

**GUILHERME NUNES FONTANS**

**SISTEMA *WEB* PARA MANTER INOVAÇÔES DE IDEIAS INTERNAS**

**Porto Alegre – RS**

**2016**

**ESCOLAS E FACULDADES QI**

**CURSO DE ANÁLISE E DESENVOLVIMENTO DE SISTEMAS**

**GUILHERME NUNES FONTANS**

**SISTEMA *WEB* PARA MANTER INOVAÇÔES DE IDEIAS INTERNAS**

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.

Professor: Tiago Minuzzi

**Porto Alegre – RS**

**2016**

# **Lista de Figuras**

[Figura 1: Servidores rodando nos sites mais visitados 11](#_Toc465982461)

[Figura 2: Conexão com banco de dados utilizando PDO 14](#_Toc465982462)

[Figura 3: Caso de uso cadastrar ideia 20](#_Toc465982463)

[Figura 4: Modelo ER 20](#_Toc465982464)

[Figura 5: Formulário de cadastro de área 22](#_Toc465982465)

[Figura 6: Mensagem de confirmação de cadastro de área 22](#_Toc465982466)

[Figura 7: Formulário de cadastro de usuário 23](#_Toc465982467)

[Figura 8: Menu lateral 24](#_Toc465982468)

[Figura 9: Menu superior com dados do usuário 24](#_Toc465982469)

[Figura 10: Lista de prêmios disponíveis para o usuário 25](#_Toc465982470)

**Lista de Abreviaturas e Siglas**

API: Application Programming Interface

CGI: Common Gateway Interface

DAO: Data Access Object

ER: Entidade Relacionamento

FastCGI: Fast Common Gateway Interface

FI: Forms Interpreter

GNU: Gnu's Not Unix

MVC: Model View Controller

OVA: Open Virtual Appliance

PDO: PHP Data Objects

PHP: PHP Hypertext Preprocessor

PHP-FIG: PHP Framework Interop Group

SGBD: Sistema de Gerenciamento de Banco de Dados

SO: Sistema Operacional

SQL: Structured Query Language

UML: Unified Modeling Language

VI: Visual Interface

# **Sumário**

[**Lista de Figuras** 3](#_Toc465982876)

[**Sumário** 5](#_Toc465982877)

[**1** **Introdução** 6](#_Toc465982878)

[**2** **Elementos da pesquisa** 7](#_Toc465982879)

[**2.1** **Problema** 7](#_Toc465982880)

[**2.2** **Justificativa** 7](#_Toc465982881)

[**2.3** **Hipóteses** 7](#_Toc465982882)

[**2.4** **Objetivos** 8](#_Toc465982883)

[**2.4.1** **Objetivo geral** 8](#_Toc465982884)

[**2.4.2** **Objetivos específicos** 8](#_Toc465982885)

[**3** **Referencial teórico** 8](#_Toc465982886)

[**3.1** **Editor de texto** 9](#_Toc465982887)

[**3.2** **Servidor WEB** 10](#_Toc465982888)

[**3.3** **PHP** 11](#_Toc465982889)

[**3.4** **SGBD** 14](#_Toc465982890)

[**3.5** **Repositório** 15](#_Toc465982891)

[**3.6** **Virtualização** 16](#_Toc465982892)

[**4** **Metodologia.** 17](#_Toc465982893)

[**5** **Banco de ideias.** 19](#_Toc465982894)

[**5.1** **Requisitos.** 19](#_Toc465982895)

[**5.2** **Modelo de Banco de Dados** 20](#_Toc465982896)

[**5.3** **Uso da ferramenta.** 21](#_Toc465982897)

[**6** **Conclusão** 25](#_Toc465982898)

[**7** **Trabalhos Futuros** 26](#_Toc465982899)

[**Referências bibliográficas** 27](#_Toc465982900)

[**Apêndices** 29](#_Toc465982901)

# **Introdução**

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 que não estavam preparadas economicamente para essa perda, vieram a falir.

Felizmente muitas companhias souberam como inovar a sua maneira de prestar seus serviços, ou vender os seus produtos, garantindo assim uma sobrevivência econômica em meio a momentos de crise.

Para que isso viesse a ocorrer, em dado momento os donos, gestores, funcionários de cargos de níveis gerenciais tiveram que reunir-se para reavaliar seus processos de negócios, a fim de procurar gargalos que possam ser melhorados ou talvez pensar em novos produtos, conseguindo de essa forma manter a lucratividade.

Muitas vezes esses gargalos e possíveis produtos que podem ser criados não são identificados apenas pela gerência da empresa, e sim pelos funcionários que ocupam cargos mais operacionais, isso é devido aos mesmos estarem mais próximos dos processos de produção do que a sua gerência.

Com o avanço da tecnologia, é possível se obter computadores com um hardware mais sofisticado e com um preço mais acessível, tendo isso em vista, consegue-se utilizar um computador comum, como um desktop e poder utilizar uma máquina virtual encima desse desktop, mantendo ainda um bom desempenho com intuito de utilizar em conjunto com outros softwares.

O problema da pesquisa abordada é a possibilidade de desenvolver um sistema web o qual possibilita que funcionários cadastrem novas ideias, seja para o seu setor, ou para outros setores, ou até mesmo sugerir uma melhoria, vindo a poder contribuir para o crescimento da organização. Pois segundo Pamella Gonçalves (2013), “todos os funcionários podem ser empreendedores e precisam ter espaço para sugerir novas ideias dentro da empresa”. Para possibilitar esse ambiente dentro das empresas o sistema arquitetado contará com uma área restrita para os administradores, no qual eles poderão avaliar as ideias cadastradas e alterar o status das mesmas, e conforme a evolução, o funcionário que cadastrou a ideia ganhará pontos, os quais poderão ser trocados por prêmios previamente cadastrados pelos administradores.

