****

**Proposta de Especificação de Software**

**Sistema para Gerenciamento de RH- Prefeitura de Cônego Marinho**

**Acadêmicos: Igor Guedes Ribeiro e Jardel Lopes Ferreira**

**Januária/MG**

**Novembro 2015**

****

**Proposta de Especificação de Software**

**Sistema para Gerenciamento de RH- Prefeitura de Cônego Marinho**

Este documento apresenta a proposta do Sistema para Gerenciamento de RH- Prefeitura de Cônego Marinho, que tem como principal função controlar e gerenciar os dados referentes aos funcionários e das secretarias da prefeitura. Para a construção do software foi feita a análise de requisitos, aplicação de técnicas de modelagem do esquema e das classes.

**Januária/MG**

**Abril 2015**

Sumário

[**1. Introdução 4**](#_Toc436824640)

[**1.1 Introdução à situação do problema 4**](#_Toc436824641)

[**1.2 Mini-Mundo 4**](#_Toc436824642)

[**1.3 Requisitos 4**](#_Toc436824643)

[**1.3.1 Requisitos Funcionais, Não-Funcionais e Suplementares. 4**](#_Toc436824644)

[**1.3.2 Requisitos Suplementares 8**](#_Toc436824645)

[**1.4 Objetivos 9**](#_Toc436824646)

[**1.4.1 Objetivos Gerais 9**](#_Toc436824647)

[**1.4.2 Objetivos Específicos 9**](#_Toc436824648)

[**2. Analise 10**](#_Toc436824649)

[**2.1 Diagrama Conceitual 10**](#_Toc436824650)

[**2.2 Diagrama Lógico do Banco de Dados 11**](#_Toc436824651)

[**2.3 Diagrama de Pacotes 12**](#_Toc436824652)

[**2.4 Diagrama de Classes 12**](#_Toc436824653)

[**2.5 Diagrama de Casos de Uso 13**](#_Toc436824654)

[**2.6 Descrição de Casos de Uso 14**](#_Toc436824655)

[**2.7 Diagrama de Estados 21**](#_Toc436824656)

[**2.8 Diagrama de Sequencia 23**](#_Toc436824657)

[**2.9 Diagrama de Implantação e Componentes 23**](#_Toc436824658)

[**3. Desenvolvimento 24**](#_Toc436824659)

[**3.1 ALM- Ambiente ALM 24**](#_Toc436824660)

[**3.1.1 Linguagem 24**](#_Toc436824661)

[**3.1.2 IDE 24**](#_Toc436824662)

[**3.1.3 Controle de versão 24**](#_Toc436824663)

[**3.1.4 SGBD 24**](#_Toc436824664)

[**3.1.5 Ferramenta de Issue/Bug Tracking 25**](#_Toc436824665)

[**3.1.6 Criação de Diagramas UML 25**](#_Toc436824666)

[**3.2 Arquitetura 25**](#_Toc436824667)

[**3.3 APIs 25**](#_Toc436824668)

[**3.4 Protótipo de Telas 26**](#_Toc436824669)

[**4. Referências 28**](#_Toc436824670)

# Introdução

# Introdução à situação do problema

Considera-se como “Relações Humanas” (RH), todo e qualquer tipo de relação social ou interação entre as pessoas. Decisivamente, em qualquer profissão o ser humano precisa estar se relacionando entre si, e a comunicação é o meio fundamental para que este relacionamento aconteça. As RH não estão só vinculadas às relações entre pessoas, estão também vinculadas ao ambiente de trabalho, onde precisam adquirir uma atitude mais formal, pois é necessário lidar com hierarquias.

O setor de RH é responsável pelo envolvimento de todos os setores da empresa, com a intenção de alcançar a organização. Dessa forma é fundamental que o responsável pelo setor de RH tenha um amplo conhecimento da empresa e suas políticas e tenha também um bom relacionamento com todos que fazem parte da organização. O setor é ainda responsável pelo recrutamento e seleção de futuros funcionários.

## Mini-Mundo

A prefeitura municipal de Cônego Marinho precisa desenvolver um sistema gerenciador de recursos humanos. A prefeitura possui várias secretarias, cada secretaria possui um endereço diferente e único. Um funcionário pode está alocado em somente uma secretaria, o funcionário pode ser contratado ou efetivo, caso o funcionário tenha dependentes ele recebe um salario família que é obtido pelo calculo de 5% do seu salario vezes a quantidade de dependentes. Deseja-se manter endereço de cada funcionário. O funcionário tem direito de um mês de férias por ano. A prefeitura paga gratificações sendo elas de periculosidade, insalubridade, desempenho e horas extras, o funcionário pode receber mais de uma gratificação ou nenhuma, o gestor de cada secretaria é responsável de passar os dados como dias trabalhados horas extras e gratificações de cada funcionário para o administrador do sistema no dia de fechamento de folha de pagamento. A prefeitura dispõe de cargos comissionados, estes cargos são destinados a pessoas de confiança do executivo, essas pessoas também ficam lotadas nas secretárias para ajudarem os gestores, os funcionários comissionados podem ser uma parte contratada, mas obrigatoriamente entre deve haver no mínimo 30% de efetivos.

