**CENTRO ESTADUAL DE EDUCAÇÃO TECNOLÓGICA PAULA SOUZA DA ETEC DA ZONA LESTE**

**Novotec Desenvolvimento de Sistemas AMS**

**Gabriel Messias da Silva**

**Giovanna Aparecida Vivencio Rodrigues**

**Guilherme Felix Barreto**

**SAFE GUARDIAN:**

**Sistema de Detecção de Quedas**

**São Paulo**

**2024**

**Gabriel Messias da Silva**

**Giovanna Aparecida Vivencio Rodrigues**

**Guilherme Felix Barreto**

**SAFE GUARDIAN:**

**Sistema de Detecção de Quedas**

Trabalho de Conclusão de Curso apresentado ao Curso Ensino Médio com Habilitação Profissional de Técnico em Desenvolvimento de Sistemas AMS da Etec Zona Leste, orientado pelo Professor Jeferson Roberto de Lima, como requisito parcial para obtenção do título de técnico em Desenvolvimento de Sistemas.

**São Paulo**

**2024**

**Página de Dedicatória**

Dedicamos este projeto a todos os idosos e indivíduos suscetíveis que enfrentam desafios diários com coragem e resiliência. Que nosso projeto possa trazer mais segurança e tranquilidade, oferecendo um cuidado mais eficaz e uma resposta rápida em momentos de risco que, infelizmente, as quedas podem proporcionar.

**Página de Agradecimento**

Este projeto só pôde dar um grande passo em nossa vida acadêmica e se tornar realidade graças ao apoio de muitas pessoas que foram fundamentais para seu desenvolvimento. Agradecemos a todos que nos apoiaram e, de certa forma, nos auxiliaram com suas dúvidas e sugestões, nos ajudando a aperfeiçoar ainda mais o Safe Guardian.

Aos nossos colegas e amigos que estiveram ao nosso lado ao longo do nosso desenvolvimento, muitas vezes nos frustrando ou aumentando nossa ansiedade, mas sempre nos motivando a seguir em frente e continuar desenvolvendo nosso projeto de maneira eficaz, expressamos nossa eterna gratidão. Esse apoio foi essencial para o nosso progresso.

E aos nossos familiares, que pacientemente nos ouviram em momentos de desabafo, frustração, noites em claro, lágrimas e até em momentos de raiva, devemos imensa gratidão. O maior apoio que poderíamos receber veio justamente de vocês. Agradecemos profundamente por todo o suporte e pela torcida incansável que nos motivou até a conclusão deste projeto.

Gostaríamos de expressar nossos sinceros agradecimentos ao nosso coordenador, professor e orientador, Jeferson Roberto de Lima que, com sua abordagem e cobrança rígida, porém descontraída e cheias de sarcasmo, sempre nos desafiou, mas também nos apoiou imensamente, trazendo ensinamentos e dedicação ao longo do projeto. Agradecemos nosso coordenador Rogério Bezerra, que trouxe críticas e sugestões, tornando o processo mais eficiente e melhorando o projeto. Devemos nossos agradecimentos especiais aos nossos professores, Carlos Alberto e Marcelo Collado, por todo tempo e atenção que nos deram em momentos de dúvidas, nos auxiliando em momentos de angústia e oferecendo apoio constante para a construção do nosso projeto.

Por fim, agradecemos a Deus por sua infinita bondade e amor, que nos protegeram e guiaram em nossa jornada acadêmica, fortalecendo-nos nos momentos de cansaço e concedendo-nos vigor quando nos faltavam forças.

“Na juventude deve-se acumular o saber. Na velhice fazer uso dele.."

**Jean-Jacques Rousseau**

**Resumo**

Este projeto aborda um sistema destinado ao auxílio na detecção de quedas para idosos e indivíduos suscetíveis. O objetivo principal é proporcionar uma resposta rápida em casos de emergência, minimizando os riscos associados a quedas e aumentando a qualidade de vida desses indivíduos. A pesquisa foi conduzida por meio de um estudo misto, que incluiu tanto métodos quantitativos quanto qualitativos para descobrir o número de idosos que sofreram este evento e suas consequências após a queda. O sistema utiliza um dispositivo vestível, que é acoplado ao corpo dos utilizadores, equipado com sensores que monitoram continuamente os seus movimentos, onde em casos de queda, o dispositivo enviará alertas imediatos para cuidadores ou familiares mediantes de uma aplicação móvel conectada, permitindo uma intervenção rápida. Espera-se que o sistema proposto tenha um impacto positivo, reduzindo incidência de lesões graves, fornecendo uma rápida assistência em casos emergenciais e promovendo confiança desses indivíduos em suas atividades diárias.

**Palavras chaves**: Detecção de quedas. Idosos. Dispositivo vestível.

**Abstract**

This research addresses a system designed to help detect falls for the elderly and susceptible individuals. The main objective of this research is to provide a quick response in emergency cases, minimizing the risks associated with falls and increasing the quality of life of these individuals. The research was conducted through a mixed study, which included both quantitative and qualitative methods to discover the number of elderly people who survived these events and their consequences after the fall. The system uses a wearable device, which is attached to the users’ bodies, equipped with sensors that continuously monitor their movements. In cases of a fall, the device will send immediate alerts to caregivers or family members through a connected mobile application, allowing for quick intervention. To conclude, the proposed system is expected to have a positive impact, including reducing serious injuries, providing rapid assistance in emergency cases, and promoting the confidence of these individuals in their daily activities.

**Keywords**: Detect falls. Elderly. Wearable Device.

**LISTA DE ILUSTRAÇÕES**

[Figura 1 – ESP32 16](#_Toc180966588)

[Figura 2 – Sensor MPU6050 16](#_Toc180966589)

[Figura 3 - Placa PCI com barra de pinos 17](#_Toc180966590)

[Figura 4 – Modelo Jumpers Fêmea-Fêmea 18](#_Toc180966591)

[Figura 5 - Resistores 100k 18](#_Toc180966592)

[Figura 6 – Suporte de pilha 19](#_Toc180966593)

[Figura 7 - Módulo TP4056 19](#_Toc180966594)

[Figura 8 - Bateria de Li-ion 18650 20](#_Toc180966595)

[Figura 9 - Exemplo de modelagem 3D 21](#_Toc180966596)

[Figura 10 – Exemplo simples de código de ligamento de leds 22](#_Toc180966597)

[Figura 11 – Representação visual do código de leds 24](#_Toc180966598)

[Figura 12 – Logotipo FIGMA 26](#_Toc180966599)

[Figura 13 – Exemplo de wireframe de baixa fidelidade 27](#_Toc180966600)

[Figura 14 – Exemplo de wireframe de alta fidelidade 28](#_Toc180966601)

[Figura 15 – Exemplo de código para a construção de um formulário em HTML 30](#_Toc180966602)

[Figura 16 – Resultado do formulário em HTML 31](#_Toc180966603)

[Figura 17 – Exemplo de código em CSS 32](#_Toc180966604)

[Figura 18 – Resultado da estilização em CSS 33](#_Toc180966605)

[Figura 19 – Exemplo de código em JavaScript 34](#_Toc180966606)

[Figura 20 – Mensagem de erro da codificação em JavaScript 34](#_Toc180966607)

[Figura 21 – Mensagem de conclusão da codificação em JavaScript 35](#_Toc180966608)

[Figura 22 – Inserção do JavaScript no HTML 35](#_Toc180966609)

[Figura 23 – Atributo "OnSubmit" em HTML 35](#_Toc180966610)

[Figura 24 – Estrutura de Pasta 36](#_Toc180966611)

[Figura 25 – Exemplo de Formulário de Login utilizando React Native 37](#_Toc180966612)

[Figura 26 – Continuação do formulário de login em React Native 39](#_Toc180966613)

[Figura 27 – Resultado do formulário utilizando React Native 40](#_Toc180966614)

[Figura 28 – Logotipo Firebase 41](#_Toc180966615)

[Figura 29 – Console Firebase 42](#_Toc180966616)

[Figura 30 – Logotipo da API CallMeBot 42](#_Toc180966617)

[Figura 31 - Caso de Uso do sistema 44](#_Toc180966618)

[Figura 32 - Diagrama de Atividade: Confirmar Resolução da Queda 51](#_Toc180966619)

[Figura 33 - Diagrama de Atividade: Detectar a Queda 52](#_Toc180966620)

[Figura 34 - Diagrama de Atividade: Emparelhar Dispositivo 53](#_Toc180966621)

[Figura 35 - Diagrama de Atividade: Enviar Mensagem 54](#_Toc180966622)

[Figura 36 - Diagrama de Atividade: Gerenciar Bateria 55](#_Toc180966623)

[Figura 37 - Diagrama de Atividade: Manter Dados Emergenciais 56](#_Toc180966624)

[Figura 38 - Diagrama de Atividade - Notificar Bateria Baixa do Dispositivo 57](#_Toc180966625)

[Figura 39 - Diagrama de Atividade: Notificar Queda 58](#_Toc180966626)

[Figura 40 - Diagrama de Atividade: Visualizar Histórico de Quedas 59](#_Toc180966627)

[Figura 41 - Diagrama de Sequência: Confirmar Resolução da Queda 60](#_Toc180966628)

[Figura 42 - Diagrama de Sequência: Detectar a Queda 61](#_Toc180966629)

[Figura 43 - Diagrama de Sequência: Emparelhar Dispositivo 62](#_Toc180966630)

[Figura 44 - Diagrama de Sequência: Enviar Mensagem 63](#_Toc180966631)

[Figura 45 - Diagrama de Sequência: Gerenciar Bateria 64](#_Toc180966632)

[Figura 46 - Diagrama de Sequência: Manter Dados Emergenciais 65](#_Toc180966633)

[Figura 47 - Diagrama de Sequência: Notificar Bateria Baixa do Dispositivo 66](#_Toc180966634)

[Figura 48 - Diagrama de Sequência: Notificar Queda 67](#_Toc180966635)

[Figura 49 - Diagrama de Sequência: Visualizar Histórico de Quedas 68](#_Toc180966636)

[Figura 50 - Diagrama de Máquina de Estado: Visualizar histórico de Quedas 69](#_Toc180966637)

[Figura 51 - Diagrama de Máquina de Estado: Confirmar Resolução de Queda 70](#_Toc180966638)

[Figura 52 - Diagrama de Máquina de Estado: Enviar Mensagem 71](#_Toc180966639)

[Figura 53 - Diagrama de Máquina de Estado: Gerenciar Bateria 72](#_Toc180966640)

[Figura 54 - Diagrama de Máquina de Estado: Notificar Bateria baixa do dispositivo 73](#_Toc180966641)

[Figura 55 - Diagrama de Máquina de Estado: Notificar Queda 74](#_Toc180966642)

[Figura 56 - Protótipo do dispositivo físico Safe Guardian 75](#_Toc180966643)

[Figura 57 - Wireframe de Baixa Fidelidade: Tela de carregamento 76](#_Toc180966644)

[Figura 58 - Wireframe de Alta Fidelidade: Tela de carregamento 77](#_Toc180966645)

**LISTA DE QUADROS**

[Quadro 1 - Documentação do caso de uso: Enviar Mensagem 46](#_Toc180965051)

[Quadro 2 - Documentação do caso de uso: Detectar a Queda 46](#_Toc180965052)

[Quadro 3 - Documentação do caso de uso: Gerenciar Bateria 47](#_Toc180965053)

[Quadro 4 - Documentação do caso de uso: Notificar Bateria Baixa do Dispositivo 47](#_Toc180965054)

[Quadro 5 - Documentação do caso de uso: Confirmar Resolução da queda 47](#_Toc180965055)

[Quadro 6 - Documentação do caso de uso: Manter Dados Emergenciais 48](#_Toc180965056)

[Quadro 7 - Documentação do caso de uso: Visualizar Histórico de Quedas 48](#_Toc180965057)

[Quadro 8 - Documentação do caso de uso: Emparelhar Dispositivo 49](#_Toc180965058)

[Quadro 9 - Documentação do caso de uso: Notificar queda 50](#_Toc180965059)

**LISTA DE ABREVIATURA E SIGLAS**

Acrilonitrilo Butadieno Estireno (ABS);

Fabricação por Filamento Fundido(FFF);

Internet of Things (IoT);

iPhone Operating System (IOS);

Unified Modeling Language (UML);

Not Only SQL (NOSql);

Application Programming Interfaces (APIs);

Hypertext Markup Language (HTML);

JavaScript (JS);

Cascading Style Sheets (CSS);

Placa de Circuito Impresso (PCI);

Requisitos Funcionais (RF);

Requisitos Não Funcionais (RNF);

**SUMÁRIO**

[1 INTRODUÇÃO 14](#_Toc181038127)

[2 REFERENCIAL TEÓRICO 15](#_Toc181038128)

[2.1 Quedas em idosos 15](#_Toc181038129)

[2.2 Tecnologias Utilizadas 15](#_Toc181038130)

[2.2.1 IOT 15](#_Toc181038131)

[2.2.1.1 ESP32 16](#_Toc181038132)

[2.2.1.2 Módulo MPU6050 16](#_Toc181038133)

[2.2.1.3 Placa de Circuito Impresso (PCI) 17](#_Toc181038134)

[2.2.1.4 Jumpers 18](#_Toc181038135)

[2.2.1.5 Resistores 18](#_Toc181038136)

[2.2.1.6 Suporte de pilha 19](#_Toc181038137)

[2.2.1.7 Módulo TP4056 19](#_Toc181038138)

[2.2.1.8 Bateria recarregável Li-ion 20](#_Toc181038139)

[2.2.2 Modelagem 3D 20](#_Toc181038140)

[2.2.3 C++ 21](#_Toc181038141)

[2.2.4 UML - Unified Modeling Language 24](#_Toc181038142)

[2.2.5 FIGMA 25](#_Toc181038143)

[2.2.6 Wireframe 26](#_Toc181038144)

[2.2.7 React 28](#_Toc181038145)

[2.2.7.1 HTML 29](#_Toc181038146)

[2.2.7.2 Cascading Style Sheet (CSS) 31](#_Toc181038147)

[2.2.7.3 JavaScript 33](#_Toc181038148)

[2.2.7.4 React Native 35](#_Toc181038149)

[2.2.8 Firebase 41](#_Toc181038150)

[2.2.9 API CallMeBot 42](#_Toc181038151)

[3 DESENVOLVIMENTO 43](#_Toc181038152)

[3.1 Diagrama de Caso de Uso 43](#_Toc181038153)

[3.1.1 Documentação dos Casos de Uso 44](#_Toc181038154)

[3.1.1.1 Documentação dos Casos de Uso da API 46](#_Toc181038155)

[3.1.1.2 Documentação dos Casos de Uso do Dispositivo 46](#_Toc181038156)

[3.1.1.3 Documentação dos Casos de Uso do Cuidador 47](#_Toc181038157)

[3.1.1.4 Documentação dos Casos de Uso Automático 49](#_Toc181038158)

[3.2 Diagrama de Atividade 50](#_Toc181038159)

[3.3 Diagrama de Sequência 59](#_Toc181038160)

[3.4 Diagrama de Máquina de Estado 68](#_Toc181038161)

[3.5 Protótipo do dispositivo físico 74](#_Toc181038162)