A metodologia a ser utilizada no desenvolvimento do projeto é baseada em pesquisas sobre ferramentas de software livre, as quais proporcionam alto desempenho da aplicação e mantém a estrutura do projeto organizada, facilitando assim a manutenção e distribuição do mesmo. Posteriormente será distribuído um clone da máquina virtual utilizada no desenvolvimento da aplicação, podendo ser baixado e importado em um virtualizador da preferência do usuário final.

Espera-se encontrar resultados positivos, mostrando que é possível desenvolver um sistema web para que as empresas possam inovar suas ideias e proporcionar uma maior interação entre a gerencia e a operação, trazendo assim oportunidades de buscarem juntos uma maneira de manter a sua organização viva, assim como a evolução.

# **Elementos da pesquisa**

## **Problema**

É possível desenvolver um software em ambiente web, de fácil instalação, para apoiar os gestores de uma empresa na captação de novas ideias ou sugestões de melhoria de seus colaboradores? Bem como fazer com que os funcionários acompanhem o status da sua ideia?

## **Justificativa**

Com o impacto ocasionado no ano de 2009, pela crise mundial econômica, houve um grande número de empresas que chegaram à falência, isso devido a grande perda de seus clientes e falta de preparação econômica para sobreviver.

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.

- Realizar a análise da distribuição do sistema pronto para ser virtualizado.

- Desenvolver a implementação do sistema.

# **Referencial teórico**

“O empreendedor é aquele que vive alguns anos como ninguém quer para passar muitos anos como ninguém pode”. Essa frase foi dita por Antônio Salvador, presidente do Comitê de Gestão de Pessoas, que aconteceu no dia 26/07/2013, para explicar a essência do empreendedorismo, que é investir em uma ideia, apesar de perdas iniciais, e colher bons resultados no futuro (AMCHAM, 2013).

Atualmente algumas empresas realizam reuniões em períodos determinados, no qual podem ouvir novas ideias por parte de seus funcionários, obtendo melhorias dentro de determinada área da empresa, cabendo ao empreendedor dar ouvidos a essas ideias ou não.

Outros tipos de sugestões também seriam ideias que possam trazer inovações, vindo a gerar novos produtos para serem comercializados, ou simplesmente para adoção interna, premiando os autores dessas ideias, como foi o caso da empresa Infraero em seu concurso “Ideia em ação”, no qual ocorre desde 2000. Para João Jordão, Diretor de Gestão Operacional e Navegação Aérea da Infraero, os trabalhos desenvolvidos e premiados pelo projeto "Ideia em Ação" ajudam a demonstrar a capacidade de gestão da Infraero, além de revelar talentos entre os empregados. “Os trabalhos atestam que a empresa tem pessoas capacitadas e que podem desenvolver novos produtos e serviços e aplicá-los em prol das atividades aeroportuárias", pontuou o diretor (INFRAERO, 2010).

Tal motivação não precisa estar presente apenas em empresas grandes, pode-se desenvolver um software de apoio ao pequeno empresário o qual pode vir a obter novas ideias de seus funcionários, motivando-os e ouvindo sugestões de quem possui contato direto com o serviço, já que muitas vezes os mesmos acabam entendendo melhor do negócio do que o próprio gestor. Segundo Mamona, 2012: “Um líder sábio reconhece que são seus funcionários que produzem lucros. Um gerente brilhante deve ficar em recompensar os seus funcionários que contribuem para o bem-estar de uma organização”.

Seguem abaixo os recursos utilizados para o desenvolvimento do software o qual irá servir de apoio para cadastro de novas ideias e das premiações que podem ser obtidas.

## **Editor de texto**

Para efetuar o desenvolvimento do software, seria necessário utilizar um editor de texto para programar o código fonte, tendo em vista a sua simplicidade e capacidade de ganhar mais produtividade trabalhando alternadamente com os modos de operação disponíveis. O editor de texto que se enquadra nesses requisitos é o VIM, sendo por isso empregado a utilização deste no desenvolvimento.

O VIM é um editor de texto extremamente configurável, criado para permitir a edição de forma eficiente, tornando-o produtiva e confortável. Também é uma aprimoração do editor VI, um tradicional programa dos sistemas Unix. Ocorre que o mesmo possui uma série de mudanças em relação a esse último. O próprio slogan do VIM é Vi IMproved, ou seja, Vi Melhorado. O Vim é tão conhecido e respeitado entre programadores, e tão útil para programação, que muitos o consideram uma verdadeira IDE (VIM, 2016).

Capaz de reconhecer mais de 500 sintaxes de linguagens de programação e marcação, é possível efetuar mapeamento de teclas de atalhos para execução de comandos, macros, abreviações, busca por Expressões Regulares, entre outras facilidades. Também conta com uma comunidade bastante atuante, sempre disposta a criar plug-ins para facilitar o desenvolvimento com o editor, mantendo assim o VIM, ao lado do Emacs, outro editor de texto, um dos editores mais usados nos sistemas GNU/Linux, embora esteja também disponível em outros sistemas, como o Windows e o Macintosh.

