**INSTITUTO FEDERAL DO PARANÁ**

**LUIS GUILHERME DE JESUS GOMES**

**SISTEMA DE SOLICITAÇÃO DE DIÁRIAS E PASSAGENS**

**PARANAVAÍ**

**2018**

**LUÍS GUILHERME DE JESUS GOMES**

**SISTEMA DE CONTROLE DE PRESTAÇÃO DE SERVICOS**

Trabalho da matéria de Projeto Integrador do Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas, como parte dos requisitos iniciais de avaliação.

Orientador: Frank Willian Cardoso de Oliveira

**PARANAVAÍ**

**2018**

**RESUMO**

Este trabalho visa apresentar as metodologias de desenvolvimento, bem como as ferramentas de software utilizadas no desenvolvimento do projeto integrador, que une todas as principais matérias do curso de Tecnologia em Analise e Desenvolvimento de Sistemas, em uma só, proporcionando uma maior integração e entendimento do aluno para com os conceitos aprendidos.

Palavras Chave: desenvolvimento, ferramentas, software.

**SUMÁRIO**

[1. INTRODUÇÃO 5](#_Toc513370784)

[2. MÉTODOS DE TRABALHO 6](#_Toc513370785)

[**2.1.** Scrum 6](#_Toc513370786)

[**2.2.** PROGRAMAÇÃO EXTREMA 7](#_Toc513370787)

[3. FERRAMENTAS 8](#_Toc513370788)

[**3.1.** NETBEANS 8](#_Toc513370789)

[**3.2.** GIT 10](#_Toc513370790)

[**3.3.** BITBUCKET 10](#_Toc513370791)

[4. PLATAFORMA, LINGUAGEM E FRAMEWORKS 10](#_Toc513370792)

[**4.1.** HISTÓRICO DO JAVA 10](#_Toc513370793)

[**4.2.** LINGUAGEM DE PROGRAMAÇÃO JAVA E JAVA EE 11](#_Toc513370794)

[**4.3.** HIBERNATE 12](#_Toc513370795)

[**4.4.** JAVA SERVER FACES 13](#_Toc513370796)

[**4.5.** PRIMEFACES 14](#_Toc513370797)

[5. MODELAGEM 15](#_Toc513370798)

[**5.1.** REQUISITOS FUNCIONAIS 15](#_Toc513370799)

[**5.2.** REQUISITOS NÃO FUNCIONAIS 17](#_Toc513370800)

[**5.3.** DIAGRAMA DE CASOS DE USO 19](#_Toc513370801)

[**5.4.** DIAGRAMA DE ENTIDADES 19](#_Toc513370802)

[**5.5.** DESCRIÇÕES DE CASOS DE USO 20](#_Toc513370803)

[**5.6.** DIAGRAMA DE CLASSES 24](#_Toc513370804)

[6. CONCLUSÃO 26](#_Toc513370805)

[REFERÊNCIAS 27](#_Toc513370806)

# INTRODUÇÃO

O desenvolvimento de software na atualidade ainda é uma tarefa tão complexa como a maioria das pessoas acredita ser, o desenvolvimento correto e organizado depende de dedicação da equipe de desenvolvimento como um todo.

Neste trabalho serão apresentados os métodos, ferramentas e documentos utilizados na construção do software, resultado do estudo realizado durante o segundo ano do curso de Tecnologia em Analise e Desenvolvimento de Software do Instituto Federal do Paraná, campus de Paranavaí.

A organização do trabalho está assim disposta: na Seção 2 descrevem-se os Métodos de trabalho como o Scrum e Programação Extrema; na seção 3 as Ferramentas, que se subdividem em Netbeans, GIT e BitBucket e a seção número 4 que trata de Plataforma, linguagens e frameworks que são apresentadas JavaEE, a linguagem de programação Java, o Hibernate, o Java Server Faces (JSF), e por fim, o Primefaces; na seção 5 apresentamos a modelagem do software, com todos os requisitos, diagramas e descrições de casos de uso.

# MÉTODOS DE TRABALHO

Com o passar do tempo às equipes envolvidas no desenvolvimento de software e realização de projetos, acabaram por definir metodologias que tem por objetivo agilizar a realização e conclusão das atividades.

## Scrum

Scrum é uma metodologia ágil que envolve uma equipe em um desenvolvimento de determinado projeto, as etapas são chamadas iterações ou sprints (que significa arrancada em inglês) as sprints, como o próprio nome sugere é uma etapa em que as tarefas devem ser realizadas com a maior rapidez possível.

Cada pessoa da equipe possui uma titulação, para objetivar as atividades realizadas por cada um, tem-se então o Scrum Master que atua como líder da equipe e é como um instrutor, tanto do Product Owner quanto do time.

