

**CENTRO ESTADUAL DE EDUCAÇÃO TECNOLÓGICA PAULA
SOUZA**

ESCOLA TÉCNICA ESTADUAL DA ZONA LESTE

MTEC Desenvolvimento De Sistemas AMS

Amanda Farias da Rocha

Beatriz Silva de Andrade

Carlos Henrique Rodrigues Barile

SONORIS: Sistema de Auxílio à Pessoa Com Deficiência Auditiva

São Paulo

2025

Amanda Farias da Rocha

Beatriz Silva de Andrade

Carlos Henrique Rodrigues Barile

SONORIS: Sistema de Auxílio à Pessoa Com Deficiência Auditiva

Trabalho de Conclusão de Curso
apresentado ao MTEC Desenvolvimento de
Sistemas AMS da Etec da Zona Leste, orientado
pelo Prof. Esp. Jeferson Roberto de Lima, como
requisito parcial para a obtenção do título de técnico
em Desenvolvimento de Sistemas.

São Paulo

2025

SONORIS

Sistema de Auxílio ao Deficiente Auditivo

Amanda Farias da Rocha

Beatriz Silva de Andrade

Carlos Henrique Rodrigues Barile

Aprovada em __/__/____.

BANCA EXAMINADORA

Prof. Esp. Jeferson Roberto de Lima

Universidade do Jeferson

Prof. (Professor avaliador)

Universidade do Avaliador

Prof. (Professor avaliador)

Universidade do Avaliador

DEDICATÓRIA

Dedico (amigos e familiares surdos)

AGRADECIMENTOS

Em primeiro lugar, somos gratos a Deus por toda a saúde e sabedoria que ele nos proporciona.

Além disso, estamos profundamente agradecidos aos nossos pais e irmãos por todo o incentivo e o suporte que nos foi dado durante os 3 anos do curso de desenvolvimento de sistemas AMS e no decorrer da confecção do trabalho de conclusão de curso.

Agradecemos a todos os nossos amigos, em especial à Sakiri Moon Cestari, Lucas Bonfim e Kelvyn Marola, pelo apoio emocional e suporte que nos foi oferecido durante todo o processo de desenvolvimento e idealização deste trabalho.

Também, nossos animais domésticos — Fonseca, Nenem, Kira, Choco, Fofinha e Max —, por todo o apoio emocional que nos deram.

À Escola Técnica Estadual de São Paulo, nossos maiores e mais importantes agradecimentos, pelo apoio, ensino e recursos essenciais para a idealização e realização de nosso projeto, nosso êxito não seria possível sem o esforço e dedicação dos profissionais do ensino, Carlos Alberto e Marlon Marques.

Bem como, o conhecimento e mestria dos nossos coordenadores Jeferson Roberto Lima e Rogério Bezerra, que nos proporcionaram com uma visão clara para o sucesso da Sonoris.

Vocês fizeram nosso projeto se tornar realidade!

Muito obrigado a todos!

“Epigrafe”

Autor

RESUMO

ResumoResumoResumo

Palavras-Chave: acessibilidade; comunicação; deficiência auditiva; transcrição.

ABSTRACT

AbstractAbstractAbstract

Keywords: accessibility; communication; auditive; transcription.

LISTA DE ILUSTRAÇÕES

Figura 1 - Panorama das pessoas com deficiência no Brasil	18
Figura 2 - Raspberry PI 3B.....	20
Figura 3 - Microfone USB Omnidirecional.	21
Figura 4 - Display LCD Touch.....	23
Figura 5 - Exemplo de <i>wireframe</i> de baixa fidelidade.....	25
Figura 6 - Exemplo de <i>wireframe</i> de alta fidelidade.	26
Figura 7 - Exemplo de código Python.	27
Figura 8 - Resultado do código Python.	28
Figura 9 - Exemplo de funções Dart.....	29
Figura 10 - Exemplo de código Dart.....	30
Figura 11 - Resultado do código Dart.....	31
Figura 12 - Exemplo de código Flutter.....	32
Figura 13 - Resultado do código Flutter.	33
Figura 14 - Logo do Firebase.	34
Figura 15 - Exemplo de diagrama de casos de uso	35
Figura 16 - Exemplo de diagrama de sequência	39
Figura 17 - Exemplo de diagrama de máquina de estados	41
Figura 18 - Exemplo de diagrama de atividades	42
Figura 19 - Exemplo de modelagem 3D.....	43
Figura 20 - Exemplo de impressão 3D	44
Figura 21 - Gatos brancos e a surdez	46
Figura 22 - Conceituação da logo	46
Figura 23 - Idealização da logo	47
Figura 24 - Paleta de cores da Sonoris	48
Figura 25 - Tipografia da Sonoris	48
Figura 26 - Diagrama de casos de uso	50
Figura 27 - Wireframe da página de bem-vindo	54
Figura 28 - Wireframe da página de cadastro	55
Figura 29 - Wireframe da página de modo de funcionamento.....	56
Figura 30 - Wireframe da página de bluetooth	57
Figura 31 - Wireframe da página de pareamento.....	58
Figura 32 - Wireframe da página de dispositivo configurado.....	59

Figura 33 - Wireframe da página de login	60
Figura 34 - Wireframe da página inicial	61

LISTA DE ABREVIATURAS E SIGLAS

Ambulatório Medico de Especialidades (AME).

Aplicativo (APP).

Ahead of Time (AOT).

Bateria de Íon-Lítio (BIL).

Bluetooth Low Energy (BLE).

Corrente Alternada (CA).

Computer-Aided Design (CAD).

Corrente Contínua (CC).

Código de Endereçamento Posta (CEP).

Cadastro de Pessoas Físicas (CPF).

Frequency-Hopping Spread Spectrum (FHSS).

Instituto Brasileiro de Geografia e Estatística (IBGE).

Internet of Things (IoT).

Instituto Nacional de Educação dos Surdos (INES).

Just in Time (JIT).

Liquid Crystal Display (LCD).

Língua Brasileira de Sinais (LIBRAS).

Package Installer for Python (PIP).

Regras de Negócio (RN).

Requisitos Funcionais (RF).

Requisitos Não Funcionais (RNF).

Tecnologia Assistiva (TA).

Tridimensional (3D).

Unified Modeling Language (UML).

Uninterruptible Power Supply (UPS).

Universal Serial Bus (USB).

User Experience (UX).

User Interface (UI).

LISTA DE QUADROS

Quadro 1 - Exemplo de documentação de casos de uso	37
Quadro 2 - Documentação do caso de uso cadastrar conta	51

SUMÁRIO

1	INTRODUÇÃO	15
2	REFERENCIAL TEÓRICO	18
2.1	Deficiência Auditiva.....	18
2.1.1	Problemas Enfrentados	19
2.1.2	Importância da acessibilidade.....	20
2.2	Tecnologias Utilizadas	20
2.2.1	Internet of Things (IoT).....	20
2.2.2	Raspberry Pi	20
2.2.3	Microfone Omnidirecional	21
2.2.4	Display LCD Touch	22
2.2.5	Bateria de Íon-Lítio (BIL)	23
2.2.6	Uninterruptible Power Supply (UPS)	23
2.2.7	Bluetooth Low Energy	23
2.2.8	Wireframes	24
2.2.9	Design Thinking.....	26
2.2.10	UI & UX	26
2.2.11	Python	27
2.2.12	Vosk	28
2.2.13	Dart	28
2.2.14	Flutter	31
2.2.15	Banco de dados	33
2.2.16	Firebase.....	34
2.2.17	Unified Modeling Language (UML).....	34
2.2.18	Prototipação do Case.....	43
3	DESENVOLVIMENTO.....	45

3.1	Identidade visual	45
3.2	Levantamento de Requisitos	48
3.3	Documentação	50
3.3.1	Diagrama de Casos de Uso	50
3.3.2	Documentação dos Casos de Uso	50
3.3.3	Diagrama de Máquina de Estados	52
3.3.4	Diagrama de Sequência	52
3.3.5	Diagrama de Atividade	52
3.4	Implementação do IoT	54
3.5	Wireframes da Sonoris.....	Erro! Indicador não definido.
4	CONSIDERAÇÕES FINAIS	63
	REFERÊNCIAS.....	64

1 INTRODUÇÃO

Neste capítulo, será abordado o tema, principais objetivos, objetivos específicos, hipóteses e justificativas do projeto Sonoris que tem como proposta a criação de um dispositivo que realiza a transcrição de áudio em conjunto com um aplicativo que armazena as conversas, personaliza as legendas e configura o dispositivo de forma a auxiliar a pessoa com deficiência auditiva.

O objetivo principal deste projeto é permitir que as pessoas com deficiência auditiva tenham mais autonomia, especialmente em contextos educativos e profissionais, visto que o dispositivo foi projetado para captar a fala humana em uma grande área ao seu redor e salvar transcrições no aplicativo automaticamente, para revisar reuniões e aulas posteriormente.

Já para os objetivos específicos, foi feita uma pesquisa detalhada sobre as dificuldades enfrentadas por essas pessoas e, logo após, a elaboração do protótipo do dispositivo com o objetivo de ser agradável ao usuário e que interaja de forma eficiente e rápida com o aplicativo, o qual será projetado cuidadosamente utilizando a ferramenta Figma, a fim de atender aos critérios de simplicidade e conforto que o projeto busca como objetivo.

Esse aplicativo possibilita a configuração das legendas da transcrição da fala humana, que serão apresentadas no *display* do dispositivo, assim como permite personalizar as respostas rápidas para caso precise, o usuário possa ter uma conversa agradável com a pessoa ouvinte.

Tudo isso será documentado utilizando o diagrama de caso de uso, diagrama de sequência, diagrama de atividade e diagrama de máquina de estado, todos pertencentes à linguagem unificada de modelagem (UML), que, conforme Booch, Rumbaugh e Jacobson (2012), é fundamental para facilitar a compreensão de toda a aplicação e suas funcionalidades como cadastrar a conta do usuário.

As pessoas com deficiência auditiva ainda enfrentam preconceito e obstáculos comunicativos que prejudicam a sua inclusão e aceitação na sociedade, consequentemente levando ao sentimento de exclusão, consoante a Magno (2021).

Analisando esses problemas, foi concluído que eles são causados por um desinteresse e uma falta de engajamento das pessoas ouvintes em quebrar essas barreiras, segundo AME (2021). Também pertinente ao problema, é a falta de meios de educação inclusiva para essas pessoas, como intérpretes em palestras e aulas, o que pode dificultar uma educação de qualidade e consequentemente a entrada ao mercado de trabalho.

Mediante a esses problemas, a Sonoris nasceu dessa urgente necessidade de reduzir as barreiras de comunicação enfrentadas diariamente pelas pessoas com deficiência auditiva. Ela oferece não apenas uma ferramenta de acessibilidade que garanta a autonomia do usuário, mas também traz uma solução inovadora e acessível, que auxilia os surdos a participem ativamente das interações cotidianas e permite o armazenamento de aulas e palestras captadas pelo dispositivo para posteriormente serem revisadas, promovendo a inclusão social e uma verdadeira qualidade de vida para a pessoa.

