

---

Estudo de caso - Modelagem de software acessível  
na Web

*Filipe Del Nero Grillo*  
*Prof<sup>a</sup>. Dr<sup>a</sup>. Renata Pontin de Mattos Fortes*

---



# Sumário

---

---

<b>1</b>	<b>Introdução</b>	<b>1</b>
1.1	Contexto e Motivação . . . . .	1
1.2	Objetivos . . . . .	2
1.3	Organização do estudo de caso . . . . .	3
<b>2</b>	<b>Trabalhos relacionados</b>	<b>5</b>
<b>3</b>	<b>Protocolo do estudo de caso</b>	<b>9</b>
3.1	Questões de pesquisa . . . . .	9
3.2	Estratégia de seleção . . . . .	10
3.3	Coleta de dados . . . . .	10
3.3.1	Entrevistas . . . . .	10
3.3.2	Observação . . . . .	11
3.3.3	Medições e métricas . . . . .	12
3.4	Triangulação . . . . .	13
3.5	Ameaças à validade . . . . .	14
3.6	Piloto . . . . .	16
3.7	Plano de Análise . . . . .	19
<b>4</b>	<b>Resultados</b>	<b>21</b>
4.1	Arthur . . . . .	21
4.2	Ford . . . . .	27
4.3	Análise . . . . .	33
<b>5</b>	<b>Conclusões e trabalhos futuros</b>	<b>39</b>
5.1	Conclusões . . . . .	39
5.2	Trabalhos futuros . . . . .	41
	<b>Referências</b>	<b>45</b>
<b>A</b>	<b>Guia para entrevista pré-uso</b>	<b>45</b>
A.1	Assuntos a serem abordados . . . . .	45
A.2	Questões . . . . .	45

<b>B</b>	<b>Tarefas para a fase de observação</b>	<b>49</b>
B.1	Treinamento na linguagem textual . . . . .	49
B.2	Lista de objetivos . . . . .	49
<b>C</b>	<b>Guia para entrevista pós-uso</b>	<b>53</b>
C.1	Assuntos a serem abordados . . . . .	53
C.2	Questões . . . . .	53
<b>D</b>	<b>Carta-convite</b>	<b>55</b>
<b>E</b>	<b>Termo de confidencialidade</b>	<b>57</b>
<b>F</b>	<b>Transcrições das entrevistas com o piloto</b>	<b>61</b>
F.1	Pré-entrevista . . . . .	61
F.2	Pós-entrevista . . . . .	66
<b>G</b>	<b>Transcrições das entrevistas - Arthur</b>	<b>69</b>
G.1	Pré-entrevista . . . . .	69
G.2	Pós-entrevista . . . . .	72
<b>H</b>	<b>Transcrições das entrevistas - Ford</b>	<b>75</b>
H.1	Pré-entrevista . . . . .	75
H.2	Pós-entrevista . . . . .	81

---

# Lista de Figuras

---

1.1	Captura de tela da AWMo com um diagrama de classes aberto para edição em modo textual. . . . .	3
3.1	Foto da configuração do ambiente do experimento, conforme foi conduzido durante a execução piloto. O computador da direita, utilizado pelo sujeito, executa o Morae <i>Recorder</i> e o computador à esquerda, utilizado pelo pesquisador, executa o Morae <i>Observer</i> . . . . .	13
4.1	Gráfico mostrando a quantidade de tempo utilizado por Arthur para completar cada um dos objetivos propostos neste estudo de caso. . . . .	23
4.2	Gráfico extraído da gravação de vídeo da fase de observação contendo o número de falhas inseridas por Arthur ao decorrer da execução de cada objetivo. . . . .	25
4.3	Gráfico extraído da gravação de vídeo da fase de observação contendo o número de vezes em que mensagens de erro do interpretador foram exibidas para Arthur separados por objetivo. . . . .	26
4.4	Captura de tela de um dos momentos em que o interpretador retornou mensagens de erro para Arthur. Foi aplicado um mosaico sobre o rosto de Arthur para preservar sua identidade. . . . .	26
4.5	Gráfico mostrando a quantidade de tempo utilizado por Ford para completar os objetivos propostos neste estudo de caso. . . . .	29
4.6	Gráfico mostrando o número de falhas inseridas por Ford ao decorrer da execução dos objetivos. . . . .	30
4.7	Gráfico mostrando o número de vezes em que mensagens de erro do interpretador de texto foram exibidas para Ford, separados por objetivo. . . . .	31
4.8	Gráfico mostrando a quantidade de tempo utilizado por cada um dos sujeitos para completar os objetivos propostos neste estudo de caso. . . . .	33
4.9	Gráfico mostrando o número de falhas inseridas por cada um sujeitos ao decorrer da execução dos objetivos. . . . .	34
4.10	Gráfico mostrando o número de vezes em que mensagens de erro do interpretador de texto foram exibidas para os sujeitos, separados por objetivo. . . . .	35

4.11 Gráficos ilustrando a categorização das falhas exibidas nas Tabelas 4.1 e 4.2 sendo: (a) - quantidade de falhas de Arthur em cada categoria e separadas por objetivo, (b) - quantidade de falhas de Ford em cada categoria e separadas por objetivo, (c) - proporção entre as categorias das falhas cometidas por Arthur e (d) - proporção entre as categorias das falhas cometidas por Arthur . . . . . 38

# Lista de Tabelas

---

---

3.1	Ameaças à validade interna . . . . .	15
3.2	Ameaças à validade externa . . . . .	16
4.1	Categorização das falhas inseridas por Arthur ao longo do estudo. . . . .	36
4.2	Categorização das falhas inseridas por Ford ao longo do estudo. . . . .	37





---

# Introdução

---

## 1.1 Contexto e Motivação

Atualmente no Brasil, o número de pessoas que possuem algum tipo de deficiência é de 45,62 milhões. De acordo com o Censo 2010, este número representa cerca de 23,92% de toda a população do país (IBGE, 2010).

Com o aumento de serviços de utilidade pública sendo oferecidos pelo governo por meio da Internet e outros meios eletrônicos, tendência chamada de *e-gov* (sigla para governo eletrônico), aumenta-se a preocupação com relação ao acesso desses cidadãos. Por esse motivo, existem algumas iniciativas que buscam melhorar a acessibilidade desses serviços tais como a portaria e-MAG (Modelo de Acessibilidade em Governo Eletrônico) (Santanna, 2007) entre outras leis e decretos que abordam desde meios eletrônicos como o e-MAG quanto o acesso a ambientes físicos, como a obrigatoriedade da construção de rampas para pessoas que utilizam cadeiras de rodas.

Mais especificamente no âmbito da computação, os deficientes visuais têm historicamente uma participação mais ativa pois os códigos de programas de computadores são essencialmente no formato de texto e, portanto, mais facilmente acessíveis por meio de tecnologias assistivas tais como leitores de tela e mostradores de braille, por exemplo.

No entanto, mesmo com diversas iniciativas, alguns tipos de dados ainda permanecem intrinsecamente dependentes de sentidos específicos, como a visão. Os modelos de *software* por exemplo, são geralmente representados na forma de diagramas visuais, usualmente compostos por formas geométricas e conectores ligando essas formas geométricas. Na maioria das vezes também existe conteúdo escrito nestes diagramas, porém o significado

das notações visuais é tão grande que por meio da leitura do texto apenas, o entendimento do conteúdo fica bastante prejudicado.

Além disso, com o crescimento da disciplina de engenharia de *software* e linguagens visuais como a UML (*Unified Modeling Language*) e consequente aumento de atividades de modelagem que utilizam essas linguagens visuais, o acesso de deficientes visuais fica prejudicado, sendo necessário o auxílio de outras pessoas tanto para a leitura quanto para a edição desses modelos.

Neste contexto, foi desenvolvida a AWMo (*Accessible Web Modeler*), uma ferramenta Web desenhada para permitir que seus usuários modelem diagramas de classe da UML em duas visões distintas: a *visão gráfica* na qual desenvolvedores de *software* com visão podem desenhar os diagramas de maneira tradicional, arrastando os elementos e conectando-os visualmente e a *visão textual*, em que desenvolvedores de *software* com ou sem deficiência visual podem acessar e desenhar os mesmos diagramas utilizando uma linguagem textual desenvolvida especialmente para a AWMo.

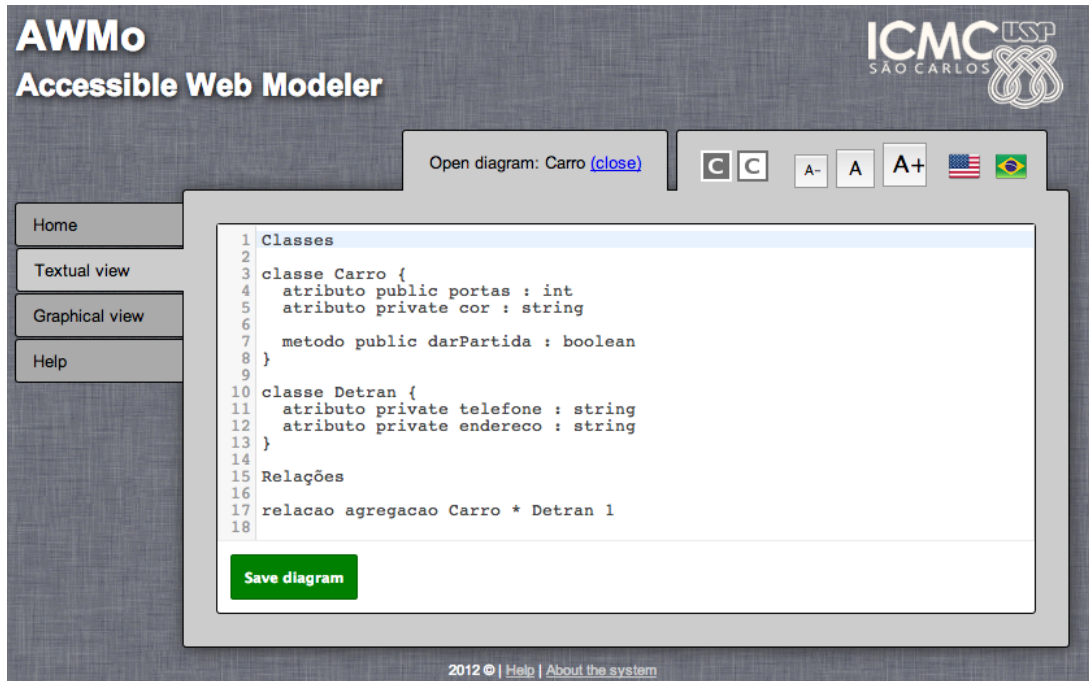
O conceito chave da abordagem proposta na AWMo é o de prover duas visões diferentes para a edição do mesmo modelo, contanto que ambas as visões permitam que os usuários realizem as mesmas tarefas e alcancem os mesmos resultados. Dessa forma, se um usuário deficiente visual criar um novo diagrama de classes na AWMo, editá-lo e salvá-lo, um usuário com visão poderá abrir o mesmo diagrama na visão gráfica e continuar o trabalho ou realizar alterações nele que, por sua vez, serão novamente acessíveis na visão textual e vice-versa.

A Figura 1.1 ilustra a tela do editor textual da AWMo. Pode-se notar que no menu à esquerda é possível alternar entre o modo visual e o modo gráfico de edição para o mesmo diagrama que encontra-se aberto para edição.

A AWMo foi desenvolvida como parte do projeto de mestrado do autor, intitulado “*Uma ferramenta acessível de apoio à modelagem de software na Web*” no ICMC-USP sob orientação da Profa. Dra. Renata Pontin de Mattos Fortes.

## 1.2 Objetivos

Neste relatório, o objetivo é descrever um estudo de caso que foi realizado para avaliar a viabilidade da proposta da AWMo. No estudo de caso, a AWMo proposta foi objeto de investigação, para verificar se ela permite acesso a modelos de *software* para deficientes visuais, bem como se a estratégia de uso da AWMo proposta é apropriada para ser utilizada em ambientes reais, tais como empresas que trabalham com desenvolvimento de *software* e empregam profissionais com deficiência visual.



**Figura 1.1:** Captura de tela da AWMo com um diagrama de classes aberto para edição em modo textual.

### 1.3 Organização do estudo de caso

Neste capítulo foram apresentadas as considerações iniciais sobre o estudo de caso realizado, o contexto no qual ele se insere e seus objetivos.

O restante deste relatório é organizado da seguinte forma: o Capítulo 2 ilustra outros trabalhos que foram realizados no mesmo contexto ou similares ao proposto. O Capítulo 3 apresenta o protocolo deste estudo de caso. O Capítulo 4 mostra os dados coletados durante a execução do estudo com os sujeitos selecionados. Por fim, o Capítulo 5 apresenta as conclusões obtidas com a realização deste trabalho e possibilidades de trabalhos futuros. O Apêndice A possui um guia que foi utilizado para a realização da entrevista inicial, o Apêndice B contém as tarefas que foram realizadas pelos sujeitos durante a fase de observação deste estudo de caso. O Apêndice C contém um guia utilizado para a realização da entrevista posterior à observação. O Apêndice D contém a carta-convite que foi enviada para os potenciais sujeitos deste estudo. O Apêndice E contém o termo de confidencialidade que foi assinado pelos pesquisadores e participantes a fim de assegurar a anonimidade dos dados que serão publicados. Por fim, os Apêndices F, G e H contém as transcrições das entrevistas conduzidas.



---

## Trabalhos relacionados

---

Encontram-se na Literatura diversos trabalhos que abordam a acessibilidade com focos bastante distintos. Alguns exploram novas formas de interação para permitir o acesso, como é o caso do trabalho de King et al. (2004), que buscou possibilitar a visualização e navegação em diagramas UML para deficientes visuais. Nesse projeto foram utilizados quatro dos doze diagramas UML e explorou-se a interação por meio de *joysticks* e leitores de tela para que o usuário pudesse navegar no espaço do diagrama e escutar seu conteúdo. O sistema, intitulado TeDUB, foi avaliado com 34 usuários cegos ou com baixa visão, dos quais 11 eram estudantes de computação e 23 eram profissionais na área. Uma demanda que surgiu dos usuários foi a possibilidade de criação e edição de diagramas (King et al., 2004). Esta demanda por edição e criação dos diagramas é o ponto que a AWMo pretende atender.

Os trabalhos de Metatla et al. abordam a exploração não visual de informações representadas graficamente, como é o caso em que é utilizada uma representação hierárquica para diagramas relacionais de maneira que eles podem ser explorados por meio de interfaces de áudio. Em um outro trabalho foram realizadas comparações entre estratégias para construção de diagramas em interfaces que utilizam apenas áudio (Metatla et al., 2008).

Em (Siegfried, 2006), foi desenvolvida uma linguagem de *scripting* para permitir que deficientes visuais consigam trabalhar com paradigmas visuais de programação, no caso a linguagem BASIC.

O trabalho de Cohen et al. (2006) desenvolveu uma ferramenta chamada PLUMB com o objetivo de ensinar grafos para deficientes visuais. O PLUMB define uma interface de áudio, na qual a interação do usuário ocorre por meio de uma caneta e um computador

do tipo *tablet*. Durante o uso, o usuário recebe indicações sonoras distintas quando seu ponteiro se encontra sobre um nó ou uma aresta do grafo (Cohen et al., 2006).

No caso de (Freire et al., 2010), foi realizado um estudo de caso para avaliar um protótipo de uma lousa digital construído para permitir o acesso por deficientes visuais. A idéia explorada foi a mediação, ou seja, desenvolver um ambiente no qual mediadores podem inserir informações e os usuários com deficiência visual possam acessá-las. Neste caso específico, o conteúdo se refere a descrições textuais para desenhos feitos sobre a lousa. O estudo de caso foi realizado com um estudante em uma aula experimental de geografia sobre a atmosfera e utilizou, de maneira informal a técnica *think aloud*, contou com a participação de um professor e um mediador. Posteriormente, foi conduzida uma entrevista para conhecer melhor o participante e identificar suas impressões e problemas em relação ao protótipo.

Em (Shinohara e Tenenber, 2007), os autores realizaram um estudo de caso com uma estudante de ensino superior com cegueira congênita. Neste estudo a participante, Sara, foi incumbida de realizar uma série de doze tarefas nas quais ela deveria interagir com algum equipamento ou objeto a fim de estudar sua interação com artefatos tecnológicos. Os pesquisadores optaram por um *design* de pesquisa que utilizou apenas um sujeito e uma análise em profundidade dos dados coletados através de observações. As principais técnicas utilizadas para coleta de dados foram *Technology tours*, *Personal History* e *Guided Speculation*.

Em (Kaczmirek e Wolff, 2007), os autores mostram um conjunto de doze recomendações para a elaboração de *surveys* cujo público alvo fossem deficientes visuais. Para tal, o trabalho leva em consideração o conhecimento dos processos cognitivos e padrões de acessibilidade. Embora este trabalho não se utilize de questionários que serão respondidos de maneira autônoma pelos participantes, as recomendações se mostraram muito importantes durante a preparação do material de apoio para a condução de entrevistas e observações de estudo, que serão melhor detalhados no Capítulo 3.

Os pesquisadores (Andronico et al., 2006) exploraram a acessibilidade de motores de busca com o objetivo de mostrar que, com pequenas modificações em seu código fonte, é possível tornar sua interação mais fácil, mais eficiente e menos frustrante para indivíduos com deficiência visual. O trabalho avaliou os principais problemas de acessibilidade e usabilidade dos motores de busca, propôs um conjunto de oito recomendações para o tratamento dos problemas identificados, tais como uso de alertas sonoros e adicionar *links* de navegação e ajuda. Por fim, foram aplicadas as recomendações ao buscador do Google e foram analisados os desafios para a realização destas alterações, porém nenhum estudo foi conduzido com usuários reais para avaliar essa versão modificada do buscador.

Finalmente, observa-se que os trabalhos citados se aproximam das características deste trabalho por ângulos distintos. Alguns são mais relacionados à parte técnica e ao desenvolvimento da AWMo, outros tratam da condução de estudos de caso com deficientes, mas todos tratam de alguma forma do assunto central desta proposta: acessibilidade de software para usuários com deficiência visual.





---

# Protocolo do estudo de caso

---

Estudos de caso são estudos empíricos que buscam investigar fenômenos contemporâneos em seu contexto (Yin, 1994). O protocolo de um estudo de caso é utilizado como documentação para os procedimentos planejados e decisões tomadas ao longo do planejamento, execução e posterior análise dos dados do estudo de caso (Runeson e Höst, 2008).

Para entender a relação dos usuários com a técnica de interação proposta para modelos de *software*, foram conduzidas entrevistas presenciais e observação da utilização da AWMo por sujeitos pré-selecionados, enquanto realizavam determinadas tarefas definidas.

## 3.1 Questões de pesquisa

As questões de pesquisa são as questões que se busca responder por meio da análise do estudo de caso. Para este trabalho temos as seguintes questões de pesquisa:

1. A ferramenta AWMo permite o acesso, por deficientes visuais, aos diagramas de classe da UML?
2. A linguagem textual da AWMo impõe alguma barreira para a utilização da abordagem proposta?
3. Qual o maior desafio enfrentado na questão do acesso e desenvolvimento de modelos visuais por usuários com deficiência visual?

## 3.2 Estratégia de seleção

O problema que a AWMo busca abordar é o acesso a modelos de *software* para deficientes visuais. Portanto, para a condução deste estudo foi necessária a participação de indivíduos que possuíssem conhecimento em desenvolvimento de *software* e que também fossem deficientes visuais. Portanto, o conjunto de usuários que se buscou era bem delimitado.

Foram selecionados os sujeitos que atenderam aos seguinte critérios:

- Pessoa com deficiência visual;
- Indivíduo que trabalhasse com desenvolvimento de *software* ou fosse estudante e estivesse cursando algum curso relacionado com as áreas de programação e desenvolvimento de *software*.

## 3.3 Coleta de dados

A coleta de dados ocorreu de duas maneiras distintas. De acordo com a definição de Lethbridge et al. (2005), podemos separar as técnicas de coleta de dados em três tipos: de primeiro grau, na qual o pesquisador está em contato direto com os sujeitos e os dados são coletados em tempo real. Segundo grau, em que o pesquisador indiretamente coleta dados sem interagir diretamente com os sujeitos. E terceiro grau, quando o pesquisador se utiliza da análise independente de artefatos ou documentos já existentes.

Neste estudo, foram utilizadas entrevistas presenciais como técnica de coleta de dados de primeiro grau, e observação do sujeito ao utilizar a AWMo para realizar um conjunto de tarefas pré-estabelecidas como uma técnica de coleta de dados de segundo grau.

### 3.3.1 Entrevistas

Foram realizadas duas sessões de entrevistas. Uma que foi realizada antes da observação do uso da AWMo, para coletar informações sobre o participante e sua experiência com desenvolvimento e modelagem de *software*, e outra sessão de entrevista, que foi realizada posteriormente ao uso da ferramenta, para obter informações sobre a experiência do usuário durante a utilização. Essas sessões são chamadas daqui em diante de entrevistas pré-uso e pós-uso, respectivamente.

Ambas as entrevistas foram realizadas de maneira semiestruturada, ou seja, foram utilizados roteiros que guiavam as questões e pontos que deveriam ser abordados durante a entrevista, porém, a ordem destas questões não precisava necessariamente ser seguida de maneira linear. Essa abordagem semiestruturada possibilita ao entrevistador incluir ou

remover perguntas de acordo com o rumo da entrevista, tornando-a assim uma entrevista oportunística, o que é bastante útil para se aumentar ou aprofundar o entendimento (Lazar et al., 2010).

As questões que foram utilizadas como guia para a entrevista pré-uso encontram-se no Apêndice B e as questões que guiaram a entrevista pós-uso encontram-se no Apêndice D.

As entrevistas planejadas buscam obter respostas mais amplas e nas quais os entrevistados tenham a liberdade de colaborar com mais informações; portanto, buscou-se trabalhar com questões abertas em detrimento de questões fechadas ou com conjuntos de respostas pré-definidos, que podem inibir os entrevistados de elaborarem suas respostas. Esta escolha torna o processo de análise dos dados mais trabalhoso, pois ao invés de simplesmente contar ou tabular os dados, é necessária uma análise interpretativa das respostas (Lazar et al., 2010).

