FACULDADE DE TECNOLOGIA DE PORTO ALEGRE

CURSO TECNÓLOGO DE ANALISE E DESENVOLVIMENTO DE SISTEMAS – PORTO ALEGRE



|  |
| --- |
| BANCO DE IDEIAS |
| Guilherme Nunes Fontans |

Relatório de atividades desenvolvidas durante o Estágio Supervisionado e apresentada ao Curso Superior de Tecnológica em Análise e Desenvolvimento de Sistemas da Faculdade de Tecnologia de Porto Alegre, como pré-requisito para a obtenção do título de Graduação Tecnólogo em Análise e Desenvolvimento de Sistemas.

Supervisor: Prof. Eduardo Reus

Porto Alegre, dezembro de 2016.

# **INTRODUÇÃO**

Muitas vezes dentro de uma empresa, é bom ter uma opinião de fora do seu setor, pois essa opinião seria de alguém que está vendo os produtos ou processos de uma maneira diferente, e pode ter uma ideia nova, ou uma ideia para melhorar oque já existe. Por isso a proposta foi desenvolver um sistema web para realizar o gerenciamento das ideias propostas pelos funcionários, tanto para outros setores, ou para o seu setor, podendo ser cadastrado melhorias de processos ou novas ideias.

O software criado chama-se Banco de Ideias, no qual é permitido 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.

O intuito da escolha da criação desse software é poder proporcionar mais participação por parte dos funcionários na escolha de novos produtos para serem criados pela empresa, trazendo assim mais incentivo a participarem devido aos prêmios que podem ser obtidos.

No decorrer desse relatório será apresentado as tecnologias utilizadas durante o desenvolvimento do software e como o mesmo está estruturado.

# **REQUISITOS**

O levantamento dos requisitos de um software é uma tarefa importante a ser realizada na sua ideação. Apesar de ser um processo longo e árduo, o mesmo possui um papel fundamental, pois é através dela que surgem as demais etapas que compõem o software.

Com base nas observações efetuadas na empresa, foi detectado quais seriam as necessidades básicas que os usuários finais teriam. Diante disso identificou-se que o sistema conteria as principais funcionalidades: cadastro, alteração, busca e remoção de ideias dos usuários, áreas e prêmios, poder solicitar prêmios, exportação das telas de listagem dos itens para pdf, notificação de email aos administradores quando um prêmio é solicitado e um acesso através de login e senha.

Partindo dessa análise, foi criado o documento de análise de requisitos, para fins de obter maio clareza do que deveria ser feito, servindo de consulta durante o desenvolvimento do software. O documento de análise de requisitos está presente no apêndice A, presente em anexo neste relatório.

Baseado na análise de requisitos foi criado o diagrama de caso de uso de todo o sistema, tendo assim uma melhor clareza do que é possível cada ator efetuar quais os passos necessários para efetuar uma tarefa dentro do sistema.

Na figura 1 é apresentado através da UML, o diagrama de caso de uso do requisito manter ideia. A figura apresentada representa as quatro operações que o usuário administrador pode efetuar no menu de ideias, no qual o mesmo pode visualizar todas as ideias cadastradas, excluir uma ideia desde que o status da mesma permita essa operação. O usuário administrador também pode cadastrar e alterar as ideias cadastradas, operações estas que automaticamente afetam os pontos do usuário que cadastrou a ideia.

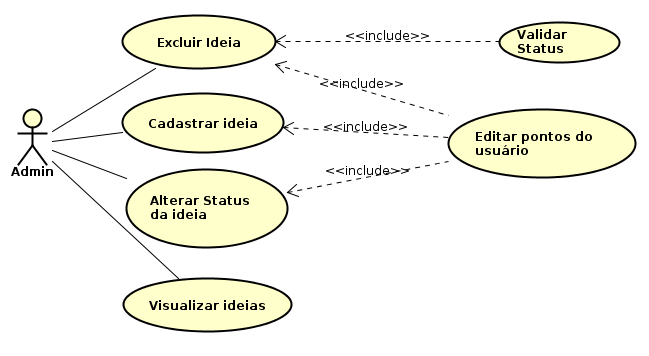


Figura 1: Caso de uso manter ideia

A descrição completa de todos os passos utilizados durante o cadastro de uma ideia esta descrito no diagrama de sequência, presente no apêndice B, presente no anexo deste relatório, assim como o Diagrama de caso de uso das demais funcionalidades do sistema, presente no apêndice C.

# **TECNOLOGIAS UTILIZADAS**

Após efetuar a análise de requisitos do sistema, foi feito a escolha das tecnologias que foram utilizadas para iniciar o desenvolvimento do software. Essas tecnologias são apresentadas em detalhes nas subseções abaixo.

