

UNIVERSIDADE FEDERAL DO PARANÁ

PEDRO MORMUL CORREIA

**CHARTORG: REDE SOCIAL PARA EMPRESAS**

Curitiba

2021

PEDRO MORMUL CORREIA

**CHARTORG: REDE SOCIAL PARA EMPRESAS**

Trabalho apresentado como requisito parcial à obtenção de grau de Tecnólogo em Análise e Desenvolvimento de Sistemas do Setor de Educação Profissional e Tecnológica da Universidade Federal do Paraná.

Orientadora: Professora Mestre Andreia de Jesus

Curitiba

2021

## **AGRADECIMENTOS**

Agradeço aos meus professores do Setor de Educação Profissional e Tecnológica da UFPR que me ajudaram a obter todo o conhecimento que hoje tenho para concluir esse projeto e ao longo de minha carreira profissional.

Também agradeço a meus familiares que me incentivaram a concluir meus estudos e me deram oportunidade para realizar tal objetivo.

## SUMÁRIO

<b>1 INTRODUÇÃO.....</b>	<b>8</b>
1.1 PROBLEMA.....	8
1.2 OBJETIVOS GERAL.....	9
1.3 OBJETIVOS ESPECÍFICOS.....	9
1.4 JUSTIFICATIVA.....	10
1.5 ORGANIZAÇÃO DO DOCUMENTO.....	11
<b>2 FUNDAMENTAÇÃO TEÓRICA.....</b>	<b>12</b>
2.1 ASPECTOS CONCEITUAIS DO NEGÓCIO.....	12
2.2 REDES SOCIAIS.....	12
2.3 GERÊNCIA DE PROJETOS.....	14
2.4 LINGUAGENS DE PROGRAMAÇÃO.....	17
2.5 FRAMEWORKS.....	25
2.6 SISTEMA GERENCIADOR DE BANCO DE DADOS.....	27
2.7 TRABALHOS CORRELATOS.....	28
<b>3 METODOLOGIA DO TRABALHO.....</b>	<b>31</b>
3.1 REQUISITOS DO SISTEMA.....	31
3.1.1 Levantamento de requisitos.....	31
3.1.2 Especificação de requisitos.....	32
<b>3.2 GERÊNCIA DO PROJETO.....</b>	<b>33</b>
3.3 ANÁLISE DO PROJETO.....	36
3.3.1 Análise dos Dados.....	36
3.3.2 Análise das Funcionalidades.....	36
3.4 TECNOLOGIAS E LINGUAGENS APLICADAS.....	37
<b>4 APRESENTAÇÃO DO SOFTWARE CHARTORG.....</b>	<b>40</b>
4.1 ARQUITETURA DO SOFTWARE.....	40
4.2 O SISTEMA CHARTORG.....	41
<b>5 CONSIDERAÇÕES FINAIS.....</b>	<b>47</b>
5.1 RECOMENDAÇÕES PARA TRABALHOS FUTUROS.....	47

<b>REFERÊNCIAS.....</b>	<b>49</b>
<b>APÊNDICE A - TRANSCRIÇÃO DA ENTREVISTA.....</b>	<b>53</b>
<b>APÊNDICE B - TRANSCRIÇÃO DO RESUMO DAS SPRINTS.....</b>	<b>54</b>
<b>APÊNDICE C - DIAGRAMA DE CASO DE USO.....</b>	<b>56</b>
<b>APÊNDICE D – ESPECIFICAÇÃO DE CASOS DE USO.....</b>	<b>57</b>
<b>APÊNDICE E - DIAGRAMA DE ENTIDADE RELACIONAMENTO.....</b>	<b>87</b>
<b>APÊNDICE F - DIAGRAMA DE CLASSE DE NEGÓCIO.....</b>	<b>88</b>
<b>APÊNDICE G - DIAGRAMAS DE SEQUÊNCIA.....</b>	<b>89</b>

## RESUMO

Atualmente as empresas e seus funcionários passam por uma fase de transformação digital. Cada vez mais a tecnologia está presente diariamente nos ambientes de trabalho, por meio de softwares que auxiliam em contratações; softwares que auxiliam na organização do trabalho; plataformas que facilitam a comunicação entre os funcionários das empresas; dentre outras ferramentas. Com o foco em auxiliar as empresas a passarem por essa transformação digital, melhorar a qualidade da comunicação e interação dos seus funcionários , propõe-se com este projeto o desenvolvimento de uma plataforma denominada **CHARTORG: Rede Social Para Empresas**. A proposta é disponibilizar para as empresas uma ferramenta que possibilita a comunicação e interação dos funcionários via um sistema web para compartilhamento e comunicação de ideias em tempo real. As ferramentas aplicadas no desenvolvimento da plataforma web foram: linguagem *Ruby*, HTML, CSS, SQL, *frameworks Ruby on Rails*, banco de dados *Postgres*. Como resultado obteve-se um software com todos os requisitos para interação de seus usuários, com todo o código escrito com tecnologia aberta de forma que novos desenvolvedores e/ou técnicos possam utilizar e empregar em suas empresas com facilidade.

**Palavras-chave:** Comunicação; Interação de Funcionários; Rede Social; Sistema Web Tecnologia.

## ABSTRACT

Currently, companies and their employees are going through a phase of digital transformation. More and more technology is present daily in work environments, through software that helps in hiring; software that helps in the organization of work; platforms that facilitate communication between company employees; among other tools. With the focus on helping companies to go through this digital transformation, improving the quality of communication and interaction of their employees, it is proposed with this project the development of a platform called **CHARTORG: Social Network for Companies**. The proposal is to provide companies with a tool that enables the communication and interaction of employees via a web system for sharing and communicating ideas in real time. The tools applied in the development of the web platform were: Ruby language, HTML, CSS, SQL, Ruby on Rails frameworks, Postgres database. As a result, a software was obtained with all the requirements for the interaction of its users, with all the code written with open technology so that new developers and / or technicians can use and employ in their companies with ease.

**Keywords:** Communication; Employee Interaction; Social Network; Web Technology System.

## 1 INTRODUÇÃO

Atualmente as empresas e seus funcionários passam por uma fase de transformação digital. Cada vez mais a tecnologia está presente em ambientes de trabalho, por meio de softwares que auxiliam em contratações; softwares que auxiliam na organização do trabalho; plataformas que facilitam a comunicação entre os funcionários das empresas; dentre outras ferramentas.

Esta transformação digital tornou-se ainda mais evidente com a pandemia provocada pela Covid-19 em 2020, que obrigou o isolamento das pessoas e promoveu o trabalho remoto em massa. Com isso, as empresas buscaram formas de aproximar seus funcionários por meio da tecnologia. Porém, ferramentas de comunicação não são simples de se dar manutenção, pois muitas vezes exigem conhecimento avançado para suas configurações e/ou seus preços acabam afastando empresas de pequeno e médio porte. Além disso, tais ferramentas não necessariamente irão proporcionar a mesma interação que ocorre de forma presencial no dia a dia das empresas.

Segundo Adeline Daniele (2015), empresas que não oferecerem trabalho remoto terão dificuldades de manter talentos, visto que as tecnologias da informação e comunicação tornam a vida do profissional menos burocrática, facilitando e agilizando o processo de interação diária, tornando mais cômodas e práticas as atividades empresariais.

Com o foco em auxiliar as empresas a passarem por essa transformação digital e possibilitar ainda mais que a interação dos seus funcionários seja fácil e de maneira simples, propõe-se com este projeto o desenvolvimento de uma plataforma denominada **CHARTORG: Rede Social Para Empresas**. A proposta é disponibilizar para empresas e funcionários uma ferramenta que possibilita interação via um sistema web para compartilhamento e comunicação de ideias em tempo real.

### 1.1 PROBLEMA

De acordo com uma pesquisa realizada pelo *Project Management Institute Brasil* (PMI), instituição internacional que faz associação de profissionais de gestão de projetos, 76% de 300 grandes empresas observam que a falta de clareza na comunicação dentro de seu ambiente de trabalho é

um dos principais motivos para que não se possa obter o sucesso em projetos ou objetivos da empresa.

Segundo Daniella Doyle (2019), 35% das empresas não têm um setor responsável pela comunicação e 75% têm dificuldade em localizar as informações na empresa. Ainda segundo o site, essa questão de falta de informação desenvolve consequências graves como retrabalhos e prejuízos financeiros.

Com isso, empresas como *Microsoft* (2021?) estão apostando cada vez mais em softwares que unem seus funcionários, como o *Microsoft Teams*, mostrando a importância que tais softwares têm para a melhoria da comunicação. Exemplos de benefícios que o uso da tecnologia traz para a comunicação é o aumento de produtividade dos funcionários, melhorando a integração entre os pares e, com isso, diminuindo erros por meio de uma comunicação assertiva. Assim, os funcionários podem ser mais produtivos sem perderem tempo e recursos com retrabalhos.

## 1.2 OBJETIVOS GERAL

O objetivo deste trabalho é disponibilizar uma plataforma online para aprimorar e auxiliar o processo de comunicação interna de funcionários dentro de uma empresa. Para tanto, buscar-se-á desenvolver uma rede social para atender as necessidades de ambientes de trabalho.

## 1.3 OBJETIVOS ESPECÍFICOS

- Entrevistar um profissional do RH de uma empresa;
- Identificar as características de um sistema do tipo rede social;
- Levantar requisitos por meio da análise de softwares similares, disponíveis no mercado;
- Modelar um banco dados com informações genéricas a qualquer empresa;
- Implementar o organograma da empresa, para que o funcionário consiga identificar o responsável por um determinado departamento ou área;
- Modelar e implementar uma rede social empresarial.

## 1.4 JUSTIFICATIVA

Atualmente a comunicação é um componente inerente a todos os processos administrativos de uma empresa, assim como a autora Profa. Dra. Thais Zaninelli (2013), que em seu artigo cita fatores que influenciam o processo de inovação no contexto colaborativo, citando a sua importância, contendo diversos motivos, como a necessidade de divulgar uma certa notícia ou comunicado e, isso, deve ser feito de forma assertividade, a fim de alcançar benefícios na produtividade do trabalho.

De acordo com o Instituto *McKinsey* (2021), realizado com 1674 empresas, 82% fizeram o uso de uma ferramenta de comunicação em 2013, com isso mais da metade, 62%, tiveram um aumento da produtividade de seus funcionários. Sendo assim, empresas têm cada vez mais buscado formas para serem transparentes com seus colaboradores e seu público alvo, procurando diversas estratégias para que seja possível diminuir a falta de comunicação e/ou melhorar a qualidade da comunicação na empresa.

No caso das empresas de pequeno e médio porte, os softwares livres para comunicação são essenciais, considerando que o custo médio de aquisição destas ferramentas são cada vez mais um diferencial no momento da escolha. Um exemplo de custo é o *Slack*<sup>1</sup>, software baseado na troca de mensagens utilizado para comunicação entre pessoas de uma comunidade, que possui um preço de 8 dólares por pessoa, dados de 2020. Com isso, tais ferramentas de código livre irão trazer não só benefícios para o processo de comunicação, mas também para a parte financeira da empresa.

Considerando o contexto apresentado, a proposta da plataforma **CHARTORG** visa disponibilizar uma opção a mais para as empresas no que diz respeito a recursos tecnológicos acessíveis para a promoção da comunicação interna de seus funcionários. Com isso, promover a melhoria na vida profissional dos funcionários, através de interações em uma rede social voltada para o ambiente de trabalho. Outro benefício que a plataforma dispõe é a possibilidade das empresas acompanharem as necessidades de seus

---

<sup>1</sup>Disponível em: < <https://app.slack.com/plans/T024FT6BZ?geocode=pt-br>>. Acesso em: novembro de 2020.

funcionários, pois se trata de um local onde se pode compartilhar experiências, processos de trabalho e até celebrações relacionadas ao ambiente profissional.

## 1.5 ORGANIZAÇÃO DO DOCUMENTO

Para uma melhor apresentação e entendimento deste Trabalho de Conclusão de Curso, este documento está organizado segundo os seguintes capítulos:

- **Fundamentação Teórica:** Compreende a apresentação da teoria aplicada no trabalho, considerando os aspectos conceituais do negócio, dinâmica de Redes Sociais, conceitos de Gestão de Projetos, linguagem de programação, *framework*, sistema gerenciados de banco de dados, trabalhos correlatos;
- **Metodologia do Trabalho:** Demonstra aplicação dos fundamentos teóricos que foram utilizados durante o desenvolvimento deste trabalho;
- **Apresentação do Software CHARTORG:** Explica como foi criado e implementado a arquitetura e procurando abordar suas funcionalidades;
- **Considerações Finais:** Contém os aprendizados adquiridos e recomendações para trabalhos futuros.

## 2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo apresenta as regras de negócio do problema a ser resolvido; o conceito de gerência de sistemas aplicado; as metodologias e tecnologias utilizadas no desenvolvimento da plataforma; e uma análise de trabalhos correlatos.

### 2.1 ASPECTOS CONCEITUAIS DO NEGÓCIO

Desde que foi concretizado o desejo de desenvolver uma plataforma com o conceito de comunicação e interação através de uma rede social voltado ao contexto empresarial, com o objetivo de auxiliar o dia a dia do funcionário por meio da simplificação da interação entre colaboradores, houve a necessidade de se embasar conceitualmente em assuntos relacionados com o tema. Matrizes diversas foram estudadas, desde a área de recursos humanos de empresas a redes sociais empregadas para o mercado de trabalho. O objetivo deste trabalho é demonstrar o conceito de comunicação e redes sociais aplicado ao negócio.

Cabe conceituar que no âmbito deste trabalho o sentido da comunicação está intimamente relacionado à interação dos colaboradores por meio de uma rede social em um ambiente empresarial.

O software desenvolvido neste trabalho, foi estruturado através de pesquisas e estudos realizados durante o seu processo de desenvolvimento, explorando conceitos mais relevantes para sua implementação.

### 2.2 REDES SOCIAIS

Uma rede social pode ser definida como uma estrutura social formada por indivíduos com um ou mais objetivos em comum, sendo esses objetivos o elo de ligação para que ocorra a interação entre os membros da rede, além da simplicidade de comunicação entre quaisquer membros de sua rede social. Mencionado pelos autores, Sônia Cristina Vermelho, Ana Paula Machado Velho, Valdecir Bertoncello, em seu artigo Conceito de Redes Sociais e seus Pesquisadores, além disso, Marteleto ainda completa:

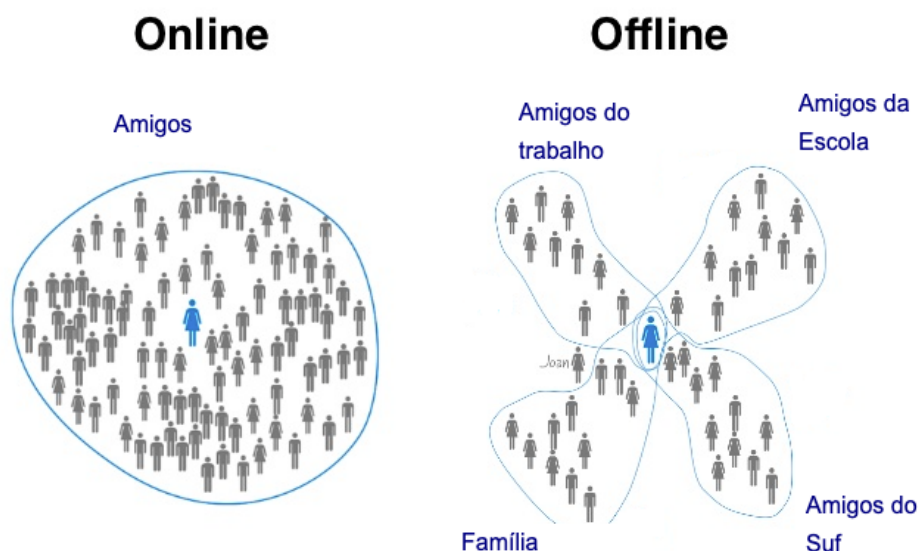
A rede social, derivando deste conceito, passa a representar um conjunto de participantes autônomos, unindo ideias e recursos em torno de valores e interesses compartilhados. (MARTELETO, 2001 p.72).

Com a modernização da tecnologia, pode-se observar que cada vez mais são desenvolvidos softwares com o intuito de integrar pessoas por meio da tecnologia, e-mails, mensagens instantâneas, postagens de notícias, etc.

Estruturalmente, uma rede social é constituída por: atores, pessoas que são responsáveis pela comunicação feita na rede social; estrutura e local onde se baseia a rede social.

A rede social pode ser *online* ou *offline*. As redes *online* compreendem sites utilizados para interações de pessoas, por exemplo: *Facebook*, mídias e redes sociais virtuais. As redes *offline* são ambientes físicos onde presencialmente as pessoas interagem entre si. A FIGURA 1 demonstra os dois tipos de redes sociais descritos.

FIGURA 1– Demonstração de uma rede social



FONTE: Bridging the gap between our online and offline social network<sup>2</sup>, adaptado pelo Autor.

<sup>2</sup> Disponível em: <[https://pt.slideshare.net/padday/bridging-the-gap-between-our-online-and-offline-social-network/6-Her\\_family](https://pt.slideshare.net/padday/bridging-the-gap-between-our-online-and-offline-social-network/6-Her_family)>. Acesso em: novembro de 2020.

Ainda com base no conceito apresentado, a rede social possui diferentes categorias de motivações que são utilizadas entre elas para que ocorra a manifestação e interação de seus membros. Assim como observado no texto Rede Social: característica, estrutura e importância social, pela MTI(2016), empresa pública de tecnologia da informação, podemos dividi-los abaixo:

- Redes comunitárias: podem ser divididas em online e offline. Essas são redes que são ligadas por um interesse em comum de seus participantes; como melhorar a situação do local de seus participantes, podendo ser uma associação de moradores de um condomínio ou grupo de Facebook;
- Redes profissionais: mais conhecida como networking, é a rede com intuito de fazer interações para promover o meio profissional. Um exemplo é a rede social online é o LinkedIn.

Com isso, foi definido que as características de redes sociais profissionais são mais relevantes e atendem ao negócio explorado neste trabalho, por se tratar de um meio de comunicação onde o objetivo é relacionado ao trabalho.

### 2.3 GERÊNCIA DE PROJETOS

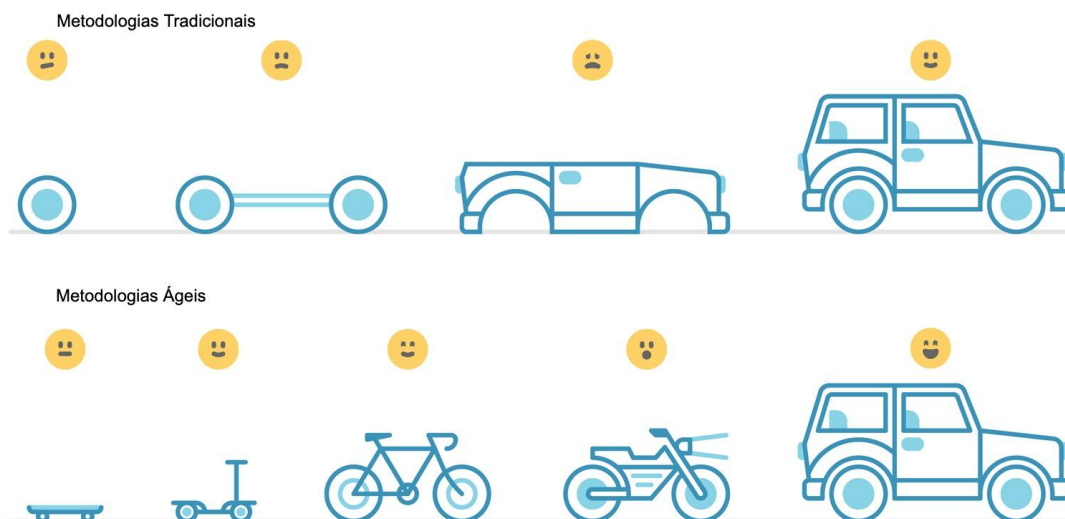
O gerenciamento de projetos é o uso de conhecimentos, habilidades e técnicas aplicadas na criação, manutenção e gerenciamento de um projeto, com o intuito de entregar um software capaz de atender todos os requisitos técnicos e funcionais da modelagem proposta (PMBOK, 2017). Desta forma, no início de um projeto é necessário seguir algumas práticas que irão auxiliar nas mais diversas etapas do projeto, lidando com adversidades, tempo, custo, qualidade e, por fim, sua entrega e metrificação. Parte-se da premissa para o gerenciamento de projetos ter clara a definição do objetivo e escopo aprovado pelo cliente. Mudanças podem ocorrer para que seja possível a entrega do projeto em determinado escopo, prazo e custo, sendo necessário realizar pesquisas de mercado para verificar se sua hipótese ou projeto é ainda válido.

As metodologias de gerenciamento de projetos contêm categorias e com elas características específicas que são seguidas ao longo do projeto, divididas em (PMBOK, 2017):

- Metodologias Tradicionais: como, por exemplo, PRINCE 2 e PMBOK. As suas vantagens são: a previsibilidade financeira com custos mais fixos; previsibilidade de entrega com calendários mais assertivos; documentação extensa;
- Metodologias Ágeis: como, por exemplo, *Scrum*, XP, *Lean*. As suas vantagens são: a flexibilidade do projeto e entregas, sabendo que às vezes o que foi definido no início de seu projeto pode não fazer mais sentido; processos mais simples; entregas gradativas, onde o cliente pode ter a visão do valor do produto desde o começo.

É possível observar na FIGURA 2 que um projeto baseado em metodologias tradicionais o valor é entregue apenas no final, onde o cliente vai receber seu objeto completo, em metodologias ágeis o cliente irá receber valores gradualmente.

FIGURA 2 – Metodologia tradicional versus ágeis



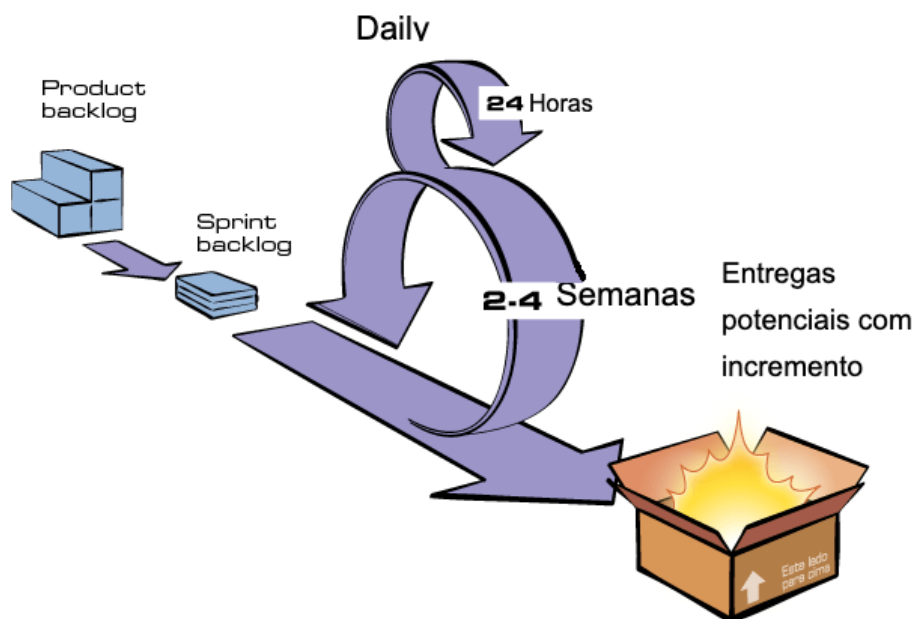
FONTE: Pagina da IEEP educação<sup>2</sup>

### 2.3.1 Metodologia *Scrum*

*Scrum* é uma metodologia de gestão de projetos ágeis voltados para o planejamento de software, criado em 1995, por Jeff Sutherland e Ken Schwaber (SCRUM, 1993). Eles se basearam no termo utilizado no esporte *Rugby*, onde jogadores se unem para em conjunto chegarem ao objetivo em comum. Da mesma forma, neste projeto, em que autor e orientadora se uniram para alcançar um objetivo em comum, flexibilizar um projeto dividindo-o em entregas que gradualmente são evoluídas através de um *Backlog*, uma lista de atividades que o time se compromete a entregar; dividido em ciclos chamados de *Sprints*, ou seja, um determinado período de tempo para realizar as atividades, representado esse tempo por uma *Time Box* que é indicado o tempo para a realização de atividades propostas. O *Scrum* é um composto de (SCRUM ORG, 2021?):

- *Product Backlog*: etapa em que é levantado ideias e funcionalidades para alimentar o projeto e seus objetivos;
- *Sprint Planning Meeting*: reuniões que ocorrem no momento inicial das *sprints*, com o objetivo de priorizar atividades do *Product Backlog* para a próxima interação;
- *Daily Scrum*: reuniões diárias que são utilizadas para o acompanhamento do andamento das atividades propostas para uma *sprint*, FIGURA 3.

FIGURA 3 – Daily scrum meeting



FONTE: Adaptado de: Desenvolvimento Ágil<sup>3</sup>

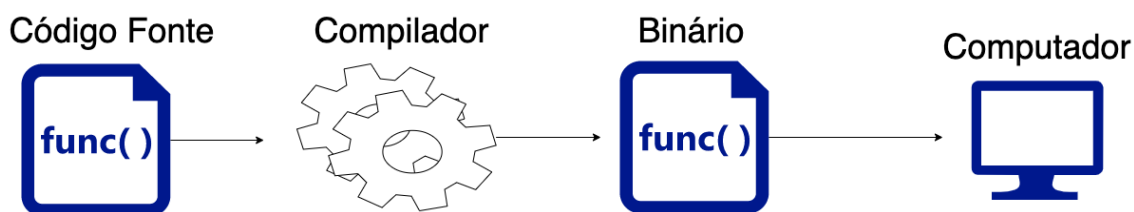
Na FIGURA 3, podemos observar como funciona o ciclo da metodologia *Scrum*, através de suas cerimônias, desde sua ideia até a entrega efetiva, passando por todas as fases previamente explicadas no capítulo 2.3.

## 2.4 LINGUAGENS DE PROGRAMAÇÃO

Linguagens de programação são uma série de conjuntos, possuindo regras de sintaxe e semântica, através de um código fonte, criadas para que se permita a descrição detalhada de uma série de instruções onde um determinado computador deve reproduzir. Escritas por meio de algoritmos pelo programador e interpretada pelo computador, dessa forma devendo ser convertido em binário, conforme demonstrado na FIGURA 4.

