



Sistema RH: Desenvolvimento de um sistema para gestão de pessoas, processos e atividades no departamento de recursos humanos.

**Samuel Hubner Tomaz
Luciana Rocha Cardoso**

**Curso: Análise e desenvolvimento de sistemas Período: 6º Área de
Pesquisa: Tecnologia**

Resumo: Esse trabalho visa apresentar um sistema *web* que tem como objetivo fazer o gerenciamento de pessoas, processos e atividades, auxiliando diretamente o departamento de recursos humanos. Tal sistema irá fazer uso de uma arquitetura cliente/servidor que trabalhará de maneira a auxiliar os operadores a fazerem uma melhor gestão dos funcionários da empresa em que trabalham, aumentando assim a organização e o controle dos mesmos. O *software* visa tornar todo o trabalho de gestão e controle fácil e intuitivo, apresentando uma interface simples de fácil uso, de modo a torná-lo um *software* acessível a todos. Tal *software* ficará disponível na rede mundial de computadores, podendo ser acessado de qualquer dispositivo que possua *internet* e um navegador *web*.

Palavras-chave: Controle, sistema, recursos humanos, pessoas, atividades.

1. INTRODUÇÃO

O departamento de Recursos Humanos (RH) possui papel fundamental para a empresa, pois tal departamento é responsável em oferecer suporte aos administradores e funcionários. Assim como todo departamento, o RH deve acompanhar a evolução na forma como empresa é gerida em um mundo onde a tecnologia da informação é um ponto fundamental. Um bom sistema é possível oferecer aos seus utilizadores ferramentas para que uma boa gestão seja feita. De modo a ser possível fazer um bom controle e planejamento de seus funcionários.

Um planejamento pode ser descrito como um plano que visa alcançar um determinado objetivo, tal planejamento consiste na administração e gestão de ferramentas que se relacionam com a estruturação e organização de um objetivo. (OLIVEIRA, 1999)

As empresas que contam com um departamento de RH, agregam valor e demonstram mais status e confiança para seu cliente, tendo em vista que contam com um departamento específico para cuidar, supervisionar e motivar os seus funcionários, GARCIA (2015).

O departamento de RH possui papel fundamental na empresa, por este motivo fazer uso de um sistema auxiliar pode ajudar e muito o rendimento de tal setor, de modo a melhorar a empresa como um todo.

Ao se observar a grande crescente no número de empresas abertas durante os anos, a busca pelo alto rendimento se tornaram cada vez mais recorrente nesse ramo. Diversas pessoas são contratadas e despedidas a todo momento, o alto número de empregados, atrelado ao constante aumento no número de pessoas buscando emprego, leva ao setor RH uma importante responsabilidade perante toda empresa,

tal setor deve ser eficiente na execução de suas tarefas para que toda a empresa funcione como o desejado.

A grande dificuldade do setor é descobrir uma maneira para que o fluxo de trabalho seja compatível com o fluxo de movimentação de funcionários. O RH tem como função gerir todo os recursos humanos da empresa, o que gera a pergunta: Quais os recursos que o setor de recursos humanos precisa para trabalhar de forma mais eficiente?

Com o intuito de responder tal questão, foi criado um *software* que tem como principal objetivo deixar todo o trabalho do RH mais eficiente e prático, onde é possível efetuar o controle de diversos itens como funcionários e empresa, tudo de maneira rápida, visando assim agilizar todo o fluxo de controle.

Para se ter um bom setor de RH, em primeiro lugar, é necessário investir em profissionais capacitados para a gestão dos recursos de uma empresa. Também vale a pena investir em softwares de gestão que facilitem a tarefa. (AUSLAND.2016).

Portanto o *software* desenvolvido justifica-se em entregar para os setores de RH uma ferramenta rápida e prática cujo seu principal objetivo é garantir uma maior eficiência no processo de gestão do mesmo.

2. REFERENCIAL TEÓRICO

Nesta seção é possível encontrar conceitos das técnicas, ferramentas e tecnologias utilizadas no processo de planejamento e desenvolvimento do *software*. Além de encontrar o conceito de Recursos Humanos (RH).

2.1. Recursos Humanos (RH)

Recursos Humanos, ou também conhecido como RH, pode ser definida como todas as pessoas que fazem parte de uma determinada organização. Dentro dessa área existe a gestão de Recursos Humanos, tal gestão fica responsável por administrar os funcionários visando sempre em alinhar os objetivos da empresa de modo a desenvolver uma satisfação profissional. É possível citar algumas funções que a gestão de RH faz, como: Seleção e Recrutamento de funcionários, definição de salários e cargos e o desenvolvimento corporativo e pessoal.

Conforme dita por Ana Paula Coutinho de Sousa (2009):

Ao DRH compete: manter atualizados os dados cadastrais dos servidores e demais documentos pertinentes ao RH, compilando informações quando solicitadas; controlar a frequência dos funcionários; examinar, emitir pareceres e informações em processos que envolvam matéria relativa à legislação de pessoal; promover a seleção e recrutamento de pessoal para o provimento de cargos e preenchimentos de empregos (SOUSA, 2009)

Chiavenato (1975, *apud* ALMEIDA, TEIXEIRA, MARTINELLI,1993, p.17) entende que a Gestão de Recursos Humanos "consiste no planejamento, na

organização, no desenvolvimento, na coordenação e no controle de técnicas capazes de promover o desempenho eficiente do pessoal.”

2.2. Linguagem de modelagem unificada (UML)

A UML visa auxiliar na observação do sistema e seus relacionamentos na forma de diagramas, apoiando a prática de engenharia de *software* orientada a objetos e fazendo uso de técnicas que apresentam a visualização lógica e estrutural, de forma abstrata e intuitiva, sobre os componentes e os processos da solução a ser desenvolvida e é utilizada por projetistas de sistemas que utilizam o paradigma orientado a objetos para a modelagem de dados (JACOBSON, 2005, BEZERRA, 2015).

Os diagramas da UML podem ser divididos em Diagramas Estruturais e Diagramas Comportamentais.

2.2.1. Diagramas Estruturais

Segundo Jacobson (2005), os diagramas estruturais são:

De Componentes: Associa-se à linguagem de programação, sua finalidade é indicar os componentes do *software* e seus relacionamentos. Esse diagrama é fundamental e o mais utilizado na UML, o diagrama de componentes é apresentado no apêndice A através da figura 9.

De Pacotes: Representa os subsistemas englobados de forma a determinar as partes que o compõem. VIDAL (2003) entende que os diagramas de pacotes possuem como objetivo fazer a representatividade de itens como os sistemas, subsistemas, as camadas e domínios do mesmo. Em seu texto ele afirma: “O Diagrama de Pacotes serve para mostrar como as Classes são divididas dentro de módulos; são diagramas lógicos e não necessariamente implicam numa divisão física de classes.”. Portanto, tais pacotes podem ser utilizados para apresentar de forma distinta as diferentes visões que a arquitetura do sistema pode apresentar.

De Objeto: O diagrama de objeto se relaciona com o diagrama de classes, é praticamente um complemento dele. GUEDES (2014) afirma em seu livro que os diagramas de objetos: “Fornecem uma visão dos valores armazenados pelos objetos de um Diagrama de Classe em um determinado momento da execução do software a ser desenvolvido.” Tal diagrama trabalha com os vínculos entre si, de modo a serem instâncias das associações presentes no diagrama de classes. O mesmo é representado na Apêndice A através da figura 5.

De implantação: Determina as necessidades de *hardware* e características físicas do Sistema.

De Estrutura: Descreve a estrutura interna de um classificador. Estes diagramas representam as partes estáticas da solução e podem ser elementos conceituais ou físicos do mundo real.

De Classe: Este diagrama é fundamental e o mais utilizado na UML. Ele serve de apoio aos outros diagramas e mostra o conjunto de classes com seus atributos.

Antônio Geraldo da Rocha Vidal (2003) descreveu que:

O Diagrama de Classes é um produto de modelagem de estruturas de informação, que tem como objetivo fornecer uma visão estática do Sistema, em termos das classes (descrevendo suas propriedades (dados e conteúdo) e seu comportamento) e dos vários tipos de relacionamentos entre elas, em um alto nível de abstração.

Um diagrama de classe pode ser estruturado em classes, atributos, operações e relacionamentos. VIDAL (2003) refere-se a classes como um objeto do mundo real, onde o mesmo é estruturado no sistema através de atributos e comportamentos que são identificados por um nome. Os atributos por sua vez, servem para compor a classe e descrever todas suas propriedades. As operações tem como principal objetivo fazer a manipulação das propriedades dos objetivos que compõem uma classe. Por fim, os relacionamentos fazem as conexões entre os objetos, possuindo informações como a multiplicidade. Tal diagrama é representado na Apêndice A através da figura 4.

2.2.2. Diagramas Comportamentais

Segundo Jacobson (2005), os diagramas comportamentais são:

De Máquina de Estados: Visa acompanhar mudanças sofridas por um objeto dentro de um processo.

