

**ANDRÉ GOMES ANDRADE**

**PAINEL ADMINISTRATIVO COM CONTROLE DE NÍVEIS DE  
USUÁRIO E CONTROLE DE ACESSO DE NAVEGADORES**

**TEÓFILO OTONI – MG**

**FACULDADES UNIFICADAS DE TEOFILO OTONI**

**2016**

**ANDRÉ GOMES ANDRADE**

**PAINEL ADMINISTRATIVO COM CONTROLE DE NÍVEIS DE  
USUÁRIO E CONTROLE DE ACESSO DE NAVEGADORES**

Monografia apresentada ao Curso de Sistemas de Informação das Faculdades Unificadas de Teófilo Otoni, como requisito parcial à obtenção do título de Bacharel em Sistemas de Informação. Área de Concentração: Programação Web. Orientador: Prof. Salim Ziad Pereira Aouar.

**TEÓFILO OTONI – MG**

**FACULDADES UNIFICADAS DE TEOFILO OTONI**

**2016**

## **AGRADECIMENTOS**

Agradeço a todas as pessoas de minha família que me apoiaram para concluir o curso de Sistemas de Informação. A minha mãe que bancou o final do meu curso junto com meu pai. Agradeço ao meu Pai que, com o salário que ganha, tirou de onde não tinha para poder me ajudar. Agradeço ao meu irmão Amós pelo apoio em algumas situações.

“Os homens pensam que possuem uma mente, mas é a mente que os possui”

**Bob Marley**

## LISTA DE SIGLAS E ABREVIATURAS

ANSI – American National Standards Institute

BD – Banco de Dados

CPE - Customer Premises Equipment

CSS - Cascading Style Sheets

GIS - Geographic Information Systems

GNU - General Public License

HTML - Hypertext Markup Language

MIT - Massachusetts Institute of Technology

OLAP — Online Analytical Processing

PHP - Personal Home Pages

PHP/FI - Personal Home Pages/Forms Interpreter

POO - Programação Orientada a Objetos

POP - Point of Presence

POSH — Plain Old Semantic HTML

SBD – Sistema de Banco de Dados

SGBD - Sistema de Gerenciamento de Banco de Dados

SIG – Sistemas de Informação Gerenciais

SQL – Structured Query Language

T.I – Tecnologia de Informação

VBA - Visual Basic Applications

W3C - World Wide Web Consortium

WWW - World Wide Web

## LISTA DE FIGURAS/ TABELAS/ GRÁFICOS

<b>Figura 1</b> – Computador.....	05
<b>Figura 2</b> – Logomarca do PHP.....	09
<b>Figura 3</b> - Pontos de acesso .....	10
<b>Figura 4</b> - Servidores Web.....	12
<b>Figura 5</b> - Total de números de Websites.....	13
<b>Figura 6</b> – Web Server Developers.....	13
<b>Figura 7</b> – Componentes de um sistema de Banco de dados.....	17
<b>Figura 8</b> – Logomarca do Navicat Premium.....	20
<b>Figura 9</b> - NetBeans .....	22
<b>Figura 10</b> – Navicat Premium.....	23
<b>Figura 11</b> – Escolha da língua e instalação do wamppserver.....	24
<b>Figura 12</b> - Termos de licença do WampServer .....	25
<b>Figura 14</b> - Informações do WampServer 3.....	26
<b>Figura 14</b> - Definição do diretório do WampServer 3.....	26
<b>Figura 15</b> - Instalação do WampServer 3.....	27
<b>Figura 16</b> - Escolha do <i>Browser</i> do WampServer 3.....	28
<b>Figura 17</b> - Escolha do editor de texto do WampServer 3.....	28
<b>Figura 18</b> - Pacotes instalados junto ao WampServer 3.....	29
<b>Figura 19</b> - Iniciando os serviços do WampServer 3.....	29
<b>Figura 20</b> - Tela de Login.....	30
<b>Figura 21</b> - Tela Principal.....	31
<b>Figura 22</b> - Controle de Usuários.....	32
<b>Figura 25</b> - Tráfego por navegador.....	32
<b>Figura 24</b> - Cadastro de Post.....	33

<b>Figura 25</b> - Cadastro de Post/Categorias.....	34
<b>Figura 26</b> - Cadastro de Post/Autores.....	34
<b>Figura 27</b> - Edição de Post.....	35
<b>Figura 28</b> - Cadastro de categorias.....	36
<b>Figura 29</b> - Cadastro de Subcategorias.....	36
<b>Figura 30</b> - Tela de edição de categorias.....	37
<b>Figura 31</b> - Tela de edição de categorias/ Ver no site.....	37
<b>Figura 32</b> - Cadastro de empresas.....	38
<b>Figura 33</b> - Tela de edição de empresas.....	39
<b>Figura 34</b> - Parte do código onde cria Usuário.....	46
<b>Figura 635</b> - Parte do código onde atualiza Usuário.....	47
<b>Figura 36</b> - Todas as classes.....	47
<b>Figura 77</b> - Parte do código onde cria Categoria.....	48
<b>Figura 38</b> - Parte do código onde lista as categorias das notícias.....	48
<b>Figura 39</b> - Parte do código onde cria Postagens.....	49
<b>Figura 40</b> - Parte do código onde cria Postagens II.....	49
<b>Figura 41</b> - Parte do código onde edita as postagens.....	50
<b>Figura 42</b> - Parte do código onde lista as empresas cadastradas.....	50
<b>Figura 48</b> - Parte do código onde faz a edição das empresas.....	51
<b>Figura 44</b> - Index do site.....	51
<b>Figura 45</b> - Index do painel.....	52
<b>Figura 46</b> - Parte do código onde faz o controle do nível de usuário.....	52
<b>Figura 47</b> - Banco de Dados.....	53

## RESUMO

Este trabalho de pesquisa tem como objetivo o desenvolvimento de um Painel Administrativo para um portal de notícias, que pode vir a colocar notícias em tempo real no Vale do Mucuri e ter empresas vinculadas ao site como parceiras. Por meio de pesquisa e observação foi possível coletar e desenvolver todas as funcionalidades do Painel Administrativo. O desenvolvimento partiu de conhecimentos adquiridos no referencial teórico e pesquisas práticas, dando ênfase na Programação Web, CSS3, HTML5 e Banco de Dados. Logo em seguida foi realizada a modelagem do banco de dados, e posteriormente o desenvolvimento das interfaces gráficas do painel visando às funcionalidades do portal administrativo para interação com o usuário. Portanto, o painel visa colocar o portal de notícias prático e de fácil acesso aos usuários, trazendo assim a flexibilidade e praticidade que proporcionam uma perspectiva futura ao site, podendo ser usado futuramente por empresas do ramo ou até mesmo pelo autor no âmbito empresarial.

**Palavras-Chave:** Painel Administrativo. Portal de Notícias. Desenvolvimento Web.



## SUMÁRIO

<b>INTRODUÇÃO .....</b>	<b>2</b>
<b>1 REFERENCIAL TEÓRICO .....</b>	<b>5</b>
1.1 O QUE É COMPUTADOR? .....	5
1.2 SISTEMAS DE INFORMAÇÃO GERENCIAIS (SIG).....	6
1.3 TECNOLOGIAS E SISTEMAS DE INFORMAÇÃO .....	7
1.4 LINGUAGENS DE PROGRAMAÇÃO ORIENTADA A OBJETOS .....	8
<b>1.4.1 PHP .....</b>	<b>8</b>
1.5 AMBIENTE WEB .....	10
<b>1.5.1 Apache.....</b>	<b>11</b>
<b>1.5.2 HTML.....</b>	<b>14</b>
<b>1.5.3 Marcação Semântica .....</b>	<b>15</b>
<b>1.5.4 CSS .....</b>	<b>15</b>
<b>1.5.5 JQuery .....</b>	<b>16</b>
<b>1.5.6 Banco de dados.....</b>	<b>16</b>
1.6 SISTEMAS DE BANCO DE DADOS.....	18
<b>1.6.1 Base de dados x SGBD.....</b>	<b>18</b>
<b>1.6.2 Sql .....</b>	<b>19</b>
<b>2 DESENVOLVIMENTO DO SOFTWARE.....</b>	<b>21</b>
2.1 MÉTODOS E FERRAMENTAS UTILIZADAS.....	21
2.1.1 NetBeans.....	21
<b>2.1.2 Navicat Premium .....</b>	<b>23</b>
<b>2.1.3 WampServer.....</b>	<b>24</b>
2.2 O SOFTWARE.....	30
<b>3 REULTADOS E DISCUSSÕES.....</b>	<b>40</b>
<b>CONCLUSÃO .....</b>	<b>42</b>
<b>REFERÊNCIAS.....</b>	<b>44</b>

## INTRODUÇÃO

Esta monografia tem por objetivo o desenvolvimento de um portal de notícias para o Vale do Mucuri. Para isso foi desenvolvido um Painel Administrativo que atenda as necessidades e funcionalidades desse portal.

Com toda essa modernidade, para que possa colocar no ar notícias e também trabalhar com empresas da região, mostrando a elas o uso da Tecnologia da Informação (TI) em âmbito empresarial e pessoal.

Este novo mercado proporciona flexibilidade e competitividade que, através de novos métodos, resultam na otimização de vendas e do marketing da empresa, promovendo desenvolvimento nos anúncios das empresas cadastradas no painel.

A utilização de um portal para mostrar notícia e anunciar empresas pode ser um novo método utilizado por pequenas, médias e grandes empresas com eficiência maior que outros meios de publicidade, buscando um público específico.

Para gerenciar esse portal foi criado um Painel Administrativo que facilita o uso do site para o administrador e para o editor. Além disso, o Painel facilita:

- Cadastro de novas empresas;
- Ativar, desativar e editar as notícias e as empresas cadastradas.

O site mostra essas empresas de forma organizada, separando por tipo, as notícias são exibidas no portal por ordem de data de cadastro.

Mediante todas as informações analisadas, levantou-se a seguinte questão problema: será viável implantação de um portal de notícias no Vale do Mucuri?

Com relação à pesquisa realizada foram levantadas as seguintes hipóteses:

H0: Não seria viável a criação de um painel administrativo para um portal de notícias, pois o mesmo poderia não suprir as necessidades do mercado.

H1: Não seria viável colocar o site em um servidor *Web*, pois o custo seria alto para um investimento inicial.

H2: Seria viável a construção de um Painel Administrativo para manutenção do portal de notícias, que seria alimentado através do painel.

H3: Seria viável a criação de um portal de notícias da região para mostrar as notícias em tempo real e teria empresas parceiras para manter o site funcionando perfeitamente.

Após os levantamentos, foram instituídos objetivos específicos para ajudar no andamento da pesquisa, com a finalidade de ter uma linha de trabalho para o desenvolvimento do painel administrativo do portal de notícias do Vale do Mucuri:

- Diagnosticar problemas que possam atrapalhar o andamento do site;
- Realizar uma pesquisa bibliográfica para dar direção mais exata ao desenvolvimento do site;
- Desenvolver um painel administrativo que atenda às necessidades do site;
- Testar todas as funcionalidades do *software*;
- Fornecer a primeira versão do sistema com todas as necessidades testadas e funcionando.

Esta pesquisa é descritiva e explicativa, feita através de um processo de análise, a fim de desenvolver soluções cabíveis para otimização do projeto.

A presente pesquisa mostra novos métodos de trabalho e novos meios de divulgação, bem como inovações tecnológicas para pequenas e médias empresas da região, de modo que proporciona confiabilidade. Essas metodologias proporcionam ampliação de uma realidade bastante conhecida por grandes empresas.

Utilizou-se de pesquisa bibliográfica através de leituras que auxiliam no conhecimento do tema proposto e de interpretações das informações adquiridas numa pesquisa de campo. Será possível analisar problemas que eventualmente possam atrapalhar o andamento do projeto, e finalmente por meio da prática

desenvolveram-se mecanismos de trabalho que antecedam problemas e gerem soluções.

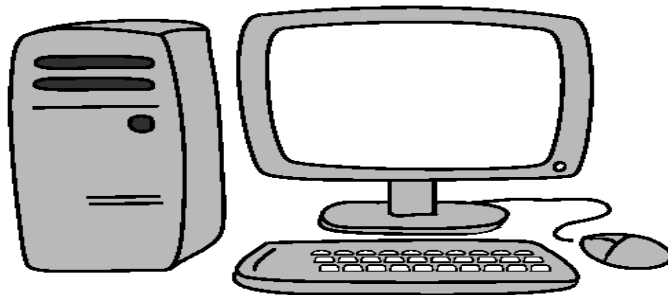
Com esse conhecimento, veio a possibilidade de gerenciar e desenvolver sites, compreendendo e processando informações de forma eficaz em um mercado em bastante crescimento e carente de profissionais capacitados, de uma forma sólida e técnica.

O trabalho proposto está disposto em capítulos, e através de um referencial teórico bastante estruturado, onde foi abordada a evolução do projeto, entre alguns tópicos abordados, destacam-se a linguagem de programação PHP5, CSS3 e HTML5, seguido da preparação do ambiente *Web*.

# 1 REFERENCIAL TEÓRICO

## 1.1 O QUE É COMPUTADOR?

**FIGURA 9 - COMPUTADOR**



Fonte: O próprio autor (Criado em 18 de setembro de 2016)

O computador figura 1 nada mais é que um recurso que permite usar várias funções ao mesmo tempo com velocidade e precisão, exemplo: armazenagem de dados, pesquisa, seleção, classificação, recuperação de dados, execução de cálculos e etc.

Ele executa essas funções com incrível velocidade (inacessível por outros meios) com uma confiabilidade muito alta, através de rotinas exclusivas para cada tarefa.

Porém, ele se aplica a determinadas situações. Ao contrário, torna-se facilmente um recurso ocioso, sem estrutura, mal planejado, em desacerto com a realidade do processamento em questão. (CORNACHIONE, Edgard B. Jr., 2007, p. 39-40).

Segundo Cornachione (2007, p.40):

As situações mais abrangentes em que se indica o uso do computador são aquelas em que se verificam as seguintes características:

- Grande volume de serviço;
- Ciclo de transações (repetição);
- Dependência de grande quantidade de cálculos;
- Relativa estabilidade das operações (devido ao fato de as mudanças de sistemas serem onerosas);
- Aproveitamento de dados (preocupação em evitar reprocessamento);
- Necessidade em maior precisão informações (confiança no sistema).
- Necessidade e menor tempo de resposta na obtenção das informações.

Quando determinado processamento se encontra nessa situação, uma das prováveis soluções emergentes é implantar o “computador” e torná-lo um processamento eletrônico de dados.

Atualmente, as tecnologias computacionais (equipamentos, lógica, comunicação, conectividade, fácil relacionamento/uso etc.) estão-se tornando cada vez mais acessíveis. Essa acessibilidade se dá, entre outros fatores, tanto em termos financeiros (recursos computacionais mais baratos) quanto em termos de existências (hoje, devido à pesquisa e divulgação apropriada, é possível tomar conhecimento sobre a existência das mais variadas tecnologias de informática).

## 1.2 SISTEMAS DE INFORMAÇÃO GERENCIAIS (SIG)

Em um cenário com uma competitividade cada vez mais alta com diversos seguimentos, para ter êxito precisa-se maximizar constantemente os resultados. Devem-se manter os sistemas funcionando muito bem com manutenções e um bom

relacionamento com o cliente, para isso um sistema deve ter ferramentas de otimização do gerenciamento das informações disponíveis em sua base de dados para possibilitar os gerentes, gestores e executivos a tomarem suas decisões com base em informações atualizadas, diminuindo cada vez mais o tempo de respostas aos diversos comportamentos do mercado e da própria empresa. Uma das ferramentas que se pode utilizar é o SIG.<sup>1</sup>

### 1.3 TECNOLOGIAS E SISTEMAS DE INFORMAÇÃO

A tecnologia está transformando o mundo, com ela o meio empresarial está sendo revolucionado de uma maneira muito rápida, trazendo soluções rápidas e ágeis para o meio. O uso da TI em empresas afeta ela de uma forma impressionante, por isso está sendo utilizada por mais empresas cada vez mais.

Martinz et all (2012, p.4) afirmam que:

A TI pode ser conceituada como recursos tecnológicos e computacionais para a geração, armazenamento e uso da informação, ou seja, é um conjunto de todas as atividades e soluções providas por recursos de computação. O significado de TI pode ainda ser utilizado em um sentido mais amplo, referindo para descrever um conjunto de diversos sistemas de informação, usuários e gestão de toda uma organização. Então a TI é a preparação, coleta, transporte e recuperação, armazenamento, acesso, apresentação e transformação de informações.

---

<sup>1</sup> < [http://tupi.fisica.ufmg.br/michel/docs/Artigos\\_e\\_textos/Ciencia\\_da\\_informacao/SIG\\_conceitos.pdf](http://tupi.fisica.ufmg.br/michel/docs/Artigos_e_textos/Ciencia_da_informacao/SIG_conceitos.pdf)>

## 1.4 LINGUAGENS DE PROGRAMAÇÃO ORIENTADA A OBJETOS

A programação orientada a objetos (POO) foi criada para interligar o mundo real ao mundo virtual. Fundamental é simular o mundo real dentro do computador; para isso, nada mais natural que utilizar objetos.

Na POO o programador é responsável por moldar o mundo dos objetos e explicar para estes objetos como eles devem interagir entre si. Os objetos “conversam” uns com os outros através de envio de mensagens, e o papel principal do programador é especificar quais serão as mensagens que cada objeto pode receber, e também qual a ação que aquele objeto deve realizar ao receber aquela mensagem. Uma mensagem é um pequeno texto que os objetos conseguem entender, e por questões técnicas não pode conter espaços. Junto com algumas dessas mensagens ainda é possível passar algumas informações para o objeto (parâmetros); dessa forma, dois objetos conseguem trocar informações entre si facilmente.<sup>2</sup>

Segundo Pablo Dall'Oglio (2004, p 173):

São diversas as vantagens de se utilizar o modelo orientado a objetos no desenvolvimento de aplicações. Desde a similaridade dos objetos com a nossa percepção do mundo real, facilitando o entendimento do sistema e a provável manutenção do mesmo; a reusabilidade proporcionada pela forma de organização do sistema através de uma hierarquia em árvore; o encapsulamento dos objetos, que permitem trabalharmos diretamente com atributos, evitando variáveis globais em nossas aplicações, e até a elegância do código, que tende a ficar mais enxuto, organizado e de fácil compreensão.

### 1.4.1 PHP

É uma das linguagens mais usadas na atualidade, é uma linguagem de script de código aberto (Open Source), adequada especialmente para desenvolvimento web que pode ser embutida dentro do HTML.

