

**GUILHERME NUNES FONTANS**

**DESENVOLVIMENTO DE SISTEMA *WEB* PARA MICROEMPRESAS**

**Porto Alegre – RS**

**2016**

**ESCOLAS E FACULDADES QI**

**CURSO DE ANÁLISE E DESENVOLVIMENTO DE SISTEMAS**

**GUILHERME NUNES FONTANS**

**DESENVOLVIMENTO DE SISTEMA *WEB* PARA MICROEMPRESAS**

Projeto de Pesquisa apresentado ao Curso de ANÁLISE E DESENVOLVIMENTO DE SISTEMAS das Escolas e Faculdades QI, como requisito para elaboração do trabalho de conclusão de curso.

Orientador: Prof. Thiago Minuzzi

**Porto Alegre – RS**

**2016**

**Lista de Figuras**

[Figura 1: Servidores rodando nos sites mais visitados. 6](#_Toc461023361)

[Figura 2: Conexão com banco de dados utilizando PDO 9](#_Toc461023362)

Lista de Abreviaturas e Siglas

API: Application Programming Interface, 10

CGI: Common Gateway Interface, 6

FastCGI: Fast Common Gateway Interface, 6

FI: Forms Interpreter, 6

IDE: Integrated Development Environment, 5

OVA: Open Virtual Appliance, 5

PDO: PHP Data Objects, 8

PHP: PHP Hypertext Preprocessor, 7

PHP-FIG: PHP Framework Interop Group, 8

RAM: Randon Acess Memory, 5

SGBD: Sistema de Gerenciamento de Banco de Dados, 5

SQL: Structured Query Language, 8

VIM: Vi Improvement, 5

Sumário

[1 Introdução 3](#_Toc460968630)

[2 Elementos da pesquisa 3](#_Toc460968631)

[2.1 Problema 3](#_Toc460968632)

[2.2 Justificativa 3](#_Toc460968633)

[2.3 Hipóteses 4](#_Toc460968634)

[2.4 Objetivos 4](#_Toc460968635)

[2.4.1 Objetivo geral 4](#_Toc460968636)

[2.4.2 Objetivos específicos 4](#_Toc460968637)

[3 Referencial teórico 4](#_Toc460968638)

[3.1 Editor de texto 4](#_Toc460968639)

[3.2 Servidor WEB 4](#_Toc460968640)

[3.3 PHP 5](#_Toc460968641)

[3.4 SGBD 7](#_Toc460968642)

[3.5 Repositório 7](#_Toc460968643)

# **Introdução**

# **Elementos da pesquisa**

## **Problema**

É possível desenvolver um software em ambiente web de fácil instalação para apoiar os gestores de uma empresa para obter novas ideias ou sugestão de melhoria de seus funcionários e fazer com que os funcionários acompanhem o status da sua ideia?

## **Justificativa**

Devido ao impacto ocasionado pela crise mundial econômica no ano de 2009, muitas empresas tiveram uma grande perda de seus clientes e como consequência, as mesmas que não estavam preparadas economicamente para essa perda, vieram a falir.

Mas e as empresas que se mantiveram abertas? Como fazer para manter os clientes atuais e conquistar novos clientes, ou até mesmo manter seus gastos?

Muitas empresas de porte maior tem um setor específico apenas para levantar ideias, porém empresas pequenas não tem a disposição de espaço e nem pessoas para isso.

## **Hipóteses**

- Sim, é possível efetuar o desenvolvimento de um software para gerenciamento de ideias.

- Não, não é possível.

## **Objetivos**

### **Objetivo geral**

Desenvolver um sistema para apoiar as empresas no armazenamento de ideias internas, no qual o mesmo permita o cadastro de prêmios (folgas, bonificações em dinheiro) por parte do administrator, nos quais podem ser trocados por pontos, que serão atribuídos conforme status da ideia apresentada pelo funcionário.

### **Objetivos específicos**

- Fazer a análise da linguagem utilizada e o diagrama de classes.

- Escolha do SGBD que será utilizado no sistema e Modelagem do banco.

- Efetuar a análise da distribuição do sistema (OVA).