De Caso de Uso: Geral e informal para fases de levantamento e análise de Requisitos do Sistema. Vidal (2003) descreve os diagramas de casos de uso como produtos de modelagem funcional que representam um sistema como um agrupado de interações com os vários atores presentes do mesmo. Tal diagrama é constituído por alguns elementos básicos como:

- **Ator:** O ator é considerado um agente externo onde tal agente faz a interação direta com o sistema e é sempre representado por um boneco estilizado
- **Use Case:** Esse elemento visa representar um grupo de ações que o sistema tende a executar, tal ação só é executada a partir de uma interação do ator, gerando um resultado final que é redirecionado para um ou mais atores. Vidal (2003) afirma que: “As ações executadas dentro de um Use Case podem envolver comunicação com vários outros atores, cálculos e funções realizadas pelo Sistema.” O *use case* geralmente é representado por uma elipse com a descrição da ação em seu interior.
- **Associação:** Tal elemento tende a representar uma ligação simples que mostra a comunicação de um ator com um use case.

De Atividades: Descreve os passos a serem percorridos para a conclusão de uma atividade.

De Interação: Nome atribuído a diagramas de sequências e de comunicação.

2.3. Sistema de Gerenciamento de Banco de Dados (SGBD)

Os SGBDS podem ser considerados sistemas computadorizados que visam armazenar informação de modo a permitirem que o usuário busque e atualize essas informações. Os SGBDS foram criados com o intuito de servir como um auxílio para os desenvolvedores e/ou indivíduos que precisam fazer a manipulação de dados em presentes em grandes bancos, auxiliando assim todo o processo de execução da atividade (DATE, 2004).

Segundo Rob e Coronel (2011), no que diz respeito ao SGBD:

O sistema de gerenciamento de bancos de dados (SGBD) é um conjunto de programas que gerenciam a estrutura do banco de dados e controlam o acesso aos dados armazenados. Até certo ponto, o banco de dados se assemelha a um arquivo eletrônico com conteúdo muito bem organizado com a ajuda de um software poderoso, conhecido como sistema de gerenciamento de banco de dados.

2.4. Mysql

O MySQL pode ser definido como um SGBD que faz uso da linguagem SQL (Linguagem de consulta estruturada) como interface. Tal SGBD suporta alguns recursos como o *Unicode*, *Hot Backups* e outros. O MYSQL é um dos bancos de dados mais utilizados em aplicações voltadas à web. Segundo Milani (2007) o MySQL se destaca por ser um SGBD extremamente rápido e confiável, sua rapidez se dá pelo fato de o mesmo armazenar os dados da tabela em modo ISAM, que por sua vez apresenta um código de baixo nível.

O MySQL se destaca também por ser uma aplicação muito portátil, Milani (2007) afirma o seguinte sobre sua portabilidade: “Desenvolvido utilizando as linguagens de programação C e C++, unido com o uso de *GNU Atomake*, *Autoconf* e *Libtool*, torna o MySQL uma aplicação altamente portátil”.

O MySQL foi escolhido por motivos de ser uma aplicação eficiente e simples, além de funcionar em diversos sistemas operacionais como *Windows* e *Linux* e ser extremamente rápido.

2.5. Arquitetura Cliente Servidor

Tal arquitetura pode ser encontrada em uma aplicação distribuída onde são distribuídas tarefas entre o servidor e a aplicação cliente.

A aplicação cliente ou também conhecida por “*front-end*” pode ser definida como um processo que se inter-relaciona com o usuário, por meio gráfico geralmente, ou seja, nela o usuário efetua ações que serão interagidas com o servidor. Já os servidores são conhecidos como “*back-end*” e tem como principal função receber e responder as requisições enviadas pelo cliente. (MCKIE, 1997, RENAUD, 1993).

Essa arquitetura foi utilizada no desenvolvimento do sistema por ser a arquitetura usada em aplicações *Web*, além da mesma ser projetada de modo a garantir uma facilidade de uso e segurança.

2.6. Arquitetura modelo-visão-controlador (MVC)

Pode ser definida em uma arquitetura de *software* que visa dividir a aplicação em três principais camadas: visão (*view*), modelo (*model*), e controlador (*controller*). Tal arquitetura era muito utilizada em aplicações voltada para *desktops*, porém acabou sendo muito utilizado em aplicações *Web*, pelo fato dos grandes números de *Frameworks* que implementam tal arquitetura.

Segundo GAMMA (2000), a camada modelo (*model*) fica responsável por gerir todos os dados da aplicação. Atendendo às requisições para efetuar as alterações de dados já armazenados. A camada visão (*view*) fica responsável por gerenciar a lógica das apresentações, de modo a fornecer uma interface gráfica para aplicação, gerenciando também a apresentação de informações ao usuário. A camada controladora (*controller*) fica responsável por gerenciar toda a lógica de negócio, recebendo as informações de entrada do usuário, e retornando tais interações para a camada modelo ou para a própria camada visão.

Tal estrutura foi escolhida no desenvolvimento pelo fato de:

- Da divisão da aplicação ser em camadas: a primeira camada é a manipulação da lógica de dados (Modelo ou *Model*), a segunda camada é a de interface com o usuário (Visão ou *View*) e a terceira camada é a de lógica de negócios (Controlador ou *Controller*).
- A possibilidade de uma mesma lógica de negócios através de várias interfaces.
- O isolamento da camada Modelo das demais camadas do sistema, visando facilitar a sustentabilidade do código.
- A implementação do controlador permite o recebimento dos eventos da interface e os converte em ações na camada Modelo.
- Se um meio para utilizar o reaproveitamento de código.

Luciano e Alves (2011) afirmam o seguinte sobre a arquitetura MVC:

A proposta é útil para possibilitar ao usuário visualizar o mesmo modelo sob diferentes contextos e/ou ponto de vista. No MVC as entradas do usuário, o modelo do mundo real e a interface são explicitamente separados e tratados por três tipos de objetos, onde cada um é especializado em suas próprias tarefas.

2.7. Programação orientado a objetos

O paradigma de programação orientado a objetos prioriza elementos conceituais do domínio do problema, além de auxiliar no projeto e implementação de aplicações, os modelos de classe advindos da UML, são utilizados para a compreensão de problemas entre os envolvidos no desenvolvimento das aplicações. Neste paradigma, são implementadas um conjunto de classes que definem os objetos que irão compor o *software*. Nestas classes são determinados os comportamentos (métodos) e estados (atributos) de seus objetos, além dos relacionamentos entre eles (JACOBSON, 2005, BEZERRA, 2015).

A utilização da programação orientada a objetos se deu pelo fato de a mesma tornar todo o código mais fácil de modificar e dar manutenção, além de que os

sistemas desenvolvidos com base na orientação de objetos são mais fáceis de ler e de entender.

2.8. PHP

É uma linguagem de *script open source*, interpretada no lado servidor, muito utilizada, de fácil aprendizado e especialmente desenvolvida para a *Web* por possibilitar escrever *scripts* dentro de marcações HTML. O diferencial do PHP em relação a outras linguagens é a variedade de banco de dados que ele é capaz de manipular, utilizando extensões específicas ou uma camada de abstração como o PDO (*PHP Data Object*) (WELLING, 2005).

A linguagem PHP foi escolhida para o desenvolvimento do *software* pelo fato da mesma ser simples, fácil de aprender, eficiente e possuir uma vasta lista de conteúdo sobre o assunto. O PHP será utilizado para a criação de toda a parte “*back-end*” do sistema, ele ficará responsável por gerir as regras de negócios, validar dados e efetuar a comunicação com o banco de dados.

2.8.1. PDO

Segundo o PHP (2001), a interface PDO basicamente permite a comunicação com qualquer banco de dados, fazendo uso de *drivers* específicos de cada banco de dados para realizar as comunicações.

2.8.2. *Laravel Framework*

O *Laravel* é um *Framework* PHP que utiliza a arquitetura MVC com o intuito de auxiliar o desenvolvimento de aplicações proporcionando segurança, códigos limpos e simples, performance e agilidade na codificação, pois, emprega o uso de boas práticas de programação utilizando o padrão PSR-2 onde estão definidas propostas para uma melhor codificação com a linguagem PHP. Com o intuito de criar as interfaces gráficas, o *Laravel* faz uso de uma *Engine de template*, o *Blade*, que possui inúmeras ferramentas que auxiliam na criação de interfaces amigáveis e funcionais de maneira rápida, além de evitar a duplicidade de códigos.

Para realizar a comunicação com o Banco de Dados o *Laravel* faz uso do *ActiveRecord* chamada de *Eloquent* ORM, trazendo funcionalidades para facilitar as operações de DML. Com uma simples configuração e pouco código é possível configurar a conexão com Banco de Dados e começar o trabalho. (LARAVEL, 2015).

O sistema foi desenvolvido fazendo uso do *Laravel* pois ele facilita o desenvolvimento de diversos pontos como:

- Com ele é fácil criar um sistema de rotas intuitivo.
- Ele faz uso do *Composer* para gerenciar suas dependências, o que permite uma rápida configuração do ambiente.
- Possui uma vasta gama de conteúdo sobre o mesmo.
- Possui uma documentação de fácil entendimento.
- Grande comunidade de desenvolvedor com ampla participação da comunidade

O Laravel se destaca também por ser o mais popular em comparação às outras opções de framework quando se fala em PHP. Segundo o GOOGLE TRENDS (2021) o Laravel manteve a liderança absoluta quando se diz a respeito de interesses de busca por frameworks PHP. Em uma análise feita durante os últimos 5 anos, o Laravel possui uma porcentagem de 75% de interesse em relação a todas opções, tal fato mostra o quão grande e popular é o framework Laravel.

