GreatIdea: Ferramenta para auxiliar o desenvolvimento de ideias por meio do trabalho coletivo

Erick H. L. Silva[1], Daiane M. Tomazeti[2]

[1]Superior em Tecnologia em Análise e Desenvolvimento de Sistemas - Instituto Federal de São Paulo - Câmpus Hortolândia (IFSP)

erickhleite@gmail.com[1], daianetomazeti@ifsp.edu.br[2]

Abstract. *In the last decade the speed of technological growth has increased considerably and together with this movement it becomes even more difficult to come up with a new idea to work with. At the Federal Institute of São Paulo, Câmpus Hortolândia, it can be noted that this phenomenon affects students who are looking for ideas to develop, either as personal projects or as a conclusion work.. The GreatIdea system is designed to help students and teachers in the câmpus share ideas with each other and to encourage them to develop these ideas through common work, helping each other from the beginning to the end of project development. This article aims to present the GreatIdea system, the methodology used in its development from the beginning until its conclusion and also the expected results regarding the impact that the application can cause in the institute.*

Resumo. *Na última década a velocidade do crescimento tecnológico têm aumentado de forma considerável e juntamamente a este movimento se torna cada vez mais difícil elaborar uma ideia nova para se trabalhar. No Instituto Federal de São Paulo câmpus Hortolândia pode-se notar como este fenômeno afeta os discentes que buscam ideias para desenvolver, sejam como projetos pessoais ou como trabalho de conclusão de curso. O sistema GreatIdea foi pensado para auxiliar discentes e docentes do câmpus a compartilharem ideias entre si, além de incentivá-los a desenvolver essas ideias através do trabalho comum, ajudando uns aos outros do início até o fim do projeto. Este artigo têm como objetivo apresentar o sistema GreatIdea, a metodologia utilizada em seu processo de implementação e também os resultados esperados referentes ao impacto que a aplicação pode causar na instituição.*

1. Introdução

Nos tempos atuais, o número de ideias inovadoras que surgem dia após dia têm crescido de forma significativa [Canaltech 2015], este fenômeno se dá por conta da facilidade de acesso a informação, que cresce juntamente a evolução tecnológica. O conceito de banco de ideias surgiu para ajudar no armazenamento das mesmas para que possam ser implementadas futuramente.

Este trabalho realizou um estudo de caso baseado no IFSP Câmpus Hortolândia, no qual foram abordados os possíveis benefícios de se possuir um banco de ideias aplicado ao mesmo. Dessa forma, foi realizado o desenvolvimento de uma aplicação *Web* que consiste em um banco de ideias *online*, mas com um diferencial, nela além de se consultar ideias disponíveis para implementação, é possível também auxiliar pessoas a consolidar seus projetos por meio do trabalho coletivo, a aplicação auxilia na formação de um time de desenvolvimento para trabalhar neste projeto e acompanhar o processo até que ele chegue ao seu término.

Por fim, este trabalho está estruturado da seguinte forma:

A seção 2 mostra os trabalhos similares ao GreatIdea que foram identificados durante o processo de pesquisa deste trabalho e resume a forma com que estes trabalhos funcionam.

Na seção 3 são explicados alguns conceitos teóricos e técnicos importantes para o melhor aproveitamento deste trabalho.

A seção 4 apresenta toda a metodologia utilizada durante o processo de desenvolvimento do projeto.

Na seção 5 são apresentados os requisitos do sistema, que foram elaborados no início do projeto e utilizados como metas a serem cumpridas ao longo do desenvolvimento da aplicação.

Na seção 6 temos os diagramas de caso de uso e de classes do sistema, os quais servem para efetuar uma descrição ilustrativa organizacional do sistema.

A seção 7 exibe o sistema de Capitão e Ajudante e explica como essa metodologia atua de forma organizacional dentro da aplicação.

Na seção 8 são exibidos os fluxos do sistema, que são algumas ações mais comuns que os usuários podem executar dentro da ferramenta, e como a aplicação deverá se comportar em relação a essas ações.

A seção 9 exibe as principais telas da aplicação e suas funcionalidades, as quais os usuários poderão interagir para realizar o objetivo proposto pela ferramenta.

Na seção 10 são exibidos os resultados obtidos através de uma seção de testes realizada com usuários para consolidar o aceite da aplicação pelo público comum.

Por fim, nas seções 11 e 12 estão descritos a conclusão e os trabalhos futuros referentes a este estudo, respectivamente.

1.1. Justificativa

Após conversa com diversos discentes do curso de Análise e Desenvolvimento de Sistemas do câmpus, pode-se identificar a dificuldade que vários deles têm de encontrar boas

ideias para iniciarem o processo de desenvolvimento do Trabalho de Conclusão de Curso, sendo que em alguns caso, os discentes não conseguem chegar a um consenso quanto ao estudo que irão realizar ainda no quarto semestre, postergando esta decisão para os semestres seguintes, situações que causam problemas quanto ao andamento do curso do discente e pode gerar transtornos maiores quanto ao tempo de conclusão no futuro. Após consultar os sistemas abertos a discentes do câmpus, pode-se constatar que o mesmo ainda não utiliza um banco de ideias próprio para projetos internos ou para auxiliar os discentes a iniciarem novos projetos. Neste estudo foi desenvolvida uma ferramenta que atua como um banco de ideias para o Instituto Federal de São Paulo - Câmpus Hortolândia, de forma a auxiliar estes discentes, fornecendo ideias para que os mesmos possam se inspirar e de fato iniciarem a implementação de um projeto para a conclusão do curso.

1.2. Objetivos gerais e específicos

Este trabalho têm por objetivo desenvolver uma ferramenta que possa prover um ambiente de compartilhamento de ideias voltada ao Instituto Federal de São Paulo câmpus Hortolândia. A aplicação foi pensada com o intuito de incentivar o compartilhamento de ideias entre discentes e docentes, além de ajudar discentes a desenvolver projetos e intermediar a colaboração no desenvolvimento de uma ideia para que ela se concretize.

