

**CENTRO ESTADUAL DE EDUCAÇÃO TECNOLÓGICA PAULA
SOUZA**

ETEC DA ZONA LESTE

Novotec Desenvolvimento de Sistemas

Desirée Constantino de Almeida Barboza

Giovana Marsigli Rodrigues

Isabelle Gomes de Souza Andrade

PILLTRACK: IoT para gerenciamento de medicamentos

São Paulo

2024

Desirée Constantino de Almeida Barboza

Giovana Marsigli Rodrigues

Isabelle Gomes de Souza Andrade

PILLTRACK: IoT para gerenciamento de medicamentos

Trabalho de conclusão de curso apresentado ao Curso Técnico em Desenvolvimento de Sistemas da ETEC Zona Leste, orientado pelo Prof. Esp. Jeferson Roberto de Lima, como requisito parcial para obtenção do título de técnico em Desenvolvimento de Sistemas.

São Paulo

2024

AGRADECIMENTOS

A conclusão deste Trabalho de Conclusão de Curso (TCC) é um grande marco em nossas vidas profissionais e acadêmicas, e é de extrema importância destacar que este projeto não é apenas um marco pessoal, mas também uma conquista compartilhada com aqueles que nos apoiaram desde o início.

Expressamos nossa profunda gratidão a Deus, por nos dar saúde e sabedoria necessárias para superar cada desafio ao longo da nossa trajetória.

Agradecemos de todo o coração às nossas famílias, pelo apoio constante e por sempre acreditarem em nosso potencial, sua paciência e encorajamento foram essenciais para chegarmos até aqui.

Aos nossos coordenadores, Jeferson Roberto de Lima e Rogério Bezerra, expressamos agradoecimento pela orientação e dedicação.

Por fim, agradecemos a todos que, de alguma forma, contribuíram para a realização deste trabalho. Cada gesto e palavra de apoio foram fundamentais para alcançarmos nosso objetivo.

Muito obrigada à todos!

“Nunca se satisfaça com o que você alcançou, porque o que você faz hoje vai determinar o que você alcançará amanhã.”

William Pollard

RESUMO

O projeto aborda a criação de um sistema completo e eficiente de gestão de medicamentos, visando assegurar a adesão a tratamentos e melhora de resultados na saúde dos pacientes, especialmente os idosos. O sistema incluirá uma caixa de medicamentos inteligente para notificações de horários de administração e controle de estoque, além de um aplicativo móvel para registro e acesso às informações sobre a rotina medicamentosa dos pacientes por médicos e cuidadores. A metodologia envolverá implementação e teste do sistema em ambiente controlado, seguido de avaliação qualitativa da experiência dos usuários e análise quantitativa dos dados de adesão ao tratamento e resultados de saúde. Espera-se que o sistema melhore significativamente a qualidade de vida dos usuários ao potencializar os resultados de saúde, com benefícios como melhor adesão ao tratamento medicamentoso, redução de erros na administração de medicamentos e melhoria dos indicadores de saúde dos pacientes. Em conclusão do estudo destacará a eficácia da abordagem tecnológica adotada na gestão de medicamentos e seus benefícios para a saúde pública, sublinhando a importância de soluções inovadoras no cuidado com pacientes crônicos, especialmente os idosos.

Palavras-Chave: Gestão, Medicamentos, IoT, Aplicativo, Tratamento, Monitoramento.

ABSTRACT

The project addresses the creation of a complete and efficient medication management system, aimed at ensuring adherence to treatment and improving the health outcomes of patients, especially the elderly. The system will include a smart medicine box for notifications of administration times and stock control, as well as a mobile app for recording and accessing information on patients' medication routine by doctors and caregivers. The methodology will involve implementing and testing the system in a controlled environment, followed by qualitative evaluation of the user experience and quantitative analysis of treatment adherence data and health outcomes. The system is expected to significantly improve users' quality of life by boosting health outcomes, with benefits such as better adherence to drug treatment, a reduction in medication administration errors and an improvement in patients' health indicators. In conclusion, the study will highlight the effectiveness of the technological approach adopted in medicines management and its benefits for public health, underlining the importance of innovative solutions in the care of chronic patients, especially the elderly.

Keywords: Management, Medicines, IoT, Application, Treatment, Monitoring.

LISTA ILUSTRAÇÕES

Figura 1 – Exemplo de estrutura básica de HTML	16
Figura 2 – Exemplo de estrutura HTML para um formulário.....	18
Figura 3 – Exemplo de estrutura HTML para um formulário projetado na web.....	19
Figura 4 – Exemplo de link CSS para ligação a página HTML.....	19
Figura 5 – Código CSS para estilização de botão HTML	20
Figura 6 – Código CSS para estilização de formulário HTML	21
Figura 7 – Resultado do formulário HTML estilizado em CSS	22
Figura 8 – Elemento “onsubmit” no HTML	23
Figura 9 – Exemplo de código JavaScript	23
Figura 10 – Resultado da programação JavaScript	24
Figura 11 – Exemplo de diagrama de Caso de Uso	25
Figura 12 – Exemplo de diagrama de Sequência.....	27
Figura 13 – Exemplo de diagrama de Atividade.....	28
Figura 14 – Exemplo de diagrama de Máquina de Estados	29
Figura 15 – Exemplo de wireframe de baixa fidelidade.....	30
Figura 16 – Exemplo de wireframe de alta fidelidade.....	31
Figura 17 – Tela de criação de projetos do Figma	32
Figura 18 – Logotipo Firebase.....	33
Figura 19 – Console Firebase	33
Figura 20 – Cloud Firestore.....	34
Figura 21 – Comando de criação projeto React Native	35
Figura 22 – Exemplo de código React Native	35
Figura 23 – Resultado da codificação React Native	37
Figura 24 – Exemplo de Modelagem 3D	38
Figura 25 – Exemplo de prototipagem 3D impressa e pintada.....	38
Figura 26 – ESP 32	39
Figura 27 – Jumpers	40
Figura 28 – Display LCD	41
Figura 29 - Motor de Passo + Módulo de Controle (Driver ULN2003).....	42
Figura 30 - Sensor Óptico TCRT5000	43
Figura 31 – Exemplo C++	44
Figura 32 – Diagrama de Caso de Uso do sistema PillTrack	45
Figura 33 - Diagrama de Atividade: Fazer Login	60
Figura 34 - Diagrama de Atividade: Visualizar relatórios.....	61
Figura 35 - Diagrama de Atividade: Detectar retirada de medicamentos	62
Figura 36 - Diagrama de Atividade: Registrar retirada de medicamentos	63
Figura 37 - Diagrama de Atividade: Notificar dose perdida	63
Figura 38 - Diagrama de Atividade: Relatar reação adversa.....	64
Figura 39 - Diagrama de Atividade: Visualizar medicamento	65
Figura 40 - Diagrama de Atividade: Manter medicamento	66
Figura 41 - Diagrama de Atividade: Cadastrar caixa.....	67
Figura 42 - Diagrama de Atividade: Cadastrar paciente.....	68

Figura 43 - Diagrama de Atividade: Vincular responsável.....	69
Figura 44 - Diagrama de Atividade: Vincular médico	70
Figura 45 - Diagrama de Atividade: Cadastrar responsável.....	71
Figura 46 - Diagrama de Atividade: Visualizar relatórios.....	72
Figura 47 - Diagrama de Atividade: Cadastrar médico.....	73
Figura 48- Diagrama de Sequência: Fazer Login.....	74
Figura 49 - Diagrama de Sequência: Visualizar relatórios	75
Figura 50 - Diagrama de Sequência: Detectar retirada de medicamentos	75
Figura 51 - Diagrama de Sequência: Registrar retirada de medicamentos	76
Figura 52 - Diagrama de Sequência: Notificar dose perdida	76
Figura 53 - Relatar reação adversa.....	77
Figura 54 - Diagrama de Sequência: Cadastrar caixa.....	77
Figura 55 - Diagrama de Sequência: Cadastrar paciente	77
Figura 56 - Diagrama de Sequência: Vincular responsável	78
Figura 57 - Diagrama de Sequência: Vincular médico	79
Figura 58 - Diagrama de Sequência: Cadastrar responsável.....	79
Figura 59 - Diagrama de Sequência: Cadastrar médico	79
Figura 60 - Diagrama de Máquina de Estados: Fazer Login	80
Figura 61 - Diagrama de Máquina de Estados: Visualizar relatórios sistema	80
Figura 62 - Diagrama de Máquina de Estados: Detectar retirada de medicamentos	80
Figura 63 - Diagrama de Máquina de Estados:Registrar retirada de medicamentos	80
Figura 64 - Diagrama de Máquina de Estados: Notificar dose perdida	81
Figura 65 - Diagrama de Máquina de Estados: Relatar reação adversa	81
Figura 66 - Diagrama de Máquina de Estados: Visualizar medicamento	81
Figura 67 - Diagrama de Máquina de Estados: Cadastrar caixa	81
Figura 68 - Diagrama de Máquina de Estados: Cadastrar paciente	81
Figura 69 - Diagrama de Máquina de Estados: Vincular médico.....	81
Figura 70 - Diagrama de Máquina de Estados: Cadastrar responsável	81
Figura 71 - Diagrama de Máquina de Estados: Visualizar relatórios	81
Figura 72 - Diagrama de Máquina de Estados: Cadastrar médico.....	81
Figura 73 - Wireframe de Baixa Fidelidade Site	81
Figura 74 - Wireframe de Alta Fidelidade Site.....	82
Figura 75 - Wireframe de Baixa e Alta Fidelidade: Login	84
Figura 76 - Wireframe de Baixa e Alta Fidelidade: Esqueceu a senha	84
Figura 77 - Wireframe de Baixa e Alta Fidelidade: Tipo de cadastro	85

LISTA DE TABELAS

Tabela 1 - Exemplo Documentação de Caso de Uso abertura de uma conta.....26

LISTA DE QUADROS

Quadro 1 - Documentação Caso de Uso (Fazer Login)	48
Quadro 2 - Documentação Caso de Uso (Detectar retirada de medicamentos) ..	Erro!
Indicador não definido.	
Quadro 3 - Documentação Caso de Uso (Registrar retirada de medicamentos)	50
Quadro 4 - Documentação Caso de Uso (Visualizar relatórios)	Erro! Indicador não definido.
Quadro 5 - Documentação Caso de Uso (Notificar dose perdida)	51
Quadro 6 - Documentação Caso de Uso (Relatar reação adversa)	51
Quadro 7 - Documentação Caso de Uso (Visualizar medicamento)	53
Quadro 8 - Documentação Caso de Uso (Manter medicamento).....	53
Quadro 9 - Documentação Caso de Uso (Cadastrar caixa)	55
Quadro 10 - Documentação Caso de Uso (Cadastrar paciente)	55
Quadro 11 - Documentação Caso de Uso (Vincular responsável)	56
Quadro 12 - Documentação Caso de Uso (Visualizar médico)	57
Quadro 13 - Documentação Caso de Uso (Cadastrar responsável)	58
Quadro 14 - Documentação Caso de Uso (Visualizar relatórios)	Erro! Indicador não definido.
Quadro 15 - Documentação Caso de Uso (Cadastrar médico)	59

LISTA DE ABREVIATURAS E SIGLAS

Cascading Style Sheet (CSS).

Duas dimensões (2D).

Eletronic Stability Program (ESP).

Ground (GND).

HyperText Markup Language (HTML).

Inter-Intregrated Circuit (I2C).

Interface do usuário (IU).

Integrated Development Environment (IDE).

Internet of Things (IoT).

JavaScript (JS).

Kilobyte (KB).

Light Emitting Diode (LED).

Liquid Cristal Display (LCD).

Milliampere (mA).

Model View Controller (MVC).

Node Package Maneger (NPM).

Not Only SQL (NoSQL).

Pulse Position Modulation (PPM).

Pulse-Width Modulation (PWM).

Real Time Clock (RTC).

Red, Green, Blue (RGB).

Sistema de Gerenciamento de Banco de Dados (SGBD).

Static Random-Access Memory (SRAM).

Structured Query Language (SQL).

Television (TV).

Tridimensional (3D).

Unified Modeling Language (UML).

Volt (V).

Voltage Common Collector (VCC).

Wireless Fidelity (Wi-Fi).

SUMÁRIO

1.	INTRODUÇÃO	13
2.	REFERENCIAL TEÓRICO	15
2.1	Dificuldades na eficácia da gestão de medicamentos	15
2.2	HTML.....	15
2.3	CSS	19
2.4	JavaScript.....	23
2.5	UML.....	25
2.5.1	UML - Diagrama de Casos de Uso.....	25
2.5.2	UML - Documentação Caso de Uso	26
2.5.3	UML - Diagrama de Sequência	27
2.5.4	UML - Diagrama de Atividade.....	28
2.5.5	UML - Diagrama de Máquina de Estados.....	29
2.6	Wireframe	30
2.7	Figma	32
2.8	Banco de dados.....	32
2.8	Firebase.....	32
2.9	Firestore	33
2.10	Node.js & NPM	34
2.11	React	35
2.12	React Native	35
2.12.1	Criação de um aplicativo em React Native	35
2.13	IoT	37
2.13.1	Modelagem 3D	38
2.13.2	ESP 32	38
2.13.3	Jumpers.....	39
2.13.4	Display LCD	Erro! Indicador não definido.
2.13.5	Motor de Passo + Módulo de Controle (Driver ULN2003)	42
2.13.6	Sensor Óptico TCRT5000	43
2.13.7	C++.....	44
3.	DESENVOLVIMENTO.....	45
3.1	Diagrama de Casos de Uso.....	45
3.2	Documentação Caso de Uso.....	46

3.3	Diagrama de Atividade	60
3.4	Diagrama de Sequência	74
3.5	Diagrama de Máquina de Estados	80
3.6	Montagem do circuito do dispositivo.....	81
3.7	Prototipação das páginas da aplicação	81
4.	CONSIDERAÇÕES FINAIS.....	92
	REFERÊNCIAS.....	94

1. INTRODUÇÃO

Segundo a Eurofarma (2018), é crucial seguir a prescrição médica nos horários corretos, independentemente da idade. O médico adapta o tratamento com base no corpo individual, considerando seus hábitos e rotinas. Em consideração a isso, este projeto foca na seriedade da gestão de medicamentos tornando uma solução tecnológica proposta visando organizar e monitorar o tratamento adequado de cada medicamento para o usuário.

O presente estudo justifica-se pela importância de garantir a adesão correta ao uso de medicamentos, considerando os desafios enfrentados por pacientes ao seguir prescrições nos horários corretos. A tecnologia tem potencial para minimizar esquecimentos e melhorar a gestão medicamentosa, tornando-se uma solução relevante para otimizar tratamentos e reduzir complicações decorrentes da má administração de medicamentos. Assim, a tecnologia evolui para atender às necessidades da sociedade, acompanhando sua evolução e isso nos leva ao surgimento da Internet das Coisas (*IoT*) com grande potencial. (ALBERTIN, 2017).

Atualmente, a Internet das Coisas, de forma rápida, se torna um cenário da realidade, pois ao olhar ao redor podemos ver que nossos dispositivos ficam mais inteligentes a cada dia (SANTOS, 2018). Com essa tecnologia, é possível automatizar tarefas demoradas ou esquecidas. Na área da saúde, um sistema *IoT* se torna uma ferramenta valiosa para pacientes e profissionais, melhorando a compreensão e gerenciamento de atividades.

A problemática gira em torno da dificuldade que muitos pacientes, especialmente idosos, têm em seguir rigorosamente os tratamentos prescritos, seja por esquecimento ou falta de disciplina, o que compromete a eficácia dos medicamentos e pode agravar condições de saúde. Com isso, aliada ao conhecimento médico, um *IoT* pode atuar na medicina preventiva, melhorando a qualidade de vida e a satisfação dos pacientes. (MASSOLA; PINTO, 2018). Conforme dados referenciados pela Sociedade de Cardiologia do Estado de São Paulo. (KATZ; FEITOSA; PINTO; FELIX, BORTOLOTTO, 2020) “estima-se que metade dos 3,2 bilhões de prescrições

médicas realizadas anualmente nos EUA não são seguidas corretamente". Desta forma a criação de um sistema auxiliador faz grande diferença na recuperação e evita o esquecimento causador de agravamentos de condições que requerem tratamento.

Assim, a utilização do sistema pode melhorar significativamente a adesão medicamentosa com o acesso fácil às informações dos medicamentos e notificações automáticas, os usuários são incentivados a seguir as orientações médicas, reduzindo o esquecimento e aumentando a eficácia dos tratamentos a longo prazo.

O acesso ao sistema, permite uma análise mais específica de responsáveis e médicos sobre a adesão ao tratamento pelos pacientes, assim como no caso de pacientes idosos, que são mais propensos a esquecer doses e horários, a criação da aplicação se torna ainda mais positiva, visto que tende a solucionar esses problemas, contribuindo para a melhoria na qualidade da assistência, assegurando a administração correta dos medicamentos, tornando assim o projeto único e inovador com uma automação na problemática abordada.

Neste artigo abordaremos para o desenvolvimento do aplicativo React Native (FACEBOOK, 2015); Firebase (GOOGLE, 2014) para fins de armazenamento de dados; C++ para a construção da programação do sistema (STROUSTRUP, 1985); UML 2 (GUEDES, 2018) para a documentação de diagramas do projeto.

2. REFERENCIAL TEÓRICO

Neste capítulo serão abordados o embasamento teórico e as tecnologias usadas, assim como seus conceitos e aplicabilidades para construção do nosso projeto de gerenciamento de medicamentos PillTrack.

2.1 Dificuldades na eficácia da gestão de medicamentos

Um dos princípios fundamentais para uma boa recuperação e tratamento é a comunicação direta entre pacientes e médicos, permitindo que juntos cheguem aos melhores resultados no combate às doenças. Pouco tempo de consulta e má comunicação aumentam a propensão ao distanciamento e à hiper formalidade no diálogo, o que contribui para esses problemas. (PIXEL DIAGNÓSTICO, 2020). Ou seja, um histórico médico detalhado pode funcionar como um mapa para prevenir futuros problemas de saúde. (UNIMED CAMPINAS).

Dessa forma, é possível utilizar recursos que facilitem esse acompanhamento. Para ter mais tempo de escutar e acompanhar seus pacientes, os médicos podem contar com tecnologias de software para gerenciamento das informações dos pacientes em um só local.

Uma pesquisa realizada por alunos da Universidade de São Paulo (USP) com profissionais da saúde e as reações mais citadas pelos profissionais incluíram termos como “gravidade”, “piora”, “preocupação”, “prejudicial” e “morte”, refletindo os efeitos da má adesão ao tratamento tanto para os pacientes quanto para os profissionais. (FERREIRA; CAMPOS, 2023).

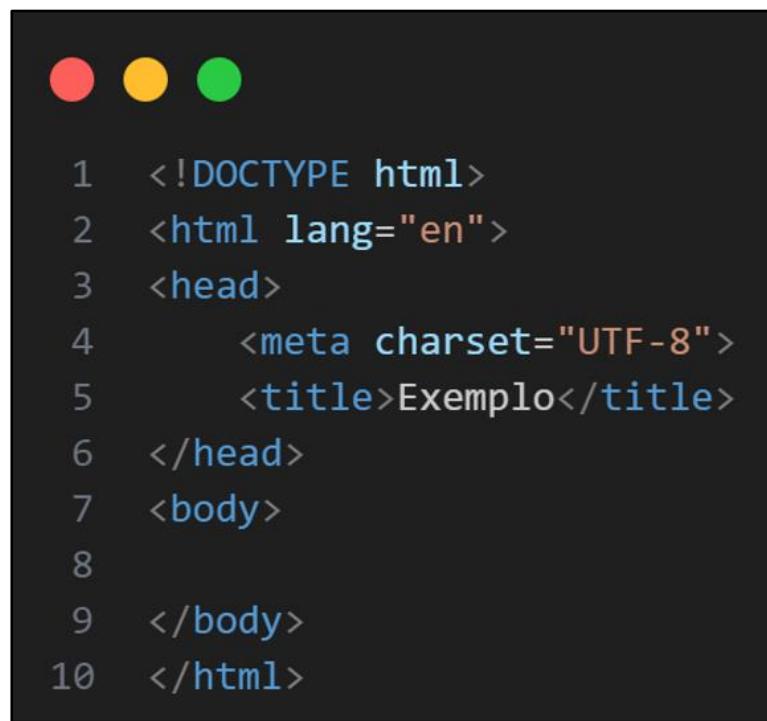
Por isso, foi pensado na aplicação PillTrack, que além de ajudar os pacientes a se medicar corretamente de acordo com a prescrição médica, ainda será possível a interação direta entre médico e paciente, através de relatórios detalhados disponibilizados pelo aplicativo que terá acesso às informações direto da caixa. Para entender como funcionará o projeto PillTrack, continue lendo o artigo, pois nosso sistema será mais bem detalhado nos próximos capítulos.

2.2 HTML

De acordo com Maurício Samy Silva (2008), HTML é a sigla para *HyperText Markup Language*, que, em português significa linguagem para marcação de hipertexto.

Os elementos HTML ou chamados de *tags* HTML, são utilizados para informar ao navegador que tipo de estrutura está sendo construída, podendo ser títulos, parágrafos, imagens, *links*, entre outros. (Alura, 2018). Segue abaixo um exemplo de estrutura básica:

Figura 1 – Exemplo de estrutura básica de HTML



```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>Exemplo</title>
6  </head>
7  <body>
8
9  </body>
10 </html>
```

Fonte: Dos próprios autores, 2024.

Para melhor compreensão, explicaremos as *tags* referente a imagem acima:

- *Doctype*: “`<!DOCTYPE html>`” essa *tag* informa ao navegador qual é o tipo de documento e a versão em HTML.
- *Html*: “`<html lang = “pt-br”></html>`” a *tag* essencial para a construção do código em HTML, serve como base e engloba todos os elementos na construção de documentos. Com o LANG, é possível indicar o idioma do texto representado.

- *Head*: “<head></head>” a tag HEAD é onde fica toda a parte inteligente da página, contendo as informações sobre o documento e o conteúdo ali publicado.
- Metatag Charset: “<meta charset = “UTF – 8”>” tag responsável por chamar qual tabela de caracteres a página está utilizando.
- *Title*: “<title></title>” tag usada para dar referência ao título da página que será inserido nas guias do navegador.
- *Body*: “<body></body>” a tag responsável pela inserção de todo o corpo do código, estabelecendo o conteúdo visual da página, no qual o usuário vê e interage.

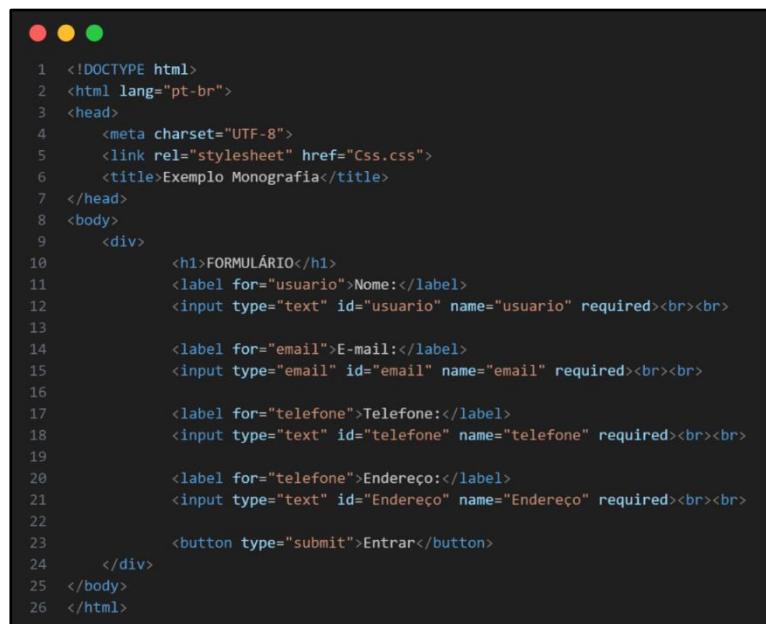
Além da estrutura básica do HTML, para a construção de uma página é necessário mais do que as tags principais como:

- *Div*: “<div></div>”, define uma divisão da página, funciona como um container para conteúdo fluxo.
- *Br*:
, não necessita de fechamento, ela executa a função de quebra de linha.
- H1, H2, H3, H4, H5, H6: “<h1>”, “<h2>”, “<h3>”, “<h4>”, “<h5>”, “<h6>”, possuem valor semântico, variando entre seis níveis hierárquicos, a página contendo apenas H1 não seria apenas para o título principal, H2 e H3 para títulos de seções e a H4, H5 e a H6 para subtítulos.
- *P*: “<p>”, principal tag usada para composição de um parágrafo.
- *Span*: “”, costumam ser utilizadas apenas para pequenas informações, como legendas de um formulário, legendas de uma imagem, entre outros.
- *B*: “”, transforma o conteúdo em negrito.
- *I*: “<i></i>”, transforma o conteúdo em itálico.
- *Hr*: “<hr>”, não necessita de fechamento, ela forma uma linha horizontal.
- *Button*: “<button>”, utilizada para inserir interatividade em uma página ou formulário e tem como função executar uma determinada ação ao receber um clique do usuário.
- *Form*: “<form></form>”, usada para criar campos de entrada interativos, nos quais as pessoas usuárias podem inserir dados.

- *Footer*: “<footer></footer>”, define um rodapé para a página, geralmente são utilizados no final da página.
- *Section*: “<section></section>”, define uma sessão para sua página.
- *Nav*: “<nav></nav>”, define um conteúdo de navegação, portanto é muito utilizado em criação de menus.
- *Header*: “<header></header>”
- *Main*: “<main></main>”, representa o conteúdo principal do seu corpo, ou seja, o conteúdo relacionado diretamente com o tópico central da página.
- *Aside*: “<aside></aside>”, representa uma seção de uma página cujo conteúdo é relacionado ao seu entorno, que pode ser considerado separado do conteúdo.
- *Img*: “”, usada para incluir uma imagem ao seu texto.
- *Video*: “<video>”, usada para indicar inserção de vídeo.
- *Input*: “<input>”, possui o atributo *type*, que varia entre diversos tipos, como “<input type = “text”>”, que é um campo que receberá qualquer caractere, já o “<input type= “email”>”, define um campo que receberá caracteres e verificará se o mesmo consiste em um e-mail válido.

Abaixo temos um exemplo em HTML que provirá de um formulário simples para melhor compreensão de algumas *tags melhor* explicadas acima:

Figura 2 – Exemplo de estrutura HTML para um formulário



```

1 <!DOCTYPE html>
2 <html lang="pt-br">
3 <head>
4   <meta charset="UTF-8">
5   <link rel="stylesheet" href="Css.css">
6   <title>Exemplo Monografia</title>
7 </head>
8 <body>
9   <div>
10    <h1>FORMULÁRIO</h1>
11    <label for="usuario">Nome:</label>
12    <input type="text" id="usuario" name="usuario" required><br><br>
13
14    <label for="email">E-mail:</label>
15    <input type="email" id="email" name="email" required><br><br>
16
17    <label for="telefone">Telefone:</label>
18    <input type="text" id="telefone" name="telefone" required><br><br>
19
20    <label for="Endereço">Endereço:</label>
21    <input type="text" id="Endereço" name="Endereço" required><br><br>
22
23    <button type="submit">Entrar</button>
24  </div>
25 </body>
26 </html>

```

Fonte: Dos próprios autores, 2024.

Temos agora reprodução gráfica do exemplo HTML projetado acima:

Figura 3 – Exemplo de estrutura HTML para um formulário projetado na web

The image shows a web-based form enclosed in a light gray border. At the top, the word "FORMULÁRIO" is written in large, bold, dark blue capital letters. Below it are four input fields, each preceded by a label in brown text: "Nome:" followed by a text input field; "E-mail:" followed by a text input field; "Telefone:" followed by a text input field; and "Endereço:" followed by a text input field. At the bottom left of the form is a blue rectangular button with the word "Entrar" in white.

Fonte: Dos próprios autores, 2024.

2.3 CSS

Abreviação para o termo *Cascading Style Sheet* que em português possui a tradução como folha de estilo em cascata, desta forma cria-se assim a sigla CSS. Surgiu em 1994 como uma proposta de Tim Berners-Lee que havia desenvolvido um navegador com um conjunto de funcionalidades de estilização padrão, sem definição de estilo feita pelo usuário ou autor. (j, 2011). Criado exclusivamente para uma linguagem de marcação e estruturação, o CSS faz o que HTML não é capaz de proporcionar, como todo o visual de um documento, cabe a essa linguagem a função de estilização do mesmo. (SILVA, 2018).

Possui um funcionamento prático, a formatação feita de seus estilos se dá por fora do documento original HTML, sendo interligado por meio de *link* criado pelo desenvolvedor, que se resulta em uma página formatada conforme a definição feita pelo arquivo CSS. (JOBSTRAIBIZER, [s.d.]).

Figura 4 – Exemplo de *link* CSS para ligação a página HTML



```

1  <!DOCTYPE html>
2  <html lang="pt-br">
3  <head>
4      <meta charset="UTF-8">
5      <link rel="stylesheet" href="Css.css">
6      <title>Exemplo Monografia</title>

```

Fonte: Dos próprios autores, 2024.