2.9. HTML

HTML (*HyperText Markup Language* - Linguagem de Marcação de HiperTexto), inserida na *Web* como possibilidade de interligação entre documentos da *Web* (SILVA, 2011), é a base de todas as páginas da *Web*, pois o mesmo descreve e define os conteúdos de uma página e como estas se conectarão a outras páginas do site ou a outros *websites* através de *links*, o que é fundamental para *Web*. O principal objetivo do HTML está na difusão da informação publicada, tornando possível acessá-la em qualquer dispositivo que possua uma conexão com a internet e um navegador de internet.

O HTML ficou responsável pela criação de todo presente na tela, desde campos para preenchimentos, até as ancoras usadas a navegação entre as diferentes páginas.

2.10. CSS

As folhas de estilo em cascata (*Cascading Style Sheet* - CSS), possuem a finalidade de apresentação dos elementos HTML no que tange, cor, formato, posicionamento e todos os demais aspectos visuais do *site*. A programação se dá no lado do cliente, e podem ser escritas no documento HTML ou em arquivos externos do tipo "css" que são incorporados na página através de um *link* (SILVA, 2012).

O CSS ficou responsável por efetuar toda a estilização.

2.11. JAVASCRIPT

Segundo (SILVA, 2010), o *JavaScript* foi desenvolvido com o intuito de possibilitar interatividade as páginas HTML e possui um motor incorporado nos navegadores, ou seja, seu funcionamento está condicionado as funcionalidades hospedadas no *browser* do usuário é a principal linguagem de programação no lado cliente em navegadores *Web*. É uma linguagem de *scripts* que se baseia em orientação a objetos por prototipagens e possui tipagem de dados fraca e dinâmica, além de suporte à programação funcional.

Você pode usar *Java Script* para ajudar os usuários a interagirem melhor com páginas *Web*. O *Java Script* também pode ser usado para controlar o navegador, comunicar-se de forma assíncrona com o servidor, alterar de forma dinâmica o conteúdo de uma página *Web*, e para desenvolver jogos e aplicações móveis e *desktop*. O *Java script*, sendo uma linguagem de *script* não pode funcionar por conta própria.

Ele é executado, de fato, nos navegadores dos visitantes. (PRESCOT. 2016).

2.12. O software e seus requisitos

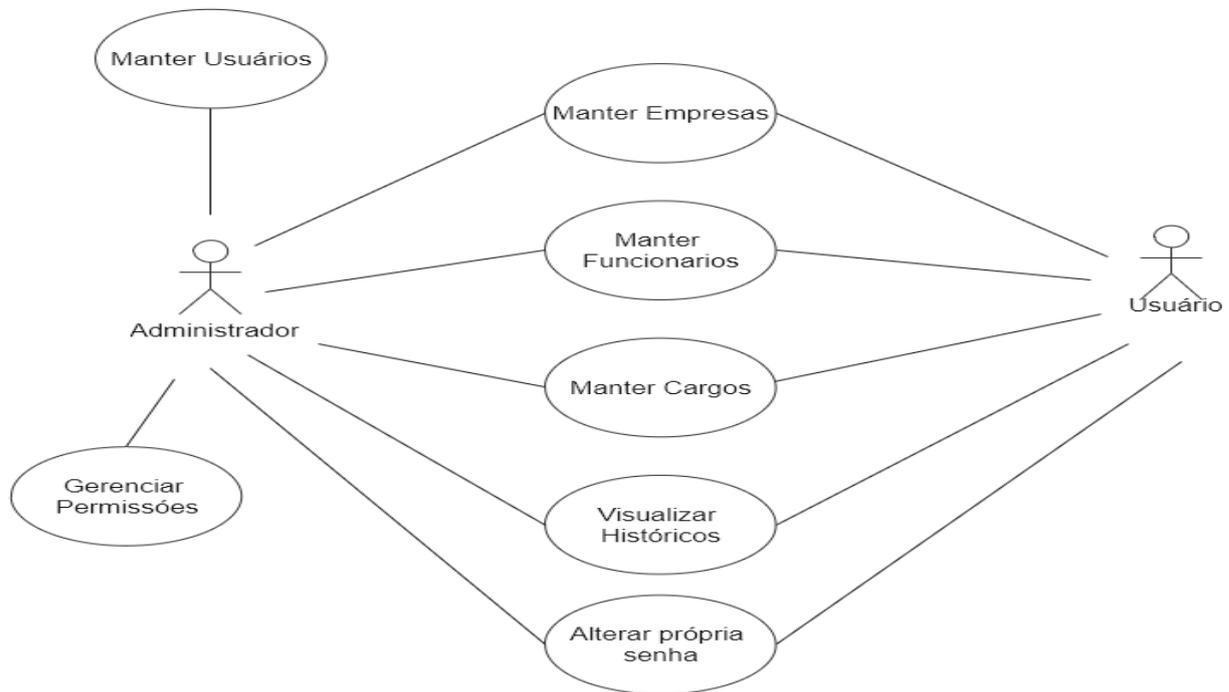
A usabilidade de um *software* é ponto chave para definir se o mesmo é um bom produto ou não. “A usabilidade é capacidade do sistema em fazer com que o usuário tenha sucesso na execução de suas tarefas. Fácil aprendizagem, utilização eficiente e gestão de erros são pontos fundamentais para que o usuário perceba a boa usabilidade.”(TECLÔGICA,2012).Portanto, o sistema em questão conta com um *design* responsivo e intuitivo para que o usuário possa efetuar com facilidade a execução de suas tarefas.

O foco na usabilidade foi um requisito não funcional apresentado no sistema, referente a isso alguns requisitos funcionais e não funcionais foram levantados a partir do planejamento. Dentre os requisitos funcionais estão manter diversas ferramentas do sistema, como o cadastro de usuários, empresas e funcionários, além de ser necessário autenticar o usuário. Já os requisitos não funcionais então por exemplo o foco na usabilidade para facilitar a maneira com que o usuário possa executar diferentes ações dentro do sistema.

Após os levantamentos dos requisitos funcionais e não funcionais, foram elaborados os Diagramas de caso de Uso e o Diagrama de Classe. Tais diagramas são fundamentais para o desenvolvimento do sistema.

O diagrama de caso de uso apresentado na figura 1 estão descritos os principais atuantes do sistema e suas funcionalidades, bem como o relacionamento existente entre eles, também mostra os relacionamentos existentes entre funcionalidades.

Figura 1 – Diagrama de caso de uso



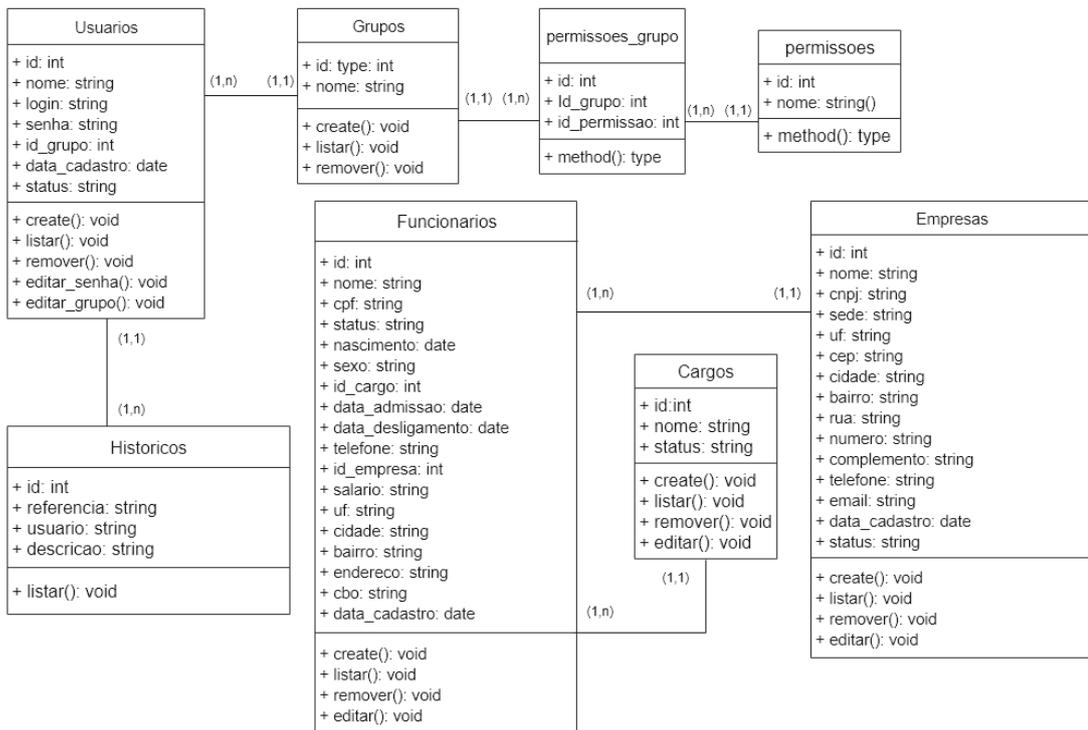
Fonte: Acervo Pessoal

O Administrador pode manter os usuários, que por sua vez deverão passar por um processo de autenticação para poder acessar o sistema. Além disso, o administrador pode fazer o gerenciamento das permissões dos usuários, de modo a definir qual parte do sistema o grupo de usuários acessa. Ele também é capaz de manter todos os tipos de cadastros, podendo fazer qualquer alteração se necessário.