<sup>3</sup> Disponível em: <<https://www.desenvolvimentoagil.com.br/scrum/>> Acesso em: novembro de 2020.

FIGURA 4 – Compilador



FONTE: Elaborado pelo autor

Existem diversas linguagens, sendo cada uma delas podendo ser empregadas em diferentes âmbitos, como o SQL, linguagem utilizada para comunicação com Sistemas Gerenciadores de Bancos de Dados; HTML, CSS e *JavaScript*, muito utilizadas por programadores para criar sites; linguagem R utilizada por cientistas de dados para manipulação de dados.

Nos tópicos seguintes, apresentam-se as linguagens de programação utilizadas para a elaboração deste projeto, sendo elas: *Ruby*, HTML, *JavaScript*, CSS e SQL.

#### a) Linguagem *Ruby*

Como a própria documentação da linguagem propõe, trata-se de uma “linguagem dinâmica, *open source* com foco na simplicidade e na produtividade. Tem uma sintaxe elegante de leitura natural e fácil escrita.” (DOCUMENTING RUBY, 2021?).

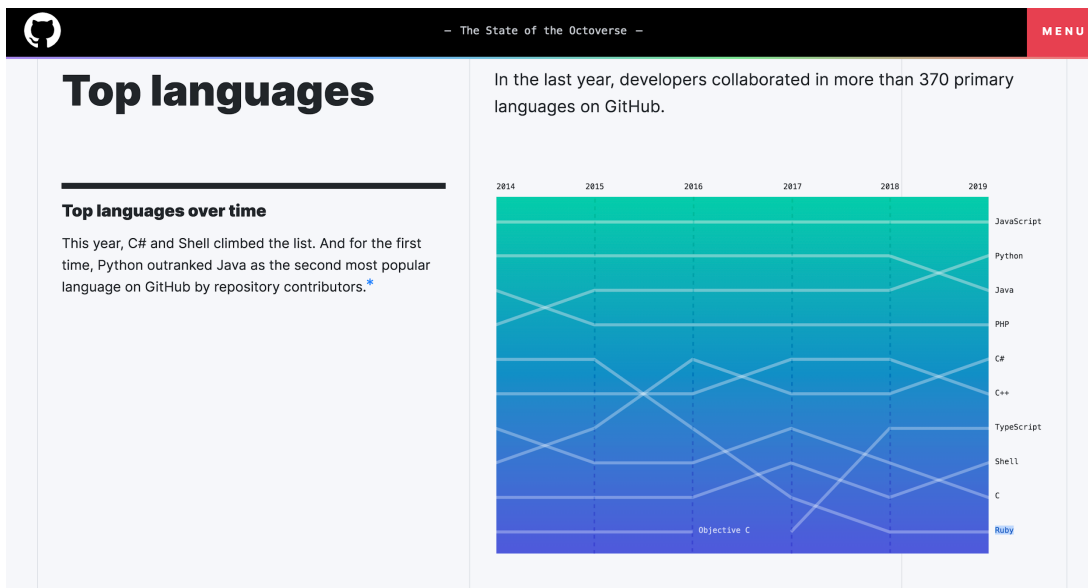
Essa linguagem foi criada em 1995 por Yukihiro "Matz" Matsumoto. É dinamicamente tipada e com gerenciamento de linguagem, (DOCUMENTING RUBY, 2021?). Foi desenvolvida no Japão para ser utilizada como uma linguagem de *script*. Sabendo disso, o criador teve como proposta para sua linguagem suporte para programação funcional, orientada a objetos, imperativa e reflexiva.

É possível verificar alguns benefícios que foram considerados na escolha da linguagem para este projeto, tais como:

- Facilidade de interpretação de sintaxe do código;
- Versatilidade para escrever um código, sendo coeso e com uma grande comunidade da linguagem;

- Grande gama de bibliotecas que são utilizadas durante todo desenvolvimento do código;
- Uma das 10 linguagens mais utilizadas nos últimos 5 anos (GITHUB, 2020), FIGURA 5.

FIGURA 5 – Top linguagens



FONTE: Github<sup>4</sup>

## b) Linguagem HTML

O HTML pode ser definido com base em seu próprio nome, Linguagem de Marcação de HiperTexto, ou seja, essa linguagem é utilizada para construir sites através de marcações de textos que interpretados por ela o navegador irá construir a estrutura de um site, anotando textos, imagens e alguns conteúdos (MDN WEB DOCS, 2019). Algumas dessas marcações são mostradas na FIGURA 6. Cada uma dessas marcações é conhecida como "Tags", responsável por reproduzir diferentes estruturas em um site.

<sup>4</sup> Disponível em: <<https://octoverse.github.com/>> Acesso em: novembro de 2020.

FIGURA 6 – Tags HTML

```
<head>, <title>, <body>, <header>, <footer>, <article>,  
<section>, <p>, <div>, <span>, <img>, <aside>, <audio>,  
<canvas>, <datalist>, <details>, <embed>, <nav>, <output>,  
<progress>, <video>, <ul>, <ol>, <li>
```

FONTE: Elaborado pelo autor

### c) Linguagem *JavaScript*

Conhecida também por JS, o *Javascript* é uma linguagem interpretada e baseada em objetos, criada pela Netscape em 1995 (MDN WEB DOCS, 2020), o intuito era que as páginas web pudessem ganhar vida. Seu navegador lançado em conjunto, o *Netscape Navigator 2.0*, adicionou dinamicidade, podendo ser utilizada para controlar elementos que previamente eram apenas estáticos, como definido acima, marcados por *tags* HTML. Após o seu lançamento, houve uma padronização da sua sintaxe para que fosse possível rodar em qualquer navegador.

O *Javascript* pode ser adicionado *inline*, ou seja, dentro de arquivos HTML, ou importados através de *scripts* HTML, onde se faz referência do arquivo JS que a página deve carregar. A FIGURA 7 apresenta um exemplo de importação de *javascript* em uma página HTML e a FIGURA 8 apresenta um exemplo de *javascript inline* em uma página HTML.

FIGURA 7 – Importação de *javascript* em uma página HTML

```

1  <!DOCTYPE html>
2  <html lang="en">
3    <head> </head>
4    <body>
5      <h1>JavaScript Loading Strategies - Approach 2</h1>
6      
7      <div class="container">
8        <p>Dynamic from JS files</p>
9        <p><span>File 1 :</span> <span id="output1">Empty</span></p>
10       <p><span>File 2 :</span> <span id="output2">Empty</span></p>
11     </div>
12
13   <p style="display: none" class="dummy-content">...
12844 </p>
12845
12846   <script src="./js/file1.js"></script>
12847   <script src="./js/file2.js"></script>
12848   <script src="./js/file3.js"></script>
12849 </body>
12850 </html>

```

FONTE: Elaborado pelo autor

FIGURA 8 – Javascript inline em uma página HTML

```

1  <script type="text/javascript">
2    //Definido o Javascript
3    function func() {
4      alert("Hello I'm inside function");
5    }
6    $('#function').click(func);
7
8    $('#anonymous-function').click(function() {
9      alert("Hello I'm inside anonymous function");
10   });
11
12   var inline_func = function() {
13     alert("Hello I'm inside inline function");
14   };
15   $('#inline-function').click(inline_func);
16 </script>

```

FONTE: Elaborado pelo autor

## d) Linguagem CSS

*Cascading Style Sheets* ou em sua tradução Folhas de Estilo em Cascata, o CSS é uma linguagem de estilo utilizada em sites para que seja

possível definir a apresentação de como o navegador irá interpretar a página, definindo como serão exibidos os elementos do HTML, as *tags* (MDN WEB DOCS, 2019).

Criada inicialmente para diferenciar o que é o conteúdo do site do que é estilo, auxiliando os programadores em melhorias de layout, cores e fontes de seus sites. Assim como o *Javascript*, é possível definir o CSS de três maneiras distintas:

- Dentro de uma tag HTML, conhecida como inline, utilizando o atributo do HTML "*<style>*". Porém, não é recomendado, pois, cada uma de suas *tags* precisam ser definidas gerando um enorme trabalho. Segue um exemplo na FIGURA 9.

FIGURA 9 – CSS dentro de uma *tag* HTML

```
<!DOCTYPE html>
<html>
<body style="background-color:black;">

<h1 style="color:white;padding:30px;">Hostinger Tutorials</h1>
<p style="color:white;">Something useful here.</p>

</body>
</html>
```

FONTE: Elaborado pelo autor

- Na página HTML, conhecido por CSS interno, adicionado em uma seção "*<HEAD>*" do HTML. Elas são usadas para que seja possível a repetição de suas classes. Porém, a desvantagem é que as classes são pertencentes apenas onde é definido esse estilo. Segue um exemplo na FIGURA 10.

FIGURA 10 – CSS dentro da página HTML

```
<head>
<style type="text/css">
p {color:white; font-size: 10px;}
.center {display: block; margin: 0 auto;}
#button-go, #button-back {border: solid 1px black;}
</style>
</head>
```

FONTE: Elaborado pelo autor

- Importado dentro de uma página por meio de um arquivo próprio .css, desta forma quaisquer alterações que ocorram em seu arquivo de estilo é refletido em todas as páginas HTML que são importadas. Segue um exemplo na FIGURA 11 e na FIGURA 12, sendo a primeira importação e a segunda definição de um arquivo .css.

FIGURA 11 – CSS próprio, importação

```
<head>
<link rel="stylesheet" type="text/css" href="style.css" />
</head>
```

FONTE: Elaborado pelo autor

FIGURA 12 – CSS próprio, definição de arquivo

```
.xleftcol {  
float: left;  
width: 33%;  
background:#809900;  
}  
.xmiddlecol {  
float: left;  
width: 34%;  
background:#eff2df;  
}
```

FONTE: Elaborado pelo autor

## e) Linguagem SQL

SQL é uma linguagem utilizada para se comunicar com banco de dados relacionais (SILVEIRA, 2019). Desenvolvida nos anos 70 nos laboratórios da IBM, seu propósito e uso é de gerar *scripts* conhecidos como *queries*, que geram consultas ou interações dentro de um banco. Entre os bancos que a utilizam podemos citar: *Oracle*, *MySQL*, *MariaDB*, *PostgreSQL*, *Microsoft SQL Server*, entre muitos outros.

A SQL é dividido em duas grandes categorias: (1) DML (*Data Manipulation Language*) usado para manipular os dados, fazer consultas e manutenção de seus dados; (2) DDL (*Data Definition Language*) usado para definir os dados de um banco e objetos. Como exemplo podemos citar os mais conhecidos comandos: SELECT, UPDATE, DELETE e INSERT, FIGURA 13.

FIGURA 13 – Exemplo de comandos SQL

```
INSERT INTO funcionarios (nome)
VALUES ('Pedro')
DELETE FROM funcionarios
WHERE nome = 'Pedro'
SELECT * FROM funcionarios
UPDATE funcionarios
SET name = 'Paulo'
```

FONTE: Elaborado pelo autor

## 2.5 FRAMEWORKS

O *framework* é um conjunto de códigos genéricos, sendo utilizado para compartilhar códigos em pacotes, que auxiliam no desenvolvimento de projetos de softwares (NOLETO, 2020). Com eles simplifica-se a maneira que se escreve determinados trechos de códigos e suas estruturas, melhorando na qualidade do projeto e ainda mais na produtividade do desenvolvedor. A seguir serão apresentados os dois *frameworks* utilizados no desenvolvimento do projeto CHARTORG.

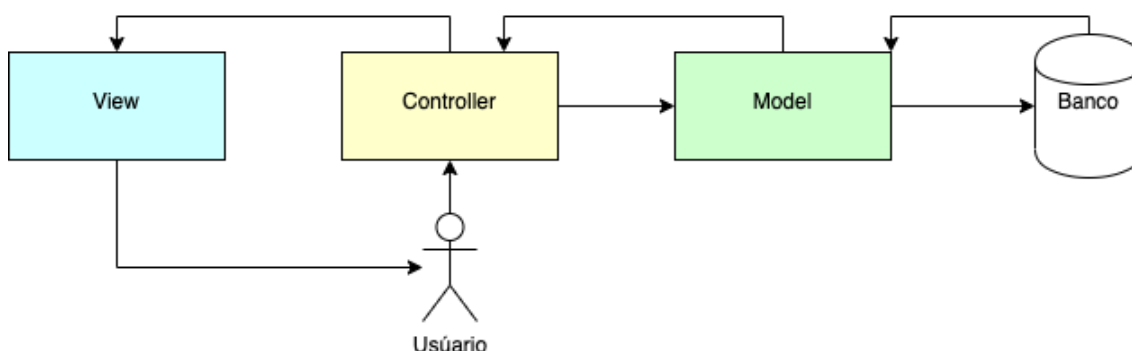
### a) *Framework Ruby on Rails*

Complementando a linguagem *Ruby* temos o *framework Ruby on Rails*, uma ferramenta escrita pela comunidade *Ruby* para desenvolvimento web. Esta ferramenta nos permite aumentar a velocidade e facilitar o desenvolvimento de sites orientados a banco de dados. Isso quer dizer que as aplicações são desenvolvidas visando os dados do produto por meio de códigos que irão auxiliar o desenvolvedor com estruturas definidas. É importante colocar que ela foi baseada no design *pattern* MVC, FIGURA 14, muito utilizado quando se trata de projetos com a interface web, auxiliando em padrões de estrutura do código. Com esse padrão podemos obter vantagens em sua utilização, como melhorar a manutenção na aplicação, a arquitetura modular e suas partes podem ser modificadas sem alterar todo contexto.

O MVC é composto pelas seguintes camadas (GUEDES, 2020):

- *Controller*: camada responsável pela interpretação da requisição e resposta da requisição;
- *Model*: camada responsável por representar o negócio, também utilizada para manipular os dados da aplicação;
- *View*: camada responsável pela interface que o usuário interage, mostrando informações pertinentes ao cliente.

FIGURA 14 – Exemplo de padrão MVC



FONTE: Elaborado pelo autor

Além disso, com o uso do *Ruby on Rails* podemos nos beneficiar do *ActiveRecord*, uma camada de mapeamento objeto-relacional que é um *ORM* (Mapeamento Relacional de Objetos). Outro design de projeto que faz com que a comunicação entre o banco de dados e a aplicação seja mais simples, podendo obter como benefício a transcrição, é o mapeamento das entidades e modelos para as tabelas e colunas diretamente proporcionais no banco.

#### b) *Framework Bootstrap*

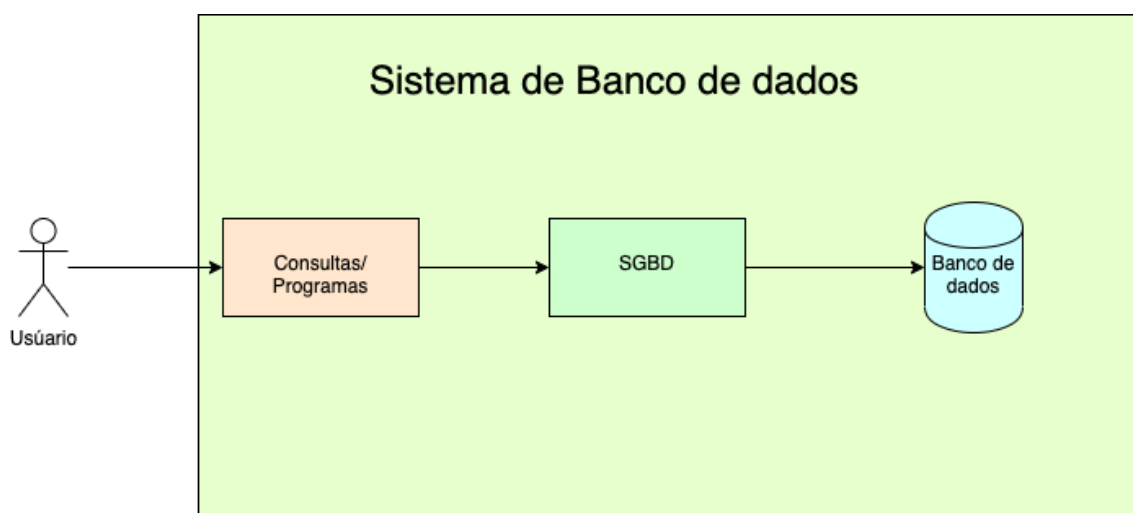
O *Bootstrap* é um *framework* de *front-end*. Nele é empregado o uso de alguns padrões de escrita utilizando HTML, CSS e *JavaScript*. Com o benefício de utilizar tais padrões, chamados de classes, que nos auxiliam na criação das *views* do nosso sistema, empregando a usabilidade e reutilização de códigos disponíveis no *framework*. Desta forma possibilita que a aplicação possa ser entregue com um visual agradável e amigável.

## 2.6 SISTEMA GERENCIADOR DE BANCO DE DADOS

Banco de dados é uma coleção de dados organizado para armazenar informações, sendo eletronicamente controladas por um Sistema Gerenciador de Banco de Dados (SGBD) (SCHIMIGUEL, 2014). Ou seja, um software cuja finalidade é salvar, listar, deletar ou atualizar os dados. Como podemos observar na FIGURA 15, em que o autor destaca que o SGBD é capaz de fazer uma ligação entre o sistema, por meio de consultas aos dados armazenados no banco de dados.

Existem diversos modelos de SGBDs, como o sistema relacional, que apresentam uma maneira de armazenar os dados por meio de tabelas, onde cada uma das suas tabelas possuem atributos e tipos de dados armazenados diferentes entre si. Exemplos de SGBD: *MySQL*, *Oracle*, *SQL Server* e *PostgreSQL*.

FIGURA 15 – Consulta SGBD



FONTE: Elaborado pelo autor

Para o armazenamento dos dados é necessário que o banco de dados supra as necessidades do sistema e seja compatível com a linguagem de programação adotada. Por esse motivo foi utilizado para este projeto o banco padrão do *Ruby on Rails*, PostgreSQL. Algumas vantagens que auxiliaram nesta escolha são:

- Gerenciamento de alto volume de dados;
- Quantidade de informações que o banco pode armazenar;

- O seu uso quando é necessário evitar redundâncias;
- O seu uso para precaver inconsistências;
- A segurança no acesso aos dados armazenados;
- O uso da linguagem SQL;
- Garantia da integridade dos dados.

## 2.7 TRABALHOS CORRELATOS

Um dos objetivos específicos desta proposta é pesquisar e analisar softwares similares para auxiliar no levantamento de requisitos para a construção da Rede Social e organograma para empresas. Após uma pesquisa inicial na Internet, foi identificada uma lista de sites com o propósito de integrar funcionários de uma empresa. Porém, somente um dos 5 (cinco) casos visitados faz uso de uma rede social promovendo integração por meio de mensagens, como apresentado no QUADRO 1.

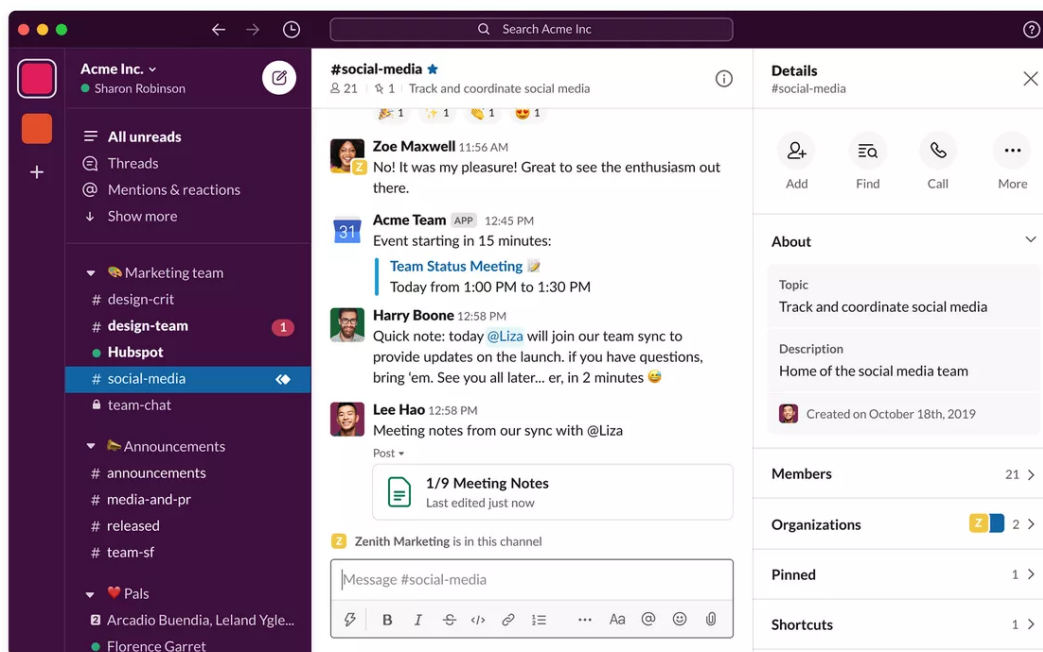
QUADRO 1 - Sites com propósito de integrar funcionários dentro das empresas

<b>Nome do Site</b>	<b>Descrição do site</b>	<b>Link do site</b>	<b>Redes Sociais</b>
<i>Team Culture</i>	Site para engajamento dos seus colaboradores	<a href="https://teamculture.com.br/">https://teamculture.com.br/</a>	Não
<i>Xerpa</i>	Plataforma de benefícios financeiros que permite aos colaboradores criarem seu cadastro no sistema de rede profissional	<a href="https://www.xerpa.com.br/">https://www.xerpa.com.br/</a>	Não
<i>Pulses</i>	Plataforma brasileira de pesquisa contínua de integração empresarial	<a href="https://www.pulses.com.br/">https://www.pulses.com.br/</a>	Não
<i>Feedz</i>	Plataforma de Engajamento de Colaboradores	<a href="https://www.feedz.com.br/">https://www.feedz.com.br/</a>	Não
<i>Slack</i>	Plataforma de comunicação comercial	<a href="http://slack.com/">http://slack.com/</a>	Sim

FONTE: Elaborado pelo autor

Dos sites e softwares listados, o único que apresenta as mesmas características de propósito do CHARTORG é o *Slack*. Trata-se de um sistema predominante no mercado em suas propostas, sendo uma plataforma com interações entre seus usuários, possibilitando que o administrador de sua organização faça o cadastro e manutenção de seus colaboradores, FIGURA 16.

FIGURA 16 – Um exemplo de conversa no sistema Slack



FONTE: Slack<sup>5</sup>

Desde que se decidiu criar uma rede social e organograma voltado para empresas em geral, com o objetivo de aproximar seus funcionários e facilitar a comunicação por meio de interações, houve a necessidade de criar um estudo para levantar e embasar mais ainda a solução proposta pela equipe. Setores empregados, recursos humanos entre outros tópicos, foram necessários para auxiliar na pesquisa. Este objetivo específico visa trazer o que foi considerado mais relevante para o desenvolvimento e amparar a compreensão do negócio, deixando o seu entendimento simples.

Para embasar a pesquisa de software foi realizada uma entrevista com a gerente de recursos humanos de uma empresa de Curitiba, APÊNDICE A. Nesta entrevista foram citados os requisitos necessários para o uso da aplicação em sua empresa, sendo os principais requisitos: interações entre os

<sup>5</sup> Disponível em: <<http://slack.com/>> Acesso em: novembro de 2020.

usuários, como posts diários e uma forma simplificada de localizar seus funcionários via sistema.

Além da entrevista, houve a necessidade de um estudo para identificar outros softwares que utilizam interações por meio de uma rede de integrantes, porém que não possuem o propósito empresarial. O objetivo foi identificar as características essenciais de uma rede social. Com esse estudo, foi possível criar o QUADRO 2.

QUADRO 2 - Comparação de softwares do tipo redes sociais

Software	Proposta	Característica possível de se agregar	Site
Twitter	Rede social e um servidor para microblogging, que permite aos usuários enviar e receber atualizações pessoais de outros contatos	Utilização de timeline para microblogging como escrever posts para sua rede social de amizades	<a href="http://twitter.com/">http://twitter.com/</a>
Facebook	Rede social que permite o usuário interagir entre seus usuários	Criar grupos e interagir apenas entre eles	<a href="http://facebook.com/">http://facebook.com/</a>
Pinterest	Rede social de fotos que traz o conceito de “mural de referências”	Criar meios para salvar mensagens para ser lido depois	<a href="https://br.pinterest.com/">https://br.pinterest.com/</a>
Instagram	Rede social para interagir através de envios de fotos e timeline específica para fotos	Criar meios de enviar fotos na timeline	<a href="https://www.instagram.com/">https://www.instagram.com/</a>

FONTE: Elaborado pelo autor

Após este estudo foi possível identificar características a serem aplicadas no CHARTORG, tais como: o uso de uma *timeline* para *microblogging*, com o objetivo de escrever *posts* para a sua rede social, utilizado pelo Twitter e Pinterest, a interação entre usuários de apenas um determinado grupo, podendo postar mensagens apenas para um esse grupo escolhido pelos usuários, como utilizado pelo Facebook e interação através de respostas e curtidas utilizado pelo Instagram.

### 3 METODOLOGIA DO TRABALHO

Este capítulo descreve a metodologia do trabalho aplicada no projeto, com as seguintes etapas: levantamento e especificação de requisitos; descrição da gerência do projeto; especificação dos diagramas de análise do sistema e descrição das ferramentas e linguagens aplicadas no projeto.

#### 3.1 REQUISITOS DO SISTEMA

Em requisitos do sistema, identificamos as necessidades, funcionalidades, regras e características do negócio que serão contemplados no desenvolvimento para o atendimento do objetivo do projeto.

##### 3.1.1 Levantamento de requisitos

Com a definição do objetivo geral do projeto iniciou-se a etapa de levantamento de requisitos. Para tanto, foi aplicada uma pesquisa na empresa Bcredi<sup>6</sup>, que atua no mercado de startups relacionada a dados financeiros, com o intuito de definir: (1) o modelo de dados; (2) definições de interações necessárias entre os usuários do sistema; (3) identificar a falta ou a complementação de funcionalidades que não constam nas aplicações analisadas previamente na seção 2.7 deste trabalho e que demonstram ser necessidades para a empresa.