## Requisitos

### Requisitos Funcionais, Não-Funcionais e Suplementares.

**Responsáveis**

|  |  |
| --- | --- |
| **Nome** | **Assinatura** |
| Igor Guedes Ribeiro | Igor |
| Jardel Lopes | Jardel |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| F1 – Controle de acesso para o operador do sistema | | | | Oculto ( ) |
| **Descrição:** O sistema deverá ter controle de acesso para autenticação do operador do sistema | | | | |
| Requisitos Não Funcionais | | | | |
| Nome | Restrição | Categoria | Desejável | Permanente |
| NF1.1 - Login com mais de 6 caracteres> | O login deve contar no mínimo 6 (seis) caracteres. |  | ( x ) | ( ) |
| NF1.2 – Senha formatada | O campo senha deve ser formatado para que a senha não seja mostrada ao digitar |  | ( ) | ( x ) |
| NF1.3 - Senha forte | A senha deve conter letras e números e será criptografada. |  | (x) | ( ) |
|  |  |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| F2 – Cadastro de secretaria | | | | **Oculto ( )** |
| **Descrição:** O sistema deverá possuir uma tela que permita o cadastro de secretária | | | | |
| ***Requisitos Não Funciona*** | | | | |
| **Nome** | **Resirtrição** | **Categoria** | **Desejável** | **Permanente** |
| NF2.1 – Id obrigatório | Cada secretária cadastrada apenas uma vez com o CNPJ próprio. |  | ( ) | ( x ) |
| NF2.2 – Acesso ao sistema | Somente o administrador do sistema poderá cadastrar secretárias |  | ( ) | ( x) |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| F3 – Cadastro de Funcionário | | | | **Oculto ( )** |
| **Descrição:** O sistema deverá possuir uma tela para cadastro de funcionários | | | | |
| ***Requisitos Não Funcionais*** | | | | |
| **Nome** | **Restrição** | **Categoria** | **Desejável** | **Permanente** |
| NF3.1Acesso ao sistema | Somente o administrador do sistema poderá cadastrar funcionários |  | ( ) | ( x) |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| F4 – Cadastro de dependente | | | | **Oculto ( )** |
| **Descrição:** O sistema deverá permitir o cadastro de dependentes | | | | |
| ***Requisitos Não Funcionais*** | | | | |
| **Nome** | **Restrição** | **Categoria** | **Desejável** | **Permanente** |
| NF4.1 - Acesso ao sistema | Somente o administrador do sistema poderá cadastrar dependentes |  | ( ) | ( x ) |
| NF4.2 Dependências | Dependente só pode ser cadastrado caso ao menos um ascendente já esteja cadastrado. |  | ( ) | ( x ) |
| NF4.3 - Multiplicidades de dependências | Um dependente deve pertencer obrigatoriamente a no mínimo 1 funcionário e no máximo 2 funcionários já cadastrado no sistema. |  | ( ) | ( x ) |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| F5 – Cadastro de cargo | | | | **Oculto ( )** |
| **Descrição:** O sistema deverá possuir uma tela para cadastro de cargos | | | | |
| ***Requisitos Não Funcionais*** | | | | |
| **Nome** | **Restrição** | **Categoria** | **Desejável** | **Permanente** |
| NF 5.1 Numero de contra cheques por página | Deverá ser gerado dois contracheques idênticos em uma mesma folha |  |  |  |
| NF 5.2 Tecnologia utilizada | Será utilizado o JasperReports e o iReport para gera contracheque |  | ( x ) | ( ) |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| F6 – Gerar contracheque | | | | **Oculto ( )** |
| **Descrição:** O sistema deverá gerar contracheque de cada funcionário | | | | |
| ***Requisitos Não Funcionais*** | | | | |
| **Nome** | **Restrição** | **Categoria** | **Desejável** | **Permanente** |
| NF 6.1 |  |  | ( ) | ( ) |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| F7 – Cadastrar Adicionais (proventos) | | | | **Oculto ( )** |
| **Descrição:** O sistema deverá permitir o cadastro de adicionais, como periculosidade, insalubridade, horas-extras, etc, para funcionários; | | | | |
| ***Requisitos Não Funcionais*** | | | | |
| **Nome** | **Restrição** | **Categoria** | **Desejável** | **Permanente** |
| NF 7.1 |  |  | ( ) | ( ) |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| F8 – Calcular Adicionais (proventos) | | | | **Oculto ( x )** |
| **Descrição:** O sistema deverá fazer o calculo de adicionais para os funcionários, | | | | |
| ***Requisitos Não Funcionais*** | | | | |
| **Nome** | **Restrição** | **Categoria** | **Desejável** | **Permanente** |
| NF8.1 |  |  | ( ) | ( ) |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| F9 – Cadastrar Descontos | | | | **Oculto ( )** |
| **Descrição:** O sistema deverá permitir o cadastro de descontos para os funcionários, | | | | |
| ***Requisitos Não Funcionais*** | | | | |
| **Nome** | **Restrição** | **Categoria** | **Desejável** | **Permanente** |
| NF9.1 |  |  | ( ) | ( ) |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| F10 – Calcular Descontos | | | | **Oculto ( x )** |
| **Descrição:** O sistema deverá fazer o calculo de descontos para os funcionários, | | | | |
| ***Requisitos Não Funcionais*** | | | | |
| **Nome** | **Restrição** | **Categoria** | **Desejável** | **Permanente** |
| NF.10.1 – calculo sob legislação | O calculo de desconto deverá ser feito de acordo com a legislação pertinente |  | ( ) | ( x ) |
| NF10.2 - Desconto ao funcionário | O calculo de desconto deverá ser feito de acordo com descontos cadastrados e que seja aplicado a determinado funcionário |  | ( ) | ( x) |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| F11– Gerenciar folha de pagamento | | | | **Oculto ( )** |
| **Descrição:** O sistema deverá permitir o gerenciamento de folha de pagamento | | | | |
| ***Requisitos Não Funcionais*** | | | | |
| **Nome** | **Restrição** | **Categoria** | **Desejável** | **Permanente** |
| NF11.1 |  |  | ( x ) | ( ) |

### Requisitos Suplementares

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nome | Restrição | Categoria | Desejável | Permanente |
| S 1 – Interface | O sistema deverá possuir uma interface com boa usabilidade. | interface | ( x ) | ( ) |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nome | Restrição | Categoria | Desejável | Permanente |
| S 2 – Portabilidade | O sistema como um todo deverá tem uma boa portabilidade de forma que seja possível a migração deste para outras tecnologia |  | ( x ) | ( ) |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nome | Restrição | Categoria | Desejável | Permanente |
| S 3 – Manutenibilidade | O sistema deverá ser fácil de dar manutenção |  | ( x ) | ( ) |