O VIM trabalha com quatro modos de operação, os quais são os seguintes:

1) Modo normal: Modo padrão quando entra no VIM, utilizado para executar comandos de manipulação de texto. Quando estiver em qualquer outro modo, pode-se usar a tecla ‘esc’ para voltar a este modo;

2) Modo de inserção: Quando de fato, será inserido texto no arquivo. Para acessar esse modo, basta pressionar a tecla ‘i’;

3) Modo visual: Serve, principalmente, para seleção de grandes blocos de texto. Para acessar este modo, basta pressionar a tecla ‘v’, porém é necessário previamente estar no modo normal;

4) Modo de comandos: Onde é executa alguns comandos no VIM ou externos, é possível configurar o editor em tempo real, sair do editor, salvar arquivos (OLIVEIRAS, 2016).

Os diferentes modos de operação do VIM o tornam mais produtivo, pois existe facilidade de se realizar operações de modificação. Desde de uma simples alteração de palavras no código através do modo de inserção, ou ainda alterações maiores através de comandos que possibilitam a utilização de expressões regulares para localizar e alterar trechos do código são suportadas pelo editor VIM.

Outra vantagem do VIM, é que ele possui um arquivo de configuração chamado vimrc, no qual pode ser incluídos plug-ins para personalizar o mesmo e salvar configurações como mudança de temas, estilo de tabulação, entre outras funcionalidades.

Plugins são um meio de estender as funcionalidades do Vim, há plugins para diversas tarefas, desde wikis para o Vim até ferramentas de auxílio à navegação em arquivos como é o caso do plugin NerdTree, que divide- se em uma janela e permite navegar pelos diretórios do sistema a fim de abrir arquivos a serem editados (BOTARO, 2015).

## **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 pouca memoria, e mais importante ainda, quantidades previsíveis dessa memória sob carga (SITE OFICIAL DO NGINX, 2016).

Segundo pesquisa realizada 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, isso pode-se perceber com a figura 1, a qual demonstra o crescimento entre os sites mais visitados.

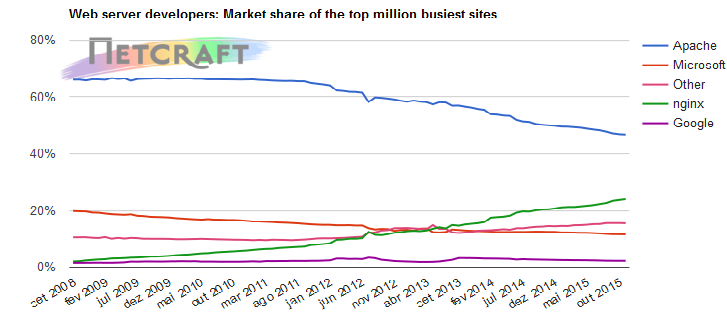


Figura 1: Servidores rodando nos sites mais visitados

Fonte: https://news.netcraft.com

Dentre as vantagens do NGINX, temos o FastCGI, recurso que 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 possa 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, permitindo assim que desenvolvedores usassem da forma como desejassem. Isso encorajou os 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 em desenvolvimento. Nos anos de 1997 e 1998, PHP/FI teve o apoio de milhares de usuários de vários países. 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).

O PHP 3.0 foi a primeira versão que se assemelha com o PHP como existe atualmente. Anterioromente a isso, o PHP/FI se encontrava ainda ineficiente e não tinha os recursos para prover uma aplicação de um eCommerce para uma universidade, no qual Andi Gutmans e Zeev Suraski de Tel Aviv, Israel, estavam trabalhando. Então eles começaram outra completa reescrita do interpretador em 1997. Em contato online com Rasmus, criador do PHP/FI, eles discutiram vários aspectos para a corrente implementação e redesenvolvimento da linguagem (SITE OFICIAL DO PHP, 2016).

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 era melhorar o desempenho das aplicações complexas, bem como 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, assim como outros colaboradores trabalhando em algo relacionado ao PHP e apoio a projetos (SITE OFICIAL DO PHP, 2016).

Olhando para a história do PHP, 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, estando presente em grandes empresas famosas, como é o caso do Facebook, Yahoo, Wikipedia e Wordpress.

Porém 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 de fácil aprendizagem e também possui um grande ecossistema em sua volta, oque vem a facilitar o trabalho com o mesmo. Ele possui uma comunidade colaborativa e com muitos eventos ao redor do país, o que ajuda na divulgação de novos recursos e de novas solução que utilizam a linguagem, como também ganhar novos colaboradores para sua continua evolução.

O PHP também possui um bom gerenciador de dependências, bem como inúmeros frameworks e componentes que foram desenvolvidos por usuários e disponibilizados livremente, e os mesmos podem 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).

Os membros do PHP-FIG é 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, submetendo-se a feedbacks às recomendações do PHP-FIG que estejam em processo de proposição. As recomendações finais dele normalmente são adotadas e implementadas pelos 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, no qual a mesma possui controle de transações e oferece 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. A PDO não é uma biblioteca completa para abstração do acesso a base de dados, uma vez que ela não faz leitura e tradução de instruções SQL, adaptando-as aos mais diversos drivers de banco de dados existentes. Ela simplesmente unifica a chamada de métodos, passando as suas extensões correspondentes e faz uso do que existe de mais recente no que diz respeito à orientação a objetos presente no PHP (OGLIO, 2009, pg 159).