Metodologia de abordagem qualitativa (Lakatos e Marconi, ano)

A Sonoris tem como público alvo pessoas com deficiência auditiva, no entanto, utilizando da captação de áudio, para transcrever e salvar conversas, também é aberto a possibilidade da utilização por qualquer pessoa que precise da ferramenta para estudos, como ferramenta de anotação e produtividade.

Para o desenvolvimento do referencial teórico, foram citados autores e textos certificados, que possuem a capacidade de descrever as necessidades do tema e auxiliar na continuidade do projeto. Utilizou-se a notícia *Pessoas com deficiência têm menor acesso à educação, ao trabalho e à renda*, elaborado pelo IBGE (2012), para a contextualização e conscientização necessária a fim de uma abordagem viável perante o assunto. Seguindo o mesmo pensamento, também foi de suma importância o livro ***Introdução à IOT: Desvendando a Internet das Coisas***, feito por Sandro Santos, visando atender às melhores práticas para a concepção do dispositivo portátil. Ademais, foi utilizado o ***UML 2: Uma abordagem prática***, de Gilleanes Guedes (2018), para a preparação e desenvolvimento da documentação e prototipagem do projeto, vendo que, a obra explica de forma clara e objetiva sobre os princípios da Linguagem de Modelagem Unificada e como utiliza-la.

Toda essa análise e documentação preparada foi de extrema importância para o desenvolvimento do projeto, que foi feito utilizando o *framework* Flutter em conjunto com a linguagem de programação Dart para a criação de todas as telas do aplicativo e o armazenamento seguro das informações no banco de dados Firebase. Para a codificação da lógica de captação da fala humana e transcrição de legendas, aplicou-se o Python em colaboração com a biblioteca Vosk, garantindo a precisão e eficiência no processamento do áudio.

Portanto, a Sonoris foi elaborada com o propósito de facilitar a comunicação e contribuir para a inclusão das pessoas com deficiência auditiva, principalmente nos âmbitos profissionais e acadêmicos, onde a comunicação é crucial. O aplicativo e o dispositivo proporcionam uma experiência intuitiva e agradável, possibilitando que o usuário (INSIRA CONCLUSÃO).

2 REFERENCIAL TEÓRICO

Nessa seção serão abordadas as principais tecnologias e embasamentos teóricos utilizados para a documentação e desenvolvimento do nosso projeto Sonoris.

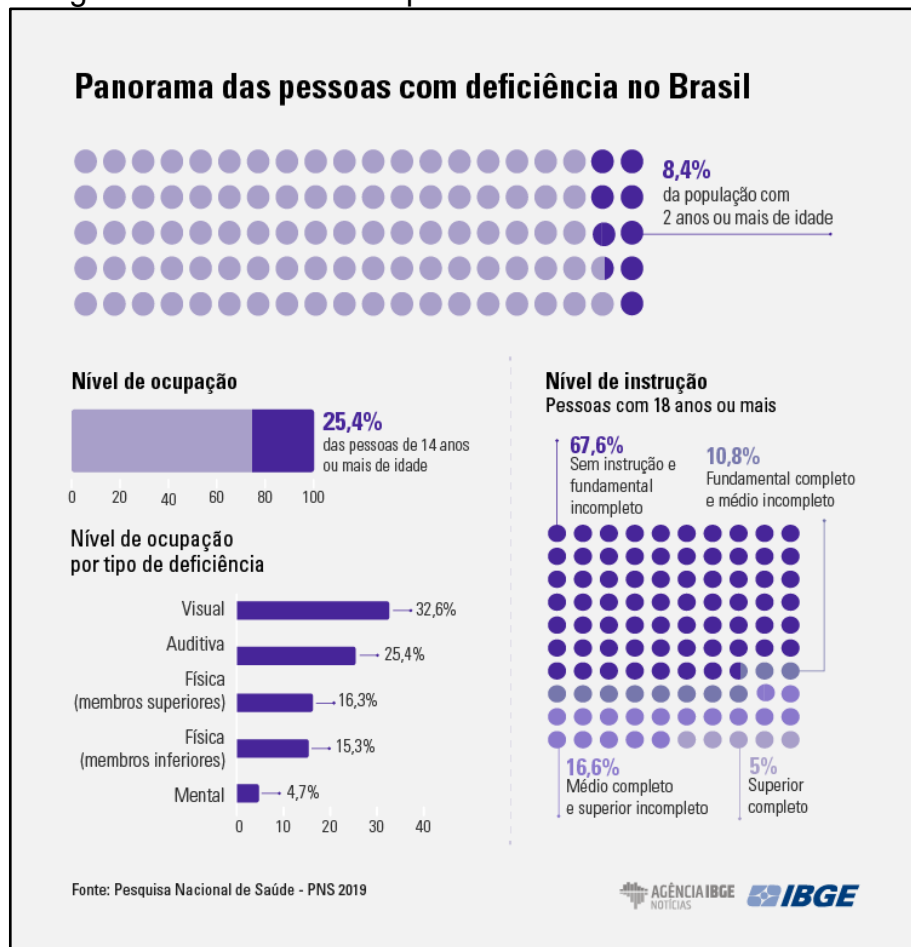
2.1 Deficiência Auditiva

Segundo levantamento realizado pelo IBGE (2010), cerca de 4,6% da população brasileira apresenta algum tipo de deficiência auditiva, sendo que 1% têm surdez profunda e, de acordo com o artigo 4º da Lei Nº 13.146, de 6 de julho de 2015, Estatuto da Pessoa com Deficiência:

Art. 4. Toda pessoa com deficiência tem direito à igualdade de oportunidades com as demais pessoas e não sofrerá nenhuma espécie de discriminação. (BRASIL, 2015, p. 23)

Toso *et al.* (2018) afirmam que as tecnologias assistivas de comunicação têm um papel importante na vida de pessoas com deficiência auditiva, facilitando a comunicação e contribuindo para sua inclusão social.

Figura 1 - Panorama das pessoas com deficiência no Brasil



Fonte: Pesquisa Nacional de Saúde (PNS), 2019.

2.1.1 Problemas Enfrentados

As maiores dificuldades enfrentadas por pessoas com deficiência auditiva se dão pela falta de oralização de pessoas com surdez profunda e pela falta de conhecimento da Libras pela população ouvinte e surda, consoante à Silva, Hora e Carvalho (2019).

Há séculos pessoas com deficiência auditiva eram vistas como incapazes de exercer quaisquer funções na sociedade da época justamente por possuírem certas limitações ou dificuldades, parafraseando UFMG (2022).

Conforme Mundo Educação (2022), no século XIX, Ernest Huet, educador e deficiente auditivo, fundador do Instituto Nacional de Educação dos Surdos (INES), veio ao Brasil a pedido de D. Pedro II para auxiliar na educação de pessoas com deficiência auditiva.

No entanto, houve bastante contrariedade a respeito do desenvolvimento da Libras no Brasil, devido à crença da época, consequentemente sendo decretado sua proibição em 1911, consoante à Brasil Escola (2024).

Todavia, os surdos lutaram por seus direitos, posteriormente resultando na criação de um projeto de lei em 1993, aprovado apenas em no dia 24 de abril de 2002 como Lei 10.436/2002, UFMG (2022). Essa lei reconhece a Libras como meio legal de comunicação no país e assegura a inclusão das pessoas com deficiência auditiva em instituições públicas. – LEI

Além dessa lei, também há o Decreto nº 5.626/2005, que regulamenta o ensino da Libras e sua difusão nas instituições de ensino. Por exemplo, como determina o Artigo 22º, escolas comuns e bilíngues abertas a alunos com necessidades especiais. –

JUSBRASIL

De acordo com Vereta e Streiechen (2024), por mais que a legislação garanta cotas para profissionais com deficiência auditiva, muitas empresas apenas cumprem a lei, sem oferecer condições para o desenvolvimento desses trabalhadores.

Conforme Magno (2021), as pessoas com deficiência auditiva ainda enfrentam preconceito e barreiras de comunicação que dificultam tanto sua inclusão no mercado quanto sua aceitação na sociedade, gerando um sentimento de exclusão.

Diante desse cenário, se mostra cada vez necessário buscar softwares mais claros e de fácil acesso para seus usuários, como diz Freitas, Maranhão e Félix (2017).

2.1.2 Importância da acessibilidade

Consoante à Rodrigues e Alves (2013), a tecnologia assistiva (TA) é uma área em ascensão, muito importante para a participação e inclusão das pessoas com deficiência em diversos contextos sociais.

De acordo com Santos *et al* (2017), a TA pode auxiliar em situações que exigem o desempenho humano, como auto higiene, esportes, tarefas profissionais ou até de lazer, neutralizando a barreira da deficiência promovendo inclusão.

2.2 Tecnologias Utilizadas

Durante o processo de desenvolvimento desse projeto foram utilizadas as seguintes tecnologias:

2.2.1 Internet of Things (IoT)

Conforme Magrani (2018), *Internet of Things* é considerado, de forma geral, um conceito onde dispositivos físicos distintos interagem entre si, a partir de sensores e objetos a fim de criar soluções práticas para o cotidiano.

Ele não é apenas a representação de ligar dispositivos pelo *smartphone*, e sim tornar esses dispositivos inteligentes, capazes de captar e analisar dados do ambiente ou das redes que ele está conectado, segundo Oliveira (2017).

Consoante a Santos (2019), IoT é especialmente útil para auxiliar a automatização de tarefas do dia a dia, sem a necessidade de intervenção humana.

2.2.2 Raspberry Pi

De acordo com Eberman *et al.* (2017), no ano de 2006, Eben Upton, Rob Mullins, Jack Lang e Alan Mycroft, tiveram a ideia de criar um microcomputador, denominado Raspberry Pi, acessível para ensinar jovens sobre lógica de programação.

Porém o sucesso do módulo Raspberry Pi e seu poder de processamento elevado mostrou que ele poderia ser usado em outros cenários, especialmente em sistemas embarcados, conforme Oliveira. (2017).

Figura 2 - Raspberry PI 3B.



Fonte: Autoria Própria, 2025.

O Raspberry Pi 3 Model B é o modelo de placa Raspberry mais vendido do mundo, tendo importantes características como Wi-Fi, *Bluetooth Low Energy* e múltiplas entradas USB para periféricos, consoante à RoboCore (2025)

2.2.3 Microfone Omnidirecional

Mercado Livre (2025) afirma que o Microfone De Mesa Omnidirecional Usb capta áudios de alta qualidade em até 3 metros, é portátil e fácil de usar, sendo ideal para reuniões, salas de aulas e afins.

Figura 3 - Microfone USB Omnidirecional.



Fonte: Autoria Própria, 2025.

2.2.4 Display LCD Touch

“Uma tela LCD é formada por uma matriz de ‘pontos’ quadrados, chamados de pixels, sendo cada pixel formado por três barras: uma vermelha, uma verde e uma azul.” (SOARES; BARBOSA, 2018, p. 4)

Figura 4 - Display LCD Touch.



Fonte: Mercado Livre, 2025.

2.2.5 Bateria de Íon-Lítio (BIL)

Azeredo (2024) aponta que as Baterias de Íon-Lítio têm grande capacidade energética entre as baterias recarregáveis, além de suportarem centenas de ciclos de carga e serem excepcionalmente leves.

De acordo com Ronaldo (2022), essa grande capacidade energética faz as baterias de íon-lítio serem bastante utilizadas em dispositivos IoT como o Raspberry Pi.