## **Linguagem de programação**

O PHP foi escolhido como linguagem de programação para ser utilizado no desenvolvimento do software, pois a linguagem possui uma sintaxe simples e proporciona os recursos necessários para atender as demandas de um sistema web, além de ter ganhado grande desempenho em sua ultima versão 7.0.10, versão esta utilizada no desenvolvimento.

Além de ser hoje a linguagem mais utilizada para servidores web no mundo, o PHP 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 chamado composer, oque garante que as versões das bibliotecas utilizadas no desenvolvimento da aplicação sejam o mesmo caso haja interação de mais desenvolvedores no futuro, não correndo risco de um desenvolvedor utilizar um dos componentes com uma versão incompatível com a versão que o software demanda.

A linguagem também possui inúmeros frameworks e componentes que foram desenvolvidos por usuários e disponibilizados livremente, e os mesmos podem ser utilizados em paralelos com outros componentes, graças aos padrões propostos pela comunidade PHP-FIG. Dentre esses frameworks, foi escolhido o micro-framework Silex para ser aplicado no sistema.

## **Servidor web**

Para servir as páginas web que os usuários da empresa irão visualizar, foi utilizado o servidor web NGINX, ganhando assim mais desempenho em relação a servidores web mais tradicionais como o apache ou httpd, pois dentre as vantagens do NGINX, o mesmo possui o FastCGI, recurso que pode ser utilizado em conjunto com o PHP.

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.

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).

## **Banco de Dados**

Para o desenvolvimento do sistema, foi necessário utilizar um SGBD, tendo em vista que os dados que serão colocados no mesmo precisam ser consultados posteriormente, e também poderão vir a sofrer mudanças. Para isso, foi escolhido o SGBD MariaDB para armazenar os dados que serão salvos no software Banco de Ideias. Tendo em vista o levantamento de requisitos efetuado, modelou-se o diagrama ER apresentado na figura 2 para a base de dados do sistema.

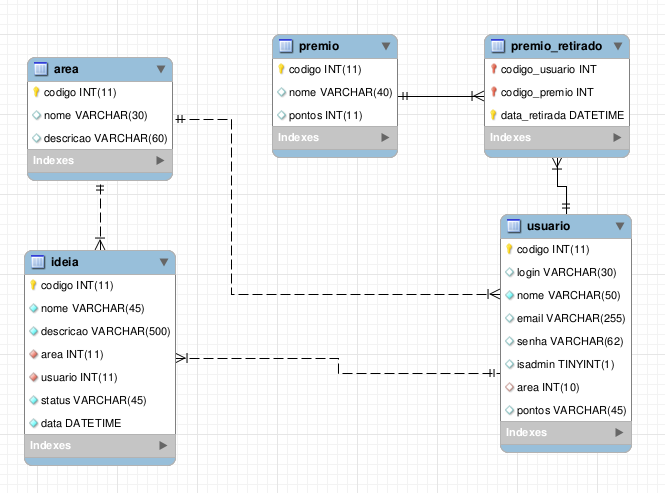


Figura 2: Modelo ER

A figura citada demonstra as cinco tabelas relacionadas entre si do sistema. Esse relacionamento foi modelado com o propósito de buscar melhor uso do banco de dados usado bem como ter melhor relação das informações armazenadas. As tabelas a seguir são apresentadas e detalhadas:

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

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

3) Premio\_retirado: usada para emissão de relatórios, para fins de ter um histórico de qual usuário retirou qual prêmio;

4) Usuário: responsável por armazenar os dados do usuário, e por efetuar o login do sistema;

5) Ideia: armazena todas ideias cadastradas pelos usuários e registra de 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.

# **ARQUITETURA**

Após escolher as tecnologias que foram utilizadas, foi iniciado o desenvolvimento do software. Nas subseções seguintes é apresentado como está a estrutura física do software e também a estrutura de seus arquivos.

## **Arquitetura física**

O sistema foi desenvolvido em uma máquina virtual cujo sistema operacional instalado é um Linux CentOS 7 no qual a mesma máquina foi disponibilizada através do seu export para o formato OVA, bastando apenas importar esse arquivo em um virtualizador para que a mesma já funcione conforme o esperado, tendo os componentes necessários para o seu funcionamento no ar, como o servidor web e o banco de dados iniciados no sistema.

A figura 3 demonstra a arquitetura do sistema após ter sido efetuado a importação da máquina virtual em um computador que tenha um virtualizador.