Esta pesquisa foi realizada no formato de entrevista, uma entrevista informal (APÊNDICE A), no ano de 2020, com a gerente de Recursos Humanos da empresa Bcredi. Para tanto, foram realizadas algumas perguntas direcionadas, considerando o tema de relacionamento dos funcionários, a fim de que fosse possível entender quais eram os principais pontos de melhoria no dia a dia do colaborador dessa empresa e como um software voltado para a área poderia auxiliar tanto na convivência quanto na performance de seus funcionários. Após a entrevista, foi possível identificar o que o software deveria atender para ser utilizado pela empresa, assim foram definidos os principais requisitos:

- O software deve auxiliar a comunicação interna de informativos;
- O software deve permitir que haja interação entre os funcionários;

---

<sup>6</sup> <https://www.linkedin.com/company/bcrediservicosfinanceiros/>

- O software deve permitir a visualização simples de informações dos funcionários;
- O software deve permitir a comunicação, manutenção de mensagens entre funcionários, departamentos e times.

A empresa Bcredi foi escolhida para a realização da entrevista por o autor do projeto já ter trabalhado na empresa, e assim, possuir contato com a companhia. Isto facilitou o acesso e a obtenção de informações, bem como a realização entrevista.

### 3.1.2 Especificação de requisitos

Esta seção apresenta os requisitos especificados para o sistema proposto conforme o levantamento descrito na seção anterior. O QUADRO 3 descreve os requisitos Funcionais e o QUADRO 4 os Requisitos não Funcionais.

QUADRO 3 - Requisitos funcionais do sistema

<b>CÓDIGO DO REQUISITO</b>	<b>NOME DO REQUISITO</b>	<b>DESCRIÇÃO DO REQUISITO</b>
RF1	Login	O sistema deve possuir um método para que o usuário possa se autenticar
RF2	Exportar Funcionários	O sistema deve possuir um para que o usuário possa exportar seus funcionários
RF3	Cadastro	O sistema deve possuir um método para que o usuário possa se cadastrar como administrador
RF4	Funcionários	O sistema deve possuir um método para que o admin possa dar manutenção dos funcionários
RF5	Time	O sistema deve possuir um método para que o admin possa dar manutenção dos times
RF6	Departamento	O sistema deve possuir um método para que o admin possa dar manutenção dos departamentos
RF7	Organograma	O sistema deve possuir um método para visualizar o gráfico de funcionários
RF8	Empresa	O sistema deve possuir um método para dar manutenção da empresa
RF9	Timeline	O sistema deve possuir um ou mais método(s) criar post e fazer interações
RF10	Perfil	O sistema deve possuir um método para dar manutenção no perfil do funcionário ou admin

RF11	Mensagem pública	O sistema deve possuir um método para enviar mensagens públicas
RF12	Mensagem privada	O sistema deve possuir um método para enviar mensagens privadas para um usuário
RF13	Mensagem departamento	O sistema deve possuir um método para enviar mensagens públicas para um departamento
RF14	Mensagem time	O sistema deve possuir um método para enviar mensagens públicas para um time
RF15	Feed time	O sistema deve possuir um feed que mostre apenas mensagens de um time
RF16	Feed próprio	O sistema deve possuir um feed que mostre apenas suas próprias mensagens

FONTE: Elaborado pelo autor

QUADRO 4 - Requisitos não funcionais do sistema

CÓDIGO DO REQUISITO	NOME DO REQUISITO	DESCRIÇÃO DO REQUISITO
RNF1	Usabilidade	O sistema deve ser fácil de ser utilizado, contendo a facilidade de aprender e facilidade de uso
RNF2	Manutenibilidade	O sistema deve possuir um código legível e de fácil manutenção por qualquer desenvolvedor
RNF3	Portabilidade	O sistema deve ser possível rodar em qualquer navegador
RNF4	Segurança	Somente pessoas autorizadas e autenticadas podem acessar as informações do sistema
RNF5	Manutenibilidade	O sistema deve ser código aberto para ser reutilizável por qualquer desenvolvedor
RNF6	Portabilidade	O sistema deve rodar em qualquer máquina de um desenvolvedor com configurações mínimas definidas na documentação

FONTE: Elaborado pelo autor

### 3.2 GERÊNCIA DO PROJETO

Esta seção apresenta a utilização do *Scrum*, descrito na Seção 2.3.1, para a gestão e acompanhamento do desenvolvimento do projeto, sendo adaptada para ser utilizada na fase de desenvolvimento, não considerando a parte de análise ou levantamento de requisitos.

O projeto iniciou com a fase de apresentação do escopo das atividades para a professora Andreia de Jesus, orientadora do trabalho, a fim de delimitar as ações e funcionalidades a serem desenvolvidas com base no período disponível para o desenvolvimento do projeto, considerando que a equipe de trabalho é formada por um integrante. Em seguida, foram definidas etapas para o início do projeto, que envolve o levantamento de requisitos e a arquitetura do projeto, com o objetivo de identificar as atividades necessárias para compor o *Product Backlog*.

As entregas foram definidas em ciclos de uma semana, com o objetivo de definir um escopo fechado para ser entregue na sua totalidade neste período, sendo definido no começo da *sprint*.

O QUADRO 5 apresenta algumas atividades funcionais definidas inicialmente no *Product Backlog* do projeto e no QUADRO 6, temos algumas atividades não funcionais definidas inicialmente.

QUADRO 5 - Atividades pré-definidas no projeto, funcionais

Nome	Objetivo
Empregado	Empregado precisa poder adicionar foto no seu perfil
Admin	Admin - Pode editar informações da empresa
Usuário	Autenticação de usuários no sistema
Usuário	Criação de perfil para o usuário dentro do sistema
Admin	Autorização do administrador dentro do sistema
Empregado	Autorização do empregado dentro do sistema
Admin	Criar apenas uma empresa, requisito para criar qualquer ação

FONTE: Elaborado pelo autor

QUADRO 6 - Atividades pré-definidas no projeto, não funcionais

Nome	Objetivo
Usuário	Arrumar o layout do projeto
Usuário	Criar um projeto <i>Ruby on Rails</i>
Usuário	Criar um dockerfile para que seja mais simples o desenvolvimento do projeto
Documentação	Criar a introdução da documentação do projeto
Documentação	Criar a fundamentação da documentação do projeto

FONTE: Elaborado pelo autor

No QUADRO 7 apresenta problemas que poderiam ocorrer durante o desenvolvimento do projeto.

QUADRO 7 - Possíveis problemas que poderiam ocorrer durante a execução do projeto

Descrição	Possível solução	Impacto
Atrasos de entregas.	Rever a entrega ou adicionar a entrega da atividade para a próxima sprint para que seja possível a realização do projeto ou estender o prazo de entrega.	Alto
Estimativas irreais com complexidade não definida.	Re-priorizar as atividades e replanejar conforme o baixo nível de conhecimento das atividades.	Médio
Sobrecarga de trabalho das pessoas envolvidas, sendo desenvolvida apenas por um único integrante.	Conforme os participantes não puderem se dedicar todo tempo do projeto, é possível que a solução seja férias para a dedicação total ao projeto.	Médio

FONTE: Elaborado pelo autor

Dos problemas citados acima o que foi mais visível durante as semanas de desenvolvimento do projeto foi a sobrecarga de trabalho, pois, como já citado, a equipe do projeto é formada por apenas um integrante. A solução foi estender o prazo da entrega final do projeto.

A seguir o QUADRO 9 (APÊNDICE B), apresenta a utilização e a organização do modelo de *sprints* que foi seguido no decorrer do projeto,

mediante a descrição de cada uma delas, contendo um resumo do que foi feito e o produto final resultante em cada uma das *sprints*.

### 3.3 ANÁLISE DO PROJETO

#### 3.3.1 Análise dos Dados

Na análise de dados podemos explorar as principais tabelas que foram modeladas previamente no sistema. Elas servem para armazenar os seus respectivos dados e podem possuir seu respectivo modelo, sendo aqui destacado as seguintes tabelas do modelo de dados do CHARTORG: *companies*, *posts*, *squads*, *employees*, *departments*, onde são armazenados os dados do sistema. O Diagrama Entidade Relacionamento pode ser visualizado no Apêndice C.

#### 3.3.2 Análise das Funcionalidades

Para auxiliar na análise do projeto, comunicação, visão e orientação da equipe, foram elaborados diagramas da linguagem de notação UML. Os diagramas proporcionaram um melhor entendimento do projeto, compreender o ambiente, os objetivos do negócio e aplicá-los diretamente ao software. Os diagramas utilizados foram: casos de uso, diagrama de entidade relacionamento e diagrama de sequência.

##### a) Diagrama e Especificação de Casos de Uso

O CHARTORG contém duas categorias de usuários: administrador e funcionário, representadas no diagrama de caso de uso, disponível no Apêndice A. Cada categoria possui um nível de acesso e pode ter diferentes visualizações para a mesma tela. O Administrador é responsável por fazer o passo a passo inicial da criação da empresa, criando também todos os funcionários que existem nela. Além disso, é responsável por criar os departamentos da empresa e fazer a ligação entre eles e seus funcionários. O funcionário possui outro tipo de visualização, capaz de interagir com o sistema por meio de postagens em seu fluxo de notícias, curtir, comentar, visualizar cada um de seus

colegas e modificar suas informações. As especificações dos casos de uso encontram-se no Apêndice B.

#### b) Diagrama de Classe de Negócio

O software possui diversas classes no seu projeto, dentre elas as principais são: "*User*" e "*Employee*" que são responsáveis pela interação em todo o sistema, representando o usuário como funcionário ou administrador e, assim, definindo suas autorizações. Além dessas duas classes tem-se a "*Department*" e a "*Squad*" que definem onde o funcionário se situa na empresa. E a classe "*Post*" é responsável pelos *posts* da *timeline* do sistema.

Dados esses modelos, podemos observar os principais relacionamentos de classes dentro do sistema, com isso, temos o uso de uma associação reflexiva, onde existe a referência em seu próprio modelo, nesse caso o modelo de "*Employee*" faz o seu uso, sendo representado pelo uso do atributo "*employee\_id*" contendo o seu supervisor, nesse mesmo modelo podemos ver o seu relacionamento com "*Department*" e a "*Squad*", "*User*", onde é possível observar o relacionamento de associação, fazendo o uso sempre da chave primária para servir de referência, podendo ser observado o diagrama de Classe no Apêndice D.

#### c) Diagrama de Sequência de Negócio

Os diagramas de sequência de negócio estão disponibilizados no Apêndice E. Os diagramas modelados representam as principais funcionalidades do sistema: criação de *posts*, curtidas, autenticação, autorização, criação de funcionários, *times*, departamentos e a *timeline*.

### 3.4 TECNOLOGIAS E LINGUAGENS APLICADAS

Na construção da arquitetura de um projeto de sistema de informação, há a necessidade de utilizar diversas tecnologias, a fim de entregar um software funcional no final do processo. A diversidade de tecnologias e

linguagens de programação variam conforme a natureza e o objetivo do sistema.

Para o projeto em questão, as seguintes linguagem e tecnologias foram aplicadas:

a) Linguagens

- *Ruby*: utilizada pela simplicidade de sua sintaxe, produtividade e fácil utilização do *framework Ruby on Rails* no desenvolvimento de aplicações web. Além disso, é uma linguagem de código livre e com o seu *framework* é possível escrever códigos que resultam em códigos HTML e CSS que simplificam a escrita e o entendimento do código fonte do projeto (DOCUMENTIN RUBY, 2021).

b) Frameworks

- *Bootstrap*: outro *framework* que é conhecido pela sua facilidade e utilização no mercado atual, pois entrega uma grande quantidade de códigos que são utilizados para padronizar projetos (LONGEN, 2020).

c) Ferramentas de Versionamento

- *Git*: versionador utilizado para registrar histórico de edições do código fonte do projeto (FERNANDES, 2019);
- *Github*: plataforma utilizada para armazenar o código do projeto por meio de repositórios (FERNANDES, 2019).

d) Ferramentas para a Construção de Diagramas

Para a construção dos diagramas de sequência foi utilizado o software *online Sequence Diagram*<sup>7</sup>. Já para modelar os demais diagramas foi utilizado o software *online Draw IO*<sup>8</sup>. Esses dois softwares, além de intuitivos e *open source*, possuem uma facilidade na hora de implementação devido a abundante documentação disponível em seus sites, assim auxiliando na hora de escrever os diagramas.

---

<sup>7</sup> <https://sequencediagram.org>

<sup>8</sup> <https://draw.io/>

#### e) Ferramenta de Desenvolvimento

Os softwares utilizados para a implementação deste projeto foram, respectivamente, *VIM* e o *DBeaver* para desenvolvimento da aplicação; *Team Gantt* para o controle das *sprints*, assim como o gerenciamento visual do projeto e desenvolvimento do protótipo. Para UML foi utilizado *Dbeaver* junto com o diagrama de entidade-relacionamento. *DBeaver* é uma ferramenta universal para bancos de dados. Com ela é possível fazer a conexão de forma simples, integrando os dados. Além disso, manipula vários SGBDs, como: *Oracle*, *Postgres*, *MySQL*. Sendo possível gerar alguns diagramas com base na conexão do banco de dados. *Team Gantt* é uma ferramenta gratuita e online, que auxilia os desenvolvedores no gerenciamento do projeto, tanto quanto seu planejamento e cronograma.

Por fim, foi utilizado o *Docker*, uma plataforma que auxilia na virtualização ao nível do sistema operacional. Simplificando a criação de pacotes que podem ser compartilhados, assim os serviços web ficam mais intercambiáveis, eficientes e flexíveis.

#### f) Ferramentas de Edição

Este documento foi escrito com *Google Documentos*. Para a edição de códigos e interface de linhas de comando foi utilizado o *VIM*, editor de texto para Unix.

#### g) Hardware

Para o desenvolvimento do software, foi necessário o uso de um computador descrito no QUADRO 8.

QUADRO 8 - HARDWARE

Fabricante	Modelo	Processador	Memória
Apple	MacBook Pro	2,3 GHz Dual-Core Intel Core i5	8 GB

FONTE: Elaborado pelo autor

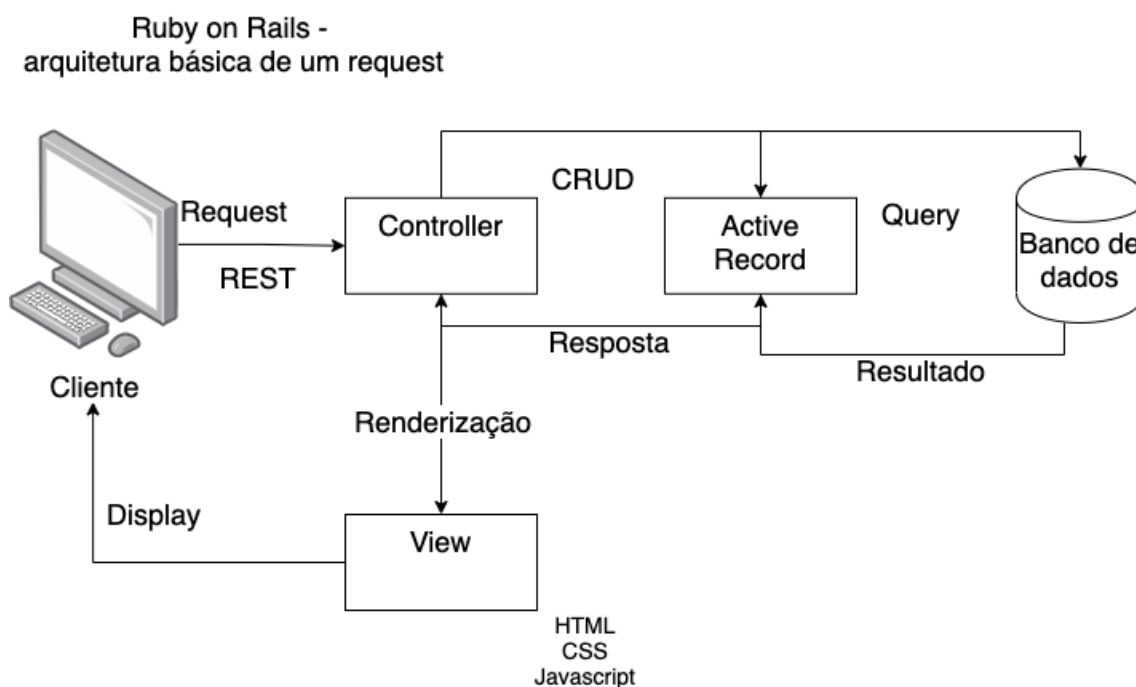
## 4 APRESENTAÇÃO DO SOFTWARE CHARTORG

Neste capítulo é apresentado a arquitetura e funcionamento do Sistema CHARTORG.

### 4.1 ARQUITETURA DO SOFTWARE

A arquitetura do sistema CHARTORG é apresentada na FIGURA 21, onde é possível observar como uma aplicação *Ruby on Rails* utiliza o padrão MVC.

FIGURA 17 – Representação arquitetura requisição



FONTE: Elaborado pelo autor

Como é possível perceber na FIGURA 17, a estrutura de um padrão de desenvolvimento *Ruby on Rails*, é composta por camadas que nelas são seguidas através do MVC.

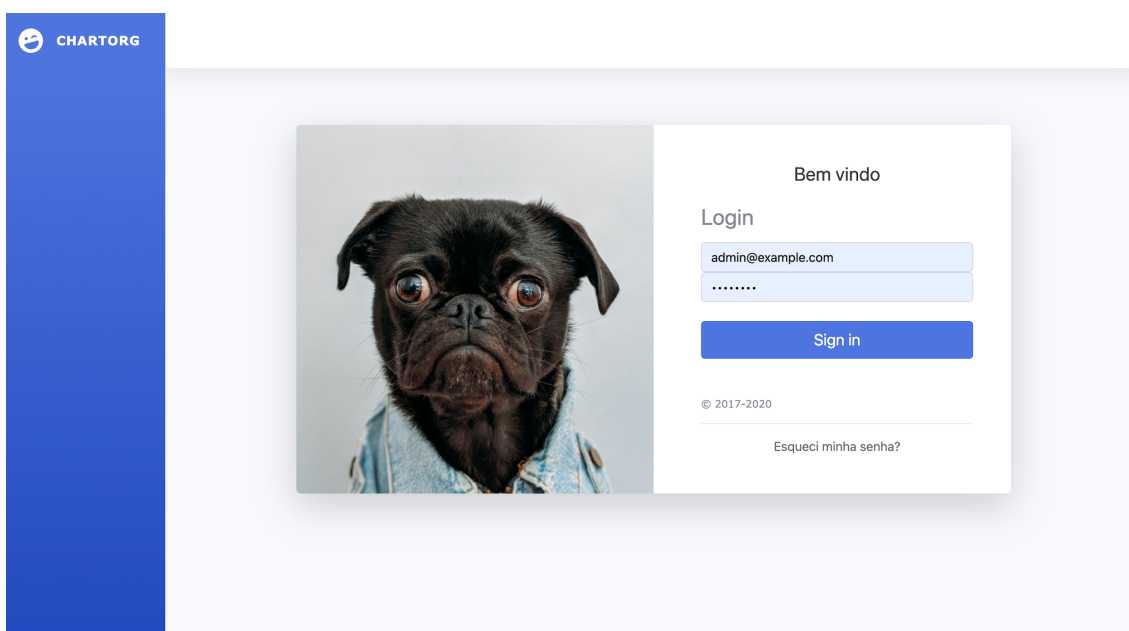
1. O Cliente faz uma requisição responsável pela comunicação com o Web Server;
2. Por meio do *Dispatcher* essa requisição é recebida, enviando-a por meio de um roteamento para a *Controller*;
3. Na *Controller* o dado é acessado por meio do *Active Record* no banco, por meio da *Model*;

4. O dado é renderizado por meio de uma *View* para o cliente.

## 4.2 O SISTEMA CHARTORG

Existe uma página de entrada para o usuário, FIGURA 22, que é realizado o cadastro dos seus dados para que seja possível o acesso à plataforma. Nesta página é requerido que o usuário tenha sua conta previamente cadastrada. Se o usuário não possuir cadastro na plataforma é necessário entrar no *link* "novo?", onde o é redirecionado para uma tela, onde será obrigatório informar algumas informações para concluir o seu cadastro.

FIGURA 18 – Tela de entrada

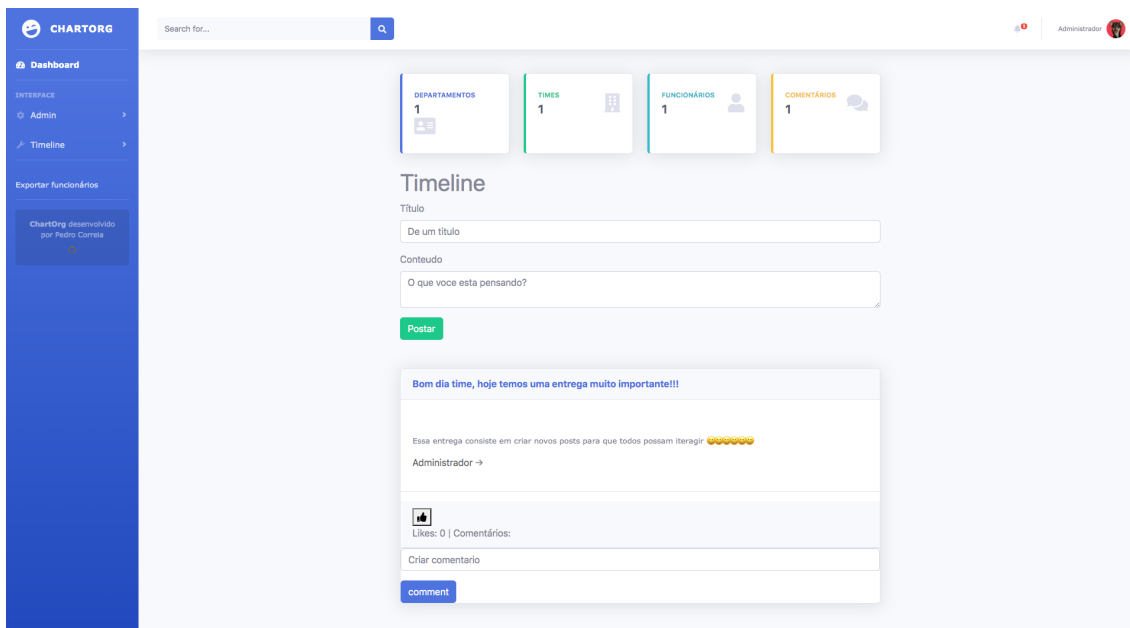


FONTE: Elaborado pelo autor

Após a realização do login pela primeira vez, o usuário é direcionado para a página de *timeline* da plataforma, FIGURA 19, para completar algumas informações com o privilégio de administrador. Possibilitando criar o perfil da empresa e interagir com a rede social, possibilitando incluir posts, realizar comentários e curtir outros.

O administrador possui acessos restritos em relação a outros usuários comuns, como dar manutenção em funcionários, usuários, times e grupos.

FIGURA 19 – Timeline



FONTE: Elaborado pelo autor

O perfil administrador possui permissão para gerar dados de funcionários para exportá-los para csv, FIGURA 20 e ver caixas, FIGURA 21, com informações adicionais sobre interação de sua plataforma.

FIGURA 20 – Exportar funcionários



FONTE: Elaborado pelo autor

FIGURA 21 – Caixas auxiliares



FONTE: Elaborado pelo autor

O perfil funcionário pode ver suas informações dentro do link onde mostra seu avatar, podendo editar suas informações, alterar seus dados cadastrais que são acessíveis a outros funcionários, tornando mais simples a busca por seu contato ou descobrir onde ele está trabalhando atualmente. Abaixo, é demonstrado o cadastro de funcionários no sistema, onde o administrador pode atualizar informações dos dados pessoais, profissionais e liberação de acesso. A partir deste cadastro o funcionário poderá acessar e interagir com o sistema, através do email disponibilizado.

FIGURA 22 – Criar funcionário

 A screenshot of a web application interface for creating a new employee. 
 - \*\*Header:\*\* 'CHARTORG' logo on the left, a search bar 'Procure um funcionário...' in the center, and 'Administrador' on the right.
 - \*\*Sidebar (left):\*\* 'Organograma' section with 'INTERFACE' sub-section containing 'Funcionalidades', 'Timeline', and 'Relatórios'. A footer note says 'ChartOrg desenvolvido por Pedro Correia'.
 - \*\*Form (right):\*\* Titled 'Funcionário'. Fields include:
 - Avatar: 'Choose File' button, 'No file chosen' text.
 - Data de início: 'dd/mm/yyyy' format with a calendar icon.
 - Nome: empty text input.
 - Cargo: empty text input.
 - Telefone: empty text input.
 - Descrição do cargo: empty text input.
 - Aniversário: 'dd/mm/yyyy' format with a calendar icon.
 - Email: empty text input.
 - Time: dropdown menu.
 - Departamento: dropdown menu.
 - Supervisor: dropdown menu.
 - 'Salvar' button at the bottom.

FONTE: Elaborado pelo autor

O perfil funcionário permite que o usuário possa interagir com outros funcionários por meio da procura por informações de sua rede, adicionar *posts*, comentários, curtir *posts*. Esse é o ambiente que o usuário pode ter a sensação de estar dentro do seu escritório informando aos seus colegas o que está acontecendo dentro do seu ambiente de trabalho, FIGURA 23.

FIGURA 23 – Tela de Post



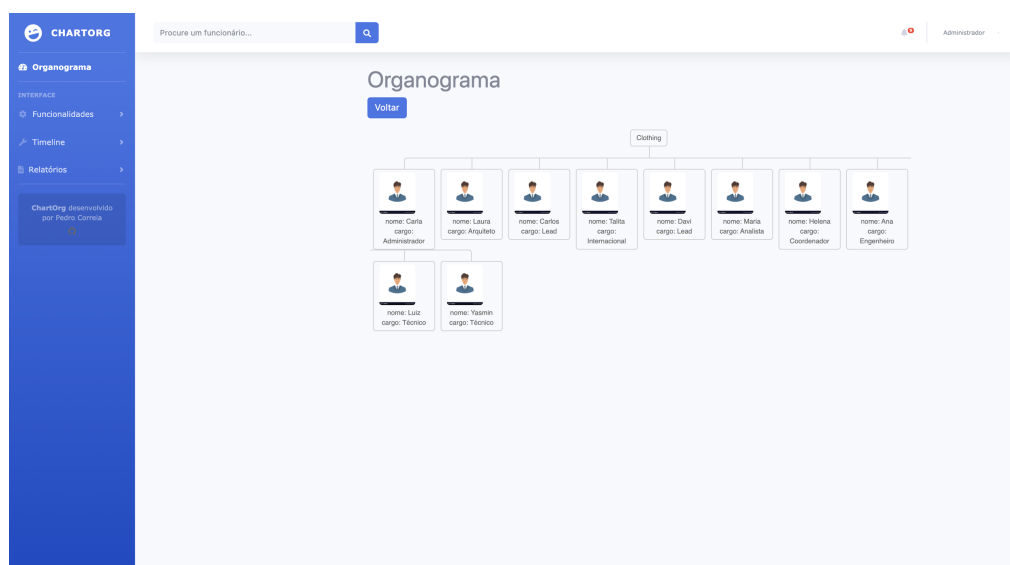
FONTE: Elaborado pelo autor

Para uma melhor visualização do organograma de como estão divididos os funcionários dentro da empresa, o administrador também possui uma maneira de poder observar tal informação. Na tela de organograma, FIGURA 24 é possível observar um gráfico de quem faz parte de um determinado departamento e qual o cargo do funcionário dentro do departamento. No *card* do funcionário existe um link onde ele é redirecionado para seu perfil completo, assim como o departamento possui um link onde é possível ver mais informações.