2.2.6 Uninterruptible Power Supply (UPS)

As UPS são um tipo de equipamento utilizado em quedas de energia elétrica, servindo como uma reserva energética para o aparelho, podendo ser utilizadas em uso residencial ou comercial, como destaca Storek (2007).

Segundo Andrade (2023), as UPS basicamente funcionam controlando a carga de uma fonte, e à convertendo de corrente alternada (CA) para corrente contínua (CC), que é apta para o uso em dispositivos eletrônicos.

2.2.7 Bluetooth Low Energy

Conforme Atoji (2010), o *Bluetooth* é uma tecnologia amplamente utilizada nos dispositivos modernos como celulares, laptops e demais periféricos, muitas vezes como um substituto dos cabos físicos para conexão de aparelhos.

Parafraseando Silva (2009), o *Bluetooth* funciona utilizando uma tecnologia de ondas de rádio com a técnica de transmissão FHSS, que ajudam a remover interferências de outros aparelhos e criam uma rede privada de conexão entre os dispositivos.

O *Bluetooth Low Energy* se difere do *Bluetooth* clássico pois foi projetado para a Internet das Coisas, sendo menos eficiente em aspectos como alcance e taxa de transmissão, mas consumindo menos energia, consoante a Reck (2016).

2.2.8 Wireframes

Os *wireframes* são protótipos responsáveis pela estruturação e organização do conteúdo do *software* na tela do usuário de maneira simplificada, como destaca Teixeira (2014).

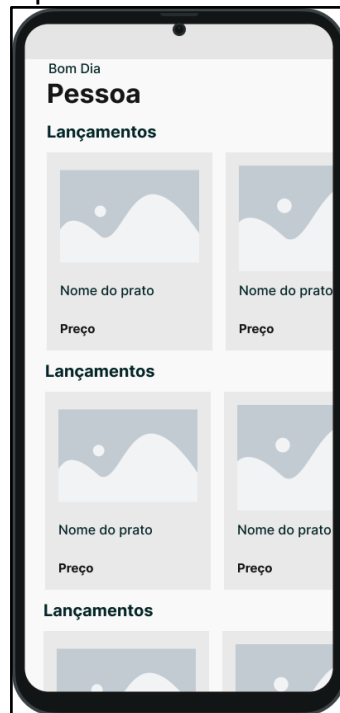
De acordo com Toledo (2014), a prototipação é fundamental no design de interfaces, pois permite testar soluções antes da versão final, garantindo que o produto atenda às necessidades dos usuários.

Conforme Alura (2022), o Figma é uma plataforma gratuita utilizada para construir interfaces de sistemas, possibilitando a prototipação deles de forma rápida e fácil.

O *wireframe* de baixa fidelidade é um protótipo superficial do site, já o *wireframe* de alta fidelidade, que apresenta a página de forma detalhada e interativa, contendo cores mais assertivas e uma navegação dinâmica, Lucidchart (2023).

Para um melhor entendimento sobre o assunto, abaixo podemos ver um exemplo do que é um *wireframe* de baixa fidelidade:

Figura 5 - Exemplo de *wireframe* de baixa fidelidade.

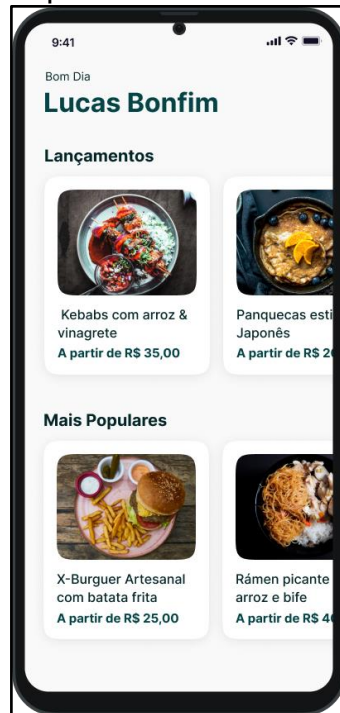


Fonte: Autoria Própria, 2025.

Este exemplo mostra como os *wireframes* de baixa fidelidade usam poucas cores, formas simples e imagens e textos genéricos, apenas com o intuito de definir a disposição dos elementos na interface.

Abaixo, o mesmo *wireframe* de baixa fidelidade após ser melhor elaborado, se tornando um *wireframe* de alta fidelidade:

Figura 6 - Exemplo de *wireframe* de alta fidelidade.



Fonte: Autoria Própria, 2025.

Já este exemplo, mostra como os *wireframes* de alta fidelidade expandem na ideia original, adicionando mais detalhes como cores, tipografia, formato de elementos e a criação de *mockups* de textos e imagens, que refletem como a interface deve ser desenvolvida de forma mais explícita.

2.2.9 Design Thinking

Segundo Melo e Abelheira (2015), o *design thinking* é uma abordagem metodológica que utiliza ferramentas do *design* para resolver problemas complexos, equilibrando criatividade e análise para manter a inovação e reduzir riscos.

O *design thinking* é uma metodologia de inovação que prioriza a perspectiva humana, utilizando empatia e prototipagem para desenvolver soluções com base nas necessidades dos usuários, parafraseando Pagani (2018).

2.2.10 UI & UX

UI é o desenvolvimento do visual da interface e de seus elementos dos quais o usuário interage, e tem como objetivo tornar a interface fácil e agradável de usar de maneira eficiente, Alura (2023).

Já a experiência do usuário (UX) é composta pela facilidade de uso, a agilidade para realizar tarefas, a satisfação emocional e a interação intuitiva, que são avaliados com a criação de protótipos e testes com usuários, segundo Chen (2018).

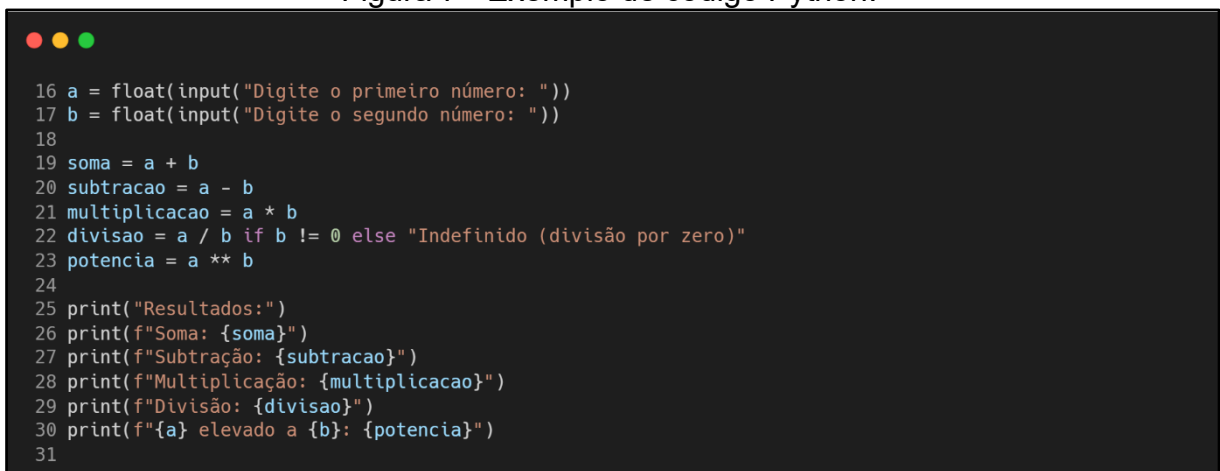
2.2.11 Python

Consoante Borges (2014), a linguagem Python foi desenvolvida por Guido van Rossum, em 1990, a partir da linguagem ABC e tinha inicialmente o foco de atender usuários profissionais em áreas de física e engenharia.

Como afirma Ramalho (2023), o Python utiliza várias ideias inspiradas na linguagem ABC, como a indentação obrigatória para estruturar blocos de código e a tipagem que não exige a declaração direta de variáveis.

Donat (2018) aponta que, Python foi a linguagem escolhida pelos criadores do Raspberry Pi seguindo o raciocínio de criar um ambiente de fácil aprendizagem, já que Python foi considerado uma linguagem eficaz e simples.

Figura 7 - Exemplo de código Python.



```

16 a = float(input("Digite o primeiro número: "))
17 b = float(input("Digite o segundo número: "))
18
19 soma = a + b
20 subtracao = a - b
21 multiplicacao = a * b
22 divisao = a / b if b != 0 else "Indefinido (divisão por zero)"
23 potencia = a ** b
24
25 print("Resultados:")
26 print(f"Soma: {soma}")
27 print(f"Subtração: {subtracao}")
28 print(f"Multiplicação: {multiplicacao}")
29 print(f"Divisão: {divisao}")
30 print(f"{a} elevado a {b}: {potencia}")
31

```

Fonte: Autoria Própria, 2025.

Linhas 16 e 17: É feita uma impressão na tela do terminal pedindo que o usuário digite dois números. As entradas são lidas com `input()` e convertidas para o tipo *float* usando a função `float()`, o que permite realizar cálculos com números decimais.

Linha 19: a soma dos dois números é armazenada na variável `soma`.

Linha 20: na variável `subtracao` é feita a subtração do segundo número com o primeiro.

Linha 21: na variável `multiplicacao` é feita uma multiplicação entre os dois números.

Linha 22: é feita a na variável divisão a divisão do primeiro numero pelo segundo. É usada em um laço condicional (*if b != 0 else ...*) para evitar erro de divisão por zero. Caso o segundo número seja zero, imprime a mensagem “Indefinido (divisão por zero)”.

Linha 23: na variável potencia, o primeiro número é elevado ao segundo com o operador ******.

Linhas 25 a 30: Os resultados das operações são imprimidos na tela usando a função `print()`.

Figura 8 - Resultado do código Python.

```
Digite o primeiro número: 5
Digite o segundo número: 3
Resultados:
Soma: 8.0
Subtração: 2.0
Multiplicação: 15.0
Divisão: 1.6666666666666667
5.0 elevado a 3.0: 125.0
```

Fonte: Autoria Própria, 2025.

O PIP é apresentado na documentação oficial de empacotamento Python (2024), como a ferramenta padrão para instalar pacotes em Python, permitindo que desenvolvedores encontrem e adicionem facilmente bibliotecas pelo terminal.

2.2.12 Vosk

A biblioteca *Vosk Speech Recognition Toolkit* processa localmente o áudio recebido, convertendo as falas em texto, e permitindo o uso de comandos de voz sem precisar de outras ferramentas conectadas, como apontam Silva e Silva (2025).

2.2.13 Dart

Bitencourt (2022) destaca que Dart é uma linguagem orientada a objetos criada pela Google, que integra princípios da programação funcional e oferece apoio para programação reativa, o que aumenta sua flexibilidade para construir aplicações atuais.

Conforme Marinho (2020), Dart utiliza AOT (Ahead of Time) e JIT (Just in Time), ou seja, o código é compilado antes da execução, trazendo maior estabilidade, rapidez na inicialização do projeto e nas atualizações dos componentes.