Os usuários conseguem manter diferentes recursos, como o cadastro de cargos, funcionários e empresas, tais operadores conseguem também mudar sua respectiva senha, desde de que o mesmo informa sua antiga senha. Tanto os usuários, quanto os administradores podem acessar os históricos referentes aos cadastros e edições feitas dentro do sistema.

Na figura 2 mostra o diagrama de Classe, esse diagrama permite a visualização das entidades, seus atributos e métodos, que compõem o sistema. É um modelo gráfico da UML que apresenta as entidades que compõem o domínio do problema. As entidades são elementos obtidos através da reflexão da abstração do problema a ser solucionado, no contexto real, para o cenário lógico. Este diagrama serve como base para a elaboração de outros diagramas da linguagem UML, e é composto por classes e suas associações existentes, ou seja, os relacionamentos entre elas (JACOBSON, 2005)

Figura 2 – Diagrama de classe



Fonte: Acervo Pessoal

3. METODOLOGIA

Nesta seção serão apresentadas técnicas, ferramentas e linguagens de marcação de texto e programação envolvidas durante o processo de desenvolvimento do sistema proposto.

Para atender ao objetivo proposto, este trabalho foi desenvolvido baseado em conceitos e técnicas de aplicações *Web*, com o intuito de disponibilizá-la para acesso em qualquer dispositivo eletrônico contemplado de um navegador de internet e acesso à rede mundial de computadores. Esta análise gerou um modelo de arquitetura de *software*, fundamentado no paradigma orientado a objetos, utilizando de diagramas da UML, os modelos que serão apresentados neste trabalho foram definidos levando em consideração o Documento de Especificação de Requisitos, e detalha os diagramas da UML, necessários para modelar a estrutura e o comportamento da solução proposta.

Estes diagramas contemplam representações gráficas apresentando uma visualização lógica e estrutural, de forma abstrata e intuitiva, sobre os componentes e os processos da solução a ser desenvolvida. Inicialmente foi desenvolvida toda a lógica do servidor que abrange um banco de dados Oracle MySQL, por se tratar de um sistema gerenciador de banco de dados (SGBD) de código aberto e livre, além de agregar a linguagem de consulta estruturada (SQL), este SGBD garantirá a integridade e segurança dos dados persistidos.

A linguagem de programação que realiza a comunicação com o SGBD é o PHP por ser especialmente desenvolvida para web, *open source* e trabalhar com qualquer banco de dados utilizando uma camada PDO, que basicamente permite a comunicação com qualquer banco de dados, fazendo uso de drivers específicos de cada banco de dados para realizar as comunicações. Conforme a linguagem escolhida para a comunicação com o SGBD, este desenvolvimento contempla de um *framework*, o Laravel 8.10, que possibilita a implementação do padrão de arquitetura de software MVC (Modelo – Visão – Controlador), que descreve como deve ser estruturada uma aplicação e que esta deve ser dividida em camadas e cada qual possui funções específicas. A camada modelo é responsável por manter as informações relacionadas ao domínio e os objetos que implementam as principais funcionalidades do sistema. Na camada Visão encontram-se os objetos relacionados à apresentação, ou seja, as interfaces gráficas com o usuário. A camada Controlador define a maneira como a Visão (interface gráfica) deve atuar, com base nas informações recebidas do usuário, além de manter as informações e o estado dos objetos da camada Modelo atualizados.

No desenvolvimento no lado do cliente está sendo empregada a tecnologia de marcação de textos HTML por permitir marcações como campos de textos, tabelas e botões, que serão interpretados pelo navegador de internet. O CSS (*Cascading Style Sheets*) é empregado para estilizar todas as páginas HTML do sistema com o intuito de torná-las mais amigáveis ao usuário. Outra linguagem muito utilizada é o *JavaScript* que permite que scripts sejam executados no browser do cliente objetivando fornecer interatividade às páginas HTML, por possuir ferramentas que se destinam a fornecer funcionalidades para simplificar a tarefa, esta linguagem vem sendo empregada para validação de formulários, apresentação de caixas de alerta e informações de forma dinâmica.

O tipo de pesquisa utilizado no trabalho de conclusão de curso foi a pesquisa exploratória, pois através dela foi feito o aprimoramento do conhecimento em ferramentas novas, como o *framework* laravel e o padrão MVC.

A coleta de dados foi feita através da coleta por observação, o material foi obtido de sites, fórum e vídeos que ensinam e exemplificam o uso de diversas ferramentas usadas no desenvolvimento do *software*.

4. DISCUSSÃO DE RESULTADOS

O levantamento dos requisitos se deu por meio de pesquisas sobre a área de Recursos Humanos, com isso foi possível levantar os requisitos funcionais e não funcionais ligados ao sistema de RH.

Entende-se por MANTER o ato de cadastrar, editar, pesquisar e excluir registros.

QUADRO 1 – Requisitos Funcionais

Id	Descrição
RF-1	Autenticar Usuários
RF-2	Manter Usuários
RF-3	Manter Empresas
RF-4	Manter Funcionários
RF-5	Manter Cargos
RF-6	Alterar Senha

Fonte: Acervo pessoal.

O levantamento de requisitos não funcionais se deu pela pesquisa de ferramentas mais utilizadas para o desenvolvimento *Web*.

QUADRO 1 – Requisitos Não Funcionais

Id	Descrição
RNF-1	O sistema deve ser desenvolvido fazendo uso da arquitetura <i>WEB</i> .
RNF-2	O sistema deve fazer uso do banco de dados MySQL
RNF-3	O sistema deve usar o <i>Framework</i> Laravel.

Fonte: Acervo pessoal.

5. CONCLUSÃO

O sistema foi desenvolvido para ser de fácil uso a usuários leigos, o mesmo é fácil e intuitivo, de modo com que qualquer pessoa que faça uso dele consiga usufruir de todos os recursos de maneira simples, rápida e segura.

Como principal forma de estudo de caso e testes foi escolhida a funcionalidade de gerenciamento de contas e por meio de suas telas foram feitos testes de caixa preta, segundo Pressman (2005) os testes de caixa preta não levam em consideração os comportamentos internos do *software*, e sim os aspectos de usabilidade com participação direta dos usuários, com intuito de identificar se o acesso as funcionalidades se dão de maneira simples. Observando os usuários acessarem estes recursos, é notável que os padrões seguidos para o desenvolvimento desta solução foram bem aplicados, pois, após pouco tempo de uso eles conseguiram memorizar e compreender as principais funcionalidades do sistema.

A solução apresentada se propôs a criar um sistema para efetuar o gerenciamento de funcionários. Tal sistema cumpre com seus objetivos, pois através dele é possível gerenciar as empresas, funcionários, grupos, cargos, usuários e outros.

6. TRABALHOS FUTUROS

Como trabalhos futuros, é possível efetuar a implementação de um recurso para lançar multas e advertências á funcionários, além da criação de relatórios gerais que visam melhorar e facilitar o controle de tais funcionários.

A solução também será capaz de lançar férias e afastamento para cada funcionário, permitindo assim um maior controle de quem está trabalhando ativamente no período desejado.

7. REFERÊNCIAS

ALMEIDA, M.; TEIXEIRA, M.; MARTINELLI, D. Por que administrar estrategicamente recursos humanos?. São Paulo. 1993

AUSLAND. A importância de uma gestão eficiente de Recursos Humanos. Disponível em: < <https://ausland.com.br/blog/a-importancia-de-uma-gestao-eficiente-de-recursos-humanos/>>. Acesso em: 29 de jun. de 2021

BEZERRA, E. Principios de Análise e Projeto de Sistemas com UML. 3.ed. Rio de Janeiro-RJ: Ed. Elsevier, 2015. ISBN 9788535226263.

BOOCH, G.; RUMBAUGH J.; JACOBSON I., UML: Guia do Usuário. 2ª edição. Rio de Janeiro. Elsevier. 2005.

CHIAVENATO, I. Introdução à teoria geral da administração. São Paulo; McGraw Hill do Brasil, 1975.

DATE, C. J. Introdução a sistemas de banco de dados. Elsevier Brasil, 2004.

GARCIA, Alessandro. Qual a importância do departamento de recursos humanos no contexto das empresas? Disponível em: <https://www.rhportal.com.br/artigos-rh/qual-a-importancia-do-departamento-de-recursos-humanos-no-contexto-das-empresas>. Acesso em: 12 de março de 2021.

GAMMA, E.; HELM R.; JOHSON R.; VLISSIDES J., Padrões de Projeto: Soluções reutilizáveis de software Orientado a Objetos. Porto Alegre: Bookman, 2000.

GUEDES, Gilleanes T. A. UML 2 – Guia Prático - 2ª Edição. Novatec, 20 de jan. de 2014.

LARAVEL. Laravel – Instalação – Laravel – O framework PHP para artesãos na web. 2015. Disponível em: Acesso em: <<https://laravel.com/docs/>> 30 de out. 2021.

LUCIANO, Josué.; ALVES, Wallison Joel Barberá. Padrão de Arquitetura MVC: MODEL-VIEW-CONTROLLER. Bebedouro, 2011

MILANI, André. MySQL – Guia do Programador. Novatec Editora. São Paulo, 8 de jan. de 2007.

