Curso de Sistemas de Informação
Universidade Estadual de Mato Grosso do Sul

Serviço Web para edição de Código Fonte

James Gotardi Castilho

Prof. Msc. Diogo Fernando Trevisan(Orientador)
Prof. Esp. Marcos Alexandre Matos Marques (Co-orientador)

Dourados -MS
Novembro de 2015

Curso de Sistemas de Informação
Universidade Estadual de Mato Grosso do Sul

Serviço Web para edição de Código Fonte

James Gotardi Castilho

Novembro de 2015

Banca Examinadora:

Prof. Msc. Diogo Fernando Trevisan (Orientador)
Área de Computação - UEMS

Prof. Msc. Jéssica Bassani de Oliveira
Área de Computação - UEMS

Prof. Alcione Ferreira
Área de Computação - UEMS

Serviço Web para edição de Código Fonte

James Gotardi Castilho

Este exemplar corresponde à redação final da monografia da disciplina Projeto Final de Curso II devidamente corrigida e defendida por James Gotardi Castilho e aprovada pela Banca Examinadora, como parte dos requisitos para a obtenção do título de Bacharel em Sistemas de Informação.

Dourados, 18 de novembro de 2015.

Prof. Msc. Diogo Fernando Trevisan
(Orientador)

Prof. Esp. Marcos Alexandre Matos Marques
(Co-orientador)

Dedicatória

Dedico este trabalho ao meu avô
Jovelino Castilho (*in Memorian*)

Agradecimentos

Agradeço primeiramente a Deus por me proporcionar sabedoria e paciência para concluir minha graduação, ao meu orientador professor Diogo que me auxiliou em muito na conclusão deste projeto, ao meu co-orientador professor Marcos que mesmo não estando mais na UEMS me permitiu a possibilidade de criar um projeto como o que foi criado.

Não posso deixar de agradecer minha família, a qual esteve presente nestes anos, meu Pai James, minha Mãe Maria, meu Irmão Paulo, a minha amada esposa Polyana que teve muita paciência comigo e que muitas vezes passou noites acordadas apenas para não me deixar sozinho estudando e ao meu filho Felipe que mesmo pequeno ainda e sem entender muito me deu força para continuar.

Aos meus amigos, amigos que levarei para sempre e que estiveram a todo momento ao meu lado: Adrian, Ana, Antonio (Nugoli), Cassiano, Fábio, Jackson, Jónison(prof), Rafael, Vinicius, e tantos outros.

Por fim agradeço a todos que fizeram parte da minha vida nesses anos de graduação.

Resumo

O mundo dos negócios vem passando por grandes mudanças, e uma delas esta relacionada com a facilidade de manipulação e armazenamento de arquivos digitais. Empresas de desenvolvimento de *software* e até mesmo profissionais liberais necessitam de boas ferramentas de programação em seu dia a dia. No mercado existem inúmeros *softwares* de edição de códigos, porém a maioria são de instalação e armazenamento local.O desenvolvimento de uma ferramenta *web* que permita criar, editar e armazenar códigos em servidores na nuvem, muda o foco de trabalho.

A necessidade de estar conectado a todo momento, faz com que seja necessário o desenvolvimento de aplicações que deem essa liberdade ao usuário, de poder realizar seu trabalho em qualquer lugar, seja em um computador pessoal ou em um *smartphone*.

O objetivo desse projeto foi exatamente esse, o desenvolvimento de uma ferramenta *on-line* que possibilite ao programador trabalhar onde estiver, permitindo que o mesmo utilize computadores, notebooks ou *smartphones*, mesmo não sendo de sua propriedade, para acessar seus projetos onde estiver. Para isso foi necessário um estudo de diversas linguagens e *frameworks*.O sistema foi desenvolvido utilizando o *PHP* como base para maioria das funções e contou com o auxílio do *JavaScript* para a criação de *layouts*.

Palavras-chave: *Editor On-line*, *Código Fonte*, *Php*, *Código Aberto*.

Sumário

Lista de Siglas

HTML *Hyper Text Markup Language*

CSS *Cascading Style Sheets*

W3C *World Wide Web Consortium*

MIT *Massachusetts Institute of Technology*

GPI *General Public Licence*

SQL *Structure Query Language*

API *Application Programming Interface*

SGBD Sistema de Gerenciamento de Banco de Dados

RF Requisitos Funcionais

RNF Requisitos Não-funcionais

BD Banco de Dados

PPA Página Pessoal de Arquivos

Lista de Tabelas

Lista de Figuras

Capítulo 1

Introdução

1.1 Introdução

Muito do que diz respeito à tecnologia está relacionado à *Internet*. Empresas têm ampliado os investimentos em conexões mais rápidas e seguras, possibilitando o uso de várias aplicações e serviços *on-line*, paralelos ou distribuídos.

Com essa constante evolução nas conexões de computadores ou outros dispositivos com a *Internet*, as pessoas têm mudado a forma como acessam a rede, pois o que antes era feito apenas por computadores que ficavam em mesas dentro de locais fechados, hoje pode ser feito de qualquer hora e lugar, com o uso de *notebooks, tablets, smartphones* entre outros.

Mas para que seja possível toda essa evolução, não bastam apenas *hardwares*[1] capazes de controlar todos esses processamentos, e sim um conjunto de *hardware* e *software*[2] capazes o suficiente de gerenciar, processar e dar suporte para os desenvolvedores dos sistemas.

É necessário o uso de linguagens de programação que deem o suporte apropriado para que seja possível o desenvolvimento de sistemas e dos *hardwares* necessários para que toda essa infraestrutura de conexão funcione apropriadamente, de modo a satisfazer os anseios dos usuários domésticos e empresariais.

Essas são conhecidas como linguagens de alto nível, pois possibilitam ao programador manipular de forma mais fácil e rápida seus códigos, permitido que o foco maior seja a lógica de programação.

1.2 Objetivos

O objetivo desse projeto foi modelar uma ferramenta *web* que permita aos usuários desenvolver e armazenar códigos fonte *on-line*, afim de alcançar alguns objetivos específicos como.

[1] Parte física dos dispositivos, placas, periféricos, *chips*.

[2] Programas e aplicativos instalados em dispositivos ou computadores.

1.2.1 Objetivos Específicos

- Oferecer ao usuário opções de escolher a linguagens de programação para edição;
- Ser compatível com os navegadores presentes no mercado atual;
- Salvar os arquivos em um servidor.

1.3 Motivação

Empresas como *Google* e *Microsoft*, gigantes no ramo da computação e *internet*, disponibilizam ferramentas como *Google Drive* e *Onedrive*, que fornecem armazenamento de arquivos em nuvem. Para utilizar essas ferramentas, os usuários necessitam de uma conta e através desta, os mesmos podem fazer *upload*[3] ou *download*[4] de seus arquivos onde estiverem.

Com esse novo ramo de mercado, o armazenamento de arquivos em nuvem, surgiu também empresas especializadas nesse tipo de serviço, como o *Dropbox* e o *Mega*, ambas com a mesma finalidade.

Essa nova visão de manipulação de arquivos, motivou a elaboração de uma ferramenta *web* voltada para área de programação, cuja finalidade é além de armazenar arquivos possa também oportunizar ao usuário desenvolver seus códigos *on-line*.

1.4 Metodologia

Para o desenvolvimento desse sistema, foi feita uma pesquisa bibliográfica, a fim de levantar os conceitos e as características das linguagens e ferramentas a serem utilizadas no desenvolvimento do sistema.

O sistema é compatível com os navegadores atuais do mercado, dando liberdade ao programador trabalhar com seu preferido. O usuário tem a possibilidade de desenvolver códigos utilizando as linguagens *PHP,HTML,CSS,JS*, além de arquivos do tipo *txt* e posteriormente outras linguagens que serão incorporadas ao projeto.

A ferramenta aqui proposta, foi desenvolvida utilizando *HTML,CSS e PHP* junto com as bibliotecas *JavaScript JQuery e JQuaey UI* além do *framework Codemirror*. O *layout*[5] é feito com *HTML* controlado pelo *CSS*.A validação das entradas é controlada pelo *PHP*, o editor foi criado com base no *Codemirror* e o *JavaScript*, juntamente com suas bibliotecas *JQuery e JQuaey UI* são responsáveis pela árvore de arquivos mostrada na página do editor, possibilitando a escolha do arquivo para edição.O Banco de Dados é gerenciado pelo *MySql*.

[3]Transferir dados de um computador local para um servidor ou outro em uma rede.
[4]Transferir dados do servidor para o computador.
[5]Desenho da página

1.5 Organização dos Capítulos

O texto do projeto está organizado em um único volume, contendo listas de figuras e tabelas. Além deste Capítulo, o volume é organizado em outros 3 capítulos e a conclusão, cujos conteúdos são apresentados a seguir.

No **Capítulo 2**, é apresentado um conceito básico sobre as linguagens utilizadas na elaboração do presente projeto. No **capítulo 3**, é explanado sobre o desenvolvimento do projeto, o motivo de utilizar as linguagens escolhidas, como se dará a utilização pelo usuário e como o sistema se comportará no servidor. O **Capítulo 4** trata sobre os testes realizados com o editor. O **Capitulo 5** é a conclusão.

Capítulo 2

Revisão Bibliográfica

Nesse capítulo são apresentados conceitos de linguagens e *frameworks* para o desenvolvimento do projeto. As linguagens aqui mostradas são a base para a implementação do projeto proposto.

2.1 *HTML*

A *Hyper Text Markup Language* (HTML) trata-se de uma linguagem de marcação, que é interpretada por um *software* específico (navegador *web*), mostrando na tela do computador a página da com suas características.

Um documento *HTML* é criado utilizando um editor de código fonte, ou mesmo em um editor de texto. Para ser interpretado, faz-se o uso de *tags*, segundo FREEMAN and FREEMAN (2006), as *tags* nada mais são que palavras ou caracteres escritos entre dois sinais, um de menor(<) outro de maior(>).

