancora(FILE *fp)	anc cont.cpp
int ancora executavel::eh ancora executavel()	anc exec.h
int ancora::grava ancora(FILE *fp)	ancora.cpp
int ancora imagem::eh ancora imagem()	anc ima.h
int ancora::obtem ident externo()	ancora.h
int ancora::recupera ancora(FILE *fp)	ancora.cpp
int ancora tabela::eh ancora tabela()	anc tab.h
int ancora texto::eh ancora texto()	anc tex.h
int ancora texto::grava ancora(FILE *fp)	anc tex.cpp
int ancora texto::obtem inicio ancora()	anc tex.h
int ancora texto::obtem fim ancora()	anc tex.h
int ancora texto::recupera ancora(FILE *fp)	anc tex.cpp
int ancora trilha::eh ancora trilha()	anc tri.h
int ancora video::eh ancora video()	anc vid.h
int baseprivada::grava(FILE *fp)	basepriv.cpp
int baseprivada::insere na base privada(No *pno)	basepriv.cpp
int baseprivada::recupera(FILE *fp)	basepriv.cpp
int baseprivada::retira da base privada(No *pno)	basepriv.cpp
int baseprivada::RetornaIdent(char *nome)	basepriv.cpp
int Cavi::GetHeight (void)	Video.h
int Cavi::GetWidth (void)	Video.h
int CFrame::OnCreate(LPCREATESTRUCT lpCreateStruct)	frame.cpp
int CGrafoObj::Hiperbase(char *nome)	grafoob.h
int CGrafoObj::LimparHiperbase()	grafoob.h
int CGrafoObj::Recuperar(char *arg)	grafoob.h
int CGrafoObj::Salvar(char *arg)	grafoob.h
int CGrafoView::Caminho(int x*temp, int y*temp, int comp)	grafovw.cpp
int CGrafoView::DeterminaEscopo(No *no)	grafovw.cpp
int CGrafoView::PaisNoNivelDeCima(No *no)	grafovw.cpp
int CGrafoView::SomaEscopoEsquerda(No *no, No *filho)	grafovw.cpp
int composicao::commit()	composic.cpp
int composicao::contem(No *no)	composic.cpp
int composicao::grava(FILE *fp)	composic.cpp
int composicao::grava no(FILE *fp)	composic.cpp
int composicao::insere no(No *pno)	composic.cpp
int composicao::muda assunto(char *string)	composic.cpp
int composicao::primeiro elo(int *id)	composic.cpp
int composicao::proximo elo(int *id)	composic.cpp
int composicao::primeiro no(char *nome)	composic.cpp
int composicao::proximo no(char *nome)	composic.cpp

int composicao::recupera(FILE *fp)	composic.cpp
int composicao::recupera_no(FILE *fp)	composic.cpp
int composicao::retira_no(No *pno)	composic.cpp
int contexto::cria ancora lambda(int ide)	contexto.cpp
int contexto::grava_no(FILE *fp)	contexto.cpp
int contexto::recupera_no(FILE *fp)	contexto.cpp
int contextousuario::eh contexto usuario()	contuser.h
int contextousuario::grava(FILE *fp)	contuser.cpp
int contextousuario::recupera(FILE *fp)	contuser.cpp
int contextoversoes::eh contexto versoes()	contvers.h
int contextoversoes::grava(FILE *fp)	contvers.cpp
int contextoversoes::grava_no(FILE *fp)	contvers.cpp
int contextoversoes::recupera(FILE *fp)	contvers.cpp
int contextoversoes::recupera_no(FILE *fp)	contvers.cpp
int contextoversoes::ObtemVersaoCorrente()	contvers.h
int CTipTop::OnCreate(LPCREATESTRUCT lpCreateStruct)	Tiptop.cpp
int elo::eh para mostrar()	elo.h
int elo::eh simultaneo()	elo.h
int elo::grava elo(FILE *fp)	elo.cpp
int elo::obtem id ancora destino()	elo.h
int elo::obtem id ancora origem()	elo.h
int elo::recupera elo(FILE *fp)	elo.cpp
int entidade::fscanf(FILE *fp, const char *format, char *item)	entidade.cpp
int entidade::fscanf(FILE *fp, const char *format, int *item)	entidade.cpp
int entidade::grava(FILE *fp)	entidade.cpp
int entidade::obtem identificador()	entidade.cpp
int entidade::recupera(FILE *fp)	entidade.cpp
int FAR PASCAL anterior_no trilha(char *, char *, char *)	hipdll.h
int FAR PASCAL ativa hiperbase(char *)	hipdll.h
int FAR PASCAL cria ancora contexto(char *, int, char *)	hipdll.h
int FAR PASCAL cria_ancora_texto(char *, int, char *, int, int)	hipdll.h
int FAR PASCAL cria audio(char *)	hipdll.h
int FAR PASCAL cria base privada(char *)	hipdll.h
int FAR PASCAL cria contexto(char *)	hipdll.h
int FAR PASCAL cria_elo(char *, int, char *, int, char *, int, int, int)	hipdll.h
int FAR PASCAL cria executavel(char *)	hipdll.h
int FAR PASCAL cria imagem(char *)	hipdll.h
int FAR PASCAL cria tabela(char *)	hipdll.h
int FAR PASCAL cria texto(char *)	hipdll.h
int FAR PASCAL cria trilha(char *)	hipdll.h
int FAR PASCAL cria versao(char *, char *)	hipdll.h
int FAR PASCAL cria video(char *)	hipdll.h
int FAR PASCAL export recuperar(char *)	grafoapi.h
int FAR PASCAL insere_na_posicao(char *, char *, char *, int)	hipdll.h
int FAR PASCAL insere na trilha(char *, char *, char *)	hipdll.h
int FAR PASCAL insere no contexto(char *, char *)	hipdll.h
int FAR PASCAL insere nome arquivo audio(char *, char *)	hipdll.h
int FAR PASCAL insere_nome_arquivo_executavel(char *, char *)	hipdll.h
int FAR PASCAL insere nome arquivo imagem(char *, char *)	hipdll.h
int FAR PASCAL insere nome arquivo tabela(char *, char *)	hipdll.h
int FAR PASCAL insere nome arquivo texto(char *, char *)	hipdll.h
int FAR PASCAL insere nome arquivo video(char *, char *)	hipdll.h
int FAR PASCAL limpar hiperbase()	hipdll.h
int FAR PASCAL limpa trilha(char *)	hipdll.h
int FAR PASCAL mostra primeiro lixo(char *)	hipdll.h
int FAR PASCAL mostra proximo lixo(char *)	hipdll.h
int FAR PASCAL muda_ancora_texto(char *, int, int, char *, int, int)	hipdll.h