Segundo Macedo e Sbrocco (2012) o Scrum não é utilizado somente em desenvolvimento de software, mas também pode ser aplicado no gerenciamento de qualquer trabalho, pois é iterativo incremental, que significa que o produto é construído em partes, ou seja, de forma iterada. As sprints devem ser muito bem planejadas, pois há a necessidade de estabelecer um tempo hábil para a execução de determinada tarefa a ser desenvolvida; esse planejamento é feito em reuniões que duram em torno de oito horas; o projeto é iniciado com o Product Owner que recebe as informações dos stakeholders e elabora uma lista de requisitos e cria um Backlog de Produto, que seria um “resumo” do trabalho realizado naquele tempo, mais conhecido como user story. No Scrum, temos os cerimoniais, que são: Sprint Planning que se trata de reunião de Planejamento, a Daily Scrum é nada mais que uma reunião pública onde todos os envolvidos podem participar, a Sprint Review acontece no último dia da Sprint onde todos são convidados e dura no máximo quatro horas, e a Sprint Retrospective que tem duração máxima de três horas e tem por objetivo identificar pontos de melhorias.

Para Pham e Pham (2011) mesmo que o Scrum e outras metodologias ágeis sejam difíceis de aplicar, se utilizadas de maneira correta geram resultados satisfatórios. Por mais trabalhosa que seja a utilização da metodologia para organização de todo o processo de desenvolvimento tanto de software como de outros projetos o Scrum é uma ferramenta essencial para a realização de um projeto desde seu planejamento até sua finalização.

## PROGRAMAÇÃO EXTREMA

Programação extrema é conhecida como XP (Extreme Programming) e Sbrocco e Macedo (2012, p.143) afirma que “A ideia básica é enfatizar o desenvolvimento rápido do projeto, visando garantir a satisfação do cliente, além de favorecer o cumprimento das estimativas”. este tipo de programação é considerada como uma metodologia ágil de desenvolvimento, a idéia é que a equipe envolvida seja pequena e que os feedbacks sejam constantes, visando a qualidade e comunicação eficiente, gerando por consequência a satisfação com os resultados. os papéis também são especificados, assim como no Scrum e no XP são Gerente de Projeto, o Coach, o Desenvolvedor, o Analista de teste, e o Redator técnico.

Existem várias etapas na realização desta metodologia, a primeira é chamada de História, onde se levantam os requisitos necessários junto ao cliente; nos Ciclos Semanais de estação são realizadas avaliações sobre o planejamento realizado, as histórias são reavaliadas e é feita uma previsão para os próximos requisitos; para que haja um acompanhamento de quem entende e do que precisa ser entregue, onde acontece o Envolvimento real com o cliente; a Implantação incremental que deve ser definida juntamente com o cliente que consiste em habilitar uma funcionalidade para ele, possibilitando o trabalho com a ferramenta.

A Programação em Pares que trata-se de dois profissionais desenvolvendo um código, onde um é o piloto que tem a tarefa de digitar o código e o segundo é o copiloto que tem a tarefa de identificar erros no trabalho e a Integração contínua, visa compartilhamento do trabalho entre os membros da equipe, o mais rápido possível, para que todos tenham a noção do que está sendo produzido.

# FERRAMENTAS

As ferramentas facilitam o trabalho de um desenvolvedor, pois visa o maior aproveitamento do tempo e segurança do processo, utilizamos essas tecnologias com mais frequência com o passar do tempo devido esses motivos, temos como exemplos o NetBeans, o Git e o BitBucket, nos tópicos a seguir será detalhado a forma como essas ferramentas atuam em um projeto de software, entre outros detalhes.

## NETBEANS

O Netbeans segundo Edson Gonçalves (2006, p.38) “é uma IDE de desenvolvi- mento integrado de código fonte aberto desenvolvida totalmente em java”, e compatível com várias plataformas como Windows, Linux, Solaris e Mac OS e possuindo vários recursos para desenvolvimento web, desktop e mobile ou móveis, segundo o autor permite o programador escrever, compilar e debugar seu código, e seu uso em empresas de produtividade alta não é apenas por economia.

Segundo Pedro Pinto (2014) a IDE suporta também as linguagens Java (principal), C, C++, PHP, Ruby, PHP, HTML5, Java Script, e várias outras.

De acordo com Edson Gonçalves (2006,p.38) “é um projeto fundado pela Sun Microsystems em Junho de 2000 que é seu maior patrocinador atualmente”, a IDE vem evoluindo a cada versão rapidamente tornando-se competitiva junto as melhores ferramentas de desenvolvimento Java do mundo. Segundo Edson (2006, p.38) “Com uma evolução rápida Encontramos atualmente diversas funcionalidades na IDE que vem atraindo cada vez mais desenvolvedores de toda parte do mundo”, nela há vários recursos e funcionalidades como um editor amigável de telas AWT/SWING, bem como suporte completo ao Java Enterprise Edition, integração com banco de dados, suporte a diversos tipos de Plug-ins para estender as funcionalidades da IDE, CVS, depurador de aplicativos e componentes como JSPS e EJBS, suporte para o desenvolvimento e utilização de Web Services, o editor possui também identação automática de código, sistema de auto completar e várias outras funcionalidades, vale mencionar que o mesmo contém recursos para controle de versões, suporte no editor para aplicações web, servlets, jsp, e aplicações visuais swing, plug-ins para UML, recursos CSS. Possuindo uma vasta biblioteca de desenvolvimento e uma documentação em português inclusive, sendo, portanto uma IDE popular e perfeita para programadores iniciantes.