Durante as entrevistas, o áudio foi gravado para consultas posteriores, pois ao se realizar anotações em tempo real é necessário muito tempo e concentração, e em razão da quantidade de informação ser muito grande, acaba-se perdendo detalhes que podem ser interessantes. A gravação do áudio das entrevistas foi realizada com o conhecimento e consentimento do participante.

### 3.3.2 Observação

Para a coleta de dados em segundo grau, ou seja, com pouca interferência do pesquisador, foi utilizado um computador instrumentado para a gravação da interação do usuário com a AWMo. Nesta parte do estudo, o usuário recebeu uma série de tarefas pré-estabelecidas para que as executasse utilizando a interface da AWMo.

Para gravação das interações do usuário participante foi utilizada a ferramenta Morae. A Morae é uma ferramenta para auxiliar na execução de testes de usabilidade desenvolvida pela TechSmith<sup>1</sup>

Com o uso da Morae foram gravadas as seguintes informações durante a execução das tarefas:

- Tela do computador e os programas que estavam sendo executados durante o procedimento;
- Interação do usuário com o *mouse*;
- Webcam mostrando o rosto do usuário ao longo do processo;

---

<sup>1</sup><http://www.techsmith.com/morae.html>

- Audio do computador para capturar o que o usuário estava falando e possíveis perguntas que pudessem vir a ser realizadas para o pesquisador em caso de dúvidas, por exemplo;

A *Morae* é composta por 3 componentes: (1) o **Recorder** que é o *software* responsável pela captura do experimento, (2) o **Observer** que possibilita que pesquisadores acompanhem o experimento remotamente em tempo real e (3) o **Manager**, que é utilizado para visualizar e analisar os dados após a coleta.

Embora seja possível conduzir estudos utilizando apenas o **Recorder** e o **Manager**, escolheu-se utilizar o **Observer**, mesmo tendo em vista que o estudo foi conduzido por apenas um pesquisador durante as sessões, pois o uso do **Observer** possibilita a adição de marcadores e comentários à captura, enquanto o estudo está sendo executado. A configuração utilizada para execução contou com o uso de dois computadores em uma mesa, um para o sujeito, executando o **Recorder** e outro para o pesquisador, executando o **Observer** e conectado ao primeiro computador.

A Figura 3.1 mostra a configuração utilizada durante a execução do estudo piloto e que também foi utilizada na execução do estudo final.

Com esta configuração, o pesquisador está sempre à disposição para auxiliar em quaisquer dúvidas ou questões que o sujeito possa vir a ter, e ainda é capaz de fazer anotações e adicionar marcadores diretamente no experimento que está sendo capturado sem interromper o sujeito.

As tarefas que foram realizadas pelos participantes podem ser encontradas no Apêndice C.

Do mesmo modo que nas entrevistas, os dados foram gravados com a autorização e consentimento dos participantes.

### 3.3.3 Medições e métricas

Com o objetivo de coletar dados qualitativos, foram extraídas algumas métricas da fase de observação do experimento, no qual o participante realizou tarefas utilizando a *AWMo*.

Para que o fluxo do trabalho não fosse interrompido, as medições foram realizadas em um momento posterior ao da coleta, utilizando os registros em vídeo gravados com a ferramenta *Morae*. As medições extraídas das gravações foram:

- **Falhas por tarefa:** medida a partir dos vídeos, o número de ações incorretas realizadas pelo usuário, ao longo de cada tarefa objetivo; ações essas que podem vir a resultar em um erro se não corrigida.



**Figura 3.1:** Foto da configuração do ambiente do experimento, conforme foi conduzido durante a execução piloto. O computador da direita, utilizado pelo sujeito, executa o Morae *Recorder* e o computador à esquerda, utilizado pelo pesquisador, executa o Morae *Observer*.

- **Erros por tarefa:** medida a partir dos vídeos, o número de vezes em que a AWMo exibiu mensagens de erro para o usuário, no decorrer da realização de cada tarefa.
- **Tempo por tarefa:** medido o tempo que o usuário utilizou para realizar cada uma das tarefas.

### 3.4 Triangulação

Triangulação é o processo de se abordar o objeto de estudo por diferentes ângulos. Essa técnica é muito importante quando se está realizando um estudo qualitativo, pois o material coletado em geral é muito rico e extenso, porém menos preciso do que dados quantitativos (Runeson e Höst, 2008).

De acordo com Stake (1995), existem quatro tipos de triangulação que podem ser realizadas em um estudo de caso: triangulação de dados, triangulação de observadores, triangulação metodológica e triangulação teórica.

Neste trabalho foram utilizados três tipos de triangulação:

- **Triangulação metodológica:** com o objetivo de validação e aumentar a confiabilidade dos dados coletados, foram combinadas uma estratégia qualitativa (entrevista) com uma estratégia quantitativa (métricas na observação). Embora essa abordagem quantitativa não seja suficiente para que conclusões sejam estatisticamente válidas, elas colaboram para a validade dos dados coletados, seja sustentando os dados das entrevistas ou expondo possíveis divergências.
- **Triangulação de dados:** do mesmo modo que na triangulação metodológica, a triangulação de dados se deu pela coleta de dados de duas fontes diferentes, nas entrevistas, de maneira direta e por meio de gravações da Morae, de maneira indireta. O uso de fontes de dados distintas provê oportunidade para se obter *insights* que poderiam ser mais difíceis de se obter quando se utiliza apenas uma forma de coleta de informações (Lazar et al., 2010).
- **Triangulação de observador:** durante a fase de análise dos dados, todo o material coletado durante as entrevistas e observações de uso da ferramenta foi analisado pelo autor com o auxílio de um segundo pesquisador. Dessa forma, buscou-se evitar que as análises fossem influenciadas pelas opiniões pessoais dos pesquisadores e refletissem de maneira mais fiel os fatos.

Com o uso das triangulações citadas e analisando os dados das três etapas (entrevista pré-uso, uso da ferramenta e entrevista pós-uso) de maneira cruzada, esperou-se mitigar o risco de que a informação fosse mal interpretada ou mal compreendida.

### 3.5 Ameaças à validade

Para assegurar a melhor qualidade dos resultados que se pretende obter com este estudo, as principais ameaças a sua validade foram tratadas e monitoradas ao longo de sua execução. Essas ameaças e as ações que poderiam ser realizadas para mitigá-las ou possíveis soluções alternativas são listadas nas Tabelas 3.1 e 3.2.

**Tabela 3.1:** Ameaças à validade interna

Ameaça	Mitigação (M) / Solução alternativa (A)
<p>As questões das entrevistas não serão interpretadas da mesma maneira pelo entrevistador e pelos entrevistados.</p>	<ul style="list-style-type: none"> <li>• (M) A entrevista será gravada para que nenhuma informação seja perdida ou mal interpretada.</li> <li>• (M) A entrevista será transcrita e os entrevistados receberão uma cópia para que possa fazer considerações, se necessário.</li> <li>• (M) O piloto que será realizado também ajudará a identificar questões que possam vir a ser mal interpretadas.</li> </ul>
<p>A amostra de informações coletadas na entrevista não é confiável.</p>	<ul style="list-style-type: none"> <li>• (M) As entrevistas serão realizadas presencialmente, diminuindo o risco de enganos.</li> <li>• (M) Os dados obtidos através das entrevistas serão confrontados, na medida do possível, com os dados coletados durante a observação, aumentando a confiabilidade de ambas as fontes.</li> <li>• (A) Caso seja constatado que mesmo com as ações acima os dados não sejam confiáveis, os mesmos serão descartados e não serão considerados para o estudo.</li> </ul>
<p>As questões não refletem a realidade dos entrevistados e seu contexto de uso da ferramenta.</p>	<ul style="list-style-type: none"> <li>• (M) Será executado um estudo piloto como citado na Seção 3.6 com o objetivo de mitigar este risco.</li> <li>• (M) As entrevistas serão realizadas de maneira semiestruturada. Portanto, será possível ao entrevistador ignorar questões que não sejam cabíveis ou mesmo acrescentar outras que sejam relevantes no momento da entrevista, documentando devidamente essas decisões posteriormente.</li> </ul>

Tabela 3.2: Ameaças à validade externa

Ameaça	Mitigação (M) / Solução alternativa (A)
Os dados obtidos não são representativos.	<ul style="list-style-type: none"> <li>• (M) Para tentar mitigar este risco, pretende-se realizar este estudo de caso com ao menos dois sujeitos. Idealmente com ainda mais pessoas, no entanto, como foi mencionado na Seção 3.2 o grupo de usuários-alvo para este estudo é bastante restrito.</li> </ul>
As conclusões não são generalizáveis.	<ul style="list-style-type: none"> <li>• (A) Esta é uma característica inerente aos estudos de caso. Ao invés de esperar que as conclusões sejam generalizáveis, espera-se que elas apresentem bons indícios sobre a capacidade da AWMo em auxiliar desenvolvedores de <i>software</i> deficientes visuais a realizarem atividades de modelagem de <i>software</i> e que os resultados possam ser utilizados como base para novas pesquisas e melhorias na própria ferramenta.</li> </ul>

### 3.6 Piloto

Para avaliar se as questões de entrevista e as tarefas que foram requisitadas eram adequadas para se coletar os dados pretendidos e se seriam suficientes para se atingir os objetivos estabelecidos, um estudo piloto foi conduzido.

O estudo piloto contou com a participação de Marvin. Marvin é um nome fictício que foi atribuído pelo autor, utilizado com o objetivo de preservar a identidade do participante.

Marvin é natural do interior de São Paulo, possui 41 anos de idade e apenas 10% de visão em ambos os olhos em função de cicatrizes em suas retinas. É formado em Matemática e atualmente está no terceiro ano de um curso de Bacharelado em Sistemas de Informação.

Com o objetivo de facilitar o uso de computadores, ele utiliza as fontes levemente ampliadas, o que torna a leitura mais fácil e ao mesmo tempo não atrapalha o uso em telas pequenas, como é o caso de seu *netbook*. Além disso, é comum que aproxime bastante o rosto da tela do computador. Também utiliza os ícones em tamanho grande e mantém poucos itens na área de trabalho. Por possuir visão parcial, não faz uso de leitores de tela.

Marvin utiliza computadores desde 1984, e iniciou com computadores como o CP 500. Em relação à programação de computadores, teve seu primeiro contato aos 18 anos durante um curso técnico de processamento de dados, com linguagens de programação como Basic e COBOL, porém deixou de praticar programação durante o período em



que cursou graduação em Matemática, período que utilizou apenas um pouco de linguagem C. A cerca de três anos começou a estudar linguagens de programação para *Web* por conta própria.

Mais recentemente, no curso de Sistemas de Informação, Marvin teve contato com Java, PHP e SQL.

Em relação a modelos e diagramas da UML, teve contato ao longo do curso de Sistemas de Informação em disciplinas como banco de dados (modelos entidade-relacionamento) e programação orientada a objetos, nas quais o conteúdo em Java era ilustrado com diagramas de classe da UML e também chegou a construir alguns diagramas utilizando a ferramenta BlueJ.

Ao longo da execução do estudo piloto foram coletados 18 minutos de áudio durante a pré-entrevista, 67 minutos de vídeo durante a fase de observação enquanto Marvin utilizava a AWMo e, posteriormente, foram capturados 16 minutos de áudio durante a pós-entrevista.

Com base na execução do estudo piloto, foi identificado que a pré-entrevista foi capaz de coletar dados sobre o sujeito e suas experiências passadas com modelagem de maneira satisfatória. A estratégia de conduzir a entrevista de maneira semiestruturada se mostrou bastante útil à medida que questões que foram adicionadas pelo pesquisador ao longo da entrevista ajudaram a melhor conhecer o sujeito. Outro aspecto importante foi o fato de que algumas das questões planejadas não refletiam a realidade de Marvin, por exemplo, a questão: *“Na sua opinião, qual o principal problema de acesso a esses diagramas?”* para a qual o sujeito respondeu:

*“...eu acho que o fato deles serem diagramas facilitou.”* e *“A não ser que você tenha que fazer um diagrama muito complexo e talvez tenha que diminuir o diagrama, não sei como funciona isso aí. Eu fiz diagramas simples, bem simplesinhos.”*

Isto ocorreu devido ao fato de Marvin possuir 10% de visão, então diagramas visuais não são, na realidade, uma barreira para ele. Podem exigir mais esforço para leitura e, especialmente se o diagrama for complexo, a imagem resultante pode ser muito grande. Isso mostrou que esta questão foi criada com foco em sujeitos que são completamente cegos, que são os usuários que supostamente terão maior benefício com o uso da AWMo. Sob esta perspectiva, o curso da entrevista foi alterado durante sua execução para melhor se adaptar à realidade de Marvin.

Na primeira versão do protocolo foi definido um conjunto de 4 tarefas objetivos para o usuário executar ao utilizar a AWMo, durante a fase de observação do estudo. No entanto, durante o estudo piloto foi possível perceber que as descrições dos objetivos causaram alguma confusão para o sujeito. Parte disso, em função de alguns nomes que foram utilizados de maneira inconsistente entre as descrições e o diagrama já existente na AWMo, como

foi o caso da denominação “Conta corrente”. O objetivo 3 foi o que apresentou os maiores problemas, pois foi solicitado que o usuário realizasse duas ações inteiramente distintas no mesmo diagrama e Marvin concluiu apenas uma delas, precisando ser lembrado da segunda parte do objetivo pelo pesquisador.

Para melhorar a formulação das tarefas objetivos, alguns dos nomes foram padronizados e o objetivo 3 foi dividido em dois outros objetivos distintos, resultando em um novo conjunto de 5 objetivos que englobam as mesmas ações do primeiro conjunto, porém de maneira mais simples de serem compreendidas.

A pós-entrevista foi conduzida de maneira similar à pré-entrevista e permitiu eliciar novos requisitos, tais como: adicionar um *link* nas mensagens de erro para que o foco do editor de texto vá diretamente para a linha com erro em questão e a exibição de número de linha no editor de código para auxiliar a navegação do usuário. Novamente, a entrevista semiestruturada com questões abertas se mostrou útil para se obter um entendimento mais profundo nos tópicos que se mostraram interessantes no decorrer da entrevista.

Após a condução do piloto, algumas das funcionalidades apontadas por Marvin foram desenvolvidas na AWMo, antes da execução do estudo com outros sujeitos, tais como: habilidade de controlar o tamanho da fonte dentro da área de texto do código utilizando a funcionalidade de aumentar fonte da barra de acessibilidade da AWMo. Outro caso especial foi a apresentação do número de linha (do código textual) e a habilidade de ir diretamente para as linhas com erro, que foram desenvolvidas depois do piloto utilizando um *plugin* para editores de código na Web chamado *code mirror*<sup>2</sup>. Porém, em um momento posterior, ao validar a acessibilidade deste editor com o leitor de tela NVDA, ele se mostrou inacessível para leitores de tela, pois substitui completamente a área de textos por outros elementos HTML que respondem a eventos JavaScript. Por isso, essas últimas funcionalidades tiveram que ser removidas para permitir a continuidade do estudo e necessitam de uma reavaliação mais detalhada, e serão alvo de investigações futuras.

A execução do estudo piloto foi muito importante, não apenas para validar a abordagem com a qual o estudo foi conduzido, mas também por permitir que os pesquisadores coletassem informações valiosas sobre a própria AWMo e identificar alguns problemas básicos, tanto na ferramenta quanto no conjunto de objetivos planejados para a fase de observação.

As transcrições das entrevistas realizadas com Marvin podem ser encontradas no Apêndice F.

---

<sup>2</sup><http://codemirror.net>

### **3.7 Plano de Análise**

Para cada um dos sujeitos analisados neste estudo de caso, os arquivos de áudio das entrevistas foram transcritos e o conteúdo das gravações em vídeo (tela e *webcam*) foram analisados em busca das medições estabelecidas na Seção 3.3.2 e de outras informações relevantes.

Após as transcrições das entrevistas terem sido revistas pelos participantes, os dados foram analisados com o objetivo de responder as questões de pesquisa estabelecidas na Seção 3.1. Nenhum dos nomes ou informações confidenciais dos participantes serão publicadas em hipótese alguma. O termo de confidencialidade que foi firmado com os participantes pode ser encontrado no Apêndice E.



---

## Resultados

---

Após a execução do estudo piloto e de terem sido melhorados os pontos observados na Seção 3.6, o estudo de caso foi conduzido com dois outros participantes: Arthur e Ford (novamente os nomes foram alterados com o objetivo de manter a confidencialidade dos participantes).

As Seções 4.1 e 4.2 apresentam os resultados das execuções dos estudos com Arthur e Ford, respectivamente, e a Seção 4.3 apresenta uma análise comparativa dos resultados dos dois participantes.

### 4.1 Arthur

A partir da pré-entrevista, foi possível conhecer um pouco mais sobre o Arthur: ele é um homem de 31 anos e está atualmente matriculado no terceiro ano de um curso de ensino a distância de Bacharelado em Sistemas de Informação, oferecido com apoio de uma universidade federal. Arthur é completamente cego em função de uma doença craniana e perdeu a visão quando tinha 15 anos de idade. Ele utiliza computadores há 7 anos e começou a aprender programação de computadores há 5 anos atrás, sendo familiarizado com as linguagens Java, C# e JavaScript e outras tecnologias para desenvolvimento Web tais como HTML e CSS.

Para auxiliar o uso de computadores, ele utiliza o leitor de tela NVDA que lê todas as informações exibidas na tela e fornece atalhos de teclado para auxiliar a navegação e interação com o conteúdo.

Arthur já teve contato com diagramas de classe UML no passado, por meio de descrições textuais e de uma abordagem que utiliza planilhas do Excel para sistematizar os diagramas. Quando questionado sobre como foi essa experiência, ele respondeu:

*“Através dessas descrições, se o diagrama for complexo, fica complicado você ter uma idéia clara. Eu consigo ter uma noção mas não considero que seja suficiente para eu ter uma idéia clara de como é o diagrama. A questão de você descrever uma imagem, ela ajuda em alguns aspectos, mas em diagramas mais complexos fica difícil, fica muito confuso você ter uma noção clara do que está ali.”*

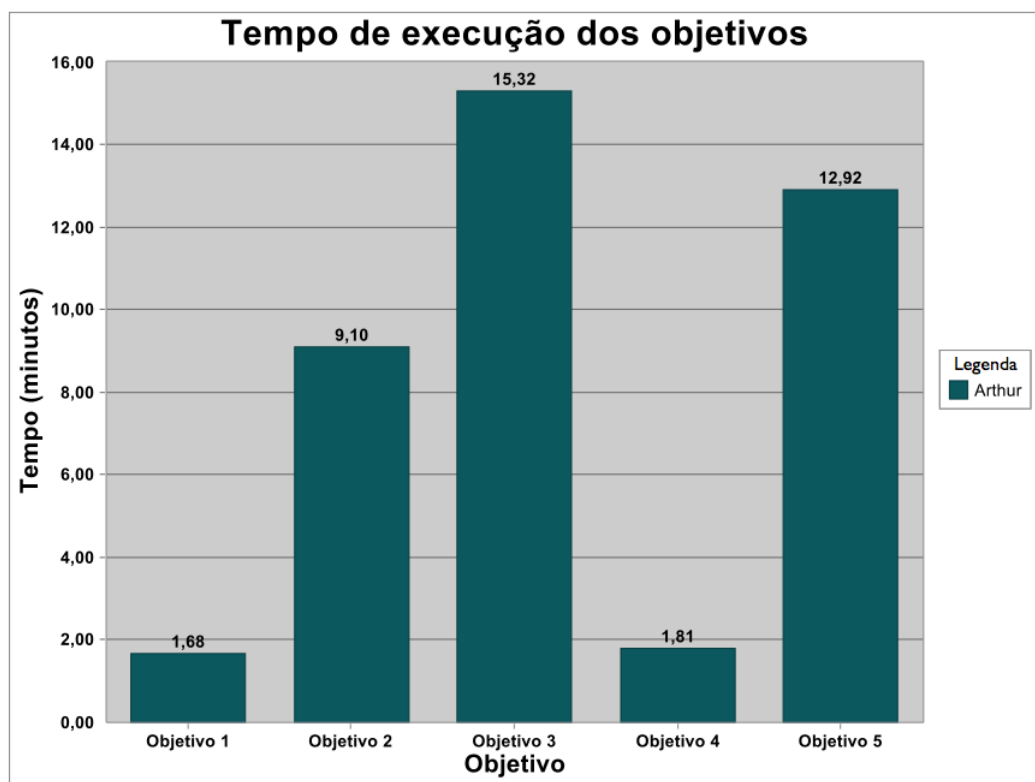
Quando questionado sobre o que ele acredita que sejam os principais problemas nesta forma de se acessar os diagramas:

*“Primeiro, a falta de praticidade. Porque se você está em uma situação real, onde você está dentro de um projeto e você precisa trabalhar com pessoas que enxergam, por exemplo, você precisa de alguém que descreva para você e isso já uma perda de tempo em determinados aspectos. Fora que você não consegue expressar o diagrama de uma forma que uma outra pessoa que enxerga... Você não consegue se comunicar com a pessoa através do diagrama, por exemplo, eu posso criar uma descrição de como eu entendo o diagrama, só que para uma pessoa que está acostumada a observar a representação gráfica, ler aquilo não vai ficar muito claro pra ela entender. Então esse é um aspecto, e a questão da leitura também, basicamente eu preciso de alguém que descreva para mim. Eu não consigo pegar o diagrama feito por uma outra pessoa e entender aquilo de uma maneira independente”.*

Durante a fase de observação foram coletados 51:44 minutos de vídeo enquanto Arthur completava o conjunto de objetivos pré-estabelecidos. O tempo utilizado em cada um dos objetivos pode ser visto na Figura 4.1.

O tempo utilizado para a leitura das instruções dos objetivos, antes de começar a trabalhar para completá-los não foi contabilizado nos tempos dos objetivos. Todas as consultas posteriores às instruções realizadas após o início da resolução foram contabilizadas na soma do tempo. Além disso, no caso do objetivo 1, Arthur interrompeu a execução para explicar ao pesquisador o funcionamento do leitor de tela e como um problema de acessibilidade encontrado poderia ser resolvido. Durante essa explicação, a contagem do tempo estava em pausa na Morae e, portanto, não foi contabilizado no total de tempo utilizado para completar o objetivo 1, por se tratarem de informações adicionais.