## **SGBD**

Um banco de dados é uma coleção organizada de dados. Existem muitas estratégias diferentes para organizá-los, a fim de facilitar o acesso e a manipulação. Um sistema de gerenciamento de banco de dados (SGBD) oferece mecanismos para armazená-los, organizá-los, recuperá-los, bem como modifica-los. Tais sistemas proveem acesso e armazenamento de informações 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, 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. Segue abaixo a figura 2 a qual demostra como é feita a conexão no SGBD utilizando PDO.

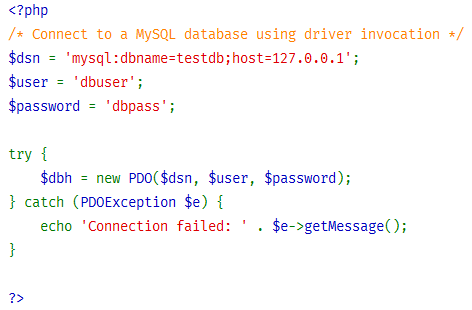


Figura 2: Conexão com banco de dados utilizando PDO

Fonte: php.net

Na figura acima, os valores necessários para efetuar a conexão com o SGBD estão sendo armazenadas em variáveis, que posteriormente são utilizadas na instanciação da classe PDO, caso ocorra algum erro, será lançada uma exceção do tipo PDOException.

O SGBD MySQL 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 sendo posteriormente 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, o 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 o seu desempenho 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 devido sua rapidez, 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, assim como 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 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, já que assim mantém 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).

O GIT será utilizado como o sistema de versionamento durante o desenvolvimento do projeto, tenho como repositório o github.

## **Virtualização**

A virtualização seja ela de serviços, aplicativos ou de servidores hoje já deixou de ser uma tendência para ser uma realidade em muitos setores, seja dentro ou fora da área da tecnologia. Pode-se definir virtualização como uma forma de se executar vários serviços, programas, ou até mesmo sistemas operacionais em um único equipamento físico. A virtualização possibilita inclusive simular hardwares diferentes em um único equipamento, como roteadores, switchs, servidores, celulares, etc. (DEVEL SISTEMAS).

A virtualização de servidores funciona com o uso de um software o qual irá simular parcial ou completamente o hardware em que será executado um sistema operacional, não necessariamente o mesmo do sistema hospedeiro, é essa a grande vantagem da virtualização, simular um hardware que não se tem nativamente (DEVEL SISTEMAS).

Tendo em vista que o foco da distribuição do software é ter uma instalação simples, optou-se por distribuir a própria máquina virtual que será desenvolvida e aplicada, tendo em vista que a mesma já terá todas as suas dependências contempladas, como o MariaDB, NGINX, PHP. Outro fato é que está máquina virtual já irá rodar em um sistema operacional configurado para suportar todos os requisitos para a aplicação se manter estável, evitando assim problemas como o do MariaDB ou o servidor WEB estarem inoperantes.

Além disso, no que diz respeito a Segurança dos dados, este tipo de distribuição permite utilizar um agendador de tarefas do sistema operacional para executar o backup da base de dados e disponibilizá-lo via SAMBA na rede em que o servidor estiver operando.

Para poder fazer essa distribuição da máquina virtual em que o sistema será desenvolvido, será exportado a mesma, e efetuado a distribuição da sua exportação, no qual o formato do arquivo é na extensão OVA, bastando apenas importar este arquivo em um virtualizador, como VMWare, ou VirtualBox.

# **Metodologia.**

Após a identificação do problema, e analisou-se casos de empresas que já obtiveram resultados positivos obtendo ideias através de seus funcionários, chegou-se a percepção de que essa metodologia traz benefícios tanto para o lado empresa quanto para o lado funcionário através do recebimento de premiações.

Para melhorar a administração desse processo, para fins de que tanto funcionários como gestores possam acompanhar o status dessas ideias sugeridas, foi desenvolvido um software, no qual os usuários comuns podem cadastrar novas sugestões no banco de ideias, e os administradores do sistema podem alterar o status das mesmas e distribuir pontos conforme o status, pontos esses que poderão vir a serem trocados por prêmios.

Tendo definido o problema e uma solução, foi realizada uma pesquisa com intuito de identificar quais as ferramentas que poderiam ser utilizadas para o desenvolvimento do projeto, tendo como foco ferramentas atuais, livres assim como o sistema operacional o qual será distribuído. Além disso, na pesquisa realizada, as ferramentas foram analisadas também por suas características de desempenho para que possibilitem uma boa performance do sistema.

- PHP 7.0.10: Como linguagem de programação, utilizando o FastCGI, podendo assim ganhar mais performance.

- Vim 7.4: Editor de texto que será adaptado com plug-ins e customizações para ter uma experiência mais produtiva no desenvolvimento do software com a linguagem escolhida.

- Nginx 1.10.1: Servidor web capaz de trabalhar em conjunto com o recurso FastCGI utilizado pela linguagem.

- MariaDB 5.5.50: SGBD sucessor do MySQL.

- Git 1.8.3.1: Utilizado para realizar o controle de versionamento durante o desenvolvimento.

Para começar o desenvolvimento, foi necessário realizar o levantamento de requisitos, ou seja, primeiro foi realizada a análise detalhada das regras de negócio e das necessidades do sistema, verificando assim o que era esperado que o sistema fizesse para garantir que o desenvolvimento ocorresse da forma correta e de acordo com o que foi descrito nos requisitos. Com o objetivo de corroborar o entendimento cita-se Fabio Gomes Rocha (2013), “parte das falhas de projetos é devida a falhas no levantamento de requisitos”.

