Desenvolvimento de Ferramenta para Elaboração de Textos Multisemióticos com Interação a Caneta e Gestos por Toque para Multidispositivos

Samuel G. Lima[1], André C. Silva[1,2], Fernanda M. P. Freire[2], Flávia L. Arantes[2]

[1]Instituto Federal de Educação, Ciência e Tecnologia – Campus Hortolândia (IFSP)
Av. Thereza Ana Cecon Breda, S/N - Vila São Pedro, Hortolândia - SP, 13183-250

samuel_smgl@hotmail.com, andre.constantino@ifsp.edu.br

[2]Núcleo de Informática Aplicada a Educação – UNICAMP
Rua Seis de Agosto, 50, Reitoria V - Cidade Universitária, Campinas – SP, 13083-873

{ffreire, farantes}@unicamp.br

Abstract. *The number of users who benefit from m-Learning (term used to describe the use of mobile devices in teaching-learning activities) is increasing, mainly due to the popularity of mobile devices such as smartphones and tablets. Interaction on these devices that are held by touch, pen or gesture, differs from the interaction occurred on desktop computers, whose interaction peripherals predominantly used are the keyboard, mouse and a medium-sized high-resolution screen. Thus, a variety of modalities is employed to interact with computing devices. We believe that the diversity of devices, in particular, interaction modes, is desirable because the choice of the most suitable mode to a teaching-learning activity is possible. With this motivation, this project aimed to investigate and evolve an application, called EdiMM, to explore the interaction with pen and touch gestures in educational contexts, motivating students to greater participation in the teaching process.*

Resumo. *O número de usuários que se beneficiam do m-Learning (termo usado para designar o uso de dispositivos móveis em atividades de ensino-aprendizagem) está aumentando, principalmente devido a popularização dos dispositivos móveis como smartphones e tablets. A interação nesses dispositivos, realizada por toque, a caneta ou por gestos, se difere da interação ocorrida em computadores desktops, cujos periféricos de interação predominantemente utilizados são o teclado, o mouse e uma tela de tamanho médio de alta resolução. Assim uma variedade de modalidades é empregada para interagir com dispositivos computacionais. Acreditamos que a diversidade de dispositivos, em particular, de modalidades, é desejável, pois é possível a escolha da mais adequada para uma atividade de ensino-aprendizagem. Com esta motivação, este projeto visou investigar e evoluir uma aplicação, chamada de EdiMM (editor multimodal), e explora a interação a caneta e gestos por toque em contextos educacionais, motivando os alunos a uma maior participação no processo de ensino.*

1. Introdução

Com a popularização dos dispositivos móveis, começou a surgir o uso desses dispositivos

com propósitos educacionais. Sung et al. (2005) definem m-Learning como qualquer tipo de educação que acontece quando o aprendiz não tem uma localização geográfica fixa ou pré-determinada; ou aquela que acontece quando o aprendiz aproveita as oportunidades de aprendizado oferecidas pelas tecnologias móveis. Os dispositivos móveis permitem ao aluno estudar em qualquer lugar (*anywhere*) e a qualquer hora (*anytime*).

Considerando os dispositivos móveis atuais, destacam-se celulares inteligentes e tablets cujas modalidades de entrada são baseadas em toque e/ou caneta. Essas duas modalidades possuem características diferentes, o que os diferencia quanto ao potencial de aplicação nas atividades de ensino e aprendizagem. É o caso da construção de textos multisemióticos, no qual Freire et al. (2015) afirmam que "as tecnologias e, em especial, as móveis, facilitam muito a produção de textos e hipertextos que incluem diferentes tipos de recursos expressivos, além da escrita". Dionísio (2006) descreve que:

> "... a interpretação e produção de textos multisemióticos requerem, além de domínio da escrita tradicional, outras habilidades ou multiletramentos, necessitando aprender a atribuir sentido articulando diferentes tipos de informação: texto verbal, vídeo, arquivos de som, imagem etc." (Dionísio, 2006).

Visando explorar o potencial da interação a caneta e por toque, este projeto propôs dar continuidade à um trabalho cujo propósito era desenvolver uma aplicação que explorasse as modalidades de interação a caneta e gestos na construção de textos multisemióticos, atribuindo o nome para aplicação de EdiMM. Considerando o objetivo proposto, foram definidos os seguintes objetivos específicos: Compreender as diferentes modalidades de interação e a relação entre elas e a tarefa do usuário; Conhecer a definição de usabilidade segundo a área de Interação Humano-Computador (IHC) e o design de interfaces segundo princípios de design centrado no usuário; Conhecer as técnicas de Design Responsivo; Compreender como a linguagem JavaScript trata os dados de diferentes modalidades e a possibilidade de realizar diferentes ações conforme dispositivo; Aplicar o conhecimento adquirido na implementação de uma aplicação web.

Na Seção 2 apresentamos o conceito de modalidade, questionamentos sobre princípios de design de interfaces e um exemplo de aplicação para apoio as atividades de ensino-aprendizagem. Na Seção 3 apresentamos os materiais e métodos utilizados no desenvolvimento do projeto. A avaliação do protótipo do EdiMM e a sua nova versão são apresentados na Seção 4. Por fim a Seção 5 apresenta os resultados alcançados no decorrer do projeto.

2. Fundamentação Teórica

Modalidade é o termo empregado para definir o modo como uma entrada do usuário e uma saída do sistema são expressas. Nigay e Coutaz (1995) definem modalidade como um método de interação que um agente pode usar para alcançar uma meta. Segundo os autores uma modalidade pode ser especificada em termos gerais como "usando fala" ou em termos mais específicos como "usando microfone". Bernsen (2008) e, posteriormente, Bernsen e Dykjær (2010) discutem que:

> "... a modalidade, ou, mais explicitamente, uma modalidade de representação da informação, é uma forma de representar a informação em algum midium. Assim, uma modalidade é definida por seu midium e sua particular 'forma' de representação" (Bernsen e Dykjær, 2010, tradução nossa).

Segundo da Silva (2014) Várias modalidades tornaram-se foco de pesquisa nas últimas décadas, entre elas a fala (reconhecimento e sintetização), a escrita manuscrita (reconhecimento) e a interação com dedos por meio de toques e gestos. Assim, surgiram

áreas de pesquisa dentro da Ciência da Computação que visam possibilitar a entrada e a saída de dados por uma modalidade, bem como explorar suas vantagens, como é o caso da Computação Baseada em Caneta e a interação por gestos, modalidades de estudo deste trabalho.

Freire et al. (2015) apresentam um estudo de viabilidade de um editor que possibilita a elaboração de um texto utilizando textos digitados, textos manuscritos e imagens. Os autores desenvolveram um protótipo da aplicação, chamada de Editor Multimodal (posteriormente chamado de EdiMM) entretanto com diversas limitações como (i) código testado apenas no navegador Chrome; (ii) código testado apenas em computadores desktop; (iii) interface de usuário que necessita de redesign e aprimoramento. Ao testarmos o editor em dispositivos com suporte a toque e a caneta, percebemos que os dados oriundos de diferentes modalidades são tratados da mesma forma, ou seja, não é possível usar a caneta para escrever com letra manuscrita e usar os dedos para acionar funcionalidades como zoom ou deslizamento da página onde está se escrevendo, por exemplo.

Em relação ao design de uma interface de usuário, destacamos que este é um processo complexo, sendo necessário considerar que o usuário foque na tarefa e não no aprendizado da tecnologia (Shneiderman e Plaisant, 2004) e operar a aplicação independente do dispositivo do usuário, focando na tarefa (Shneiderman, 2006). Assim é necessária uma consistência da interface de usuário quando acessada entre dispositivos com caraterísticas diferentes. Uma das atuais vertentes para se manter uma certa consistência entre dispositivos com telas de diferentes tamanhos é o emprego de Design Responsivo (Marcotte, 2011).

