

# UNIVERSIDADE DE SÃO PAULO

## Escola de Engenharia de São Carlos

---

Desenvolvimento de uma plataforma  
*online* para o Laboratório de  
Processamento de Sinais.

---

***Bruno Abreu Kemmer***



São Carlos - SP

# Desenvolvimento de uma plataforma *online* para o Laboratório de Processamento de Sinais.

***Bruno Abreu Kemmer***

***Orientador: Carlos Dias Maciel***

Monografia final de conclusão de curso apresentada a  
Escola de Engenharia de São Carlos – EESC-USP - para  
obtenção do título de Engenheiro de Computação.

Área de Concentração: Desenvolvimento de Aplicações  
Web.

**USP – São Carlos**  
**Junho de 2013**

AUTORIZO A REPRODUÇÃO TOTAL OU PARCIAL DESTE TRABALHO,  
POR QUALQUER MEIO CONVENCIONAL OU ELETRÔNICO, PARA FINS  
DE ESTUDO E PESQUISA, DESDE QUE CITADA A FONTE.

K31d      Kemmer, Bruno Abreu  
Desenvolvimento de uma plataforma online para o  
laboratório de processamento de sinais. / Bruno Abreu  
Kemmer; orientador Carlos Dias Maciel; coorientador  
Evandro Luis Linhari Rodrigues. São Carlos, 2013.

Monografia (Graduação em Engenharia de Computação)  
-- Escola de Engenharia de São Carlos da Universidade  
de São Paulo, 2013.

1. Desenvolvimento. 2. Web. 3. Online. I. Título.

## FOLHA DE APROVAÇÃO

**Nome:** Bruno Abreu Kemmer

**Título:** “Desenvolvimento de uma plataforma online para o laboratório de processamento de sinais”

**Trabalho de Conclusão de Curso defendido em** 21/06/2013

**Comissão Julgadora:**

**Resultado:**

Prof. Dr. José Roberto Boffino de Almeida  
Monteiro  
SEL/EESC/USP

APROVADO

M.Sc. Wagner Endo  
(Doutorando - SEL/EESC/USP)

APROVADO

M.Sc. Giovana Yuko Nakashima  
(Doutoranda - SEL/EESC/USP)

Aprovado

**Coordenador pela EESC/USP do Curso de Engenharia de Computação:**

Prof. Associado Evandro Luís Linhari Rodrigues

“Nós somos aquilo que fazemos repetidamente.  
Excelência, então, não é um modo de agir, mas um hábito.”

*Aristóteles*

# **Dedicatória**

Dedico este trabalho a minha mãe Rosa Walda Abreu Marquart e meu pai Ezequiel Cleto Kemmer, que me apoiaram durante toda a minha vida.

# **Agradecimentos**

Agradeço a todos os grandes amigos que fiz durante a faculdade, à minha família.

Ao meu orientador que me ajudou durante grande parte da faculdade com conselhos e diretrizes.

Aos colegas de trabalhos que proporcionaram tempo para a execução desse projeto.

# Resumo

O objetivo desse trabalho é construir um módulo de uma plataforma que possibilite o acesso às informações desejáveis dos usuários do laboratório de processamento de sinais da Escola de Engenharia de São Carlos, facilitando o contato e a interface com tanto o ambiente interno referente aos integrantes da Universidade de São Paulo quanto as pessoas e instituições externas. O módulo também deve ter uma manutenção prática e funcional, que possibilite uma integração com outras plataformas de base de dados de curriculums como a Plataforma Lattes<sup>1</sup> e ResearcherID<sup>2</sup>.

---

<sup>1</sup> Plataforma Lattes. Disponível em: <http://lattes.cnpq.br>. Acesso em 05.05.2013.

<sup>2</sup> ResearchID. Disponível em: <http://www.researchid.com>. Acesso em 05.05.2013.



# Sumário

<b>LISTA DE GRÁFICOS.....</b>	<b>IX</b>
<b>LISTA DE TABELAS .....</b>	<b>X</b>
<b>LISTA DE FIGURAS .....</b>	<b>XI</b>
<b>CAPÍTULO 1: INTRODUÇÃO .....</b>	<b>1</b>
1.1. CONTEXTUALIZAÇÃO E MOTIVAÇÃO .....	1
1.2. OBJETIVOS .....	1
1.3. ORGANIZAÇÃO DO TRABALHO .....	1
<b>CAPÍTULO 2: REVISÃO BIBLIOGRÁFICA.....</b>	<b>3</b>
2.1. CONSIDERAÇÕES INICIAIS .....	3
2.2. TECNOLOGIAS.....	3
2.4. CONSIDERAÇÕES FINAIS .....	18
<b>CAPÍTULO 3: MATERIAIS E MÉTODOS.....</b>	<b>19</b>
3.1. ESPECIFICAÇÃO .....	19
3.2 ARQUITETURA CONCEITO .....	19
3.3. ESTRUTURA DE DADOS .....	20
3.4. DESCRIÇÃO DAS PÁGINAS PHP .....	20
3.5. FUNÇÕES PHP .....	21
<b>CAPÍTULO 4: DESENVOLVIMENTO DO TRABALHO .....</b>	<b>22</b>
4.1. PROJETO .....	22
4.2. DESCRIÇÃO DAS ATIVIDADES REALIZADAS .....	22
4.2.1. <i>Levantamento dos requisitos</i> .....	22
4.2.2. <i>Análise dos requisitos</i> .....	24
4.2.3. <i>Implementação</i> .....	27

4.2.4. <i>Teste</i> .....	27
4.3. RESULTADOS OBTIDOS .....	27
4.4. DIFICULDADES E LIMITAÇÕES .....	34
4.5. CONSIDERAÇÕES FINAIS .....	34
<b>CAPÍTULO 4: CONCLUSÃO</b> .....	<b>35</b>
4.1. CONTRIBUIÇÕES .....	35
4.2. TRABALHOS FUTUROS .....	35
<b>REFERÊNCIAS</b> .....	<b>36</b>

# Lista de Gráficos

Gráfico 1 - Domínios reportando uso da linguagem PHP .....	5
Gráfico 2 - Market Share dos SGBDs .....	7
Gráfico 3 - Custos totais aos longo de 3 anos dos SGBDs .....	8
Gráfico 4 - <i>Market Share</i> dos servidores web em todos os sites .....	10
Gráfico 5 - Número de usuários da internet.....	15
Gráfico 6 – Percentual de sites que utilizam bibliotecas Javascript .....	17

# Lista de Tabelas

Tabela 1 - Vantagens de cada SGDB .....	5
Tabela 2 - Market share dos servidores web por desenvolvedor.....	11

# Lista de Figuras

Figura 1 - Tela inicial da aplicação phpMyAdmin .....	12
Figura 2 - Tela com um banco de dados selecionado .....	13
Figura 3 - Arquitetura conceito da plataforma .....	19
Figura 4 - Mapeamento do banco de dados .....	20
Figura 5 - Protótipo da visão do perfil de um aluno .....	23
Figura 6 - Protótipo da visão do perfil de um docente .....	24
Figura 7 - Arquitetura da plataforma no sistema Windows.....	26
Figura 8 - Diagrama de Entidade-Relacionamento da plataforma .....	26
Figura 9 – Resultado do processamento do arquivo index.php no servidor.....	28
Figura 10 - Resultado do processamento do arquivo professores.php no servidor .....	28
Figura 11 – Resultado do processamento do arquivo profile.php para o usuário selecionado .....	29
Figura 12 - Resultado do arquivo login.php .....	30
Figura 13 – Visão administrativa da plataforma – Seção “Meu Perfil” .....	30
Figura 14 – Seção de gerenciamento de foto.....	31
Figura 15 - Seção formação acadêmica .....	32
Figura 16 - Seção disciplinas .....	32
Figura 17 - Seção trabalhos .....	33
Figura 18 - Seção usuários.....	33

# **CAPÍTULO 1: INTRODUÇÃO**

## **1.1. Contextualização e Motivação**

Como o laboratório de processamento de sinais está inserido em uma universidade pública, a Universidade de São Paulo, em que se publicou 26.704 artigos no ano de 2012, 58.303 alunos matriculados na graduação, 28.498 na pós-graduação, 13.836 mestrando e 14.662 doutorandos, torna-se essencial facilitar o acesso, interno e externo, à informação gerada por seus grupos de pesquisa (USP, 2013).

É necessário facilitar o contato com seus integrantes, professores, mestrando e doutorandos, tanto localmente na universidade quanto com a comunidade externa, possibilitando parcerias nacionais e internacionais.

## **1.2. Objetivos**

A plataforma web para o laboratório de processamento de sinais tem por objetivo ser um portal para entrega de informações sobre seus integrantes, suas pesquisas em desenvolvimento e já publicadas, eventos que estejam sendo executados ou que estejam dentro da área de estudo, hyperlinks que possam auxiliar o estudo e uma área de contato com o grupo.

Implementar a área relacionada às informações dos integrantes, tanto pessoais, formação acadêmica, áreas de interesse e disciplinas que possam estar ministrando, com hyperlinks ao sistema jupyterweb (sistema de informação da Universidade de São Paulo) referente a cada disciplina ministrada. Além de possibilitar a integração com a plataforma Lattes e ResearchID.

Além disso, desenvolver uma visão administrativa, restringir seu acesso e possibilitar a manutenção da informação.

Como descrito, neste trabalho não se espera a implementação de uma plataforma completa, mas sim com o escopo para o desenvolvimento da área de perfil de seus integrantes e um módulo de administração para sua manutenção.

## **1.3. Organização do Trabalho**

Esta monografia está estruturada da seguinte forma, o capítulo 2 é dedicado aos métodos e tecnologias que envolvem o desenvolvimento da aplicação. Para cada tópico, é descrito seu estado da arte e sua evolução. Serão descritas as seguintes tecnologias: PHP, MySQL, Apache, phpMyAdmin, HTML, CSS e jQuery.

O capítulo 3 se dedica ao desenvolvimento do trabalho, sendo organizado em três seções. A seção 3.1 trata das considerações iniciais do projeto. A seção 3.2 refere-se ao escopo da plataforma, e as tarefas a serem realizadas. A seção 3.3 descreve as atividades realizadas. A seção 3.4 é dedicada aos resultados obtidos.

No capítulo 4 são feitas conclusões e projetos futuros. Além disso se discorre sobre as possibilidades de melhorias futuras para a continuação da plataforma.

# CAPÍTULO 2: REVISÃO BIBLIOGRÁFICA

## 2.1. Considerações Iniciais

Este capítulo apresenta os métodos e tecnologias que envolvem o desenvolvimento de uma aplicação web. Para cada tópico será descrito o estado da arte além de suas principais características.

## 2.2. Tecnologias

De acordo com Converse, Morgan, e Park (2004), PHP é uma linguagem *open-source* (tem seu código fonte disponibilizado), *server-side* (a compilação é executada no servidor, e disponibilizado apenas para o usuário o elemento final processado), sua codificação fica embutida em uma página HTML (*HyperText Markup Language* é uma linguagem de marcação de hipertexto interpretada por navegadores), podendo ser executada por servidores Web.