Como exemplo a ser explicado iremos usar a estilização de um botão que passara por seletores e propriedades, conforme dito por FERREIRA (2012), os seletores são estruturas para determinar quais elementos serão formatados e as propriedades são características desses elementos que pretende formatar.

Assim, conforme declarado por MAZZA (2014), existem seletores e propriedades básicas para a estilização de um botão como apresentado abaixo:

Figura 5 – Código CSS para estilização de botão HTML



```

1  button {
2      padding: 15px;
3      align-self: center;
4      width: 50%;
5      border: solid 1px;
6      border-radius: 30px;
7      background-color: #E1D9CE;
8      color: #000000;
9      font-weight: 700;
10     cursor: pointer;
11     text-transform: uppercase;
12 }

```

Fonte: Dos próprios autores, 2024.

- Seletor: O seletor é *button*, este seletor aplica essas regras a todos os botões no documento HTML a partir das propriedades declaradas.

- *padding*: Cria espaço dentro do botão entre o conteúdo e as bordas do botão.
- *align-self*: Define o alinhamento do botão.
- *width*: Define a largura do botão.
- *border*: Remove a borda do botão. Isso significa que o botão não terá uma borda visível.
- *border-radius*: Define o raio de curvatura das bordas. Isso cria cantos arredondados para o botão.
- *background-color*: Define a cor de fundo do botão.
- *color*: Define a cor do texto do botão.
- *font-weight*: Usada para definição da espessura da fonte
- *cursor*: Define o cursor do mouse ao passar sobre o botão como um ícone de ponteiro, indicando que o botão é clicável.
- *Text-transform*: Usado para especificar como o texto deve ser transformado em relação à sua capitalização e caracteres.

Logo, existem algumas maneiras de se criar uma página formatada em CSS com elementos básicos, sendo possível criar estilizações mais significativas a um documento HTML, como podemos ver na figura abaixo:

Figura 6 – Código CSS para estilização de formulário HTML

```
body {
    font-family: Arial, sans-serif;
    background-color: #F0F2EE;
    display: flex;
    height: 100vh;
    justify-content: center;
    align-items: center;
    margin: 0;
    padding: 0;
    box-sizing: border-box;
}

form {
    width: 400px;
    background-color: white;
    padding: 50px;
    border-radius: 30px;
    display: flex;
    flex-direction: column;
}

input[type="text"],
input[type="email"],
input[type="password"],

textarea {
    width: 100%;
    padding: 10px;
    margin-bottom: 15px;
    border: 3px solid #E1D9CE;
    border-radius: 30px;
    box-sizing: border-box;
    justify-content: center;
    flex-direction: column;
}

input[type="submit"] {
    width: 100%;
    padding: 10px;
    border: none;
    border-radius: 15px;
    background-color: #E1D9CE;
    color: #fff;
    cursor: pointer;
}

button {
    padding: 15px;
    align-self: center;
    width: 50%;
    border: solid 1px;
    border-radius: 30px;
    background-color: #E1D9CE;
    color: #000000;
    font-weight: 700;
    cursor: pointer;
    text-transform: uppercase;
}

label {
    font-weight: bold;
}

.error {
    color: red;
    font-size: 14px;
}
```

Fonte: Dos próprios autores, 2024.

Desta forma com o código acima resulta-se em um formulário mais apresentável e estilizado de forma simples.

Figura 7 – Resultado do formulário HTML estilizado em CSS

Fonte: Dos próprios autores, 2024.

2.4 JavaScript

Tendo sua primeira versão lançada em 1995, JavaScript foi desenvolvido pela Netscape em colaboração com a Sun Microsystems, sendo parte fundamental da tríade de tecnologias *web*, juntamente com o HTML e CSS. (Iepsen, 2022). O JavaScript é altamente eficaz e uma das linguagens mais populares do mundo, ocupando uma posição proeminente na internet. (Groner 2019). Apesar de possuir um nome semelhante, JavaScript diferentemente do Java, evolui de uma linguagem de *script* para uma linguagem de uso geral, com recursos modernos voltados para o desenvolvimento de *software* em larga escala. (Flanagan, 2013). É uma linguagem de programação do lado do cliente, o que significa que sua interpretação e execução dependem das funcionalidades do navegador do usuário, que normalmente possuem um interpretador JavaScript integrado. (Silva, 2010).

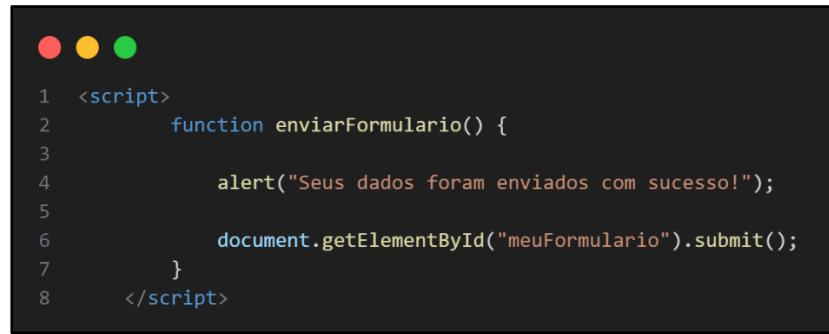
Figura 8 – Elemento “onsubmit” no HTML

```
1 <form id="meuFormulario" method="post" onsubmit="enviarFormulario();">
```

Fonte: Dos próprios autores, 2024.

- Onsubmit: é usado em um formulário HTML para especificar a ação que deve ser executada quando o formulário é enviado.

Figura 9 – Exemplo de código JavaScript



```

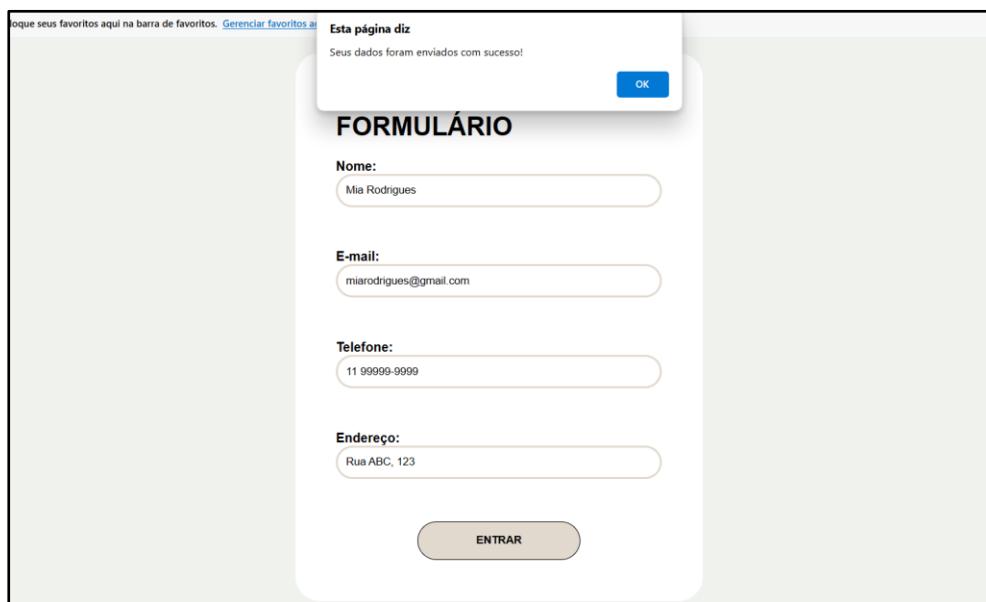
1 <script>
2     function enviarFormulario() {
3
4         alert("Seus dados foram enviados com sucesso!");
5
6         document.getElementById("meuFormulario").submit();
7     }
8 </script>

```

Fonte: Dos próprios autores, 2024.

- *Script*: “<script></script>” os códigos *script* deve ser escritos dentro da *tag SCRIP*T, ou podem ser colocados em um arquivo separado com extensão .js, que deve ser referenciado no documento HTML.
- *Function*: “function enviarFormulario()” define uma função chamada enviarFormulario() que será chamada quando necessário.
- *Alert*: “alert()” exibirá uma caixa de diálogo com uma mensagem quando a função for executada.
- *Function*: “document.getElementById().submit()” obtém o elemento HTML pelo ID usando a função document.getElementById(). Em seguida, a função submit() é chamada nesse formulário, enviando os dados do formulário para um servidor.

Figura 10 – Resultado da programação JavaScript



Fonte: Dos próprios autores, 2024.

2.5 UML

O UML, ou melhor *Unified Modeling Language*, é uma linguagem visual muito utilizada para retratar *software* orientado a objetos. É uma linguagem versátil aplicável a diversos domínios e é adotada como padrão internacional na engenharia de *software*. (GUEDES, 2018). Surgiu da união de diversas linguagens gráficas de modelagem orientadas a objetos que surgiram nos anos 80 e 90, assim desde sua introdução em 1997, vem sendo um recurso valorizado por muitos desenvolvedores. (FOWLER, 2005).

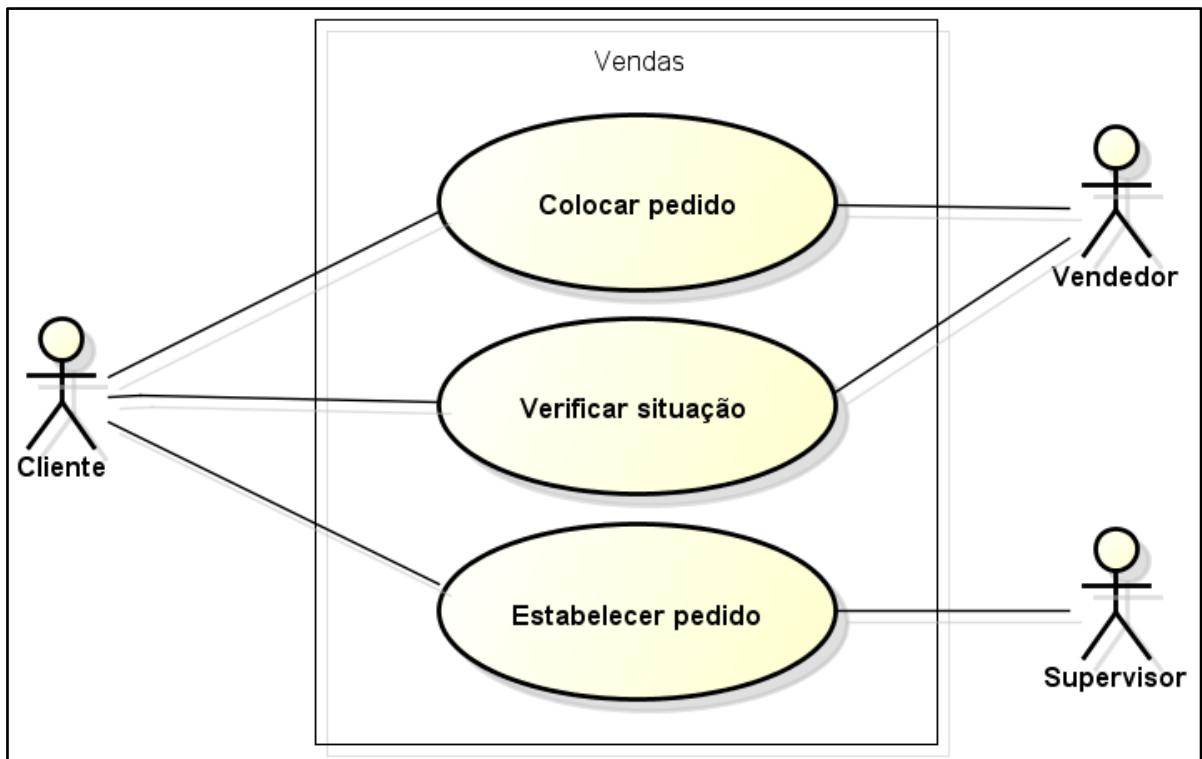
Modelagem está ligada a comunicação, e a UML concede ferramentas importantes para visualizar, especificar, construir e documentar artefatos de diversos sistemas de *software* complexos. (BOOCH, 2006). É importante modelar todo sistema antes de começar sua implementação, pois os sistemas de informação tendem a crescer em tamanho, complexidade e alcance, logo são dinâmicos e estão sempre em evolução. (GUEDES, 2018).

Portanto essa linguagem de modelagem possui diversos diagramas para o melhor entendimento e documentações dos sistemas criados, que serão mais bem abordados nos próximos capítulos.

2.5.1 UML - Diagrama de Casos de Uso

Utilizados para organização e modelagem de como é o comportamento de um sistema, por meio de conjunto de casos de uso que são funcionalidades e atores que interagem com o sistema. (BOOCH, 2006).

Figura 11 – Exemplo de diagrama de Caso de Uso



Fonte: Lima, 2009.

2.5.2 UML - Documentação Caso de Uso

A documentação de um caso de uso descreve, através de uma linguagem simples, informações nas quais os atores interagem aos casos de uso, como execução dos parâmetros, restrições e validações. (GUEDES, 2018). Abaixo podemos analisar de uma forma lúdica um exemplo de documentação de caso de uso da abertura de uma conta.

Tabela 1 - Exemplo Documentação de Caso de Uso abertura de uma conta

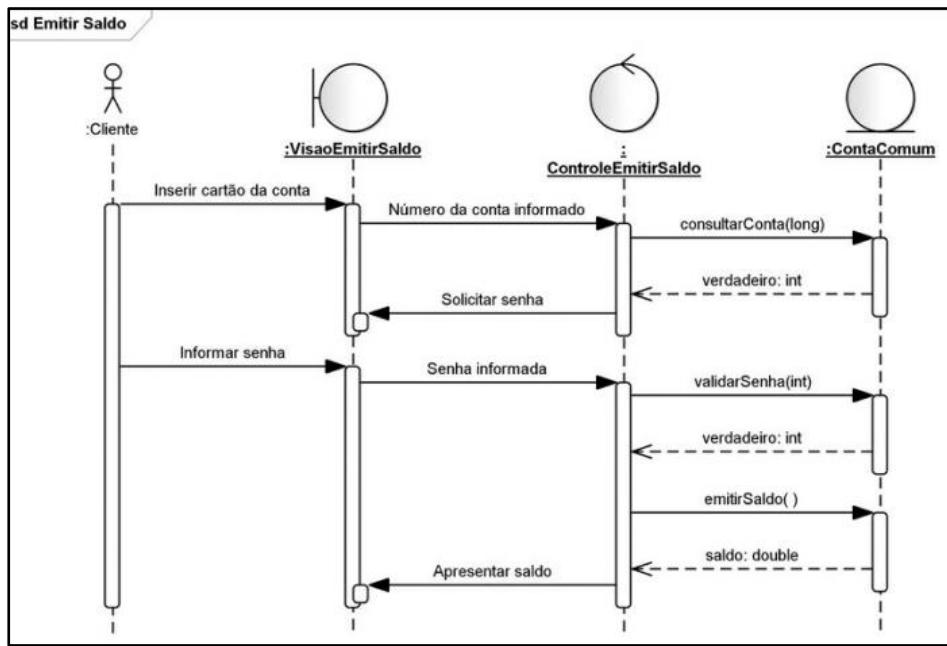
Nome do Caso de Uso		UC01 – Abrir Conta
Caso de Uso Geral		
Autor Principal	Funcionário	
Atores Secundários	Cliente	
Resumo	Esse caso de uso descreve as etapas percorridas por um cliente, intermediado por um funcionário, para abrir uma conta-corrente	
Pré-condições	O pedido de abertura precisa ter sido previamente aprovado	
Pós-condições	É necessário realizar um depósito inicial	
Cenário Principal		
Ações do Ator		Ações do Sistema
2. O funcionário informa o CPF ou CNPJ do cliente e consulta seu registro		3. Consultar cliente por seu CPF ou CNPJ
4. O cliente informa a senha da conta		5. Abrir conta
6. O cliente fornece um valor a ser depositado		7. Executar caso de uso "Realizar Depósito" para registrar o depósito do cliente 8. Emitir cartão da conta
Restrições/Validações		
1. Para abrir uma conta-corrente, é preciso ser maior de idade 2. O valor mínimo de depósito é R\$ 5,00 3. O cliente precisa fornecer algum comprovante de residência		
Cenário Alternativo – Manutenção do Cadastro do Cliente		
Ações do Ator		Ações do Sistema
		1. Executar o Caso de Uso "Gerenciar Clientes", para registrar um novo cliente ou atualizar o cadastro do cliente consultado
Cenário de Exceção – Cliente menor de idade		
Ações do Ator		Ações do Sistema
		1. Comunicar ao cliente que ele não tem a idade mínima para possuir uma conta-corrente 2. Recusar o pedido

Fonte: Guedes, 2018.

2.5.3 UML - Diagrama de Sequência

Utilizado para explorar como determinada ação ocorre, apresentando diversos exemplos de comunicações entre os elementos passados pelos objetos envolvidos nos casos de uso (FOWLER, 2005).

Figura 12 – Exemplo de diagrama de Sequência

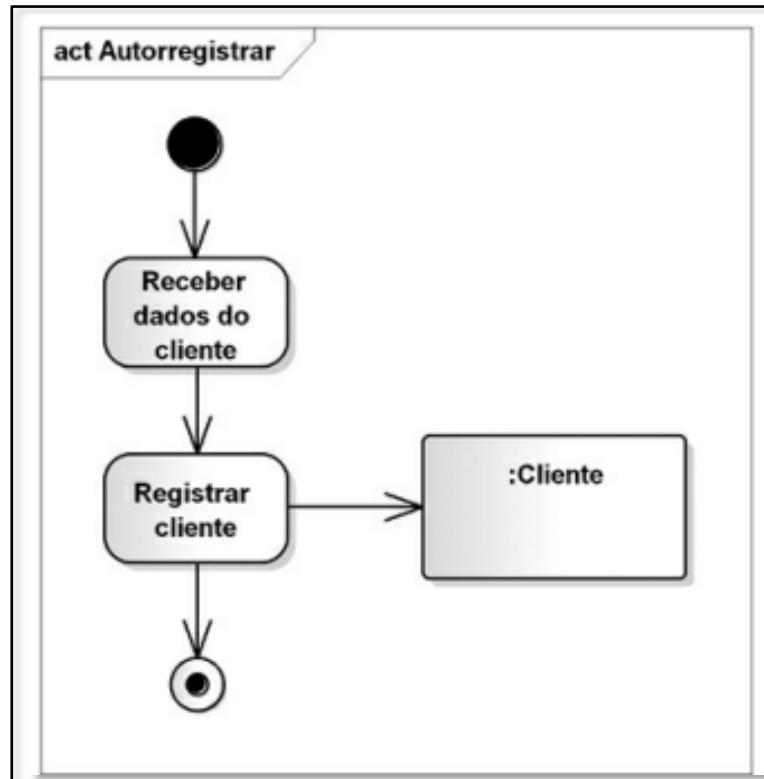


Fonte: Guedes, 2018.

2.5.4 UML - Diagrama de Atividade

Os diagramas de atividades são usados para modelar aspectos dinâmicos de um sistema, mostrando etapas sequenciais ou concorrentes de um processo computacional e o fluxo de objetos entre estados (BOOCH, 2006).

Figura 13 – Exemplo de diagrama de Atividade

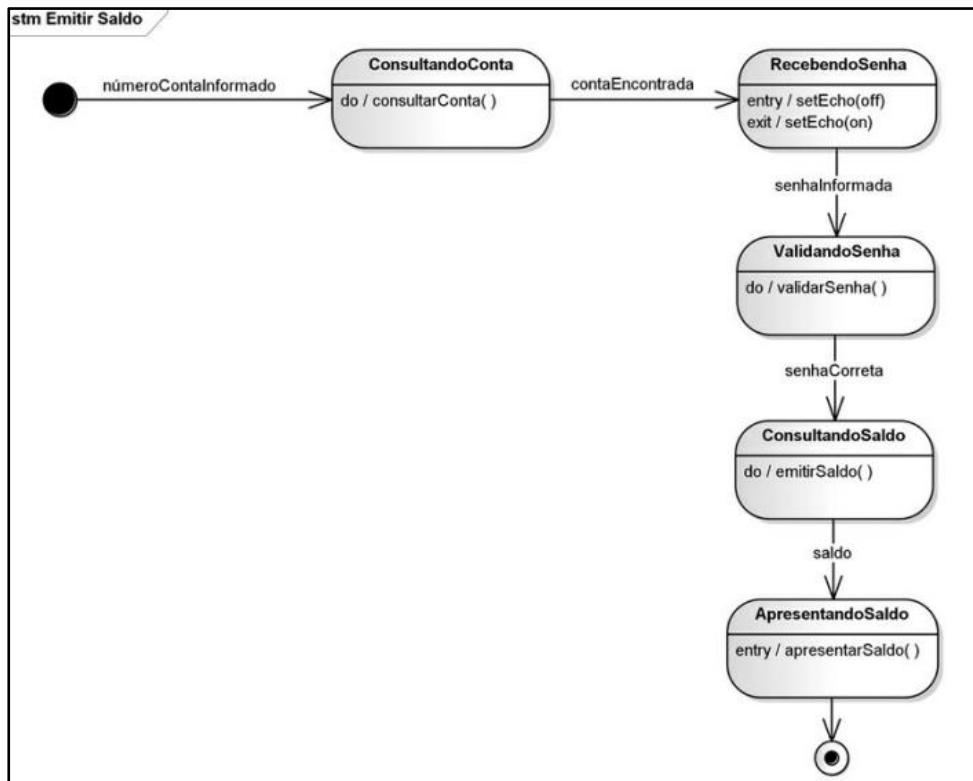


Fonte: Guedes, 2018.

2.5.5 UML - Diagrama de Máquina de Estados

Os diagramas de máquina de estados descrevem o comportamento de um sistema, mostrando o ciclo de vida de um objeto de uma classe eles são usados desde os anos 60, especialmente em técnicas orientadas a objetos (FOWLER, 2005).

Figura 14 – Exemplo de diagrama de Máquina de Estados



Fonte: Guedes, 2018.

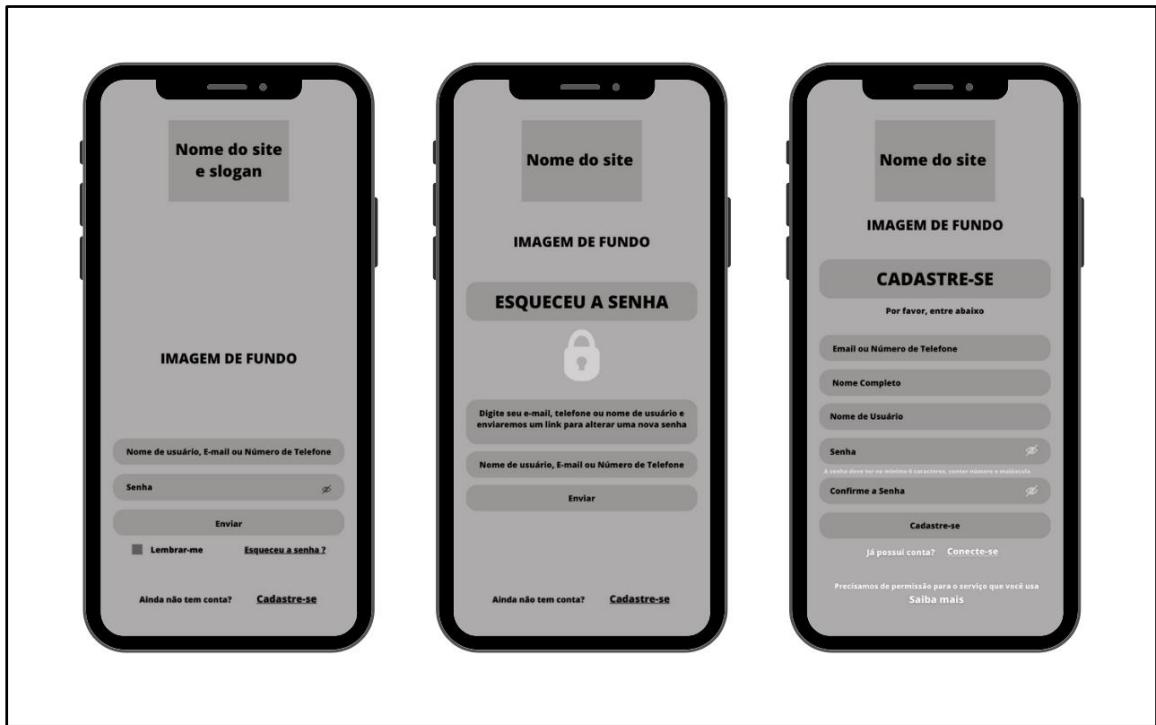
2.6 Wireframe

O termo “*wireframe*” descreve a modelagem no desenvolvimento de páginas na *web* em 3D para animações por computadores e em 2D para aplicativos móveis. (LUCIDCHART, 2023).

O aspecto crucial para utilização de *wireframes* é a aproximação para a prévia do *design* para o cliente, justamente pela capacidade de análise crítica e construtiva do mesmo. (Orgânica Digital, 2022).

Existem três tipos de *wireframe* de acordo com (ROCKCONTENT, 2019) o de Baixa Fidelidade que é feito manualmente e sem muitos detalhes, o Anotado que contém legendas e descrições, e o de Alta Fidelidade que é o mais próximo da versão final. Para maior compreensão abaixo podemos encontrar exemplos de *wireframes* de Baixa Fidelidade e de Alta fidelidade.

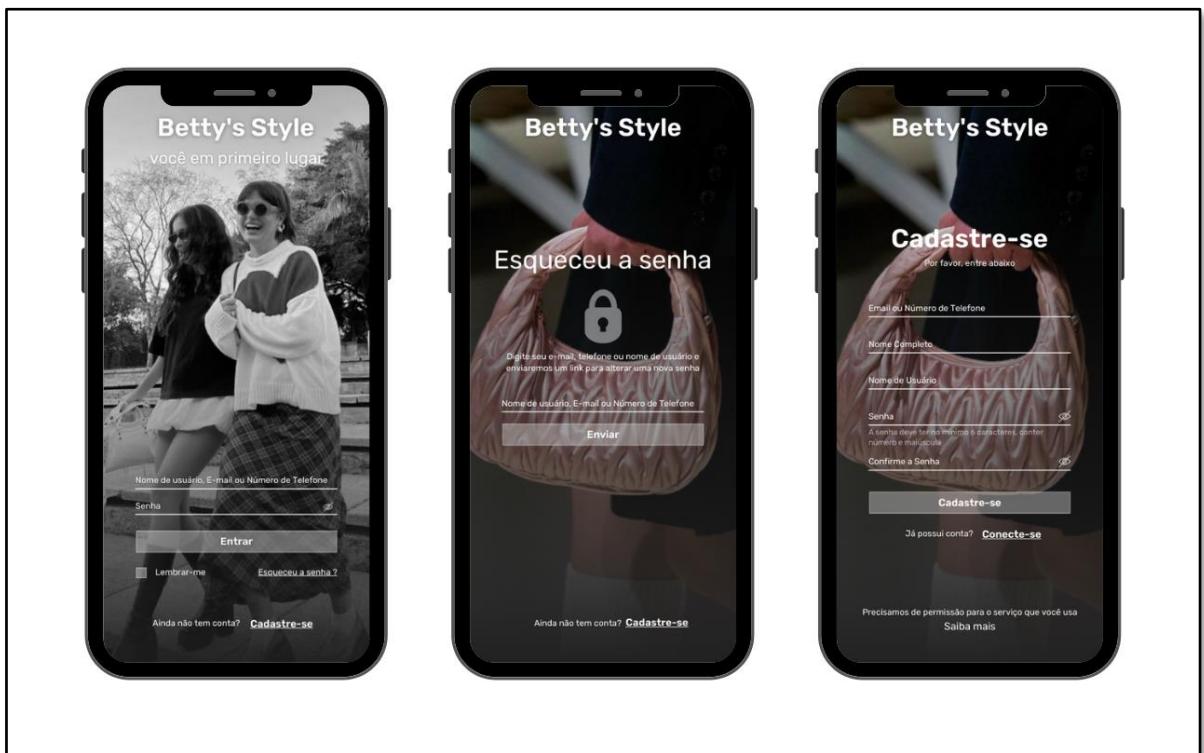
Figura 15 – Exemplo de *wireframe* de baixa fidelidade



Fonte: Dos próprios autores, 2024.

Acima, podemos ver o *wireframe* de baixa fidelidade, com as telas de *Login*, Esqueceu a Senha e Cadastre-se, e abaixo, o de alta fidelidade.