## Objetivos

### Objetivos Gerais

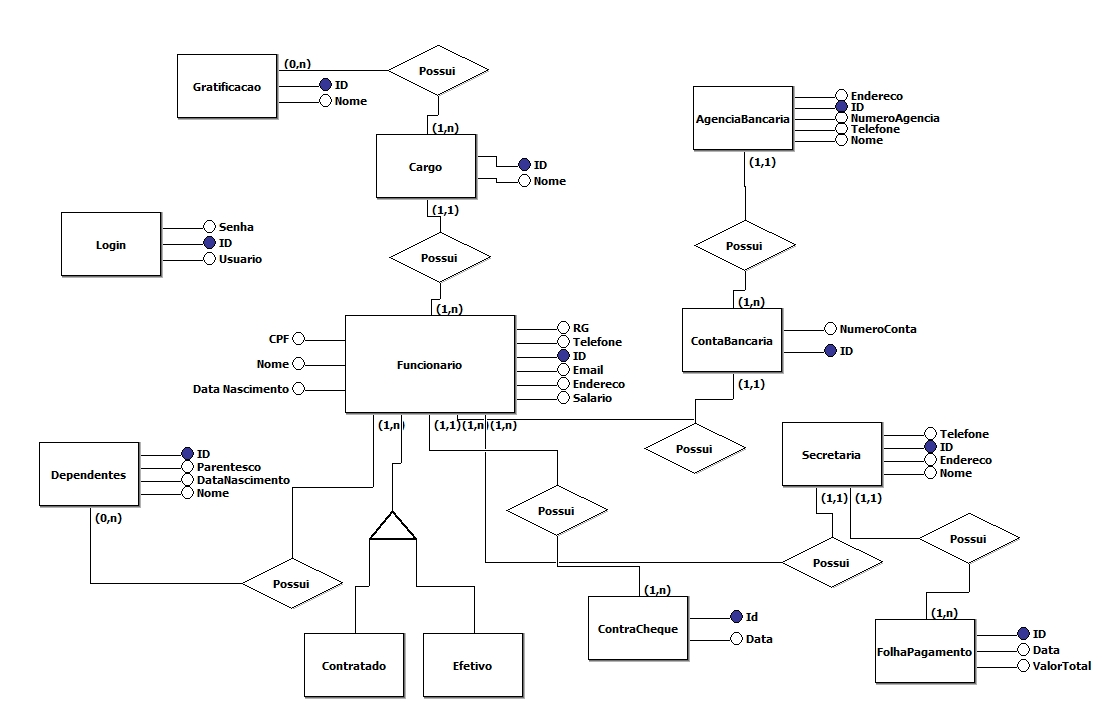
Desenvolver um sistema para gerir o setor de RH da prefeitura municipal de Cônego Marinho. O sistema deverá controlar os dados referentes aos funcionários, secretárias e a folha de pagamento.

### Objetivos Específicos

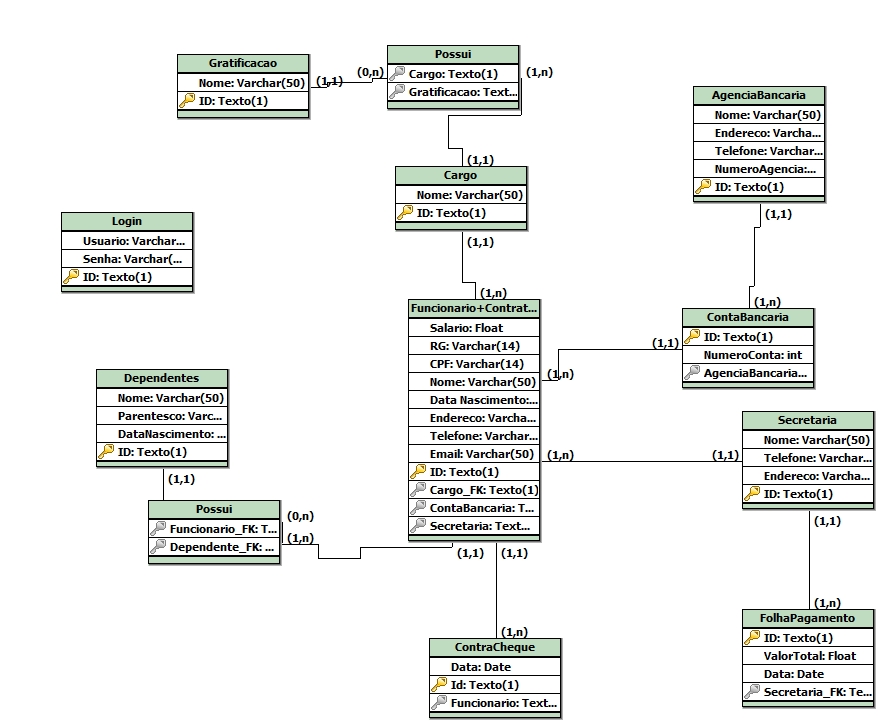
* Gerenciar as informações referentes ao funcionário;
* Gerenciar as informações das secretárias;
* Gerenciar a folha de pagamento da prefeitura;