3. Materiais e Métodos

Em relação às tecnologias empregadas para o desenvolvimento, foi definido que aplicação seria para a plataforma Web e para multidispositivos, adotando-se uma arquitetura cliente-servidor. Empregamos as tecnologias disponíveis na Tabela 1. No componente servidor, preferimos a linguagem de programação PHP para, posteriormente, integração da aplicação com ambientes virtuais de aprendizagem (AVA), como o TelEduc. Como base, resolvemos adotar o protótipo do EdiMM, conforme explicado na Seção anterior. O protótipo do EdiMM deveria possibilitar a escrita de textos manuscritos por meio de uma caneta (*stylus*).

Propomos então, a partir do código disponível, o redesign da interface de usuário adotando princípios de Projeto Centrado no Usuário (ISO 9241-210, 2010), utilizando-se a literatura como base para se obter conhecimento sobre informática na educação e o uso de dispositivos computacionais, em especial com interação a caneta e por gestos, no contexto educacional. Essas atividades tornam-se necessárias, pois, conforme descrito na fundamentação teórica, desenvolver uma interface de usuário para uma aplicação é necessário conhecer (i) a tecnologia, (ii) os usuários e (iii) as tarefas. Portanto, o processo de design é uma tarefa complexa, ainda mais se considerarmos as modalidades de interação à caneta e por gestos que, embora haja trabalhos na literatura científica, ainda existe muito a ser explorado, principalmente em relação a sua complementação, as interfaces multimodais (Bernsen, 2010).

Foram necessários dispositivos móveis e um computador para desenvolvimento deste Trabalho de Conclusão de Curso (Tabela 1). Os dispositivos foram selecionados, pois possui uma caneta, chamada *stylus*, que possibilita suporte à interação a caneta e a gestos por caneta além de identificar aspectos de melhoria de interface nos dispositivos móbiles.

Após a compreensão do código do protótipo do EdiMM (Figura 1), propomos a aplicação de técnicas de Design Responsivo (Marcote, 2011) para que a aplicação seja

responsiva conforme dispositivo utilizado na interação. Executamos testes nos dispositivos especificados na Tabela 1 e identificamos melhorias que foram realizadas até a geração de uma nova versão da aplicação. Nas próximas seções, detalhamos cada uma das principais fases realizadas para a geração da nova versão da aplicação.

Tabela 1 – Dispositivos e Ferramentas utilizados no EdiMM

<table>
<tr><th colspan="2">Dispositivos</th><th colspan="3">Tecnologias</th><th>Ferramentas</th></tr>
<tr><td colspan="2">Desktop</td><td colspan="3">Design Responsivo</td><td>Repositório</td></tr>
<tr><td>S.O. Windows 7 e 10</td><td>S.O. Ubuntu 18.1</td><td colspan="2">Bootstrap 3.3.6</td><td>HTML5</td><td>GitHub 2.12</td></tr>
<tr><td colspan="2">Smartphone Samsung Note 4 e S7</td><td colspan="3">Interface de usuário</td><td>Framework de desenvolvimento</td></tr>
<tr><td colspan="2">Android 6.0 e 7.0</td><td>HTML</td><td>JavaScript</td><td>CSS</td><td>Notepad ++ 7.5.6</td></tr>
<tr><td colspan="2">Tablet Galaxy Samsung Note 10</td><td colspan="3">Servidor</td><td>SGBD</td></tr>
<tr><td colspan="2">Android 6.0</td><td colspan="3">PHP 5.2</td><td>PHP My Admin 4.0.10</td></tr>
</table>

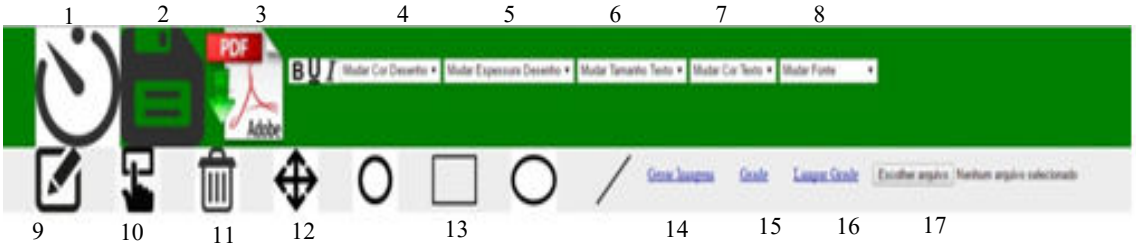
4. Avaliação e desenvolvimento do protótipo do EdiMM

O protótipo do EdiMM (Figuras 1 e 2) foi desenvolvido durante o ano de 2015 por Renato Oliveira, com orientação do professor André Constantino da Silva (IFSP) e das pesquisadoras Fernanda M. P. Freire (NIED/UNICAMP) e Flávia L. Arantes (NIED/UNICAMP), orientador e co-orientadores deste Trabalho de Conclusão de Curso. O protótipo contava com algumas funcionalidades específicas, por exemplo: escrita de texto; desenho livre; ilustrar formas geométricas; importar imagem; salvar no SGBD MySQL; gerar grade; limpar grade; importar imagem; e salvar em PDF; além disso havia a possibilidade de escolher a fonte do texto a ser utilizada e as cores limitadas através de uma caixa de seleção, também sendo possível mover ou remover os elementos (Figura 1).

4.1 Análise do EdiMM

Analisamos o código e executamos o protótipo nos equipamentos selecionados e encontramos diversos problemas relacionado à sua implementação arquitetural e suas funcionalidades, por exemplo:

- A função salvar no formato PDF não ajustava o canvas com a resolução da tela;
- As imagens importadas não serem renderizadas após o usuário acionar a função gerar PDF;
- A função escrever texto estava sobrescrevendo as demais funções ao deslizar o mouse por cima da letra no SVG;



Legenda:

1	Criar novo arquivo
2	Salvar no servidor
3	Salvar no formato PDF
4	Selecionar cor de desenho
5	Selecionar espessura de desenho
6	Modificar tamanho da fonte do texto
7	Alterar cor da fonte do texto
8	Alterar fonte do texto
9	Escrita de texto
10	Desenhar a mão livre
11	Apagar elemento
12	Mover elementos
13	Desenhar formas geométricas
14	Salvar no formato PNG
15	Selecionar grade de fundo
16	Limpar grade
17	Selecionar arquivo de imagem para importar

Figura 1. Componentes que compõe a Interface de usuário do protótipo do EdiMM executada em um computador desktop.

Figura 2. Interface de usuário do protótipo do EdiMM executada em um smartphone com tela de 5,7 polegadas.

- Foi encontrado problemas na função mover elementos ao manter selecionado o ponto que indica o início do texto com o botão esquerdo do mouse, o editor interpretava como a função mover elementos e logo acionava a movimentação do mesmo;
- Problemas de usabilidade e de responsividade, pois seu design não se

adequava aos diferentes tipos de dispositivos como smartphones ou tablets se restringindo apenas aos desktops;

- As cores nessa versão mostram-se limitada, pois, ao acionar essa funcionalidade pelo usuário, a interface apresenta uma caixa de seleção restrita com apenas nove opção de cores, sendo que o ideal seria uma paleta de cores para sua livre escolha, abrindo um leque maior de possibilidades de cores distintas;
- A versão apresenta duas ferramentas para seleção de cores, uma voltada para desenhos geométricos e a outra para escrita de textos, confundindo os usuários, pois o ideal é apenas uma paleta de cores voltada para as funções que vão se apropria dessa opção;
- Ícones confusos;
- A disponibilização dos elementos não está adequada à interface;
- Problemas na estrutura e endentação do código-fonte;
- Ausência de separação do código que compõe a interface de usuário. O código em HTML (Linguagem de Marcação), CSS (Estilos de paginação), JavaScript (Linguagem de programação interpretada), estão no mesmo arquivo, e não havia modelo de desenvolvimento empregado ao projeto.