Figura 16 – Exemplo de *wireframe* de alta fidelidade



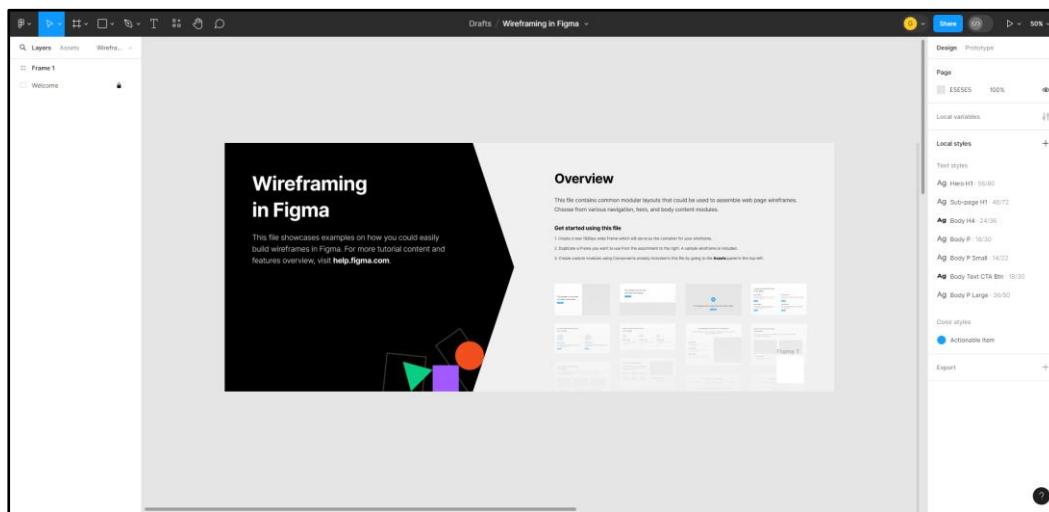
Fonte: Dos próprios autores, 2024.

2.7 Figma

O Figma é uma plataforma de design que permite criar, colaborar e prototipar interfaces de usuários e *designs* interativos. Diferentemente de outras ferramentas de design o Figma é totalmente acessível pelo navegador. (Cubos Academy, 2023).

De acordo com Ebac Online (2023), vale ressaltar a importância do Figma como uma ferramenta que tem sido muito utilizada para entrega de projetos digitais de ponta a ponta, sendo um projeto *web* ou um aplicativo *mobile*. Abaixo encontra-se uma imagem da tela principal de criação de projetos do aplicativo.

Figura 17 – Tela de criação de projetos do Figma



Fonte: Figma, 2024.

2.8 Banco de dados

Um sistema de banco de dados é basicamente um sistema computadorizado de manutenção de registros. Os registros em questão podem ser qualquer coisa que tenha significado ao indivíduo ou a organização a que o sistema deve servir. (DATE, 2023).

2.8 Firebase

Conforme Oracle (2023), o Firebase é uma plataforma de desenvolvimento de aplicativos *mobile* e *web* do Google, é vantajoso incluindo um ambiente de desenvolvimento ponta a ponta, rápido para criar aplicativos e infraestrutura escalável.

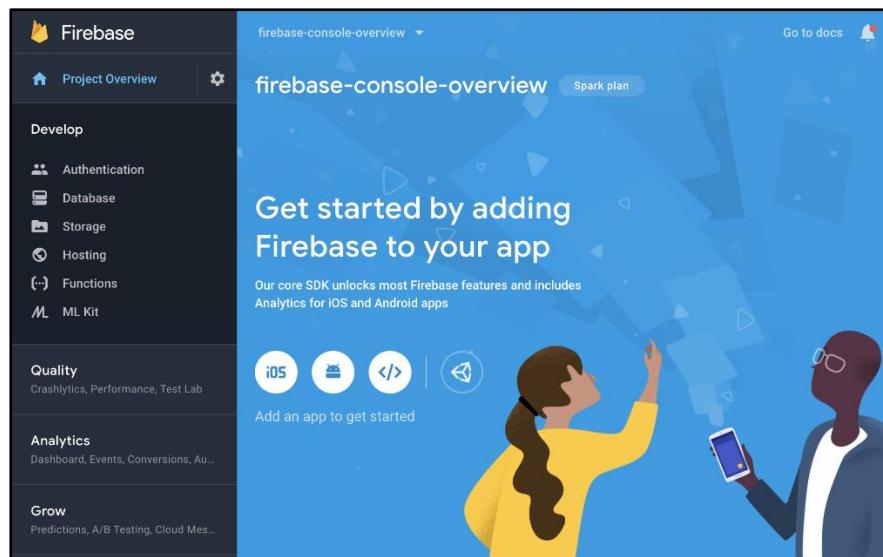
Figura 18 – Logotipo Firebase



Fonte: Firebase, 2023.

O console do Firebase oferece o ambiente mais sofisticado para gerenciar produtos, apps e configurações do nível do projeto Firebase, abaixo é possível perceber que sua lista de produtos é organizada por categorias de nível superior. (FIREBASE, 2023).

Figura 19 – Console Firebase



Fonte: Firebase, 2023.

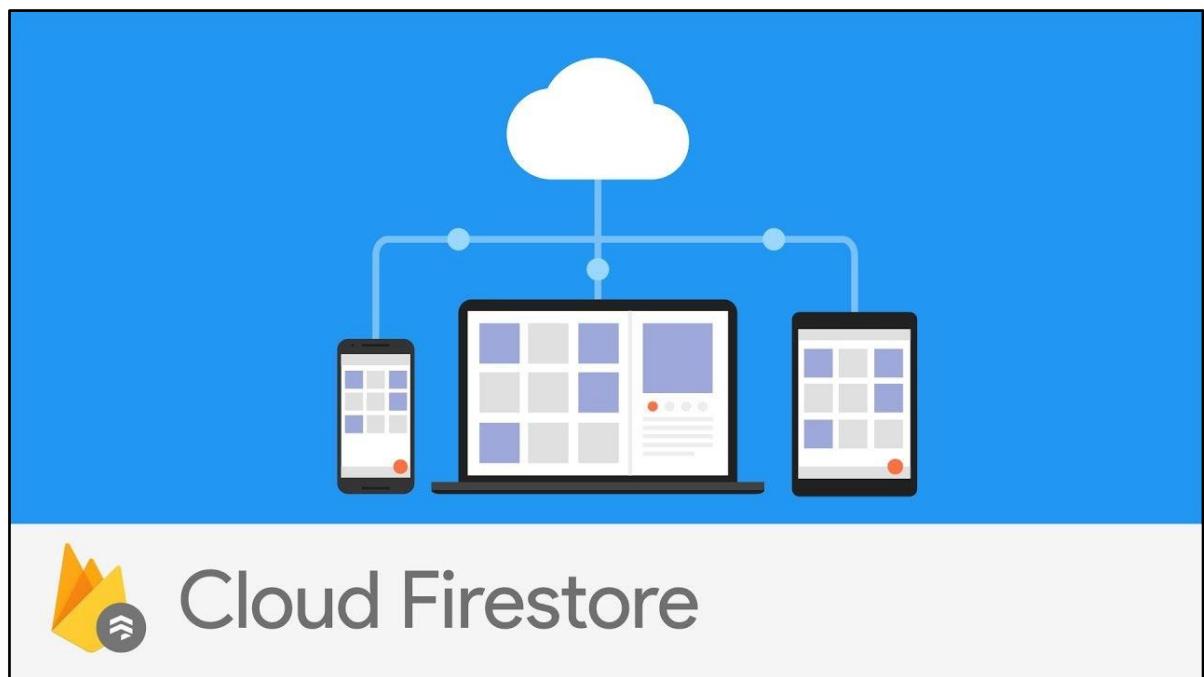
2.9 Firestore

O Cloud Firestore é um banco de dados NoSQL, que permite armazenar, sincronizar e consultar dados nos dispositivos móveis e web, além disso disponibiliza regras de segurança para acessar o banco sem precisar manter o próprio servidor. (Firebase).

Além disso, tem o foco no desenvolvimento do aplicativo usando banco de dados de documentos gerenciado e sem servidor, fazendo ajustes com o objetivo de atender

qualquer demanda sem janelas de manutenção ou inatividade. (GOOGLE CLOUD, 2024).

Figura 20 – Cloud Firestore



Fonte: Firebase, 2023.

2.10 Node.js & NPM

Node.js, criado 2009, é uma plataforma de execução de JavaScript no servidor, conhecida por sua arquitetura assíncrona e eficiência em operações de I/O, ideal para desenvolver APIs, aplicações de tempo real e *backends* escaláveis. (PEREIRA, 2014).

O Node.js utiliza um modelo *non-blocking thread*, otimizando o processamento ao eliminar paralisações por I/O, isso permite criar aplicações escaláveis e eficientes, sem esperas prolongadas, devido à sua arquitetura orientada a eventos. (DUARTE, 2020).

NPM é o gerenciador de pacotes do Node.js, facilitando a gestão de dependências, acesso a bibliotecas JavaScript e controle de versões, simplificando o desenvolvimento e compartilhamento de projetos. (GADO, 2021).

2.11 React

React é uma biblioteca em JavaScript que simplifica e agiliza a tarefa de desenvolvimento de interfaces de usuário interativas e de alto desempenho, é usado por grandes plataformas como Facebook e Instagram. (SILVA, 2021).

Suas aplicações são compostas por componentes, que são para da IU (interface do usuário) que possui sua própria lógica e aparência. (React, 2024).

2.12 React Native

O React Native, baseado no React e desenvolvido pelo Facebook, é um *framework* JavaScript para criar aplicativos multiplataforma. Ele combina desenvolvimento nativo com React, oferecendo uma experiência autêntica aos usuários. (Escudelario; Pinho, 2020).

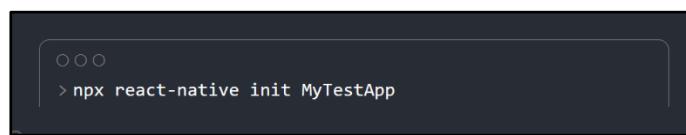
Com uma abordagem declarativa e compartilhamento de código, ele agiliza o desenvolvimento, possibilitando que equipes criem aplicativos nativos de forma eficiente. (React Native).

O funcionamento do React Native depende do Node.js, que converte o código JavaScript para plataformas como Android e iOS. Além disso, o Node.js gerencia as bibliotecas e pacotes essenciais para o projeto. (DevMedia).

2.12.1 Criação de um aplicativo em React Native

Para criar um aplicativo React Native, você pode utilizar o seguinte comando no Prompt de Comando, após instalar o Node.js e entrar no diretório desejado:

Figura 21 – Comando de criação projeto React Native

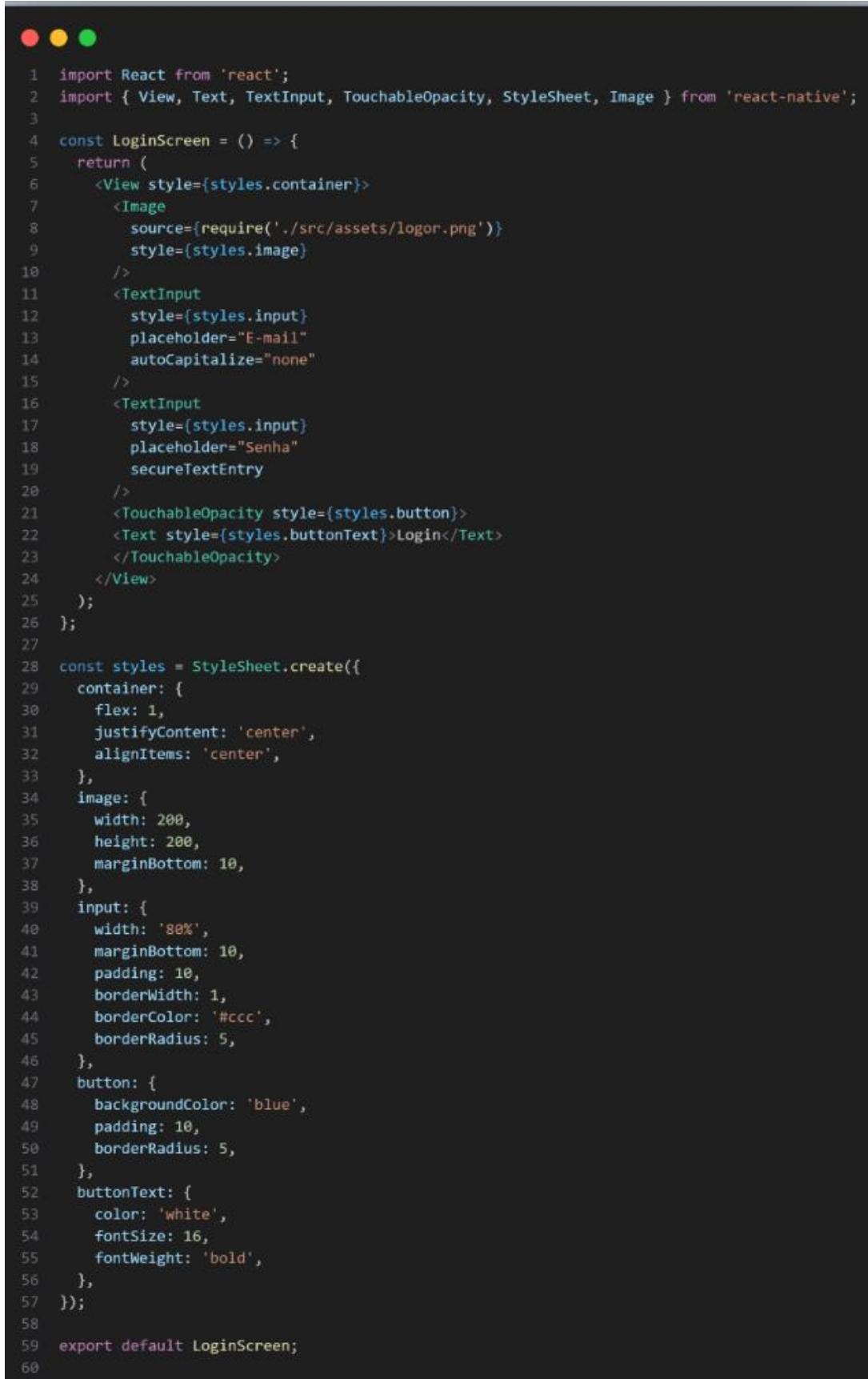


```
npx react-native init MyTestApp
```

Fonte: React Native, 2024.

Isso criará um aplicativo React Native chamado MyTestApp no diretório atual. Após a criação do projeto, você poderá desenvolver a aplicação utilizando a IDE de sua preferência.

Figura 22 – Exemplo de código React Native

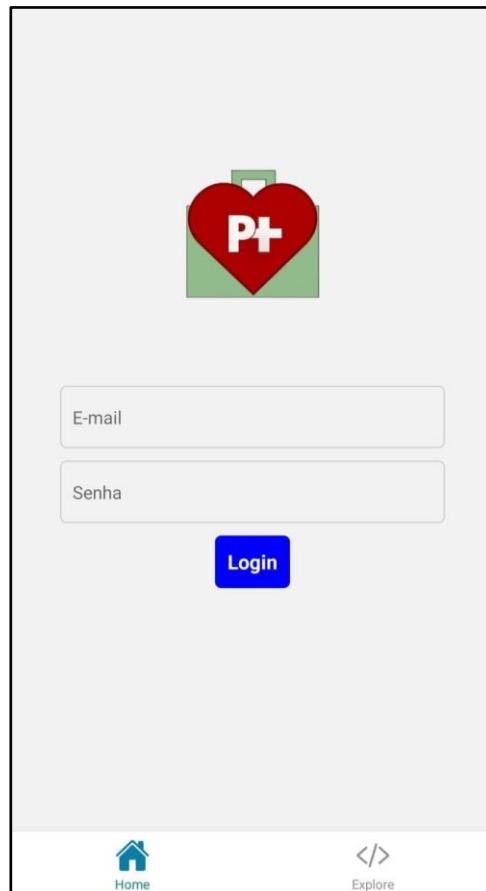


```
1 import React from 'react';
2 import { View, Text, TextInput, TouchableOpacity, StyleSheet, Image } from 'react-native';
3
4 const LoginScreen = () => {
5   return (
6     <View style={styles.container}>
7       <Image
8         source={require('../src/assets/logo.png')}
9         style={styles.image}
10      />
11      <TextInput
12        style={styles.input}
13        placeholder="E-mail"
14        autoCapitalize="none"
15      />
16      <TextInput
17        style={styles.input}
18        placeholder="Senha"
19        secureTextEntry
20      />
21      <TouchableOpacity style={styles.button}>
22        <Text style={styles.buttonText}>Login</Text>
23      </TouchableOpacity>
24    </View>
25  );
26};
27
28 const styles = StyleSheet.create({
29   container: {
30     flex: 1,
31     justifyContent: 'center',
32     alignItems: 'center',
33   },
34   image: {
35     width: 200,
36     height: 200,
37     marginBottom: 10,
38   },
39   input: {
40     width: '80%',
41     marginBottom: 10,
42     padding: 10,
43     borderWidth: 1,
44     borderColor: '#ccc',
45     borderRadius: 5,
46   },
47   button: {
48     backgroundColor: 'blue',
49     padding: 10,
50     borderRadius: 5,
51   },
52   buttonText: {
53     color: 'white',
54     fontSize: 16,
55     fontWeight: 'bold',
56   },
57 });
58
59 export default LoginScreen;
60
```

Fonte: Dos próprios autores, 2024.

Aqui, exibimos o formulário criado com React Native, junto com o código equivalente.

Figura 23 – Resultado da codificação React Native



Fonte: Dos próprios autores, 2024.

2.13 IoT

A Internet das Coisas, conhecida como, IoT é a rede de dispositivos e, em geral, as coisas que estão conectadas e se comunicam entre si e através de redes, para realizar determinadas tarefas sem exigir interação entre seres humanos. (SANTOS, 2018).

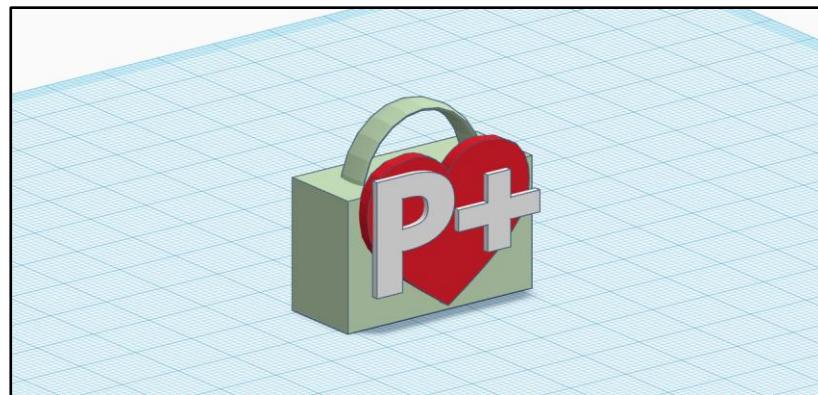
O “produto IoT” vai além do produto inteligente e do produto conectado, por explorarem toda a capacidade da internet em produtos físicos, sendo assim, ele é efetivamente um sistema, ou, melhor dizendo um sistema de sistemas. (SINCLAIR, 2018).

O avanço das tecnologias, especialmente as digitais, iram afetar profundamente todas as estruturas econômicas e sociais, e talvez, a mais impactante e pervasiva dessas tecnologias digitais seja a internet das coisas. (MAGRANI, 2018).

2.13.1 Modelagem 3D

A modelagem 3D é considerada um processo de desenvolvimento de personagens, objetos ou cenários em três dimensões, ou seja, possuem profundidade além de altura e largura. (LOPES, 2023). Abaixo temos um exemplo de Modelagem 3D.

Figura 24 – Exemplo de Modelagem 3D



Fonte: Dos próprios autores, 2024.

Acima, apresentamos o protótipo de um modelo criado para impressão 3D. Abaixo, você pode ver a foto do modelo após ser impresso e pintado.

Figura 25 – Exemplo de prototipagem 3D impressa e pintada



Fonte: Dos próprios autores, 2024.

2.13.2 ESP 32

Desenvolvido pela Espressif Systems, o ESP-32 é um microcontrolador eficiente e de baixo custo, excelente para projetos de IoT, entretenimento e automação

residencial, possuindo suporte de WIFI, Bluetooth e outros tipos de conexão. (MAKIYAMA, 2023).

Posto isso, ESP-32 se torna ideal para projetos de IoT visto que possui capacidade de conexão à internet e a outros dispositivos, com processador dual-core e 500 KB de SRAM, que permite execução de programas complexos. (PEREIRA, 2020). Assim o ESP-32 possui 36 pinos digitais com 16 utilizáveis como PWM, e portando entradas e saídas, onde entradas, são como pressionar um botão, que enviam sinais ao microcontrolador, e pode ativar saídas como *LEDs* e motores. (ELETRÔNICA ÔMEGA, 2021).

Para o projeto PillTrack, ele será utilizado para a monitoração de uso medicamentoso em tempo real, fazendo conexão por *Wi-Fi* entre o aplicativo, o banco de dados e a caixa de remédios inteligente. Através dele será possível o envio de informações de uso de medicamentos a partir do momento em que o paciente se medicar de acordo com a prescrição médica e o recebimento de notificação de cada dose diária. Sendo assim ele será eficiente para facilitar o controle de medicamentos, na imagem a seguir, temos um ESP-32 para demonstração:

Figura 26 – ESP 32



Fonte: RoboCore, 2024.

2.13.3 Jumpers

Os *jumpers* são fios elétricos que fazem conexões entre os componentes, possuem diversas cores, espessuras e tamanhos, são usadas principalmente em Protoboards e Arduinos (MONK, 2014).

Nesta aplicação usaremos os jumpers fêmea com fêmea, que são utilizados para a conexão de pinos machos e jumpers fêmea com macho, pois faremos uso do ESP-32 e da Protoboard, sendo assim esse é o tipo de jumper ideal para o projeto. Abaixo temos uma imagem demonstrativa de alguns tipos de jumpers:

Figura 27 – Jumpers



Fonte: Casa da Robótica, 2024.

2.13.4 Buzzer Passivo

Os buzzers são dispositivos eletromecânicos que transformam energia elétrica em som audível, sendo amplamente utilizados em diversas aplicações para emitir alertas ou melodias devido ao seu baixo custo e facilidade de conexão e operação. (Alvarez, 2023).

O buzzer passivo reproduz o som conforme a forma do sinal elétrico que o aciona, permitindo imitar sons específicos, ao contrário do buzzer ativo que emite apenas um apito com timbre próprio. (GUIMARÃES, 2017).

O *buzzer* passivo será usado para emitir som no horário em que o paciente precisar tomar os medicamentos prescritos pelo médico.

Figura 28 – Buzzer Passivo



Fonte: Eletrônica Cuiabá, 2024.

2.13.5 Display LCD

Criado pelo engenheiro George Heilmer em 1964, *LCD (Liquid Crystal Display)* é uma tecnologia que utiliza de cristais líquidos e polarizadores de luz para formar imagens, comuns em eletrônicos como celulares e TVs (HIGA; MARQUES, 2023).

Os displays *LCD* alfanuméricos são encontrados em diversos aparelhos, possuem interfaces práticas e embora sejam uma tecnologia de mais de vinte anos, continuam populares e econômicos. (PUHLMANN, 2015).

A interface para conectar um microcontrolador é padronizada, variando entre 14 e 16 pinos conforme a presença de *backlight*, que possui uma corrente típica de 60 mA e máxima de 75 mA, com tensões diretas de 3,5V e 3,6V, respectivamente. (PUHLMANN, 2015).

O Display LCD, será essencial para a criação do projeto, pois através dele o usuário poderá ver de forma interativa na parte frontal o horário, nome da caixa e o compartimento que está sendo ativado o motor. Na figura subsequente, é possível analisar a estrutura de uma placa LCD:

Figura 29 – Display LCD



Fonte: ArduCore, 2024.

2.13.6 Motor de Passo + Módulo de Controle (Driver ULN2003)

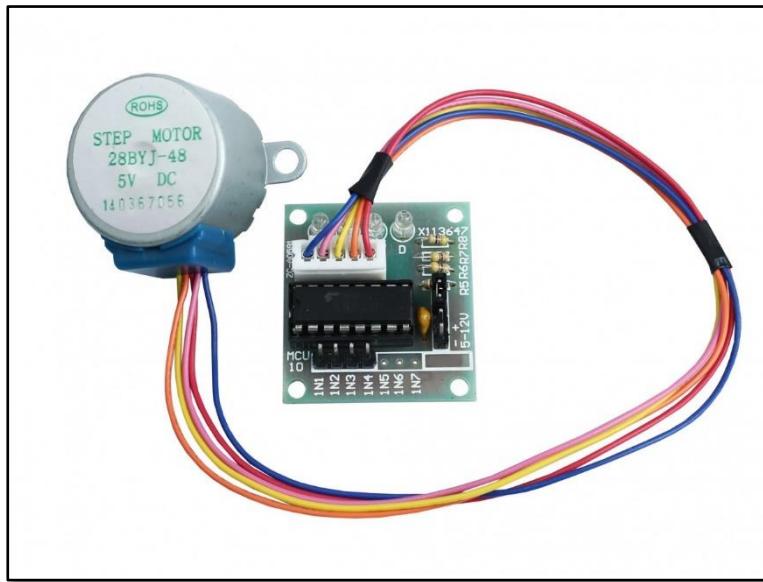
O Motor de Passo + Módulo de Controle (*Driver ULN2003*), de acordo com Viana (2022), é um motor elétrico de corrente contínua em que seu eixo gira em passos precisos, movendo-se por uma quantidade fixa de graus, sendo precisos de ângulo, velocidade, posição e sincronismo.

O motor é unipolar possuindo 4 enrolamentos que são chamados de fases, sendo assim as mesmas são conectadas juntas. (Eletrogate). Geralmente este motor é usado juntamente com o *Driver ULN2003* que permite ser controlado facilmente pelo motor de passo com um microcontrolador. (VIANA, 2022).

O componente, será o responsável pelo gerenciamento da queda dos remédios diretamente para o copinho da caixa PillTrack, pois, ao girar, seu movimento deverá ser preciso, permitindo assim que o compartimento onde ficaram armazenados os remédios gire 180° , liberando a queda da pílula, uma por vez. Ao concluir o “despejo” da pílula, o sistema voltará automaticamente para sua posição original.

Na figura abaixo, uma imagem do Motor de Passo + Módulo de Controle (*Driver ULN2003*) será apresentada:

Figura 30 - Motor de Passo + Módulo de Controle (*Driver ULN2003*)



Fonte: Usinainfo, 2024.

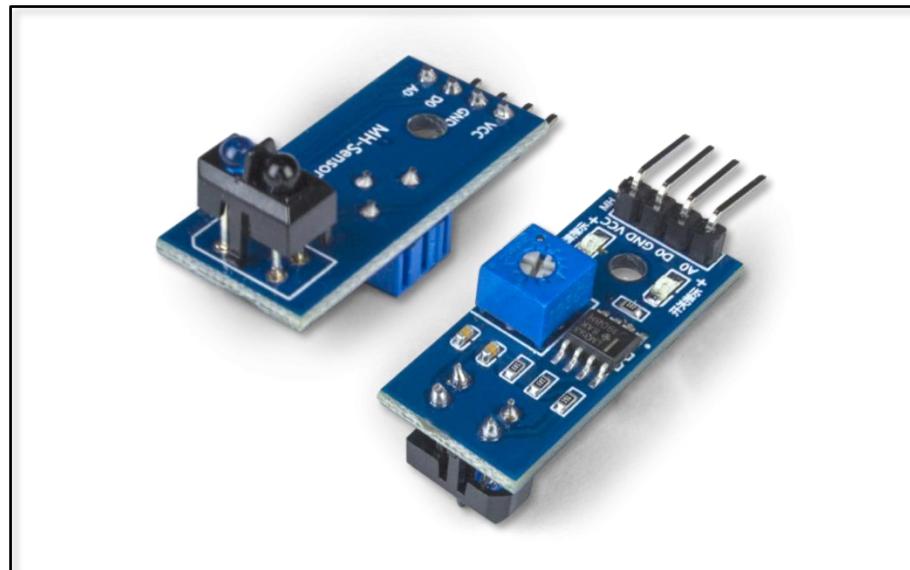
2.13.7 Sensor Óptico TCRT5000

O TCRT5000 é um sensor óptico reflexivo que utiliza a tecnologia de reflexão infravermelha para detectar a presença de objetos. (ELETROGATE, 2017). Ele é composto por um LED infravermelho e um fototransistor IR, ambos presos num suporte plástico, quando um objeto se aproxima do sensor, a luz infravermelha é refletida para o fototransistor, ativando-o. (Arduino e Cia).

Este sensor será de extrema importância para detectar a retirada do medicamento através de sua luz infravermelha, pois, ao ser removido, sua reflexão é interrompida, dessa forma o sensor emitirá a ausência do remédio enviando um sinal para o sistema, que será interpretado pelo aplicativo que o usuário ingeriu os medicamentos, podendo assim atualizar o relatório de forma que o paciente está seguindo as prescrições médicas corretamente.

Agora, será possível ver uma imagem do Sensor Óptico TCRT5000:

Figura 31 - Sensor Óptico TCRT5000

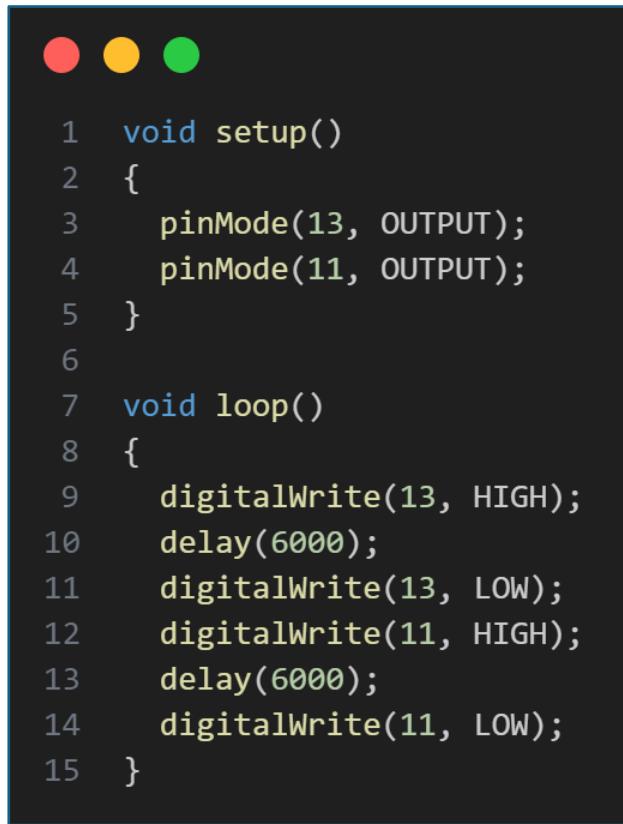


Fonte: RoboCore, 2024.

