****

**CENTRO UNIVERSITÁRIO DA GRANDE DOURADOS**





**LUCAS RIBEIRO DAL VESCO**

**SISTEMA DE GERENCIAMENTO E AUTOMAÇÃO DE ROTINAS DE RH**

Dourados

2025

****

**CENTRO UNIVERSITÁRIO DA GRANDE DOURADOS**





LUCAS RIBEIRO DAL VESCO

**SISTEMA DE GERENCIAMENTO E AUTOMAÇÃO DE ROTINAS DE RH**

Trabalho de Conclusão de Curso apresentado ao Curso de Engenharia de Software da Faculdade de Ciências Exatas e Agrárias como pré-requisito para obtenção do título de Bacharel em Engenharia de Software.

Orientador: Prof. (M.Sc.) Felipe Perez.



Dourados

2025

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

SUMÁRIO

[1](#_tyjcwt) ESCOPO DO SISTEMA 1

[1.1](#_3dy6vkm) DADOS INICIAIS 1

[1.2](#_26in1rg) MOTIVAÇÃO E PROBLEMÁTICA ABORDADA PELO SOFTWARE 1

[1.2.1](#_lnxbz9) Definição e importância 1

[1.2.2](#_35nkun2) Contextualização 2

[1.2.3](#_1ksv4uv) O Público-alvo 2

[1.3](#_44sinio) JUSTIFICATIVA DO PROJETO 2

[1.4](#_2jxsxqh) ENTREGAS DO PROJETO 2

[1.5](#_z337ya) OBJETIVOS DO SISTEMA 2

[1.6](#_3j2qqm3) CRITÉRIOS DE ACEITAÇÃO DO SISTEMA 2

[1.7](#_1y810tw) CONSULTOR DO SISTEMA 2

[1.8](#_4i7ojhp) ENTREVISTA COM O CONSULTOR DO SISTEMA 3

[2](#_2xcytpi) REQUISITOS DO SISTEMA 4

[2.1](#_1ci93xb) METODOLOGIA DE LEVANTAMENTO DE REQUISITOS 4

[2.2](#_2bn6wsx) REQUISITOS 4

[2.3](#_qsh70q) MATERIAIS E MÉTODOS (LINGUAGEM E FERRAMENTAS UTILIZADAS) 4

[2.3.1](#_3as4poj) Casos de Usos Gerais 5

[2.3.2](#_1pxezwc) Atores envolvidos 6

[2.4](#_49x2ik5) CASOS DE USO ESPECÍFICOS 6

[2.4.1](#_2p2csry) Controlar Monitores 6

[2.4.2](#_147n2zr) (Inserir outros casos de usos) 7

[2.5](#_3o7alnk) ARQUITETURA DO SISTEMA 7

[2.6](#_23ckvvd) DIAGRAMAS 8

[2.6.1](#_ihv636) Modelo de Classes (ou DER se não usar Orientação a Objetos) 8

[3](#_2grqrue) DESENVOLVIMENTO DE SOFTWARE 9

[3.1](#_vx1227) METODOLOGIA DE DESENVOLVIMENTO DE SOFTWARE 9

[3.1.1](#_3fwokq0) Ambientes de desenvolvimento/produção 9

[3.1.2](#_1v1yuxt) Bibliotecas principais 9

[3.2](#_4f1mdlm) MÓDULOS DO CRONOGRAMA DE DESENVOLVIMENTO 9

[3.3](#_2u6wntf) MOCKUPS 9

[4](#_3tbugp1) CONCLUSÃO 10

[5](#_28h4qwu) REFERÊNCIAS 11

**HISTÓRICO DE ALTERAÇÕES**

| **Data** | **Versão** | **Descrição** | **Autor** |
| --- | --- | --- | --- |
| 14/03/2025 | 0.1.0 | Inclusão dos requisitos funcionais e não funcionais |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

O versionamento do documento será feito utilizando os parâmetros baseados na metodologia semver. O documento só será considerado na versão 1.0 quando completar os capítulos 1, 2 e 3. Toda alteração no documento deve constar na tabela acima.

| Versionamento numeração x.y.z | | |
| --- | --- | --- |
| X | MAJOR | Alterações drásticas (Inclusão/Alteração Caso de Uso Geral)  Adição de novos capítulos (4 e 5) |
| Y | MINOR | Adição/Remoção de Funcionalidades |
| z | PATCH | Correções ortográficas e/ou tipográficas |

**CONVENÇÕES, TERMOS E ABREVIAÇÕES**

A correta interpretação deste documento exige o conhecimento de algumas convenções e termos específicos, que são descritos a seguir.