Segundo (Php.net, 2013), suas principais características são:

- Velocidade e robustez;
- Programação estruturada e orientada a objetos;
- Portabilidade – independência de plataforma;
- Tipagem dinâmica; (o tamanho das variáveis é definido na execução)
- Sintaxe similar a C/C++ e Perl;
- *Open Source*

Em sua última versão disponibilizada em julho de 2004, o PHP versão 5, introduz um novo modelo de orientação a objeto adicionando novos elementos contidos no paradigma da programação orientada a objetos, incluindo a reformulação dos construtores e adição de destrutores, visibilidade de acesso, abstração de objeto e interfaces de objetos. O tratamento de objetos foi completamente reescrito, permitindo um desempenho melhor além de outras vantagens. Enquanto na versão anterior era preciso muito esforço para atender à orientação a objetos e aos padrões de projeto, o PHP versão 5 veio para sanar essa deficiência, embora ainda sofra, nesse sentido, por problemas devido a ser uma

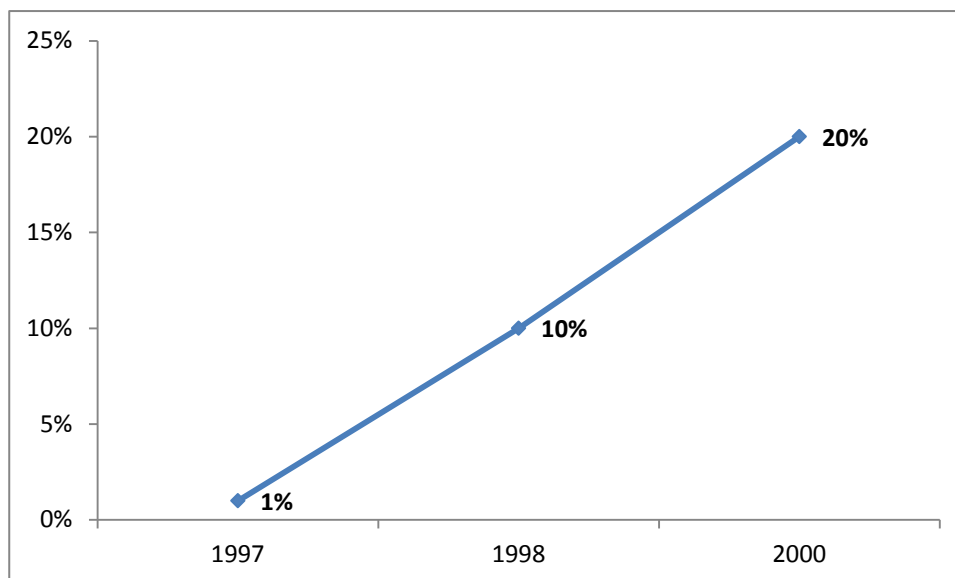


linguagem de tipagem fraca, ou seja, não sendo obrigatório a declaração do tipo das variáveis antes da execução das aplicações.

A linguagem PHP foi criada por Rasmus Lerforf em 1995, inicialmente como uma ferramenta para executar scripts Perl, para gerar estatísticas de acesso para seu curriculum online. Ele nomeou esse conjunto como '*Personal Home Page Tools*' que deu origem ao nome PHP (Personal Home Page). No momento em que foram requeridas mais funcionalidades, Rasmus escreveu uma implementação em C muito mais complexa e maior, e que poderia ter conexão com bancos de dados, possibilitando assim aos usuários o desenvolvimento de aplicativos dinâmicos para a Web.

Em 1997, houve a segunda versão da linguagem, chegando ao número de 50.000 domínios reportando que utilizavam a versão 2.0 (1% dos domínios da internet). Após essa versão o código foi reescrito, com a versão 3.0, que se parece com o PHP existente nos dias atuais. Esta versão foi criada por Andi Gutmans e Zeev Suraski ainda no ano de 1997. A motivação da dupla era desenvolver um *eCommerce* (comércio eletrônico) para um projeto em sua universidade. Suas principais características eram sua forte capacidade de extensibilidade (adendos ao código que possibilitam o reuso e economia de tempo de desenvolvimento), uma base sólida para conexões com diversos bancos de dados, protocolos (padrões utilizados para comunicação entre aplicações), e APIs (*Application Programming Interface*, interfaces de programação de aplicativos), o que atraiu milhares de desenvolvedores. Ao final de 1998, a linguagem estava com dezenas de milhares de usuários e centenas de milhares de Web sites relatando que o utilizavam, chegando a 10% dos servidores Web da Internet. O foco da versão seguinte 4.0, lançada oficialmente em maio de 2000 foi na performance, objetivo o qual foi alcançado, obtendo melhoras significativas. A expectativa era de que milhões de sites o utilizassem na época, totalizando 20% dos domínios da internet.

O gráfico 1 mostra a evolução da quantidade de domínios da internet reportados que utilizam a linguagem ao longo das 3 versões disponibilizadas.



**Gráfico 1 - Domínios reportando uso da linguagem PHP**

Fonte: (Php.net, 2013)

De acordo com (MySQL.com, 2013) o MySQL é um sistema de gerenciamento de bancos de dados (SGDB), que utiliza a linguagem SQL (*Structured Query Language*) como sua interface. Ele é rodado como um servidor provendo acessos para múltiplos usuários a diferentes bancos de dados. É um sistema *Open Source*, tendo seu código disponível no termos do *General Public Licence* (GNU).

**Tabela 1 - Vantagens de cada SGDB**

Fonte: (Suehring, 2002)

SGDB	Vantagens	Inconveniências
Oracle	Versatilidade, estabilidade e segurança	Chance de aumento de TCO ( <i>Total Cost Ownership</i> , custo total de propriedade, somatório de todos os custos relacionados a solução).
MS SQL Server	Estabilidade e segurança.	Relativamente uma solução de alto TCO e proprietária, como pode ser visto no gráfico 3.
PostgreSQL	Uma base de dados que tem seu uso crescente e	Precisa ser largamente implementada para ser utilizada em negócios de

	de baixo TCO.	grande porte.
Informix	Estável.	Geralmente tem um TCO mais alto.
MySQL	Oferece um melhor caso cenário em muitas formas, baixo TCO, alta estabilidade, alta segurança.	Nem todas as versões disponíveis podem oferecer todo o potencial de capacidades do MySQL.

Em sua última versão (5.5), há duas variantes, uma *open source*, e outra comercial (necessita licenciamento) *Enterprise Server*. As duas tem um código fonte em comum com principais características: portabilidade (suporta quase todas as plataformas disponíveis atualmente), logo obtém também um alto nível de compatibilidade, desempenho e estabilidade, baixo uso de recursos de hardware, facilidade de uso, suporte a um controle transacional (controle de cada operação efetuada), *trigger* (recurso de programação executado na ocorrência de um evento, exemplo: sempre que for executada uma operação de inserção execute outro comando previamente estruturado), *stored procedures* (programas armazenados no servidor, pré-compilados, para executar alguma lógica para a manipulação de dados, podendo retornar ou não algum valor), possibilidade do usuário criar funções, particionamento (possibilidade de distribuir partes de tabelas individuais ao longo do sistema de arquivos de acordo com regras previas), agendamento de eventos, melhor controle das tabelas de log, uma aplicação interna especial para o upgrade de versões (verificando incompatibilidades) e clusterização (possibilidade de utilização de múltiplos servidores sendo tratados como uma única máquina).

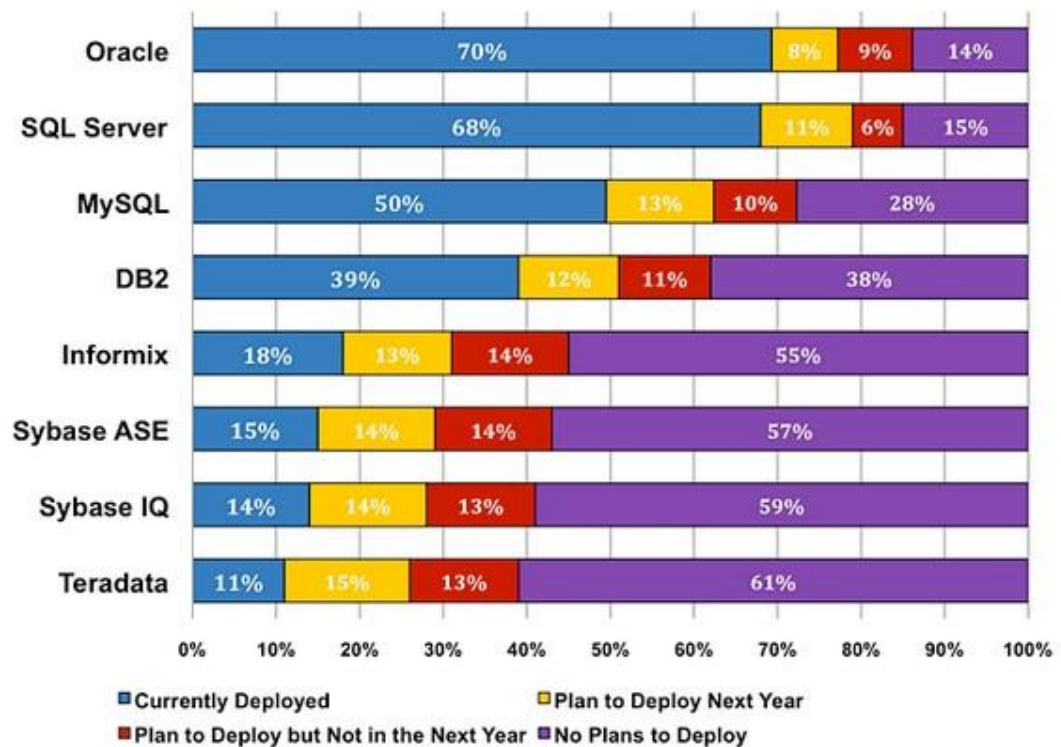
Segundo MySQL.com (2013), seu desenvolvimento começou no início de 1994, com Michael Widenius e David Axmark, e sua primeira versão foi lançada 23 de maio de 1995. Em 2008, a empresa MySQL foi adquirida pela Sun Microsystems, que por sua vez foi adquirida pela Oracle Corporation em 27 de janeiro de 2010.

O gráfico 2, mostra um estudo feito pelo Gartner Group (empresa especializada em pesquisas sobre tecnologia) sobre o *market share* (quota de mercado) existente entre os

SGBDs no ano de 2008. MySQL está a frente do SGBD da IBM (DB2), porém atrás do SGDB da Microsoft (SQL Server) e da Oracle (Oracle Database), para os quais precisam de licenciamento para sua utilização regularizada.

#### Gartner Group

##### Database Installations and Deployment Plans - 2008



Gartner Study Shows Strong Growth in the DBMS Market - 2008

**Gráfico 2 - Market Share dos SGBDs**

**Fonte: (Feinberg, 2008)**

No gráfico 3, tem-se o custo total de propriedade de três anos dos principais SGDBs utilizados.