---

<sup>2</sup> <<http://www.hardware.com.br/artigos/programacao-orientada-objetos/>>.



Segundo Pablo Dall'Oglio (2004, p 21):

Como todos sabem, o PHP é uma linguagem de Script consistente e flexível, com diversas extensões, desempenho superior e uma grande comunidade. Criada por Rasmus Lerdorf em 1995. A partir de 1997 o core do PHP foi reescrito por Zeev Suravski e Andi Gutmans, formando o que conhecemos hoje por Zend Engine. Na Versão 5 do PHP o core Zend Engine atinge a versão 2.0, possibilitando novos recursos que veremos neste artigo.

O mesmo autor completa:

A linguagem de programação PHP, cujo logotipo vê-se na figura 2, foi criada no outono de 1994 por Rasmus Lerdorf. No início era formada por um conjunto de scripts voltados à criação de páginas dinâmicas que Rasmus utilizava para monitorar o acesso ao seu currículo na internet. À medida que essa ferramenta foi crescendo em funcionalidades, Rasmus teve de escrever uma implementação em C, a qual permitia às pessoas desenvolverem de forma muito simples suas aplicações para web. Rasmus nomeou essa versão de PHP/FI (Personal Home Pages/Forms Interpreter) e decidiu disponibilizar seu código na web, em 1995, para compartilhar com outras pessoas, bem como receber ajuda e correção de bugs.

**FIGURA 2 – LOGOMARCA DO PHP**



Fonte: PHP Programando com Orientação a Objetos, 2007 p. 21

## 1.5 AMBIENTE WEB

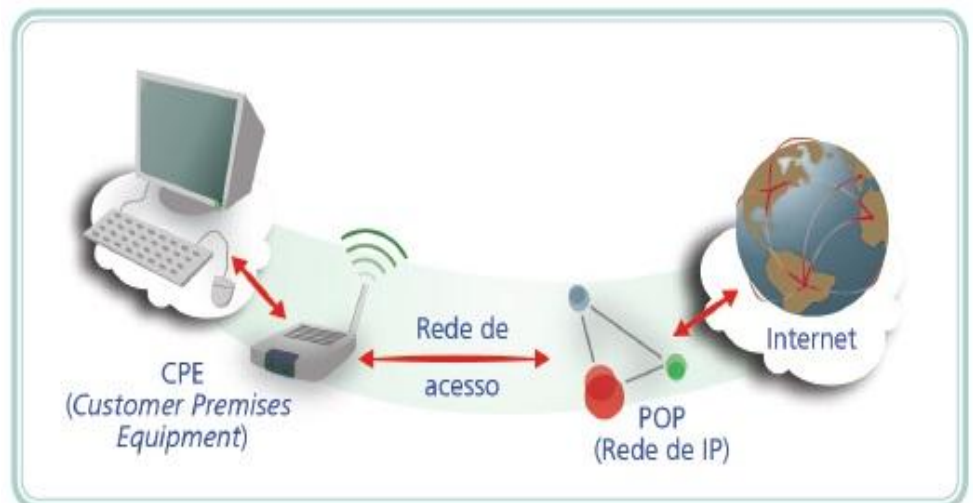
Ambiente onde são instalados e configurados todos os programas que vão ser utilizados no desenvolvimento Web. Aqui eles são tratados de forma a facilitarem o desenvolvimento web para que possam otimizar a plataforma de desenvolvimento.

Segundo Boniati e Silva (2013, p. 17):

A internet é uma rede de computadores de acesso público e ilimitado (sem um “dono”) que utiliza a infra-estrutura de telecomunicações. Embora não exista um dono, existem consórcios internacionais, como o W3C (*World Wide Web Consortium*), com a tarefa de agregar empresas filiadas na tentativa de, em conjunto, desenvolver padrões para a internet.

O acesso a internet se dá, normalmente, por meio de um ISP (Internet Service Provider ou Provedor de Serviço de Internet) e utiliza-se de, pelo menos, três componentes (CPE, Rede de Acesso e POP) ilustrados na Figura 3.

**FIGURA 3 - PONTOS DE ACESSO**



Fonte: Fundamentos do desenvolvimento Web, 2013 p. 17

- CPE (Customer Premises Equipment) e o equipamento que conecta o dispositivo a rede de acesso (exemplo: modem).
- Rede de acesso e o tipo de infraestrutura que liga o dispositivo ao provedor de internet (exemplos: cabos de cobre, fibra ótica, Wi-Fi).
- POP (Point of Presence) e o ponto de presença do provedor onde estão os equipamentos que atribuem ao dispositivo um endereço IP, dando-lhe acesso a internet.

### 1.5.1 Apache

O Apache por ser um servidor Web gratuito de fonte aberta. Conseguiu superar duas grandes empresas comerciais, a Microsoft e a Netscape, como uma plataforma de servidor Web. Essa superação vem de diversos fatores, um deles é que ele é altamente configurável com um projeto modular.

Funciona muito bem com Perl, PHP e outras linguagens de scripts. Grande parte das aplicações Web ainda é formada por scripts. Perl é excelente no mundo dos scripts e o apache facilita muito o uso do Perl. Tem uma estrutura formada com vários módulos, permitindo alterar a estrutura, e até mesmo, criação de novos módulos para adaptar as necessidades do usuário, além de ser multiplataforma. (KABIR, 2002, p. 4-5).

Segundo Kabir, (2002, p.6):

Um dos maiores recursos que o Apache oferece é o fato de funcionar em quase todas as plataformas de computação mais utilizadas. No início, o Apache era principalmente um servidor Web baseado no Unix, mas essa característica não é verdadeira. O apache não apenas funciona na maioria das variedades do Unix(se não em todas), mas também no windows 2000/NT/9x e em muitos outros sistemas operacionais das classes de servidores e de desktop, como o Amiga OS 3.x e o OS/2.

Segundo a Netcraft, em Agosto de 2016 houve uma queda de 3,8% nos usuários do apache por conta da perda de um grande número de dispositivos de consumidor-NAS rodando Apache. Em contrapartida os dispositivos têm aumentado

de forma constante seus usuários desde o início do ano. Já em agora em Outubro teve declínio acentuado. A maioria desses dispositivos são ligados por meio de linhas de internet de casa, portanto, susceptíveis de vir e ir de mês para mês. Para evitar tantas perdas, a Apache este mês está distribuindo por vários provedores de consumo. E por outro lado ela sempre vê o crescimento entre os provedores de hospedagem web. Abaixo seguem gráficos comparativos entre empresas de servidores web.<sup>3</sup>

A figura 4 mostra uma lista dos maiores desenvolvedores web, mostrando a porcentagem que cada um hospeda no mundo.

**FIGURA 4 - SERVIDORES WEB**

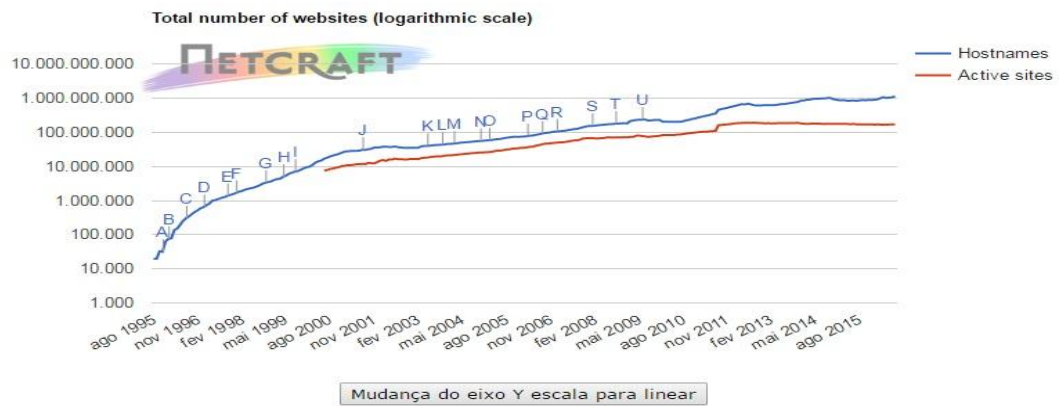
Desenvolvedor	julho 2016	Por cento	agosto 2016	Por cento	Alterar
Microsoft	378655759	35,26%	445105755	38,58%	3,32
Apache	340551074	31,72%	300028832	26,01%	-5,71
nginx	170896716	15,92%	181606297	15,74%	-0.17
Google	22552901	2,10%	22111431	1,92%	-0.18

Fonte: <http://news.netcraft.com/archives/category/web-server-survey/>

A figura 5 mostra o total de Websites existentes no mundo até o momento.

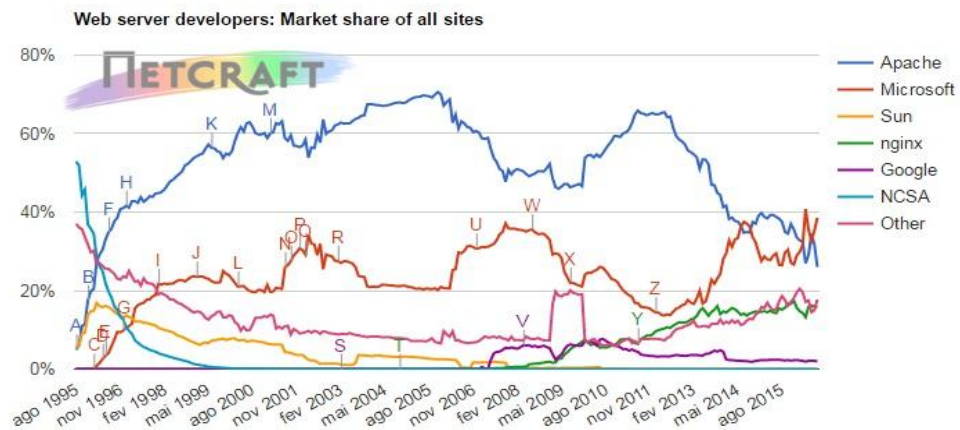
---

<sup>3</sup> <<http://news.netcraft.com/archives/category/web-server-survey/>>

**FIGURA 5 - TOTAL DE NÚMEROS DE WEBSITES (LOGARITHMIC SCALE)**

Fonte: <http://news.netcraft.com/archives/category/web-server-survey/>

A figura 6 mostra um gráfico mostrando qual o maior servidor web da atualidade, que vem sendo a apache desde 1995 e deu uma oscilada de 2014 para frente.

**FIGURA 6 - WEB SERVER DEVELOPERS MARKET SHARE OF ALL SITES**

Fonte: <http://news.netcraft.com/archives/category/web-server-survey/>

## 1.5.2 HTML

O HTML é uma abreviação de (Hypertext Markup Language) que significa Linguagem de Marcação de Hipertexto. O HTML é uma linguagem de publicação de conteúdo na WEB (texto, imagem, vídeo, áudio, etc.).

Segundo Eis e Ferreira (2012, p.26-27):

O HTML é baseado em Hipertexto que é a forma de organizar o conteúdo não linear. Hipertexto são conjuntos de elementos ou nós ligados por conexões. Esses elementos podem ser palavras, imagens, áudio etc. Estes elementos conectados formam uma grande rede de informação. A conexão feita em um hipertexto é algo imprevisto que permite a comunicação de dados, organizando conhecimentos e guardando informações relacionadas.

Nós marcamos a informação, dando significado a estes objetos, tornando-os legíveis para os meios de acesso. Assim, quando marcamos um título com h1, h2 ou h3, indicamos para os meios de acesso que determinado bloco de texto é um título, com uma determinada importância e assim por diante com os outros elementos.

Dessa forma cada meio de acesso pode decidir o que fazer com esta informação. O browser, por exemplo, carrega a informação na tela do usuário. Já o robô do Google guarda a informação e a utiliza em suas buscas. Cada meio de acesso tem sua função, logo, utiliza a informação de maneiras diferentes.

### 1.5.2.1 HTML 5

O WHATWG desenvolve a HTML5 em conjunto com o W3C. Ambos mantêm em seus sites uma versão das especificações que diferem ligeiramente em pequenos detalhes. A versão do WHATWG é menos restritiva do que a versão do W3C. Por exemplo: em vários itens da especificação, apresenta exemplos ilustrativos e informações sobre suporte da funcionalidade descrita, nos navegadores modernos. Essas informações adicionais não constam da versão do W3C (SILVA, 2011, p.26).

### 1.5.3 Marcação Semântica

Sabe-se que h1 é um título dentro de um documento HTML, e h2 subtítulo, p parágrafo e assim por diante. E isso pela marcação semântica dentro do HTML nas normas da W3C, ela acompanha todo o desenvolvimento do site. Por isso, todo desenvolvedor deve estar por dentro de todas as mudanças da marcação.

Manian et al afirmam (2014, p.4):

Acreditamos na importância da marcação semântica (às vezes, chamado de POSH — Plain Old Semantic HTML —, sigla em inglês para (Bom e Velho HTML Semântico). Acreditamos que o HTML é um elemento de design e que, antes de adicionar uma camada de apresentação (que aprimora a marcação subjacente), é importante focar a construção de uma sólida fundação de conteúdo bem estruturado.

Marcação semântica é auto descritiva e usa os elementos corretos de HTML para o trabalho certo. Por exemplo, você poderia marcar um cabeçalho da seguinte forma:

```
<div id="heading" style="font-size:300%; padding: 10px;">My heading</div>
```

### 1.5.4 CSS

O CSS (Cascading Style Sheets) é uma folha de estilos em cascata, utilizada para formatar cores, posição, características de fonte, alinhamento, borda e etc. Essas informações são entregues pelo HTML e interpretada pelo navegador Web.

Segundo Eis e Ferreira (2012, p.137):

O CSS formata a informação que é entregue pelo HTML. Essa informação pode ser qualquer coisa: imagem, texto, vídeo, áudio ou qualquer outro elemento criado. Grave isso: CSS formata a informação. Essa formatação na maioria das vezes é visual, mas não necessariamente. No CSS Aural, nós manipulamos o áudio entregue ao visitante pelo sistema lê a tela. Nós controlamos volume, profundidade do som, tipo da voz ou em qual caixa de som a voz sairá. De certa forma você está formatando a informação que está em formato de áudio e que o visitante está consumindo ao entrar no site. O CSS prepara essa informação para que ela seja consumida da melhor maneira possível.

### 1.5.5 JQuery

O JQuery é uma biblioteca *javascript* fácil de utilizar, leve e faz com que os desenvolvedores escreva menos códigos e faça mais em menos tempo.

Segundo Silva (2010, p. 25):

É uma biblioteca JavaScript que foi criada por John Resig, feita para simplificar a criação de efeitos visuais e de interatividade em web sites, logo foi disponibilizada como software livre, de uso e emprego regido segundo licença conforme regras da MIT(Massachusetts Institute of Technology) e pelo GPL(GNU General Public License). Resumindo pode-se usar a biblioteca tanto para projetos pessoais como comerciais.

O jQuery usa scripts de uma forma tão simples que com meia dúzia de linhas faz o mesmo que o javascript faz usando umas 40 linhas.

### 1.5.6 Banco de dados

Banco de dados é o local onde fica armazenado dados de maneira estruturada e sem redundância. Esses dados são acessados por um programa específico, ou um web site.

Segundo Elmasri e Navathe, (1989, p.3):

Os bancos de dados e os sistemas de bancos de dados se tornaram componentes essenciais no cotidiano da sociedade moderna. No decorrer do dia, a maioria de nós se depara com atividades que envolvem alguma interação com os bancos de dados. Por exemplo, se formos ao banco para efetuarmos um depósito ou retirar dinheiro, se fizermos reservas em um hotel ou para a compra de passagens aéreas, se acessarmos o catálogo de uma biblioteca informatizada para consultar uma bibliografia, ou se comprarmos produtos — como livros, brinquedos ou computadores — de um fornecedor por intermédio de sua página Web, muito provavelmente essas atividades envolverão uma pessoa ou um programa de computador que acessará um banco de dados. Até mesmo os produtos adquiridos em supermercados, em muitos casos, atualmente, incluem uma atualização automática do banco de dados que mantém o controle do estoque disponível nesses estabelecimentos.

Essas interações são exemplos do que podemos denominar aplicações tradicionais de banco de dados, no qual a maioria das informações que são armazenadas e acessadas apresenta-se em formatos textual ou numérico.

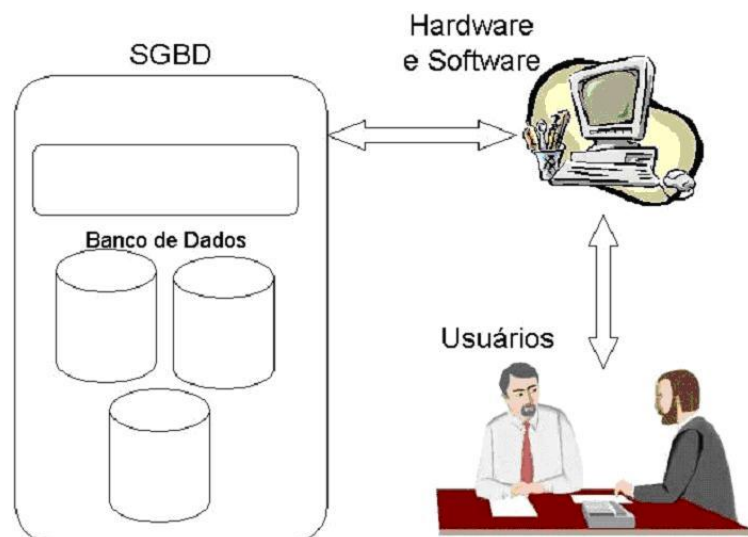


Nos últimos anos, os avanços tecnológicos geraram aplicações inovadoras e interessantes dos sistemas de banco de dados.

Os bancos de dados de multimídia podem, agora, armazenar figuras, vídeos e mensagens sonoras. Os sistemas de informações geográficas (geographic information systems — GIS) são capazes de armazenar e analisar mapas, dados do tempo e imagens de satélite. Os data warehouses e os online analytical processing (OLAP) — processamento analítico on-line — são utilizados em muitas empresas para extrair e analisar as informações úteis dos bancos de dados para a tomada de decisões.

A tecnologia de bancos de dados ativos (active database technology) e real time (tempo real) são usados no controle de processos industriais e de produção (indústria). As técnicas de pesquisa em banco de dados estão sendo aplicadas na World Wide Web (WWW) para aprimorar a recuperação de informações necessárias pelos usuários da Internet.