## GIT

O GIT como informa seu próprio site é um sistema de controle de versões de arquivos gratuitos e de código aberto, que pode ser utilizado para a gerência de pequenos projetos ou para a direção de projetos de grandes proporções. Um dos diferenciais do GIT é seu sistema de ramificações onde pode- se criar vários ramos independentes entre si, e depois decidir se deseja compartilhá-los, mesclá-los ao seu ramo principal ou excluí-los.

Esse sistema ainda traz como vantagem sua distribuição criptografada, que possibilita que cada um dos desenvolvedores tenham uma cópia do servidor principal, e que ninguém de fora do projeto possa acessar os dados gerando uma maior segurança, além de que cada desenvolvedor terá um backup do servidor, guardado localmente, e que a cada mudança poderá ser salvo, substituindo a versão principal.

## BITBUCKET

O BitBucket é um serviço de hospedagem online para projetos de software que utilizam GIT. Nesse sistema só é preciso realizar requests push e pull para atualizar os repositórios (web e local), ele funciona basicamente para hospedar e gerenciar os códigos do desenvolvimento na plataforma (GIT), até então podendo trabalhar sozinho ou em até 5 pessoas gratuitamente e online, sem precisar baixar e instalar aplicações.

# PLATAFORMA, LINGUAGEM E FRAMEWORKS

A escolha da plataforma de desenvolvimento de um software é indispensável, e assim a plataforma e o framework devem funcionar satisfatoriamente com a linguagem, enquanto isso a realidade atual facilita cada vez mais os processos que envolvem o desenvolvimento dos projetos de software.

## HISTÓRICO DO JAVA

Com a evolução, computadores que ocupavam salas e que custava milhões, se tornavam cada vez mais acessíveis. Nesta época as linguagens de programação estruturadas eram bastante utilizadas, como exemplo temos o basic, assembly, cobol entre outras. O termo programação orientada a objetos, estava pouco conhecido entre os programadores, e com o forte crescimento da linguagem de programação C/C++, trouxe uma revolução para época, pois C era uma linguagem apenas estruturada, e o C+

+ possibilitou a orientação de objetos.

Deitel (2010) afirma que Java é uma linguagem de programação fortemente orientada a objetos, com a popularização de computadores pessoais, modificaram a vida de todos na época, desde pessoas, as grandes corporações, tendo essa visão a Sun Micro Systems, financiou um projeto de uma nova linguagem de programação orientada a objetos baseada em C/C++, e a batizou com o nome de Oak em homenagem a árvore de carvalho que via todos os dias de sua sala na empresa. Mais tarde descobriram que já havia uma linguagem com esse nome.

Segundo Deitel (2010) a equipe da Sun Micro Systems, após visitar uma cafeteria, ao ouvirem o nome Java (cidade que forneciam um café importado), e considerou interessante, consequentemente o nome foi definido pela equipe, e que esta linguagem seria chamada de Java. No início eles investiram em dispositivos eletrônicos, mas o mercado não aceitou como a equipe esperava e com isso a crise chegou à empresa.

Ainda segundo Deitel (2010) a internet evoluiu a cada dia e a Sun viu uma oportunidade para o uso da linguagem na web proporcionar mais interatividade na internet, com animações e tornado páginas mais dinâmicas, deu um novo fôlego ao Java onde foi anunciado na conferência em 1995 da empresa. A linguagem Java foi de fácil entendimento e os programadores aderiram, e logo Java se espalhou em vários dispositivos, e estava em grandes servidores de várias corporações.

## LINGUAGEM DE PROGRAMAÇÃO JAVA E JAVA EE

Com a linguagem de programação Java, pode-se trabalhar com várias situações, por ser uma linguagem versátil a mesma possui três principais versões, que são:

* + - Java SE (Java Standard Edition), a versão padrão, com ambiente de programação e as principais bibliotecas do Java;
    - Java EE (Java Enterprise Edition), Java para aplicações de grande porte, voltada para corporações, e desenvolvimento web;
    - Java ME (Java Mobile Edition), Java para dispositivos mobiles;



Primeiramente para começar a programar em Java há necessidade do uso de uma IDE (Integrated development Environment), ambientes de programação integrado com a linguagem, e na máquina utilizada a versão do Java JDK ( Java development Kit), kit para desenvolvimento em Java, que é basicamente a linguagem de programação.

O Java EE sua maior diferença do SE, são as aplicações webs, onde o Java é executado em um servidor robusto, onde deve suportar várias conexões, e não permitir que o sistema das grandes corporações tenham problemas, o Java pode trabalhar em conjunto com o Tomcat, o Java Server Faces e muitos outros frameworks.