2.13.8 C++

Desenvolvida por Bjarne Stroustrup na Bell Laboratories nos anos 80, C++ foi criado sendo uma linguagem de programação de baixo nível e fornecendo recursos adicionais a linguagem C. (DEITEL; PAUL, 2015). Sua sintaxe versátil suporta programação orientada a objetos, procedural, genérica e funcional, tornando-a uma escolha poderosa para uma variedade de aplicações em diferentes plataformas. (Locaweb). A filosofia do C++ de construir sobre si mesmo simplifica a criação de compiladores e leva a uma linguagem com uma biblioteca padrão e muitas opções de compiladores. (VILLAS-BOAS, 2001). A seguir, temos um exemplo de código em C++:

Figura 32 – Exemplo C++



```
1 void setup()
2 {
3     pinMode(13, OUTPUT);
4     pinMode(11, OUTPUT);
5 }
6
7 void loop()
8 {
9     digitalWrite(13, HIGH);
10    delay(6000);
11    digitalWrite(13, LOW);
12    digitalWrite(11, HIGH);
13    delay(6000);
14    digitalWrite(11, LOW);
15 }
```

Fonte: Do próprio autor, 2024.

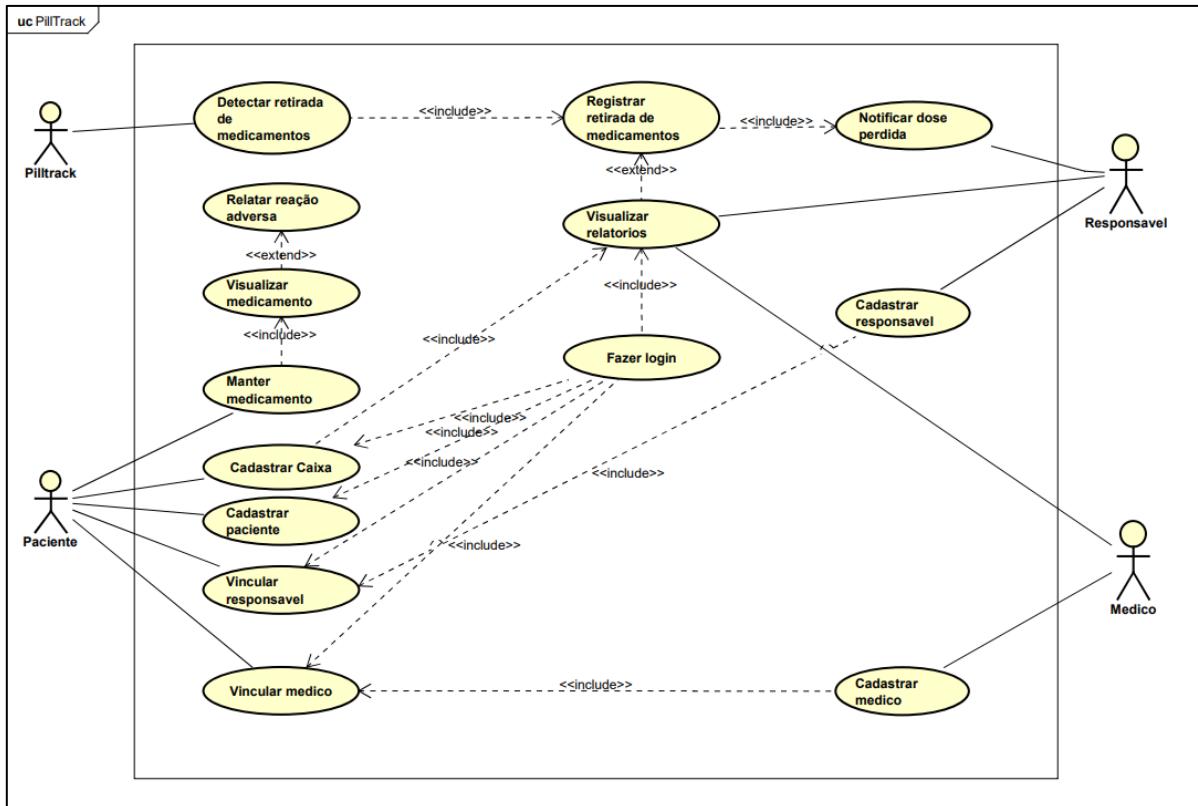
3 DESENVOLVIMENTO

Neste capítulo será abordado a metodologia utilizada para a criação do sistema PillTrack, fundamentando-se em métodos de documentação UML, montagem do circuito e as imagens das telas do sistema.

3.1 Diagrama de Casos de Uso

Segue abaixo uma imagem onde pode-se examinar detalhadamente o Diagrama de Caso de Uso do sistema PillTrack onde são evidenciadas interações do sistema com os atores, que incluem: Dispositivo PillTrack, Paciente, Responsável e Medico, onde apresenta ações que cada um deles podem executar. Neste diagrama, constam-se ao todo, 14 elipses de caso de uso.

Figura 33 – Diagrama de Caso de Uso do sistema PillTrack



Fonte: Dos próprios autores, 2024.

3.2 Documentação Caso de Uso

Apresenta-se a seguir a documentação dos casos de uso vistos no capítulo anterior, seguindo as normas dos componentes UML, segmentando-se entre requisitos funcionais, requisitos não funcionais e regras de negócio.

Os requisitos funcionais são aqueles que demonstram as funcionalidades do sistema, sendo assim, são primordiais para o funcionamento. Os requisitos não funcionais demonstram os atributos requeridos pelos usuários podendo ser associados com confiabilidade, segurança, visual e desempenho, não sendo primordial para o desenvolvimento do sistema. Já as regras de negócio, são nuances imprescindíveis que o usuário almeja no sistema.

Atores Genéricos (Todos):

- RF01: Todos os atores devem ser capazes de realizar o login no sistema para acessar suas funcionalidades específicas.
- RF02: Todos os atores poderão ter acesso a visualização dos relatórios de uso de medicamentos;

Requisitos Funcionais do Sistema PillTrack:

- RF01: O sistema deve detectar quando os medicamentos são retirados;
- RF02: O sistema deve permitir o registro da retirada dos medicamentos;
- RF03: O sistema deve notificar o responsável em caso de uma dose perdida.

Requisitos Funcionais do Paciente:

- RF01: O paciente pode relatar efeitos colaterais ou reações adversas;
- RF02: O paciente pode consultar detalhes sobre o medicamento;
- RF03: O paciente pode atualizar ou gerenciar informações sobre o medicamento;
- RF04: O paciente pode registrar uma caixa de medicamento no sistema;
- RF05: O paciente pode se cadastrar no sistema;
- RF06: O paciente pode associar um responsável a si;
- RF07: O paciente pode vincular um médico a si para acompanhar seu tratamento;

Requisitos Funcionais do Responsável:

- RF01: O responsável pode se cadastrar no sistema;

Requisitos Funcionais do Médico:

- RF01: O médico pode se registrar no sistema;

Requisitos Não Funcionais do Sistema:

- RF01: Todos os atores devem realizar login seguro com autenticação robusta;
- RF02: O sistema deve responder rapidamente às consultas, como visualização de relatórios e medicamentos;
- RF03: O sistema deve garantir a precisão das informações registradas, como a retirada de medicamentos e reações adversas;
- RF04: A interface do sistema deve ser intuitiva, permitindo que pacientes, médicos e responsáveis naveguem facilmente entre as funções;
- RF05: O sistema deve ser capaz de notificar rapidamente o responsável em caso de dose perdida ou outras emergências.

Regras de Negócio:

- RF01: Permitir apenas usuários autorizados e exigir autenticação para acesso ao sistema, restringindo o acesso a dados sensíveis;
- RF02: Registrar a administração de medicamentos, atualizando automaticamente o estoque e permitindo apenas uma confirmação por dose;
- RF03: Enviar notificações de administração de medicamentos e alertas, além de notificações aos cuidadores em caso de falha na administração;
- RF04: Permitir que médicos e cuidadores visualizem informações dos pacientes;
- RF05: Gerar relatórios de adesão ao tratamento, registrando todas as interações com o sistema.

Documentação dos Casos de Uso Atores Genéricos (Todos):

Quadro 1 - Documentação Caso de Uso (Fazer Login)

Nome do Caso de Uso	Fazer Login (RF01)
Autor Principal	Todos os atores do sistema
Resumo	Este caso de uso descreve o procedimento para efetuar o login
Pré-condições	Para efetuar login, é preciso ter um cadastro no sistema
Cenário Principal	
Ações do Ator	Ações do Sistema
	1. Solicita e-mail e senha
2. Insere dados solicitados pelo sistema	
	3. Verifica dados corretos
	4. Realiza login
Cenário Alternativo – Dados incorretos	
Ações do Ator	Ações do Sistema
	1. Informa que os dados estão incorretos

	2. Recusa solicitação de login
--	--------------------------------

Fonte: Dos próprios autores, 2024.

Quadro 2 - Documentação Caso de Uso (Visualizar relatórios)

Nome do Caso de Uso	Visualizar relatórios (RF02)
Ator Principal	Paciente, Médico e Responsável
Resumo	Esse caso de uso descreve o procedimento que o sistema deve possibilitar a visualização dos relatórios de uso de medicamentos.
Pré-condições	Para visualizar o relatório é preciso fazer login, detectar ou não a retirada dos medicamentos e ser registrado a retirada desses medicamentos no sistema.
Cenário Principal	
Ações do Ator	Ações do Sistema
1. Fazer Login	
2. Acessar área de relatórios	
3. Escolher um mês	
	4. Exibir relatórios gerados
Cenário Alternativo- Relatórios não existentes	
Ações do Ator	Ações do Sistema
	1. Exibe mensagem de sem relatórios

Fonte: Dos próprios autores, 2024.

Documentação dos Casos de Sistema PillTrack

Quadro 3 - Documentação Caso de Uso (Detectar retirada de medicamentos)

Nome do Caso de Uso	Detectar retirada de medicamentos (RF01)
Ator Principal	Sistema PillTrack

Resumo	Esse caso de uso descreve o procedimento de detecção do sistema quando os medicamentos são retirados.
Pré-condições	Para detectar a retirada é preciso ter cadastrado os medicamentos no aplicativo.
Pós-condições	A detecção do medicamento é registrada no sistema e visível nos relatórios
Cenário Principal	
Ações do Ator	Ações do Sistema
1. Sinaliza a hora de retirar medicamento	
2. Retirar medicamento	
3. Detectar a retirada do medicamento	
4. Registar a retirada do medicamento no relatório	
	5. Verificar informação
	6. Exibir informações
Cenário Alternativo - Medicamento não retirado	
Ações do Ator	Ações do Sistema
	1. Registrar a não retirada do medicamento no horário certo

Fonte: Dos próprios autores, 2024.

Quadro 4 - Documentação Caso de Uso (Registrar retirada de medicamentos)

Nome do Caso de Uso	Registrar retirada de medicamentos (RF02)
Autor Principal	Sistema PillTrack
Resumo	Esse caso de uso descreve o procedimento que o sistema deve registrar a retirada dos medicamentos.

Pré-condições	Para registrar a retirada de medicamentos é preciso detectar ou não a retirada deste medicamento
Pós-condições	Após o registro da retirada do medicamento ele pode ser visualizado pelo paciente e os médicos e responsáveis vinculados
Cenário Principal	
Ações do Ator	Ações do Sistema
1.Registar a retirada do medicamento no relatório	
	2.Verificar informação
	3.Exibir informações

Fonte: Dos próprios autores, 2024

Quadro 5 - Documentação Caso de Uso (Notificar dose perdida)

Nome do Caso de Uso	Notificar dose perdida (RF04)
Ator Principal	Responsável
Resumo	Esse caso de uso descreve o procedimento que o sistema deve notificar o responsável em caso de uma dose perdida.
Pré-condições	Para notificar o responsável é necessário ser vinculado à caixa pelo paciente.
Cenário Principal	
Ações do Ator	Ações do Sistema
	1. Enviar notificações para o responsável

Fonte: Dos próprios autores, 2024.

Documentação dos Casos de Uso Pacientes

Quadro 6 - Documentação Caso de Uso (Relatar reação adversa)

Nome do Caso de Uso	Relatar reação adversa (RF01)
Autor Principal	Paciente
Resumo	O paciente relata uma reação adversa ao medicamento para que o responsável e o médico sejam notificados e o registro seja armazenado.
Pré-condições	O paciente deve estar cadastrado e logado no sistema.
Pós-condições	A reação adversa é registrada no sistema e visível para o responsável e o médico.
Cenário Principal	
Ações do Ator	Ações do Sistema
1. O paciente faz login no sistema.	
	2. O sistema verifica as credenciais e concede acesso.
3. O paciente seleciona "Relatar Reação Adversa".	
	4. O sistema exibe um formulário para o relato.
5. O paciente preenche os detalhes da reação.	
	6. O sistema valida as informações e registra o relato.
Cenário Alternativo	
Ações do Ator	Ações do Sistema
1. O paciente não preenche todos os campos obrigatórios.	

	2. O sistema notifica o paciente sobre o erro e solicita a correção.
--	--

Fonte: Dos próprios autores, 2024.

Quadro 7 - Documentação Caso de Uso (Visualizar medicamento)

Nome do Caso de Uso	Visualizar medicamento (RF02)
Autor Principal	Paciente
Resumo	O paciente pode consultar detalhes sobre os medicamentos que está utilizando e foram cadastrados por ele.
Pré-condições	O paciente deve estar cadastrado e logado no sistema.
Cenário Principal	
Ações do Ator	Ações do Sistema
1. O paciente faz login no sistema.	
	2. O sistema verifica as credenciais e concede acesso.
3. O paciente acessa sua caixa e o compartimento do medicamento.	
	4. O sistema exibe os dados do medicamento cadastrado naquele compartimento.
Cenário Alternativo	
Ações do Ator	Ações do Sistema
1. O paciente seleciona um compartimento sem nenhum medicamento cadastrado.	
	2. O sistema exibe o formulário para cadastrar um medicamento.

Fonte: Dos próprios autores, 2024.

Quadro 8 - Documentação Caso de Uso (Manter medicamento)

Nome do Caso de Uso	Manter medicamento (RF03)
Autor Principal	Paciente
Resumo	Esse caso de uso descreve o procedimento que o paciente faz para cadastrar e gerenciar os medicamentos no sistema.
Pré-condições	Para manter medicamentos é preciso fazer login e conectar caixa no sistema
Pós-condições	Após manter medicamentos, é feito a detecção da retirada de medicamentos e o registro de medicamentos no sistema
Cenário Principal	
Ações do Ator	Ações do Sistema
1. Acessar página de cadastro de medicamentos	
2. Selecionar compartimento da caixa	
	3. Exibir formulários
4. Cadastrar as informações	
	5. Salvar as informações no sistema.
6. Selecionar edição	
	7. Exibir formulários
8. Editar informações	
	9. Salvar as informações no sistema.
10. Selecionar opção excluir	
	11. Excluir informações do sistema
Cenário Alternativo - Informação invalida	
Ações do Ator	Ações do Sistema

	1. Exibe mensagem de informação inválida
--	--

Fonte: Dos próprios autores, 2024.

Quadro 9 - Documentação Caso de Uso (Cadastrar caixa)

Nome do Caso de Uso	Cadastrar caixa (RF04)
Autor Principal	Paciente
Autor Secundário:	Sistema PillTrack
Resumo	Este caso de uso descreve como o paciente cadastra uma nova caixa no sistema PillTrack
Pré-Condições	O paciente deve estar autenticado no sistema
Cenário Principal	
Ações do Ator	Ações do Sistema
1. Acessa a função “Cadastrar caixa”	
	2. Exibe um formulário solicitando informações da caixa
3. Preenche as informações solicitadas pelo sistema	
4. Confirma o cadastro	
	5. Valida as informações e cobra a nova caixa no banco de dados
Cenário Alternativo – Dados incorretos	
Ações do Ator	Ações do Sistema
	1. Informa que os dados estão incorretos
	2. Pede para que o usuário preencha o formulário novamente

Fonte: Dos próprios autores, 2024.

Quadro 10 - Documentação Caso de Uso (Cadastrar paciente)

Nome do Caso de Uso	Cadastrar Paciente (RF05)
Autor Principal	Paciente
Autor Secundário	Sistema PillTrack
Resumo	Este caso de uso descreve o processo para a realização do cadastro de um paciente
Cenário Principal	
Ações do Ator	Ações do Sistema
1. Acessa a opção “Cadastrar Paciente”	
	2. Exibe um formulário com campos obrigatórios
3. Preenche as informações solicitadas pelo sistema	
4. Confirma o cadastro	
	5. Valida as informações e registra o paciente no banco de dados
Cenário Alternativo – Dados incorretos	
Ações do Ator	Ações do Sistema
	1. Informa que os dados estão incorretos
	2. Pede para que o usuário preencha o formulário novamente

Fonte: Dos próprios autores, 2024.

Quadro 11 - Documentação Caso de Uso (Vincular responsável)

Nome do Caso de Uso	Vincular Responsável (RF06)
Autor Principal	Paciente
Resumo	O paciente pode vincular um responsável para acompanhar o tratamento.

Pré-condições	O paciente deve estar cadastrado e logado no sistema. O responsável deve estar cadastrado no sistema.
Pós-condições	O responsável é vinculado ao paciente e pode acessar os relatórios de tratamento.
Cenário Principal	
Ações do Ator	Ações do Sistema
1. O paciente faz login no sistema.	
	2. O sistema verifica as credenciais e concede acesso.
3. O paciente seleciona "Vincular Responsável".	
	4. O sistema exibe um campo para inserir o CPF do responsável.
5. O paciente insere o CPF do responsável.	
	6. O sistema valida o CPF e vincula o responsável ao paciente.
Cenário Alternativo	
Ações do Ator	Ações do Sistema
1.O paciente insere um CPF de responsável não cadastrado.	
	2.O sistema exibe uma mensagem solicitando que o responsável se cadastre primeiro.

Fonte: Dos próprios autores, 2024.

Quadro 12 - Documentação Caso de Uso (Visualizar médico)

Nome do Caso de Uso	Vincular médico (RF07)
Autor Principal	Paciente

Resumo	O paciente pode vincular um médico para supervisionar o tratamento através de relatórios.
Pré-condições	O paciente deve estar cadastrado e logado no sistema. O médico deve estar cadastrado no sistema.
Pós-condições	O médico é vinculado ao paciente e pode acessar os relatórios de tratamento compartilhados com ele.
Cenário Principal	
Ações do Ator	Ações do Sistema
1. O paciente faz login no sistema.	
	2. O sistema verifica as credenciais e concede acesso.
3. O paciente acessa seu perfil e seleciona "Vincular Médico".	
	4. O sistema exibe um campo para inserir o CRM do médico.
5. O paciente insere o CRM do médico.	
	6. O sistema valida o CRM e vincula o médico ao paciente.
Cenário Alternativo	
Ações do Ator	Ações do Sistema
1. O paciente insere um ID de médico não cadastrado.	
	2. O sistema exibe uma mensagem solicitando que o médico se cadastre primeiro.

Fonte: Dos próprios autores, 2024.

Documentação dos Casos de Uso Responsável

Quadro 13 - Documentação Caso de Uso (Cadastrar responsável)

Nome do Caso de Uso	Cadastrar responsável (RF01)
Autor Principal	Responsável
Autor Secundário	Sistema PillTrack
Resumo	Este caso de uso descreve o processo para a realização do cadastro de um responsável
Cenário Principal	
Ações do Ator	Ações do Sistema
1. Acessa a opção “Cadastrar Responsável”	
	2. Exibe um formulário com campos obrigatórios
3. Preenche as informações solicitadas pelo sistema	
4. Confirma o cadastro	
	5. Valida as informações e registra o responsável no banco de dados
Cenário Alternativo – Dados incorretos	
Ações do Ator	Ações do Sistema
	1. Informa que os dados estão incorretos
	2. Pede para que o usuário preencha o formulário novamente

Fonte: Dos próprios autores, 2024.

Documentação dos Casos de Uso Médico

Quadro 14 - Documentação Caso de Uso (Cadastrar médico)

Nome do Caso de Uso	Cadastrar médico (RF01)
Autor Principal	Médico
Autor Secundário	Sistema PillTrack

Resumo	Este caso de uso descreve o processo para a realização do cadastro de um médico
Cenário Principal	
Ações do Ator	Ações do Sistema
1. Acessa a opção “Cadastrar Médico”	
	2. Exibe um formulário com campos obrigatórios
3. Preenche as informações solicitadas pelo sistema	
4. Confirma o cadastro	
	5. Valida as informações e registra o paciente no banco de dados
Cenário Alternativo – Dados incorretos	
Ações do Ator	Ações do Sistema
	1. Informa que os dados estão incorretos
	2. Pede para que o usuário preencha o formulário novamente

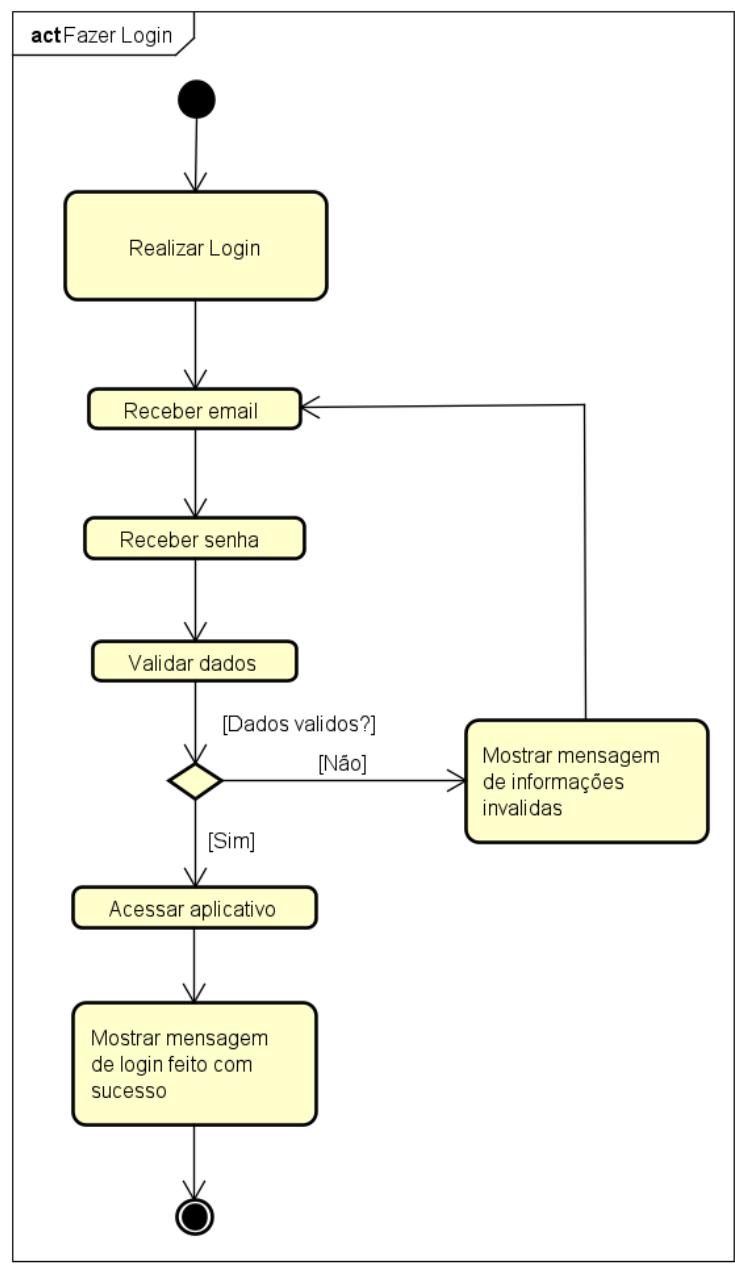
Fonte: Dos próprios autores, 2024.

3.3 Diagrama de Atividade

Os Diagramas de Atividade são usados para detalhar os estágios sequenciais de cada processo do sistema. Todos os diagramas a seguir mostraram os fluxos de ações, detalhadamente através de cada funcionalidade do sistema PillTrack.

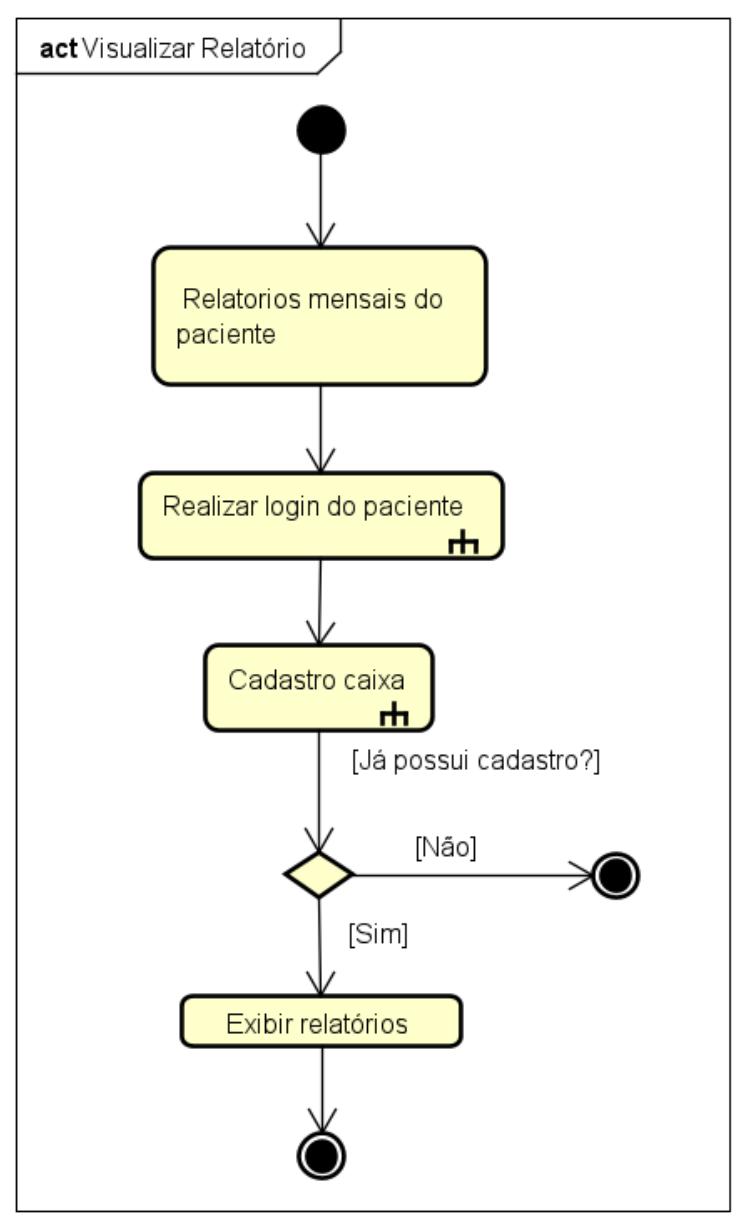
Diagrama de Atividade Atores Genéricos (Todos):

Figura 34 - Diagrama de Atividade: Fazer Login



Fonte: Dos próprios autores, 2024.

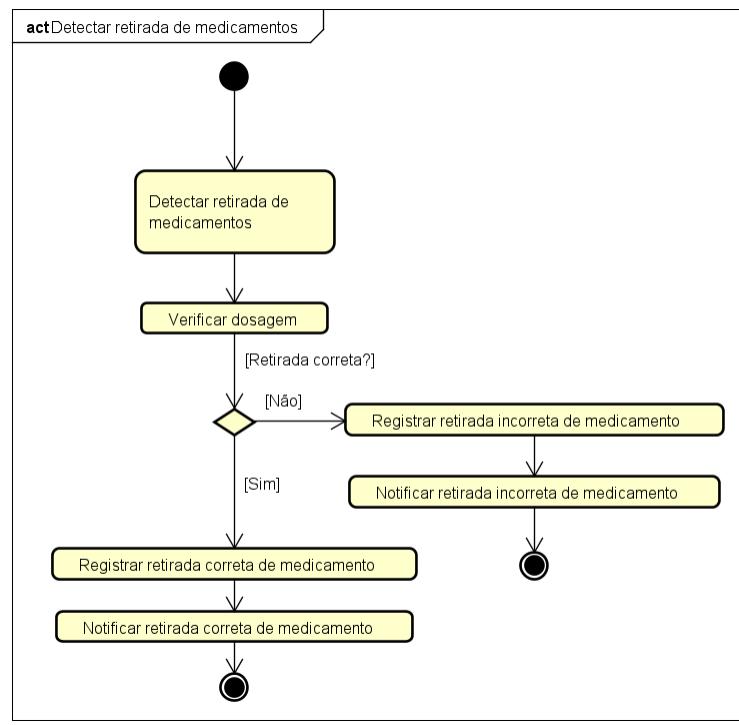
Figura 35 - Diagrama de Atividade: Visualizar relatórios



Fonte: Dos próprios autores, 2024.