**FIGURA 7 - COMPONENTES DE UM SISTEMA DE BANCO DE DADOS.**



Fonte: <<http://www.devmedia.com.br/conceitos-fundamentais-de-banco-de-dados/1649#ixzz3bYBAENgN>>

## 1.6 SISTEMAS DE BANCO DE DADOS

Segundo Prof. Bittencourt (2004 p.3):

Sistema de Banco de Dados é um sistema de software composto pelos programas de aplicação, pelo SGBD e pelo BD, para um conjunto de aplicações de uma mesma organização.

Programas de aplicação, colocado na definição acima, são programas que realizam funções da aplicação. EX.: cálculo das deduções e impostos, a partir da receita apurada, dos custos computados e da legislação em vigor. Eles também são os responsáveis pela garantia das restrições de integridade que não podem ser controladas pelo SGBD. Implementam interfaces e relatórios específicos. Acessam o BD através do SGBD para consulta e atualização dos dados da aplicação. V

$SBD = BD + SGBD + PA$

De acordo com (DATE,1985), um SBD é dividido em módulos que tratam de partes, em separado, cada uma das responsabilidades do sistema geral. Estes componentes fundamentais são:

- Gerenciador de Arquivos que trata da alocação do espaço para armazenamento e das estruturas de dados utilizadas para representar a informação armazenada no disco;
- Gerenciador de Banco de Dados fornece a interface entre os dados de baixo nível armazenados no disco e os programas aplicativos e de consulta submetidos ao sistema;
- Processador de Consultas traduz as consultas escritas em uma linguagem de alto nível para instruções de baixo nível que o gerenciador do banco de dados entende;
- Pré-compilador DML converte comandos DML embutidos em um aplicativo para chamadas de procedimento normal na linguagem hospedeira;
- Arquivos de Dados armazenam banco de dados por si mesmos;
- Dicionário de Dados é o componente responsável pelo armazenamento dos metadados sobre a estrutura;

### 1.6.1 Base de dados x SGBD

Um SGBD é um Sistema de gerenciamento de banco de dados, serve para criar um banco de dados, e manipular as informações contidas em um banco de dados.

Segundo Neves e Rua (2005 p .16):

Por vezes confundimos os conceitos Base de Dados e Sistema de Gestão de Base de Dados (SGBD).

De forma a clarificar esta eventual confusão procedemos à definição dos dois conceitos:

• Uma Base de Dados consiste num conjunto integrado de dados, utilizável para múltiplos objetivos e acessíveis por múltiplos tipos de utilizadores de uma forma concorrente, refletindo os propósitos fundamentais de:

- Integração;
  - Partilha;
  - Concorrência;
  - Múltiplos objetivos;
  - Múltiplos tipos de utilizadores.
- Um SGBD consiste no software que gere todo o acesso a uma ou mais bases de dados, permitindo a definição, acesso concorrente, manipulação e controlo dos dados, assegurando a integridade, segurança e recuperação das bases de dados.

## 1.6.2 Sql

SQL é uma linguagem de computador para trabalhar com conjuntos de fatos e as relações entre elas. Programas de banco de dados relacional, como o Microsoft Office Access, usam o SQL para trabalhar com dados. Ao contrário de muitas linguagens de computador, SQL não é difícil de ler e entender, mesmo para um iniciante. Como muitas linguagens de computador, SQL é um padrão internacional que é reconhecido pelos corpos de padrões como ISO e ANSI.<sup>4</sup>

Pode-se utilizar SQL para descrever os conjuntos de dados que podem ajudá-lo a responder perguntas. Quando se usa SQL, deve-se usar a sintaxe correta. Sintaxe é o conjunto de regras pela qual os elementos de um idioma são

---

<sup>4</sup> < <https://support.office.com/pt-br/article/Access-SQL-conceitos-b%C3%A1sicos-vocabul%C3%A1rio-e-sintaxe-444d0303-cde1-424e-9a74-e8dc3e460671> >

combinados corretamente. Sintaxe SQL baseado na sintaxe inglesa e usa muito dos mesmos elementos como Visual Basic para sintaxe Applications (VBA).<sup>5</sup>

### 1.6.2.1 Navicat Premium

**FIGURA 8 - LOGOMARCA DO NAVICAT PREMIUM**



Fonte: <<https://www.navicat.com/products/navicat-premium>>

Navicat Premium é uma ferramenta de administração de banco de dados (como mostra a figura 8), que lhe permite conectar simultaneamente a MySQL, MariaDB, SQLServer, Oracle, PostgreSQL e bancos de dados SQLite a partir de uma única aplicação.

Navicat premium combina funções de outros produtos navicat e suporta a maioria dos recursos usados em sistemas de gerenciamento de banco de dados modernos, tais como procedimentos armazenados e eventos, triggers, funções, vistas, etc.

Navicat Premium permite transferir dados de forma eficiente em vários sistemas de banco de dados ou arquivos de texto simples com formato de SQL e codificação. Trabalhos em lotes para diferentes tipos de bancos de dados pode ser programado para ser executado em um momento específico. Outras características incluem um Export/ Import Wizard, o Query Builder, o Report Builder, sincronização de dados, Backup, Job Scheduler, e muito mais.

Navicat fornece aos desenvolvedores profissionais com características sofisticadas específicas para suas necessidades, mas é fácil de aprender para novos usuários para os servidores de banco de dados.<sup>6</sup>

---

<sup>5</sup> < <https://support.office.com/pt-br/article/Access-SQL-conceitos-b%C3%A1sicos-vocabul%C3%A1rio-e-sintaxe-444d0303-cde1-424e-9a74-e8dc3e460671>>

<sup>6</sup> < <https://www.navicat.com/products/navicat-premium>>

## 2 DESENVOLVIMENTO DO SOFTWARE

### 2.1 MÉTODOS E FERRAMENTAS UTILIZADAS

De início, foi feito um estudo maior em relação à linguagem de programação PHP 5, HTML 5 e CSS3 para o desenvolvimento do painel administrativo. Foi criado um *template* para melhor visualização das postagens publicadas no Admin do site.

Através do SGBD Navicat Premium, citado no item 1.6.1.1, foi criado um banco de dados que terá fácil manipulação através do SGBD, a base do banco do sistema armazena as informações criadas no sistema do site. Tanto dados de navegação, quanto dados de postagens e também dados dos usuários do sistema.

Para o desenvolvimento do sistema foi utilizado uma IDE gratuita NetBeans na versão 8.1 que ajuda a organizar as linguagens de programação. O acesso ao sistema é feito por navegador Web local.

O sistema desenvolvido será hospedado no endereço [www.mucurinonet.com.br](http://www.mucurinonet.com.br), em ambiente local e fica armazenado no servidor local Wampp.

#### 2.1.1 NetBeans

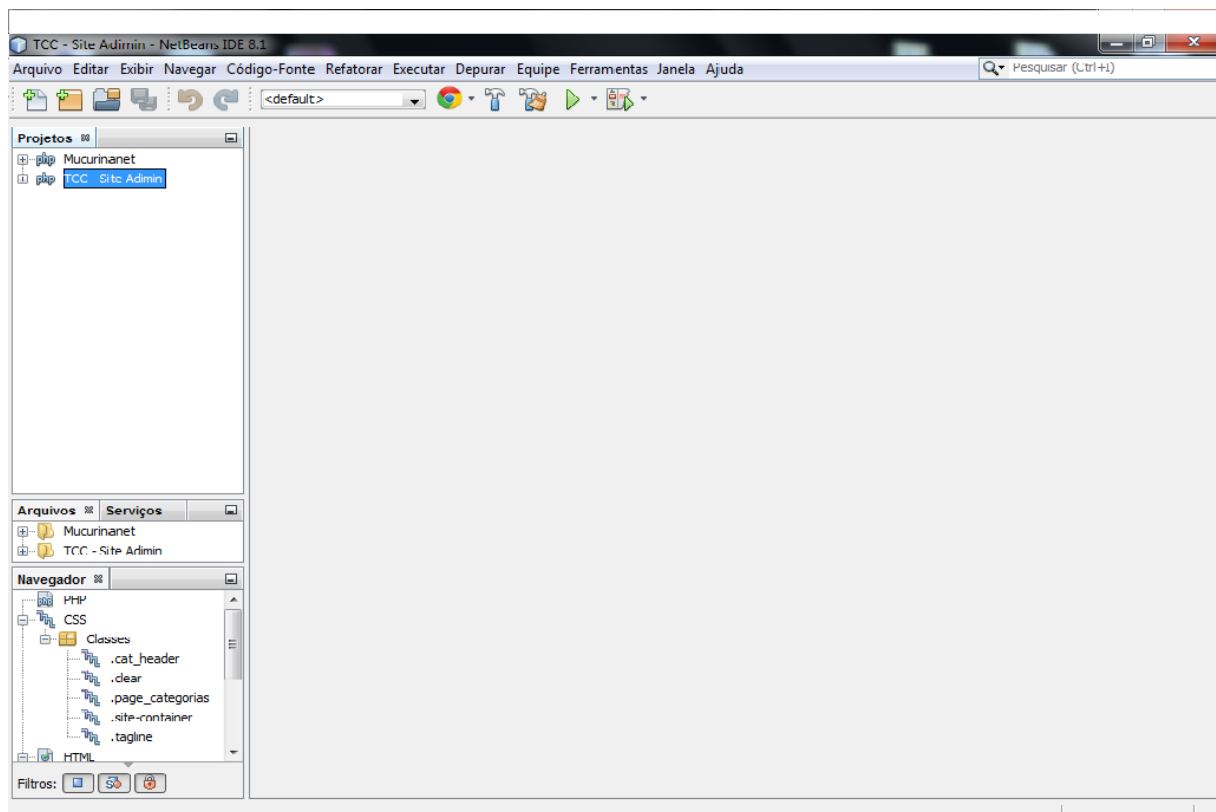
O NetBeans é um projeto open-source livre para ser usado comercialmente e não comercialmente. O código fonte é livre para reuso dentro dos termos de uso. Tendo mais de 18 milhões de downloads do NetBeans IDE e mais de 800.000

desenvolvedores. O projeto prospera e continua a crescendo, graças a sua facilidade de uso.<sup>7</sup>

O IDE NetBeans permite que você rapidamente e facilmente desenvolva em Java, para aplicativos móveis e aplicações web, bem como HTML5 com HTML, JavaScript e CSS. O IDE fornece também um grande conjunto de ferramentas para PHP e C/C ++. É livre e de código aberto e tem uma grande comunidade de usuários e desenvolvedores ao redor do mundo.<sup>8</sup>

A figura 9 representa a tela principal do NetBeans.

**FIGURA 9 – NETBEANS**



Fonte: próprio autor

---

<sup>7</sup> <https://netbeans.org/about/index.html>

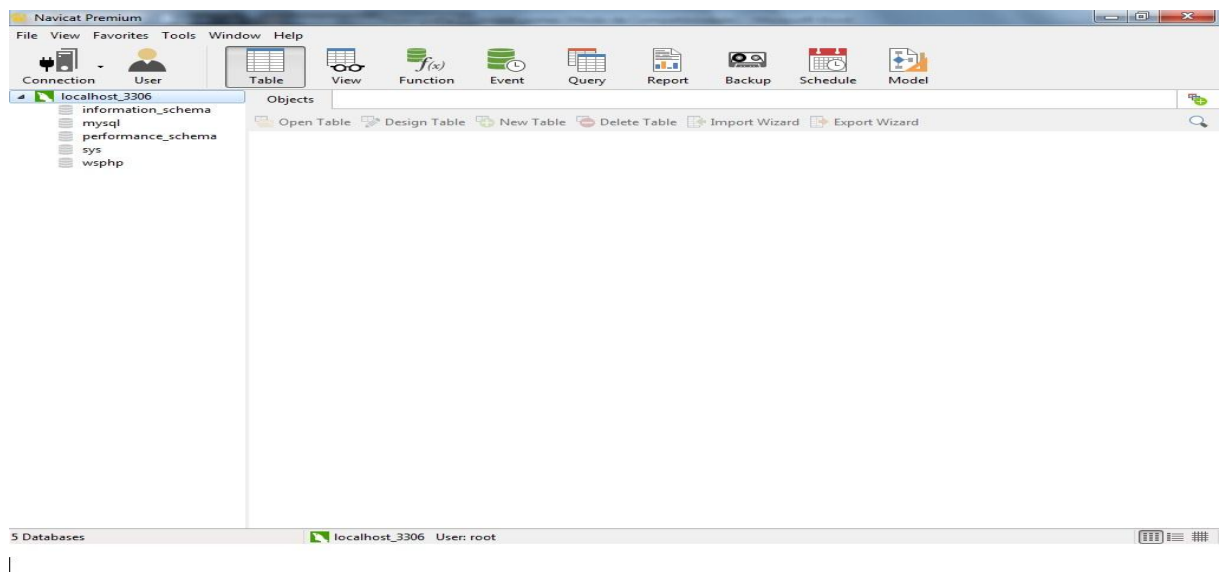
<sup>8</sup> <https://netbeans.org/features/index.html>

## 2.1.2 Navicat Premium

A Navicat Premium uma ferramenta rápida, confiável e acessível. O programa foi construído para simplificar o gerenciamento do banco de dados, esquematizado para acolher as necessidades dos administradores dos bancos de dados e desenvolvedores de todas as empresas. Nele é permitido criar, organizar, acessar e compartilhar informação do banco de forma segura e fácil. Ele é conhecido e confiável, utilizado ao redor do mundo todos os dias por empresas globais, instituições educacionais e agencias governamentais. Disponível para MySQL, SQLite, PostgreSQL e Oracle para administração e desenvolvimento de banco de dados local ou remoto.<sup>9</sup>

A figura 10 mostra a inicialização do Navicat Premium, já feita a conexão com o Banco.

**FIGURA 10 - NAVICAT PREMIUM**



Fonte: próprio auto

---

<sup>9</sup> [http://www.software.com.br/p/navicat?gclid=CjwKEAjw-Oy\\_BRDg4lqok57a4kcSJADsuDK1g59Pf4CAQL9zmy3k8pCXENObjKWgsMpj6cn8iieuEhoCazrw\\_wcB](http://www.software.com.br/p/navicat?gclid=CjwKEAjw-Oy_BRDg4lqok57a4kcSJADsuDK1g59Pf4CAQL9zmy3k8pCXENObjKWgsMpj6cn8iieuEhoCazrw_wcB)

### 2.1.3 WampServer

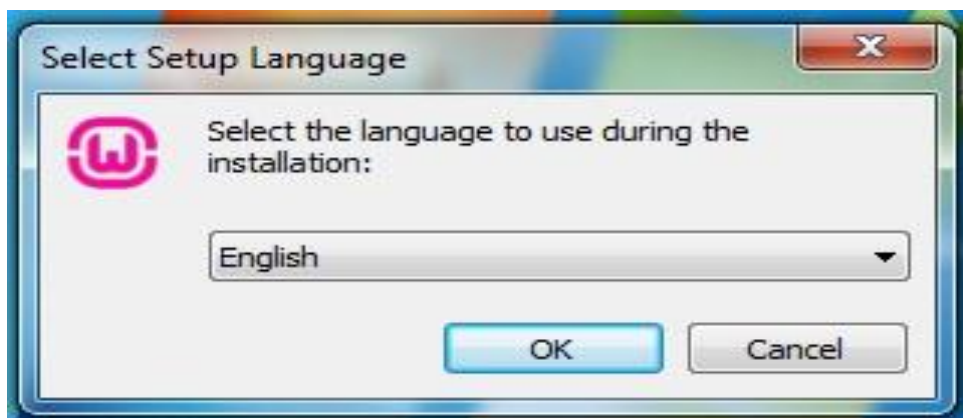
O Wampserver versão 3.0.4, disponibiliza todos os recursos necessários para criação do ambiente de desenvolvimento Web. Vem com um pacote de programas com Apache, PHP, MySQL, PHPMyAdmin, SqlBuddy e XDebug. O que foi utilizado nesta pesquisa foi Apache que é o servidor Web local compatível com o protocolo HTTP dos *browsers Web*, já o PHP é o responsável por processar a programação e os scripts. O MySQL é o sistema gerenciador do banco de dados, atualmente o mais usado no mercado, ele funciona de forma estruturada e relacional de código aberto utilizado na maioria das aplicações gratuitas para gerir banco de dados, utiliza linguagem SQL (*Structured Query Language* – Linguagem de consulta estruturada), como interface, permitindo que pudesse ser realizados testes localmente.

#### 2.1.3.1 Instalação do WampServer 3

Após ter baixado o WampServer 3 no link (<http://www.wampserver.com/en/>), deverá ser executado para iniciar a instalação.

A figura 11 ilustrada abaixo mostra a tela de inicialização da instalação do WampServer 3, que pede a língua para instalação.

**FIGURA 11 - ESCOLHA DA LÍNGUA DE INSTALAÇÃO**

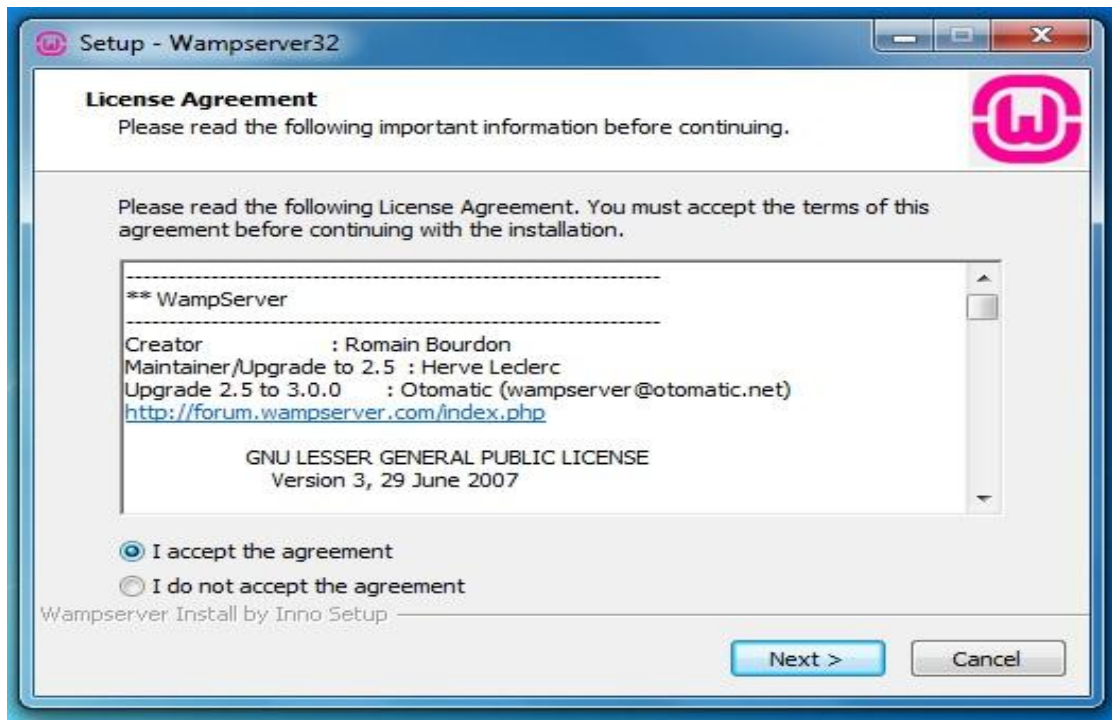


Fonte: próprio autor



Após a escolha da língua, deve-se aceitar os termos de licença para prosseguir a instalação como mostra a figura 12.