int FAR PASCAL muda_ancora_contexto(char *, int, int, char *)	hipdll.h
int FAR PASCAL muda_assunto(char *, char *)	hipdll.h
int FAR PASCAL muda_atributos(char *, char *, int)	hipdll.h
int FAR PASCAL muda_dificuldade(char *, int)	hipdll.h
int FAR PASCAL muda_elo(char *, int, char *, int, char *, int, int, int)	hipdll.h
int FAR PASCAL muda_funcao_didatica(char *, char *)	hipdll.h
int FAR PASCAL muda_palavra_chave(char *, int, char *)	hipdll.h
int FAR PASCAL muda_tipo(char *, char *)	hipdll.h
int FAR PASCAL muda_vcorrente(char *, char *)	hipdll.h
int FAR PASCAL numero_de_ancoras(char *)	hipdll.h
int FAR PASCAL numero_de_elos(char *)	hipdll.h
int FAR PASCAL primeira_ancora(char *, char *)	hipdll.h
int FAR PASCAL primeiro_contexto_pai(char *, char *)	hipdll.h
int FAR PASCAL primeiro_elo(char *, int *)	hipdll.h
int FAR PASCAL primeiro_no_contexto(char *, char *)	hipdll.h
int FAR PASCAL primeiro_no_hiperbase(char *)	hipdll.h
int FAR PASCAL primeiro_no_trilha(char *, char *, char *)	hipdll.h
int FAR PASCAL proxima_ancora(char *, char *)	hipdll.h
int FAR PASCAL proximo_contexto_pai(char *, char *)	hipdll.h
int FAR PASCAL proximo_elo(char *, int *)	hipdll.h
int FAR PASCAL proximo_no_contexto(char *, char *)	hipdll.h
int FAR PASCAL proximo_no_hiperbase(char *)	hipdll.h
int FAR PASCAL proximo_no_trilha(char *, char *, char *)	hipdll.h
int FAR PASCAL Quantos_contextos_pertence(char *)	hipdll.h
int FAR PASCAL recupera_assunto(char *, char *)	hipdll.h
int FAR PASCAL recupera_dificuldade(char *, int *)	hipdll.h
int FAR PASCAL recupera_elo(char *, int, char *, int *, char *, int *, int *, int *)	hipdll.h
int FAR PASCAL recupera_funcao_didatica(char *, char *)	hipdll.h
int FAR PASCAL recupera_identificador_interno(char *, int)	hipdll.h
int FAR PASCAL recupera_nome_ancora(char *, int, char *)	hipdll.h
int FAR PASCAL recupera_nome_arquivo_audio(char *, char *)	hipdll.h
int FAR PASCAL recupera_nome_arquivo_executavel(char *, char *)	hipdll.h
int FAR PASCAL recupera_nome_arquivo_imagem(char *, char *)	hipdll.h
int FAR PASCAL recupera_nome_arquivo_tabela(char *, char *)	hipdll.h
int FAR PASCAL recupera_nome_arquivo_texto(char *, char *)	hipdll.h
int FAR PASCAL recupera_nome_arquivo_video(char *, char *)	hipdll.h
int FAR PASCAL recupera_palavra_chave(char *, int, char *)	hipdll.h
int FAR PASCAL recuperar(char *)	hipdll.h
int FAR PASCAL recupera_tipo(char *, char *)	hipdll.h
int FAR PASCAL regras_tutoriais(char *, char *, char *, int, char *, char *, int *)	hipdll.h
int FAR PASCAL retira_ancora(char *, int)	hipdll.h
int FAR PASCAL retira_da_hiperbase(char *)	hipdll.h
int FAR PASCAL retira_da_trilha(char *, char *)	hipdll.h
int FAR PASCAL retira_do_contexto(char *, char *)	hipdll.h
int FAR PASCAL retira_elo(char *, int)	hipdll.h
int FAR PASCAL retorna_identificador(char *)	hipdll.h
int FAR PASCAL retorna_nome(int, char *)	hipdll.h
int FAR PASCAL salvar(char *)	hipdll.h
int FAR PASCAL tipo_no(char *)	hipdll.h
int FAR PASCAL ultima_ancora(char *, char *)	hipdll.h
int FAR PASCAL ultimo_contexto_pai(char *, char *)	hipdll.h
int FAR PASCAL ultimo_no_trilha(char *, char *, char *)	hipdll.h
int grafo::CriaItem(char *nome, long tipo, long classe, int classif, char *arg)	grafos.h
int grafo::CriaItem(int id, FILE *fp)	grafos.h
int grafo::CriaLigacao(int id, FILE *fp)	grafos.h
int grafo::CriaLigacao(int id*, int id*, long tipo)	grafos.h
int grafo::DestroiItem(int id)	grafos.h
int grafo::DestroiLigacao(int id)	grafos.h

int grafo::ExisteItemFilho(int id*, int id*)	grafos.h
int grafo::LeGrafo(FILE *fp)	grafos.h
int grafo::NBugs()	grafos.h
int grafo::NConceitos()	grafos.h
int grafo::NExemplos()	grafos.h
int grafo::NExercicios()	grafos.h
int grafo::NItens()	grafos.h
int grafo::NLinks()	grafos.h
int grafo::Nresultados()	grafos.h
int grafo::NTopicos()	grafos.h
int grafo::PrimeiroItemGrafo()	grafos.h
int grafo::PrimeiroLinkGrafo()	grafos.h
int grafo::PrimeiroTopicoGrafo()	grafos.h
int grafo::ProximoItemGrafo()	grafos.h
int grafo::ProximoLinkGrafo()	grafos.h
int grafo::ProximoTopicoGrafo()	grafos.h
int grafo::RetiraItem(item *item*)	grafos.h
int grafo::RetiraLigacao(ligacao *link)	grafos.h
int grafo::RetiraTopico(item *topico)	grafos.h
int grafo::SalvaGrafo(FILE *fp)	grafos.h
int hipwindow::ativa hiperbase(char *nome)	hip.cpp
int hipwindow::grava hiperbase(char *nome)	hip.cpp
int hipwindow::primeiro no hiperbase(char *nome)	hip.cpp
int hipwindow::proximo no hiperbase(char *nome)	hip.cpp
int hipwindow::recupera hiperbase(char *nome)	hip.cpp
int hipwindow::retira(No *pno)	hip.h
int hipwindow::RetornaIdent(char *nome)	hip.cpp
int hipwindow::obtem identificador ultimo no()	hip.cpp
int insere elo(elo *e)	Composic.h
int lista ancoras::grava(FILE *fp)	List anc.cpp
int lista ancoras::insere(ancora *pontancora)	List anc.h
int lista ancoras::recupera(No *n, FILE *fp)	List anc.cpp
int lista ancoras::retira(ancora *pontancora)	List anc.h
int lista elos::grava(FILE *fp)	List elo.cpp
int lista elos::insere(elo *elo*, elo *elo*)	List elo.h
int lista elos::insere(elo *pelo)	List elo.h
int lista elos::recupera(No *pno, FILE *fp)	List elo.cpp
int lista elos::retira(elo *pelo)	List elo.h
int lista::insere(qqtipo a)	Lista.cpp
int lista::insere(qqtipo anterior, qqtipo a)	Lista.cpp
int lista nos::grava(FILE *fp)	List no.cpp
int lista nos::insere(No *pant, No *pno)	List no.h
int lista nos::insere(No *pno)	List no.h
int lista nos::recupera(FILE *fp)	List no.cpp
int lista nos::retira(No *pno)	List no.h
int lista::num elementos()	Lista.h
int lista::retira(qqtipo a)	Lista.cpp
int no audio::cria ancora audio(int ide)	No audio.cpp
int no audio::eh audio()	No audio.h
int no audio::grava(FILE *fp)	No audio.cpp
int no audio::recupera(FILE *fp)	No audio.cpp
int No::commit()	No.cpp
int No::contem ancora(ancora *pnc)	No.cpp
int No::criaversao(char *nome no)	No.cpp
int no executavel::cria ancora executavel(int ide)	No exec.cpp
int no executavel::eh executavel()	No exec.h
int no executavel::grava(FILE *fp)	No exec.cpp
int no executavel::recupera(FILE *fp)	No exec.cpp
int No::grava(FILE *fp)	No.cpp
int No::grava no(FILE *fp)	No.cpp
int no imagem::cria ancora imagem(int ide)	No image.cpp
int no imagem::eh imagem()	No image.h
int no imagem::grava(FILE *fp)	No image.cpp
int no imagem::recupera(FILE *fp)	No image.cpp

int no::insere ancora(ancora *p)	No.h
int no::insere contexto(No *pontno)	No.h
int No::MudaNome(char *novo nome)	No.cpp
int no::num contextos()	No.h
int no::numero ancoras()	No.h
int No::obsolete()	No.cpp
int no::ObtemCV()	No.h
int No::obtem indice(int ide)	No.cpp
int No::pertence contexto(No *pontno)	No.cpp
int no::recupera dificuldade()	No.h
int No::recupera(FILE *fp)	No.cpp
int No::recupera no(FILE *fp)	No.cpp
int no::retira ancora(ancora *p)	No.h
int no::retira contexto(No *pontno)	No.h
int No::retorna indice()	No.cpp
int no tabela::cria ancora tabela(int ide)	No tab.cpp
int no tabela::eh tabela()	No tab.h
int no tabela::grava(FILE *fp)	No tab.cpp
int no tabela::recupera(FILE *fp)	No tab.cpp
int no terminal::grava no(FILE *fp)	Terminal.cpp
int no terminal::muda funcao didatica(char *string)	Terminal.cpp
int no terminal::muda nome arquivo(char *nome)	Terminal.cpp
int no terminal::muda tipo(char *stipo)	Terminal.cpp
int no terminal::muda palavras_chaves(int pos, char *chaves)	Terminal.cpp
int no terminal::recupera no(FILE *fp)	Terminal.cpp
int no_texto::cria ancora texto(int ide, char *pal, int inicio, int fim)	No_texto.cpp
int no_texto::eh texto()	No_texto.h
int no_texto::grava(FILE *fp)	No_texto.cpp
int no_texto::recupera(FILE *fp)	No_texto.cpp
int No::UltCV(int ultcv)	No.cpp
int no video::cria ancora video(int ide)	no video.cpp
int no video::eh video()	no video.h
int no video::grava(FILE *fp)	no video.cpp
int no video::recupera(FILE *fp)	no video.cpp
int obtem identificador()	entidade.h
int pilha::number()	grafovw.h
int Pilha::pop(char *string)	grafovw.cpp
int Pilha::push(char *string)	grafovw.cpp
int retira elo(elo *e)	Composic.h
int trilha::anterior_no_trilha(char *nome, char *nome liberdade)	Trilha.cpp
int trilha::cria ancora lambda(int ide)	Trilha.cpp
int trilha::eh trilha()	Trilha.h
int trilha::grava(FILE *fp)	Trilha.cpp
int trilha::insere no(No *pno, contexto *pno liberdade)	Trilha.cpp
int trilha::insere_no(No *pno, contexto *pno_liberdade, int posicao)	Trilha.cpp
int trilha::insere no(No *pno)	Trilha.cpp
int trilha::insere no(No *pno, int posicao)	Trilha.cpp
int trilha::primeiro_no_trilha(char *nome, char *nome liberdade)	Trilha.cpp
int trilha::proximo_no_trilha(char *nome, char *nome liberdade)	Trilha.cpp
int trilha::recupera(FILE *fp)	Trilha.cpp
int trilha::recupera no(FILE *fp)	Trilha.cpp
int trilha::retira no(No *pno)	Trilha.cpp
int trilha::ultimo_no_trilha(char *nome, char *nome liberdade)	Trilha.cpp
int tutor::primeira estrategia(contexto *pai)	Tutor.cpp
int tutor::segunda estrategia(contexto *pai)	Tutor.cpp
int tutor::tabela dificil(char *origem, char *destino)	Tutor.cpp
int tutor::tabela facil(char *origem, char *destino)	Tutor.cpp