MCKIE, Stewart “Everything you wanted to know about Client/Server computing but were afraid to ask” Eye on Technology. 1997

OLIVEIRA, Djalma de Pinho Rebouças de. Planejamento Estratégicos: Conceitos Metodologia e Práticas. São Paulo. Atlas, 1999.

PRESCOTT, Prestron. Programação em Javascript.. Babelcube Inc. 2016

PRESSMAN, R. S., Software Engineering: A Practitioner’s Approach, McGraw-Hill, 6th ed, Nova York, NY, 2005

PHP. PHP – Introdução - Manual. 2001. Disponível em: <http://php.net/manual/pt_BR/intro.pdo.php> Acesso em: 30 de jun. 2021.

RENAUD, P. E. Introdução aos Sistemas Cliente/Servidor IBPI Press, RJ 1993.

ROB, Peter.; CORONEL, Carlos. Sistemas de Banco de Dados. 2011

SILVA, M. S. HTML 5: A Linguagem de Marcação que Revolucionou a Web. 1. ed. São Paulo-SP: Novatec, 2011. ISBN 9788575222614.

SILVA, M. S. CSS3: Desenvolva Aplicações Web Profissionais com uso dos Poderosos Recursos de Estilização das CSS3. 1. ed. São Paulo-SP: Novatec, 2012. ISBN 9788575222898.

SILVA, M. S. JavaScript: Guia do Programador. 1. ed. São Paulo–SP: Novatec, 2010. ISBN 9788575222485.

SOUSA, Ana Paula Coutinho de.; As práticas de gestão de pessoas, processos, atividades e controles no departamento de recursos humanos: Estudo de caso da empresa Reason Tecnologia S.A. Florianópolis. 2009

TECLÔGICA.O que é usabilidade. Teglôgica, 2012. Disponível em: <<https://blog.teclogica.com.br/o-que-e-usabilidade/>>. Acesso em: 29 de jun. de 2021.

TRENDS, GOOGLE. Gráfico de interesses. 2021. Disponível em: <<https://trends.google.com.br/trends/explore?q=laravel,Codelgniter,symfony,cakephp,Zend>>. Acesso em: 06 de jul de 2021

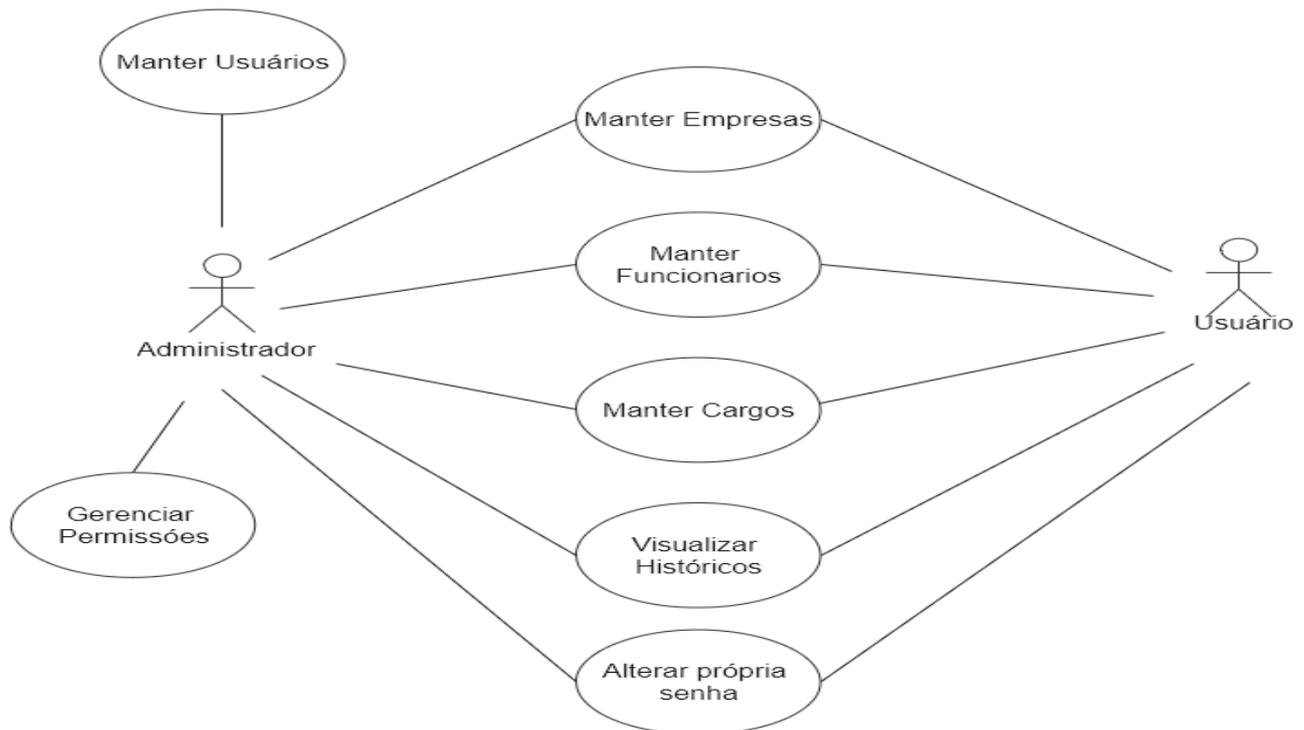
VIDAL, Antônio Geraldo da Rocha. Projeto Físico de Sistemas de Informação. Agosto de 2003.

WELLING, L.; LAURA, T. PHP e MySQL desenvolvimento web. 3. ed. Rio de Janeiro-RJ: Elsevier, 2005.

APÊNDICE A

a) Diagrama de Caso de Uso

Figura 1 – Diagrama de casos de uso



Fonte: Acervo pessoal

O diagrama de casos de uso tem como objetivo ilustrar a maneira como os casos apresentados são interligados nos sistemas e aos usuários. Tais usuários são representados no diagrama como atores, tais atores são administradores e usuário.

Ator: **Administrador**

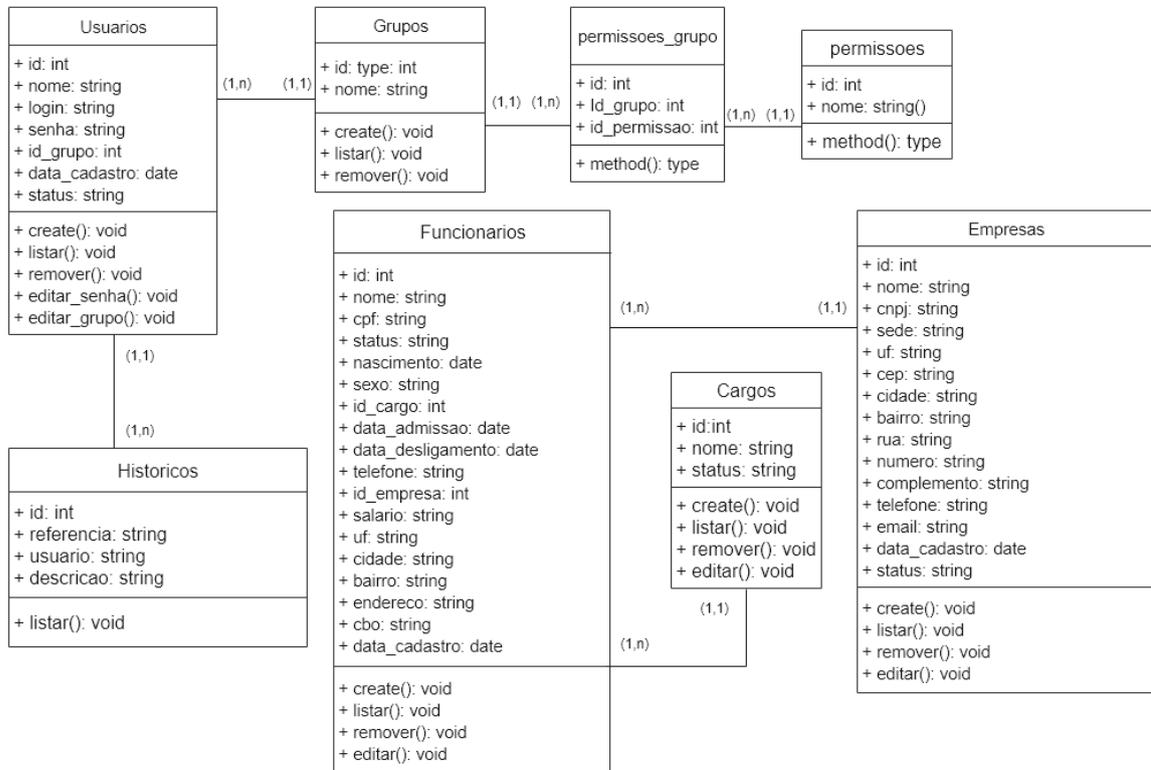
O administrador tem acesso total ao sistema, o mesmo pode manter todos os recursos nele presente, como manter cargos, funcionários e empresas. Além de ser o responsável por cadastrar novos usuários e gerenciar as permissões do mesmo.