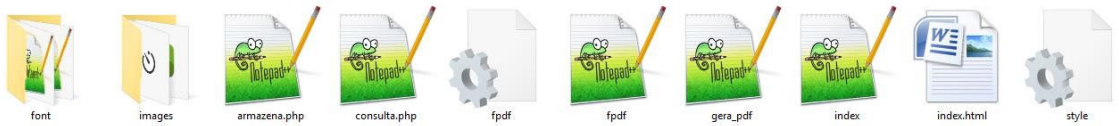


Figura 3. Organização dos arquivos de código-fonte que compõe o protótipo do EdiMM.

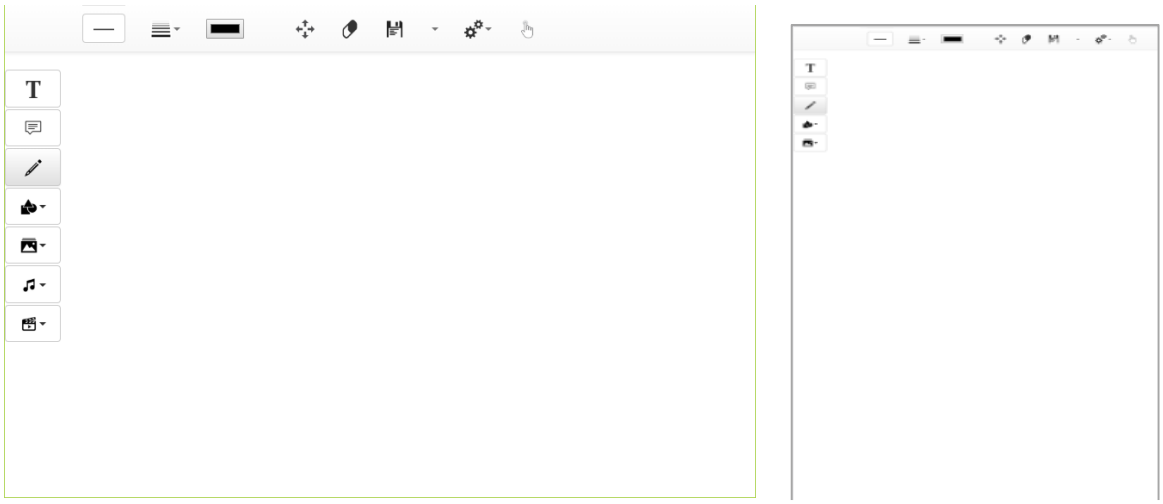
4.2 Evolução do protótipo EdiMM

Após uma análise detalhada do EdiMM, foram estudas e implementadas modificações, como por exemplo, a reestruturação do código e especificada uma nova interface de usuário.

Durante o estudo do código e das funcionalidades do EdiMM, foi especificada uma arquitetura que divide as responsabilidades e separa os scripts em arquivos separados, organizando o código da aplicação. Como padrão adotou-se que cada pasta corresponde apenas a sua linguagem de desenvolvimento; o código interno foi endentado, funções foram organizadas e as variáveis renomeadas de acordo sua responsabilidade, garantindo a qualidade e manutenibilidade do código do produto (Figura 4).

Figura 4. Organização dos arquivos de código-fonte que compõe a nova versão do EdiMM.

Uma nova interface de usuário foi projetada com objetivo de melhorar a usabilidade da ferramenta em diferentes dispositivos (Figura 5). Na implementação desta nova interface, utilizou-se o framework *Bootstrap* com o objetivo de redimensionar o layout de acordo com a resolução da tela do dispositivo utilizado.



(a) (b)

Figura 5. Nova "versão da interface de usuário do EdiMM executada em (a) um computador de mesa e (b) um smartphone com tela de 5.7".

A utilização do Design Responsivo na aplicação nos ajudou a alcançar um dos objetivos descrito no início deste artigo, pois a aplicação possui a necessidade de ser multidispositivo, permitindo que os usuários utilizem seus dispositivos para a produção de textos multissemiótico. Além disso, a aplicação tem como objetivo explorar a multimodalidade por gestos e toque na forma de interação dos usuários com a aplicação, pensando nesses aspectos a interface também foi modelada de forma que seja de fácil compreensão e aprendizado na visão de um usuário, facilitando o acesso as ferramentas da aplicação e se adequando as funcionalidades que estão sendo executadas pelos usuários (Figura 6).

Tabela 2 – Listagem das funcionalidades existentes e evoluídas e das novas funcionalidades.

Novas funcionalidades		**Funcionalidades existentes e evoluídas**	
01	Forma geométrica selecionada	14	Paletas de cores
02	Realizar download (SVG)	15	Salvar em PDF
03	Linha vertical	16	Salvar em PNG
04	Linha horizontal	17	Mover componentes
05	Grade	18	Escrita de texto
06	Grade milimetrada	19	Desenho livre
07	Multimodalidade	20	Formas geométricas
08	Multitoque	21	Apagar componente
09	Aprimoramento na interface	22	Salvar no servidor
10	Design Responsivo	23	Limpar layout
11	Caixa de texto	24	Espessura da borda
12	Importa imagem / áudio	25	Novo arquivo
13	Cores de fundo	26	Alterar tipo / tamanho da fonte do texto

Após realizar ajustes a função responsável pelas cores com o propósito de melhorar a interface da aplicação frente à usabilidade dos usuários, já é possível atribuir a cor especificada as demais funções como, por exemplo, formas geométricas, escrita de textos e linhas de grade. Foram incluídos rótulos de textos em formato de dica (*tooltip*) para as diversas ferramentas e funcionalidades para o usuário saber o que cada botão realiza (para acionar é necessário deixar o ponteiro do mouse por cima da função por alguns segundos) e os ícones foram modificados para melhor espelhar a natureza de cada

funcionalidade de acordo com sua imagem (Figura 6).