**Identificação dos requisitos**

Por convenção, a referência a requisitos é feita através do nome da subseção onde eles estão descritos, seguidos do identificador do requisito, de acordo com a especificação a seguir:

[nome da subseção. identificador do requisito]

Por exemplo, o requisito funcional [Recuperação de dados.RF016] deve estar descrito em uma subseção chamada “Recuperação de dados” (que indica um subsistema), em um bloco identificado pelo número [RF016]. Já o requisito não-funcional [Confiabilidade.NF008] deve estar descrito na seção de requisitos não-funcionais de Confiabilidade, em um bloco identificado por [NF008].

Os requisitos devem ser identificados com um identificador único. A numeração inicia com o identificador [RF001] ou [NF001] e prossegue sendo incrementada à medida que forem surgindo novos requisitos.

**Prioridades dos requisitos**

Para estabelecer a prioridade dos requisitos, nos capítulos 3 e 4, foram adotadas as denominações “essencial”, “importante” e “desejável”.

**Essencial** é o requisito sem o qual o sistema não entra em funcionamento. Requisitos essenciais são requisitos imprescindíveis, que têm que ser implementados impreterivelmente.

**Importante** é o requisito sem o qual o sistema entra em funcionamento, mas de forma não satisfatória. Requisitos importantes devem ser implementados, mas, se não forem, o sistema poderá ser implantado e usado mesmo assim.

**Desejável** é o requisito que não compromete as funcionalidades básicas do sistema, isto é, o sistema pode funcionar de forma satisfatória sem ele. Requisitos desejáveis podem ser deixados para versões posteriores do sistema, caso não haja tempo hábil para implementá-los na versão que está sendo especificada.

# ESCOPO DO SISTEMA

## DADOS INICIAIS

**Nome do software:**

AutoGRH.

**Patrocinador**

Lucas Ribeiro Dal Vesco

**Público-alvo**

Empresas de médio a grande porte, que possuem uma quantidade elevada de colaboradores.

**Stakeholders**

Empresas nas quais o sistema será implementado, seus colaboradores e gestores, bem como contadores responsáveis, caso a empresa possua.

**Equipe Básica**

**Analistas/Desenvolvedores:**

Lucas Ribeiro Dal Vesco

**Orientadores:**

Felipe Perez

**Consultor:**

Gilberto Ribeiro Dal Vesco

## MOTIVAÇÃO E PROBLEMÁTICA ABORDADA PELO SOFTWARE

### Definição e importância

O setor de Recursos Humanos (RH) é uma parte fundamental de qualquer empresa, sendo responsável pela gestão do capital humano e pelo desenvolvimento organizacional. Seu papel vai muito além do recrutamento e seleção de funcionários, abrangendo também o bem-estar dos colaboradores, a cultura organizacional e o cumprimento das leis trabalhistas. A importância do RH está diretamente relacionada ao sucesso e à produtividade da empresa. Um RH eficiente contribui para um ambiente de trabalho positivo, promovendo a motivação dos colaboradores e aumentando a retenção de talentos. Além disso, auxilia na capacitação profissional, garantindo que os funcionários estejam preparados para enfrentar desafios e contribuir para o crescimento da organização.

### Contextualização

Quando uma empresa passa a possuir um número mais elevado de funcionários, a quantidade de informações que a empresa deve armazenar aumenta, e as rotinas habituais se tornam mais demoradas e propensas a erros. Para compensar o aumento da carga de trabalho, se faz necessária a implementação de ferramentas mais eficientes.