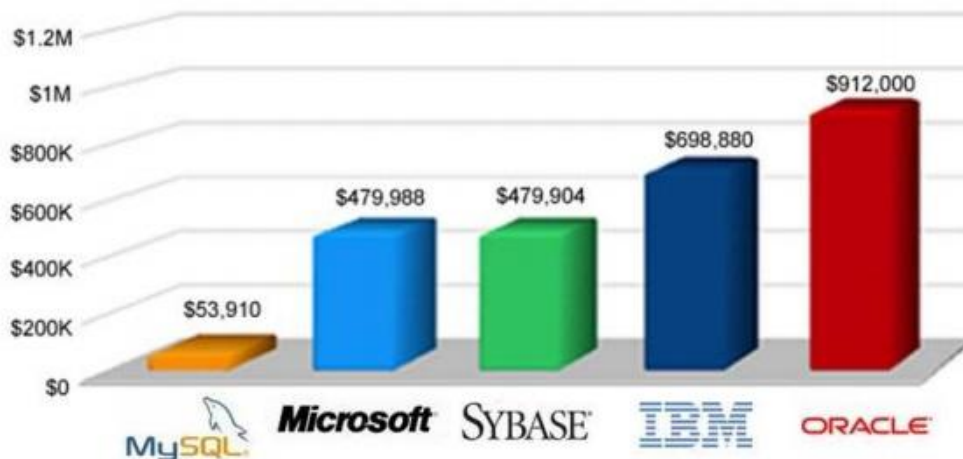


Gráfico 3 - Custos totais aos longo de 3 anos dos SGDBs

Fonte: (MySql, 2009)

Segundo Laurie e Laurie (1999), o servidor Apache é um dos principais servidores web existentes na infraestrutura da internet atualmente, é um *software open source*, além de ser *freeware* (sem custo de licenciamento), seu código fonte e instalador podem ser obtidos no site <http://www.apache.org>.

Servidores web tem como sua função principal traduzir uma URL (*Uniform Resource Locator*, um endereço para um recurso existente em uma rede, podendo ser internet, uma rede externa, ou intranet.) Uma rede interna, tem sua sintaxe como: `<scheme>://<host>:<port>/<path>`, esta é recebida em um nome de arquivo ou em um nome de programa no caminho para uma página, essa requisição é processada, e novamente enviada pela internet como saída da requisição feita pelo *browser* (programa de computador que possibilita que usuários interajam com servidores web).

Usando como exemplo a URL: <http://www.bkemmer.com/admin/>, temos como `<scheme>` HTTP (*Hypertext Transfer Protocol*, protocolo de comunicação utilizado na internet), `<host>` é [www.bkemmer.com](http://www.bkemmer.com), este é o caminho para que o servidor DNS (*Domain Name System*, servidor que traduz um nome de domínio em um endereço IP *Internet Protocol*, possibilita encontrar fisicamente a máquina que o tem como identificação) encontre o IP da máquina que contém o servidor web que executará a

requisição. Como não foi especificada uma *<port>* a requisição retorna na porta padrão que seria a 80, para o protocolo HTTP, e *<path>* /admin/, representando a navegação necessária dentro da estrutura de arquivos.

O protocolo TCP/IP (*Transmission Control Protocol / Internet Protocol*) é um grupo de protocolos que possibilitam que computadores se comuniquem mesmo estando em redes diferentes.

Um típico endereço IP seria: 192.168.0.1; existem quatro segmentos separados por pontos, cada parte corresponde a um *byte* (oito *bit*, o qual é a menor unidade utilizada na computação), portanto o tamanho total do endereço são quatro *bytes*. Por convenção, nesse número está contido, à esquerda, o endereço da rede e à direita, o endereço do *host*. Para se descobrir o quais campos representam o a rede e os que representam os computadores que estão internos na rede se utiliza *subnet masks* (máscaras de subrede, é um número de 32 bits) , estas especificam até qual bit deve ser utilizado como endereço de rede e por consequência os bits adjacentes a direita serão usados como endereço do *host*.

Se o endereço de rede é o mesmo, é fixada uma conexão TCP/IP e se inicia a troca de pacotes de dados entre os computadores. Caso contrário, os pacotes são enviados a um roteador que identifica o caminho a ser percorrido até a rede especificada pelo endereço de rede. Existem dois meios que utilizam IP como forma de comunicação:

- UDP (*User Data Protocol*), os pacotes são enviados porém não existe a garantia de recebimento no destino. O servidor DNS utiliza UDP, porém não é utilizado pelo servidor Apache.
- TCP (*Transmission Control Protocol*), pacotes de qualquer tamanho podem ser enviados, a ordem e o recebimento podem ser garantidos. Utilizado pelo servidor Apache.

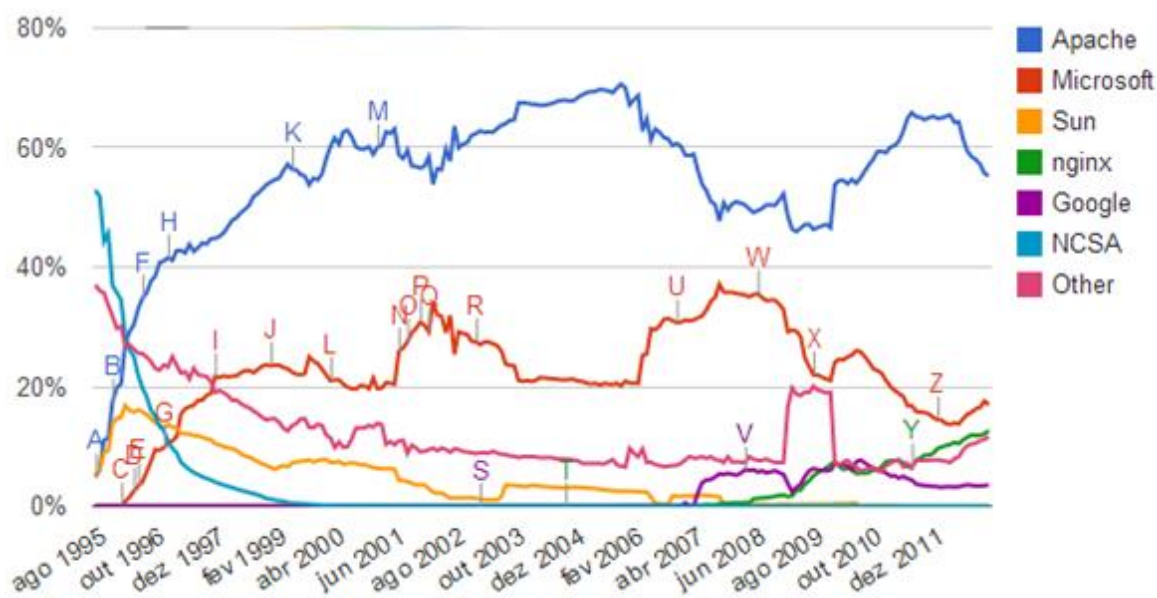
Um servidor web pode receber diferentes tipos de requisições de serviços, SMTP (*Simple Mail Transfer Protocol*, requisições de e-mail), DNS, HTTP, entre outras. O servidor decide como executar cada requisição, de acordo com a porta que esta foi enviada: SMTP, utiliza a porta 119, DNS, utiliza a porta 53 e HTTP, utiliza a porta 80.

O administrador do servidor pode adicionar qualquer serviço para qualquer porta (respeitando que são enviados dois bytes para determinar a porta, logo como cada byte suportam 255 combinações, existem 65025 opções de portas disponíveis).

O servidor Apache em seu estado idle (ocioso), verifica requisições ao endereço IP especificado no seu arquivo de configuração. Quando uma requisição é recebida, seu *header* (cabeçalho) é analisado, seguem-se as regras as quais estão presentes no arquivo de configurações e executa a ação apropriada. Na maioria dos casos, o servidor Apache recebe uma URL, executa seus *scripts* (códigos executados no interior de programas, no caso o servidor Apache), encontra um arquivo e o retorna pela internet.

De acordo com Apache (2013), em fevereiro de 1995, o servidor web mais popular foi desenvolvido por Rob McCool no NCSA (*National Center for Supercomputing Applications*, da Universidade de Illinois), porém seu desenvolvimento parou com a saída de Rob em 1994. Entretanto seu uso estava difundido entre os *webmasters* (desenvolvedores de páginas da internet), que se reuniram em uma lista de e-mail, criada por Brian Behlendorf e Cliff Skolnick. Utilizando como base o código NCSA httpd 1.3, foi disponibilizada uma versão beta (0.6.2) ao público em abril de 1995. Em 1º de dezembro de 1995, foi disponibilizada a versão 1.0.

Pelo gráfico 4, nota-se que o servidor Apache é líder de mercado desde abril de 1996 e pela tabela 2 é visto que em janeiro de 2013 o servidor Apache estava presente em 55,26% de todas as páginas web.



**Gráfico 4 - Market Share dos servidores web em todos os sites**

Fonte: (Netcraft, 2013)

**Tabela 2 - Market share dos servidores web por desenvolvedor**

**Fonte: (Netcraft, 2013)**

<b>Desenvolvedor</b>	<b>Dezembro 2012</b>	<b><i>Market Share</i></b>	<b>Janeiro 2013</b>	<b><i>Market Share</i></b>	<b>Alteração</b>
Apache	352.951.511	55,70%	348.119.032	55.26%	-0.43
Microsoft	111.570.010	17,61%	106.619.177	16,93%	-0.68
Nginx	76.460.756	12,07%	79.640.472	12,64%	0.58
Google	21.870.614	3,45%	22.573.858	3,58%	0.13

De acordo com Delisle (2012), o phpMyAdmin é uma aplicação desenvolvida na linguagem PHP, gratuita e de código aberto a qual oferece uma interface completa para gerenciar bancos de dados MySQL através da internet. Desde o início de seus desenvolvimento

O phpMyAdmin necessita de um servidor Web, como o Apache descrito no tópico anterior, já que o utiliza para enviar comandos na linguagem SQL para um servidor MySQL e receber dados do mesmo.