# Analise

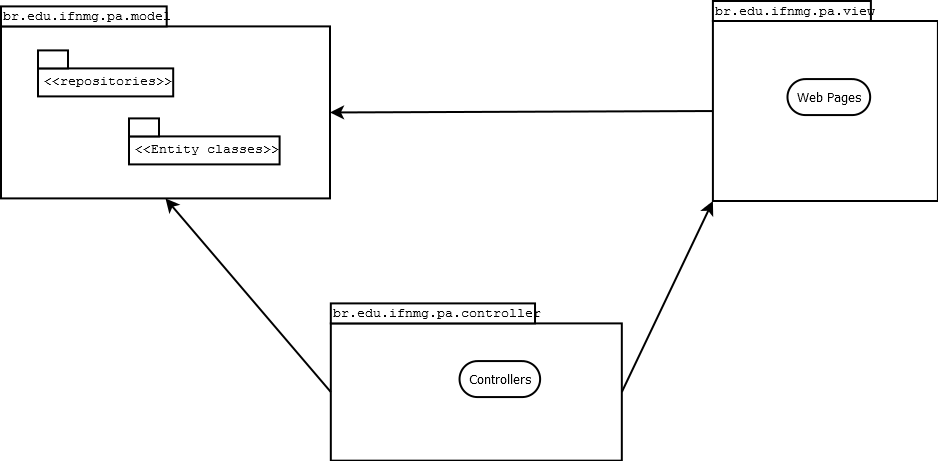
## Diagrama Conceitual

****

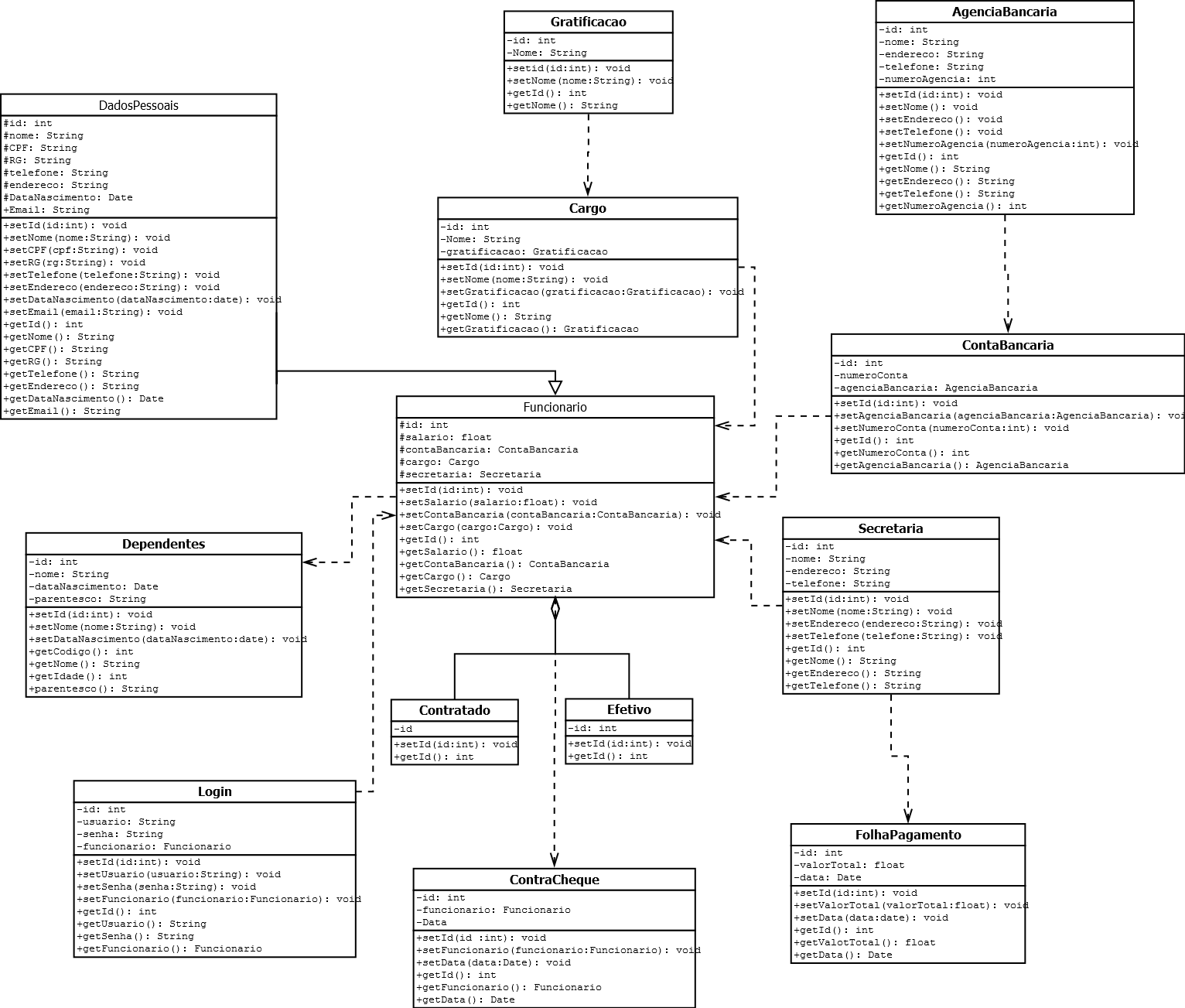
## Diagrama Lógico do Banco de Dados

****

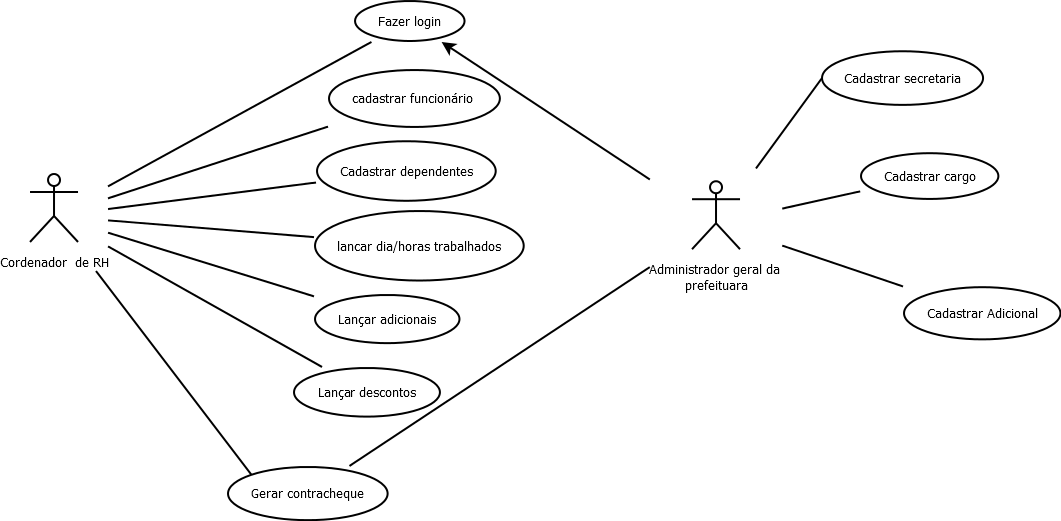
## Diagrama de Pacotes

****

## Diagrama de Classes

****

## Diagrama de Casos de Uso

****

## Descrição de Casos de Uso

|  |  |
| --- | --- |
| Nome do caso de uso | Cadastrar usuário |
| Descrição | Este caso de uso descreve os passos para que um usuário do sistema seja cadastrado no sistema. |
| Ator | Administrador do sistema |
| Pré-condições | O usuário não está cadastrado no sistema |
| Pós-condições | O usuário já encontra-se cadastrado no sistema. |
| Fluxo normal | 1. Na tela de cadastro de novo usuário, o usuário informa: Nome , sexo, data de nascimento, endereço, telefone, CPF, nome de usuário e senha. 2. O usuário salva o dados 3. O sistema emite a mensagem, usuário cadastrado com sucesso. |
| Fluxo secundário | 1. Quando a senha for digitada ela deve possuir números e letras, caso contrario o sistema informa senha invalida 2. O CPF deve conter apenas numero e sem espaços, caso seja digitado com espaço ou com outro caractere o sistema informa que o CPF digitado é invalido e que deve conter apenas numero. 3. Caso o CPF já esteja cadastrado no sistema é emitida uma mensagem dizendo que o CPF já está cadastrado, e não é possível cadastrar novamente. |

|  |  |
| --- | --- |
| Nome do caso de uso | Autenticar Usuário |
| Descrição | Este caso de uso descreve os passos para que um usuário do sistema consiga acessar a parte operacional do sistema. |
| Ator | Coordenador de RH, Administrador geral da prefeitura, administrador do sistema |
| Pré-condições | O usuário não está logado no sistema |
| Pós-condições | O usuário encontra-se logado no sistema |
| Fluxo normal | 1. Na tela inicial do sistema o usuário digita nome de usuário e senha para logar. 2. O sistema autentica o usuário. 3. O sistema emite uma mensagem dizendo login efetuado com sucesso. |
| Fluxo secundário | 1. Se a senha digitada for inválida o sistema exibirá uma mensagem de senha inválida e pedirá para digitar a senha correta. 2. Caso o nome de usuário não esteja cadastrado, o sistema exibe a mensagem de erro, informando que o usuário não está cadastrado e pede para efetuar cadastro. 3. Caso a senha seja digitada 5 vezes errada o sistema bloqueia o usuário e pede para contatar o administrador do sistema |

|  |  |
| --- | --- |
| Nome do caso de uso | Cadastra funcionário |
| Descrição | Este caso de uso especifica como cadastrar um funcionário no sistema. |
| Ator | Coordenador de RH |
| Pré-condições | O funcionário não está cadastrado no sistema. |
| Pós-condições | O funcionar já está cadastrado no sistema |