Exemplo de estrutura da linguagem:

public class HelloWorld{

public static void main(String[] args){ System.out.println("Hello, World");

}

}

Sierra e Bates apresenta que as aplicações java por ser executadas na JVM (Java Virtual Machine), torna-se uma linguagem de programação multiplataforma, pois se estiver instalado na máquina de trabalho ou pessoal , independentemente do sistema, todas as aplicações Java funcionarão, e ainda a aplicação pode ser portada por outro sistema sem a necessidade de alteração do código.

## HIBERNATE

Devido a dificuldade de trabalhar com aplicações orientadas a objetos e banco de dados relacionais e a demanda por desenvolvimento mais rápido no mercado, segundo Izalmo Primo(2016) surgiram várias soluções para facilitar a integração entre elas e os banco de dados relacionais e resolver este problema bem como poupar o programador de muito trabalho manual, para Primo a principal e mais popular solução foi o Hibernate que é uma ferramenta ou framework de mapeamento de objeto relacional ORM object relational mapping que ajuda a persistir objetos java em um banco de dados relacional, ele faz acesso ao banco e gera todos os códigos SQL necessários.

Segundo (Primo, 2016) o Hibernate foi iniciado por uma equipe de desenvolvedores sob liderança de Gavin King de forma autônoma para resolver os problemas de persistência de dados ORM no EJB 2.0 que era muito complicado e complexo, teve sua primeira versão divulgada em 2004 e após isso o projeto tornou-se vasto e popular demais, e Gavin King aceitou entrar para a JBoss Group que atualmente o mantém sob a licença de software livre LGPL de gratuidade, a funcionalidade dele. Segundo (Izalmo, 2016) consiste em mapear um modelo de objetos para um modelo de dados relacional baseado em um esquema Entidade/Relacionamento ele faz o mapeamento das classes Java para tabelas do banco de dados e dos tipos de dados Java para os tipos de dados SQL e também facilita as consultas e retorno de dados.

Para Izalmo Primo(2017) ”A meta do projeto Hibernate é aliviar os desenvolvedores de 95% das tarefas comuns de programação relacionadas à persistência”, ele eliminou os trabalhos manuais com ORM dos dados feitas com SQL e JDBC, em outras palavras para o autor o Hibernate é uma ferramenta de consulta e persistência se tornando uma das soluções melhores que existe poupando trabalhos cansativos e manuais desta parte como a codificação manual com SQL e o JDBC, ele acessa e intermédia entre o banco e aplicação e gera os comandos SQL necessário para a aplicação, faz a conversão entre registros e objetos, já que o objetivo da ferramenta é

eliminar o desenvolver da maior parte dos trabalhos manuais relacionados a persistência de dados ou seja a parte da integração dos software com banco de dados.

Portanto, para Primo (2016) o Hibernate localiza-se entre a aplicação e o banco de dados intermediando e fornecendo também consultas e facilidades para retorno dos dados, isso acaba reduzindo significativamente o tempo no desenvolvimento dos softwares eliminando as operações manuais dos dados construídos com SQL e JDBC, ele permite também a portabilidade da aplicação tornando-a portável para qualquer banco de dados SQL.

## JAVA SERVER FACES

Java Server Faces também conhecido como JSF, foi criado com a intenção de facilitar a produtividade no desenvolvimento de interfaces visuais, é um framework baseado em componentes de interface com o usuário, e estruturado na arquitetura do padrão de projeto MVC (Model View Controller). Os componentes são a base para a construção da interface com o usuário, são demonstrados por meios de tags (Etiquetas) em um JSP, e logo após “transformados” em HTML. O JSF é um Framework dirigido a eventos, como por exemplo, a chamada de um método, este framework é considerado um padrão de mercado na atualidade e está presente na especificação JEE 5.0.

O Java Server Faces é fácil de usar e produtivo, pois de modo geral há uma organização na divisão de funções na construção das aplicações Web.

É composto de uma tag padrão, que é a letra h. Exemplo de sintaxe do JSF:

<h:body>

<h:title>Página</h:title>

</h:body>

## PRIMEFACES

O PrimeFaces é para o profissional que busca alta qualidade visual e alta produtividade, pois ele é considerado uma biblioteca de componentes para aplicações com JSF, ou seja, a ferramenta possui uma ampla quantidade de componentes, desde botões, campos de texto em geral, agregando também o Ajax, que é um plug-in integrado, que facilita o desenvolvimento, não havendo a necessidade do programador configurar os componentes para o Ajax, o próprio PrimeFaces já efetua a configuração deste componente.

Os componentes são compostos de uma tag padrão, que é a letra p. Exemplo sintaxe do PrimeFaces:

<p:body></p:body>

A grande vantagem na utilização da plataforma é que possibilita uma maior opção de componentes estilizados e possui um padrão de componentes dos mesmos.

# MODELAGEM

A modelagem de um sistema funciona como a construção da planta de uma casa, a cada passo é definida uma parte da estrutura do projeto, com a modelagem de um software é possível construí-lo de forma a evitar erros, desperdício de tempo, desvios de funcionalidade entre outras anomalias que poderiam aparecer ao programar sem ter um “mapa” para seguir.