O objetivo 1 consistia em permitir que o usuário utilizasse as funcionalidades de acessibilidade da AWMo. No entanto, pelo fato de Arthur ser completamente cego, a única assistência que ele utilizava era o leitor de tela, que estava previamente instalado no computador e, portanto, ele utilizou o tempo deste objetivo para conhecer a ferramenta AWMo, com qual ele estava tendo o primeiro contato. Ele navegou por toda a estrutura da aplicação aprendendo a localização de menus, botões e outros recursos. No decorrer



**Figura 4.1:** Gráfico mostrando a quantidade de tempo utilizado por Arthur para completar cada um dos objetivos propostos neste estudo de caso.

dos objetivos seguintes, foi possível perceber que o sujeito começou a adotar diferentes estratégias de navegação para alcançar mais rapidamente o local desejado dentro da AWMo.

O objetivo 2 consistia no tempo utilizado para entender o diagrama “Sistema Bancário” e explicar seu conteúdo para o pesquisador. Aqui Arthur mostrou entender o conteúdo consideravelmente rápido, considerando-se que ele possuía conhecimento prévio das estruturas da linguagem textual em si, porém estava tendo contato com este diagrama específico pela primeira vez. Ele levou um total de 9,1 minutos para entender o diagrama e explicá-lo ao pesquisador.

Como esperado, os objetivos 3 e 5 foram os que levaram mais tempo para serem completados pois eles são mais complexos e, portanto, a probabilidade de o sujeito cometer algum engano é maior. No entanto, era esperado que o objetivo 4 tomasse mais tempo e fosse difícil para ser resolvido, pois ao remover a conta poupança Arthur teria de buscar todas as referências a essa classe. Porém, haviam apenas duas referências no diagrama proposto que consistiam em dois métodos na classe Pessoa: `AbrirContaPoupanca()` e `FecharContaPoupanca()`, sendo que ambos tinham valores de retorno *booleanos*. Estes dois métodos são relacionados à classe `ContaPoupanca`, porém esse fato não foi mapeado no diagrama por meio de tipo de retorno ou parâmetros de entrada. Neste caso, se os dois

métodos tivessem valores de retorno ou parâmetros do tipo ContaPoupanca, o interpretador da AWMo teria mostrado erros quando a classe fosse removida, pois os métodos estariam referenciando uma classe que não poderia ser encontrada.

Uma observação importante, em relação às tarefas 3 e 5, é o fato de que Arthur encontrou bastante dificuldade com o teclado do computador que ele utilizou durante o estudo, e pode-se notar claramente por meio da gravação de vídeo que grande parte das falhas e erros poderiam ter sido evitadas se o sujeito estivesse utilizando um computador ou teclado com o qual ele já estivesse familiarizado. Infelizmente, devido a instrumentação necessária para realizar as gravações, Arthur teve que utilizar um computador fornecido pelo pesquisador e, mesmo com tempo fornecido para que ele pudesse configurar o computador e o leitor de tela de acordo com suas preferências, o teclado não pode ser alterado por se tratar de um notebook.

A Figura 4.2 mostra o número de falhas inseridas por Arthur ao longo de cada um dos objetivos. Uma vez que os objetivos 3 e 5 são os que envolvem mais edição, foram ao longo do desenvolvimento destas tarefas os que apresentaram o maior número de falhas. Comparando-se o número de falhas inseridas, com o número de vezes que o interpretador mostrou mensagens de erro ao usuário exibidos na Figura 4.3, podemos observar que, durante o processo de correção dos erros apontados, em geral Arthur cometeu mais de uma falha por vez, isto explica o motivo pelo qual o número de vezes que o interpretador retornou mensagens de erros ao usuário é inferior ao número de falhas inseridas. Além disso, houveram situações em que Arthur foi capaz de detectar uma falha que ela gerasse uma mensagem de erro, ou seja, antes de salvar o diagrama.

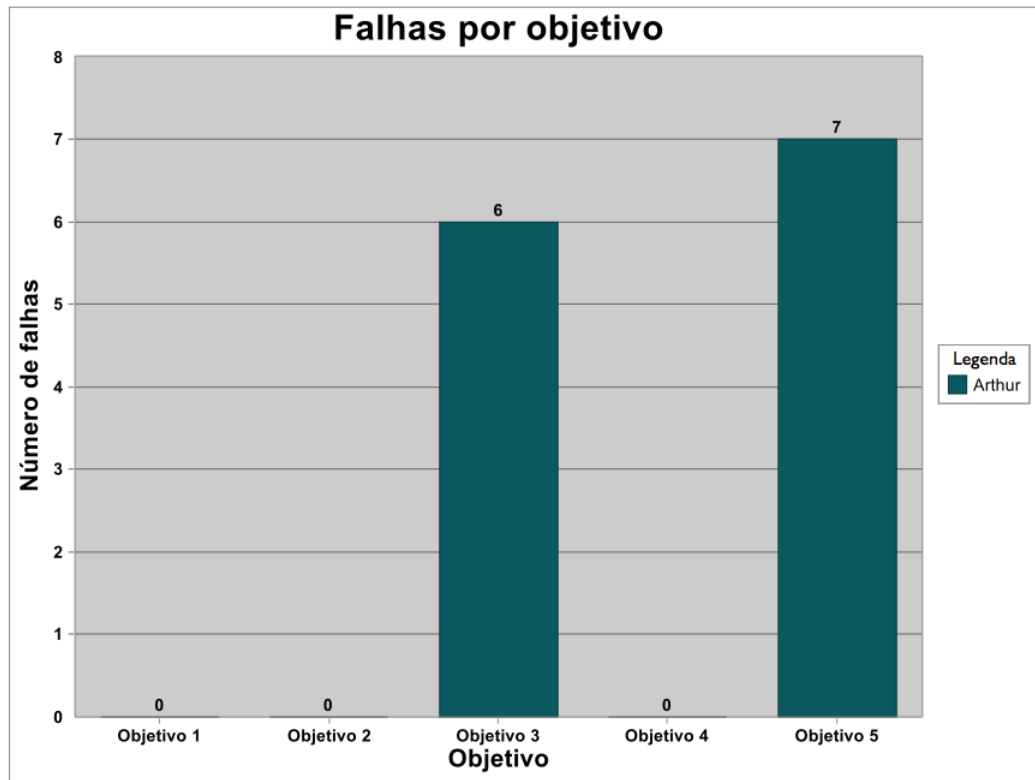
A Figura 4.4 mostra uma captura de tela retirada da sessão gravada com a ferramenta Morae. A figura exhibe uma das situações em que o interpretador da AWMo retornou erros ao sujeito em resposta a ativação do botão "Salvar diagrama".

Após a fase de observação, a pós-entrevista foi conduzida com o objetivo de entender melhor a experiência de uso, as dificuldades que Arthur encontrou e o que contribuiu para que os objetivos fossem completados.

Ao longo da fase de observação do estudo, Arthur identificou um problema de acessibilidade nos botões de controle de tamanho de fonte da AWMo, exceto isso, não foram encontradas maiores dificuldades ao utilizar a ferramenta:

*“Não achei nada que fosse difícil de trabalhar, achei a interface simples, acessível, intuitiva. Só sinalizei um problema, que não chega a impedir a utilização que são os botões do menu do topo, não estão rotulados, a função deles está indicada só pelo atributo title e quando você está navegando no modo de leitura da página, o leitor de tela não detecta a funcionalidade do botão, você precisa navegar pelo foco. Mas, tirando isso que também não é nenhum problema que impeça a utilização, o resto eu achei totalmente*





**Figura 4.2:** Gráfico extraído da gravação de vídeo da fase de observação contendo o número de falhas inseridas por Arthur ao decorrer da execução de cada objetivo.

*acessível e tranquilo de utilizar, mesmo o feedback com relação aos erros de sintaxe da linguagem, são simples de você encontrar na página.”*

Além disso, quando questionado se mudaria algo na AWMo:

*“Olha, honestamente, observando a ferramenta, a dificuldade que você tem é a normal que você tem em qualquer página que é o fato de você não conhecer a página. Você precisa primeiro saber onde estão as coisas, estudar primeiro a página para você ter uma representação, ter claro na sua mente como as coisas estão dispostas para você encontrá-las mas rapidamente mas é o contato inicial. Após isso você já consegue se localizar e fica tranquilo de utilizar. Não vejo nada que devesse melhorar, mais aqueles pontos dos botões e link de abrir que são mudanças básicas e simples de fazer, mas de forma geral eu fiquei satisfeito com a interface.”*

Com o objetivo de melhor investigar os sentimentos de Arthur a respeito da ferramenta, foi questionado se ele acredita que seria prático utilizar a AWMo diariamente, ao trabalhar com desenvolvimento de *software*:

*“Eu acho que sim, por exemplo, se eu me encontrasse numa equipe de desenvolvimento e fosse trabalhar maciçamente com UML e os demais desenvolvedores se interessassem em*

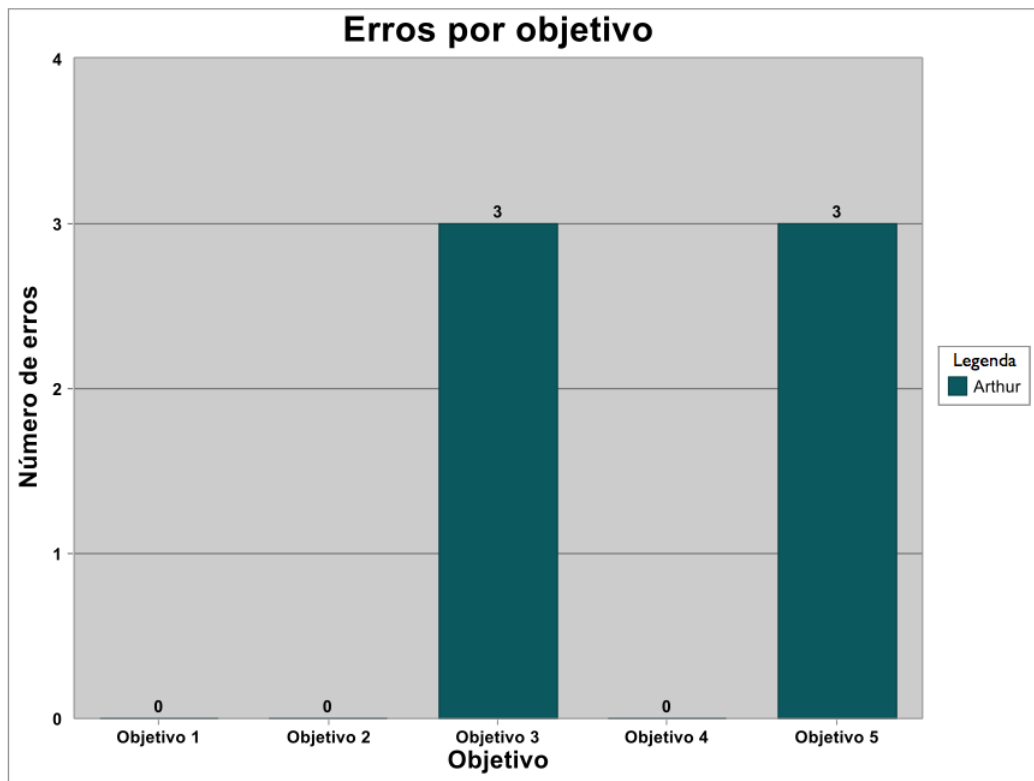


Figura 4.3: Gráfico extraído da gravação de vídeo da fase de observação contendo o número de vezes em que mensagens de erro do interpretador foram exibidas para Arthur separados por objetivo.

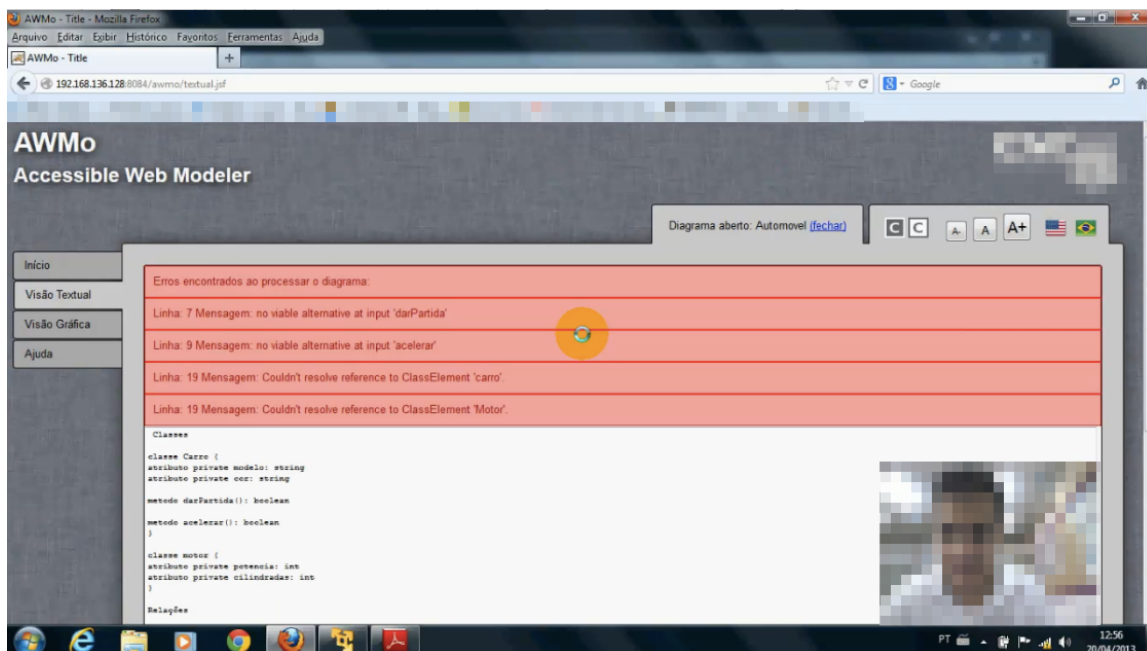


Figura 4.4: Captura de tela de um dos momentos em que o interpretador retornou mensagens de erro para Arthur. Foi aplicado um mosaico sobre o rosto de Arthur para preservar sua identidade.

*aprender a linguagem, que é uma coisa simples e rápida de se fazer, eu acho que seria interessante sim.*”

No entanto, ele se esqueceu que a AWMo também oferece uma visão gráfica que permitiria que usuários com visão colaborassem com ele sem a necessidade de aprender a linguagem textual. Quando foi lembrado sobre isto pelo pesquisador ele disse: “*Exatamente, esse é o ponto.*”

Alguns pontos que foram levantados como melhorias para serem implementadas na AWMo:

- Adicionar o atributo *label* nos botões de controle de fonte;
- Melhorar a descrição das ações de abrir o diagrama na lista de diagramas da página inicial.

Por fim, foi questionado se ele, como um desenvolvedor de *software*, teria interesse em contribuir com o projeto da AWMo, se ela fosse publicada como um projeto de código aberto:

*“Eu me interessaria, achei um projeto bacana de utilizar e de colaborar.”*

## 4.2 Ford

Ford é um homem de 35 anos e atualmente é funcionário em um departamento de investimentos corporativos de uma grande instituição financeira. Ele possui glaucoma congênito e nasceu cego. Aos 20 dias de nascido, passou por uma cirurgia que restaurou 30% de sua visão. Ford viveu com baixa visão durante grande parte de sua vida, porém aos 30 anos de idade passou por uma crise que o deixou praticamente cego novamente. Ele é capaz de reconhecer luz e alguns objetos, mas sem formas definidas e é clinicamente considerado cego.

Ele utiliza computadores desde os 9 anos de idade quando ganhou um curso de informática que ensinava MS Logo, mas em casa ia além do conteúdo passado em aula lendo e aprendendo sozinho e programando algoritmos de geração de números aleatórios e jogos de perguntas e repostas. Posteriormente, ganhou um TK85, depois um PC e atualmente é formado em Análise de Sistemas e utiliza um Mac em casa.

Já utilizou as linguagens de programação Basic, dBase, Clipper, VB3, todas as versões de Delphi, VB6, ASP, PHP e Java. Atualmente trabalha com .NET mas disse estar sempre buscando aprender novas linguagens e tecnologias.

Para auxiliar seu uso de computadores, ele utiliza o leitor de tela NVDA em 99% do tempo e ocasionalmente utiliza o alto contraste no Windows quando o leitor de tela falha em ler alguma janela, embora admita que o uso do alto contraste seja bastante raro.

Ford já teve contato com modelos da UML, entidade-relacionamento, DFDs (*Data flow diagrams*) nos cursos de graduação que frequentou, durante a fase de sua vida em que possuía 30% de visão, portanto aprendeu UML como um vidente e por isso tem memória de como os modelos são representados visualmente e já chegou a utilizar o UML profissionalmente para documentação e comunicação entre as equipes de análise e desenvolvimento. Porém deixou de utilizar os diagramas quando perdeu a visão.

Ao ser questionado se já havia utilizado algum tipo de tecnologia assistiva para UML ele respondeu:

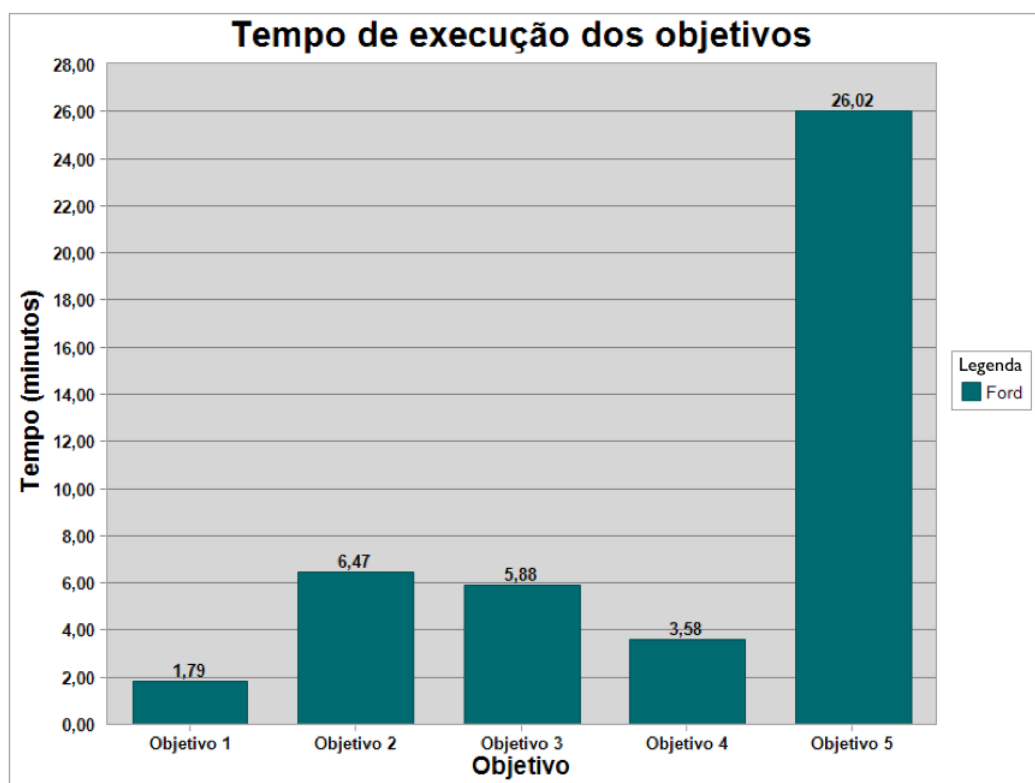
*“Não, tecnologia assistiva não. Assim, sempre tentei imaginar alguma coisa assim, inclusive a sua ideia foi sensacional, porque foi até uma coisa que estava comentando com um colega meu era exatamente isso né, pra deficientes seria muito mais fácil uma linguagem declarativa e que fizesse essa conversão, porque aí os videntes poderiam ler os modelos graficamente como eles leem hoje e a gente teria essa facilidade porque a universidade do Paraná, teve uma aluna lá que fez um projeto sobre como ensinar UML para um deficiente visual a partir do uso de tabelas. Cara, bacana, é super bom para o lado educacional, o deficiente vai entender mas quando ele sair para o mercado, das duas uma: ou ele vai ter que perder um tempão com alguém da equipe dele falando, cara, olha é isso, e o cara vai documentando na tabela e aí se depois tiver uma mudança de projeto, ele vai ter que ter esse trabalho ir lá e atualizar, sendo que com uma linguagem declarativa como é o caso do seu projeto, se você consegue fazer uma tradução bidirecional. Se alguém for lá na ferramenta e alterar o diagrama, automaticamente atualiza a linguagem e alterei a linguagem, já altera o diagrama. Então, quer dizer, tem uma interação muito maior e muito mais rápida né.”*

Quando questionado sobre os principais problemas de acesso de diagramas para deficientes visuais:

*“...as tecnologias assistivas hoje ainda não conseguem chegar num ponto de transformar realmente qualquer elemento da tela em algo interativo. Então se o cara que desenvolveu lá o Visio tivesse pensado que cada um dos componentes que vai no formulário tem que ter foco, tem que ter interação com o teclado, tem que ter ali um label que descreva o que é aquilo para que o NVDA, o JAWS ou qualquer outro leitor consiga interagir né, a gente não consegue.”*

Na fase de observação, foram coletados 54:14 minutos de vídeo durante o tempo em que Ford completava o conjunto de objetivos propostos por este estudo. A distribuição do tempo utilizado em cada um dos 5 objetivos pode ser vista no gráfico 4.5 .

Da mesma forma que ocorreu com a participação de Arthur descrita na seção anterior, Ford não necessitava de nenhuma das funcionalidades de acessibilidade da AWMo por utilizar unicamente o leitor de tela como tecnologia assistiva para a navegação em *sites*



**Figura 4.5:** Gráfico mostrando a quantidade de tempo utilizado por Ford para completar os objetivos propostos neste estudo de caso.

e aplicações web e portanto, utilizou apenas pouco tempo para se familiarizar com a localização dos menus e outros itens da interface.