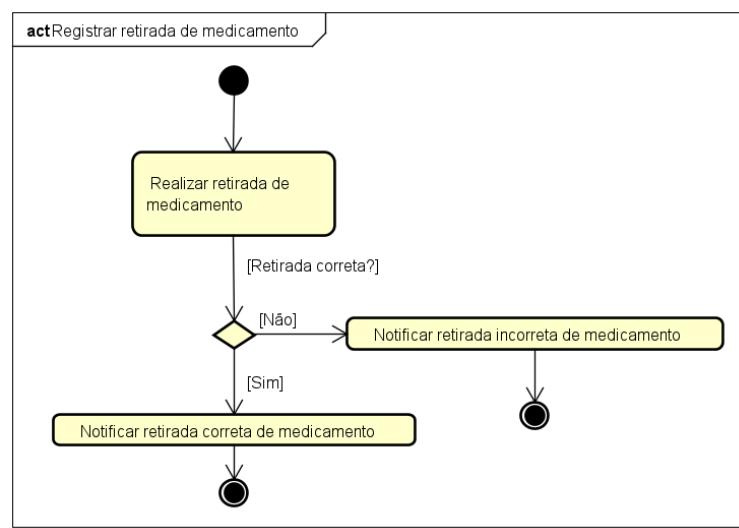
Diagrama de Atividade Sistema PillTrack

Figura 36 - Diagrama de Atividade: Detectar retirada de medicamentos



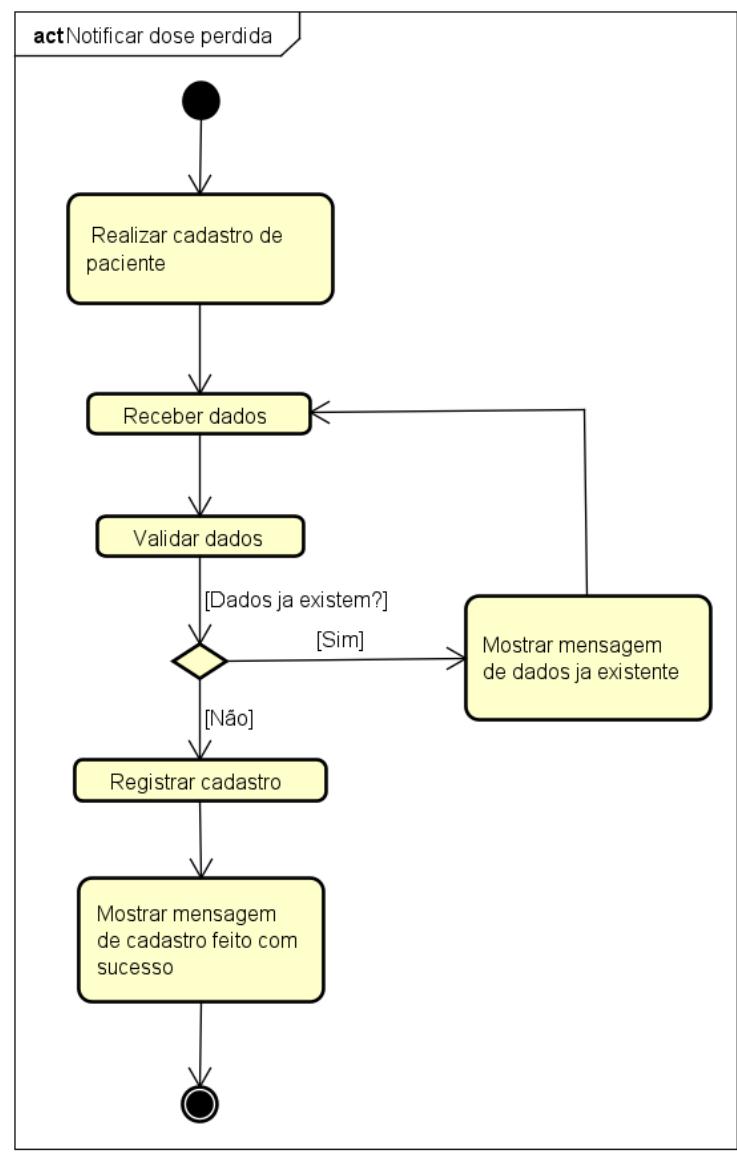
Fonte: Dos prop rios autores, 2024.

Figura 37 - Diagrama de Atividade: Registrar retirada de medicamentos



Fonte: Dos prop rios autores, 2024.

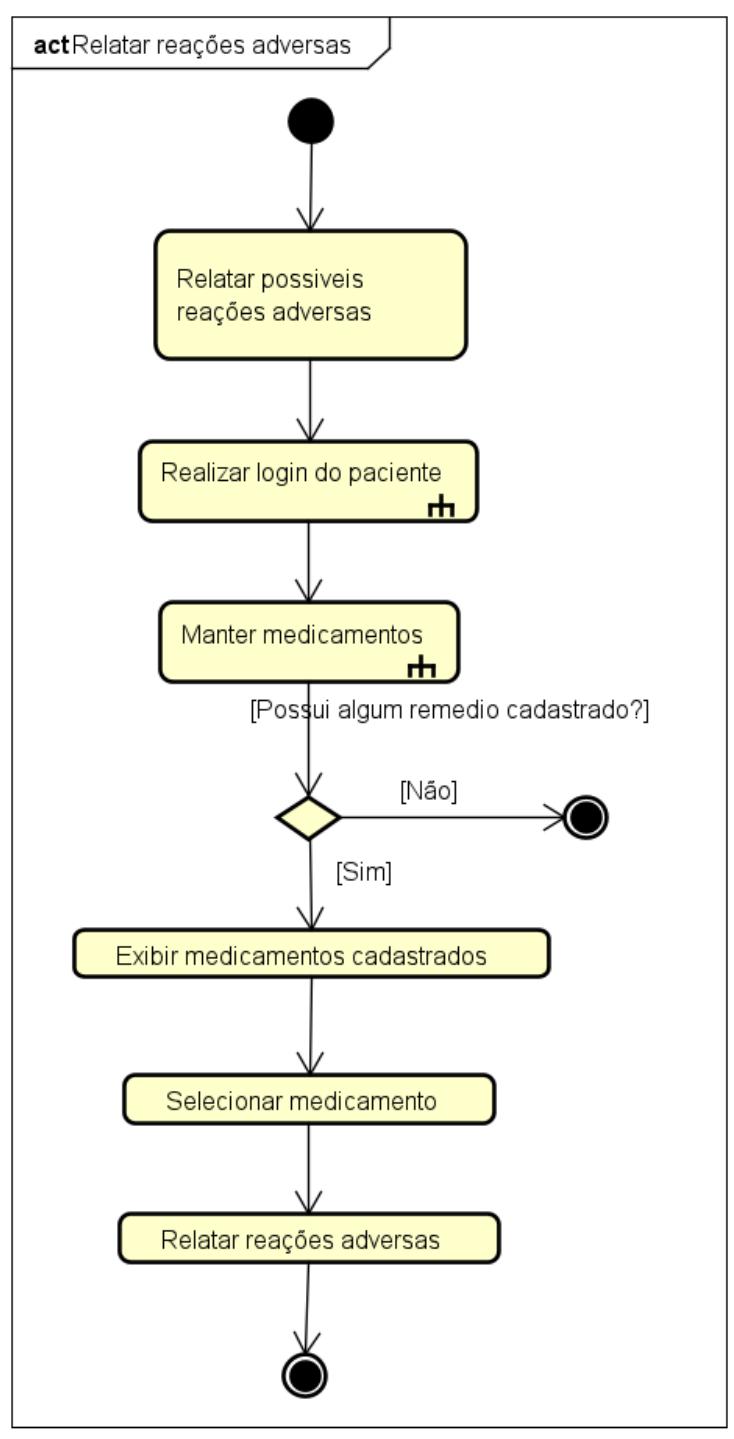
Figura 38 - Diagrama de Atividade: Notificar dose perdida



Fonte: Dos próprios autores, 2024.

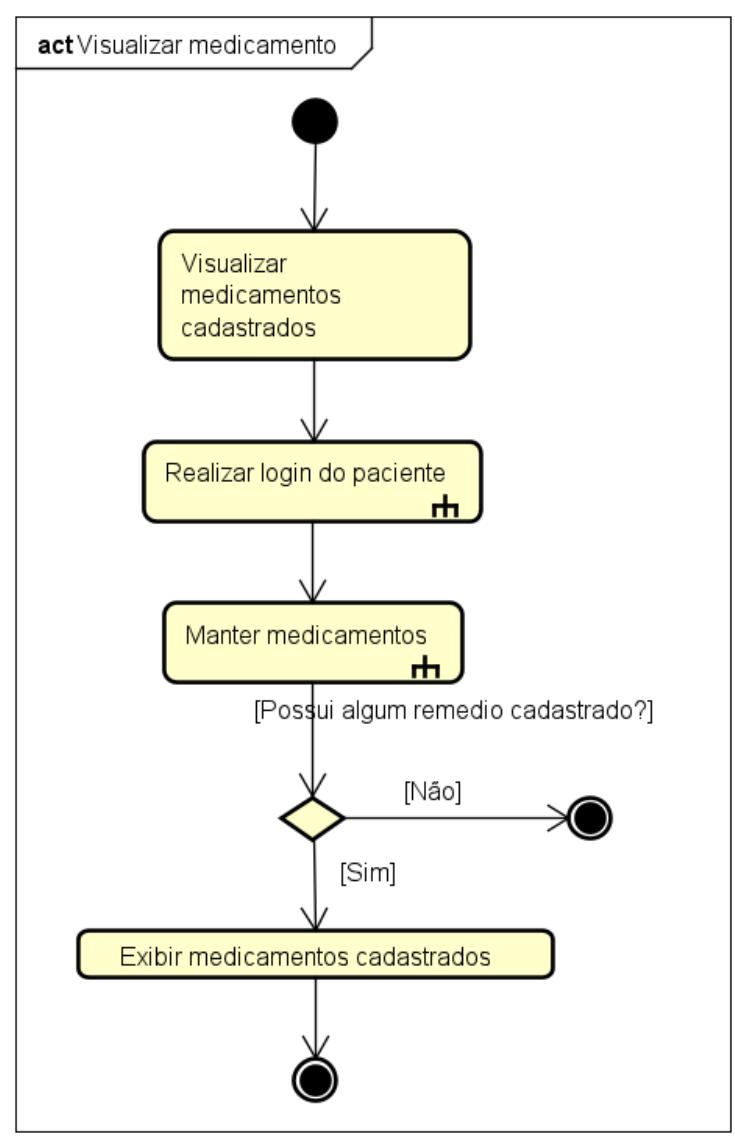
Diagrama de Atividade Paciente

Figura 39 - Diagrama de Atividade: Relatar reação adversa



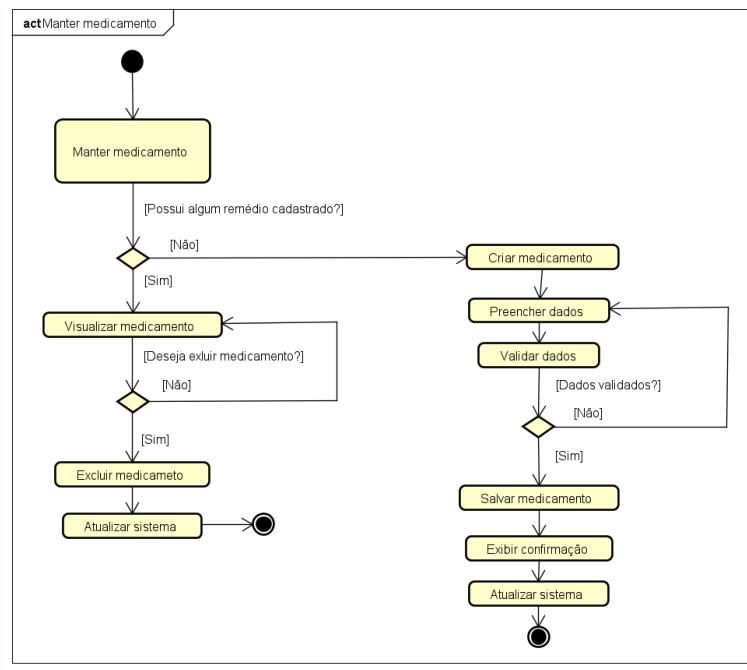
Fonte: Dos próprios autores, 2024.

Figura 40 - Diagrama de Atividade: Visualizar medicamento



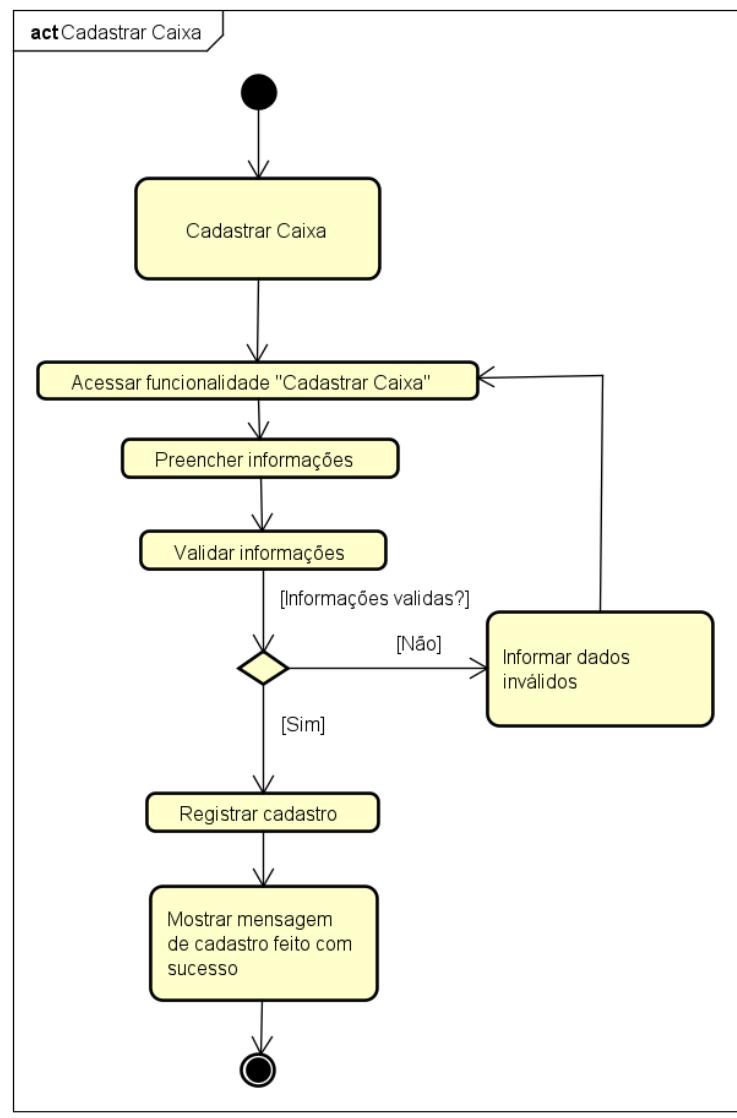
Fonte: Dos próprios autores, 2024.

Figura 41 - Diagrama de Atividade: Manter medicamento



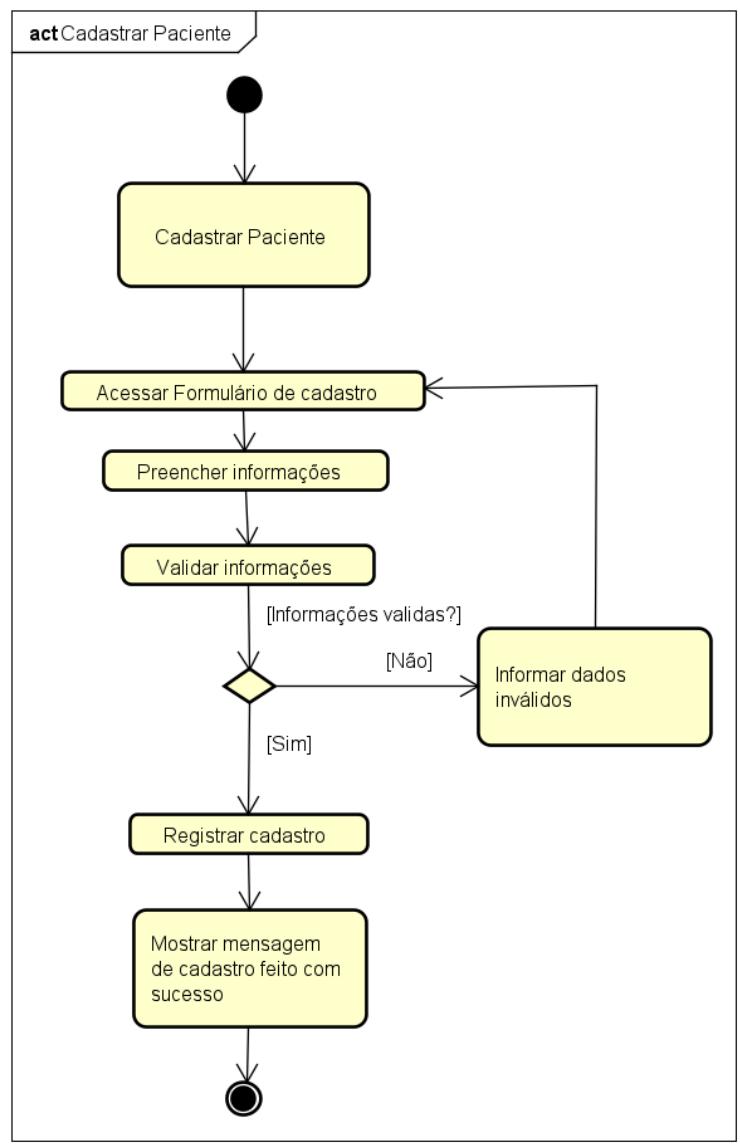
Fonte: Dos próprios autores, 2024.

Figura 42 - Diagrama de Atividade: Cadastrar caixa



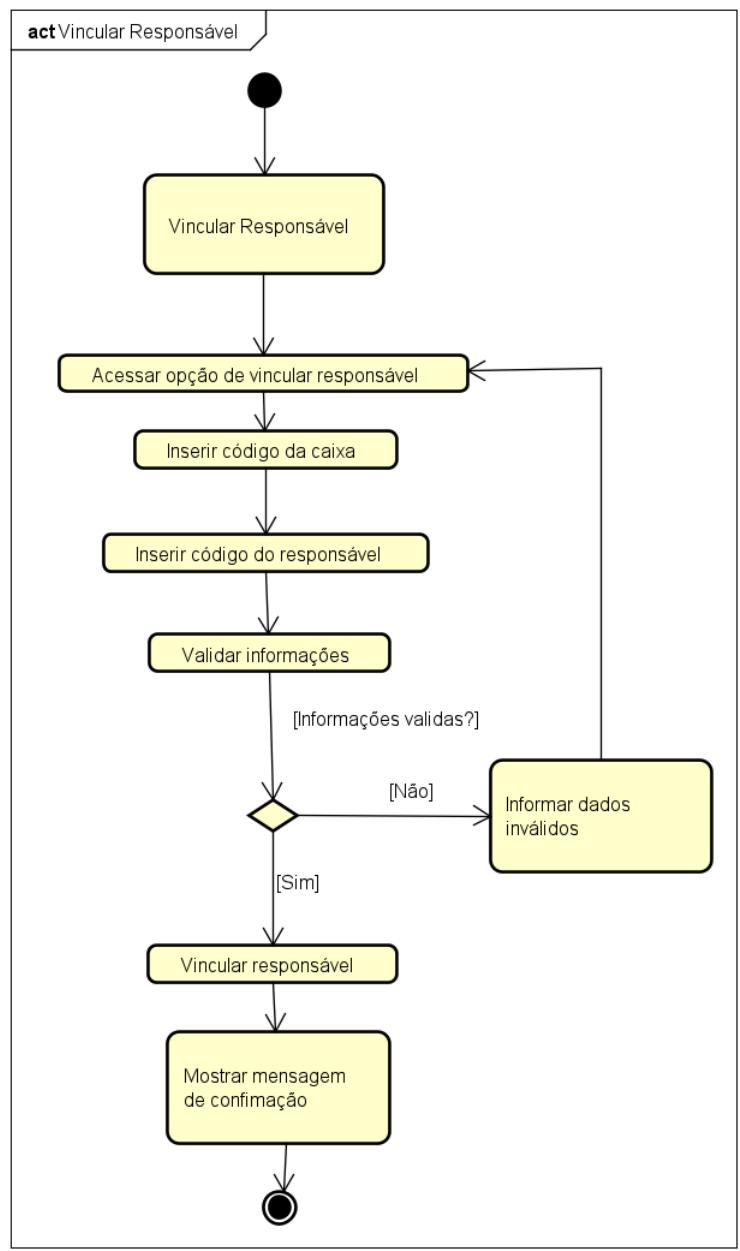
Fonte: Dos próprios autores, 2024.

Figura 43 - Diagrama de Atividade: Cadastrar paciente



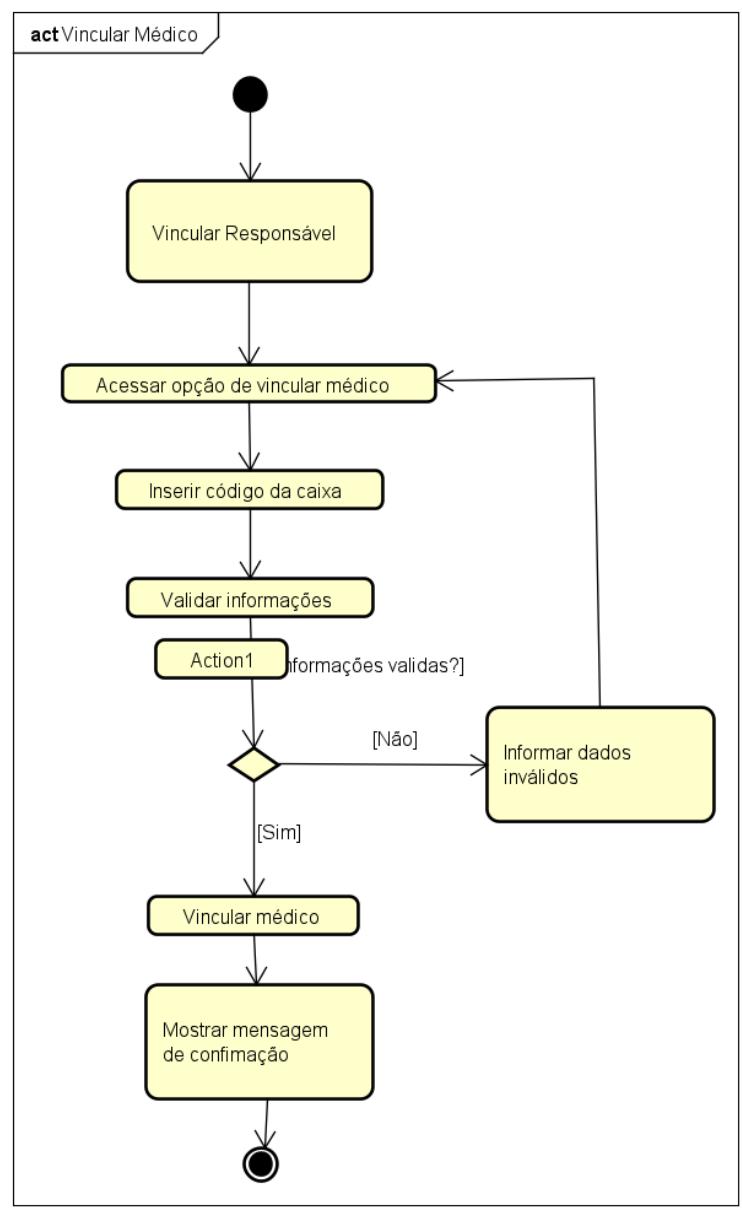
Fonte: Dos próprios autores, 2024.

Figura 44 - Diagrama de Atividade: Vincular responsável



Fonte: Dos próprios autores, 2024.

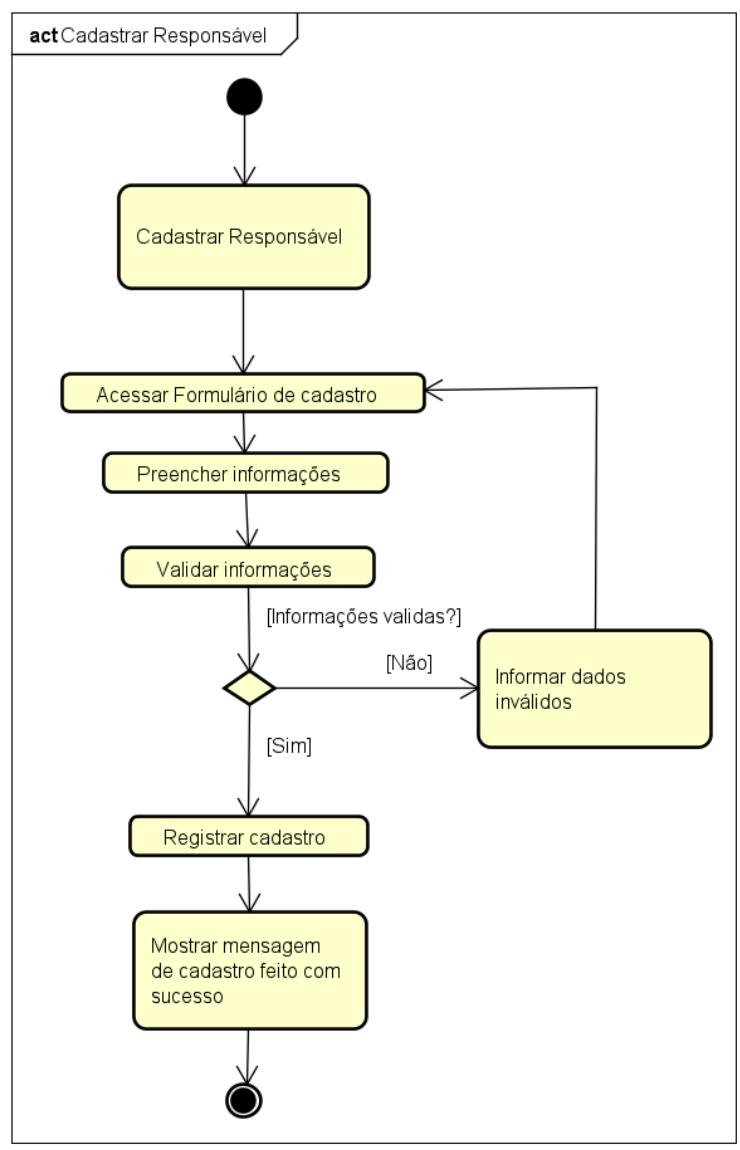
Figura 45 - Diagrama de Atividade: Vincular médico



Fonte: Dos próprios autores, 2024.

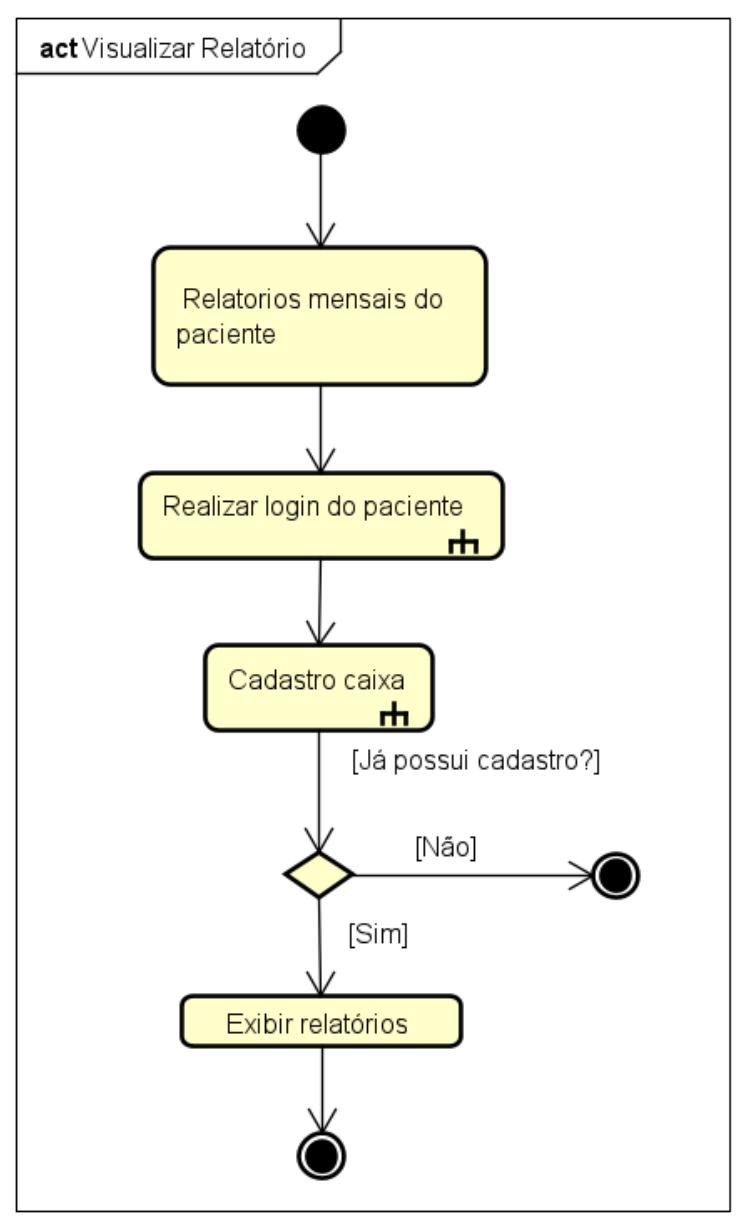
Diagrama de Atividade Responsável

Figura 46 - Diagrama de Atividade: Cadastrar responsável



Fonte: Dos próprios autores, 2024.