Após os requisitos serem verificados, foi feito o modelo ER para fins de ficar especificado como seria a estrutura do banco de dados utilizado pelo software.

Após efetuar a análise de pressupostos e a criação do modelo ER, o desenvolvimento foi realizado utilizando as ferramentas descritas na seção três deste artigo, e de acordo com os padrões de projeto MVC e DAO, garantindo que o sistema fosse feito de forma organizada e clara.

Ao final do desenvolvimento do software, foi disponibilizado no repositório de dados disponível em https://github.com/guilhermefontans/banco-de-ideias, o clone da máquina virtual utilizada para desenvolvê-lo, o qual trata-se de um arquivo no formato OVA, contendo como sistema operacional Linux com a distribuição CentOS 7. Neste caso basta efetuar o download do arquivo e importá-lo em um virtualizador da escolha do usuário final, podendo ser VirtualBox, VMWare, Hyper-vision entre outros. Posteriormente a máquina virtual vai ser importada no virtualizador, ao ligar a mesma e concluir a carga do seu SO, será apresentado o IP que será utilizado para acessar a aplicação em um navegador web.

# **Banco de ideias.**

O software Banco de ideias tem como objetivo principal, proporcionar a todos os funcionários de uma organização, oportunidade de sugerir melhorias ou inovações para gestores da sua e das demais áreas da empresa, através do seu cadastro de ideias que o software proporciona.

Para cada ideia cadastrada, o usuário receberá um ponto, no qual poderá trocar pelas premiações cadastradas, quando a ideia passar do estado “nova” para “em andamento’, o usuário receberá mais cinco pontos, para ideias que forem aprovadas, o usuário recebera dez pontos, caso a ideia seja excluída quando ainda estiver no estado inicial, o funcionário perderá o ponto ganho.

O gestor de uma área, através do seu acesso privilegiado de usuário do tipo “admin”, poderá previamente cadastrar prêmios, e os pontos necessários para poder ter o direito de obter esse prêmio.

O usuário que alcançar o número de pontos necessários de um prêmio, poderá solicitar a troca pelos seus pontos, nos quais serão descontados automaticamente da sua conta.

## **Requisitos.**

O levantamento dos requisitos de um determinado software é a tarefa mais crucial a ser realizada na sua criação, pois apesar de geralmente ser um processo longo e árduo a ser implementado pelo desenvolvedor, possui um papel fundamental, pois é através dela que surgem as demais etapas que compõem o software.

Com base nas funcionalidades básicas que um sistema deveria ter para atender a proposta do software que foi desenvolvido, foi detectado quais são os requisitos que o sistema deveria contemplar, esses requisitos podem ser vistos no Apêndice A, presente em anexo a este artigo. Diante disso identificou-se que sistema conteria as seguintes funções principais: cadastro, alteração, busca e remoção de ideias dos seguintes itens: usuários, áreas e prêmios. Possuir um tela para solicitação de prêmio, geração de relatório e um acesso através de login e senha.

Na figura 3 é apresentado através da UML, o diagrama de caso de uso do requisito cadastrar ideia.

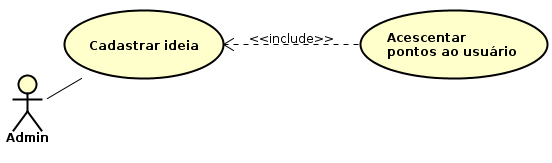


Figura 3: Caso de uso cadastrar ideia

Fonte: Autor

A figura acima demostra as etapas que ocorrem para que seja efetuado o cadastro de uma ideia no sistema, são elas: após estar conectado o usuário solicita a tela de cadastro de ideia, o sistema irá salvar a ideia com o usuário logado, e irá acrescentar os respectivos pontos pro mesmo.

## **Modelo de Banco de Dados**

Conforme mencionado anteriormente na seção 3.4, foi optado pelo autor utilizar o SGBD MariaDB para armazenar os dados que serão salvos no software Banco de Ideias. Tendo em vista o levantamento de requisitos efetuado, pode-se traçar o modelo ER da base de dados do sistema, este no qual e apresentado na figura 4.

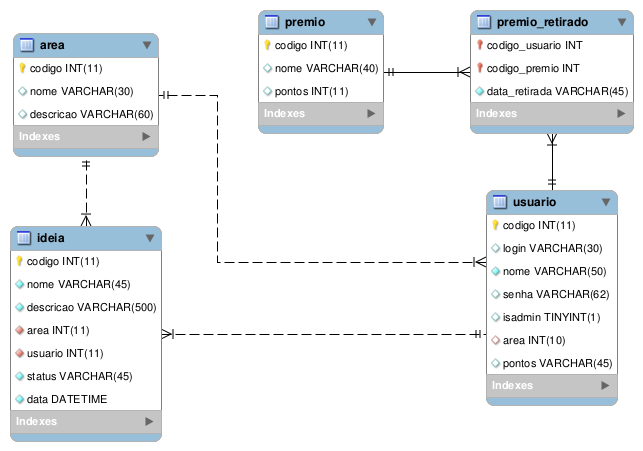


Figura 4: Modelo ER

Fonte: Autor

A imagem acima apresenta as cinco tabelas relacionadas entre si que o sistema possui, esse relacionamento foi criado com o propósito de obter as informações necessárias quando necessário. As tabelas são identificadas como:

1) Área, utilizada para informações das possíveis áreas que o usuário pode possuir;

2) Prêmio, utilizada para armazenar os prêmios disponíveis, assim como os pontos necessários para a solicitação do mesmo;

3) Premio\_retirado, tabela utilizado para geração de relatórios, para fins de ter um histórico de qual usuário retirou qual prêmio;

4) Usuário, tabela responsável por armazenar os dados do usuário, e também utilizada para efetuar o login do sistema;

5) Ideia, tabela que armazena todas ideias cadastradas pelos usuários, assim como registrar para qual área é a ideia.