[3.6 Prototipação das telas do aplicativo 75](#_Toc181038163)

[4 CONSIDERAÇÕES FINAIS 97](#_Toc181038164)

[REFERÊNCIAS 99](#_Toc181038165)

# INTRODUÇÃO

A população mundial está envelhecendo rapidamente e, conforme relatado pela Organização Pan-Americana de Saúde (OPAS, 2021), em 2020, mais de 8% da população tinha 65 anos ou mais, estimando-se que essa porcentagem ultrapasse 30% até o final do século, diante desse cenário, desafios crescentes vem à saúde pública, especialmente no que tange à segurança e autonomia de idosos e indivíduos suscetíveis a quedas. As quedas representam uma das principais causas de morte e hospitalização entre idosos, com um aumento notável de óbitos no Brasil entre 2000 e 2019, conforme Gonçalves et al. (2022). Um sistema de detecção e resposta a quedas pode, assim, contribuir para a redução dessas estatísticas, além de proporcionar maior tranquilidade aos cuidadores e familiares. Visando essa problemática, o projeto irá abordar o desenvolvimento de um dispositivo de detecção de quedas para idosos e indivíduos vulneráveis, o qual integra um sistema composto por hardware e aplicativo para monitoramento e resposta rápida em casos de emergência. Nesse mesmo contexto, o sistema de detecção de quedas proposto neste projeto visa minimizar os riscos associados a esse tipo de evento, pensando na melhoria da qualidade e segurança do usuário através desse recurso. O projeto envolve uma série de etapas essenciais para o desenvolvimento de um sistema de detecção de quedas. Inicialmente, realizaremos pesquisas aprofundadas sobre o tema e identificaremos tecnologias disponíveis. Com base nessa pesquisa, refinaremos conceitos e definiremos as especificações do projeto, levantando requisitos funcionais e não funcionais. Em seguida, estruturaremos diagramas UML, incluindo Casos de Uso, Sequência, Atividade e Máquina de Estado. Além disso, selecionaremos as linguagens adequadas para o desenvolvimento do dispositivo e aplicativo. Iremos idealizar o dispositivo necessário e desenvolveremos o software conforme as especificações definidas, estabelecendo um local estratégico para anexar o dispositivo, garantindo a precisão na detecção de quedas. Nos capítulos posteriores, serão abordados o processo de desenvolvimento do sistema Safe Guardian — Sistema de detecção de quedas — juntamente ao referencial teórico, considerações finais e referências.

# REFERENCIAL TEÓRICO

Neste presente capítulo, será apresentado uma visão abrangente das bases teóricas e tecnológicas que fundamentam o projeto Safe Guardian, abordando conceitos relacionados ao envelhecimento populacional e os riscos de quedas, a tecnologia assistiva e os principais componentes, como sensores, incluindo acelerômetro e giroscópio que auxiliarão na construção do dispositivo.

## Quedas em idosos

Segundo Maia (2011), cerca de 30% dos indivíduos com mais de 65 anos de idade caem ao menos uma vez por ano, podendo destacar que esses incidentes ocorrem de maneira habitual. Indivíduos de todas as idades estão suscetíveis a quedas.

Consoante a Gomes et al. (2014), os idosos institucionalizados podem apresentar até três vezes mais chances de cair dos que vivem em suas próprias residências, aos quais 39,8% possuem a idade entre 80 a 89 anos.

As quedas são vistas como um problema de saúde geral, pois está relacionada a altas taxas de morbimortalidade, destacando também o alto custo social e econômico, conforme dito por Oliveira (2014).

## Tecnologias Utilizadas

A seguir, serão listadas todas as tecnologias empregadas no desenvolvimento do projeto Safe Guardian, abrangendo linguagens de programação, frameworks e os componentes eletrônicos e outros recursos que constituirão o dispositivo.

### IOT

A Internet das Coisas é um conceito que se refere à conexão de objetos físicos à internet, permitindo a coleta e compartilhamento de dados. Esses objetos podem interagir entre si, criando um ambiente conectado (MAGRANI, 2018, p.20). O IoT tem sido amplamente reconhecida por sua capacidade de revolucionar diversos setores, desde a agricultura até a indústria, promovendo eficiência e melhorando a qualidade de vida (SINCLAIR, 2018).

#### ESP32

É um chip projetado para atuar como um microcontrolador em dispositivos móveis e aplicações de internet das coisas. Ele possui em sua construção um módulo que contém WI-FI e Bluetooth integrados (MARTINS, 2019).

Figura 1 – ESP32

Tela de um aparelho eletrônico

Descrição gerada automaticamente com confiança baixa

Fonte: Autoria Própria, 2024.

#### Módulo MPU6050

O sensor MPU6050 combina um acelerômetro e um giroscópio, permitindo medir a aceleração e o ângulo dos movimentos em três eixos (X, Y, Z). Frequentemente utilizado em smartphones para interpretação de gestos (VARASCHIM et al., 2022).

Figura 2 – Sensor MPU6050

Circuito eletrônico em superfície de madeira

Descrição gerada automaticamente com confiança baixa

Fonte: Autoria Própria, 2024.

#### Placa de Circuito Impresso (PCI)

Em subsequente a Engenharia Hibrida (2023) uma Placa de Circuito Impresso (PCI) é uma estrutura composta por camadas de condutores elétricos, interconexões metálicas, isolantes e outros componentes, que serve como base para o suporte e a conexão dos componentes montados em sua superfície.

Figura 3 - Placa PCI com barra de pinos

Uma imagem contendo circuito

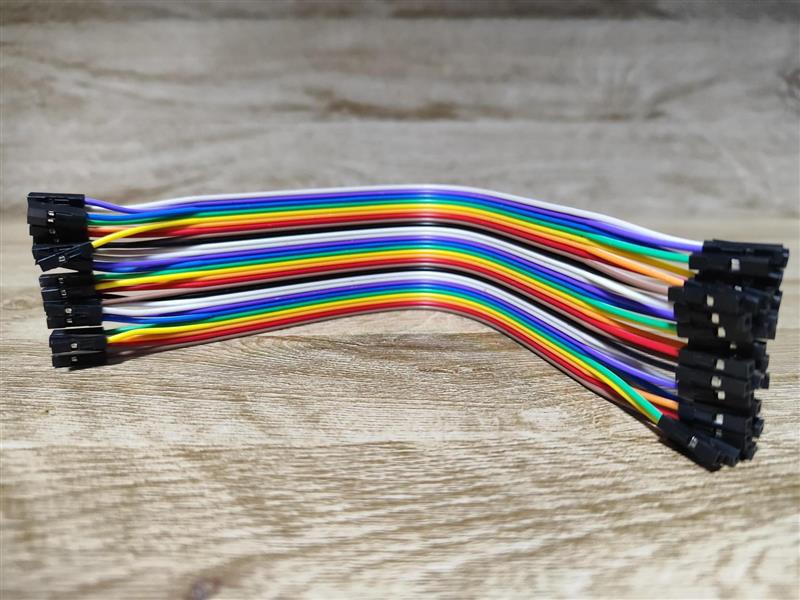
Descrição gerada automaticamente

Fonte: Autoria Própria, 2024.

#### Jumpers

São pequenos fios que tem como intuito interligar dois pontos do circuito em um projeto eletrônico (MENDES, 2022).

Figura 4 – Modelo Jumpers Fêmea-Fêmea



Fonte: Autoria Própria, 2024.

#### Resistores

Segundo MarkerHero (2024) resistores são componentes passivos que limitam ou controlam o fluxo de corrente elétrica em um circuito, oferecendo uma resistência específica, medida em ohms (Ω), que dificulta a passagem da corrente.

Figura 5 - Resistores 100k

Pingente de metal

Descrição gerada automaticamente com confiança baixa

Fonte: Autoria Própria, 2024.

#### Suporte de pilha

Componente eletrônico utilizado para conectar e sustentar pilhas em circuitos eletrônicos (SOLDAFRIA, 2024).

Figura 6 – Suporte de pilha

Uma imagem contendo mesa

Descrição gerada automaticamente

Fonte: Autoria própria, 2024.

#### Módulo TP4056

De acordo com MarkerHero (2023) o módulo TP4056 é um chip de carregamento que é usado para carregar baterias de polímero de lítio (Li-Po) ou íon de lítio (Li-Ion) . Ele é bem visto em circuitos eletrônicos portáteis.

Figura 7 - Módulo TP4056

Circuito eletrônico em superfície de madeira

Descrição gerada automaticamente com confiança média

Fonte: Autoria Própria, 2024.

#### Bateria recarregável Li-ion

De acordo com Iberdrola (2021) uma bateria de íon de lítio ou também conhecida por bateria de Li-ion é uma bateria recarregável que é constituído por lítio como um dos eletrodos.

Figura 8 - Bateria de Li-ion 18650

Uma imagem contendo no interior, banana, mesa, deitado

Descrição gerada automaticamente

Fonte: Autoria Própria, 2024.

### Modelagem 3D

Autodesk aponta que modelagem 3D, realizada por meio de software, consiste na criação de objetos ou formas tridimensionais, sendo utilizado por diversos setores. Nunes e Pereira (2015) descrevem que, na modelagem 3D, o posicionamento dos objetos ocorre em três eixos cartesianos: X (horizontal), Y (vertical) e Z (profundidade), permitindo movimentos em todas as direções.

Posteriormente à modelagem, é possível realizar a impressão 3D. Conforme Cunico (2015), a fabricação aditiva, ou impressão 3D, adiciona material em camadas conforme a geometria de um modelo 3D, geralmente criado via software CAD.

Figura 9 - Exemplo de modelagem 3D

Texto

Descrição gerada automaticamente com confiança média

Fonte**:** (SOARES, 2019).

### C++

O C++ é uma linguagem de baixo nível e orientada a objetos, sendo similar e compatível com o seu antecessor C. Devido ao seu poder e agilidade, a linguagem é amplamente utilizada na criação de programas em larga escala (DAVIS, 2016, p.2).

Além disso, por ser compilada diretamente na máquina, o C++ oferece alta performance em diversos sistemas operacionais. Essa proximidade com o hardware permite otimizar o código para aproveitar as capacidades específicas de cada plataforma (MARQUES, 2023).

Figura 10 – Exemplo simples de código de ligamento de leds

Texto

Descrição gerada automaticamente

Fonte: Autoria Própria, 2024.

A função de cada linha de código é:

* #define: Esse comando define constantes para os números dos pinos utilizados para as lâmpadas vermelha e azul.
* void setup(): Esta linha marca o início da função ‘setup()’, que é uma função especial no Arduino/ESP32 que é executada uma vez ao iniciar o programa.
* pinMode(): Configura os pinos especificados (4 e 2 respectivamente) como saídas, o que significa que serão utilizados para ligar e desligar as lâmpadas.
* Serial.begin(): Inicializa a comunicação serial com uma taxa de transmissão de 115200 bits por segundo.
* Serial.printIn(): Imprime uma mensagem na porta serial para indicar que as lâmpadas foram ligadas.
* Void loop(): Marca o início da função ‘loop()’, que é executada continuamente após a função ‘setup ()’.
* digitalWrite(): Define o estado (ligado/desligado) de um pino digital.
* delay(): Pausa a execução do programa por um determinado número de milissegundos.

Abaixo temos uma representação visual do código em ação, demonstrando a conexão dos LEDs ao ESP32.

Figura 11 – Representação visual do código de leds

Interface gráfica do usuário

Descrição gerada automaticamente

Fonte: Autoria Própria, 2024.

### UML - Unified Modeling Language

A UML, ou Linguagem de Modelagem Unificada,tumb é uma linguagem visual de modelagem que oferece um conjunto de diagramas padronizados para representar conceitos de criação de software de maneira compreensível (GUEDES, 2018, p.20).

Essa ferramenta, como destacado por Rumbaugh et al. (2006), é valiosa para análise e design de sistemas, permitindo que os desenvolvedores comuniquem efetivamente as estruturas de um sistema por meio de diagramas visuais.

Seguindo a afirmação de Booch et al. (2006), a UML é constituída por alguns diagramas essenciais, sendo eles: Diagrama de Casos de Uso, Diagrama de Atividade, Diagrama de sequência e, por fim, Diagrama de Máquina de estados.

O diagrama de casos de uso destaca as funcionalidades gerais oferecidas pelo sistema, sem adentrar nas especificações detalhadas dessas tais funcionalidades (GUEDES, 2018, p.60). Embora não se aprofunde em detalhes específicos, permanece sendo um recurso fundamental para a organização e modelagem do sistema (RUMBAUGH et al., 2006).

O diagrama de atividades é utilizado para exibir o fluxo de uma atividade para outra em um sistema. Ele enfatiza o controle de fluxo, representando sequências de ações e decisões e o envolvimento de objetos ou entidades (BOOCH et al., 2006).

O diagrama de sequência é uma representação visual da interação entre objetos em um processo, mostrando a ordem das mensagens trocadas. Ele se baseia em casos de uso e no diagrama de classes para ser construído (GUEDES, 2018, p.37).

Um diagrama de máquina de estados mostra o comportamento de um sistema através de estados, transições, eventos e atividades, sendo importante para modelar interfaces, classes ou colaborações (RUMBAUGH et al., 2006).

### FIGMA

Segundo Santos da Silva et al. (2020), o FIGMA é uma plataforma de design gráfico, acessível via navegador e compatível com diversas plataformas, permitindo a colaboração em tempo real entre equipes multidisciplinares e simplificando o uso individual por link. Como observado por Cadore (2021), o FIGMA permite a criação, desenvolvimento claro e objetivo de layouts, facilitando para o designer, uma vez que não requer instalação de software em dispositivos como o computador.

Conforme apresentado por Oliveira (2022), a ferramenta beneficia-se de uma comunidade ativa que cria sistemas de design, extensões e diretrizes de estilo para aprimorar o design de interfaces. O FIGMA facilita a criação de produtos digitais, tornando a interface do projeto mais prática, eficiente e de qualidade. Isso proporciona agilidade e satisfação aos usuários (Souza, 2023).

Figura 12 – Logotipo FIGMA



Fonte: (FIGMA, 2021).

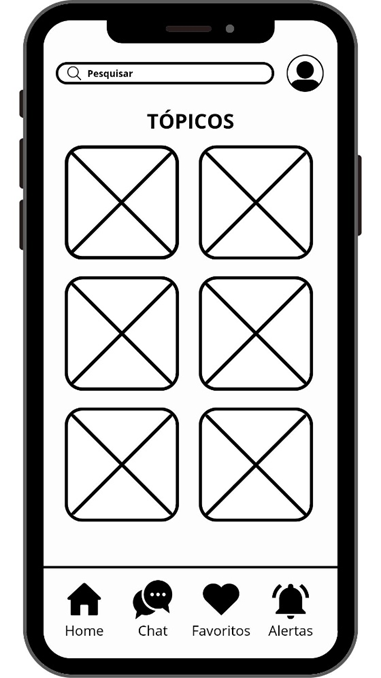
### Wireframe

Os wireframes são representações visuais que definem a estrutura de uma página, destacando sua hierarquia e elementos principais. Geralmente são criados em tons de cinza e sem imagens, auxiliando os desenvolvedores (TEIXEIRA, 2014).