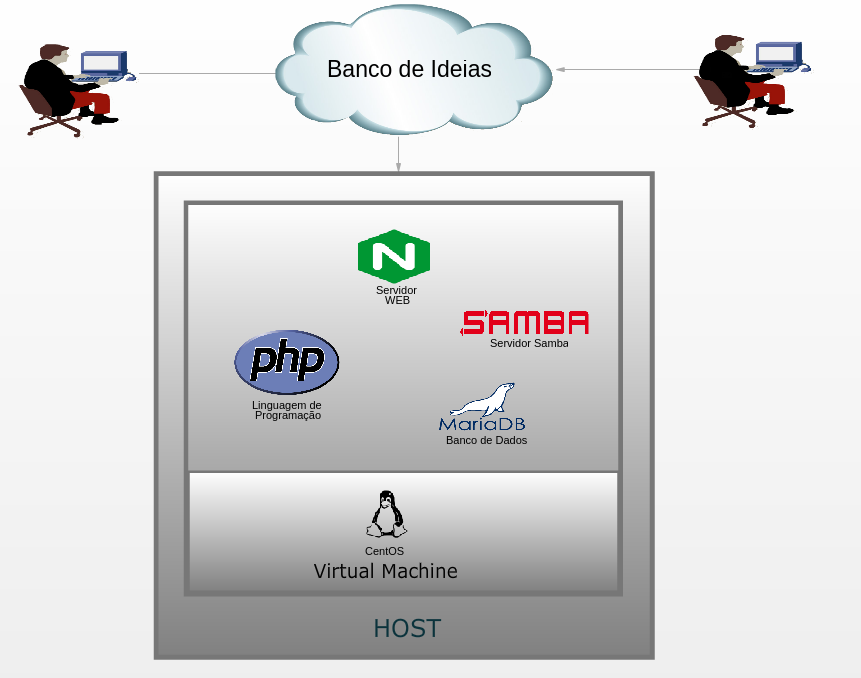


Figura 3: Arquitetura do sistema

A figura acima apresenta a estrutura do sistema, no qual o host representa a máquina em que o arquivo com a extensão OVA será importado, gerando assim a maquina virtual já configurada, na qual já possui a linguagem de programação, o servidor WEB e o banco de dados configurados para trabalharem em conjunto para disponibilizarem as páginas do sistema para o usuário final através da URL do mesmo.

Também está configurado um servidor samba para poder compartilhar pastas do sistema com demais computadores de outros sistemas operacionais, está configuração tem o intuito de permitir o acesso desses demais computadores a pasta em que ficam armazenados os backups do sistema, sendo efetuado o backup das páginas todos os dias as 00:20 e o backup da base de dados todos os dias as 01:20. Ambos os processos de backup são efetuados através do agendador de tarefas do sistema operacional disponibilizado na máquina virtual e disponíveis no compartilhamento [\\ip-do-sistema\banco-de-ideias](file:///\\ip-do-sistema\banco-de-ideias).

Ao importar a máquina virtual, a mesma está configurada para capturar um IP válido na rede da empresa através do protocolo DHCP, este IP será exibido ao término da carga do sistema conforme exibido na figura 4. Esta apresenta uma mensagem contendo o endereço completo que o usuário deverá utilizar no navegador para poder acessar o software. O IP informado foi obtido via DHCP durante a carga do sistema operacional.

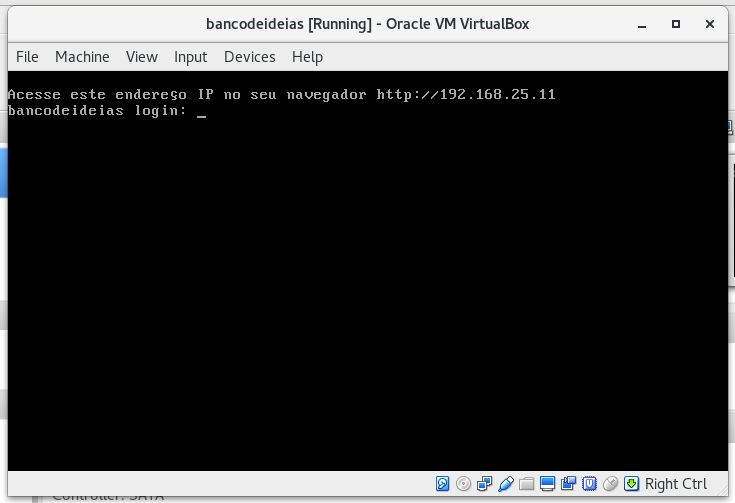


Figura 4: Tela com informação do IP

## **Arquitetura dos arquivos**

A implementação foi realizada utilizando as ferramentas descritas na seção dois deste relatório, e de acordo com os padrões de projeto MVC e DAO, garantindo que o sistema fosse feito de forma organizada e clara.