2. Trabalhos correlatos

São projetos existentes que podem possuir similaridades com a proposta realizada pelo trabalho em questão. Ao longo das pesquisas para realizar o desenvolvimento deste trabalho, pode-se observar algumas aplicações com propostas similares, são elas o Trello e o Meetup.

2.1. Trello

Trello é um aplicativo gerenciador de projetos que funciona em ambiente *web* com a proposta de auxiliar pessoas a organizarem o processo de implementação de seus projetos através da utilização de Kanban. O principal ponto positivo do Trello é a facilidade de organização e visualização dos projetos através de seus quadros, que promove um ganho considerável de produtividade. Por outro lado, o sistema de compartilhamento dos projetos com outras pessoas ainda é bastante simplicado na ferramenta, visto que nela só se é possível adicionar pessoas previamente conhecidas aos seus projetos para quem visualizá-lo.

2.2. Meetup

O Meetup é uma ferramenta que funciona em ambiente *Web* cuja proposta é a formação de grupos de pessoas para a realização de eventos presenciais. A principal limitação do Meetup se dá exatamente no fato de que os encontros são realizados de forma presencial, o que limita a sua área de atuação quando a ideia é formar um grupo de pessoas para uma atividade *online*.

3. Fundamentação Teórica

3.1. Qualidade de *software*

Assim como em outros setores, a garantia da qualidade de um produto de *software* é de extrema importância. Atualmente existem órgãos que fornecem níveis de maturidade para

empresas que trabalham com o desenvolvimento de sistemas, para certificar as empresas que produzem *softwares* respeitando os níveis aceitáveis de qualidade no Brasil, o órgão responsável por esta certificação é o MPS.BR [Softex 2019]. Para garantir a qualidade de um produto de *software* são utilizadas técnicas de testes e validações como por exemplo: teste unitário, teste de integração, teste de sistema, teste de aceitação, homologação e teste de regressão, durante e após o período de desenvolvimento. Estas técnicas verificam se o produto atende aos requisitos funcionais e não funcionais do projeto.

Na seção 10 pode-se observar os testes com usuários realizados para este trabalho, com o intuito de detectar possíveis erros de execução e também para obter um teste de aceitação realizado com usuários reais.

3.2. UML

UML - Linguagem de Modelagem Unificada é um conjunto de padrões voltados a elaboração da estrutura de projetos de *software* com o intuito de padronizar e otimizar a visualização, elaboração, documentação e especificação de estruturas de projetos de *software* [Booch 2012]. Com seu vasto portifólio de diagramas voltados a esse segmento, a UML é fundamental para manter intactas as diretrizes da Engenharia de *software* nos dias de hoje e é recomendado sua utilização em projetos de sistemas que visam otimizar ganhos e reduzir custos durante o planejamento.

3.3. Kanban

Kanban é um termo de origem japonesa que significa literalmente "cartão" ou "sinalização" [Bernardo 2014]. A empresa de automóveis Toyota foi a responsável pela introdução desse método que consta na utilização de cartões *Post-it*, para sinalizar o andamento dos fluxos de produção em empresas de fabricação em série. O Kanban está ligado ao conceito de *just in time* que significa "no momento certo", é um outro modelo japonês que têm o foco de eliminar estoques e agilizar a produção [Lima 2008]. O Kanban foi escolhido, pois permite planejar as tarefas de forma mais compreensível, completa e com maior visibilidade, centralizando todas as informações mais importantes do projeto em um quadro único, organizado por colunas que definem o andamento da tarefa evitando assim a necessidade de diversas formas de visualização das tarefas que serão executadas durante o processo de implementação do sistema.

Segundo [Mariotti 2018], "O Kanban, com a combinação de fluxo, contribui para a redução do estresse da equipe e melhora a previsibilidade e colaboração, refletindo com isso, nas datas de vencimento para entrega de tarefas".

As tarefas são divididas entre as colunas:

- *Backlog* de cada entrega (coluna com todas as tarefas);
- *To Do* (trabalhos para fazer);
- *In Progress* (trabalhos em andamento);
- *In Analysis* (aguardando análise ou revisão);
- *Done* de cada entrega (tarefas finalizadas).

Na figura 1 é exibido o quadro Kanban do projeto GreatIdea, que foi utilizado durante a elaboração deste estudo.

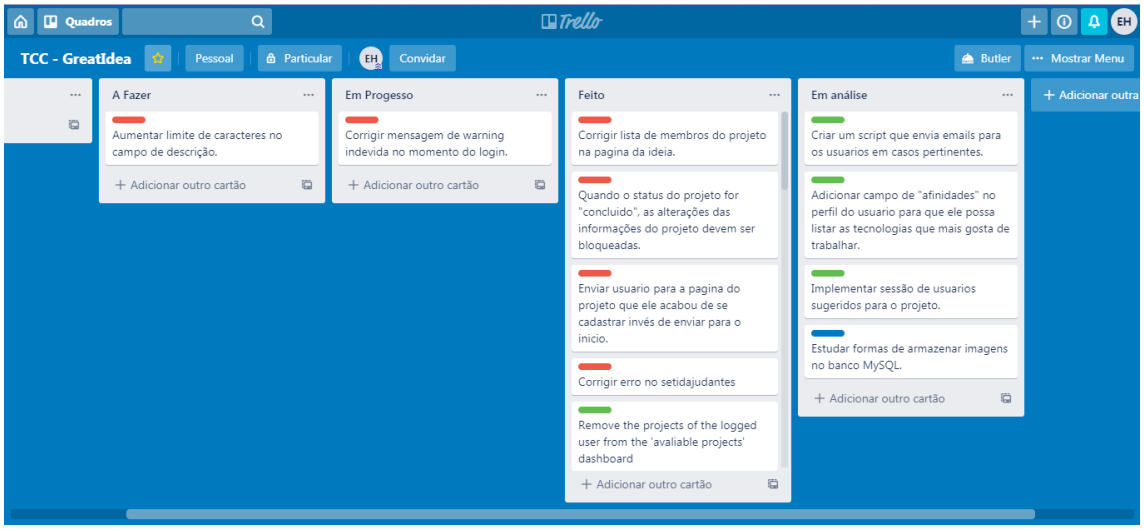

Figura 1. Painel Kanban utilizado no projeto.