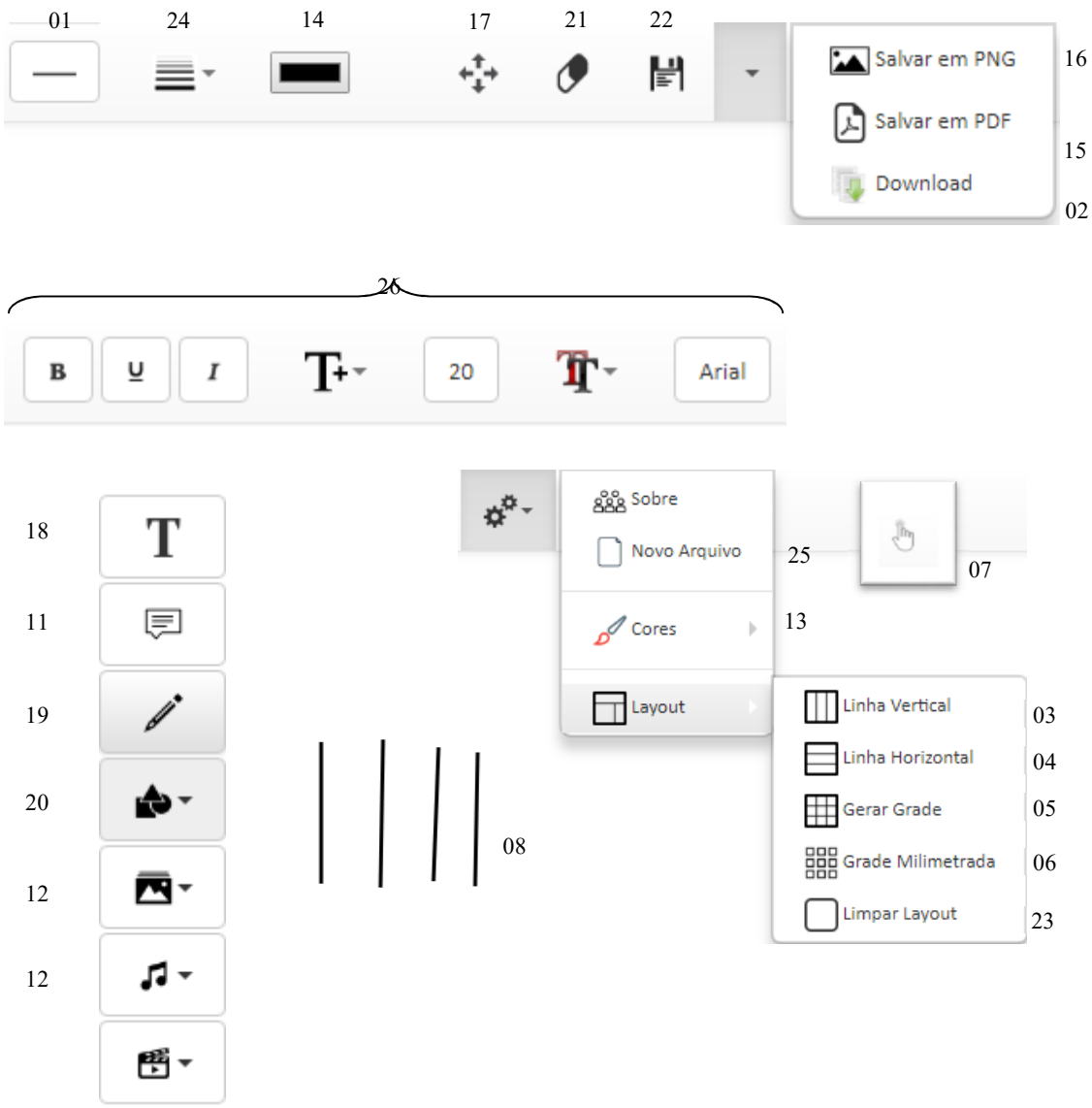


Figura 6. Elementos da nova versão da interface de usuário do EdiMM executada em um computador desktop.

Pensando na interface da aplicação frente à usabilidade dos usuários, as opções de espessura dos desenhos, que antes eram especificadas em valores, foram substituídas por barras de diferenças espessuras, assim o usuário assemelha a imagem a utilização da função no caso desenho e a espessura esperada. A função desenho livre foi considerada como *default* ao iniciar a página. Outra modificação importante foi a criação de uma barra de ferramentas superior contendo as funcionalidades relacionadas à ferramenta selecionada pelo usuário no menu lateral esquerdo, evitando a "poluição visual" de ações contendo funcionalidades que não estão disponíveis na ferramenta selecionada, por exemplo: caso o usuário escolha a ferramenta escrita de texto no menu lateral esquerdo, a barra de ferramentas superior vai apresentar somente as funções correspondentes a função texto, por exemplo, cor da letra, tamanho da letra, tipo de fonte e formação (itálico, negrito e sublinhado). No caso da ferramenta desenho geométrico, as opções na barra de

funcionalidades são a figura geométrica, a cor e a espessura do traço (Figura 6).

Outra adição foi o acréscimo de desenho a mão livre com suporte a *Multitouch* para dispositivos *Touch Screen*, ou seja, a capacidade de reconhecer a presença de dois ou mais pontos de contato com a superfície de detecção de toque. Ao acionar a funcionalidade de desenho a mão livre em um dispositivo móvel, por exemplo, permite ao usuário desenhar traços com mais de um dedo ou que dois ou mais usuários interajam na mesma superfície construindo o texto semiótico colaborativamente.

Também foi adicionado um botão para que o usuário possa indicar se deseja usar os dados capturados pela tela sensível ao toque como conteúdo para o texto multisemiótico ou se deseja utilizar esses dados para acionar funcionalidades da aplicação. Isso é útil no caso de dispositivos com diversas modalidades de entrada, por exemplo, o usuário pode utilizar o dedo para selecionar funcionalidades e a caneta para escrever.

Além disso, foram apresentados e corrigidos bugs persistentes em algumas funcionalidades, por exemplo: a função "Gerar PDF"; as imagens importadas agora são renderizadas ao acionar a opção de inserir imagem; a movimentação está designada somente para a função mover componentes, e a escrita de texto não estão mais sobrescrevendo as demais funcionalidades; o uso da função paleta de cores já implementada para colorir os elementos internamente: ao acionar a opção cores no menu superior o usuário poderá escolher se o elemento será colorido internamente ou somente será colorido as bordas dos elementos, através de uma caixa de seleção. Além da função paleta de cores também foram trabalhadas as demais ferramentas, por exemplo, as linhas dos elementos; após a opção "Linha Tracejada" ser selecionada no menu superior, as linhas ou bordas dos elementos vão adquirir um determinado espaçamento resultando em traços na linha (Figura 7).

A função gerar grade na versão inicial atribuía somente linhas horizontais ao SVG (Figura 8), essa função foi estendida com o propósito de possibilitar a atribuição de diferentes layouts na aplicação, a saber, linha vertical, linha horizontal, grade e grade milimetrada (Figura 9), necessitando somente ao usuário indicar o layout desejado no menu superior da aplicação disponível na guia configuração. Um exemplo de uso dos layouts de grade é possibilitar a construção de gráficos matemáticos.

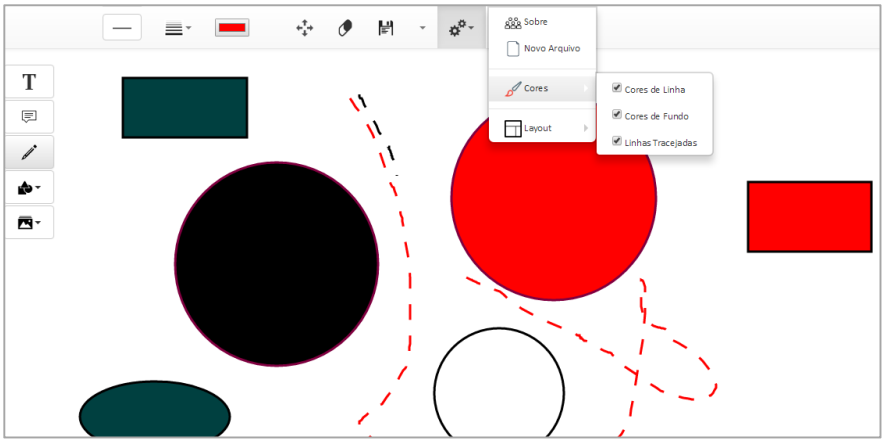


Figura 7. "Função Desenho Livre tracejado e coloração interna aos elementos sendo executado em um Tablet com tela de 10.1".

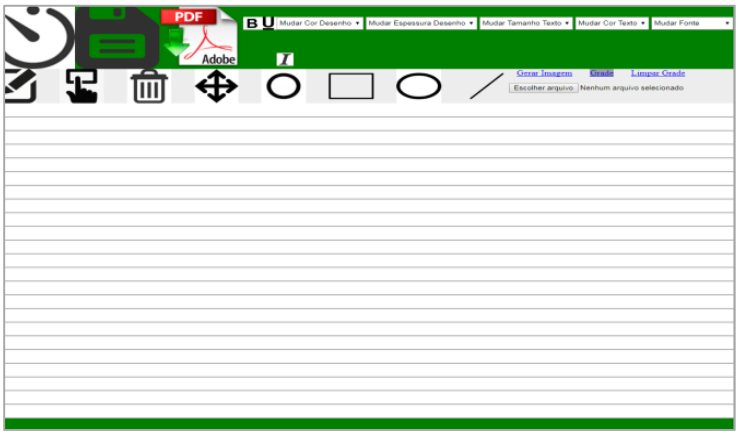


Figura 8. Função Grade Milimetrada no protótipo do Editor Multimodal.