Na visualização detalhada do funcionário, FIGURA 25, o usuário pode saber qual o cargo do seu colega, seu número de telefone, sua data de

nascimento, desde quando está na empresa, qual departamento ou time que está participando no momento, além de um link para que seja possível enviar um e-mail para o colega, se desejar. Adicionalmente, nesta tela encontra-se disponível um link para o time que ele faz parte, com detalhes dos participantes do time, com acesso a uma lista de colegas, seus cargos e telefones para poder manter contato, também uma breve descrição do objetivo do time.

FIGURA 24 – Organograma



FONTE: Elaborado pelo autor

FIGURA 25 – Visualizar funcionário

The screenshot displays the 'Visualizar funcionário' (View Employee) interface in the CHARTORG system. The interface includes a search bar at the top with the text 'Procure um funcionário...' and a search icon. A sidebar on the left contains navigation options: 'Organograma', 'INTERFACE', 'Funcionalidades', 'Timeline', and 'Relatórios'. The main content area shows the profile of Sr. Luiz Miguel Moura. The profile includes a photo placeholder, a name, a title, a phone number, a description, a date of birth, and a date of entry into the company. Below the profile information, there are several fields and a button:

**VectorStock** Illustration by VectorStock

**Sr. Luiz Miguel Moura**

**Meu cargo:** Investidor Gerente de TI

**Meu telefone:** (65) 98549-4257

**Minha descrição:** Saúde

**Data de nascimento:** 12/28/1993

**Data que entrei na empresa:** 06/05/2020

Eu participo dos times:

Sou do time: Games

Meu departamento é: Toys

[twana\\_ernser@ufpr.com](mailto:twana_ernser@ufpr.com)

FONTE: Elaborado pelo autor

Seguindo essa visualização detalhada, também existe uma tela, FIGURA 26, onde é possível observar uma tabela contendo a lista de funcionários que fazem parte de um grupo de departamento ou time.

FIGURA 26 – Visualização do departamento

Procure um funcionário...

Administrador

Departamento: Electronics

Funcionários que fazem parte do departamento

Nome	Cargo	Telefone
Alice Madeira	Estrategista Futuro	(31) 97075-6321
Caio Videira Filho	Executivo de Varejo	(99) 94955-8834
Maria Eduarda da Mata	Estrategista de Hospitalidade Corporativo	(62) 96034-9630
Ladislau Alves	Facilitador Agricultura Pleno	(33) 97600-0765
Luana Taveira	Frente Supervisor de Industrial	(88) 92253-7792
Bruno Ximenes	Associado de Propaganda	(67) 97236-1462
Raul Moniz Jr.	Designer Investidor	(48) 94869-5249
Pablo da Rosa	Administrador de Propaganda	(85) 97270-9058
Suêlen Rolim	Coordenador Legado	(95) 96809-7826
Maria Clara Gabeira Jr.	Principal Designer de Construção	(38) 90036-3572
Dra. Ryan Aguiar	Humano Administrador de Educação	(64) 92637-2797
João Carneiro	Trainee Planejador de Hospitalidade	(19) 92485-8763
Anthony Mangueira	Desenvolvedor TI Central	(66) 96812-8888
Laura de Oliveira	Distrital Administrador de Tecnologia	(18) 98565-2488
Lavínia Aparício	Associado de Mineração Humano	(61) 95197-9621
Sra. Ana Vitória de Sá	Engenheiro de Contabilidade Pleno	(46) 90941-3182
Núbia Tavares	Internacional Representante de Administração	(31) 98036-6043
Matheus Barbosa	Gerente de Legal Investidor	(45) 95774-1438
Catarina Freitas Filho	Estrategista de Hospitalidade	(48) 92197-2607
Dr. Mércia Porto	Contato de Administração	(21) 91677-1844
Rodrigo Albuquerque	Corporativo Administrador de Agricultura	(48) 91383-0686
Milena Duarte	Orquestrador Governamental Senior	(85) 96317-4116

Editar departamento

FONTE: Elaborado pelo autor

## 5 CONSIDERAÇÕES FINAIS

Observando as pesquisas realizadas e apresentadas neste documento pode-se observar que, em geral, os sistemas que tem como o objetivo criar uma rede social em uma empresa, não possuem interações como *posts* e organogramas em seu software. Portanto, o projeto CHARTORG veio com o intuito de atender essa demanda de possibilitar a interação entre os funcionários de uma empresa em um ambiente próprio e melhorar a visão da empresa por meio de um organograma funcional.

Os dois principais benefícios que o CHARTORG pode trazer para a empresa no seu dia a dia é: (1) comunicação entre os funcionários, melhorando a produtividade através da comunicação efetiva; (2) simplicidade de organização e de sua empresa dentro de um projeto escalável e de fácil manutenção.

Este trabalho propiciou a oportunidade para o autor do projeto aprofundar os conhecimentos e aprimorar os conceitos didáticos em complemento à formação no curso de Tecnologia em Análise e Desenvolvimento de Sistemas do Setor de Educação Profissional e Tecnológica da UFPR. Adicionalmente, permitindo a experiência de colocar em prática seu conhecimento em um projeto que pode ser utilizado por uma empresa real.

Também pode ser observado pelo autor do projeto que foi possível obter conhecimentos não apenas na área de tecnologia, mas também, em outras áreas, como no caso deste projeto, comunicação e relações humanas.

Desse modo, a jornada desse projeto foi muito positiva, começando pelo levantamento de todo o conteúdo agregado, desde o processo inicial do desenvolvimento, concepção das ideias, programação das telas e reuniões com a orientadora, passando pelas suas diversas etapas e por fim a entrega do projeto.

### 5.1 RECOMENDAÇÕES PARA TRABALHOS FUTUROS

Como um dos objetivos do projeto é apresentá-lo como um objeto de conclusão de um curso de graduação, foi necessário definir um escopo plausível ao tempo disponibilizado para o desenvolvimento de um Trabalho de

Conclusão de Curso. Logo, pode-se identificar três oportunidades de melhorias que podem ser implementadas para ampliar o escopo do software:

#### 5.1.1 Mensagens privadas

O sistema possui uma página para cadastrar comentários onde todos que são parte da empresa podem visualizar. Pode-se observar que seria útil a criação de envio de mensagem privada, onde apenas um usuário ou um grupo de usuários pode recebê-la.

#### 5.1.2 Criar lista de funções e cargos

A padronização de alguns dados pode ser vantajosa quando uma empresa possui uma lista de cargos e funções já pré-determinada, com isso, futuramente pode ser incluído no sistema uma classe para adicionar listas de funções e cargos.

#### 5.1.3 Importar funcionários de uma arquivo existente

Pode-se observar que o cadastro de novos usuários é de forma individual, ou seja, será necessário muito tempo para inserir uma empresa completa. Para esta necessidade, pode ser implementada uma funcionalidade de carga de dados com vários funcionários.

## REFERÊNCIAS

AVRANE-CHOPARD, Julie; POTTER, Jaime; MUHLMANN, David. **How to develop soft skills**. Mckinsey. 11 nov. 2019. Disponível em: <<https://www.mckinsey.com/business-functions/organization/our-insights/the-organization-blog/how-to-develop-soft-skills>>. Acesso em: 9 fev. 2021.

BOOCH, G. **UML: guia do usuário**. Rio de Janeiro: Elsevier, 2012. p. 14-31,71-72,250.

DANIELE, A. **Empresas que não oferecem trabalho remoto terão dificuldade para reter talentos**. Exame. 7 ago. 2015. Disponível em: <<https://exame.com/carreira/empresas-que-nao-oferecem-trabalho-remoto-tera-o-dificuldade-para-reter-talentos-diz-executivo/>>. Acesso em: 7 nov. 2020.

DOCUMENTING RUBY(Org.). **Rails Guides**. 2021?. Disponível em: <<http://documenting-ruby.org>>. Acesso em: 3 nov. 2020.

DOYLE, D. **Exemplos de falha na comunicação em empresas: confira os 7 principais e saiba como evitá-los**. Siteware. 21 fev. 2019. Disponível em: <<https://www.siteware.com.br/comunicacao/exemplos-falha-comunicacao-empresas>>. Acesso em: 2 nov. 2020.

ESPINHA, R.G. **Metodologias de gestão de projetos**. Artia. Disponível em: <<https://artia.com/blog/gestao-de-projetos-o-que-e-para-que-serve/#metodologias-de-gestao-de-projetos/>>. Acesso em: 9 nov. 2020.

FERNANDES, D. **Git & Github: O que é? Por que? Como iniciar?**. Rocketseat. Disponível em: <<https://blog.rocketseat.com.br/iniciando-com-git-github/>>. Acesso em: 2 jan. 2021.

FLANAGAN, D; MATSUMOTO, Y. **A Linguagem de Programação Ruby**. Rio de Janeiro: Alta Books, 2009.

GILLES, B.P. **Tudo sobre Metodologia Scrum: o que é e como essa ferramenta pode te ajudar a poupar tempo e gerir melhor seus projetos**.

**Treasy**. 14 fev. 2016. Disponível em: <<https://www.treasy.com.br/blog/scrum>>. Acesso em: 3 nov. 2020.

GITHUB. **GitHut 2.0**. 2020. Disponível em: <<https://github.info/>>. Acesso em: 5 dez. 2020.

GUEDES, M. **O que é MVC?**. Treina Web. 22 jun. 2020. Disponível em: <<https://www.treinaweb.com.br/blog/o-que-e-mvc/>>. Acesso em: 7 dez. 2020.

HERTEL, R.. **CSS Inline, Interno ou Externo: Entenda a Diferença entre eles**. Hostinger. 1 dez. 2020. Disponível em: <<https://www.hostinger.com.br/tutoriais/diferenca-entre-estilos-css#Op%C3%A7%C3%A3o-3-8211-CSS-Inline>>. Acesso em: 8 jan. 2021.

LINKEDIN. **Linkedin**. 2021? Disponível em: <<https://www.linkedin.com/feed/>>. Acesso em: 3 jan. 2021.

LONGEN, A. **O Que é Bootstrap?** Guia para Iniciantes. Hostinger. 14 out. 2020. Disponível em: <<https://www.hostinger.com.br/tutoriais/o-que-e-bootstrap>>. Acesso em: 1 dez. 2020.

MARTELETO, R.M. **Análise de redes sociais - aplicação nos estudos de transferência da informação**. Ci. Inf., Brasília , v. 30, n. 1, p. 71-81, Apr. 2001. Disponível em: <[http://www.scielo.br/scielo.php?script=sci\\_arttext&pid=S0100-1965200100010009&lng=en&nrm=iso](http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0100-1965200100010009&lng=en&nrm=iso)>. Acesso em 24 Mar. 2021.

MDN Web Docs, **CSS**. 2021? Disponível em: <<https://developer.mozilla.org/pt-BR/docs/Web/CSS>>. Acesso em 09/11/2020.

MDN Web Docs, **HTML: Linguagem de Marcação de Hipertexto**. MDN Web Docs. 2021? Disponível em: <<https://developer.mozilla.org/pt-BR/docs/Web/HTML>>. Acesso em: 9 nov. 2020.

MDN Web Docs, **Sobre JavaScript**. MDN Web Docs. 2021? Disponível em: <[https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/About\\_JavaScript](https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/About_JavaScript)>.

Acesso em: 3 jan. 2021.

MICROSOFT. **Microsoft**. 2021? Disponível em:

<<https://www.microsoft.com/pt-br/microsoft-teams>>. Acesso em: 5 nov. 2020.

MTI(Org.). **Rede Social: característica, estrutura e importância social**. MTI.

19 jan. 2016. Disponível em:

<<http://www.mti.mt.gov.br/-/rede-social-caracteristica-estrutura-e-importancia-social>>. Acesso em: 2 fev. 2021.

NOGUEIRA, A. **UML - Unified Modeling Language - Introdução e Histórico**.

Linha De Código. Disponível em:

<<http://www.linhadecodigo.com.br/artigo/763/uml-unified-modeling-language-introducao-e-historico.aspx#:~:text=A%20cria%C3%A7%C3%A3o%20da%20UML%20iniciou,lan%C3%A7ado%20em%20outubro%20de%201995>>. Acesso em: 3 jan. 2021.

NOLETO, C. **Framework: o que é, como ele funciona e para que serve?**.

Betrybe. 13 fev. 2020. Disponível em:

<<https://blog.betrybe.com/framework-de-programacao/o-que-e-framework/>>.

Acesso em: 2 nov. 2020.

PROJECT MANAGEMENT INSTITUTE (Org.). **Um Guia do Conhecimento em Gerenciamento de Projetos: PMI(Guia PMBOK 6)**. Pennsylvania. ed.

[S.I.]: PMI, 2019.

RAMOS, A. **O que é MVC?**. Tableless. 26 fev. 2015. Disponível em:

<<https://tableless.com.br/mvc-afinal-e-o-que/#:~:text=MVC>>. Acesso em: 4 jan. 2021.

RUBY DOC. **Ruby Doc**. 2021?. Disponível em: <<https://ruby-doc.org/>>. Acesso em: 2 jan. 2021.

SCHIMIGUEL J. **Gerenciamento de banco de dados análise comparativa de**

**sgbd**. 2014. Disponível em:

<<https://www.devmedia.com.br/gerenciamento-de-banco-de-dados-analise-com-parativa-de-sgbd-s/30788>>. Acesso em: 9 fev. 2021.

SCRUM ORG(Org.). **Scrum**. 2021? Disponível em: <<https://www.scrum.org>>. Acesso em: 6 jan. 2021.

SILVEIRA, P. **O que é SQL?**. Alura. 14 set. 2019. Disponível em: <<https://www.alura.com.br/artigos/o-que-e-sql#:~:text=SQL%20significa%20Standard%20Query%20Language,SQL%20Server%2C%20entre%20muitos%20outros>>. Acesso em: 3 fev. 2021.

SLACK. **Slack**. 2021? Disponível em: <<https://slack.com/intl/en-br/>>. Acesso em: 4 nov. 2020.

TOMAEL, M.I.; ALCARA, A. R.; DI CHIARA, I. G. **Das redes sociais à inovação**. Ci. Inf., Brasília , v. 34, n. 2, p. 93-104, Aug. 2005. Disponível em <[http://www.scielo.br/scielo.php?script=sci\\_arttext&pid=S0100-19652005000200010&lng=en&nrm=iso](http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0100-19652005000200010&lng=en&nrm=iso)>. access on 24 Mar. 2021.

TWITTER. **Bootstrap**. 2021?. Disponível em: <<https://getbootstrap.com/>>. Acesso em: 5 jan. 2021.

UNIVERSIDADE FEDERAL DO PARANÁ. SISTEMA DE BIBLIOTECAS. **Manual de normalização de documentos científicos : de acordo com as normas da ABNT**. v. : digital il. . ed. atual. [S. l.]: UFPR, 2017. Disponível em: <https://hdl.handle.net/1884/45654>. Acesso em: 22 mar. 2021.

VERMELHO, S.C; VELHO, A.P.M.; BERTONCELLO, V.. **Sobre o conceito de redes sociais e seus pesquisadores**. Educação e Pesquisa, São Paulo, v. 41, n. 4, p. 863-881, dez. 2015. Acesso em 24 mar. 2021. Epub 10-Abr-2015.

W3SCHOOLS. **W3schools**. 2021? Disponível em: <<https://www.w3schools.com/css>>. Acesso em: 3 nov. 2020.

ZANINELLI, T.B. **A gestão da informação e da comunicação como fatores que influenciam o processo de inovação no contexto colaborativo**. Perspectivas em Ciência da Informação, n. 2, v. 18, p. 39-59, 2013.

## APÊNDICE A - TRANSCRIÇÃO DA ENTREVISTA

Entrevistado(a): Gerente de Recursos Humanos Bcredi

Realizada em: 02/11/2020

P. Quais são as maiores dificuldades dentro de uma empresa?

R. Atualmente, temos diversas etapas que enfrentamos e elas dependem da maturidade e tamanho da empresa, em geral, podemos elencar que a falta de comunicação assertiva e a dificuldade de criar processos, são as mais prejudiciais independente da empresa.

P. De que maneira um software pode auxiliar a sua empresa nesses pontos que você elencou?

R. Incluindo a comunicação em um único lugar, conseguimos centralizar a informação e manter um histórico, além de receber *feedbacks* baseados na interação dos times, outro ponto interessante é ter uma ferramenta que pode oficializar as movimentações internas e externas e manter as pessoas atualizadas de como está a estrutura organizacional.

P. Quais são as funcionalidades que julga ser necessário para o uso desse software em sua empresa?

R. Distribuição dos times em equipes de trabalho, atualização dos dados básicos de cada colaborador e da empresa, canal centralizado de comunicados com interações, *post* de conquistas e entregas ou definição das metas anuais e os avanços mensais/trimestrais.

P. Que tipo de informações você precisa ter acesso no sistema?

R. Dados pessoais dos colaboradores, hierarquias, times de trabalho, se o funcionário está ativo ou inativo.

P. Quem precisa ter acesso ao sistema?

R. Colaboradores, líderes, recursos humanos, e administradores da empresa.

## APÊNDICE B - TRANSCRIÇÃO DO RESUMO DAS SPRINTS

QUADRO 9 - Resumo das sprints

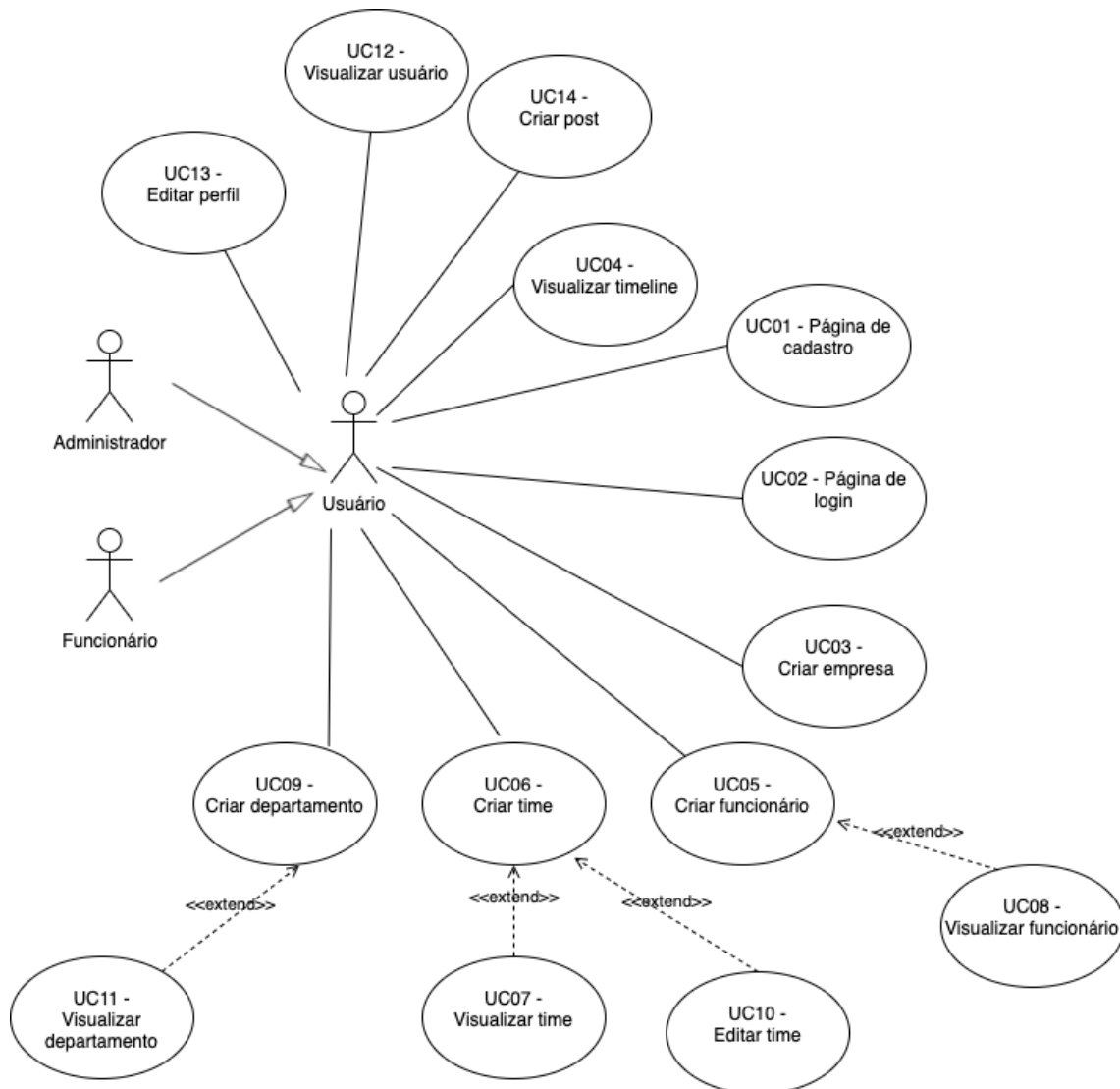
Número da sprint	Descrição da sprint	Produto final da sprint
1	<p>Demonstração para a orientadora do protótipo inicial do projeto, o modelo foi demonstrado também o primeiro documento UML que serviria de guia base para a implementação do projeto com a criação e após isso a implementação deste esquema. Além disso, deu-se início ao projeto <i>Ruby on Rails</i> com todo o seu esqueleto para que começar o desenvolvimento do projeto e essa etapa foi criada uma tela de login e cadastro, junto a suas autenticações necessárias tanto para administradores quanto para seus funcionários que nela continham campos necessários para a criação de usuários do sistema e campos para fazer login no sistema, respectivamente</p>	<ul style="list-style-type: none"> <li>- Modelo UML</li> <li>- <i>Projeto inicial Ruby on Rails</i></li> <li>- Autenticação e telas iniciais</li> </ul>
2	<p>Foram definidos quais os papéis de cada um das categorias de usuários, junto a suas permissões, dessa vez o papel de administrador seria responsável pela manutenção, adição e exclusão de seus funcionários. Também foi necessária a criação da modelagem através de objetos propostos pelo protótipo de UML apresentado na primeira semana.</p>	<ul style="list-style-type: none"> <li>- Papéis de categorias de usuários</li> <li>- Criação de modelagem proposta inicial</li> </ul>
3	<p>Implementados o modelo de companhias, departamentos e times, assim associando a cada usuário a sua determinada um relacionamento de um para um, em simultâneo, um usuário funcionário pode ter vários funcionários dependendo dele, nesse caso a proposta é que tenhamos um colaborador gerente contendo vários gerenciados, tendo isso em mente foram necessários criar métodos para a sua manutenção por completo, respectivamente com suas telas para auxiliar o cliente.</p>	<ul style="list-style-type: none"> <li>- Implementação de companhia</li> <li>- Implementação de departamentos</li> <li>- Implementação de times</li> <li>- Telas necessárias para os modelos propostos</li> </ul>
4	<p>Perfil do usuário, focando em adicionar informações pessoais, como tipo de cargo, telefone, descrição do empregado, data de nascimento. Nesse ciclo deu-se início ao desenvolvimento do organograma para que fosse</p>	<ul style="list-style-type: none"> <li>- Implementação do perfil do usuário</li> <li>- Início do organograma</li> </ul>

	possível uma melhor visualização de onde o funcionário está presente e qual o seu papel na companhia.	
5	Continuação do organograma dos funcionários da companhia, nesse ciclo foi fundamental adicionar, cargos, funções, descrições e nomes em cada um dos respectivos funcionários para ser mais simples a forma de obter informações, também como citado na sprint 3 foi preciso adicionar uma hierarquia para que pudéssemos demonstrar quem é o superior do respectivo usuário.	- Finalizado o organograma
6	Começar o desenvolvimento de uma <i>Timeline</i> , essa com o objetivo de trazer ainda mais a função de rede social para o projeto, através de comentários públicos criados pelo usuário onde todo mundo que faz parte de da companhia pudesse interagir, contendo curtidas, respostas ao comentário do autor.	- Criação da Timeline
7	Início desta documentação do projeto, através de pesquisas correlacionadas com o tema proposto e adição das pesquisas realizadas verbais e não verbais que através dela, foi realizado o projeto.	- Início da documentação
8	Adição de uma funcionalidade para que o administrador pudesse exportar dados do sistema assim como id do usuário, data de início, nome, título, telefone, data de nascimento, e-mail, time e departamento.	- Função de exportar dados dos funcionários
9	Continuidade da documentação e finalização.	- Finalização da documentação

FONTE: Elaborado pelo autor

## APÊNDICE C - DIAGRAMA DE CASO DE USO

FIGURA 27 – Diagrama caso de uso



FONTE: Elaborado pelo autor

## APÊNDICE D – ESPECIFICAÇÃO DE CASOS DE USO

### UC01 – Página de cadastro

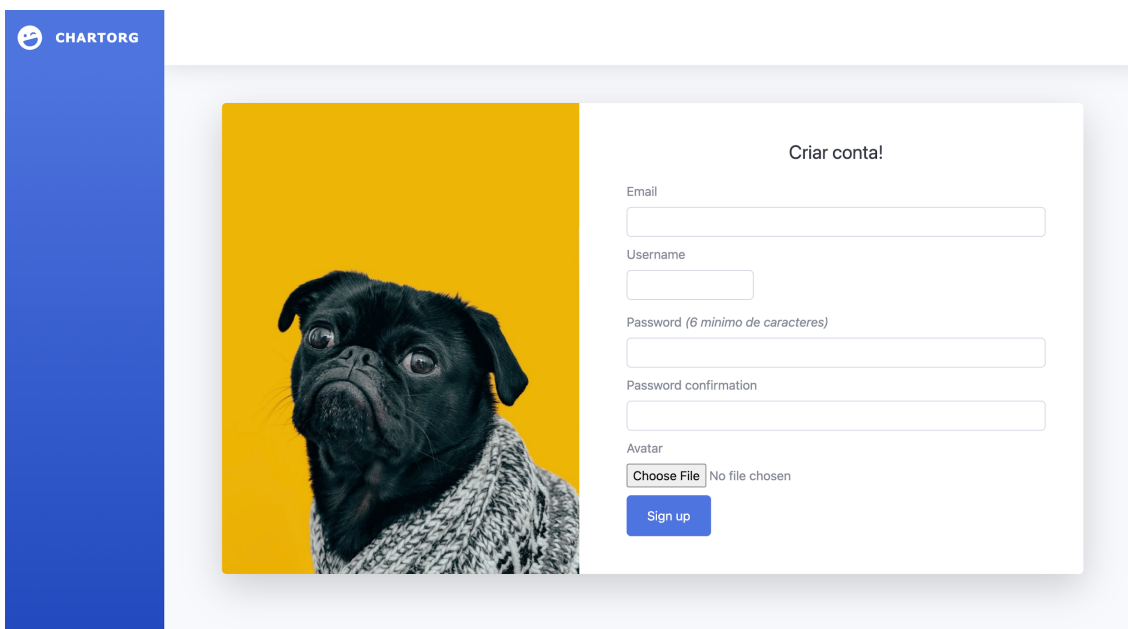
#### Descrição

Este caso de uso serve para que uma pessoa possa se cadastrar no software.