Abaixo, um exemplo de código de uma calculadora básica no terminal:

Figura 9 - Exemplo de funções Dart.

```
16 import 'dart:io';
17
18 void main() {
19   double numUm = 0;
20   double numDois = 0;
21   String operacao = "";
22
23   void soma(){
24     print(numUm + numDois);
25   }
26
27   void subtracao(){
28     print(numUm - numDois);
29   }
30
31   void multiplicacao(){
32     print(numUm * numDois);
33   }
34
35   void divisao(){
36     print(numUm / numDois);
37   }
38
39   void calculo(){
40     switch(operacao){
41       case "+":
42         soma();
43         break;
44
45       case "-":
46         subtracao();
47         break;
48
49       case "*":
50         multiplicacao();
51         break;
52
53       case "/":
54         divisao();
55         break;
56     }
57   }
```

Fonte: Autoria Própria, 2025.

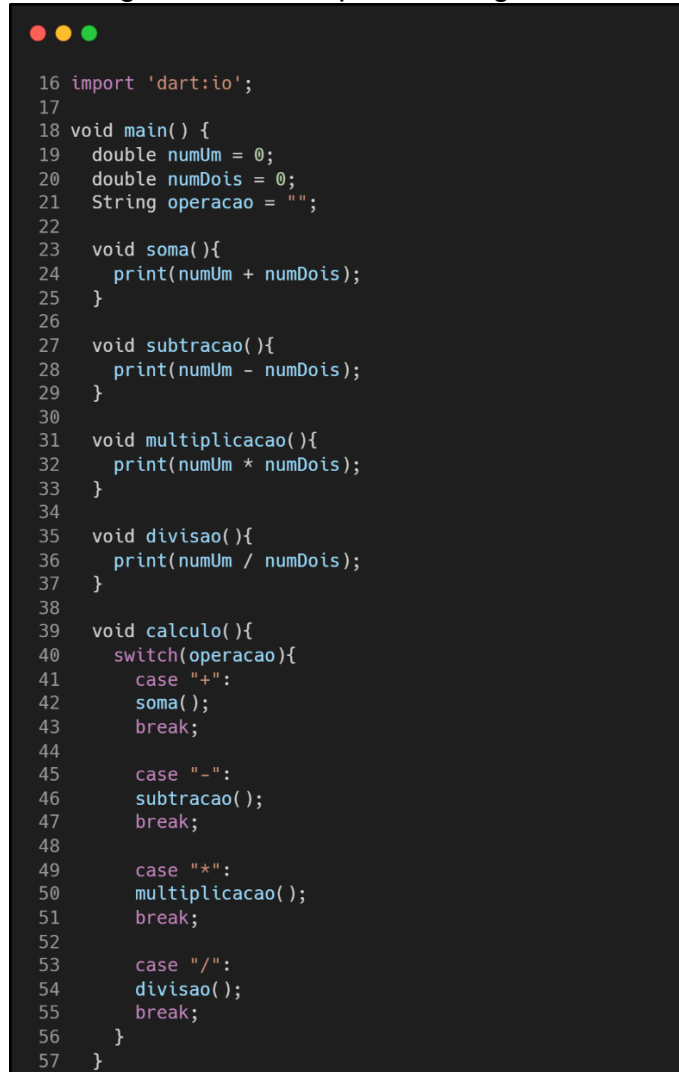
Linhas 16: São feitas as importações necessárias para a execução. `dart:io` é uma biblioteca que faz a leitura de entradas via terminal.

Linhas 19 a 20: As variáveis `numUm`, `numDois` e `operacao` são criadas. Elas vão armazenar os dois números e o tipo de operação que o usuário irá digitar.

Linhas 23 a 37: São criadas quatro funções — `soma`, `subtracao`, `multiplicacao` e `divisao`, uma para cada operação matemática. Cada uma faz o cálculo correspondente, usando os números armazenados nas variáveis, e exibe o resultado diretamente na tela do terminal.

Linhas 39 a 57: A função `calculo` é criada. Nela, é usada a estrutura `switch` para verificar qual operação foi escolhida e executar a função correspondente ao símbolo digitado.

Figura 10 - Exemplo de código Dart.



```

16 import 'dart:io';
17
18 void main() {
19   double numUm = 0;
20   double numDois = 0;
21   String operacao = "";
22
23   void soma(){
24     print(numUm + numDois);
25   }
26
27   void subtracao(){
28     print(numUm - numDois);
29   }
30
31   void multiplicacao(){
32     print(numUm * numDois);
33   }
34
35   void divisao(){
36     print(numUm / numDois);
37   }
38
39   void calculo(){
40     switch(operacao){
41       case "+":
42         soma();
43         break;
44
45       case "-":
46         subtracao();
47         break;
48
49       case "*":
50         multiplicacao();
51         break;
52
53       case "/":
54         divisao();
55         break;
56     }
57   }

```

Fonte: Autoria Própria, 2025.

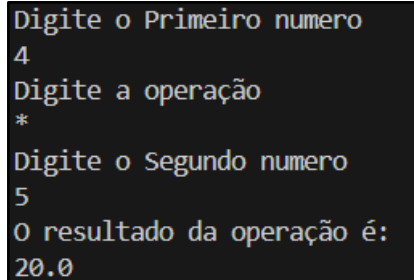
Linhas 58 a 64: Essas linhas fazem uma impressão na tela pedindo pelo primeiro número, e depois, usam o `stdin.readLineSync()` para ler o valor digitado. O valor é convertido para o tipo `double`, se o valor digitado não for vazio.

Linhas 66 a 70: É feita uma impressão na tela do terminal solicitando ao usuário a operação desejada. Ela é lida e armazenada na variável `operacao`.

Linhas 72 a 78: Nessas linhas, uma impressão na tela é realizada que pede pelo segundo número desejado, que também é lido e convertido da mesma forma que o primeiro.

Linhas 80 a 81: Por fim, o programa imprime a mensagem indicando o resultado da operação e chama a função `calculo()`, que realiza o cálculo com base nos dados armazenados.

Figura 11 - Resultado do código Dart.

A screenshot of a terminal window with a dark background. It shows the execution of a Dart program. The prompts and user inputs are as follows:
- Prompt: "Digite o Primeiro numero"
- Input: "4"
- Prompt: "Digite a operação"
- Input: "*" (multiplication)
- Prompt: "Digite o Segundo numero"
- Input: "5"
- Output: "O resultado da operação é:"
- Output: "20.0"
The text is displayed in a monospaced font with some color coding (green for prompts, white for inputs and outputs).

```
Digite o Primeiro numero
4
Digite a operação
*
Digite o Segundo numero
5
O resultado da operação é:
20.0
```

Fonte: Autoria Própria, 2025.

2.2.14 Flutter

Segundo a Alura (2022), o Flutter é um *framework* criado pela Google, que utiliza a linguagem Dart e tem como principal objetivo o desenvolvimento de aplicações móveis para diferentes sistemas operacionais.

De acordo com Marinho (2020), o *framework* Flutter foi desenvolvido com o propósito de otimizar o carregamento das aplicações, tornando-as mais rápidas em um ambiente multiplataforma a partir de um único código.

Grande parte dos elementos do Flutter são *widgets*, que são classes em Dart. Esses *widgets* normalmente são organizados em uma estrutura hierárquica chamada de "árvore de *widgets*", que definem a composição da UI, conforme Zammetti (2020).

O exemplo abaixo é uma aplicação básica em Flutter que exibe uma interface com uma tarefa a ser realizada.

Figura 12 - Exemplo de código Flutter.

```

16 home: Scaffold(
17   appBar: AppBar(
18     title: Text('Tarefas'),
19   ),
20   body: Column(
21     mainAxisAlignment: MainAxisAlignment.spaceBetween,
22     children: [
23       Stack(
24         children: [
25           Container(color: Colors.blue, height: 140),
26           Container(
27             color: Colors.white,
28             height: 100,
29             child: Row(
30               mainAxisAlignment: MainAxisAlignment.spaceBetween,
31               children: [
32                 Container(color: Colors.black12, width: 72, height: 100),
33                 Text('Aprender Flutter'),
34                 ElevatedButton(
35                   onPressed: () {}, child: Icon(Icons.check_box)),
36               ],
37             ),
38           ),
39         ],
40       ),
41     ],
42   ),
43   floatingActionButton: FloatingActionButton(
44     onPressed: () {},
45     child: Icon(Icons.add),
46   ),
47 );
48

```

Fonte: Autoria Própria, 2025.

A seguir, encontra-se uma explicação breve sobre o funcionamento do código desenvolvido acima:

Linha 16: A propriedade `home` define a tela principal do aplicativo, que é construída utilizando o *widget Scaffold*. Esse *widget* funciona como uma estrutura simples que organiza os elementos visuais principais da tela.

Linhas 17 a 19: Na parte superior da tela, o *widget AppBar* cria uma barra de aplicativo, exibindo o título “Tarefas”.

Linhas 20 a 42: O conteúdo principal fica dentro do *widget body*, que contém uma *Column*. A *Column* é um *widget* que organiza seus filhos alinhando-os em coluna com espaçamento entre eles.

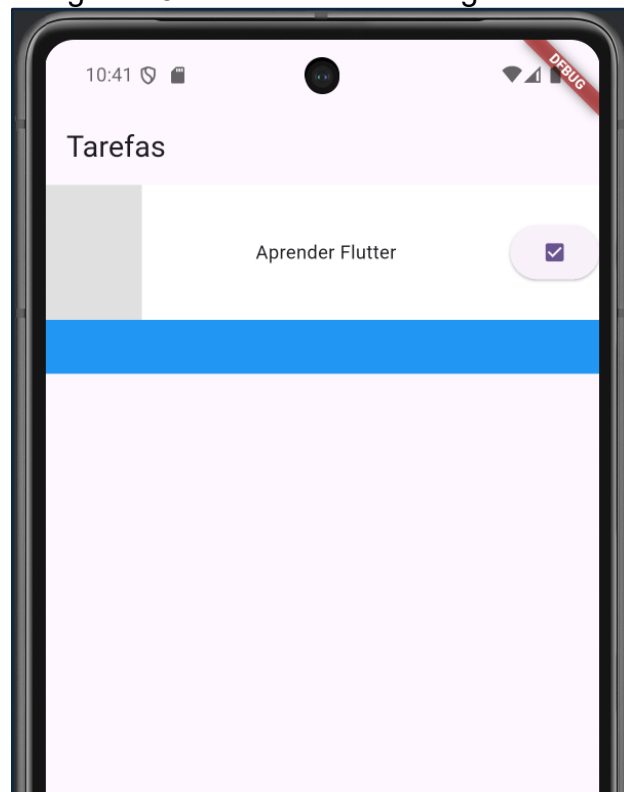
Linhas 23 a 40: Dentro dessa *Column*, há um *widget Stack*, que permite que os filhos se empilhem, posicionando um em cima do outro. No caso, são dois *widgets*

Container: o primeiro, com cor azul e maior altura, serve como fundo; o segundo, branco e menor, fica sobreposto para exibir o conteúdo principal.

Linhas 26 a 37: O Container branco contém um *widget Row*, que organiza seus elementos em linha horizontal. Essa linha inclui três *widgets*: um Container quadrado cinza claro, um *widget Text* com o título “Aprender Flutter” e um botão elevado (ElevatedButton) com o ícone de caixa de seleção (*Icon*).

Linhas 43 a 47: Para finalizar, o *widget FloatingActionButton* cria um botão flutuante no canto inferior direito da tela, com o ícone (*Icon*) de “adicionar”.