### O Público-alvo

Empresas de médio a grande porte, que possuem um número elevado de colaboradores.

## JUSTIFICATIVA DO PROJETO

Este tipo de solução costuma estar embutido em sistemas mais abrangentes, resultando muitas vezes em uma necessidade de sobreposição ou substituição de sistemas caso a empresa já utilize de outros softwares para áreas diferentes. A proposta do projeto é criar um programa dedicado para esta área, de forma genérica e que não conflita com possíveis sistemas que já estejam em uso no ambiente da empresa.

## ENTREGAS DO PROJETO

* Documento de Requisitos
* Sistema codificado com os requisitos implementados

## OBJETIVOS DO SISTEMA

O Software tem como objetivo atuar na área de recursos humanos de uma empresa, gerenciando dados dos funcionários, e automatizando rotinas.

O sistema tem como objetivos:

* armazenamento dos dados de registro dos funcionários
* controle e histórico de férias disponíveis, valores pendentes e descontos
* cálculo da folha de pagamento, com descontos e adicionais
* atualização de salário
* registro do histórico de ações
* geração de relatórios

## CRITÉRIOS DE ACEITAÇÃO DO SISTEMA

Todas as funcionalidades do website devem ser testadas através do emprego de:

* Testes de Usabilidade;
* Testes de Software;
* Teste nos Navegadores;

## CONSULTOR DO SISTEMA

Gilberto Ribeiro Dal Vesco

CPF:730.057.381-91

Contato: gilbertofilho@moveisdalvesco.com.br

## ENTREVISTA COM O CONSULTOR DO SISTEMA

**Quais informações dos funcionários precisam ser armazenadas no sistema?**

Nome, endereço, rg, cpf, Pis, carteira de trabalho, data nascimento, telefone, contato de emergência, data de admissão, cargo, salário (mesmos dados da ficha de registro do funcionário)

**O sistema deve gerar cálculos automáticos da folha de pagamento?**

Sim

**Você precisa de dashboards para visualização rápida de dados?**

Sim

**O sistema deve permitir a criação e publicação de vagas de emprego?**

Não

**Quais processos do RH são mais demorados e poderiam ser automatizados?**

Geração da folha de pagamento

Calculo de férias

**Quais tipos de relatórios são essenciais para a tomada de decisão no RH?**

Relatório de funcionários com férias vencidas

Relatório de funcionários com férias a vencer

Relatório de dias de férias disponíveis (ex. se um funcionário tirou férias parcial quantos dias ainda tem direito referente aquele período)

Relatório de faltas (por mês e por ano) por funcionário e geral

Relatório de alterações salariais (por funcionário)

**Há necessidade de um sistema de auditoria para rastrear mudanças nos dados?**

Sim

**Você deseja que o sistema faça backups automáticos para evitar a perda de dados?**

Sim

**Quais níveis de permissão devem existir no sistema para diferentes usuários?**

Administrador, e usuário comum.

Apenas o administrador poderá excluir informações em definitivo.

**O sistema precisa funcionar offline?**

Sim

**O acesso ao sistema deve ser feito apenas dentro da empresa ou também remotamente?**

Apenas dentro da empresa

# REQUISITOS DO SISTEMA

Neste capítulo, serão apresentados os aspectos técnicos do projeto Website a ser desenvolvido.

## METODOLOGIA DE LEVANTAMENTO DE REQUISITOS

O projeto terá início em Março de 2025, abrangendo as etapas de

planejamento e desenvolvimento do sistema. O projeto será dividido em duas

etapas, a primeira etapa sendo denominada de Levantamento de Requisitos, onde os

analistas levantarão os requisitos do sistema, através de entrevistas e

observações no recinto. Após a coleta de dados, o analista elaborará uma

versão preliminar dos requisitos do sistema (que serão descritos no capítulo 2

desse documento).