lista ancoras::ancora * primeiro()	List anc.h
lista ancoras::ancora * proximo()	List anc.h
lista ancoras::ancora * proximo(ancora * pontancora)	List anc.h
lista ancoras::~~lista ancoras()	List anc.cpp
lista elos::elo * anterior()	List elo.h
lista elos::elo * anterior(elo * pelo)	List elo.h
lista elos::elo * primeiro()	List elo.h
lista elos::elo * proximo()	List elo.h
lista elos::elo * proximo(elo * pelo)	List elo.h
lista elos::elo * topo()	List elo.h
lista elos::elo * ultimo()	List elo.h
lista elos::~~lista elos()	List elo.cpp
lista::lista()	lista.cpp
lista nos::~~lista nos()	List no.cpp
lista nos::lista nos()	List no.h
lista nos::No *anterior()	List no.h
lista nos::No *anterior(No *pno)	List no.h
lista nos::No * primeiro()	List no.h
lista nos::No * proximo()	List no.h
lista nos::No * proximo(No *pno)	List no.h
lista nos::No * retira topo()	List no.h
lista nos::No * topo()	List no.h
lista nos::No * ultimo()	List no.h
no audio::no audio(char *nome no):no terminal(nome no)	No audio.cpp
no audio::no audio(int id):no terminal(id)	No audio.cpp
No * baseprivada::Retorna(char *nome)	basepriv.cpp
No * baseprivada::RetornaNo(int Id)	basepriv.cpp
No *CGrafoView::FilhosNoNivelDeBaixo(No *no, int &i)	grafovw.cpp
No *CGrafoView::UltimoNoAninhado(char *s)	grafovw.cpp
No *composicao::primeiro no()	composic.cpp
No *composicao::proximo no()	composic.cpp
no_executavel::no_executavel(char *nome no):no terminal(nome no)	No_exec.cpp
no_executavel::no_executavel(int id):no terminal(id)	No_exec.cpp
No *FAR PASCAL _export primeiro_ponteiro_no(char *nome cont)	hipger.cpp
No *FAR PASCAL export proximo_ponteiro no(char *nome cont)	hipger.cpp
No *FAR PASCAL export retorna_ponteiro no(int id)	hipger.cpp
No *FAR PASCAL primeiro_ponteiro no(char *)	hipdll.h
No *FAR PASCAL proximo_ponteiro no(char *)	hipdll.h
No *FAR PASCAL retorna_ponteiro no(int)	hipdll.h
No * hipwindow::RetornaBase(char *nome)	hip.cpp
No * hipwindow::Retorna(char *nome)	hip.cpp
No * hipwindow::RetornaNo(int Id)	hip.cpp
no imagem::no imagem(char *nome no):no terminal(nome no)	No image.cpp
no imagem::no imagem(int id):no terminal(id)	No image.cpp
No::No(char *nome no):entidade()	No.cpp
No::No(int id):entidade()	No.cpp
No *No::primeiro contexto pai()	No.cpp
No *No::proximo contexto pai()	No.cpp
no::No *ultimo contexto pai()	No.h
no tabela::muda ancora tabela(ancora tabela *p, int ide)	No tab.cpp
no tabela::no tabela(char *nome no):no terminal(nome no)	No tab.cpp
no tabela::no tabela(int id):no terminal(id)	No tab.cpp
no texto::no texto(char *nome no):no terminal(nome no)	No texto.cpp
no texto::no texto(int id):no terminal(id)	No texto.cpp
no terminal::no terminal(char *nome no):No(nome no)	Terminal.cpp
no terminal::no terminal(int id):No(id)	Terminal.cpp
no video::no video(char *nome no):no terminal(nome no)	no video.cpp
no video::no video(int id):no terminal(id)	no video.cpp
qqtipo lista::anterior()	Lista.cpp
qqtipo lista::anterior(qqtipo alguem)	Lista.cpp
qqtipo lista::primeiro()	Lista.cpp
qqtipo lista::proximo()	Lista.cpp

qgtipo lista::proximo(qgtipo alguem)	Lista.cpp
qgtipo lista::retira_topo()	Lista.cpp
qgtipo lista::topo()	Lista.h
qgtipo lista::ultimo()	Lista.cpp
static int Cmidi::GetNumMidiDevices (void)	Sound.h
static int Cwave::GetNumWaveDevices (void)	Sound.h
trilha::trilha(char *nome no):composicao(nome no)	Trilha.cpp
trilha::trilha(int id):composicao(id)	Trilha.cpp
tutor::BOOL jah foi visto()	Tutor.h
tutor::No * recupera no atual()	Tutor.h
tutor::tutor()	Tutor.cpp
UINT CGrafoView::GetMouseOverTip(CPoint point)	grafovw.cpp
virtual CGrafoView::~CGrafoView()	grafovw.h
virtual int ancora::eh ancora audio ()	ancora.h
virtual int ancora::eh ancora contexto()	ancora.h
virtual int ancora::eh ancora executavel()	ancora.h
virtual int ancora::eh ancora imagem()	ancora.h
virtual int ancora::eh ancora tabela()	ancora.h
virtual int ancora::eh ancora texto()	ancora.h
virtual int ancora::eh ancora trilha()	ancora.h
virtual int ancora::eh ancora video()	ancora.h
virtual int contexto::eh contexto()	contexto.h
virtual int eh base privada()	basepriv.h
virtual int no::eh anotacao()	No.h
virtual int no::eh audio()	No.h
virtual int no::eh base_privada()	No.h
virtual int no::eh contexto()	No.h
virtual int no::eh contexto usuario()	No.h
virtual int no::eh contexto versoes()	No.h
virtual int no::eh executavel()	No.h
virtual int no::eh imagem()	No.h
virtual int no::eh hiperbase publica()	No.h
virtual int no::eh tabela()	No.h
virtual int no::eh terminal()	No.h
virtual int no::eh texto()	No.h
virtual int no::eh trilha()	No.h
virtual int no::eh video()	No.h
virtual int no terminal::eh terminal()	Terminal.h
virtual void CGrafoView::AssertValid()	grafovw.h
void ancora contexto::muda conteudo(int ide, int id no)	anc cont.cpp
void ancora_texto::muda_conteudo(int ide, char *pal, int inicio, int fim)	anc_tex.cpp
void ancora trilha::muda conteudo(int ide)	anc tri.cpp
void CAboutDlg::DoDataExchange(CDataExchange* pDX)	grafo.cpp
void Cavi::Close (void)	Video.h
void CFrame::AssertValid()	frame.cpp
void CFrame::CreateFrame(HWND hWndParent)	frame.cpp
void CFrame::Dump(CdumpContext& dc)	frame.cpp
void CFrame::OnClose()	frame.cpp
void CFrame::OnDestroy()	frame.cpp
void CFrame::OnFileOpen()	frame.cpp
void CFrame::OnPaint()	frame.cpp
void CFrame::OnSize(UINT nType, int cx, int cy)	frame.cpp
void CGrafoApp::OnAppAbout()	grafo.cpp
void CGrafoApp::OnAppExit()	grafo.cpp
void CGrafoObj::AssertValid()	grafoob.cpp
void CGrafoObj::Dump(CdumpContext& dc)	grafoob.cpp
void CGrafoObj::Serialize(CArchive& ar)	grafoob.cpp
void CGrafoView::AssertValid()	grafovw.cpp
void CGrafoView::Coordenada(int deslx, int desly, int width, int height, int espix, int espy)	grafovw.cpp
void CGrafoView::DesenhaArco(elo *Elo)	grafovw.cpp
void CGrafoView::DestacaArco(elo *Elo)	grafovw.cpp
void CGrafoView::DesenhaNo(No *no)	grafovw.cpp