Na figura a seguir, é apresentado como foi criado a camada de modelo do sistema, o diagrama de todo o sistema está presente no apêndice D em anexo a este relatorio.

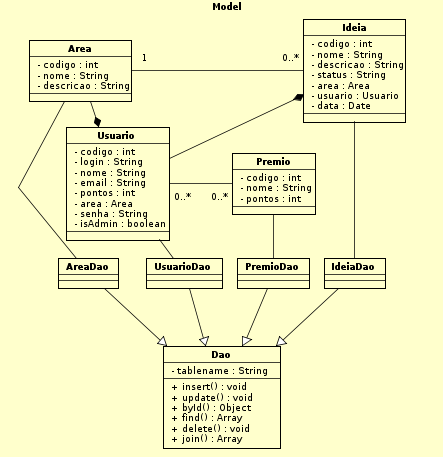


Figura 5: Diagrama de classe

A figura 5 demonstra as nove classes que compõem a camada de modelo da aplicação. Essa estrutura foi modelada com o proposito de criar as dependências entre que existem entre os objetos do sistema, e utilizando o pilar ‘herança’ da orientação a objetos para fins de facilitar a manutenção do código. As classes a seguir são apresentadas e detalhadas:

1) Area, Premio, Usuario e Ideia: são as classes utilizadas para efetuar a representação das mesmas dentro do sistema, no qual cada um dos seus atributos são representados dentro do diagrama;

2) Dao: classe abstrata que possui as funções responsáveis pelas operações de inclusão, busca, exclusão e atualização de um objeto no banco de dados;

3) AreaDao, PremioDao, UsuarioDao e IdeiaDao: são as classes que herdam a classe Dao para que cada objeto tenha a sua classe de persistência relativa a sua tabela no banco de dados.

## **Framework**

Para fins de agilizar o desenvolvimento do software, foi utilizado o micro-framework chamado Silex, no qual o mesmo auxilia no mapeamento das URLs do sistema, disponibilizando assim um maior controle na renderização das páginas conforme solicitação.

Dentre as vantagens do Silex, temos a integração com diversos componentes do framework Synfony e Laravel, no qual os componentes utilizados estão sendo apresentados a seguir:

Monolog: Utilizado para gerar logs do sistema, esse componente foi de extrema importância para efetuar o debug durante o desenvolvimento do sistema, ajudando a identificar problemas durante esse oprocesso;

Request: Utilizado para capturar as informações passadas por POST pelos formulários do sistema.

Twig: Engine utilizado nas paginas de view, fazendo com que ficasse mais limpo o código, não sendo necessário misturar tags php com html.

IlluminateDatabase: Componente que efetua todas as operações entre a aplicação e o banco de dados, não sendo necessário criar nenhuma query manualmente dentro das classes do sistema.

Snappy: utilizado para efetuar o export das telas de apresentação das grids do sistema para pdf.

# **BANCO DE IDEIAS**

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

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, e a mesma 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.

## **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 6 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.

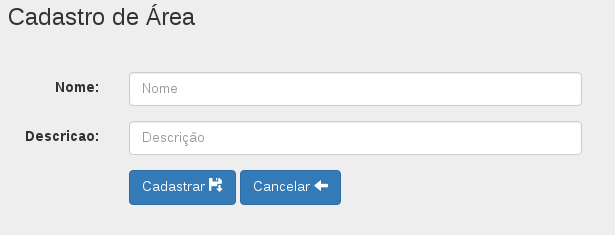


Figura 6: Formulário de cadastro de área

Na figura 7 é apresentada a mensagem retornada para o usuário após a confirmação da criação da área.

