UNIVERSIDADE DE SÃO PAULO

Instituto de Ciências Matemáticas e de Computação

ISSN 0103-2569

AMeG: ACCESSIBLE MENU GENERATOR
MANUAL TÉCNICO E OPERACIONAL

HUMBERTO LIDIO ANTONELLI
RENATA PONTIN DE MATTOS FORTES

Nº 403

RELATÓRIOS TÉCNICOS

São Carlos – SP
Jan./2015

Universidade de São Paulo
Instituto de Ciências Matemáticas e de Computação
Programa de Pós-Graduação em Ciências da Computação e Matemática Computacional

AMeG: *Accessible Menu Generator*
Manual técnico e operacional

Humberto Lidio Antonelli
Renata Pontin de Mattos Fortes

Departamento de Ciências de Computação (SCC)

Caixa Postal 668 – 13560-970 São Carlos – SP

Janeiro de 2015

Os autores agradecem à
Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP)
Processo 2013/23966-8, pelo apoio financeiro a esta pesquisa.

SUMÁRIO

1 INTRODUÇÃO 5
1.1 Contexto e motivação 5
1.2 Objetivos 6
1.3 Organização deste manual 7

2 INSTALANDO A AMEG 9
2.1 Pré-requisitos do servidor 9
2.2 Instalação 9

3 UTILIZANDO A AMEG 13
3.1 Pré-requisitos do cliente 13
3.2 Navegação 14
3.3 Acessibilidade 14
3.4 Editor textual 15

4 UTILIZANDO A LINGUAGEM AMENU 19
4.1 Estrutura da linguagem 20
4.2 Criando um novo modelo de menu 21
4.2.1 Definições iniciais 21
4.2.2 Configurações do menu 22
4.2.3 Itens 23
4.2.4 Imagens 25
4.2.5 Submenus 25
4.3 Exemplo completo de um menu definido em AMenu 26

5 INFORMAÇÕES TÉCNICAS 29
5.1 Tecnologias empregadas 29
5.1.1 Aplicação Web 30
5.1.2 Linguagem AMenu 31
5.2 Licença 36
5.3 Como contribuir com o projeto 37

Referências 39

APÊNDICE A GRAMÁTICA EM NOTAÇÃO EBNF DA LINGUAGEM AMENU 41

APÊNDICE B EXEMPLO COMPLETO DE UM MENU WEB ACESSÍVEL 45

CAPÍTULO

1

INTRODUÇÃO

Atualmente, devido à popularização da Internet, grande parte das tarefas diárias (compras, movimentações financeiras, comunicação entre as pessoas, etc.) passaram a ser desempenhadas por meio da Web, tornando-a indispensável na rotina da maioria das pessoas. Assim, com o crescimento do número de usuários e de recursos oferecidos na Web, a acessibilidade passou a ser um fator muito importante, tendo em vista o caráter essencial de se promover a inclusão digital de pessoas com deficiências ou com restrições temporárias de mobilidade.

Segundo dados do XII Censo Demográfico realizado pelo Instituto Brasileiro de Geografia e Estatística (IBGE) em 2010[1], 45,6 milhões de brasileiros possuíam algum tipo de deficiência. Esse número representa cerca de 23,91% de toda a população do país. Entretanto, a obtenção de interfaces que satisfaçam e possibilitem que todos os usuários, em especial aqueles que possuem algum limitação temporária ou permanente, realizem suas tarefas não é simples, visto a diversidade de perfil das pessoas e suas diferentes necessidades (NICHOLL, 2001).

1.1 Contexto e motivação

Acessibilidade na Web refere-se a percepção, o entendimento, a navegação e a interação com a Web, por todas as pessoas, que livres de barreiras (ou com barreiras reduzidas), ainda que contando com recursos de Tecnologia Assistiva, também possam utilizar ou contribuir com a geração de conteúdo (Web 2.0) e desenvolvimento de sistemas Web (RUTTER *et al.*, 2006; GRILLO; FORTES, 2014). Portanto, em termos gerais, a definição de acessibilidade na Web visa garantir o uso da Web por todas as pessoas, independente de suas limitações ou deficiências.

Todavia, apesar da acessibilidade ser um requisito fundamental a ser atendido para que todos possam acessar o conteúdo disponibilizado na Web, muitos desenvolvedores, principalmente aqueles que estão começando a trabalhar com projetos para Internet, em geral, não possuem

[1] http://biblioteca.ibge.gov.br/visualizacao/periodicos/94/cd_2010_religiao_deficiencia.pdf

conhecimento suficiente sobre acessibilidade, ou ainda apresentam dificuldades para projetar interfaces Web acessíveis(FREIRE; RUSSO; FORTES, 2008; MIÑÓN *et al.*, 2014).

O processo de inclusão de acessibilidade requer, inicialmente, a conscientização e o treinamento dos desenvolvedores sobre a importância em implementar esse requisito em seus artefatos. Além disso, os detalhes específicos, que precisam ser levados em consideração no desenvolvimento de uma solução acessível, exige que os desenvolvedores conheçam tecnologias específicas de acessibilidade.

Assim, com o objetivo de orientar os projetistas na criação de artefatos acessíveis, diversos documentos propõem *guidelines* de acessibilidade. De acordo com Kelly *et al.* (2005), o documento de referência nessa areá é a *Web Content Accessibility Guidelines* (WCAG), elaborado pela WAI[2]. Atualmente, a WCAG está na versão 2.0, e é composto por princípios e orientações de caráter geral que orientam a criação do conteúdo das páginas Web, explicando como torná-lo mais acessível às pessoas com deficiência.

Apesar da WCAG 2.0 sugerir *guidelines* para a criação de conteúdo Web acessível, grande parte dos desenvolvedores ainda não adotam essas diretrizes em seus projetos, resultando em diversos problemas de acessibilidade. Um desses problemas é a dificuldade em relação à navegação pelo conteúdo disponibilizado em um site, que em alguns casos pode ser impossível para alguns usuários. Um caso típico é o de menus de navegação mal estruturados e que não possuem acessibilidade. Sendo assim, é importante que sejam pesquisadas abordagens que facilitem a adoção dessas recomendações, a fim de enriquecer a Web com a produção de conteúdos acessíveis e, consequentemente, permitir que todos possam acessá-la sem dificuldades ou, pelo menos, com barreiras reduzidas.

Neste contexto, foi desenvolvida a *Accessible Menu Generator* (AMeG), uma ferramenta projetada para auxiliar os desenvolvedores na criação de menus Web acessíveis, em conformidade com as normas de acessibilidade descritas na WCAG 2.0. Para tanto, todos os requisitos de acessibilidade necessários para este elemento da aplicação foram criteriosamente estudados e modelados em uma linguagem específica de domínio (DSL - *Domain Specific Language*), de modo que os desenvolvedores não precisam se preocupar com a implementação (o código) deles. Essa abstração do modelo de um menu acessível foi possível em razão da utilização da DSL *Accessible Menu* (AMenu), que foi projetada especialmente para o desenvolvimento deste trabalho, e que constitui a base para criação de modelos de menus de navegação Web que abstraem as principais características existentes nesse domínio.

1.2 Objetivos

O principal objetivo deste manual é orientar os usuários da AMeG nos procedimentos de utilização da ferramenta, apresentando de forma simples e concisa seus componentes a fim

[2] http://www.w3c.org/WAI/

de que seus usuários criem menus de navegação Web com o máximo de acessibilidade. Além disso, este manual descreve os procedimentos necessários para instalação da ferramenta em um servidor Web requerido para seu funcionamento.

Por fim, este manual traz detalhes técnicos sobre a linguagem AMenu, bem como as instruções necessárias para que outros desenvolvedores de software também possam colaborar com o aperfeiçoamento da linguagem e o desenvolvimento da ferramenta, contribuindo, assim, com a criação de uma Web cada vez mais acessível.

1.3 Organização deste manual

Neste capítulo foi apresentada a contextualização e motivação que levaram ao desenvolvimento da ferramenta AMeG, bem como os objetivos deste manual.

O restante deste trabalho é organizado da seguinte forma: o Capítulo 2 fornece instruções sobre como instalar a ferramenta AMeG em um servidor Web. O Capítulo 3 contém um manual ensinando como utilizar a AMeG para criação de menus de navegação Web acessíveis, passando por todas as páginas e ações possíveis em cada uma delas. No Capítulo 4 encontra-se um manual detalhado sobre como utilizar a linguagem textual AMenu, responsável pela abstração dos modelos de menus que são utilizados pela AMeG para geração de menus acessíveis. Por fim, no Capítulo 5 são mostrados detalhes técnicos da arquitetura e organização da AMeG bem como instruções sobre como contribuir com o projeto.

CAPÍTULO

2

INSTALANDO A AMEG

Este capítulo tem como objetivo conduzir o leitor na instalação da ferramenta AMeG em um servidor Web, bem como apresentar os pré-requisitos necessários para esta instalação. Ao final do capítulo, a AMeG estará instalada e pronta para ser utilizada por qualquer usuário.

2.1 Pré-requisitos do servidor

A AMeG foi desenvolvida utilizando a plataforma Java em que os programas são executados sobre uma *Java Virtual Machine* (JVM). Dessa forma, qualquer plataforma de hardware ou equipamento eletrônico que possa executar uma máquina virtual conseguirá executar Java. Assim, a AMeG pode ser executada tanto em máquinas com o sistema operacional Windows quanto Linux, porém, este manual concentra-se apenas em instruções para a instalação em uma máquina Linux.

O processo de instalação segue o mesmo para ambos os sistemas, embora alguns detalhes como localização de diretórios podem ser diferentes de um sistema para o outro. Para funcionamento da AMeG, são necessários:

- Java 1.6 ou superior;
- JavaServer Faces (JSF) 2.2 ou superior;
- Apache Tomcat 7.0.23 ou superior.

2.2 Instalação

Para instalar a AMeG, é necessário primeiramente realizar o *download* do código-fonte disponível publicamente no repositório do GitHub[1]. Na Figura 1 é apresentada a estrutura de

[1] https://github.com/lordantonelli/AMeG

diretórios da AMeG após o *download* do código-fonte.

Figura 1 – Estrutura de diretórios da base de código da AMeG

Fonte: Elaborada pelo autor.

O diretório `dist` possui uma versão de distribuição da AMeG em formato `war`. Esse arquivo é utilizado pelos servidores Web para executar a aplicação, nele está contido todo o código-fonte compilado da aplicação. Para instalar a AMeG no servidor Apache Tomcat, basta copiar o arquivo `ameg.war` para dentro do diretório `webapps` da sua instalação do Tomcat, conforme o Código 1.

Código 1: Instalação da AMeG no servidor Tomcat

```
1 cp ameg.war /var/lib/tomcat7/webapps/
```

Após este passo, o Tomcat irá automaticamente extrair o conteúdo do arquivo e criará o diretório `ameg` dentro do diretório `webapps`. Dentro deste diretório encontram-se todos os arquivos necessários para a execução da AMeG, não sendo necessária nenhuma configuração adicional.

Após esse procedimento, já é possível acessar a AMeG e verificar se tudo foi instalado corretamente. Para isso, basta acessar o endereço URL[2] da ferramenta em qualquer navegador. No caso de uma instalação na máquina local, este endereço é normalmente: `http://localhost:8080/ameg`.