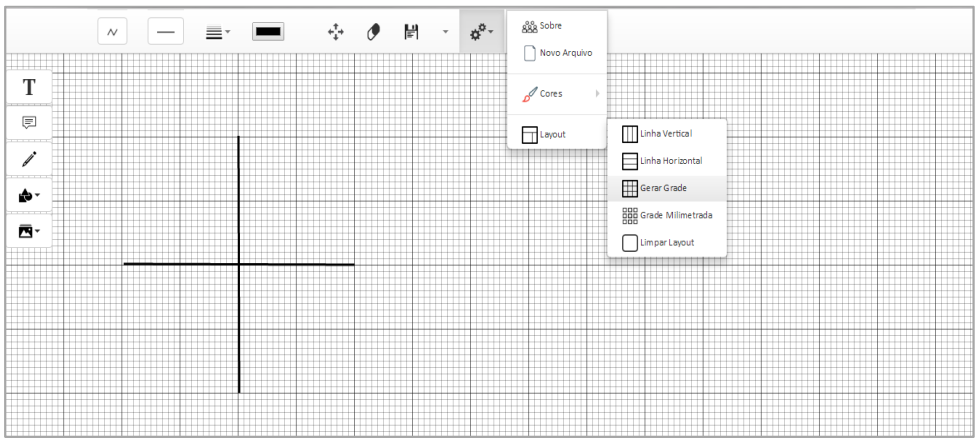


Figura 9. Função grade milimetrada sendo executado em um notebook com tela de 14 polegadas.

Também foi modificada a forma da interação com textos digitados por meio da inclusão de uma caixa de texto (acionável por meio do menu lateral esquerdo da aplicação). Essa função permite ao usuário acrescentar uma caixa de texto na área de produção do texto sendo possível redimensioná-la de acordo com o texto descrito em seu interior. A função também permite utilizar recursos já conhecidos do teclado pelos usuários, por exemplo: caixa alta; quebra de linha; backspace entre outros (Figura 10).

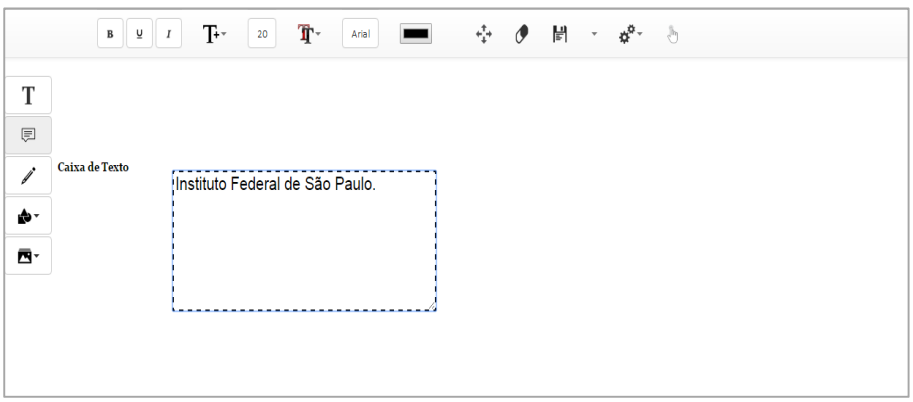


Figura 10. Função Caixa de Texto sendo executado em um Notebook com tela de 14 polegadas.

4.3 SGBD utilizado no EdiMM

O modelo conceitual representa a abstração do banco de dados relacionado ao projeto; embora não fosse a proposta deste trabalho realizar modificações na estrutura do banco de dados, achamos pertinente registrar a estrutura do banco da aplicação. A Figura 11 apresenta a entidade “svgtable” e dois atributos incluídos: “id” e “tag_svg”. O atributo “id” é responsável por armazenar o código de acesso ao conteúdo desenvolvido, esse atributo possui uma marcação na qual se destaca como sendo um atributo chave nessa entidade. O atributo “tag_svg” é responsável por armazenar o conteúdo desenvolvido na aplicação (Figura 11).

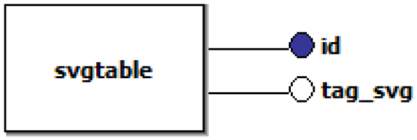


Figura 11. Modelo conceitual do EdiMM.

O modelo físico passa o nível de abstração das entidades para a visão estrutural da base de dados. A Figura 3.11 apresenta a tabela “svgtable” e os valores suportados em seus campos. Definiu-se que o campo “id” armazena valores no formato texto possuindo uma limitação de 50 caracteres que será utilizado pelos usuários para informar qual documento desejam acessar, assim, permite ao usuário recuperar o conteúdo salvo na aplicação, podendo ser visualizado ou modificado. Por isso, este código é informado ao usuário quando este solicita que o seu texto multimodal seja salvo. Decidiu-se adotar o tipo *varchar* e este ser o campo de identificação da tupla, sendo assim, a *primary key* e permitindo somente chaves diferenciadas serem armazenadas na tabela, impedindo a ocorrência de eventos indesejáveis, por exemplo, chaves duplicadas a conteúdos distintos.

O objetivo do campo “tag_svg” é permitir armazenar o conteúdo do texto elaborado pelo usuário, que está em SVG formato baseado em XML, que utiliza tags (etiquetas) para descrever os dados, o que acarreta no alto número de bytes para armazenar seus dados, mas, por outro lado, pode ser interpretado em diferentes máquinas; oferecendo a portabilidade desejada para nosso editor. Assim, definiu-se que este campo seria do tipo *longblob*, suportando no máximo 4.294.967.295 bytes - 4 GB de armazenamento de informação (Figura 12).

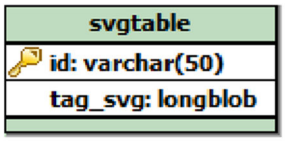


Figura 12. Modelo físico do EdiMM.