Figura 13 - Resultado do código Flutter.



Fonte: Autoria Própria, 2025.

2.2.15 Banco de dados

Como afirma Date (2004), um banco de dados é um sistema de armazenamento e manutenção de registros de dados digitais, que permite que os usuários busquem e atualizem esses dados que podem conter qualquer tipo de informação relevante.

Elmasri e Navathe (2011) definem chave primária como a chave candidata escolhida para identificar unicamente cada registro de tabela, cujos valores não podem ser nulos, fazendo com que os dados sejam confiáveis.

2.2.16 Firebase

Consoante Bernardino e Barros (2025), o Firebase é uma plataforma que opera com a Google Cloud, facilitando e agilizando o desenvolvimento de apps com a disponibilização de serviços de sincronização em nuvem.

O Firebase disponibiliza vários *plug-ins* para Flutter que facilitam a conexão de aplicações aos seus serviços, melhorando a experiência e qualidade no aplicativo em menos tempo e mais facilmente, Firebase (2025).

Figura 14 - Logo do Firebase.



Fonte: Firebase, 2025.

Como aponta Rodrigues (2021), o Cloud Firestore é um banco de dados NoSQL do Firebase que armazena e sincroniza dados em tempo real, mesmo quando offline, e disponibiliza alto desempenho e escalabilidade nas consultas.

2.2.17 Unified Modeling Language (UML)

Segundo Guedes (2018), a linguagem unificada de modelagem é uma linguagem ilustrativa de modelagem versátil, muito utilizada para projetar a estrutura dos *softwares* baseados no modelo de orientação a objeto.

A modelagem é essencial para auxiliar a compreensão da aplicação e de suas funcionalidades, além de documentar toda sua trajetória e os processos que foram tomados para sua conclusão, como afirma Booch, Rumbaugh e Jacobson (2012).

De acordo com Fowler (2005), a UML foi criada a partir da unificação de outras linguagens de modelagem diferentes dos anos 90, facilitando o desenvolvimento de diagramas e documentações padronizadas.

Ao todo, a UML possui 14 diagramas oficiais, que, conforme Guedes (2018), são separados entre dois grupos: diagramas estruturais e diagramas comportamentais.

Booch, Rumbaugh e Jacobson (2012) afirmam que os diagramas estruturais auxiliam a visualização e documentação da estrutura que compõe o sistema, enquanto os diagramas comportamentais ajudam a compreender os aspectos que são alterados. Um exemplo de ambos os diagramas é uma casa: ela possui aspectos estáticos, como as portas, janelas, mesas, cadeiras, e aspectos dinâmicos, como a passagem de pessoas pelos cômodos e ar.

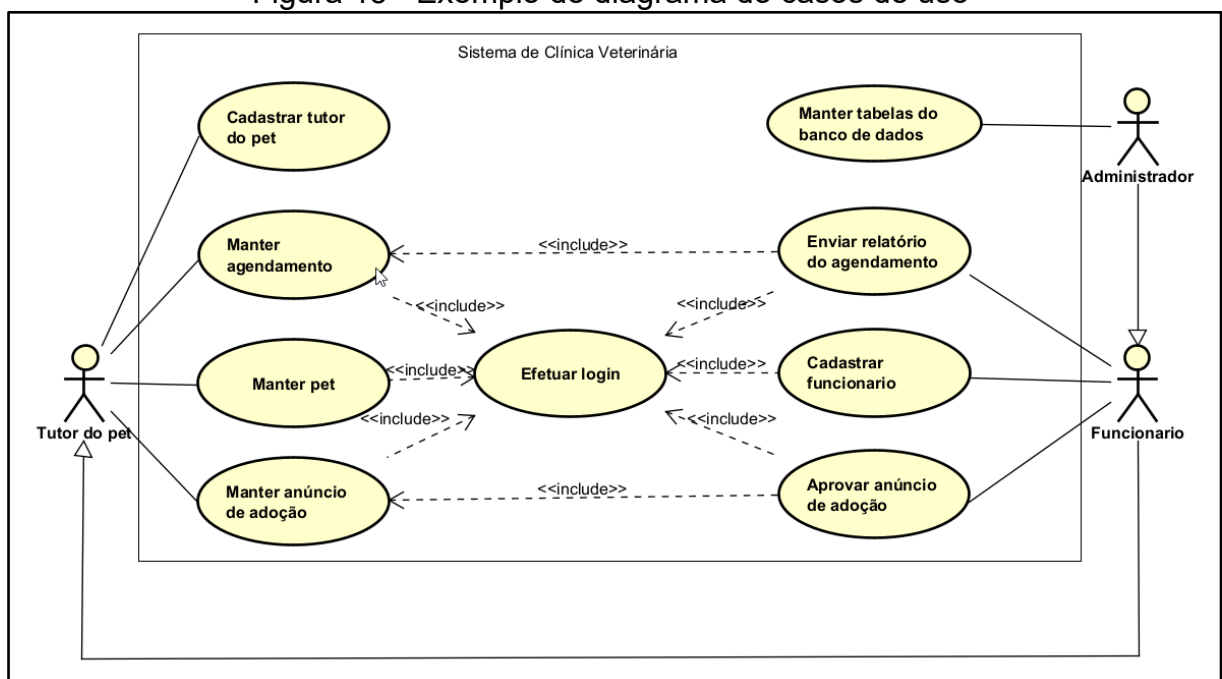
No desenvolvimento desse projeto será utilizado os seguintes diagramas: diagrama de caso de uso, diagrama de classe, diagrama de sequência, diagrama de máquina de estados e diagrama de atividade.

2.2.17.1 Diagrama de Casos de Uso

O diagrama de casos de uso tem como finalidade ilustrar as principais funcionalidades do sistema de forma simplificada para que seja fácil o entendimento das interações que o usuário terá com o *software*, como destaca Guedes (2018).

Fowler (2005) aponta que, no diagrama de casos de uso, temos atores que representam indivíduos que interagem com a aplicação. Cada uma dessas interações entre usuário e sistema é uma funcionalidade diferente que o *software* possui.

Figura 15 - Exemplo de diagrama de casos de uso



Fonte: Autoria Própria, 2024.

A figura acima apresenta o diagrama de casos de uso de um sistema de clínica veterinária, onde temos, ao todo, três atores fundamentais, sendo eles:

O “tutor do pet”, que representa o cliente da clínica veterinária e pode se cadastrar no sistema, gerenciar seus agendamentos de consulta, cadastrar seus animais domésticos e, por último, administrar seus anúncios de adoções.

O “funcionario”, responsável por enviar os relatórios do agendamento após ele ser concluído, cadastrar outros funcionários e verificar os anúncios de adoção para garantir que o anúncio seja verídico.

Por último, o “administrador”, que tem como única função gerenciar as tabelas do banco de dados da aplicação, garantindo a manutenção fácil e rápida da mesma.

Este “administrador” tem as mesmas funções que um funcionário, assim como um funcionário exerce as mesmas funcionalidades que o tutor do pet.

Além disso, a grande parte dessas essas interações realizadas pelos atores necessitam que seja efetuado login antes de sua execução. Esse tipo de relacionamento é chamado de `<<include>>` e, de acordo com Booch, Rumbaugh e Jacobson (2012), significa que um caso de uso está atrelado a outro.

Iremos utilizar esse exemplo de diagrama de caso de uso para fazer a elaboração dos demais exemplos referentes ao *Unified Modeling Language* (UML).

2.2.17.2 Documentação de Casos de Uso

Guedes (2018) destaca que, a documentação de casos de uso é usada para detalhar os casos de uso, informando os atores que interagem com o mesmo, o processo para chegar até ele, suas condições e restrições.

Como descrito anteriormente, será utilizado um dos casos de uso apresentado como exemplo no capítulo acima referente aos diagramas de casos de uso para construir uma documentação do mesmo. Mais especificamente será aplicado o caso de uso “Cadastrar tutor do pet”, que apresenta o processo que o usuário levará para cadastrar uma conta no sistema da clínica veterinária.

Quadro 1 - Exemplo de documentação de casos de uso

Nome do Caso de Uso		Cadastrar tutor do pet	
Ator Principal		Tutor do pet	
Atores Secundários		Funcionario; Administrador	
Resumo		Esse caso de uso mostra as etapas percorridas pelo usuário para cadastrar sua conta no sistema	
Pré-condições			
Pós-condições			
		Cenário Principal	
Ações do Ator		Ações do Sistema	
1. O cliente informa seu e-mail e senha		2. Consulta se já existe um cliente com o e-mail fornecido no sistema	
3. O cliente fornece o nome completo, data de nascimento, CPF, CEP e telefone		4. Cadastra a conta 5. Redireciona a página de perfil	
Restrições/Validações		1. Verifica se todos os campos estão preenchidos 2. Consulta se o cliente é maior de 16 anos de idade 3. CPF e telefone devem ser válidos	
Cenário de Exceção – Cliente menor de idade			
Ações do Ator		Ações do Sistema	
		1. Comunica que o cliente não tem a idade mínima para ser cadastrado no sistema 2. Recusa o pedido de cadastro	

Fonte: Aatoria Própria, 2025.

No caso acima, o ator principal é o tutor do pet e os atores secundários são os funcionários e administradores do sistema.

No cenário ideal, o fluxo descrito apresenta o processo de ações e etapas necessários para a criação de um conta caso não tenham complicações como o cliente ter idade menor de 16 anos e os campos não estejam preenchidos apropriadamente.

Já no cenário onde o cliente é menor do que a idade requerida (16 anos ou mais), o sistema informa que o cadastro foi recusado pois o indivíduo não condiz com os requisitos.