# **Referencial teórico**

Atualmente, há um grande número de usuários e de compradores na Internet, segundo dados do Ibope Nielsen, em 2012, mais de 90 milhões de brasileiros estão conectados na *web.*

## **Editor de texto**

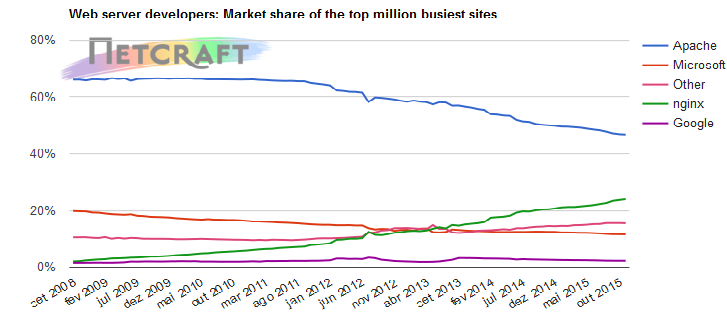
A ferramenta que será utilizada para desenvolvimento será o editor de texto VIM. Pois de acordo com o site oficial, o vim é um editor de texto altamente configurável construído para fazer criação e alteração de qualquer tipo de texto muito eficiente. Por se tratar de ter uma grande possibilidade de customizações e se tratar de um editor de texto simples, o vim atende a todos os requisitos de uma IDE, porém com um uso muito menor de memória RAM.

## **Servidor WEB**

Ao contrário de servidores tradicionais, Nginx não depende de threads para manipular as solicitações. Em vez disso, usa uma arquitetura muito mais escalável orientada a eventos (assíncrono). Esta arquitetura utiliza pequeno, mas mais importante ainda, quantidades previsíveis de memória sob carga (SITE OFICIAL DO NGINX, 2016).

Segundo pesquisa realizada pela em novembro de 2015 pela Netcraft, o NGINX tem hoje mais de 16% do mercado de servidores WEB, ficando atrás apenas do Apache, porém o percentual de servidores rodando o NGINX está crescendo cada vez mais rápido, a figura 1 demonstra tal crescimento entre os sites mais visitados.

Figura : Servidores rodando nos sites mais visitados.



Dentre as vantagens do NGINX, temos o FastCGI, recurso no qual pode ser utilizado em conjunto com a linguagem selecionada para efetuar o desenvolvimento da aplicação.

O principal objetivo da FastCGI é reduzir a sobrecarga associada à interface dos programas de servidor web e CGI , permitindo que um servidor para lidar com mais solicitações de páginas web ao mesmo tempo.(PIERRO, 2013, pg 533).

Essa ferramenta se adapta aos requisitos do projeto e será utilizada no desenvolvimento.

## **PHP**

O PHP como é conhecido hoje, é na verdade o sucessor para um produto chamado PHP/FI. Criado em 1994 por Rasmus Lerdof, a primeira encarnação do PHP foi um simples conjunto de binários Common Gateway Interface (CGI) escrito em linguagem de programação C. Originalmente usado para acompanhamento de visitas para seu currículo online, ele nomeou o conjunto de scripts de "Personal Home Page Tools" mais frequentemente referenciado como "PHP Tools. Em Junho de 1995, Rasmus liberou o código fonte do PHP Tools para o público, o que permitiu que desenvolvedores usassem da forma como desejassem. Isso permitiu - e encorajou - usuários a fornecerem correções para bugs no código, e em geral, aperfeiçoá-lo." (SITE OFICIAL DO PHP, 2016).

Apesar de ter tido um curto período de desenvolvimento, ele continuava a desfrutar uma crescente popularidade em um ainda jovem mundo web desenvolvimento, Em 1997 e 1998, PHP/FI teve o apoio de milhares de usuários ao redor do mundo. Uma pesquisa Netcraft de Maio de 1998 indicou que cerca de 60.000 domínios relataram ter cabeçalhos contendo "PHP", indicando que o servidor de hospedagem de fato tinha o PHP instalado. Este número pode ser equiparado com aproximadamente 1% de todos os domínios da Internet da época (SITE OFICIAL DO PHP, 2016).

PHP 3.0 foi a primeira versão que se assemelha com o PHP como existe hoje. PHP/FI se encontrava ainda ineficiente e não tinha recursos que precisava para prover uma aplicação eCommerce que estavam desenvolvendo para um projeto da Universidade, Andi Gutmans e Zeev Suraski de Tel Aviv, Israel, começaram outra completa reescrita do interpretador em 1997. Abordando Rasmus online, eles discutiram vários aspectos para a corrente implementação e redesenvolvimento do PHP. Em um esforço para melhorar a engine e iniciar a construção em cima da base de usuários existentes do PHP/FI, Andi, Rasmus, e Zeev decidiram colaborar no desenvolvimento de uma nova e independente linguagem de programação. Essa nova linguagem foi lançada com um novo nome, que removeu a impressão do limitado uso pessoal que o nome PHP/FI 2.0 tinha mantido. Foi renomeado simplesmente para 'PHP', com o significado se tornando um acrônimo recursivo - PHP: Hypertext Preprocessor (SITE OFICIAL DO PHP, 2016).

No inverno de 1998, logo após o PHP 3.0 ter sido oficialmente lançado, Andi Gutmans e Zeev Suraski começaram a trabalhar em uma reescrita do core do PHP. Os objetivos do projeto eram melhorar performance das aplicações complexas, e melhorar a modularização do código base do PHP (SITE OFICIAL DO PHP, 2016).