3.4. Aplicação *Web*

São sistemas de computadores que são executados com a utilização de um navegador, podem ser acessados através de um computador por meio de um endereço URL (Universal Resource Locator) e não requer instalação local do cliente visto que estes sistemas operam diretamente no servidor que fornece os dados. Para o desenvolvimento de um sistema *Web*, pode-se utilizar diversas linguagens de programação, paradigmas de programação e modelos de implementação, para este trabalho, como linguagem de programação foram utilizados o HTML (Linguagem de Hipertexto), para estruturar as telas da aplicação de forma a posicionar seus componentes e elementos, o CSS (Cascading Style Sheets) foi utilizado para realizar a estilização das telas de forma a facilitar a interface com o usuário, tornando o *layout* mais atrativo e amigável a qualquer um que venha a acessá-lo, e a linguagem PHP para operar no *backend* da aplicação, ou seja, fazer com que os dados sejam processados e passados entre as telas de forma a realizar o funcionamento da aplicação. Como paradigma de programação, foi utilizado o paradigma de Orientação a Objetos, através do MVC (Model-View-Controller) e DAO (Dynamic Access Object).

3.5. Metodologias Ágeis

São metolodogias de desenvolvimento de projetos que surgiram na década de 80 mas só se popularizaram depois dos anos 2000. Desde então, as metodologias ágeis são alinhadas com o Manifesto Ágil [Kent Beck 2001] de Desenvolvimento criado em 2001 e buscam trazer mais agilidade e simplicidade no processo de desenvolvimento de projetos. [Paredes 2019].

Neste trabalho, as metodologias ágeis foram essenciais para a organização do projeto, definindo todas as etapas de ação durante o processo de implementação, tais como erros a serem tratados, alterações necessárias e funcionalidades a serem implementadas a cada execução. Dessa forma, o processo de desenvolvimento ocorreu de forma mais fluida, uma vez que pode-se realizar o desenvolvimento de forma sequencial atuando sob um item por vez, realizando testes de pontos de função a cada *sprint*.

3.6. Banco de dados MySQL

Segundo [Elmasri 2010] Um banco de dados é uma coleção de dados relacionados, possuindo as seguintes propriedades implícitas:

- Representa algum aspecto do mundo real (minimundo ou universo de discurso);
- É uma coleção de dados logicamente coerentes e
- É projetado, construído e instanciado (“povoado”) para uma aplicação específica.

Um banco de dados MySQL permite que um sistema possa persistir seus dados mesmo quando o processo que executa a aplicação no computador é encerrado. É muito utilizado em aplicações que armazenam dados de modo geral. Um banco de dados MySQL pode ainda oferecer ferramentas para realizar o levantamento de dados, estatísticas, indicadores e até mesmo gráficos, para tornar o tratamento de grandes quantidades de dados mais fácil e simples de se analisar. Neste projeto, o banco de dados foi essencial para que a aplicação possa realizar consultas mais complexas, buscando dados específicos dentro de determinadas condições passadas através de códigos SQL.

3.7. Modelo de Entidade Relacionamento (MER)

O Modelo Entidade Relacionamento (também chamado Modelo ER, ou simplesmente MER), como o nome sugere, é um modelo conceitual utilizado na Engenharia de *software* para descrever os objetos (entidades) envolvidos em um domínio de negócios, com suas características (atributos) e como elas se relacionam entre si (relacionamentos). [Rodrigues 2014]

Na figura 2 é mostrado o Modelo de Entidade Relacionamento do GreatIdea.

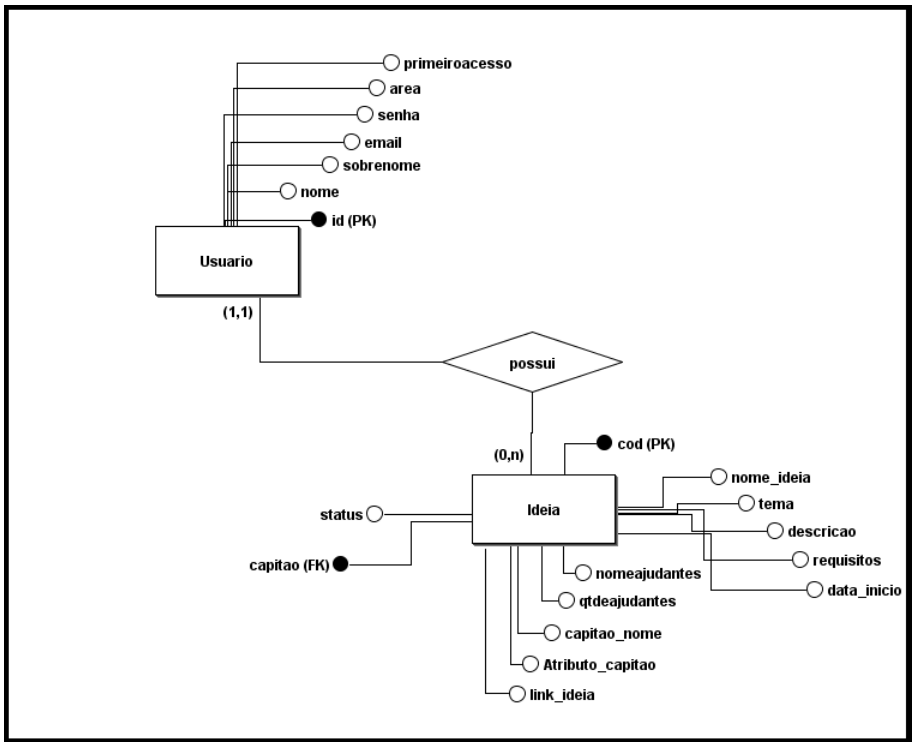

Figura 2. Modelo de Entidade Relacionamento.