[2] Uniform Resource Locator (URL)

Entretanto, tanto a URL quanto a porta utilizada podem variar de acordo com a instalação e configuração do Tomcat que foi utilizada. Cabe ressaltar também que qualquer alteração no código-fonte original, o projeto deve ser recompilado e a nova versão deve ser colocada no servidor Tomcat para que as alterações sejam disponibilizadas.

CAPÍTULO

3

UTILIZANDO A AMEG

Uma vez que a AMeG está instalada e pronta para uso, a ferramenta pode ser acessada a partir de qualquer navegador Web moderno que suporte os pré-requisitos necessários para seu funcionamento correto, os quais são apresentados a seguir, bem como as instruções para utilização da ferramenta.

3.1 Pré-requisitos do cliente

Por se tratar de uma aplicação Web, o requisito principal que se faz necessário para utilização da AMeG é um navegador (*browser*). Entretanto, deve-se buscar preferencialmente versões recentes do navegador que será utilizado. Alguns navegadores que foram testados são:

- Google Chrome[1];
- Internet Explorer[2];
- Mozila Firefox[3];
- Safari[4];

A ferramenta AMeG deverá funcionar em outros navegadores modernos, embora não tenha sido exaustivamente testada nos demais navegadores. Além disso, é preciso que o navegador utilizado permita a execução de Javascript, caso contrário isso poderá comprometer o correto funcionamento e apresentação visual da AMeG.

[1] http://www.google.com.br/chrome/
[2] http://windows.microsoft.com/pt-br/internet-explorer/download-ie
[3] https://www.mozilla.org/pt-BR/firefox/new/
[4] http://www.apple.com/br/safari/

3.2 Navegação

A ferramenta AMeG apresenta dois menus a partir dos quais o usuário pode interagir; esses menus são os principais componentes de navegação dentro da aplicação. O menu lateral à esquerda e a barra de acessibilidade localizada no topo da página.

Como a ferramenta não disponibiliza nenhum tipo de armazenamento permanente dos menus gerados, o menu lateral esquerdo permanece inalterado com três opções: Início, Sobre a AMeG e Ajuda. O item "Início" apresenta a página inicial da ferramenta, em que podem ser gerados os menus acessíveis. Essa página da aplicação é melhor detalhada na Seção 3.4.

A segunda opção do menu (Sobre a AMeG) informa ao usuário detalhes sobre a origem da ferramenta e seus responsáveis. No último item (Ajuda) são disponibilizadas algumas informações técnicas sobre a ferramenta, bem como as instruções de apoio para o usuário poder utilizá-la. Na Figura 2 pode ser vista a aparência da ferramenta.

Figura 2 – Captura de tela mostrando a aparência da AMeG

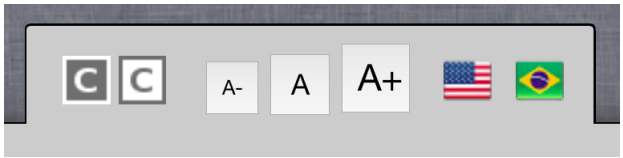
Fonte: Elaborada pelo autor.

3.3 Acessibilidade

A AMeG contém algumas características de acessibilidade desenvolvidas internamente. Essas características estão presentes na barra de navegação superior, na qual estão dispostas as funcionalidades que auxiliam o usuário na utilização da ferramenta. Na Figura 3 pode ser observada a barra de ferramentas de acessibilidade.

Da esquerda para a direita: Ativar alto contraste, Desativar alto contraste, Diminuir tamanho da fonte, Fonte em tamanho normal, Aumentar tamanho da fonte, Alterar idioma para Inglês e Alterar idioma para Português.

Figura 3 – Captura de tela mostrando a barra de ferramentas de acessibilidade



Fonte: Elaborada pelo autor.

As funcionalidades descritas tem como objetivo facilitar o uso da aplicação por pessoas com diferentes capacidades visuais e também por usuários que sejam fluentes apenas em Inglês ou Português. Dessa forma, é possível atingir uma gama maior de usuários por meio da inclusão de pessoas que possuam alguma limitação, seja ela uma barreira visual ou de idioma. Atualmente, a ferramenta limita-se a dois idiomas, porém estes podem ser estendidos facilmente graças a modularização do código-fonte.

3.4 Editor textual

O editor textual da AMeG está presente na página inicial da aplicação, sendo composto por uma área de texto que ocupa toda a área visível e um único botão para gerar o menu a partir do texto inserido. O texto permanece disponível para o usuário até a aplicação ser encerrada. A Figura 4 mostra um exemplo de menu pronto para ser gerado.

A utilização do editor textual está condicionada ao conhecimento das estruturas da linguagem AMenu, utilizada pela AMeG para geração dos menus acessíveis. Essa linguagem textual é descrita em detalhes e com exemplos práticos no Capítulo 4.

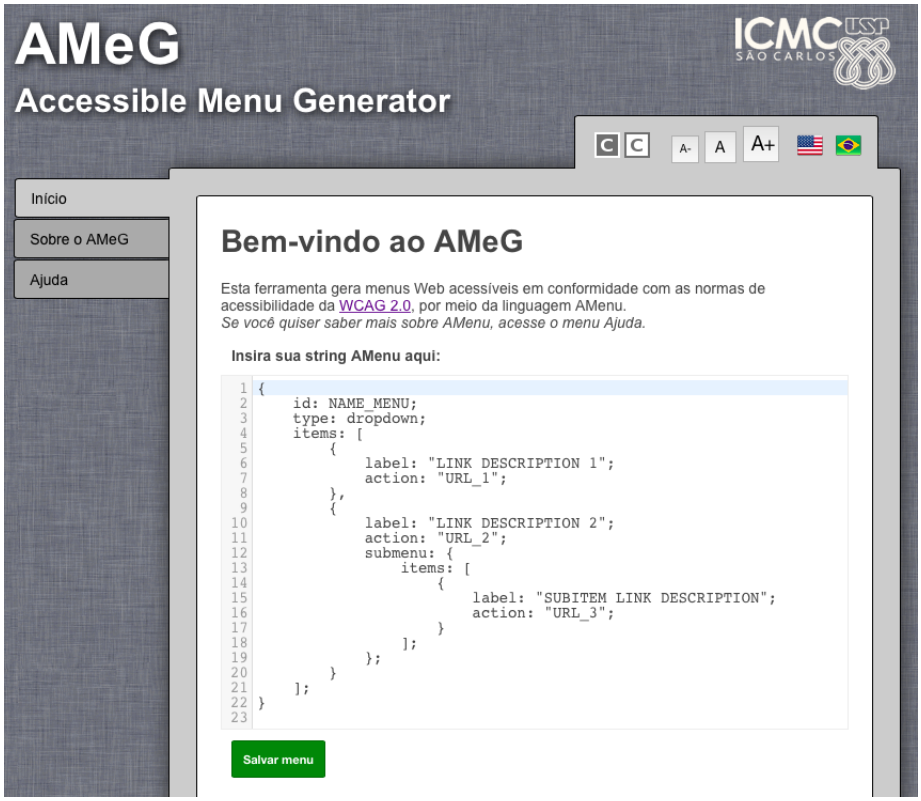
Após inserido o conteúdo do menu, de acordo com a linguagem AMenu, este pode ser gerado clicando no botão “Gerar menu”. Caso o conteúdo inserido não esteja de acordo com a estrutura exigida pela linguagem, a ferramenta retornará uma mensagem de erro e o número da linha em que o problema foi localizado. Assim, o usuário pode corrigí-lo e gerar novamente o menu.

Caso a estrutura esteja correta, uma mensagem de sucesso será apresentada na parte superior e o resultado da transformação será exibido logo abaixo da área de texto.

O resultado da transformação apresenta todas as informações necessárias de como utilizar o código-fonte do menu acessível gerado pela ferramenta, sendo este composto por três trechos de códigos, que se referem às partes: HTML[5], CSS[6] e Javascript. Na Figura 5 pode ser visto um exemplo da transformação gerada pela AMeG.

[5] *HyperText Markup Language* (HTML)
[6] *Cascading Style Sheets* (CSS)

Figura 4 – Captura de tela mostrando o editor da AMeG



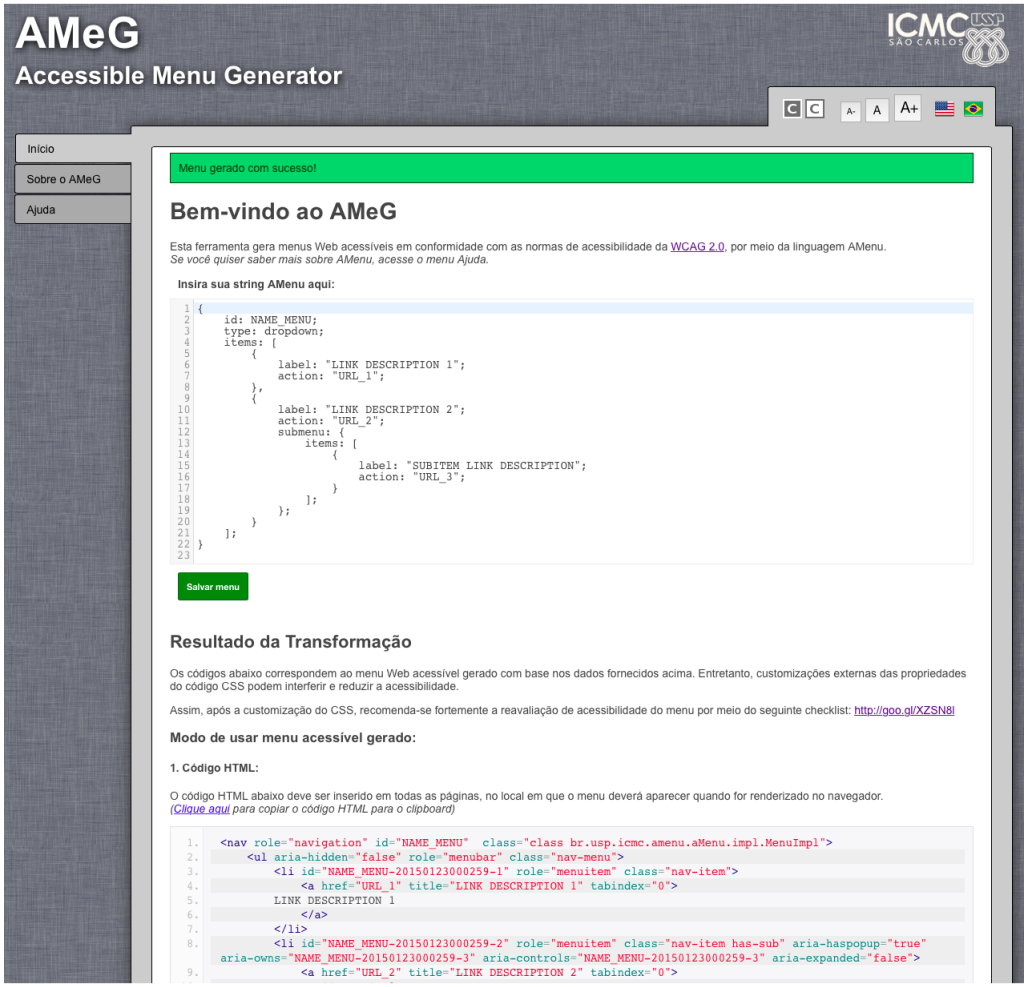
Fonte: Elaborada pelo autor.