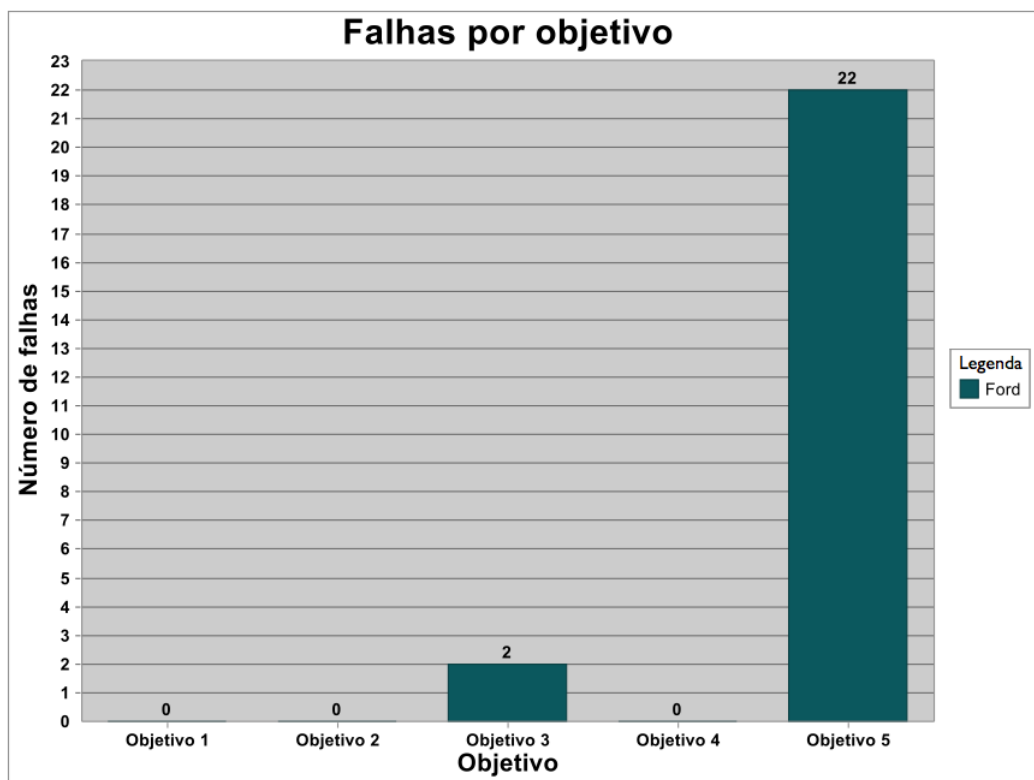
No objetivo 2 Ford foi capaz de compreender o modelo “Sistema Bancário” rapidamente, sendo que levou 6,47 minutos para ler e explicar com suas próprias palavras o funcionamento do sistema que se encontrava modelado naquele diagrama.

No objetivo 3 era requisitado que o sujeito especializasse a classe Pessoa, existente no diagrama para tratar tanto pessoas jurídicas quanto pessoas físicas e sugeria-se utilizar o relacionamento de herança. Ford completou este objetivo muito rapidamente cometendo apenas duas falhas que foram esquecer-se de adicionar os modificadores de acesso aos atributos da classe “PessoaJuridica” que ele havia criado. No entanto, Ford detectou a falha sozinho e as corrigiu antes de ser alertado sobre o problema pelo interpretador da linguagem textual. No total, Ford levou 5,88 minutos para completar esta tarefa.

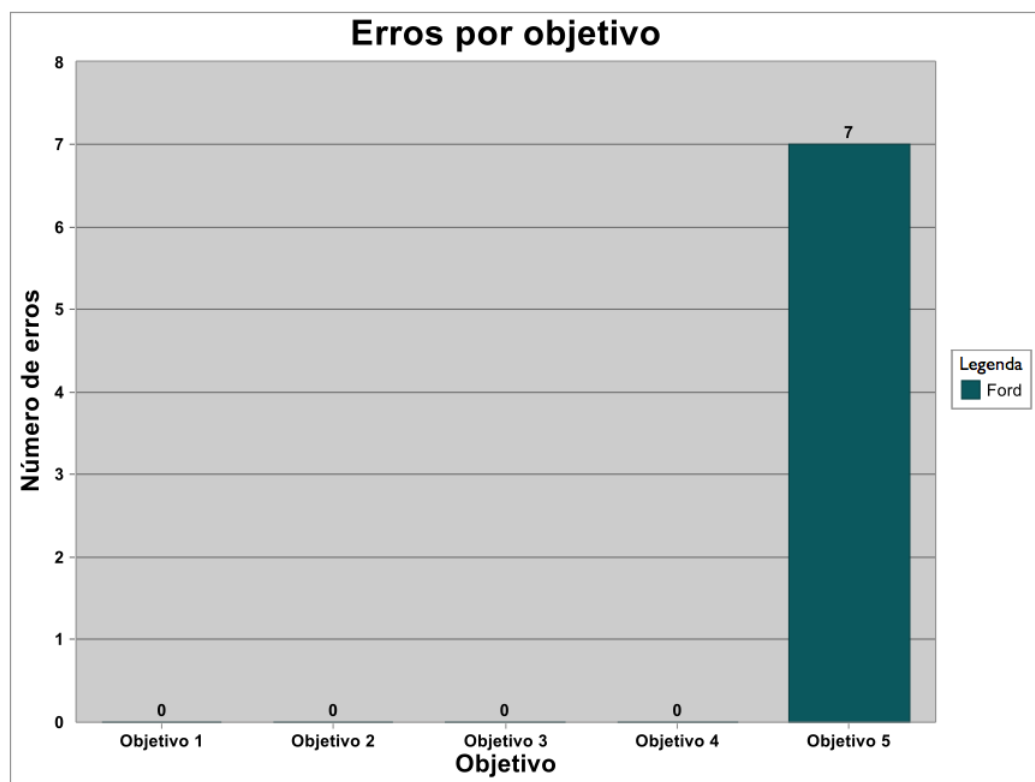
Para completar o objetivo 4, Ford utilizou 3,58 minutos sendo que não cometeu nenhuma falha e conseqüentemente não houveram erros. Mesmo com a questão de os métodos relacionados a poupança não estarem relacionados com a classe “Poupança” como explicado na Seção anterior, Ford encontrou e removeu os métodos “AbrirPoupanca” e “FecharPoupanca” que se encontravam na classe “Pessoa”.

O objetivo 5 foi o que Ford encontrou maiores dificuldades, ele levou um total de 26,02 minutos para completar este objetivo e cometeu 22 falhas ao longo deste período, que resultaram em 7 retornos com mensagens de erros do interpretador. Grande parte das falhas foram erros de sintaxe como falta de modificador de acesso ou grafia incorreta de tipos básicos de dados (81,8%) e as demais foram erros de digitação como troca de letras ou falta de espaço entre duas palavras (18,2%). Além disso, para criar o relacionamento de composição entre as classes “Carro” e “Motor” Ford precisou consultar o manual da linguagem textual para se lembrar como esse relacionamento deveria ser declarado. O manual da linguagem estava disponível a qualquer momento na forma de um documento PDF no *desktop* do computador onde o experimento foi executado e Ford realizou essa consulta de maneira independente, sem qualquer auxílio do pesquisador.

A Figura 4.6 mostra o número de falhas inseridas por Ford, distribuídas pelos objetivos propostos, e na Figura 4.7, pode-se observar que, assim como no caso de Arthur, falhas foram corrigidas em conjunto, ou seja, a cada vez que o usuário se deparava com mensagens de erros do interpretador da AWMo, mais de uma falha foi corrigida, no caso de Ford foram 3,14 falhas corrigidas em média a cada vez que foram exibidas mensagens de erro.



**Figura 4.6:** Gráfico mostrando o número de falhas inseridas por Ford ao decorrer da execução dos objetivos.



**Figura 4.7:** Gráfico mostrando o número de vezes em que mensagens de erro do interpretador de texto foram exibidas para Ford, separados por objetivo.

Após a fase de observação, a pós-entrevista foi conduzida para entender melhor a experiência de uso da AWMo pela perspectiva de Ford. Ao falar sobre as dificuldades de uso da ferramenta, Ford disse:

*“Cara, na verdade, a dificuldade foi lembrar a sintaxe, mas foi a primeira vez que eu usei, então com certeza usando isso no dia a dia se torna um pouco mais fácil. E as mensagens de erro, elas poderiam conter ali a mais, além da linha, que eu acho que facilita pra você chegar na linha e tudo mais. Seria alguma coisa como, por exemplo, a própria linha mesmo como algumas linguagens fazem na compilação, ela colocar ali um caractere do lado da onde ele acha que está o problema né. Ou então, pelo menos a coluna, do tipo: deu problema na linha 11, coluna 5. Porque, quando ele tava lendo rápido, quando ele lia método pra mim, o cérebro tem essa mania de querer adivinhar as coisas, então existe até leitura dinâmica com leitor de tela e quando ele fala método apesar de estar errado eu entendia certo. Foi aí que eu soletrei pra ver o que é que tava acontecendo.”*

Quando questionado se mudaria algo na ferramenta ou na linguagem textual, Ford respondeu:

*“Cara, na linguagem eu achei a estrutura bem bacana, eu achei que ficou bem legal. Porque ela segue uma linha de linguagem orientada a objetos mesmo, então por exemplo, para alguém que desenvolve com .NET hoje, com C# ou Java, PHP ou com qualquer coisa*

*que hoje em dia tem alguma orientação a objeto, tá bem legal mesmo, tá bem definido. E na ferramenta acho que seria o que já comentei mesmo: alguns facilitadores para a pessoa interagir mais com a sintaxe num primeiro momento, porque depois também vai ficar tão natural que ela vai indo e, por exemplo, Web, pelo menos eu nunca vi, se alguém fez isso o cara é mais maluco do que eu imaginava, seria fazer uns auto-completes que a gente tem nas IDEs hoje...”*

Além disso, sobre a questão a acessibilidade da ferramenta, Ford comentou:

*“...por você ter feito ela Web, ficou muito legal porque roda em qualquer lugar né. Isso é um diferencial bacana e o ambiente Web por si só ele já é um ambiente muito mais acessível. Se você tivesse feito, por exemplo quando você me falou que era em Java eu já tava imaginando a interface Swing, que é um inferno pra se configurar acessibilidade pra aquilo, não fica bom de jeito nenhum, todo ano eu sofro com aquele imposto de renda. Então, por ter feito ele em Web, você já começou com uma plataforma bem bacana e que estende muito até o próprio uso, porque, dentro de uma empresa, por exemplo, você pode ganhar algumas vantagens, como ter a instalação disso em um único servidor na empresa e a empresa inteira usar, que vai facilitar muito. Ou então mesmo, hoje em dia né tá muito comum ir pra nuvem. Mas pensando no lado da acessibilidade é muito legal porque o ambiente, ele é bem acessível.”*

Alguns pontos de melhoria que foram identificados foram:

- Adicionar a coluna em que ocorre o erro, nas mensagens de erro do interpretador
- Adicionar botões para auxiliar o usuário a utilizar estruturas da linguagem como relacionamentos
- Utilizar WAI-ARIA para destacar as mensagens de erro e controlar melhor o foco do cursor para facilitar a navegação.

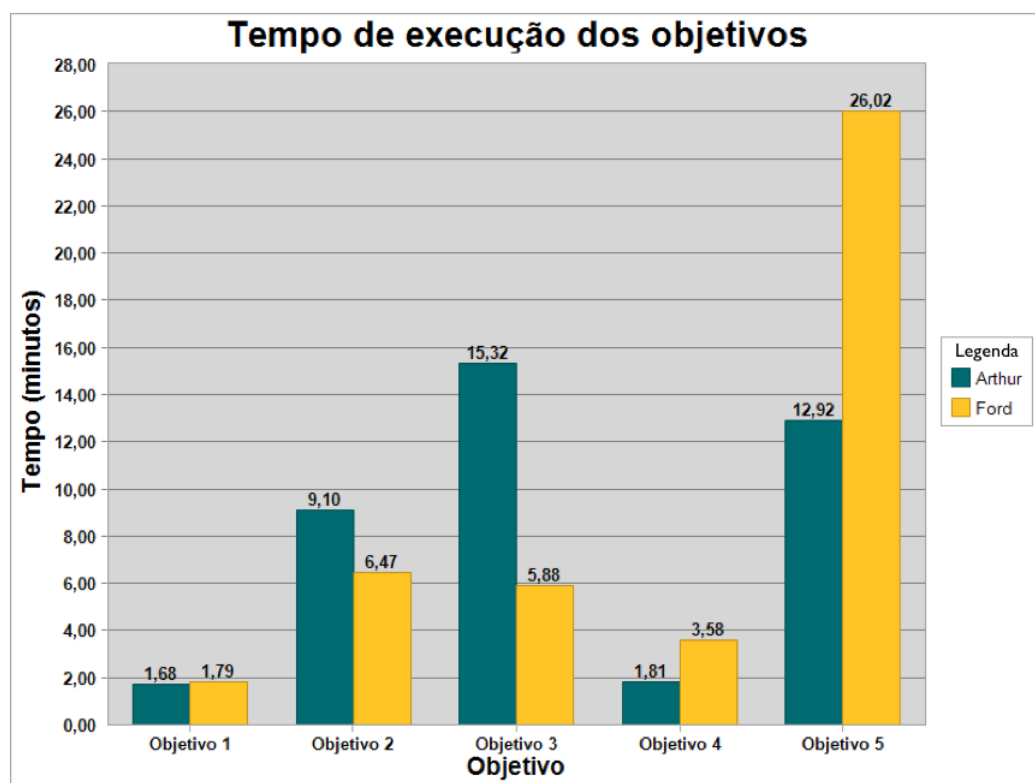
Por fim, pelo fato de Ford ser um desenvolvedor de *software*, foi questionado se teria interesse em contribuir para o projeto da AWMo, que foi publicada como um projeto de *software* livre:

*“Com toda certeza, com toda certeza. E acho que faltam iniciativas desse tipo e toda a iniciativa que vem tem que ser muito apoiada porque é tão difícil olharem pra gente né, que pô, você tem um trabalho de pesquisa, um trabalho de desenvolvimento. Um negócio desse open source eu seria com certeza um dos colaboradores, isso aí você pode contar mesmo.”*



### 4.3 Análise

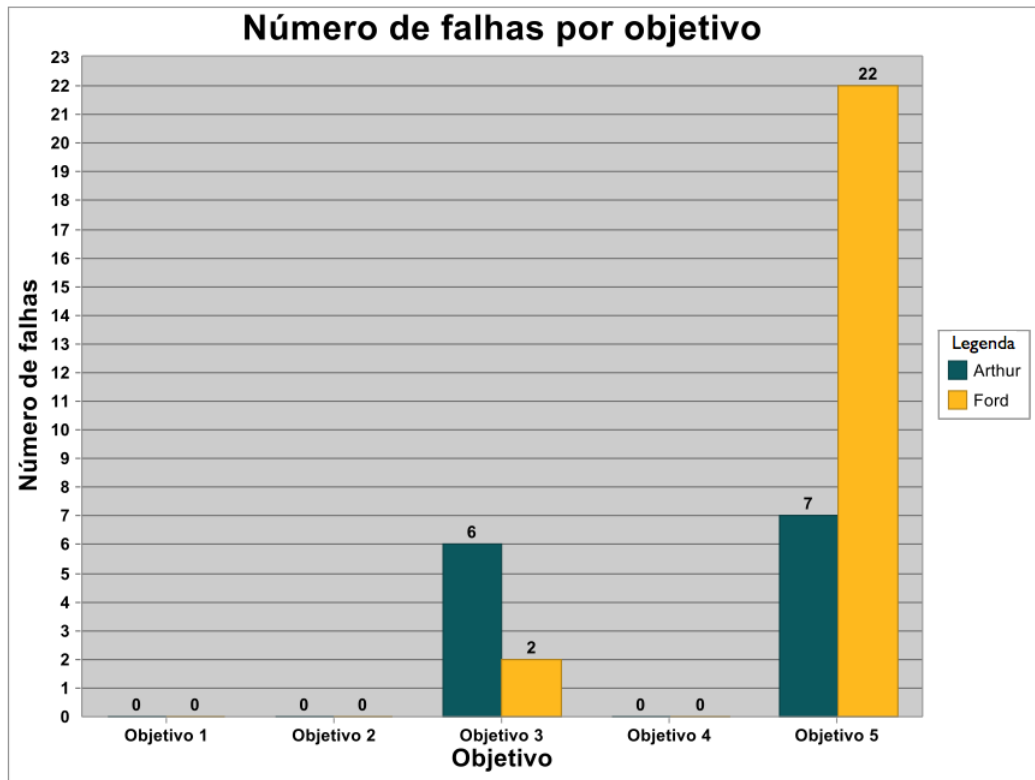
As Figuras 4.8, 4.9 e 4.10, mostram os tempos utilizados, as falhas inseridas e os erros exibidos pelos participantes lado a lado. É possível ver que os comportamentos foram ligeiramente distintos. No objetivo 1 ambos fizeram tempos bastante parecidos, e ambos desempenharam a mesma tarefa: conhecer a localização dos itens de interface e se familiarizar com a AWMo, nenhuma falha ou erro foi cometido durante a realização desse objetivo, por nenhum dos participantes.



**Figura 4.8:** Gráfico mostrando a quantidade de tempo utilizado por cada um dos sujeitos para completar os objetivos propostos neste estudo de caso.

No objetivo 2, Ford utilizou um tempo menor, sendo 6,47 minutos contra 9,10 minutos de Arthur. Essa diferença pequena pode ser explicada pelo fato de Ford trabalhar em uma instituição financeira. Embora o sistema utilizado nesse estudo de caso seja fictício, trata-se de um modelo de um *software* de sistema bancário e, portanto, Ford possuía familiaridade com o assunto modelado. Novamente, nenhuma falha ou erro foi cometido durante esse objetivo, ambos os participantes souberam explicar bem o funcionamento do sistema modelado.

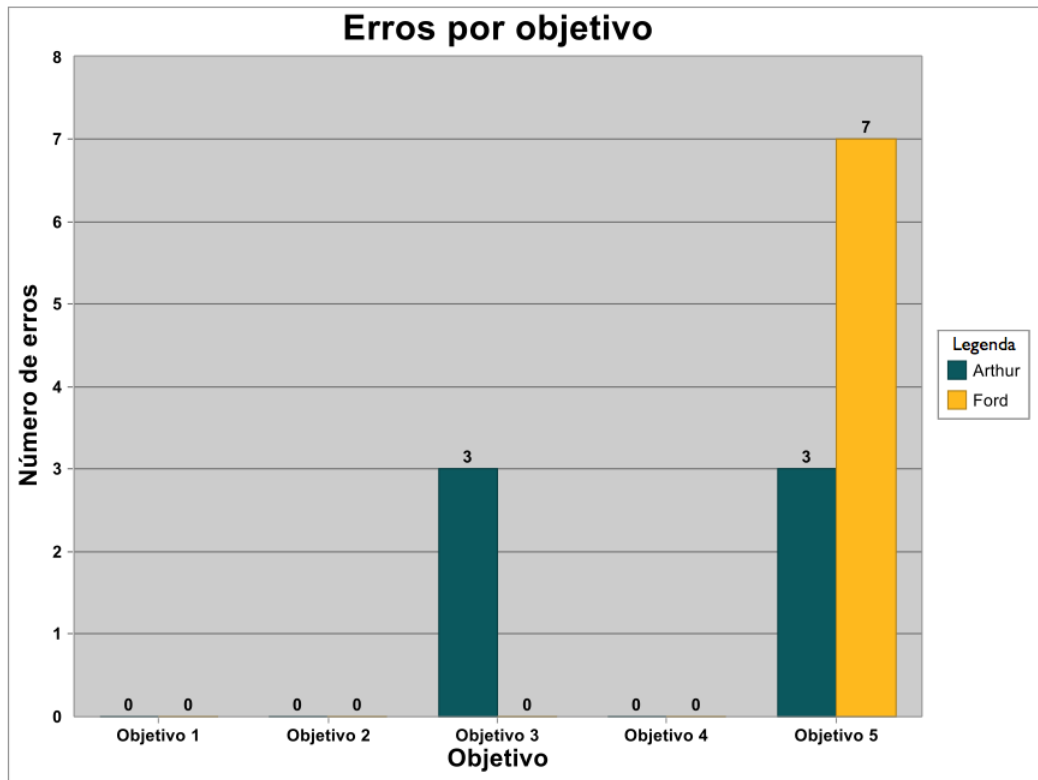
No objetivo 3, novamente Ford utilizou menos tempo e aqui a diferença entre os dois participantes foi maior, 5,88 minutos de Ford contra 15,32 minutos de Arthur. Este objetivo demandava que a classe Pessoa fosse especializada em classes para representar pessoas



**Figura 4.9:** Gráfico mostrando o número de falhas inseridas por cada um dos sujeitos ao decorrer da execução dos objetivos.

físicas e pessoas jurídicas. Ford cometeu um número menor de falhas do que Arthur e Ford corrigiu ambas as 2 falhas antes de salvar o diagrama, e portanto, não houve nenhum erro exibido pela AWMo. A quantidade de falhas e erros cometidos ajudam a explicar a diferença no tempo, porém, um outro fator que deve ser levado em consideração é o fato de que Ford aprendeu UML enquanto possuía visão, e portanto tem uma familiaridade maior com seus conceitos do que Arthur, que só teve contato com a UML após ter perdido a visão.

No objetivo 4, por outro lado, Ford levou um tempo maior do que Arthur, pois Ford removeu os métodos “AbrirPoupanca” e “FecharPoupanca” mesmo não tendo nenhuma relação sintática direta com a classe poupança que pudesse ser apontado pelo interpretador, o que auxiliaria na identificação da forma correta de execução da tarefa. Portanto, embora Ford tenha levado mais tempo, é possível afirmar que o resultado obtido por ele foi melhor do que o que foi obtido por Arthur, sob a perspectiva conceitual da modelagem pretendida. Como o critério de sucesso definido foi a ação de salvar o diagrama, ambos os participantes obtiveram sucesso no desempenho do objetivo. No entanto, sob uma análise mais criteriosa, é possível afirmar que o resultado obtido por ele foi melhor do que o que foi obtido por Arthur, o que torna inconclusiva a análise da diferença entre os tempos utilizados. Nenhum dos participantes cometeram erros ou falhas.



**Figura 4.10:** Gráfico mostrando o número de vezes em que mensagens de erro do interpretador de texto foram exibidas para os sujeitos, separados por objetivo.

Por fim, no objetivo 5 nota-se uma maior discrepância entre os resultados. Assim como nos outros casos, ambos os participantes foram capazes de concluir o objetivo com sucesso. No entanto, Ford levou 26,02 minutos para completar o objetivo enquanto Arthur levou apenas 12,92 minutos, ou seja, o tempo de Ford foi 2 vezes maior. Na Figura 4.9 podemos ver que Ford inseriu 22 falhas ao longo da execução do objetivo 5, enquanto Arthur inseriu apenas 7, novamente o número de falhas inseridas por Ford foi mais do que 3 vezes maior e, conseqüentemente, o número de erros vistos na Figura 4.10 também foi maior: 7 de Ford contra 3 de Arthur.