3.8. Engenharia de *software*

Segundo [Sommerville 2013] engenharia de *software* é uma disciplina que se preocupa com todos os aspectos de produção de *software* garantindo um processo que seja eficiente do início ao fim de sua elaboração. A engenharia de *software* disponibiliza algumas diretrizes e ferramentas para garantir a tríplice restrição composta por Escopo, Prazo e Custo, onde pessoas podem aplicar essas ferramentas de forma a garantir a qualidade no resultado final.

4. Metodologia utilizada

Foi realizado o desenvolvimento da aplicação utilizando diretrizes da Engenharia de *software*, Orientação a Objetos e Padrões de Projeto sendo eles MVC e DAO, sendo iniciado pelo levantamento dos requisitos funcionais e não funcionais e elaboração dos diagramas de caso de uso e de classes respeitando as normas da UML, além da utilização da ferramenta Kanban durante todo o processo de desenvolvimento da aplicação. A aplicação foi desenvolvida voltada para o ambiente *Web* utilizando as seguintes linguagens de programação: HTML, CSS, JavaScript, além de um banco de dados MySQL.

5. Requisitos do GreatIdea

Os requisitos do sistema são as características que o sistema deve possuir para satisfazer a proposta inicial e cumprir o que foi solicitado pelo cliente ou idealizador em sua elaboração. Estes são classificados em **Requisitos Funcionais** e **Requisitos Não-Funcionais**.

Requisitos não funcionais são restrições aos serviços ou funções oferecidas pelo sistema. Incluem restrições de tempo, restrições no processo de desenvolvimento e restrições impostas pelas normas [Sommerville 2011]. A relação dos requisitos não funcionais (RNF) do aplicativo está detalhada na tabela de requisitos não funcionais.

Requisitos Não-Funcionais	
RNF - 01	Utilização de Tecnologias para desenvolvimento de *software* em ambiente *Web*.
RNF - 02	Interfaces que sigam um padrão lógico para facilitar a utilização de telas do sistema.
RNF - 03	Garantir a persistência dos dados utilizando tecnologias de armazenamento de dados como um banco de dados.
RNF - 04	Utilizar metodologias de desenvolvimento durante o processo de implementação da aplicação. (Ex: Metodologias Ágeis)

Tabela 1. Tablela de Requisitos Não-Funcionais da aplicação.

Requisitos funcionais são declarações de serviços que o sistema deve fornecer, de como o sistema deve reagir a entradas específicas e de como o sistema deve se comportar em determinadas situações [Sommerville 2011]. A relação dos Requisitos Funcionais (RF) do sistema e o caso de uso (*UserCase*) relacionado estão detalhados na tabela 2.

Requisitos Funcionais	
RF01 - Efetuar Login	Será disponibilizada uma seção para que o usuário possa efetuar login no sistema utilizando os dados "email" e "senha".
RF02 - Cadastrar Usuário	Caso o usuário não possua informações de login, poderá efetuar um cadastro no sistema utilizando seus dados pessoais para então estar apto a utilizar a aplicação.
RF03 - Cadastrar Ideia	O usuário poderá criar um novo projeto dentro da aplicação, este projeto deverá possuir informações sobre a ideia e deverá conduzir seu desenvolvimento, o mesmo ficará disponível para que outros usuários possam participar do processo de implementação.
RF04 - Dashboard: Lista de ideias	Será exibida uma lista de todas as ideias que estão aptas a receber novos colaboradores, os projetos que já contêm o número máximo de participantes não deverão aparecer na lista.
RF05 - Dashboard: Lista de ideias do usuário	Será exibida uma lista com os projetos do usuário que está utilizando a aplicação, estes projetos devem ser editáveis pelo autor da ideia.
RF06 - Dashboard: Editar ideias	Na lista de ideias do usuário, será possível selecionar o projeto para uma visualização mais detalhada e editar suas informações caso necessário.

Tabela 2. Tablela de Requisitos Funcionais da aplicação.

6. Diagrama de caso de uso

O Diagrama de caso de uso é uma das ferramentas que compõem a UML [Booch 2012], ele auxilia na modelagem de um produto de *software* possibilitando a construção do sistema de modo teórico ilustrativo, facilitando seu entendimento.

Segundo [Vazquez 2016], podemos dizer que um caso de uso é um documento narrativo que descreve a sequência de eventos de um ator que usa um sistema para completar um processo. Um caso de uso representa uma unidade discreta da interação entre um ator (humano, dispositivo ou outro *software*) e o sistema.

Neste estudo o diagrama de caso de uso foi importante para a definição mais clara de quais são as implementações necessárias para que a ferramenta possa contemplar todas as atividades as quais são esperadas que o usuário possa executar dentro da aplicação. Baseado no diagrama de caso de uso pode-se ainda elaborar os fluxos sistemáticos desta aplicação, que define quais são os passos esperados pelo desenvolvedor os quais o usuário deverá seguir para que os caso de uso possam ser contemplados, de forma a fazer com que o sistema possa ser intuitivo, funcional e simples de se utilizar.

Na figura 3 é possível visualizar o diagrama de caso de uso do sistema GreatIdea,

elaborado durante a fase de planejamento do *software* e utilizado como referência durante todo o processo de seu desenvolvimento.