void CGrafoView::DestacaNo(No *no)	grafovw.cpp
void CGrafoView::DeterminaNivel(No *no, int nivel)	grafovw.cpp
void CGrafoView::Dump(CDumpContext& dc)	grafovw.cpp
void CGrafoView::OnAtualizar()	grafovw.cpp
void CGrafoView::OnDraw(CDC* pDC)	grafovw.cpp
void CGrafoView::OnFileNew()	grafovw.cpp
void CGrafoView::OnFileOpen()	grafovw.cpp
void CGrafoView::OnFileSave()	grafovw.cpp
void CGrafoView::OnInitialUpdate()	grafovw.cpp
void CGrafoView::OnLButtonDown(UINT nFlags, CPoint point)	grafovw.cpp
void CGrafoView::OnLButtonUp(UINT nFlags, CPoint point)	grafovw.cpp
void CGrafoView::OnMouseMove(UINT nFlags, CPoint point)	grafovw.cpp
void CGrafoView::OnNomeIdentificador()	grafovw.cpp
void CGrafoView::OnRButtonDown(UINT nFlags, CPoint point)	grafovw.cpp
void CGrafoView::OnRButtonUp(UINT nFlags, CPoint point)	grafovw.cpp
void CGrafoView::OnTimer(UINT nIDEvent)	grafovw.cpp
void CGrafoView::OnUpdateAtualizar(CCmdUI* pCmdUI)	grafovw.cpp
void CGrafoView::OnUpdateEditNomIdent(CCmdUI* pCmdUI)	grafovw.cpp
void CGrafoView::Posicao(No *no)	grafovw.cpp
void CGrafoView::Refresh()	grafovw.cpp
void CGrafoView::Roteamento(elo *Elo)	grafovw.cpp
void CGrafoView::SetDataObject()	grafovw.cpp
void Cmidi::Stop (void)	Sound.h
void composicao::copia nos(lista nos *lista)	composic.cpp
void contexto::copia contexto(contexto *nc)	contexto.cpp
void contexto::muda_ancora_contexto(ancora *p, int ide, int id no)	contexto.cpp
void contextoversoes::CriaAncoraCV(No *n)	contvers.cpp
void CtipTop::HideCurrentTip()	Tiptop.cpp
void CTipTop::OnPaint()	Tiptop.cpp
void CTipTop::ShowTip(UINT nID, UINT x, UINT y, char *szItemName)	Tiptop.cpp
void Cwave::Stop (void)	Sound.h
void elo::muda ancora destino(int id)	elo.h
void elo::muda ancora origem(int id)	elo.h
void elo::muda mostra conteudo contexto(int tipo)	elo.h
void entidade::troca identificador(int id)	entidade.cpp
void FAR PASCAL export ProcessDllIdle()	grafoapi.h
void FAR PASCAL export WindowResize(int, int)	grafoapi.h
void grafo::AtualizaGrafo()	grafos.h
void grafo::Atualiza(item *item*, int topico)	grafos.h
void grafo::MudaArquivo(char *nome)	grafos.h
void hipwindow::atualiza base ativa(No *pno)	hip.cpp
void lista::insere topo(qgtipo a)	Lista.cpp
void lista_nos::inicializar(int lib mem)	List_no.cpp
void lista_nos::insere topo(No *pno)	List_no.h
void lista::reinicia lista()	Lista.cpp
void no_audio::copia ancoras audio(lista ancoras *ancoras)	No_audio.cpp
void no_audio::copia audio(no_audio *nt)	No_audio.cpp
void no_audio::muda ancora audio(ancora audio *p, int ide)	No_audio.cpp
void no_executavel::copia ancoras_executavel(lista_ancoras *ancoras)	No_exec.cpp
void no_executavel::copia executavel(no_executavel *nt)	No_exec.cpp
void no_executavel::muda ancora_executavel(ancora_executavel *p, int ide)	No_exec.cpp
void no_imagem::copia ancoras_imagem(lista_ancoras *ancoras)	No_image.cpp
void no_imagem::copia imagem(no_imagem *nt)	No_image.cpp
void no_imagem::muda ancora_imagem(ancora_imagem *p, int ide)	No_image.cpp
void no::muda dificuldade(int nivel)	No.h
void no_tabela::copia ancoras_tabela(lista_ancoras *ancoras)	No_tab.cpp

void no_tabela::copia_tabela(no_tabela *nt)	No_tab.cpp
void no_texto::copia_ancoras_texto(lista_ancoras *ancoras)	No_texto.cpp
void no_texto::copia_texto(no_texto *nt)	No_texto.cpp
void no_texto::muda_ancora_texto(ancora *p, int ide, char *pal, int inicio, int fim)	No_texto.cpp
void no_video::copia_video(no_video *nt)	no_video.cpp
void pilha::inicializar()	grafovw.h
void trilha::copia_trilha(trilha *nc)	Trilha.cpp
void trilha::limpa_trilha()	Trilha.cpp
void troca_ident_externo(int id)	Ancora.h
void tutor::muda_historico(trilha *no)	Tutor.h
void tutor::muda_nivel(Botoes_Tutor nivel)	Tutor.h
void tutor::muda_no_atual(No *no)	Tutor.h

3.6 Quadro de Operações-Procedimentos de Implementação

Nesse quadro identifica-se os procedimentos que implementam as operações da interface e, de acordo com sua funcionalidade, os procedimentos são alocados à interface ou a um dos temas definidos anteriormente.

Nas primeiras colunas do quadro são colocadas as opções do menu e as operações de cada opção (descrição da interface). Na próxima coluna são colocados os procedimentos que implementam cada operação, de acordo com a hierarquia de chamadas descrita no quadro de chamadas.

O nível de profundidade que os procedimentos são detalhados nesse quadro depende do interesse em questão (Costa 97). No caso deste trabalho, a documentação recuperada será utilizada na re-engenharia do sistema para mudança do domínio de aplicação. Como o foco principal dessa re-engenharia se concentra na interface do sistema, o quadro de operações-procedimentos de implementação está detalhado até o nível das chamadas das operações exportadas pela DLL.

As próximas colunas do quadro são utilizadas para alocar os procedimentos à interface ou a um dos temas definidos. Assim, tem-se uma coluna para “interface” e uma para cada tema definido. Uma vez que, para melhor visualização, os quadros estão separados por temas (um quadro para o tema *Autoria* e um quadro para o tema *Navegação*), e como são detalhados apenas procedimentos de implementação da interface, com exceção das chamadas à DLL, as referidas colunas foram excluídas.

A seguir são apresentados os quadros de operações-procedimentos de implementação para os temas *Autoria* e *Navegação*, respectivamente.

Opções da Interface para o tema AUTOMIA			
Opções	Operações	Operações Aninhadas	Procedimentos
Hiperbase	Criar Abrir Salvar Sair		<pre>To_fo_Principal.o_bu_CriarHiperbaseClick(Sender: TObject); To_fo_Principal.Ler_DLL; extern "C" int FAR PASCAL _export cria_base_privada(char *nome); To_fo_Principal.o_bu_AbrirHiperbaseClick(Sender: TObject); To_fo_Principal.Ler_DLL; extern "C" int FAR PASCAL _export recuperar(char *nome); To_fo_Principal.o_bu_GravarHiperbaseClick(Sender: TObject); extern "C" int FAR PASCAL _export salvar(char *nome); To_fo_Principal.o_mn_SairClick(Sender: TObject); To_fo_Principal.FormCloseQuery(Sender: TObject; var CanClose: Boolean);</pre>
Janelas	Lado_a_lado Cascata Organizar_ícones 1_Nós_e_Elos Ativar_janela_n		<pre>To_fo_Principal.o_mn_LadoALadoClick(Sender: TObject); To_fo_Principal.o_mn_CascataClick(Sender: TObject); To_fo_Principal.o_mn_OrganizarIconesClick(Sender: TObject);</pre>
Ajuda	Conteúdo		<pre>To_fo_Principal.o_mn_ConteudoClick(Sender: TObject);</pre>
Nós	b_Criar_nó		<pre>To_fo_Nos.o_pm_CriarNoClick(Sender: TObject); extern "C" int FAR PASCAL _export muda_funcao_didatica(char *nome_no, char *categoria); extern "C" int FAR PASCAL _export muda_dificuldade(char *nome_no, int nivel); extern "C" int FAR PASCAL _export muda_palavra_chave(char *nome_no, int pos, char *palavra); extern "C" int FAR PASCAL _export cria_contexto(char *nome) extern "C" int FAR PASCAL _export insere_no_contexto(char *nome_cont, char *nome_no); extern "C" int FAR PASCAL _export cria_texto(char *nome); Muda Atributos;</pre>