Ator: **Funcionário**

O funcionário ficará responsável por gerenciar as principais ferramentas do sistema, como por exemplo, cadastrar, editar e remover usuário. O mesmo poderá também visualizar o histórico de cadastro e edição de tais ferramentas.

b) Diagrama de Classes

Figura 2 – Diagrama de classes



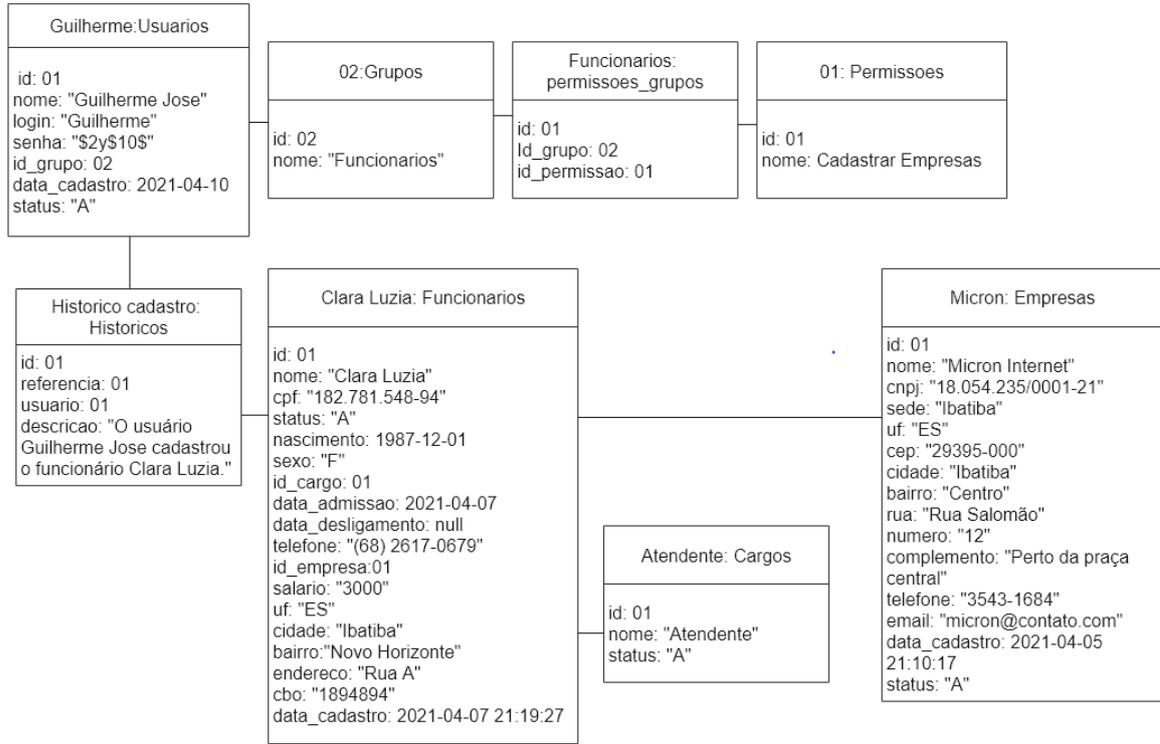
Fonte: Acervo pessoal

O diagrama de classes tem como objetivo mostrar os relacionamentos presentes no sistema, além de exibir os campos presentes em cada respectiva tabela no banco de dados. Em tal diagrama é possível observar por exemplo, o relacionamento que a tabela Funcionarios possui com as tabelas de empresas e cargos.

c) Diagrama de Objetos

O diagrama de objetos tem como objetivo apresentar com detalhes algumas instâncias presentes no sistema, além de mostrar as ligações entre as principais tabelas.