A modelagem desse projeto contem os itens:

* Requisitos Funcionais;
* Requisitos Não Funcionais;
* Descrição de Caso de Uso;
* Diagrama de Classes;
* Diagrama de Entidade Relacional.

Itens responsáveis por melhorar o entendimento e definir uma estrutura a seguir durante todo o desenvolvimento do sistema.

## REQUISITOS FUNCIONAIS

Um requisito de sistema de software especifica uma função que o sistema deve ser capaz de realizar. Esses requisitos são apresentados de acordo com a solicitação do usuário**.**

|  |  |
| --- | --- |
| **RF001** | O sistema deve ser capaz de cadastrar um usuário (Servidor ou colaborador eventual), exigindo nome completo, matrícula SIAPE, e-mail, log-in e senha diferenciando suas permissões de acesso ao sistema. |
| **RF002** | O sistema deve ser capaz de exibir as telas de dados de acordo com o privilégio do usuário logado; |
| **RF003** | O software deve possuir um formulário de solicitações de diárias e passagens para ser preenchido e enviado pelo usuário logado com campos obrigatórios como dados do solicitante, motivo, horários de saída e chegada, tipo de transporte, e outros campos como, por exemplo, o de permissão de inclusão de anexos; |
| **RF004** | O software deve exibir um formulário com possibilidade de upload de arquivos, |

|  |  |
| --- | --- |
|  | para posterior análise na prestação de contas da viagem aprovada; |
| **RF005** | O software deve apresentar um formulário de relatório da viagem feita pelo  servidor para preenchimento, com campos como: saída, chegada, datas, percurso, descrição da viagem e demais campos necessários e obrigatórios; |
| **RF007** | O software deve apresentar um formulário de notas técnicas da viagem feita pelo servidor para preenchimento permitindo anexos ,com motivos da viagem, compatibilidade da qualificação do servidor com a natureza da atividade, entre outros; |
| **RF008** | O sistema deve permitir que o usuário administrador tenha acesso a todas as telas e possa visualizar ações de todos os outros usuários, bem como aprovar a solicitações ou recusar; |
| **RF009** | O sistema deve permitir a visualização do formulário de solicitações por outro usuário de maior privilégio; |
| **RF010** | O sistema deve ser capaz de arquivar as solicitações aprovadas em um relatório  de solicitações de diárias e passagens aprovadas; |
| **RF011** | O sistema deve ser capaz de arquivar as solicitações recusadas para um  relatório de solicitações recusadas; |
| **RF012** | O sistema deve permitir ao usuário logado a checagem do status da sua solicitação, se negada ou aprovada, caso negada, esclarecimentos sobre motivos da negativa; |
| **RF013** | O software deve identificar cada solicitação enviada como um código de proposta e seu status. |
| **RF014** | O sistema deve poder apresentar um relatório de todas as viagens aprovadas e  seus servidores associados a ela; |

## REQUISITOS NÃO FUNCIONAIS

Em engenharia de sistemas de software, um requisito não funcional é aquele que descreve como sistema deve realizar algo ou alguma ação.

|  |  |
| --- | --- |
| **RNF001** | O sistema deverá ser online, acessado pela web de qualquer computador ou  dispositivo conectado a internet; |
| **RNF002** | O software deverá ser desenvolvido em linguagem JAVA EE; |
| **RNF003** | O sistema deverá ser modelado utilizando UML; |
| **RNF004** | O software deve identificar/autenticar todos os seus usuários com login e senha; |
| **RNF005** | O sistema deve diferenciar os níveis de acesso (ADMINISTRADOR e SERVIDOR) de informação aos usuários; |
| **RNF006** | O sistema deve bloquear o acesso após três tentativas de login sem sucesso; |
| **RNF007** | O software deve identificar/autenticar todos os seus usuários com login e  senha; |
| **RNF008** | O sistema deve associar um status a toda proposta encaminhada visível para o solicitante, em quatro categorias: enviada, em análise, aprovada e recusada. |
| **RNF009** | Cada solicitação deve ter um número identificador; |
| **RNF010** | O sistema deverá estar disponível vinte e quatro horas por dia; |

## DIAGRAMA DE CASOS DE USO

O diagrama de casos de uso mostra como são distribuídas as funções através de um “mapa”, com opções (extend) e obrigações (include) a cada passo tomado.