A criação do Modelo ER não serviu apenas para demostrar as tabelas que compõem o banco de dados, mas também como um norte durante a etapa de construção das classes do sistema.

## **Uso da ferramenta.**

O Sistema possui inicialmente um usuário chamado ‘admin’, no qual possui permissões para cadastrar novos usuários, áreas e prêmios, sendo necessário estar logado no sistema.

Devido a um usuário estar ligado diretamente a uma área dentro da empresa, o usuário previamente cadastrado estará na área ‘Administração’, para adicionar mais áreas, é necessário clicar no botão “Novo”, situado acima da lista de áreas cadastradas. Na figura 5 é demostrado a tela de cadastro da área, na qual o usuário preenche o nome da área e adiciona uma descrição, após ocorrer a validação dos dados, o botão de “Cadastrar” será habilitado para que o usuário possa submeter os dados.



Figura 5: Formulário de cadastro de área

Fonte: Autor

Na figura 6 é apresentada a mensagem retornada para o usuário após a confirmação da criação da área.



Figura 6: Mensagem de confirmação de cadastro de área

Fonte: Autor

Após o cadastro da área, a mesma já estará disponível para associar a um usuário na tela de cadastro de usuários, conforme demostrada na figura 7, apresentando as áreas em um botão do tipo *combobox.*

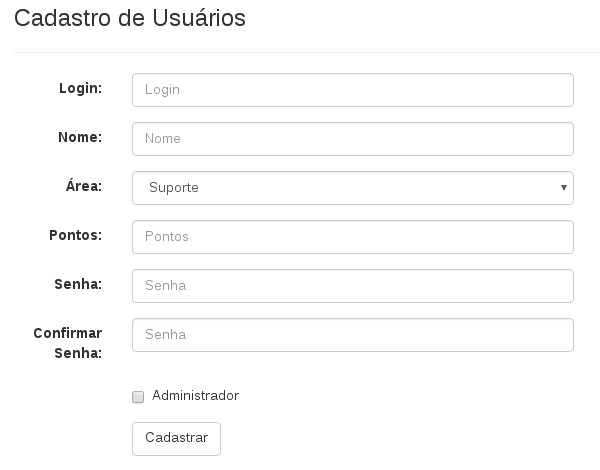


Figura 7: Formulário de cadastro de usuário

Fonte: Autor

Na figura acima, além de apresentar as áreas disponíveis para associar ao usuário, também são apresentados os campos de preenchimento obrigatório para adição do usuário, e um botão do tipo *checkbox*, para indicar se o usuário será administrador ou não.

Após a inclusão do usuário, o mesmo já estará disponível para efetuar o acesso ao sistema conforme login e senha cadastrados. Uma vez logado, o usuário estará apto a criar uma nova ideia, acessando o menu de ideias.

Na adição de ideias, o campo estado sempre estará com o valor “Novo”, podendo ser alterado apenas após a ideia já estiver cadastrada no sistema, a descrição completa de como funciona a edição de ideias está sendo apresentado no RF05, presente no apêndice A em anexo.

Indo no menu de prêmios, o usuário poderá verificar quais são os prêmios que estão cadastrados no sistema, assim como verificar quantos pontos são necessários para poder solicitar a retirada do mesmo. Os pontos do usuário ficam disponíveis no menu de usuários, localizado na área esquerda da tela do sistema, e na tela de solicitação de prêmios, também localizado na mesma área. A figura 8 demonstra a apresentação dos menus laterais do sistema, no qual o usuário ao logar no sistema se encontrara acessando o menu ‘Home’.

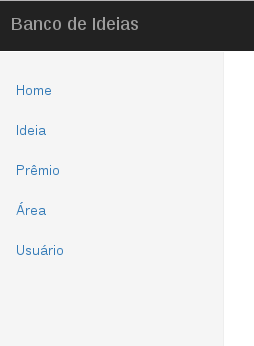


Figura 8: Menu lateral

Fonte: Autor

Com o intuito de trazer uma melhor experiência ao usuário, para fins de força-lo a procurar pela quantidade de pontos que o mesmo possui, tais pontos são apresentados ao lado do seu nome em uma barra superior que acompanha todas as telas do software, conforme demonstrado na figura 9.



Figura 9: Menu superior com dados do usuário

Fonte: Autor

Os usuários administradores do sistema poderão alterar o status das ideias cadastradas, ou excluirá caso a mesma se enquadre nos requisitos da regra de negócio, também são os usuários que poderão efetuar a adição de áreas, usuários, prêmios e ideias de todo o sistema, assim como alterar e excluir os mesmos, os demais usuários não poderão ter essas ações de manter esses tópicos, mas sim, apenas visualizar.

Após obter os pontos necessários para poder trocar por um dos prêmios cadastrados, o botão de “solicitar prêmio” estará disponível ao lado da listagem do mesmo na tabela de prêmios, podendo o usuário clicar no botão que automaticamente os pontos serão descontados do usuário, e o responsável pela área do usuário será notificado por e-mail para poder tomar as devidas providências para que o prêmio seja entregue ao usuário solicitante, a figura 10 apresenta a listagem de prêmios, quando o mesmo já está disponível para o usuário logado no sistema.