A grande maioria de seus utilizadores são:

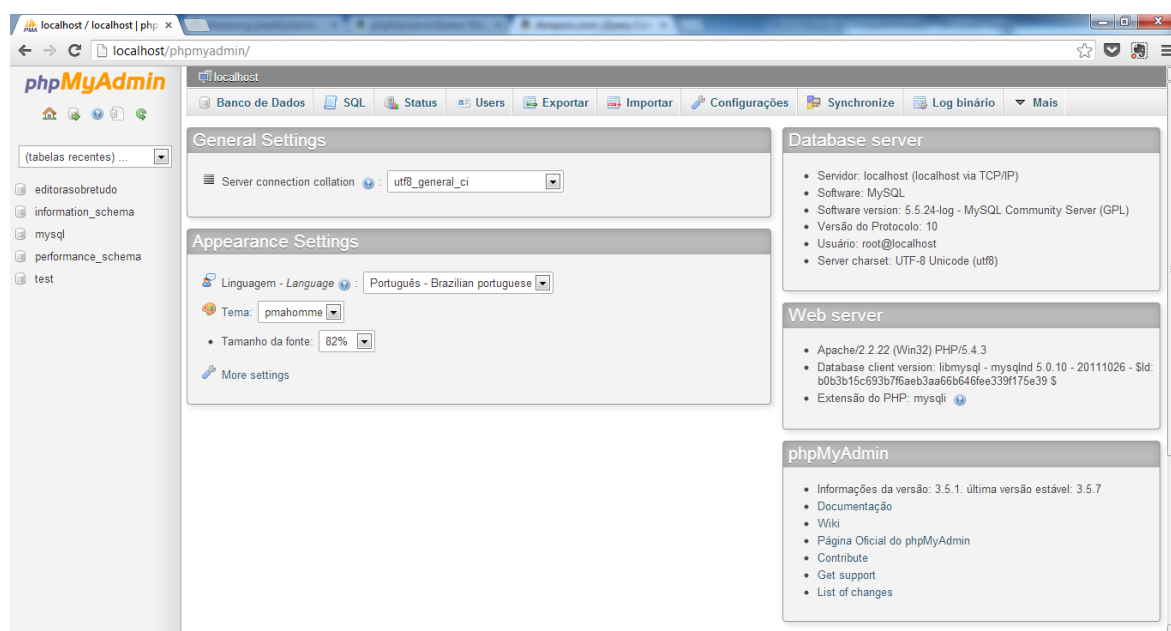
- Desenvolvedores Web
- Administradores de banco de dados.
- Alunos e professores de TI (Tecnologia da Informação)

O phpMyAdmin permite ao usuário tanto gerenciar as estruturas de dados (gerenciar usuários, alterar, deletar elementos como tabelas, bancos de dados, colunas, chaves, entre outros) quanto alterar os dados (criar, editar, deletar, ordenar, buscar em tabelas ou bancos de dados, além de dar a opção de importar ou exportar para vários formatos, além de outras funcionalidades.



Segundo (phpMyAdmin, 2013), seu desenvolvimento iniciou em 1998, por Tobias Ratschiller, inspirado por um projeto similar, mas por falta de tempo acabou abandonando o projeto no ano de 2000. Mas nesse momento o phpMyAdmin já era uma das mais populares ferramentas de administração de servidores MySQL com uma grande base de usuários e desenvolvedores.

A Figura 1, mostra a tela inicial da aplicação do phpMyAdmin, a esquerda há as bancas de dados existentes no sistema de gerenciamento de bancos de dados MySQL, e no menu superior, as opções existentes: Banco de dados (análise dos dados existentes no banco de dados selecionado), SQL (seção a qual dá a possibilidade de executar queries, trechos de código SQL, direto no servidor SQL), *status* (mostra como está o servidor no momento, quantidade de dados enviados e recebidos por hora, quantidade máxima de conexões de usuários simultâneas por hora, número de falhas), *users* (mostra as opções existentes para cada usuário incluindo seus dados, como senha entre outros aspectos), exportar (possibilidade de exportar para diversos formatos), importar (importar dados também de diversos formatos), configurações (as configurações técnicas do banco de dados selecionado), *synchronize* (modo de sincronização entre diferentes bancos de dados) e log binário (arquivo de log gerado pelo servidor SQL).



**Figura 1 - Tela inicial da aplicação phpMyAdmin**

**Fonte: phpMyAdmin, 2013**

A Figura 2 mostra a tela de um banco de dados selecionado; pode-se ver as tabelas existentes, e as ações disponíveis para cada uma delas que são: visualizar (visualizar os itens internos da coluna específica, estrutura (seus campos internos), procurar (busca elementos específicos), inserir (inserir manualmente), limpar e eliminar. Além disso, mostra também as informações sobre a tabela, como número de registros, tipo de dados, tamanho e entre outros dados.

Tabela	Ação	Registros	Tipo	Colação	Tamanho	Sobrecarga
columns_priv	Visualizar Estrutura Procurar Inserir Limpar Eliminar	0	MyISAM	utf8_bin	4 KB	-
db	Visualizar Estrutura Procurar Inserir Limpar Eliminar	2	MyISAM	utf8_bin	5.9 KB	-
event	Visualizar Estrutura Procurar Inserir Limpar Eliminar	0	MyISAM	utf8_general_ci	2 KB	-
func	Visualizar Estrutura Procurar Inserir Limpar Eliminar	0	MyISAM	utf8_bin	1 KB	-
general_log	Visualizar Estrutura Procurar Inserir Limpar Eliminar	2	CSV	utf8_general_ci	desconhecido	-
help_category	Visualizar Estrutura Procurar Inserir Limpar Eliminar	38	MyISAM	utf8_general_ci	24.6 KB	-
help_keyword	Visualizar Estrutura Procurar Inserir Limpar Eliminar	465	MyISAM	utf8_general_ci	105.5 KB	-
help_relation	Visualizar Estrutura Procurar Inserir Limpar Eliminar	1,029	MyISAM	utf8_general_ci	28 KB	-
help_topic	Visualizar Estrutura Procurar Inserir Limpar Eliminar	508	MyISAM	utf8_general_ci	454.4 KB	-
host	Visualizar Estrutura Procurar Inserir Limpar Eliminar	0	MyISAM	utf8_bin	2 KB	-
ndb_binlog_index	Visualizar Estrutura Procurar Inserir Limpar Eliminar	0	MyISAM	latin1_swedish_ci	1 KB	-
plugin	Visualizar Estrutura Procurar Inserir Limpar Eliminar	0	MyISAM	utf8_general_ci	1 KB	-
proc	Visualizar Estrutura Procurar Inserir Limpar Eliminar	0	MyISAM	utf8_general_ci	2 KB	-
procs_priv	Visualizar Estrutura Procurar Inserir Limpar Eliminar	0	MyISAM	utf8_bin	4 KB	-
proxies_priv	Visualizar Estrutura Procurar Inserir Limpar Eliminar	1	MyISAM	utf8_bin	5.7 KB	-
servers	Visualizar Estrutura Procurar Inserir Limpar Eliminar	0	MyISAM	utf8_general_ci	1 KB	-
slow_log	Visualizar Estrutura Procurar Inserir Limpar Eliminar	2	CSV	utf8_general_ci	desconhecido	-
tables_priv	Visualizar Estrutura Procurar Inserir Limpar Eliminar	0	MyISAM	utf8_bin	4 KB	-
time_zone	Visualizar Estrutura Procurar Inserir Limpar Eliminar	0	MyISAM	utf8_general_ci	1 KB	-
time_zone_leap_second	Visualizar Estrutura Procurar Inserir Limpar Eliminar	0	MyISAM	utf8_general_ci	1 KB	-
time_zone_name	Visualizar Estrutura Procurar Inserir Limpar Eliminar	0	MyISAM	utf8_general_ci	1 KB	-
time_zone_transition	Visualizar Estrutura Procurar Inserir Limpar Eliminar	0	MyISAM	utf8_general_ci	1 KB	-
time_zone_transition_type	Visualizar Estrutura Procurar Inserir Limpar Eliminar	0	MyISAM	utf8_general_ci	1 KB	-
user	Visualizar Estrutura Procurar Inserir Limpar Eliminar	4	MyISAM	utf8_bin	2.2 KB	-

**Figura 2 - Tela com um banco de dados selecionado**

**Fonte: phpMyAdmin, 2013**

Segundo w3Schools (2010), HTML (*HyperText Markup Language*), é a linguagem de publicação para o www (World Wide Web).

A linguagem utiliza *markup tags* (elementos com início e fim definidos e uma palavra descrevendo seu conteúdo, exemplo: <p>Olá</p> representa um parágrafo com delimitadores <p> para o início e </p> para seu final) para descrever páginas Web. As tags não serão exibidas, porém serão interpretados para depois serem exibidos pelo *Web browser*.

Atualmente, HTML está na sua versão 5.0; esta provê suporte nativo a recursos multimídia, além de recursos para aparelhos de baixa potência como *smartphones* (aparelhos de telefonia móveis com sistema operacional e capacidade de computação) e *tablets* (computadores móveis operados por uma tela sensível ao toque).

De acordo com Alexander, Kmiec, Lam e Raggett (1997), a *World Wide Web* começou no CERN (laboratório de partículas físicas em Genebra, Suíça). O protótipo da linguagem foi desenvolvido em 1992, por Tim Berners-Lee. Tim trabalhava no departamento de serviços de computação do laboratório. Muitas vezes, as pesquisas envolviam a colaboração entre cientistas ao redor do mundo; Tim teve a idéia de possibilitar os cientistas a remotamente organizar e agrupar as informações pertinentes, não somente disponibilizar para *download* (ato de descarregar um arquivo da internet para seu computador) mas adicionar um *link* (uma ligação) ao arquivo em si.

Como os métodos de publicação na época eram específicos para a plataforma em que estes eram utilizados, ou de utilização complexa, Tim criou seu próprio protocolo, HTTP (*HyperText Transfer Protocol*), o formato de texto a ser utilizado com este protocolo seria HTML. Tim demonstrou seu uso em um NeXT (computador pessoal desenvolvido na década de 1990).

O HTML é fortemente baseado em SGML (*Standart Generalized Mark-up Language*), um método internacional de marcação de texto em unidades estruturais como parágrafos, títulos, lista de itens, entre outros. Porém o tópico principal que o SGML não incluía eram os hyperlinks em si, além disso a linguagem HTML é mais simples que a SGML o que aumentou sua chance de sucesso.

Em abril de 1993, a NCSA, lançou a versão 1.0 de seu próprio *browser*, o Mosaic; outros *browsers* que existiam no momento eram Lynx (um *browser* que suportava apenas modo texto para terminais e máquinas apenas com o sistema operacional DOS, desenvolvido por Lou Montulli) e Arena (desenvolvido em seu tempo livre por Dave Raggett, um pesquisador dos laboratórios da Hewlett-Packard).

Em 1994, na primeira conferência *World Wide Web* foi definido o padrão HTML 2, agrupando expansões da linguagem já utilizadas nos browsers, porém não padronizadas.