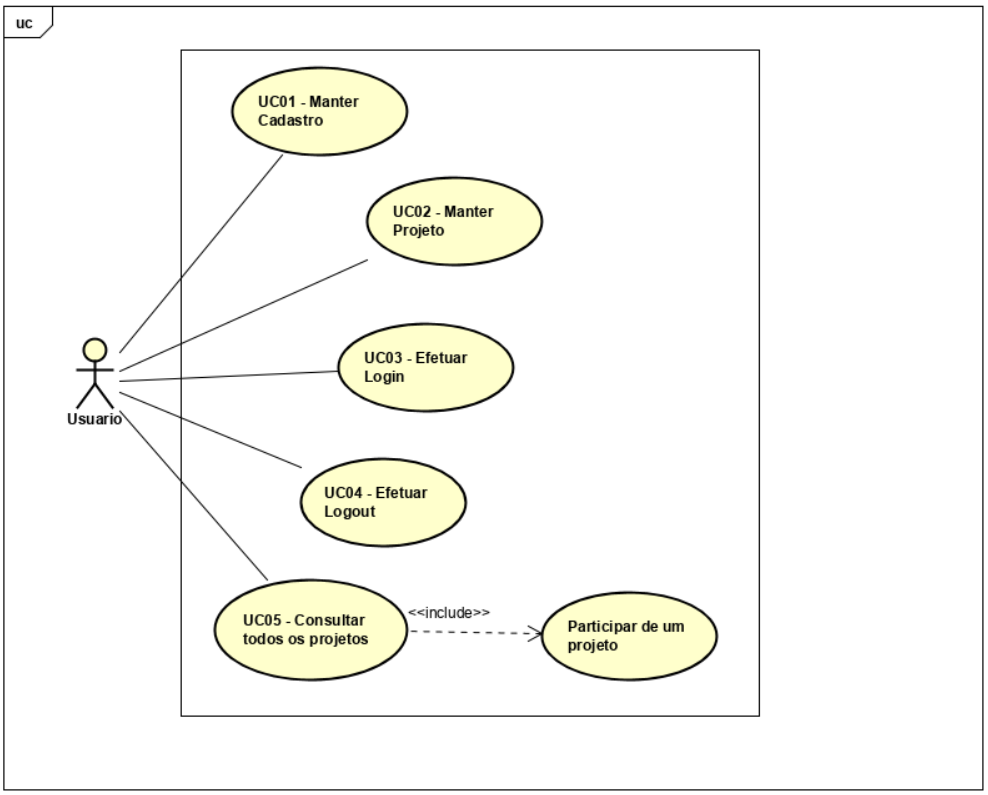

Figura 3. Diagrama de caso de uso do GreatIdea.

6.1. Diagrama de classes

Para Booch [Vazquez 2016] Um diagrama de classes é um diagrama que mostra um conjunto de classes, interfaces e colaborações e seus relacionamentos. São relacionamentos estruturais entre instâncias e especificam que objetos de uma classes estão ligados a objetos de outras classes. Desta forma, pode-se dizer que o diagrama de classes é a representação ilustrativa de como as classes de um sistema orientado a objetos estão distribuídas. O diagrama de classes pode representar:

- Classes;
- Associação;
- Dependência;
- Agregação;
- Composição;
- Generalização.

A figura 4 mostra o diagrama de classes elaborado e utilizado durante o processo de desenvolvimento da aplicação GreatIdea.

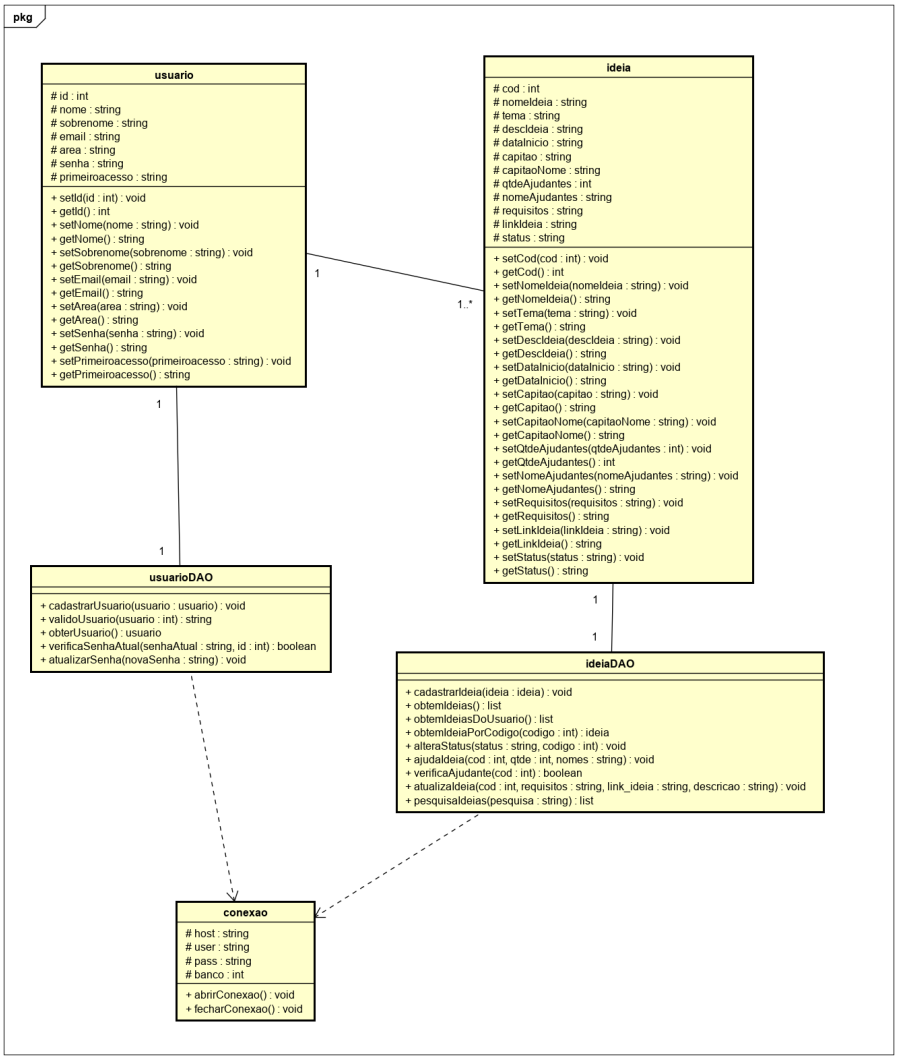

Figura 4. Diagrama de caso de uso do GreatIdea.