A versão atual da AMeG restringe-se apenas ao editor textual para criação de menus acessíveis. No entanto, isto deve-se exclusivamente a restrição de tempo para implementação do protótipo gráfico e a necessidade de cumprimento do cronograma relacionado ao projeto de mestrado no qual a AMeG foi desenvolvida.

O desenvolvimento da interface textual foi priorizado de modo que fosse possível estudar o principal objetivo da ferramenta que é permitir a geração de menus de navegação Web com o máximo de acessibilidade. Contudo, uma interface gráfica que facilite a utilização da linguagem AMenu pode ser facilmente implementada e integrada a ferramenta, uma vez que se possui uma estrutura de dados central para tratamento das informações fornecidas pelo usuário.

O Capítulo 5 deste documento apresenta mais detalhes técnicos de como a AMeG está estruturada e quais os passos necessários para que se possa colaborar e contribuir com o desenvolvimento e aprimoramento deste projeto, inclusive com a implementação da interface gráfica, por exemplo.

Figura 5 – Captura de tela mostrando o editor da AMeG



Fonte: Elaborada pelo autor.

CAPÍTULO

4

UTILIZANDO A LINGUAGEM AMENU

A linguagem textual AMenu, utilizada na aplicação AMeG, foi desenvolvida para permitir a modelagem do domínio de menus de navegação Web em alto nível de abstração. Tal modelagem tem como base estudos dos conceitos sobre os elementos de navegação Web, focando nas estruturas e características específicas e necessárias para criação de menus. Além disso, inclui também estudos sobre as *guidelines* de acessibilidade da WCAG 2.0 e ARIA 1.0, a fim de compreender os requisitos de acessibilidade voltados para o contexto de navegação, bem como o modo de aplicar essas diretrizes no desenvolvimento de menus de navegação Web.

Portanto, a AMenu constitui a base para a construção de modelos de menus Web, que por sua vez servem como entrada para um conjunto de transformações que vão gerar o código de menus acessíveis. Essas transformações encapsulam o conhecimento específico de menus acessíveis e por meio delas é possível gerar o código referente a implementação dos menus Web acessíveis sem a necessidade de conhecimentos técnicos de acessibilidade pelos desenvolvedores. Além disso, ferramentas de transformação evitam a introdução de erros acidentais, tais como erros de digitação e de sintaxe, e os erros conceituais podem ser identificados em um nível mais alto de abstração.

Atualmente a linguagem encontra-se na versão 1.0, abstraindo as características fundamentais do domínio de menus. Essa versão da AMenu contém também três modelos diferentes de transformações que encapsulam os requisitos de acessibilidade necessários para esse domínio. Desse modo, a AMenu é capaz de gerar os seguintes tipos de menus acessíveis:

- *Drop-down*
- *Fly out*
- *Accordion*

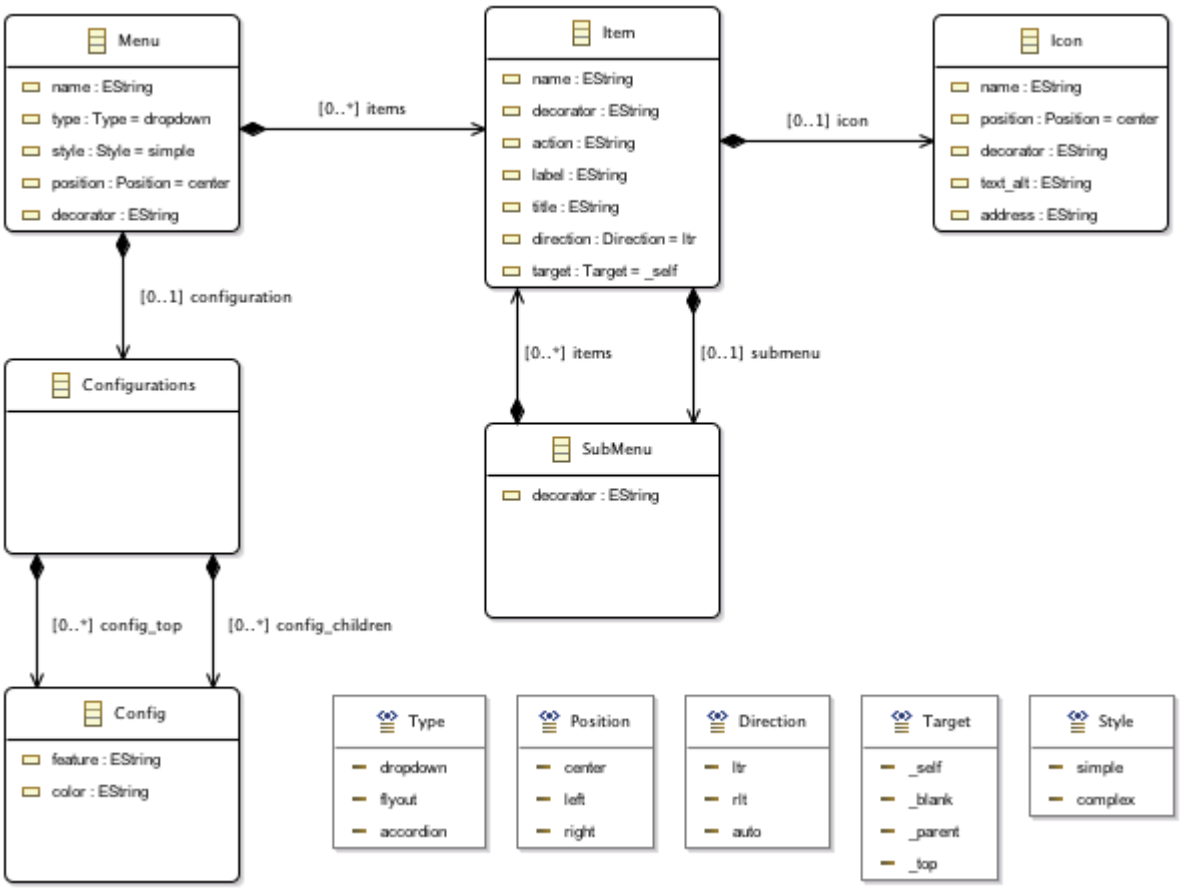
Na versão atual da AMeG, a linguagem AMenu é utilizada como uma forma de interação

puramente textual, uma vez que seu principal objetivo é garantir que qualquer desenvolvedor possa criar menus Web com o máximo de acessibilidade, sem, no entanto, ter conhecimentos específicos sobre este requisito.

4.1 Estrutura da linguagem

A AMenu foi implementada utilizando o Xtext, podendo ser facilmente estendida para abstrair novas propriedades que por ventura venham a se tornar necessárias. Apesar de sua simplicidade, a AMenu é suficiente para dar origem ao código completo de um menu de navegação Web acessível. A Figura 6 representa o metamodelo conceitual da AMenu.

Figura 6 – Metamodelo da AMenu



Fonte: Elaborada pelo autor.

De acordo com o diagrama apresentado na Figura 6, o metamodelo projetado é composto pela definição de um único menu constituído de um ou mais itens, além do tipo referente ao modelo que será gerado, pode-se incluir outros atributos opcionais, como configurações de cores, nome e estilo do modelo. Cada item do menu deve conter obrigatoriamente a ação a ser

executada (por exemplo, o redirecionamento para uma outra página) e o texto (*label*) que informa ao usuário sobre essa ação. Tais atributos compõem as informações básicas necessárias para geração da estrutura de navegação. Outros atributos como nome, título e ícone (ou imagem) são opcionais, pois são informações complementares que podem ser necessárias ou não. Além disso, um item pode ser composto por outro menu, gerando uma estrutura de dados como uma árvore.

4.2 Criando um novo modelo de menu

A estrutura sintática utilizada para a criação de um menu na linguagem AMenu é relativamente simples, uma vez que a linguagem busca-se aproximar ao máximo da sintaxe de linguagens de programação convencionais como C e Java. Um exemplo do código completo de um menu desenvolvido utilizando a AMenu, incluindo o código-fonte gerado pela AMeG, pode ser consultado no Apêndice B.

De modo geral, a especificação de todos os atributos sempre é formada iniciando-se pelo nome da propriedade, seguido do símbolo de "=", o respectivo valor do atributo e símbolo de ";" para finalizar. O Código 2 mostra um exemplo de declaração de um atributo. Um ponto que deve ser destacado é que a ordem de declaração dos atributos não sofre nenhuma influência no momento da compilação, desde que eles estejam dentro do bloco de declaração ao qual pertençam.

Código 2: Exemplo de declaração de atributo

```
1 class = "menu-dropdown";
```

Todavia, em casos que a especificação do atributo ocorrer dentro de colchetes "[···]", o símbolo de ";" não deve ser colocado, uma vez que os atributos declarados são caracterizados como uma lista de atributos e estes devem vir separados por virgula ",". Além disso, cabe ressaltar que igual a maioria das linguagens de programação, os espaços em branco e quebras de linha são ignorados no momento da compilação do código.

4.2.1 Definições iniciais

Um menu pode ser definido como um conjunto de opões apresentadas na tela, em que a seleção e execução de uma (ou mais) das opções resulta em uma mudança de estado da interface (PAAP; COOK, 1997). Assim, um dos atributos obrigatórios na declaração de um novo menu são os itens que o compõem. Além disso, deve ser especificado também o tipo de menu que se deseja gerar de acordo com as opções disponíveis na atual versão da linguagem.

Outros cinco atributos opcionais de configuração do menu também podem ser declarados, conforme a necessidade do desenvolvedor. O conjunto completo de atributos que fazem parte das

definições iniciais estão listados no Quadro 1, bem como os valores que podem ser atribuídos a cada um.

Quadro 1 – Propriedades do menu

Atributo	Valor	Tipo	Descrição
id		Opcional	Define um identificador para o menu. Todo identificador deve começar por uma letra ou "_", seguido de uma ou mais letras, números ou "_". Os identificadores são exclusivos, ou seja, não podem se repetir.
type	`dropdown`, `flyout` ou `accordion`	Obrigatório	Define o tipo de menu referente ao modelo que está sendo criado.
position	`left`, `center` ou `right`	Opcional	Define a posição que o menu deverá aparecer na aplicação. A opção `left` é utilizada como padrão caso este atributo não seja especificado.
style	`simple` ou `complex`	Opcional	Define o estilo de apresentação do menu. A opção `simple` é utilizada como padrão caso este atributo não seja especificado.
class	"$\langle value \rangle$" ou '$\langle value \rangle$'	Opcional	Define o nome das classes de CSS que o menu pode ter.
config	$[\langle value_1 \rangle, \cdots, \langle value_n \rangle]$	Opcional	Define algumas configurações de layout que o menu pode assumir, diferentes das configurações iniciais. Mais informações são apresentadas na Seção 4.2.2.
items	$[\langle value_1 \rangle, \cdots, \langle value_n \rangle]$	Obrigatório	Define os itens que constituem o menu. Maiores detalhes de como definir os itens são apresentados na Seção 4.2.3.