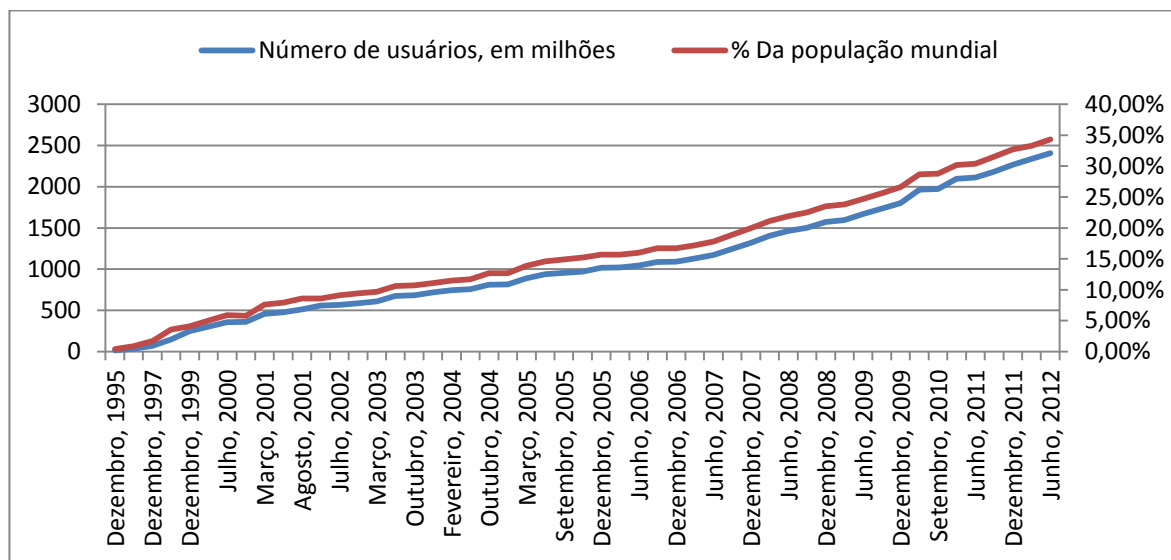
Em novembro de 1994, foi fundada a Netscape Communications Corporation, a qual criou o *browser* Netscape, desenvolvido por Marc Andreessen, após sair do projeto de desenvolvimento do *browser* Mosaic. O Netscape se tornou um sucesso, devido ao fato de prover acesso a WWW a todos incluindo os que tinham uma conexão de internet de velocidade reduzida, algo frequente na época.

No final de 1994, foi formado o *World Wide Web Consortium* (W3C), com a liderança de Tim Berners-Lee, e formado pelas principais empresas e desenvolvedores da linguagem na época.

Em março de 1995 foi desenvolvida a HTML 3 e a Microsoft desenvolveu a versão 1.0 do seu *browser* o Internet Explorer em agosto de 1995. Em 1999, foi lançado o HTML 4.01. Segundo W3C (2013), a linguagem XHTML (*Extensible HyperText Markup Language*), na versão 1.1 se tornou uma recomendação W3C em 31 de maio de 2001. Esta é um conjunto da linguagem XML (*Extensible Markup Language*), também uma linguagem de marcação porém mais rígida que a SGML. Como os elementos da linguagem XHTML precisam ser bem-formados (significa que precisam satisfazer uma lista de regras de sintaxe), portanto podem passar por um analisador de sintaxe, auxiliando a codificação.

O HTML na sua versão 5.0 foi uma recomendação da W3C em dezembro de 2012, e tem a intenção de substituir além da versão HTML 4.0, o XHTML. Seu foco é adicionar conteúdo multimídia, os quais eram processados por softwares externos, como flash, e prover uma sintaxe que possa ser escrita usando tanto a sintaxe HTML quanto XHTML.

No gráfico 5, é visto o crescimento da internet ao passar dos anos, atingindo 35% da população mundial em Junho de 2012.



**Gráfico 5 - Número de usuários da internet**

**Fonte: (Internet World Stats, 2012)**

Segundo W3C (2013), CSS (*Cascading Style Sheets*), é a linguagem que descreve como as páginas Web serão apresentadas, incluindo suas cores, disposição e fontes de todo texto presente.

Além dos aspectos citados, possibilita a apresentação a dispositivos que tenham configuração distintas, desde telas grandes a pequenas ou mesmo uma formatação específica quando o destino é uma impressora.

O CSS, mesmo sendo muitas vezes empregado em conjunto com a linguagem HTML, é independente da mesma, a linguagem CSS pode ser utilizada para descrever o estilo de qualquer linguagem baseada na linguagem XML.

A separação entre o código HTML (deve ter seu foco no conteúdo e estrutura) e o código CSS (contendo o estilo), facilita a manutenção, o reuso de códigos para definir o estilo através das páginas, e deixá-las personalizadas para cada ambiente, além de possibilitar um nível de acessibilidade aos sites.

As especificações CSS são mantidas pelo W3C; atualmente, o CSS está na sua versão 3.0, contendo: *media queries* (possibilidade do conteúdo se ajustar a diferentes resoluções de tela; exemplo: desde um *smartphone* até uma tela de alta definição), *namespaces* (criação de um ambiente; neste, as definições tem seu escopo apenas ao ambiente definido), *selectors* (utilizados para selecionar elementos tanto da linguagem HTML quanto documentos XML, para poder prover estilo a suas propriedades, podem ser selecionados tanto utilizando seus nomes, atributos, contexto, entre outros), além de outras funcionalidades.

Folhas de estilo existem desde o início da linguagem SGML na década de 1980, porém as folhas de estilo em cascata foram desenvolvidas para prover estilo às informações presentes nas páginas Web. Com a evolução da internet, a necessidade cada vez maior de um controle sobre a aparência das páginas, e o desejo de personalização dos desenvolvedores Web, trouxe um nível alto de complexidade aos códigos HTML, e conforme os códigos cresciam dificultavam sua manutenção. Em dezembro de 1996, o CSS versão 1.0 já era uma recomendação W3C.

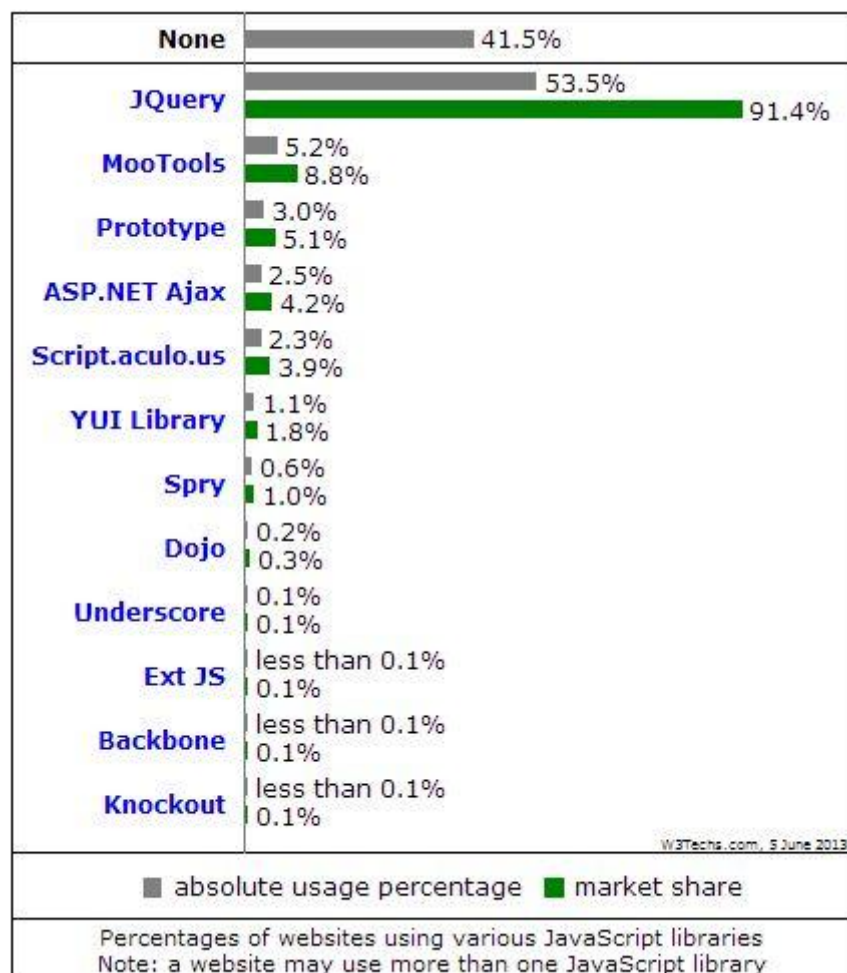
Segundo, jQuery Foundation (2013), jQuery é uma biblioteca JavaScript pequena, rápida e com muitos recursos. Pode facilitar manipulações HTML, gerenciamento de

eventos, animações e Ajax ( Asynchronous Javascript and XML, uso de tecnologias JavaScript e XML de modo assíncrono). Esta tem seu código aberto.

Suas principais funcionalidades são:

- Redução em linhas de código.
- Reutilização de código via plugins.
- Implementação de recursos CSS, nas versões 1, 2 e 3.
- Aumento da compatibilidade entre browsers.

Foi desenvolvida em janeiro de 2006, por John Resig e atualmente é utilizada por cerca de 55% dos 10 mil sites mais visitados do mundo. É portanto a biblioteca mais popular da linguagem JavaScript. Como pode ser visto no gráfico 6, tem um *market share* de 91,4%.



**Gráfico 6 - Percentual de sites que utilizam bibliotecas Javascript**

## **2.4. Considerações Finais**

Nesse capítulo foi apresentado o conjunto de tecnologias necessárias para a implementação e manutenção de uma plataforma web, seu estado da arte e sua evolução histórica. No capítulo seguinte será apresentado o desenvolvimento do trabalho efetuado, a metodologia utilizada e a arquitetura resultante.

## CAPÍTULO 3: MATERIAIS E MÉTODOS

Esse capítulo contém as etapas de projeto: especificação; estruturas de dados; organização das principais páginas PHP e suas funções; organização das páginas HTML e javascript, além de mostrar a arquitetura conceito do sistema.

### 3.1. Especificação

Desenvolver um módulo de uma página web que disponibilize os dados de seus integrantes, segmentados por seu campo de atuação dentro do laboratório e um módulo de administração para sua manutenção.

Será utilizado o ambiente de desenvolvimento web WampServer, para a plataforma Windows, neste está contido: servidor web (Apache); módulo PHP e o sistema de gerenciamento de bancos de dados (MySQL), além disso também contém a aplicação PhpMyAdmin.

### 3.2 Arquitetura conceito

A figura 3 mostra a arquitetura conceito da plataforma.