O SGBD (Sistema Gerenciador de Banco de Dados) utilizado atualmente no projeto é o MySQL. O banco de dados responsável por armazenar a tabela “svgtable” possui o nome de “EditorMm”. A representação e a criação da estrutura da tabela “svgtable” no SGBD MySQL são descritas na Figura 13.

```
CREATE TABLE IF NOT EXISTS 'svgtable' (
   'id' varchar(50) NOT NULL,
   'tag_svg' longblob NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

Figura 13. Código SQL da criação da tabela svgtable no SGBD MySQL.

4.4 Utilizando Git Hub no EdiMM

O GitHub é um Serviço de Web Hosting compartilhado para projetos que usam o controle de versionamento Git. Os serviços do GitHub contem planos comerciais e gratuitos para projetos de código aberto, além de possuir funcionalidades encontradas em uma rede social como, por exemplo, *feeds*, *followers*, wiki e um gráfico que transparece aos colaboradores envolvidos no projeto o índice de contribuição dos colaboradores no controle de versões do repositório especificado (Figura 14).

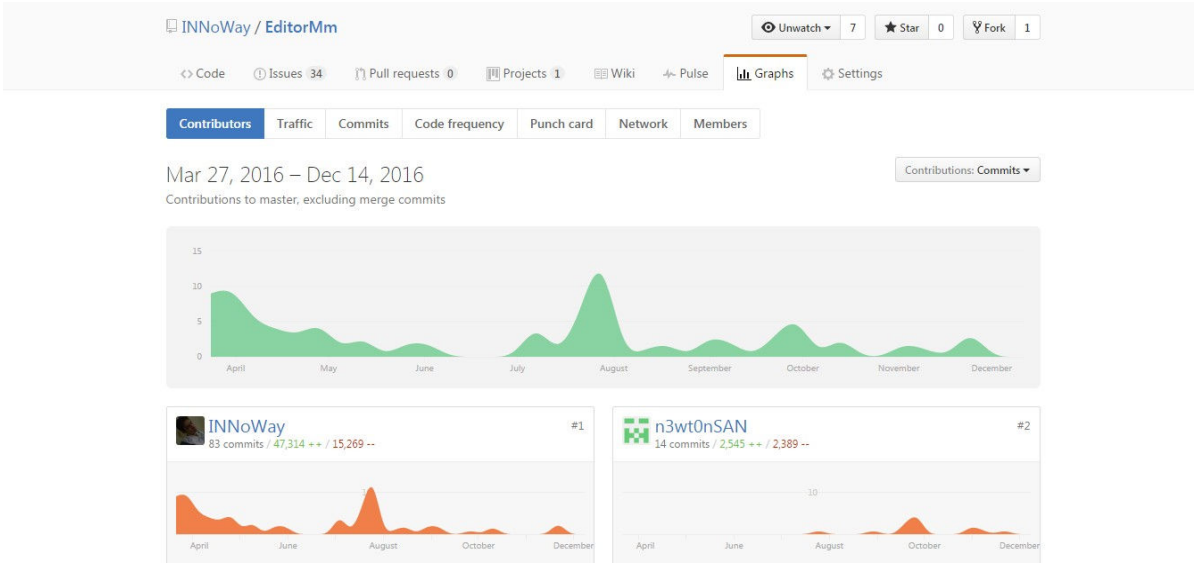


Figura 14. Gráfico ilustrando a contribuição dos colaboradores.
(Fonte: https://github.com/INNoWay/EditorMm/graphs/contributors)

O GitHub permite dois níveis de permissão: o proprietário do repositório (administrador) e colaboradores (membros do projeto). O proprietário do repositório tem controle total do repositório, por exemplo: convidar colaboradores; alterar a visibilidade do repositório do público para o privado ou de privado para público; mesclar uma solicitação de recebimento em uma filial protegida mesmo se não há revisões aprovadas; eliminar o repositório etc.

O EdiMM está localizado no repositório EditorMm disponível no endereço (https://github.com/INNoWay/EditorMm). A equipe do NIED/UNICAMP participa do projeto atualmente e contribuem através do GitHub de diferentes formas como, por exemplo, realizando testes na aplicação; sugerindo melhorias na interface; corrigindo bugs encontrados nas funções; contribuem com a manutenibilidade do código fonte; e, atualmente, atribuindo responsabilidade aos envolvidos no projeto, e para saber quem são os membros que estão ou estiveram envolvidos no projeto do EdiMM até momento pode ser localizado através da guia: Configurações/Sobre

A intenção do projeto futuramente é expandir a comunidade para proporcionar as pessoas a contribuírem com o projeto, com novas ideias e soluções, desenvolvendo e visando a participação da comunidade auxiliando sempre na melhoria da aplicação.

O projeto armazenado no GitHub atualmente possui diversas tarefas a serem atribuídas e implementadas na aplicação; grande parte dessas tarefas já foram realizadas, porém os testes na aplicação são realizados em diferentes plataforma e navegadores, consistindo em um número relativamente grande de bugs que ainda podem estar atrelado ao projeto.

Com base nos testes que ainda estão sendo realizados pelos membros envolvidos no projeto, será possível analisar e documentar os problemas encontrados e as tecnologias utilizadas no projeto. Os testes no momento estão sendo realizado apenas pela equipe

envolvida no projeto e só estão verificando problemas relacionados às funcionalidades das funcionalidades disponíveis no EdiMM, assim os futuros membros poderão realizar testes mais precisos como de usabilidades e funcionais e até mesmo os desenvolvedores da comunidade vão estar preparados para entender a estrutura e as tecnologias empregadas no EdiMM, e resolver os problemas atrelados a ele rapidamente de forma eficaz e eficiente.

4.5 Estudos das tecnologias para suporte a diferentes modalidades

Após as modificações realizadas no código, estudamos como os navegadores tratam os dados provenientes das modalidades mouse, toque e caneta. Também foram estudadas as padronizações definidas pela W3C, com o objetivo de se produzir um código que não fique limitado as tecnologias e implementações atuais dos navegadores.

Segundo a W3C, para o tratamento de diferentes modalidades de entrada, os navegadores deverão suportar três tipos de eventos (Tabela 3): *mouse events* (eventos de mouse), *touchevents* (eventos de toque) e *pointer events* (eventos de ponteiro). Antigamente, os navegadores davam suporte a somente entradas pelo mouse e pelo teclado, assim os navegadores dispunham, por meio da linguagem JavaScript, métodos para acionar ações para tratar eventos somente desses dispositivos. Por exemplo, o programador pode incluir uma função em JavaScript que será acionada quando o usuário dispor o apontador do mouse sobre um botão, bastando que inclua a chamada a esta função no atributo *onmouseover* do referido botão. A Tabela 3 apresenta os eventos de mouse bem como também os *touchevents* e de ponteiro. Nas subseções a seguir explicaremos os *touchevents* e de ponteiro.

Tabela 3 - Eventos de Mouse, de Ponteiro e de toque
(Fonte: https://docs.webplatform.org/wiki/concepts/Pointer_Events, tradução nossa)

Ação\Tipo de Evento	**Mouse event**	**Pointer event**	**Touch event**
Usuário aciona o dispositivo sobre o elemento.	mousedown	pointerdown	touchstart
Usuário libera o dispositivo sobre o elemento.	mouseup	pointerup	touchend
Usuário desliza o dispositivo sobre o elemento.	mousemove	pointermove	touchmove
Usuário está com o dispositivo sobre o elemento.	mouseover	pointerover	-
Usuário movimenta o dispositivo para fora do elemento ou de seus filhos.	mouseout	pointerout	-
Usuário movimenta o dispositivo para dentro do elemento.	mouseenter	pointerenter	-
Acionado quando o navegador percebe que o dispositivo não irá mais gerar eventos.	-	pointercancel	touchcancel

4.5.1 Touch Events