No Código 3 pode ser observada a declaração dos atributos iniciais. As reticências presentes nas linhas 6 e 7 representam os valores que compõem cada atributo, os quais são detalhados nas Seções 4.2.2 e 4.2.3 respectivamente.

Código 3: Exemplo de código das definições iniciais

```
1 id       = menu_principal;
2 type     = flyout;
3 position = left;
4 style    = complex;
5 class    = "default";
6 config   = [ ... ];
7 items    = [ ... ];
```

4.2.2 Configurações do menu

O atributo `config` é opcional, mas caso seja declarado deve-se especificar pelo menos um elemento na lista. Os valores dos atributos são formados por cerquilha "#" seguido de três ou seis dígitos hexadecimais, por exemplo: `#FFF` que representa a cor branca ou `#000000` que presenta a cor preta.

O conjunto completo de atributos que fazem parte das configurações do menu estão listados no Quadro 2, bem como os valores que podem ser atribuídos a cada um.

Quadro 2 – Propriedades das configurações do menu

Atributo	Valor	Tipo	Descrição
top-background	#HHH ou #HHHHHH	Opcional	Define a cor de fundo dos itens do menu principal.
top-background-hover	#HHH ou #HHHHHH	Opcional	Define a cor de fundo do item do menu principal quando este receber foco ou estiver aberto.
top-color	#HHH ou #HHHHHH	Opcional	Define a cor do texto dos itens do menu principal.
top-colo-hover	#HHH ou #HHHHHH	Opcional	Define a cor do texto do item do menu principal quando este receber foco ou estiver aberto.
top-border	#HHH ou #HHHHHH	Opcional	Define a cor da borda do menu principal.
children-background	#HHH ou #HHHHHH	Opcional	Define a cor de fundo dos itens dos submenus.
children-background-hover	#HHH ou #HHHHHH	Opcional	Define a cor de fundo do item do submenu quando este receber foco ou estiver aberto.
children-color	#HHH ou #HHHHHH	Opcional	Define a cor do texto dos itens dos submenus.
children-colo-hover	#HHH ou #HHHHHH	Opcional	Define a cor do texto do item do submenu quando este receber foco ou estiver aberto.
children-border	#HHH ou #HHHHHH	Opcional	Define a cor da borda dos submenus.

Obs.: A letra "H" deve ser substituída por um dígito em hexadecimal.

No Código 4 pode ser observada a declaração dos atributos de configuração do leiaute do menu. As reticências presentes nas linhas 5 e 18 representam os valores de outros atributos das configurações iniciais do menu.

Código 4: Exemplo de código das configurações de leiaute do menu

```
5  ...
6  config = [
7      top-background            = #4CB6EA,
8      top-background-hover      = #FFF,
9      top-color                 = #FFF,
10     top-color-hover           = #000000,
11     top-border                = #4CB6EA,
12     children-background       = #FFFFFF,
13     children-background-hover = #FFF,
14     children-color            = #000,
15     children-color-hover      = #4CB6EA,
16     children-border           = #BBBBBB
17 ];
18 ...
```

4.2.3 Itens

Os itens são os principais elementos que compõem um menu, uma vez que é por meio deles que os usuários navegam e interagem com a aplicação Web. Um item possui duas propriedades obrigatórias: `label` que descreve a ação que será executada, e `action` que contém

a ação a ser executada. Entretanto outros atributos opcionais podem ser definidos.

Os atributos declarados de cada item devem estar dentro de um bloco de conteúdo delimitado por chaves "$\{\cdots\}$". O conjunto completo de atributos que fazem parte do item do menu estão listados no Quadro 3, bem como os valores que podem ser atribuídos a cada um.

Quadro 3 – Propriedades dos itens

Atributo	Valor	Tipo	Descrição
id		Opcional	Define um identificador para o menu. Todo identificador deve começar por uma letra ou "_", seguido de uma ou mais letras, números ou "_". Os identificadores são exclusivos, ou seja, não podem se repetir.
class	"$\langle value \rangle$" ou '$\langle value \rangle$'	Opcional	Define o nome das classes de CSS que o item pode ter.
action	"$\langle value \rangle$" ou '$\langle value \rangle$'	Obrigatório	Define a URL ou a ação que o item irá executar. Caso o item contenha um submenu associado, seu valor não é obrigatório.
label	"$\langle value \rangle$" ou '$\langle value \rangle$'	Obrigatório	Define o texto que descreve o conteúdo do item.
title	"$\langle value \rangle$" ou '$\langle value \rangle$'	Opcional	Define o texto que descreve o conteúdo do item de forma mais completa. Ex. O significado de uma abreviatura.
dir	`ltr`, `rlt` ou `auto`	Opcional	Especifica a direção que o texto do item deve ser lido.
target	`_self`, `_blank`, `_top` ou `_parent`	Opcional	Indica onde será aberto o recurso (*link* ou *script*) vinculado ao item.
icon	$\langle value \rangle$	Opcional	Define uma imagem para o item. Maiores detalhes de como definir os itens são apresentados na Seção 4.2.4.
submenu	$[\langle value_1 \rangle, \cdots, \langle value_n \rangle]$	Opcional	Define um submenu para o item. Maiores detalhes de como definir os itens são apresentados na Seção 4.2.5.

No Código 5 pode ser observada a declaração dos atributos correspondente a um item do menu. A reticência presente na linha 6 representa os valores de outros atributos das configurações iniciais do menu. Nas linhas 16 e 17 as reticências representam os valores que compõem cada atributo, os quais são detalhados nas Seções 4.2.4 e 4.2.5 respectivamente.

Código 5: Exemplo de código de um item do menu

```
6  ...
7  items = [
8      {
9          id      = item_contato;
10         class   = "item-top";
11         action  = "contato.hmtl";
12         label   = "Contato";
13         title   = "Informações de contato da empresa";
14         dir     = auto;
15         target  = _self;
16         icon    = { ... };
17         submenu = { ... };
18     }
19 ];
```

4.2.4 Imagens

As imagens são caracterizadas com um atributo opcional e decorativo, sendo que a maioria dos sites não faz uso delas em seus menus. Entretanto, caso o desenvolvedor declare este atributo, além do local onde está armazenada a imagem, é obrigatória a especificação de um texto alternativo, a fim de que os usuários com deficiência visual possam identificar o conteúdo da imagem.

Assim como no itens do menu, os atributos declarados de cada imagem devem estar dentro de um bloco de conteúdo delimitado por chaves "$\{\cdots\}$". O conjunto completo de atributos que fazem parte da especificação de uma imagem estão listados no Quadro 4, bem como os valores que podem ser atribuídos a cada um.

Quadro 4 – Propriedades das imagens

Atributos	Valor	Tipo	Descrição
id		Opcional	Define um identificador para o menu. Todo identificador deve começar por uma letra ou "_", seguido de uma ou mais letras, números ou "_". Os identificadores são exclusivos, ou seja, não podem se repetir.
class	"$\langle value \rangle$" ou '$\langle value \rangle$'	Opcional	Define o nome das classes de CSS que a imagem pode ter.
position	`left`, `center` ou `right`	Opcional	Define a posição que da imagem em relação ao texto do item.
text-alt	"$\langle value \rangle$" ou '$\langle value \rangle$'	Obrigatório	Define o texto alternativo que descreve a imagem.
address	"$\langle value \rangle$" ou '$\langle value \rangle$'	Obrigatório	Define a URL onde a imagem está localizada. O

No Código 6 pode ser observada a declaração dos atributos correspondente à imagem de um item do menu. As reticências presentes nas linhas 15 e 23 representam a declaração de outros atributos pertencentes ao item.

Código 6: Exemplo de código da imagem de um item do menu

```
15 ...
16 icon = {
17     id       = img_menu_mail;
18     class    = "image-menu";
19     position  = left;
20     text-alt   = "Símbolo de e-mail";
21     address   = "imagens/mail.png";
22 };
23 ...
```

4.2.5 Submenus

Assim como os menus, o submenus são compostos obrigatoriamente de pelo menos um item, sendo que cada item pode conter apenas um único submenu. A declaração de um submenu

pelo desenvolvedor implica na geração de uma estrutura de árvore em que o menu é a raiz e os submenus as ramificações da árvore.

Em geral, os submenus em sua maioria permanecem ocultos para o usuário final, de modo que para serem visualizados deve-se interagir com o item que o agrega. Este é um ponto crucial, no que diz respeito aos requisitos de acessibilidade, visto que se não for tratado de maneira adequada, alguns usuários podem não conseguir acessar a informação, e assim não percebem a existência do submenu. Portanto, tendo em vista essa problemática, a AMenu implementa em suas transformações todos os requisitos de acessibilidade definidos pela WCAG 2.0 e pela ARIA 1.0.

No Quadro 5 pode ser visto o conjunto completo de atributos que fazem parte da especificação de um submenu, bem como os valores que podem ser atribuídos a cada um.

Quadro 5 – Propriedades dos submenus

Atributos	Valor	Tipo	Descrição
id		Opcional	Define um identificador para o menu. Todo identificador deve começar por uma letra ou "_", seguido de uma ou mais letras, números ou "_". Os identificadores são exclusivos, ou seja, não podem se repetir.
class	"$\langle value \rangle$" ou '$\langle value \rangle$'	Opcional	Define o nome das classes de CSS que o submenu pode ter.
items	$[\langle value_1 \rangle, \cdots, \langle value_n \rangle]$	Obrigatório	Define os itens que constituem o submenu. Detalhes de como definir os itens são apresentados na Seção 4.2.3.

No Código 7 pode ser observada a declaração dos atributos correspondentes a um submenu. As reticências presentes nas linhas 16 e 22 representam a declaração de outros atributos pertencentes ao item.

Código 7: Exemplo de código do submenu de um item do menu

```
16 ...
17 submenu = {
18     id      = submenu_telefone;
19     class   = "submenu";
20     items   = [ ... ];
21 };
22 ...
```

4.3 Exemplo completo de um menu definido em AMenu

Com o objetivo de demonstrar de maneira mais abrangente a utilização de AMenu, o Código 8 apresenta um exemplo completo da definição do menu com os dados utilizados nas seções anteriores. Um exemplo mais realista pode ser consultado no Apêndice B, que inclui, além

do código-fonte do menu utilizando a AMenu, os códigos-fonte em HTML, CSS e Javascript referentes à versão de uso.