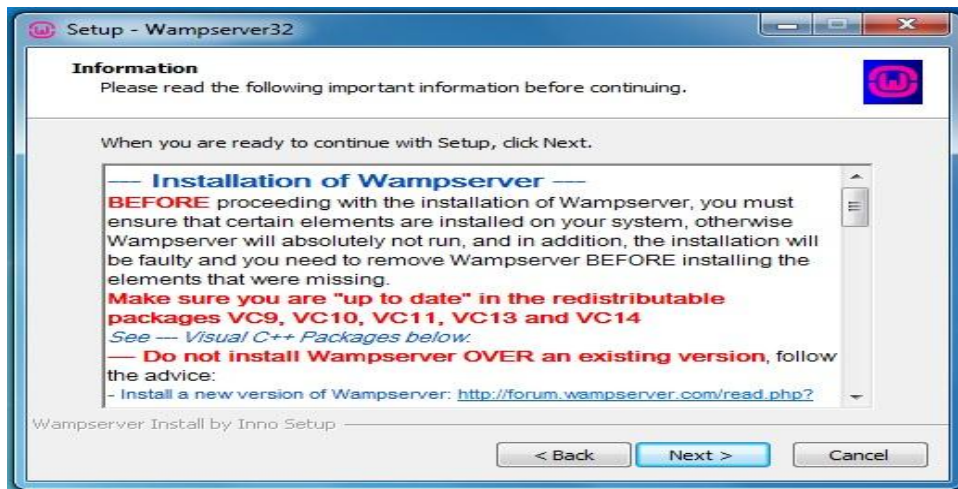
**Figura 12 – Termos de licença do WampServer 3**



Fonte: próprio autor

Após aceitar os termos o WampServer 3 pede para ler algumas informações de uso como mostra na figura 13, leia e aceite para prosseguir a instalação.

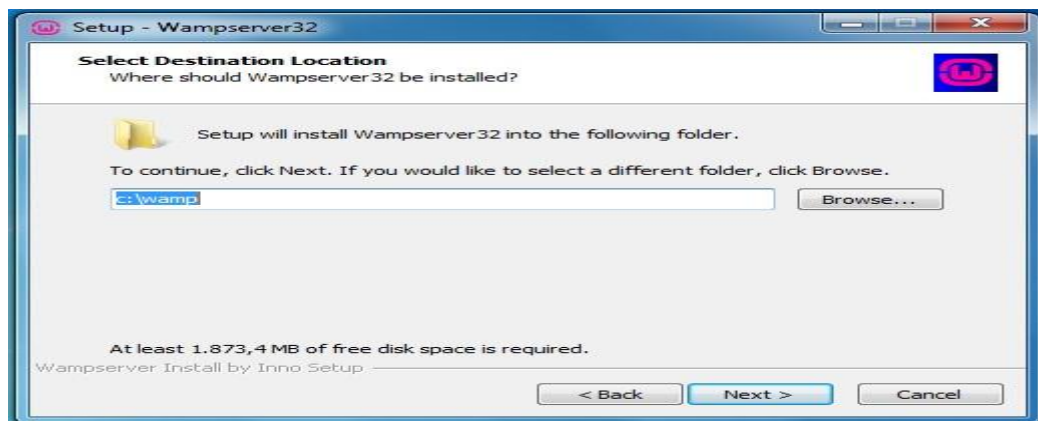
**Figura 13 – Informações do WampServer 3**



Fonte: próprio autor

O diretório de instalação padrão do WampServer é a unidade principal do computador, onde fica instalado o sistema operacional do computador. Basta manter o local e seguir com a instalação como mostra a figura 14.

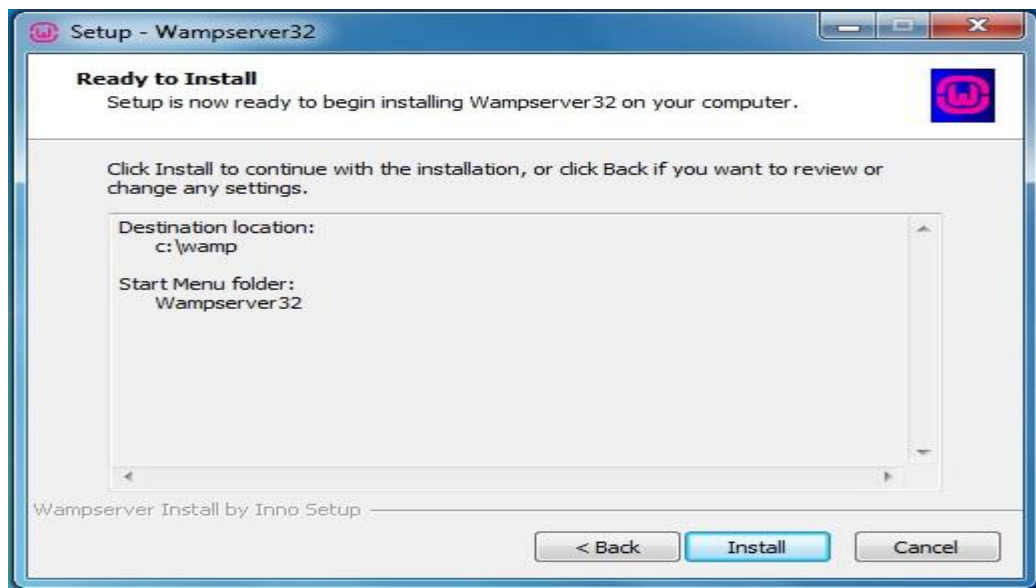
**Figura 14 – Definição do diretório do WampServer 3**



Fonte: próprio autor

Depois de ter feito toda a configuração correta, instala-se o WampServer 3, prosseguindo para a próxima etapa como mostra na figura 15.

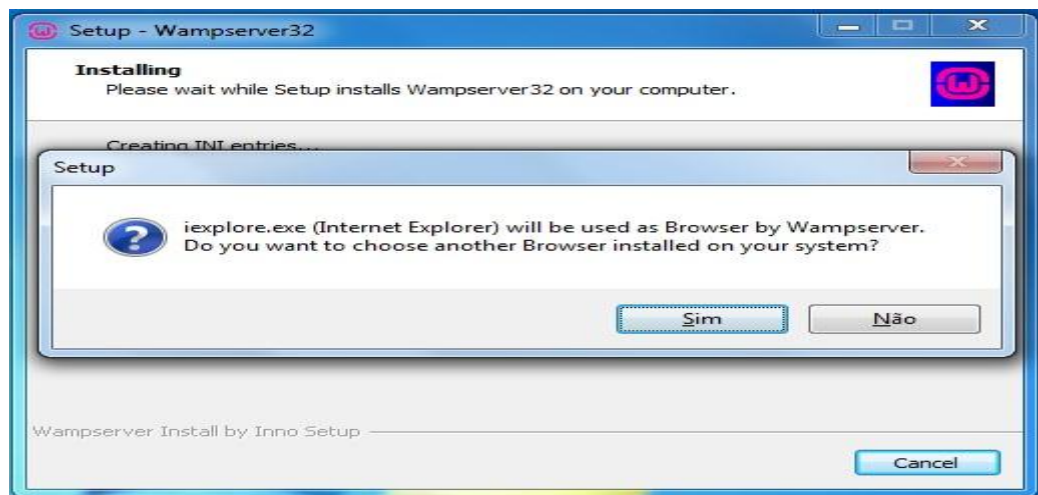
**Figura 15 – Instalação do WampServer 3**



Fonte: próprio autor

Seguindo com as configurações necessárias para o ambiente de desenvolvimento, o programa pede para selecionar qual navegador padrão o WampServer 3 deverá usar, como mostra na figura 16. Vem por padrão o Internet Explorer; se desejar mudar clique em não, ou clique em sim e prossiga a instalação.

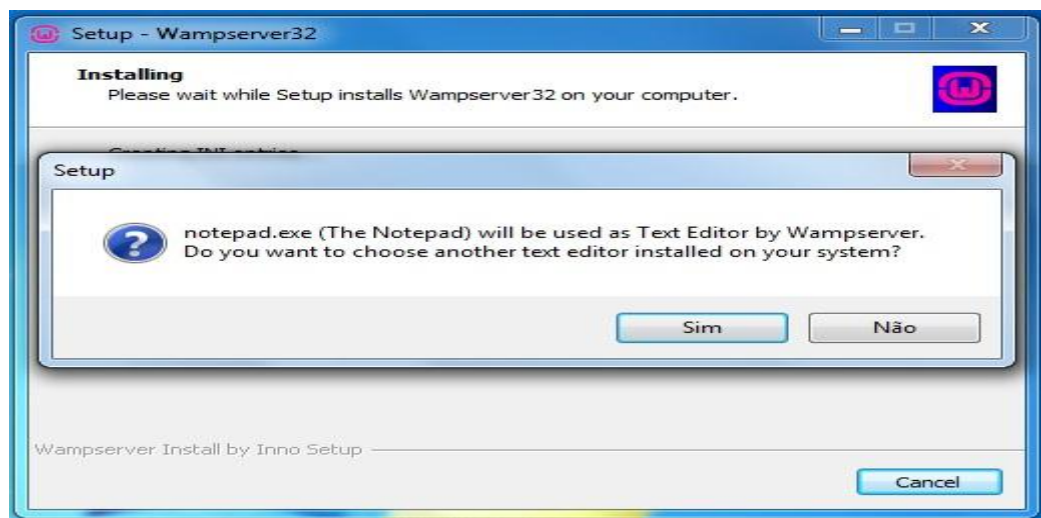
**Figura 16** – Escolha do *Browser* do WampServer 3



Fonte: próprio autor

Após escolha do navegador padrão é pedido o editor de texto padrão, clique em sim e siga a instalação como mostra na figura 17.

**Figura 17** – Escolha do editor de texto do WampServer 3



Fonte: próprio autor

Logo após a instalação e configuração do WampServer 3 mostrará uma tela com os pacotes instalados como mostra a figura 18.

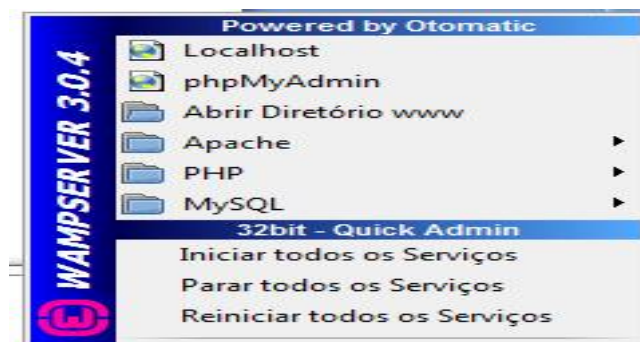
**Figura 18** – Pacotes instalados junto ao WampServer 3



Fonte: próprio autor

Como mostra a figura 19 todos os serviços devem ser inicializados, para que o ambiente de desenvolvimento esteja corretamente configurado.

**Figura 19** – Iniciando os serviços do WampServer 3



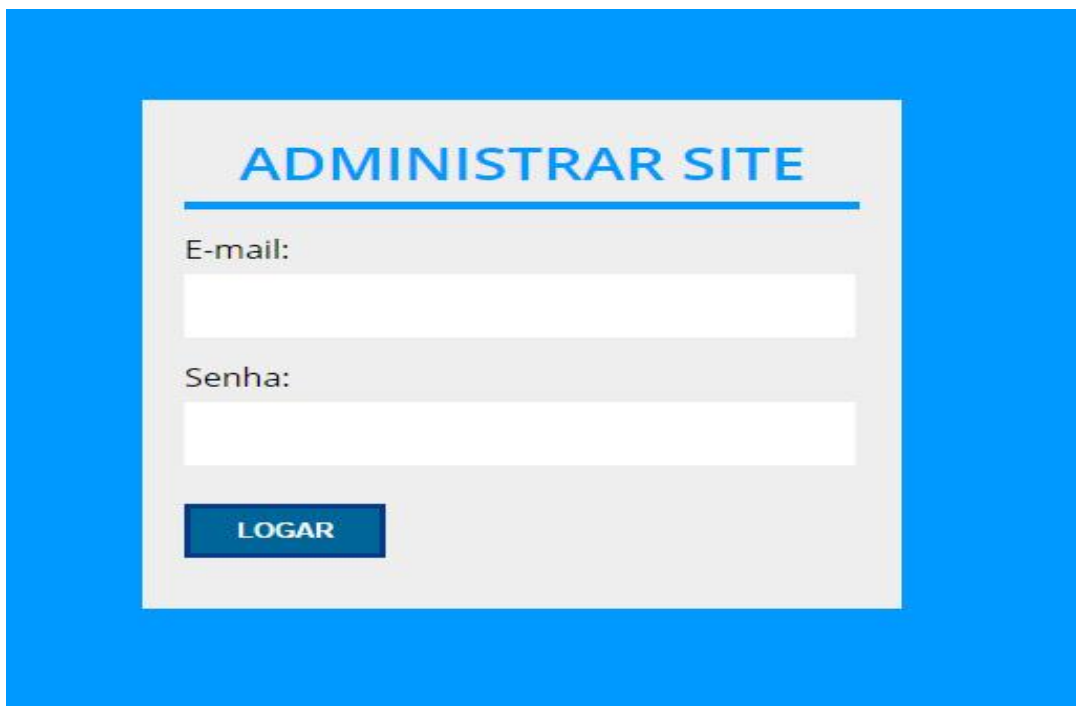
Fonte: próprio autor

## 2.2 O SOFTWARE

Neste subcapítulo serão apresentados os layouts desenvolvidos para o sistema, que tem como Template um portal de notícias da região do Vale do Mucuri. Serão apresentadas também todas as funções do painel administrativo e do site.

De acordo com a figura 20, logo abaixo, inicia o software com a tela de login, sendo que só os administradores do site podem usar essa tela.

**Figura 20** – Tela de Login

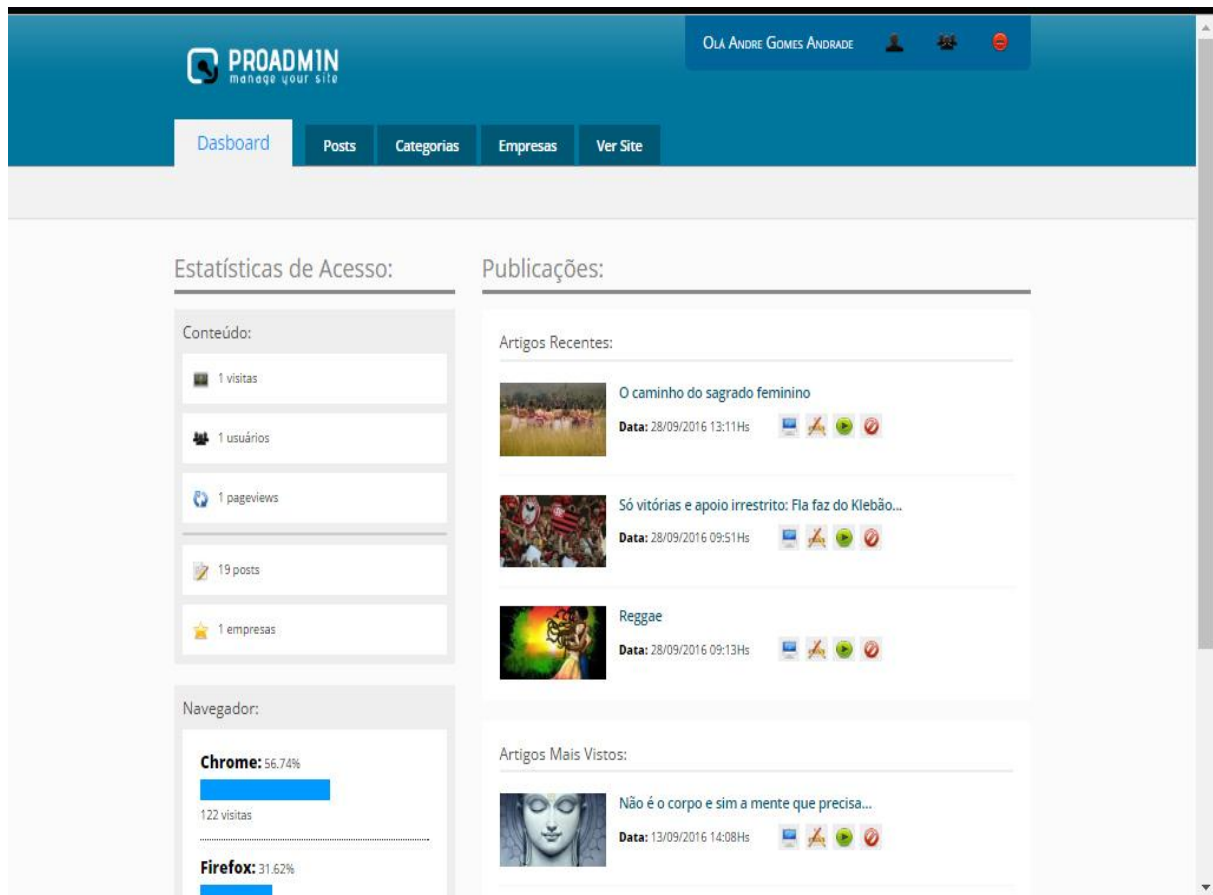
A imagem mostra a tela de login do sistema administrativo. O título "ADMINISTRAR SITE" está em azul no topo. Abaixo dele, há dois campos de entrada: "E-mail:" e "Senha:". Cada campo é seguido por uma barra de entrada de texto. No final, há um botão azul com o texto "LOGAR" em branco.

Fonte: próprio autor

Após colocar e-mail e senha corretos, o painel vai para a página principal, onde mostra as publicações mais recentes, controle de tráfego por navegador e tem links para adicionar e editar postagens, cadastrar e editar empresas e criar

categorias para as postagens. Como mostra na figura 21. Também há o controle de usuários logo acima.