#### Data View

**DV01** tela de cadastro.

FIGURA 28 – Criar conta



The screenshot shows a registration form titled "Criar conta!" on the CHARTORG website. The form is set against a yellow background featuring a black pug dog. The form fields are: Email, Username, Password (6 mínimo de caracteres), Password confirmation, and Avatar. The Avatar field includes a "Choose File" button and the text "No file chosen". A blue "Sign up" button is located at the bottom of the form.

FONTE: Elaborado pelo autor

#### Pré-condições

Não se aplica.

#### Pós-condições

Após o fim normal deste caso de uso a aplicação deve:

1. Ser redirecionado para timeline(DV04).

#### Ator Primário

Usuário.

### **Fluxo de Eventos Principal**

1. O usuário preenche o campo Nome Completo(R1).
2. O usuário preenche o campo Endereço de e-mail (R1) (R2).
3. O usuário preenche o campo Senha (R1) (R3).
4. O usuário adiciona uma foto em "avatar" (R1) (R3).
5. O usuário pressiona o botão “sign-up”.
6. A aplicação verifica se o e-mail do usuário já existe (E2).
7. A aplicação salva o usuário com seus parâmetros;
8. A aplicação mostra a notificação “Usuário cadastrado!”.
9. A aplicação redireciona para Timeline(DV04).
10. O caso de uso é encerrado.

### **Fluxos Alternativos**

Não se aplica.

### **Regras de Negócios**

R1. Campo obrigatório.

R2. O campo precisa ser do tipo e-mail.

R3. A atenção guarda as senhas no banco de forma criptografada.

### **Fluxo de Exceção**

E1. A aplicação verifica que o e-mail já existe.

1. A aplicação apresenta a notificação “Usuário já existe!”.
2. O caso de uso é finalizado.

## UC02 – Página de login

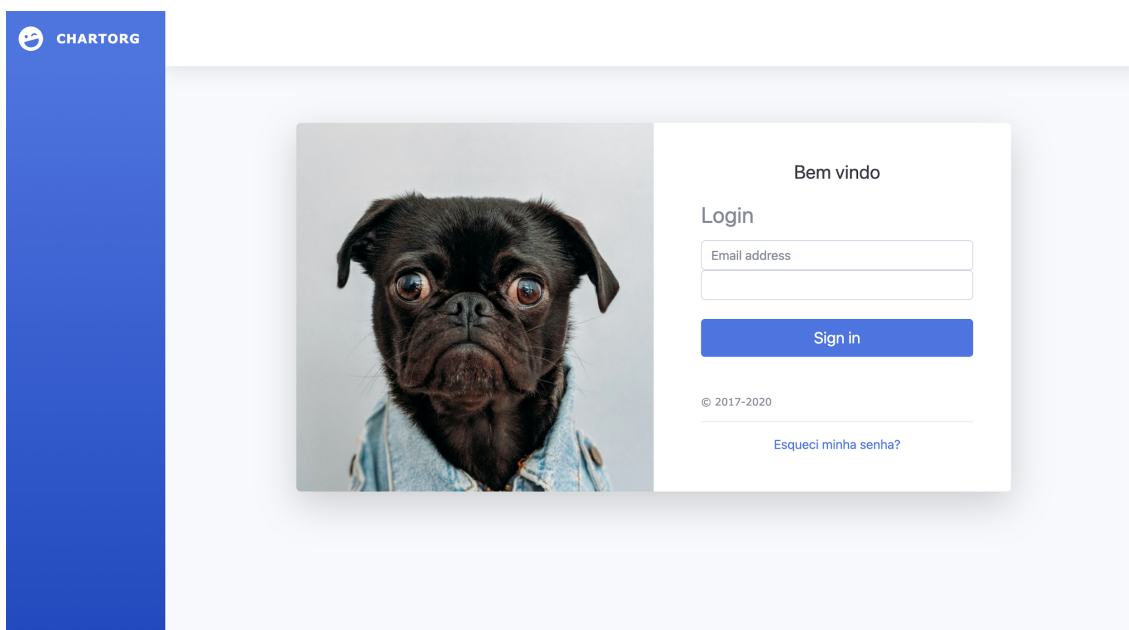
### Descrição

Este caso de uso serve para que o usuário possa se logar no software.

### Data View

**DV02** tela de login.

FIGURA 29 – Tela de login



FONTE: Elaborado pelo autor

### Pré-condições

Não se aplica.

### Pós-condições

Após o fim normal deste caso de uso a aplicação deve:

1. Ser redirecionado para timeline(DV04).

### Ator Primário

Usuário.

### Fluxo de Eventos Principal

1. O usuário preenche o campo e-mail (R1).
2. O usuário preenche o campo senha (R1) (R3).
3. O usuário pressiona o botão “sign in”.
4. A aplicação armazena na sessão o usuário.
5. O usuário é redirecionado para timeline(DV04).
6. O caso de uso é encerrado.

### **Fluxos Alternativos**

**A1.** O usuário não possui cadastro.

1. O usuário aperta o botão novo?
2. O usuário é direcionado para a tela de cadastro(DV01).

### **Regras de Negócios**

**R1.** Campo obrigatório.

**R2.** A aplicação verifica se a senha é válida para o e-mail.

**R3.** A aplicação salva na sessão o usuário.

### **Fluxo de Exceção**

**E1.** A aplicação verifica que não existe a combinação de e-mail e senha:

1. A aplicação apresenta a notificação “E-mail invalido”.
2. O caso de uso é finalizado.

## UC03 - Criar empresa

### Descrição

Este caso de uso serve para que o usuário possa criar a sua empresa no software.

### Data View

**DV03** Tela de criar empresa.

FIGURA 30 – Tela de criar empresa

The screenshot shows a web application interface for creating a company. The main content area is titled 'Companhia' and contains a form with the following fields:

- Nome:** Empresa fake
- Url:** empresa.com.br
- Email da empresa:** email@empresafake.com.br

A 'Criar' button is located at the bottom of the form. The interface also features a search bar at the top, a user profile 'Administrador' in the top right, and a sidebar on the left with navigation options like 'Dashboard' and 'Criar sua empresa'.

FONTE: Elaborado pelo autor

### Pré-condições

Este caso de uso pode iniciar somente se:

1. A aplicação tiver executado o UC02 – Login.

### Pós-condições

Após o fim normal deste caso de uso a aplicação deve:

1. Ser redirecionado para timeline(DV04).

### Ator Primário

Usuário.

### **Fluxo de Eventos Principal**

1. O usuário preenche o campo nome (R1).
2. O usuário preenche o campo URL (R1).
3. O usuário preenche o campo e-mail da empresa (R1).
4. O usuário pressiona o botão “criar”.
5. A empresa é criada e registrada no banco.
6. O usuário é redirecionado para timeline(DV04).
7. O caso de uso é encerrado.

### **Fluxos Alternativos**

Não se aplica.

### **Regras de Negócios**

**R1.** Campo obrigatório.

**R2.** A aplicação verifica se a senha é válida para o e-mail.

### **Fluxo de Exceção**

Não se aplica.

## UC04 -Visualizar timeline

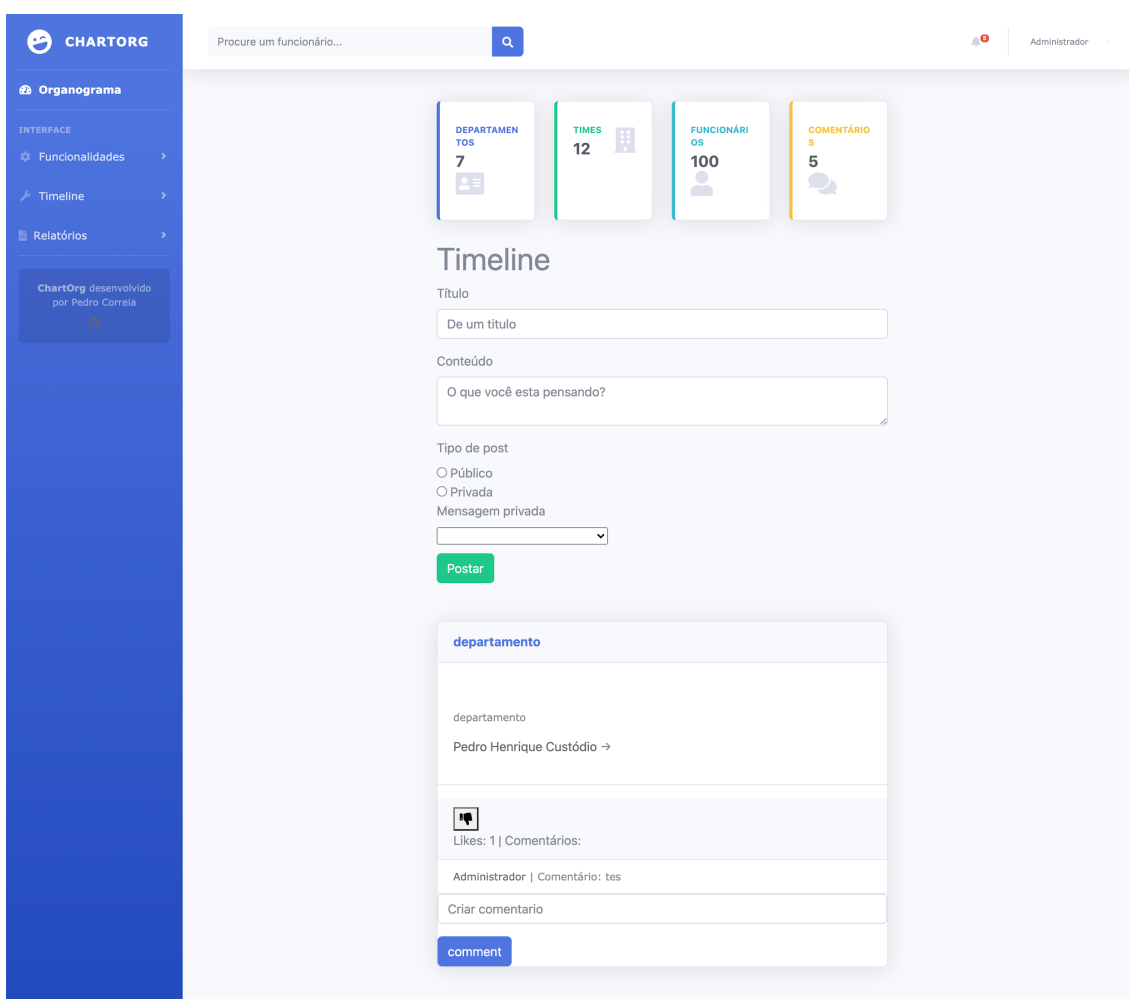
## Descrição

Este caso de uso é onde o cliente é redirecionado.

## Data View

DV04 Tela de timeline.

FIGURA 31 – Tela de timeline



FONTE: Elaborado pelo autor

## Pré-condições

Este caso de uso pode iniciar somente se:

1. A aplicação tiver executado o UC02 – Login.

## Pós-condições

Não se aplica.

### **Ator Primário**

Usuário.

### **Fluxo de Eventos Principal**

1. O usuário se loga.
2. A aplicação busca os posts para o usuário logado.
3. A aplicação busca os comentários para o usuário logado.
4. A aplicação busca a quantidade de comentários.
5. A aplicação busca a quantidade de times.
6. A aplicação busca a quantidade de departamentos.
7. A aplicação busca a quantidade de funcionários.
8. A aplicação mostra informações, como quantidade de comentários, funcionários, times, departamentos.
9. A aplicação exibe a tela (DV04).
10. Fim de caso uso.

### **Fluxos Alternativos**

Não se aplica.

### **Regras de Negócios**

Não se aplica.

### **Fluxo de Exceção**

Não se aplica.

## UC05 - Criar funcionário

**Descrição**

Este caso de uso serve para que o usuário possa cadastrar um funcionário.

**Data View**

**DV05** Tela de criar funcionário.

FIGURA 32 – Tela de criar funcionário

The screenshot shows the 'Criar funcionário' form in the CHARTORG application. The form is titled 'Funcionário' and contains the following fields and controls:

- Avatar:** A file upload field with a 'Choose File' button and the text 'No file chosen'.
- Data de início:** A date input field with the placeholder 'dd/mm/yyyy' and a calendar icon.
- Nome:** A text input field.
- Cargo:** A text input field.
- Telefone:** A text input field.
- Descrição do cargo:** A text input field.
- Aniversário:** A date input field with the placeholder 'dd/mm/yyyy' and a calendar icon.
- Email:** A text input field.
- Time:** A dropdown menu.
- Departamento:** A dropdown menu.
- Supervisor:** A dropdown menu.
- Salvar:** A button to save the form.

The left sidebar shows the application menu with 'Organograma' selected. The top navigation bar includes a search bar with the text 'Procure um funcionário...' and a search icon, and the user's name 'Administrador'.

FONTE: Elaborado pelo autor

**Pré-condições**

Este caso de uso pode iniciar somente se:

1. A aplicação tiver executado o UC02 – Login.

**Pós-condições**

Após o fim normal deste caso de uso a aplicação deve:

1. Ser redirecionado para timeline(DV04).

### **Ator Primário**

Usuário.

### **Fluxo de Eventos Principal**

1. O usuário preenche o campo avatar (R1).
2. O usuário preenche o campo data de início (R1)(R2).
3. O usuário preenche o campo nome (R1).
4. O usuário preenche o campo cargo (R1).
5. O usuário preenche o campo telefone (R1).
6. O usuário preenche o campo, descrição do cargo (R1).
7. O usuário preenche o campo aniversário (R1)(R2).
8. O usuário preenche o campo e-mail (R1)(R3).
9. O usuário preenche o campo time.
10. O usuário preenche o campo departamento.
11. O usuário preenche o campo supervisor.
12. O usuário pressiona o botão “criar”(R4).
13. A aplicação salva o registro no banco.
14. O usuário é redirecionado para timeline(DV04).
15. O caso de uso é encerrado.

### **Fluxos Alternativos**

Não se aplica.

### **Regras de Negócios**

- R1.** Campo obrigatório.
- R2.** Campo precisa ser do tipo data.
- R3.** Campo precisa ser do tipo e-mail.
- R3.** Campo precisa ser do tipo e-mail.
- R3.** Precisa ser salvo no banco.

**Fluxo de Exceção**

**E1.** A aplicação verifica que já existe o e-mail.

1. A aplicação apresenta a notificação “E-mail já existe”.
2. O caso de uso é finalizado.

## UC06 - Criar Times

**Descrição**

Este caso de uso serve para que o usuário possa cadastrar os times.

**Data View**

**DV06** Tela de criar times.

FIGURA 33 – Tela de criar times

CHARTORG

Procure um funcionário...

Administrador

Funcionário

Avatar  
Choose File No file chosen

Data de inicio  
dd/mm/yyyy

Nome

Cargo

Telefone

Descrição do cargo

Aniversário  
dd/mm/yyyy

Email

Time

Departamento

Supervisor

Salvar

FONTE: Elaborado pelo autor

**Pré-condições**

Este caso de uso pode iniciar somente se:

1. A aplicação tiver executado o UC02 – Login.

**Pós-condições**

Após o fim normal deste caso de uso a aplicação deve:

1. Ser redirecionado para visualizar o times(DV07).

### **Ator Primário**

Usuário.

### **Fluxo de Eventos Principal**

1. O usuário preenche o campo nome (R1).
2. O usuário preenche o campo descrição (R1).
3. O usuário pressiona o botão "criar"(R2).
4. A aplicação salva o registro no banco.
5. O usuário é redirecionado para visualizar o time.
6. O caso de uso é encerrado.

### **Fluxos Alternativos**

Não se aplica.

### **Regras de Negócios**

**R1.** Campo obrigatório.

**R2.** A aplicação salva o registro.

### **Fluxo de Exceção**

Não se aplica.

## UC07 - Visualizar Times

### Descrição

Este caso de uso serve para que o usuário possa visualizar os times.

### Data View

**DV07** Tela de visualizar times.

FIGURA 34 – Tela de visualizar times

The screenshot shows the CHARTORG application interface. On the left is a blue sidebar with navigation options: Organograma, INTERFACE, Funcionalidades, Timeline, and Relatórios. The main content area is titled 'Times' and contains a table with the following data:

Nome	Descrição	
Tools	Tools	Mostrar Editar Deletar
Shoes	Shoes	Mostrar Editar Deletar
Industrial	Industrial	Mostrar Editar Deletar
Health	Health	Mostrar Editar Deletar
Jewelry	Jewelry	Mostrar Editar Deletar
Electronics	Electronics	Mostrar Editar Deletar
Sports	Sports	Mostrar Editar Deletar
Games	Games	Mostrar Editar Deletar
Beauty	Beauty	Mostrar Editar Deletar
Automotive	Automotive	Mostrar Editar Deletar
Garden	Garden	Mostrar Editar Deletar
Toys	Toys	Mostrar Editar Deletar

At the bottom of the table area, there is a button labeled 'Novo time'.

FONTE: Elaborado pelo autor

### Pré-condições

Este caso de uso pode iniciar somente se:

1. A aplicação terá executado o UC02 – Login.

### Pós-condições

Não se aplica.

### Ator Primário

Usuário.

**Fluxo de Eventos Principal**

1. A aplicação busca os times existentes no banco.
2. O usuário visualiza os times existentes.
3. O caso de uso é encerrado.

**Fluxos Alternativos**

Não se aplica.

**Regras de Negócios**

Não se aplica.

**Fluxo de Exceção**

Não se aplica.

## UC08 - Visualizar funcionários

### Descrição

Este caso de uso serve para que o usuário possa visualizar os funcionários.

### Data View

**DV08** Tela de visualizar funcionários.

FIGURA 35 – Tela de visualizar funcionários

Data de início	Nome	Função	Telefone	Descrição	Data de nascimento	Email	
02/20/2017	Pedro Henrique Custódio	Supervisor de Industrial Lead	(34) 93500-3498	Mineração	01/19/1992	moses@ufpr.org	Mostrar Editar Deletar
07/19/2020	Oliver Siqueira	Arquiteto de Agricultura Lead	(71) 94083-4245	TI	12/24/1990	theron@ufpr.org	Mostrar Editar Deletar
06/10/2017	Rafaela Conceição Neto	Analista Educação Consumidor	(49) 93754-0564	Varejo	07/17/1999	monroe@ufpr.name	Mostrar Editar Deletar
10/17/2016	Sra. Isabel Sales	Regional Técnico de Legal	(53) 95628-2745	Hospitalidade	06/11/1985	felicidad@ufpr.co	Mostrar Editar Deletar
05/24/2020	Marcelo Louzada Jr.	Principal Analista de Imobiliária	(19) 90808-7366	Marketing	09/12/1974	ramon.strosin@ufpr.net	Mostrar Editar Deletar
07/02/2016	Srta. Guilherme Longuinho	Desenvolvedor Administração Distrital	(96) 92693-6481	Tecnologia	04/08/1971	pandora@ufpr.info	Mostrar Editar Deletar
04/18/2016	Sra. Isabella Mendes	Coordenador Central	(22) 94421-9890	Educação	06/05/1971	erich.reichel@ufpr.info	Mostrar Editar Deletar

FONTE: Elaborado pelo autor

### Pré-condições

Este caso de uso pode iniciar somente se:

1. A aplicação tiver executado o UC02 – Login.

### Pós-condições

Não se aplica.

### Ator Primário

Usuário.

### Fluxo de Eventos Principal

1. A aplicação busca os funcionários existentes no banco.
2. O usuário visualiza os funcionários existentes.
3. O caso de uso é encerrado.

**Fluxos Alternativos**

Não se aplica.

**Regras de Negócios**

Não se aplica.

**Fluxo de Exceção**

Não se aplica.

## UC09 – Criar departamentos

### Descrição

Este caso de uso serve para que o usuário possa cadastrar os departamentos.

### Data View

**DV09** Tela de criar departamentos.

FIGURA 36 – Tela de criar departamentos

The screenshot displays the 'Criar departamentos' (Create departments) screen in the CHARTORG application. The interface includes a blue sidebar with navigation options: 'Organograma', 'Funcionalidades', 'Timeline', and 'Relatórios'. The main content area features a search bar at the top with the placeholder 'Procure um funcionário...'. Below the search bar, there is a form titled 'Departamento' with two input fields: 'Nome' and 'Descrição'. A 'Criar' button is positioned at the bottom left of the form. The top right corner shows the user's role as 'Administrador'.

FONTE: Elaborado pelo autor

### Pré-condições

Este caso de uso pode iniciar somente se:

1. A aplicação tiver executado o UC02 – Login.

### Pós-condições

Após o fim normal deste caso de uso a aplicação deve:

1. Ser redirecionado para visualizar departamentos(DV11).

**Ator Primário**

Usuário.

**Fluxo de Eventos Principal**

1. O usuário preenche o campo nome(R1).
2. O usuário preenche o campo descrição(R1).
3. O usuário pressiona o botão “criar”(R1).
4. A aplicação salva o registro no banco.
5. O caso de uso é encerrado.

**Fluxos Alternativos**

Não se aplica.

**Regras de Negócios**

**R1.** Campo obrigatório.

**R2.** A aplicação salva o registro.

**Fluxo de Exceção**

Não se aplica.

## UC11 - Visualizar departamentos

**Descrição**

Este caso de uso serve para que o usuário possa visualizar os departamentos.

**Data View**

**DV11** Tela de visualizar departamentos.

FIGURA 37 – Tela de visualizar departamentos

Nome	Descrição	
Electronics	Electronics	Mostrar <a href="#">Editar</a> <a href="#">Deletar</a>
Clothing	Clothing	Mostrar <a href="#">Editar</a> <a href="#">Deletar</a>
Health	Health	Mostrar <a href="#">Editar</a> <a href="#">Deletar</a>
Grocery	Grocery	<a href="#">Mostrar</a> <a href="#">Editar</a> <a href="#">Deletar</a>
Toys	Toys	<a href="#">Mostrar</a> <a href="#">Editar</a> <a href="#">Deletar</a>
Computers	Computers	Mostrar <a href="#">Editar</a> <a href="#">Deletar</a>
Shoes	Shoes	Mostrar <a href="#">Editar</a> <a href="#">Deletar</a>

Novo departamento

FONTE: Elaborado pelo autor

**Pré-condições**

Este caso de uso pode iniciar somente se:

1. A aplicação tiver executado o UC02 – Login.

**Pós-condições**

Não se aplica.

**Ator Primário**

Usuário.

**Fluxo de Eventos Principal**

1. A aplicação busca os departamentos existentes no banco.
2. O usuário visualiza os departamentos existentes.
3. O caso de uso é encerrado.

**Fluxos Alternativos**

Não se aplica.

**Regras de Negócios**

Não se aplica.

**Fluxo de Exceção**

Não se aplica.

## UC12 - Visualizar usuários

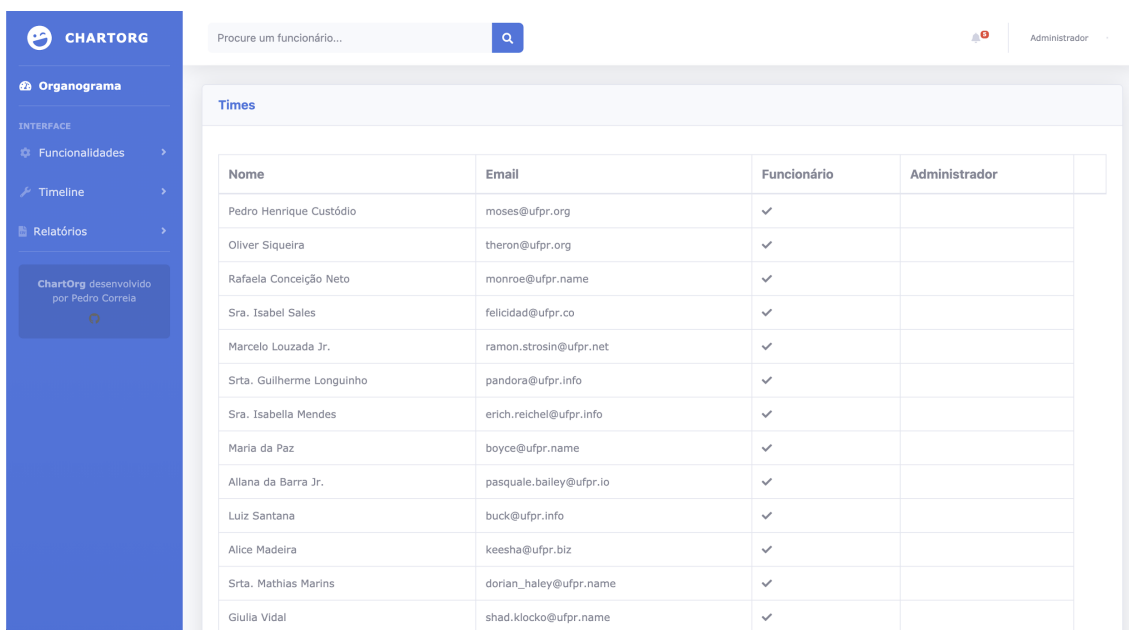
### Descrição

Este caso de uso serve para que o usuário possa visualizar os usuários.