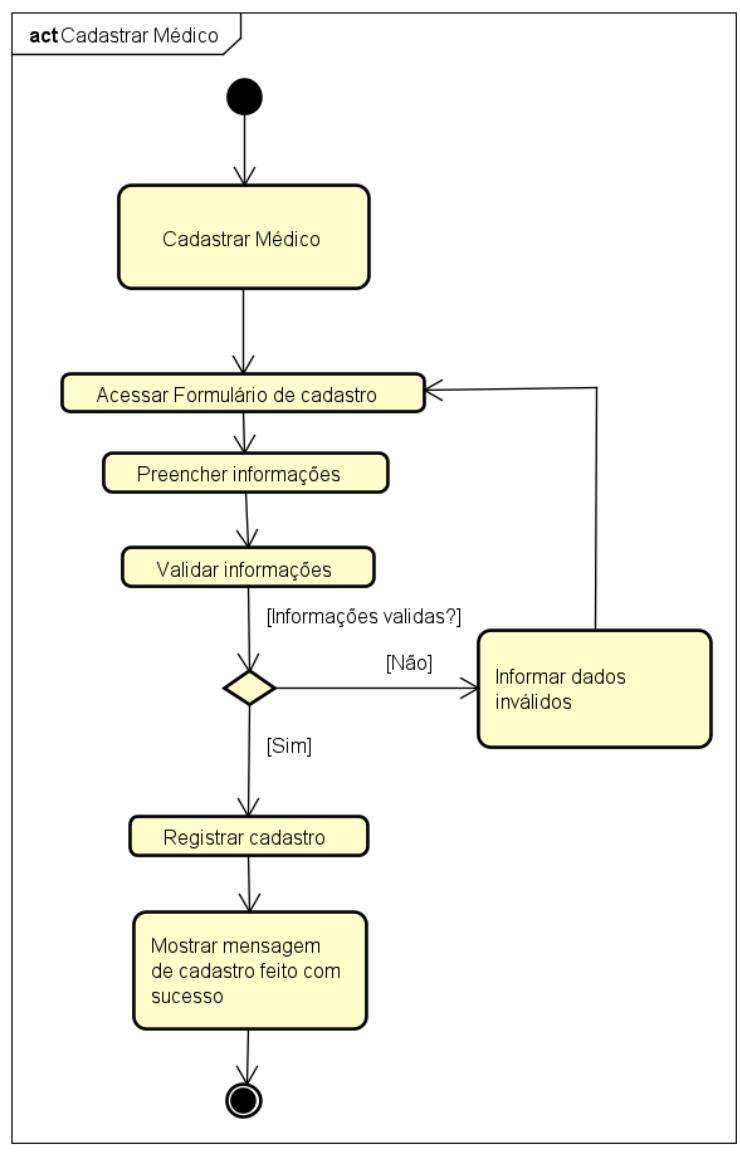
Figura 47 - Diagrama de Atividade: Visualizar relatórios



Fonte: Dos próprios autores, 2024.

Diagrama de Atividade Médico

Figura 48 - Diagrama de Atividade: Cadastrar médico



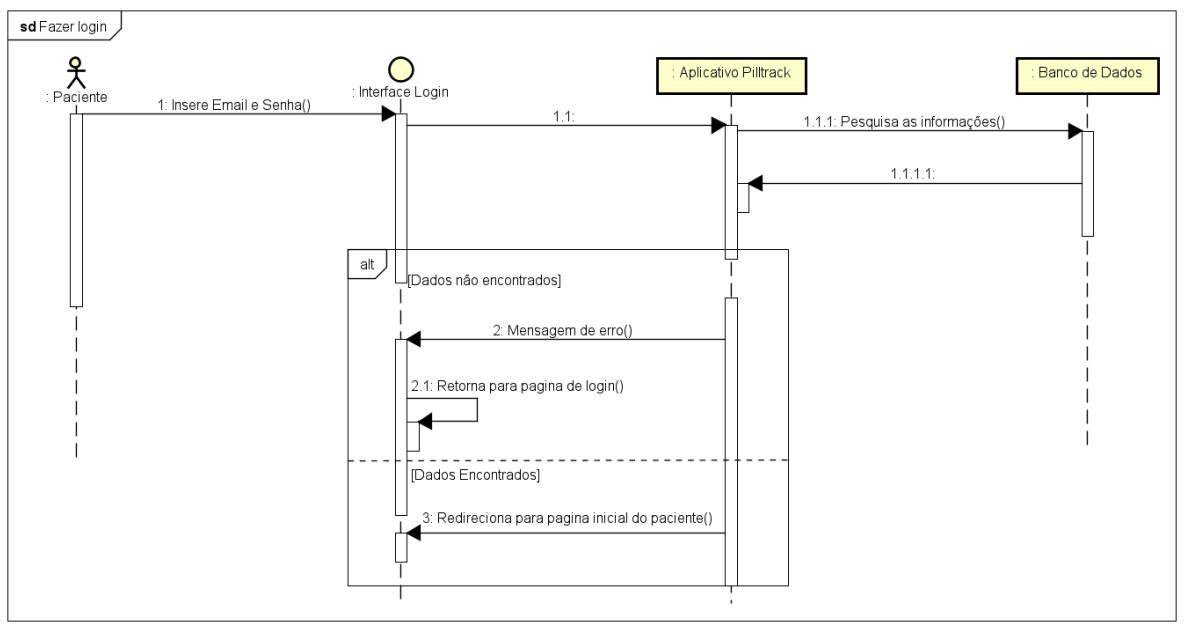
Fonte: Dos próprios autores, 2024.

3.4 Diagrama de Sequência

Agora, abordaremos os Diagramas de Sequência que são utilizados, principalmente, para o usuário ter uma maior noção das interações das execuções de cada função do sistema. Neste diagrama, é nítida a participação minuciosa dos atores e a classificação de cada interação das execuções de uma ação do sistema.

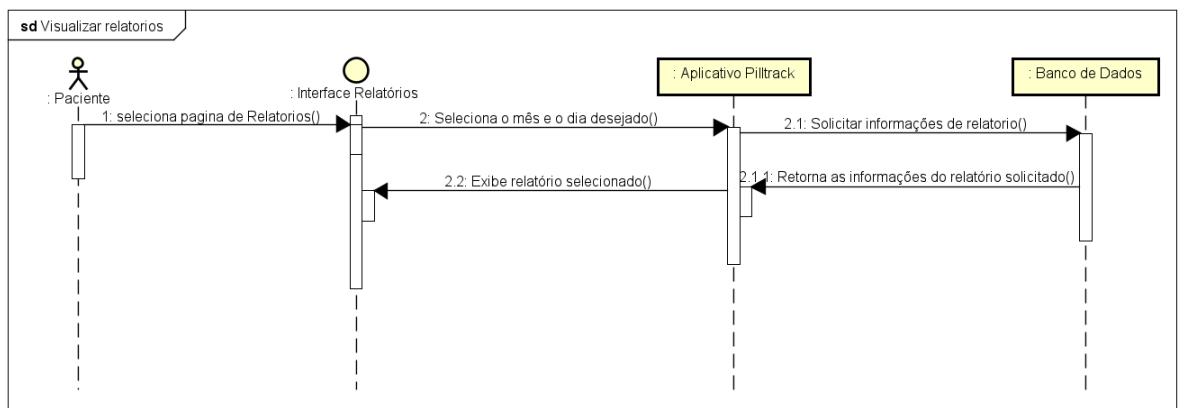
Diagrama de Sequência Atores Genéricos (Todos)

Figura 49- Diagrama de Sequência: Fazer Login



Fonte: Dos próprios autores, 2024.

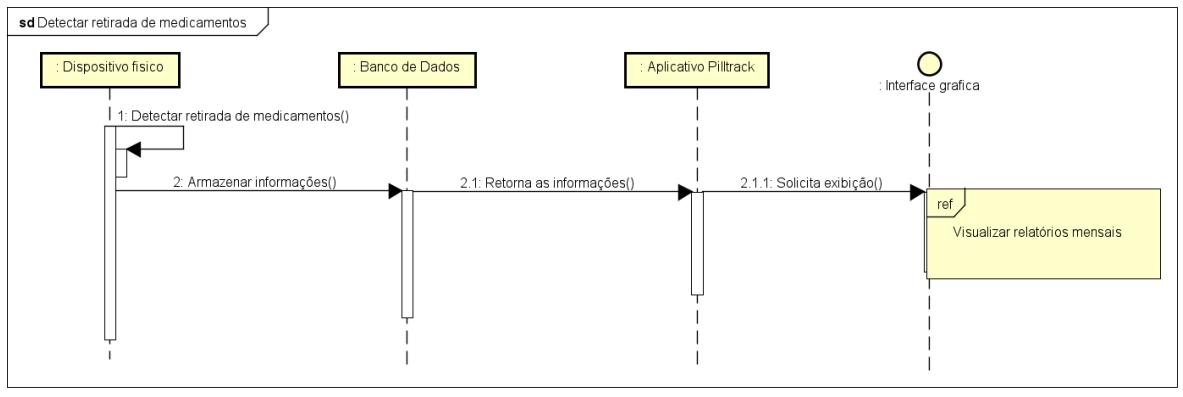
Figura 50 - Diagrama de Sequência: Visualizar relatórios



Fonte: Dos próprios autores, 2024.

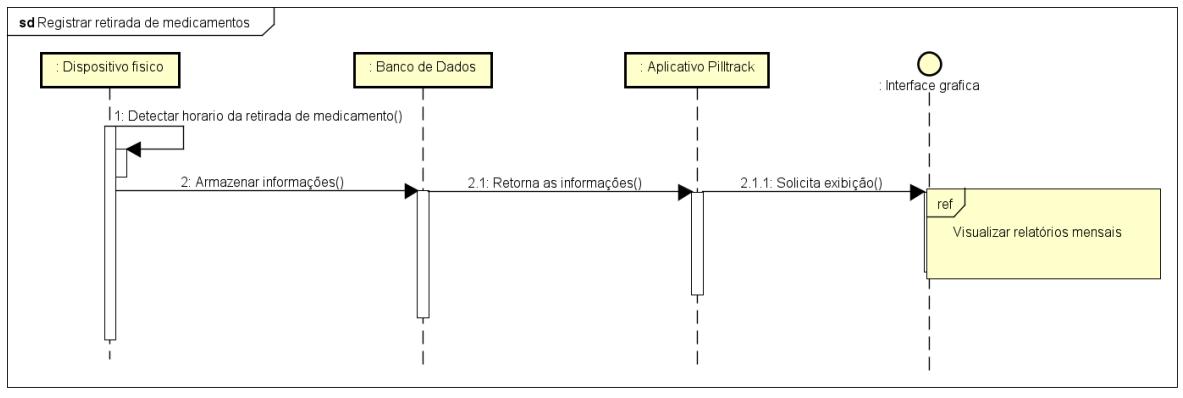
Diagrama de Sequência Sistema PillTrack

Figura 51 - Diagrama de Sequência: Detectar retirada de medicamentos



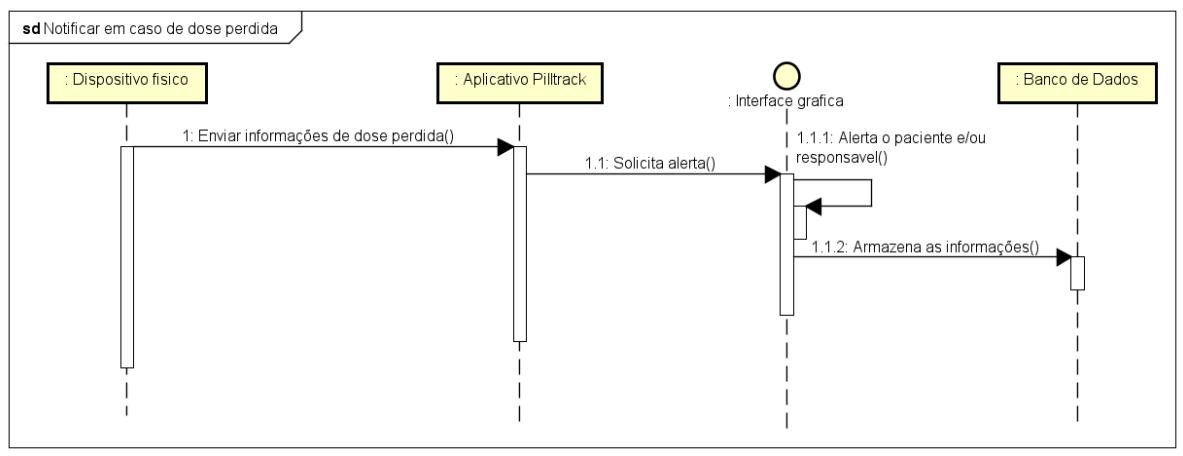
Fonte: Dos próprios autores, 2024.

Figura 52 - Diagrama de Sequência: Registrar retirada de medicamentos



Fonte: Dos próprios autores, 2024.

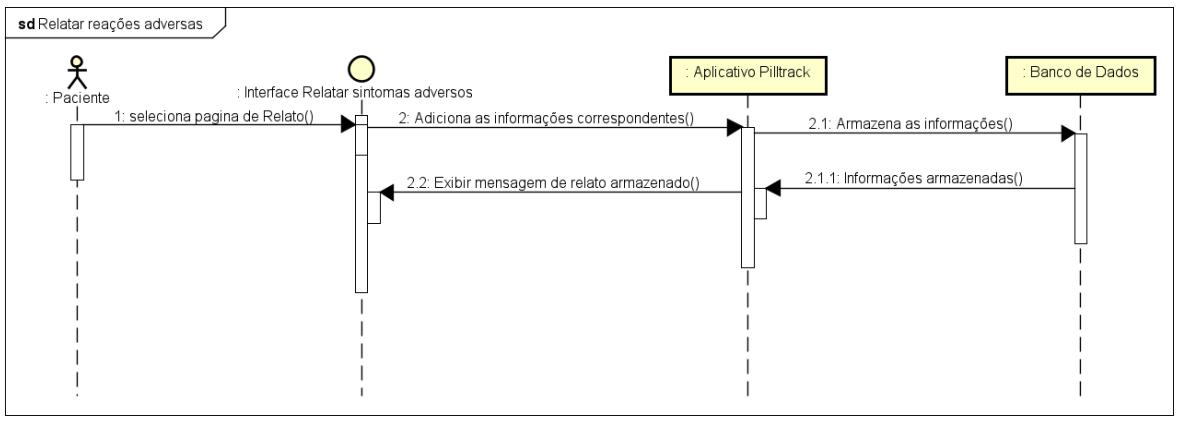
Figura 53 - Diagrama de Sequência: Notificar dose perdida



Fonte: Dos próprios autores, 2024.

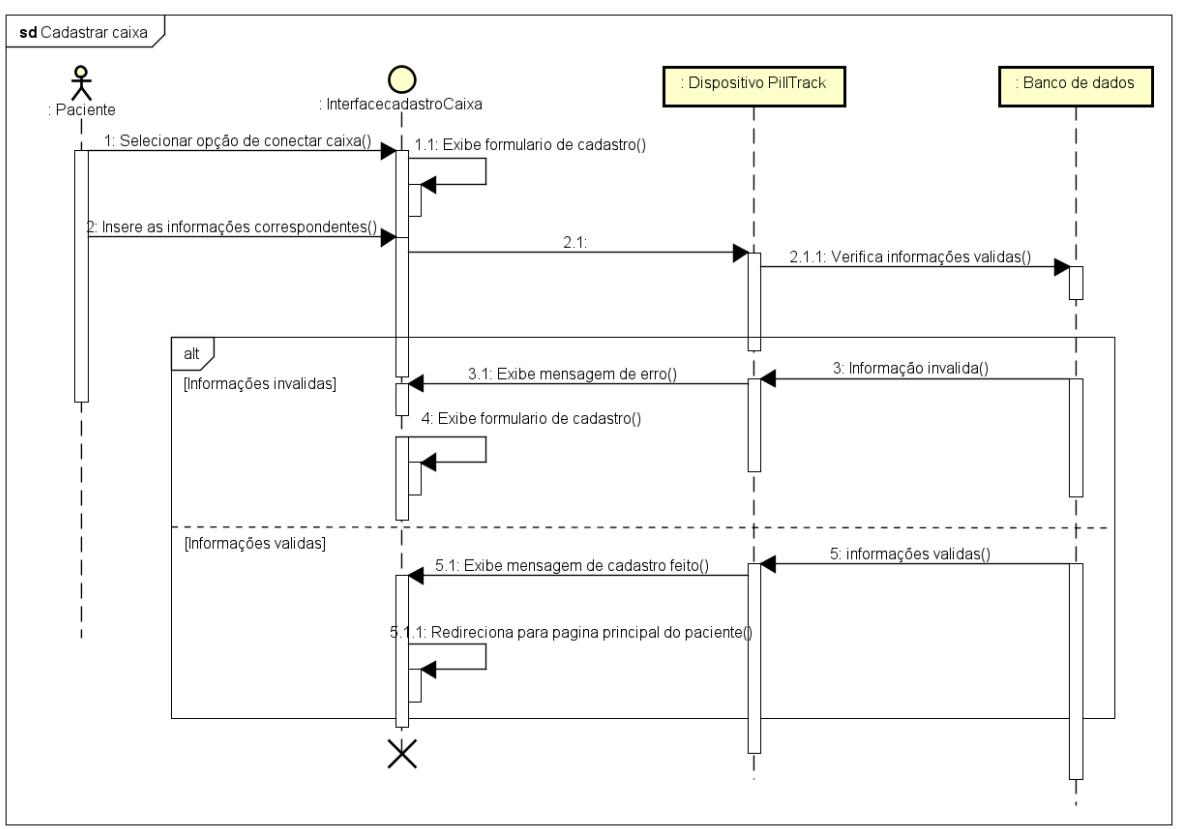
Diagrama de Sequência Paciente

Figura 54 - Relatar reação adversa



Fonte: Dos próprios autores, 2024.

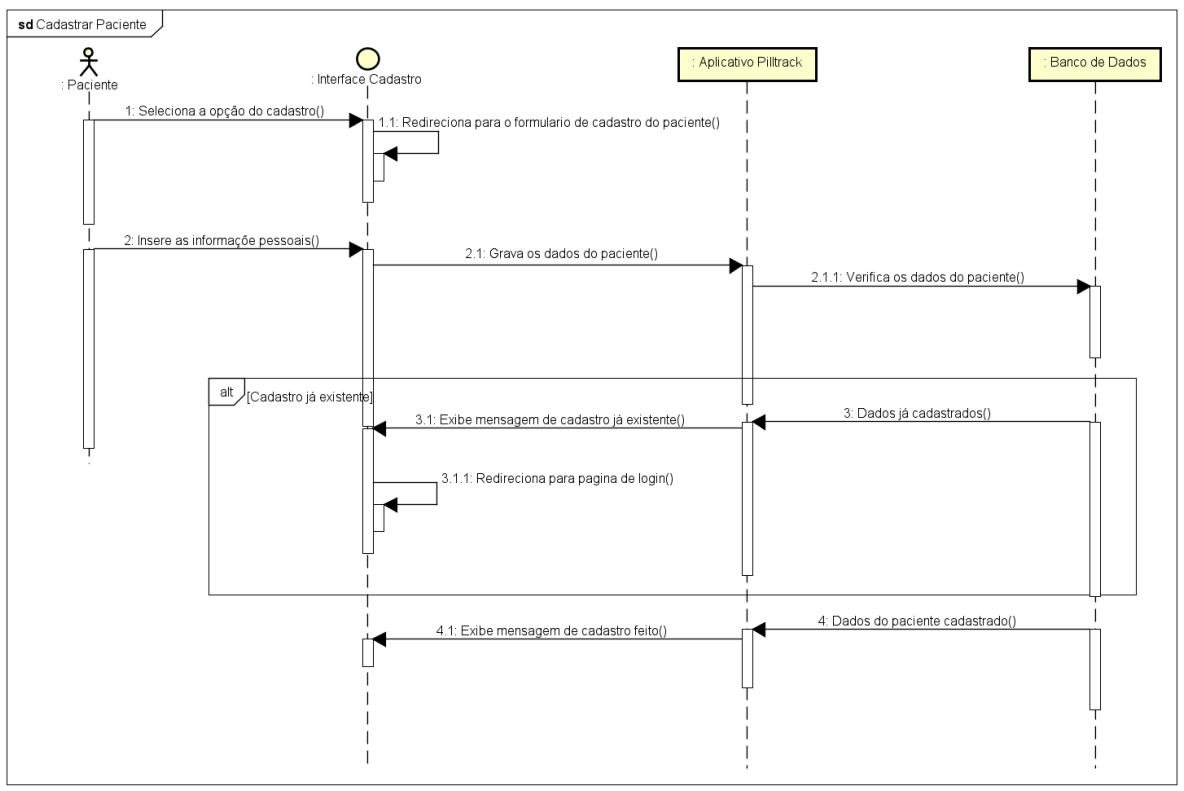
Figura 55 - Diagrama de Sequência: Cadastrar caixa



Fonte: Dos próprios autores, 2024.

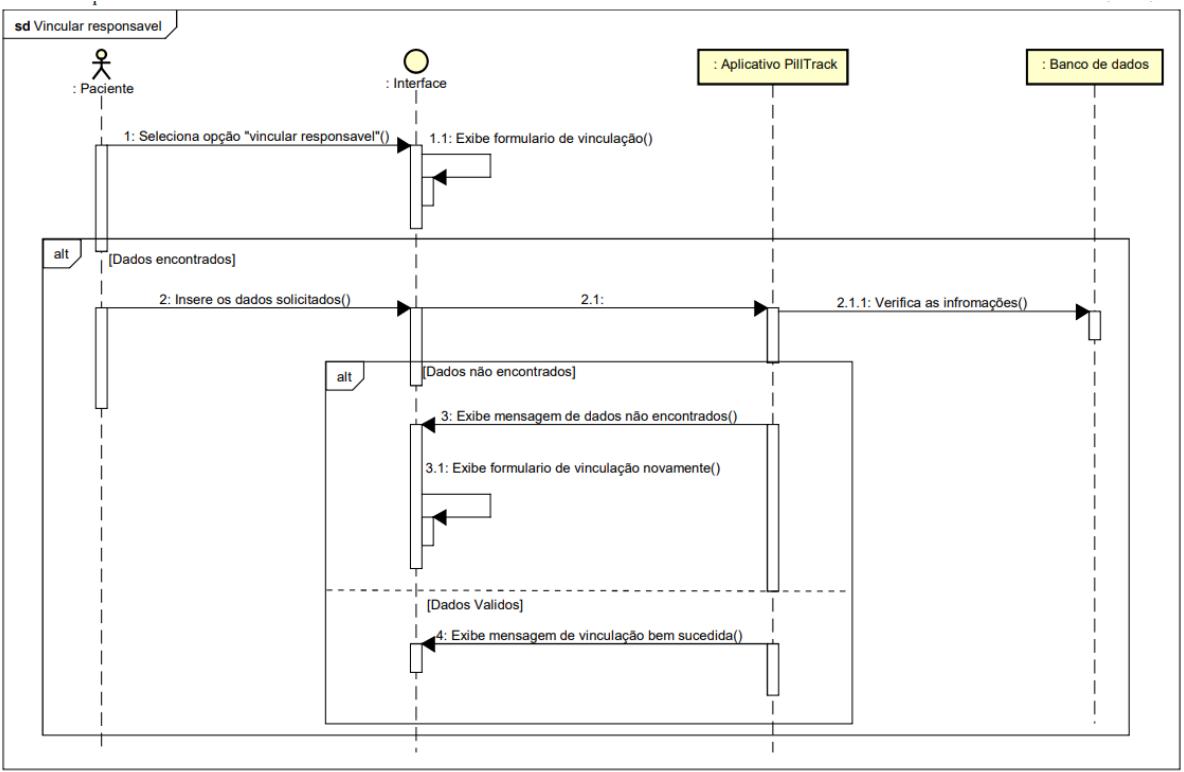
Figura 56 - Diagrama de Sequência: Cadastrar paciente

Fonte: Dos próprios autores, 2024.



Fonte: Dos próprios autores, 2024.

Figura 57 - Diagrama de Sequência: Vincular responsável

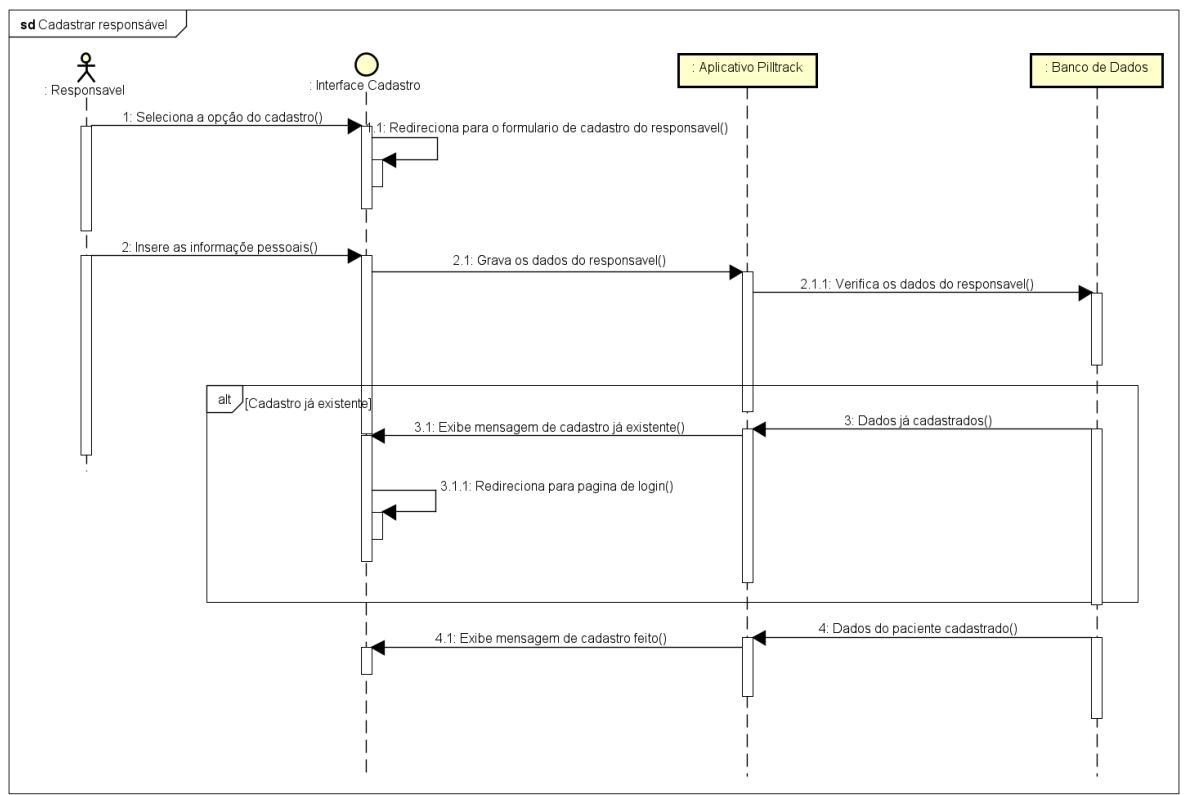


Fonte: Dos próprios autores, 2024.

Figura 58 - Diagrama de Sequência: Vincular médico
 Fonte: Dos próprios autores, 2024.