Figura 10: Lista de prêmios disponíveis para o usuário

Fonte: Autor

# **Conclusão**

O presente artigo trouxe uma possível solução como opção para empresas que buscam ouvir mais os seus funcionários, tanto para busca de inovações ou melhorias internas de seus processos, solução essa proporcionada pelo software Banco de Ideias, no qual possibilita o cadastro de sugestões por parte dos funcionários.

O sistema foi desenvolvido pelo autor em uma máquina virtual com o sistema operacional Linux CentOS 7, no qual a mesma máquina está sendo distribuída no repositório de dados GitHub, acessível em <https://github.com.br/guilhermefontans/banco-de-ideias>, na distribuição já está incluído e configurado todos os componentes utilizados para o desenvolvimento e estabilidade do sistema, como um servidor web, banco de dados e linguagem de programação instalada, assim como os demais itens citados na seção 3 do corrente artigo, e vindo a cumprir os objetivos descritos na seção 2 do mesmo.

Durante o desenvolvimento do software, pode-se observar a importância de efetuar o levantamento de requisitos, para fins de ter uma visão ampla de como o sistema deverá se comportar, e também a importância de ter uma base de dados modelada antes de começar o desenvolvimento, pois se pode obter uma direção mais precisa em relação a estrutura das classes que irão compor o sistema, porém como o software será distribuído livremente, e com o seu código aberto, é de extrema importância ter uma solução que garanta que o sistema permanecerá funcionado caso haja alguma modificação em seus fontes, solução está que serve de proposta para trabalhos futuros.

# **Trabalhos Futuros**

Posteriormente, implementar testes unitários para cada método existente nas classes do sistema, para fins de garantir que não haja quebra do sistema devido a alguma modificação feita por quem desejar modificar o código fonte do sistema, tal implementação pode ser feita através do componente PHPUnit.

Com a aplicação da solução acima, o software Banco de Ideias poderá encorajar mais os usuários que desejarem desenvolver novas *features* para o sistema, tendo em vista que na medida e que acrescentarem novas funcionalidades, poderão se certificar de que nada parou de funcionar.

# **Referências bibliográficas**

AMCHAM. **Estimulo as novas ideias entre funcionários transforma empresas.** Disponível em: < http://www.amcham.com.br/gestao-empresarial/noticias/estimulo-as-novas-ideias-entre-funcionarios-transforma-as-empresas-629.html>. Acesso em Set. 2016.

AURELIO. **Vi, Vim e Venci**. Disponível em <http://aurelio.net/vim/vi-vim-venci.html>. Acesso em Set. 2016.

BOTARO, Marcio. Disponível em: <https://github.com/cassiobotaro/vimbook>. Acesso em Set. 2016.

DEVEL SISTEMAS, **Virtualizações de Servidores, Vantagens e Desvantagens.** Disponível em <http://www.develsistemas.com.br/virtualizacao-de-servidores-vantagens-e-desvantagens>. Acesso em Set 2016.

GILMORE, W. Jason. **PHP and MySQL: from novice to professional**. Ed New York:

Apr, 2010.

GOMES, Fábio. **Engenharia de Requisitos: introdução e certificação**. Disponível em:<<http://www.devmedia.com.br/engenharia-de-requisitos-introducao-e-certificacao/28058>>. Acesso em Set. 2016.

IMASTERS, **PHP é muito melhor do que você pensa**. Disponível em: <http://imasters.com.br/linguagens/php/php-e-muito-melhor-do-que-voce-pensa/?trace=1519021197&source=single>. Acesso em Set. 2016.

INFRAERO. **Empregados da Infraero recebem prêmio por ideias inovadoras adotadas na empresa**. Disponível em: <http://www.infraero.com.br/index.php/es/prensa/noticias/3986-1512-empregados-da-infraero-recebem-premio-por-ideias-inovadoras-adotadas-na-empresa.html>. Acesso em Ago 2016.

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

MARIADB. Disponível em: <http://en.wikipedia.org/wiki/MariaDB>. Acesso em Set 2016.

MAYECREATE. **5 razões do porque o PHP é uma grande linguagem de programação.** Disponível em: <http://www.mayecreate.com/2013/12/5-reasons-php-great-programming-language>. Acesso em Ago. 2016.

MEDRADO, Adler. **Porque aprender PHP.** Disponível em: <https://phpparatodos.com.br/2015/11/14/por-que-usar-php>. Acesso em Set. 2016.

MAMONA, Karla Snatana. **9 Coisas que um chefe não pode dizer a um empregado.** Disponível em: <http://www.infomoney.com.br/carreira/gestao-e-lideranca/noticia/2620273/coisas-que-chefe-nao-pode-dizer-empregado>. Acesso em Set. 2016.

NAVATHE, Ramez Elmasri Shamkant B. **Sistemas de Banco de Dados**. 2010.

NETCRAFT, **Web Server Survey**. Disponível em <https://news.netcraft.com/active-sites.html>. Acesso em Set 2016.

OLIVEIRA, Willia. **Começando com o editor de texto VIM**. Disponivel em: <[https://woliveiras.com.br/posts/Comecando-com-o-editor-de-texto-VIM](https://woliveiras.com.br/posts/Comecando-com-o-editor-de-texto-VIM/)>. Acesso em Set 2016.

OGLIO, Pablo Dall’. **PHP Programando Orientação a Objetos.** Ed Novatec 2009.