As *tags* no documento *HTML* servem para mostrar ao navegador *web* o que ele precisa interpretar, ou seja, elas definem o que é exibido por um *browser*[6]. Elas podem estar dentro de outras *tags* e também receber atributos que definem as propriedades da página.

Os atributos em um documento *HTML*, servem para definir as propriedades de uma *tag* e sempre são colocadas na *tag* de abertura, são compostos por um nome e um valor.

A **Tabela 2.1** mostra alguns exemplos.

[6]Navegador *web*.

Tags de um Documento *HTML*	
Tag	**Significado**
<html> ... </html>	Informa início e o fim do Html
<head> ... </head>	Inicia e termina o cabeçalho
<title> ... </title>	Define um Título para a página
<body> ... </body>	Define o corpo da página
...	Inicia e termina um parágrafo
<h1,h2, ... ,h5> ... </h1,/h2, ... ,/h5>	Define um Título, um subtítulo
	Insere uma imagem na página

Tabela 2.1: Tags *HTML*.

Quando uma página da *internet* é criada, ela não é composta apenas de *HTML*, mas sim de outras linguagens que ajudam o *site* a ficar com uma aparência mais atraente ao usuário e mais funcional também. Geralmente são usados elementos da linguagem *CSS, JavaScript e Frameworks*, que serão abordados em outros tópicos.

Segundo FREEMAN and FREEMAN (2006), o *HTML* surgiu no início da década de 90, criado pelo inglês *Tim Berners-Lee*. A intenção de *Lee* era criar uma linguagem em que fosse possível fazer a comunicação entre computadores do laboratório e outras instituições de ensino e também para exibir documentos científicos de forma simples e objetiva.

Desde sua criação até hoje, o *HTML* passou por algumas versões, em cada uma delas recebendo melhoras, para que cada vez mais os navegadores *web* pudessem ser compatíveis com o *HTML* sem a necessidade de ter que modificar o código a cada nova versão. Na **Tabela 2.2** é apresentada a evolução do *HTML*.

Histórico de versões do *HTML*	
Ano	**Versão**
1991	*Tim Berners-Lee* cria o *HTML*
1992	Com a liberação de bibliotecas, o *HTML* surge para o mundo
1993	Cria-se padrões para as principais características do *HTML*, surgindo à versão 2.0
1995	Surge o *HTML* 3.0, uma extensão da versão 2.0
1997	*HTML* 3.2, *Netscape e Internet Explorer* desenvolvem seus próprios padrões
1998	Nasce o *HTML* 4.0, com muitos recursos introduzidos, entre eles *Css e JavaScript*
1999	Entra no mercado o *HTML* 4.01, pequenas modificações em relação à versão anterior
2008	Começa a surgir o *HTML* 5
2014	É lançado o *HTML* 5, no dia 29 de Outubro

Tabela 2.2: Histórico do *HTML*.

2.2 CSS

O *Cascading Style Sheets* (CSS) formata toda a informação, desde imagens, vídeos, áudio e texto entregues pelo documento *HTML*. Com o *CSS* é possível redimensionar fontes, definir configurações para partes específicas do *HTML*, como por exemplo, definir uma configuração para o *layout* do *site*.

Mesmo existindo muitas páginas, é necessário apenas que crie um *link*[7] em cada página *HTML* para que o *CSS* entre em ação, sem a necessidade de mudar página uma a uma individualmente.

Com o *CSS* além das páginas *web* ficarem atraentes visualmente, elas também ficaram mais simples de serem manipuladas, com isso o código *HTML* fica mais enxuto e de fácil compreensão para o profissional que posteriormente irá manter a página.

Segundo W3C (2012),com a popularização da *internet*, foi preciso criar alguma forma de deixar os sites mais atrativos para os usuários e menos complexos para os desenvolvedores, pois as páginas eram muito grandes e com *layouts* complexos. O *CSS* veio para deixar o código mais limpo, criado em 1996 pela *World Wide Web Consortium* (W3C). A **Tabela 2.3** mostra o histórico das versões do *CSS*.

Histórico de versões *CSS*	
Ano	**Versão**
1996	*CSS*-1, o W3C lançou a versão oficial do *CSS*, entre suas capacidades poderia formatar os textos, cores e margens
1997	*CSS*-P, essa versão quando lançada tinha o propósito de posicionar os elementos em uma página *HTML* da forma mais precisa possível, então o W3C lançou essa versão intermediária do *CSS*
1998	*CSS*-2, engloba todo o conteúdo das duas primeiras versões, porém dando mais ênfase na facilidade de acesso e também na capacidade do *CSS* de se adaptar a qualquer tipo de mídia, seja ela visual ou auditiva. Essa versão ainda é utilizada
Atual	*CSS*-3, a terceira versão do *CSS* ainda está em desenvolvimento, sabe-se que uma das adições será o formato *SVG*, que permitirá a inclusão de novas formas nos documentos *Html*, como círculos, curvas e outras formas.

Tabela 2.3: Histórico do *CSS*.

[7]Ligação de uma página com outra.

2.3 PHP

Segundo PHP.NET (2015), o *PHP* é uma linguagem de *script* muito utilizada por desenvolvedores especialmente adequada para *web*. Em geral o *PHP* é embutido dentro do *HTML*. Abaixo um exemplo de uma página *HTML* com o *PHP* embutido.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
    <head>
        <title>Exemplo</title>
    </head>
    <body>
        <!-- esse e um comentario html -->
        <?php

        //esse e um comentario PHP
        echo "Ola, eu sou um script PHP!";
        /*esse tambem e um comentario PHP*/
        ?>