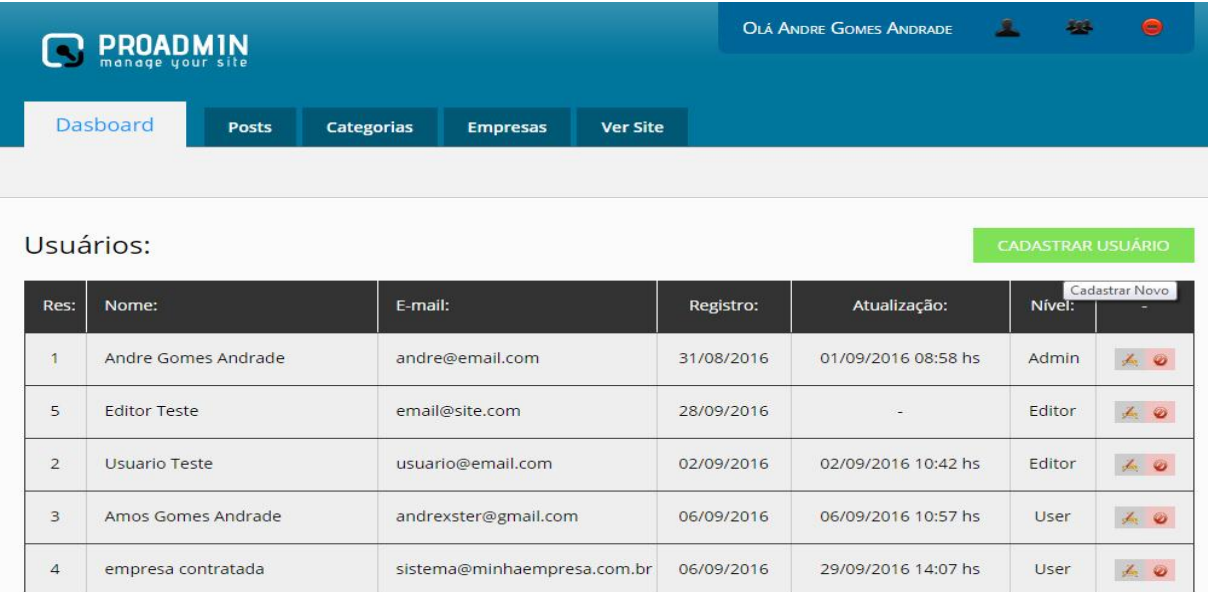
**Figura 21 – Tela Principal**













Fonte: próprio autor

A figura 22 mostra o controle de usuários do site, onde aparece Editor, Admin e Usuário. O usuário tem acesso apenas ao site, o editor pode editar algumas matérias e o administrador ativa e desativa postagens, controla os usuários, podendo cadastrar e excluir usuários. A painel Administrativo só pode ser acessado pelo administrador, sendo que pelo código pode ser alterado por permissões de níveis 1, 2 e 3.

**Figura 22 – Controle de Usuários**



Usuários: CADASTRAR USUÁRIO

Res:	Nome:	E-mail:	Registro:	Atualização:	Nível:	Cadastrar Novo
1	Andre Gomes Andrade	andre@email.com	31/08/2016	01/09/2016 08:58 hs	Admin	 
5	Editor Teste	email@site.com	28/09/2016	-	Editor	 
2	Usuario Teste	usuario@email.com	02/09/2016	02/09/2016 10:42 hs	Editor	 
3	Amos Gomes Andrade	andrexster@gmail.com	06/09/2016	06/09/2016 10:57 hs	User	 
4	empresa contratada	sistema@minhaempresa.com.br	06/09/2016	29/09/2016 14:07 hs	User	 

Fonte: próprio autor

A figura 23 mostra o controle de tráfego de navegadores, mostrando qual o navegador mais utilizado para uso do site.

**Figura 23 – Tráfego por navegador**



Fonte: próprio autor



A figura 24 mostra a tela de cadastro de postagens onde coloca uma imagem, um título, o conteúdo das postagens, data, categoria e o autor; por último pode-se colocar uma galeria de imagens referente à postagem.

**Figura 24 – Cadastro de Post**

Criar Post:

Enviar Capa:  
[Escolher arquivo] Nenhum arquivo selecionado

Titulo:  
[ ]

Conteúdo:  
[Rich text editor with toolbar]

Endereço: [ ] Words: 0

Data: [13/10/2016 09:38:19] Categoria: [Selezione a categoria: v] Author: [Andre Gomes Andrade v]

Enviar Galeria:  
[Escolher arquivos] Nenhum arquivo selecionado

[CADASTRAR] [CADASTRAR & PUBLICAR]

Fonte: próprio autor

A Figura 25 mostra as categorias cadastradas no painel, para prosseguir deve-se ter categorias cadastradas.

**Figura 25 – Cadastro de Post/Categorias**

Criar Post:

Enviar Capa:  
 Nenhum arquivo selecionado

Título:

Conteúdo:

Endereço:

Data:

Selecione a categoria:  
 Baladas  
 »» Cartos Chagas  
 »» Nanuque  
 »» Teofilo Otoni  
 esportes  
 »» Futebol  
 Eventos  
 musicas  
 »» Eletronicas  
 »» Reggae  
 Noticias  
 »» aconteceu  
 »» Cartos Chagas  
 »» geral  
 »» Nanuque  
 »» Teofilo Otoni  
 Selecione a categoria: ▾

Words: 0

Author:

Fonte: próprio autor

A figura 26 mostra os autores cadastrados, que são os usuários, sendo que só o editor e o administrador que podem cadastrar notícias.

**Figura 26 – Cadastro de Post/Autores**

Enviar Capa:  
 Nenhum arquivo selecionado

Título:

Conteúdo:

Endereço:

Words: 0

Data:

Categoria:  
 Selecione a categoria: ▾

Author:

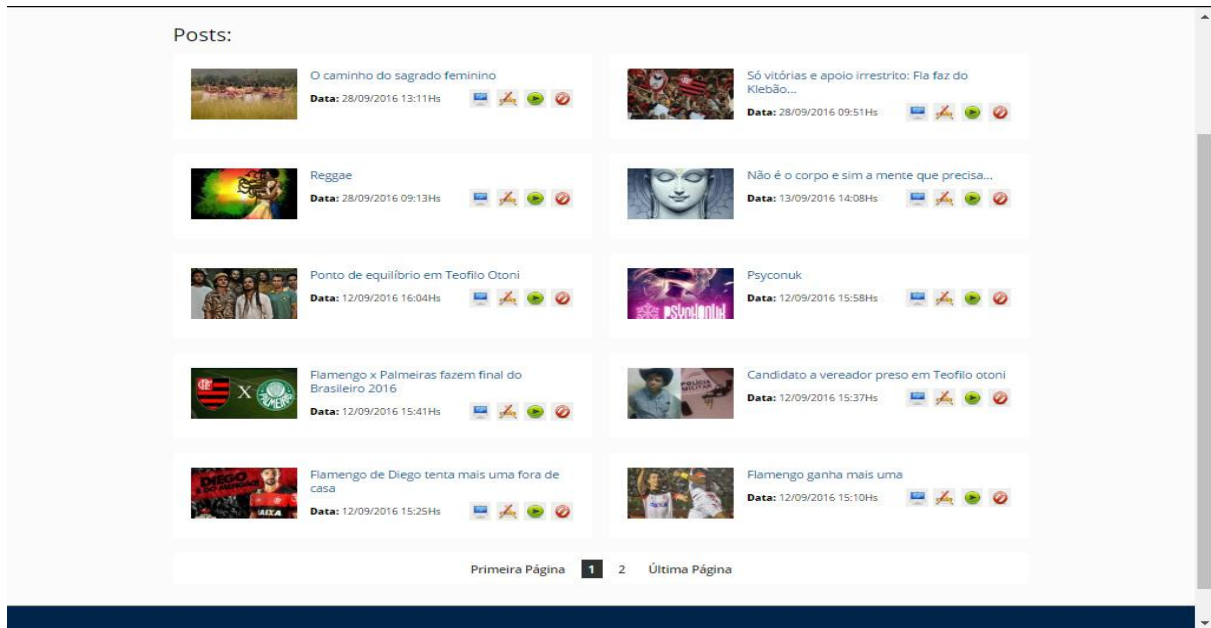
Enviar Galeria:  
 Nenhum arquivo selecionado

CADASTRAR CADASTRAR & PUBLICAR

Fonte: próprio autor

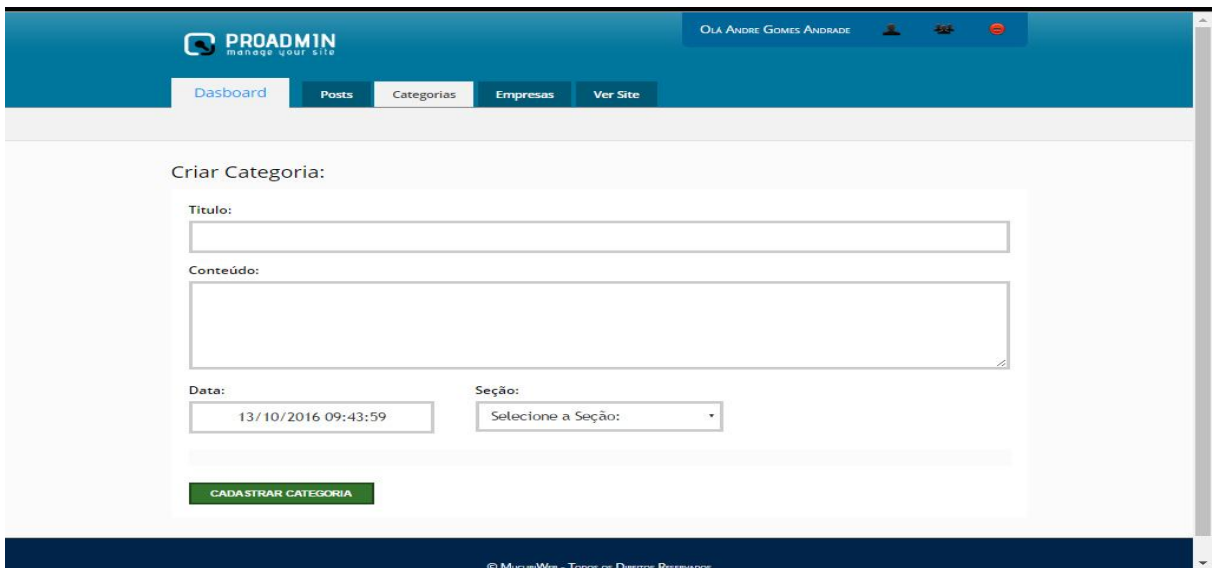
A figura 27 vem com a tela de edição de postagens, onde o administrador do site pode ver as postagens ativas e desativas do site, ele pode editar, excluir e ativar ou desativar as postagens, também pode ver elas no site.

**Figura 27 – Edição de Post**



Fonte: próprio autor

Para criar uma categoria como mostra a figura 28, deve-se cadastrar um título, um conteúdo e uma data e uma seção, que define se a categoria vai ser mãe ou filha. A categoria mãe engloba as categorias filhas, que são onde ficam as matérias. Apenas o administrador do site tem acesso a essa página.

**Figura 28 – Cadastro de categorias**

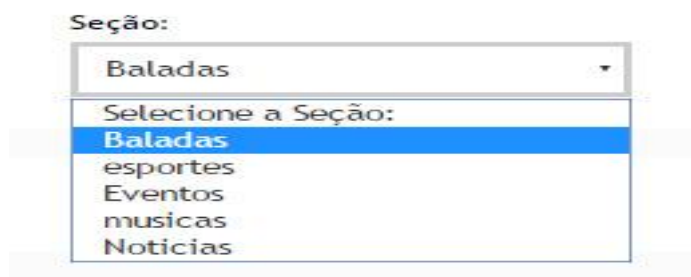
The screenshot shows the PROADMIN web application interface. At the top, there is a navigation bar with the logo 'PROADMIN manage your site' and the user name 'OLA ANDRE GOMES ANDRADE'. Below the navigation bar, there are tabs for 'Dashboard', 'Posts', 'Categorias', 'Empresas', and 'Ver Site'. The main content area is titled 'Criar Categoria:' and contains a form with the following fields:

- Título:** A text input field.
- Conteúdo:** A large text area for content.
- Data:** A date and time input field showing '13/10/2016 09:43:59'.
- Seção:** A dropdown menu with the text 'Selecione a Seção:' and a downward arrow.

At the bottom of the form is a green button labeled 'CADASTRAR CATEGORIA'. The footer of the page contains the text '© Microweb - Todos os Direitos Reservados'.

Fonte: próprio autor

As subcategorias são criadas escolhendo em qual categoria mãe elas devem ficar como mostra a figura 29.

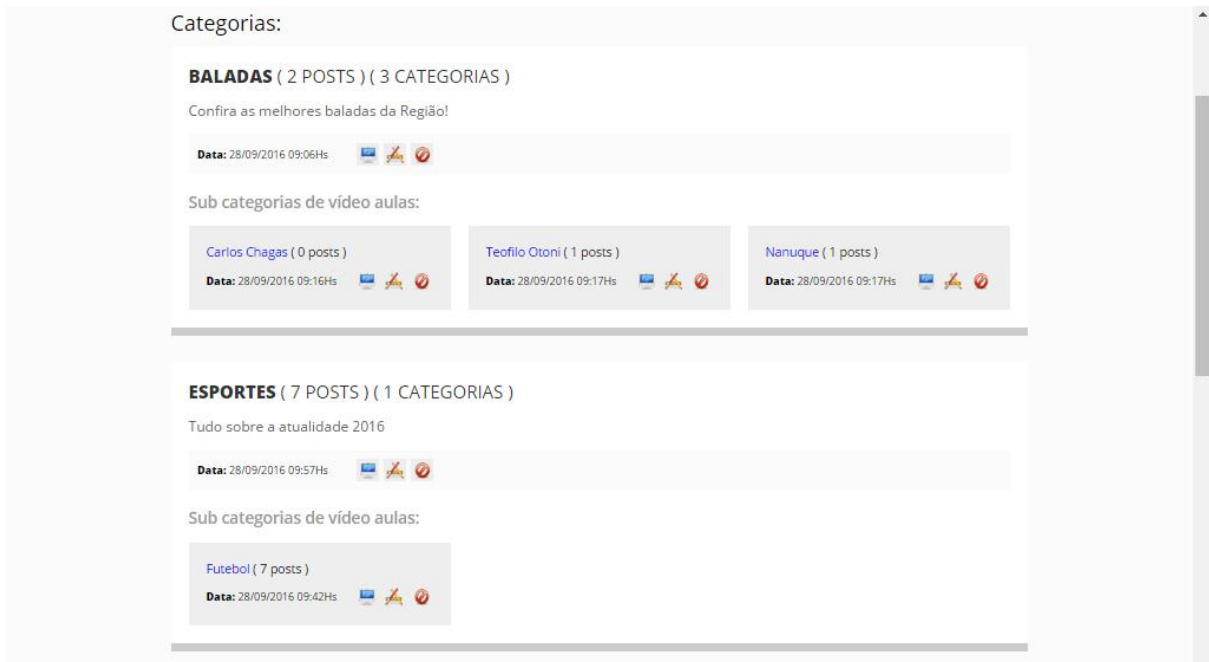
**Figura 29 – Cadastro de Subcategorias**

The screenshot shows a dropdown menu labeled 'Seção:'. The menu is open, displaying a list of options. The first option is 'Baladas', which is currently selected. Below it, there is a header 'Selecione a Seção:' followed by a list of subcategories: 'Baladas', 'esportes', 'Eventos', 'musicas', and 'Noticias'.

Fonte: próprio autor

A próxima tela como mostra a figura 30, separa as categorias mães e filhas, para que possam ser visualizadas e editadas, ativadas e desativadas.

**Figura 30 – Tela de edição de categorias**



Fonte: próprio autor

A tela a seguir mostra como ver a categoria no site, onde mostra todas as postagens daquela categoria, como mostra a figura 31.

**Figura 31 – Tela de edição de categorias/ Ver no site**



Fonte: próprio autor

A próxima tela como mostra a figura 32, é a tela de cadastro de empresas, onde adiciona uma imagem, o nome da empresa, o ramo de atividade da empresa, site da empresa, facebook da empresa, endereço e indicação que será mostrado na tela a seguir.

**Figura 32 – Cadastro de empresas**

Cadastrar Empresa:

Logo da empresa: Exatamente 578x288px (JPG ou PNG)  
Escolher arquivo Nenhum arquivo selecionado

Nome da Empresa:

Ramo de atividade:

Sobre a empresa:

Site da Empresa:  Facebook Page:

Nome da rua / Número:

Estado UF:  Cidade:  Indicação:

© MUCURIWEB - TODOS OS DIREITOS RESERVADOS

Fonte: próprio autor

Após cadastrar a empresa, vem a parte da indicação que é a separação do ramos das empresas, que tem onde ficar, comer, se divertir e comprar, como mostra a figura 32.

Essa parte do painel mostra a edição do cadastro de empresas, ilustrado na figura 33.

**Figura 33 – Tela de edição de empresas**

The screenshot displays the PROADMIN web application interface. At the top, there is a header with the PROADMIN logo and the text 'manage your site'. To the right of the header, the user's name 'OLA ANDRE GOMES ANDRADE' is visible. Below the header, a navigation menu contains the following items: 'Dashboard', 'Posts', 'Categorias', 'Empresas' (which is highlighted), and 'Ver Site'. The main content area is titled 'Empresas:' and contains a grid of eight company entries. Each entry consists of a logo, the company name, the user 'CARLOS CHAGAS / MG', and the creation date. The entries are:

Logo	Company Name	User	Date
Bistrô Dos anjos	Bistrô Dos anjos	CARLOS CHAGAS / MG	13/10/2016 11:09Hs
Hotel CarlosChaguense	Hotel CarlosChaguense	CARLOS CHAGAS / MG	13/10/2016 11:06Hs
Hotel Fic2	Hotel Fic2	CARLOS CHAGAS / MG	13/10/2016 10:33Hs
Lanchonete	Lanchonete	CARLOS CHAGAS / MG	13/10/2016 11:12Hs
Bistrô Dos anjos	Bistrô Dos anjos	CARLOS CHAGAS / MG	13/10/2016 11:09Hs
Hotel Fic1	Hotel Fic1	CARLOS CHAGAS / MG	13/10/2016 10:31Hs
Hotel Tchotcho	Hotel Tchotcho	CARLOS CHAGAS / MG	13/10/2016 11:08Hs
Restaurante Fic	Restaurante Fic	NULL / MG	13/10/2016 11:11Hs

At the bottom of the page, there is a footer with the text '© MucurWeb - Todos os Direitos Reservados'.