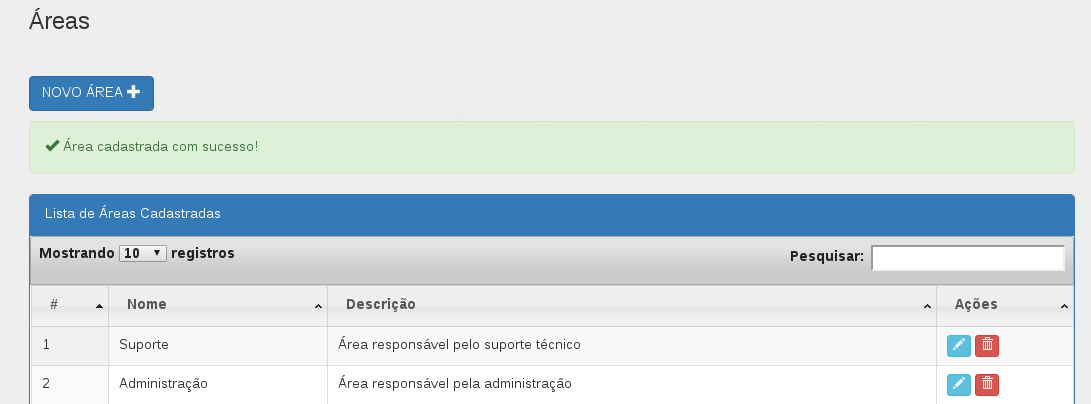


Figura7: Mensagem de confirmação de cadastro de área

Após realizado o cadastro da área, a mesma já estará disponível para associar um usuário na tela de cadastro de usuários, conforme demostrada na figura 8, apresentando as áreas em um botão do tipo *combobox.*

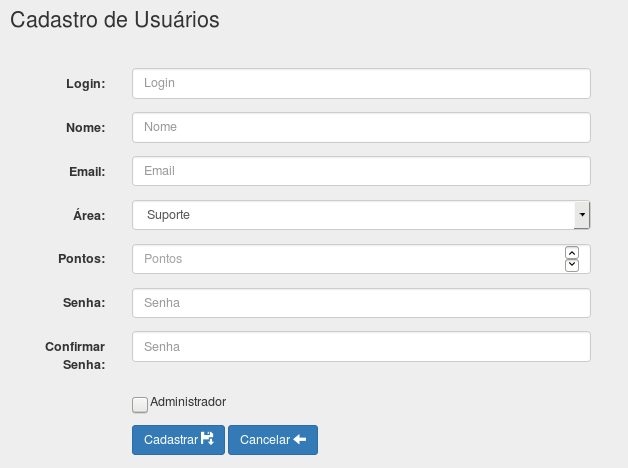


Figura 8: Formulário de cadastro de usuário

Na figura 8, 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.

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 9 demonstra a apresentação dos menus laterais do sistema, no qual o usuário encontrará acessando o menu ‘Home’.

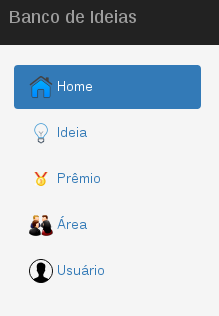


Figura 9: Menu lateral

Com o intuito de trazer uma melhor experiência ao usuário, os pontos são apresentados ao lado do seu nome em uma barra superior que acompanha todas as telas do software, conforme demonstrado na figura 10.



Figura 10: Menu superior com dados do usuário

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, os mesmos também 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, ficará ativo o botão de “solicitar”, disponível na coluna “solicitar troca” da listagem de prêmios, podendo o usuário clicar e confirmar a solicitação da troca para que automaticamente os pontos sejam descontados da sua conta, e o responsável pela área do usuário seja notificado por e-mail para poder tomar as devidas providências para que o prêmio seja entregue ao usuário solicitante. A figura 11 demonstra a listagem de prêmios, quando o mesmo já está disponível para o usuário logado no sistema.

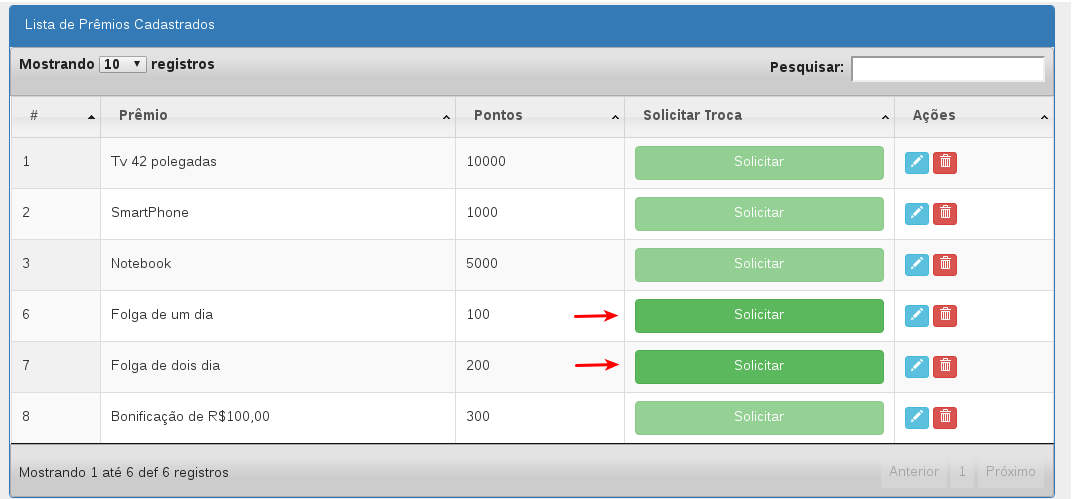


Figura 12: Lista de prêmios disponíveis para o usuário

A figura 11 apresenta a tela de listagem de prêmios cadastrados no sistema, no qual que permite a solicitação da troca dos pontos pelos prêmios encontra-se inativo, sendo ativado apenas para aqueles itens cujos pontos tenham sido alcançados pelo usuário.