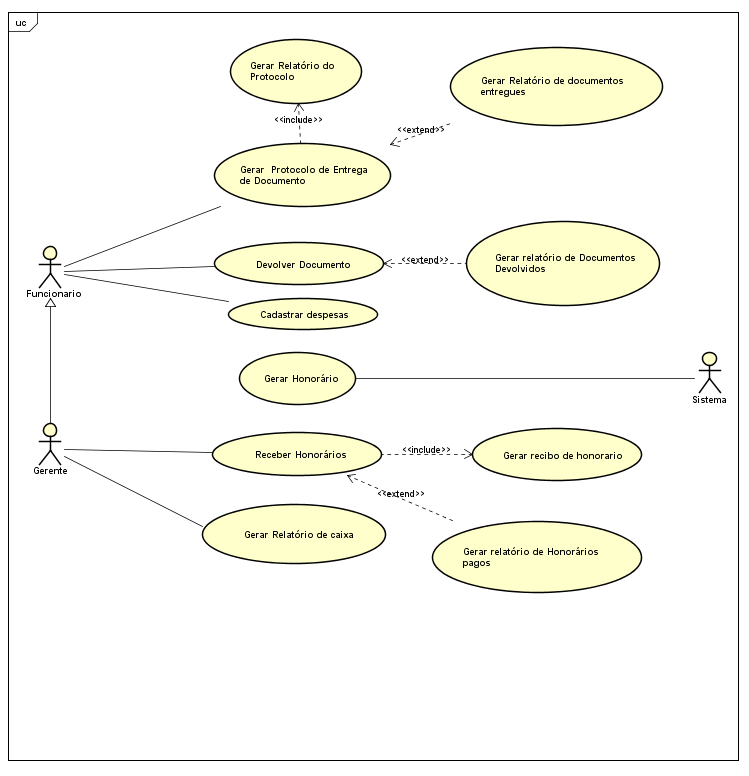


Figura 1 diagrama de casos de uso

## DIAGRAMA DE ENTIDADES

O diagrama de entidades mostra como estão dispostas as tabelas, ou entidades no banco de dados e os relacionamentos entre as tabelas.

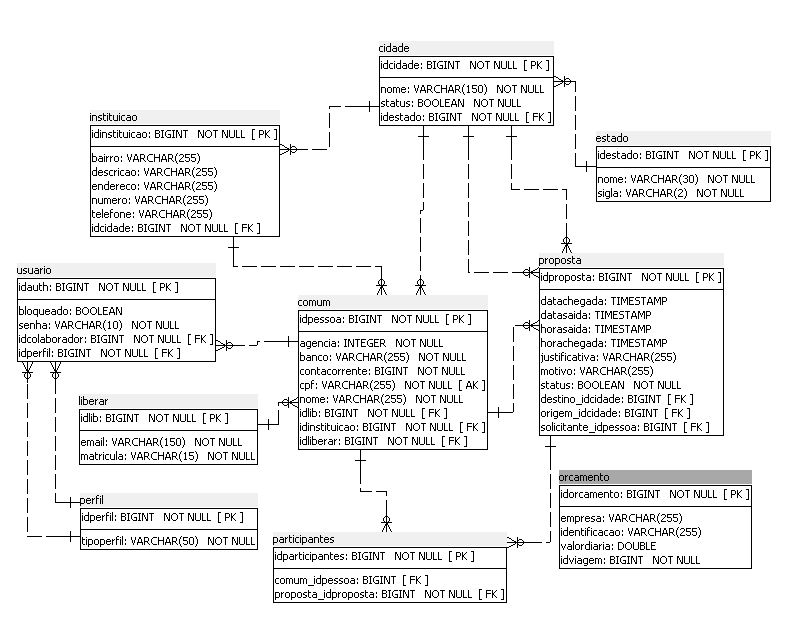


Figura 2 diagrama de entidades

## DESCRIÇÕES DE CASOS DE USO

Em cada descrição de caso de uso é detalhado como irá funcionar cada caso de uso em si, e como as ações podem ser tomadas.