Fonte: próprio autor

### 3 REULTADOS E DISCUSSÕES

O desenvolvimento do painel administrativo trouxe uma visão contextualizada, mostrando diversas vezes que existiam possibilidades teóricas adaptadas e outras eliminadas durante o processo de desenvolvimento, levando em conta o conhecimento obtido através do referencial teórico, e identificação de problemas específicos com relação às necessidades práticas do sistema.

O desenvolvimento do painel administrativo veio através da aplicabilidade dos conceitos elaborados no capítulo do referencial teórico.

Ao longo do desenvolvimento houve a necessidade de realizar busca por mais conhecimentos, como a prática da linguagem PHP5, CSS3 e HTML5, a construção do banco de dados. Apesar disso, o software atende as necessidades específicas como o cadastro de empresas, o cadastro de postagem, levando em consideração as necessidades do portal de notícias vinculado ao painel administrativo.

Percebeu-se que não é relevante o desenvolvimento de funcionalidades sofisticadas, já que não seria aplicado no uso do projeto. Tendo em vista a associação da teoria e da prática, o corpo da pesquisa vai conter uma aplicação prática e de fácil uso para o usuário, o editor e principalmente o administrador.

Através de referências bibliográficas e de um estudo complementar, teve-se a percepção da necessidade do estudo específico do conceito de desenvolvimento web. As referências sobre PHP e também toda estrutura do ambiente web, proporcionaram uma noção inicial para a prática do desenvolvimento do painel com os conceitos aprendidos.

A construção do banco de dados foi feita através de estudo e de scripts associados ao site. Todo ele é relacionado ao painel administrativo que passa informações ao site, que funciona localmente.



Sobre os resultados obtidos no desenvolvimento do software, conclui-se que houve melhorias em relação às habilidades e conhecimentos sobre ambiente Web e também aprimorou-se os conhecimentos em relação à programação Web. Em relação ao painel, ele atende as necessidades do tema proposto para a Monografia, servindo ao site em tempo real quando for colocado em servidor web, o portal de notícias do vale do mucuri pode vir a ser colocado no ar futuramente.

É relevante lembrar que a TI foi primordial à pesquisa, uma vez que abrange toda tecnologia necessária para o desenvolvimento da plataforma para desenvolvimento do painel e do site. A TI é uma aliada muito favorável para micro empreendedores por ter uma agilidade e organização muito grande e facilidade de uso. Seus procedimentos são confiáveis, pois são descartados procedimentos inseguros sem projeção alguma ao crescimento. Existe um grande receio de empresas de pequeno porte em relação ao uso de tecnologia, devido à falta de conhecimento do que a TI pode favorecer dentro de uma empresa, mas esse cenário está mudando com a informatização de várias microempresas.

## CONCLUSÃO

Através do trabalho proposto acima, foi possível constatar a necessidade do desenvolvimento de um painel administrativo para um portal de notícias na região, para proporcionar agilidade ao site e otimizar todas as funcionalidades do mesmo.

Foi realizada uma análise de viabilidade de implantação de um painel administrativo, por meio de objetos específicos descritos logo abaixo com finalidade de atingir o objetivo geral do projeto.

**Diagnosticar problemas que possam atrapalhar o andamento do site:** Através de diagnósticos de dificuldades encontradas ao longo do desenvolvimento da pesquisa.

**Realizar uma pesquisa bibliográfica para dar direção mais exata ao desenvolvimento do site:** Através de estudos direcionados à programação web e banco de dados foi possível desenvolver um planejamento para todo o desenvolvimento.

**Desenvolver um painel administrativo que atenda às necessidades do site:** A interface gráfica foi criada especialmente para o painel administrativo, com interação fácil para o usuário.

**Testar todas as funcionalidades do software:** Realizar testes simulando um ambiente real onde todas as funcionalidades do sistema foram testadas, prevenindo falhas no sistema futuramente e otimizando os serviços do painel, dando uma maior segurança ao sistema.

**Fornecer a primeira versão do sistema com todas as necessidades testadas e funcionando:** Após todo o ambiente preparado, instalado e configurado corretamente para um melhor funcionamento de todo o processo de desenvolvimento, e uma melhor eficiência nas funcionalidades disponíveis.

Em relação ao problema abordado, foram criadas as seguintes repostas para as hipóteses desta pesquisa, classificando-as como nulas, inválidas e válidas:

**H0: Não seria viável a criação de um painel administrativo para um portal de notícias, pois o mesmo poderia não suprir as necessidades do mercado.**

Está hipótese é nula pois o painel foi desenvolvido com todas as novas tecnologias do mercado e atende todas as funcionalidades do site.

**H1: Não seria viável colocar o site em um servidor web, pois o custo seria alto para um investimento inicial.**

Está hipótese é inválida pois o site pode gerar um lucro maior do que os gastos que teria para estar no ar.

**H2: Seria viável a construção de um painel administrativo para manutenção do portal de notícias, que seria alimentado através do painel.**

Foi possível validar essa hipótese, pois um sistema precisa de um painel para edição, cadastro e manutenção do mesmo.

**H3: Seria viável a criação de um portal de notícias para a região para mostrar as notícias em tempo real e teria empresas parceiras para manter o site funcionando perfeitamente.**

Foi possível validar essa hipótese, pois a região atende a necessidade do site, com poucas empresas cadastradas no portal ele teria recursos necessários para se manter no ar.

Ao final, mediante estudo bibliográfico ao longo do curso que proporcionou uma estrutura para desenvolvimento do projeto teórico e para o desenvolvimento do painel administrativo, são perceptíveis as contribuições no âmbito acadêmico e pessoal, onde obteve-se aprimoramento dos conhecimentos dos conteúdos relacionados. O conhecimento pessoal foi ampliado por meio de pesquisas e estudos propostos para a realização desta pesquisa. Essa monografia tem uma relevância grande e sistêmica para o profissional da área através da crítica, análise e reflexão em relação ao problema solucionado, preparando o profissional para o mercado de trabalho, trazendo autonomia e flexibilidade no mercado de trabalho, para um destaque pessoal do profissional nas atividades relacionadas ao desenvolvimento.

Após a conclusão do trabalho foi possível apresentar sugestões para próximos trabalhos, pensando em sempre melhorar o *software* já em funcionamento, buscando inovações como vender produtos diretamente do site e também um sistema responsivo que se adapta a qualquer tamanho de tela.

## REFERÊNCIAS

BITTENCOURT, Rogério Gonçalves. *Aspectos Básicos de Banco de Dados*. 1ed. Florianópolis: Adobe® Acrobat eBookReader™, 2004.

BONIATI, Bruno Batista; SILVA, Teresinha Letícia. *Fundamentos do desenvolvimento Web*. 1 Ed. Rio Grande do Sul. Colégio Agrícola de Frederico Westphalen, 2013.

CORNACHIONE, Edgard B. Jr. *Informática: Aplicada às Áreas de Contabilidade, Administração e Economia*. 3 ed. São Paulo. Editora Atlas S.A, 2007.

DALL'OGGIO, Pablo. *PHP Programando com Orientação a Objetos*. 1. ed. São Paulo: Novatec Editora, 2007.

\_\_\_\_\_. *PHP5: Orientação a Objetos e Design Patterns*, 1 ed. 2004. Disponível em: [www.cin.ufpe.br/~jfp/~fpcursophp/~fphp5.pdf&usg=AFQjCNEVVvEud\\_ac9v4ZehIMf10yBauzWA&sig2=D6xwG8Npj1ISmfChhUi8MQ&bv m=bv.133178914,d.Y2l](http://www.cin.ufpe.br/~jfp/~fpcursophp/~fphp5.pdf&usg=AFQjCNEVVvEud_ac9v4ZehIMf10yBauzWA&sig2=D6xwG8Npj1ISmfChhUi8MQ&bv m=bv.133178914,d.Y2l)>. Acesso em: 18, Set. de 2016.

\_\_\_\_\_. *PHP-GTK: Criando Aplicações Gráficas com PHP*. 1. ed. São Paulo: Novatec Editora, 2004.

DAVID, Marcio Frayze. *Programação Orientada a Objetos*. 2007. Disponível em: <http://www.hardware.com.br/artigos/programacao-orientada-objetos/>>. Acessado em 12 de Set. 2016.

EICHSTAEDT, John F; OLIVEIRA, Djalma de Pinho Rebouças de. *Sistemas de Informações Gerenciais: Estratégicas Táticas Operacionais*. 12ª Ed. São Paulo: Editora Atlas, 2008, 299 páginas. Disponível em: [http://tupi.fisica.ufmg.br/michel/docs/Artigos\\_e\\_textos/Ciencia\\_da\\_informacao/SIG\\_conceitos.pdf](http://tupi.fisica.ufmg.br/michel/docs/Artigos_e_textos/Ciencia_da_informacao/SIG_conceitos.pdf) >. Acessado em 18 de Set. 2016.

EIS, Diego; FERREIRA, Elcio. *HTML5 e CSS3: com farinha e pimenta*. 1. ed. São Paulo. Tableless 2012.

KABIR, J. Mohammed. *Apache Server 2: A Bíblia*. 1 ed. Rio de Janeiro. Editor Campus Ltda. 2002.

KORTH, H.F. e SILBERSCHATZ. *Sistemas de Bancos de Dados, Makron Books*, 2a. edição revisada, 1994. Disponível em: <<http://www.devmedia.com.br/conceitos-fundamentais-de-banco-de-dados/1649#ixzz3bYBAENgN>>. Acessado em 15/09/2016.

MARTINZ, et al. *Tecnologia e Sistemas de Informação e Suas Influências na Gestão e Contabilidade*. 9 ed. Rio de Janeiro. 2012. Disponível em: <<http://www.aedb.br/seget/fotos2012.php>>. Acesso em: 19 Set. de 2016.

MICROSOFT. *Access SQL: conceitos básicos, vocabulário e sintaxe*. Microsoft. 2007. Disponível em: <<https://support.office.com/pt-br/article/Access-SQL-conceitos-b%C3%A1sicos-vocabul%C3%A1rio-e-sintaxe-444d0303-cde1-424e-9a74-e8dc3e460671>>. Acessado em 18 de Set. 2016.

MURPHY et al, *Introdução ao HTML5 e CSS3: A Evolução da Web*. 1 ed. Rio de Janeiro: Starlin Alta Editora. 2014.

NAVICAT. Disponível em: <<https://www.navicat.com/products/navicat-premium>>. Acessado em 18 de Set. 2016.

NETCRAFT. 2016. Disponível em: <<http://news.netcraft.com/archives/category/web-server-survey/>>. Acessado em 18/09/2016.

NEVES, Pedro M. C; Ruas, Rui P F. *O guia prático do mysql*. 1 Ed. Lisboa. Centro Atlântico, Ltda, 2005.

O'BRIEN, James A. *Sistemas de informação e as decisões gerenciais na era da internet*. 2. ed. São Paulo: Saraiva, 2004.

PRESSMAN, Roger. *Engenharia de software: uma abordagem profissional*. 7 ed. São Paulo: AMGH Editora Ltda. 2011.

SILVA, M. Samy. *HTML5: A Linguagem de marcação que revolucionou a WEB*. 1 ed. São Paulo. Novatec Editora Ltda. 2011.

\_\_\_\_\_. *jQuery: a Biblioteca do programador*. 1 ed. São Paulo. Novatec Editora Ltda. 2010.

SILVA, M. Samy. *JQuery: A biblioteca do Programador*. 3ª edição. São Paulo/SP . Novatec Editora Ltda 2010. Disponível em: <<http://livrosdomaujor.com.br/jquery/jq3/>>. Acessado em 18 de Set. 2016.

## APÊNDICE I

### O CÓDIGO

Figura 34 – Parte do código onde cria Usuário

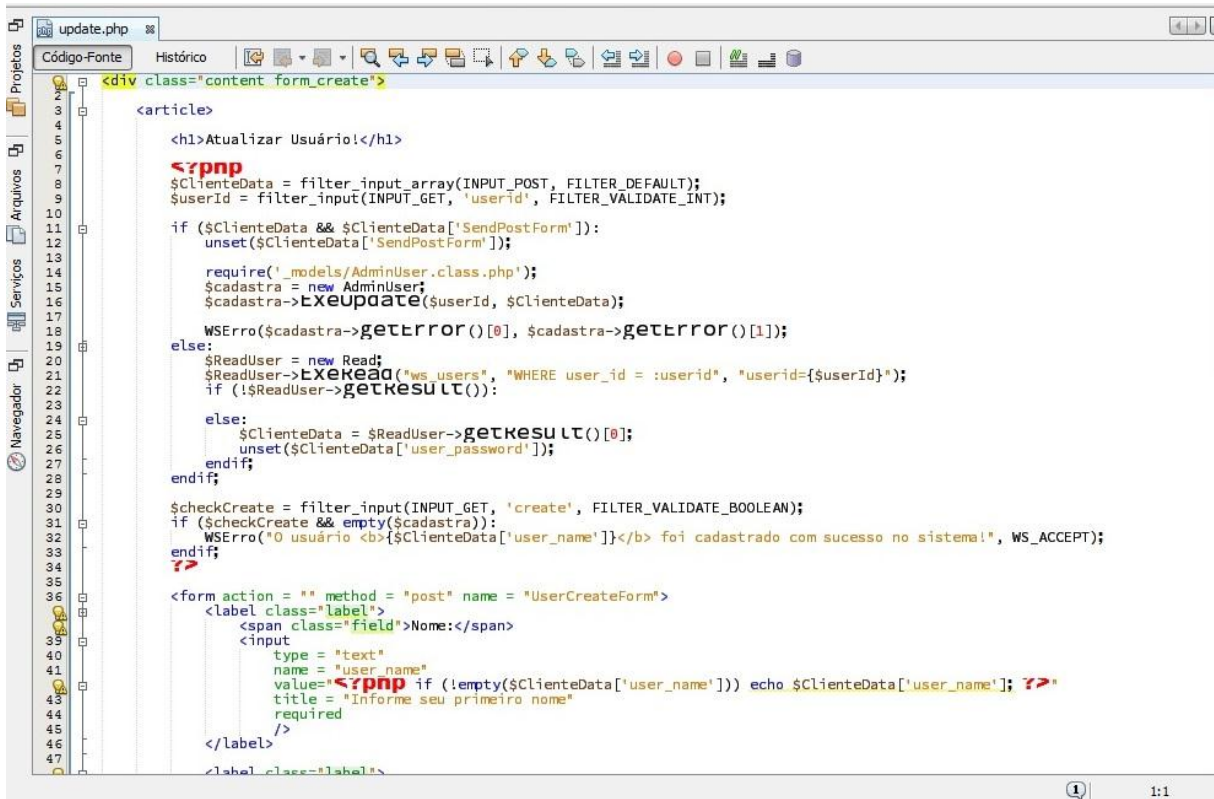
```

1 <div class="content form_create">
2
3 <article>
4
5 <h1>Cadastrar Usuário!</h1>
6
7
8 <?php
9 $ClienteData = filter_input_array(INPUT_POST, FILTER_DEFAULT);
10 if ($ClienteData && $ClienteData['SendPostForm']):
11     unset($ClienteData['SendPostForm']);
12
13     require('_models/AdminUser.class.php');
14     $cadastra = new AdminUser;
15     $cadastra->execute($ClienteData);
16
17     if ($cadastra->getResult()):
18         header("Location: painel.php?exe=users/update&create=true&userid={$cadastra->getResult()}");
19     else:
20         WSError($cadastra->getError()[0], $cadastra->getError()[1]);
21     endif;
22 <?
23
24 <form action="" method="post" name="UserCreateForm">
25 <label class="label">
26 <span class="field">Nome:</span>
27 <input
28     type="text"
29     name="user_name"
30     value="<?php if (!empty($ClienteData['user_name'])) echo $ClienteData['user_name']; ??"
31     title="Informe seu primeiro nome"
32     required
33 </input>
34 </label>
35
36 <label class="label">
37 <span class="field">Sobrenome:</span>
38 <input
39     type="text"
40     name="user_lastname"
41     value="<?php if (!empty($ClienteData['user_lastname'])) echo $ClienteData['user_lastname']; ??"
42     title="Informe seu sobrenome"
43     required
44 </input>
45 </label>
46

```

Fonte: Próprio autor

Figura 35 – Parte do código onde atualiza Usuário



```

update.php
Código-Fonte Histórico
<div class="content form_create">
<article>
<h1>Atualizar Usuário!</h1>
<?php
$ClienteData = filter_input_array(INPUT_POST, FILTER_DEFAULT);
$userId = filter_input(INPUT_GET, 'userid', FILTER_VALIDATE_INT);

if ($ClienteData && $ClienteData['SendPostForm']):
    unset($ClienteData['SendPostForm']);
    require('._models/AdminUser.class.php');
    $cadastra = new AdminUser;
    $cadastra->update($userId, $ClienteData);

    WSError($cadastra->getError()[0], $cadastra->getError()[1]);
else:
    $ReadUser = new Read;
    $ReadUser->read("ws_users", "WHERE user_id = :userid", "userid={$userId}");
    if (!$ReadUser->getResult()):
        else:
            $ClienteData = $ReadUser->getResult()[0];
            unset($ClienteData['user_password']);
        endif;
    endif;

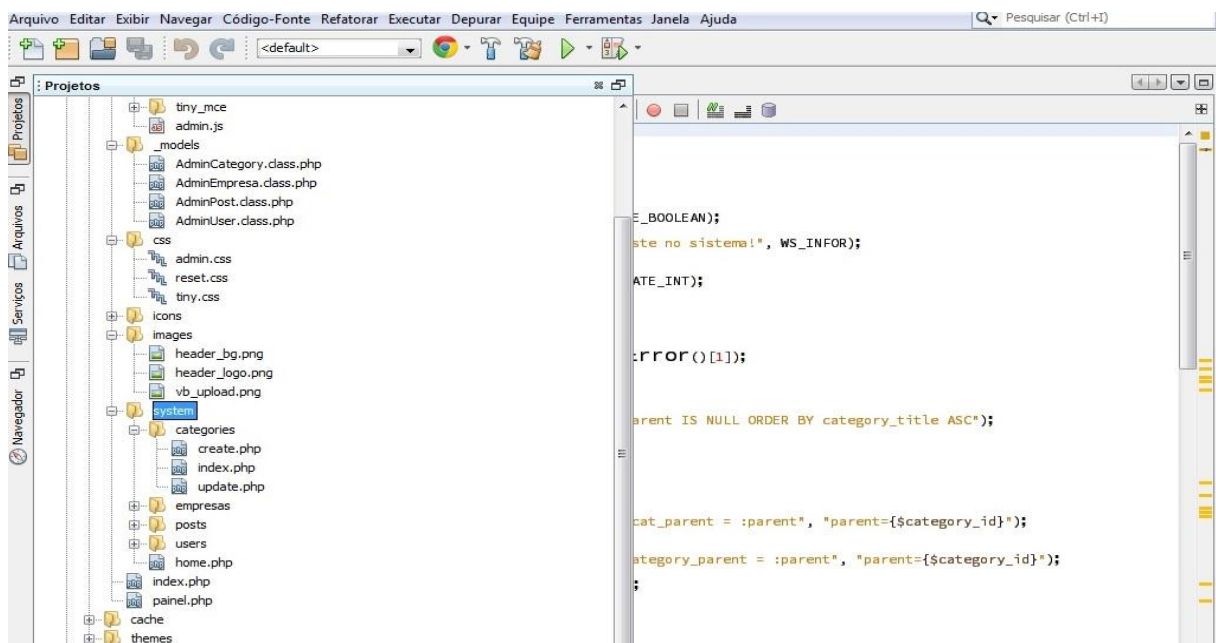
    $checkCreate = filter_input(INPUT_GET, 'create', FILTER_VALIDATE_BOOLEAN);
    if ($checkCreate && empty($cadastra)):
        WSError("0 usuário <b>{$ClienteData['user_name']}</b> foi cadastrado com sucesso no sistema!", WS_ACCEPT);
    endif;
    ?>

<form action="" method="post" name="UserCreateForm">
<label class="label">
<span class="field">Nome:</span>
<input
    type="text"
    name="user_name"
    value="<?php if (!empty($ClienteData['user_name'])) echo $ClienteData['user_name']; ?>"
    title="Informe seu primeiro nome"
    required
/>
</label>
<label class="label">