## REQUISITOS

O sistema deverá prover os seguintes requisitos:

**Requisitos Funcionais**

* O sistema deve permitir o cadastro de funcionários com todas as informações necessárias (nome, endereço, CPF, RG, etc.).
* O sistema deve permitir a edição e atualização dos dados cadastrais dos funcionários.
* O sistema deve permitir a exclusão lógica de funcionários, mantendo o histórico de informações.
* O sistema deve realizar cálculos automáticos da folha de pagamento.
* O sistema deve calcular descontos e adicionais na folha de pagamento.
* O sistema deve calcular automaticamente o saldo de férias de cada funcionário.
* O sistema deve registrar o histórico de férias dos funcionários.
* O sistema deve permitir o agendamento e a aprovação de férias pelos gestores.
* O sistema deve gerar relatórios de funcionários com férias vencidas.
* O sistema deve gerar relatórios de funcionários com férias a vencer.
* O sistema deve gerar relatórios detalhados de dias de férias disponíveis por funcionário.
* O sistema deve gerar relatórios de alterações salariais de funcionários.
* O sistema deve permitir o cadastro de níveis de acesso para usuários (administrador e usuário comum).
* O sistema deve permitir que apenas o administrador exclua informações em definitivo.
* O sistema deve realizar backup automático dos dados periodicamente.
* O sistema deve permitir a atualização de salários e benefícios dos funcionários.
* O sistema deve permitir a consulta rápida de informações dos funcionários.
* O sistema deve emitir alertas sobre funcionários com férias vencidas.
* O sistema deve armazenar e permitir a consulta ao histórico de alterações de dados dos funcionários.
* O sistema deve armazenar e permitir a consulta ao histórico de alterações de dados dos funcionários.
* O sistema deve ser acessível via login e senha.
* O sistema deve permitir que a quantidade de faltas dos funcionários seja inserida manualmente como um número inteiro, sem controle automático de ponto.

**Requisitos Não-Funcionais**

* O sistema deve ser acessível apenas dentro da rede local da empresa.
* O sistema deve ser dividido em backend (servidor) e frontend (acessível via navegador na LAN).
* O sistema deve implementar um mecanismo de logs para auditoria.
* O sistema deve realizar backups automáticos diários dos dados armazenados.
* O sistema deve possuir um mecanismo de logs para rastreamento de ações dos usuários.

## MATERIAIS E MÉTODOS (LINGUAGEM E FERRAMENTAS UTILIZADAS)

O desenvolvimento do sistema **AutoGRH** será realizado utilizando tecnologias modernas voltadas para aplicações web. O sistema será estruturado em uma arquitetura **cliente-servidor**, onde o frontend será desenvolvido em **React** e o backend em **Golang**.

**Linguagens e Tecnologias Utilizadas:**

**Frontend:**

* Linguagem: JavaScript
* Framework: React.js
* Comunicação com o backend: Fetch API

**Backend:**

* Linguagem: Golang
* Banco de Dados: MySQL

**Ambiente de Desenvolvimento e Infraestrutura:**

* IDE: Visual Studio Code
* Versionamento de Código: GitHub

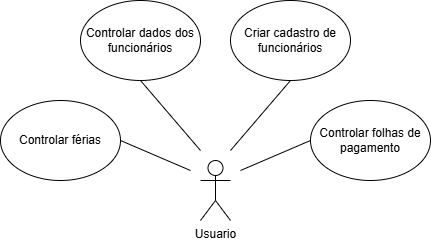
**Gerenciamento de Dependências:**

* Frontend: npm
* Backend: Go Modules

**Servidor de Aplicação:**

* O backend será hospedado em um servidor local dentro da rede da empresa.
* O frontend será acessível via navegador dentro da LAN corporativa.

### Casos de Usos Gerais



**Criar cadastro de funcionários**

**Descrição:** Permite que o usuário, que pode ser tanto um funcionário do RH quanto o gerente, a criar uma nova entrada de funcionário no sistema

**Prioridade: Essencial**

**Pré-condições:** O usuário deve estar logado no sistema.

**Entrada:** Recebe como entrada os dados do funcionário a ser cadastrado.

**Controlar dados dos funcionários**

**Descrição:** Permite que o usuário insira, altere ou delete (caso tenha permissão) informações e documentos pertencentes ao funcionário cadastrado.