7. Sistema de Capitão e Ajudante

Ao criar um novo projeto dentro da plataforma GreatIdea, o usuário pode contar com a ajuda de outros membros da aplicação para desenvolver sua ferramenta, para que essa metodologia funcione, foi criado um sistema de hierarquia em que o usuário pode participar de uma ideia como Capitão e como Ajudante da mesma. Quando o usuário cria um projeto dentro da plataforma, ele é automaticamente definido como o capitão do mesmo atuando como autor da ideia, este usuário passa a ser o responsável por gerir o processo de desenvolvimento conforme suas necessidades, distribuir funções conforme a equipe formada e suas características, definir os requisitos, criar diretórios compartilhados para que os usuários possam acessar os recursos necessários, além de definir quando o projeto atingiu seu término e já pode ser concluído. A figura 5 mostra um exemplo de projeto utilizando a distribuição das funções de cada usuário, a figura mostra ainda as tecnologias que cada usuário têm grande afinidade, a fim de mostrar como a diversificação de habilidades dentre os integrantes do projeto podem ajudar a suprir todas as necessidades durante o processo de elaboração.

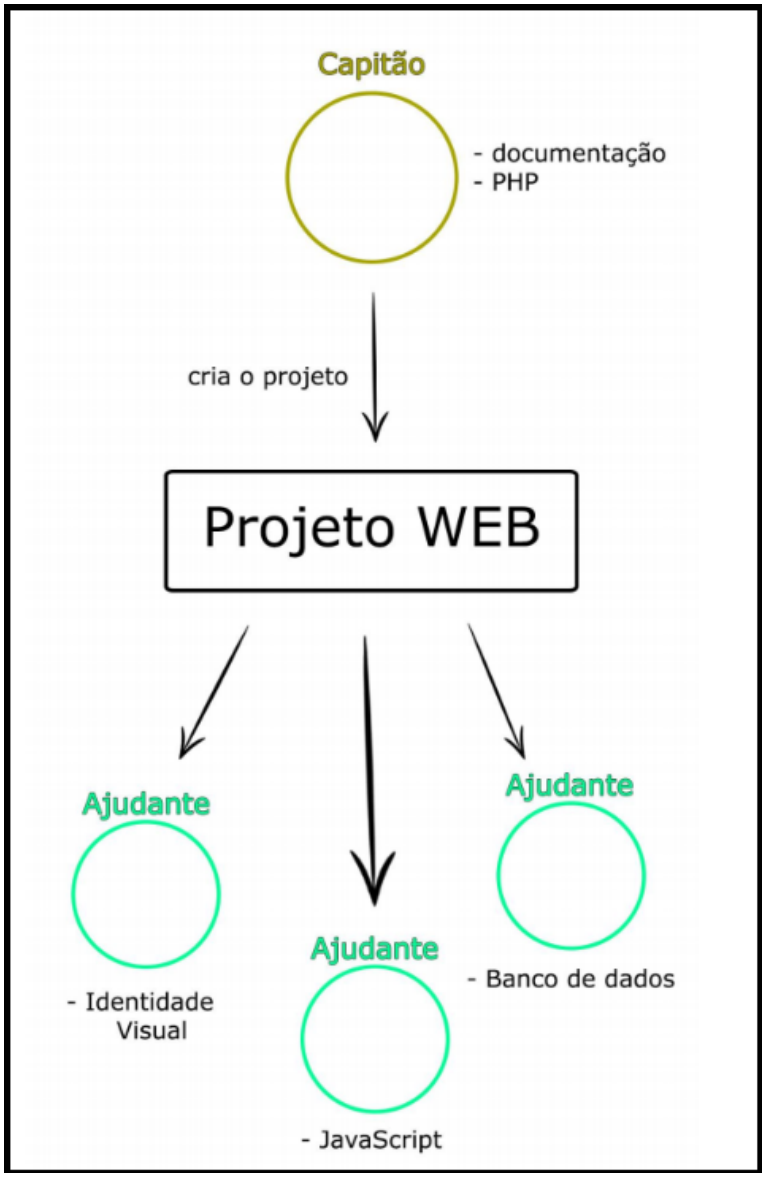

Figura 5. Ilustração de um projeto fictício com um usuário capitão e 3 usuários ajudantes participantes.

8. Fluxos do sistema

Fluxos do sistema compõem uma lista que definem as ações que são esperadas que um usuário realize durante a utilização de um sistema, definindo os passos e documentando-os. Abaixo temos dois dos principais fluxos de sistema da aplicação GreatIdea.

8.1. Fluxo 1 - Criar ideia

FP1. Usuário efetua *login* na aplicação GreatIdea; (UC03)
FA1. Usuário efetua cadastro na aplicação GreatIdea (UC01);
FP2. Usuário clica no botão "Criar ideia"; Aciona caso de uso Criar Ideia (UC02);
FP3. Usuário preenche os campos: Nome da ideia; Tema; Descrição (UC02);
FP4. Usuário Clicar no botão "Criar ideia". Exibe a mensagem "Ideia cadastrada com sucesso!"(UC02);

FA2. Usuário não preencheu todos os campos necessários para a criação da ideia. Exibe a mensagem "Este campo é obrigatório"no campo não preenchido (UC02).

8.2. Fluxo 2 - Participar de uma ideia

FP1. Usuário efetua *login* na aplicação (UC03);
FP2. Usuário Escolhe um projeto na *dashboard* e seleciona "Ver Mais"(UC05);
FP3. Usuário clica no botão "Quero participar!". Exibe a mensagem "Você agora é um ajudante desta ideia!"(UC05); FA1. O projeto está cheio ou com *status* concluído (UC05).