### Data View

DV12 Tela de visualizar usuários.

FIGURA 38 – Tela de visualizar usuários



The screenshot shows the CHARTORG application interface. On the left is a blue sidebar with navigation options: Organograma, INTERFACE, Funcionalidades, Timeline, and Relatórios. The main content area has a search bar at the top with the text 'Procure um funcionário...' and a search icon. Below the search bar is a table titled 'Times' with the following data:

Nome	Email	Funcionário	Administrador
Pedro Henrique Custódio	moses@ufpr.org	✓	
Oliver Siqueira	theron@ufpr.org	✓	
Rafaela Conceição Neto	monroe@ufpr.name	✓	
Sra. Isabel Sales	felicidad@ufpr.co	✓	
Marcelo Louzada Jr.	ramon.strosin@ufpr.net	✓	
Srta. Gullherme Longuinho	pandora@ufpr.info	✓	
Sra. Isabella Mendes	erich.reichel@ufpr.info	✓	
Maria da Paz	boyce@ufpr.name	✓	
Allana da Barra Jr.	pasquale.bailey@ufpr.io	✓	
Luiz Santana	buck@ufpr.info	✓	
Alice Madeira	keesha@ufpr.biz	✓	
Srta. Mathias Marins	dorian_haley@ufpr.name	✓	
Giulia Vidal	shad.klocko@ufpr.name	✓	

FONTE: Elaborado pelo autor

### Pré-condições

Este caso de uso pode iniciar somente se:

1. A aplicação tiver executado o UC02 – Login.

### Pós-condições

Não se aplica.

### Ator Primário

Usuário.

### Fluxo de Eventos Principal

1. A aplicação busca os usuários existentes no banco.
2. O usuário visualiza os usuários existentes.
3. O caso de uso é encerrado.

**Fluxos Alternativos**

Não se aplica.

**Regras de Negócios**

Não se aplica.

**Fluxo de Exceção**

Não se aplica.

## UC13 - Editar perfil

**Descrição**

Este caso de uso serve para que o usuário possa editar seu perfil.

**Data View**

DV12 Tela de editar perfil.

FIGURA 39 – Tela de editar perfil

CHARTORG

Procure um funcionário...

Administrador

Organograma

INTERFACE

- Funcionalidades
- Timeline
- Relatórios

ChartOrg desenvolvido por Pedro Correia

Funcionário

Email  
admin@mail.com

Username  
Administrador

Password (Se você nao for mudar pode deixar em branco)  
6 mínimo de caracteres

Password confirmation

Current password (Precisamos da sua senha para confirmar as alteracoes)

Avatar  
Choose File No file chosen

Editar

Voltar

FONTE: Elaborado pelo autor

**Pré-condições**

Este caso de uso pode iniciar somente se:

1. A aplicação tiver executado o UC02 – Login.

**Pós-condições**

Não se aplica.

**Ator Primário**

Usuário.

### **Fluxo de Eventos Principal**

1. A aplicação busca o usuário existente no banco.
2. O usuário preenche o campo e-mail(R1)(R3).
3. O usuário preenche o campo username(R1)(R2).
4. O usuário preenche o campo password(R1)(R2).
5. O usuário preenche o campo password confirmation(R1)(R2).
6. O usuário preenche o campo current password(R1).
7. O usuário pressiona o botão “editar”(R4)(R5).
8. A aplicação salva os novos dados do usuário existente no banco.
9. O caso de uso é encerrado.

### **Fluxos Alternativos**

Não se aplica.

### **Regras de Negócios**

- R1.** Campo obrigatório.
- R2.** Campo precisa ser do tipo senha.
- R3.** Campo precisa ser do tipo e-mail.
- R4.** A aplicação verifica se a senha está correta.
- R5.** A aplicação salva o registro.

### **Fluxo de Exceção**

**E1.** A aplicação verifica que não existe a combinação de e-mail e senha estão incorretos.

1. A aplicação apresenta a notificação “Senha incorreta!”.
2. O caso de uso é finalizado.

**E2.** A aplicação verifica que o e-mail já existe.

1. A aplicação apresenta a notificação “E-mail já existe!”.
2. O caso de uso é finalizado.



## UC14 – Criar post

**Descrição**

Este caso de uso serve para que o usuário possa criar post.

**Data View**

**DV14** Tela de criar post.

FIGURA 40 – Tela de criar post

The screenshot shows the CHARTORG application interface. On the left is a blue sidebar with the logo and navigation menu. The main area features a search bar at the top, a dashboard with four statistics cards, and a 'Timeline' section with a form to create a post. The form includes a title field, a content area, and radio buttons for post type, with a dropdown for 'Mensagem privada'.

FONTE: Elaborado pelo autor

**Pré-condições**

Este caso de uso pode iniciar somente se:

1. A aplicação tiver executado o UC02 – Login.

**Pós-condições**

Não se aplica.

**Ator Primário**

Usuário.

**Fluxo de Eventos Principal**

1. O usuário preenche o campo título(R1).
2. O usuário preenche o campo conteúdo(R1).
3. O usuário pressiona o botão “postar”(R2).
4. A aplicação salva os dados do post no banco.
5. O caso de uso é encerrado.

### **Fluxos Alternativos**

**A1.** O usuário clica no ícone de joia em um post existente.

1. A aplicação grava ou apaga a curtida.
2. O Caso de Uso é finalizado.

**A2.** O usuário adiciona um comentário em um post existente.

1. A aplicação grava o comentário.
2. O Caso de Uso é finalizado.

**A3.** O usuário clica em exportar funcionários.

1. A aplicação gera um csv com os funcionários cadastrados.
2. O Caso de Uso é finalizado.

**A4.** O usuário clica no ícone de sino.

1. A aplicação mostra os últimos 3 comentários.
2. O Caso de Uso é finalizado.

### **Regras de Negócios**

**R1.** Campo obrigatório.

**R2.** A aplicação salva o registro.

### **Fluxo de Exceção**

Não se aplica.

## UC15 – Visualizar organograma

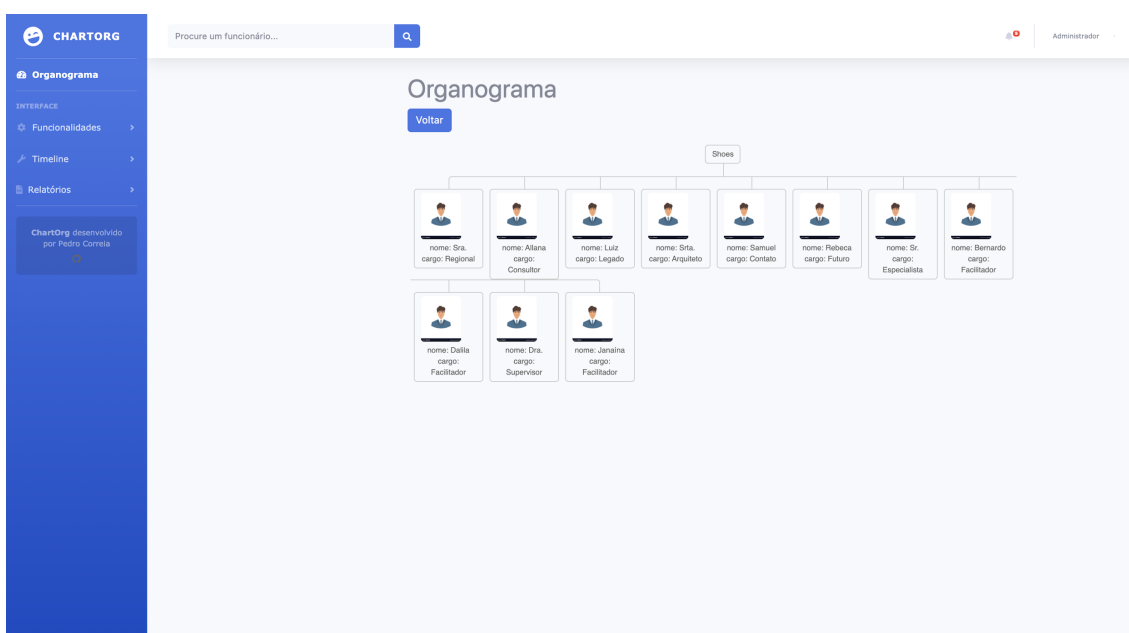
### Descrição

Este caso de uso serve para que o usuário visualize o organograma.

### Data View

**DV15** Tela de visualizar organograma.

FIGURA 41 – Tela de visualizar organograma



FONTE: Elaborado pelo autor

### Pré-condições

Este caso de uso pode iniciar somente se:

1. A aplicação tiver executado o UC02 – Login.

### Pós-condições

Não se aplica.

### Ator Primário

Usuário.

### Fluxo de Eventos Principal

1. A aplicação busca os funcionários do departamento no banco.
2. O usuário visualizar o gráfico.
3. O caso de uso é encerrado.

**Fluxos Alternativos**

Não se aplica.

**Regras de Negócios**

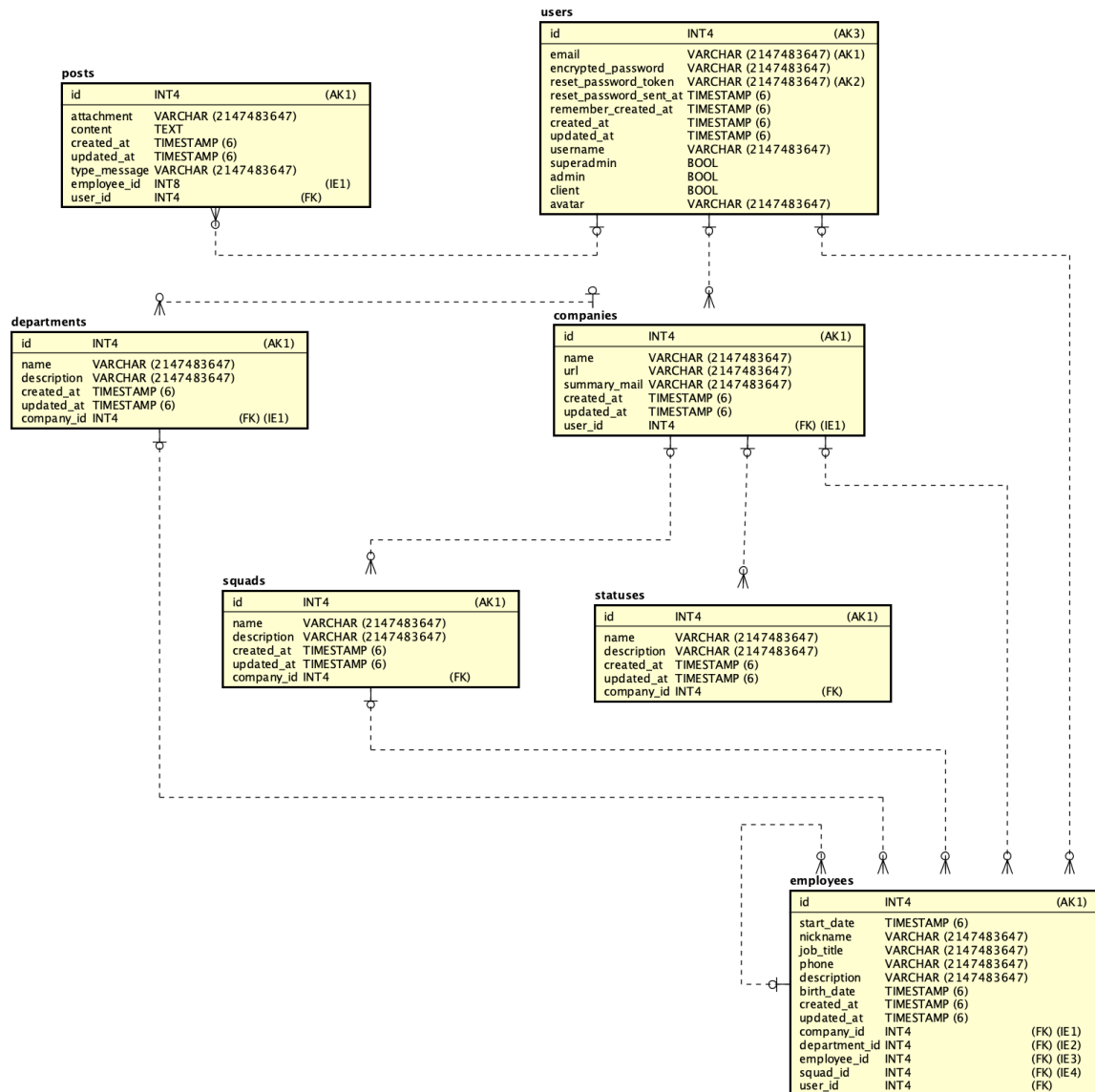
Não se aplica.

**Fluxo de Exceção**

Não se aplica.

## APÊNDICE E - DIAGRAMA DE ENTIDADE RELACIONAMENTO

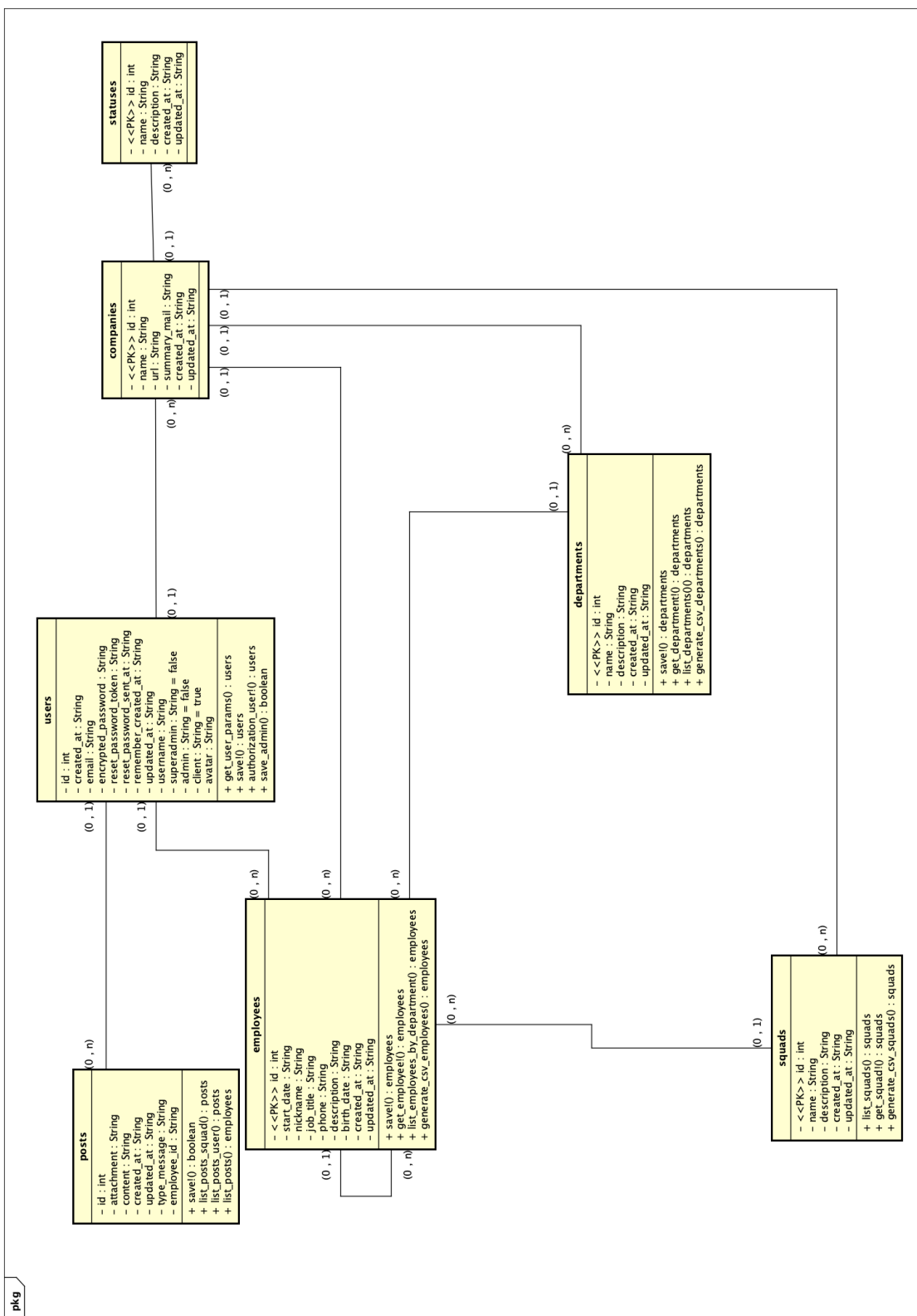
FIGURA 42 – Diagrama de entidade relacionamento