Os *touchevents* oferecem a capacidade de interpretar atividades em telas sensíveis ao toque realizadas por meio dos dedos. O *Touch Events* fornece os seguintes tipos de entrada:

- *Touchstart*: disparado quando o usuário coloca um ou mais dedos na tela.

- *Touchmove*: disparado enquanto o usuário move o dedo sobre a tela.
- *Touchend*: disparado quando o usuário para de tocar na tela.
- *Touchleave*: disparado quando o usuário move seus dedos fora da área de escuta do evento.
- *Touchcancel*: disparado quando um gesto de toque for cancelado, por exemplo, se o usuário move os dedos fora da própria tela.

Touchevents foram padronizados pela W3C em 2014; mesmo assim, até a realização deste trabalho, nem todos os navegadores ou versão dos navegadores ainda suportam esses eventos. A Tabela 4 apresenta as versões dos navegadores com suporte (ou não) a *touchevents*.

Tabela 4 – *Mouse Events*, *Pointer Events* e *Touch Events*
(Fonte: https://docs.webplatform.org/wiki/concepts/Pointer_Events)

IE	Edge	Firefox	Chrome	Safari	Opera	IOS Safari	Opera Mobile	Chrome for Android	Firefox for Android	IE Mobile
			29 ~							
			47 ~							
			48 ~							
		45 *	49 ~							
8 *		46 *	50 ~		37 ~	9.2 ~				
11 *	13 *	47 *	51 ~	9.1 *	38 ~	9.3 ~	37 ~	50 ~	46 ~	11 !
	14 *	48 *	52 ~	10 *	39 ~					
		49 *	53 ~	TP *	40 ~					
		50 *	54 ~							

Suporta	~
Não Suporta	*
Suporta parcialmente	!
Suporte Desconhecido	x

4.5.2 Pointer Events

Todos os navegadores web de hoje tem suporte para os eventos de mouse *mousedown, mousemove, mouseup* etc; porém muitos usuários podem não estar utilizando um mouse (por exemplo, os usuários podem estar interagindo com uma página web usando seus dedos em um smartphone, ou usando uma caneta *stylus*). Considerando esse fato, os desenvolvedores de navegadores decidiram fornecer um modelo unificado para todos os três tipos de entrada (caneta, toque ou mouse), criando eventos chamados de *Pointer Events* (*pointerdown, pointerrmove, pointerup*, entre outros), uma representação abstrata de dispositivos de entrada, que possibilita aos desenvolvedores escreverem código único para ações semelhantes (Figura 15) como, por exemplo, dispor o dispositivo sobre o elemento ou mover o dispositivo fora do elemento. Uma das vantagens do uso de *Pointer Events* o é que ele foi projetado para ser compatível com os dispositivos de entrada atuais e cobrir futuros paradigmas de interação.

Em geral, pode-se dizer que *Pointers Events* herdam e se estendem de eventos do mouse, por isso tem todas as propriedades habituais que os eventos do mouse possuem porém os diferentes navegadores podem tratar de forma diferente suas propriedades, podendo suportá-las por completo ou apenas parcialmente, até mesmo rejeitando suas

propriedades. O *Pointer Events* fornece tratamento para os seguintes eventos de entrada:

- pointerdown: é acionado quando o usuário deseja disparar ação sobre o elemento (clicando com o mouse ou tocando com o dedo, por exemplo).
- pointerup: é acionado quando o usuário deseja libera a ação sobre o elemento após o disparo (liberando o botão do mouse ou retirando o dedo, por exemplo).
- pointerover: é acionado quando um usuário move o dispositivo sobre o elemento (movendo o mouse ou movendo a caneta, por exemplo).
- pointerout: é acionado quando um usuário move o dispositivo para fora do elemento (movendo o mouse ou movendo a caneta, por exemplo).
- pointerenter: é acionado quando um usuário move o dispositivo para dentro de um elemento (movendo o mouse ou movendo a caneta, por exemplo).
- pointerleave: é acionado quando um usuário move o dispositivo para fora de um elemento (movendo o mouse ou movendo a caneta, por exemplo).

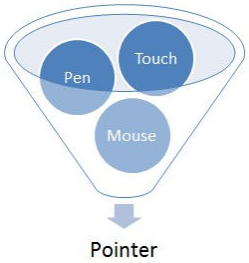


Figura 15. Um ponteiro é uma representação abstrata de dispositivos de entrada que acionam coordenadas específicas (ou um conjunto de coordenadas) na tela. (Fonte: https://www.w3.org/TR/pointerevents/)

A tabela a seguir relata as versões dos navegadores mais utilizados atualmente, assim podemos saber, quais são as versões dos navegadores que oferecem suporte para os *Pointer Events* (Tabela 5).

Tabela 5. Suporte aos *Pointer Events* nos principais navegadores (Fonte: http://caniuse.com/search%3Dpointer%20events)

IE	Edge	Firefox	Chrome	Safari	Opera	IOS Safari	Opera Mobile	Chrome for Android	Firefox for Android	IE Mobile
			29 *							
			47 *							
			48 *							
		45 *	49 *							
8 *		46 *	50 *		37 *	9.2 *				
11 ~	13 ~	47 *	51 *	9.1 *	38 *	9.3 *	37 *	50 *	46 *	11 ~
	14 ~	48 !	52 *	10 *	39 *					
		49 !	53 *	TP *	40 *					
		50 !	54 *							

Suporta
Não Suporta
Suporta parcialmente
Suporte Desconhecido

4.5.3 Tratamento para as diferentes modalidades

Utilizamos os eventos para ponteiro no projeto pois, nos últimos anos, tem havido uma grande variedade de dispositivos móveis que utilizam outros mecanismos de entrada não se limitando somente a toque como, por exemplo, a caneta *stylus*. Nosso objetivo foi averiguar a possibilidade de tratar de forma diferente os dados provenientes de diferentes modalidades (toque ou caneta).

O algoritmo a seguir exemplifica o tratamento para chamada de funções para desenho de círculo conforme o tipo de dispositivo utilizado pelo usuário. Quando existe caneta no equipamento do usuário (conteúdo da variável *stylusIsEnable* é verdadeiro) utiliza-se os *Pointer Events*; quando o usuário está utilizando um dispositivo com tela sensível ao toque (*touchIsEnable* é verdadeiro) utiliza-se os *touchevents*; e, independente do dispositivo, o usuário poderá utilizar o mouse, pois serão acionados os eventos de mouse.