Opções	Operações	Operações Aninhadas	Procedimentos
Nós (continuação)	b_Eliminar_nó_da_base b_Retirar_nó_do_contexto b_Visualizar_conteúdo_do_nó b_Visualizar_contexto_selecionado b_Voltar_para_o_contexto_anterior b_Editar_nó		To_fo_Nos.o_pm_ApagarNoClick(Sender: TObject); Extern "C" int FAR PASCAL _export tipo_no(char *nome); Extern "C" int FAR PASCAL _export limpa_trilha(char *nome_trilha); Extern "C" int FAR PASCAL _export retira_da_hiperbase(char *nome_no); To_fo_Nos.o_pm_RetirarNoClick(Sender: TObject); Extern "C" int FAR PASCAL _export retira_do_contexto(char *nome_contexto, char *nome_no); To_fo_Nos.o_pm_ApresentarConteudoDoNoClick(Sender: TObject); extern "C" int FAR PASCAL _export tipo_no(char *nome); To_fo_Nos.Visualizar_No_Terminal(Tipo: char; Nome_No: Ptochar); To_Fo_Nos.ApresentarNoTrilha(NoSelecioneado: Ptochar); To_fo_Trilha.FormCreate(Sender: TObject); To_fo_Trilha.FormClose(Sender: TObject; var Action: TCloseAction); To_Fo_Nos.ApresentarNoTrilha(NoSelecioneado: Ptochar); extern "C" int FAR PASCAL _export primeiro_no_trilha(char *nome_trilha, char *nome_no, char *nome_liberdade); extern "C" int FAR PASCAL _export tipo_no(char *nome); extern "C" int FAR PASCAL _export proximo_no_trilha(char *nome_trilha, char *nome_no, char *nome_liberdade); To_fo_Nos.o_pm_VisualizarContextoClick(Sender: TObject); To_fo_Nos.Ler_Nos; To_fo_Nos.o_pm_SubirContextoClick(Sender: TObject); To_fo_Nos.o_pm_EditarNoClick(Sender: TObject); extern "C" int FAR PASCAL _export tipo_no(char *nome); extern "C" int FAR PASCAL _export recupera_dificuldade(char *nome_no, int *nivel); extern "C" int FAR PASCAL _export recupera_funcao_didatica(char *nome_no, char *categoria); extern "C" int FAR PASCAL _export recupera_palavra_chave(char *nome_no, int pos, char *palavra);

Opções	Operações	Operações Aninhadas	Procedimentos
Nós (continuação)	b_Editar_nó		<pre> extern "C" int FAR PASCAL _export recupera_nome_arquivo_texto(char *nome_texto, char *arquivo); extern "C" int FAR PASCAL _export recupera_nome_arquivo_video(char *nome_video, char *arq_video); extern "C" int FAR PASCAL _export recupera_nome_arquivo_audio(char *nome_audio, char *arquivo); extern "C" int FAR PASCAL _export recupera_nome_arquivo_imagem(char *nome_imagem, char *arquivo); extern "C" int FAR PASCAL _export muda_atributos(char *nome_velho, char *nome_novo, int num_estado); extern "C" int FAR PASCAL _export insere_nome_arquivo_texto(char *nome_texto, char *arquivo); extern "C" int FAR PASCAL _export insere_nome_arquivo_video(char *nome_video, char *arq_video); extern "C" int FAR PASCAL _export insere_nome_arquivo_audio(char *nome_audio, char *arquivo); extern "C" int FAR PASCAL _export insere_nome_arquivo_imagem(char *nome_imagem, char *arquivo); extern "C" int FAR PASCAL _export muda_dificuldade(char *nome_no, int nivel); extern "C" int FAR PASCAL _export muda_funcao_didatica(char *nome_no, char *categoria); extern "C" int FAR PASCAL _export muda_palavra_chave(char *nome_no, int pos, char *palavra); To_fo_Nos.Ler_Nos; To_fo_EditarNo.FormCreate(Sender: TObject); extern "C" int FAR PASCAL _export primeiro_no_hiperbase(char *nome); extern "C" int FAR PASCAL _export proximo_no_hiperbase(char *nome); To_fo_EditarNo.FormActivate(Sender: TObject); To_fo_EditarNo.FormCloseQuery(Sender: TObject; var CanClose: Boolean); </pre>

Opções	Operações	Operações Aninhadas	Procedimentos
Nós (continuação)	b_Browser_Estrutural (continuação) b_Visualizar_conteúdo_do_nó b_Abrir_nova_janela_com_contexto_selecionado b_Browser_Gráfico		extern "C" int FAR PASCAL _export proximo_no_contexto(char *nome_cont, char *nome_no); extern "C" int FAR PASCAL _export recupera_funcao_didatica(char *nome_no, char *categoria); extern "C" int FAR PASCAL _export recupera_dificuldade(char *nome_no, int *nivel); extern "C" int FAR PASCAL _export tipo_no(char *nome); To_fo_BrowserEstrutural.o_bu_ApresentarConteudoClick(Sender: TObject); extern "C" int FAR PASCAL _export tipo_no(char *nome); To_fo_Nos.Visualizar_No_Terminal(Tipo: char; Nome_No: Ptochar); To_Fo_Nos.ApresentarNoTrilha(NoSelecionado: Ptochar); To_fo_BrowserEstrutural.o_bu_AbrirJanelaClick(Sender: TObject); extern "C" int FAR PASCAL _export tipo_no(char *nome); To_fo_Principal.Apresentar_Browser_Grafico(p_Contexto: Ptochar); extern "C" BOOL FAR PASCAL _export MostrarJanela(HWND hWndParent); extern "C" void FAR PASCAL _export ApresentaContexto(char *no); To_fo_Principal.o_bu_BrowserGraficoClick (Sender: TObject); To_fo_Principal.Apresentar_No(p_NomeDoNo: Ptochar; ApresentarNosTerminais: integer) To_fo_Nos.o_bu_BrowserGraficoClick(Sender: TObject); To_fo_Principal.Apresentar_No(p_NomeDoNo: Ptochar; ApresentarNosTerminais: integer); To_fo_Roteiro.o_mp_Visualizar_GrafoClick(Sender: TObject); To_fo_Principal.Apresentar_No(p_NomeDoNo: Ptochar; ApresentarNosTerminais: integer); To_fo_BrowserGrafico.FormResize(Sender: TObject); To_fo_BrowserGrafico.FormClose(Sender: TObject; var Action: TcloseAction);
		b_Ajuda_hipgrafo b_Voltar_nó b_Zoom_in b Zoom out	

Opções	Operações	Operações Aninhadas	Procedimentos
Nós (continuação)	b_Browser_Gráfico (continuação)	Visualizar_Browser_Gráfico b_Browser_Gráfico Explorar_nó	
Elos	b_Criar_elo		<pre> To_fo_Nos.Ler_Elos; extern "C" int FAR PASCAL _export numero_de_elos(char *nome_contexto); extern "C" int FAR PASCAL _export primeiro_elo(char *nome, int *id); extern "C" int FAR PASCAL _export recupera_elo(char *nome_contexto, int id_elo, char *id_no_orig, int *id_anc_orig, char *id_no_dest, int *id_anc_dest, int *simultaneo, int *mostra_conteudo); extern "C" int FAR PASCAL _export retorna_nome(int ident, char *nome); extern "C" int FAR PASCAL _export recupera_nome_ancora(char *nome, int id_ancora, char *nome_ancora); extern "C" int FAR PASCAL _export proximo_elo(char *nome, int *id); To_fo_Nos.o_lb_ListaElosClick(Sender: TObject); extern "C" int FAR PASCAL _export recupera_elo(char *nome_contexto, int id_elo, char *id_no_orig, int *id_anc_orig, char *id_no_dest, int *id_anc_dest, int *simultaneo, int *mostra_conteudo); extern "C" int FAR PASCAL _export retorna_nome(int ident, char *nome); extern "C" int FAR PASCAL _export tipo_no(char *nome); To_fo_Nos.o_bu_CriarEloClick(Sender: TObject); To_fo_CriarElos.Inicializa_Browser(Contexto_Inicial: String); extern "C" int FAR PASCAL _export retorna_identificador(char *nome); extern "C" int FAR PASCAL _export numero_de_elos(char *nome_contexto); extern "C" int FAR PASCAL _export tipo_no(char *nome); extern "C" int FAR PASCAL _export recupera_identificador_interno(char *nome, int ide); extern "C" int FAR PASCAL _export cria_elo(char *nome_contexto, int id_elo, char *origem, int anc_origem, char *destino, int anc_destino, int simultaneo, int mostra_conteudo); </pre>