Wireframe é um desenho bem simples e básico, muitas vezes feito à mão ou com ferramentas simples. Ele não se preocupa com detalhes precisos, apenas mostra formas gerais (RABELLO, 2023).

Segundo Giulya Vasconcelos (2023), wireframe de baixa fidelidade é um esboço simples e básico do layout, sem muitos detalhes. Já o wireframe de alta fidelidade, é um desenho mais detalhado e realista, incluindo imagens reais, textos e cores.

Figura 13 – Exemplo de wireframe de baixa fidelidade



Fonte: Autoria Própria, 2024.

Figura 14 – Exemplo de wireframe de alta fidelidade



Fonte: Autoria Própria, 2024.

### React

Segundo Silva (2021), o React é uma biblioteca que simplifica o processo de desenvolvimento de interfaces de usuário de maneira interativa. Essa biblioteca é amplamente adotada por empresas renomadas, como o Facebook.

Como apontado por Escudelario e Pinho (2020), o React é uma linguagem simples, pois seus conceitos fundamentais são poucos e fáceis de compreender.

Ao empregar o React para o desenvolver interfaces, ele simplifica a construção de telas por meio da utilização de componentes. Essas peças permitem criar partes reutilizáveis no código, tornando-o mais organizado e fácil de manter.

#### HTML

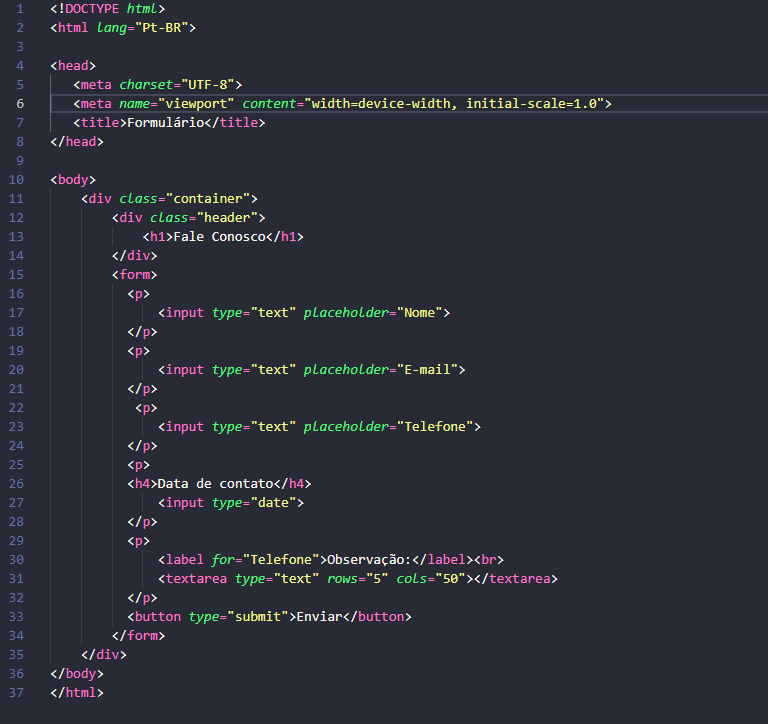
De acordo com Pedroso (2007), HTML, sigla dada para Hypertext Markup Language, é uma linguagem de marcação para hipertexto que permite a estruturação e organização dos elementos em uma página da web.

Segundo Patel (2023), o HTML tem o papel de organizar o conteúdo de uma página da web por meio de elementos como cabeçalhos, parágrafos, formulários e botões, cada um representado por tags.

As tags de um documento HTML incluem:

* Doctype: “<!DOCTYPE html>” é a declaração inicial obrigatória que indica ao navegador a versão do HTML utilizada, prevenindo incompatibilidades.
* HTML: “<html>” define o início e o fim do código HTML, encapsulando todo o conteúdo da página.
* Cabeçalho: Localizado na tag “<head>”, contém informações gerais sobre o documento, como título da página e referências a arquivos de estilização.
* Link: A tag “<link>” é utilizada para referenciar recursos externos ou arquivos de estilização no projeto.
* Script: “<script>” permite a inclusão de scripts JavaScript executáveis dentro do HTML, possibilitando interatividade na página.
* Estilo: “<style>” é utilizado para inserir código de estilização diretamente no HTML, sem a necessidade de um arquivo externo.
* Título da Página: Dentro da tag “<title>”, é declarado o título da página que aparece nas guias do navegador.
* Corpo: “<body>” contém todos os elementos visuais da página, sendo o espaço onde o conteúdo é exibido ao usuário.
* Títulos e Subtítulos: Utilizam as tags “<h1>” a “<h6>”, variando em tamanho, sendo “<h1>” o maior e “<h6>” o menor.
* Parágrafo: “<p>” define o início de um novo parágrafo de texto na página.
* Formulário: Iniciado com “<form>”, é utilizado para coletar dados do usuário, especificando o método de envio e ação do formulário.
* Caixa de Texto: Utiliza a tag “<input>” para criar campos de entrada de dados, onde o usuário insere informações.
* Botão: “<button>” cria um botão que executa uma ação quando clicado.

Figura 15 – Exemplo de código para a construção de um formulário em HTML



Fonte: Autoria Própria, 2024.

A figura abaixo representa o resultado do código disposto na figura 15.

Figura 16 – Resultado do formulário em HTML



Fonte: Autoria Própria, 2024.

#### Cascading Style Sheet (CSS)

O CSS tem como sua função original devolver uma estruturação de conteúdo em páginas da web, segundo Mauricio Silva (2015). A separação entre estrutura e apresentação no HTML facilita a organização do código.

O CSS é amplamente adotado devido à sua praticidade e capacidade de executar funções anteriormente realizadas em JavaScript, o que simplifica o desenvolvimento e aprimora a experiência do usuário na web (ZEMEL, 2015).

Seletores encadeados e agrupados são fundamentais no CSS, permitindo selecionar elementos com base em suas estruturas e condições específicas para aplicar formatação (EIS E FERREIRA, 2012).

Abaixo encontra-se um exemplo de codificação em CSS.

Figura 17 – Exemplo de código em CSS

Texto

Descrição gerada automaticamente

Fonte: Autoria Própria, 2024.

Diante desse código, obtemos a seguinte estilização do formulário mostrado acima na figura 16:

Figura 18 – Resultado da estilização em CSS

Interface gráfica do usuário

Descrição gerada automaticamente

Fonte: Autoria Própria, 2024.

#### JavaScript

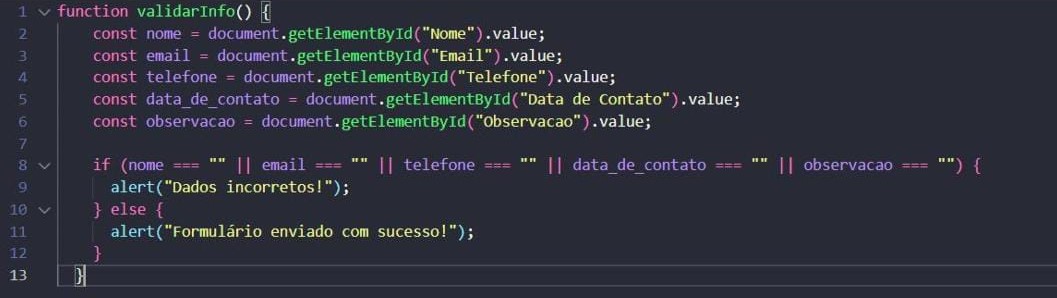
De acordo com Flanagan (2013), JavaScript é uma linguagem de alto nível, dinâmica e não tipada, que define o comportamento das páginas da web. Sua sintaxe é superficialmente semelhante à linguagem Java.

As funcionalidades do JavaScript são inúmeras. Com ele, é possível modificar o conteúdo e as características de uma página adicionando layouts complexos, melhorando a interação do navegador com o usuário (IEPSEN, 2022).

Em JavaScript, expressões são declarações que podem ser avaliadas pelo interpretador e não se limitam apenas a valores numéricos, incluindo variáveis, funções, strings e literais (SILVA, 2010).

Para que possamos exemplificar o funcionamento do JavaScript, a figura abaixo demonstrará um código de alertas para um formulário.

Figura 19 – Exemplo de código em JavaScript



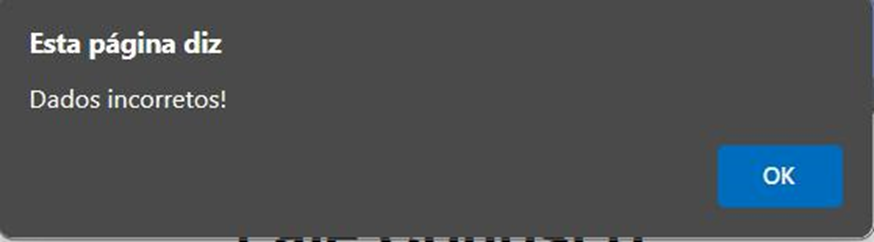
Fonte: Autoria Própria, 2024.

Diante desse código, podemos observar que se declara uma função que irá validar todos os campos do formulário, informando se foram preenchidos ou não.

As linhas 2 a 6 definem variáveis representando os campos do formulário, cujos valores são avaliados na linha 8 por meio de uma condicional para verificar se estão preenchidas ou vazias.

Em caso de o usuário não ter preenchido algum campo, a linha 9 será executada, e o seguinte alerta será emitido:

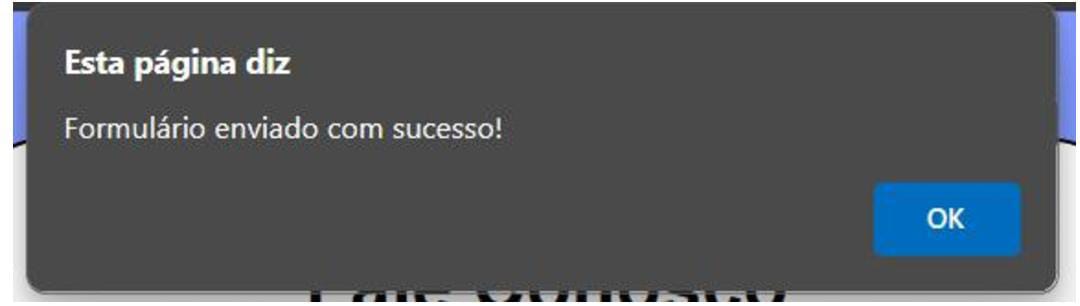
Figura 20 – Mensagem de erro da codificação em JavaScript



Fonte: Autoria Própria, 2024.

Se todos os campos estiverem preenchidos corretamente, a linha 9 será ignorada, e a linha 11 será executada, exibindo um alerta especificado abaixo:

Figura 21 – Mensagem de conclusão da codificação em JavaScript



Fonte: Autoria Própria, 2024.

Para garantir a execução eficiente do código JavaScript, é fundamental incorporá-lo dentro de uma marcação de script no documento HTML, conforme mostrado na imagem fornecida:

Figura 22 – Inserção do JavaScript no HTML



Fonte: Autoria Própria, 2024.

Para especificar qual função JavaScript deve ser executada quando o formulário é enviado, empregamos o atributo “onsubmit”, o qual incluirá o nome da função dada, conforme a imagem:

Figura 23 – Atributo "OnSubmit" em HTML



Fonte: Autoria Própria, 2024.

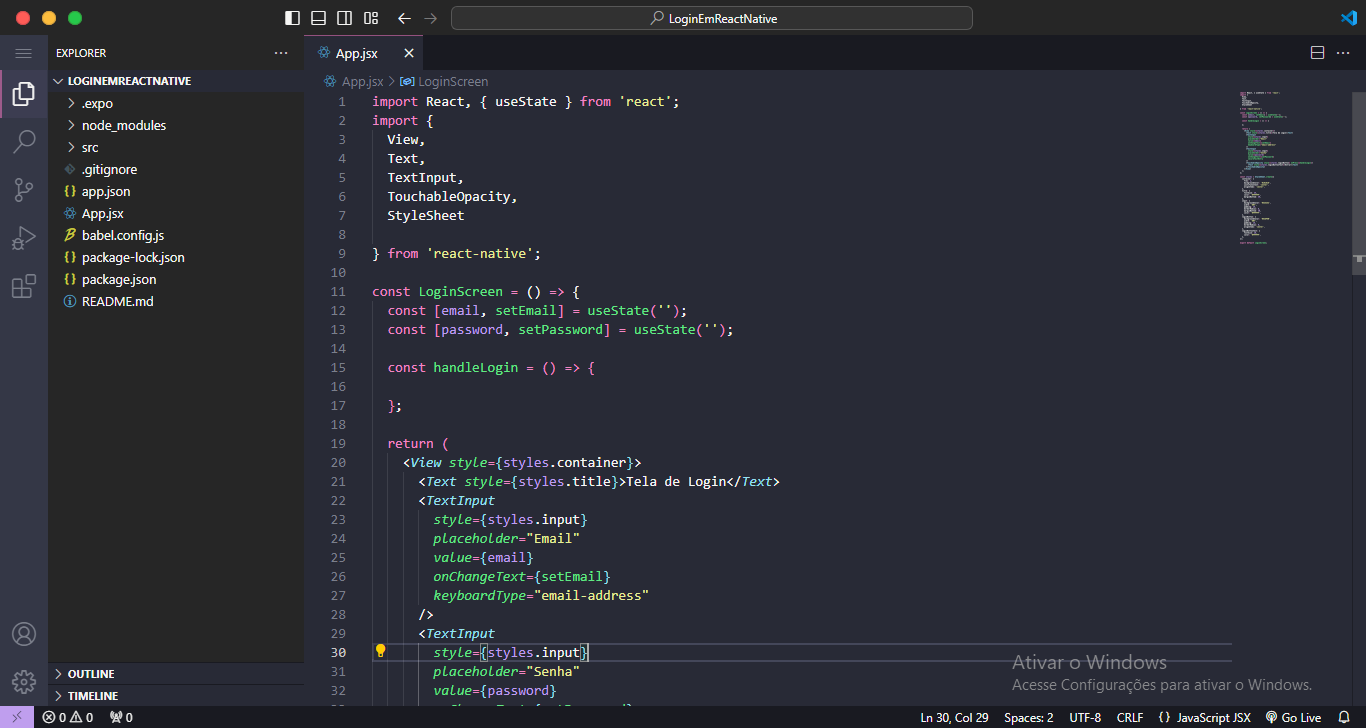
#### React Native

O React Native, criado pelo Facebook, é um framework que segue os princípios da biblioteca React. Conforme dito por Escudelario e Pinho (2020), o React Native foi desenvolvido durante uma maratona de programação interna em 2013 e lançado em 2015.

Como apontado por Costa Filho (2022), o React Native é uma biblioteca utilizada para desenvolver aplicativos que funcionem em diversos sistemas operacionais, como iOS ou Android, oferecendo assim, uma abordagem eficaz para o desenvolvimento de aplicativos.

De acordo com Falcão (2022), o React Native proporciona uma experiência de usuário aprimorada e um desempenho superior em comparação com outras tecnologias multiplataforma.