O objetivo 5 exigia que os participantes criassem um modelo totalmente novo a partir de especificações simples fornecidas nas instruções do objetivo e, portanto, era necessário que o participante tivesse maior domínio sobre a linguagem textual da AWMo. Em comparação com os objetivos anteriores, quando se está alterando um modelo pré-existente, é mais fácil encontrar exemplos das estruturas da linguagem no próprio modelo em que se está trabalhando, ao passo que para se construir um modelo totalmente novo é necessário lembrar melhor das estruturas da linguagem textual ou utilizar o manual da linguagem para consulta.

As Tabelas 4.1 e 4.2 mostram uma categorização das falhas inseridas por ambos os participantes durante a execução dos objetivos propostos por este estudo de caso. Para tal, foram definidas as seguintes categorias de falhas:

- **Gramatical:** ocasiões em que o participante errou palavras reservadas e tipos de dados ou utilizou incorretamente estruturas da linguagem.
- **Lógica:** ocasiões em que o modelo ou o objeto modelado não tem sentido lógico ou difere do resultado que se pretendia alcançar.
- **Digitação:** ocasiões em que o participante cometeu um erro de digitação, seja troca, ausência ou excesso de caracteres.

**Tabela 4.1:** Categorização das falhas inseridas por Arthur ao longo do estudo.

Objetivo	Tempo	Tipo	Observação
3	00:21:48	Digitação	Buscando tecla de atalho
	00:21:56	Digitação	Apagou a quebra de linha
	00:23:25	Digitação	Buscando tecla, apagou espaço
	00:25:54	Digitação	Faltou espaço entre palavras
	00:27:19	Digitação	Pressionou seta para cima acidentalmente
	00:28:38	Gramatical	Tipo básico com letra maiúscula
5	00:41:46	Gramatical	Faltou modificador de acesso do método
	00:42:23	Gramatical	Faltou modificador de acesso do método
	00:42:48	Digitação	Faltou espaço entre palavras
	00:44:14	Digitação	Pressionou “Page Up” junto com “Enter”
	00:46:02	Digitação	Pressionou seta para cima acidentalmente
	00:46:04	Gramatical	Referência a classe com case diferente
	00:46:15	Gramatical	Referência a classe com case diferente

Nota-se que nenhuma falha de lógica foi cometida, não houve nenhuma construção incorreta dos modelos e, como já citado, os participantes concluíram com sucesso os objetivos propostos. A Figura 4.11 mostra alguns gráficos para efeitos de comparação entre essas falhas. Podemos ver que no caso de Arthur, a maior parte das falhas foi de digitação, representando 62% de todas as falhas cometidas por ele. Esse dado reforça o fato observado por meio dos vídeos de que Arthur encontrou dificuldades em se adaptar ao teclado do notebook no qual foi conduzido o estudo de caso.

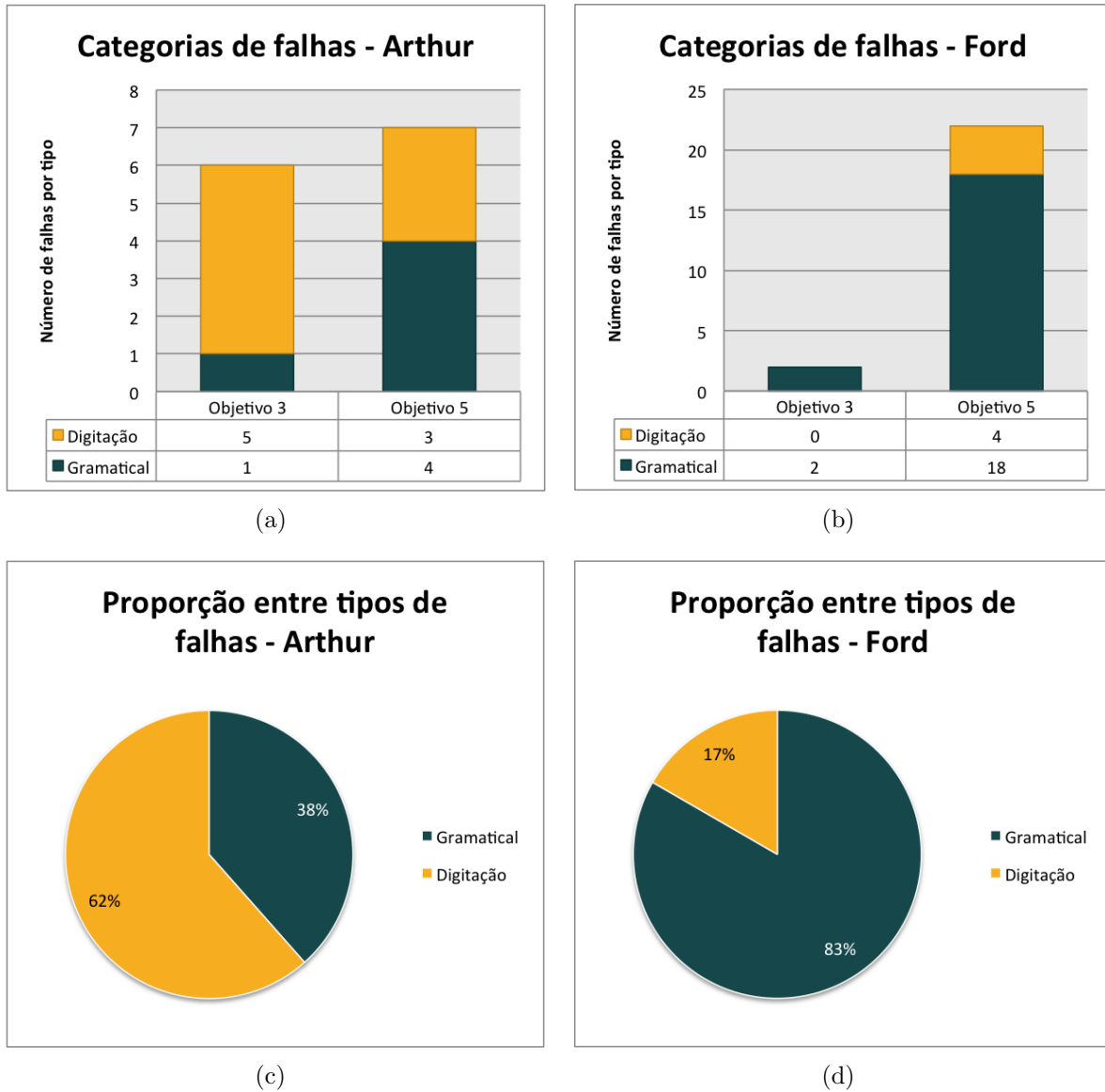
Já no caso de Ford, a quantidade de falhas de digitação foi bem menor, proporcionalmente à quantidade de falhas gramaticais, sendo que apenas 17% das falhas foram de digitação contra 83% de falhas gramaticais. Este número elevado de falhas, principalmente no objetivo 5 como mostra a Figura 4.11(b), mostra que o participante teve

**Tabela 4.2:** Categorização das falhas inseridas por Ford ao longo do estudo.

Objetivo	Tempo	Tipo	Observação
3	00:18:04	Gramatical	Faltou modificador de acesso do atributo
	00:18:29	Gramatical	Faltou modificador de acesso do atributo
5	00:28:36	Gramatical	Faltou modificador de acesso do atributo
	00:28:48	Gramatical	Faltou modificador de acesso do atributo
	00:29:13	Gramatical	Trocou boolean por bool
	00:29:20	Gramatical	Faltou modificador de acesso do método
	00:29:26	Gramatical	Trocou boolean por bool
	00:29:35	Gramatical	Faltou modificador de acesso do método
	00:29:39	Gramatical	Trocou boolean por bool
	00:29:44	Digitação	Escreveu “mrtodo” ao invés de “metodo”
	00:29:50	Gramatical	Faltou modificador de acesso do atributo
	00:29:52	Gramatical	Trocou boolean por bool
	00:30:29	Digitação	Faltou espaço entre palavras
	00:30:56	Gramatical	Faltou modificador de acesso do método
	00:31:00	Gramatical	Trocou boolean por bool
	00:31:08	Gramatical	Faltou modificador de acesso do método
	00:31:11	Gramatical	Trocou boolean por bool
	00:31:22	Gramatical	Faltou modificador de acesso do método
	00:31:24	Gramatical	Trocou boolean por bool
	00:31:27	Digitação	Faltou espaço entre palavras
	00:31:29	Gramatical	Faltou modificador de acesso do método
	00:31:31	Gramatical	Trocou boolean por bool
00:32:51	Gramatical	Faltou palavra-chave “relacao”	
00:35:20	Digitação	Escreveu “booeanl” ao invés de “boolean”	

bastante dificuldade com as palavras chave e estruturas de linguagem textual. Grande parte dessas falhas foram esquecimento de modificador de acesso, que define se um método ou atributo é público ou privado e outro problema bastante recorrente foi a troca do tipo de dado “boolean” por “bool”. Ambos são erros de gravidade baixa, e indicam apenas falta de familiaridade com a linguagem textual. Tanto pela observação quanto pelos dados coletados durante as entrevistas, não houveram indícios de problemas graves ou dificuldades maiores relacionadas à linguagem textual.

Para ambos os participantes, o objetivo 5 foi o que gerou mais falhas do tipo gramatical em função da tarefa de construção de um modelo novo, escrito inteiramente pelos participantes.



**Figura 4.11:** Gráficos ilustrando a categorização das falhas exibidas nas Tabelas 4.1 e 4.2 sendo: (a) - quantidade de falhas de Arthur em cada categoria e separadas por objetivo, (b) - quantidade de falhas de Ford em cada categoria e separadas por objetivo, (c) - proporção entre as categorias das falhas cometidas por Arthur e (d) - proporção entre as categorias das falhas cometidas por Arthur

---

## Conclusões e trabalhos futuros

---

### 5.1 Conclusões

Levando-se em consideração que os objetivos utilizados na fase de observação não foram demasiadamente complexos, mas, similares a tarefas as quais um desenvolvedor de *software* pode encontrar ao trabalhar com diagramas de classe da UML em projetos reais, o estudo de caso realizado reproduziu situações representativas para avaliação da AWMo.

Assim, foi possível avaliar a viabilidade da proposta da AWMo. No estudo de caso, a AWMo proposta foi objeto de investigação, para verificar se ela permite acesso a modelos de *software* para deficientes visuais, bem como se a estratégia de uso da AWMo proposta é apropriada para ser utilizada em ambientes reais, tais como empresas que trabalham com desenvolvimento de *software* e empregam profissionais com deficiência visual.

No objetivo 3, por exemplo, Arthur levou 15,32 minutos e Ford levou 5,88 minutos para completá-la, sendo que foi necessário a modificação de uma classe e criação de duas outras com relação de herança com a primeira. Isso representa um tempo bastante pequeno, principalmente se levarmos em consideração que ambos tiveram o primeiro contato com o diagrama durante a realização do objetivo 2, por cerca de 9 (Arthur) e 6 (Ford) minutos apenas.

O mesmo pode ser observado no objetivo 5, durante o qual, os participantes precisaram criar um modelo completamente novo a partir de especificações simples fornecidas pelas instruções do objetivo. Considerando que ambos estavam tendo seu primeiro contato com a AWMo, e estavam lendo as instruções do objetivo pela primeira vez, Arthur levou menos de 13 minutos para criar o novo diagrama, abrí-lo na visão textual e modelá-lo

de acordo com as especificações e Ford levou cerca de 26 minutos. Apesar da diferença entre os tempos dos dois participantes, acreditamos fortemente que a abordagem utilizada pela AWMo para modelagem de *software* não traz nenhuma sobrecarga de trabalho ou aumento no tempo utilizado na atividade de modelagem enquanto traz o benefício de que, uma vez que o modelo seja criado com uso da linguagem textual como feito pelos participantes, a representação gráfica está automaticamente pronta para ser utilizada por usuários com visão.

No caso específico de Ford, que levou um tempo maior, o próprio participante comentou sobre isso em alguns trechos extraídos da pós-entrevista:

*“Cara, na verdade, a dificuldade foi lembrar a sintaxe, mas foi a primeira vez que eu usei, então com certeza usando isso no dia a dia se torna um pouco mais fácil. E as mensagens de erro, elas poderiam conter ali a mais, além da linha, que eu acho que facilita pra você chegar na linha e tudo mais. Seria alguma coisa como, por exemplo, a própria linha mesmo como algumas linguagens fazem na compilação, ela colocar ali um caractere do lado da onde ele acha que está o problema né.”*

*“...a maior dificuldade que a pessoa vai sentir sinceramente, se é uma pessoa que usa a Web, é lembrar da sua sintaxe. Como eu disse novamente, depois de dois ou três diagramas que a pessoa fizer, isso vai tá tão natural pra ela que ela não vai sentir diferença, mas acho que num primeiro momento você ter ali os atalhos já ajudaria, já seria um facilitador pelo menos, pra ao invés dele ter que recorrer ao manual ele dá um “enterzinho” ali e já insere”*

Nota-se que, embora a quantidade de falhas gramaticais tenha sido alta, o próprio participante acredita que esse fato ocorreu devido à falta de prática com a linguagem, tanto que quando questionado se mudaria algo na estrutura da linguagem respondeu:

*“Cara, na linguagem eu achei a estrutura bem bacana, eu achei que ficou bem legal. Porque ela segue uma linha de linguagem orientada a objetos mesmo, então por exemplo, para alguém que desenvolve com .NET hoje, com C# ou Java, PHP ou com qualquer coisa que hoje em dia tem alguma orientação a objeto, tá bem legal mesmo, tá bem definido.”*

Portanto, há indícios de que apenas pouco ou nenhum tempo foi adicionado ao processo de modelagem de *software* se comparado com o processo de construção de modelos por meio de descrições textuais em linguagem natural para deficientes visuais, bem como, o entendimento dos diagramas por usuários com visão, que não são acostumados com as descrições textuais, é natural pois existe o acesso à representação gráfica da maneira como estão familiarizados a trabalhar. Isso indica que a AWMo se mostra como uma opção viável para facilitar o acesso de desenvolvedores de *software* deficientes visuais a modelos de *software* e promover a colaboração e comunicação efetiva e de maneira



independente entre usuários com e sem visão para atividades de modelagem de *software*, como mencionado pelos próprios participantes ao longo das entrevistas.

Além disso, tanto pelas informações coletadas nas entrevistas, quando ao analisar a eficiência com que os participantes utilizaram a linguagem textual é possível ver que a linguagem em si não se apresentou como uma barreira para a utilização da abordagem proposta pela AWMo. Embora exista espaço para melhorias, com pouco tempo de estudo independente, utilizando apenas uma apostila em PDF fornecida pelo pesquisador, os participantes foram capazes de aprender a linguagem textual e utilizá-la de maneira eficaz durante todos os objetivos propostos neste estudo.

## 5.2 Trabalhos futuros

Como trabalhos futuros, os autores pretendem executar o estudo de caso em questão com mais sujeitos e projetar diferentes experimentos para avaliar a interface gráfica e o uso da AWMo por usuários com visão.

Do ponto de vista técnico, a AWMo possui uma arquitetura bastante flexível que permite que futuros colaboradores desenvolvam novas formas de visualização e edição de modelos sem afetar os que já existem. Uma dessas possíveis formas de visualização, é uma representação em HTML, utilizando *links* para melhorar a navegação do usuário no conteúdo do modelo durante sua leitura e permitindo que a leitura possa ser realizada de maneira não linear.

Com isso em mente, a AWMo será disponibilizada como uma ferramenta *open source* e um manual técnico será publicado para auxiliar futuros desenvolvedores que possam vir a contribuir com a ferramenta.



---

## Referências

---

---

- Andronico, P.; Buzzi, M.; Castillo, C.; Leporini, B. Improving search engine interfaces for blind users: a case study. *Universal Access in the Information Society*, v. 5, p. 23–40, 2006.
- Cohen, R. F.; Meacham, A.; Skaff, J. Teaching graphs to visually impaired students using an active auditory interface. *SIGCSE Bull.*, v. 38, p. 279–282, 2006.
- Freire, A. P.; Linhalis, F.; Bianchini, S. L.; Fortes, R. P.; da Graça C. Pimentel, M. Revealing the whiteboard to blind students: An inclusive approach to provide mediation in synchronous e-learning activities. *Computers & Education*, v. 54, n. 4, p. 866 – 876, 2010.
- IBGE *Censo Demográfico 2010 - Resultados Preliminares da Amostra*. Rio de Janeiro: Instituto Brasileiro de Geografia e Estatística, 2010.
- Kaczmirek, L.; Wolff, K. Survey design for visually impaired and blind people. In: Stephanidis, C., ed. *Universal Access in Human Computer Interaction. Coping with Diversity*, v. 4554 de *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, p. 374–381, 2007.
- King, A.; Blenkhorn, P.; Crombie, D.; Dijkstra, S.; Evans, G.; Wood, J. Presenting UML Software Engineering Diagrams to Blind People. In: Miesenberger, K.; Klaus, J.; Zagler, W.; Burger, D., eds. *Computers Helping People with Special Needs*, v. 3118 de *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, p. 626–626, 2004.
- Lazar, J.; Feng, J.; Hochheiser, H. *Research methods in human-computer interaction*. Indianapolis, IN: Wiley, 2010.

- Lethbridge, T. C.; Sim, S. E.; Singer, J. Studying software engineers: Data collection techniques for software field studies. *Empirical Softw. Engg.*, v. 10, n. 3, p. 311–341, 2005.  
Disponível em <http://dx.doi.org/10.1007/s10664-005-1290-x>
- Metatla, O.; Bryan-Kinns, N.; Stockman, T. Using hierarchies to support non-visual access to relational diagrams. In: *Proceedings of the 21st British HCI Group Annual Conference on People and Computers: HCI...but not as we know it - Volume 1*, BCS-HCI '07, Swinton, UK, UK: British Computer Society, 2007, p. 215–225 (*BCS-HCI '07*, ).
- Metatla, O.; Bryan-Kinns, N.; Stockman, T. Comparing interaction strategies for constructing diagrams in an audio-only interface. In: *Proceedings of the 22nd British HCI Group Annual Conference on People and Computers: Culture, Creativity, Interaction - Volume 2*, BCS-HCI '08, Swinton, UK, UK: British Computer Society, 2008, p. 65–69 (*BCS-HCI '08*, ).
- Runeson, P.; Höst, M. Guidelines for conducting and reporting case study research in software engineering. *Empirical Software Engineering*, v. 14, n. 2, p. 131–164, 2008.  
Disponível em <http://www.springerlink.com/index/10.1007/s10664-008-9102-8>
- Santanna, R. Portaria Nº 3, de 7 de maio de 2007. Institucionaliza o Modelo de Acessibilidade em Governo Eletrônico - e-MAG no âmbito do Sistema de Administração dos Recursos de Informação e Informática - SISP. *Diário Oficial [da] República Federativa do Brasil*, p. 103, n. 87, Seção 1, ISSN 1677-7042, 2007.
- Shinohara, K.; Tenenberg, J. Observing sara: a case study of a blind person's interactions with technology. In: *Proceedings of the 9th international ACM SIGACCESS conference on Computers and accessibility*, Assets '07, New York, NY, USA: ACM, 2007, p. 171–178 (*Assets '07*, ).  
Disponível em <http://doi.acm.org/10.1145/1296843.1296873>
- Siegfried, R. M. Visual programming and the blind: the challenge and the opportunity. *SIGCSE Bull.*, v. 38, p. 275–278, 2006.
- Stake, R. *The art of case study research*. Sage Publications Inc, 1995.  
Disponível em [http://books.google.com/books?hl=no&lr=&id=ApGdBx76b9kC&oi=fnd&pg=PR11&dq=case+study&ots=KrNKk2Fnbo&sig=tBVSrcFVI4tzDG3AzErMhF\\_xe4](http://books.google.com/books?hl=no&lr=&id=ApGdBx76b9kC&oi=fnd&pg=PR11&dq=case+study&ots=KrNKk2Fnbo&sig=tBVSrcFVI4tzDG3AzErMhF_xe4)
- Yin, R. K. *Case Study Research: Design and Methods*. 2nd ed. SAGE Publications Inc., 1994.

---

# Guia para entrevista pré-uso

---

## A.1 Assuntos a serem abordados

A entrevista que será realizada antes do sujeito utilizar a AWMo foi projetada para abordar os assuntos a seguir (mas não é limitada a eles):

- Experiência do entrevistado com programação;
- Contato anterior com a UML ou outra linguagem de modelagem;
- Identificar os problemas de acesso a diagramas;
- Entender a expectativa do entrevistado em relação a AWMo; e
- Colher requisitos para novos desenvolvimentos.

## A.2 Questões

Para ser utilizada como guia para a realização da entrevista, a lista de questões a seguir foi elaborada. No entanto, como se pretende realizar entrevistas semiestruturadas, não será necessário que as questões sejam colocadas na ordem que se encontram.

### 1. Dados pessoais

Esta questão busca obter informações sobre o sujeito que possam ser utilizadas para categorizar as outras informações coletadas ou no auxílio na avaliação dos dados coletados posteriormente. Entre os dados que busca-se obter estão:

- Idade;
- Se possui alguma deficiência visual, qual e como ela afeta o seu trabalho; e
- Se utiliza alguma configuração especial para facilitar o uso de computadores? (Como fontes ampliadas, alto contraste, etc.),

**2. Qual a sua experiência com o uso de computadores?**

Esta questão visa obter mais informações sobre o entrevistado de maneira qualitativa.

**3. Qual sua experiência com programação de computadores?**

Busca-se registrar quantos anos de experiência o sujeito possui com programação e com quais linguagens e tecnologias ele trabalha ou já trabalhou.

**4. Você já teve contato com diagramas da UML?**

Se o entrevistado nunca teve nenhum contato com diagramas de classe da UML, será necessário uma breve introdução sobre o assunto antes da realização das tarefas, na fase de observação do uso da ferramenta.

**5. Se sim, como foi essa experiência?**

Esta questão busca entender o que o entrevistado conhece de UML. Até que ponto o entrevistado utilizou a UML e quais foram suas impressões.

**6. Se não, você já teve contato com alguma outra linguagem de modelagem?**

Esta questão busca entender até que ponto o entrevistado utilizou o quais foram suas impressões.