Opções	Operações	Operações Aninhadas	Procedimentos
Elos (continuação)	b_Criar_elo (continuação)		<pre> To_fo_CriarElos.Ler_Contexto_Origem(Indice: LongInt; Contexto_Inicial: String); extern "C" int FAR PASCAL _export primeiro_no_contexto(char *nome_cont, char *nome_no); To_fo_CriarElos.Ler_Contexto_Origem(Indice: LongInt; Contexto_Inicial: String); extern "C" int FAR PASCAL _export proximo_no_contexto(char *nome_cont, char *nome_no); extern "C" int FAR PASCAL _export tipo_no(char *nome); To_fo_CriarElos.Ler_Contexto_Destino(Indice: LongInt; Contexto_Inicial: String); Extern "C" int FAR PASCAL _export primeiro_no_contexto(char *nome_cont, char *nome_no); extern "C" int FAR PASCAL _export tipo_no(char *nome); To_fo_CriarElos.Ler_Contexto_Destino(Indice: LongInt; Contexto_Inicial: String); extern "C" int FAR PASCAL _export proximo_no_contexto(char *nome_cont, char *nome_no); To_fo_CriarElos.Inicializa_Browser(Contexto_Inicial: String); To_fo_CriarElos.Ler_Contexto_Origem(Indice: LongInt; Contexto_Inicial: String); To_fo_CriarElos.Ler_Contexto_Destino(Indice: LongInt; Contexto_Inicial: String); To_fo_CriarElos.No_Origem_Selecionado; extern "C" int FAR PASCAL _export tipo_no(char *nome); extern "C" int FAR PASCAL _export primeira_ancora(char *nome, char *nome_ancora); extern "C" int FAR PASCAL _export proxima_ancora(char *nome, char *nome_ancora); To_fo_CriarElos.No_Destino_Selecionado; extern "C" int FAR PASCAL _export tipo_no(char *nome); extern "C" int FAR PASCAL _export primeira_ancora(char *nome, char *nome_ancora); extern "C" int FAR PASCAL _export proxima_ancora(char *nome, char *nome_ancora); To_fo_CriarElos.o_ol_BrowserOrigemMouseUp(Sender: TObject; Button: TMouseButton; Shift: TshiftState; X,Y: Integer); To_fo_CriarElos.No_Origem_Selecionado; To_fo_CriarElos.o_ol_BrowserDestinoMouseUp(Sender: TObject; Button: TMouseButton; Shift: TshiftState; X,Y: Integer); </pre>

Opções	Operações	Operações Aninhadas	Procedimentos
Elos (continuação)	b_Criar_elo (continuação)		To_fo_CriarElos.No_Destino_Selecionado; To_fo_CriarElos.o_cb_AncoraOrigemClick(Sender: TObject); To_fo_CriarElos.o_cb_AncoraDestinoClick(Sender: TObject); To_fo_CriarElos.o_ol_BrowserOrigemClick(Sender: TObject); extern "C" int FAR PASCAL _export recupera_funcao_didatica(char *nome_no, char *categoria); extern "C" int FAR PASCAL _export recupera_dificuldade(char *nome_no, int *nivel); extern "C" int FAR PASCAL _export tipo_no(char *nome); To_fo_CriarElos.o_ol_BrowserDestinoClick(Sender: TObject); extern "C" int FAR PASCAL _export recupera_funcao_didatica(char *nome_no, char *categoria); extern "C" int FAR PASCAL _export recupera_dificuldade(char *nome_no, int *nivel); extern "C" int FAR PASCAL _export tipo_no(char *nome); To_fo_Nos.o_bu_EditarEloClick(Sender: TObject); extern "C" int FAR PASCAL _export recupera_elo(char *nome_contexto, int id_elo, char *id_no_orig, int *id_anc_orig, char *id_no_dest, int *id_anc_dest, int *simultaneo, int *mostra_conteudo); To_fo_CriarElos.Inicializa_Browser(Contexto_Inicial: String); extern "C" int FAR PASCAL _export retorna_nome(int ident, char *nome); To_fo_CriarElos.No_Origem_Selecionado; extern "C" int FAR PASCAL _export retorna_identificador(char *nome); extern "C" int FAR PASCAL _export tipo_no(char *nome); extern "C" int FAR PASCAL _export recupera_identificador_interno(char *nome, int ide); extern "C" int FAR PASCAL _export muda_elo(char *nome_contexto, int id_elo, char *origem, int anc_origem, char *destino, int anc_destino, int simultaneo, int mostra_conteudo);
	b_Alterar_elo		To_fo_Nos.o_bu_ApagarEloClick(Sender: TObject); extern "C" int FAR PASCAL _export retira_elo(char *nome_contexto, int id_elo); To_fo_Nos.Ler_Elos; To_fo_Nos.o_lb_ListaElosClick(Sender: TObject);
	b_Eliminar_elo		

Opções	Operações	Operações Aninhadas	Procedimentos
b_Roteiro			<pre> To_fo_Principal.o bu RoteiroClick(Sender: TObject); To_fo_Roteiro.FormCreate (Sender: TObject); extern "C" int FAR PASCAL _export primeiro_no_hiperbase(char *nome); extern "C" int FAR PASCAL _export tipo_no(char *nome); extern "C" int FAR PASCAL _export proximo_no_hiperbase(char *nome); extern "C" int FAR PASCAL _export primeiro_no_trilha(char *nome_trilha, char *nome_no, char *nome_liberdade); To_fo_Roteiro.DesenharRoteiro(p_NoLiberdade, p_NoSelecioneado: String); extern "C" int FAR PASCAL _export proximo_no_trilha(char *nome_trilha, char *nome_no, char *nome_liberdade); To_fo_Roteiro.DesenharRoteiro(p_NoLiberdade, p_NoSelecioneado: String); To_fo_Roteiro.SelecionarNos(p_IndiceInicial: Longint; p_ContextoInicial: String); Extern "C" int FAR PASCAL _export primeiro_no_contexto(char *nome_cont, char *nome_no); extern "C" int FAR PASCAL _export tipo_no(char *nome); extern "C" int FAR PASCAL _export recupera_funcao_didatica(char *nome_no, char *categoria); extern "C" int FAR PASCAL _export recupera_dificuldade(char *nome_no, int *nivel); extern "C" int FAR PASCAL _export recupera_palavra_chave(char *nome_no, int pos, char *palavra); extern "C" int FAR PASCAL _export recupera_assunto(char *nome_no, char *assunto); To_fo_Roteiro.SelecionarNos(p_IndiceInicial: Longint; p_ContextoInicial: String); extern "C" int FAR PASCAL _export proximo_no_contexto(char *nome_cont, char *nome_no); To_fo_Roteiro.o dg_DesenhaRoteiro(Sender: TObject; Col, Row: Longint; Rect: TRect; State: TGridDrawState); extern "C" int FAR PASCAL _export tipo_no(char *nome); To_fo_Roteiro.o o_Nos_SelecionadosClick(Sender: TObject); To_fo_Roteiro.o dg_NoSelecioneado(Sender: TObject; Col, Row: Longint; var CanSelect: Boolean); extern "C" int FAR PASCAL _export tipo_no(char *nome); </pre>

Opções	Operações	Operações Aninhadas	Procedimentos
b_Roteiro (continuação)	b_Selecionar_nós b_Criar_Roteiro Inserir_no_Roteiro Browser_Gráfico		<pre> extern "C" int FAR PASCAL _export recupera_funcao_didatica(char *nome_no, char *categoria); extern "C" int FAR PASCAL _export recupera_dificuldade(char *nome_no, int *nivel); To_fo_Roteiro.o_cb_Palavra_ChaveClick(Sender: TObject); To_fo_Roteiro.o_cb_Palavra_ChaveKeyPress(Sender: TObject; var Key: Char); To_fo_Roteiro.o_mp_Remover_PalavraClick(Sender: TObject); To_fo_Roteiro.o_cb_Funcao_DidaticaChange(Sender: TObject); To_fo_Roteiro.o_cb_DificuldadeChange(Sender: TObject); To_fo_Roteiro.FormClose(Sender: TObject; var Action: TCloseAction); To_fo_Roteiro.o_b_Selecionar_NosClick(Sender: TObject); To_fo_Roteiro.SelecionarNos(p_IndiceInicial: LongInt; p_ContextoInicial: String); To_fo_Roteiro.o_b_Criar_RoteiroClick(Sender: TObject); extern "C" int FAR PASCAL _export limpa_trilha(char *nome_trilha); extern "C" int FAR PASCAL _export retira_da_hiperbase(char *nome_no); extern "C" int FAR PASCAL _export cria_trilha(char *nome); extern "C" int FAR PASCAL _export insere_no_contexto(char *nome_cont, char *nome_no); extern "C" int FAR PASCAL _export insere_na_trilha(char *nome_trilha, char *nome_no, char *no_liberdade); To_fo_Roteiro.o_mp_InserirNoRoteiroClick (Sender: TObject); extern "C" int FAR PASCAL _export primeiro_contexto_pai(char *nome, char *nome_pai); extern "C" int FAR PASCAL _export tipo_no(char *nome); To_fo_Roteiro.DesenharRoteiro(p_NoLiberdade, p_NoSelecioneado: String); To_fo_Roteiro.DesenharRoteiro(p_NoLiberdade, p_NoSelecioneado: String); To_fo_Principal.Apresentar_Browser_Grafico(p_Contexto: Ptochar); extern "C" BOOL FAR PASCAL _export MostrarJanela(HWND hWndParent); extern "C" void FAR PASCAL _export ApresentaContexto(char *no); : integer); </pre>