| Fluxo normal | 1. O coordenador de RH abre o formulário de cadastro de funcionário 2. O formulário é preenchido, sendo informado como dados do funcionário: Nome, CPF, RG, data de nascimento, cargo, escolaridade, telefone, endereço, sexo. No momento que os dados são informados o sistema já valida os dados em cada campo. 3. O coordenador de RH salva os dados. 4. O sistema emite uma mensagem de sucesso |
| Fluxo secundário | 1. No item 2 do fluxo normal O campo CPF deve ser preenchido somente com números, caso contrario o sistema emite uma mensagem de erro pedindo para digitar um CPF com apenas numero. 2. No item 2 do fluxo normal caso o CPF digitado seja invalido o sistema emite uma mensagem de erro, dizendo que o CPF é inválido. 3. No item 2 do fluxo normal caso o CPF digitado já esteja cadastrado o sistema emite mensagem de erro dizendo que o usuário com aquele CPF já está cadastrado. 4. No momento de salvar caso algum campo não esteja preenchido, o sistema emite mensagem de erro informando que há campo sem preencher. |

|  |  |
| --- | --- |
| Nome do caso de uso | Cadastrar dependentes |
| Descrição | Este caso de uso descreve como cadastrar um dependente de um funcionário no sistema |
| Ator | Coordenador de RH |
| Pré-condições | O funcionário relacionado com o dependente deve estar cadastrado no sistema.  O dependente não está cadastrado no sistema. |
| Pós-condições | O dependente já está cadastrado no sistema |
| Fluxo normal | 1. O coordenador de RH entra no Formulário de cadastro de dependentes. 2. O formulário é preenchido, sendo informado como dados do dependente: Nome, CPF, RG, data de nascimento, cargo, escolaridade, telefone, endereço, sexo, Funcionário o qual ele é dependente, grau de parentesco, os dados informados são validados no momento do preenchimento. 3. O ator salva os dados no sistema. 4. O sistema emite uma mensagem de sucesso. |
| Fluxo secundário | 1. No item 2 do fluxo normal O campo CPF deve ser preenchido somente com números, caso contrario o sistema emite uma mensagem de erro pedindo para digitar um CPF com apenas numero. 2. No item 2 do fluxo normal caso o CPF digitado seja invalido o sistema emite uma mensagem de erro, dizendo que o CPF é inválido. 3. No item 2 do fluxo normal caso o CPF digitado já esteja cadastrado o sistema emite mensagem de erro dizendo que o usuário com aquele CPF já está cadastrado. 4. O CPF e RG não são obrigatórios, caso algum outro campo além desses dois não sejam preenchidos, no momento de salvar o sistema emite mensagem de erro informando que tal campo obrigatório não está preenchido. 5. Caso já exista um dependente com o mesmo nome para o funcionário informado, o sistema emite uma mensagem dizendo que o dependente já está cadastrado. 6. Caso o funcionário informado não esteja cadastrado, o sistema emite uma mensagem de erro dizendo que o funcionário não está cadastrado, e não é possível cadastrar aquele dependente, antes de cadastrar o funcionário. |