**7. Na sua opinião, qual o principal problema de acesso a esses diagramas?**

Questão que pode ser complementar à anterior, se o entrevistado já teve contado com a UML. No entanto, ela pode ser descartada se o entrevistado nunca teve nenhum contato anterior com a UML ou qualquer outra linguagem de modelagem.

**8. Você acredita que o acesso a atividades de modelagem é importante para a realização do seu trabalho?**

Busca-se descobrir se em algum momento o entrevistado já sentiu necessidade de utilizar modelagem ou que poderia ter realizado alguma tarefa de maneira melhor, caso utiliza-se.

**9. O que você espera de uma ferramenta que busca o acesso e edição de diagramas por meio de uma interface textual?**

De maneira qualitativa, busca-se entender o que o entrevistado espera da ferramenta

e a possibilidade de coletar novos requisitos para evolução da AWMo e realização de trabalhos futuros.





---

## Tarefas para a fase de observação

---

Para a fase de observação deste estudo de caso, os participantes serão convidados a utilizar a AWMo para realizar uma série de tarefas pré-estabelecidas enquanto suas interações com a ferramenta serão gravadas.

### B.1 Treinamento na linguagem textual

Para utilizar a visão textual da AWMo é necessário que o usuário conheça a linguagem por meio da qual são expressados os diagramas de classe nessa ferramenta. Antes que o usuário inicie o uso da ferramenta ele será instruído sobre como é o funcionamento da linguagem textual da AWMo. Para tal, o sujeito receberá antecipadamente material para que irá explicar o funcionamento da linguagem com diversos exemplos de uso. Momentos antes do início da observação, serão sanadas quaisquer dúvidas existentes a respeito da linguagem textual para que isso não se apresente como uma dificuldade durante a utilização do sistema.

### B.2 Lista de objetivos

A lista de tarefas a seguir será entregue aos participantes, assim como quaisquer arquivos ou pré-configurações que sejam necessárias para a efetiva realização das mesmas.

**Objetivo 1** - Ajuste a AWMo de acordo com suas preferências (Tamanho de fonte, contraste e idioma) utilizando as funcionalidades de acessibilidade existentes na própria AWMo. Se necessário pode-se utilizar de configurações do navegador e/ou sistema

operacional para tornar a utilização da ferramenta o mais confortável possível.

Este objetivo tem a função de avaliar como o usuário irá localizar e interagir com os controles de acessibilidade do AWMo e configurar de modo que o restante das atividades sejam realizadas de maneira mais confortável para o usuário.

**Tarefas necessárias para a realização deste objetivo:**

1. Navegar até a barra de navegação de acessibilidade;
2. Utilizar os *links* para alterar as configurações de acessibilidade da AWMo; e
3. Repetir o item 2 até que a AWMo esteja configurada de acordo com as preferências do usuário.

**Objetivo 2** - Abrir o diagrama “Sistema Bancário” já existente na AWMo e entender seu conteúdo e explicá-lo com suas palavras ao pesquisador.

Este objetivo tem a função de avaliar quais as dificuldades de entendimento da linguagem textual da AWMo e também preparar o sujeito para os próximos objetivos que serão realizados utilizando o mesmo diagrama como base.

**Tarefas necessárias para a realização deste objetivo:**

1. Navegar na página inicial até a tabela de diagramas;
2. Clicar no *link* “Textual” ao lado do diagrama Sistema Bancário;
3. Navegar até a área de texto onde está o conteúdo do diagrama Sistema Bancário;
4. Ler o diagrama buscando compreender as informações nele contidas; e
5. Explicar o modelo ao pesquisador.

**Objetivo 3** - Ainda no diagrama Sistema Bancário: O engenheiro responsável pela modelagem do sistema bancário se esqueceu que o banco em questão atende tanto a pessoas físicas quanto a pessoas jurídicas. Sua tarefa é corrigir este problema: a Pessoa, dona de uma conta, pode ser tanto uma pessoa física quanto uma pessoa jurídica (empresa). Represente essa informação no diagrama de classes. (Se possível, utilize o relacionamento de herança). Implemente os métodos e atributos que julgar necessário.

Este objetivo busca avaliar se o sujeito está confortável ao utilizar as estruturas da linguagem e quais serão suas dificuldades ao realizar uma alteração pontual em um diagrama já existente e desenvolvido por outra pessoa.

**Tarefas necessárias para a realização deste objetivo:**

1. Assuma-se que o diagrama Sistema Bancário já foi aberto no objetivo 2;
2. Navegar no texto até a classe Pessoa;
3. Inserir duas novas classes (Pessoa Física e Pessoa Jurídica); e
4. Fazer com que as duas novas classes herdem a classe Pessoa.

**Objetivo 4** - Ainda no diagrama Sistema Bancário. Na versão atual o modelo do sistema está considerando o funcionamento de contas do tipo Poupança. No entanto, o cliente deseja que a conta Poupança seja desenvolvida apenas no futuro, portanto, remova a Poupança desta versão do modelo. (Remova também as relações da Poupança com as outras classes onde for cabível).

Este objetivo busca verificar se a navegação no diagrama textual é simples. O sujeito irá remover uma classe e deverá buscar nas outras se algum atributo ou método possui relação com a classe que acabou de remover.

**Tarefas necessárias para a realização deste objetivo:**

1. Assuma-se que o diagrama Sistema Bancário já foi aberto no objetivo 2;
2. Encontrar a classe Poupança;
3. Apagar o conteúdo da classe poupança;
4. Buscar Atributos, Métodos ou Relações que utilizam a classe Poupança e removê-los; e
5. Salvar o diagrama.

**Objetivo 5** - Crie um novo diagrama e chame-o de Automóvel. Em seguida abra o diagrama para edição no modo textual e insira as classes Carro e Motor. Sendo que Carro deve possuir os atributos “cor” e “modelo” e o Motor deve possuir os atributos “potência” e “cilindros”. Represente o fato de que um carro é composto por um motor. Crie os métodos que achar necessário.

Este objetivo busca analisar como o usuário se comporta ao criar um diagrama inteiramente novo a partir de uma especificação simples.

**Tarefas necessárias para a realização deste objetivo:**

1. Navegar até a página inicial;
2. Navegar até o campo de criação de novo diagrama;
3. Digitar o nome do diagrama: “Automóvel”;
4. Pressionar o botão “Novo diagrama”;

5. Navegar até lista de diagramas;
6. Abrir o diagrama recém criado ativando o *link* “Textual”;
7. Navegar até a caixa de texto;
8. Inserir classes, atributos, métodos e relacionamentos até que o modelo esteja completo; e
9. Salvar o diagrama.

---

# Guia para entrevista pós-uso

---

## C.1 Assuntos a serem abordados

A entrevista que será realizada após a utilização da AWMo foi projetada para abordar os assuntos a seguir (mas não é limitada à eles):

- Problemas e dificuldades encontradas;
- Funcionalidades e sua importância;
- Funcionalidades faltantes;
- Possibilidade de contribuição com o desenvolvimento da AWMo

## C.2 Questões

Para ser utilizada como guia para a realização da entrevista, a lista de questões a seguir foi elaborada. No entanto, como se pretende realizar entrevistas semiestruturadas, não será necessário que as questões sejam colocadas na ordem que se encontram.

1. **Quais foram as principais dificuldades que você encontrou ao utilizar a AWMo?**

Esta questão tem o objetivo de ajudar a entender quais foram os possíveis problemas de interação que o usuário encontrou ao experimentar a ferramenta e além disso,

entender quais foram os motivos por trás dessas dificuldades: se a ferramenta representa uma mudança de paradigma muito grande, a falta de experiência ou problemas de usabilidade ou funcionamento da AWMo.

**2. Quais foram as funcionalidades que você achou mais interessantes? Em ordem de preferência.**

Esta questão será utilizada de maneira qualitativa para entender quais são as funcionalidades que os usuários mais utilizou/gostou na AWMo para ajudar a entender as suas necessidades e quais os aspectos que ele observou na ferramenta.

**3. O que você mais sentiu falta enquanto utilizava a ferramenta?**

O principal objetivo dessa questão é identificar aspectos essenciais para a utilização da ferramenta que foram deixados de fora no seu desenvolvimento, dessa forma, espera-se que quaisquer problemas críticos existentes na ferramenta possam ser sanados.

**4. Você mudaria algo na AWMo? O que?**

Esta questão tem o objetivo de colher novos requisitos e ao mesmo tempo identificar falhas na construção da AWMo que podem ser melhoradas/adicionadas em versões futuras.

**5. Se a AWMo for divulgada como um projeto de código aberto, você teria interesse em se tornar um colaborador do projeto?**

Com esta questão busca-se entender se o projeto da AWMo é motivador para a comunidade de desenvolvedores de *software* deficientes visuais e, com isso, incentivar seu contínuo desenvolvimento pela comunidade.

---

## Carta-convite

---

Gostaríamos de convidá-lo(a) para participar do estudo de caso “Modelagem de software acessível na Web”. Este estudo de caso tem como objetivo avaliar uma alternativa de acesso à diagramas de classe da UML utilizando uma linguagem textual. Para tal será utilizado um protótipo funcional de uma ferramenta Web, chamado AWMo (*Accessible Web Modeler*), desenvolvido especialmente para este estudo de caso.

A AWMo permite que diagramas de classe da UML sejam acessados por duas visões distintas: uma visão gráfica, onde se cria e altera os diagramas de maneira tradicional e uma visão textual por onde os mesmos diagramas podem ser criados e alterados por meio do uso de uma linguagem textual desenvolvida especificamente para a AWMo.

O estudo será composto por 3 partes. Inicialmente será realizada uma entrevista com questões abertas para conhecê-lo melhor e conhecer também suas experiências em computação.

Depois você irá utilizar a AWMo para realizar um conjunto pequeno de tarefas, enquanto um programa no computador irá gravar a tela e todos os movimentos do mouse e teclado para que eles possam ser analisados posteriormente. Um pesquisador estará ao seu lado para tirar quaisquer dúvidas que possam surgir durante esta etapa.

Por fim, será realizada mais uma entrevista com questões abertas para entender melhor como foi sua experiência de uso da AWMo e quais foram suas impressões.

Todas as informações coletadas serão utilizadas exclusivamente para avaliar a AWMo e a técnica de interação textual proposta. Por isso, não se preocupe, quaisquer informações referentes ao estudo que venham a ser publicadas serão totalmente anônimas. Seu nome e nenhuma outra informação que possa identificá-lo serão citados.

Agradecemos a sua colaboração e informamos que este estudo faz parte de testes finais de um Mestrado realizado no Instituto de Ciências Matemáticas e de Computação da USP e conta com auxílio financeiro do CNPq.

Qualquer dúvida, por favor, entre em contato:

**Nome:**

Filipe Del Nero Grillo

Profa. Dra. Renata Pontin de Mattos Fortes

**Email:**

grillo@icmc.usp.br

renata@icmc.usp.br



---

## Termo de confidencialidade

---

1. Você está sendo convidado para participar da pesquisa “Modelagem de software acessível na Web”.
2. Esta pesquisa tem como intuito observar a interação de usuários com um sistema Web para coletar indícios de usabilidade e acessibilidade.
  - (a) O sistema utilizado por esta pesquisa chama-se AWMo (*Accessible Web Modeler*) e possui a finalidade de criar e editar diagramas de classe da UML (*Unified Modelling Language*) por meio de interfaces de texto e tradicional (gráfica). O intuito da ferramenta é fornecer uma visão acessível para acesso, criação e edição de diagramas de classe.
  - (b) Você foi selecionado para desenvolver o papel de um engenheiro de *software* utilizando a AWMo para realizar atividades de modelagem.
  - (c) Sua participação nesta pesquisa consistirá em interagir com o sistema Web AWMo. Será realizada uma pequena entrevista antes da interação e posteriormente uma nova entrevista será conduzida para que você possa relatar sua experiência, bem como pontos positivos e negativos.
3. Sua participação neste estudo de caso envolve riscos mínimos e é voluntária (você não terá nenhum benefício financeiro por participar deste estudo), podendo você recusar-se a participar ou retirar seu consentimento em qualquer fase da pesquisa, sem penalização alguma.

- (a) A qualquer momento você pode desistir de participar e retirar seu consentimento.
  - (b) Sua recusa não trará nenhuma prejuízo em sua relação com o pesquisador ou com a Instituição.
4. A pesquisa será realizada em um ambiente de sua preferência. O equipamento necessário será levado pelo pesquisador. Durante a pesquisa você será assistido pelo pesquisador que estará conduzindo o estudo de caso. O recurso a ser utilizado é um computador e o tempo do estudo de caso não ultrapassará o limite de 2 (duas) horas.
5. A qualquer momento, antes ou durante a pesquisa, você poderá solicitar esclarecimentos a respeito dos procedimentos ou qualquer outra questão relacionada com a pesquisa.
6. Seus dados pessoais envolvidos na pesquisa serão confidenciais.
- (a) Os dados coletados no estudo de caso serão analisados e todos os participantes receberão por e-mail os resultados da análise dos dados coletados, por meio de artigo publicado ou um capítulo específico da dissertação de mestrado sobre o estudo de caso desenvolvido.
  - (b) Toda e qualquer informação coletada durante o estudo é tratada como confidencial. Os dados não serão divulgados de forma a possibilitar sua identificação, para isso serão utilizados nomes e siglas fictícias para a apresentação dos dados.
7. Os resultados obtidos com esta pesquisa serão utilizados para investigar melhorias na AWMo e na técnica proposta para permitir acesso não visual para modelos de *software*.
8. Você receberá uma cópia deste termo onde consta o contato dos pesquisadores, podendo tirar quaisquer dúvidas sobre o projeto e sua participação, agora ou a qualquer momento. Você poderá também, entrar em contato com os pesquisadores no endereço Avenida Trabalhador são-carlense, 400 - Centro - São Carlos, SP ou através dos emails abaixo:

**Nome:**

Filipe Del Nero Grillo

Profa. Dra. Renata Pontin de Mattos Fortes

**Email:**

grillo@icmc.usp.br

renata@icmc.usp.br

APÊNDICE E. TERMO DE CONFIDENCIALIDADE

---

São Carlos, \_\_\_\_\_ de \_\_\_\_\_ de \_\_\_\_\_

---

Participante:

CPF:

---

Pesquisador responsável:

CPF:



---

# Transcrições das entrevistas com o piloto

---

## F.1 Pré-entrevista

**Filipe:** Quantos anos você tem?

**Marvin:** 41 anos

**Filipe:** O que fez você vir fazer o curso de computação?

**Marvin:** Eu já fiz outro curso aqui, eu fiz matemática. Faz tempo, eu me formei em 98. Só que não dá muito dinheiro esse negócio né? Então eu resolvi vir fazer sistemas de informação. Eu vim no embalo dessa onda que o pessoal tá aí de “ah agora computação vai dar dinheiro”, “o mercado tá defasado, está precisando de gente”. Aí eu falei: “Ah, então eu vou lá fazer o curso”, prestei vestibular e entrei.

Eu já estava estudando algumas coisinhas e o curso de sistemas de informação é a minha cara mesmo. Foi até um amigo meu que é professor aqui, ele falou assim: “Mário, se eu fosse você eu vinha fazer esse curso, é a sua cara, tudo que você está estudando, eles dão aqui no curso. Se eu fosse você eu vinha fazer. Entra no site e dá uma olhada”.

Aí eu entrei no site e vi qual que era a grade do curso de informática que chamava né? Mas mudaram o nome. Aí eu olhei lá e falei “Nossa, é tudo que eu quero aprender mesmo”. Aí eu liguei para ele e falei “Olha, vou fazer o vestibular, posso ficar na sua casa?”

**Filipe:** Você não é de São Carlos?

**Marvin:** Não, sou de Araçatuba.

**Filipe:** Você gosta do curso?

**Marvin:** A eu curto, muito legal.

**Filipe:** Você possui alguma deficiência visual?

**Marvin:** Eu tenho uma cicatriz no fundo do olho, na retina, que me deixa com 10% da visão apenas. Nos dois olhos. Eu disfarço muito bem, as pessoas pensam que eu enxergo, mas não enxergo não.

**Filipe:** E para você usar o computador, você utiliza alguma configuração específica? Algum *software*?

**Marvin:** Eu costumo colocar os ícones grandes e olho de perto quando não tem jeito. Por exemplo, eu uso um *netbook*, prefiro porque ele é mais portátil e o *netbook* se você aumentar muito a fonte, ele perde uma parte da tela. Por exemplo, se você tem uma janela, ela fica maior que a tela aí você não consegue ver os botões na parte de baixo. A tela em si não é rotativa (não se desloca). Aí fica complicado, então eu deixo a fonte no tamanho melhor possível e é pequeno, então as vezes eu preciso colocar o rosto perto da tela para enxergar. Eu deixo os ícones grandes e deixo o *desktop* com pouquíssima coisa, para os ícones grandes não fiquem incômodos.

**Filipe:** Você não costuma utilizar leitores de tela?

**Marvin:** Não eu não gosto de usar aquilo, a lupa você fala?

**Filipe:** Não, aquele que lê mesmo, que vai falando o que está escrito.

**Marvin:** Não, aquilo lá é complicado. Eu até uma vez resolvi configurar pra ver se era legal, mas o meu não tinha, pois eu comprei um computador que vem com Windows 7 starter, aí não vem nada.

**Filipe:** Há quanto tempo você usa computadores?

**Marvin:** Bastante tempo, uso desde criança, da sétima série acho. Desde 84. O computador naquela época era o CP 500. Depois do CP 500, que usava o processador Z80, que era um processador de 8 bits, bem legalzinho, era interessante. Aí começaram a desenvolver os 16 bits: 8086, 80286, 386 e 486. Teve muito desenvolvimento de lá pra cá. Aí do Pentium aí começou pentium 1, 2, 3.

**Filipe:** E com programação de computadores?

**Marvin:** Programação eu fiz um curso quando eu tinha uns 18 anos. Eu fiz o técnico em processamento de dados. Só que esse curso era voltado para você terminar o técnico e sair trabalhando, ele não te preparava para a universidade. Então eu posso dizer que, depois que eu entrei na matemática, eu praticamente deixei de lado aquilo ali e comecei a estudar matemática, então eu fiquei defasado rápido, não me aperfeiçoei naquela época. Então, não me serviu de muita coisa, foi praticamente um colegial que eu fiz. Mas naquela época eu aprendi a programar em Basic, aquele Basic interpretado que vinha no computador. Você nem sonha o que que é isso. E o COBOL, no tempo em que o COBOL era a linguagem de programação de computadores de grande porte, por exemplo, tinham duas linguagens, era o Fortran para universidade e programação mais científica e a programação comercial, banco de dados, toda aquela coisa era feita com COBOL. Acho que eu lembro até hoje alguma coisa.

Aí eu terminei o colegial e vim pra cá fazer matemática. Fiz um ano de cursinho e depois vim pra cá. No curso de matemática estudei um pouquinho de computação: linguagem C básica, alguma coisinha a mais do que a matemática, mas não mexi muito com computador não. Tive programação básica, depois cálculo numérico e depois eu fiz C, a mais, então acabei aprendendo alguma coisinha a mais. Quando saí eu acabei indo trabalhar com outras coisas e depois voltei pra cá.

Um pouco antes de vir, eu era muito interessado em programação de sites dinâmicos, que eu estava vendo que ia ter mercado, que o pessoal estava querendo aprender. Em todo lugar que você ia, tinha revista, tinha livro. Eu falei “Vamos aprender esse negócio”. Me falaram que era fácil, a parte de HTML é facinho.

Eu aprendi isso, fiquei interessado e depois vim fazer o curso de sistemas de informação.

**Filipe:** Você está no terceiro ano do curso, você já teve bastante matérias de programação?

**Marvin:** Um pouco de java e agora eu fiz banco de dados, tive que programar um pouco em PHP. O pessoal quis usar e eu achei interessante a idéia. Ela funciona muito bem, é interpretada diretamente no browser. Eu achei legal isso aí. E tem a facilidade de conexão com banco de dados também.

**Filipe:** Com diagramas UML, você já teve algum contato? Já chegou a utilizar?

**Marvin:** Quando você aprende banco de dados e java você acaba vendo alguma coisa. Estou confundindo com XML, desculpa. O UML, o que eu vi foi o seguinte: tem um livro de programação orientada a objetos usando Java, que o autor usa o BlueJ e ele faz um

diagrama UML simplificado.

**Filipe:** O protótipo da AWMo também utiliza um digrama UML simplificado. Então você já teve contato, lendo, mas já chegou a construir um desses diagramas?

**Marvin:** Fazer, você faz, porque você faz as setinhas (as linhas né) e as classes e ele vai criando o diagrama. De uma maneira não tão direta, você realmente pega para fazer, pega a setinha... Só que ali é um pouquinho específico né, as caixinhas são classes.

**Filipe:** É exatamente isso. Na sua opinião, qual sua principal dificuldade? pelo fato deles serem diagramas?

**Marvin:** Não, eu acho que o fato deles serem diagramas facilitou. A proposta do livro é justamente isso. Você fala de classes e objetos olhando pra um código. Alí você fala de classes e objetos olhando para diagramas e o diagrama é mais gráfico, assim, é mais intuitivo.

**Filipe:** No caso o fato de você possuir apenas 10% da visão não atrapalhou o seu acesso aos diagramas.

**Marvin:** Nisso especificamente, não. Porque os diagramas são meio grandinhos. A não ser que você tenha que fazer um diagrama muito complexo e talvez tenha que diminuir o diagrama, não sei como funciona isso aí. Eu fiz diagramas simples, bem simplesinhos.

**Filipe:** Uma das coisas que a AWMo visa tratar são os usuário que possuem 0% de visão. Nesse caso, o diagrama é uma barreira bem maior. Como você imagina que seria isso?

**Marvin:** Eu acho que o diagrama é estritamente visual, se a pessoa não vê, ele não aproveita o que tem de mais importante no diagrama. Acho que esse que é o problema.

**Filipe:** No caso essa construção de diagramas da UML, as atividades de modelagem. Você acredita que isso é uma atividade importante no trabalho de desenvolvimento de *software*.

**Marvin:** Eu vi diagramas, não especificamente UML, na modelagem de banco de dados. Eu não consigo imaginar uma pessoa fazer um banco de dados importante, em qualquer sentido, sem a diagramação. Eu acho que não dá pra fazer, a coisa sai tropeçando e depois vai dar um monte de problema. Se você faz um diagrama você enxerga melhor. Então o diagrama é fundamental, a modelagem tem que ser via diagrama. Então ele é importante no desenvolvimento sim.