**Prioridade: Essencial**

**Pré-condições:** O usuário deve estar logado no sistema, e para ações de remoção deve possuir permissão de administração.

**Entrada:** Recebe como entrada os dados do funcionário a serem inseridos ou alterados.

### 

**Controlar férias**

**Descrição:** Permite que o usuário crie e aprove pedidos de férias para funcionários cadastrados e acesse o histórico de férias.

**Prioridade: Essencial**

**Pré-condições:** O usuário deve estar logado no sistema, e deve ter permissão de administrador para aprovação de pedidos.

**Entrada:** Recebe como entrada o número de dias de férias a serem requisitados e o numero de faltas do ano em questão.

### 

**Controlar folhas de pagamento**

**Descrição:** Permite que o usuário gere um relatório de folha de pagamento mensal, acesse o histórico de relatórios passados, e faça pedidos e aprovação para vales de adiantamento salarial dos funcionários cadastrados.

**Prioridade: Essencial**

**Pré-condições:** O usuário deve estar logado no sistema, e deve possuir permissão de administrador para aprovações de vales.

**Entrada:** Recebe como entrada o número de faltas do mês em questão, ou o valor de vale a ser solicitado.

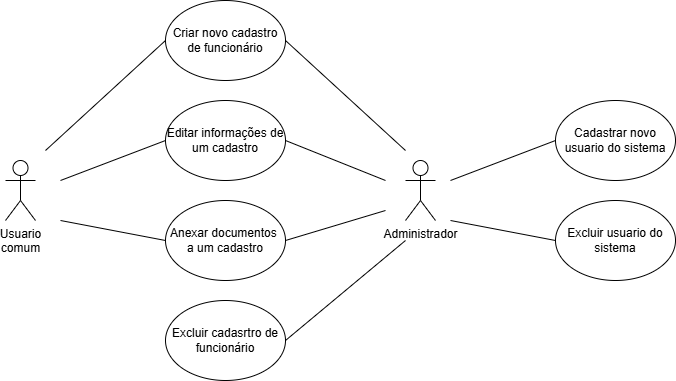
### Atores envolvidos

Neste sistema, há dois atores que irão usar diretamente este sistema proposto:

* Usuário comum: É o usuário comum, funcionário responsável pelo setor de RH da empresa. Possui permissão para Criar e modificar dados, assim como criar pedidos e acessar relatórios.
* Administrador: É o proprietário ou gerente da empresa. Possui todas as permissões do usuário comum, além da permissão para ações de remoção de dados e aprovação de pedidos.

## CASOS DE USO ESPECÍFICOS

### Controlar Funcionários



**Criar novo cadastro de funcionário**

**Descrição:** Este caso de uso permite que o usuário comum ou administrador adicione um novo funcionário no sistema.

**Prioridade: Essencial**

**Pré-condições:** O usuário comum ou administrador deve estar logado no sistema.

**Entrada:** Recebe como entrada o nome do funcionário, bem como suas informações e documentação trabalhista

**Saída e pós-condições:** Um novo funcionário é cadastrado no sistema.

**Editar informações de um cadastro**

**Descrição:** Este caso de uso permite que o usuário/administrador altere as informações de um funcionário previamente cadastrado no sistema.

**Prioridade: Essencial**

**Pré-condições:** O usuário comum ou administrador deve estar logado no sistema.

**Entrada:** Recebe como entrada o cadastro a ser alterado, e as informações a serem inseridas nele.

**Saída e pós-condições:** Um cadastro de funcionário já existente é alterado no sistema.

### 

**Anexar documentos a um cadastro**

**Descrição:** Este caso de uso permite que o usuário/administrador anexe documentos relevantes ao cadastro de um funcionário, como avisos e atestados.