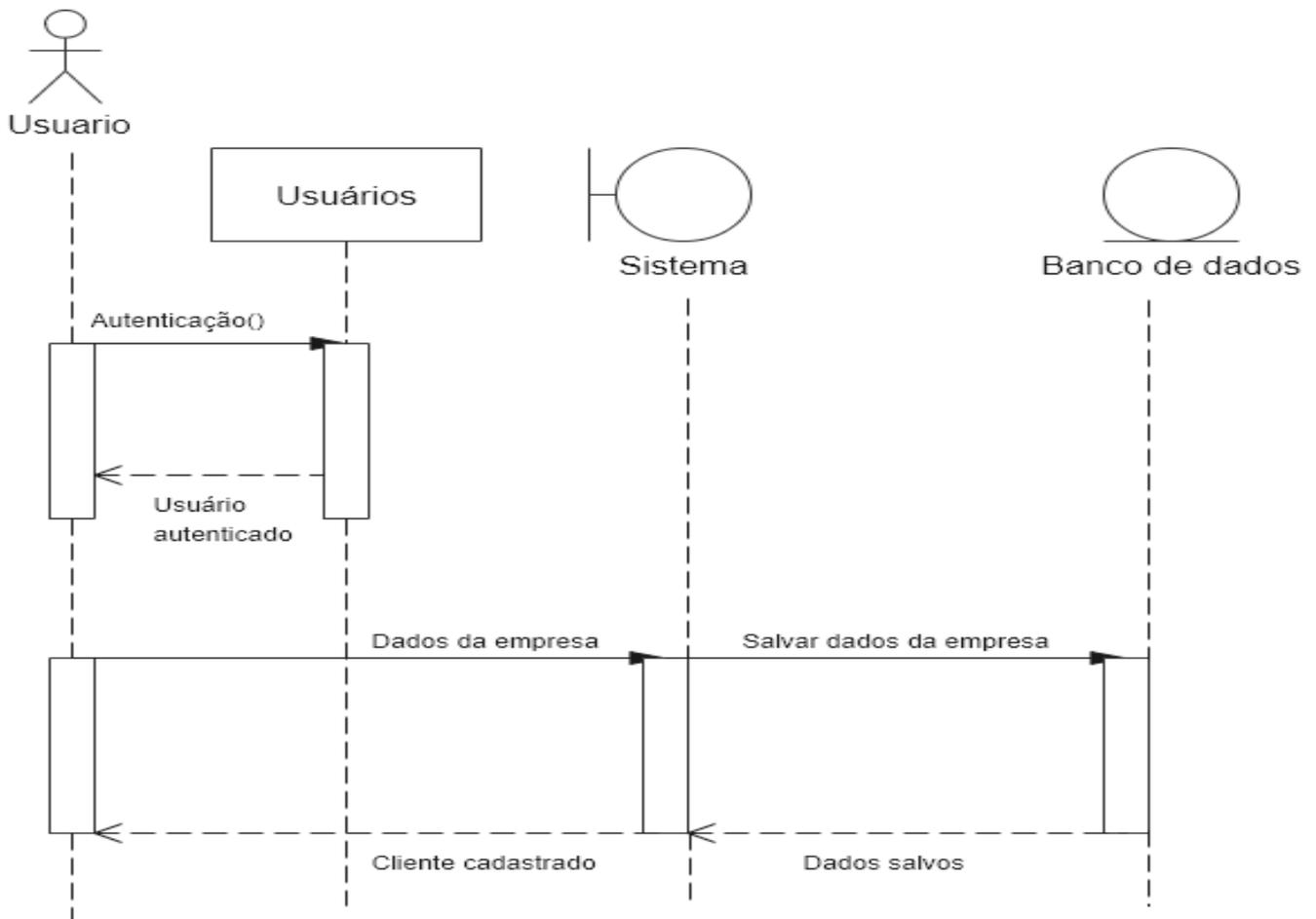
Figura 3 – Diagrama de objetos



Fonte: Acervo pessoal

d) Diagrama de Sequência

Figura 4 – Diagrama de Sequência

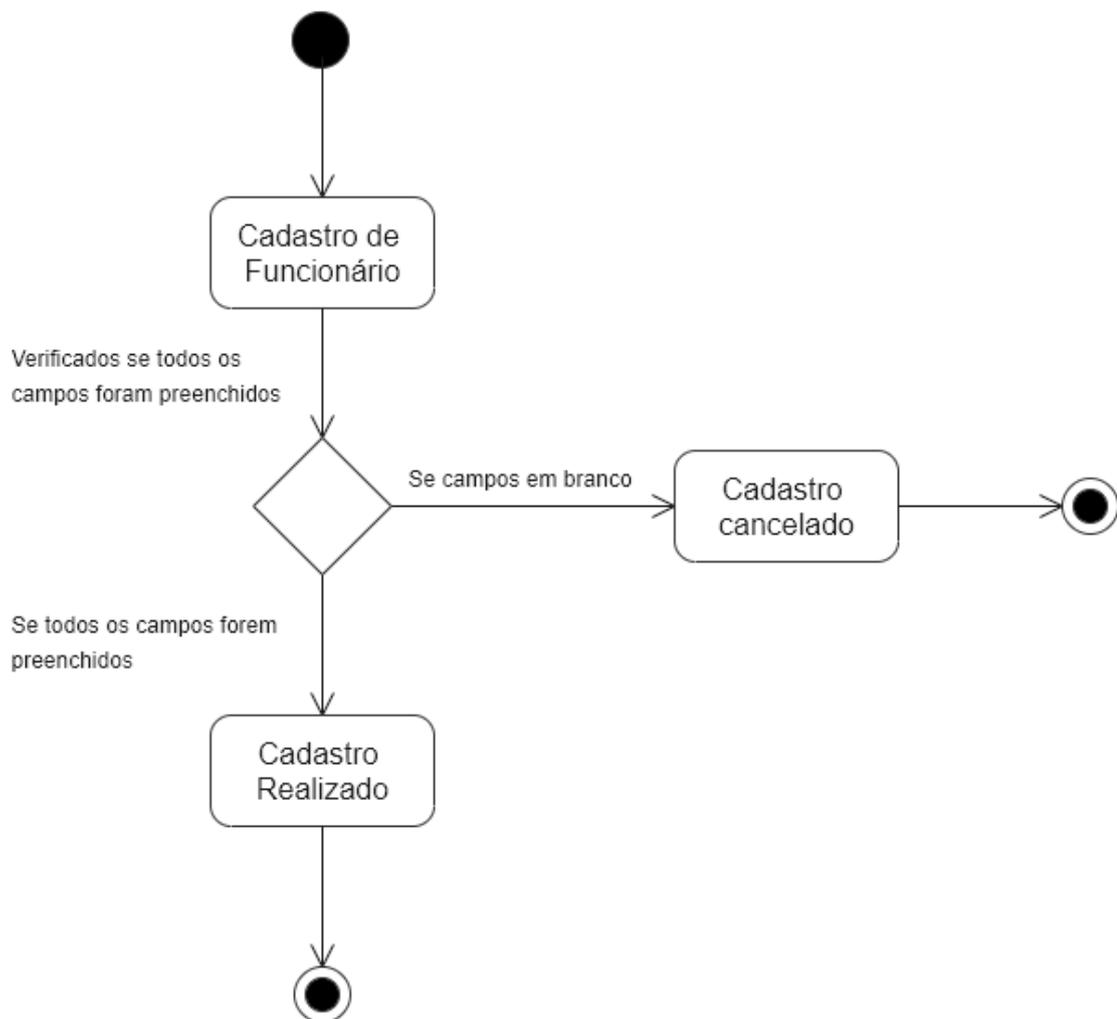


Fonte: Acervo pessoal

O diagrama sequencial visa exibir como é feito o fluxo de procedimentos, métodos e funções do sistema. No caso acima é mostrado como funciona o fluxo de cadastro de uma empresa, nele é possível observar que primeiramente é verificado se o usuário está logado, após tal verificação o sistema envia os dados inseridos pelo usuário para serem salvos no banco de dados caso os dados estejam de acordo com o esperado.

e) Diagrama de Máquina de Estados

Figura 5 – Diagrama de máquina de Estados



Fonte: Acervo pessoal

O diagrama de máquina de estado tem como objetivo mostrar as transações necessárias para que ocorra a mudança nos estados do objetivo. No caso apresentado acima é feita uma verificação se todos os campos do cadastro foram preenchidos, se caso os mesmos não forem preenchidos a transação é cancelada, caso contrario a transação é efetuada e os dados são cadastrados.

f) Diagrama de Atividades

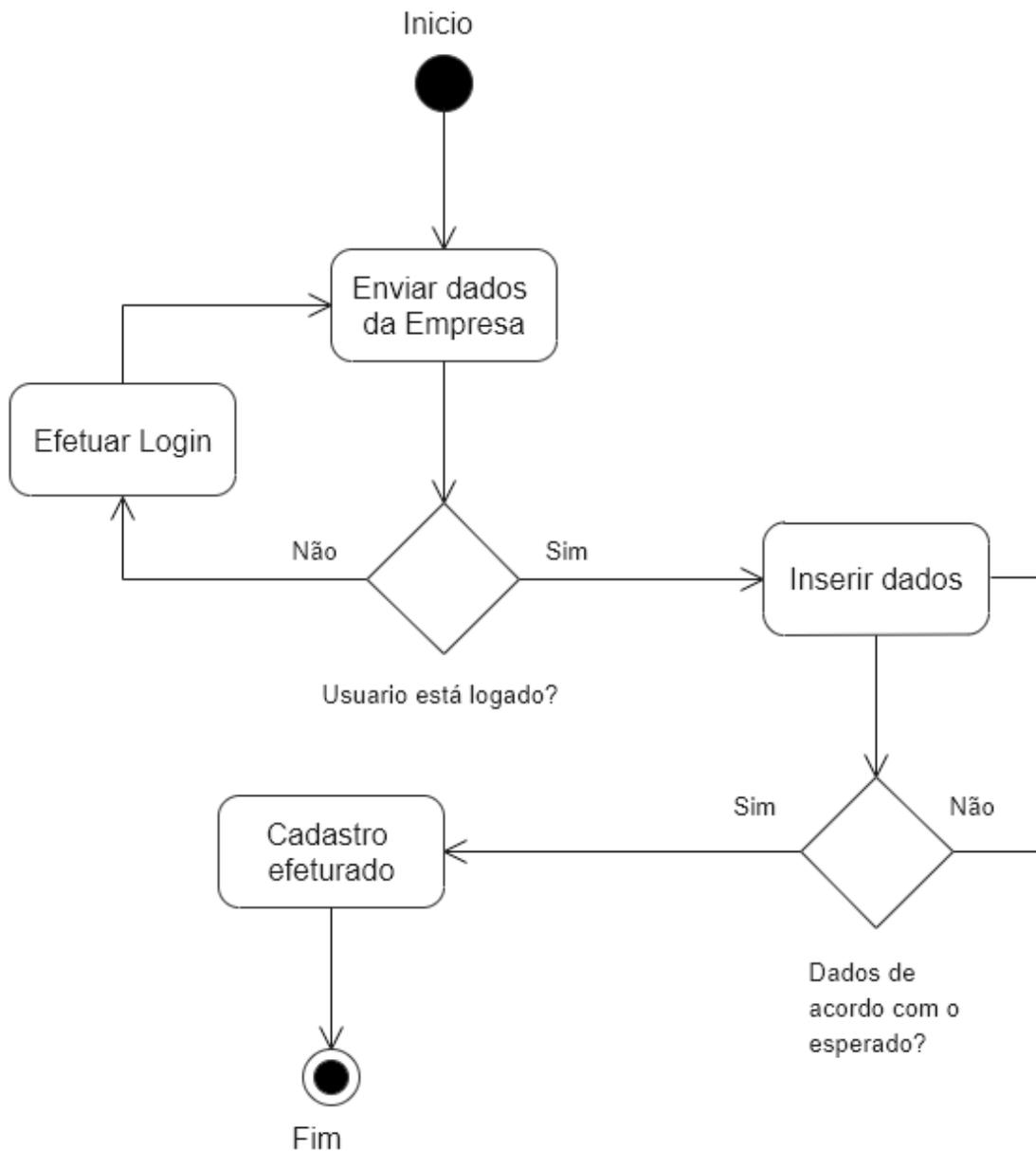


Figura 6 – Diagrama de atividades

Fonte: Acervo pessoal

O diagrama de atividades tende a exibir o fluxo das transações, exibindo deste a parte da validação se o usuário está logado, até a parte de inserção dos dados. O fluxo trabalha de forma contínua, ou seja, ele só dará continuidade se todos os dados ali informados estão de acordo com o esperado.

g) Diagrama de Componentes

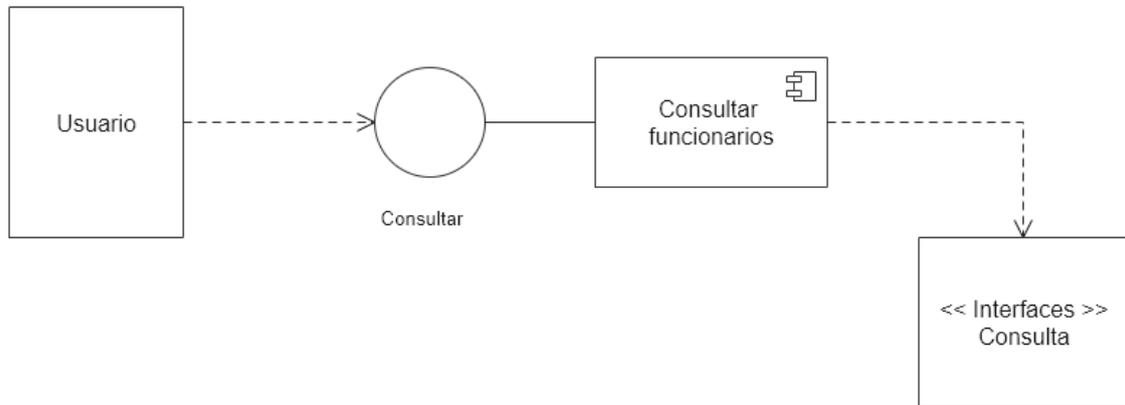


Figura 7 – Diagrama de componentes
Fonte: Acervo pessoal

O diagrama de componentes tem como objetivo exibir de maneira estática a forma com que ocorre a implementação dos dados do sistema, mostrando os também os componentes utilizados.

APÊNDICE B

Neste apêndice encontram-se as principais telas do sistema e suas principais características.

a) Tela de login

Figura 8 – Tela de login



A imagem mostra a tela de login do Sistema de RH. No topo, o título "Sistema de RH" é exibido em uma fonte cinza. Abaixo, há um formulário com os seguintes elementos: um ícone de usuário e o rótulo "Usuário" acima de um campo de entrada; um ícone de cadeado e o rótulo "Senha" acima de um campo de entrada; um botão de opção desativado com o rótulo "Lembrar"; e um botão azul com o texto "Entrar" e um ícone de seta para a direita.