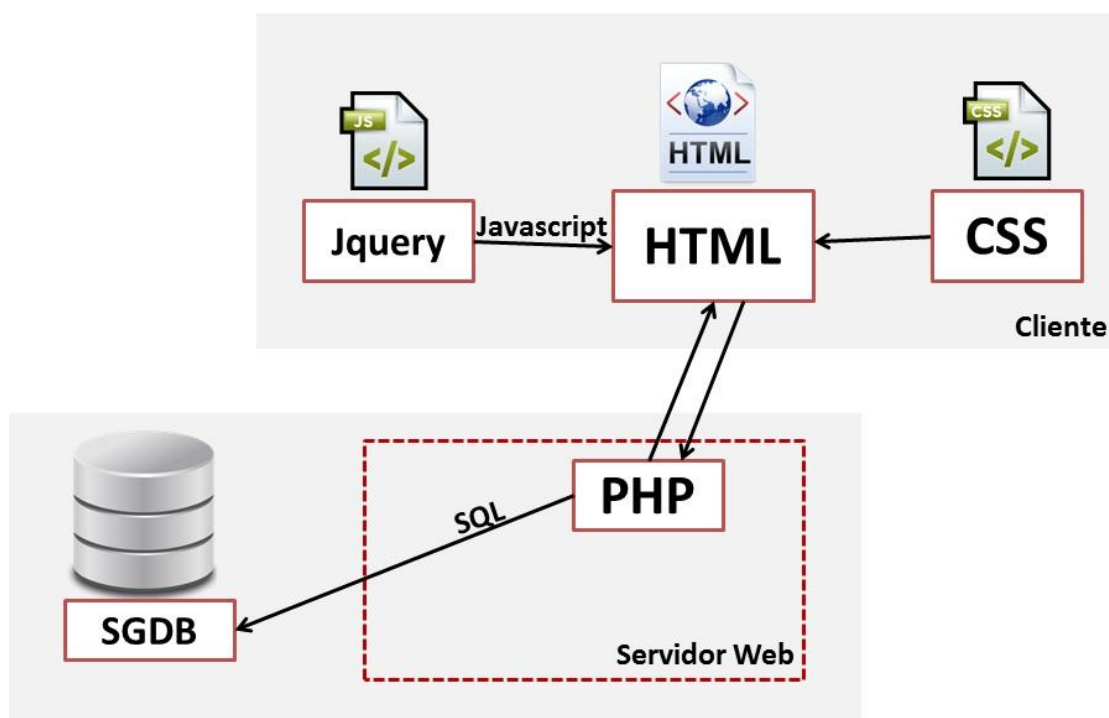


Figura 3 - Arquitetura conceito da plataforma



Nela pode ser vista a divisão do paradigma cliente-servidor, na parte superior estão os elementos processados no *browser* do usuário, separados no HTML, o conteúdo e no CSS, sua apresentação, já na parte inferior estão os elementos processados no servidor, o módulo PHP dentro do servidor web que acessa o SGDB via comandos SQL, e recebe requisições via métodos HTTP das páginas HTML e retorna páginas HTML como resultado do processamento efetuado pelo servidor web.

### 3.3. Estrutura de dados

Como pode ser visto na figura 4, a estrutura será interligada pela chave “idparticipant”, está é chave primária da tabela “participants” e chave estrangeiras das outras com exceção da tabela “new\_user”, já que esta contém apenas os usuários que receberam um e-mail para se cadastrarem no sistema e ainda não o fizeram.

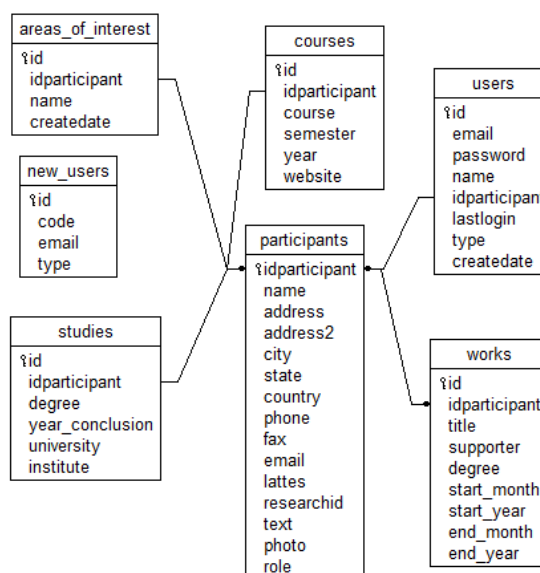


Figura 4 - Mapeamento do banco de dados

### 3.4. Descrição das páginas PHP

O script `index.php` retira do banco de dados as linhas, segmentadas por: professores; mestrandos; doutorandos; colaboradores e iniciação científica. Já os scripts: `professores.php`; `mestrandos.php`; `doutorandos.php`; `colaboradores.php` e `ic.php`, retornam as linhas apenas que contém os papéis que o nome do arquivo já exemplifica.

No script `profile.php`, é executada a *query* no banco de dados, com a entrada de dados recebida pelo método GET do HTTP (uma página web para informação para outra na URL, no caso, passa o elemento a ser buscado no banco de dados). Este após receber as linhas referentes aos atributos que o elemento buscado tem os posiciona na tela.

No módulo administrativo, temos como principais scripts: `index.php`, este executa `head.php` que contém o cabeçalho da página e `topo.php` que contém o menu de escolha (padrão para todos os scripts presentes nesse módulo), a partir dessa página o usuário pode entrar nas seguintes páginas: `perfil.php`, executa uma *query* com a chave “`idparticipant`” do usuário que está com o *login* ativo recebe seus dados e os posiciona na tela para uma alteração; `photo.php`, executa o script "PHP & JQuery image upload & crop" para submissão da imagem e ajustes ;`studies.php`, recebe as informações relativas a universidade do usuário ativo; `áreas.php`, recebe as informações das áreas de interesse do usuário ativo ;`courses.php`, recebe as linhas referentes aos cursos do usuário; `trabalhos.php` contém as linhas de trabalhos do usuários e `usuários.php`, página para o gerenciamento das permissões dos usuários.

Todas as páginas executam o arquivo `config.php`, este contém as funções PHP utilizadas e descritas na seção abaixo além de conter as variáveis para a conexão com o banco de dados, e no arquivo `config.php` é executado a página `login.php`, esta faz a verificação de usuário e senha. No arquivo `imagizer.php`, está a função para o upload das imagens ao site.

### 3.5. Funções PHP

A função `ProtegePagina`, verifica se tanto a variável referente ao usuário quanto a senha estão inicializadas e pertencem ao usuário atual. A função `VerificaPerm` verifica se o usuário tem permissão para acessar a página atual e funções de data e validação de strings.

# **CAPÍTULO 4: DESENVOLVIMENTO DO TRABALHO**

Nesse capítulo será apresentado o escopo do projeto, a proposta de trabalho e a metodologia utilizada.

Além disso será mostrada a arquitetura resultante do sistema, o modelo de entidade-relacionamento da plataforma e as telas resultantes da implementação.

## **4.1. Projeto**

O escopo desse trabalho é prover uma interface entre o laboratório de pesquisa e o ambiente externo, que fosse acessível, ao mesmo tempo em que mostrasse o máximo de dados relevantes sobre seus participantes, com manutenção prática, além de ser funcional.

A proposta do trabalho é criar uma versão modular de um site, com o perfil de seus integrantes segmentados conforme sua área de atuação no laboratório (professores, mestrandos, doutorandos, colaboradores e alunos de iniciação científica), podendo, assim, ser embutido, o módulo desenvolvido, ao site já existente do grupo, além de possibilitar uma expansão do módulo para uma implementação completa.

O projeto consiste nas seguintes fases:

1. Levantamento dos requisitos – Nessa fase são feitas entrevistas com os usuários do atual sistema e dos futuros usuários, levantado quais requisitos gostariam que houvesse, no caso, quais dados são interessantes disponibilizar. E desenho de um protótipo de parte do site.
2. Análise dos requisitos – Estudam-se os itens recebidos, quais produtos existentes no mercado são mais aderentes ao projeto e é definido qual a arquitetura a ser utilizada.
3. Implementação – Nessa etapa, desenvolve-se o código de cada parte do módulo em questão.
4. Teste – Com o sistema online, executam-se os testes e os ajustes finais.

## **4.2. Descrição das Atividades Realizadas**

### **4.2.1. Levantamento dos requisitos**

Foram levantadas os perfis necessários ao site:

- Professor
- Colaborador
- Mestrando
- Doutorando
- Iniciação Científica

As diferenças entre os usuários:

- Para professores é interessante que tenham a possibilidade de adicionar disciplinas.
- Para “alunos” (mestrando, doutorando e iniciação científica), campo, título do trabalho e tipo de bolsa.
- No perfil colaborador, os mesmos campos de docentes porém sem a possibilidade de adicionar disciplinas.

**LPS**  
Laboratório de Processamento de Sinais

Principal   Pessoal   Pesquisas   Eventos   Contatos

**FOTO**

**DADOS:**  
 Nome:  
 Work Address:  
 RUA, numero, bairro  
 CEP, cidade, estado pais  
 tel, fax  
 email  
 Education: Opção de colocar multiplas linhas de estudo  
 Areas of Interest:  
 multiplas linhas  
 Publications  
 International journals  
 Braziliant Journals  
 Refereed Conference Publications

Campo para a pessoa colocar um texto de sua preferencia

Trabalhos  
tipo de bolsa

Link lattes   Research ID button

Área Restrita

**Figura 5 - Protótipo da visão do perfil de um aluno**

Como pode ser visto na Figura 5, foram levantados os dados necessários para o perfil de um aluno (mestrando, doutorando, iniciação científica), e como seria interessante que fosse disponibilizada essa informação. Imagem validada com o usuário final, além de mostrar o módulo desenvolvido embutido no layout do site atual, (cabeçalho e menu).

Já a Figura 6, mostra o segundo protótipo disponibilizado, mostrando a visão do perfil de um docente.

**LPS**  
Laboratório de Processamento de Sinais

Principal   Pessoal   ▾   Pesquisas   Eventos   Contatos

FOTO	<b>DADOS:</b> Nome: Work Address: RUA, numero, bairro CEP, cidade, estado país tel, fax email Education: Opção de colocar multiplas linhas de estudo Areas of Interest: multiplas linhas Publications International journals Braziliant Journals Refereed Conference Publications	Campo para a pessoa colocar um texto de sua preferencia
	Disciplinas: Nome da materia   link do jupyter	
Link lattes	Research ID botton	

[Área Restrita](#)

**Figura 6 - Protótipo da visão do perfil de um docente**

Não foi desenvolvido um protótipo para os perfis classificados como “colaboradores” já que se diferenciavam minimamente com o perfil dos professores.

Foi determinado que seria necessário um módulo administrativo para o site, mas como pode ser visto foi colocado no protótipo apenas um hyperlink (Área Restrita), o qual redirecionaria para está parte do site, porém como será descrito, foi utilizada uma outra abordagem.

#### 4.2.2. Análise dos requisitos

Após o levantamento dos requisitos, desenvolvimento dos protótipos, feita a revisão bibliográfica dos temas (principalmente do paradigma cliente-servidor), decidiu-se pela arquitetura que pode ser vista na Figura 7.

No retângulo inferior, temos os módulos, os quais são executados no servidor; este tem como servidor web, o software open source Apache, que executa os arquivos PHP. Embutidos nestes arquivos PHP existem queries SQL que executam buscas no servidor de banco de dados (SGDB) Mysql, este último pode estar na mesma máquina física que o

servidor, ou conectado nele via rede. Estas buscas retornam visões com os resultados, que são processados pelo servidor web, e como resultado final é gerado um arquivo html (este contém o conteúdo da página). No arquivo HTML tem um link para um arquivo css, que contém as regras de apresentação. Além disso, algumas páginas contêm funções que utilizam a linguagem Javascript (códigos executados no browser do usuário), e algumas funções da biblioteca JQuery. Na parte “cliente” está o produto final que são interpretadas pelo browser do usuário.

A caixa que contém phpmyadmin, representa o programa de administração de banco de dados que será executado no servidor web, este provê uma interface gráfica para facilitar o desenvolvimento de queries administrativas, exemplo: “SELECT \* FROM TABELA”, mostra todos os elementos, isso pode ser feito apenas usando uma seção própria que mostra todas as linhas existentes da tabela, para o gerenciamento dos usuários que podem acessar o banco de dados, as ações que forem executadas nele, são posteriormente transformadas em queries e enviadas e executadas pelo MySQL (SGDB).