**Prioridade: Essencial**

**Pré-condições:** O usuário comum ou administrador deve estar logado no sistema.

**Entrada:** Recebe como entrada cadastro do funcionário em questão, e o documento a ser anexado

**Saída e pós-condições:** Um novo documento é anexado a um cadastro do sistema.

**Excluir cadastro de funcionário**

**Descrição:** Este caso de uso permite que o administrador exclua um funcionário cadastrado no sistema.

**Prioridade: Essencial**

**Pré-condições:** O administrador deve estar logado no sistema.

**Entrada:** Recebe como entrada o cadastro do funcionário a ser excluído do sistema

**Saída e pós-condições:** Um funcionário é excluído do sistema.

**Cadastrar novo usuário do sistema**

**Descrição:** Este caso de uso permite que o administrador cadastre um novo usuário comum para utilizar o sistema.

**Prioridade: Essencial**

**Pré-condições:** O administrador deve estar logado no sistema.

**Entrada:** Recebe como entrada o login e senha do novo usuário.

**Saída e pós-condições:** Um novo usuário é cadastrado no sistema.

**Excluir um usuário do sistema**

**Descrição:** Este caso de uso permite que o administrador exclua um usuário comum do sistema.

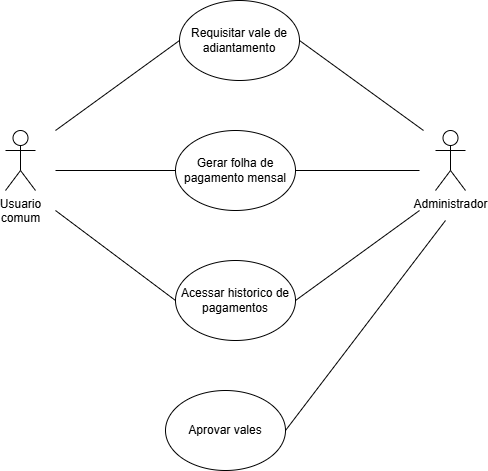
**Prioridade: Essencial**

**Pré-condições:** O administrador deve estar logado no sistema.

**Entrada:** Recebe como entrada o login e senha do usuário a ser excluído.

**Saída e pós-condições:** Um usuário é excluído do sistema.

* + 1. **Controlar folhas de pagamento**



**Requisitar vale de adiantamento**

**Descrição:** Este caso de uso permite que o usuário/administrador crie uma solicitação de vale de adiantamento para um funcionário

**Prioridade: Essencial**

**Pré-condições:** O usuário/administrador deve estar logado no sistema.

**Entrada:** Recebe como entrada o nome do usuário e o valor requisitado.

**Saída e pós-condições:** Uma requisição de vale é criada no sistema.

**Gerar folha de pagamento mensal**

**Descrição:** Permite que o usuário/administrador gere uma folha de pagamento com os valores salariais de cada funcionário calculados levando em consideração descontos e benefícios.

**Prioridade: Essencial**

**Pré-condições:** O usuário/administrador deve estar logado no sistema.

**Entrada:** Recebe como entrada mês da folha de pagamento.

**Saída e pós-condições:** Um relatório da folha de pagamento é gerado.

**Acessar histórico de pagamentos**

**Descrição:** Este caso de uso permite que o usuário/administrador acesse um histórico das folhas de pagamento geradas previamente.