Figura 24 – Estrutura de Pasta

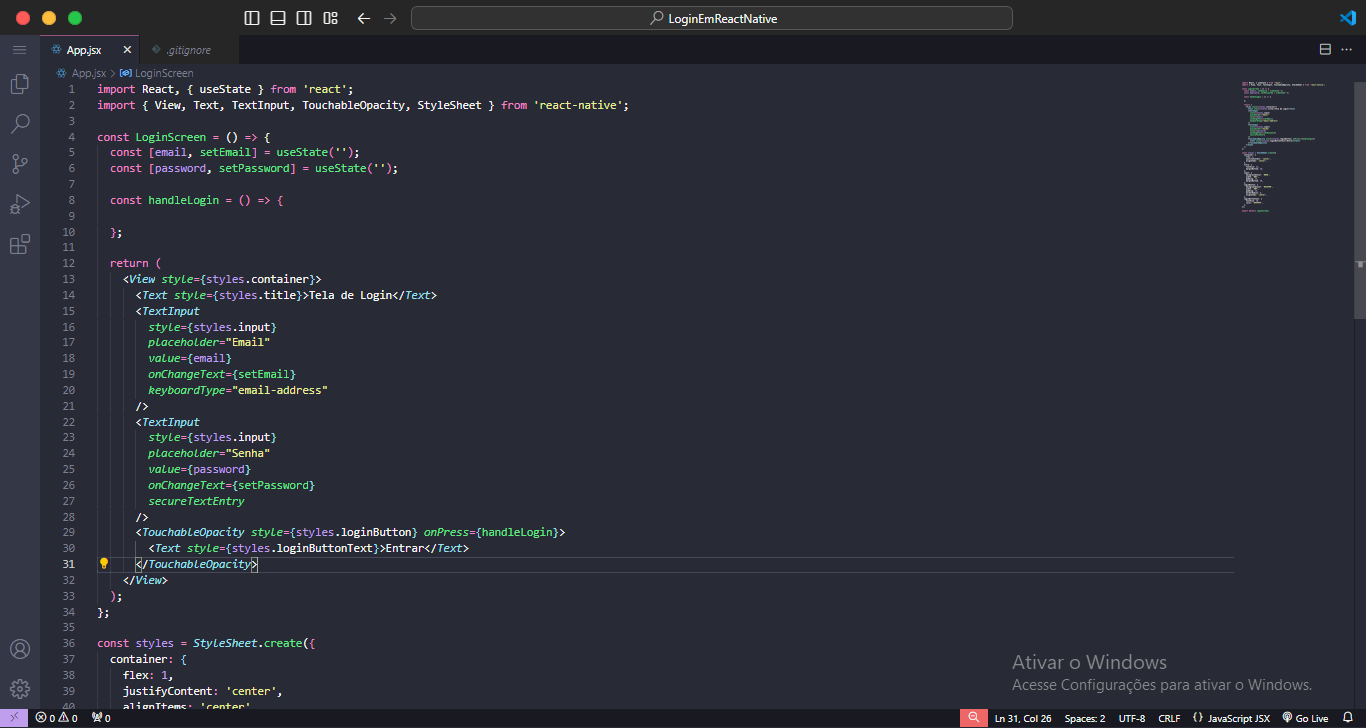


Fonte: Autoria Própria, 2024.

No exemplo dado, a estrutura de diretórios para desenvolver um formulário em React Native segue o seguinte padrão:

* .expo: Contém arquivos de configuração específicos do Expo, como chaves de API e configurações de ambiente.
* node\_modules: Contém todos os módulos de terceiros necessários para o projeto, como componentes e utilitários.
* src: Pasta onde fica todo o código-fonte do aplicativo, mantendo a organização e separação da lógica de negócios. Aqui estão arquivos como imagens, telas, estilos e serviços.
* app.json: Contém configurações globais para o aplicativo React Native, como nome, ícone e versão.
* Arquivos JSX: São arquivos JavaScript que contêm código escrito no formato JSX (JavaScript XML). JSX é uma extensão de sintaxe que permite escrever marcação XML (como HTML) dentro do JavaScript.
* babel.config.js: Define as configurações para o Babel, uma ferramenta para converter código ECMAScript 2015+ para versões compatíveis com JavaScript mais antigo.
* package-lock.json e package.json: Contêm informações sobre as dependências do projeto, scripts e outras configurações. O package.json especifica as dependências, scripts de build e versão do Node.js. O package-lock.json garante a consistência das versões das dependências entre diferentes ambientes.

Figura 25 – Exemplo de Formulário de Login utilizando React Native

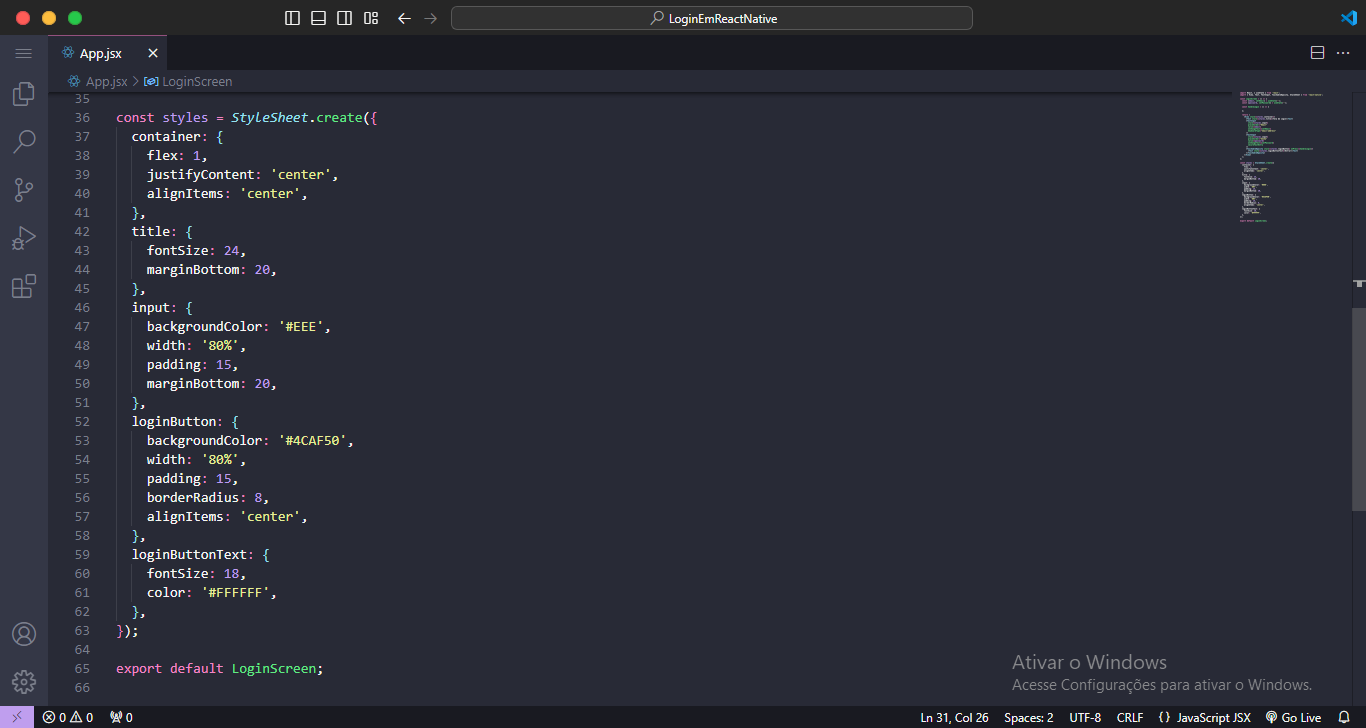


Fonte: Autoria Própria, 2024.

Neste exemplo de código, apresentamos uma tela de login desenvolvida utilizando a tecnologia React Native. Abaixo, teremos a explicação de cada linha de código:

* Importações: React e useState do pacote react.
* Componentes:<View>, <Text>, <TextInput>, <TouchableOpacity>, e `StyleSheet` do pacote react-native.
* <View>: Usada para criar uma estrutura de layout e agrupar outros componentes. No exemplo, é usada para envolver os campos de entrada de texto e o botão de login.
* <Text>: Exibe o texto "Tela de Login" com um estilo específico.
* `TextInput` para o Email: Cria um campo de entrada de texto para o email. O texto "Email" é definido como placeholder. O valor do campo é definido pelo estado `email`. A função `setEmail` é chamada sempre que o texto é alterado no campo de entrada. O teclado é configurado para exibir o layout adequado para inserção de endereços de email.
* <TextInput> para a Senha: Cria um campo de entrada de texto para a senha. O texto "Senha" é definido como placeholder. O valor do campo é definido pelo estado `password`. A função `setPassword` é chamada sempre que o texto é alterado no campo de entrada. O texto digitado é ocultado para proteger a privacidade do usuário.
* **`**TouchableOpacity**:** Cria um botão no aplicativo. No exemplo, é usado para criar um botão de login.
* Estados useState, email e setEmail: Armazenam e atualizam o valor do email digitado pelo usuário.
* password e setPassword: Armazenam e atualizam o valor da senha digitada pelo usuário.
* Função handleLogin: Será utilizada para lidar com a ação de login, mas está vazia neste exemplo do código e não executando nenhuma ação.
* Estilos StyleSheet:Contém definições de estilos para os componentes, como cor, tamanho, margens, entre outros estilos. Utilizados para estilizar o container, título, campos de entrada e botão.

Figura 26 – Continuação do formulário de login em React Native



Fonte: Autoria Própria, 2024.

* Objeto styles: Contém definições de estilos para os componentes da tela de login.
* Propriedade StyleSheet.create: Utilizada para criar um objeto StyleSheet, que contém estilos para os componentes React Native.
* Estilo container: Define o estilo para o contêiner principal da tela de login.

flex: 1: Faz com que o contêiner ocupe todo o espaço disponível na tela.

justifyContent: “center”: Centraliza os itens verticalmente dentro do contêiner.

* Estilo title: Define o estilo para o título da tela de login.

fontSize: 24: Define o tamanho da fonte como 24 pixels.

* marginBottom: 20: Adiciona uma margem inferior de 20 pixels para separar o título dos campos de entrada.
* Estilo input: Define o estilo para os campos de entrada de texto (email e senha).

backgroundColor: '#EEE': Define a cor de fundo como cinza claro.

width: '80%': Define a largura como 80% da largura total disponível.

padding: 15: Adiciona preenchimento interno de 15 pixels para os campos de entrada.marginBottom: 20: Adiciona uma margem inferior de 20 pixels para separar os campos de entrada.

* Estilo loginButton: Define o estilo para o botão de login.

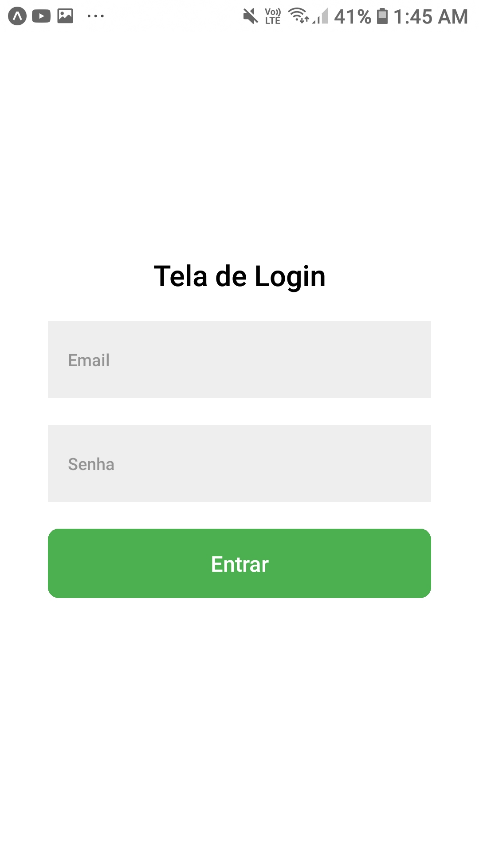
backgroundColor: '#4CAF50': Define a cor de fundo como verde.

padding: 15: Adiciona preenchimento interno de 15 pixels para o botão. borderRadius: 8: Adiciona cantos arredondados ao botão com raio de 8 pixels.

alignItems: “center”: Centraliza os itens horizontalmente dentro do botão.

* Estilo loginButtonText: Define o estilo para o texto dentro do botão de login. fontSize: 18: Define o tamanho da fonte como 18 pixels. Color: '#FFFFFF': Define a cor do texto como branca.

Figura 27 – Resultado do formulário utilizando React Native



Fonte: Autoria Própria, 2024.

### Firebase

De acordo com Machado (2021), o Firebase, através do Cloud Firestore, facilita a administração e consulta de dados em escala global. Suas funcionalidades permitem organizar dados em coleções, priorizando segurança e escalabilidade.

Conforme Andrade (2018) menciona, o Firebase é uma plataforma "backend as a service" desenvolvida pela infraestrutura do Google, ajudando os desenvolvedores a acelerarem o processo de desenvolvimento. É amplamente utilizada em aplicativos móveis.

Ximenes et al. (2023) salientam que, ao contrário dos bancos de dados convencionais, o Firestore opera com um modelo de dados flexível. As informações são estruturadas em documentos e coleções, facilitando o trabalho do desenvolvedor.

Figura 28 – Logotipo Firebase



Fonte: (FIREBASE GOOGLE, 2024).

Figura 29 – Console Firebase

Fonte: Autoria Própria, 2024.

### API CallMeBot

De acordo com a documentação da Mozilla Developer (2024), APIs são construções disponíveis em linguagens de programação que facilitam a criação de funcionalidades complexas ao abstrair o código. Assim, permitindo o uso de sintaxes mais simples.

Segundo Romeiro (2022), a API “CallMeBot” é um serviço de redirecionamento de mensagens que não tem custo, desde que os limites estabelecidos pelo fornecedor da ferramenta não sejam ultrapassados.

Figura 30 – Logotipo da API CallMeBot

Placa branca com letras pretas em fundo branco

Descrição gerada automaticamente com confiança baixa

Fonte: (BOTOSTORE, 2022).

# DESENVOLVIMENTO

No presente capítulo, serão apresentados os principais aspectos do processo de criação do sistema Safe Guardian, com foco na utilização dos métodos de documentação UML, no desenvolvimento das telas do aplicativo móvel, bem como na concepção e prototipagem do dispositivo físico. Serão descritos os diagramas UML, como caso de uso, classe, atividade e máquina de estado, essenciais para o planejamento da arquitetura do sistema. Além disso, serão detalhadas as interfaces do aplicativo e as funcionalidades do dispositivo, mostrando como ambos se integram para oferecer uma solução eficiente na detecção de quedas.

## Diagrama de Caso de Uso

Nessa seção, será realizada uma análise detalhada do diagrama de caso de uso do sistema Safe Guardian, um dos principais elementos da documentação UML. O diagrama de caso de uso demonstrará as interações entre os atores externos do sistema, como o cuidador, e os componentes internos, sendo eles o Safe Guardian — dispositivo físico, e a API CallMeBot. Ele nos permite visualizar como cada funcionalidade é acionada e quais são as responsabilidades de cada ator dentro do ecossistema do sistema, sendo elas 9 elipses ao todo.

Figura 31 - Caso de Uso do sistema



Fonte: Autoria Própria, 2024

### Documentação dos Casos de Uso

Conforme explicado anteriormente, a seguir apresentamos a documentação dos casos de uso com base na UML. A documentação está organizada em tópicos para facilitar a compreensão, abordando: Requisitos Funcionais, Requisitos Não Funcionais e Regras de Negócio.