```

Fonte: Próprio autor

Figura 36 – Todas as classes



Fonte: Próprio autor

Figura 37 – Parte do código onde cria Categoria

```

Arquivo Editar Exibir Navegar Código-Fonte Refatorar Executar Depurar Equipe Ferramentas Janela Ajuda
create.php
Código-Fonte Histórico
35 <label class="label">
36 <span class="field">Título:</span>
37 <input type="text" name="category_title" value="<?php if (isset($data)) echo $data['category_title']; ?>" />
38 </label>
39
40 <label class="label">
41 <span class="field">Conteúdo:</span>
42 <textarea name="category_content" rows="5"><?php if (isset($data)) echo $data['category_content']; ?></textarea>
43 </label>
44
45 <div class="label_line">
46
47 <label class="label_small">
48 <span class="field">Data:</span>
49 <input type="text" class="formDate center" name="category_date" value="<?php = date('d/m/Y H:i:s'); ?>" />
50 </label>
51
52 <label class="label_small left">
53 <span class="field">Seção:</span>
54 <select name="category_parent">
55 <option value="null"> Selecione a Seção: </option>
56 <?php
57 $readSes = new Read;
58 $readSes->LExeRead("ws_categories", "WHERE category_parent IS NULL ORDER BY category_title ASC");
59 if (!$readSes->getResultSet());
60 echo '<option disabled="disabled" value="null"> Cadastre antes uma seção! </option>';
61 else:
62 foreach ($readSes->getResultSet() as $ses):
63 echo "<option value='{$ses['category_id']}'> " .
64     if ($ses['category_id'] == $data['category_parent']):
65 echo ' selected="selected" ' ;
66 endif;
67 echo "> {$ses['category_title']} </option>";
68 endforeach;
69 endif;
70 </select>
71 </div>
72
73
74
75
76

```

Fonte: Próprio autor

Figura 38 – Parte do código onde lista as categorias das notícias

```

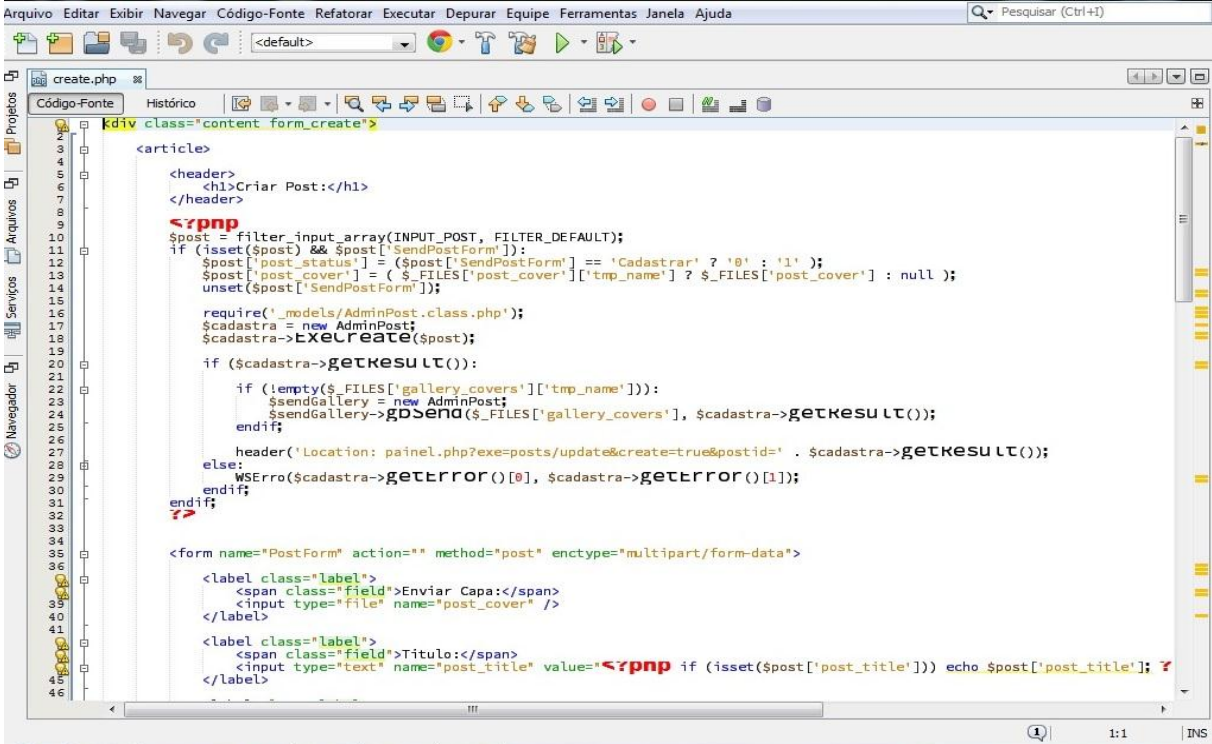
Arquivo Editar Exibir Navegar Código-Fonte Refatorar Executar Depurar Equipe Ferramentas Janela Ajuda
index.php
Código-Fonte Histórico
2 <div class="content_cat_list">
3
4 <section>
5
6 <h1>Categorias:</h1>
7
8 <?php
9 $empty = filter_input(INPUT_GET, 'empty', FILTER_VALIDATE_BOOLEAN);
10 if ($empty):
11 WSErro("Você tentou editar uma categoria que não existe no sistema!", WS_INFOR);
12 endif;
13
14 $delCat = filter_input(INPUT_GET, 'delete', FILTER_VALIDATE_INT);
15 if ($delCat):
16 require ('_models/AdminCategory.class.php');
17 $deletar = new AdminCategory;
18 $deletar->LExeDelete($delCat);
19 WSErro($deletar->getError()[0], $deletar->getError()[1]);
20 endif;
21
22
23 $readSes = new Read;
24 $readSes->LExeRead("ws_categories", "WHERE category_parent IS NULL ORDER BY category_title ASC");
25 if (!$readSes->getResultSet());
26 else:
27 foreach ($readSes->getResultSet() as $ses):
28 extract($ses);
29
30 $readPosts = new Read;
31 $readPosts->LExeRead("ws_posts", "WHERE post_cat_parent = :parent", "parent={$category_id}");
32
33 $readCats = new Read;
34 $readCats->LExeRead("ws_categories", "WHERE category_parent = :parent", "parent={$category_id}");
35
36 $countSesPosts = $readPosts->getRowCount();
37 $countSesCats = $readCats->getRowCount();
38
39 <?php
40 <section>
41
42 </headers>

```

Fonte: Próprio autor



Figura 39 – Parte do código onde cria Postagens



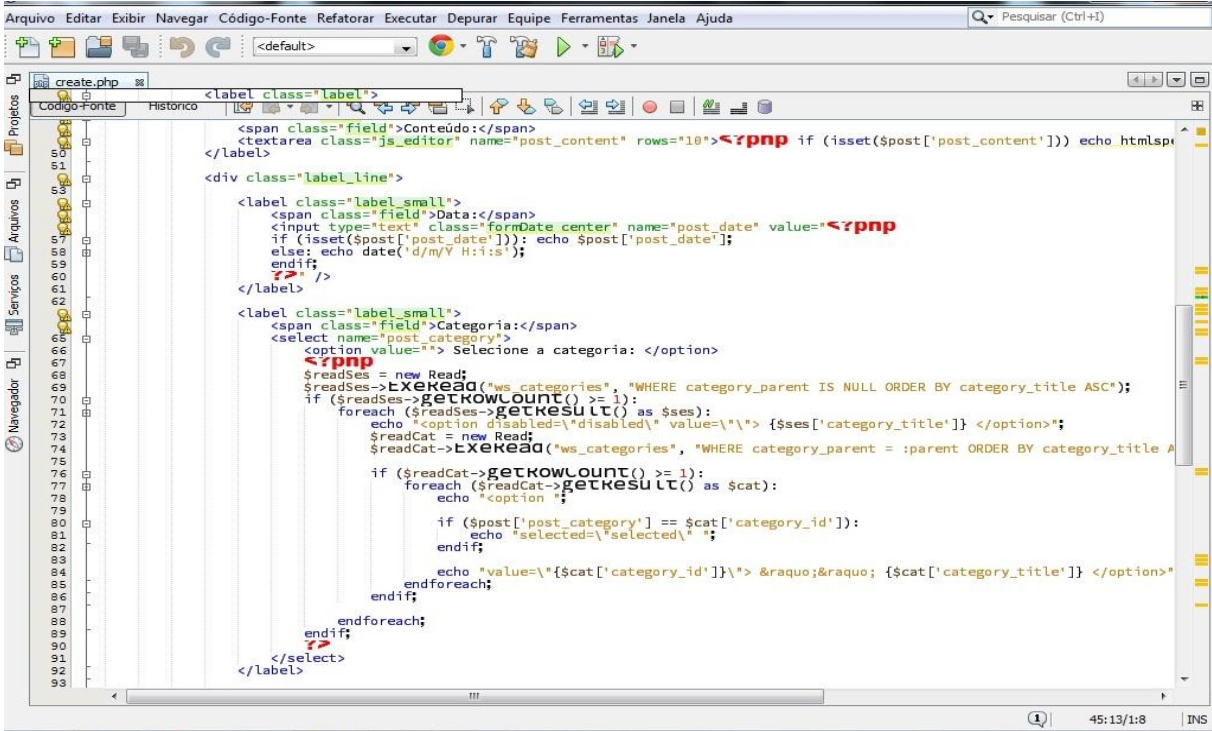
```

Arquivo Editar Exibir Navegar Código-Fonte Reforçar Executar Depurar Equipe Ferramentas Janela Ajuda
create.php
Código-Fonte Histórico
1 kdiv class="content form_create">
2 <article>
3
4 <header>
5 <h1>Criar Post:</h1>
6 </header>
7
8
9 <?php
10 $post = filter_input_array(INPUT_POST, FILTER_DEFAULT);
11 if (isset($post) && $post['SendPostForm']):
12 $post['post_status'] = ($post['SendPostForm'] == 'Cadastrar' ? '0' : '1' );
13 $post['post_cover'] = ( $_FILES['post_cover']['tmp_name'] ? $_FILES['post_cover'] : null );
14 unset($post['SendPostForm']);
15
16 require('_models/AdminPost.class.php');
17 $cadastra = new AdminPost;
18 $cadastra->EXECCreate($post);
19
20 if ($cadastra->GETRESULT()):
21
22     if (!empty($_FILES['gallery_covers']['tmp_name'])):
23         $sendGallery = new AdminPost;
24         $sendGallery->GDSEND($_FILES['gallery_covers'], $cadastra->GETRESULT());
25     endif;
26
27     header('Location: painel.php?exe=posts/update&create=true&postid=' . $cadastra->GETRESULT());
28 else:
29     WSError($cadastra->GETERROR()[0], $cadastra->GETERROR()[1]);
30 endif;
31 endif;
32
33
34 <form name="PostForm" action="" method="post" enctype="multipart/form-data">
35
36 <label class="label">
37 <span class="field">Enviar Capa:</span>
38 <input type="file" name="post_cover" />
39 </label>
40
41 <label class="label">
42 <span class="field">Titulo:</span>
43 <input type="text" name="post_title" value="<?php if (isset($post['post_title'])) echo $post['post_title']; ?>
44 </label>
45
46

```

Fonte: Próprio autor

Figura 40 – Parte do código onde cria Postagens II



```

Arquivo Editar Exibir Navegar Código-Fonte Reforçar Executar Depurar Equipe Ferramentas Janela Ajuda
create.php
Código-Fonte Histórico
47 <label class="label">
48 <span class="field">Conteúdo:</span>
49 <textarea class="js_editor" name="post_content" rows="10"><?php if (isset($post['post_content'])) echo htmlspecialchars($post['post_content']); </textarea>
50 </label>
51
52 <div class="label_line">
53
54 <label class="label_small">
55 <span class="field">Data:</span>
56 <input type="text" class="formDate center" name="post_date" value="<?php if (isset($post['post_date'])) echo $post['post_date']; else: echo date('d/m/Y H:i:s'); </input>
57 </label>
58
59 <label class="label_small">
60 <span class="field">Categoria:</span>
61 <select name="post_category">
62 <option value=""> Selecione a categoria:</option>
63 <?php
64 $readSes = new Read;
65 $readSes->EXECREAD("ws_categories", "WHERE category_parent IS NULL ORDER BY category_title ASC");
66 if ($readSes->GETKOWLOUNT() >= 1):
67     foreach ($readSes->GETRESULT() as $ses):
68         echo <option disabled="disabled" value=""> {$ses['category_title']} </option>;
69         $readCat = new Read;
70         $readCat->EXECREAD("ws_categories", "WHERE category_parent = :parent ORDER BY category_title ASC");
71         if ($readCat->GETKOWLOUNT() >= 1):
72             foreach ($readCat->GETRESULT() as $cat):
73                 echo <option >
74
75                 if ($post['post_category'] == $cat['category_id']):
76                     echo "selected="selected" ";
77                 endif;
78                 echo "value="{$cat['category_id']} "&#34;&#34; {$cat['category_title']} </option>;
79             endforeach;
80         endif;
81     endforeach;
82 endif;
83 </select>
84 </label>
85
86
87
88
89
90
91
92
93

```

Fonte: Próprio autor

Figura 41 – Parte do código onde edita as postagens

```

Arquivo Editar Exibir Navegar Código-Fonte Refatorar Executar Depurar Equipe Ferramentas Janela Ajuda
index.php
Código-Fonte Histórico
<div class="content_list_content">
  <section>
    <h1>Posts:</h1>
    <?php
    $empty = filter_input(INPUT_GET, 'empty', FILTER_VALIDATE_BOOLEAN);
    if ($empty):
      WSErrro("Óppsss: Você tentou editar um post que não existe no sistema!", WS_INFOR);
    endif;

    $action = filter_input(INPUT_GET, 'action', FILTER_DEFAULT);
    if ($action):
      require ('_models/AdminPost.class.php');

      $postAction = filter_input(INPUT_GET, 'post', FILTER_VALIDATE_INT);
      $postUpdate = new AdminPost;

      switch ($action):
        case 'active':
          $postUpdate->ExeStatus($postAction, '1');
          WSErrro("O status do post foi atualizado para <b>ativo</b>. Post publicado!", WS_ACCEPT);
          break;

        case 'inactive':
          $postUpdate->ExeStatus($postAction, '0');
          WSErrro("O status do post foi atualizado para <b>inativo</b>. Post agora é um rascunho!", WS_ALERT);
          break;

        case 'delete':
          $postUpdate->ExeDelete($postAction);
          WSErrro($postUpdate->getError()[0], $postUpdate->getError()[1]);
          break;

        default :
          WSErrro("Ação não foi identifica pelo sistema, favor utilize os botões!", WS_ALERT);
      endswitch;
    endif;

    $posti = 0;
    $getPage = filter_input(INPUT_GET, 'page', FILTER_VALIDATE_INT);
    $Pager = new Pager('painel.php?exe=posts/index&page=');
    $Pager->ExePager($getPage, 10);
  </div>
  
```

Fonte: Próprio autor

Figura 42 – Parte do código onde lista as empresas cadastradas

```

Arquivo Editar Exibir Navegar Código-Fonte Refatorar Executar Depurar Equipe Ferramentas Janela Ajuda
create.php
Código-Fonte Histórico
else:
  header("Location: painel.php?exe=empresas/update&create=true&emp={$_cadastra->getResult()}");
endif;
if;

<?php
<form name="PostForm" action="" method="post" enctype="multipart/form-data">
  <label class="label">
    <span class="field">Logo da empresa: <sup>Exatamente 578x288px (JPG ou PNG)</sup></span>
    <input type="file" name="empresa_capa" />
  </label>

  <label class="label">
    <span class="field">Nome da Empresa:</span>
    <input type="text" name="empresa_title" value="<?php if (isset($data['empresa_title'])) echo $data['empresa_title']; ?>" />
  </label>

  <label class="label">
    <span class="field">Ramo de atividade:</span>
    <input type="text" name="empresa_amo" value="<?php if (isset($data['empresa_amo'])) echo $data['empresa_amo']; ?>" />
  </label>

  <label class="label">
    <span class="field">Sobre a empresa:</span>
    <textarea name="empresa_sobre" rows="3"><?php if (isset($data['empresa_sobre'])) echo $data['empresa_sobre']; ?></textarea>
  </label>

  <div class="label_line">
    <label class="label_medium">
      <span class="field">Site da Empresa:</span>
      <input type="url" placeholder="http://www.mucurinanet.com.br" name="empresa_site" value="<?php if (isset($data['empresa_site'])) echo $data['empresa_site']; ?>" />
    </label>

    <label class="label_medium">
      <span class="field">Facebook Page:</span>
      <input type="text" placeholder="mucurinanet" name="empresa_facebook" value="<?php if (isset($data['empresa_facebook'])) echo $data['empresa_facebook']; ?>" />
    </label>
  </div>

  <label class="label">
    <span class="field">Nome da rua / Número:</span>
  </label>
  