Diagrama de Sequência Responsável

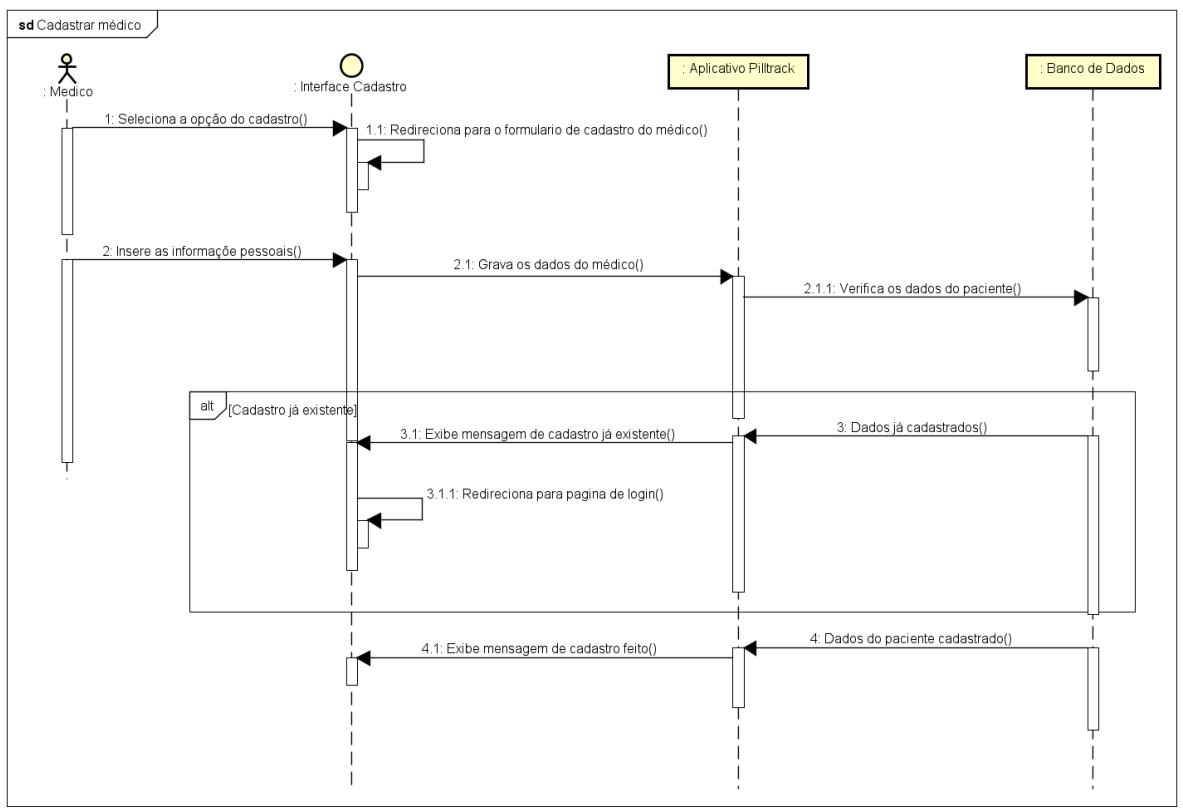
Figura 59 - Diagrama de Sequência: Cadastrar responsável



Fonte: Dos próprios autores, 2024.

Diagrama de Sequência Médico

Figura 60 - Diagrama de Sequência: Cadastrar médico



Fonte: Dos próprios autores, 2024.

3.5 Diagrama de Máquina de Estados

Diagramas de Máquina de Estados incluem transições, estados e eventos que causam mudanças do objeto em resposta a eventos específicos, informando como ele se move em diferentes situações. É considerado muito útil, para indicar sistemas interativos e principalmente como eles reagem a diferentes fatores desencadeantes.

Durante este capítulo, apresentaremos os diagramas de Máquina de Estados do sistema PillTrack. Ao todo, 13 diagramas foram necessários.

Diagrama de Máquina de Estados Atores Genéricos (Todos)

Figura 61 - Diagrama de Máquina de Estados: Fazer Login

Figura 62 - Diagrama de Máquina de Estados: Visualizar relatórios sistema

Diagrama de Máquina de Estados Sistema PillTrack

Figura 63 - Diagrama de Máquina de Estados: Detectar retirada de medicamentos

Figura 64 - Diagrama de Máquina de Estados: Registrar retirada de medicamentos

Figura 65 - Diagrama de Máquina de Estados: Notificar dose perdida

Diagrama de Máquina de Estados Paciente

Figura 66 - Diagrama de Máquina de Estados: Relatar reação adversa

Figura 67 - Diagrama de Máquina de Estados: Visualizar medicamento

Figura 68 - Diagrama de Máquina de Estados: Cadastrar caixa

Figura 69 - Diagrama de Máquina de Estados: Cadastrar paciente

Figura 70 - Diagrama de Máquina de Estados: Vincular médico

Diagrama de Máquina de Estados Responsável

Figura 71 - Diagrama de Máquina de Estados: Cadastrar responsável

Figura 72 - Diagrama de Máquina de Estados: Visualizar relatórios

Diagrama de Máquina de Estados Médico

Figura 73 - Diagrama de Máquina de Estados: Cadastrar médico

3.6 Montagem do circuito do dispositivo

3.7 Prototipação das páginas da aplicação

Nesta seção serão apresentados os *wireframes* de baixa e alta fidelidade correspondente aos componentes gráficos do *website* e do aplicativo do sistema.

É importante ressaltar que o website será unicamente informativo, aonde será permitido que o usuário possa compreender o funcionamento do sistema. Adicionalmente, através do site, será possível entrar em contato diretamente com desenvolvedoras através das redes sociais. Já o aplicativo estará interligado com o dispositivo físico, concedendo ao usuário realizar o cadastro e conectar-se diretamente com o sistema.

Apresentaremos agora, as interfaces gráficas do website, seguido do *wireframe* de baixa e depois o de alta:

Figura 74 - *Wireframe* de Baixa Fidelidade Site



Fonte: Dos próprios autores, 2024.

Figura 75 - Wireframe de Alta Fidelidade Site

 PillTrack

Home Sobre Como Funciona? Desenvolvedoras

PillTrack: IoT para gerenciamento de medicamentos

A PillTrack é um sistema inteligente com notificações e controle de estoque para melhorar a saúde dos idosos. Um aplicativo facilita o acesso à informações por cuidadores e médicos.

[Leia mais sobre](#)



Sobre à PillTrack

#1 Aplicativo para gestão de medicamentos

A PillTrack é um sistema inteligente com notificações e controle de estoque para melhorar a saúde dos idosos. Um aplicativo facilita o acesso à informações por cuidadores e médicos.

[Recursos](#)



Como Funciona?

-  **Adquira o Dispositivo PillTrack**
Entre em contato conosco para adquirir o caixa de medicamentos inteligente.
-  **Instale o Aplicativo**
Baixe e instale o aplicativo PillTrack em um dispositivo móvel (disponível para Android e iOS).
-  **Crie um perfil**
Crie uma conta no aplicativo. Insira informações básicas do paciente. Configure o caixa conectando-o ao aplicativo e organize os medicamentos nos compartimentos desejados.
-  **Configure a medicação**
Insira os detalhes dos medicamentos, programe horários e doses conforme a prescrição médica, e comece a receber notificações e alertas.

Desenvolvedoras

Criadoras por trás do sistema PillTrack



Desirée Constantino de Almeida Barboza
Desenvolvedora
[LinkedIn](#) [Instagram](#) [GitHub](#)



Giovana Marsiglio Rodrigues
Desenvolvedora
[LinkedIn](#) [Instagram](#) [GitHub](#)



Isabelle Gomes de Souza Andrade
Desenvolvedora
[LinkedIn](#) [Instagram](#) [GitHub](#)

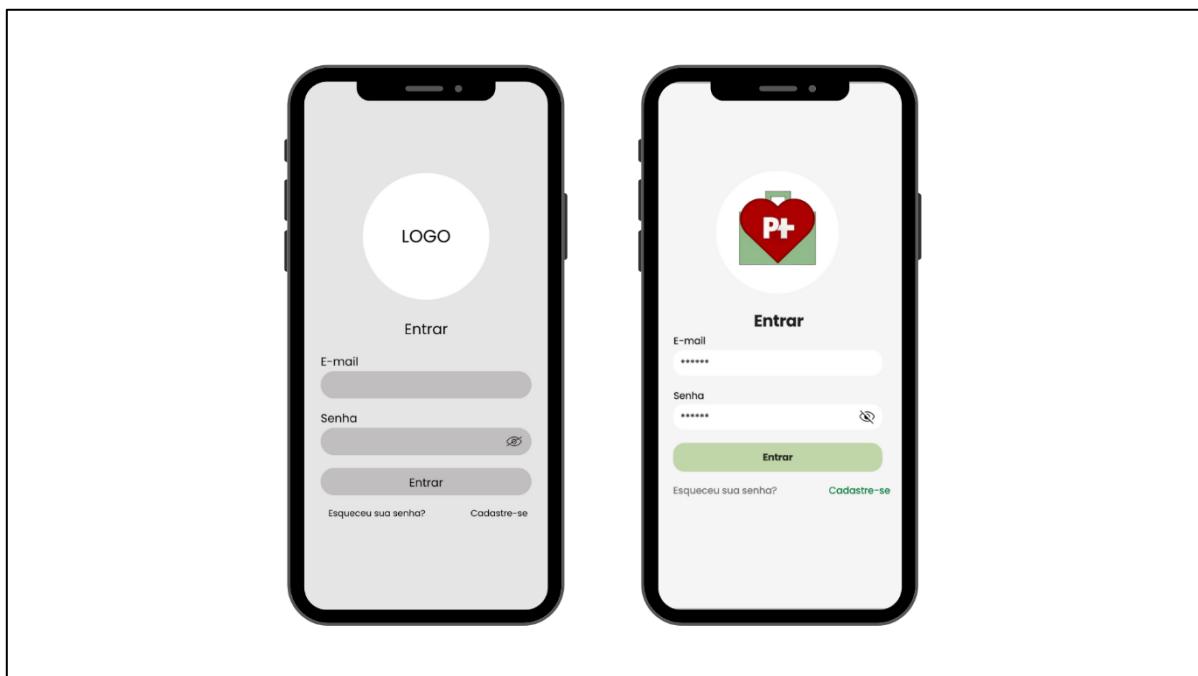
Comunicado informativo O PillTrack, um projeto escolar, ajuda na administração de medicamentos, fornecendo notificações de lembrete, registro de administração, controle de estoque e acesso remoto para cuidadores e médicos.

Contato  pilltrack@gmail.com

Fonte: Dos próprios autores, 2024.

Agora, serão exibidos os *wireframes* de baixa e alta fidelidade do aplicativo, um ao lado do outro para melhor visualização. A primeira tela a ser apresentada é a tela de “Login”, através dela será possível se cadastrar, redefinir senha e entrar no sistema por e-mail e senha.

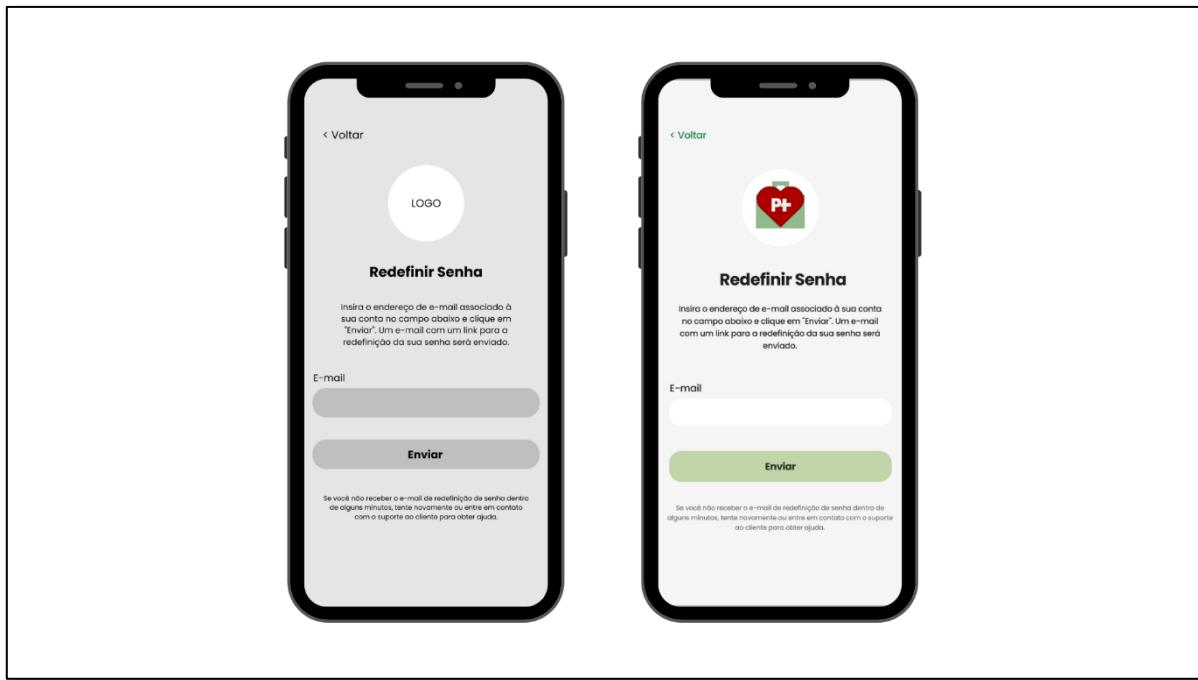
Figura 76 - Wireframe de Baixa e Alta Fidelidade: Login



Fonte: Dos próprios autores, 2024.

Em seguida, a tela de “Esqueceu a senha”, através dela o usuário poderá redefinir a própria senha via email.

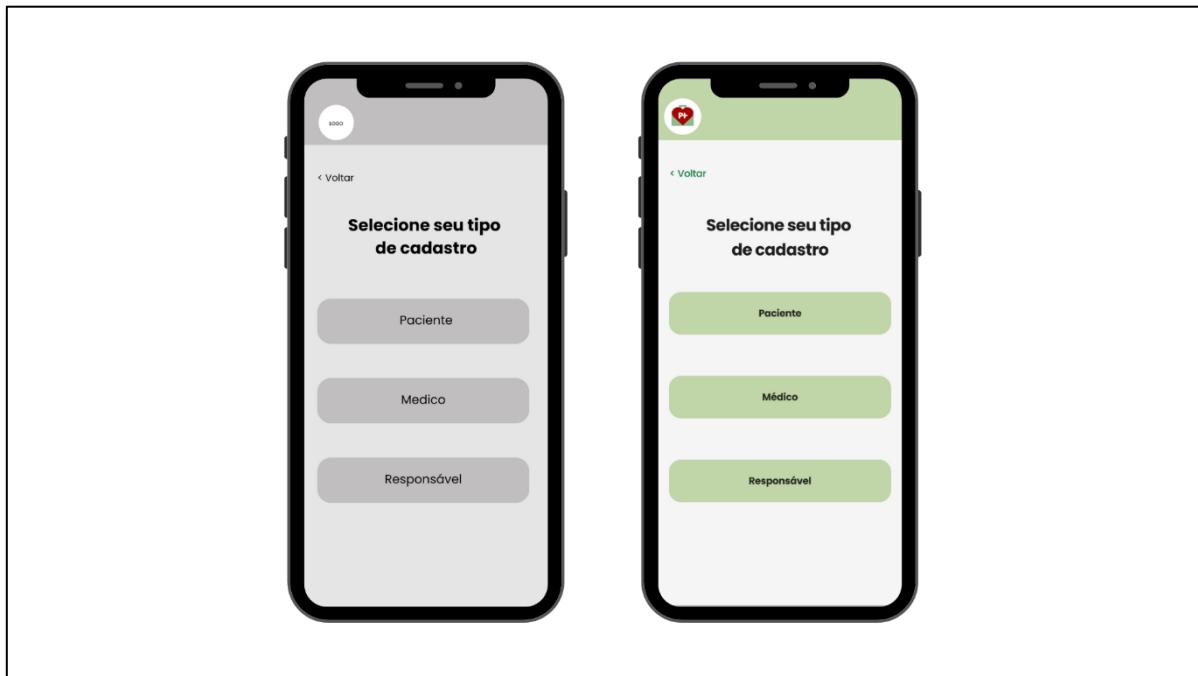
Figura 77 - Wireframe de Baixa e Alta Fidelidade: Esqueceu a senha



Fonte: Dos próprios autores, 2024.

Caso o usuário selecione a opção “cadastre-se”, ele será direcionado diretamente para a tela “Tipo de Cadastro”, na qual, ele terá que selecionar a forma que irá se cadastrar no sistema, podendo ser: Paciente, Médico e Responsável.

Figura 78 - Wireframe de Baixa e Alta Fidelidade: Tipo de cadastro

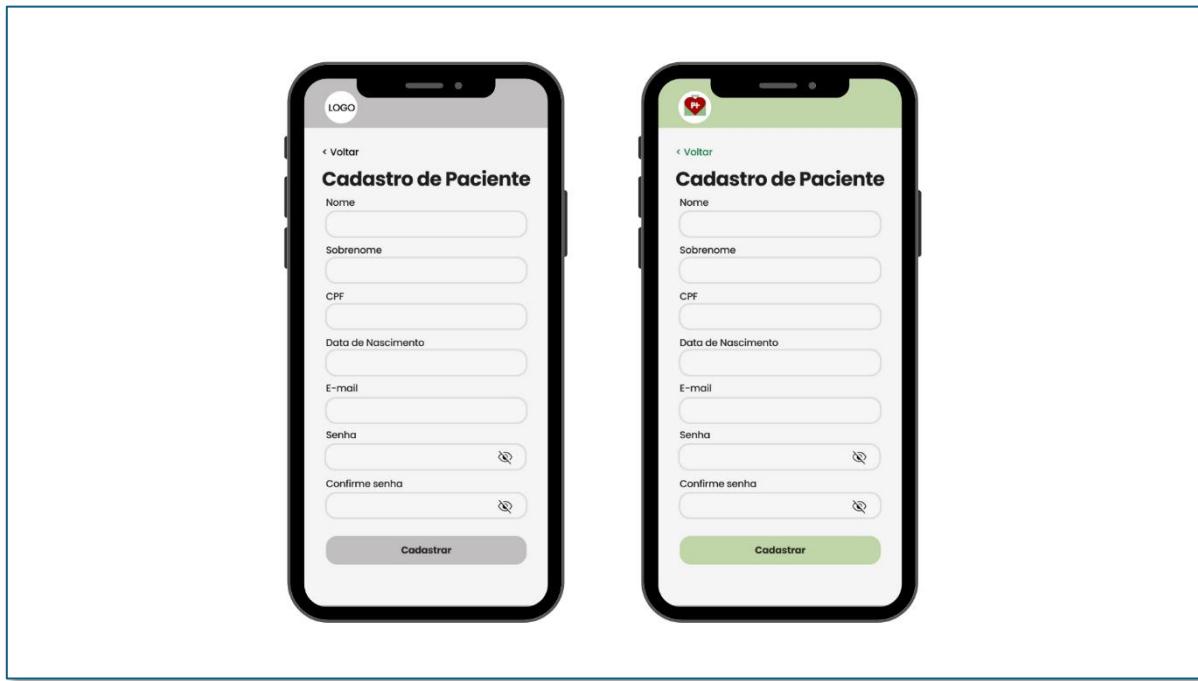


Fonte: Dos próprios autores, 2024.

A partir dessa seção serão apresentadas as telas do usuário paciente, começando pelo cadastro. Na seguinte imagem temos sua representação gráfica, nesta tela o

paciente irá cadastrar seu nome, sobrenome, CPF, data de nascimento, e-mail e senha.

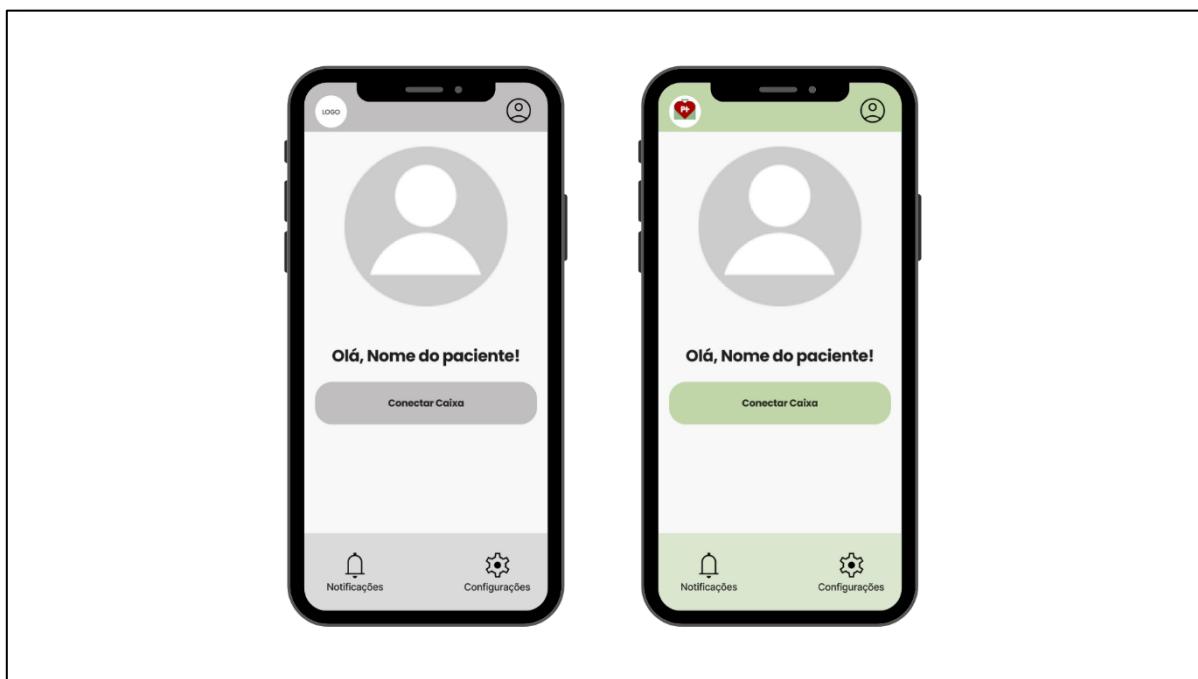
Figura 79 - Wireframe de Baixa e Alta Fidelidade: Cadastro paciente



Fonte: Dos próprios autores, 2024.

Na tela abaixo, será apresentada a Home do paciente antes do cadastro da caixa.

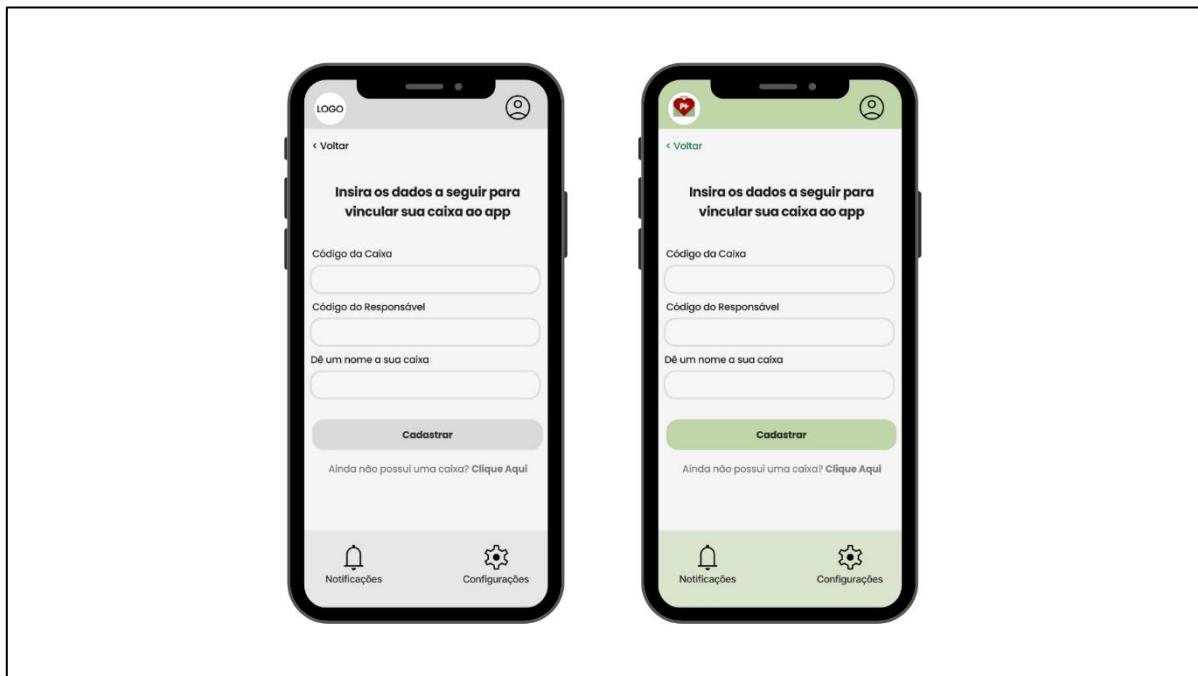
Figura 80 - Wireframe de Baixa e Alta Fidelidade: Home paciente



Fonte: Dos próprios autores, 2024.

Nesta tela, o paciente poderá vincular sua caixa ao aplicativo, através do código da caixa. Além disso, ele poderá inserir um responsável e dar um nome para a caixa.

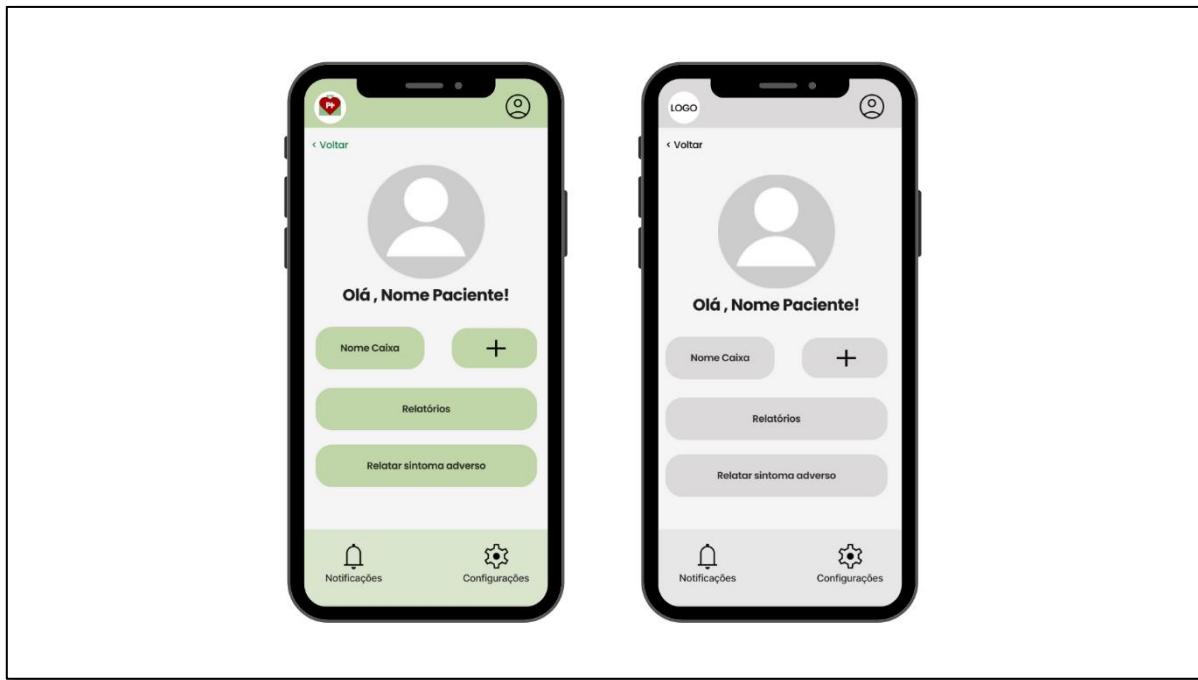
Figura 81 - *Wireframe* de Baixa e Alta Fidelidade: Cadastro da caixa



Fonte: Dos próprios autores, 2024.

Após o cadastro da caixa no aplicativo a home apresentará uma nova identidade visual, dessa vez, o paciente poderá cadastrar mais uma caixa, visualizar relatórios e relatar sintomas adversos.

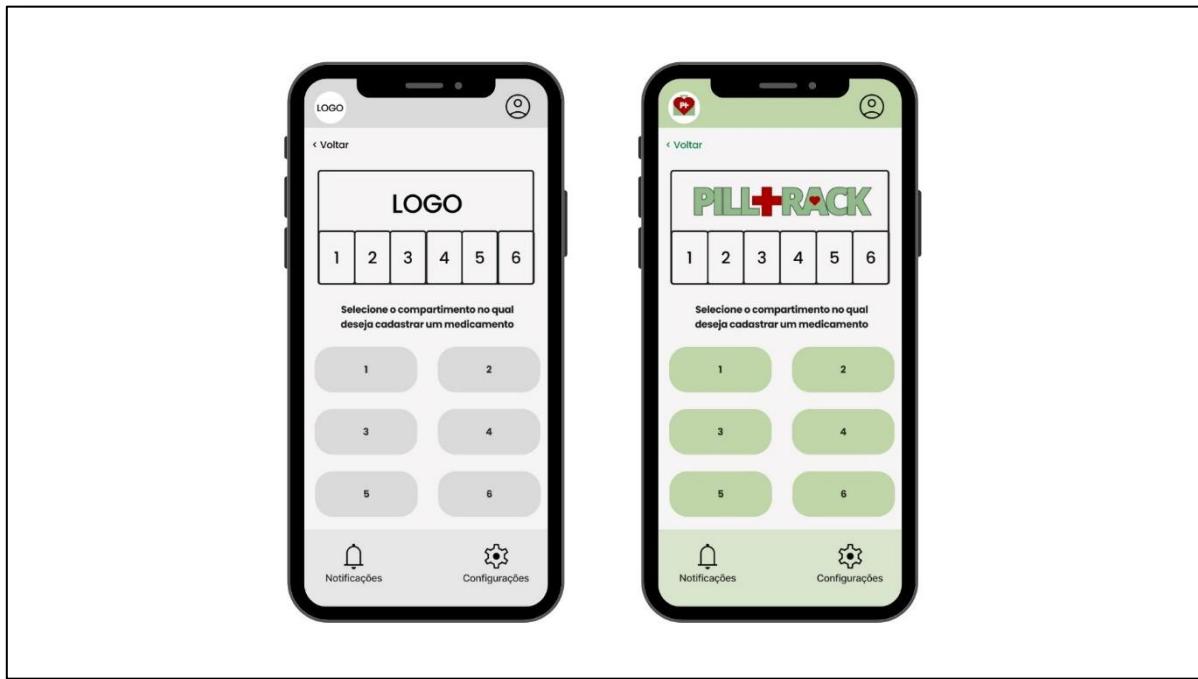
Figura 82 - *Wireframe* de Baixa e Alta Fidelidade: Home após cadastro da caixa



Fonte: Dos próprios autores, 2024.