Requisitos funcionais especificam as funcionalidades que o sistema deve oferecer, como a detecção de quedas e a notificação ao cuidador, garantindo que o Safe Guardian cumpra seus objetivos principais. Já os requisitos não funcionais determinam características de qualidade, como desempenho, segurança e usabilidade, garantindo que o sistema funcione de maneira eficiente e acessível. As regras de negócio estabelecem as diretrizes e restrições que o sistema deve seguir, garantindo que ele esteja alinhado aos objetivos e políticas operacionais, melhorando sua consistência e confiabilidade para o cliente.

Requisitos Funcionais da API CallMeBot:

* RF01: Enviar mensagem.

Requisitos Funcionais do Dispositivo Safe Guardian:

* RF01: Detectar a queda;
* RF02: Gerenciar bateria.

Requisitos Funcionais do Cuidador:

RF01: Notificar bateria baixa do dispositivo;

RF02: Confirmar resolução da queda;

RF03: Manter dados emergenciais;

RF04: Visualizar histórico de quedas.

Requisitos Funcionais Sistema:

* RF01: Notificar queda;
* RF02: Emparelhar dispositivo.

Requisitos Não Funcionais do Sistema:

* RNF01: A interface do sistema deve ser intuitiva e fácil de navegar.
* RNF02: A API CallMeBot deverá ser acionada de forma eficaz assim que identificado uma queda;

Regras de Negócio do Sistema:

• É obrigatório o cadastramento de pelo menos 1 parente ou cuidador do utilizador do sistema.

Tabelas descritivas dos casos de uso:

#### Documentação dos Casos de Uso da API

Quadro 1 - Documentação do caso de uso: Enviar Mensagem

|  |  |
| --- | --- |
| **Nome do Caso de Uso** | Enviar Mensagem |
| **Ator Principal** | API CallMeBot |
| **Resumo** | A API CallMeBot será responsável por enviar as mensagens de alerta quando necessário. |
| **Pré-condições** | O sistema deve detectar uma queda. |
| **Cenário Principal** | |
| **Ações do Ator** | **Ações do Sistema** |
|  | 1. O sistema deve identificar a necessidade de enviar a notificação. |
|  | 2. A API CallMeBot é acionada. |
| 3. A API envia a mensagem para o WhatsApp do cuidador. |  |

Fonte: Autoria Própria, 2024

#### Documentação dos Casos de Uso do Dispositivo

Quadro 2 - Documentação do caso de uso: Detectar a Queda

|  |  |
| --- | --- |
| **Nome do Caso de Uso** | Detectar a Queda |
| **Ator Principal** | Dispositivo Safe Guardian |
| **Resumo** | O dispositivo deve detectar uma queda do usuário. |
| **Pré-condições** | O idoso deve estar utilizando o dispositivo. |
| **Cenário Principal** | |
| **Ações do Ator** | **Ações do Sistema** |
| 1. O dispositivo detecta a queda. |  |
| 2. Um alerta é gerado e enviado para o cuidador. |  |
|  | 3. O cuidador é notificado via mensagem. |
| **Cenário de Exceção – Falso Positivo** | |
| **Ações do Ator** | **Ações do Sistema** |
| 1. O dispositivo detecta um tropeço busco. |  |
|  | 2. O cuidador é notificado e verifica a situação para confirmar ou desconsiderar o evento. |

Fonte: Autoria Própria, 2024

Quadro 3 - Documentação do caso de uso: Gerenciar Bateria

|  |  |
| --- | --- |
| **Nome do Caso de Uso** | Gerenciar Bateria |
| **Ator Principal** | Dispositivo Safe Guardian |
| **Resumo** | O dispositivo monitora o nível de bateria. |
| **Pré-condições** | O dispositivo deve estar em operação contínua. |
| **Cenário Principal** | |
| **Ações do Ator** | **Ações do Sistema** |
| 1. O dispositivo monitora constantemente o nível de bateria. |  |
| 2. Se a bateria estiver baixa, um alerta é gerado. |  |
|  | 3. O alerta é enviado ao cuidador por meio do aplicativo móvel. |

Fonte: Autoria Própria, 2024

#### Documentação dos Casos de Uso do Cuidador

Quadro 4 - Documentação do caso de uso: Notificar Bateria Baixa do Dispositivo

|  |  |
| --- | --- |
| **Nome do Caso de Uso** | Notificar Bateria Baixa do Dispositivo |
| **Ator Principal** | Cuidador |
| **Resumo** | O sistema notifica o cuidador quando o nível de bateria estiver baixo. |
| **Pré-condições** | O dispositivo detectou que a bateria está em um nível crítico. |
| **Cenário Principal** | |
| **Ações do Ator** | **Ações do Sistema** |
|  | 1. O dispositivo envia uma notificação de bateria baixa ao cuidador. |
| 2. O cuidador recebe a notificação pelo aplicativo. |  |
|  | 3. O cuidador recarrega o dispositivo através de um cabo USB C. |

Fonte: Autoria Própria, 2024

Quadro 5 - Documentação do caso de uso: Confirmar Resolução da queda

|  |  |
| --- | --- |
| **Nome do Caso de Uso** | Confirmar resolução da queda |
| **Ator Principal** | Cuidador |
| **Resumo** | Após o recebimento de um alerta de queda, o cuidador deve confirmar se a situação for resolvida. |
| **Pré-condições** | Uma queda deve ser detectada e notificada. |
| **Cenário Principal** | |
| **Ações do Ator** | **Ações do Sistema** |
| 1. O cuidador recebe a notificação da queda. |  |
| 2. O cuidador verifica a condição do idoso. |  |
| 3. O cuidador confirma no sistema que a situação foi resolvida. |  |

Fonte: Autoria Própria, 2024.

Quadro 6 - Documentação do caso de uso: Manter Dados Emergenciais

|  |  |
| --- | --- |
| **Nome do Caso de Uso** | Manter Dados Emergenciais |
| **Ator Principal** | Cuidador |
| **Resumo** | O sistema permite que o cuidador mantenha os dados de emergência atualizados. |
| **Pré-condições** | O cuidador tem acesso ao sistema de gerenciamento do dispositivo por meio da aplicação móvel. |
| **Cenário Principal** | |
| **Ações do Ator** | **Ações do Sistema** |
| 1. O cuidador acessa o aplicativo. |  |
| 2. O cuidador atualiza os dados do idoso através da aba de informações pessoais. |  |
|  | 3. O sistema atualiza e salva as informações no banco de dados. |
| **Cenário de Exceção – Erro ao Salvar** | |
| **Ações do Ator** | **Ações do Sistema** |
|  | 1. O sistema envia um erro na tela sobre o erro de conexão. |
| 2. O cuidador deve tentar enviar os dados novamente para verificar se não houve erro de digitação. |  |
|  | 3. O sistema salva os dados corretamente. |

Fonte: Autoria Própria, 2024.

Quadro 7 - Documentação do caso de uso: Visualizar Histórico de Quedas

|  |  |
| --- | --- |
| **Nome do Caso de Uso** | Visualizar Histórico de Quedas |
| **Ator Principal** | Cuidador |
| **Resumo** | O cuidador visualiza o histórico de quedas do usuário através do aplicativo. |
| **Pré-condições** | O usuário já teve quedas registradas no sistema. |
| **Cenário Principal** | |
| **Ações do Ator** | **Ações do Sistema** |
| 1. O cuidador acessa o aplicativo. |  |
| 2. O cuidador acessa a aba de histórico de quedas. |  |
|  | 3. O sistema exibe um formulário contendo as informações da queda (como horário, data, nome do idoso, responsável). |
| **Cenário de Exceção – Não há quedas registradas** | |
| **Ações do Ator** | **Ações do Sistema** |
| 1. O cuidador acessa a aba de histórico de quedas. |  |
|  | 2. O sistema não exibe nenhum dado, pois ainda não houve nenhuma queda para registrar. |

Fonte: Autoria Própria, 2024.

#### Documentação dos Casos de Uso Automático

Quadro 8 - Documentação do caso de uso: Emparelhar Dispositivo

|  |  |
| --- | --- |
| **Nome do Caso de Uso** | Emparelhar Dispositivo |
| **Ator Principal** | Automático |
| **Resumo** | O sistema irá emparelhar o dispositivo utilizando uma rede Wi-Fi. |
| **Pré-condições** | O dispositivo físico deve estar ativo. |
| **Cenário Principal** | |
| **Ações do Sistema** | |
|  |  |
|  |  |
|  |  |
|  |  |
| **Cenário de Exceção -** | |
| **Ações do Ator** | **Ações do Sistema** |
|  |  |
|  |  |

Fonte: Autoria Própria, 2024.

Quadro 9 - Documentação do caso de uso: Notificar queda

|  |  |
| --- | --- |
| **Nome do Caso de Uso** | Notificar Queda |
| **Ator Principal** | Sistema Safe Guardian |
| **Resumo** |  |
| **Pré-condições** |  |
| **Cenário Principal** | |
| **Ações do Ator** | **Ações do Sistema** |
|  |  |
|  |  |
|  |  |
|  |  |
| **Cenário de Exceção -** | |
| **Ações do Ator** | **Ações do Sistema** |
|  |  |
|  |  |

Fonte: Autoria Própria, 2024.

## Diagrama de Atividade

O diagrama de atividade descreve o fluxo de ações de forma clara e visual contendo as decisões dentro do processo do sistema, combinando de forma sequencial as etapas que serão realizadas por cada funcionalidade do sistema.

Figura 32 - Diagrama de Atividade: Confirmar Resolução da Queda

Diagrama

Descrição gerada automaticamente

Fonte: Autoria Própria, 2024.

Figura 33 - Diagrama de Atividade: Detectar a Queda

Diagrama

Descrição gerada automaticamente

Fonte: Autoria Própria, 2024.

Figura 34 - Diagrama de Atividade: Emparelhar Dispositivo

Diagrama

Descrição gerada automaticamente

Fonte: Autoria Própria, 2024.

Figura 35 - Diagrama de Atividade: Enviar Mensagem

Diagrama

Descrição gerada automaticamente

Fonte: Autoria Própria, 2024.

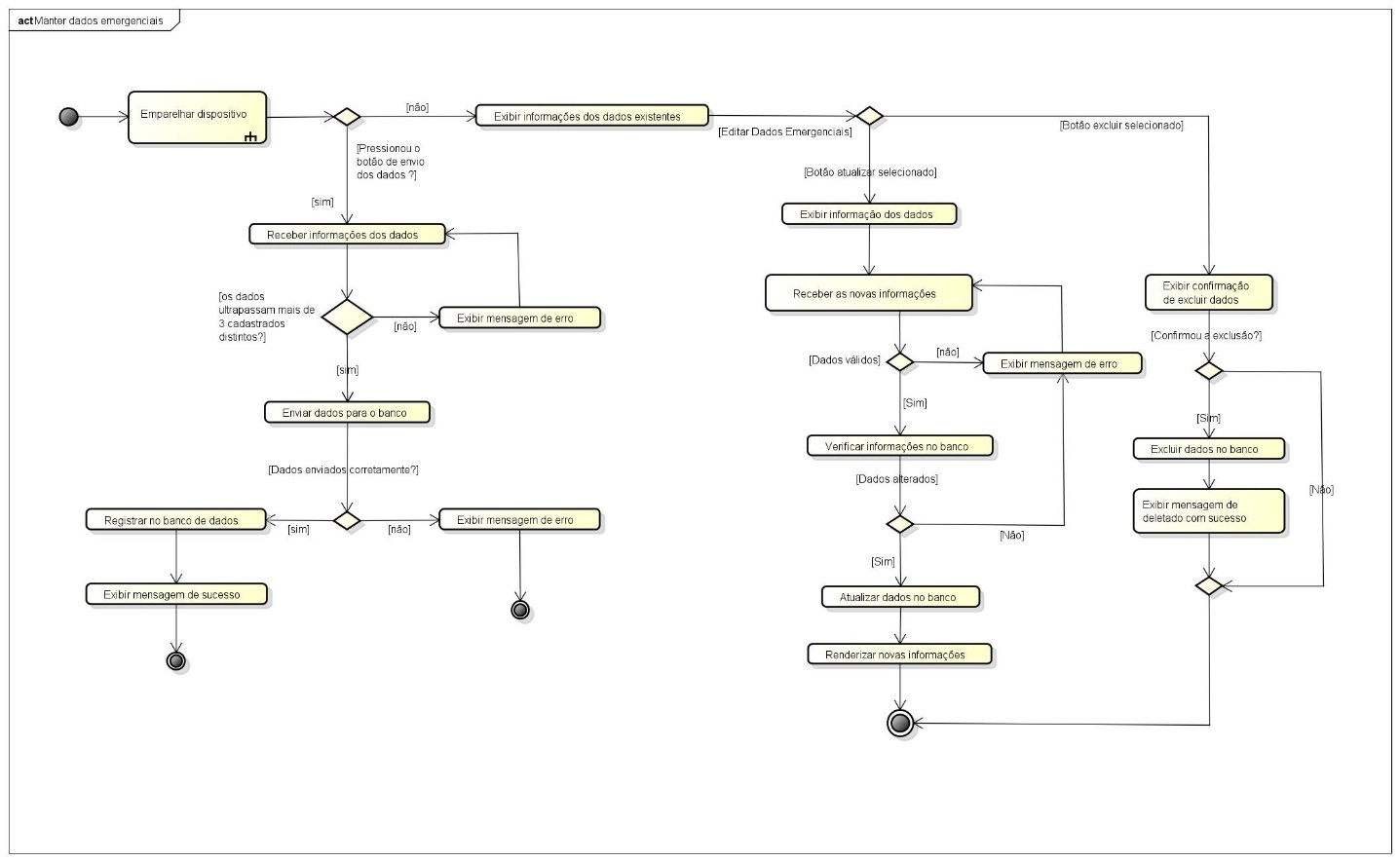
Figura 36 - Diagrama de Atividade: Gerenciar Bateria

Diagrama

Descrição gerada automaticamente

Fonte: Autoria Própria, 2024.

Figura 37 - Diagrama de Atividade: Manter Dados Emergenciais



Fonte: Autoria Própria, 2024.

Figura 38 - Diagrama de Atividade - Notificar Bateria Baixa do Dispositivo

Diagrama

Descrição gerada automaticamente

Fonte: Autoria Própria, 2024.

Figura 39 - Diagrama de Atividade: Notificar Queda

Diagrama

Descrição gerada automaticamente

Fonte: Autoria Própria, 2024.

Figura 40 - Diagrama de Atividade: Visualizar Histórico de Quedas

Diagrama

Descrição gerada automaticamente

Fonte: Autoria Própria, 2024.

## Diagrama de Sequência

Por sua vez, o diagrama de sequência mostra, de forma ordenada, as interações entre os principais componentes do sistema: Dispositivo físico, aplicativo móvel, banco de dados, API CallMeBot e os cuidadores. Foram elaborados 9 diagramas, detalhando como o sistema combina suas funcionalidades. Esses diagramas oferecem uma visão clara do funcionamento do sistema, assegurando eficiência e rapidez nas repostas.