    </body>
</html>
```

Seu nome é um acrônimo recursivo de ***P**HP:**H**ypertext **P**repocessor*(Processador de Hipertextos *PHP*). Muito de sua sintaxe é derivada de outras linguagens como o *C, Java e Perl* além de características próprias, por esse motivo o tempo de aprendizagem é rápido. De acordo com PHP.NET (2015), o objetivo do *PHP* é permitir que desenvolvedores *web* criem páginas geradas dinamicamente e de forma mais rápida.

Além de ser *open source*, o *PHP* oferece suporte para vários Sistemas Operacionais como *Linux, Windows e MAC OS*. Uma de suas características, segundo PHP.NET (2015), mais forte e significativa é o suporte a vários Bancos de Dados.

Atualmente o *PHP* encontra-se na versão 5.6.14. A **Tabela 2.4** mostra o histórico de versões do *PHP*.

Histórico do *PHP*	
Ano	**Versão**
1994	*PHP Tools*, criado por *Rasmus Lerdof*, o *PHP* originalmente era conhecido por *Personal Home Page Tools*.
1997	*PHP/FI*, é oficialmente lançada a nova versão, podendo ser atribuída o nome de *PHP 2.0*.
1998	*PHP 3.0*, lançada com várias correções da versão anterior agora o *PHP* abria espaço para o *eCommerce*.
2000	*PHP 4.0*, lançada oficialmente no ano de 2000, essa versão trouxe muitas melhorias, além de incluir suporte para maioria dos servidores *webs*, agora o *PHP* conseguia trabalhar com mais eficiência em cima de diversos Bancos de Dados, possibilitando o desenvolvimento de aplicações mais complexas.
2004	*PHP 5.0*, versão atual da linguagem.

Tabela 2.4: Histórico do *PHP*, (PHP.NET (2015)).

2.4 JavaScript

O *JavaScript* nasceu com a finalidade de permitir que desenvolvedores *web* criassem seus *scripts* diretamente no documento *HTML*, conforme diz *Brendan Eich*, no prefácio do livro *JavaScript Bible,Gold Edition.*GOODMAN (2001).

De acordo com FELIX (2005), o *JavaScript* é utilizado principalmente para dar mais dinamismo às páginas *web*, facilitando assim a manipulação de dados em formulários, como, por exemplo, verificar se os campos estão preenchidos e os dados validados corretamente.

Segundo MORRISON (2008), para que uma página *web* tenha mais vida é utilizado o *JavaScript*, pois permite ao usuário obter uma resposta da página.

O *JavaScript* foi desenvolvido pela *Netscape*, de acordo com GRILLO and de MATTOS FORTES (2008), seu nome inicial era *"Mocha"* ,e até antes de seu lançamento em 1995, era conhecida por *LiveScript*, vindo a ser chamada de *JavaScript* apenas antes de ser lançada junto à versão 2.0 do seu navegador.

O nome *JavaScript* segundo MORRISON (2008), deve-se ao fato de ser muito parecida com a sintaxe do *Java*, mesmo estas não tendo relação alguma. Com a similaridade dos nomes, houve muita confusão por parte dos usuários, dificultando assim a expansão da linguagem.

O *JavaScript* é uma linguagem que herdou muitas características do *C, C++* e do *Java*, tornando-a fácil de manipular e com inúmeras possibilidades de programação para os

web sites[8].

A **Tabela 2.5** mostra o histórico de versões do *JavaScript*.

Linha do Tempo *JavaScript*	
Ano	**Versão**
1995	Lançada a primeira versão o *JavaScript* 1.0
1996	Versão 1.1 é lançada
1997	Surge a versão 1.2 da linguagem
1998	A versão 1.4 é lançada no segundo semestre desse ano
2000	Lançada a versão 1.5
2005	Lançada a versão 1.6 no final desse ano
2006	Versão 1.7 da linguagem aparece no mercado
2008	Lançada a versão 1.8 no primeiro semestre
2010	*JavaScript* 1.8.5 é lançada e é a atual versão da linguagem

Tabela 2.5: Histórico do *JavaScript*.

2.5 *Framework*

Segundo MATTSSON (2006), *frameworks* são geradores de códigos utilizados para facilitar o desenvolvimento de aplicações dentro de uma aplicação que necessite desse recurso, sem a necessidade de reescrever o código do início, e assim fazendo o uso de códigos prontos, fazendo apenas alterações necessárias ao desenvolvimento.

2.5.1 Codemirror

De acordo com o site CODEMIRROR (2015), *Codemirror* é um editor de texto versátil implementado em *JavaScript* para o navegador. É uma ferramenta pública de código aberto que permite a configurar e utilizar conforme a necessidade do desenvolvedor do projeto.

Uma rica *Application Programming Interface* (API) de programação e permite selecionar temas diferentes para personalizar o *CodeMirror*, ajustando conforme a necessidade do aplicativo, e estendendo-o com novas funcionalidades.

No **Capitulo 3** é explicado a razão de usar o *Codemirror* e como foi utilizado no desenvolvimento do projeto.

[8]Páginas com conteúdo disponível na Internet.

2.6 Bibliotecas

São uma coleção de subprogramas que são utilizados pelos desenvolvedores ao criar um sistema, pois possuem dados e códigos auxiliares, que permitem trabalhar de maneira mais eficiente.

2.6.1 JQuery

Segundo SILVA (2011), *JQuery* é uma biblioteca *JavaScript*, oferecida como um *software* livre e aberto, sendo assim, pode ser utilizada por qualquer programador para desenvolvimento de projetos pessoais ou comerciais. Ele, por ter código livre e aberto, segue as regras estabelecidas pelo *Massachusetts Institute of Technology* (MIT)[9]e pelo *General Public Licence* (GPI)[10].

O *JQuery* foi criado por *Jhon Resing* em 2005, conforme o mesmo diz, no prefácio do livro *JQuery in Action*[11] (BIBEAULT and KATZ (2010)), a essência do *JQuery* é ser uma forma simples de se programar usando o *JavaScript*, tanto para desenvolvedores experientes como iniciantes na área *web*.

Seu objetivo é adicionar uma interatividade maior com a página *web*, deixando-a mais acessível, tornando uma experiência agradável para o usuário. Ele é usado para modificar a interface da página, desde efeitos visuais a alterar conteúdos.

De acordo com SILVA (2011), desde sua criação, o *JQuery* visa manter uma conformidade com os padrões da *web*, sendo acessível por qualquer navegador que suportar o *CSS-3*, mas para que esse código esteja de acordo com os padrões é preciso que o desenvolvedor os crie em acordo com o *W3C*.

2.6.2 JQuery UI

O *JQuery UI*,foi criado em 2007 por um grupo de desenvolvedores, ele nada mais é do que uma biblioteca *JavaScript*. Segundo SILVA (2012), o *JQuery UI* foi criado com um propósito específico, ser uma biblioteca para o *JQuery*.

A respeito do *JQuery UI* SILVA (2012) página 32, diz que:

> "Seu princípio geral de desenvolvimento adota a mesma filosofia que norteia a biblioteca jQuery, que é: "fazer mais escrevendo menos"."

Conforme SILVA (2012), com a utilização da biblioteca *JQuery UI*, o desenvolvedor tem a possibilidade de criar interfaces para o usuário com o mínimo de código possível, porém mantendo uma similaridade com a sintaxe do *JQuery*.

[9]Instituto de Tecnologia de *Massachusetts* nos Estados Unidos.
[10]Licença Pública Geral para distribuição de *software*
[11]JQuery em Ação.

2.7 Banco de Dados

De acordo com MILANI (2010), um banco de dados é um conjunto de dados organizados em tabelas e separados por assuntos afim de tornar a vida do usuário mais prática, rápida e confiável. Segundo RIBEIRO (2004), um banco de dados (ou base de dados) é uma coleção de dados logicamente relacionados, com algum significado. Ainda de acordo com CÁCERES (2012), um banco de dados representa abstratamente uma parte do mundo real, que é de interesse de uma determinada aplicação.

Várias são as definições de banco de dados pelos autores, mas pode-se fazer uma analogia com um Banco financeiro convencional, onde sempre que é necessário algum serviço especifico como empréstimo, saque ou financiamento é o local apropriado para tal necessidade, um Banco de dados funciona da mesma maneira, ele guarda informações relevantes a determinada atividade e sempre que é necessário buscar, inserir ou excluir alguma informação é no BD que é feita e organizada essas e outras operações.

2.7.1 Linguagem *SQL*

De acordo com MILANI (2010), o *Structure Query Language* (SQL)[12] é uma linguagem utilizada para realizar as interações com dados armazenados pela maioria dos bancos de dados relacionais existentes.

Por ser uma linguagem pensada para consulta aos dados, ferramentas como o *MySql e PhpMyAdmin* a utilizam para interagir com o BD. Dessa mesma forma o *PHP* para realizar interações com o banco de dados, como excluir, adicionar, buscar ou alterar informações, utiliza-se dessa linguagem.

Em geral, possui três recursos básicos para efetuar diversas interações com o BD, são elas:

- Consultas:
  - São realizadas para obter informações armazenadas no banco de dados.
- Atualizações:
  - São realizadas três tipos de atualização: Inclusão de dados, manutenção/atualização de dados e a exclusão de dados do BD.
- Filtros e ordenações:
  - Permite retornar buscas especificas para o usuário, ordenando os resultados de acordo com um modelo pré-estabelecido.

A seguir algumas instruções *SQL*.

[12]Linguagem Estruturada de Consulta