```

Fonte: Próprio autor

Figura 43 – Parte do código onde faz a edição das empresas

```

4
5
6 <header>
7 <h1>Atualizar Empresa:</h1>
8 </header>
9
10 <?php
11 $empresa = filter_input(INPUT_GET, 'emp', FILTER_VALIDATE_INT);
12 $data = filter_input_array(INPUT_POST, FILTER_DEFAULT);
13
14 if ($data && $data['SendPostForm']):
15     $data['empresa_status'] = ($data['SendPostForm'] == 'Atualizar' ? '0' : '1');
16     $data['empresa_capa'] = ($FILES['empresa_capa']['tmp_name'] ? $FILES['empresa_capa'] : 'null');
17
18     unset($data['SendPostForm']);
19     require('_models/AdminEmpresa.class.php');
20     $cadastra = new AdminEmpresa;
21     $cadastra->exeupdate($empresa, $data);
22
23     WSError($cadastra->getError()[0], $cadastra->getError()[1]);
24 else:
25     $readEmp = new Read;
26     $readEmp->exehead("app_empresas", "WHERE empresa_id = :emp", "emp={$empresa}");
27     if (!$readEmp->getResult()):
28         header("Location: painel.php?exe=empresas/index&empty=true");
29     else:
30         $data = $readEmp->getResult()[0];
31     endif;
32 endif;
33
34 $checkCreate = filter_input(INPUT_GET, 'create', FILTER_VALIDATE_BOOLEAN);
35 if ($checkCreate && empty($cadastra)):
36     WSError("A empresa <b>{$data['empresa_titulo']}</b> foi cadastrada com sucesso no sistema!", WS_ACCEPT);
37 endif;
38
39 <form name="PostForm" action="" method="post" enctype="multipart/form-data">
40
41 <label class="label">
42 <span class="field">Logo da empresa: <sup>Exatamente 578x288px (JPG ou PNG)</sup></span>
43 <input type="file" name="empresa_capa" />
44 </label>
45

```

Fonte: Próprio autor

Figura 44 – Index do site

```

1 <!DOCTYPE html>
2 <?php
3 require './_app/Config.inc.php';
4
5 <html lang="pt-br">
6 <head>
7 <meta charset="UTF-8">
8 <meta name="title" content="8832298">
9
10 <!--[if lt IE 9]>
11 <script src="../../_cdn/html5.js"></script>
12 <![endif]-->
13
14 <?php
15 $link = new Link;
16 $link->getIags();
17
18 <link rel="stylesheet" href="{INCLUDE_PATH} /css/reset.css">
19 <link rel="stylesheet" href="{INCLUDE_PATH} /css/style.css">
20 <link rel="stylesheet" href="{HOME} /_cdn/shadowbox/shadowbox.css">
21 <link href="http://fonts.googleapis.com/css?family=Baumans" rel="stylesheet" type="text/css">
22
23 </head>
24 <body>
25
26 <?php
27 require(REQUIRE_PATH . '/inc/header.inc.php');
28
29 if(!require($link->getPactn())):
30     WSError("Erro ao incluir arquivo de navegação", WS_ERROR, true);
31 endif;
32
33 require(REQUIRE_PATH . '/inc/footer.inc.php');
34
35 </body>
36
37 <script src="{HOME} /_cdn/jquery.js"></script>
38 <script src="{HOME} /_cdn/jcycle.js"></script>
39 <script src="{HOME} /_cdn/jmask.js"></script>
40 <script src="{HOME} /_cdn/shadowbox/shadowbox.js"></script>
41 <script src="{HOME} /_cdn/_plugins.conf.js"></script>
42 <script src="{HOME} /_cdn/_scripts.conf.js"></script>
43
44 </html>
45

```



Fonte: Próprio autor

Figura 45 – Index do painel

```

16 </head>
17 <body class="login">
18
19
20 <div id="login">
21 <div class="boxin">
22 <h1>Administrar Site</h1>
23
24
25 <?php
26 $login = new Login(3);
27 if ($login->checklogin()):
28     header('Location: painel.php');
29 endif;
30
31 $dataLogin = filter_input_array(INPUT_POST, FILTER_DEFAULT);
32 if (!empty($dataLogin['AdminLogin'])):
33
34     $login->execlogin($dataLogin);
35     if (!$login->getResult()):
36         WSError($login->getError()[0], $login->getError()[1]);
37     else:
38         header('Location: painel.php');
39     endif;
40 endif;
41
42
43 $get = filter_input(INPUT_GET, 'exe', FILTER_DEFAULT);
44 if (!empty($get)):
45     if ($get == 'restrito'):
46         WSError('<b>Oppsss:</b> Acesso negado. Favor efetue login para acessar o painel!', WS_ALERT);
47     elseif ($get == 'logoff'):
48         WSError('<b>Sucesso ao deslogar:</b> Sua sessão foi finalizada. Volte sempre!', WS_ACCEPT);
49     endif;
50 endif;
51
52
53 <form name="AdminLoginForm" action="" method="post">
54 <label>
55 <span>E-mail:</span>
56 <input type="email" name="user" />
57 </label>
58 <label>
59 <span>Senha:</span>
60 <input type="password" name="pass" />
61 </label>
62

```

Fonte: Próprio autor

Figura 46 – Parte do código onde faz o controle do nível de usuário

```

1 <?php
2 ob_start();
3 session_start();
4 require('../_app/Config.inc.php');
5
6 $login = new Login(3);
7 $logoff = filter_input(INPUT_GET, 'logoff', FILTER_VALIDATE_BOOLEAN);
8 $getexe = filter_input(INPUT_GET, 'exe', FILTER_DEFAULT);
9
10 if (!$login->checklogin()):
11     unset($_SESSION['userlogin']);
12     header('Location: index.php?exe=restrito');
13 else:
14     $userlogin = $_SESSION['userlogin'];
15 endif;
16
17 if ($logoff):
18     unset($_SESSION['userlogin']);
19     header('Location: index.php?exe=logoff');
20 endif;
21
22 <!DOCTYPE html>
23 <html lang="pt-br">
24
25 <head>
26 <meta charset="UTF-8">
27 <title>Site Admin</title>
28 <!--[if lt IE 9]-->
29 <script src="../cdn/html5.js"></script>
30 <![endif]-->
31
32 <link href="http://fonts.googleapis.com/css?family=Open+Sans:400,600,800" rel="stylesheet" type="text/css">
33 <link rel="stylesheet" href="css/reset.css" />
34 <link rel="stylesheet" href="css/admin.css" />
35 </head>
36
37 <body class="painel">
38
39 <header id="navadmin">
40 <div class="content">
41
42 <h1 class="logomarca">Pro Admin</h1>
43
44 <ul class="system_nav radius">
45 <li class="username">Olá <?=$userlogin['user_name']; ?> <?=$userlogin['user_lastname']; ?></li>
46 <li><a class="icon profile radius" href="painel.php?exe=users/profile">Perfil</a></li>
47 <li><a class="icon users radius" href="painel.php?exe=users">Usuários</a></li>
48 <li><a class="icon logout radius" href="painel.php?exe=logoff">Sair</a></li>

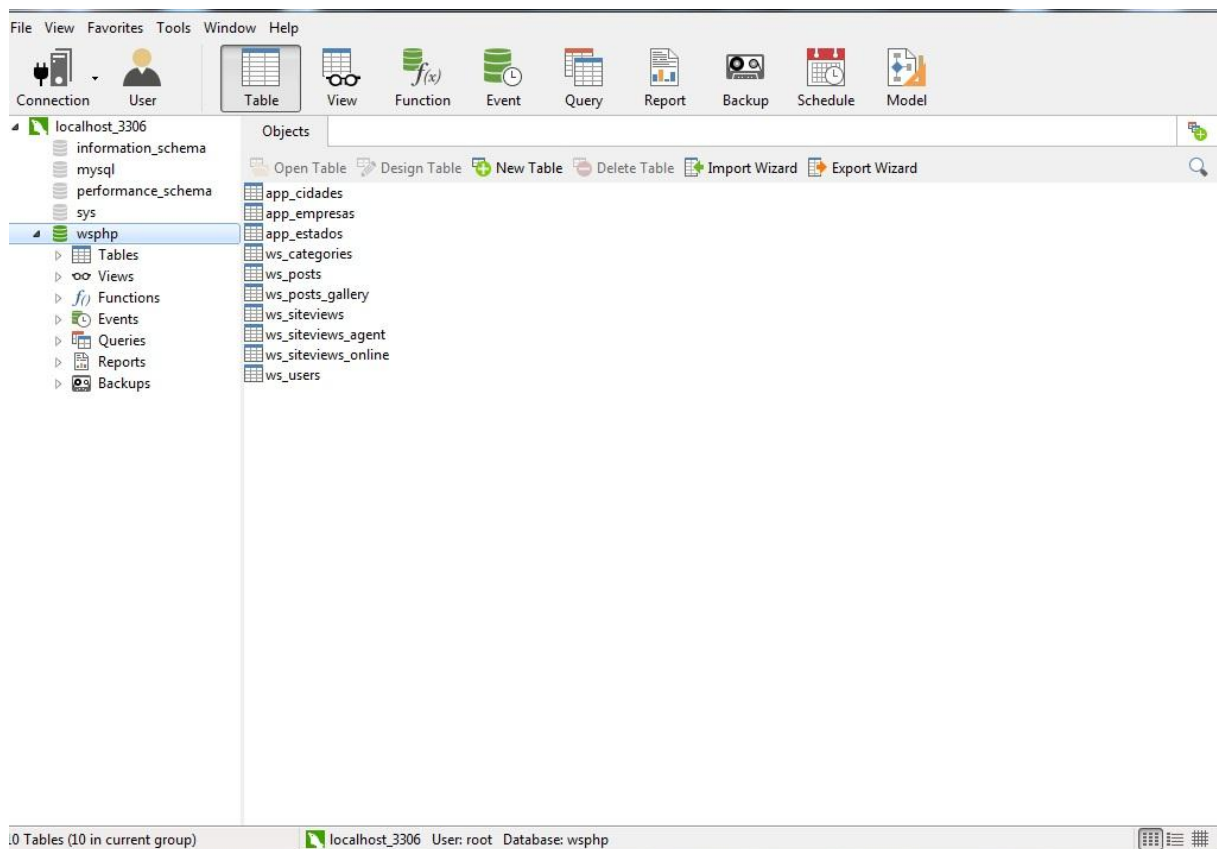
```

Fonte: Próprio autor

## APÊNDICE II

### BANCO DE DADOS

Figura 47 – Banco de Dados



Fonte: Próprio autor

### APÊNDICE III

#### MODELO FISICO DO BANCO DE DADOS

```
-----  
-- Table structure for `app_cidades`  
-----  
DROP TABLE IF EXISTS `app_cidades`;  
  
CREATE TABLE `app_cidades` (  
  `cidade_id` int(11) NOT NULL AUTO_INCREMENT,  
  `estado_id` int(11) DEFAULT NULL,  
  `cidade_nome` varchar(75) CHARACTER SET latin1 DEFAULT NULL,  
  `cidade_uf` varchar(5) DEFAULT NULL,  
  PRIMARY KEY (`cidade_id`)  
)  
  
ENGINE=InnoDB AUTO_INCREMENT=5566 DEFAULT CHARSET=utf8;  
-----  
-- Table structure for `app_empresas`  
-----  
DROP TABLE IF EXISTS `app_empresas`;
```

```

CREATE TABLE `app_empresas` (
  `empresa_id` int(11) NOT NULL AUTO_INCREMENT,
  `empresa_title` varchar(255) DEFAULT NULL,
  `empresa_name` varchar(255) DEFAULT NULL,
  `empresa_capa` varchar(255) DEFAULT NULL,
  `empresa_amo` varchar(255) DEFAULT NULL,
  `empresa_sobre` text,
  `empresa_site` varchar(255) DEFAULT NULL,
  `empresa_facebook` varchar(255) DEFAULT NULL,
  `empresa_endereco` varchar(255) DEFAULT NULL,
  `empresa_uf` int(11) DEFAULT NULL,
  `empresa_cidade` int(11) DEFAULT NULL,
  `empresa_categoria` varchar(255) DEFAULT NULL,
  `empresa_status` int(11) DEFAULT '0',
  `empresa_views` decimal(10,0) DEFAULT NULL,
  `empresa_date` timestamp NULL DEFAULT NULL,
  `empresa_last_view` timestamp NULL DEFAULT NULL ON UPDATE
CURRENT_TIMESTAMP,
  PRIMARY KEY (`empresa_id`)
)

```

```
ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

```

-- -----
-- Table structure for `app_estados`
-- -----

```

```
DROP TABLE IF EXISTS `app_estados`;
```

```

CREATE TABLE `app_estados` (
  `estado_id` int(11) NOT NULL,
  `estado_nome` varchar(75) CHARACTER SET latin1 DEFAULT NULL,
  `estado_uf` varchar(5) CHARACTER SET latin1 DEFAULT NULL,
  `estado_regiao` varchar(75) DEFAULT NULL,

```

```
PRIMARY KEY (`estado_id`)
)

ENGINE=InnoDB DEFAULT CHARSET=utf8;

-- -----
-- Table structure for `ws_categories`
-- -----

DROP TABLE IF EXISTS `ws_categories`;

CREATE TABLE `ws_categories` (
  `category_id` int(11) NOT NULL AUTO_INCREMENT,
  `category_parent` int(11) DEFAULT NULL,
  `category_name` varchar(255) CHARACTER SET latin1 DEFAULT NULL,
  `category_title` varchar(255) CHARACTER SET latin1 DEFAULT NULL,
  `category_content` text CHARACTER SET latin1,
  `category_date` timestamp NULL DEFAULT NULL,
  PRIMARY KEY (`category_id`)
)

ENGINE=InnoDB DEFAULT CHARSET=utf8;

-- -----
-- Table structure for `ws_posts`
-- -----

DROP TABLE IF EXISTS `ws_posts`;

CREATE TABLE `ws_posts` (
  `post_id` int(11) NOT NULL AUTO_INCREMENT,
  `post_name` varchar(255) CHARACTER SET latin1 DEFAULT NULL,
```



```

`post_title` varchar(255) CHARACTER SET latin1 DEFAULT NULL,
`post_content` longtext CHARACTER SET latin1,
`post_cover` varchar(255) CHARACTER SET latin1 DEFAULT NULL,
`post_date` timestamp NULL DEFAULT NULL,
`post_author` int(11) DEFAULT NULL,
`post_category` int(11) DEFAULT NULL,
`post_cat_parent` int(11) DEFAULT NULL,
`post_views` decimal(10,0) DEFAULT '0',
`post_last_views` timestamp NULL DEFAULT NULL ON UPDATE
CURRENT_TIMESTAMP,
`post_status` int(11) DEFAULT NULL,
`post_type` varchar(255) CHARACTER SET latin1 DEFAULT NULL,
PRIMARY KEY (`post_id`)
)

```

```
ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

```
-- -----
```

```
-- Table structure for `ws_posts_gallery`
```

```
-- -----
```

```
DROP TABLE IF EXISTS `ws_posts_gallery`;
```

```

CREATE TABLE `ws_posts_gallery` (
  `post_id` int(11) DEFAULT NULL,
  `gallery_id` int(11) NOT NULL AUTO_INCREMENT,
  `gallery_image` varchar(255) CHARACTER SET latin1 DEFAULT NULL,
  `gallery_date` timestamp NULL DEFAULT NULL ON UPDATE
CURRENT_TIMESTAMP,
  PRIMARY KEY (`gallery_id`)
)

```

```
ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

```
-----
```

```
-- Table structure for `ws_siteviews`
```

```
-----  
  
DROP TABLE IF EXISTS `ws_siteviews`;
```

```
CREATE TABLE `ws_siteviews` (  
  `siteviews_id` int(11) NOT NULL AUTO_INCREMENT,  
  `siteviews_date` date NOT NULL,  
  `siteviews_users` decimal(10,0) NOT NULL,  
  `siteviews_views` decimal(10,0) NOT NULL,  
  `siteviews_pages` decimal(10,0) NOT NULL,  
  PRIMARY KEY (`siteviews_id`),  
  KEY `idx_1` (`siteviews_date`)  
)
```

```
ENGINE=InnoDB AUTO_INCREMENT=4 DEFAULT CHARSET=utf8;
```

```
-----  
-- Table structure for `ws_siteviews_agent`  
-----
```

```
DROP TABLE IF EXISTS `ws_siteviews_agent`;
```

```
CREATE TABLE `ws_siteviews_agent` (  
  `agent_id` int(11) NOT NULL AUTO_INCREMENT,  
  `agent_name` varchar(255) CHARACTER SET latin1 NOT NULL,  
  `agent_views` decimal(10,0) NOT NULL,  
  PRIMARY KEY (`agent_id`),  
  KEY `idx_1` (`agent_name`)  
)
```

```
ENGINE=InnoDB AUTO_INCREMENT=2 DEFAULT CHARSET=utf8;
```

```
-----  
-- Table structure for `ws_siteviews_online`  
-----
```

```
DROP TABLE IF EXISTS `ws_siteviews_online`;
```

```
CREATE TABLE `ws_siteviews_online` (
  `online_id` int(11) NOT NULL AUTO_INCREMENT,
  `online_session` varchar(255) CHARACTER SET latin1 NOT NULL,
  `online_startview` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
  `online_endview` timestamp NOT NULL DEFAULT '0000-00-00 00:00:00' ON
UPDATE CURRENT_TIMESTAMP,
  `online_ip` varchar(255) CHARACTER SET latin1 NOT NULL,
  `online_url` varchar(255) CHARACTER SET latin1 NOT NULL,
  `online_agent` varchar(255) CHARACTER SET latin1 NOT NULL,
  `agent_name` varchar(255) DEFAULT NULL,
  PRIMARY KEY (`online_id`)
)
```

```
ENGINE=InnoDB AUTO_INCREMENT=4 DEFAULT CHARSET=utf8;
```

```
-- -----
-- Table structure for `ws_users`
-- -----
```

```
DROP TABLE IF EXISTS `ws_users`;
CREATE TABLE `ws_users` (
  `user_id` int(11) NOT NULL AUTO_INCREMENT,
  `user_name` varchar(255) CHARACTER SET latin1 NOT NULL,
  `user_lastname` varchar(255) CHARACTER SET latin1 NOT NULL,
  `user_email` varchar(255) CHARACTER SET latin1 NOT NULL,
  `user_password` varchar(255) CHARACTER SET latin1 NOT NULL,
  `user_registration` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
  `user_lastupdate` timestamp NULL DEFAULT NULL ON UPDATE
CURRENT_TIMESTAMP,
  `user_level` int(11) NOT NULL DEFAULT '1',
  PRIMARY KEY (`user_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```