Figura 41 - Diagrama de Sequência: Confirmar Resolução da Queda

Diagrama

Descrição gerada automaticamente

Fonte: Autoria Própria, 2024.

Figura 42 - Diagrama de Sequência: Detectar a Queda

Diagrama

Descrição gerada automaticamente

Fonte: Autoria Própria, 2024.

Figura 43 - Diagrama de Sequência: Emparelhar Dispositivo

Diagrama

Descrição gerada automaticamente

Fonte: Autoria Própria, 2024.

Figura 44 - Diagrama de Sequência: Enviar Mensagem

Diagrama

Descrição gerada automaticamente

Fonte: Autoria Própria, 2024.

Figura 45 - Diagrama de Sequência: Gerenciar Bateria

Diagrama

Descrição gerada automaticamente

Fonte: Autoria Própria, 2024.

Figura 46 - Diagrama de Sequência: Manter Dados Emergenciais

Diagrama

Descrição gerada automaticamente

Fonte: Autoria Própria, 2024.

Figura 47 - Diagrama de Sequência: Notificar Bateria Baixa do Dispositivo

Diagrama

Descrição gerada automaticamente

Fonte: Autoria Própria, 2024.

Figura 48 - Diagrama de Sequência: Notificar Queda

Diagrama

Descrição gerada automaticamente

Fonte: Autoria Própria, 2024.

Figura 49 - Diagrama de Sequência: Visualizar Histórico de Quedas

Diagrama

Descrição gerada automaticamente

Fonte: Autoria Própria, 2024.

## Diagrama de Máquina de Estado

O diagrama de máquina de estados apresenta de forma clara e visual as diversas condições e transições do sistema, destacando as respostas a eventos específicos. Este diagrama organiza sequencialmente os estados que o sistema pode assumir, juntamente com as ações correspondentes, proporcionando uma compreensão aprofundada do seu comportamento em diferentes situações.

Figura 50 - Diagrama de Máquina de Estado: Visualizar histórico de Quedas

Diagrama

Descrição gerada automaticamente

Fonte: Autoria Própria, 2024.

Figura 51 - Diagrama de Máquina de Estado: Confirmar Resolução de Queda

Diagrama

Descrição gerada automaticamente

Fonte: Autoria Própria, 2024.

Figura 52 - Diagrama de Máquina de Estado: Enviar Mensagem

Diagrama

Descrição gerada automaticamente

Fonte: Autoria Própria, 2024.

Figura 53 - Diagrama de Máquina de Estado: Gerenciar Bateria

Diagrama

Descrição gerada automaticamente

Fonte: Autoria Própria, 2024.

Figura 54 - Diagrama de Máquina de Estado: Notificar Bateria baixa do dispositivo

Diagrama

Descrição gerada automaticamente

Fonte: Autoria Própria, 2024.

Figura 55 - Diagrama de Máquina de Estado: Notificar Queda

Diagrama

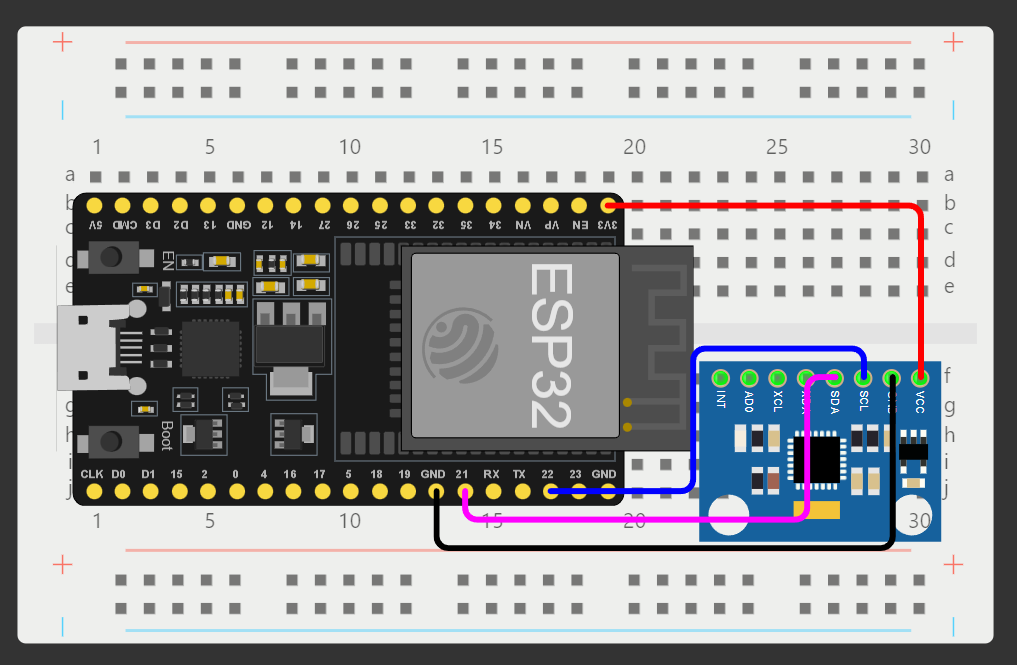
Descrição gerada automaticamente

Fonte: Autoria Própria, 2024.

## Protótipo do dispositivo físico

Nesta seção, será apresentada uma figura que ilustra a interconexão e a organização dos componentes eletrônicos do dispositivo físico do sistema Safe Guardian, projetado para detectar quedas. Essa representação visual ajudará a compreender a estrutura e o funcionamento do sistema.

Figura 56 - Protótipo do dispositivo físico Safe Guardian



Fonte: Autoria Própria, 2024.

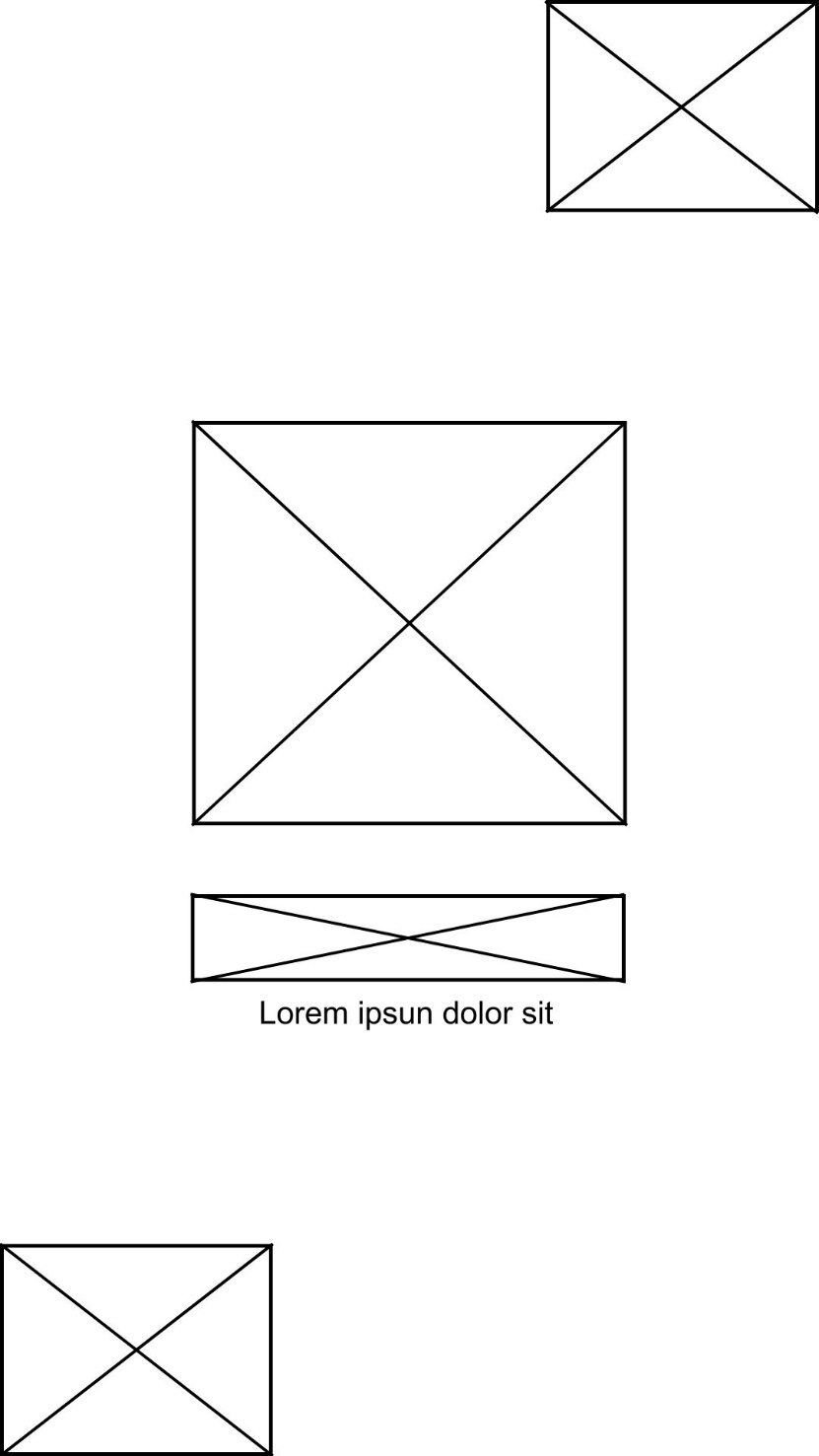
## Prototipação das telas do aplicativo

Esta seção apresenta as interfaces gráficas dos elementos visuais do aplicativo mobile, elaboradas com base em wireframes de baixa e alta fidelidade. Cada detalhe foi minuciosamente projetado para garantir uma experiência visual de excelência, priorizando o conforto e a usabilidade para os usuários, sejam cuidadores ou familiares do idoso ou indivíduo suscetível.

O primeiro wireframe criado exibe a tela de carregamento do aplicativo. No centro da tela, estão posicionados o logotipo da marca e o nome do sistema, acompanhados de uma frase de impacto logo abaixo, reforçando os valores e a missão do produto.

Este rascunho inicial da tela, um wireframe de baixa fidelidade, destaca a estrutura e o posicionamento básico dos elementos. Ele serve como uma referência para futuras refinamentos no design e na experiência do usuário:

Figura 57 - Wireframe de Baixa Fidelidade: Tela de carregamento



Fonte: Autoria Própria, 2024.

Subsequente, este é o wireframe de alta fidelidade, com detalhes visuais e interativos próximos do design final:

Figura 58 - Wireframe de Alta Fidelidade: Tela de carregamento



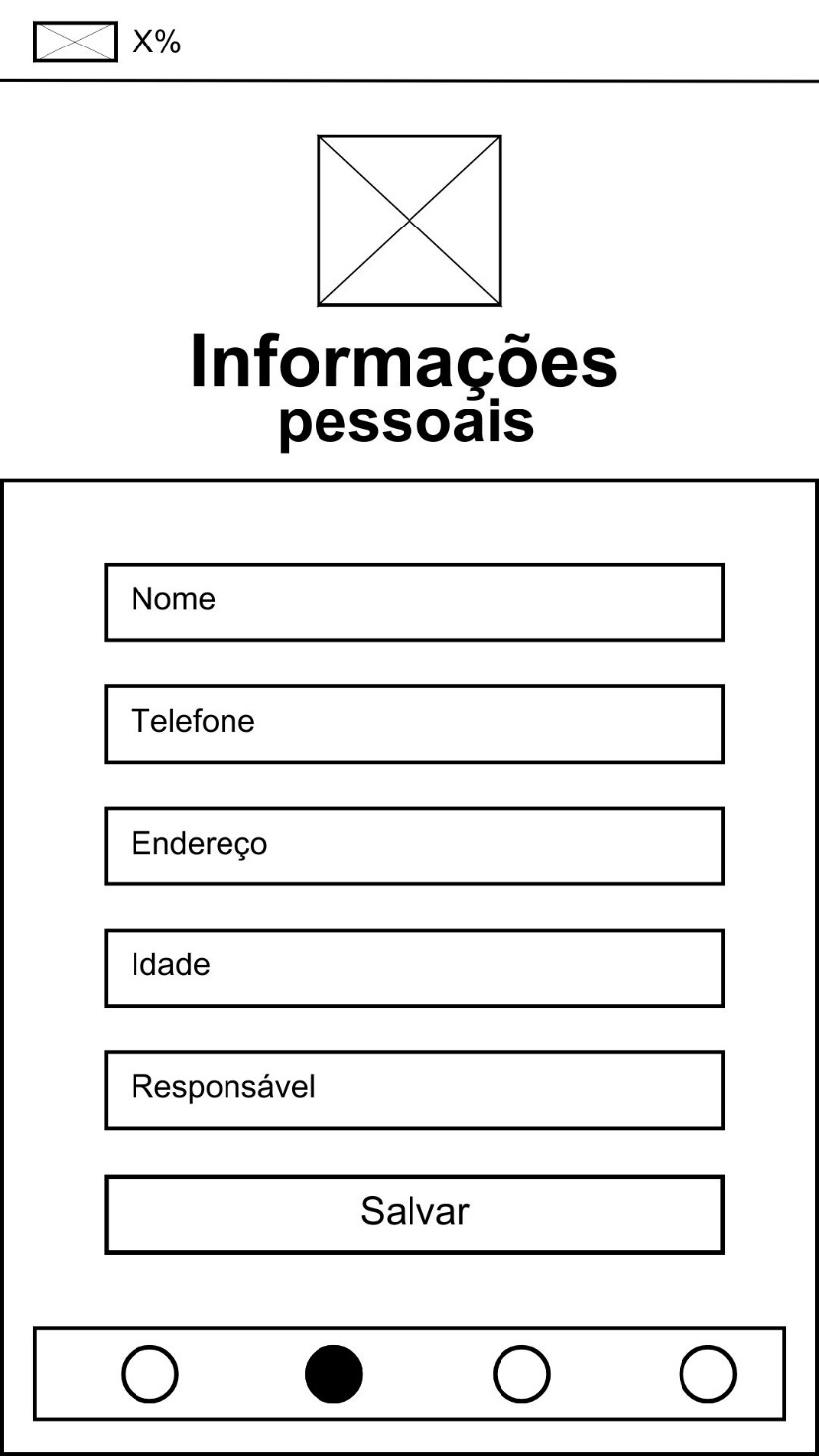
Fonte: Autoria Própria, 2024.

O segundo wireframe apresenta um carrossel com imagens, acompanhado de um breve texto explicativo abaixo. Este espaço é dedicado para compartilhar nossa missão e introduzir a equipe, destacando o comprometimento por trás do nosso produto. Aqui temos uma representação preliminar da tela, usando um wireframe de baixa fidelidade. Este esboço define a base para o layout e organização dos componentes, sem detalhes visuais avançados:

IMAGEM X E Y

O terceiro wireframe apresenta um formulário destinado ao preenchimento das informações pessoais do usuário do dispositivo físico. A tela inclui um botão de salvar para armazenar os dados inseridos, além de uma barra de menu na parte inferior para facilitar a navegação, e conter uma navbar simples indicando o nível de bateria do dispositivo físico. Na figura abaixo é possível visualizar a versão esboço da tela:

Figura 59 - Wireframe de Baixa Fidelidade: Tela de informações pessoais



Fonte: Autoria Própria, 2024.