```
Inicio

//caso o dispositivo de entrada seja caneta alert ("Usando pointer");
se Pointer Events = true então
      Adicione o evento de ponteiro,
Senão

//caso o dispositivo de entrada for o toque alert ("Usando touch");
se Touch Events = true então
      Adicione o evento de Toque,
Senão

//disponível independente do dispositivo de entrada alert ("Usando mouse");
      Adicione o evento de Mouse,

Fim
```

Esse mecanismo está relacionado também ao contexto de várias modalidades, pois através dessa função conseguimos perceber os diferentes tratamentos que poderão ser utilizados pela ferramenta. Sendo assim ao reconhecer que um determinado dispositivo utiliza caneta e toque como eventos de entrada, ele poderá realizar tratamentos diferenciados as funções disponíveis na ferramenta, por exemplo: a utilização da caneta para escrever ou desenhar na tela e o dedo para atribuir aproximação ou afastamento da tela (zoom), como também deslizar a tela para obter mais espaço livre para realizar anotações (Figura 16).

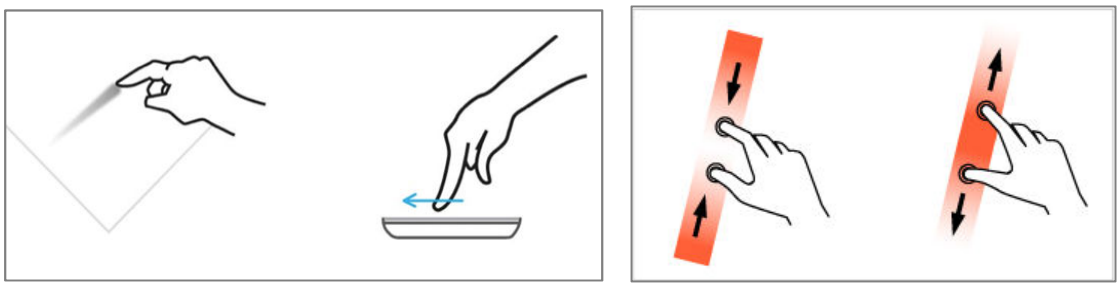


Figura 16. Movimento dos dedos para (a) deslizamento dos elementos na tela e (b) para aproximação ou afastamento dos elementos na tela (zoom).

5. Conclusão

A partir de um código de protótipo do EdiMM, este Trabalho de Conclusão de Curso elaborou uma nova versão da ferramenta com suporte a diferentes modalidades na interação (chamada de aplicação multimodal) e na elaboração de textos (chamado de texto multisemiótico) modificando o seu código para melhor legibilidade e compreensão, tornando-o mais fácil de ser mantido, corrigindo funcionalidades e a sua interface de usuário, melhorando a sua usabilidade.

O código do protótipo foi estudado e sua interface avaliada e, com base nos estudos sobre os conceitos da área de Interação Humano-Computador, como usabilidade e métodos de investigação de usabilidade, uma nova interface foi sugerida. Foram desenvolvidas novas funcionalidades e corrigidos bugs relacionados as funcionalidades já existentes anteriormente, além de uma reestruturação no código fonte e na arquitetura realizada anteriormente. Embora o protótipo já contava com o suporte necessário para as modalidades de caneta e gestos para uso em contextos educacionais, a aplicação não estava funcionando nos dispositivos testados; esses problemas foram corrigidos e não se encontram na versão atual.

Em seguida, foram realizadas investigações sobre o tratamento de dados oriundo de caneta e de gestos na tecnologia JavaScript, incluindo suas limitações em relação aos navegadores atuais, como apresentamos nas tabelas. Ressalta-se que este é um ponto de inovação deste projeto, pois ainda existem poucas aplicações Web que se aproveitam da interação a caneta e suportam as atividades de ensino-aprendizagem.

Foram realizadas reuniões junto da equipe do NIED/UNICAMP, onde o desenvolvimento de novas funcionalidades e melhorias, relacionado ao projeto foi e está sendo discutidas para estabelecer metas e processos afim de contribuir com a pesquisa científica em parceria entre as duas instituições.

Ressalta-se que este trabalho foi apresentado no 7° Congresso de Iniciação Científica e Tecnológica na cidade de Matão com o tema “Prototipação e desenvolvimento de aplicação para elaboração de textos multisemióticos com interação a caneta e gestos por toque” cujo objetivo foi apresentar a ferramenta utilizada dentro do contexto de ensino e aprendizagem na elaboração de textos multisemióticos. O projeto foi apresentado na modalidade pôster na V SNCT (Semana Nacional de Ciência e Tecnologia) realizada no Instituto Federal de Educação Ciência e Tecnologia – Campus Hortolândia, recebendo a premiação de: “Mostra de Arte e Cultura Mostra de trabalho científico”

Como trabalhos futuros, sugere-se a investigação do uso do EdiMM em contextos de ensino-aprendizagem, suporte a outras modalidades de interação (como gestos por meio de rastreamento de mãos por meio suportados pelo Kinect ou LeapMotion) e a inclusão de suporte a comunicação síncrona para suportar trabalhos colaborativos em diversos equipamentos.

Agradecimentos

Agradeço as pesquisadoras Dra. Fernanda Maria Pereira Freire e a Dra. Flávia Linhalis Arantes, ambas do NIED/UNICAMP, pelas contribuições e coorientação. Agradeço ao pesquisador Dr. André Constantino da Silva por me orientar e ao IFSP campus Hortolândia pela bolsa concedida em 2016 por meio do programa PIBIFSP e ao CNPq pelo fomento ao projeto N. 462478/2014-9 (Chamada Universal MCTI/CNPq 14/2014), que disponibilizou os equipamentos para a realização deste trabalho.

Referências

ALTY, J.; MCCARTNEY, C. Design of a Multi-Media Presentation System For A Process Control Environment, In: EUROGRAPHICS MULTIMEDIA WORKSHOP, Session 8: Systems, Stockholm, Suécia, 1991.

BARBOSA, S.D.J.B; DA SILVA, B.S. Interação Humano-Computador. Rio de Janeiro: Campus/Elsevier Editora Ltda, 2010. 408 p.

BERNSEN, N.O. Multimodality Theory. In: TZOVARAS, D. (Ed.), Multimodal User Interfaces: From signal to interaction. Berlim, Alemanha: Springer-Verlag Berlin Heidelberg, 2008. p. 5-28.

BERNSEN, N.O; DYKJÆR, L. Multimodal Usability. Berlim, Alemanha: Springer-Verlag Berlin Heidelberg, 2010. 431 p.

DA SILVA, A. C. Interação Multimodal em Ambientes de EaD: proposta de arquitetura e impactos. 2014. Tese (Doutorado em Computação) – Instituto de Computação, Universidade Estadual de Campinas, Campinas, Brasil.

GITHUBHELP. What are the different access permissions?. Disponivel em: https://help.github.com/articles/what-are-the-different-access-permissions/. Acesso 13 de dezembro de 2016.

GITHUBHELP. Permission levels for a user account repository. Disponível em: https://help.github.com/articles/permission-levels-for-a-user-account-repository/. Acesso 14 de dezembro de 2016.

ISO 9241-210. ISO Human-centred design for interactive systems. 2010.

LIMA, J.R.; CAPITÃO, Z. e-Learning e e-Conteúdos: aplicações das teorias tradicionais e modernas de ensino e aprendizagem à organização e estruturação de e-cursos. Famalicão, Portugal: Centro Atlântico, 2003. 287 p.

KARPOV, A.; CARBINI, S.; RONZHIN, A.; VIALLET, J.E. Two SIMILAR Different Speech and Gestures Multimodal Interfaces. In: TZOVARAS, D. (Ed.), Multimodal User Interfaces: From signal to interaction. Berlim, Alemanha: Springer-Verlag Berlin Heidelberg, 2008. p. 155-184.

MARCOTTE, E. Responsive Web Design. A Book Apart, 2011. 150 p.

NIELSEN, J. Usability Engineering. EUA: Morgan Kaufmann, 1993. 362 p.

NIGAY, L.; COUTAZ, J. A Generic Platform for Addressing the Multimodal Challenge. In: SIGCHI CONFERENCE ON HUMAN FACTORS IN COMPUTING SYSTEMS, 13., 1995, Colorado, EUA. Proceedings of... Nova Iorque, EUA: ACM Press / Addison-Wesley Publishing Co., 1995. p. 98-105.

NORMAN, D. A.; NIELSEN, J. Gestural Interfaces: A Step Backward In Usability. Interactions, v. 17, n. 5, p. 46-49, 2010.

PREECE, J.; ROGERS, Y.; SHARP, H.; BENYON, D.; HOLLAND, S.; CAREY, T. Human-Computer Interaction: Methods for User-Centred Design. Addison-Wesley. 1994.

ROCHA, H.; BARANAUSKAS, C. Design e Avaliação de Interfaces Humano-Computador. Núcleo de Informática Aplicada à Educação, UNICAMP, 2003.

SHNEIDERMAN, B. O Laptop de Leonardo. Rio de Janeiro: Nova Fronteira. 2006. 228 p.

SHNEIDERMAN, B.; PLAISANT, C. Designing the User Interface: Strategies for Effective Human-Computer Interaction, 4. edição, Reading, MA, EUA: Addison-Wesley. 2004. 672 p.

SUNG, M.; GIPS, J.; EAGLE, N.; MADAN, A.; CANEEL, R.; DEVAUL, R.; BONSEN, J.; PENTLAND, A. Mobile-IT Education (MIT. EDU): m-learning applications for classroom settings. Journal of Computer Assisted Learning, v. 21, p. 229–237, maio 2005.

TEICHE, A.; RAI, A. K.; YANC, C.; MOORE, C.; SOLMS, D.; ÇETIN, G.; RIGGIO, J.; RAMSEYER, N.; D´INTINO, P.; MULLER, L.; KHOSHABEH, R.; BEDI, R.; BINTAHIR, M.T.; HANSEN, T.; ROTH, T.; SANDLER, S. Multitouch Technologies. NUI Group. 2009. 90 p. Disponível em: http://nuigroup.com/log/nuigroup_book_1/