* + 1. Cadastrar Colaborador

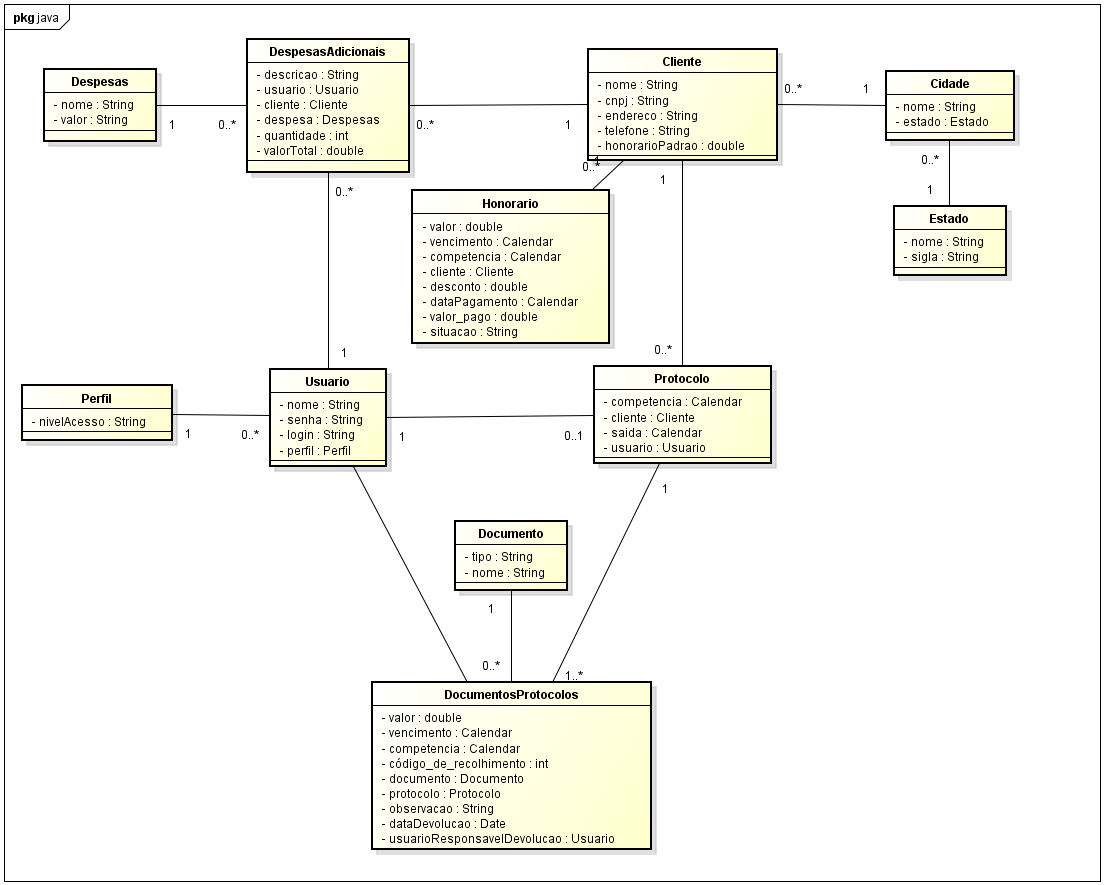
|  |  |
| --- | --- |
| UC001-  Cadastrar (Colaborador) |  |
| Objetivos: | Permitir o cadastro de servidor ou colaborador, cada qual com seu id, nome, e-mail, matricula, cargo, login e senha. |
| Requisitos: | RF001 |
| Atores: | Funcionário autorizado, admistrador. |
| Pré-condições: | O usuário deverá estar logado no sistema |
| Frequência de uso: | constante |
| Pós-condições: | O sistema deverá seguir online após o término do cadastro |
| Campos: | Código, nome, email, Matricula SIAPE(Caso Servidor), cpf, rg, telefone, endereco, cargo, login e senha |
| Fluxo principal: | FP1   1. O usuário faz log-in e acessa “cadastro de Servidor” 2. O sistema exibe a listagem de Servidores cadastrados 3. O usuário seleciona um servidor para consultar 4. O sistema exibe os dados do servidor consultado 5. fim do caso de uso |
| Fluxo alternativo: | A.1   1. No passo 2 do fluxo principal o funcionário clica no botão “Novo Servidor” 2. O sistema exibe a tela de cadastro de Servidor e solicita o preenchimento dos campos obrigatórios 3. O Usuário preenche os campos requeridos e Salva 4. O sistema valida os dados preenchidos 5. O sistema volta ao passo 2 do fluxo principal F1   A.2   1. No passo 2 do fluxo principal o funcionário clica no botão “Alterar Colaborador” 2. Sistema exibe a tela com os campos preenchidos 3. Usuário altera uma informação e salva 4. O sistema valida os dados preenchidos 5. Sistema volta ao passo 2 do FP1   A.3   1. No passo 2 do fluxo principal o funcionário clica no botão “Excluir Servidor” 2. Sistema exibe a tela perguntando se desejar realmente excluí o Servidor 3. Usuário confirma 4. O sistema volta ao passo 2 do FP1 |
| Fluxo de exceção: | E1 No passo c do fluxo alternativo A1 O usuário deixou de preencher algum campo obrigatório  a) O sistema desabilita o botão salvar e emite uma mensagem de aviso |
|  | b) volta ao passo c do fluxo alternativo A1 c)O sistema volta ao passo 2 FP1 |
| Fluxo de exceção: | E2 No passo c do fluxo alternativo A2 usuário digitou um valor não válido para o campo   1. O Sistema exibe uma mensagem de aviso 2. Usuário altera a informação e clica em salvar 3. O sistema valida os dados preenchidos 4. O sistema volta ao passo 2 FP1 |
| Validações | Os campos nome, cpf, rg, cargo, telefone, endereço devem ser preenchidos obrigatoriamente |