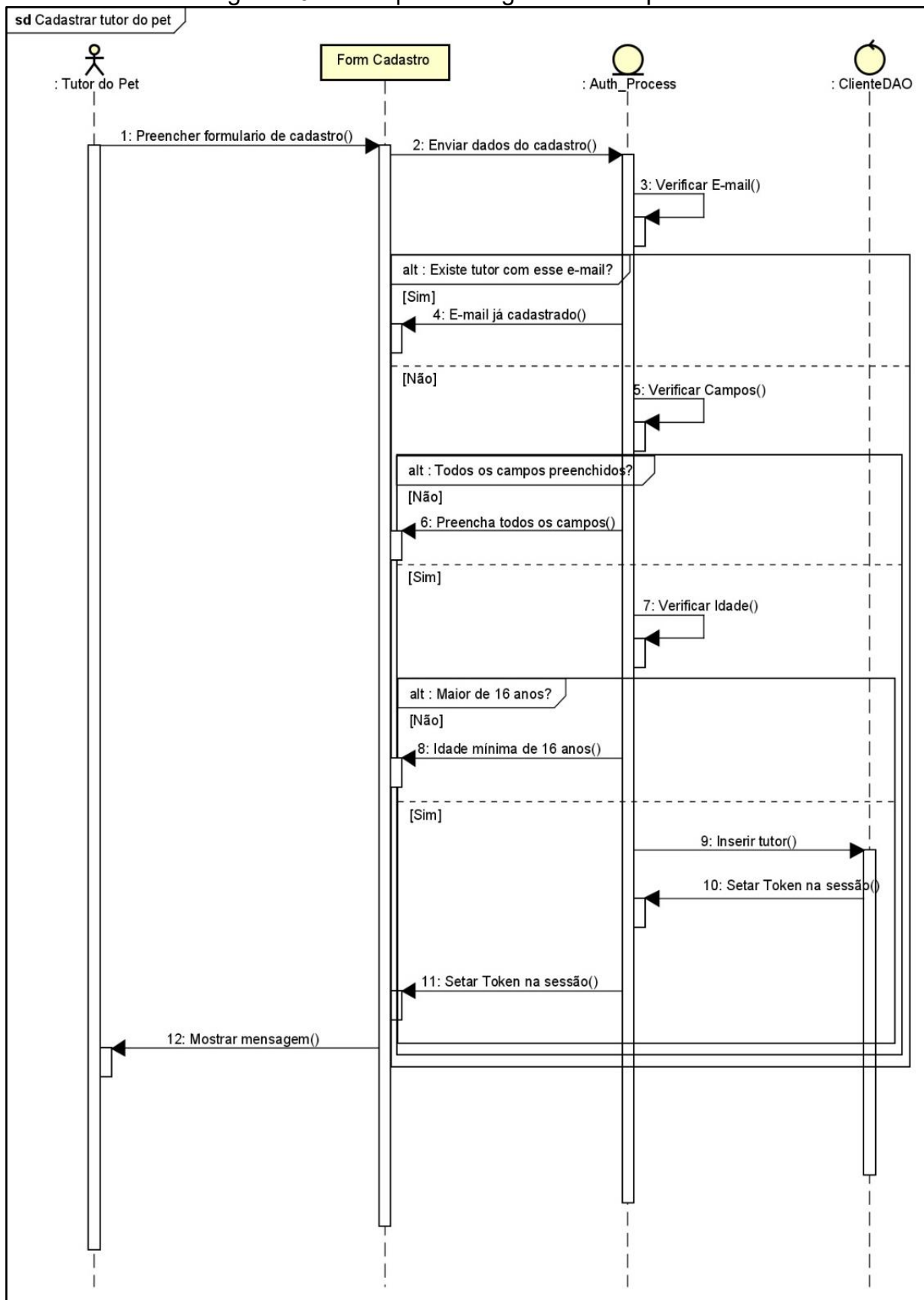
2.2.17.3 Diagrama de Sequência

Segundo Booch, Rumbaugh e Jacobson (2012), o diagrama de sequência é classificado como um “diagrama de interação”, que apresenta ações de determinados objetos, se distinguindo dos demais diagramas por ter foco na linha temporal.

Ele descreve as interações realizadas pelo usuário com o sistema – nomeadas de *lifelines*, em português “linhas de vidas” –, apresentadas verticalmente na página e representadas por uma linha tracejada, consoante a Fowler (2005).

Logo abaixo, um exemplo sobre o diagrama de sequência:

Figura 16 - Exemplo de diagrama de sequência



Fonte: Autoria Própria, 2025.

Pode-se enxergar que o diagrama começa com o ator Tutor do pet, que tem como objetivo cadastrar sua conta no site. Para isso, ele preencherá um formulário de cadastro, que é enviado de forma assíncrona para a *lifeline* “Form Cadastro”.

Essa linha de vida envia esses dados para o “Auth_Process”, que verifica se o e-mail fornecido existe no banco de dados do sistema.

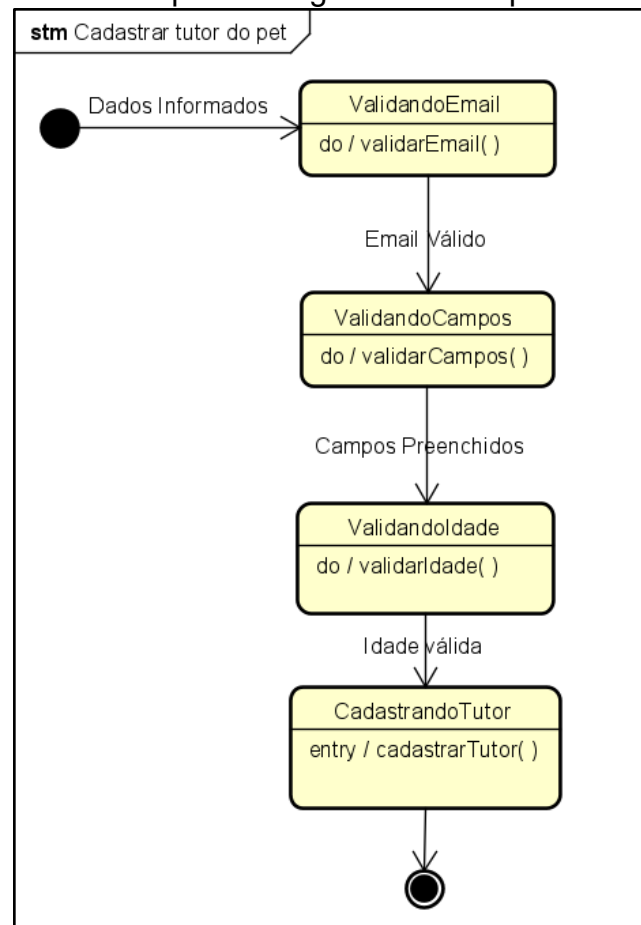
Caso já exista, retorna uma mensagem de erro para a *lifeline* “Form Cadastro”. No entanto, caso não exista, verificará se todos os campos do formulário estão preenchidos e se estiverem, analisará se a idade do tutor é superior a 16 anos de idade.

Se todas essas verificações estiverem corretas, irá cadastrar o tutor no “ClienteDAO”, gerando um *token* de sessão que será levado até a “Form Cadastro”, cujo mostrará uma mensagem ao tutor informando que seu cadastro foi concluído com sucesso.

2.2.17.4 Diagrama de Máquina de Estados

Baseado em Fowler (2005), o diagrama de máquina de estados é utilizado para descrever de forma aprofundada os comportamentos de um sistema conforme suas mudanças de estado.

Figura 17 - Exemplo de diagrama de máquina de estados



Fonte: Autoria Própria, 2025.

Esse diagrama inicia no círculo totalmente tingido de preto – que representa o ponto de partida – e segue passando dados para realizar o cadastro do tutor do pet.

Após isso, essas informações fornecidas são passadas para funções que fazem a validação das mesmas para confirmar que está tudo nos conformes antes do cadastro do cliente no sistema.

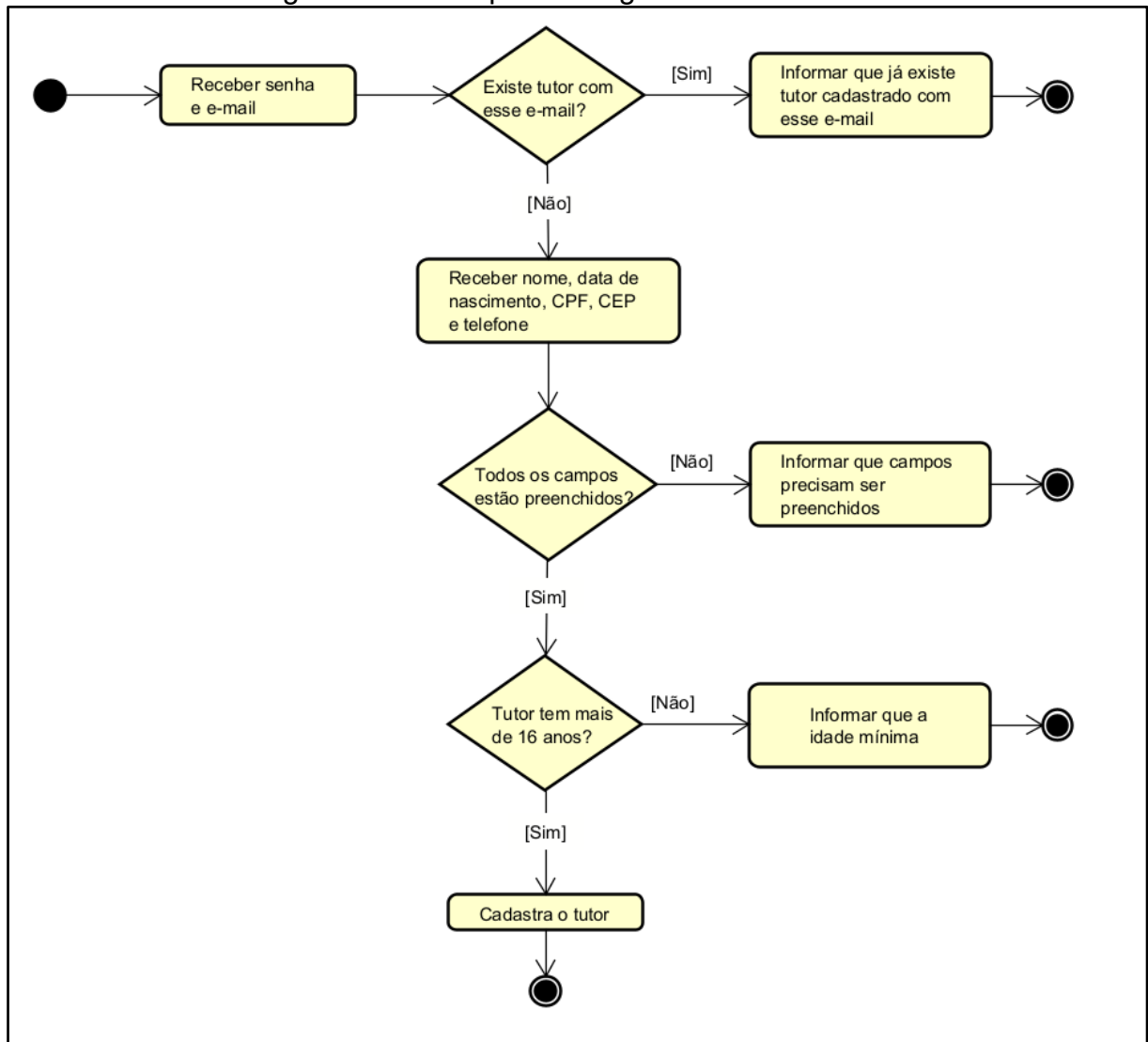
2.2.17.5 Diagrama de Atividades

De acordo com Fowler (2005), o diagrama de atividades é um fluxo que indica o percurso de interação do usuário durante a utilização de um sistema. Ele começa no laço inicial e segue uma ordem de ações até o nó final.

Esse diagrama é comumente utilizado para descrever a lógica proposta em um caso de uso, explicando de forma mais clara e coesa sobre suas etapas, conforme Guedes (2018).

O que o torna importante, pois, segundo Booch, Rumbaugh e Jacobson (2012), ele é considerado essencial para a modelagem das funcionalidades de um sistema já que facilita a compreensão delas.

Figura 18 - Exemplo de diagrama de atividades



Fonte: Autoria Própria, 2025.