Código 8: Exemplo completo de código de um menu

```
16 id        = menu_principal;
17 type      = flyout;
18 position  = left;
19 style     = complex;
20 class     = "default";
21 config = [
22     top-background            = #4CB6EA,
23     top-background-hover      = #FFF,
24     top-color                 = #FFF,
25     top-color-hover           = #000000,
26     top-border                = #4CB6EA,
27     children-background       = #FFFFFF,
28     children-background-hover = #FFF,
29     children-color            = #000,
30     children-color-hover      = #4CB6EA,
31     children-border           = #BBBBBB
32 ];
33 items = [
34     {
35         id      = item_contato;
36         class   = "item-top";
37         action  = "contato.hmtl";
38         label   = "Contato";
39         title   = "Informações de contato da empresa";
40         dir     = auto;
41         target  = _self;
42         icon    = {
43             id      = img_menu_mail;
44             class   = "image-menu";
45             position  = left;
46             text-alt   = "Símbolo de e-mail";
47             address   = "imagens/mail.png";
48         };
49         submenu = {
50             id      = submenu_telefone;
51             class   = "submenu";
52             items   = [
53                 {
54                     id      = item_contato_telefone;
55                     class   = "item-children";
56                     action  = "telefones.hmtl";
57                     label   = "Telefones";
```

```
58                          title    = "Número de telefones da empresa";
59                          dir      = auto;
60                          target   = _self;
61                          icon     = {
62                              id       = img_menu_phone;
63                              class    = "image-menu";
64                              position   = left;
65                              text-alt    = "Símbolo de telefone";
66                              address    = "imagens/phone.png";
67                          };
68                      }
69                  ];
70              };
71          }
72  ];
```

CAPÍTULO

5

INFORMAÇÕES TÉCNICAS

Este capítulo descreve os principais detalhes técnicos que compõem a AMeG, com o objetivo de facilitar que futuros desenvolvedores possam colaborar com a manutenção e aprimoramento do ambiente. Deste modo, espera-se que este projeto alcance um número cada vez maior de usuários, contribuindo assim com a criação de sites mais acessíveis por quaisquer pessoas, independente de suas limitações.

5.1 Tecnologias empregadas

A AMeG é composta por duas partes principais:

- Aplicação Web
- Linguagem AMenu

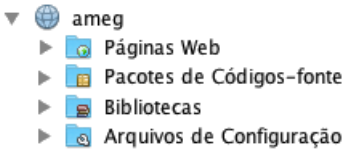
Em termos práticos, toda a infraestrutura que compõem a linguagem AMenu, utilizada pela AMeG, está contida dentro de uma biblioteca Java chamada `br.usp.icmc.amenu_1.0.jar`. Mais detalhes sobre a biblioteca e seu desenvolvimento são descritos na Seção 5.1.2.

A Figura 7 mostra a estrutura do projeto exibida pelo NetBeans[1], na qual os arquivos estão organizados em quatro categorias:

(a) Páginas Web: nesta categoria encontram-se os arquivos referentes às páginas Web da aplicação, desde os arquivos `.xhtml`, os quais são compostos pelo código HTML e *tag* do JSF, até arquivos de folhas de estilos (CSS), Javascript e as imagens que são exibidas nas páginas.

[1] http://netbeans.org/

Figura 7 – Estrutura do projeto com vista no NetBeans



Fonte: Elaborada pelo autor.

(b) Pacotes de Códigos-fonte: os arquivos de código-fonte Java da aplicação e de mapeamento estão inseridos nessa categoria.

(c) Bibliotecas: todas as bibliotecas (`.jar`) necessárias para o correto funcionamento da aplicação estão incluídas nessa categoria, inclusive a biblioteca da linguagem AMenu e suas dependências.

(d) Arquivos de Configurações: esta categoria contém atalhos criados pelo NetBeans para mostrar os arquivos de configuração espalhados pelas pastas do projeto, de modo que possam ser encontrados rapidamente. Aqui encontram-se os arquivos: *MANIFEST.MF*, *context.xml*, *faces-config.xml* e *web.xml*.

5.1.1 Aplicação Web

A AMeG faz uso do *framework* JavaServer Faces (JSF) para a construção de interfaces de usuário baseadas em componentes para aplicações Web. Além disso, para o desenvolvimento da ferramenta, utilizou-se uma arquitetura de referência que garantiu que o projeto fosse iniciado com uma base sólida e seguindo critérios de acessibilidade (FORTES *et al.*, 2013).

Dentre as páginas Web que podem ser vistas na categoria *Páginas Web*, conforme a Figura 7, tem-se:

about.xhtml: contém informações sobre a origem da ferramenta e seus autores, bem como o reconhecimento pela utilização da arquitetura de referência cedida por outros desenvolvedores e pelo apoio financeiro recebido pela Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP).

help.xhtml: página de ajuda com alguns detalhes técnicos e as instruções necessárias para auxiliar os usuários na utilização da AMeG.

index.xhtml: esta é a página inicial da AMeG e que agrega o editor textual utilizado para geração dos menus acessíveis. Nessa página é exibida, inicialmente, uma grande área de texto e um botão para gerar o menu de acordo com as informações fornecidas. Após o processamento de geração do menu, esta página pode listar as mensagens de erros

detectadas no processo, que precisam ser corrigidas para que o menu acessível possa ser efetivamente gerado, ou é apresentado o código-fonte do menu acessível gerado e como utilizá-lo.

processa.xhtml: página utilizada apenas para processamento das informações fornecidas pelo usuário na página inicial e geração do código-fonte do menu acessível. Após o processamento, a aplicação é novamente redirecionada para página inicial com o resultado produzido. Portanto, esta página não contém nenhum conteúdo a ser exibido para o usuário.

Ademais, no diretório `utils`, existem outros arquivos de páginas Web que contêm porções que se repetem em todas ou na maioria das páginas acessadas pelos usuários finais da ferramenta. Por exemplo, pode-se encontrar o menu lateral, a barra de funcionalidades no topo, o cabeçalho, rodapé e o *template* que os agregam.

5.1.2 Linguagem AMenu

A linguagem AMenu representa um artefato fundamental na abordagem proposta pela AMeG, uma vez que esta linguagem encapsula a maioria dos requisitos de acessibilidade necessários, permitindo a criação de menus de navegação Web acessíveis pelos desenvolvedores sem a necessidade de conhecimentos técnicos específicos.

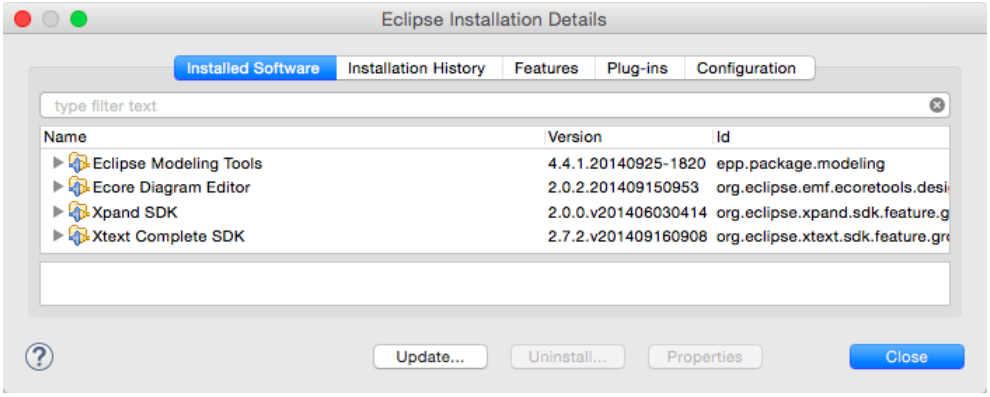
A AMeG foi desenvolvida com o fim específico de criação de menus de navegação Web acessíveis. Entretanto, a estrutura na qual a ferramenta foi projetada permite que a geração de novos elementos possa ser realizada sem muito esforço, apenas com a definição de novos modelos de transformação e aprimoramento da linguagem.

No Apêndice A encontra-se a definição completa da gramática que define a linguagem AMenu. Esta gramática é livre de contexto e está representada em notação *Extended Backus-Naur Form* (EBNF). A gramática da AMenu foi implementada utilizando o Xtext, podendo ser facilmente estendida para abstrair novas propriedades que porventura venham a se tornar necessárias.

A partir da definição da gramatica, o Xtext é capaz de gerar toda a infraestrutura da linguagem textual, as quais precisam ser extraídas em um biblioteca isolada para que possa ser incluída na aplicação Web da AMeG. Apesar de sua simplicidade, a AMenu é suficiente para dar origem ao código completo de um menu de navegação Web acessível. Todo o conhecimento relacionado a detalhes de acessibilidade são embutidos nas transformações por meio da tecnologia Xtend. Essas transformações traduzem os modelos definidos pela AMenu em código-fonte HTML, CSS e Javascript. O código produzido por meio das transformações pode ser facilmente integrado com o restante de uma aplicação Web. A seguir, encontram-se as instruções de como gerar a infraestrutura da linguagem com o Xtext e como extraí-la do projeto do Eclipse.

Para o desenvolvimento da linguagem foi utilizado o Eclipse Modeling Tools[2], uma versão da IDE Eclipse que já é distribuída gratuitamente com os pacotes e ferramentas de modelagem da *Eclipse Foundation* necessários. A Figura 8 mostra uma captura de tela com os detalhes da versão utilizada neste projeto. Recomenda-se fortemente que se utilize as versões mostradas nesta figura para se editar a gramática do Apêndice A.

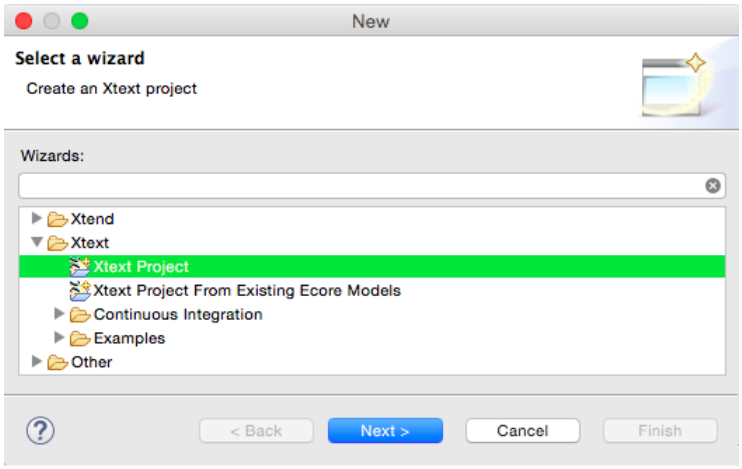
Figura 8 – Captura de tela mostrando os detalhes da instalação do Eclipse utilizado para desenvolvimento da AMenu



Fonte: Elaborada pelo autor.

Para editar a gramática, deve-se criar um novo projeto do Xtext no Eclipse, como mostrado na Figura 9.

Figura 9 – Captura de tela mostrando a criação de um novo projeto Xtext



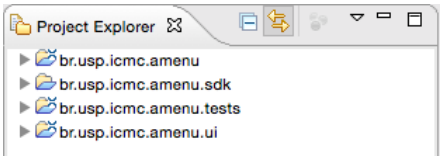
Fonte: Elaborada pelo autor.

A partir dessas definições, serão criados quatro projetos no *workspace* do Eclipse, como mostra a Figura 10. O projeto terminado com o nome da linguagem (neste caso `.amenu`) é o

[2] http://eclipse.org/downloads/packages/eclipse-modeling-tools/lunar

projeto no qual a gramática será definida e sua infraestrutura será gerada pelo Xtext. Para os propósitos da AMenu, os outros três projetos não são utilizados.

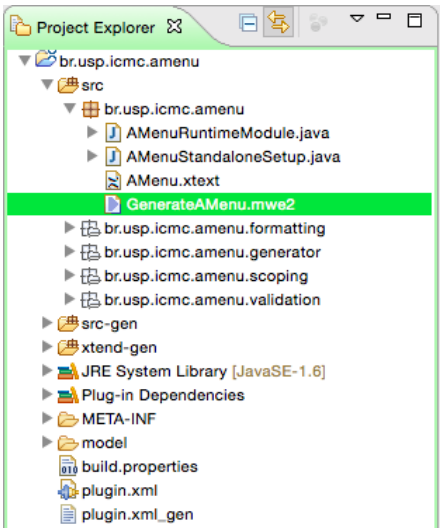
Figura 10 – Captura de tela mostrando os quatro projetos criados pelo Eclipse



Fonte: Elaborada pelo autor.