Caso o usuário deseje cadastrar algum medicamento, ele será redirecionado para a seguinte página. Nesta tela, ele irá selecionar qual compartimento da caixa ele quer cadastrar o remédio.

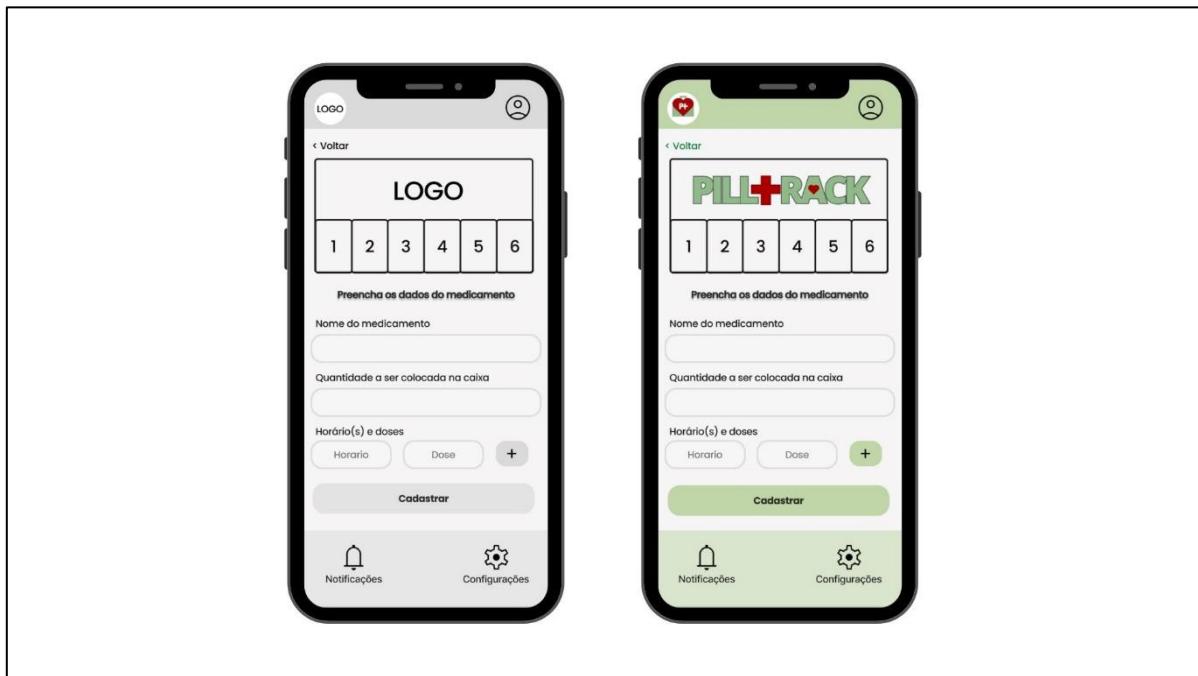
Figura 83 - Wireframe de Baixa e Alta Fidelidade: Cadastrar medicamento



Fonte: Dos próprios autores, 2024.

Depois, ele será redirecionado para a tela de inserção de informações do medicamento, como nome do medicamento, quantidade a ser colocada na caixa, horários e doses.

Figura 84 - Wireframe de Baixa e Alta Fidelidade: Detalhar cadastro de medicamento



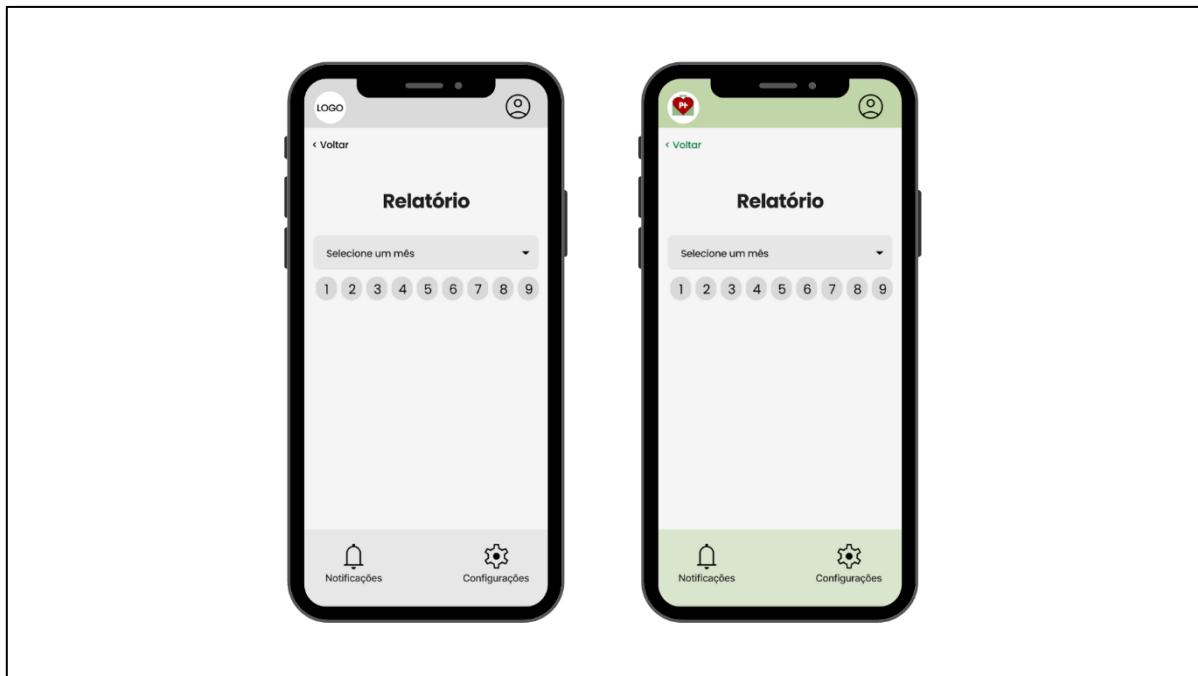
Fonte: Dos próprios autores, 2024.

O paciente também poderá editar o medicamento caso a dose e o horário mudem de horário e quantidade.

Figura 85 - Wireframe de Baixa e Alta Fidelidade: Editar medicamento

O paciente também terá acesso aos relatórios gerados pelo sistema através da *home*. Esta tela também estará disponível para os médicos e responsáveis.

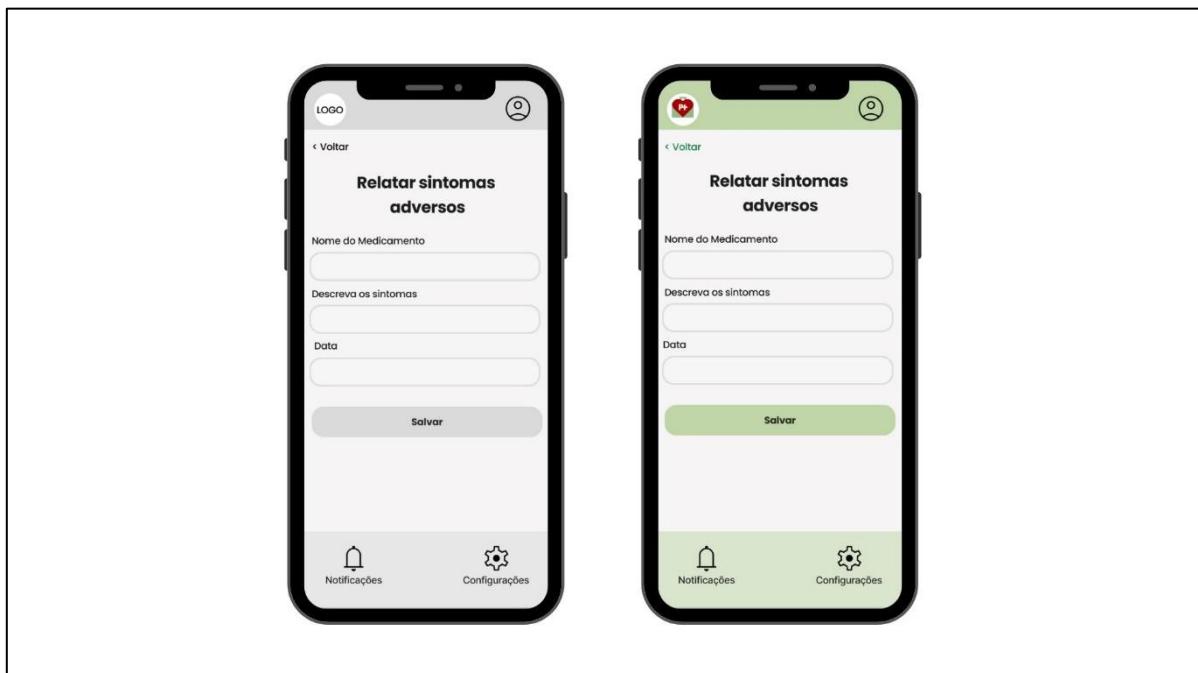
Figura 86 - Wireframe de Baixa e Alta Fidelidade: Relatórios



Fonte: Dos próprios autores, 2024.

Caso o paciente tenha que relatar algum sintoma adverso, a tela a ser apresentada será a da próxima figura ilustrativa.

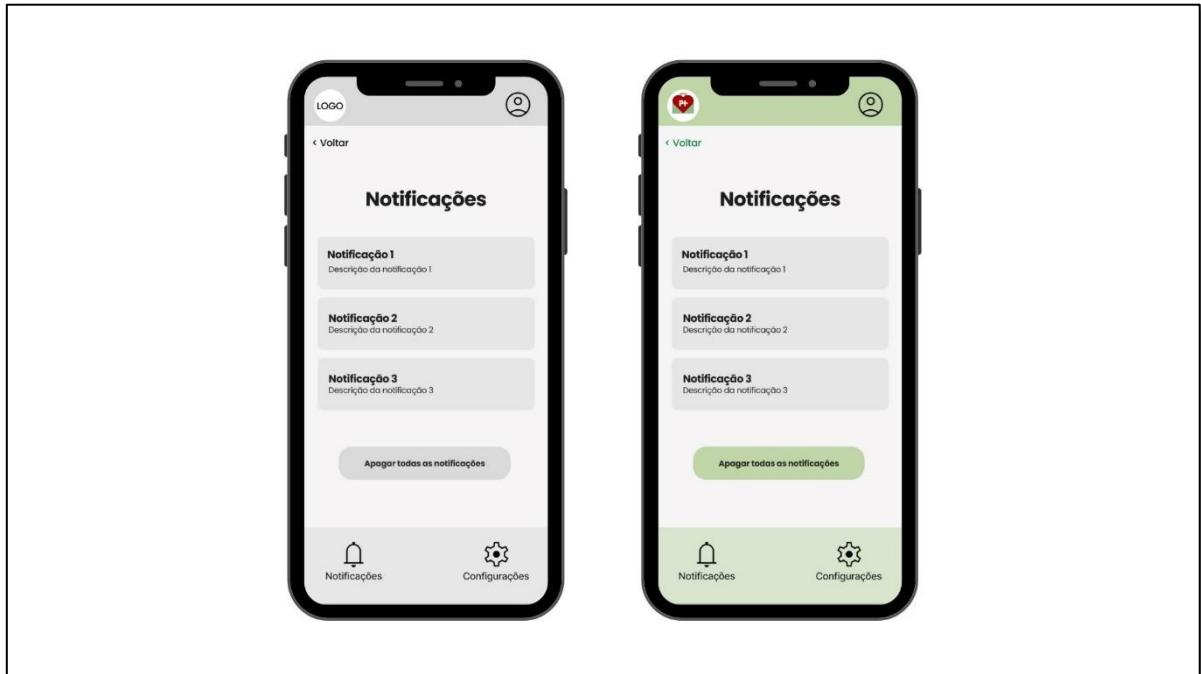
Figura 87 - Wireframe de Baixa e Alta Fidelidade: Relatar sintoma adverso



Fonte: Dos próprios autores, 2024.

As notificações serão enviadas por *pop-up*, mas também estarão acessíveis na parte inferior da tela. A imagem correspondente será apresentada de forma detalhada a seguir. Essa tela também estará disponível para visualização do responsável.

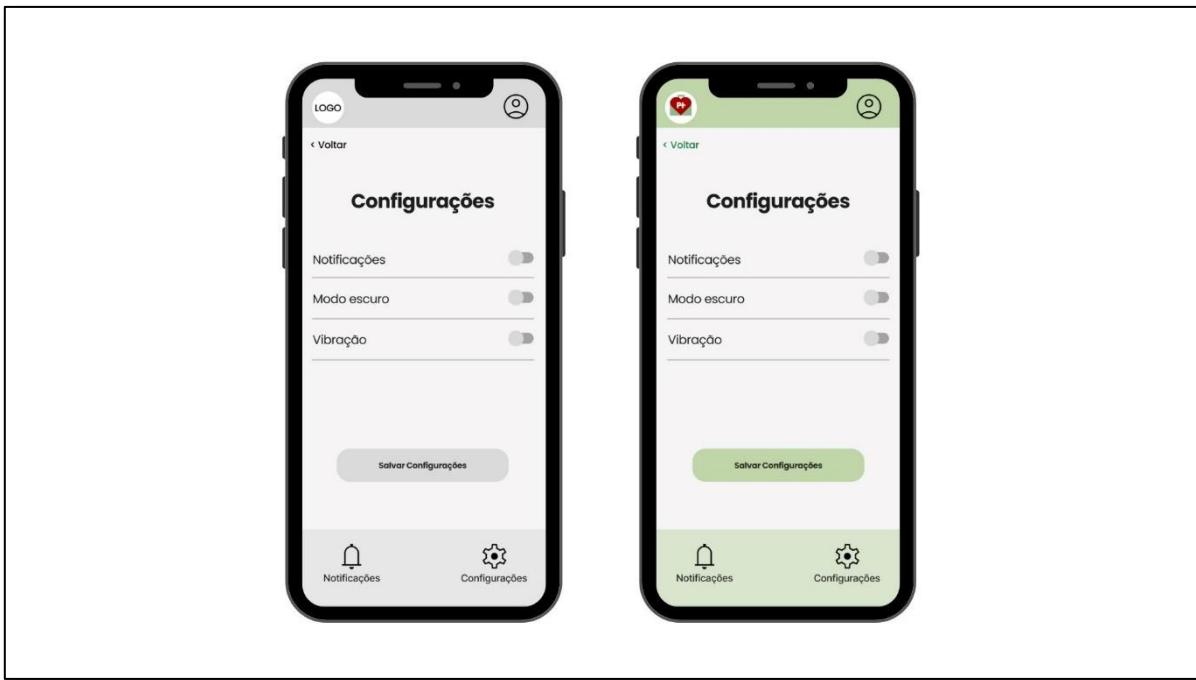
Figura 88 - Wireframe de Baixa e Alta Fidelidade: Notificações



Fonte: Dos próprios autores, 2024.

As configurações serão encontradas ao lado das notificações, através dessa tela o usuário poderá ativar e desativar as notificações, modo escuro e vibração do celular.

Figura 89 - Wireframe de Baixa e Alta Fidelidade: Configurações



Fonte: Dos próprios autores, 2024.

4 CONSIDERAÇÕES FINAIS

O sistema PillTrack visa ajudar, especialmente aqueles que são idosos e encontram dificuldades na gestão de medicamentos, a se automedicar corretamente, melhorando a adesão ao tratamento. Através do aplicativo, será possível a melhora na comunicação entre o paciente e o médico, para que o profissional possa ter certeza de que o paciente está seguindo sua prescrição corretamente, onde o paciente terá uma eficácia precisa ao tratamento.

Para garantir precisão no controle de horários e na liberação das pílulas, utilizaremos o microcontrolador ESP-32 que se conectará ao aplicativo via Bluetooth e o Motor de Passo que será responsável por liberar a pílula na hora correta. Para o aplicativo, foi considerado que a melhor escolha para seu desenvolvimento seria o framework de software de interface de usuário UI, React Native, que está disponível tanto para Android como para IOS.

Ainda que o sistema não tenha passado por testes de segurança e confiabilidade, é correto dizer que o PillTrack foi criado para propor uma melhora no estilo de vida saudável dos usuários do sistema. Ansiamos para que o projeto tenha um

desenvolvimento e contribua para a vida dos pacientes que precisam de auxílio especial para se automedicar corretamente.

REFERÊNCIAS

- BOOCH, Grady; RUMBAUGH, James; JACOBSON, Ivan. **UML: Guia Do Usuário.** 2. ed. [S. I.]. GEN LTC, 2006.
- DATE, Chris John. **Introdução a Sistemas de Banco de Dados.** 1. ed. São Paulo: GEN LTC, 2023.
- DEITEL, Harvey; DEITEL, Paul. **C++ Como Programar.** 5. ed. São Paulo: Pearson Prentice Hall, 2015.
- DUARTE, Luiz. **Programação Web com Node.js.** 6. ed. [S.I.]: LuizTools, 2017.
- ELETRÔNICA ÔMEGA. **E-book Internet das Coisas para Iniciantes com ESP-32.** 1. ed. [S.I.]: Arduino Ômega, 2021. Livro Digital.
- ESCUDELARIO, Bruna; PINHO, Diego. **React Native: Desenvolvimento de Aplicativos Mobile com React.** [S.I.]: Casa do Código, 2020.
- FLANAGAN, David. **JavaScript: O Guia Definitivo.** 6. ed. Porto Alegre: Bookman, 2013.
- FOWLER, Martin. **UML Essencial: Um Breve Guia para a Linguagem Padrão.** [S.I.]: [s.n.].
- GRONER, Loiane. **Estruturas de Dados e Algoritmos com JavaScript: Escreva um Código JavaScript Complexo e Eficaz Usando a Mais Recente ECMAScript.** 2. ed. [S.I.]: Novatec Editora, 2019.
- GUEDES, G. T. A. **UML 2 - Uma Abordagem Prática.** [S.I.]: Novatec Editora, 2018.
- IEPSEN, Edécio Fernando. **Lógica de Programação e Algoritmos com JavaScript - 2ª Edição: Uma Introdução à Programação de Computadores com Exemplos e Exercícios para Iniciantes.** 2. ed. São Paulo: Novatec Editora, 2022.
- JOBSTRAIBIZER, Flávia. **Criação de Sites com o CSS.** São Paulo: Universo dos Livros Editora, 2009.

LUCIANO, J.; ALVES, W. J. B. **Padrão de Arquitetura MVC: Model-View-Controller.** [S.I.]: [s.n.], 2011.

MAGRANI, Eduardo. **A Internet das Coisas.** 1. ed. Rio de Janeiro: FGV Editora, 2018.

MONK, Simon. **30 Projetos com Arduino.** 2. ed. Porto Alegre: Artmed, 2014.

PEREIRA, Caio Ribeiro. **Aplicações Web Real-Time com Node.js.** [S.I.]: Casa do Código, 2014.

SILVA, Maurício Samy. **Criando Sites com HTML: Sites de Alta Qualidade com HTML e CSS.** 1. ed. São Paulo: Novatec Editora, 2008.

SILVA, Maurício Samy. **JavaScript - Guia do Programador: Guia Completo das Funcionalidades de Linguagem JavaScript.** 1. ed. São Paulo: Novatec Editora, 2010.

SILVA, Maurício Samy. **React - Aprenda Praticando: Desenvolva Aplicações Web Reais com Uso da Biblioteca React e de Seus Módulos Auxiliares.** 1. ed. São Paulo: Novatec, 2021.

SANTOS, S. **Introdução IoT: Desvendando a Internet das Coisas.** North Charleston, SC, USA: Createspace Independent Publishing Platform, 2018.

VILLAS-BOAS, Sergio Barbosa. **C/C++ e Orientação a Objetos em Ambiente Multiplataforma.** Rio de Janeiro, RJ, Brasil: [s.n.], 2001.

PEREIRA, Marcelo Robson Sousa. **A aplicação do microcontrolador ESP32 no ensino: medindo posições em função do tempo utilizando o sensor VL53L0X associado ao ESP32.** Orientador: Victor Montero Del Águila. 2020. 24 f. Trabalho de Conclusão de Curso (Especialização) - Departamento de Pós-Graduação, Universidade Federal do Amapá, Macapá, 2020. Disponível em: <http://repositorio.unifap.br:80/jspui/handle/123456789/898>. Acesso em: 20 maio. 2024.

PUHLMANN, Henrique Frank W. **Módulo de display LCD.** [S.I.]: [s.n.], 2015.

ALBERTIN, A. L.; ALBERTIN, R. M. DE M. **A internet das coisas irá muito além das coisas.** GV-executivo, v. 16, n. 2, p. 12–17, 2017.

KATZ, Marcelo; FEITOSA, Gustavo Freitas; PINTO, Ibraim Masciarelli F.; FELIX, Marcelo de Maria; BORTOLOTTO, Luiz Aparecido. **Uso da tecnologia para engajar pacientes e otimizar a adesão terapêutica.** Revista da Sociedade de Cardiologia do Estado de São Paulo, [S.I.], v. 30, n. 3, p. 352-357, 20 outubro 2020. Disponível em: [07_revistasocesp_v30_03.pdf](#). Acesso em: 10 de março de 2024.

MASSOLA, S. C.; PINTO, G. S. **Uso da internet das coisas (IoT) a favor da saúde.** Revista Interface Tecnológica, v. 15, n. 2, p. 124–137, 2018.

ALVAREZ, Danilo. **Buzzer com ESP32 - Curso ESP32 básico.** Portal Vida de Silício, 6 de novembro de 2023. Disponível em:
<https://portal.vidadesilicio.com.br/buzzer-com-esp32-curso-esp32-basico>. Acesso em: 23 de maio de 2024.

AGREDA, Victor. **O que é Firebase?** Oracle, 27 de janeiro de 2023. Disponível em: <https://developer.oracle.com/pt-BR/learn/technical-articles/what-is-firebase>. Acesso em: 19 de maio de 2024.

BECKER, Lauro. **Wireframes, o que são e por que os utilizamos?** Orgânica Digital, 20 de abril de 2022. Disponível em: <https://www.organicadigital.com/blog/o-que-sao-wireframes-e-por-que-os-utilizamos/>. Acesso em: 27 de abril de 2024.

CAMPINAS, U. **Entenda o impacto e a importância do acompanhamento médico.** Disponível em: <https://www.unimedcampinas.com.br/blog/viver-com-saudade/entenda-o-impacto-e-a-importancia-do-acompanhamento-medico>. Acesso em: 26 de março de 2024.

ELETROGATE. **Guia do Motor de Passo 28BY-48 = Driver ULN3003.** Elotrogate, 23 de julho de 2018. Disponível em: <https://blog.eletrogate.com/guia-completo-do-motor-de-passo-28byj-48-driver-uln2003/>. Acesso em: 19 de outubro de 2024.

ELETROGATE. **Sensor Óptico TCRT5000 com Arduino.** Disponível em: <https://blog.eletrogate.com/sensor-optico-tcrt5000-com-arduino/>. Acesso em: 19 jun. 2024.

EUROFARMA. **A importância de tomar remédio na hora certa.** 2018. Disponível em: <https://eurofarma.com.br/artigos/a-importancia-de-tomar-remedio-na-hora-certa-inclusive-para-idosos>. Acesso em: 12 de março de 2024.

FERREIRA, A. P. C.; CAMPOS, E. M. P. **A Equipe de Saúde Diante do Paciente Não Aderente ao Tratamento.** Psicologia: Ciência e Profissão, v. 43, p. e244855, 20 de fevereiro de 2023. Disponível em: <https://www.scielo.br/j/pcp/a/TQtxVL3fdgXTYvhyRfXFvJp/>. Acesso em: 26 de março de 2024.

FIREBASE. **Make your app the best it can be with Firebase and generative AI.** [S.I.]. Disponível em: <https://firebase.google.com/?hl=pt-br>. Acesso em: 19 de maio de 2024.

FIREBASE. **Cloud Firestore | Armazene e sincronize dados do app em escala global.** [S.I.]. Disponível em: <https://firebase.google.com/products/firestore?hl=pt-br>. Acesso em: 18 de junho de 2024.

GADO, Wesley. **O que é NPM e como usar uma biblioteca instalada por ele.** Blog da TreinaWeb. Disponível em: <https://www.treinaweb.com.br/blog/o-que-e-npm-e-como-usar-uma-biblioteca-instalada-por-ele/>. Acesso em: 15 de maio de 2024.

GOOGLE CLOUD. **Firebase.** [S.I.]. Disponível em: <https://cloud.google.com/firestore#benefits>. Acesso em: 18 de junho de 2024.

GUIMARÃES, Fábio. **Buzzer - Como usar com o Arduino.** Mundo Projetado, 11 de agosto de 2017. Disponível em: <https://mundoprojetado.com.br/buzzer-como-usar-com-o-arduino>. Acesso em: 23 de maio de 2024.

HIGA, P.; MARQUES, A. **O que é LCD? Conheça os tipos e as vantagens dessa tecnologia.** Tecnoblog. Disponível em: <https://tecnoblog.net/responde/o-que-e-lcd/>. Acesso em: 21 de maio de 2024.

LOCAWEB. **C++: guia sobre a linguagem de programação.** Disponível em: <https://www.locaweb.com.br/blog/temas/codigo-aberto/c-plus-plus/>. Acesso em: 26 de maio de 2024.

LOPES, Michele. **Modelagem 3D: o que é e como funciona.** Ebac Online, 29 de setembro de 2023. Disponível em: <https://ebaconline.com.br/blog/modelagem-3d-o-que-e-e-como-funciona/>. Acesso em: 26 de maio de 2024.

LOPES, Michele. **O que é Figma e como usar?** Ebac Online, 02 de outubro de 2023. Disponível em: <https://ebaconline.com.br/blog/o-que-e-figma-e-como-usar/>. Acesso em: 26 de abril de 2024.

MAKIYAMA, M. **Placa ESP32: O que é, para que serve e uso!** Disponível em: <https://victorvision.com.br/blog/placa-esp32>. Acesso em: 20 de maio de 2024.

MOURA, Beatriz; HENRIQUE, Mateus. **O que é HTML? Suas tags - Parte 1: Estrutura Básica.** Alura, 8 de fevereiro de 2024. Disponível em: <https://www.alura.com.br/artigos/o-que-e-html-suas-tags-parte-1-estrutura-basica>. Acesso em: 23 de abril de 2024.

O que é wireframe? **Lucidchart.** Disponível em: <https://www.lucidchart.com/pages/pt/o-que-e-wireframe>. Acesso em: 27 de abril de 2024.

PIXEL DIAGNÓSTICOS. **A importância da comunicação entre paciente e profissional da saúde.** Disponível em: <https://pixeldiagnosticocom.br/blog/post/2020-10-10-a-importancia-da-comunicacao-entre-o-paciente-e-o-profissional-de-saude>. Acesso em: 26 de março de 2024.

PUHLMANN, H. F. W. **Módulo de Display LCD.** Disponível em: <https://embarcados.com.br/modulo-de-display-lcd>. Acesso em: 25 junho de 2024.

REACT. **Aprenda React.** Disponível em: <https://pt-br.react.dev/learn>. Acesso em: 18 de junho de 2024.

REACT NATIVE. **React Native - A framework for building native apps using React.** Disponível em: <https://reactnative.dev/>. Acesso em: 03 de maio de 2024.

RODRIGUES, Kauê. **Introdução ao figma: guia completo para iniciantes no figma.** Cubos Academy, 06 de novembro de 2023. Disponível em: <https://blog.cubos.academy/figma-para-iniciantes/>. Acesso em: 23 de abril de 2024.

Sensor óptico reflexivo TCRT5000 com Arduino. Arduino e Cia. Disponível em: <https://www.arduinoecia.com.br/sensor-optico-reflexivo-tcrt5000-arduino/>. Acesso em: 18 jun. 2024.

VIANA, Carol Correia. **Como acionar motor de passo com Driver ULN2003 e Arduino.** Blog da Robótica, 30 de junho de 2022. Disponível em: <https://www.blogdarobotica.com/2022/06/30/como-acionar-motor-de-passo-com-driver-uln2003-e-arduino/>. Acesso em 02 de outubro de 2024.

Wireframes: quais os tipos e as principais ferramentas de criação. **Rockcontent**, 26 de fevereiro de 2019. Disponível em: <https://rockcontent.com/br/blog/wireframes/>. Acesso em: 27 de abril de 2024.