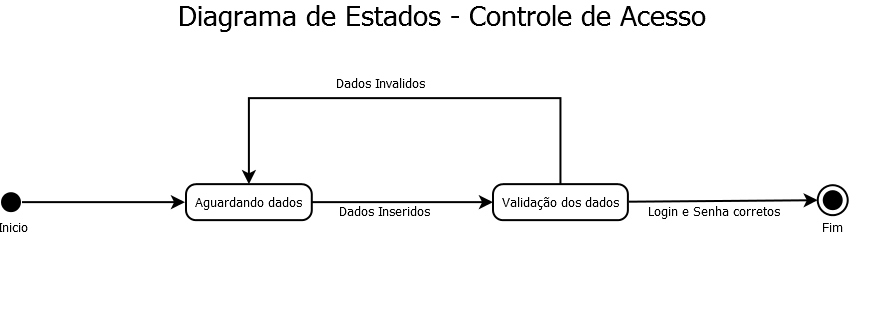
|  |  |
| --- | --- |
| Nome do caso de uso | Cadastrar cargo |
| Descrição | Este caso de uso descreve os passos necessários para cadastrar um novo cargo no sistema. |
| Ator | Administrador geral da prefeitura |
| Pré-condições | O cargo não pode está cadastrado no sistema |
| Pós-condições | O cargo já está cadastrado no sistema |
| Fluxo normal | 1. O ator entra na dela de cadastro de cargo. 2. O formulário de cadastro de cargo é preenchido com: Nome do cargo, Código, vencimento, secretaria a qual ele pertence, data de criação. 3. Os dados são salvos. 4. O sistema emite mensagem de sucesso. |
| Fluxo secundário | 1. Caso o código informado já esteja cadastrado o sistema emite uma mensagem dizendo que aquele cargo já está cadastrado, e pede para informar um código diferente 2. No momento de salvar se todos os campos não estiverem preenchido o sistema emite mensagem de erro informando que há campos sem preencher. |

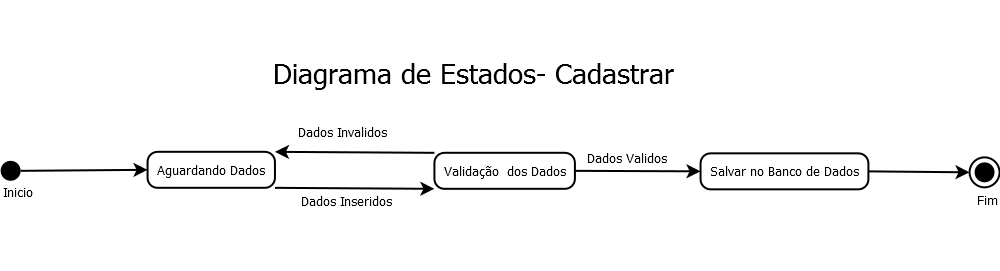
|  |  |
| --- | --- |
| Nome do caso de uso | Cadastrar desconto |
| Descrição | Este caso de uso descreve os passos necessários para cadastrar um novo tipo de desconto no sistema. |
| Ator | Administrador geral da prefeitura |
| Pré-condições | O desconto não está cadastrado no sistema. |
| Pós-condições | O desconto está cadastrado no sistema |
| Fluxo normal | 1. O ator entra na tela de cadastro de desconto 2. É feito o preenchimento do formulário de cadastro de desconto, onde é informado os valores dos campos, código, descrição, porcentagem, e data de criação. 3. Salvar os dados. 4. O sistema emite mensagem e sucesso. |
| Fluxo secundário | 1. Caso o código informado já esteja cadastrado o sistema emite uma mensagem dizendo que aquele desconto já está cadastrado, e pede para informar um código diferente 2. No momento de salvar se todos os campos não estiverem preenchido o sistema emite mensagem de erro informando que há campos sem preencher. |

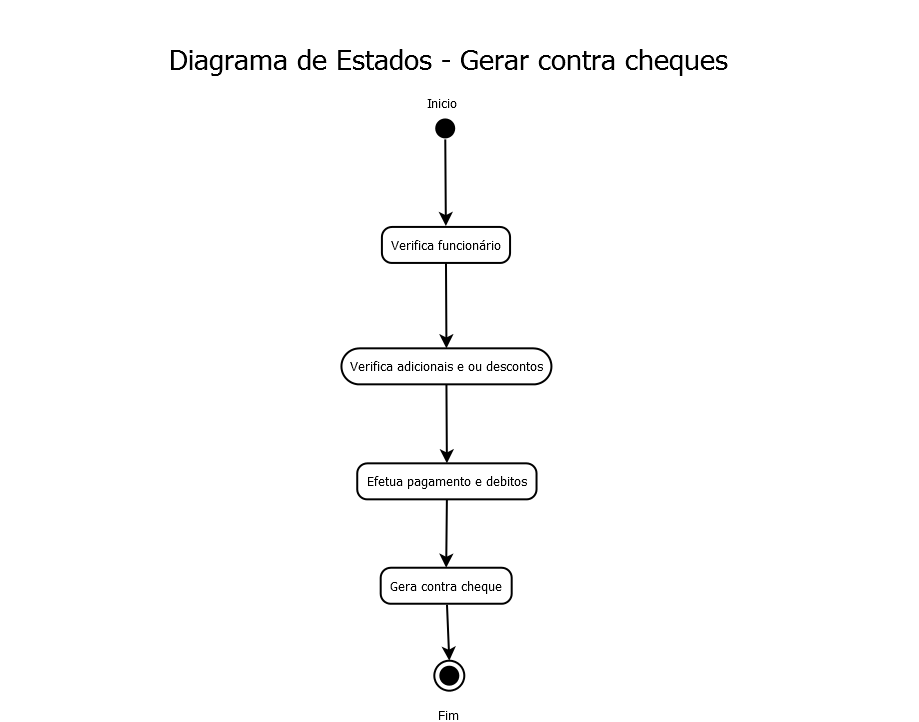
|  |  |
| --- | --- |
| Nome do caso de uso | Cadastrar Adicional |
| Descrição | Esse caso de uso descreve os passos necessários para cadastrar um novo adicional no sistema. |
| Ator | Administrador geral da prefeitura |
| Pré-condições | O adicional não está cadastrado no sistema. |
| Pós-condições | O adicional está cadastrado no sistema. |
| Fluxo normal | 1. Entrar na tela de cadastro de adicional. 2. Preencher o formulário de cadastro de adicional informando, código, descrição, porcentagem. 3. Salvar os dados. 4. O sistema emite mensagem e sucesso. |
| Fluxo secundário | 1. Caso o código informado já esteja cadastrado o sistema emite uma mensagem dizendo que aquele adicional já está cadastrado, e pede para informar um código diferente 2. No momento de salvar se todos os campos não estiverem preenchido o sistema emite mensagem de erro informando que há campos sem preencher. |