FONTE: Elaborado pelo autor

## APÊNDICE F - DIAGRAMA DE CLASSE DE NEGÓCIO

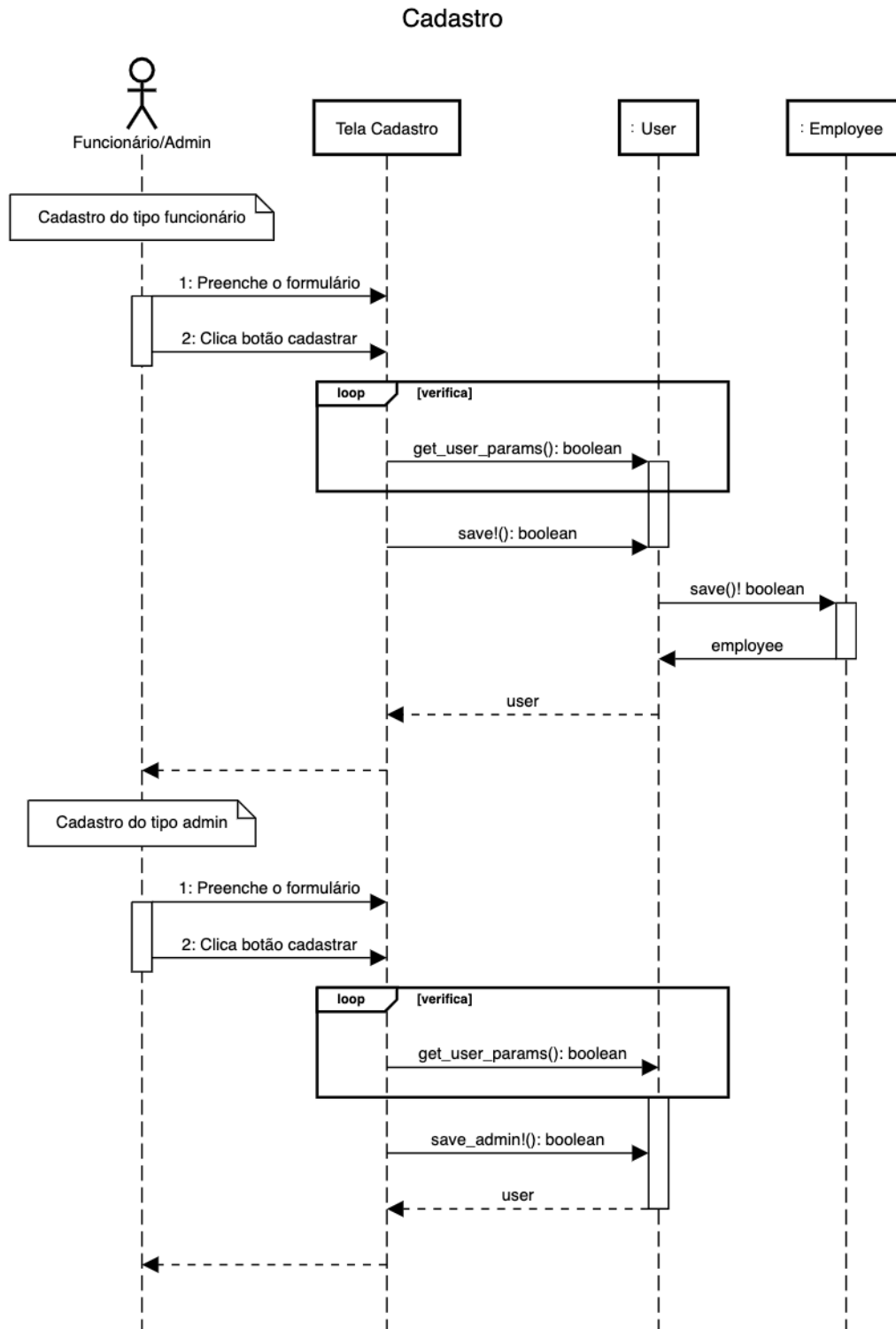
FIGURA 43 – Diagrama de classe



FONTE: Elaborado pelo autor

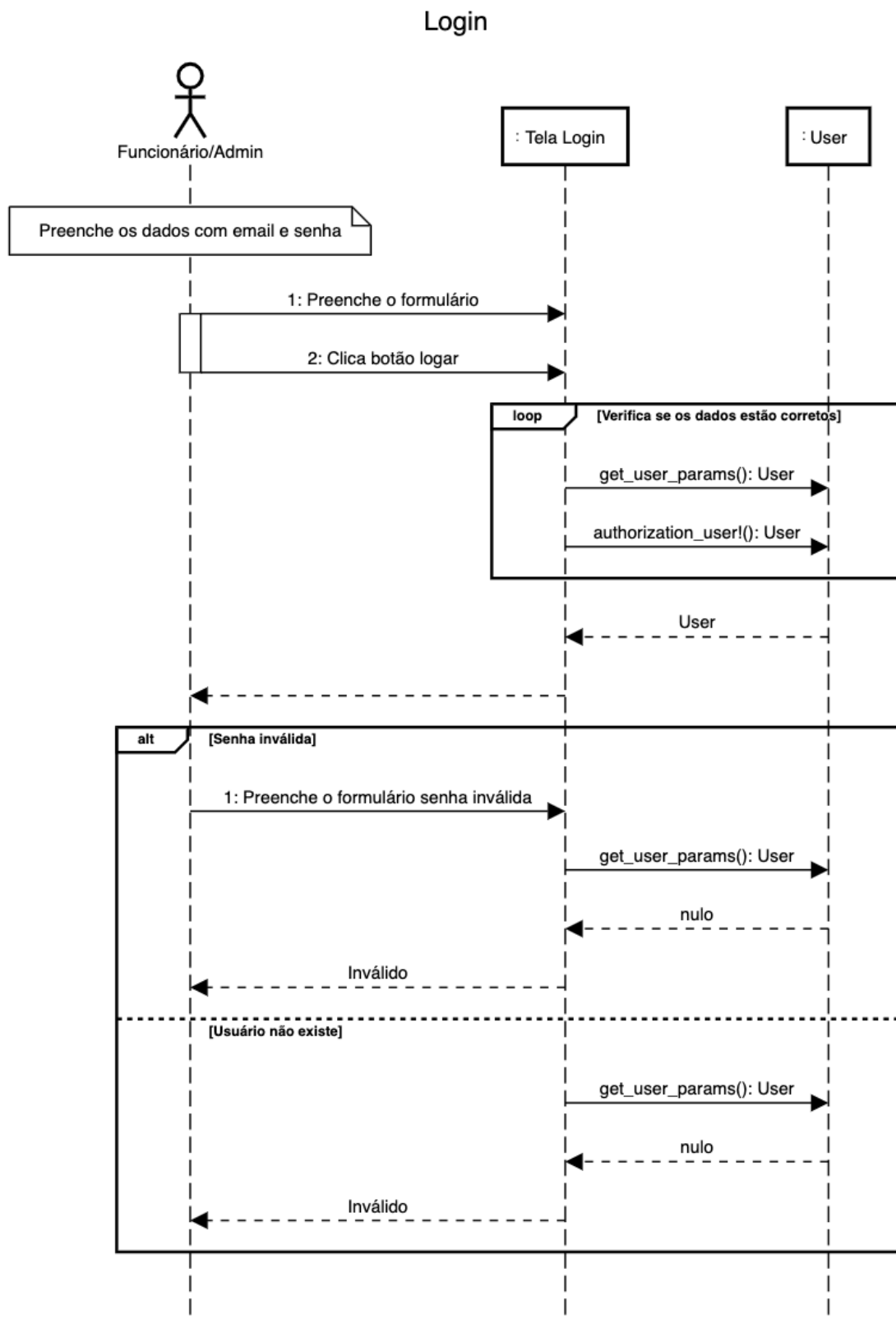
## APÊNDICE G - DIAGRAMAS DE SEQUÊNCIA

FIGURA 44 – Diagrama de sequência de cadastro



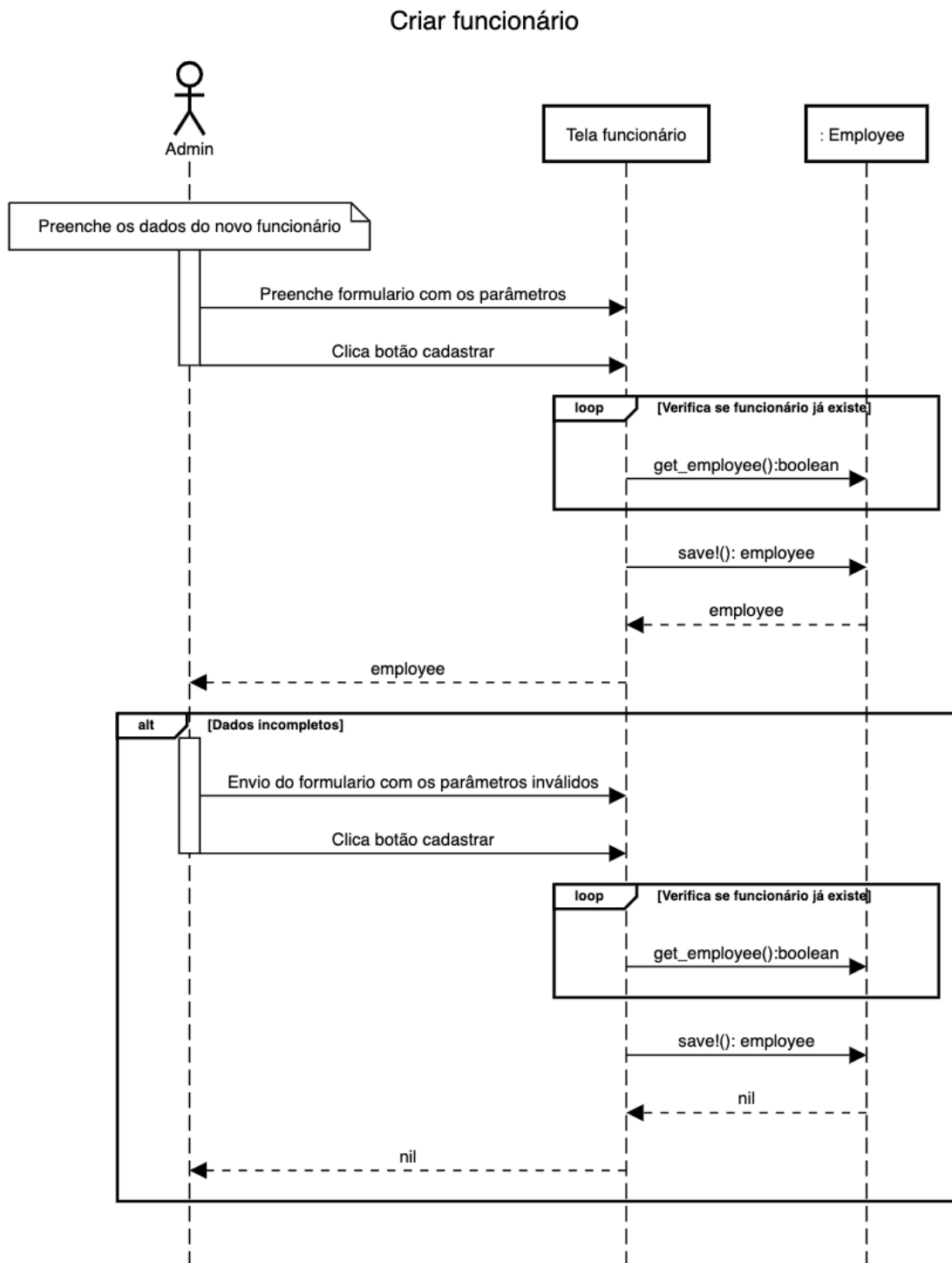
FONTE: Elaborado pelo autor

FIGURA 45 – Diagrama de sequência de login



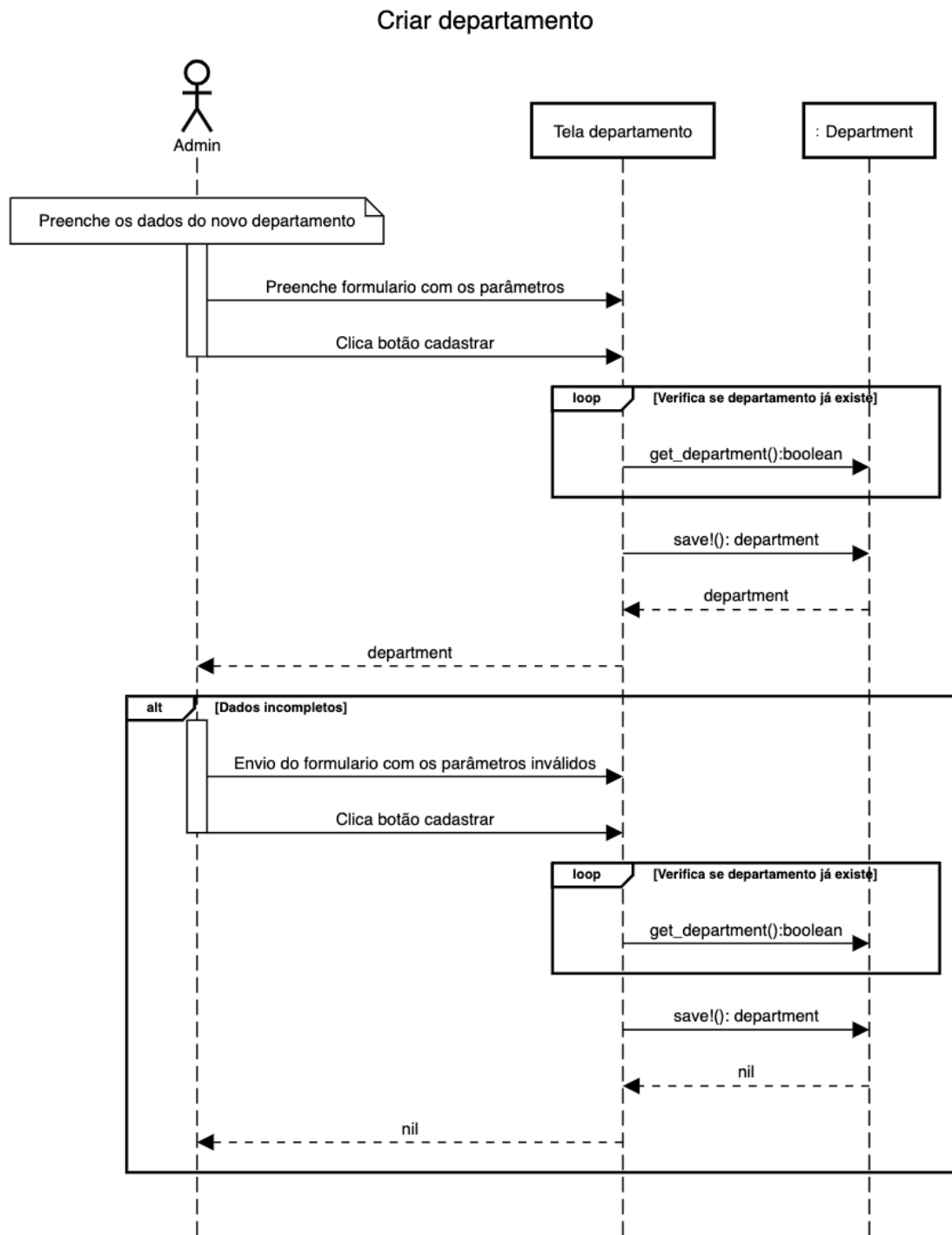
FONTE: Elaborado pelo autor

FIGURA 46 – Diagrama de sequência de cadastro de funcionário



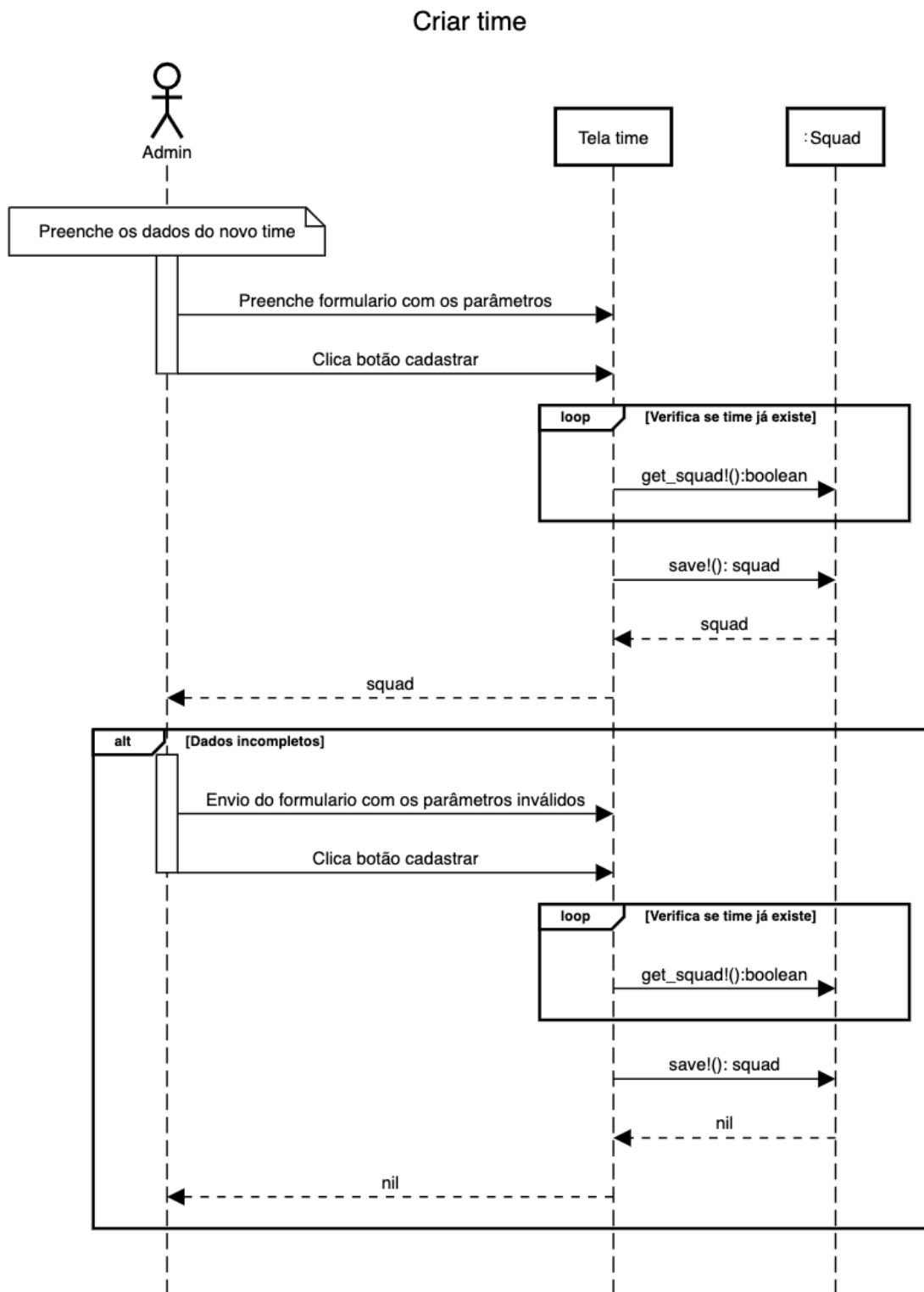
FONTE: Elaborado pelo autor

FIGURA 47 – Diagrama de sequência de cadastro de departamento



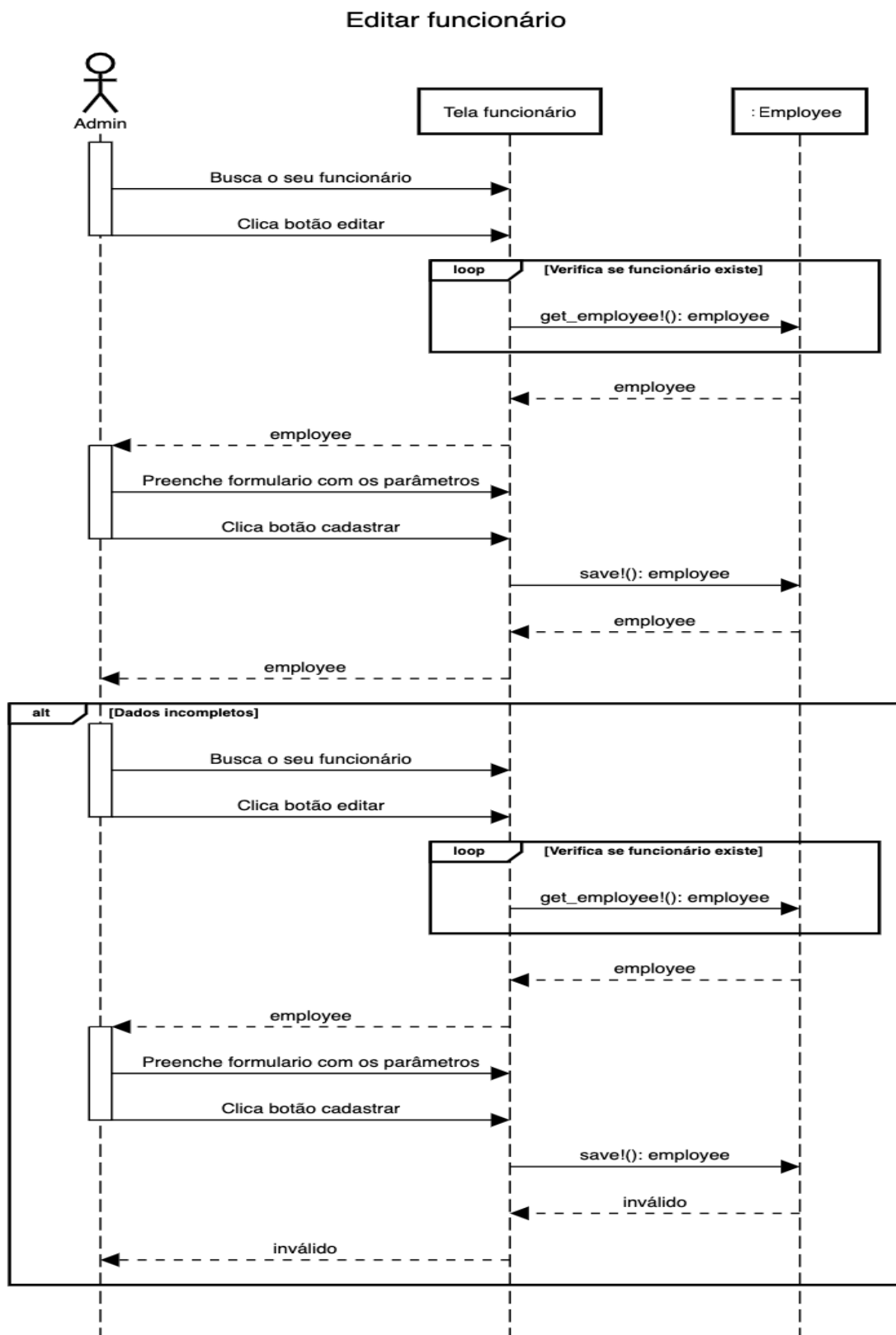
FONTE: Elaborado pelo autor

FIGURA 48 – Diagrama de sequência de cadastro de time



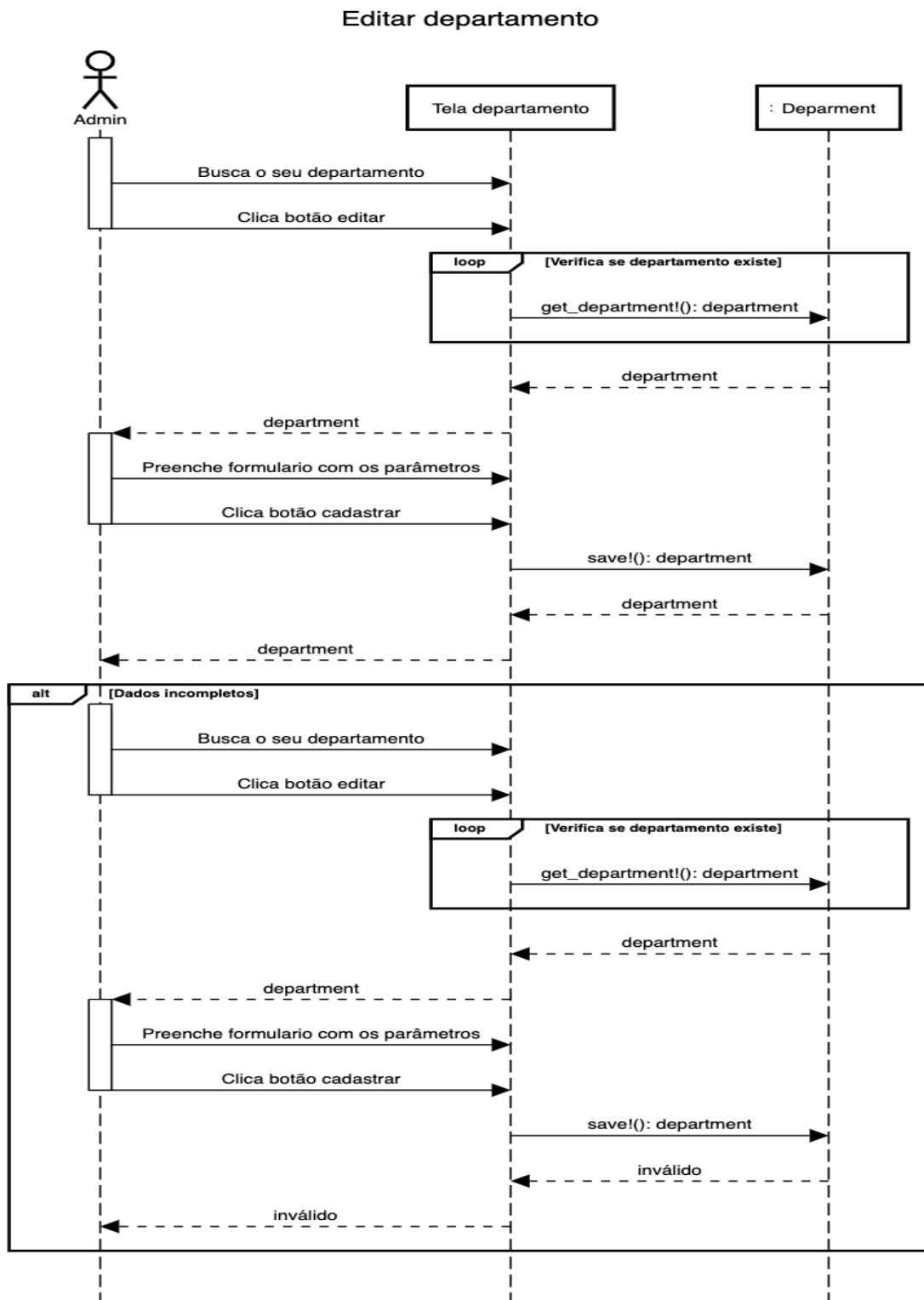
FONTE: Elaborado pelo autor

FIGURA 49 – Diagrama de sequência de editar funcionário



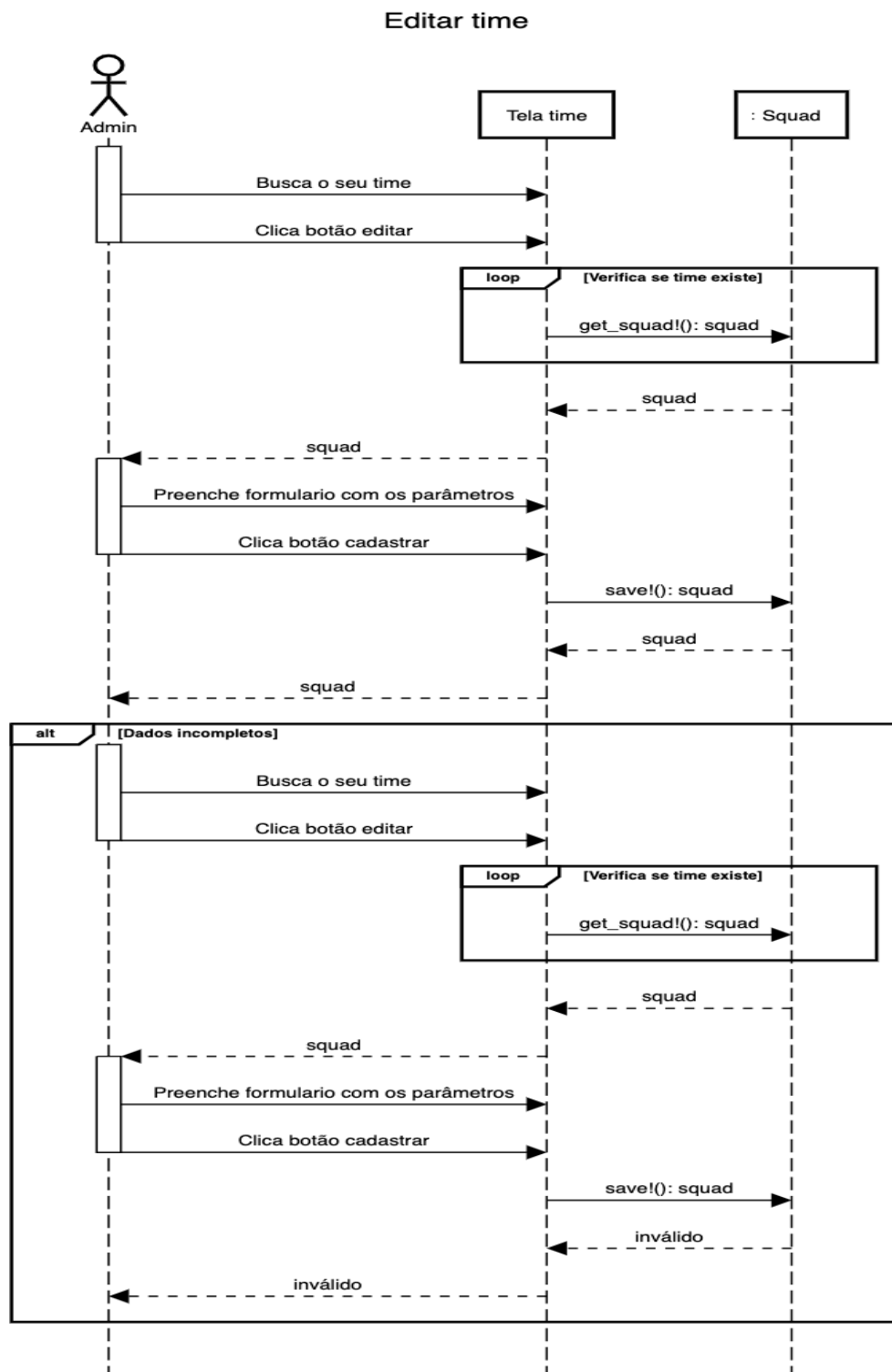
FONTE: Elaborado pelo autor

FIGURA 50 – Diagrama de sequência de editar departamento