**Prioridade: Essencial**

**Pré-condições:** O usuário/administrador deve estar logado no sistema.

**Entrada:** Recebe como entrada o mês da folha de pagamento a ser requisitada.

**Saída e pós-condições:** Uma lista de folhas de pagamento antigas é exibida.

**Excluir um usuário do sistema**

**Descrição:** Este caso de uso permite que o administrador aprove ou negue um vale solicitado.

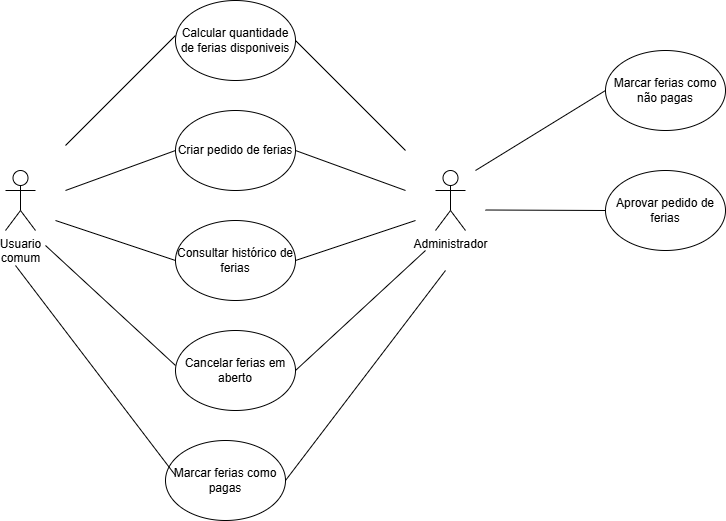
**Prioridade: Essencial**

**Pré-condições:** O administrador deve estar logado no sistema.

**Entrada:** Recebe como entrada a decisão do administrador.

**Saída e pós-condições:** o vale é excluído ou adicionado à folha de pagamento.

* + 1. **Controlar férias**



**Calcular quantidade de férias disponíveis**

**Descrição:** Este caso de uso permite que o usuário/administrador consulte a quantidade de dias disponíveis para férias de um funcionário.

**Prioridade: Essencial**

**Pré-condições:** O usuário/administrador deve estar logado no sistema.

**Entrada:** Recebe como entrada o nome do funcionário em questão.

**Saída e pós-condições:** O número de dias que o funcionário possui disponível para férias.

**Criar pedido de férias**

**Descrição:** Este caso de uso permite que o usuário/administrador faça uma solicitação para férias em nome de um funcionário.

**Prioridade: Essencial**

**Pré-condições:** O usuário/administrador deve estar logado no sistema.

**Entrada:** Recebe como entrada a data e duração do período de férias.

**Saída e pós-condições:** Um pedido de férias é criado no sistema.

**Consultar histórico de férias**

**Descrição:** Este caso de uso permite que o usuário/administrador consulte o histórico de férias de um funcionário.

**Prioridade: Essencial**

**Pré-condições:** O usuário/administrador deve estar logado no sistema.

**Entrada:** Recebe como entrada o funcionário associado ao histórico desejado.

**Saída e pós-condições:** Uma lista de férias antigas aprovadas de um funcionário.

**Cancelar férias em aberto**

**Descrição:** Este caso de uso permite que o usuário/administrador cancele um pedido de férias em aberto.

**Prioridade: Essencial**

**Pré-condições:** O usuário/administrador deve estar logado no sistema.

**Entrada:** Recebe como entrada o pedido a ser cancelado.

**Saída e pós-condições:** o pedido é excluído do sistema.

**Marcar férias como pago**

**Descrição:** Este caso de uso permite que o usuário/administrador mude o estado de um período de férias do histórico para “pago”.

**Prioridade: Essencial**

**Pré-condições:** O usuário/administrador deve estar logado no sistema, e o estado do período de férias em questão deve estar fechado.

**Entrada:** Recebe como entrada um período de férias, fechado e não pago, de um histórico de férias de um funcionário.

**Saída e pós-condições:** O pedido de férias em questão é marcado como pago.

**Marcar férias como não pago**

**Descrição:** Este caso de uso permite que o administrador mude o estado de um período de férias do histórico para “não pago”.

**Prioridade: Essencial**

**Pré-condições:** O usuário/administrador deve estar logado no sistema, e o estado do período de férias em questão deve estar como “não pago”.