Após essa etapa era necessário observar quais eram os relacionamentos entre os agentes que interagem no site e seus atributos, como pode ser visto na Figura 8. Feita essa análise pode-se mapear esses campos no banco de dados.

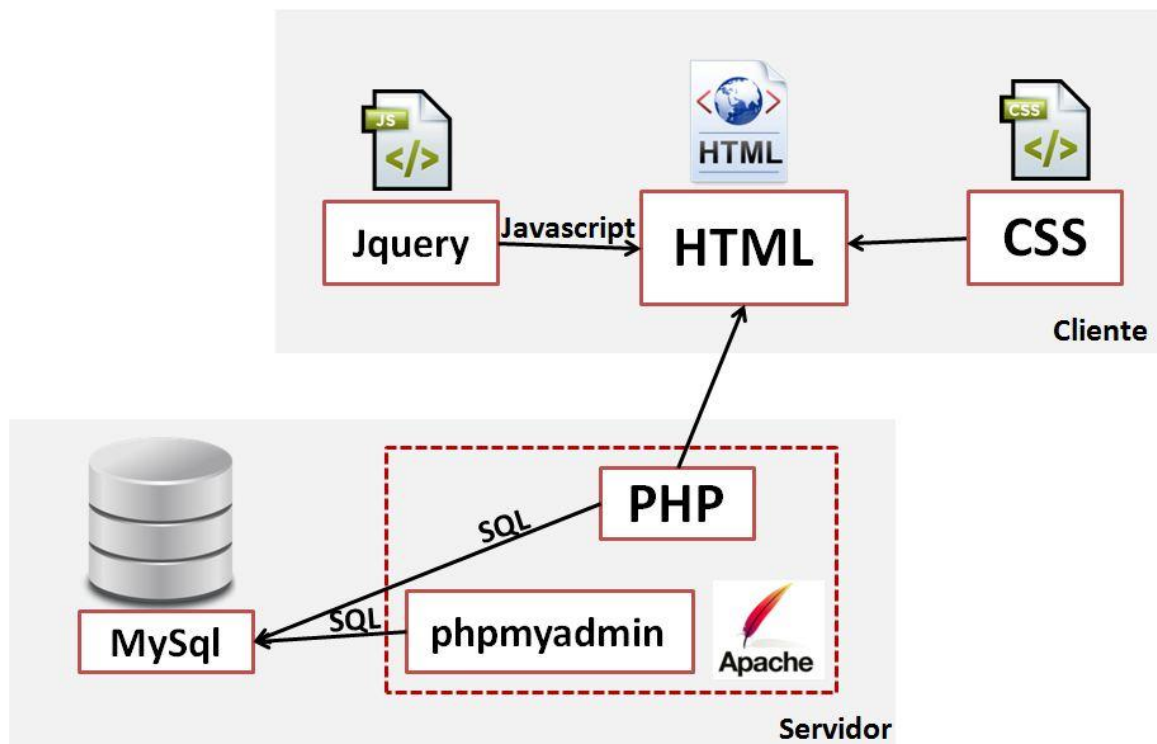


Figura 7 - Arquitetura da plataforma no sistema Windows

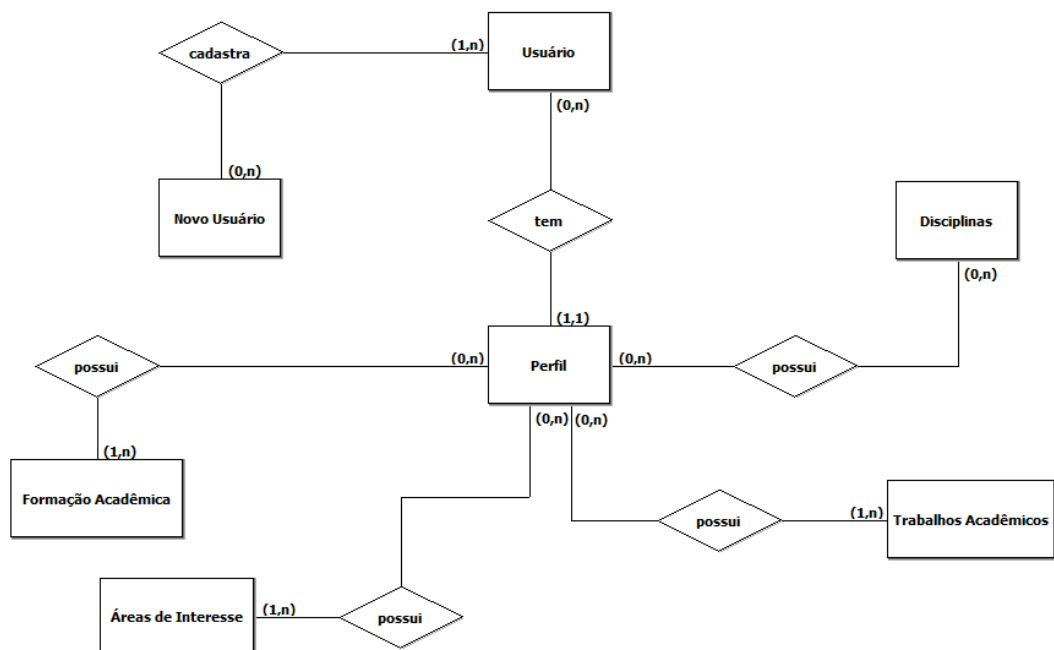


Figura 8 - Diagrama de Entidade-Relacionamento da plataforma

### **4.2.3. Implementação**

Foi implementado um arquivo PHP para cada perfil de usuário, além do conjunto de arquivos necessário para o módulo administrativo do site, os arquivos css contendo as regras de apresentação.

Para a integração com o site existente foi desenvolvido uma versão inicial mostrando todos os integrantes segmentados por perfil, em forma de lista.

### **4.2.4. Teste**

Ao longo da implementação foram encontradas falhas e estas foram sendo corrigidas. Foram vistas diferenças entre as apresentações em cada *browser*, como será comentado adiante.

## **4.3. Resultados Obtidos**

Foi obtido um conjunto de arquivos PHP (7 arquivos) referentes à apresentação dos dados para mostrar todos os segmentos (possíveis, professores, colaboradores e cada segmento de aluno), e à página inicial (index.php, Figura 9), podendo assim ser acrescentado ao site já existente do grupo, e um arquivo para mostrar os colaboradores de cada perfil (5 arquivos, exemplo visto do resultado do arquivo professores.php na Figura 10). Também existe um arquivo (profile.php, Figura 11) que executa a query para obter os dados do participante em questão.



Professores:

- [Bruno Abreu Kemmer](#)
- [Matheus Alejandro](#)

Mestrandos:

Doutorandos:

Colaboradores:


Iniciação Científica:

- [Pedro Alvares](#)

Figura 9 – Resultado do processamento do arquivo index.php no servidor



Figura 10 - Resultado do processamento do arquivo professores.php no servidor



### Bruno Abreu Kemmer

Endereço: Rua Bacaetava, 121  
 Cidade: São Paulo / SP  
 País: Brasil  
 Telefone: 5511997452461  
 Formação Acadêmica:  
 2012 - Graduação  
 Engenharia de Computação  
 Universidade de São Paulo - Escola de  
 Engenharia de São Carlos

Áreas de Interesse:  
 Business Intelligence

Aluno do curso de engenharia de computação, tem interesses diversos e além do interação web que existe hoje em dia.

### Disciplinas:

Trabalho de conclusão de curso (1º/2013)  
[Link Jupiter](#)

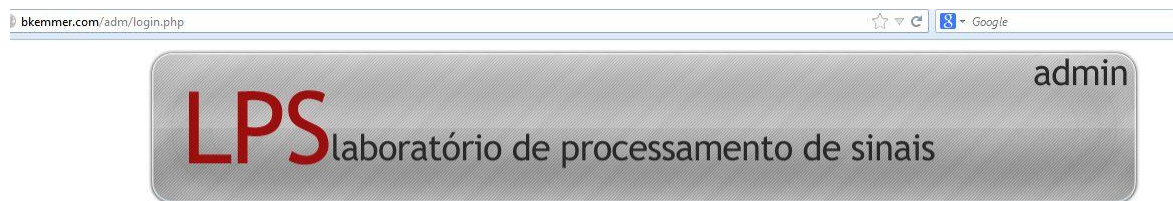
### Trabalhos:

TESTE 1  
 (1/2012 - 1/2012)  
 Iniciação Científica  
 Agência Financiadora: CAPES

[CV Lattes](#)

**Figura 11 – Resultado do processamento do arquivo profile.php para o usuário selecionado**

Além desses, foram implementados um conjuntos de 16 arquivos PHP para executar todos os requisitos necessários para a manutenção efetiva do módulo, estes podem ser acessados no arquivo index.php que está presente na pasta /adm/, este redireciona o usuário para o arquivo login.php, que pode ser visto na figura 12, e a partir deste arquivo pode-se utilizar todos os outros que estão presentes no módulo administrativo.



bkemmer.com/adm/login.php

admin

**LPS** laboratório de processamento de sinais

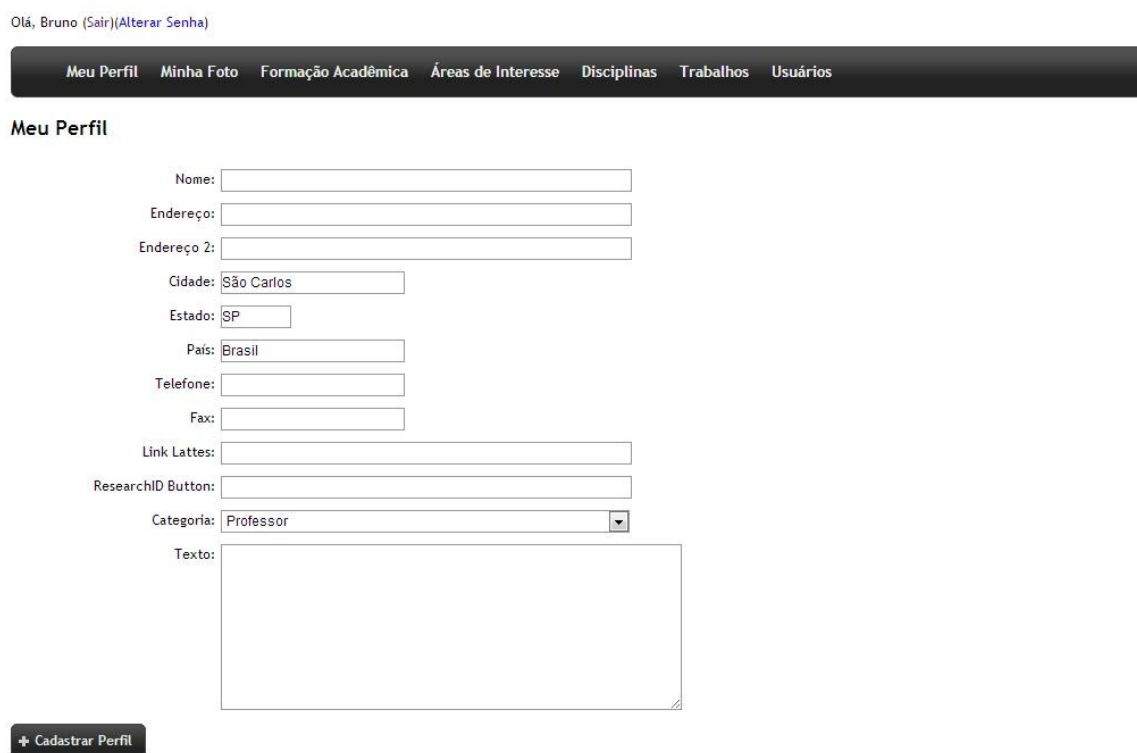
E-mail:

Senha:

[Entrar](#) [Esqueci minha senha](#)

**Figura 12 - Resultado do arquivo login.php**

No primeiro menu encontra-se o botão “Meu Peril”, e no momento em que o usuário clica nele é redirecionado para o resultado do processamento do arquivo, perfil.php que pode ser visto na figura 13.