9. Funcionamento do sistema

9.1. Tela de *login*

Ao acessar a aplicação GreatIdea, o usuário se depara inicialmente com uma tela de *login*, onde será necessário inserir os dados de *login* e senha para acessar a aplicação, caso o usuário não possua cadastro na aplicação ainda, um formulário para cadastro também está disponível nesta tela para que o usuário possa se cadastrar e acessar os recursos da ferramenta, a tela de *login* aparece na imagem abaixo.

Figura 6. Tela de cadastro e *login* do sistema GreatIdea.

9.2. *Dashboard* com a lista de ideias disponíveis para participação

Depois de efetuar *login* na aplicação o usuário se depara com a tela principal da aplicação, que consiste em uma *dashboard* que contém uma lista com todos os projetos disponíveis na plataforma no momento do acesso, nesta tela é possível visualizar o título das ideias disponíveis, tema, nome de seu criador e a quantidade de pessoas que já estão participando do projeto em questão, além de um botão chamado "ver mais", que ao clicado exibe uma página detalhada contendo as informações relevantes do projeto. Neste local há ainda um menu lateral localizado a esquerda, com as seguintes opções:

- Início;
- Nova Ideia;

- Minhas Ideias;
- Editar Perfil;
- Sair.

Ao clicar na opção Início, o usuário é direcionado para a *dashboard* com a lista de todos os projetos. Logo abaixo temos a opção Novo Projeto, onde é possível realizar a criação de uma nova ideia na plataforma, que ficará associada ao usuário em questão; A opção Meus Projetos exibe uma lista com todos os projetos dos quais o usuário logado faz parte, seja como Capitão ou como Ajudante (Conforme explicado na seção 7); O botão Editar Perfil, permite que o usuário acesse seu perfil para visualizar informações pessoais e também para editar sua senha de acesso a plataforma. Por fim, a opção Sair desloga o usuário da plataforma e o direciona de volta a tela de *login*.

Figura 7. *Dashboard* da ferramenta listando todos os projetos disponíveis para participação.

9.3. Pesquisa de ideias por tema

Por fim, na interface da aplicação, é possível notar uma barra de pesquisa presente na parte superior da tela, através dela o usuário pode buscar ideias para participar de acordo com o seu tema de interesse, ao digitar um tema e pressionar o botão de busca, o retorno será uma lista de ideias que estão de acordo com o tema pesquisado pelo usuário.

9.4. Página detalhada da ideia

Ao selecionar a opção Ver Mais de uma ideia, o usuário se depara com uma tela contendo diversas informações sobre este projeto, além das informações já apresentadas na *dashboard*, o usuário pode encontrar a data de início do projeto, o nome dos membros participantes, uma breve descrição sobre a ideia, o *link* do projeto na *web* (ex: *link* do Github), seus requisitos e seu *status* atual. Quando o número de cinco participantes for atingido, o capitão do projeto pode selecionar o botão Começar! para alterar seu *status* e começar a inserir mais informações. A imagem abaixo mostra a tela de forma detalhada aplicada em um projeto de exemplo.

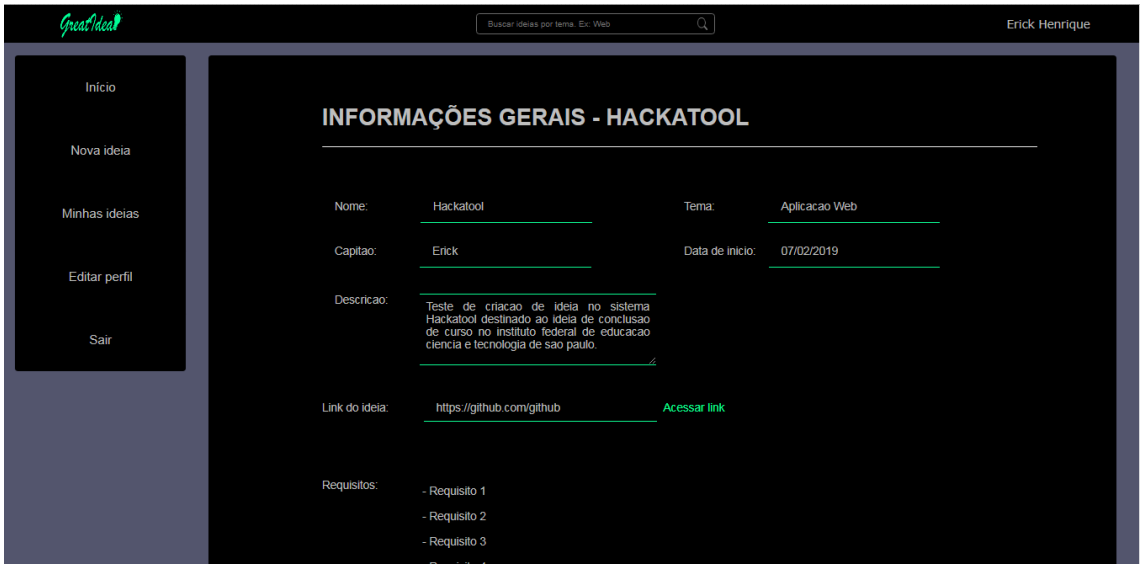

Figura 8. Tela contendo as informações detalhadas de um projeto de exemplo.

Essas informações podem estar incompletas em um primeiro momento, visto que o projeto se encontra com *status* Aguardando Ajudantes e está só começando.

9.5. Editar Perfil

Caso o usuário queira alterar sua senha de *login* ou visualizar suas informações pessoais cadastradas na aplicação, a opção Editar Perfil pode ser acionada de modo a exibir todas as informações relacionadas a este usúario. Caso o usuário faça a alteração da senha através desta opção, o mesmo será redirecionado para a tela de *login* para que efetue *login* novamente para que seja feita a autenticação da nova senha cadastrada sem a necessidade de uma confirmação via e-mail. Na página de editar perfil o usuário pode ainda atualizar suas competências, selecionando as tecnologias que gostaria de trabalhar ou que possuem maior afinidade, de forma com que os outros membros da plataforma possam conhecer melhor seu perfil de desenvolvedor.