Fonte: Acervo pessoal

Em tal tela, o usuário já cadastrado preenche os dados do campo, para que o mesmo possa ser autenticado e por fim ter acesso ao sistema.

b) Listagem de empresas

Por essa tela o usuário poderá listar todas as empresas já cadastradas no sistema, por ela também é possível efetuar buscas de empresas, CNPJ, sede e telefones específicos através da ferramenta de busca. O usuário também poderá ordenar toda a listagem da forma que desejar. Tal tela também possui uma ferramenta de paginação que tem como objetivo agrupar x (número escolhido pelo usuário) itens por tela, evitando assim o scroll excessivo, além disso o usuário poderá ver os históricos de cada empresa, desativar uma empresa em específico e ir para telas de cadastro e edição das mesmas.

Figura 9 – Listagem de empresas

Nome	CNPJ	Sede	Telefone
ADSA	11.111.111/1342-23	ADDSADSA	(12) 34213-42342
Caio e Augusto Advocacia Ltda	18.456.484/8918-47	Ibatiba	(28) 9962-6447
Eloí e Evelyn Casa Noturna ME	18.054.235/0001-21	São Paulo	(28) 3545-0154
Evelyn e Bruno Padaria ME	18.008.398/0001-78	Ibatiba	(28) 4441-5445
Levi e Manuel Buffet ME	18.409.718/0001-00	Inupi	(29) 3950-4848
Lorenzo e Mariana Locações de Automóveis ME	51.519.559/0001-63	Marapé	(12) 3213-22132
Luciana e Stella Pizzaria ME	43.242.341/2421-42	Ibatiba	(28) 99982-6647
Lúcia e Pedro Henrique Lavanderia ME	53.215.331/2521-51	Santana de Parnaíba	(11) 3560-1386
Nicole Marcenaria Ltda	11.111.111/1111-11	Goiás	(11) 9834-68905
Sarah e Edson Restaurante ME	23.714.490/0001-93	Ibatiba	(38) 9498-489

Fonte: Acervo pessoal

c) Cadastro de empresa

Em tal tela o usuário poderá efetuar o cadastro de uma nova empresa, informando os dados presentes na mesma. Tal formulário também é utilizado para efetuar futuras edições na empresa desejada.

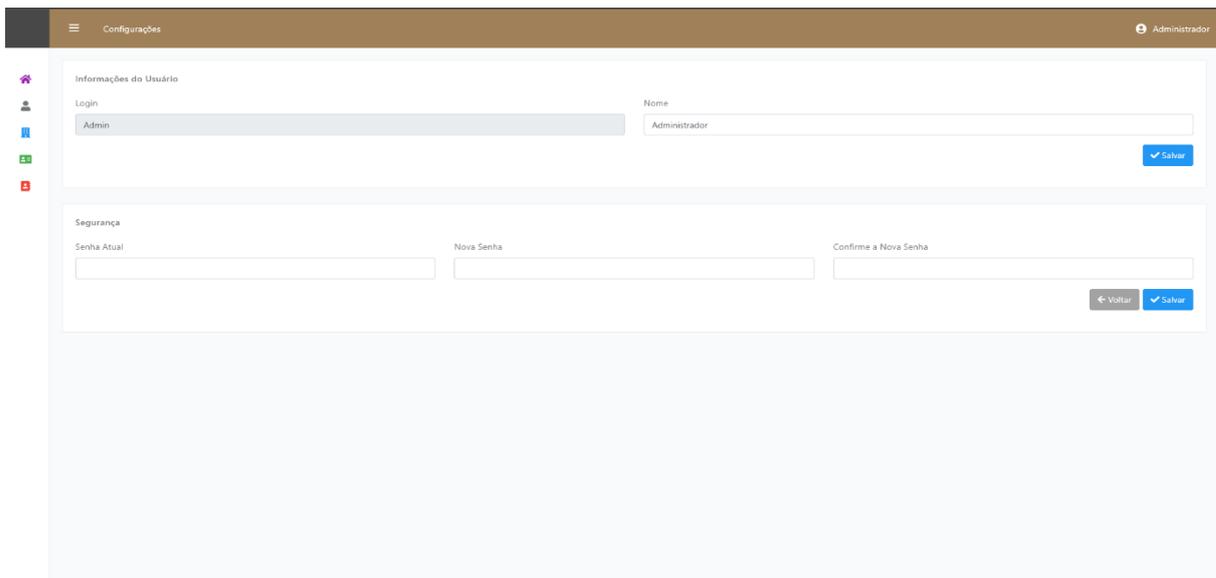
Figura 10 – Cadastro de empresa

Fonte: Acervo pessoal

d) Tela de conflagração de usuário

Por essa tela o usuário poderá editar algumas informações como o nome e a senha, para alterar a senha ele deverá inserir a senha atual e depois informar a nova senha desejada.

Figura 11 – Tela de conflagração de usuário



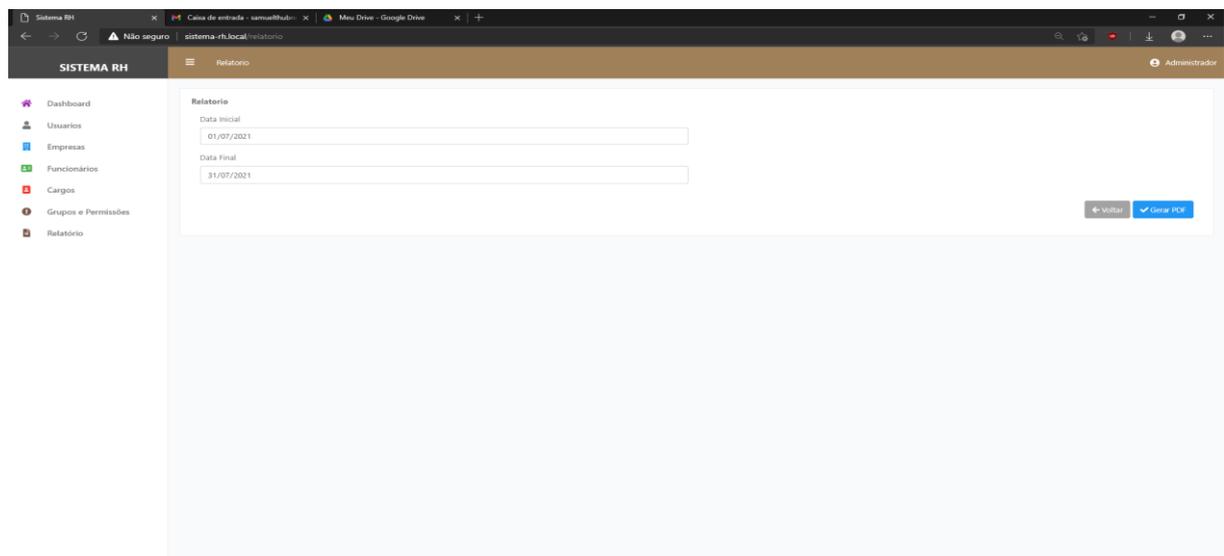
The screenshot shows a web interface for user configuration. At the top, there is a header with a menu icon, the text 'Configurações', and the user role 'Administrador'. On the left, there is a sidebar with icons for 'Informações do Usuário', 'Segurança', and other settings. The main content area is divided into two sections. The first section, 'Informações do Usuário', contains two input fields: 'Login' with the value 'Admin' and 'Nome' with the value 'Administrador'. A blue 'Salvar' button is located to the right of these fields. The second section, 'Segurança', contains three input fields: 'Senha Atual', 'Nova Senha', and 'Confirme a Nova Senha'. Below these fields are two buttons: a grey 'Voltar' button and a blue 'Salvar' button.

Fonte: Acervo pessoal

e) Relatório de funcionários.

O usuário poderá gerar um pdf com informações referentes aos funcionários cadastrados e desligados no período escolhido pelo mesmo.

Figura 12 - Tela para seleção de data para relatório



The screenshot shows a web interface for generating a report. At the top, there is a header with a menu icon, the text 'Relatório', and the user role 'Administrador'. On the left, there is a sidebar with icons for 'Dashboard', 'Usuarios', 'Empresas', 'Funcionários', 'Cargos', 'Grupos e Permissões', and 'Relatório'. The main content area is titled 'Relatório' and contains two input fields: 'Data Inicial' with the value '01/07/2021' and 'Data Final' with the value '31/07/2021'. A blue 'Gerar PDF' button is located to the right of these fields.

Fonte: Acervo pessoal

Após o usuário informar a data, o pdf é gerado com as respectivas informações.

Figura 13 - Relatório em formato PDF

Sistema RH - Relatório de Funcionarios
De 01/07/2021 até 31/07/2021

Funcionários cadastrados no periodo

Nome	Empresa	Cargo	Data Admissão
Fátima Lavinia Pereira	Eloá e Evelyn Casa Noturna ME	Programador	12/07/2021
Laura Luciana	Eloá e Evelyn Casa Noturna ME	Supervisor Operacional	03/07/2021
Oliveira Juliana Campos	Eloá e Evelyn Casa Noturna ME	Atendente SAC	13/07/2021

Funcionários desligados no periodo

Nome	Empresa	Cargo	Data Admissão	Data Desligamento
Julia Orlando	Eloá e Evelyn Casa Noturna ME	Supervisor Operacional	07/04/2020	12/07/2021
Vitor César Santos	Tereza e Betina Transportes Ltda	Supervisor Operacional	29/06/2021	12/07/2021

Fonte: Acervo pessoal