```
1 INSERT INTO <tabela> (<campo1>, <campo2>) VALUES (<valor1>,<valor2>)
2
3 UPDATE <tabela> SET <campo> = <valor> (WHERE <condicoes>)
4
5 SELECT <campo> FROM <tabela> ORDER BY <campo/criterio> [ASC|DESC]
6
7 DELETE FROM <tabela> [WHERE <condicoes>]
```

2.7.2 MySQL

Histórico

Nos anos 90 quando três desenvolvedores, (*David Axmark, Allan Larsson e Michael ŞMontyŤ Widenius*),necessitavam de uma interface *SQL* que fosse compatível com as rotinas *ISAM*[13] que eram utilizadas em suas tabelas e aplicações, acabaram por desenvolver o *MySql*.Segundo MILANI (2007), os desenvolvedores tentaram utilizar a *API mSQL*, porém não era rápida o suficiente, então utilizando dessa *API* como base, escreveram em *C e C++* uma nova o *MySql* que teve sua primeira versão lançada em 1996.

Com o ótimo resultado obtido por essa nova *API*, o *MySQL* começou a ser difundido e seus criadores fundaram a empresa responsável por sua manutenção, que é a *MySQL AB*.

O *MySQL* se tornou muito conhecido e popular sendo cada vez mais utilizado.Como qualquer sistema ou programa, novas versões foram lançadas sempre corrigindo erros e adequando-se as necessidades dos usuários.Sua mais recente versão é a 5.7.6.

O Banco de Dados *MySQL*

De acordo com de SOUZA (2014), o *MySQL* é um dos Sistema de Gerenciamento de Banco de Dados (SGBD) mais popular, por ele ser otimizado para *Web*, é amplamente utilizado para criação de sistemas *on-lines* que necessitem de armazenamento de informações, como *sites* de *e-comerce*.

Entre suas características,ainda segundo de SOUZA (2014) estão:

- Compatibilidade com linguagens como *PHP, Java, Python, C#, Ruby e C/C++*;
- Baixa exigência de processamento (em comparação como outros *SGBD*);
- Vários sistemas de armazenamento de dados (*tatabase engine*), como *MyISAM, MySQL Cluster, CSV, Merge, InnoDB*, entre outros;
- Recursos como *transactions* (transações), conectividade segura, indexação de campos de texto, replicação, etc;
- Instruções em *SQL* como indicam o nome.

[13] *Indexed Sequential Access Method*, Metódo de acesso sequencial indexado

O *MySQL* foi utilizado por ser compatível com diversos Sistemas Operacionais presentes no mercado atual, pois por ser baseado em *C/C++* ele é facilmente portável entre plataformas.

Capítulo 3

Desenvolvimento

Neste capítulo são apresentados Análise de requisitos, fluxogramas que auxiliam na compreensão do projeto desenvolvido. É também apresentada as telas do sistema.

Para criação e configuração do Banco de Dados e servidor Apache foi utilizado o *Xampp*. Além do programa citado, também foi utilizado o *notepad++* para a editar os arquivos e a ferramenta *DIA* para criar os fluxogramas.

3.1 Análise de Requisitos

Segundo SOMMERVILLE (2006), um requisito pode ser uma descrição detalhada e de alto nível de um sistema ou serviço e também algumas especificações técnicas de sistemas.

Uma Análise de Requisitos consiste em identificar os requisitos funcionais e não-funcionais do sistema, de forma que os interesses do cliente sejam compreendidos da melhor forma possível, proporcionando ao programador informações mais precisas, facilitando o processo de codificação do sistema.

Devido o aumento expressivo das tecnologias e a necessidade por mobilidade foi proposto um sistema *on-line* cuja finalidade é possibilitar aos usuários do sistema utilizar um editor de código-fonte em qualquer lugar a qualquer hora, desde que este possua os mecanismos necessários para tal finalidade.

3.1.1 Requisitos Funcionais

Requisitos Funcionais é uma maneira de informar ao desenvolvedor quais as tarefas que o sistema deverá apresentar, essas que são indispensáveis ao funcionamento correto depois de concluído.

Lista com os Requisitos Funcionais (RF).

- RF1 Permitir o cadastro usuários;
- RF2 Permitir o acesso a usuários já cadastrados;

- RF3 Permitir criar novos projetos;