Figura 9. Sessão para selecionar competências na tela de editar perfil.

10. Testes com Usúarios

Para realizar os testes da aplicação foram selecionados três discentes voluntários do câmpus e dois Analistas de *software* que atualmente trabalham em uma empresa de médio porte para atuar como usuários da ferramenta, seguindo um fluxo de telas para percorrer todas as funcionalidades do sistema. Ao início dos testes, todos os usuários foram convidados a fazerem um breve cadastro na ferramenta, preenchendo seus dados pessoais, em seguida efetuaram *login* na aplicação. Agora visualizando a tela inicial da ferramenta, os usuários foram incentivados a criarem novos projetos da aplicação, preenchendo dados como o nome da ideia, uma breve descrição sobre a mesma e a quantidade de participantes que deseja incluir em seu projeto. Em seguida, os usuários foram convidados a escolher uma ideia na tela inicial da aplicação para colaborar, os voluntários selecionaram a ideia de seu interesse, visualizaram seus detalhes na opção "ver mais"e selecionaram a opção "participar". Por fim, os usuários atualizaram suas competências selecionando quais tecnologias possui afinidade no menu "Editar Perfil". A tabela abaixo mostra as perguntas realizadas e as respostas dos voluntários:

Usuários	**Usaria para aprender novas tecnologias?**	**Usaria para desenvolver um projeto pessoal?**	**Você usaria o sistema novamente?**	**Recomendaria para um amigo?**
Aluno 1	Provavelmente sim	Provavelmente Sim	Sim	Talvez
Aluno 2	Sim	Provavelmente Sim	Provavelmente Sim	Sim
Aluno 3	Sim	Provavelmente Sim	Sim	Sim
Analista 1	Provavelmente Sim	Dificilmente	Provavelmente Sim	Sim
Analista 2	Provavelmente Sim	Provavelmente Sim	Sim	Sim

Tabela 3. Perguntas e respostas dos voluntários nos testes do GreatIdea

11. Conclusão

Este estudo propôs o desenvolvimento de um banco de ideias para o Instituto Federal de Educação, Ciência e Tecnologia de São Paulo - Câmpus Hortolândia, que até então não possuia um sistema que possa atuar como um banco de ideias para ajudar discentes e docentes a interagirem compartilhando e implementando ideias e criando soluções por meio do trabalho coletivo. Conforme seção 10 a aplicação GreatIdea possui grande aceitação por parte do público interno e externo a comunidade do câmpus, de forma a evidenciar a importância de uma ferramenta para centralizar e implementar projetos para estas pessoas.

O GreatIdea possibilita não somente o compartilhamento de ideias que possam vir a se tornar grandes projetos, mas também intermedia a participação de outras pessoas da comunidade nesse projeto, para que ele possa atingir de maneira mais consistente os resultados propostos inicialmente pelo autor da ideia.

12. Trabalhos Futuros

Após a conclusão deste trabalho, a aplicação *Web* GreatIdea poderá ser capaz de centralizar ideias da comunidade do câmpus e da comunidade externa, ela pode ainda facilitar o processo de implementação das ideias que possuem escopo e planejamento definidos de maneira a intermediar o seu processo de desenvolvimento auxiliando na formação de

equipes para trabalhar no projeto em questão. Para o futuro acreditamos que a aplicação GreatIdea pode ser aprimorada ainda mais com maior atenção na formação de equipes utilizando a tecnologia de Inteligência Artificial, essa tecnologia pode ajudar na formação desses times de desenvolvimento utilizando algoritmos que possam comparar os perfis dos usuários da plataforma baseados em suas atividades anteriores recentes, traçando assim os melhores perfis para atuar em cada projeto e atingir de forma mais satisfatória o resultado proposto por cada um.

Referências

Bernardo, K. (2014). Kanban: Do início ao fim!

Booch, J. (2012). *UML, Guia do Usuário*. Elsevier.

Canaltech (2015). 15 inovações tecnológicas surgidas nos últimos 15 anos.

Elmasri, R. (2010). *Sistemas de Banco de Dados*. Pearson Universidades.

Kent Beck, Mike Beedle, J. G. (2001). Manifesto for agile software development.

Lima, M. (2008). O que é just in time?

Mariotti, F. (2018). *Kanban: o ágil adaptativo - Revista Engenharia de Software Magazine 45*. Devmedia.

Paredes, A. (2019). As metodologias ágeis mais utilizadas e suas vantagens dentro da empresa.

Rodrigues, J. (2014). Modelo entidade relacionamento (mer) e diagrama entidade-relacionamento (der).

Softex (2019). Sobre o mps.br.

Sommerville, I. (2011). *Engenharia de Software*. Pearson Prentice Hall.

Sommerville, I. (2013). *Software Engineering 9*. Addison-Wesley.

Vazquez, C. E. (2016). *Engenharia de Requisitos: software orientado ao negócio*. Brassport.

Documento Digitalizado Público

Entrega do artigo do TCC do aluno Erick H. L. da Silva

Assunto: Entrega do artigo do TCC do aluno Erick H. L. da Silva
Assinado por: Daiane Tomazeti
Tipo do Documento: Declaração
Situação: Finalizado
Nível de Acesso: Público
Tipo do Conferência: Cópia Simples

Documento assinado eletronicamente por:

- **Daiane Mastrangelo Tomazeti**, **PROFESSOR ENS BASICO TECN TECNOLOGICO**, em 27/04/2020 13:51:15.

Este documento foi armazenado no SUAP em 27/04/2020. Para comprovar sua integridade, faça a leitura do QRCode ao lado ou acesse https://suap.ifsp.edu.br/verificar-documento-externo/ e forneça os dados abaixo:

Código Verificador: 399967
Código de Autenticação: 6d7ecda7da