|  |  |
| --- | --- |
| Nome do caso de uso | Lançar pagamento |
| Descrição | Este caso de uso descreve os passos necessários para lançar o pagamento mensal para o funcionário, preenchendo a folha de pagamento do funcionário. |
| Ator | Coordenador de RH |
| Pré-condições | O funcionário deve estar cadastrado  Os descontos que são feitos para o funcionário devem estarem cadastrado.  Os adicionais que o funcionário tem direito devem estar cadastrados. |
| Pós-condições | Folha de pagamento preenchida.  Possibilidade de gerar contracheque para aquele funcionário. |
| Fluxo normal | 1. O coordenador de RH entra na tela de lançar pagamento. 2. É feito a busca do funcionário para lançar pagamento. 3. É lançada a quantidade de dias/horas trabalhadas por mês pelo funcionário. 4. Calcular pagamento bruto em relação ao vencimento básico e a quantidade de dias trabalhados. 5. É lançado os adicionais. 6. Calcular adicionais no mês 7. É lançado o desconto 8. Calcular descontos do mês 9. para calcular o valor liquido 10. O operador do sistema salva os dados 11. O sistema emite mensagem de sucesso 12. O operador do sistema pede para gerar a folha de pagamento. |
| Fluxo secundário | 1. Caso o usuário procurado não esteja cadastrado deverá ser feito o cadastro do funcionário. 2. Caso o desconto não esteja cadastrado deverá ser feito o cadastro do desconto 3. Se o adicional a ser lançado não estiver cadastrado deverá ser feito o cadastro do adicional. |

## Diagrama de Estados

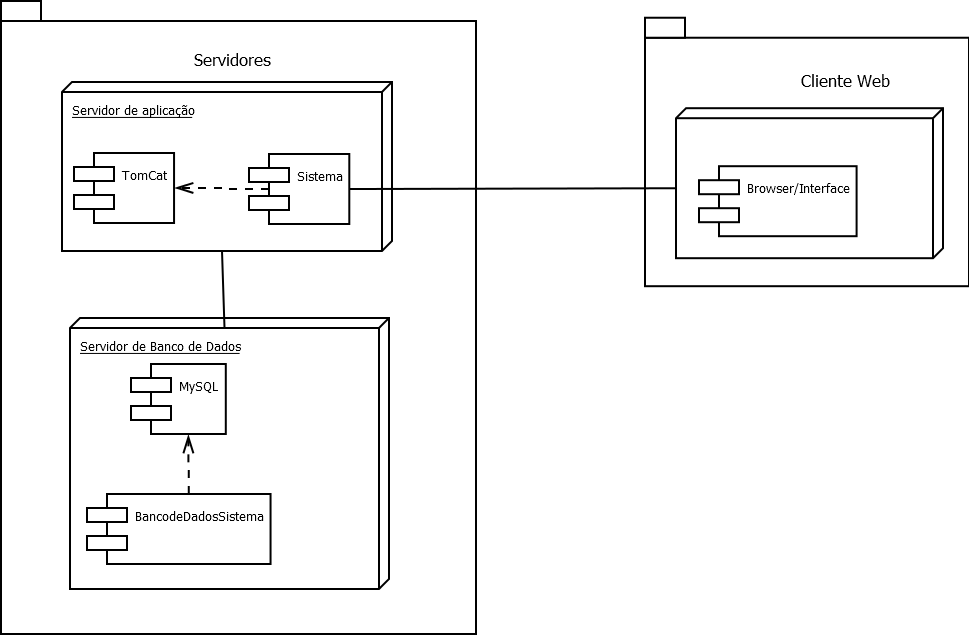
****

****

****

## Diagrama de Sequencia

## Diagrama de Implantação e Componentes

****

# Desenvolvimento

## ALM- Ambiente ALM

### Linguagem

Optou-se pela linguagem Java, por ser uma linguagem orientada a objetos, além de ser multiplataforma. As principais características da linguagem são:

* Orientação a objeto - Baseado no modelo de Smalltalk e Simula67;
* Portabilidade - Independência de plataforma - "write once run anywhere";
* Recursos de Rede - Possui extensa biblioteca de rotinas que facilitam a cooperação com protocolos TCP/IP, como HTTP e FTP;
* Segurança - Pode executar programas via rede com restrições de execução;

## IDE

A IDE utilizada será o NetBeans 8.0.1. Como [1] cita que o NetBeans IDE oferece suporte abrangente e de primeira classe para as tecnologias e melhorias de especificação Java mais recentes.

A IDE NetBeans é um editor completo, [1] salienta que a IDE oferece suporte a JDK 8, JDK 7, Java EE 7, inclusive a seus aprimoramentos de HTML5 relacionados, e a JavaFX2.

### Controle de versão

A ferramenta escolhida para controle de versão será o Git. [2] diz que o Git é um controle de versão distribuída e é utilizado principalmente para gerenciar versões de software, desenvolvidos por um ou mais desenvolvedores, podendo ser implementado novas funcionalidades ficando registrado tudo em um histórico, o qual é permitido retroceder sempre que necessário, além dos integrantes do projeto, participar enviando correções, atualizações, etc. Alterações não comprometem o projeto principal, pois o dono do projeto sempre optará por incluir ou não as alterações realizadas. Utilizando-se o Git, é evidente algumas vantagens marcantes, tais como: consistência, velocidade, espaço e simplicidade.

### SGBD

Será implementado o SGBD( Sistema Gerenciador de Banco de Dados) MySQL, por ser um SGBD portável, com muita facilidade de uso, leve e um software livre. O MySQL está presente em grandes empresas tais como cita [3]: NASA, HP, Nokia, Sony e muitas outras grandes empresas;

### Ferramenta de Issue/Bug Tracking

A Ferramenta de Issue/Bug Tracking escolhida será o GitHub. Conforme [4], o Github é um serviço web que oferece diversas funcionalidades extras aplicadas ao git. Resumindo, você poderá usar gratuitamente o github para hospedar seus projetos pessoais. Além disso, quase todos os projetos/frameworks/bibliotecas sobre desenvolvimento open source estão no github, e você pode acompanhá-los através de novas versões, contribuir informando bugs ou até mesmo enviando código e correções.