- RF4 Permitir editar projetos criados;
- RF5 Permitir salvar projetos em uma pasta no servidor;
- RF6 Permitir criar novos documentos;
- RF7 Permitir salvar seus códigos em uma pasta no servidor;
- RF8 Permitir editar documentos em uma pasta no servidor;
- RF9 Permitir escolher a linguagem de programação apropriada para seu código;
- RF10 O editor deve identificar palavras reservadas da linguagem escolhida pelo usuário;
- RF11 Permitir criar quantos documentos forem necessários para o projeto;
- RF12 Informar ao usuário sempre que existirem erros na hora de criar, salvar ou editar um arquivo ou projeto;

3.1.2 Requisitos Não-funcionais

De acordo com SOMMERVILLE (2006), os requisitos não-funcionais informam restrições do sistema ou funções que lhe atribuem algum valor, porém não são tão importantes quanto os Requisitos Funcionais.

Lista com os Requisitos Não-funcionais (RNF).

- RNF1 É preciso utilizar informações válidas quando realizar o cadastro;
- RNF2 Vários usuários podem ser cadastrados no *site*;
- RNF3 O usuário tem direito a apenas um cadastro por *e-mail*;

3.2 Fluxograma

Nas **Figuras 3.1 - 3.4** , são mostrados os fluxogramas que exemplificam o fluxo de informações no sistema em determinadas situações.

A **Figura 3.1** mostra o fluxo de informações que estarão presentes na *home page* do site.

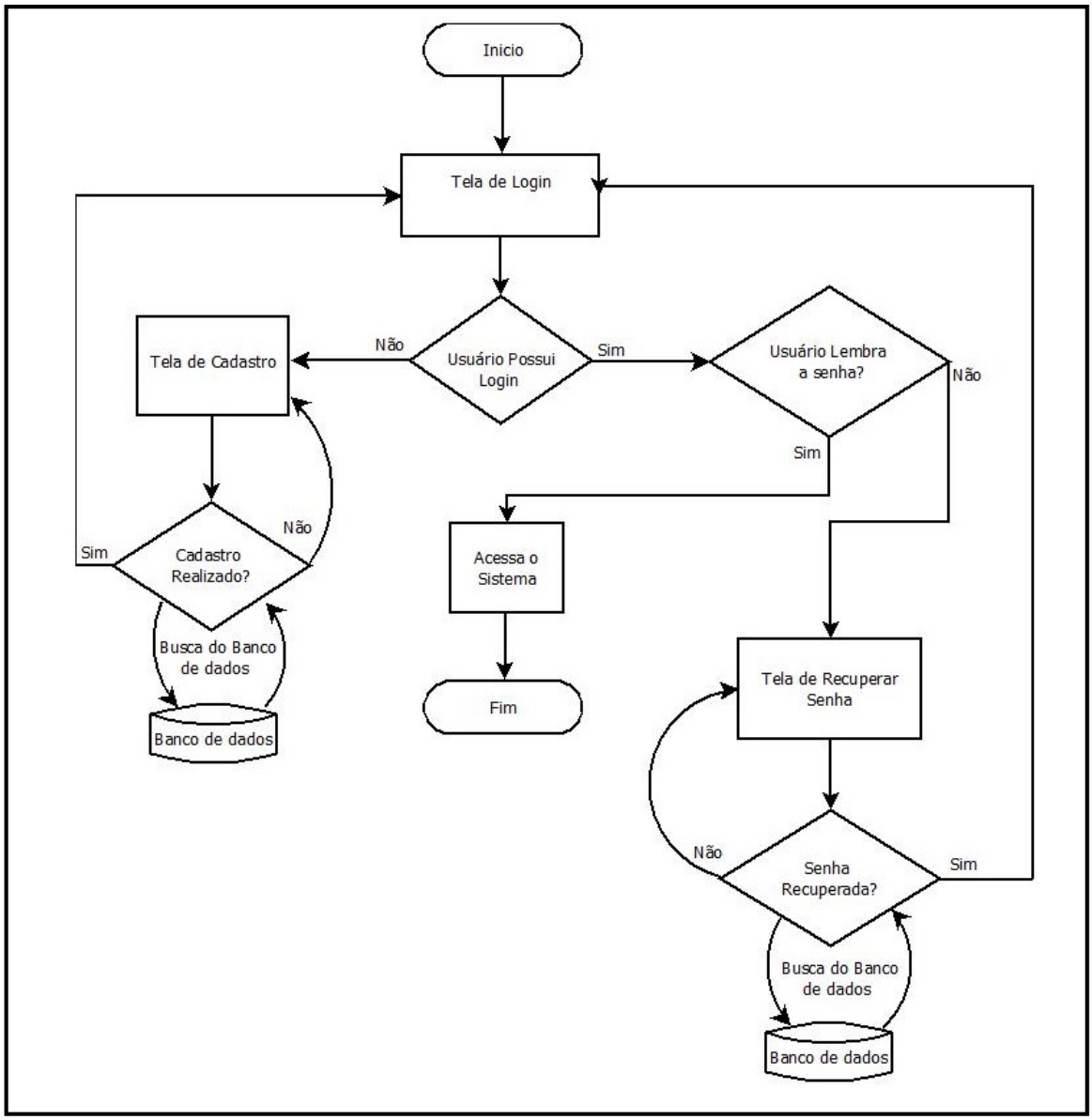


Figura 3.1: Fluxograma da Tela Inicial.

A **Figura 3.2** mostra o fluxo das informações no momento em que o usuário pode criar ou editar um projeto já salvo em sua pasta pessoal do servidor do sistema.

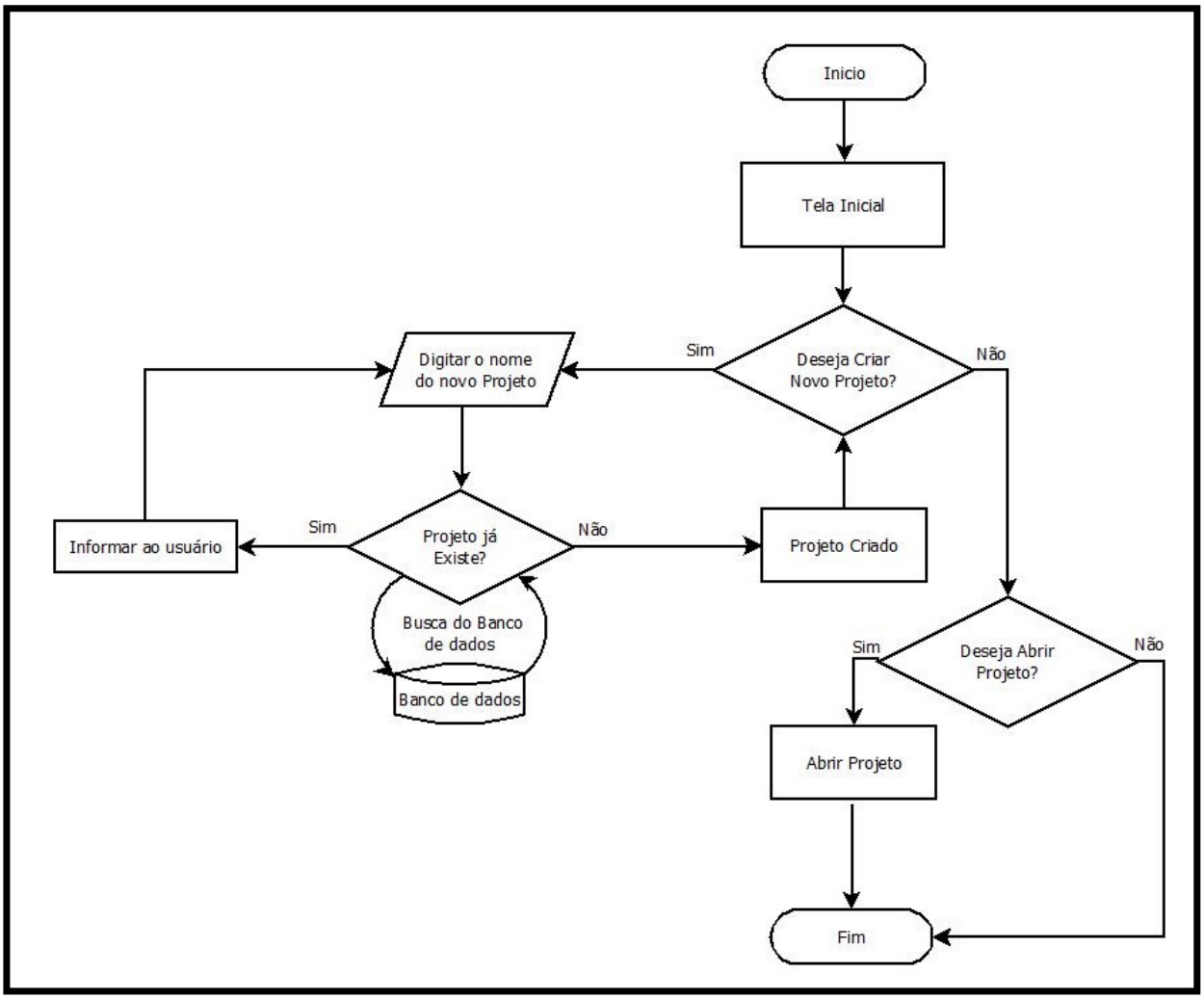


Figura 3.2: Fluxograma de Criação ou Edição de um projeto.

A **Figura 3.3** mostra o fluxo das informações no momento em que usuário escolhe entre criar um novo arquivo ou simplesmente abrir um arquivo já criado no editor.

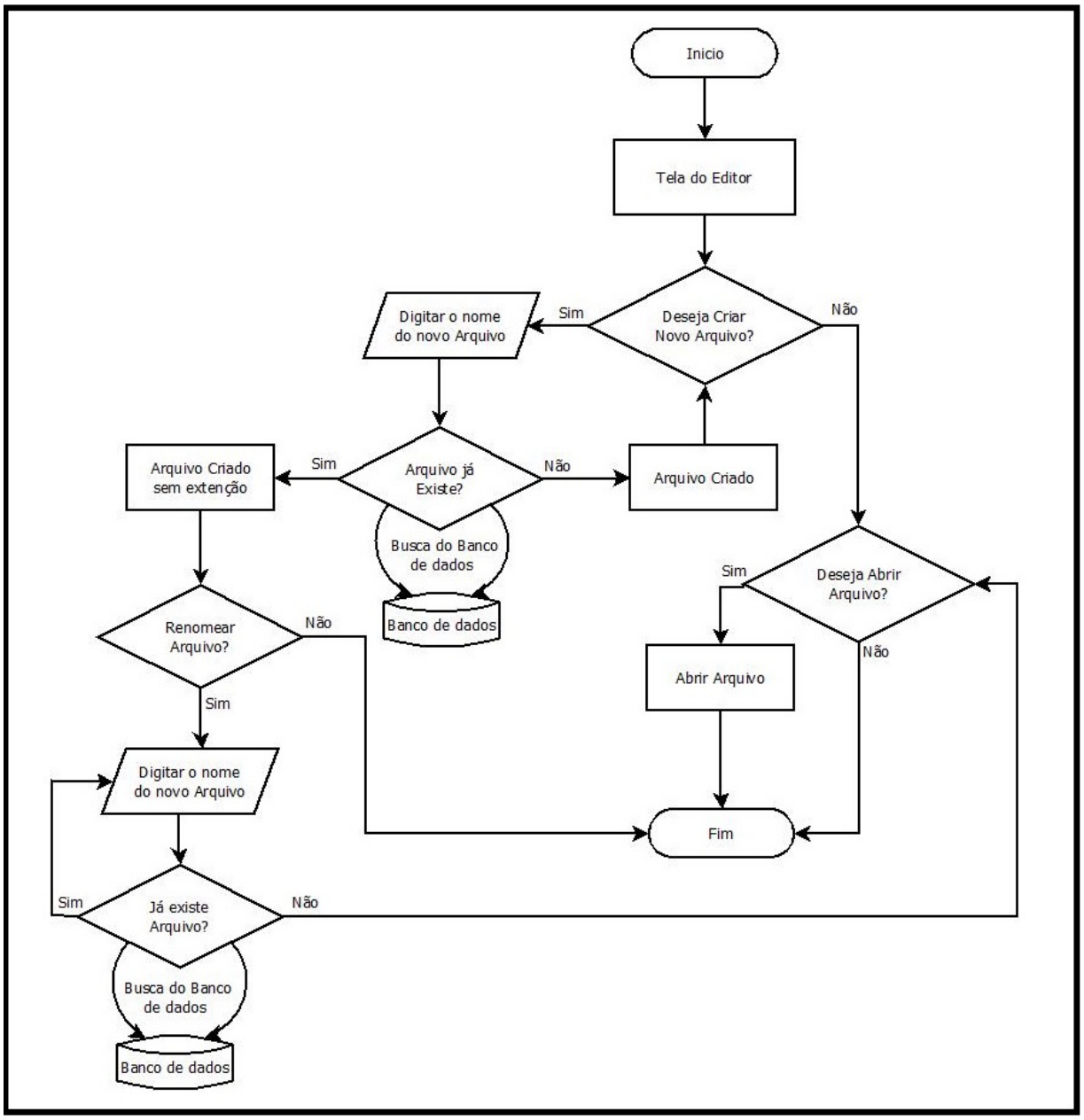


Figura 3.3: Fluxograma de Criação de Arquivo.

A **Figura 3.4** mostra o fluxo das informações no momento em que usuário abre o arquivo selecionado no editor.

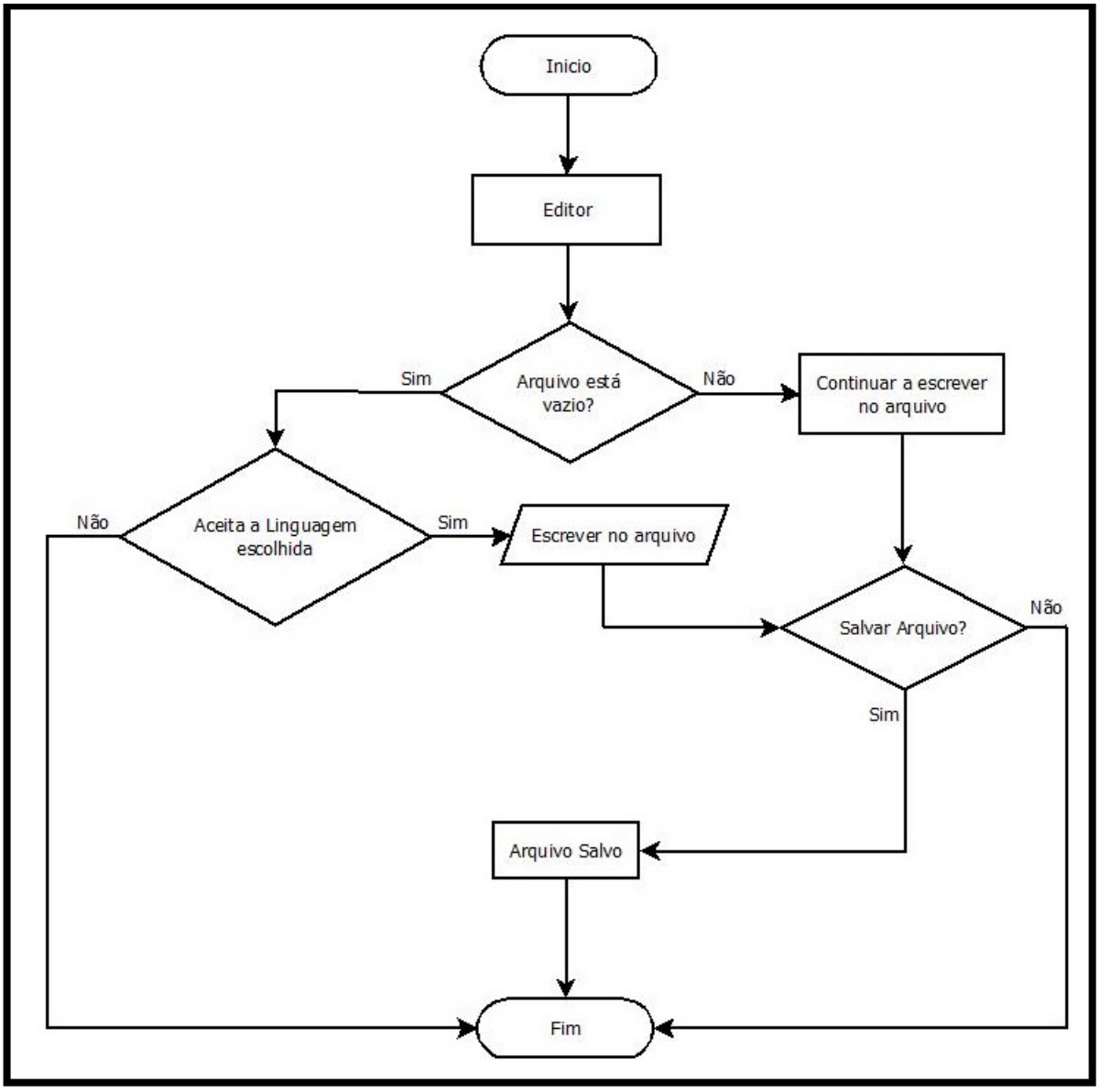


Figura 3.4: Fluxograma de Edição de Arquivo.

3.3 Funções do Sistema

Nesta sessão serão descritas as principais funções contidas no sistema.

3.3.1 Função Cadastrar

O arquivo caduser.php é responsável pelo cadastro do usuário no *site*.

Ao acessar a página, o usuário poderá cadastrar-se preenchendo um formulário com seus dados, Nome de Usuário, *E-mail* e criando uma senha. Após o preenchimento desse formulário ao pressionar o botão "CADASTRAR" o usuário é redirecionado para a página inicial onde se encontra a tela de *login* para que possa realizar o acesso ao *site*.

A **Figura 3.5** mostra a tela de Cadastro.

Figura 3.5: Tela de Cadastro no Site.

O cadastro do usuário é dividido em partes, é feita uma busca no Banco de Dados para verificar a existência de Nome de Usuário ou *E-mail* já existentes, é feita uma comparação das senhas digitadas e por fim é criada a pasta com o nome do usuário, onde ficarão seus arquivos futuros.

3.3.2 Função *Login*

O arquivo validalogin.php é a responsável por validar as informações fornecidas pelo usuário.

Para que se possa ter acesso a todas as funcionalidades do sistema, é preciso que o usuário realize o *login* no site, preenchendo um formulário com seus dados de usuário que são nome de usuário e senha. Ao pressionar o botão "OK", é enviado uma requisição ao servidor comparando o nome de usuário e a senha fornecidos com os que estão no BD, ao confirmar as informações o usuário é redirecionado para sua Página Pessoal de Arquivos (PPA), onde ele terá acesso a todos seus arquivos, projetos e também ao editor.

A **Figura 3.6** mostra a tela de *Login*.

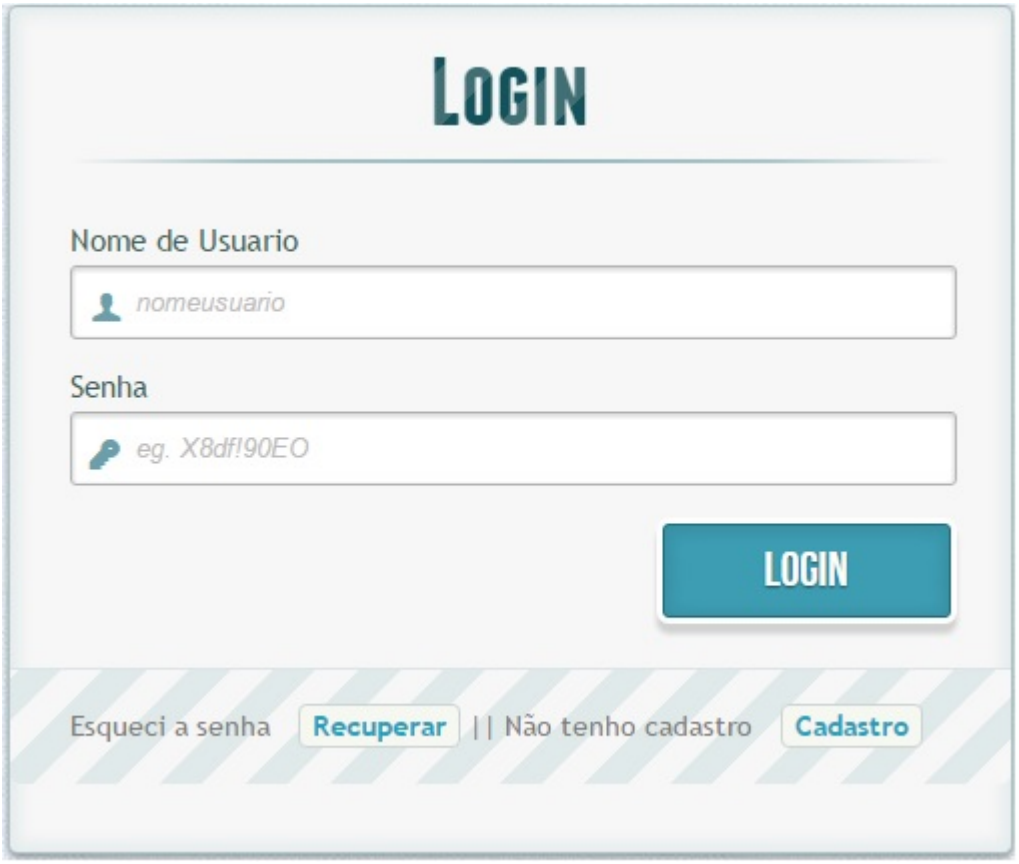


Figura 3.6: Tela de *Login* no Site.

3.3.3 Função recuperar senha