PIERRO, Massimo Di. **Web2Py**. Ed Lulu Press Inc. 2012

PRESMAN, Roger S. **Engenharia de Software uma abordagem Profissional**. Ed Amgh 2011.

SITE OFICIAL DO GIT. Disponível em: <https://github.com>. Acesso em Set. 2016.

SITE OFICIAL DO MARIADB. Disponível em: <https://mariadb.org>. Acesso em Set. 2016.

SITE OFICIAL DO MARIADB. Disponível em: <http://php.net>. Acesso em Set. 2016.

SITE OFICIAL DO VIM. Disponível em: <https://vim.org>. Acesso em Set. 2016.

SITE OFICIAL DO VIM. Disponível em: <https://nginx.org>. Acesso em Set. 2016.

SOMMERVILLE, I. **Engenharia de Software**. Prentice Hall Brasil, 2003.

VIMBOOK. **O editor de texto vim**. Disponível em: <https://www.gitbook.com/book/cassiobotaro/vimbook/details>. Acesso em Set. 2016.

# **Apêndices**

**APÊNDICE A – Requisitos do sistema**

|  |
| --- |
| **Requisitos Funcionais** |

|  |  |
| --- | --- |
| **RF01 – Funcionalidade do Sistema** | |
| **Pré-Requisitos** | - O Servidor do sistema estar ligado |
| **Pós-Requisitos** |  |
| **Descrição** | - O sistema deverá disponibilizar uma área para os usuários comuns realizarem o cadastro de novas ideias para a empresa e usuários administradores manterem as ideias, o usuário comum poderá enxergar apenas as suas ideias, já o administrador enxergará todas. |

|  |  |
| --- | --- |
| **RF02 – Login no Sistema** | |
| **Pré-Requisitos** | - O Servidor do sistema estar ligado |
| **Pós-Requisitos** | - Usuário logado |
| **Descrição** | - O sistema deverá solicitar o usuário e senha de quem realizar as operações. |

|  |  |
| --- | --- |
| **RF03 – Cadastro de funcionários** | |
| **Pré-Requisitos** | - RF02 |
| **Pós-Requisitos** | - Funcionário cadastrado. |
| **Descrição** | - O sistema deverá ter uma área para cadastrar os dados dos funcionários, no qual já estará incluso os dados do usuário. |

|  |  |
| --- | --- |
| **RF04 – Cadastro de ideias** | |
| **Pré-Requisitos** | - RF02 |
| **Pós-Requisitos** | - Ideia cadastrada |
| **Descrição** | - O sistema devera permitir que um funcionário cadastre novas ideias para a empresa, no qual o mesmo poderá acompanhar apenas o status das ideias que informou. |

|  |  |
| --- | --- |
| **RF05 – Alterar o status das ideias** | |
| **Pré-Requisitos** | - RF02 e ter um perfil de administrador. |
| **Pós-Requisitos** | - Ideias atualizadas |
| **Descrição** | - O sistema devera permitir que o usuário administrador altere o status das ideias cadastradas.  - Caso a ideia seja nova, o usuário ganha 1 ponto.  - Caso a ideia vá para análise, o usuário ganha 5 pontos.  - Caso a ideia seja aprovada, o usuário ganha 10 pontos.  - Uma ideia só pode ser excluída se for nova.  - Caso a ideia seja excluída, o usuário perderá um ponto. |

|  |  |
| --- | --- |
| **RF06 – Manter prêmios** | |
| **Pré-Requisitos** | - RF02 e ter perfil de administrador |
| **Pós-Requisitos** | - Prêmio mantido |
| **Descrição** | - O sistema deve permitir que o usuário administrador possa cadastrar, alterar, excluir e atualizar um prêmio |

|  |  |
| --- | --- |
| **RF07– Retirada de prêmios** | |
| **Pré-Requisitos** | - RF02 e usuário ter pontos suficientes para a retirada do prêmio. |
| **Pós-Requisitos** | - Premio retirado |
| **Descrição** | - O sistema deverá ter uma seção para o usuário poder solicitar a troca dos pontos pelos prêmios cadastrados, após a retirada do prêmio os pontos necessários serão descontados do funcionário |

|  |  |
| --- | --- |
| **RF08 – Geração de Relatórios** | |
| **Pré-Requisitos** | - RF03, RF05, RF06 e RF07 |
| **Pós-Requisitos** | - Relatório gerado |
| **Descrição** | - O sistema deverá permitir que seja gerado relatórios das ideias cadastradas, que poderá ser filtrado pelo status das ideias.  - O sistema deverá permitir que seja gerado relatórios dos funcionários, que poderá ser filtrado por ideias por funcionário.  - O sistema deverá permitir que seja gerado relatórios dos prêmios, que poderá ser filtrado por quantidade de prêmios já entregues e disponíveis em estoque. |

|  |
| --- |
| **Requisitos Não Funcionais** |

|  |  |
| --- | --- |
| **RNF01 – Portabilidade** | |
| **Pré-Requisitos** | - |
| **Pós-Requisitos** | - |
| **Descrição** | - O sistema deve ser desenvolvido de modo que suas funcionalidades e interfaces sejam compatíveis com o navegador firefox >= 35 ou google chrome >=45 |

|  |  |
| --- | --- |
| **RNF02 – Desempenho** | |
| **Pré-Requisitos** | - |
| **Pós-Requisitos** | - |
| **Descrição** | - O sistema deve demorar no máximo dois segundos para realizar o carregamento das suas páginas quando tiver um número de usuários conectados menor que 30. |