### Criação de Diagramas UML

A ferrementa escolhida para criação de diagramas UML será o DIA. Segundo [5] o DIA é um programa de criação de diagrama baseado em GTK + para GNU / Linux, MacOS X, Unix e Windows, e é liberado sob a licença GPL.

[5] Salienta também que o DIA é inspirado no programa comercial do Windows “Visio”, embora seja voltada para diagramas informais para uso casual. O Dia pode ser utilizado para desenhar tipos diferentes de diagramas, possuindo diversos objetos especiais para ajudar a desenhar diagramas de entidade relacionamento, diagramas UML, fluxogramas, diagramas de rede, e muitos outros; além da possibilidade de adicionar suporte para novos formatos de gravação de arquivos em XML, usando um subconjunto de SVG para desenhar a forma.

## Arquitetura

A arquitetura de desenvolvimento a ser utilizada será o padrão MVC( Model- View- Controller). Para [6] o padrão MVC é uma forma de quebrar uma aplicação, ou até mesmo um pedaço da interface de uma aplicação, em três partes: o modelo, a visão e o controlador. O sistema MVC tem como principais vantagens:

* Múltiplos visualizadores, podendo testar sistemas múltiplos sem necessidade de paralisar o sistema principal.
* Fácil de criar novas plataformas para o sistema, necessitando apenas criar os visualizadores e controles necessários para a adaptação.
* Torna o sistema escalável e portável.
* É possível desenvolver paralelamente as camadas, pois são independentes.

## APIs

As ferramentas utilizadas para o desenvolvimento do software serão:

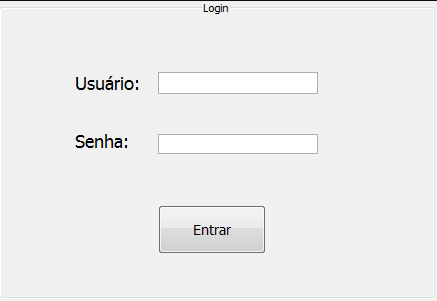
* **Java EE** - Segundo [7] o Java EE (Enterprise Edition) é uma plataforma amplamente usada que contém um conjunto de tecnologias coordenadas que reduz significativamente o custo e a complexidade do desenvolvimento, implantação e gerenciamento de aplicações de várias camadas centradas no servidor.
* **JPA-** Como salienta [7] o JPA- Java Persistense API é um framework que permite aos desenvolvedores gerenciar os dados utilizando o mapeamento relacional de objetos (ORM) em aplicações construídas na plataforma Java.
* **JSF-** Segundo [8] o JSF veio para padronizar o uso de componentes ricos pelas aplicações, além de se tornar uma referência às empresas com garantia de extensibilidade de seus componentes, ou seja, é possível a criação de novos componentes que se encaixam nas necessidades de sua aplicação.

Essa possibilidade de customização de componentes é um recurso poderoso de extensão para atender às demandas específicas de reuso presentes nas aplicações web atuais.

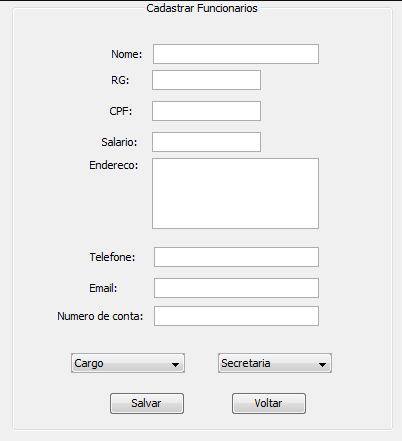
* **EJB 3.0:** Conforme [8] salienta o EJB 3.0 é uma arquitetura gerenciada de componente do lado do servidor utilizada para encapsular a lógica de negócios de uma aplicação. A tecnologia EJB permite o desenvolvimento rápido e simplificado de aplicações distribuídas, transacionais, seguras e portáteis baseadas na tecnologia Java.

## Protótipo de Telas

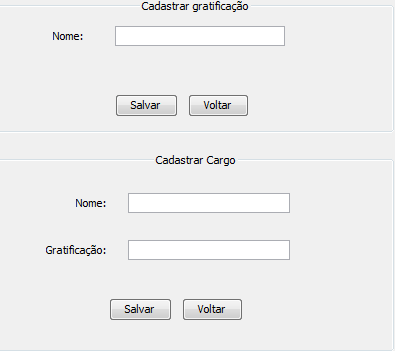
**Controle de Acesso**

****

**Cadastrar Funcionários**

****

**Cargos e Gratificações**

****

# Referências

# 

[1]- Disponível em: <https://netbeans.org/community/releases/80/index_pt_BR.html> Acesso em 23/11/2015;

[2]- Disponível em: http://pt.slideshare.net/HenriqueSilvestre1/git-e-github-conceitos-bsicos Acesso em 23/11/2015;

[3]- Disponível em: <http://www.devmedia.com.br/introducao-ao-mysql/28438> Acesso em 23/11/2015;

[4]- Disponível em: http://tableless.com.br/tudo-que-voce-queria-saber-sobre-git-e-github-mas-tinha-vergonha-de-perguntar/> Acesso em 23/01/2015;

[5]- Disponível em :<https://wiki.gnome.org/action/show/Apps/Dia?action=show&redirect=Dia> Acesso em 23/11/2015;

[6] - Disponível em: <http://www.jonathanribas.com/blog/introducao-ao-modelo-mvc/> Acesso em 23/01/2015;

[7]- Disponível em Java EE. Disponível em < <https://netbeans.org/kb/trails/java-ee_pt_BR.html>> Acesso em 23/01/2015;

[8]- Disponível em: Java Server Faces (JSF): Desenvolvimento de aplicações web. Disponível em: <http://www.devmedia.com.br/java-server-faces-jsf-desenvolvimento-de-aplicacoes-web/31026#ixzz3sVsUNypl> Acesso em 24/11/2015;