O PHP 5 foi lançado em Julho de 2004 após um longo desenvolvimento e vários pré-lançamentos. O time de desenvolvimento PHP inclui dezenas de desenvolvedores, também dezenas de outros trabalhando em algo relacionado ao PHP e apoio a projetos (SITE OFICIAL DO PHP, 2016).

Olhando para a sua história, é possível verificar como a linguagem cresceu nos últimos anos, de acordo com pesquisa do site da w3tchs, o PHP é usado hoje por 82% de todos os sites cuja linguagem de programação do lado do servidor é conhecida (W3TECHS, 2016), um crescimento notável em relação ao 1% de 1998, entre famosas empresas que utilizam a linguagem, encontram-se: Facebook, Yahoo, Wikipedia e Wordpress.

Mas não é só a linguagem que torna uma tecnologia/plataforma bem sucedida, o ecossistema ao redor dela também faz muito a diferença (MEDRADO, 2015). Além de ser hoje a linguagem mais utilizada para servidores web no mundo, o PHP é uma linguagem de fácil aprendizagem e também possui um grande ecossistema em sua volta, oque vem a facilitar o trabalho com a linguagem, como uma comunidade colaborativa, muitos eventos ao redor do país, possui um bom gerenciador de dependências, inúmeros frameworks e componentes desenvolvidos por usuários e disponibilizados livremente, podendo ser utilizados em paralelos com outros componentes, graças ao PHP-FIG.

O PHP-FIG é um grupo de representantes de frameworks PHP que, de acordo com seu site, “fala sobre o que há de comum entre seus projetos e descobre maneiras de trabalhar juntos”. O PHP-FIG cria recomendações que os frameworks PHP podem implementar voluntariamente para melhorar a comunicação e o compartilhamento com outros frameworks (LOCKHART, 2015, pg 63).

O PHP-FIG é um grupo autonomeado de representantes de frameworks. Seus membros não são eleitos, e nem são especiais em nenhum aspecto que não seja a sua disposição em proporcionar melhorias a comunidade PHP. Qualquer pessoa pode pedir para ser um membro desse grupo, e qualquer pessoa pode submeter feedbacks às recomendações do PHP-FIG que estejam em processo de proposição. As recomendações finais do PHP-FIG normalmente são adotadas e implementadas por vários dos maiores e mais populares frameworks (LOCKHART, 2015, pg 63).

Para o desenvolvimento do projeto, será utilizado o PHP como linguagem de programação, tendo em vista que a mesma proporciona liberdade para a escolha de seus componentes, e a garantia que esses componentes utilizam o padrão de escrita de códigos propostos pela comunidade.

A linguagem também disponibiliza a extensão PDO para efetuar a conexão, assim como controle de transações e oferecendo uma implementação, em tese, segura contra injeções de SQL por meio do uso de prepared statements na consulta à base de dados. A extensão PHP Data Objects define uma interface leve e consistente para acessar bancos de dados em PHP. PDO fornece uma camada de abstração de acesso a dados, o que significa que, independentemente de qual banco de dados você está usando, você usa as mesmas funções para emitir consultas e buscar dados (SITE OFICIAL DO PHP, 2016).

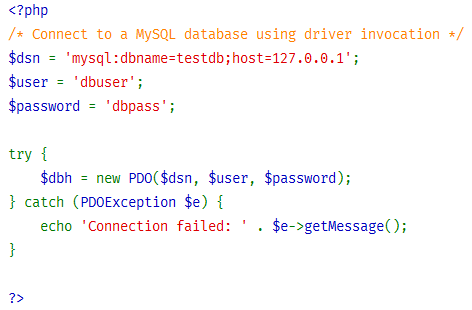
## **SGBD**

Um banco de dados é uma coleção organizada de dados. Existem muitas estratégias diferentes para organizar dados a fim de facilitar o acesso e a manipulação, e um sistema de gerenciamento de banco de dados (SGBD) oferece mecanismos para armazená-los, organizá-los, recuperá-los e modifica-los. Tais sistemas proveem acesso e armazenamento de dados sem preocupação com a representação interna dos dados no respectivo banco (NAVATHE, 2005, pg 446).

Para o desenvolvimento do sistema, se fara necessário utilizar um SGBD, tendo em vista que os dados que serão colocados no mesmo precisarão ser consultados posteriormente, e também poderão vir a sofrer mudanças.

Dentre vários sistemas de gerência de banco de dados, encontram-se o SQLServer, Oracle, PostgreeSQL, SQLite e MySQL, ficando o último em destaque entre os SGBDS que atuam junto com a linguagem PHP, no qual a própria linguagem já disponibiliza funções para trabalhar em conjunto, como a API PDO explicado na seção anterior. Na figura 2 é demonstrado como é feita a conexão no SGBD utilizando PDO.