* + 1. Emitir Proposta

|  |  |
| --- | --- |
| UC003-Emitir Proposta |  |
| Objetivos | Permitir cadastro e emissão da proposta |
| Requisitos: | RF003 |
| Atores: | Usuários |
| Pré-condições: | O usuário deverá estar logado no sistema |
| Frequência de uso: | Constante |
| Campos: | Id, solicitante, origem, destino, data saída, data chegada, hora de saída e hora de chegada, tipo de transporte, orçamento. |
| Fluxo principal: | FP1  1) O usuário loga e acessa “Nova Proposta de Viagem” |
|  | 1. O sistema exibe as propostas feitas anteriormente bem como seus status pertinentes 2. O usuário seleciona uma proposta para consulta 3. O sistema exibe os dados da proposta consultada 4. fim do caso de uso |
| Fluxo alternativo: | A1  a) no passo 2 do fluxo principal o funcionário clica no botão “Nova Proposta”  b)O sistema exibe a tela com o formulário e solicita o preenchimento dos campos obrigatórios   1. O Usuário preenche os campos requeridos e salva a proposta 2. O sistema valida os dados preenchidos e altera o status da proposta para “enviada” 3. O sistema volta ao passo 2 do FP1   A2   1. No passo 2 do fluxo principal o funcionário clica no botão “Alterar Proposta” 2. Sistema exibe a tela com os campos preenchidos 3. Usuário altera uma informação e salva 4. O sistema valida os dados preenchidos   d) sistema volta ao passo 2 do FP1  A3   1. No passo 2 do fluxo principal o funcionário clica no botão “Excluir ou cancelar proposta” 2. Sistema exibe a tela perguntando se desejar realmente excluir a proposta 3. O usuário confirma a exclusão 4. O sistema volta ao passo 2 do FP1 |
| Fluxo de exceção: | FE1 No passo b do fluxo A1 o usuário deixou de preencher algum campo obrigatório   1. O sistema desabilita o botão salvar e destaca os campos não preenchidos 2. O usuário preenche os campos 3. O sistema valida os dados novamente 4. O sistema volta ao passo 2 FP1   FE2 No passo c do fluxo A1 uma proposta em análise já existe no sistema   1. O sistema emite uma mensagem de aviso 2. O Usuário visualiza e confirma 3. O sistema fecha a tela de proposta 4. O sistema volta ao passo 2 do FP1 |
| Validações: | O preenchimento dos campos ID, solicitante, CPF, data nascimento, data viagem, data chegada, hora chegada, hora saída, orçamento. |

## DIAGRAMA DE CLASSES

Este diagrama mostra como as classes estão organizadas e como conversarão entre si, classes estas que são as representações dos objetos reais que o software vai abranger e controlar, para apresentar as funcionalidades desejadas.



**Figura 3 Diagrama de Classes**

# CONCLUSÃO

Conhecer as ferramentas, as tecnologias que serão utilizadas, e de grande valia para o desenvolvimento do projeto, evitando o gasto de tempo para determinar o que vai ser utilizado no projeto. Entre outra, os stackholders que estão no projeto terem um padrão e cada um respeitar o que o grupo esta utilizando.

# 

# REFERÊNCIAS

Controle de Versão com GIT e BitBucket Disponível em:

<<http://tecnologiadainformacaoaplicada.blogspot.com.br/2014/04/controle-de-versao-com-git-e-> bitbucket.html> Acesso em: 09 de Março de 2017.

DEITEL, Harvey M.; DEITEL Paul j. Java como programar 8a edição. São Paulo: Pearson Prentice Hall 2010.

Dominando Netbeans de Edson Gonçalves, editora CIÊNCIA MODERNA. Acesso em: 16 de Março de 2017.

Desenvolvendo com Hibernate disponível em:

<<http://www.devmedia.com.br/artigo-java-magazine-73-desenvolvendo-com-hibernate/14756>> Acesso em: 25 de Fevereiro de 2017.

Introdução ao JSF e Primefaces. Disponível em:

<<http://docente.ifrn.edu.br/fellipealeixo/disciplinas/tads-2012/desenvolvimento-de-sistemas-> corporativos/book/book-jsf-3.pdf/>. Acesso em: 09 de Março de 2017.

Netbeans disponível em: <https://pt.wikipedia.org/wiki/NetBeans>. Acesso em: 24 de Março de 2017.

Netbeans um dos melhores IDES de programação disponível em:

<https://pplware.sapo.pt/linux/chegou-o-netbeans-8-um-dos-melhores-ides-para-programacao/> Acesso em: 25 de Fevereiro de 2017.

Persistindo os dados com Hibernate disponível em:

<https:[//www.caelum.com.br/apostila-vraptor-hibernate/persistindo-os-dados-com-o-hibernate/#4-](http://www.caelum.com.br/apostila-vraptor-hibernate/persistindo-os-dados-com-o-hibernate/#4-) 2-sobre-o-hibernate> Acesso em: 25 de fevereiro de 2017.

PHAM, Andrew; PHAM, Phuong-Van. Scrum em ação: gerenciamento e desenvolvimento ágil de projetos de software. São Paulo: Novatec 2011.

SBROCCO, Jose Henrique Teixeira de Carvalho; MACEDO, Paulo Cesar de. Metodologias Ágeis: engenharia de software sob medida. São Paulo: Érica 2012.

Scrum para times – As cerimônias do Scrum, Alexandro Marques. Acesso em: 16 de Março de 2017.

SIERRA, Kathy; BATES Bert, Use a Cabeça Java. Rio de Janeiro: Altas Books Editora 2010.

Sobre o GIT disponível em: <https://git-scm.com/about> Acesso em: 09 de Março de 2017.

Plataforma Java Disponível em: <https://pt.wikipedia.org/wiki/Plataforma\_Java> (acesso em: 16/03/2017).