Uma vez que a gramática foi adicionada ao arquivo com extensão "`.xtext`" do projeto, pode-se executar o Xtext para dar início à geração da linguagem textual em si. Para tanto, deve-se executar o arquivo com extensão "`.mwe2`" utilizando o *MW2E Workflow*, mostrado em destaque na Figura 11. Para executá-lo, basta clicar com o botão direito do mouse sobre ele, escolher a opção "`Run As`" e em seguida selecionar a opção "`MW2E Workflow`".

Figura 11 – Captura de tela mostrando os arquivos necessários para gerar a linguagem

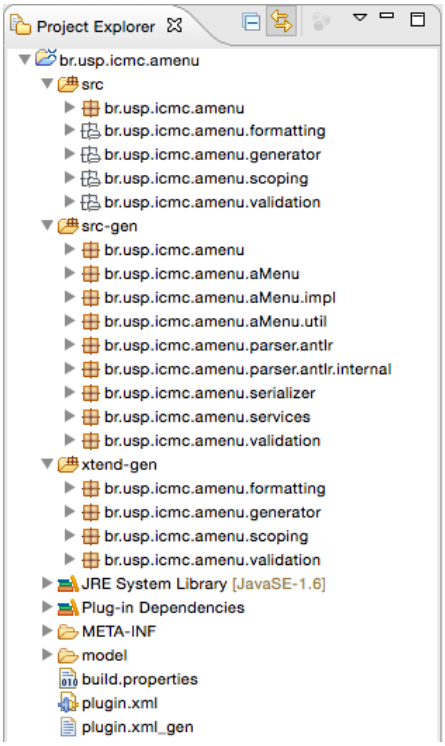


Fonte: Elaborada pelo autor.

Na primeira execução será necessário confirmar o *download* de alguns arquivos requeridos para a compilação do código pelo Xtext. Após a execução, toda a infraestrutura da linguagem textual da AMenu terá sido gerada. Os pacotes gerados podem ser vistos na Figura 12.

Após gerada a infraestrutura da linguagem, deve-se criar as transformações necessárias que são responsáveis pela geração dos códigos-fonte dos menus acessíveis. Algumas transformações já são distribuídas juntamente com o projeto da linguagem e estão localizadas dentro

Figura 12 – Captura de tela mostrando a infraestrutura gerada após a execução do *MW2E Workflow*



Fonte: Elaborada pelo autor.

do pacote "`generator`". Tais transformações são criadas utilizando o Xtend[3], cuja sintaxe da linguagem assemelha-se ao Java.

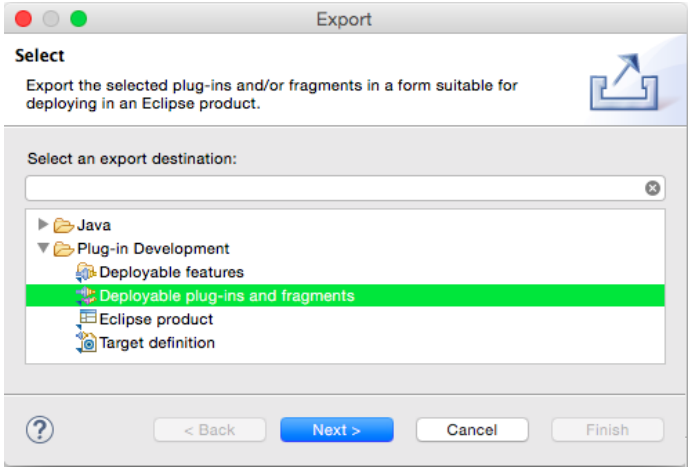
Uma vez gerada a infraestrutura da linguagem e suas transformações, deve-se exportar os pacotes como uma biblioteca JAR para que possa ser utilizada em outros projetos, incluindo a AMeG. Para isso, é preciso clicar com o botão direito do mouse sobre o projeto cujo nome termina com o nome da linguagem e em seguida seleciona a opção "`Export...`". Com isso, surgirá na tela uma nova janela, conforme mostra a Figura 13.

Deve-se, a seguir, selecionar então a opção "`Deployable plug-ins and fragments`" e clicar em "`Next`". Na janela seguinte, conforme pode ser visto na Figura 14, selecione o projeto correspondente ao qual se deseja exportar (nesse caso o projeto referente a linguagem cujo nome termina com texttt.amenu). Em seguida, clique no botão "`Finish`". Ao final do processo, será gerado um arquivo com extensão "`.jar`" contendo toda a infraestrutura da linguagem.

Entretanto, cabe ressaltar que para que o *plugin* funcione corretamente em outros projetos que for utilizado, algumas bibliotecas adicionais são necessárias, uma vez que o uso da linguagem está condicionada as dependências de parte das bibliotecas utilizadas pelo Xtext. As bibliotecas requisitadas são listadas a seguir e podem ser encontradas juntamente com a distribuição do

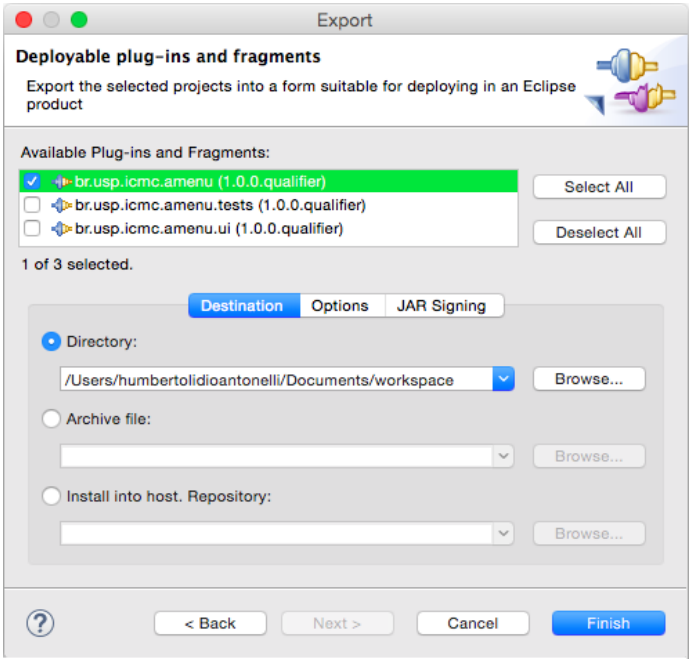
[3] http://eclipse.org/xtend/

Figura 13 – Captura de tela mostrando processo de exportação da AMenu para um *plugin* em JAVA



Fonte: Elaborada pelo autor.

Figura 14 – Captura de tela mostrando a finalização do processo de exportação da AMenu para um *plugin* em JAVA



Fonte: Elaborada pelo autor.

Eclipse, utilizada para geração da linguagem:

- com.google.guava_15.0.0.v201403281430.jar
- com.google.inject_3.0.0.v201312141243.jar
- javax.inject_1.0.0.v20091030.jar

- jstl-1.2.jar
- org.antlr.runtime_3.2.0.v201101311130.jar
- org.apache.commons.cli_1.2.0.v201404270220.jar
- org.apache.log4j_1.2.15.v201012070815.jar
- org.eclipse.emf.common_2.10.1.v20140901-1043.jar
- org.eclipse.emf.ecore.xmi_2.10.1.v20140901-1043.jar
- org.eclipse.emf.ecore_2.10.1.v20140901-1043.jar
- org.eclipse.emf.mwe.core_1.3.4.v201409021027.jar
- org.eclipse.emf.mwe.utils_1.3.4.v201409021027.jar
- org.eclipse.emf.mwe2.runtime_2.7.0.v201409021027.jar
- org.eclipse.emf_2.6.0.v20140901-1055.jar
- org.eclipse.xtend.lib.macro_2.7.2.v201409160908.jar
- org.eclipse.xtend.lib_2.7.2.v201409160908.jar
- org.eclipse.xtext.common.types_2.7.2.v201409160908.jar
- org.eclipse.xtext.util_2.7.2.v201409160908.jar
- org.eclipse.xtext.xbase.lib_2.7.2.v201409160908.jar
- org.eclipse.xtext.xbase_2.7.2.v201409160908.jar
- org.eclipse.xtext_2.7.2.v201409160908.jar

Além disso, é importante lembrar que sempre que forem realizadas alterações na gramática da linguagem AMenu, essas alterações também devem ser refletitas no código da aplicação Web, em especial nos tradutores encontrados no pacote "`functionals`" se necessário. Dependendo das mudanças realizadas na gramática da AMenu, pode haver grande impacto em todo o código da AMeG, que atualmente foi construída a partir da versão 1.0 da linguagem AMenu, que é distribuída junto com o projeto e tem sua gramática disponível no Apêndice A.

5.2 Licença

A AMeG utiliza a versão simplificada da licença BSD para permitir que a ferramenta seja distribuída e modificada como um software livre sob as condições impostas pela licença. Esta licença impõe poucas restrições quando comparada àquelas impostas por outras licenças, como a *General Public License* (GNU) ou mesmo as restrições padrão determinadas pelo *Copyright*, colocando-a relativamente próxima do domínio público.

As únicas exigências são que o nome do autor original não seja utilizado em trabalhos derivados sem permissão, visando proteger sua reputação, dado que o autor pode não ter relação

alguma com as modificações realizadas, e no caso de redistribuição do código-fonte ou binário, modificado ou não, é necessário que seja mencionado o copyright original e os termos da licença.

O Código 9 apresenta uma cópia da licença que também pode ser encontrada na raiz do projeto da AMeG e que deve ser mantida no caso de alterações em seu código e redistribuições. Entretanto, essas redistribuições não precisam estar sob a mesma licença.

Código 9: Licença BSD simplificada

```
1 Copyright (c) 2015, Humberto Lidio Antonelli
2 All rights reserved.
3
4 Redistribution and use in source and binary forms, with or without
5 modification, are permitted provided that the following conditions
6 are met:
7     * Redistributions of source code must retain the above copyright
8       notice, this list of conditions and the following disclaimer.
9     * Redistributions in binary form must reproduce the above
10      copyright notice, this list of conditions and the following
11      disclaimer in the documentation and/or other materials provided
12      with the distribution.
13    * Neither the name of the University of São Paulo nor the names
14      of its contributors may be used to endorse or promote products
15      derived from this software without specific prior written
16      permission.
17
18 THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
19 "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
20 LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS
21 FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE
22 COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT,
23 INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING,
24 BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
25 LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER
26 CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
27 LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN
28 ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE
29 POSSIBILITY OF SUCH DAMAGE.
```

5.3 Como contribuir com o projeto

Qualquer pessoa pode contribuir com o desenvolvimento e aprimoramento tanto da ferramenta AMeG quanto da linguagem AMenu, uma vez que estes projetos encontram-se hospedados no repositório do GitHub por meio dos seguintes endereços:

AMeG: https://github.com/lordantonelli/AMenu

AMenu: https://github.com/lordantonelli/AMeG

Para contribuir como os projetos, basta fazer um *fork* ou *download* do código do projeto. Alguns dos pontos que pode-se colaborar, que ainda estão fora do escopo do projeto de mestrado que motivou o desenvolvimento de AMeG e AMenu, e que foram identificados no momento da escrita deste manual, são:

- Traduzir e melhorar as mensagens de erros geradas automaticamente pelo Xtext e adicionar a coluna em que o erro foi encontrado;
- Criar uma interface gráfica para geração dos menus acessíveis, de modo que otimize a utilização da linguagem AMenu;
- Melhorar o mecanismo de customização do layout final do menu, a fim de que sejam cometidos erros posteriores de acessibilidade pelo desenvolvedor;
- Adicionar novos *templates* prontos como sugestões para os usuários da ferramenta;
- Adicionar novas transformações de modelos de menus acessíveis, como por exemplo para dispositivos móveis.