# **CONCLUSÃO**

O Banco de Ideias disponibiliza e possibilita o cadastro de sugestões por parte dos funcionários, demonstrando uma possível aproximação da empresa com os seus funcionários na busca de inovações ou melhorias internas de seus processos.

O sistema foi desenvolvido em uma máquina virtual com o sistema operacional Linux CentOS 7, no qual a mesma máquina foi importada no virtualizador da empresa, 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.

Durante o desenvolvimento do software, observou-se 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. Através deste levantamento, obteve-se uma direção mais precisa em relação à estrutura das classes que irão compor o sistema.

**ANEXO 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 – Manter áreas** | |
| **Pré-Requisitos** | - RF02 e ter perfil de administrador |
| **Pós-Requisitos** | - Área mantida |
| **Descrição** | - O sistema deve permitir que o usuário administrador possa cadastrar, alterar, excluir e atualizar uma área. |

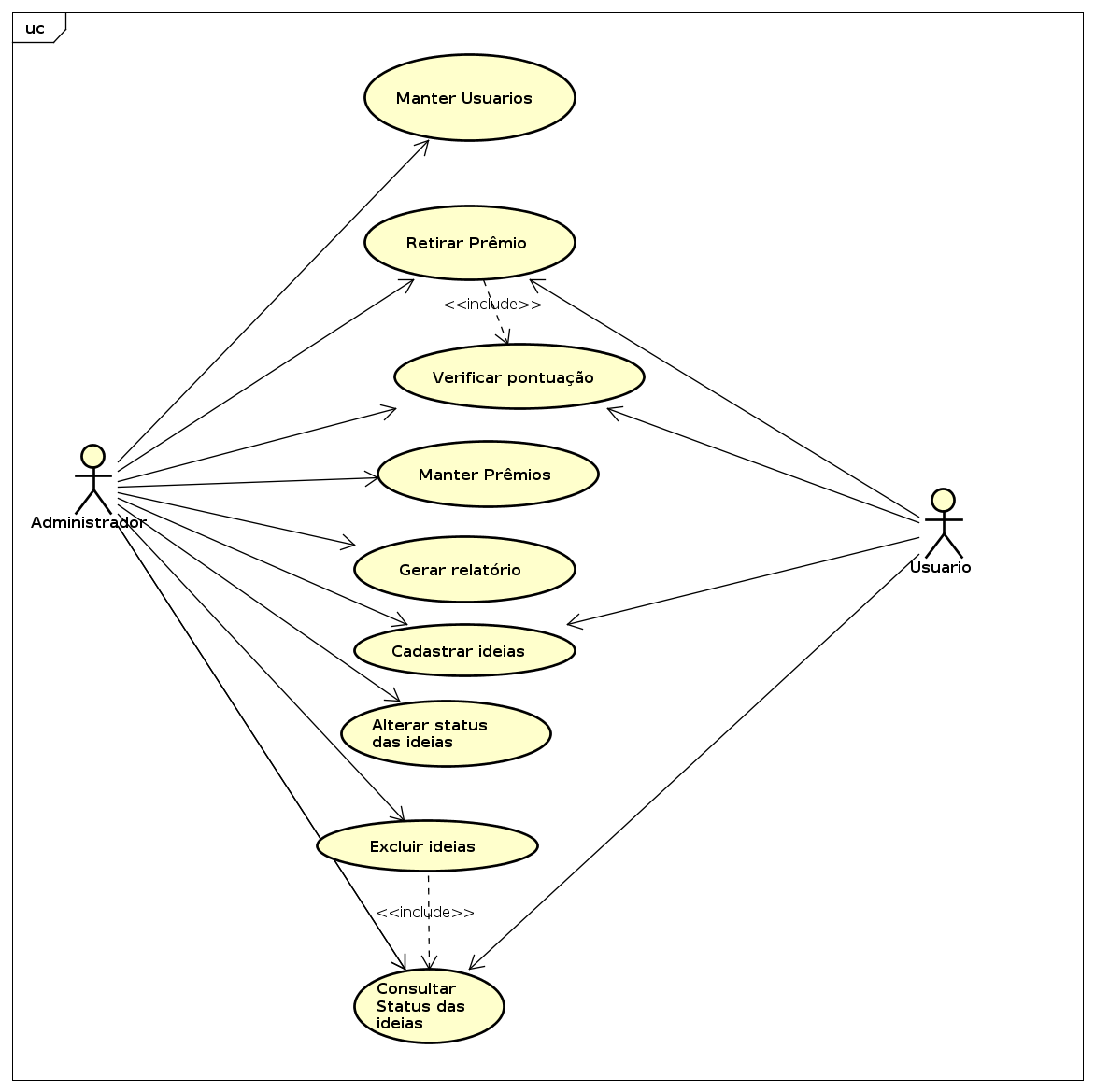
|  |  |
| --- | --- |
| **RF08– Retirada de prêmios** | |
| **Pré-Requisitos** | - RF02 e usuário ter pontos suficientes para a retirada do prêmio. |
| **Pós-Requisitos** | - Premio retirado |
| **Descrição** | - O sistema deverá ter uma seção para o usuário poder solicitar a troca dos pontos pelos prêmios cadastrados, após a retirada do prêmio os pontos necessários serão descontados do funcionário |