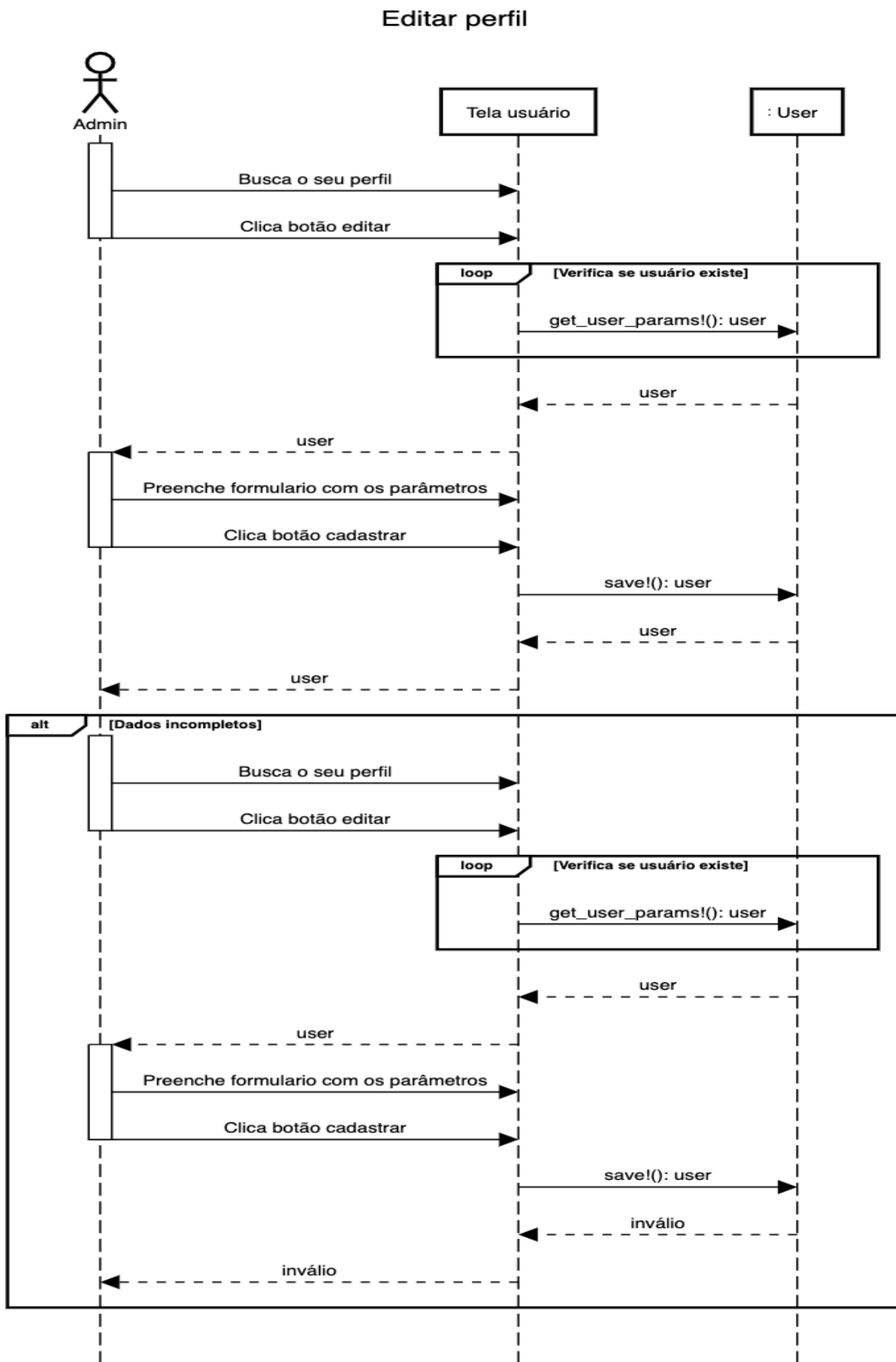
FONTE: Elaborado pelo autor

FIGURA 51 – Diagrama de sequência de editar time



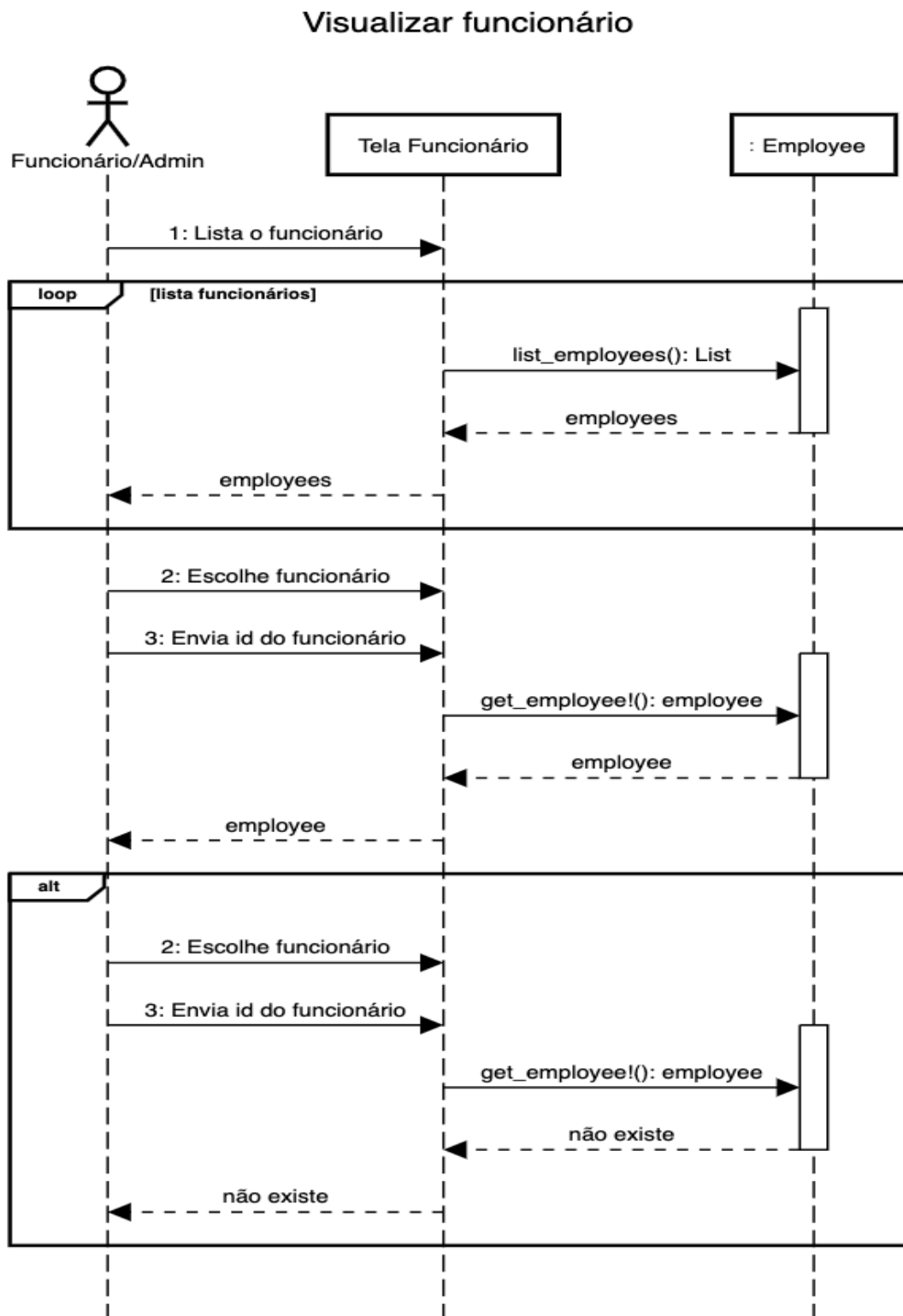
FONTE: Elaborado pelo autor

FIGURA 52 – Diagrama de sequência de editar perfil



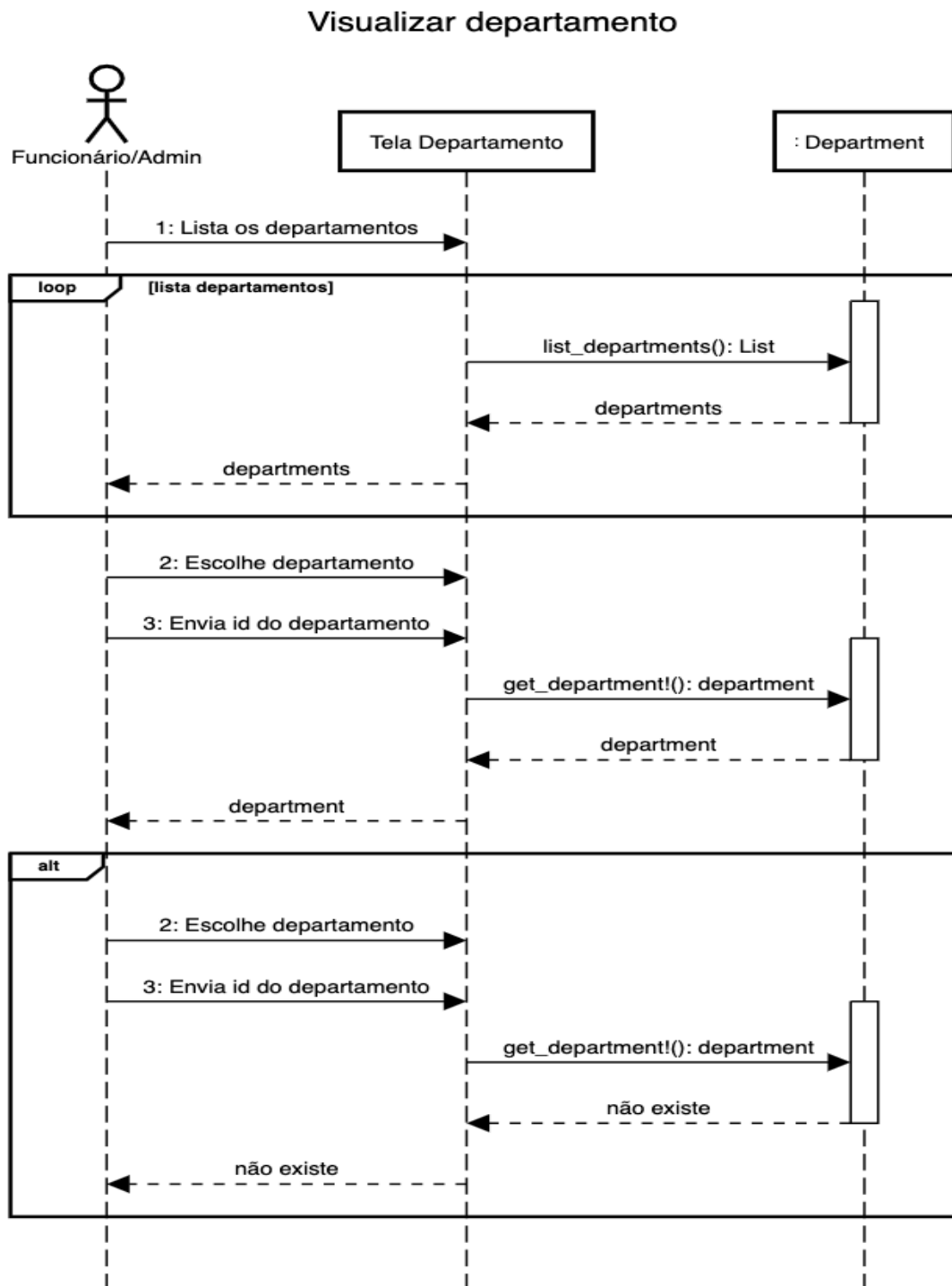
FONTE: Elaborado pelo autor

FIGURA 53 – Diagrama de sequência de visualizar funcionário



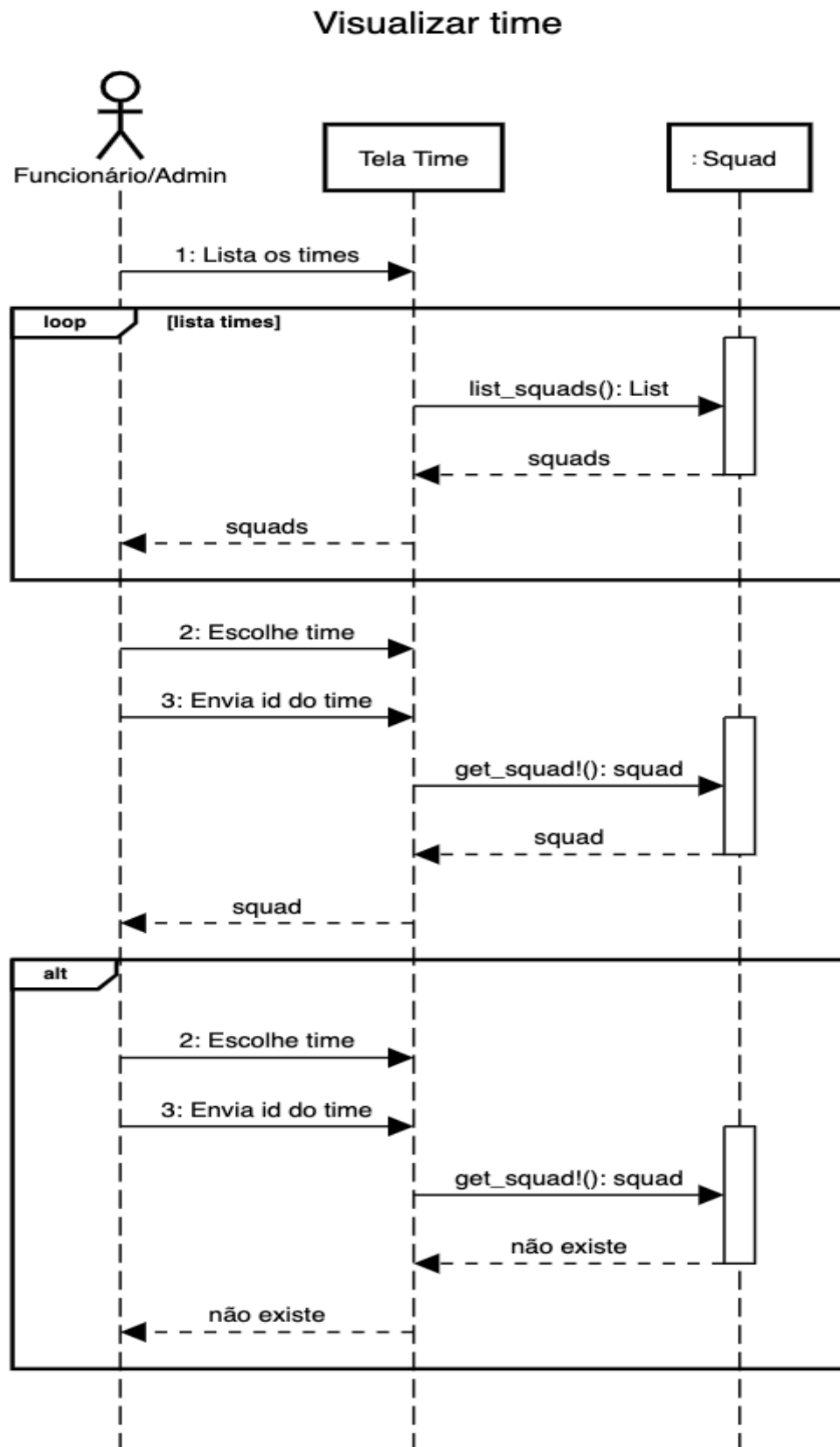
FONTE: Elaborado pelo autor

FIGURA 54 – Diagrama de sequência de visualizar departamento



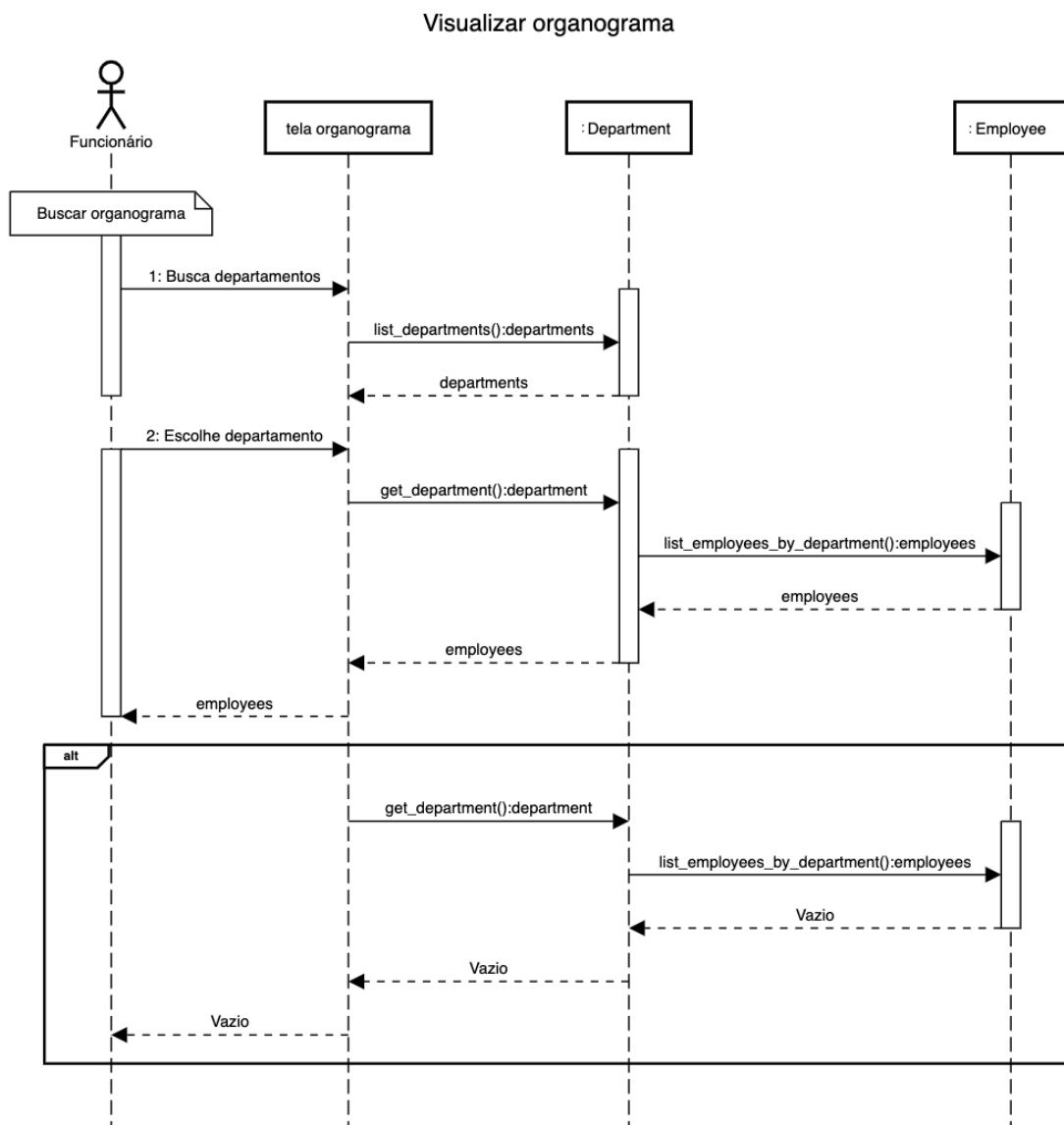
FONTE: Elaborado pelo autor

FIGURA 55 – Diagrama de sequência de visualizar time



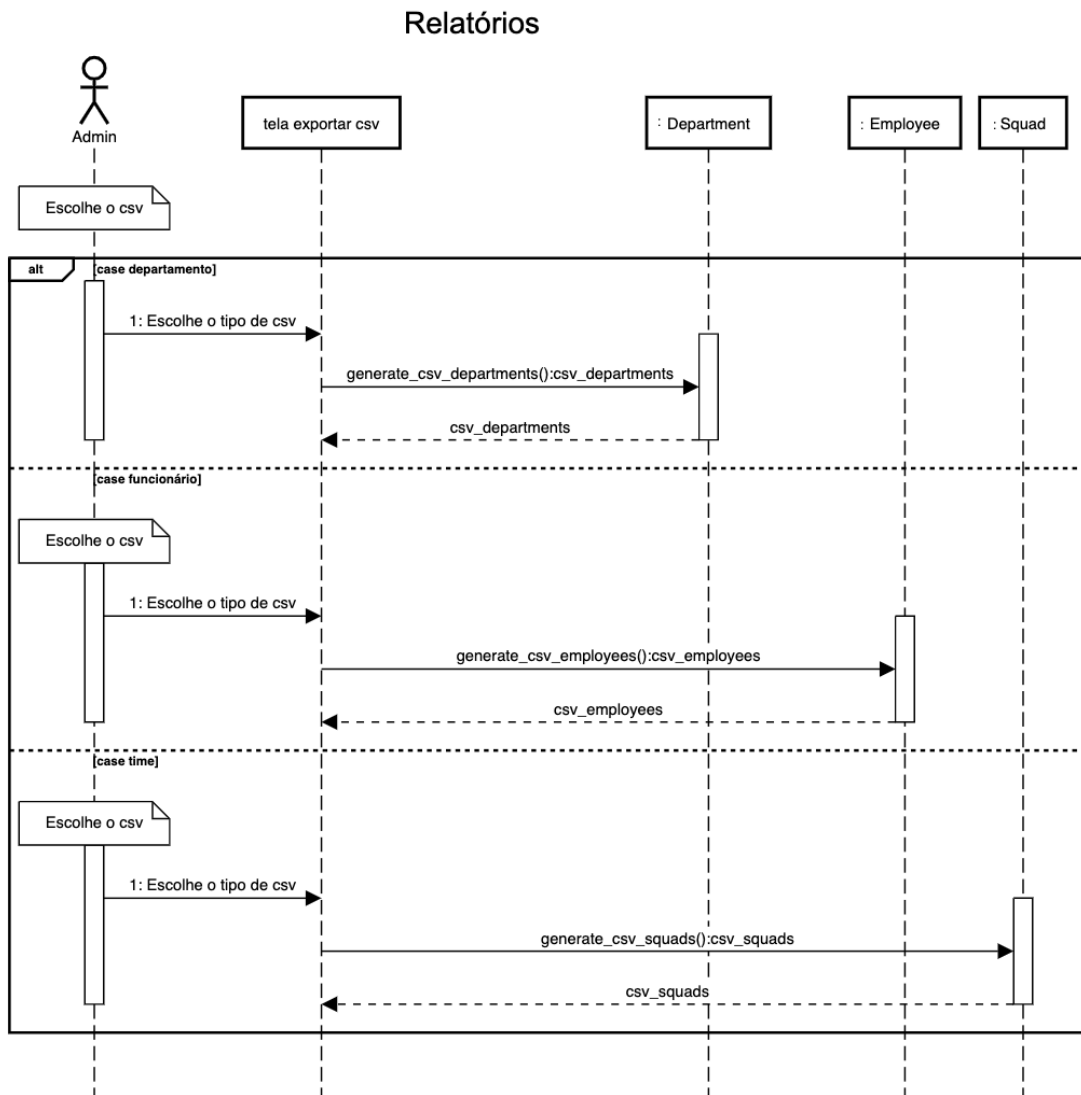
FONTE: Elaborado pelo autor

FIGURA 56 – Diagrama de sequência de visualizar organograma



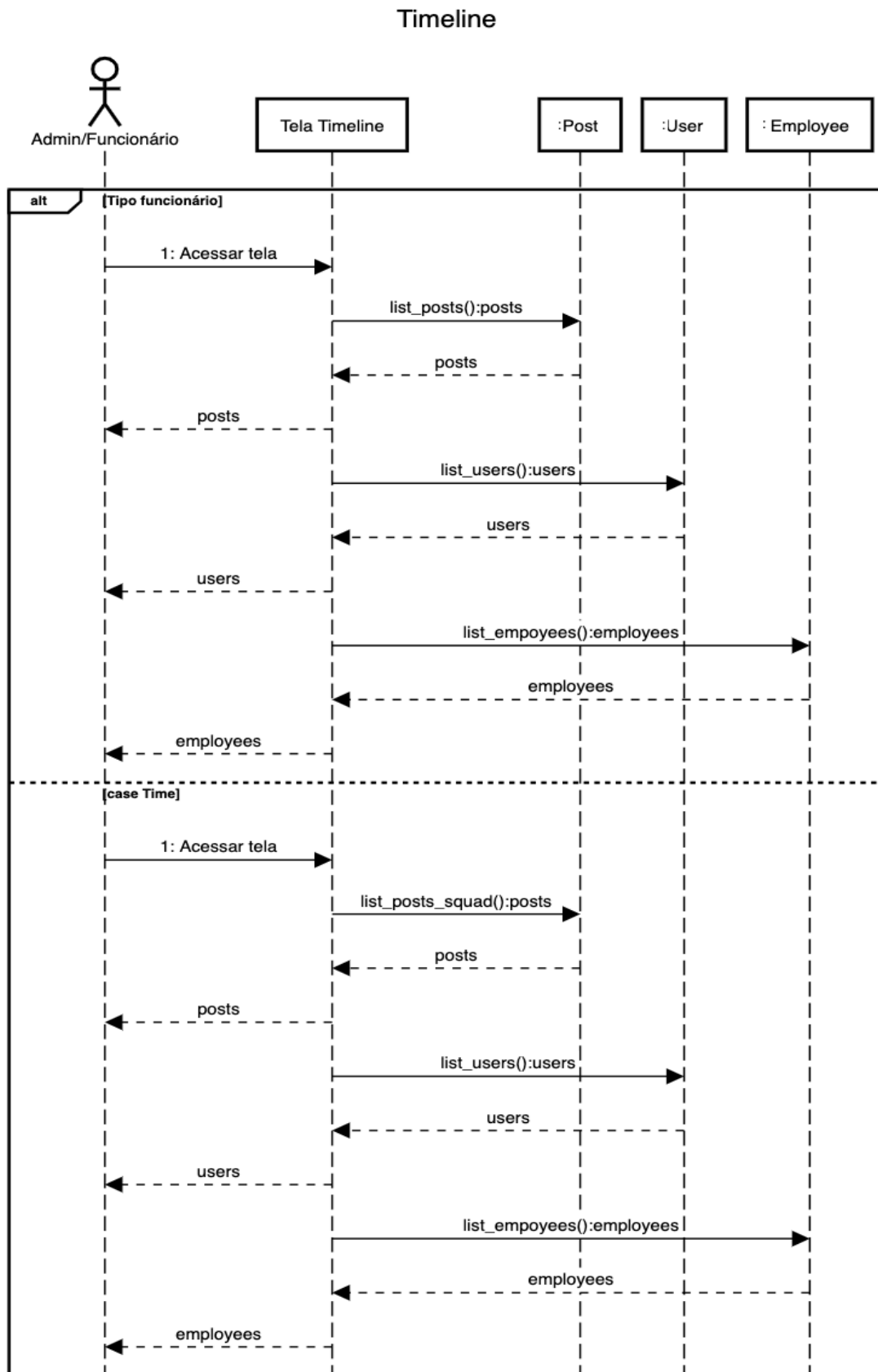
FONTE: Elaborado pelo autor

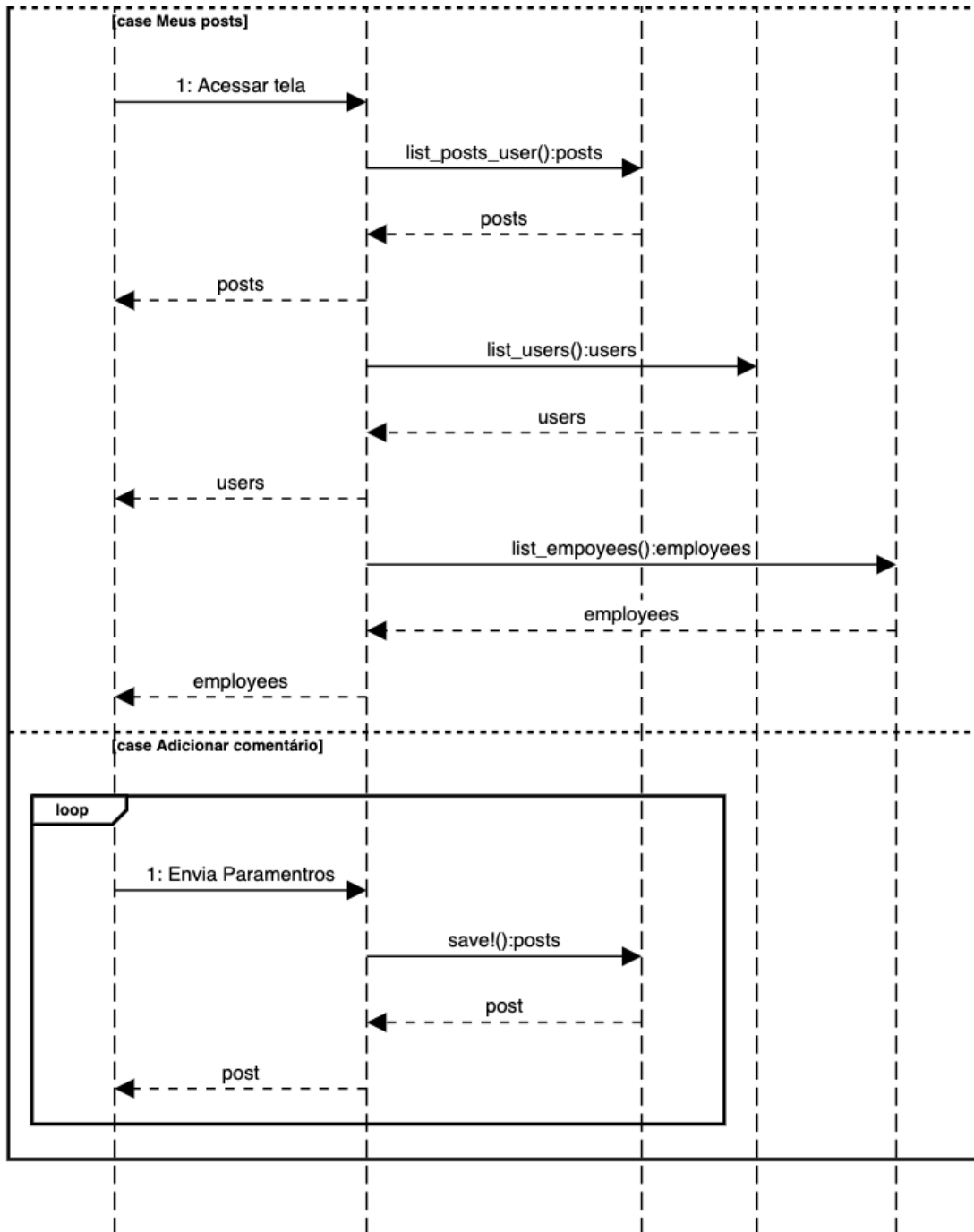
FIGURA 57 – Diagrama de sequência de exportar funcionário



FONTE: Elaborado pelo autor

FIGURA 58 – Diagrama de sequência da timeline





FONTE: Elaborado pelo autor