Além disso, caso haja qualquer dúvida sobre como contribuir com ambos os projetos, basta entrar em contato com os autores:

Humberto Lidio Antonelli `humbertoantonelli@usp.br`

Renata Pontin de Mattos Fortes `renata@icmc.usp.br`

REFERÊNCIAS

FORTES, R. P. *et al.* Creating a project towards universal access: is it possible? In: **INTERACT 2013 Workshop on Rethinking Universal Accessibility: A Broader Approach Considering the Digital Gap**. Cape Town, South Africa: [s.n.], 2013. Citado na página 30.

FREIRE, A. P.; RUSSO, C. M.; FORTES, R. P. A survey on the accessibility awareness of people involved in web development projects in brazil. In: ACM. **Proceedings of the 2008 international cross-disciplinary conference on Web accessibility (W4A)**. [S.l.], 2008. p. 87–96. Citado na página 6.

GRILLO, F. D. N.; FORTES, R. P. M. Tests with Blind Programmers Using AWMo: An Accessible Web Modeling Tool. In: STEPHANIDIS, C.; ANTONA, M. (Ed.). **Universal Access in Human-Computer Interaction. Design and Development Methods for Universal Access**. [S.l.]: Springer, 2014. p. 104–113. Citado na página 5.

KELLY, B. *et al.* Forcing Standardization or Accommodating Diversity?: A Framework for Applying the WCAG in the Real World. In: **Proceedings of the 2005 International Cross-Disciplinary Workshop on Web Accessibility (W4A)**. New York, NY, USA: ACM, 2005. (W4A '05), p. 46–54. ISBN 1-59593-219-4. Disponível em: <http://doi.acm.org/10.1145/1061811.1061820>. Citado na página 6.

MIÑÓN, R. *et al.* An approach to the integration of accessibility requirements into a user interface development method. **Science of Computer Programming**, v. 86, n. 0, p. 58 – 73, 2014. ISSN 0167-6423. Special issue on Software Support for User Interface Description Languages (UIDL 2011). Disponível em: <http://www.sciencedirect.com/science/article/pii/S0167642313001032>. Citado na página 6.

NICHOLL, A. R. J. O ambiente que promove a inclusão: Conceitos de acessibilidade e usabilidade. **Revista Assentamentos Humanos**, Marília - SP, v. 3, n. 2, p. 49–60, 2001. Disponível em: <http://www.unimar.br/publicacoes/assentamentos/assent_humano3v2/Antony%20e%20jose.htm>. Citado na página 5.

PAAP, K. R.; COOK, N. J. Design of menus. In: ____. **Handbook of Human-Computer Interaction**. 2. ed. New York, NY, USA: Elsevier Science Inc., 1997. cap. 24, p. 533–572. ISBN 0444818626. Citado na página 21.

RUTTER, R. *et al.* **Web Accessibility: Web Standards and Regulatory Compliance**. Apress, 2006. (Apresspod Series). ISBN 9781590596388. Disponível em: <http://books.google.com.br/books?id=dlJ94KZqwqcC>. Citado na página 5.

APÊNDICE

A

GRAMÁTICA EM NOTAÇÃO EBNF DA LINGUAGEM AMENU

Nas linhas 1 e 3 tem-se as instruções necessárias para que o Xtext possa gerar toda a infraestrutura da linguagem baseada na gramática que se inicia na linha 5.

Código 10: Definição da gramática da AMenu em Xtext

```
1 grammar br.usp.icmc.amenu.AMenu with org.eclipse.xtext.common.
    Terminals
2
3 generate aMenu "http://www.usp.br/icmc/amenu/AMenu"
4
5 Menu:
6     (
7         ( 'id''=' name=ID ';' )?
8       & ( 'type''=' type=Type ';' )
9       & ( 'style''=' style=Style ';' )?
10      & ( 'position''=' position=Position ';' )?
11      & ( 'class''=' decorator=STRING ';' )?
12      & ( 'config''=''[' configuration=Configurations ']'';' )?
13      & ( 'items''=''[' (items+=Item) (',' items+=Item)* ']'';' )
14    )
15 ;
16
17 Configurations:
18   (
19     (
20         'top' config_top+=Config
21       | 'children' config_children+=Config
22     )
23     (
```

```
24          ,''top' config_top+=Config
25        | ,''children' config_children+=Config
26      )*
27    )
28 ;
29
30 Config:
31    ( '-' feature=FEATURE '=' color=COLOR )
32 ;
33
34 Item:
35    '{'
36      (
37          ( 'id''=' name=ID ';' )?
38        & ( 'class''=' decorator=STRING ';' )?
39        & ( 'action''=' action=STRING ';' )?
40        & ( 'label''=' label=STRING ';' )
41        & ( 'title''=' title=STRING ';' )?
42        & ( 'dir''=' direction=Direction ';' )?
43        & ( 'target''=' target=Target ';' )?
44        & ( 'icon''=' icon=Icon ';')?
45        & ( 'submenu''=' submenu=SubMenu ';')?
46      )
47    '}'
48 ;
49
50 SubMenu:
51    '{'
52      (
53          ( 'id = ' name=ID ';' )?
54        & ( 'class''=' decorator=STRING ';' )?
55        & ( 'items''=''[' ((items+=Item) (',' items+=Item)*) ']'';' )
56      )
57    '}'
58 ;
59
60 Icon:
61    '{'
62      (
63          ( 'id''=' name=ID ';' )?
64        & ( 'position''=' position=Position ';')?
65        & ( 'class''=' decorator=STRING ';')?
66        & ( 'text-alt''=' text_alt=STRING ';' )
67        & ( 'address''=' address=STRING ';' )
68      )
69    '}'
70 ;
```

```

enum Type:
      dropdown
    | flyout
    | accordion
;

enum Style:
      simple
    | complex
;

enum Position:
    center
  | left
  | right
;

enum Direction:
    ltr
  | rlt
  | auto
;

enum Target:
    _self
  | _blank
  | _parent
  | _top
;

terminal FEATURE:
  (
      'background'
    | 'background-hover'
    | 'color'
    | 'color-hover'
    | 'border'
  )
;

terminal COLOR: '#'(HEX)(HEX)(HEX)((HEX)(HEX)(HEX))? ;

terminal HEX: ('a'..'h'|'A'..'H'|'0'..'9') ;
```

APÊNDICE

B

EXEMPLO COMPLETO DE UM MENU WEB ACESSÍVEL

O Código 11 corresponde a um menu Web escrito na linguagem AMenu. A compilação deste código-fonte pela ferramenta AMeG dará origem ao menu da Figura 15.

Código 11: Código do menu na linguagem AMenu

```
1  id       = menu_viagem;
2  type     = dropdown;
3  style    = complex;
4  position = center;
5  items    = [
6    {
7      action  = "index.html";
8      label   = "Início";
9    },
10   {
11     label   = "Guias e mapas";
12     action  = "guias-mapas.html";
13   },
14   {
15     action  = "dicas-viagem.html";
16     label   = "Dicas de Viagem";
17     submenu = {
18       items   = [
19         {
20           action  = "ecoturismo.html";
21           label   = "Ecoturismo";
22         },
23         {
24           action  = "centros-urbanos.hmtl";
25           label   = "Centros Urbanos";
```

```
26          },
27          {
28            action   = "estudantes.hmtl";
29            label    = "Estudantes";
30          },
31          {
32            action   = "familia.hmtl";
33            label    = "Família e Crianças";
34          },
35          {
36            action   = "lugares-romanticos.hmtl";
37            label    = "Lugares Românticos";
38          },
39          {
40            action   = "turismo-cultural.hmtl";
41            label    = "Turismo Cultural";
42          }
43        ];
44      };
45    },
46    {
47      action   = "sobre.hmtl";
48      label    = "Sobre o site";
49    },
50    {
51      action   = "contato.hmtl";
52      label    = "Contato";
53    }
54  ];
```

Os Códigos 12, 13 e 14 correspondem aos códigos-fontes em Javascript, CSS e HTML respectivamente gerados pela ferramenta AMeG, que devem ser inseridos na aplicação Web que está sendo desenvolvida de acordo com as instruções presentes na ferramenta.

Código 12: Código Javascript gerado pela AMeG

```
1 <script src="http://code.jquery.com/jquery-1.11.0.min.js"></script>
2 <script src="http://agua.intermidia.icmc.usp.br:8080/ameg/plugins/
    dropdown.js"></script>
3 <script>
4     $(document).ready(function() {
5         $("#menu_viagem").accessibleMenu();
6     });
7 </script>
```