**Filipe:** O objetivo da AWMo é oferecer tanto a ferramenta visual, onde você consegue desenhar com as caixinhas, a interface textual, para permitir acesso para pessoas que não tem visão nenhuma. Como você imagina que seria uma ferramenta assim? Você acredita que ela ajudaria os usuários?

**Marvin:** Depende de como ela for programada. Porque você tem que passar a informação que o diagrama passa visualmente, você tem que passar textualmente e de uma maneira que a pessoa entenda. Então, o que está subordinado a que no diagrama? Tem que ser explicado, a pessoa tem que conseguir montar o diagrama na cabeça dela pela narração do computador. Se conseguirem fazer isso com uma certa economia, de uma maneira eficiente, eu acho que fica legal sim.

**Filipe:** Esse é o grande desafio, na verdade.

**Marvin:** Isso, porque se o texto for longo, pode ser desgastante. Se você conseguir um texto conciso mas que passe toda a informação e a pessoa consiga entender com uma certa facilidade.

**Filipe:** Pode haver um problema de memória? Se o texto for longo, ao chegar no final o usuário pode não lembrar o que estava escrito no início?

**Marvin:** Sim, essas pessoas são um pouco mais treinadas nisso, mas mesmo assim se o texto for realmente muito grande, a pessoa não se atentar a isso na hora de desenvolver o protótipo, eu acho que corre o risco de fazer algo que possa desafiar a memória das pessoas sem necessidade.

**Filipe:** Se o diagrama começar a crescer?

**Marvin:** Uma coisa que eu aprendi é que em computação nada tem graça se for pequeno. O computador foi feito para resolver problemas gigantescos. Então, todo problema que você for fazer que possa se importante, ele vai ter um certo porte, nunca vai ser uma coisinha tranquila. Então esse diagrama seria grande e a narração dele seria grande e complexa, em algum sentido. Ele tem que ser grande, mas não tanto. Teria que ter uma forma de limitá-lo mas sem perder informação.

**Filipe:** E o problema da leitura sequencial? Quando se está olhando uma imagem é possível navegar em qualquer direção, no sentido que você quiser. Porém o texto ele vai sempre ser lido sequencialmente, você acredita que isso possa influenciar?

**Marvin:** Eu acho que sim, porque vai depender demais da memória da pessoa. Porque se ela tiver uma memória boa e conseguir visualizar dentro da cabeça dele, ele vai poder navegar dentro da cabeça dele, mas isso é uma coisa que acho que a maioria acaba não

conseguindo, ele vai ficar dependente do diagrama escrito. Então se ele perdeu alguma parte, ele vai ter que voltar e ler de novo.

**Filipe:** Teria que ter uma forma fácil para o usuário conseguir navegar no texto?

**Marvin:** Eu acho que o único jeito vai ser ler novamente. Então isso vai ter que ser feito de uma forma otimizada, senão fica complicado. Ele tem que voltar rapidinho e ler pra ir *linkando* as coisas.

**Filipe:** Se o usuário conseguir especificar para qual classe ele quer voltar, ajuda?

**Marvin:** Por exemplo, porque ele não tem como clicar em nada. Como ele está lidando com texto, se ele demorar ou tiver que fazer muito processamento dentro da cabeça dele pra chegar onde ele quer, ele pode acabar esquecendo o principal, que é o que ele tá fazendo. Ele vai consumir a memória dele com informações que o computador deveria fazer, que é a navegação.

## F.2 Pós-entrevista

**Filipe:** Quais foram as principais dificuldades que você encontrou ao utilizar a ferramenta?

**Mário:** Aquela da numeração (Numeração nas linhas do texto do diagrama, pois as mensagens de erro indicam o número da linha que contém o erro), que eu achei que realmente toma um tempo extra. Porque como eu, às vezes, não exergo certinho, eu dependo mais da informação que ele passa, o número da linha. Tem gente que não, vê a linha e se encontra. Mas se eu souber a linha exata, pra mim fica mais fácil. Porque aí eu me concentro naquela linha, eu sei que o erro está ali ou próximo. Então, eu preciso saber exatamente onde é a linha. Pra mim a informação do número de linha é importante. E no caso da pessoa que não enxerga nada e vai navegar ouvindo eu acho que é mais importante ainda porque ela tem que ir exatamente onde está o erro. Quando você tá vendo, você pode ir um pouquinho antes, um pouquinho depois, porque você está vendo uma faixa né. Você coloca o erro mais ou menos no meio da tela e vai olhando em volta até achar. Agora, se você está ouvindo, você tem que ir na linha exata.

**Filipe:** Você acha que ajudaria se na frente do erro houvesse um *link* “ir para linha 78”?

**Marvin:** Por exemplo! Porque ele te dá o erro, aí depois tem o *debugger*...

**Filipe:** Mas sem o *debugger*, apenas para facilitar que se encontre a linha. Como se fosse um *link* de navegação, você clica e ele coloca para você a linha no meio da tela.

**Marvin:** Na verdade, ele vai me dar uma lista de erros, você tem que guardar essa lista. Você vai no primeiro aí clica e ele vai pra lá. Vamos imaginar que nós estamos trabalhando em um código enorme em que a distância entre um erro e outro é grande. A pessoa sempre vai querer voltar na lista de erros para clicar no próximo *link*. Talvez até criando uma janelinha separada só com os erros, para você ir clicando. Sabe? para poder manipular, independente do texto. Você vai clicando nos *links*, aí vai para o texto, acerta o erro e volta nessa janela. Você pode querer recompilar no meio. Quando você acha que tem erros demais e que alguns que você corrigir vai sumir com outros. É interessante sim.

**Filipe:** Mais alguma coisa que você sentiu dificuldade? Tamanho da caixa de texto? Fonte?

**Marvin:** A fonte tem que ter tamanho alterável, junto com o restante do sistema. Acho que uma referência para os números (de linha) é o notepad++. Ele coloca os número de uma maneira que eu acho legal, ele coloca os números pequenos, discretos para não impactar a visão de quem pode ler o código. E a outra coisa é o tamanho do cursor (foi comentado durante o experimento).

**Filipe:** Das funcionalidades que você encontrou na ferramenta, quais você achou mais interessante?

**Marvin:** Nós tínhamos comentado. O contraste eu achei muito legal. Por exemplo, o Dev C++ é branco a tela, fica ruim. Geralmente eu costumo alterar. Tem outros também. Essa cor que você escolheu é muito melhor. Ela cansa menos a vista. Tem gente que gosta do branco, mas no meu caso por exemplo, eu preciso enxergar a letra.

**Filipe:** No caso do contraste que eu citei é ... (Demonstrando o uso na própria ferramenta).

**Marvin:** Na verdade pra mim ficou pior porque o fundo do conteúdo ficou mais claro. O outro modo é melhor. Aqui não é branco né? Que cor que é?

**Filipe:** É branco mesmo, talvez o cinza em volta torne menos cansativo.

**Marvin:** É eu acho que o cinza ajuda. Mas poderia não ser branco, uma cor bem clara mas não branco. Talvez um verde ou até cinza mesmo. Acho que o cinza é melhor. Mas bem clarinho pra não ser branco, porque o branco cansa bastante a vista e não pode ser escuro também se não fica ruim pra ver.

**Filipe:** Você mudaria algo na ferramenta utilizada?

**Marvin:** Só as coisas que realmente fizeram falta...

**Filipe:** Remover alguma coisa?

**Marvin:** Por exemplo, a ordem do menu está legal, o salvar tá ali. Não tem nenhum botão perigoso perto do outro. Eu estava fazendo curso de HCI e me perguntaram se eu conhecia algum caso pavoroso de usabilidade? Eu falei: Um que eu conheço é do site do ICMC, quando você vai mandar um email para um professor, tem o botão salvar e pagar do lado, encostado quase. Se você marcar, você aperta o botão errado.

**Filipe:** Se esse protótipo da AWMo, fosse divulgada como um projeto de código aberto (Por exemplo, no GitHub ou SourceForge), você se interessaria em se tornar um colaborador do projeto? Você acredita que tem futuro uma ferramenta desse tipo?

**Marvin:** Sim sim, acho interessante e acho que muita gente ia gostar, por exemplo, todo mundo que programa um pouco mas não consegue desenvolver um projeto maior por que não consegue desenvolver um diagrama. “Mas agora tem uma ferramenta lá em teste que você pode digitar o texto e usar o narrador pra você saber o que você está fazendo”. Eu tenho certeza que a pessoa vai se interessar. É difícil medir isso, quantas pessoas precisam de uma ferramenta dessa? São poucas, e dessas pessoas que precisam menos ainda vão se interessar. Pode ser que você coloque lá e fique meio frustrado com o número de pessoas.

**Filipe:** Você tem mais alguma coisa pra acrescentar? Algo que queira falar? O que você achou do projeto?

**Marvin:** O projeto em si é interessante, você praticamente coloca algumas pessoas que não iam conseguir fazer um diagrama, pra fazer.

**Filipe:** Para você que consegue enxergar os diagramas, entre a forma textual e a forma gráfica, você preferiria usar qual?

**Marvin:** Rapaz, depende do humor. Sei lá tem dia que eu acho que ia usar a gráfica numa boa. Ter a opção é muito importante. No caso de quem enxerga pouco, pode usar as duas e no caso de quem não enxerga nada, ele pode usar a textual. Então, quer dizer, você deu uma opção a mais e que, para pessoas que realmente não enxergam, é a única opção, é super válido.

**Filipe:** Muito obrigado pela sua participação.

---

# Transcrições das entrevistas - Arthur

---

## G.1 Pré-entrevista

**Filipe:** Quantos anos você tem?

**Arthur:** Eu estou com 30 anos agora.

**Filipe:** Você possui uma deficiência visual? Qual o grau?

**Arthur:** A minha deficiência visual é completa né. Cegueira bilateral. Então eu enxergo pouquíssimo, só vulto e claridade mas nada que possa me auxiliar a ter uma visão mais clara de alguma coisa.

**Filipe:** E essa deficiência é de nascença ou foi adquirida?

**Arthur:** Eu perdi a visão com 15 anos mais ou menos, foi em decorrência de uma doença craniana que eu tive.

**Filipe:** Você utilizar alguma configuração especial para facilitar seu uso do computador?

**Arthur:** Leitores de tela, atualmente eu estou utilizando mais frequentemente o NVDA, mas também já utilizei o 'JAWS for windows' e utilizo quando necessário.

**Filipe:** Você prefere o NVDA?

**Arthur:** Sim, os dois são boas ferramentas, mas atualmente estou usando mais o NVDA, até por ele ser gratuito. Como estou utilizando mais o windows 8 agora, eu ob-

servo que o NVDA está melhor do que o JAWS em alguns aspectos.

**Filipe:** Qual sua experiência com uso de computadores? A quanto tempo você usa?

**Arthur:** Acho que a uns 7 anos mais ou menos. A questão do nível de conhecimento, como eu trabalho na área, eu acredito que tenho um conhecimento um pouco maior.

**Filipe:** Qual sua experiência com programação de computadores?

**Arthur:** Acho que comecei a lidar mais com programação fazem uns 5 anos, talvez um pouco mais. Eu considero que eu tenho conhecimentos de intermediário para avançado em Java, C# e JavaScript. A parte de desenvolvimento Web, então HTML e tecnologias relacionadas. Eu considero que tenho conhecimento de intermediário para avançado em programação atualmente.

**Filipe:** Você já teve contato com diagramas da UML? Diagramas de classe ou outros diagramas da UML?

**Arthur:** Já, já tive contato através de descrições textuais da imagem do diagrama. Basicamente foi esse contato, a partir de descrições da figura do diagrama.

**Filipe:** Como foi essa experiência? Como é para você ler ou entender esses diagramas?

**Arthur:** Através dessas descrições, se o diagrama for complexo, fica complicado você ter uma idéia clara. Eu consigo ter uma noção mas não considero que seja suficiente para eu ter uma idéia clara de como é o diagrama. A questão de você descrever uma imagem, ela ajuda em alguns aspectos, mas em diagramas mais complexos fica difícil, fica muito confuso você ter uma noção clara do que está ali.

**Filipe:** Sobre essas descrições de diagramas que você já usou, ela tenta descrever a posição no espaço dos itens do diagrama ou possui mais o conteúdo?

**Arthur:** O conteúdo e algumas informações sobre a posição também: a relação da classe com outra, como está estruturado o diagrama. Basicamente isso.

**Filipe:** Na sua opinião, qual o principal problema nesse tipo de acesso? O que faz com que isso não seja tão bom?

**Arthur:** Primeiro, a falta de praticidade. Porque se você está em uma situação real, onde você está dentro de um projeto e você precisa trabalhar com pessoas que enxergam, por exemplo, você precisa de alguém que descreva para você e isso já é uma perda de tempo em determinados aspectos. Fora que você não consegue expressar o diagrama de uma forma que uma outra pessoa que enxerga... Você não consegue se comunicar com

a pessoa através do diagrama, por exemplo, eu posso criar uma descrição de como eu entendo o diagrama, só que para uma pessoa que está acostumada a observar a representação gráfica ler aquilo, não vai ficar muito claro pra ela entender. Então esse é um aspecto, e a questão da leitura também, basicamente eu preciso de alguém que descreva para mim. Eu não consigo pegar o diagrama feito por uma outra pessoa e entender aquilo de uma maneira independente.

**Filipe:** Você acredita que o acesso a essas atividades de modelagem são importantes para a realização do seu trabalho?

**Arthur:** Olha, eu acho que se for uma decisão de utilizar, pela equipe de projeto, seria importante. Agora, eu sei que existem meios alternativos de eu entender uma modelagem de classes sem utilizar a UML. Agora, logicamente que se houvesse uma forma acessível de eu realizar essa atividade, é claro que iria auxiliar em vários aspectos, principalmente na comunicação dentro do projeto.

**Filipe:** Quanto a modelagem em si, você acredita que é uma atividade importante no processo de desenvolvimento de *software*?

**Arthur:** Sim, eu acho que é uma atividade importante, pois se você não começar pela modelagem fica complicado você chegar na implementação e fazer de uma forma em que todos possam compartilhar do entendimento da estrutura daquele conjunto de classes.

**Filipe:** O que você espera de uma ferramenta que busca o acesso para edição de diagramas por meio de uma interface textual?

**Arthur:** Tendo contato com a linguagem e com a proposta, eu achei bem interessante. Eu já tinha tido contado com outras metodologias que tentam trabalhar com a UML de uma forma mais acessível, por exemplo, metodologias que utilizam planilhas do Excel de uma forma alternativa para expressar os atributos, métodos e relacionamento entre as classes. Outras com alto relevo. Porém não é uma forma muito prática, como eu havia colocado e não é uma forma que possa ser universal, que seja compartilhada tanto por pessoas que sejam deficientes visuais quanto pessoas que não sejam. Então eu achei a proposta bastante interessante, a idéia de descrever de uma forma textual, meio que uma DSL né? ... bem interessante porque se aproxima bastante da própria estrutura das classes, como as classes se estruturam e ao mesmo tempo é acessível, é a forma mais acessível, que traz maior autonomia para as pessoas cegas, realmente é através de descrições textuais.

## G.2 Pós-entrevista

**Filipe:** Ao utilizar a AWMo, quais foram as principais dificuldades que você encontrou?

**Arthur:** Com relação à ferramenta nenhuma. Não achei nada que fosse difícil de trabalhar, achei a interface simples, acessível, intuitiva. Só sinalizei um problema, que não chega a impedir a utilização que são os botões do menu do topo, não estão rotulados, a função deles está indicada só pelo atributo *title* e quando você está navegando no modo de leitura da página, o leitor de tela não detecta a funcionalidade do botão, você precisa navegar pelo foco. Mas, tirando isso que também não é nenhum problema que impeça a utilização, o resto eu achei totalmente acessível e tranquilo de utilizar, mesmo o *feedback* com relação aos erros de sintaxe da linguagem, são simples de você encontrar na página.

**Filipe:** O problema foi o teclado? (brincando)

**Arthur:** É, o que me atrapalhou um pouquinho foi o teclado, mas é falta de costume com o computador.

**Filipe:** Quais funcionalidades que você achou mais interessantes? em ordem de preferência.

**Arthur:** Eu gostei bastante da parte de edição do diagrama, do código da linguagem do diagrama, eu achei bastante interessante e achei legal também a questão do retorno dos erros de compilação, não sei se pode ser chamado de compilação, mas a interpretação da linguagem. Eles ficam posicionados de uma forma que facilita a localização ele está logo antes do campo de edição de texto, simples de encontrar. A parte de abrir o diagrama, a tabela que lista os diagramas eu achei interessante também, tá organizado e está simples de encontrar os controles.

**Filipe:** Eu senti que você teve um pouco de dificuldade para encontrar a ação para abrir o diagrama, está só escrito Textual.

**Arthur:** Isso, exatamente talvez ficasse mais lógico se colocasse “abrir”.

**Filipe:** Seria “abrir textual” e “abrir gráfico”?

**Arthur:** É tem como abrir o modo gráfico também, acho que ficaria melhor assim, ficaria mais claro.

**Filipe:** Teve algo que você sentiu falta? Que você acha que poderia existir a mais?



**Arthur:** Olha, em geral, a página está bem estruturada, as seções estão marcadas com cabeçalho, isso ajuda bastante a encontrar de forma rápida. Esse ponto que você citou de identificar a funcionalidade de uma forma um pouco mais lógica ajudaria mais, pra você não ter dúvida na hora de como se encontra o recurso. Aquela parte dos botões acho que também seria importante melhorar porque à primeira vista quando você está lendo a página de uma forma linear, dá impressão que os botões não estão identificados. Depois que você navega com o foco você consegue descobrir. Mas basicamente é isso.

**Filipe:** Em relação as mensagens de erro que você comentou, lá é impresso também o número da linha. Essa informação acaba não te ajudando?

**Arthur:** Então, o leitor de telas não consegue ler, ele não está pegando o número da linha. Ajudaria no sentido de eu ir contando.

**Filipe:** No caso de um número grande de linhas, você prefere contar até lá ou ouvir o conteúdo da linha pra se localizar?

**Arthur:** Depende, por exemplo, em IDEs quando eles tem essa informação, a própria IDE te dá a opção de você clicar no erro e ele já te coloca na linha onde ele está. Aqui no caso, como ele não lê o número da linha eu não tinha essa referência, então na realidade o que eu observei era mensagem de erro, o conteúdo dela e aí eu sabia exatamente onde estava o ponto que precisava corrigir.

**Filipe:** Você mudaria alguma coisa, o que você acha que poderia ser melhorado?

**Arthur:** Olha, honestamente, observando a ferramenta, a dificuldade que você tem é a normal que você tem em qualquer página que é o fato de você não conhecer a página. Você precisa primeiro saber onde estão as coisas, estudar primeiro a página para você ter uma representação, ter claro na sua mente como as coisas estão dispostas para você encontrá-las mas rapidamente mas é o contato inicial. Após isso você já consegue se localizar e fica tranquilo de utilizar. Não vejo nada que devesse melhorar, mais aqueles pontos dos botões e *link* de abrir que são mudanças básicas e simples de fazer, mas de forma geral eu fiquei satisfeito com a interface.

**Filipe:** Se a AWMo for divulgada como um projeto de código aberto, você, como um programador, teria interesse em se tornar um colaborador do projeto?

**Arthur:** Eu me interessaria, achei um projeto bacana de utilizar e de colaborar.

**Filipe:** Tem mais alguma coisa que você gostaria de comentar?

**Arthur:** Parabéns pela iniciativa, eu acho bastante valoroso isso né, algo que vai auxiliar mesmo nessa questão do aprendizado, do trabalho com UML, do trabalho em equipe de desenvolvimento eu acho bem interessante.

**Filipe:** Você acredita que seria prático utilizar uma ferramenta assim no dia-a-dia para trabalhar com desenvolvimento de *software*?

**Arthur:** Eu acho que sim, por exemplo, se eu me encontrasse numa equipe de desenvolvimento e fosse trabalhar maciçamente com UML e os demais desenvolvedores se interessassem em aprender a linguagem, que é uma coisa simples e rápida de se fazer, eu acho que seria interessante sim.

**Filipe:** Nesse caso, outros membros da equipe poderiam usar a visão gráfica também, não precisariam necessariamente aprender a linguagem textual.

**Arthur:** Exatamente, esse é o ponto.

**Filipe:** Muito obrigado pela sua participação, é muito importante para a continuidade e validação desse projeto.

---

# Transcrições das entrevistas - Ford

---

## H.1 Pré-entrevista

**Filipe:** Primeiro, para conhecer mais sobre você, quantos anos você tem?

**Ford:** 35 anos.

**Filipe:** Você possui uma deficiência visual? Qual?

**Ford:** Eu tenho glaucoma congênito, então eu nasci cego. Fiz uma cirurgia com 20 dias de nascido que me permitiu ter 30% da visão, então eu fui baixa visão durante muito tempo da minha vida mas a 5 anos eu tive uma crise que me deixou praticamente cego. Eu consigo reconhecer luz, alguns objetos assim, mas nada com um forma muito definida então, clinicamente hoje eu sou considerado cego mesmo, apesar de ter esse pequeno resquício.

**Filipe:** Você utiliza alguma configuração especial para o uso de computadores?

**Ford:** Eu uso o leitor de tela e a tela com alto contraste, porque com alto contraste ainda e muito próximo do monitor com letras ampliadas eu ainda consigo ler. Como eu fui alfabetizado né, em tinta que a gente chama, eu ainda consigo ler. Mas eu diria pra você que hoje 99% do tempo é só leitor de tela. O Mac, por exemplo, ele não tem alto contraste, ele faz inversão de cores, que é muito ruim. Então, no Mac eu me viro 100% com o leitor. No Windows, por usar muito tempo eu ainda uso de vez em quando o alto contraste. E tem aquele negócio, as vezes o NVDA não tá conseguindo ler uma janela e eu me aproximando eu consigo ver que tem uma janela ali e tem um campo de texto ou

alguma coisa assim eu consigo fazer, mas é muito raro, muito raro mesmo.