Figura : Conexão com banco de dados utilizando PDO



Esse SGBD surgiu em 1996, como um projeto interno da empresa TcX DataKonsult AB, na Suécia. Rapidamente, tornou-se muito popular, pois embora não tivesse alguns recursos que outros SGBDs ofereciam, primava pela escalabilidade e bom desempenho, qualidades consideradas mais importantes por muitos. Seu crescimento fez com que fosse adquirido pela Sun Microsystems que foi, pouco depois, comprada pela Oracle Corporation (GILMORE, 2010, pg 477).

Com o intuito de garantir o código aberto, o próprio fundador do MySQL criou outro banco de dados chamado MariaDB, no qual surgiu como um [fork](https://pt.wikipedia.org/wiki/Fork) do [MySQL](https://pt.wikipedia.org/wiki/MySQL) após sua aquisição pela [Oracle](https://pt.wikipedia.org/wiki/Oracle) (MARIADB, 2016). De acordo com o site oficial, o MariaDB é mantido até à data corrente com a última versão do MySQL a partir do mesmo ramo e na maioria dos aspectos MariaDB vai funcionar exatamente como MySQL. Todos os comandos, interfaces, bibliotecas e APIs que existem no MySQL também existem no MariaDB.

Tendo em vista que o desenvolvedor já tem familiaridade com o banco de dados em questão e reconhece a sua performance em relação ao seu antecessor, o SGBD utilizado para armazenar os dados do sistema será o MariaDB. Em seu site oficial, o SGBD também menciona que o MariaDB é usado porque ele é rápido, escalável e robusto, com um rico ecossistema de mecanismos de armazenamento, plugins e muitas outras ferramentas que o tornam muito versátil para uma ampla variedade de casos de uso.

## **Repositório**

O controle de versão é um dos fatores de extrema importância no desenvolvimento de um software, pois geralmente são feitas várias alterações para que se alcance o resultado esperado e consequentemente a satisfação do cliente e dos usuários do mesmo. Por esta razão é comum o uso de sistemas desenvolvidos com o proposito de registrar as várias versões que um sistema pode ter, versões estas que vão deste o inicio de sua construção até a entrega final do software (PRESMAN, 2006).

Dentre os sistemas de versionamento existentes no mercado, podemos citar o git, que de acordo com o seu site oficial, começou com um tanto de destruição criativa e controvérsia acirrada. O kernel (núcleo) do Linux é um projeto de software de código aberto de escopo razoavelmente grande. Durante a maior parte do período de manutenção do kernel do Linux (1991-2002), as mudanças no software eram repassadas como patches e arquivos compactados. Em 2002, o projeto do kernel do Linux começou a usar um sistema DVCS proprietário chamado BitKeeper.

Em 2005, o relacionamento entre a comunidade que desenvolvia o kernel e a empresa que desenvolvia comercialmente o BitKeeper se desfez, e o status de isento-de-pagamento da ferramenta foi revogado. Isso levou a comunidade de desenvolvedores do Linux (em particular Linus Torvalds, o criador do Linux) a desenvolver sua própria ferramenta baseada nas lições que eles aprenderam ao usar o BitKeeper. Alguns dos objetivos do novo sistema eram (SITE OFICIAL DO GIT):

* Velocidade
* Design simples
* Suporte robusto a desenvolvimento não linear (milhares de branches paralelos)
* Totalmente distribuído
* Capaz de lidar eficientemente com grandes projetos como o kernel do Linux (velocidade e volume de dados)

Desde sua concepção em 2005, o Git evoluiu e amadureceu a ponto de ser um sistema fácil de usar e ainda assim mantém essas qualidades iniciais. É incrivelmente rápido, bastante eficiente com grandes projetos e possui um sistema impressionante de branching para desenvolvimento não-linear (SITE OFICIAL DO GIT).

Será utilizado o Git como ferramenta de controle de versão, no qual o seu repositório do código fonte e também da distribuição do formato OVA estará hospedado no site https://github.com/guilhermefontans/banco-de-ideias.

NAVATHE, Ramez Elmasri Shamkant B. 2005, Sistemas de Banco de Dados

GILMORE, W. Jason. PHP and MySQL: from novice to professional. 4 ed. New York:

Apr, 2010.

LOCKHART, Josh. PHP Moderno. Ed. Novatec: 2015.

MARIADB. Disponível em: <http://en.wikipedia.org/wiki/MariaDB>. Acesso em: 10 abr. 2014.

NGINX. Disponível em: <https://nginx.org/en/> . Acesso em 07 set. 2016.

PIERRO, Massimo Di. Web2Py. Ed Lulu Press Inc.