Código 13: Código CSS gerado pela AMeG

```
1 <style type="text/css">
2     #menu_viagem,
3     #menu_viagem ul,
4     #menu_viagem ul li,
5     #menu_viagem ul li a,
6     #menu_viagem ul li span {
7       margin: 0;
8       padding: 0;
9       border: 0;
10      list-style: none;
11      line-height: 1;
12      display: block;
13      position: relative;
14      -webkit-box-sizing: border-box;
15      -moz-box-sizing: border-box;
16      box-sizing: border-box;
17    }
18    #menu_viagem:after,
19    #menu_viagem > ul:after {
20      content: ".";
21      display: block;
22      clear: both;
23      visibility: hidden;
24      line-height: 0;
25      height: 0;
26    }
27    #menu_viagem {
28      width: auto;
29      background: #4cb6ea;
30    }
31
32    #menu_viagem > ul > li {
33      float: left;
34      display: inline-block;
35    }
36    #menu_viagem > ul > li > a ,
37    #menu_viagem > ul > li > span {
38      padding: 10px 25px 13px 25px;
39      border-top: 2px solid #4cb6ea;
40      border-bottom: 2px solid #4cb6ea;
41      border-right: 2px solid #4cb6ea;
42      text-decoration: none;
43      color: #FFFFFF;
44      cursor:pointer;
45    }
46    #menu_viagem > ul > li:first-child > a ,
```

```
47    #menu_viagem > ul > li:first-child > span {
48      border-left: 2px solid #4cb6ea;
49    }
50    #menu_viagem > ul > li.open > a,
51    #menu_viagem > ul > li.focus > a,
52    #menu_viagem > ul > li:hover > a,
53    #menu_viagem > ul > li > a:hover,
54    #menu_viagem > ul > li.active > a,
55    #menu_viagem > ul > li.open > span,
56    #menu_viagem > ul > li.focus > span,
57    #menu_viagem > ul > li:hover > span,
58    #menu_viagem > ul > li > span:hover,
59    #menu_viagem > ul > li.active > span {
60      color: #000000;
61      background: #FFFFFF;
62    }
63    #menu_viagem > ul > li.has-sub > a,
64    #menu_viagem > ul > li.has-sub > span {
65      padding-right: 45px;
66    }
67    #menu_viagem > ul > li.has-sub::after {
68      content: "";
69      position: absolute;
70      right: 15px;
71      top: 13px;
72      width: 8px;
73      height: 8px;
74      border-bottom: 2px solid #FFFFFF;
75      border-right: 2px solid #FFFFFF;
76      content: "";
77      -webkit-transform: rotate(45deg);
78      -moz-transform: rotate(45deg);
79      -ms-transform: rotate(45deg);
80      -o-transform: rotate(45deg);
81      transform: rotate(45deg);
82      -moz-transition: all .3s ease;
83      -ms-transition: all .3s ease;
84      -o-transition: all .3s ease;
85      transition: all .3s ease;
86      -webkit-transition: all 0.3s ease;
87    }
88    #menu_viagem > ul > li.has-sub.focus::after,
89    #menu_viagem > ul > li.has-sub:hover::after,
90    #menu_viagem > ul > li.has-sub.active::after{
91      border-bottom-color: #000000;
92      border-right-color: #000000;
93    }
```

```
94    #menu_viagem > ul > li.has-sub.open::after {
95      border-bottom-color: #4cb6ea;
96      border-right-color: #4cb6ea;
97      background: #FFFFFF;
98      top: 36px;
99      right: 14px;
100     width: 10px;
101     height: 10px;
102   }
103   #menu_viagem ul ul {
104     position: absolute;
105     left: -9999px;
106     top: 60px;
107     padding-top: 8px;
108     font-size: 13px;
109     opacity: 0;
110     -webkit-transition: opacity 0.2s ease-in;
111     -moz-transition: opacity 0.2s ease-in;
112     -ms-transition: opacity 0.2s ease-in;
113     -o-transition: opacity 0.2s ease-in;
114     transition: opacity 0.2s ease-in;
115   }
116   #menu_viagem > ul > li > ul {
117     top: 120px;
118   }
119   #menu_viagem > ul > li.open > ul {
120     top: 100%;
121     left: 0;
122     opacity: 1;
123   }
124   #menu_viagem ul ul ul {
125     padding-top: 0;
126     padding-left: 8px;
127   }
128   #menu_viagem ul ul > li.open > ul {
129     display:block;
130     left: 180px;
131     top: 0;
132     opacity: 1;
133   }
134   #menu_viagem ul ul li a,
135   #menu_viagem ul ul li span {
136     text-decoration: none;
137     padding: 11px 25px;
138     width: 180px;
139     color: #000000;
140     background: #FFFFFF;
```

```
141       box-shadow: 0 2px 2px rgba(0, 0, 0, 0.1), 1px 1px 1px rgba(0,
      0, 0, 0.1), -1px 1px 1px rgba(0, 0, 0, 0.1);
142       border-left: 1px solid #BBBBBB;
143       border-right: 1px solid #BBBBBB;
144       cursor:pointer;
145     }
146
147     #menu_viagem ul ul li.open > a,
148     #menu_viagem ul ul li.focus > a,
149     #menu_viagem ul ul li:hover > a,
150     #menu_viagem ul ul li.active > a,
151     #menu_viagem ul ul li.open > span,
152     #menu_viagem ul ul li.focus > span,
153     #menu_viagem ul ul li:hover > span,
154     #menu_viagem ul ul li.active > span {
155       color: #4cb6ea;
156       background:#FFFFFF;
157     }
158     #menu_viagem ul ul li:first-child > a,
159     #menu_viagem ul ul li:first-child > span {
160       border-top-left-radius: 3px;
161       border-top-right-radius: 3px;
162       border-top: 1px solid #BBBBBB;
163     }
164     #menu_viagem ul ul li:last-child > a,
165     #menu_viagem ul ul li:last-child > span {
166       border-bottom-left-radius: 3px;
167       border-bottom-right-radius: 3px;
168       border-bottom: 1px solid #BBBBBB;
169     }
170     #menu_viagem > ul > li > ul::after {
171       position: absolute;
172       display: block;
173     }
174     #menu_viagem ul ul li.has-sub > a::after,
175     #menu_viagem ul ul li.has-sub > span::after {
176       content: "";
177       position: absolute;
178       width: 6px;
179       height: 6px;
180       border-bottom: 2px solid #000000;
181       border-right: 2px solid #000000;
182       -webkit-transform: rotate(-45deg);
183       -moz-transform: rotate(-45deg);
184       -ms-transform: rotate(-45deg);
185       -o-transform: rotate(-45deg);
186       transform: rotate(-45deg);
```

```
187       right: 17px;
188       top: 14px;
189       -moz-transition: all .2s ease;
190       -ms-transition: all .2s ease;
191       -o-transition: all .2s ease;
192       transition: all .2s ease;
193       -webkit-transition: -webkit-transform 0.2s ease, right 0.2s
      ease;
194     }
195     #menu_viagem ul ul li.has-sub.focus > a::after,
196     #menu_viagem ul ul li.has-sub:hover > a::after,
197     #menu_viagem ul ul li.has-sub.focus > span::after,
198     #menu_viagem ul ul li.has-sub:hover > span::after {
199       border-bottom-color: #4cb6ea;
200       border-right-color: #4cb6ea;
201     }
202     #menu_viagem ul ul li.has-sub.open > a::after,
203     #menu_viagem ul ul li.has-sub.open > span::after {
204       right: -6px;
205       top: 11px;
206       width: 10px;
207       height: 10px;
208       background:#FFFFFF;
209       border-bottom: 1px solid #BBBBBB000;
210       border-right: 1px solid #BBBBBB000;
211     }
212
213     #menu_viagem.align-center > ul {
214       float: none;
215       text-align: center;
216     }
217     #menu_viagem.align-center > ul > li {
218       float: none;
219       margin-left: -4px;
220     }
221     #menu_viagem.align-center ul ul {
222       text-align: left;
223     }
224     #menu_viagem.align-right ul ul {
225       text-align: right;
226     }
227     #menu_viagem.align-right > ul > li:first-child > a,
228     #menu_viagem.align-right > ul > li:first-child > span {
229       border-left-style: none;
230     }
231     #menu_viagem.align-right > ul > li:last-child > a,
232     #menu_viagem.align-right > ul > li:last-child > span {
```

```
233         border-left: 2px solid #4cb6ea;
234       }
235       #menu_viagem.align-right > ul > li.open > ul,
236       #menu_viagem.align-right > ul > li:hover > ul {
237         left: auto;
238         right: 0;
239       }
240       #menu_viagem.align-right ul ul ul {
241         padding-right: 8px;
242       }
243       #menu_viagem.align-right ul ul > li.open > ul,
244       #menu_viagem.align-right ul ul > li:hover > ul {
245         left: auto;
246         right: 100%;
247         opacity: 1;
248       }
249       #menu_viagem.align-right > ul > li {
250         float: right;
251       }
252       #menu_viagem.align-right ul ul li.has-sub > a::after,
253       #menu_viagem.align-right ul ul li.has-sub > span::after {
254         right: auto;
255         left: 17px;
256         -webkit-transform: rotate(135deg);
257         -moz-transform: rotate(135deg);
258         -ms-transform: rotate(135deg);
259         -o-transform: rotate(135deg);
260         transform: rotate(135deg);
261       }
262       #menu_viagem.align-right ul ul li.has-sub.open > a::after,
263       #menu_viagem.align-right ul ul li.has-sub.active > a::after,
264       #menu_viagem.align-right ul ul li.has-sub.open > span::after,
265       #menu_viagem.align-right ul ul li.has-sub.active > span::after {
266         left:-6px;
267       }
268 </style>
```

Código 14: Código HTML gerado pela AMeG

```
1 <nav role="navigation" id="menu_viagem"  class="align-center">
2     <ul aria-hidden="false" role="menubar" class="nav-menu">
3         <li id="menu_viagem-20150123105857-1" role="menuitem" class="
    nav-item">
4             <a href="index.html" title="Início" tabindex="0">
5         Início
6             </a>
7         </li>
```

```
8        <li id="menu_viagem-20150123105857-2" role="menuitem" class="
    nav-item">
9            <a href="guias-mapas.html" title="Guias e mapas"
    tabindex="0">
10       Guias e mapas
11           </a>
12       </li>
13       <li id="menu_viagem-20150123105857-3" role="menuitem" class="
    nav-item has-sub" aria-haspopup="true" aria-owns="
    menu_viagem-20150123105857-4" aria-controls="
    menu_viagem-20150123105857-4" aria-expanded="false">
14           <a href="dicas-viagem.html" title="Dicas de Viagem"
    tabindex="0">
15       Dicas de Viagem
16           </a>
17           <ul id="menu_viagem-20150123105857-4" aria-hidden="true"
    role="menu" aria-expanded="false" aria-labelledby="
    menu_viagem-20150123105857-3" class="sub-nav-group sub-nav class
    br.usp.icmc.amenu.aMenu.impl.SubMenuImpl">
18               <li id="menu_viagem-20150123105857-5" role="menuitem"
     class="nav-item">
19                   <a href="ecoturismo.html" title="Ecoturismo"
    tabindex="-1">
20               Ecoturismo
21                   </a>
22               </li>
23               <li id="menu_viagem-20150123105857-6" role="menuitem"
     class="nav-item">
24                   <a href="centros-urbanos.hmtl" title="Centros
    Urbanos" tabindex="-1">
25               Centros Urbanos
26                   </a>
27               </li>
28               <li id="menu_viagem-20150123105857-7" role="menuitem"
     class="nav-item">
29                   <a href="estudantes.hmtl" title="Estudantes"
    tabindex="-1">
30               Estudantes
31                   </a>
32               </li>
33               <li id="menu_viagem-20150123105857-8" role="menuitem"
     class="nav-item">
34                   <a href="familia.hmtl" title="Família e Crianças"
     tabindex="-1">
35               Família e Crianças
36                   </a>
37               </li>
```

```
38                  <li id="menu_viagem-20150123105857-9" role="menuitem"
     class="nav-item">
39                      <a href="lugares-romanticos.hmtl" title="Lugares
    Românticos" tabindex="-1">
40                  Lugares Românticos
41                      </a>
42                  </li>
43                  <li id="menu_viagem-20150123105857-10" role="menuitem
    " class="nav-item">
44                      <a href="turismo-cultural.hmtl" title="Turismo
    Cultural" tabindex="-1">
45                  Turismo Cultural
46                      </a>
47                  </li>
48              </ul>
49          </li>
50          <li id="menu_viagem-20150123105857-11" role="menuitem" class=
    "nav-item">
51              <a href="sobre.hmtl" title="Sobre o site" tabindex="0">
52          Sobre o site
53              </a>
54          </li>
55          <li id="menu_viagem-20150123105857-12" role="menuitem" class=
    "nav-item">
56              <a href="contato.hmtl" title="Contato" tabindex="0">
57          Contato
58              </a>
59          </li>
60      </ul>
61  </nav>
```

A Figura 15 mostra o menu gerado pela ferramenta AMeG renderizado no navegador.

Figura 15 – Captura de tela mostrando o menu renderizando no navegador

Fonte: Elaborada pelo autor.