Na próxima figura, vemos a versão de alta fidelidade que simula a experiência final do usuário, com detalhes completos de layout e estilo:

Figura 60 - Wireframe de Alta Fidelidade: Tela de informações pessoais

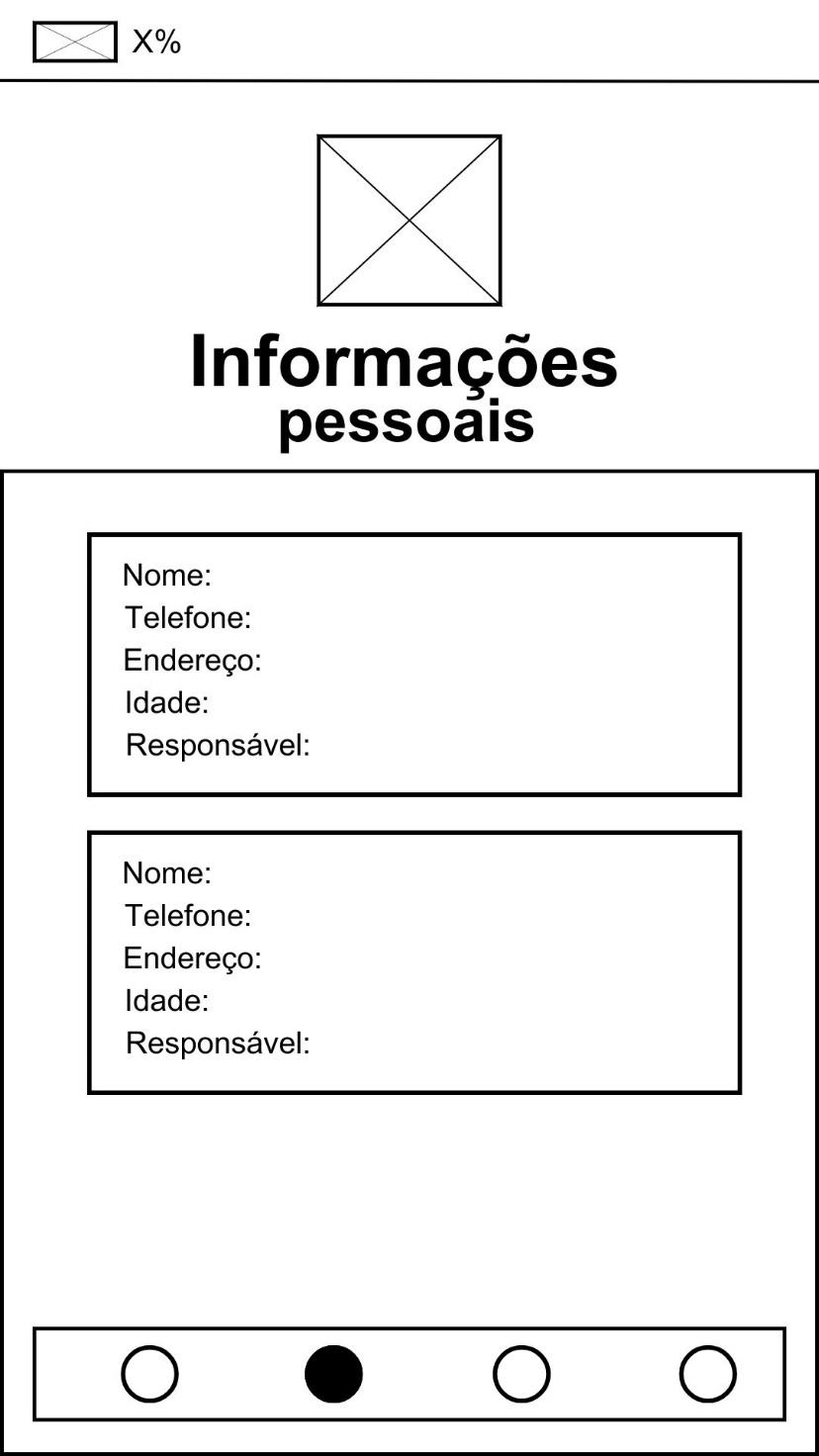
Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

Fonte: Autoria Própria, 2024.

Após o preenchimento dos dados no terceiro wireframe, a próxima tela exibirá as informações salvas, permitindo ao usuário visualizar os dados registrados de forma organizada. A imagem X demonstra o wireframe de baixa fidelidade da tela de dados das informações pessoais:

Figura 61 - Wireframe de Baixa Fidelidade: Tela de dados de Informações Pessoais



Fonte: Autoria Própria, 2024.

Em subsequente, a figura abaixo exibe o resultado final da tela através do wireframe de alta fidelidade:

Figura 62 - Wireframe de Alta Fidelidade: Tela de dados de Informações Pessoais

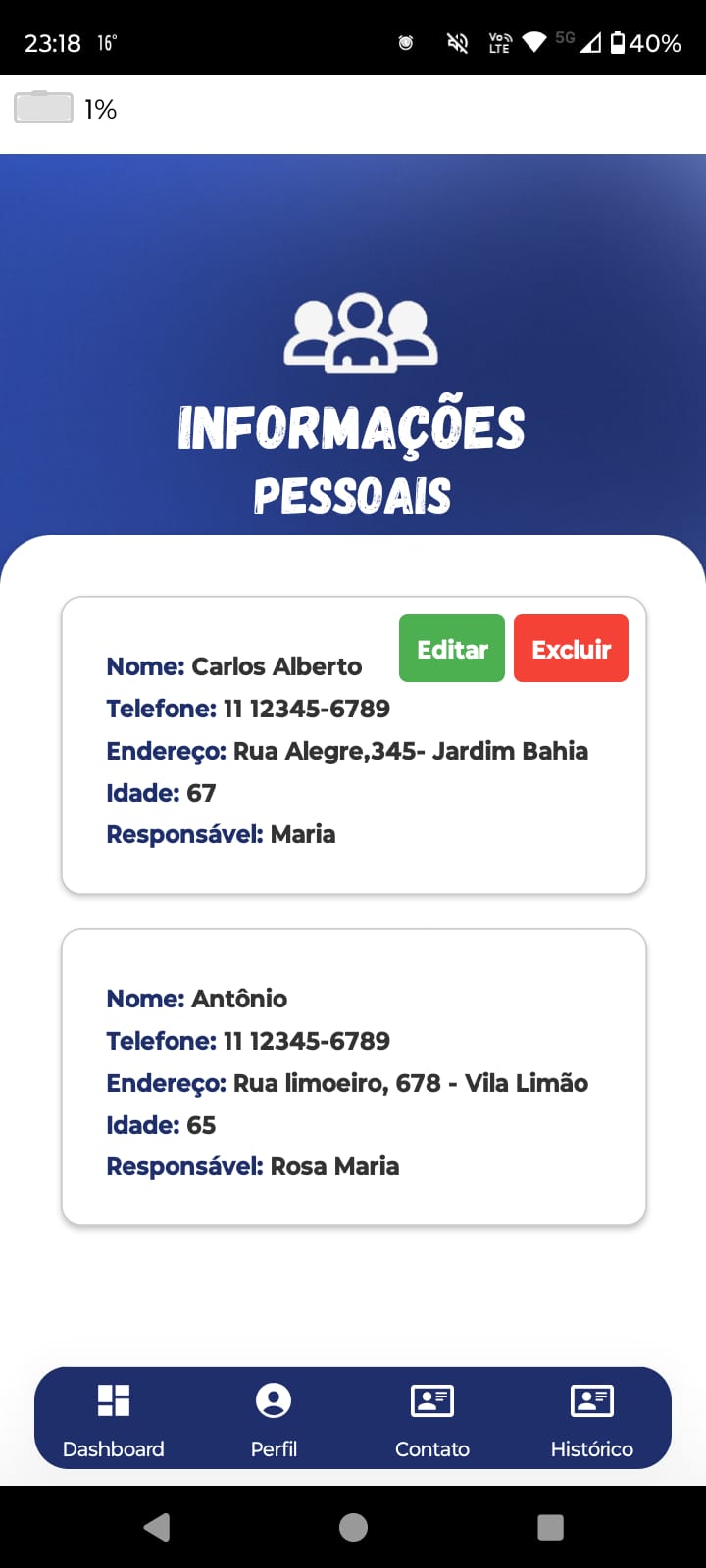
Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

Fonte: Autoria Própria, 2024.

Texto, Carta

Descrição gerada automaticamente



Diagrama, Texto, Carta

Descrição gerada automaticamenteDiagrama

Descrição gerada automaticamente

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

Forma

Descrição gerada automaticamente



Diagrama

Descrição gerada automaticamente com confiança média



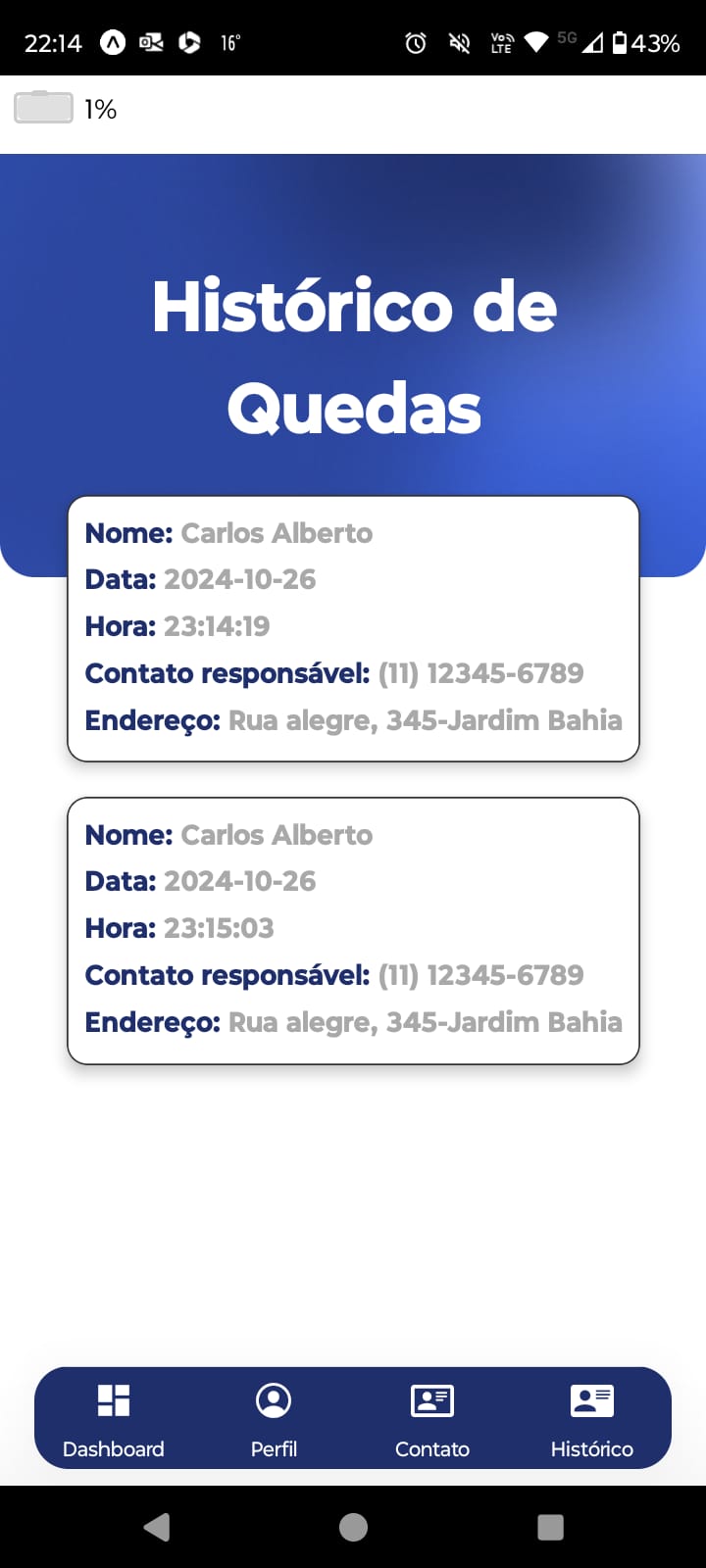
Diagrama

Descrição gerada automaticamente



Carta

Descrição gerada automaticamente com confiança baixa



Diagrama

Descrição gerada automaticamente

# CONSIDERAÇÕES FINAIS

A conclusão do projeto de desenvolvimento do sistema de detecção de quedas para idosos e indivíduos suscetíveis destaca a importância de soluções tecnológicas para enfrentar os desafios de uma população envelhecida, assegurando a segurança e a autonomia dos usuários. O sistema proposto, ao integrar hardware e software, oferece uma resposta rápida em caso de quedas, com o intuito de reduzir ao menos uma pequena parcela das consequências graves associadas a esses eventos.

Apesar de algumas limitações, como a necessidade de aprimorar a precisão dos sensores em ambientes adversos e otimizar o desempenho do aplicativo em dispositivos de baixa capacidade, futuramente, nosso sistema está comprometido em abordar essas lacunas. Esses desafios, embora possam parecer pequenos, têm o potencial de impactar significativamente a vida dos idosos.

As perspectivas futuras para o projeto incluem a melhoria da comunicação entre dispositivos e a integração com bancos de dados em tempo real, mais robustos e especializados. Uma pesquisa mais aprofundada sobre bancos de dados que armazenam informações relevantes para o público mais velho pode oferecer maior segurança, escalabilidade e atualizações constantes, elevando ainda mais as funcionalidades do sistema.

Além disso, a implementação de um chip de geolocalização para identificar a localização exata onde o usuário sofreu a queda é uma sugestão que pode ser crucial para as equipes de emergência, permitindo uma resposta mais rápida e eficiente. Outro avanço significativo seria a aplicação de técnicas de machine learning para analisar padrões de movimento e prever quedas antes que ocorram, aprimorando a prevenção de incidentes.

Essas inovações permitirão ao sistema não apenas detectar quedas, mas também monitorar continuamente os usuários, possibilitando intervenções rápidas e personalizadas. Assim, o projeto não apenas atende às necessidades imediatas, mas também abre caminho para soluções mais abrangentes, que podem melhorar a qualidade de vida e a segurança de idosos e pessoas suscetíveis a quedas. Em última análise, estamos empenhados em transformar essas pequenas vulnerabilidades em oportunidades de aprimoramento, assegurando que a segurança dos idosos nunca seja comprometida. Esperamos que nosso projeto contribua para a redução, mesmo que mínima, da incidência de acidentes decorrentes de quedas entre idosos e indivíduos suscetíveis. Cada porcentagem a menos representa uma vida mais segura e um aumento na qualidade de vida dos usuários, refletindo nosso compromisso com a proteção e o bem-estar dessa população.

# REFERÊNCIAS

ANDRADE, Giovana Lorena Costa d**e. Desenvolvimento em nuvem: um estudo de caso utilizando o Firebase como servidor backend**. 2018. Trabalho de Conclusão de Curso (Graduação em Ciência da Computação) – Universidade do Estado do Rio Grande do Norte, Faculdade de Ciências Exatas e Naturais, Departamento de Informática, Mossoró, RN, 2018. Disponível em: <https://di.uern.br/tccs/html/ltr/PDF/014005697.pdf.Acesso> em: 05 de maio 2024.