O arquivo recupera.php é responsável por enviar um *E-mail* para o usuário com seu nome de usuário e senha caso o mesmo tenha esquecido.

Para que seja realizada essa operação o usuário informa seus dados, é feita uma busca no BD e caso estejam corretos é enviada uma mensagem, para o *E-mail* cadastrado, com

seus dados de acesso caso contrário e informado que o usuário não é cadastrado.

A **Figura 3.7** mostra a tela de recuperar o acesso.

Figura 3.7: Tela de recuperação de acesso.

3.3.4 Acesso ao Banco de Dados

Para que seja realizado o cadastro/*login* e a recuperação de senha é necessário que haja uma conexão com o Banco de Dados, essa conexão é realizada pelo arquivo config.php que é responsável por verificar a conexão com o servidor do Banco e com o Banco de Dados, caso não haja conexão é informado o erro, seja ele qual for, como por exemplo falta de acesso a internet.

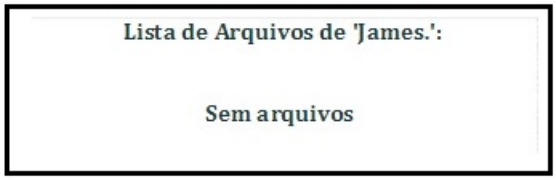
3.3.5 Função Criar Projeto

O arquivo criaprojeto.php é chamada sempre que o usuário decide iniciar um novo projeto. Dentro da pasta do usuário no servidor é criada uma nova pasta com o nome do novo projeto, podendo ser renomeada posteriormente.

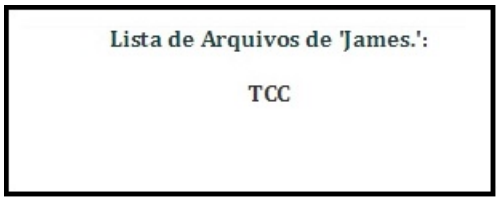
A **Figura 3.8**, basta escrever o nome e clicar no botão "Criar", caso já exista um projeto com o nome digitado é informado ao usuário.

Figura 3.8: Espaço para criação de novo projeto.

Após o usuário acessar o sistema, na página inicial ao lado esquerdo é apresentada uma lista de seus projetos, caso haja algum já criado, como mostrado na **Figura 3.9 (b)** , caso o usuário não tenha nenhum projeto ainda sua lista de projetos estará vazia, como a **Figura 3.9 (a)**.



(a) Lista de Projetos vazia



(b) Lista de projetos criados

Figura 3.9: Lista de Projetos

3.3.6 Permissão de acesso

Para que o usuário possa acessar seus projetos e arquivos é necessário que o mesmo tenha autorização para isso, assim evita-se manipular dados que não lhe pertence e também que seus dados sejam acessados por outros.

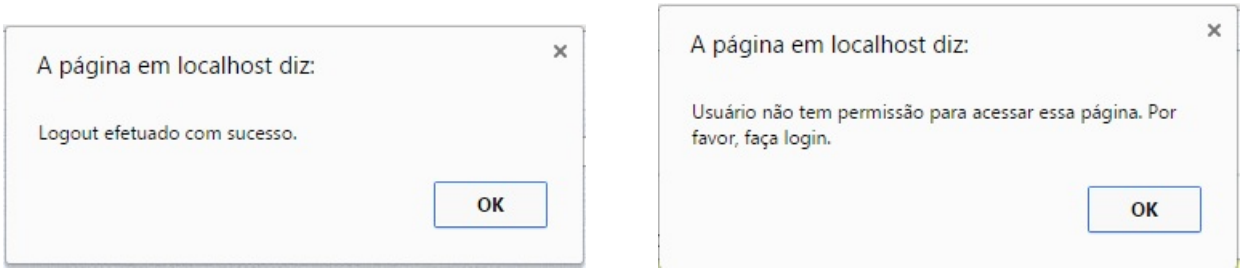
Esse controle é feito pelo arquivo verifica.php, como o próprio nome já diz essa função verifica se o usuário tem permissão de acesso a determinada página, evitando violação de dados.

3.3.7 Função *Logout*

Como em qualquer sistema, após o usuário terminar seu trabalho é efetuado o *logout* no *site*, fazendo com que seus dados permaneçam seguros e impedidos de serem acessados por outros.

O arquivo logout.php é responsável por realizar o encerramento da sessão do usuário impossibilitando que outros utilizadores do mesmo computador tenham acesso a seus arquivos, garantindo a privacidade dos dados.

Essa função destrói a sessão e realiza o *logout* do usuário informando-o sobre a saída bem sucedida, e caso tente voltar a página para ter acesso sem realizar o *login* é informado que o usuário não possui permissão de acesso, esses dois casos são mostrados na **Figura 3.10**.



(a) *Logout realizado na página* (b) Sem permissão de acesso

Figura 3.10: Alertas da função logout.php

3.3.8 Editor

A **Figura 3.11** mostra o Editor carregado com um arquivo e ao lado todos os arquivos contidos no projeto.

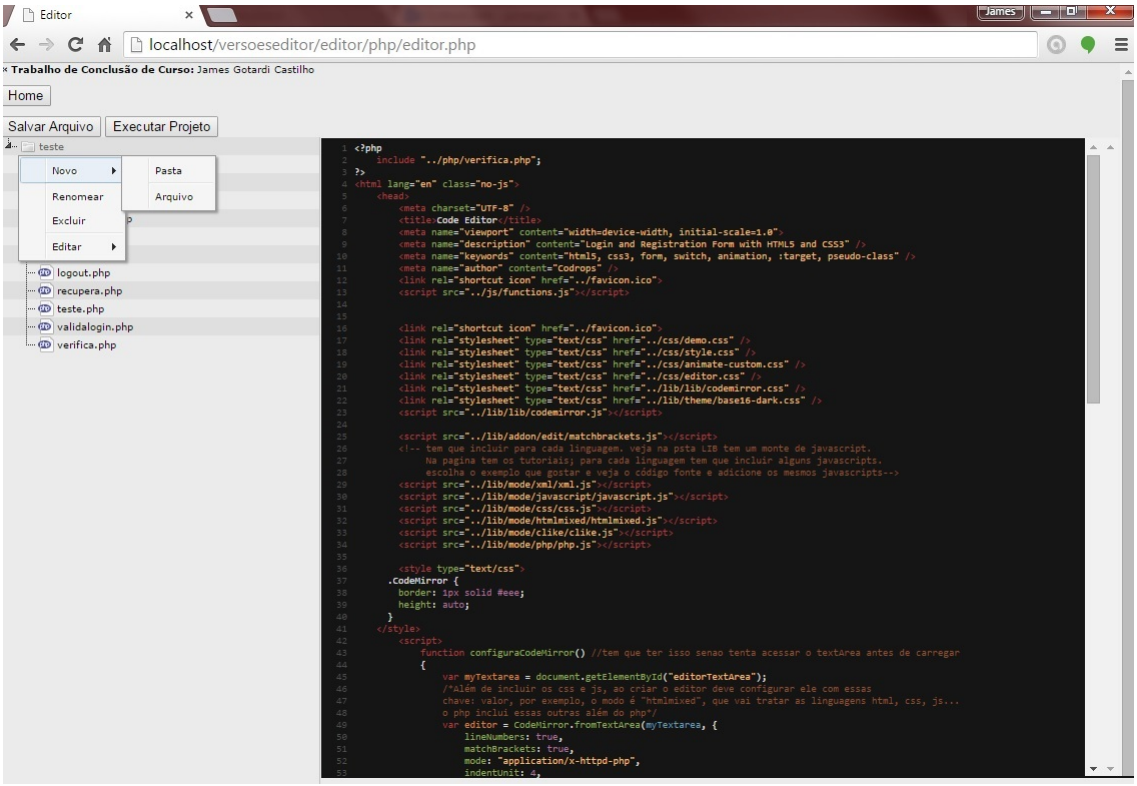


Figura 3.11: Editor carregado com um código.

Como é vista no **Figura 3.11**, a página do editor é dividida em duas partes, a esquerda fica a árvore de arquivos, que mostra todo o conteudo do projeto selecionado, e a direita o editor.

O editor é criado e configurado a partir do *framework Codemirror*, é feita a configuração *CSS* para que seja carregado arquivos maior que a tela, habilitando a barra de rolagem vertical, após entra em ação o *JavaScript*, é criado um *script* que carrega as informações necessárias, criando o editor com o tema escolhido e para as linguagens definidas, como mostra no arquivo editor.php.

O trecho de código a seguir mostra como é feita a configuração de criação do editor.