Acima, um exemplo de diagrama de atividades. Ele começa no laço inicial, anteriormente descrito no diagrama de máquina de estados, e percorre um fluxo de ações, em que o sistema recebe a senha e e-mail do tutor do pet e verifica se já existe uma conta cadastrada com essas informações no site. Caso exista, o tutor é informado e o ciclo é concluído. No entanto, caso não exista, ele informará ao sistema seu nome, data de nascimento, CPF, CEP e telefone, que, por sua vez, receberá esses dados e realizará uma verificação novamente, dessa vez, para garantir que todos os campos estão preenchidos. Caso estejam, continua o curso, com mais uma verificação para

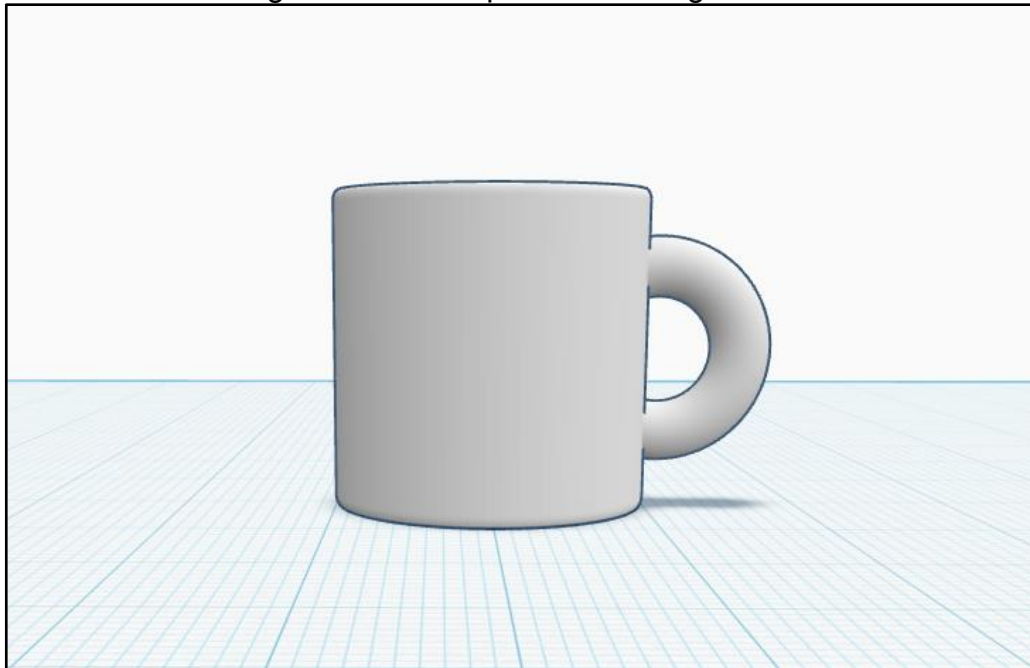
certificar-se de que o tutor do pet tem idade superior a 16 anos antes de cadastrar sua conta no banco de dados do sistema.

2.2.18 Prototipação do Case

Conforme Lor (2018), a modelagem 3D pode simular ambientes e materiais em uma determinada cena, permitindo que um objeto criado artificialmente possa ser replicado fisicamente.

Utensílios de computação gráfica, em específico a modelagem 3D (CAD), possibilitam facilidade em eventuais modificações no desenvolvimento do projeto, segundo Foggiatto, Volpato e Bontorin (2007).

Figura 19 - Exemplo de modelagem 3D

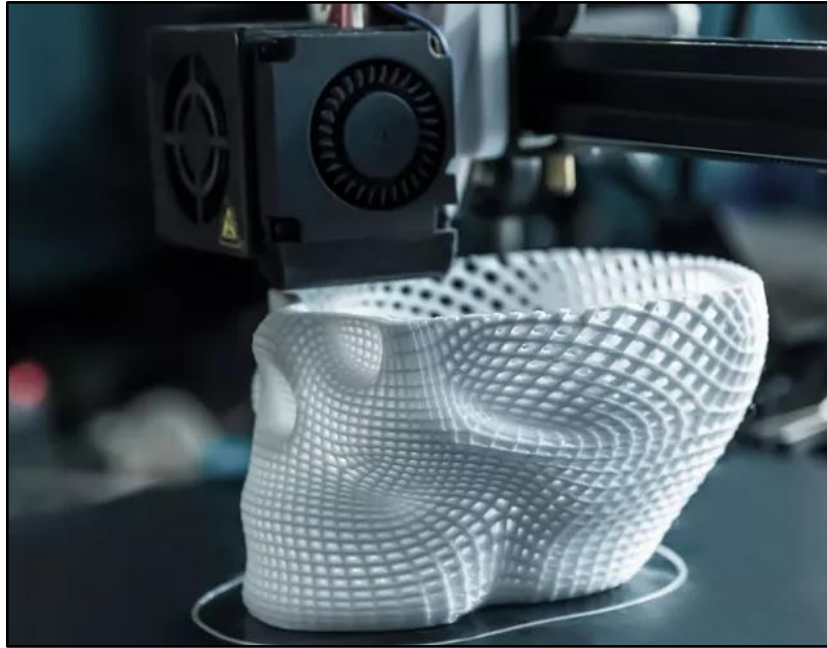


Fonte: Autoria Própria, 2025.

De acordo com Aguiar (2006) a impressão 3D é uma técnica de construir sólidos previamente modelados em 3D, depositando camada por camada, se diferenciando de outros métodos que esculpem o material até a formação da forma.

Como afirma a Secretaria de Estado da Educação do Paraná (2018), a popularização da impressão 3D a tornou uma ferramenta muito importante dentro de espaços escolares, permitindo a prototipação de qualquer ideia.

Figura 20 - Exemplo de impressão 3D



Fonte: UFRJ, 2025.

3 DESENVOLVIMENTO

Durante o seguimento desse capítulo, serão abordadas as etapas percorridas no que diz respeito à concepção e idealização do projeto, documentação do projeto, baseada nos conceitos demonstrados por Gilleanes Guedes em seu livro **UML 2**, bem como Grady Booch, James Rumbaugh e Ivar Jacobson com **UML: Guia do Usuário** e Martin Fowler em **UML Essencial: Um Breve Guia para a Linguagem-Padrão de Modelagem de Objetos**, utilizados para a preparação dos diagramas tratados no referencial teórico. Além disso, com base nessa documentação, será detalhada a construção do dispositivo portátil e a concepção e elaboração das telas do aplicativo móvel, assim concluindo o projeto Sonoris.

3.1 Identidade visual

A comunicação entre pessoas com deficiência auditiva e ouvintes ainda é uma barreira significativa nos dias de hoje, principalmente nos ambientes acadêmicos e profissionais, onde a troca de informações é crucial. Esse fator se dá pela falta de conhecimento necessário sobre a Língua Brasileira de Sinais (LIBRAS) por parte de pessoas ouvintes, e até mesmo demais pessoas com deficiência.

Diante disso, foi realizado diversas pesquisas sobre tecnologias inclusivas que buscam ajudar essas pessoas nesse âmbito. No entanto, observou-se que a uma grande lacuna de tecnologias acessíveis nessa área. Assim, chegou-se à conclusão de desenvolver um dispositivo portátil capaz de auxiliar na vida cotidiana dos surdos, captando e transcrevendo a fala humana em legendas.

O nome desse projeto, Sonoris, foi cuidadosamente selecionado com o propósito de ser fácil de se identificar e memorável para as pessoas, por remeter a palavra “sonoro”, que significa aquilo que produz som.

A identidade visual de uma marca é fundamental para seu reconhecimento para o público-alvo e para ela se estabelecer de forma formal no mercado. O que a define são as combinações do logotipo, paleta de cores e tipografia, capazes de transmitir as emoções necessárias as pessoas que utilizam a aplicação.

Durante a concepção da logo, o objetivo inicial era passar a ideia principal desse projeto, que tem como foco auxiliar pessoas com deficiência auditiva, o que, depois de muita pesquisa, concluiu-se que a representação visual da Sonoris seria feita por meio de um gato branco de olhos azuis que apresentam cerca de 80% de probabilidade de nascer com surdez devido a sua genética.

Figura 21 - Gatos brancos e a surdez



Fonte: Gatinho Branco, 2015.

Durante a conceituação da logo, foi elaborado um esboço com o objetivo de avaliar se ela está adequada antes de sua idealização. Nessa ideia inicial, o gato está em volta de um círculo, brincando com um novelo de lã que, na versão final, representa as ondas sonoras.

Figura 22 - Conceituação da logo



Fonte: Autoria Própria, 2025.

E, após todas as revisões feitas no esboço e conceituação do logo, foi elaborada sua versão final, com cores azuis que remetem aos olhos dos gatos brancos, além do novelo de lã laranja que contrasta com o tom azul presente em toda a logo.

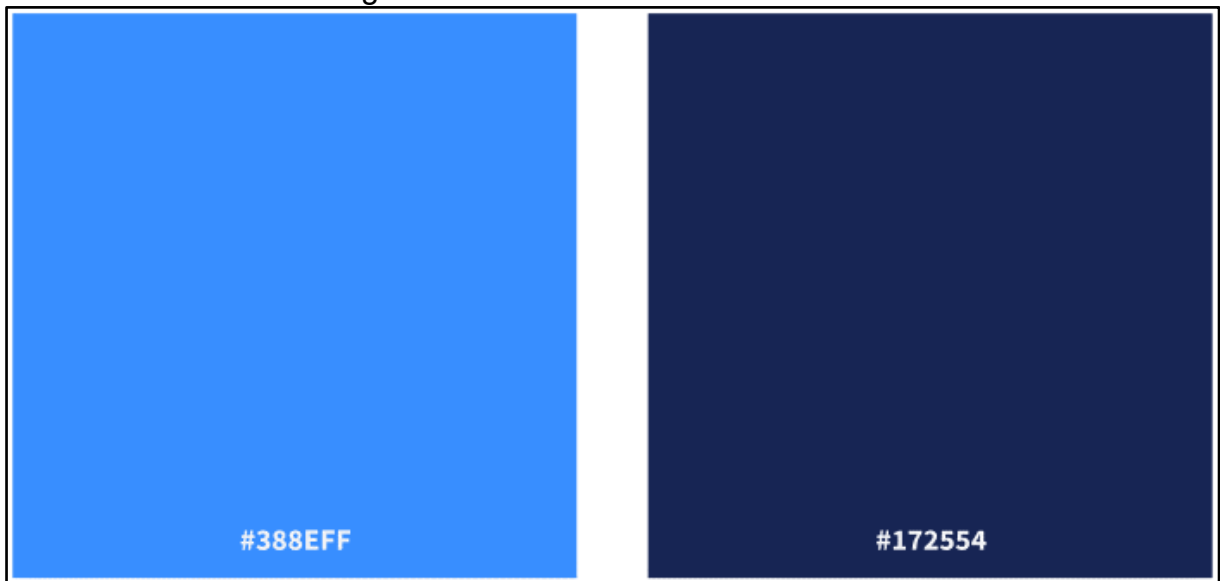
Figura 23 - Idealização da logo



Fonte: Autoria Própria, 2025.

Essas cores azuis não foram apenas utilizadas para representar melhor o gato de olhos azuis, mas também foram usadas pois transmitem aos usuários a sensação de estabilidade, fidelidade, proteção e compromisso.