**Filipe:** Você falou do alto contraste e da inversão de cores, qual a diferença entre as duas técnicas?

**Ford:** O alto contraste, vou te mostrar. Você nota que o fundo está preto, a caixinha de texto está com contorno branco mas a foto continua com as cores originais. Se fosse inversão de tela, ele inverteria tudo, essa imagem estaria com o fundo preto e as cores de dentro teriam virado um laranja. Então, o alto contraste aplica um padrão de cores que realmente dá um grande contraste, no meu caso eu trabalho com preto e branco e preto e amarelo que eu consigo enxergar melhor, mas conheço pessoas que, por exemplo, o fundo tem que estar laranja, ou azul. Existem algumas combinações e um grau de diferença entre essas cores.

**Filipe:** A ferramenta possui uma opção de alto contraste também. No caso trabalhamos com as cores preto e branco.

**Ford:** É o mais comum. A maioria das pessoas vão trabalhar com fundo preto e com letras claras.

**Filipe:** Qual a sua experiência com o uso de computadores? A quanto tempo você utiliza?

**Ford:** Cara, eu mexo no computador desde os 9 anos de idade. Eu ganhei um curso de informática na época para aprender a programar em MS Logo, aquela linguagem da tartaruga lá pra você fazer desenhos na tela. Só que, como eu sempre fui devorador de livros, quando eu enxergava, eu ganhei os livrinhos do curso e não ficava só no que era passado na aula. Eu chegava em casa, pegava o livrinho e ficava folheando e chegava na próxima aula todo empolgado pra testar aquilo que eu tinha lido e foi aí que eu aprendi a programar né. Com 9 anos eu posso dizer que já programava. Fazia algoritmos de geração de números “randômicos”, joguinhos de perguntas e respostas. Comecei com isso e daí pra frente eu nunca mais parei de mexer com computador, eu ganhei um TK85, depois ganhei meu primeiro PC

**Filipe:** Quanto à experiência com programação, você programa desde os 9 anos e já trabalhou com que linguagens?

**Ford:** Assim, no PC eu trabalhei com esse MS Logo e no TK85 eu programei em Basic que vinha em ROM. No PC eu passei pelo dBase, pelo Clipper, pelo VB3, pelo Delphi, todas as versões. Depois voltei a mexer com VB6, mexi com ASP, com PHP, Java e hoje eu estou focado em .NET. Já brinquei com algumas outras coisas como ColdFusion, tentei

estudar um pouquinho de python recentemente e estou sempre vendo coisinhas diferentes né, mexendo com um pouquinho de tudo. É que eu sempre brinquei né, tem essa briga de Java e .NET mas cara, você pode bater um prego com um martelo ou com uma chave de fenda, a diferença é o trabalho que vai dar com cada uma das duas, depende o que você vai fazer.

**Filipe:** Você já teve contato com diagramas da UML?

**Ford:** Já sim, nas graduações que eu fiz a gente tinha aulas de UML lá, que eu enxergava então eu ainda lembro dos modelos visualmente como é que é.

**Filipe:** Então você aprendeu UML tradicionalmente, com visão.

**Ford:** Sim, mas profissionalmente eu nunca cheguei a usar muito. Nas empresas que eu trabalhei, teve uma que trabalhava ainda na época dos DFDs (*Data Flow Diagrams*) e a outra que usava, na verdade para documentar mesmo alguma coisa do que realmente para passar o trabalho para o desenvolvedor né. Então, tinha a equipe de análise lá que ia lá e fazia todos os diagramas, chegava pra gente e entrega os casos de uso e falava faz isso.

**Filipe:** Então depois que você perdeu a visão, você praticamente não utiliza mais?

**Ford:** Não, não usei mais.

**Filipe:** Você chegou a ter contato com alguma tecnologia assistiva para UML?

**Ford:** Não, tecnologia assistiva não. Assim, sempre tentei imaginar alguma coisa assim, inclusive a sua ideia foi sensacional, porque foi até uma coisa que estava comentando um amigo. Foi aí que rolou o papo, ele falou: caramba cara, o Filipe falou que tinha tentado entrar em contato contigo, será que ele conseguiu e tal? Eu falei: ele não conseguiu entrar em contato comigo, você pode passar meu contato pra ele e aí ele te passou. A gente estava comentando exatamente isso né, pra deficientes seria muito mais fácil uma linguagem declarativa e que fizesse essa conversão, por que aí os videntes poderiam ler os modelos graficamente como eles leem hoje e a gente teria essa facilidade porque a universidade do Paraná, teve uma aluna lá que fez um projeto sobre como ensinar UML para um deficiente visual a partir do uso de tabelas. Cara, bacana, é super bom para o lado educacional, o deficiente vai entender mas quando ele sair para o mercado, das duas uma: ou ele vai ter que perder um tempão com alguém da equipe dele falando, cara, olha é isso, e o cara vai documentando na tabela e aí se depois tiver uma mudança de projeto, ele vai ter que ter esse trabalho ir lá e atualizar, sendo que com uma linguagem declarativa como é o caso do seu projeto, se você consegue fazer uma tradução bidirecional. Se alguém for lá na ferramenta e alterar o diagrama, automaticamente atualiza a linguagem e alterei

a linguagem, já altera o diagrama. Então, quer dizer, tem uma interação muito maior e muito mais rápida né.

**Filipe:** Você já usou alguma outra linguagem de modelagem? MER, Fluxogramas, máquinas de estado?

**Ford:** Sim, acabei vendo o MER, DFDs para os fluxos de dados, fluxogramas, diagramas de estado, diagramas de deploy, de componentes.

**Filipe:** Mas esse contato ocorreram quando você ainda tinha visão?

**Ford:** Sem visão eu nunca quis mexer com mais nada porque nenhuma ferramenta hoje realmente é acessível né. Em algumas ferramentas você até consegue jogar os componentes na tela, mas fica tudo largado, tudo jogado e você não consegue ler aquilo de volta. Não é fácil fazer isso. Uma brincadeira que a gente fez aqui dentro, com a ferramenta PowerDesigner da SyBase. E esse PowerDesigner tem uma integração via VBScript, via VBA, que permite que você interaja com os modelos. Então um amigo precisava fazer uma documentação de alguns diagramas e ele montou uma macro que pegava o diagrama e todo o metadado desse diagrama, jogava numa tabela excel pra ele poder trabalhar mais fácil e aí depois ele salvava esse excel e aí conseguia dentro do PowerDesigner importar de novo. Só que entra novamente naquele caso, beleza pra gente fazer uma coisa rápida dessa é bacana, funciona super bem. O que eu vejo dificuldade é como você encaixa isso num processo do dia a dia. Numa reunião, por exemplo, com a sua linguagem descritiva lá, eu poderia estar numa reunião conversando com as pessoas e se a gente definir que precisa mudar alguma coisa, conforme a pessoa tá fazendo ali eu vou pegando com o NVDA, vou lá edito, mudo e salvo, beleza, eu estou tendo o entendimento da mesma coisa que todo mundo está vendo ali num papel riscado ali no meio da reunião.

**Filipe:** Na sua opinião, qual seria o principal problema de acesso a esses diagramas para deficientes visuais? Não só a UML mas para outros também.

**Ford:** Acho que o maior problema realmente é isso né, assim, as tecnologias assistivas hoje ainda não conseguem chegar num ponto de transformar realmente qualquer elemento da tela em algo interativo. Então se o cara que desenvolveu lá o Visio tivesse pensado que cada um dos componentes que vai no formulário tem que ter foco, tem que ter interação com o teclado, tem que ter ali um *label* que descreva o que é aquilo para que o NVDA, o JAWS ou qualquer outro leitor consiga interagir né, a gente não consegue. Uma coisa que eu gostaria muito de usar, por exemplo, e não consigo porque não tem nenhuma ferramenta acessível e aí eu tenho que usar um outro formato que é o *outliner*, mas é pra fazer mapas mentais. Não tem nenhuma ferramenta que é acessível, e assim, você

vê as pessoas conversando pelas ferramentas, você percebe que é um negócio fantástico pra você modelar o seu raciocínio e que é super simples de fazer, porque você tem uma caixinha no meio e conforme você vai tocando no sinalzinho de mais ele vai criando mais caixinhas a partir daquele ponto e ramificando e fazendo o *layout* e depois navegar nessa idéias. Bacana, não consigo, aí eu tenho que pegar um *outliner* e fazer essa estrutura de tópicos mesmo né ou então né, mal de programador e até um truque eu uso pra fazer apresentação no PowerPoint, não sei se você conhece. Você pega e cria um arquivo txt e aí você faz o seguinte, tudo que tiver na margem ele considera como um novo slide e você vai fazendo indentação com o Tab para cada nível de tópico que você quer fazer. Então o título do slide está colado na margem esquerda, você dando um Tab ele coloca um *bullet* no primeiro nível, aí você dá dois Tab ele coloca um *bullet* no segundo nível. Então, quando eu quero fazer uma apresentação de PowerPoint eu organizo meu pensamento assim, eu vou colocando o título das palestras, por exemplo eu vou palestrar agora dia 18 em um evento. Então eu coloquei um título para cada slide aí eu dou um Enter e aí dou um Tab, aí eu coloco o primeiro tópico aí dou um Enter, Tab e coloco o segundo tópico, dou Enter e coloco o terceiro tópico. Se tem um sub-tópico eu dou um Enter e aí dois Tabs e coloco o sub-tópico. Depois, no PowerPoint eu consigo importar essa estrutura de tópicos e ele gera pra mim todos os slides. É um truque que não foi criado para acessibilidade né, a Microsoft criou isso aí pra outra coisa, mas que a gente acabou descobrindo e facilita muito quando eu preciso fazer uma apresentação. Então quando eu vou fazer um *mindmap* eu acabo fazendo isso também. Tem ferramenta que importa essa estrutura. Mas não é o ideal.

**Filipe:** Você acredita que as atividades de modelagem, tanto de UML quando de banco de dados, por exemplo, são importantes para a realização do trabalho de um desenvolvedor de *software*.

**Ford:** Cara, eu acho que assim, é um meio de comunicação legal, ele é mais fácil de comunicar, principalmente com pessoas não técnicas e eu acho que ele é bem bacana para parte de comunicação, eu não desmereço esse tipo de trabalho, acho que ele é importante sim. A única crítica que eu faço, é que eu não conheço nenhum lugar no mundo que mantêm isso realmente atualizado né. Assim, eu acho que a solução ideal desse problema, e talvez até pode ser como dica para evolução do seu projeto, seria isso estar integrado nas IDEs de forma que as mudanças em documentação gerem mudanças em código e mudanças em código atualizem a documentação. Porque, se eu pegar hoje qualquer sistema aqui, sem medo de errar, e pegar qualquer documentação não reflete o código-fonte. Então assim, eu acho que até a própria definição da UML coloca muito disso, que ele não é nem um documento técnico né, ele é mais um documento de comunicação da equipe e

realmente, pra quem enxerga, e eu lembro muito disso, você bate o olho em um documento de classe e você entende o sistema na hora. Se está bem modelado e se tá todos papéis ali bem definidos, você bate o olho e fala “ah é aqui que eu tenho que mexer”. Ou então, tem aquela matriz de causa de causa e efeito né, você sabe que se você mexeu naquela classe, pelas setinhas estão entrando e saindo delas, você já sabe, tá se eu mexi aqui eu vou ter problema aqui, aqui e aqui e isso as vezes no código-fonte, é difícil de mapear. Ontem mesmo eu estava fazendo um projeto em casa e tava codificando lá e vi que o nome de uma classe não tava legal e, tudo bem, o Visual Studio tem uma ferramenta de *refactor* lá, eu vou lá e mando fazer um *refactor* e dar *rename*, ele vai trocar me todos os lugares que aquilo é referenciado. Mas se você pensar que você está usando uma IDE que não tem esse recurso, cara você altera o nome de uma propriedade e na hora que você der F5 pra compilar é um abraço, vai aparecer as vezes um milhão de erros porque você mudou uma propriedade. Então, eu acredito que seja sim muito importante, mas eu só ressalvo esse ponto de que a evolução tem que vir junto com isso, de como eu consigo colocar isso no processo de modo que se torne uma coisa mais automática mesmo.

**Filipe:** Seria algo mais para o lado de desenvolvimento orientado a modelos? Do modelo você transforma em código e vice-versa.

**Ford:** Sim, sim. Aí sim você tem realmente um diagrama que está sempre atualizado e porque, assim é um problema hoje, não é que eu não confie no modelo, eu não confio na documentação que a gente tem hoje porque ninguém atualizou.

**Filipe:** O que você espera de uma ferramenta que busca o acesso para edição de diagramas por meio de uma interface textual?

**Ford:** Cara, eu espero ver realmente uma coisa legal nesse sentido, de que eu vou conseguir ali, de forma declarativa, escrever um modelo né, pensar num sistema e escrever o modelo e que enquanto eu consigo falar pra você: olha, eu acho que a propriedade valor aqui que declarou como *float*, ela tem que ser obrigatória. E aí a pessoa que tá conversando comigo, tá vendo ali do lado o modelo e batendo o olho na classe, ele fala, a tá, ele tá falando dessa classe e dessa propriedade né. Não aquele negócio abstrato ao extremo, de eu estar conversando com você sobre o modelo e eu estar imaginando uma coisa e você imaginando outra totalmente diferente, porque a gente tá em realidades diferentes. Eu tenho esse *background* de conhecer os modelos mas eu imagino para pessoas que nasceram cegas, assim, eu na pós-graduação, quando o professor fala de UML, como eu tenho memória visual, eu lembrava o que era, mas assim, me colocando no lugar de uma pessoa que nunca enxergou eu falava cara, o cara ia sair dessa aula xingando tanto, mas xingando tanto, porque o cara falava porque isso aqui, na classe tal e classe representada aqui por



esse objeto. A UML define lá no metamodelo que classe tem que ser representada por esse símbolo aqui e pessoa assim: mas que símbolo? Do que que você tá falando? A aula não está preparada pra esse tipo de coisa. Tinha pontos que eram mais didáticos, do tipo: olha tem que lembrar que a composição é representada por um losango preenchido. Isso dá pra entender, legal, mas e a organização disso? E aí, tudo bem, tem gente que tenta fazer, acho muito legal isso, novamente como projeto educacional eu acho bacana, tem professor que tem o trabalho de pegar aquela caneta que você escreve com ela e fica em alto-relevo. Então vai lá, pega uma cartolina gigantesca e desenha todos os modelos para o deficiente ir passando a mão e vendo como é que é o formato das coisas. Mas a gente tem que lembrar que no mundo acadêmico é tudo muito bonito né. Mas quando a gente vem aqui pra fora, o pessoal não vai ter tempo num projeto, imagina na empresa, tem um projeto pra semana que vem e o cara te pede, por exemplo, o mapa do *site*. Você não vai ter tempo pra chegar pro cara e falar: olha tá aqui eu peguei uma cartolina e desenhei a estrutura do site em relevo pra você, vai usando isso aqui pra te guiar nas páginas. Não, infelizmente, a gente vai ter que chegar e decorar a estrutura de diretório ou alguma coisa assim, porque cada dia mais é muito dinâmico, o cliente te pede uma coisa hoje e amanhã ele já quer uma coisa diferente. Então como é que você dinamiza isso né? A tua ferramenta é realmente um primeiro passo, eu sei que é um protótipo, mas é um primeiro passo realmente pra gente chegar num ponto em que você vai poder chegar numa equipe e poder falar: a qual ferramenta você usa? Ah eu uso a ferramenta X, que... ah legal, vamos discutir esse modelo então e realmente chegar de igual pra igual, isso, isso aquilo. E aí o cara chegar e falar, vamos implementar um *strategy* nisso daí. Aí como o deficiente já conhece o *strategy*, na hora ele já consegue escrever no modelo seguindo essa convenção de padrão de projeto e tudo mais né.

## H.2 Pós-entrevista

**Filipe:** Quais foram as principais dificuldades que você encontrou?

**Ford:** Cara, na verdade, a dificuldade foi lembrar a sintaxe, mas foi a primeira vez que eu usei, então com certeza usando isso no dia a dia se torna um pouco mais fácil. E as mensagens de erro, elas poderiam conter ali a mais, além da linha, que eu acho que facilita pra você chegar na linha e tudo mais. Seria alguma coisa como, por exemplo, a própria linha mesmo como algumas linguagens fazem na compilação, ela colocar ali um caractere do lado da onde ele acha que está o problema né. Ou então, pelo menos a coluna, do tipo: deu problema na linha 11, coluna 5. Porque, quando ele tava lendo rápido, quando ele lia método pra mim, o cérebro tem essa mania de querer adivinhar as coisas,

então existe até leitura dinâmica com leitor de tela e quando ele falava método apesar de estar errado eu entendia certo. Foi aí que eu soletrei pra ver o que é que tava acontecendo.

**Filipe:** Aconteceu com o “Boolean” também né? Estava “Booleanl”.

**Ford:** Isso, o booleano ficou com um L no final, então quer dizer, até pra ele falar mesmo né... e outra coisa a sintaxe do NVDA também não é a melhor coisa do universo, então a gente acaba aceitando algumas coisas, que depois passam a ser normais pra você, mas que não são tão certas. Mas fora isso, achei bem legal. Uma coisa que me surpreendeu positivamente foi o fato dele validar a sintaxe ali e te mostrar os erros, isso eu achei muito bom, porque normalmente o que a gente faz é isso. A gente tá tão acostumando né, eu no .NET é assim: eu tô lá codificando, codifico, codifico, codifico. Bom beleza, terminei a funcionalidade, deixa eu dar um F5 agora pra rodar, pra fazer o teste funcional dela né. E aí sim que vão aparecer os erros de compilação pra mim e que eu vou corrigir. Então, tendo isso facilita pra caramba, porque você pode escrever o diagrama todo e no final ir validando e refinando né.

**Filipe:** Qual funcionalidade da ferramenta você achou mais interessante?

**Ford:** Eu acho assim, alguns pontos positivos: por você ter feito ela Web, ficou muito legal porque roda em qualquer lugar né. Isso é um diferencial bacana e o ambiente Web por si só ele já é um ambiente muito mais acessível. Se você tivesse feito, por exemplo quando você me falou que era em Java eu já tava imaginando a interface Swing, que é um inferno pra se configurar acessibilidade pra aquilo, não fica bom de jeito nenhum, todo ano eu sofro com aquele imposto de renda. Então, por ter feito ele em Web, você já começou com uma plataforma bem bacana e que estende muito até o próprio uso, porque, dentro de uma empresa, por exemplo, você pode ganhar algumas vantagens, como ter a instalação disso em um único servidor na empresa e a empresa inteira usar, que vai facilitar muito. Ou então mesmo, hoje em dia né tá muito comum ir pra nuvem. Mas pensando no lado da acessibilidade é muito legal porque o ambiente, ele é bem acessível.

Talvez para alguns pontos de melhoria, não sei se seria a próxima questão. Seria usar um pouco mais os atributos de ARIA né, do WAI-ARIA, pra identificar um pouco melhor por exemplo a área de mensagens de erro. Talvez controlar um pouco melhor o foco, então quando eu entro, já entro com o foco em determinado controle. Mas, como é a primeira vez que eu tô usando também, é tudo novo. Depois que aquilo vai entrando no dia-a-dia, a gente vai com teclas de atalho e vai localizando a forma mais fácil de chegar nos itens né.

**Filipe:** Seguindo a mesma linha, o que você mais sentiu falta enquanto estava usando a ferramenta?

**Ford:** Olha, eu não sei como você vai fazer isso na Web, mas também não precisa. Mas o que mais eu senti falta na hora, era realmente a sintaxe. Agora falando com você me veio a ideia, o que você podia ter ali pra talvez facilitar um pouco seria, como a gente tem nos editores, aquela barra de ferramentas em cima, em que você já tivesse os elementos ali prontos e se eu desse um “Enter”, por exemplo, no Composição, ele já jogasse pra mim o texto da composição ali dentro e eu só precisasse trocar os elementos.

**Filipe:** Seria um atalho para você utilizar as estruturas da linguagem?

**Ford:** Exatamente. Porque a maior dificuldade que a pessoa vai sentir sinceramente, se é uma pessoa que usa a Web, é lembrar da sua sintaxe. Como eu disse novamente, depois de dois ou três diagramas que a pessoa fizer, isso vai tá tão natural pra ela que ela não vai sentir diferença, mas acho que num primeiro momento você ter ali os atalhos já ajudaria, já seria um facilitador pelo menos, pra ao invés dele ter que recorrer ao manual ele dá um “enterzinho” ali e já insere.

**Filipe:** E você mudaria alguma coisa na aplicação ou na linguagem textual?

**Ford:** Cara, na linguagem eu achei a estrutura bem bacana, eu achei que ficou bem legal. Porque ela segue uma linha de linguagem orientada a objetos mesmo, então por exemplo, para alguém que desenvolve com .NET hoje, com C# ou Java, PHP ou com qualquer coisa que hoje em dia tem alguma orientação a objeto, tá bem legal mesmo, tá bem definido. E na ferramenta acho que seria o que já comentei mesmo: alguns facilitadores para a pessoa interagir mais com a sintaxe num primeiro momento, porque depois também vai ficar tão natural que ela vai indo e, por exemplo, Web, pelo menos eu nunca vi, se alguém fez isso o cara é mais maluco do que eu imaginava, seria fazer uns auto-completes que a gente tem nas IDEs hoje mas isso pra Web eu acho que num...

**Filipe:** Se a AWMo for divulgada como um projeto de código aberto, você, como um programador, teria interesse em se tornar um colaborador do projeto?

**Ford:** Com toda certeza, com toda certeza. E acho que faltam iniciativas desse tipo e toda a iniciativa que vem tem que ser muito apoiada porque é tão difícil olharem pra gente né, que pô, você tem um trabalho de pesquisa, um trabalho de desenvolvimento. Um negócio desse *open-source* eu seria com certeza um dos colaboradores, isso aí você pode contar mesmo.

**Filipe:** Fico muito feliz, inclusive se você puder divulgar para o restante da comunidade. Para gente é difícil achar o contato de vocês.

**Ford:** Olha, eu te recomendo, é que são 1 milhão de e-mails, mas se cadastra e muda sua assinatura para Web, no google groups: cegos\_programadores. Tem hoje 150 membros cadastrados e o número cresce toda semana. Toda semana aparecem um ou dois malucos novos que entram num curso de graduação em sistemas de informação ou ciência da computação e fala: “caraca, tô ferrado. Quero fazer esse curso e tô perdido aqui, tô voando na aula de UML”. Cara, eu já sei até o que eu vou falar: “Cara, passa isso aqui para o seu professor”.