Olá, Bruno (Sair)(Alterar Senha)

[Meu Perfil](#) [Minha Foto](#) [Formação Acadêmica](#) [Áreas de Interesse](#) [Disciplinas](#) [Trabalhos](#) [Usuários](#)

**Meu Perfil**

Nome:

Endereço:

Endereço 2:

Cidade:

Estado:

País:

Telefone:

Fax:

Link Lattes:

ResearchID Button:

Categoria:

Texto:

[+ Cadastrar Perfil](#)

**Figura 13 – Visão administrativa da plataforma – Seção “Meu Perfil”**

O segundo botão (“Minha Foto”) redireciona para a página photo.php, que pode ser vista na figura 14, nela o usuário pode fazer o upload de uma foto, e selecionar apenas a parte que o interessa.

## Foto

### Upload Foto

Arquivo:

### Gerenciar sua Foto

Clique na imagem para cortar



**Figura 14 – Seção de gerenciamento de foto**

No botão seguinte, “Formação Acadêmica”, o usuário pode adicionar todas as formações que obteve na sua carreira, como pode ser visto na figura 15.

### Estudos


 O estudo foi cadastrado com sucesso

Universidade:


Instituto/Faculdade:

Estudos:

Fim:

 Cadastrar Estudos

### Gerenciar Sua Formação Acadêmica

ID	Estudos	Universidade/Instituto	Encerrado	Funções
3	Graduação	Escola de Engenharia de São Carlos - Engenharia de Computação/Universidade de São Paulo	2012	 Excluir

**Figura 15 - Seção formação acadêmica**

No botão “Disciplinas”, o usuário pode adicionar todas as disciplinas que está atualmente ministrando (essa opção só está disponível para professores), e adicionar a elas o semestre que está sendo ministrada além de disponibilizar um link para o sistema de informação (no caso o sistema Jupyterweb, da Universidade de São Paulo), isso pode ser visto na figura 16.

### Disciplinas

 A disciplina foi cadastrada com sucesso

Nome Disciplina:

Link Júpiter:

Semestre:

Ano:

 Cadastrar Disciplina

### Gerenciar Suas Disciplinas

ID	Disciplina	Semestre/Ano	Funções
10	Trabalho de conclusão de curso	01º/2013	 Excluir

**Figura 16 - Seção disciplinas**

A figura 17 mostra todos os trabalhos já efetuados pelo dono do perfil selecionado, podendo adicionar quantos forem necessários.

Olá, Bruno (Sair)(Alterar Senha)

Meu Perfil Minha Foto Formação Acadêmica Áreas de Interesse Disciplinas Trabalhos Usuários

### Trabalhos

! O trabalho foi cadastrado com sucesso

Título:

Agência Financiadora:

Estudos:

Início:  /

Fim:  /

Cadastrar Trabalho

### Gerenciar Seus Trabalhos

ID	Título	Estudos	Agência Financiadora	Encerrado	Funções
3	TESTE 1	Iniciação Científica	CAPES	12/2012	<a href="#">Excluir</a>

Figura 17 - Seção trabalhos

A figura 18, mostra como o sistema permite o gerenciamento das permissões dos usuários, a primeira tabela mostra os usuários que receberam um e-mail com um link para fazer o cadastro, e a segunda tabela mostra os usuários que estão cadastrados no sistema.

Olá, Bruno (Sair)(Alterar Senha)

Meu Perfil Minha Foto Formação Acadêmica Áreas de Interesse Disciplinas Trabalhos Usuários

### Cadastrar Usuário

E-mail:

Permissão:

Finalizar cadastro

### Gerenciar Novos Usuários

ID	Email	Funções
0	cdmaciel@gmail.com	<a href="#">Excluir</a>
0	ich_gaelzer@hotmail.com	<a href="#">Excluir</a>
0	bruno.kemmer@oracle.com	<a href="#">Excluir</a>

### Gerenciar Usuários

ID	Nome	Email	Criado em:	Funções
8	Bruno	brunokemmer@gmail.com	10/03/2013	
2	Carlos Maciel	cdmaciel@gmail.com	07/10/2012	<a href="#">Excluir</a>

Figura 18 - Seção usuários

## **4.4. Dificuldades e Limitações**

O resultado obtido cumpriu os objetivos do projeto, proporcionou um módulo de gerenciamento do perfil dos integrantes do laboratório de processamento de sinais, de um modo dinâmico e que possibilite uma manutenção prática. Essa solução pode ser efetivamente utilizada na plataforma atual, embutindo os arquivos no servidor atual em que o site está hospedado, com o único requisito de ter um servidor web instalado e adicionar corretamente os links para os arquivos php.

Algumas lições foram aprendidas: no desenvolvimento de uma aplicação web é interessante fazer testes, pelo menos, nos principais browsers disponíveis atualmente, já que estes não são padronizados; logo é necessário encontrar uma versão a qual fique com qualidade em todos. A dificuldade maior, muitas vezes, se encontra na apresentação dos elementos, um trabalho mais qualitativo do que analítico.

Outra dificuldade encontrada foi o número de tecnologias envolvidas no desenvolvimento web, tanto em linguagens (php, css, html, javascript) quanto em softwares (mysql, phpmyadmin, apache), cada uma delas com um nível de dificuldade diferente porém todas envolvem um aprendizado.

## **4.5. Considerações Finais**

Nesse capítulo discorreu-se sobre a evolução do trabalho, desde sua concepção, planejamento, metodologia utilizada e os resultados obtidos, além de mostrar algumas dificuldades levantadas.

No capítulo seguinte, serão apresentadas as conclusões levantadas após a execução e análise dos resultados do projeto.

# **CAPÍTULO 4: CONCLUSÃO**

## **4.1. Contribuições**

O trabalho efetuado proporcionou uma visão de uma arquitetura de um sistema web. Como o projeto envolve o uso de diversas tecnologias (4 linguagens e 3 softwares) para que o sistema seja executado, fazer todas funcionarem e cooperarem para que o sistema seja executado sem falhas e de modo mais perto do ideal foi um desafio alcançado.

O trabalho proporcionou ao aluno uma visão histórica da evolução da internet ao longo dos anos, seu crescimento vertiginoso no passado recente, e conhecimentos técnicos de cada uma das tecnologias utilizadas, tópico extremamente importante no mundo cada vez mais conectado em que vivemos.

## **4.2. Trabalhos Futuros**

Como trabalhos futuros, o módulo desenvolvido poderia ser expandido, aumentando a usabilidade e praticidade do site atualmente utilizado pelo grupo, e deixando sua manutenção mais prática.

A integração com redes sociais é algo valioso para aumentar o alcance do grupo e da universidade como um todo. Além da possibilidade de uso de uma ferramenta como o twitter para postar artigos relevantes tanto do grupo quanto de terceiros que os administradores julguem interessante compartilhar com o público.



# REFERÊNCIAS

ACHOUR, M; BETZ, F; DOVGAL, A. et al. “Manual do PHP” In: Php.net, 07/abr/2013, Disponível em <[http://php.net/manual/pt\\_BR/index.php/](http://php.net/manual/pt_BR/index.php/)>.

ALEXANDER, I; KMIEC, Michael; LAM, J; RAGGETT, D. Raggett on HTML 4, Boston, Ed. Addison-Wesley Professional, 1997.

Apache, “About the Apache HTTP Server Project - The Apache HTTP Server Project” In: Apache, 07/abr/2013, Disponível em <[http://httpd.apache.org/ABOUT\\_APACHE.html](http://httpd.apache.org/ABOUT_APACHE.html)>.

CONVERSE, T.; MORGAN, C.; PARK, J. PHP5 and MySQL Bible, Hoboken, Ed. Wiley, 2004.

DELISLE, Marc. phpMyAdmin Starter. Birmingham, Ed. Packt Publishing Ltd, 2011.

FEINBERG, Donald. “Gartner Study Shows Strong Growth in the DBMS Market”. In: Gartner, 2008.

Internet World Stats, “Internet World Stats” In: Internet World Stats, 2012. Disponível em <<http://www.internetworldstats.com/emarketing.htm>>.

jQuery Foundation, “jQuery” In jQuery, 2013. Disponível em <<http://jquery.com/>>.

LAURIE, B.; LAURIE, P. Apache: The Definitive Guide. Sebastopol, Ed. O`Reilly, 1999.

MySQL. “*CIO Guide: The Strategic Value of MySQL*”. In: Mysql, 2009, Disponível em <[http://www.theleanmachine.com/MySQL\\_WhitePaper\\_CIO\\_v6.pdf](http://www.theleanmachine.com/MySQL_WhitePaper_CIO_v6.pdf)>

MySQL.com.”*MySQL :: Developer Zone*”. In: MySQL.com, 04/abr/2013. Disponível em <<http://dev.mysql.com>>.

Netcraft. “January 2013 Web Server Survey | Netcraft” In: Netcraft, 07/abr/2013. Disponível em <<http://news.netcraft.com/archives/2013/01/07/january-2013-web-server-survey-2.html>>.

phpMyAdmin. “phpMyAdmin - About” In: phpmyadmin, 08/abr/2013. Disponível em <[http://www.phpmyadmin.net/home\\_page/about.php](http://www.phpmyadmin.net/home_page/about.php)>.

SUEHRING, Steve. Mysql – A Bíblia. São Paulo, Ed Campus, 2002.

USP (Universidade de São Paulo) . “Usp em números”. In: USP, 04/jun/2013. Disponível em < <http://www5.usp.br/usp-em-numeros>>.

W3C (World Wide Web Consortium), “Cascading Style Sheets” In: W3C, 2013. Disponível em <<http://www.w3.org/Style/CSS/>>.

w3Schools. Learn HTML and CSS with w3Schools. Hoboken, Ed. Wiley, 2010.

W3techs, “Usage of JavaScript libraries for websites” In: W3techs, 05/jun/2013. Disponível em < [http://w3techs.com/technologies/overview/javascript\\_library/all](http://w3techs.com/technologies/overview/javascript_library/all)>.