```
1
2     <style type="text/css">
3       .CodeMirror {
4         border: 1px solid #eee;
5         height: 80%;
6         overflow-y: scroll;
7       }
8     </style>
```

```
9          <script>
10             var editor = null;
11             var currentFile = null;
12         function configuraCodeMirror()
13         {
14             editor =new CodeMirror(document.getElementById("data"),{
15                 lineNumbers: true,
16                 matchBrackets: true,
17                 mode: "application/x-httpd-php",
18                 indentUnit: 4,
19                 indentWithTabs: true,
20                 viewportMargin: Infinity
21               });
22             editor.setOption("theme", "base16-dark");
23         }
24         </script>
```

A árvore de arquivos é criada usando o modelo *jstree*, feito a partir do *JavaScript* para mostrar todos os arquivos pertencentes ao projeto aberto, além disso é possível criar,renomear,excluir,copiar e colar pastas e arquivos.

Capítulo 4

Testes de Linguagens

Este capítulo trata a respeito dos testes realizados no editor, testes estes realizados em um computador e em um *smartphone*, mostrando que as linguagens citadas no decorrer do desenvolvimento funcionam e são aceitas pelo Editor.

4.1 Teste 1 realizado em Computador: Linguagem *PHP*

A seguir uma sequência de passos para criar e editar um projeto, criando um arquivo *.php* para comprovar a identificação da linguagem.

Passo 1: Primeiro deve-se escolher o nome para o projeto para então criá-lo, como mostra a **Figura 4.1**.

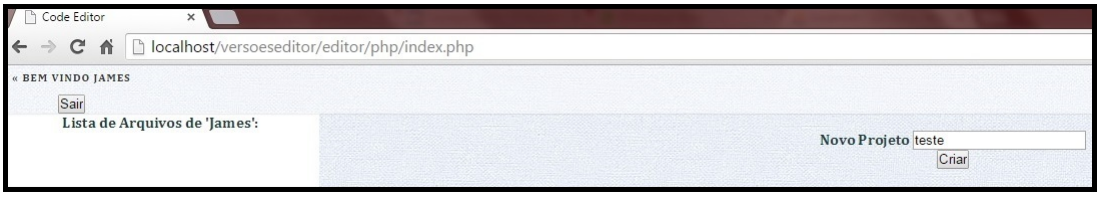


Figura 4.1: Criando o projeto.

Passo 2: Se o nome escolhido ainda não estiver em uso em outro projeto, é informado ao usuário a criação com sucesso, como mostra a **Figura 4.2**.

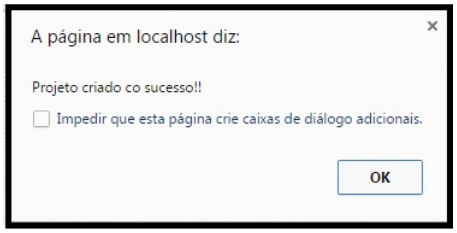


Figura 4.2: Projeto criado.

Passo 3: Ao lado esquerdo da página são mostrados todos os projetos inclusive o recém criado. Para abrí-lo no editor basta clicar em cima dele, como mostra a **Figura 4.3**.

Figura 4.3: Lista de projetos.

Passo 4: Quando aberto o projeto na página do editor, são mostradas todas as pastas e arquivos contidos no projeto, como mostra a **Figura 4.4**.

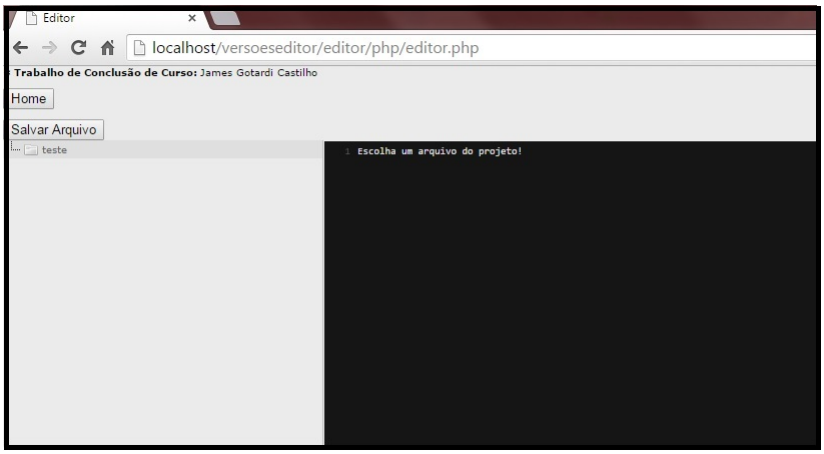


Figura 4.4: Página do editor.

Passo 5: Como o projeto foi recém criado é preciso adicionar arquivos a ele. Para isso basta clicar o botão direito do *mouse* sobre a pasta raiz ou qualquer outra pasta dentro do projeto e adicionar o novo arquivo ou nova pasta, como mostra a **Figura 4.5**.

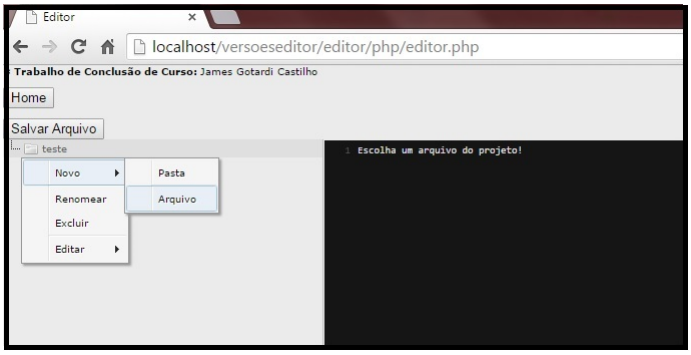


Figura 4.5: Criando novo arquivo ou pasta.

Passo 6: É criado o arquivo. A princípio recebe o nome de "Novo", então, é preciso colocar um nome para o arquivo e sua extensão para que seja reconhecido pelo editor, como mostram as **Figura 4.6** e **4.7**.

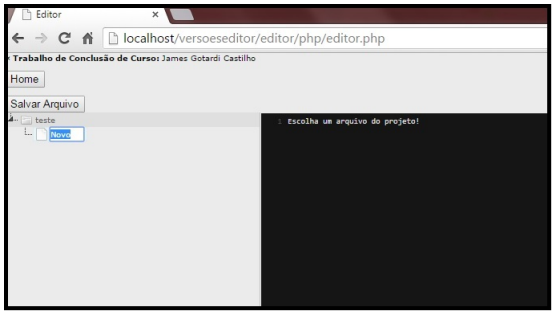


Figura 4.6: Arquivo criado sem extensão.

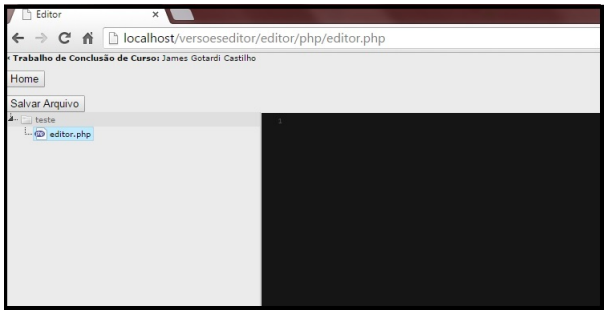


Figura 4.7: Arquivo com nome e extensão reconhecida.

Passo 7:Para abrir o arquivo no editor basta clicar sobre o mesmo. Se este tiver um tipo suportado pelo editor ele será aberto e estará pronto para edição, como mostra a **Figura 4.8**.

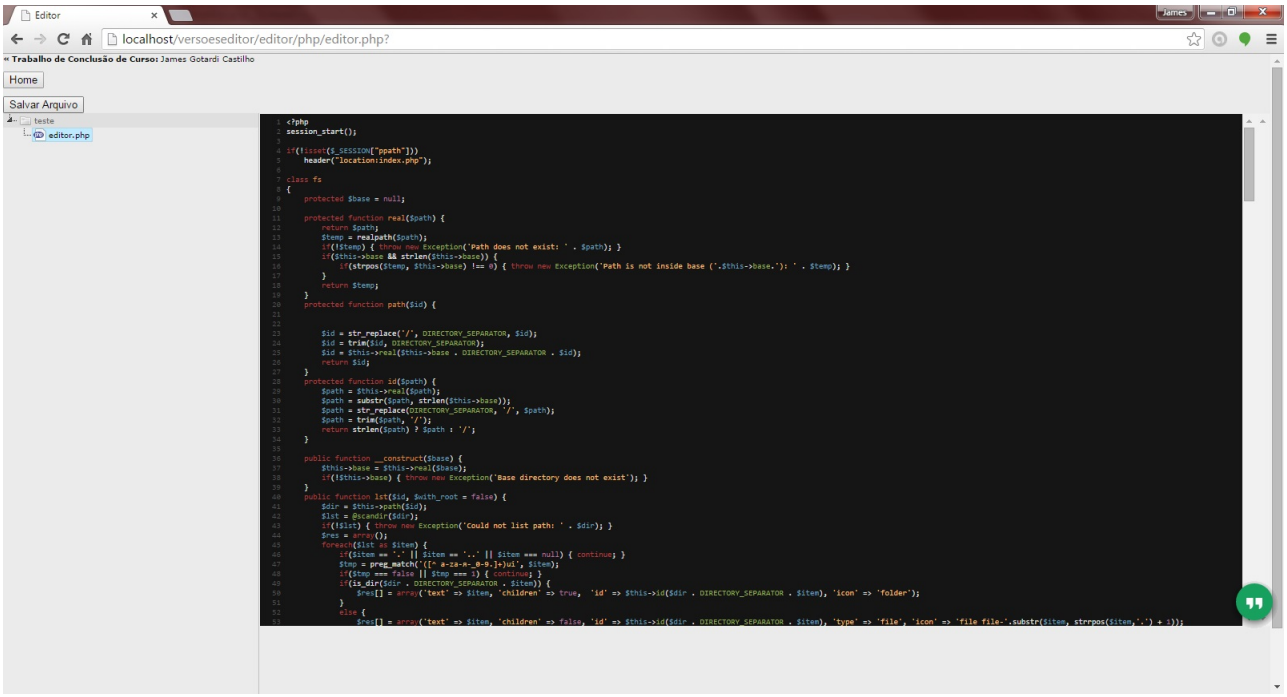


Figura 4.8: Arquivo aberto no editor.

4.2 Teste 2 realizado em *Smartphone*: Linguagem *JS*

Os testes realizados através de um *smartphone* são semelhantes aos realizados no computador. Os passos são iguais em praticamente tudo, com exceção do modo de criar arquivos ou pasta, pois como não existe o botão direto no celular, basta apenas segurar em cima da pasta desejada para abrir as opções de criar arquivos ou pastas assim como renomear, copiar, recortar ou colar.

Figura 4.9 teste realizado no *smartphone*.

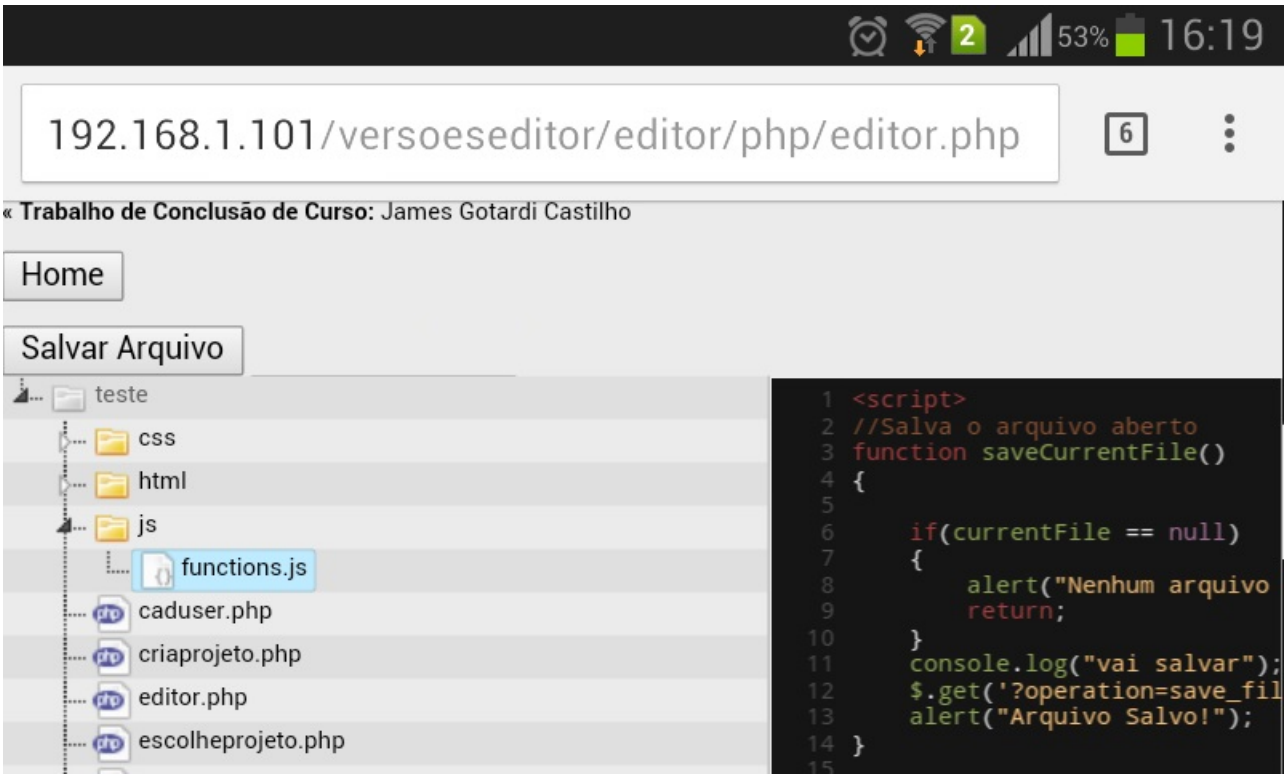


Figura 4.9: Teste *smartphone* da estrutura *JavaScript*.

Capítulo 5

Conclusão

Atualmente, quando se fala em algo relacionado a área de informática, logo se pensa em *internet*, pois tudo está conectado através dela, empresas ou pessoas. Porém quando fala-se em programação a história muda, pessoas associam desenvolvimento de programas a *softwares* instalados em computadores, e deixam de lado a *internet*.