AUTODESK**. Software de Modelagem 3D**. Disponível em: https://www.autodesk.com.br/solutions/3d-modeling-software. Acesso em: 17 de maio de 2024.

BOOCH, Grady; RUMBAUGH, James. **UML: guia do usuário**. 2. ed. Rio Janeiro: Elsevier, 2006. 474 p. Tradução Fábio Freitas da Silva e Cristina de Amorim Machado. Disponível em: https://www.google.com.br/books/edition/UML/ddWqxcDKGF8C?hl=pt-BR&gbpv=0&kptab=overview. Acesso em: 04 maio 2024.

BOTOSTORE. **CallMeBot API: Telegram bot para envio de mensagens e chamadas de voz.** Botostore, 2024. Disponível em: https://botostore.com/c/callmebot\_txtbot/. Acesso em: 13 jul. 2024.

CADORE, Douglas Cristiano. **UX DESIGN: um estudo sobre o desenvolvimento de produtos e serviços focado na experiência dos usuários.** 2021. 72 f. TCC (Graduação) - Curso de Administração, Universidade de Caxias do Sul, Guaporé, 2021. Cap. 60. Disponível em: https://repositorio.ucs.br/xmlui/handle/11338/9383. Acesso em: 03 maio 2024.

COSTA FILHO, Gilvan Ferreira. **Desenvolvimento multiplataforma com react native:** um estudo de caso do aplicativo vegbook. 2022. 42 f. TCC (Graduação-Bacharelado) - Curso de Sistemas e Mídias Digitais, Universidade Federal do Ceará, Fortaleza, 2022. Cap. 15. Disponível em: <https://repositorio.ufc.br/handle/riufc/68135> Acesso em: 27 abr. 2024.

CUNICO, Marlon Wesley Machado. **Impressoras 3D: o novo meio produtivo.** Curitiba: Concep3D Pesquisas Científicas, 2015. 172 p. Disponível em: https://www.google.com.br/books/edition/Impressoras\_3D/CybwCQAAQBAJ?hl=pt-BR&gbpv=0. Acesso em: 02 maio 2024.

DAVIS, Stephen R. **C++: para leigos**. 7. ed. Brasil: Alta Books, 2019. 472 p.

EIS, Diego; FERREIRA, Elcio. **HTML E CSS3 com Farinha e Pimenta.** São Paulo: Tableless, 2012. 139p. Disponível em: https://github.com/jorgeclesio/LTP-FADESA/blob/master/html5-e-css3-com-farinha-e-pimenta-diego-eis-e-elcio-ferreira.pdf. Acesso em 18 maio 2024.

ELETROGATE. **Suporte para 4 baterias 18650**. Disponível em: https://www.eletrogate.com/suporte-para-4-baterias-18650-serie. Acesso em 05 maio 2024.

ENGENHARIA HÍBRIDA. **Placas eletrônicas: o que são e para que servem,** 2023. Disponível em: https://www.engenhariahibrida.com.br/post/placas-eletronicas-o-que-sao-para-que-servem. Acesso em: 12 set. 2024.

ESCUDELARIO, Bruna; PINHO, Diego. **React Native**: desenvolvimento de aplicativos mobile com react. Brasil: Casa do Código, 2020. 209 p. Disponível em: <https://www.google.com.br/books/edition/React_Native/keDdDwAAQBAJ?hl=en&gbpv=0>. Acesso em: 24 abr. 2024.

FALCÃO, Filipe Dourado. **DESENVOLVIMENTO DO APLICATIVO TURISTANDO BEBERIBE**: utilizando react native. 2022. 67 f. TCC (Graduação) - Curso de Sistemas e Mídias Digitais, Universidade Federal do Ceará, Fortaleza, 2022. Cap. 21. Disponível em: <https://repositorio.ufc.br/handle/riufc/69029>. Acesso em: 28 abr. 2024.

FIGMA. **Figma SVG Logo & Wordmark** Disponível em: <https://www.figma.com/community/file/930374612850356203/figma-svg-logo-wordmark>. Acesso em 05 de maio 2024.

FIREBASE GOOGLE. **Como representar a marca Firebase.** Disponível em:https://firebase.google.com/brand-guidelines?hl=pt-br. Acesso em: 13 de maio 2024.

FLANAGAN, David. **JavaScript – O Guia Definitivo.** 6 ed. Porto Alegre: Bookman, 2013. 4 p. Disponível em: https://github.com/free-educa/books/blob/main/books%2FJavaScript%20O%20Guia%20Definitivo\_v2.pdf. Acesso em 19 maio 2024.

GOMES, Erika Carla Cavalcanti et al. Fatores associados ao risco de quedas em idosos institucionalizados: uma revisão integrativa. **Ciência & Saúde Coletiva**, v. 19, p. 3543-3551, 2014.

Gonçalves ICM, Freitas RF, Aquino EC, Carneiro JA, Lessa AC. **Tendência de mortalidade por quedas em idosos, no Brasil, no período de 2000–2019.** Rev Bras Epidemiol. 2022; 25:e220031. https://doi.org/10.1590/1980-549720220031.2. Acessado em: 16 de Mar. 2024.

GUEDES, Gilleanes Thorwald Araujo. **UML 2: uma abordagem prática.** 3. ed. São Paulo: Novatec Editora, 2018. 496 p. Disponível em: https://www.google.com.br/books/edition/UML\_2\_Uma\_Abordagem\_Pr%C3%A1tica\_3

IBERDROLA. **Baterias de lítio-ion Como funcionam?** Disponível em: https://www.iberdrola.com/inovacao/baterias-ion-litio. Acesso em: 11 de ago. 2024.

IEPSEN, Edécio Fernando. **Lógica de Programação e Algoritmos com JavaScript: Uma Introdução à Programação de Computadores com Exemplos e Exercícios Para Iniciantes.** 2 ed. São Paulo: Novatec Editora, 2022. 13 p. Disponível em: https://books.google.com.br/books?id=35ZkEAAAQBAJ&printsec=frontcover&hl=pt-BR&source=gbs\_ge\_summary\_r&cad=0#v=onepage&q&f=false. Acesso em 19 maio de 2024

MACHADO, Kheronn Khennedy. **Angular 11 e Firebase**: construindo uma aplicação integrada com a plataforma do google. São Paulo: Casa do Código, 2021. 177 p. Disponível em: https://www.google.com.br/books/edition/Angular\_11\_e\_Firebase/cRi9DwAAQBAJ?hl=pt-BR&gbpv=0. Acesso em: 28 abr. 2024.

MAGRANI, Eduardo. **A Internet das Coisas**. Rio de Janeiro: Fgv Editora, 2018. 192 p. Disponível em: https://repositorio.fgv.br/server/api/core/bitstreams/b50af2ba-b001-4b1d-a1ad-5df985f6d1bb/content. Acesso em: 30 abr. 2024.

MAIA, Bruna Carla et al. Consequências das quedas em idosos vivendo na comunidade. **Revista brasileira de geriatria e gerontologia**, v. 14, p. 381-393, 2011.

MARKERHERO. **O que é Resistor: funcionamento, calculadora e tabela de cores.** Disponível em: https://www.makerhero.com/guia/componentes-eletronicos/resistor/. Acesso em: 11 de ago. 2024.

MARKERHERO. **Saiba como funciona o carregador de bateria TP4056.** Disponível em: https://www.makerhero.com/blog/saiba-como-funciona-o-modulo-tp4056/. Acesso em: 11 de ago. de 2024.

MARQUES, Jideon. **O Curso de Codificação C++:** Santa Catarina: Clube de Autores, 2023. 658 p. Disponível em: https://www.google.com.br/books/edition/O\_Curso\_De\_Codificação\_C++/QJbqEAAAQBAJ?hl=pt-BR. Acesso em: 01 maio 2024.

MARTINS, Victor Ferreira. **Automação Residencial usando protocolo MQTT, Node-RED e Mosquitto Broker com ESP32 e ESP8266**. 2019. 53 f. TCC (Graduação) - Curso de Engenharia de Controle e Automação, Universidade Federal de Uberlândia, Uberlândia, 2019. Disponível em: https://repositorio.ufu.br/handle/123456789/28522. Acesso em: 1 maio 2024.

MENDES, A. R. C., Hiroshi, J., OLIVEIRA, L. P. D., ROMA, D. S., & SILVA, E. V. D. A. D. (2022). **Sistema de detecção de enchentes Antares.**

MOZILLA, Developer . **Introdução às Web APIs.** Disponível em:https://developer.mozilla.org/pt-BR/docs/Learn/JavaScript/Client-side\_web\_APIs/Introduction. Acesso em: 02 de maio 2024.

NUNES, Eunice P. dos Santos; PEREIRA, Fernando Lima (org.). Tendências e Técnicas em Sistemas Computacionais. **Comunicação Por Campo de Proximidade: Tecnologia, Aplicações e Questões de Segurança**, Cuiabá, v. 1, n. 1, p. 1-161, nov. 2015. Anual. Disponível em: https://www.researchgate.net/profile/Silvio-Quincozes-2/publication/360614842\_Tendencias\_e\_Tecnicas\_em\_Sistemas\_Computacionais\_-\_Comunicacao\_por\_campo\_de\_proximidade\_Tecnologia\_aplicacoes\_e\_questoes\_de\_seguranca/links/62817c1c4f1d90417d6f5624/Tendencias-e-Tecnicas-em-Sistemas-Computacionais-Comunicacao-por-campo-de-proximidade-Tecnologia-aplicacoes-e-questoes-de-seguranca.pdf#page=10. Acesso em: 17 maio 2024.

OLIVEIRA, Adriana Sarmento de et al. Fatores ambientais e risco de quedas em idosos: revisão sistemática. **Revista Brasileira de Geriatria e Gerontologia**, v. 17, p. 637-645, 2014.

OLIVEIRA, George Moreno de. **DESENVOLVIMENTO E AVALIAÇÃO DO PLUGIN PARA O FIGMA PARA DOCUMENTAÇÃO DE ACESSIBILIDADE PARA INTERFACES - DAI**. 2022. 79 f. TCC (Graduação) - Curso de Design Digital, Campus de Quixadá, Universidade Federal do Ceará, Quixadá, 2022. Disponível em: http://repositorio.ufc.br/handle/riufc/65589. Acesso em: 01 maio 2024.

ORGANIZAÇÃO PAN-AMERICANA DA SAÚDE. **Década do Envelhecimento Saudável nas Américas (2021-2030).** Disponível em: https://www.paho.org/pt/decada-do-envelhecimento-saudavel-nas-americas-2021-2030#:~:text=A%20população%20mundial%20está%20envelhecendo,até%20o%20final%20do%20século. Acessado em: 17 Mar. 2024

PATEL, Neil. **Códigos HTML: Do Básico ao Avançado (+Principais Tags).** [S.I.].NEILPATEL, 2023. Disponível em: https://neilpatel.com/br/blog/tags-html/?amp. Acesso em 18 maio de 2024.

PEDROSO, Robertha P. **APOSTILA DE HTML.** Niterói: PETele, 2007. 5 p. https://www.telecom.uff.br/pet/petws/downloads/apostilas/HTML.pdf. Acesso em 18 maio de 2024.

RABELLO, Guilherme. **Wireframe: o que é e como criar um?** [S.l.].Siteware, 22 out. 2023. Disponível em: https://www.siteware.com.br/blog/projetos/wireframe/. Acesso em: 06 maio 2024.

ROMEIRO, Lucas de Araújo Wanderley. **Sistema de baixo custo para monitoramento e alerta remoto da temperatura de medicamentos termolábeis, com base em tecnologia Edge Computing, BLE e IoT**. 2022. 53 f. TCC (Graduação) - Curso de Engenharia Elétrica, Engenharia Elétrica (Ft Ene), Universidade de Brasília, Brasília, 2022. Disponível em: http://www.rlbea.unb.br/jspui/handle/10482/44061. Acesso em: 10 maio 2024.

SANTOS DA SILVA, Gustavo et al. **Em direção a gamificação com o APP REVIEW.ME**. FACITEC, v. 11, n. 1, 15 jan. 2020. Disponível em: https://estacio.periodicoscientificos.com.br/index.php/e-revistafacitec/article/view/1858. Acesso em: 08 maio 2024.

SILVA, Maurício Samy. **Fundamentos de HTML5 e CSS3.** 1 ed. São Paulo: Novatec Editora, 2015.

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. São Paulo: Novatec Editora, 2021. 240 p.

SINCLAIR, Bruce. **IoT**: **Como usar a "internet das coisas" para alavancar seus negócios.** São Paulo: Autêntica Business, 2018. 240 p. Disponível em: https://www.google.com.br/books/edition/IoT\_Como\_Usar\_a\_Internet\_Das\_Coisas\_Para/sklVDwAAQBAJ?hl=pt-BR&gbpv=0. Acesso em: 01 maio 2024.

SOARES, Claudio. **Modelagem 3D**. Disponível em: https://claudiosoaresdesigner.com.br/modelagem-3d.Acesso em: 05 de maio 2024.

SOLDAFRIA. **Suporte de Pilhas e Bateria**. Disponível em: <https://www.soldafria.com.br/componentes-eletronicos/suporte-de-pilha-e-bateria#:~:text=Suporte%20de%20pilha%20ou%20suporte,e%20baterias%20em%20circuitos%20eletrônicos>. Acesso em: 02 maio 2024.

SOUZA, Alex Sandro Alves de. **Estudo de viabilidade para implantação de um aplicativo de carona compartilhada na Universidade Federal do Ceará do Campus Quixadá**. 2023. 56 f. TCC (Graduação) - Curso de Ciência da Computação, Campus de Quixadá, Universidade Federal do Ceará, Quixadá, 2022. Cap. 17. Disponível em: <http://repositorio.ufc.br/handle/riufc/75543>. Acesso em: 07 maio 2024.

TEIXEIRA, Fabricio. **Introdução e boas práticas em UX DESIGN.** 1 ed. São Paulo: Casa do Código, 2014.

VARASCHIM, F. H., Soares, J. W. M., de Aguiar Possoli, F. A., Pivovar, L. E., Breganon, R., & Alves, U. N. L. T. (2022). **UTILIZAÇÃO DE UM FILTRO COMPLEMENTAR JUNTO AO SENSOR MPU 6050**. Revista Mundi Engenharia, Tecnologia e Gestão (ISSN: 2525-4782), 7(7). Disponível em: https://www.periodicos.ifpr.edu.br/index.php/MundiETG/article/view/2341/992. Acesso em: 26 abr. 2024.

VASCONCELLOS, Giulya. **Wireframe: tipos, exemplos e ferramentas.** [S.l.]. KXPtech, 24 maio 2023. Disponível em: https://kxptech.com/o-que-e-wireframe. Acesso em: 06 abr. 2024.

XIMENES, Letícia DN et al. Desenvolvimento de extensão para o Chrome com persistência em nuvem utilizando Firebase. **Sociedade Brasileira de Computação**, 2023. Disponível em: https://sol.sbc.org.br/livros/index.php/sbc/catalog/download/132/573/885-1. Acesso em: 13 de maio 2024.

ZEMEL, Tárcio. **CSS Eficiente - Técnicas e Ferramentas que Fazem a Diferença nos Seus Estilos.** 1 ed. São Paulo: Casa do Código, 2015.