**Entrada:** Recebe como entrada um período de férias pago, de um histórico de férias de um funcionário.

**Saída e pós-condições:** O pedido de férias em questão é marcado como não pago.

**Aprovar pedido de férias**

**Descrição:** Este caso de uso permite que o administrador aprove ou negue um pedido de férias.

**Prioridade: Essencial**

**Pré-condições:** O administrador deve estar logado no sistema.

**Entrada:** Recebe como entrada a decisão do administrador.

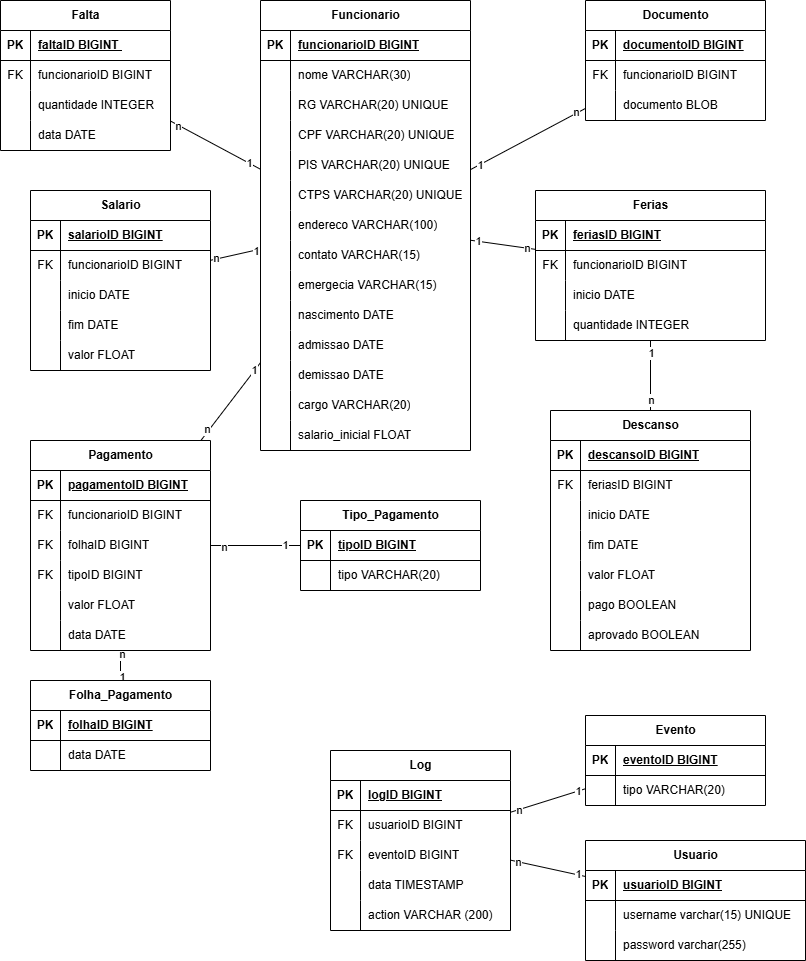
**Saída e pós-condições:** o pedido é marcado como “em aberto” caso seja aprovado, ou excluído caso seja negado.

## ARQUITETURA DO SISTEMA

Deve ser descrito como a arquitetura do sistema é composta. Deve ser feito um diagrama visual mostrando de maneira gráfica essa arquitetura.

## DIAGRAMAS

### Diagrama de entidade Relacional



### Diagrama de Classes

# DESENVOLVIMENTO DE SOFTWARE

## METODOLOGIA DE DESENVOLVIMENTO DE SOFTWARE

Deve ser descrita as etapas de desenvolvimento e os ambientes que foram utilizados (linguagens, ambientes de desenvolvimento/produção, bibliotecas principais)

### Ambientes de desenvolvimento/produção

### Bibliotecas principais

## MÓDULOS DO CRONOGRAMA DE DESENVOLVIMENTO

## MOCKUPS

# CONCLUSÃO

Nessa conclusão, deve-se conter:

* Uma breve descrição a respeito de todo o projeto e desenvolvimento.
* Dificuldades encontradas
* Trabalhos Futuros

# REFERÊNCIAS

TECHTUDO. **Como funciona um site de Sorteios.** Disponível em:<https://www.techtudo.com.br/noticias/2020/05/como-funciona-o-sorteiogram-conheca-

site-para-sorteios-no-instagram.ghtml/> Acesso em: 10 jun. 2021.

WEBMUSEUM. **A história do primeiro site publicado.** Disponível em: <https://museuweg.net/blog/conheca-a-historia-do-primeiro-site-publicado/> Acesso em: 10 jun. 2021.



Colocar em ordem alfabética

**ANEXO**

**MANUAL DO USUÁRIO**

**(NÃO PRECISA COLOCAR CAPA E FOLHA DE ROSTO DA UNIGRAN)**