Esse projeto propôs exatamente o contrário, levar o desenvolvimento para *web* também, não somente suas aplicações.

O desenvolvimento do editor que permite ao programador criar e armazenar seus códigos onde ele estiver, foi feito pensando na mobilidade, na necessidade de se estar conectado. Todo o projeto foi desenvolvido utilizando o editor *notepad++* por ser gratuito e de fácil manipulação, além do *XAMPP*, uma aplicação que contém o servidor *web apache*, o *SGBD MySQL* e o *PhpMyAdmin* utilizado para configuração do *PHP*.

Os objetivos traçados para esse projeto foram alcançados de maneira satisfatória, pois permite ao usuário do sistema criar ou editar projetos que estão salvos em sua pasta pessoal no servidor, desenvolver aplicações a algumas linguagens diferentes e ser compatível com navegadores atuais, pois um sistema que foi desenvolvido para *web* necessita dessa compatibilidade, incluindo com *smartphones*.

Até a conclusão desse projeto, o suporte para diferentes linguagens ficou restrita a apenas quatro e também a arquivos de textos normais.

O projeto propostos foi concluído, no entanto permite ainda melhoramentos, como a implantação de funções que permitam ao usuário desenvolver códigos em outras linguagens como *C,C++,Java* entre outras, além de fornecer a possibilidade de executar projetos dessas linguagens *on-line*.

Também há a possibilidade de implementar a execução dos projetos das linguagens já aceitas pelo editor e ainda fornecer mais segurança ao usuário realizando o acesso utilizando protocolos de segurança mais rigorosos, criptografia no banco de dados afim de evitar que pessoas tenham acesso ao servidor e por consequência, acesso as pastas dos usuários do *site*.

Referências Bibliográficas

BIBEAULT, B. and KATZ, Y. (2010). ***JQuery in Action***. Manning, 2 edition. 347 p.

CÁCERES, E. N. (2012). **Bancos de Dados - Conceitos Básicos**. Disponível em `http://www.dct.ufms.br/~edson/bd1/bd1.pdff`, acesso em julho de 2012.

CODEMIRROR (2015). **Codemirror**. Disponível em `https://codemirror.net/doc/manual.html`, acesso em Agosto de 2015.

de SOUZA, R. M. (2014). **História do MySql**. Disponível em `http://blog.segr.com.br/historia-do-mysql`, acesso em outubro de 2015.

FELIX, G. (2005). **JavaScript Tutorial Brasil**. Disponível em `http://www.javascript-tutorial.com.br/content-2.html`, acesso em outubro de 2012.

FREEMAN, E. and FREEMAN, E. (2006). ***Head First HTML with CSS and XHTML: a brain-friendly gyide to HTML and CSS***. O'Reilly Media, 2 edition. 580 p.

GOODMAN, D. (2001). ***JavaScript Bible,Gold Edition***. Hungry Minds, gold edition. 1514 p.

GRILLO, F. D. N. and de MATTOS FORTES, R. P. (2008). **Aprendendo JavaScript**. Disponível em `http://pt.scribd.com/doc/57107960/2/Historico-de-JavaScript`, acesso em outubro de 2012.

MATTSSON, M. (2006). ***Object-Oriented Frameworks,A survey of methodological issues***. Department of Computer Science and Business Administration,University College of Karlskrona/Ronneby, 1 edition. 130 p.

MILANI, A. (2007). ***MySQL - Guia do Programador***. Novatec Editora Ltda, 1 edition. 400 p.

MILANI, A. (2010). ***Construindo Aplicações Web com PHP e MySQL***. Novatec Editora Ltda, 1 edition. 336 p.

MORRISON, M. (2008). ***Use a Cabeça, JavaScript***. Alta Books, 1 edition. 606 p.

PHP.NET (2015). **Manual do *PHP***. Disponível em `http://php.net/manual/pt_BR/index.php`, acesso em outubro de 2015.

RIBEIRO, E. S. (2004). **Introdução a Banco de Dados**. Disponível em `http://www.eduardosilvestri.com.br/fmu/redes/intrbd08_1/conteudo/Aula04/IntroducaoaBancodeDados04.pdf`, acesso em julho de 2012.

SILVA, M. S. (2011). ***JQuery - A Biblioteca do Programador JavaScript***. Novatec Editora Ltda, 2 edition. 430 p.

SILVA, M. S. (2012). ***jQuery UI: Componentes de interface rica para suas aplicações web***. Novatec, 1 edition. 736 p.

SOMMERVILLE, I. (2006). ***Engenharia de Software***. Pearson Education, 8 edition. Cap. 6.

W3C (2012). **World Wide Web Consortium**. Disponível em `http://www.w3c.br/Home/WebHome`, acesso em agosto de 2012.