Opções	Operações	Operações Aninhadas	Procedimentos
b_Roteiro (continuação)	Browser_Gráfico (continuação)		To_fo_Principal.o_bu_BrowserGraficoClick(Sender: TObject); To_fo_Principal.Apresentar_No(p_NomeDoNo: Ptochar; ApresentarNosTerminais: integer); To_fo_Nos.o_bu_BrowserGraficoClick(Sender: TObject); To_fo_Principal.Apresentar_No(p_NomeDoNo: Ptochar; ApresentarNosTerminais: integer); To_fo_Roteiro.o_mp_Visualizar_GrafoClick(Sender: TObject); To_fo_Principal.Apresentar_No(p_NomeDoNo: Ptochar; ApresentarNosTerminais: integer); To_fo_BrowserGrafico.FormResize(Sender: TObject); To_fo_BrowserGrafico.FormClose(Sender: TObject; var Action: TCloseAction);
	Limpar_roteiro Apagar_nó Mudar_nó_liberdade	b_Ajuda_hipgrafo b_Voltar_nó b_Zoom_in b_Zoom_out Visualizar_Browser_Gráfico b_Browser_Gráfico Explorar_nó	To_fo_Roteiro.o_mp_Limpar_TrilhaClick(Sender: TObject); To_fo_Roteiro.o_mp_Apagar_NoClick(Sender: TObject); To_fo_Roteiro.o_mp_MudarNoLiberdadeClick(Sender: TObject); extern "C" int FAR PASCAL _export primeiro_contexto_pai(char *nome, char *nome pai);
b_Glossário			To_fo_Principal.o_bu_GlossarioClick(Sender: TObject); To_fo_Glossario.FormCreate(Sender: TObject); To_fo_Glossario.FormClose(Sender: TObject; To_fo_Glossario.o_ed_PalavraABuscarKeyPress(Sender: TObject; var Key: Char); To_fo_Glossario.o_dbn_ContrólesClick(Sender: TObject; Button: TNavigateBtn);
b_Criar_Hiperbase			To_fo_Principal.o_bu_CriarHiperbaseClick(Sender: TObject); To_fo_Principal.Ler_DLL; extern "C" int FAR PASCAL _export cria_base_privada(char *nome);

Opções	Operações	Operações Aninhadas	Procedimentos
b_Abrir_Hiperbase			To_fo_Principal.o_bu_AbrirHiperbaseClick(Sender: TObject); To_fo_Principal.Ler_DLL; extern "C" int FAR PASCAL export recuperar(char *nome);
b_Salvar_Hiperbase			To_fo_Principal.o_bu_GravarHiperbaseClick(Sender: TObject); extern "C" int FAR PASCAL export salvar(char *nome);

Opções de interface de linha NAVEGAÇÃO		
Opções	Operações	Procedimentos
Janela de abertura da hiperbase	Estudante	<pre> To_fo_Principal.FormActivate(Sender: TObject); extern "C" int FAR PASCAL _export recuperar(char *nome); extern "C" int FAR PASCAL _export cria_base_privada(char *nome); extern "C" int FAR PASCAL _export cria_trilha(char *nome); extern "C" int FAR PASCAL _export insere_no_contexto(char *nome_cont, char *nome_no); extern "C" int FAR PASCAL _export ativa_hiperbase(char *nome); extern "C" int FAR PASCAL _export primeiro_no_trilha(char *nome_trilha, char *nome_no, char *nome_liberdade); To_fo_Principal.Apresentar_No(p_NomeDoNo: Ptochar; ApresentarNosTerminais: integer); extern "C" int FAR PASCAL _export insere_na_trilha(char *nome_trilha, char *nome_no, char *no liberdade); </pre>
Controles	b_Ajuda b_Avança_histórico b_Avança_tópico b_Bibliografia b_Contexto	<pre> To_fo_Controles.o_bu_AjudaClick(Sender: TObject); extern "C" int FAR PASCAL _export primeiro_contexto_pai(char *nome, char *nome_pai); extern "C" int FAR PASCAL _export primeiro_no_contexto(char *nome_cont, char *nome_no); extern "C" int FAR PASCAL _export recupera_funcao_didatica(char *nome_no, char *categoria); extern "C" int FAR PASCAL _export proximo_no_contexto(char *nome_cont, char *nome_no); extern "C" int FAR PASCAL _export primeiro_contexto_pai(char *nome, char *nome_pai); To_fo_Principal.Apresentar_No(p_NomeDoNo: Ptochar; ApresentarNosTerminais: integer); To_fo_Controles.o_bu_ProximoClick(Sender: TObject); To_fo_Principal.Apresentar_No(p_NomeDoNo: Ptochar; ApresentarNosTerminais: integer); To_fo_Controles.o_bu_ProximoTopicoClick(Sender: TObject); extern "C" int FAR PASCAL _export proximo_no_trilha(char *nome_trilha, char *nome_no, char *nome_liberdade); extern "C" int FAR PASCAL _export ativa_hiperbase(char *nome); extern "C" int FAR PASCAL _export insere_na_trilha(char *nome_trilha, char *nome_no, char *no liberdade); To_fo_Principal.Apresentar_No(p_NomeDoNo: Ptochar; ApresentarNosTerminais: integer); To_fo_Controles.o_bu_BibliografiaClick(Sender: TObject); extern "C" int FAR PASCAL _export primeiro_contexto_pai(char *nome, char *nome_pai); extern "C" int FAR PASCAL _export primeiro_no_contexto(char *nome_cont, char *nome_no); extern "C" int FAR PASCAL _export recupera_funcao_didatica(char *nome_no, char *categoria); extern "C" int FAR PASCAL _export proximo_no_contexto(char *nome_cont, char *nome_no); To_fo_Principal.Apresentar_No(p_NomeDoNo: Ptochar; ApresentarNosTerminais: integer); To_fo_Controles.o_bu_ContextoClick(Sender: TObject); To_fo_Principal.Apresentar_No(p_NomeDoNo: Ptochar; ApresentarNosTerminais: integer); </pre>