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

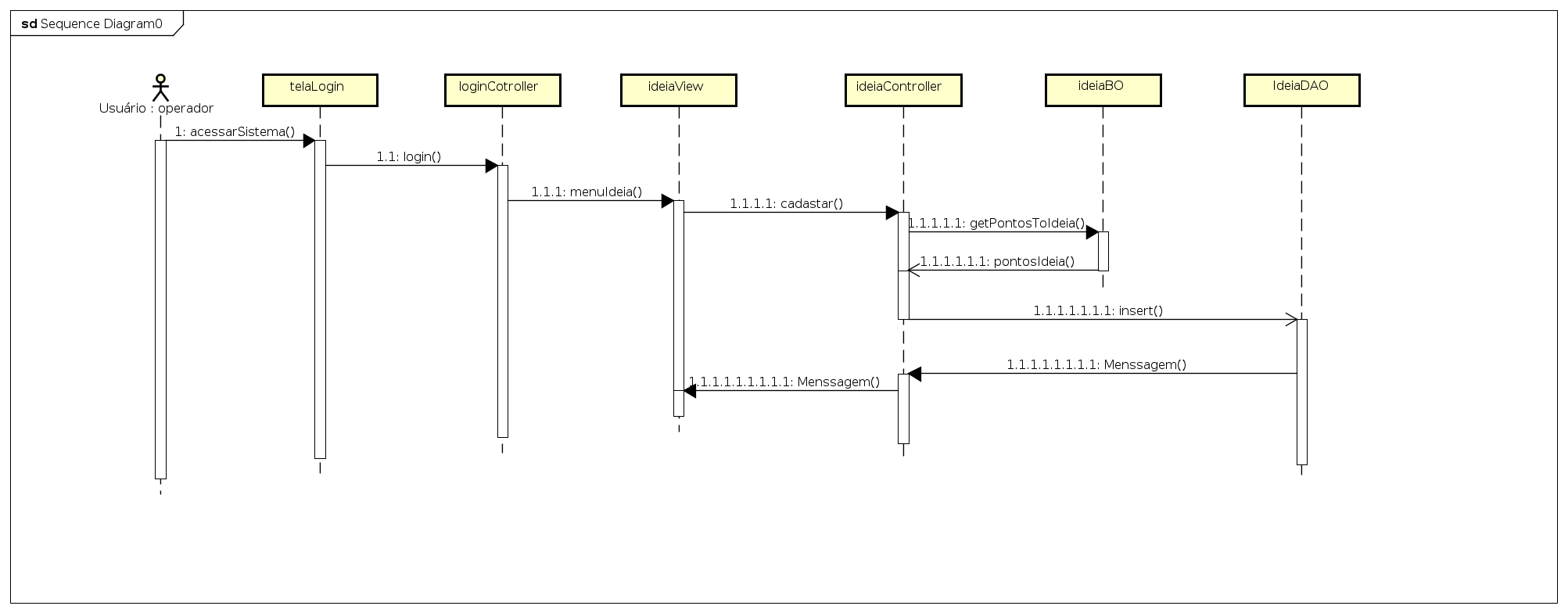
|  |  |
| --- | --- |
| **RNF01 – Portabilidade** | |
| **Pré-Requisitos** | - |
| **Pós-Requisitos** | - |
| **Descrição** | - O sistema deve ser desenvolvido de modo que suas funcionalidades e interfaces sejam compatíveis com o navegador firefox >= 35 ou google chrome >=45 |

|  |  |
| --- | --- |
| **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. |