Opções	Operações	Procedimentos
Controles (continuação)	b_Está_difícil	<pre> To_fo_Controles.o_bu_DificilClick(Sender: TObject); extern "C" int FAR PASCAL _export regras_tutoriais(char *no_atual, char *escopo, char *historico, int dificuldade, char *no_result, char *lib_result, int *revisto); extern "C" int FAR PASCAL _export ativa_hiperbase(char *nome); extern "C" int FAR PASCAL _export insere_na_trilha(char *nome_trilha, char *nome_no, char *no liberdade); To_fo_Principal.Apresentar_No(p_NomeDoNo: Ptochar; ApresentarNosTerminais: integer); </pre>
	b_Está_fácil	<pre> To_fo_Controles.o_bu_FacilClick(Sender: TObject); extern "C" int FAR PASCAL _export regras_tutoriais(char *no_atual, char *escopo, char *historico, int dificuldade, char *no_result, char *lib_result, int *revisto); extern "C" int FAR PASCAL _export ativa_hiperbase(char *nome); extern "C" int FAR PASCAL _export insere_na_trilha(char *nome_trilha, char *nome_no, char *no liberdade); To_fo_Principal.Apresentar_No(p_NomeDoNo: Ptochar; ApresentarNosTerminais: integer); </pre>
	b_Exercícios	<pre> To_fo_Controles.o_bu_ExerciciosClick(Sender: TObject); extern "C" int FAR PASCAL _export primeiro_contexto_pai(char *nome, char *nome_pai); extern "C" int FAR PASCAL _export primeiro_no_contexto(char *nome_cont, char *nome_no); extern "C" int FAR PASCAL _export recupera_funcao_didatica(char *nome_no, char *categoria); extern "C" int FAR PASCAL _export proximo_no_contexto(char *nome_cont, char *nome_no); To_fo_Principal.Apresentar_No(p_NomeDoNo: Ptochar; ApresentarNosTerminais: integer); </pre>
	b_glossario	<pre> To_fo_Controles.o_bu_GlossarioClick(Sender: TObject); </pre>
	b_Histórico	<pre> To_fo_ListaDeNos.FormCreate(Sender: TObject); To_fo_ListaDeNos.FormClose(Sender: TObject; var Action: TCloseAction); To_fo_ListaDeNos.o_lb_NosDbClick(Sender: TObject); To_fo_Principal.Apresentar_No(p_NomeDoNo: Ptochar; ApresentarNosTerminais: integer); To_fo_Controles.o_bu_HistoricoClick(Sender: TObject); extern "C" int FAR PASCAL _export ativa_hiperbase(char *nome); extern "C" int FAR PASCAL _export primeiro_no_trilha(char *nome_trilha, char *nome_no, char *nome_liberdade); extern "C" int FAR PASCAL _export tipo_no(char *nome); extern "C" int FAR PASCAL _export proximo_no_trilha(char *nome_trilha, char *nome_no, char *nome_liberdade); </pre>
	b_Mais_informações	<pre> To_fo_Controles.o_bu_MaisInformacoesClick(Sender: TObject); extern "C" int FAR PASCAL _export tipo_no(char *nome); To_fo_MaisInformacoes.FormCreate(Sender: TObject); To_fo_MaisInformacoes.Ler_Contexto(Contexto_Inicial: String); extern "C" int FAR PASCAL _export primeiro_no_contexto(char *nome_cont, char *nome_no); extern "C" int FAR PASCAL _export recupera_palavra_chave(char *nome_no, int pos, char *palavra); To_fo_MaisInformacoes.Ler_Contexto(Contexto_Inicial: String); </pre>

Opções	Operações	Procedimentos
Controles (continuação)	b_Mais_informações (continuação) b_Onde_estou? b_Retrocede_histórico b_Retrocede_tópico	<pre> extern "C" int FAR PASCAL _export proximo_no_contexto(char *nome_cont, char *nome_no); extern "C" int FAR PASCAL _export tipo_no(char *nome); To_fo_MaisInformacoes.o_lb_NosSelecionadosDbClick(Sender: TObject); extern "C" int FAR PASCAL _export tipo_no(char *nome); To_fo_Nos.Visualizar_No_Terminal(Tipo: char; Nome_No: Ptochar); To_fo_MaisInformacoes.FormClose(Sender: TObject; var Action: TCloseAction); To_fo_ListaDeNos.FormCreate(Sender: TObject); To_fo_ListaDeNos.FormClose(Sender: TObject; var Action: TCloseAction); To_fo_Controles.o_bu_OndeEstouClick(Sender: TObject); extern "C" int FAR PASCAL _export primeiro_no_trilha(char *nome_trilha, char *nome_no, char *nome_liberdade); extern "C" int FAR PASCAL _export tipo_no(char *nome); extern "C" int FAR PASCAL _export proximo_no_trilha(char *nome_trilha, char *nome_no, char *nome_liberdade); To_fo_Controles.o_bu_AnteriorClick(Sender: TObject); To_fo_Principal.Apresentar_No(p_NomeDoNo: Ptochar; ApresentarNosTerminais: integer); To_fo_Controles.o_bu_TopicoAnteriorClick(Sender: TObject); extern "C" int FAR PASCAL _export anterior_no_trilha(char *nome_trilha, char *nome_no, char *nome_liberdade); extern "C" int FAR PASCAL _export ativa_hiperbase(char *nome); extern "C" int FAR PASCAL _export insere_na_trilha(char *nome_trilha, char *nome_no, char *no_liberdade); To_fo_Principal.Apresentar_No(p_NomeDoNo: Ptochar; ApresentarNosTerminais: integer); </pre>
Sashe	Sair	<pre> To_fo_Principal.FormCloseQuery(Sender: TObject; var CanClose: Boolean); extern "C" int FAR PASCAL _export ativa_hiperbase(char *nome); extern "C" int FAR PASCAL _export salvar(char *nome); To_fo_Principal.o_mn_SairClick(Sender: TObject); </pre>
Janelas	Lado_a_lado Cascata Organizar ícones Ativar_janela_n	<pre> To_fo_Principal.o_mn_LadoALadoClick(Sender: TObject); To_fo_Principal.o_mn_CascataClick(Sender: TObject); To_fo_Principal.o_mn_OrganizarIconesClick(Sender: TObject); </pre>
Ajuda	Conteúdo	<pre> To_fo_Principal.o_mn_ConteudoClick(Sender: TObject); </pre>

4 Conclusões

Este relatório apresentou a documentação resultante da aplicação do método de engenharia reversa Fusion-RE/I no sistema SASHE. Foram apresentados os documentos componentes da visão funcional do software e da visão estrutural, respectivamente.

A documentação apresentada neste trabalho é composta por um modelo de ciclo de vida com cinquenta e cinco (55) sentenças, um modelo de operações composto por sessenta e duas (62) operações e dois modelos de objetos, um para cada tema proposto. A recuperação estrutural resultou em oitenta e quatro (84) quadros de chamadas, descrevendo setecentos e vinte e dois (722) procedimentos C++ e em vinte e um (21) quadros de chamadas, descrevendo cento e sessenta e cinco (165) procedimentos Delphi. Também foram mostrados dois índices de procedimentos, contendo a localização de cada um dos procedimentos analisados. Finalmente, foram apresentados os quadros de operações-procedimentos de implementação, um para o tema *Autoria* e outro para o tema *Navegação*.

O volume de informação manipulado durante a realização deste trabalho confirma a necessidade de meios mais ágeis para o armazenamento e a interação com esse tipo de informação. Dessa forma, com a proposta de inserir toda a informação aqui descrita em um sistema hipertexto, o SASHE, esperamos tornar a manipulação dessas informações mais eficiente, bem como garantir a consistência entre os diversos documentos.

Agradecimentos

Gostaria de agradecer a Fabrício Campos Zuardi e Gustavo Cecílio Capra, ambos alunos do curso de Bacharelado em Ciências da Computação - ICMC, por colaborarem na recuperação de parte da documentação estrutural apresentada neste relatório.

Referências Bibliográficas

(Biggerstaff 89) Biggerstaff, T. Design Recovery for Maintenance and Reuse. *IEEE Computer*, v.22, n.7, pp. 36-49, 1989.

- (Casanova 91) Casanova, M. A.; Tucherman, L.; Lima, M. J.; Rangel, J. L.; Rodriguez, N. R.; Soares, L. F. G. The Nested Context Model for Hyperdocuments. Third ACM Conference on Hypertext, *Proceedings*, San Antonio, Texas, 1991, pp. 193-201.
- (Chikofsky 90) Chikofsky, E.J.; Cross II, J.H. Reverse Engineering and Design Recovery: A Taxonomy. *IEEE Software*, v.7, n.1, pp. 13-7, 1990.
- (Coleman 96) Coleman, D. et al. *Desenvolvimento Orientado a Objetos: O Método Fusion*. Ed. Campos, Rio de Janeiro, 1996.
- (Costa 97) Costa, Rejane M. *Um método de Engenharia Reversa para Auxiliar a Manutenção de Software*. Dissertação de mestrado apresentado ao ICMC-USP, São Carlos-SP, 1997.
- (Masiero 95) Masiero, P.C. *Análise Orientada a Objetos: Uma Introdução ao Método Fusion*. Documento preparado como apoio ao tutorial homônimo apresentado no IX Simpósio Brasileiro de Engenharia de Software, Recife, 1995, 49p.
- (Nunes 96) Nunes, M.G.V.; Hasegawa, R.; Vieira, F.M.C. *Hip/Windows: Um Ambiente de Autoria de Hiperbases Multimídia*, Relatório Técnico, nº 38, ICMC-USP, São Carlos-SP, 1996.
- (Nunes 97) Nunes, M.G.V.; Hasegawa, R.; Vieira, F.M.C.; Santos, G.H.R.; Fortes, R.P.M. *SASHE: Sistema de Autoria e Suporte Hiperídia para Ensino*. Notas, nº 33, ICMC-USP, São Carlos-SP, 1997.
- (Penteado 96) Penteado, R.A.D. *Um método para Engenharia Reversa Orientado a Objetos*. Tese de Doutorado apresentado ao IFSC-USP, São Carlos-SP, 1996.
- (Rational 99) Rational Corporation. URL: <http://www.rational.com>, 1999.
- (STI 99) STI - Scientific Toolworks, Inc. URL <http://www.scitools.com>, 1999.

Data	25/03/99
Proc	S.C. S.F.C.
Por	
Liv	
RS	