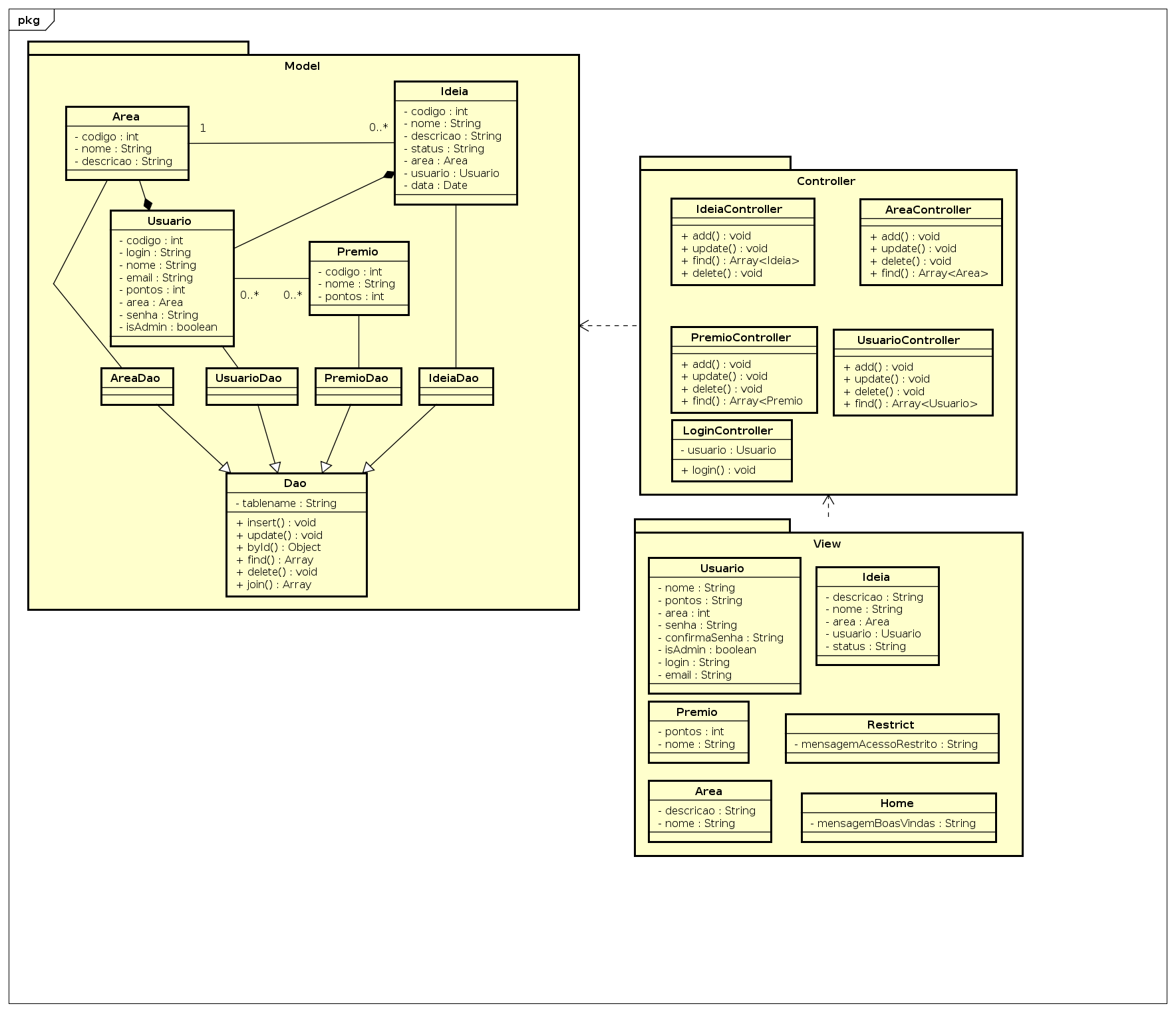
**ANEXO B – Diagrama de caso de uso**

****

**ANEXO C – Diagrama de sequência – Cadastrar ideia**

****

**ANEXO D – Diagrama de classe – Cadastrar ideia**

****

# **REFERÊNCIAS**

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.

DEVMEDIA, **Artigo SQL Magazine 63 - Utilizando UML: Diagrama de Classes**. Disponível em: <http://www.devmedia.com.br/artigo-sql-magazine-63-utilizando-uml-diagrama-de-classes/12251>. Acesso em Nov. 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.

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.

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

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

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

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

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