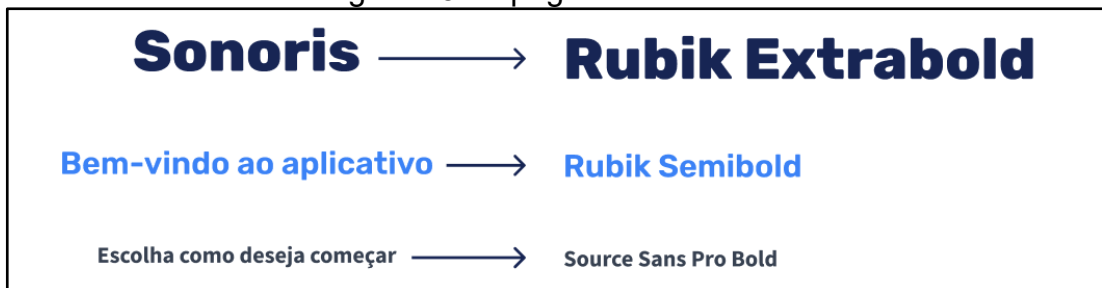
Figura 24 - Paleta de cores da Sonoris



Fonte: Autoria Própria, 2025.

E, por fim, para a tipografia, utiliza-se *Rubik* para os títulos, subtítulos e textos importantes que precisem de destaque, e *Source Sans Pro* para os demais textos do aplicativo.

Figura 25 - Tipografia da Sonoris



Fonte: Autoria Própria, 2025.

3.2 Levantamento de Requisitos

Antes de dar início a codificação do dispositivo e do aplicativo, foi feito um levantamento de requisitos para compreender as necessidades das pessoas com deficiência auditiva, documentar todas as funcionalidades que o oferece sistema e garantir sua qualidade.

As regras de negócio (RN) são todas as condições e atributos característicos que diferem a empresa das demais no mercado. Já os requisitos funcionais (RF) representam todas as funcionalidades e ações que o sistema deve ser capaz de fazer e, por fim, os requisitos não funcionais (RNF) são as características que descrevem como o sistema deve funcionar e condições que deve operar.

Regras de Negócio (RN):

- RN01 – O dispositivo só pode transcrever a fala humana até 3 metros de distância.
- RN02 – Tanto o IoT quando o App devem estar em modo de pareamento para ser pareado
- RN03 – O dispositivo só funciona pareado com o aplicativo oficial.
- RN04 – O dispositivo só funciona com o usuário logado em sua conta.
- RN05 – Cada e-mail deve ser único e não pode estar associado a mais de uma conta.
- RN06 – A senha deve ter no mínimo 8 caracteres, contendo pelo menos uma letra maiúscula, uma letra minúscula e um número.

Requisitos Funcionais (RF) do Aplicativo:

- RF01 – Cadastrar conta.
- RF02 – Efetuar login.
- RF03 – Parear dispositivo.
- RF04 – Manter conversas.
- RF05 – Editar perfil.
- RF06 – Ajustar preferências do dispositivo.
- RF07 – Manter respostas rápidas.

Requisitos Funcionais (RF) do Dispositivo:

- RF01 – Reconhecer a fala humana.
- RF02 – Reproduzir respostas rápidas.

Requisitos Não Funcionais (RNF): OTIMIZAÇÃO

- RNF01 – O aplicativo deve ser compatível com Android.
- RNF02 – O aplicativo deve utilizar fontes acessíveis.
- RNF03 – O sistema deve ter uma interface de fácil usabilidade.
- RNF04 – Deve ser possível salvar conversas com segurança.
- RNF05 – O sistema deve garantir a estabilidade e funcionamento offline parcial do dispositivo.

4 CONSIDERAÇÕES FINAIS

REFERÊNCIAS

AGUIAR, Leonardo De Conti Dias. **Um Processo para Utilizar a Tecnologia de Impressão 3D na Construção de Instrumentos Didáticos para o Ensino de Ciências**. 2016. 226 f. Dissertação (Programa de Pós-Graduação em Educação para a Ciência) – Universidade Estadual Paulista “Júlio de Mesquita Filho”, Bauru, 2016.

ALURA. **Entenda o Figma**: uma solução inovadora para projetos de design. Disponível em: <https://www.alura.com.br/artigos/figma>. Acesso em: 21 jun. 2025.

ALURA. AOVVS Sistemas de Informática S.A, c2011. **UI Design**: o que é, UX vs UI e um Guia da profissão de UI Designer | Alura. Disponível em: <<https://www.alura.com.br/artigos/ui-design#mas-afinal-o-que-e-uma-interface> >. Acesso em: 27 de maio. de 2025.

ALURA. AOVVS Sistemas de Informática S.A, c2011. **Flutter**: o que é, como funciona e por que usar | Alura. Disponível em: <https://www.alura.com.br/artigos/flutter>. Acesso em: 20 de jun. de 2025.

AME. **Comunicação entre surdos e ouvintes**: principais desafios. São Paulo, 28 abr. 2021. Disponível em: <https://www.ame-sp.org.br/comunicacao-entre-surdos-e-ouvintes-principais-desafios/>. Acesso em: 7 set. 2025.

ANDRADE, Patrick R. P. **Identificação de Falhas em Sistemas UPS (Uninterruptible Power Supply) por meio de Deep Learning em Sistema Embarcado**. 2023. Artigo (Pós-graduação em Instrumentação) – Universidade Federal de Ouro Preto (UFOP) e Instituto Tecnológico Vale (ITV), Ouro Preto, MG / CEMIG Distribuição S.A., Belo Horizonte, MG, 2023.

ATOJI, Rodolpho Lemini. **Bluetooth e NFC: estudo de caso**. 2010. Trabalho de Formatura – Universidade de São Paulo Instituto de Matemática e Estatística, São Paulo, SP, 2010.

AZEREDO, Marcelo da Costa. **Motores convencionais para aviação**. 2. ed. São Paulo, SP: Editora Dialética, 2024.

BERNARDINO, S.; DE JESUS RODRIGUES DE BARROS, L. **FIREBASE E USO NAS APLICAÇÕES ANDROID E IOS**. Revista Interface Tecnológica, Taquaritinga, SP, v. 21, n. 1, p. 279–287, 2025. DOI: 10.31510/infa.v21i1.1978.

BITENCOURT, Julio. **O guia de Dart**: fundamentos, prática, conceitos avançados e tudo mais. 1. ed. São Paulo: Casa do Código, 2022.

BOOCH, Grady; RUMBAUGH James; JACOBSON, Ivar. **UML: Guia do Usuário**. 12. ed. Rio de Janeiro, RJ: Elsevier, 2012.

BORGES, Luiz Eduardo. **Python para Desenvolvedores**: Aborda Python 3.3. 1. ed. São Paulo: Novatec Editora, 2014.

BRASIL. Lei nº 13.146, de 06 de julho de 2015. Estatuto da Pessoa Com Deficiência. **Lex:** Inclusão da Pessoa com Deficiência, Brasil, p. 23. 2. trim. 2015.

BRASIL ESCOLA. **Dia da Libras:** aprenda sobre a origem e o alfabeto da Língua Brasileira de Sinais. Brasil Escola, 24 abr. 2024. Disponível em: <https://brasilecola.uol.com.br/noticias/dia-da-libras-aprenda-sobre-a-origem-e-o-alfabeto-da-lingua-brasileira-de-sinais/3131231.html>. Acesso em: 6 set. 2025.

CHEN, Constance Jiin. **UX Design:** desenvolvimento de uma interface mobile para BTC focada na experiência do usuário. 2018. Trabalho de Conclusão de Curso (Graduação) – Faculdade de Arquitetura e Urbanismo, Universidade de São Paulo, São Paulo, 2018.

DATE, C. J. **Introdução a sistemas de bancos de dados.** 8. ed. São Paulo: Elsevier, 2004.

DONAT, Wolfram. **Programação do Raspberry Pi com Python.** 1. ed. São Paulo, SP: Novatec Editora, 2018.

EBERMAN, Elivelto *et al.* **Programação para leigos com Raspberry Pi.** 1. ed. Vitória, ES: Edifes, 2017.

ELMASRI, Ramez; NAVATHE, Shamkant B. **Sistemas de Banco de Dados.** 6. ed. São Paulo: Pearson Education do Brasil, 2011.

FOGGIATO, José; VOLPATO, Neri; BONTORIN, Ana. **RECOMENDAÇÕES PARA MODELAGEM EM SISTEMAS CAD-3D.** São Paulo: Congresso Brasileiro de Engenharia de Fabricação, 2007.

FOWLER, Martin. **UML Essencial:** Um Breve Guia para a Linguagem-Padrão de Modelagem de Objetos. 3. ed. Porto Alegre, RS: Bookman, 2005.

FREITAS, Geovane R.; MARANHÃO, Thercia L. G.; FÉLIX, Waleska M. A Deficiência Auditiva e a possibilidade de Inclusão no Mercado de Trabalho. *Id on Line Revista Multidisciplinar e de Psicologia*, Fevereiro de 2017, vol.11, n.34, p. 185-218. ISSN: 1981-1179.

GOOGLE. **Conheça o Firebase para Flutter.** Firebase. Disponível em: <https://firebase.google.com/docs/flutter?hl=pt-br>. Acesso em: 21 jun. 2025.

GUEDES, Gilleanes T. A. **UML 2:** Uma Abordagem Prática. 3. ed. São Paulo, SP: Novatec Editora, 2018.

JUSBRASIL

IBGE – INSTITUTO BRASILEIRO DE GEOGRAFIA E ESTATÍSTICA. **Censo Brasileiro de 2010:** Amostra – Pessoas com deficiência. Brasília: DF, 2012.

LUCIDCHART. Lucid Software Inc, c2008. **O que é wireframe**. Disponível em: <<https://www.lucidchart.com/pages/pt/o-que-e-wireframe#:~:text=Em%20web%20design%2C%20um%20wireframe,uma%20tela%20de%20aplicativo%20m%C3%B3vel>>. Acesso em: 26 de maio. de 2025.

MAGNO, Rodrigo. **As dificuldades da pessoa surda na sociedade brasileira**. JusBrasil, 6 mar. 2021. Disponível em: <https://www.jusbrasil.com.br/artigos/as-dificuldades-da-pessoa-surda-na-sociedade-brasileira/1176514129>. Acesso em: 7 set. 2025.

MAGRANI, Eduardo. **INTERNET DAS COISAS**. 1. ed. Rio de Janeiro, RJ: Editora FGV, 2018.

MARINHO, Leonardo H. **Flutter Framework: Desenvolva aplicações móveis no Dart Side!** 1. ed. São Paulo: Casa do Código, 2020.

MELO, Adriana; ABELHEIRA, Ricardo. **Design Thinking & Thinking Design: Metodologia, ferramentas e uma reflexão**. São Paulo: Novatec Editora, 2015.

MERCADO LIVRE. Ebazar.com.br LTDA, c1999. Microfone De Mesa Omnidirecional Com Mudo Usb Home Office. Disponível em: <https://produto.mercadolivre.com.br/MLB-1809376559-microfone-de-mesa-omnidirecional-com-mudo-usb-home-office-_JM#reviews>. Acesso em: 21 de junho. de 2025.

NEVES SILVA, Daniel. **Língua Brasileira de Sinais (Libras)**. Mundo Educação, Disponível em: <https://mundoeducacao.uol.com.br/educacao/lingua-brasileira-de-sinais-libras.htm>. Acesso em: 6 set. 2025.

OLIVEIRA, Sérgio. **Internet das Coisas com ESP8266, Arduino e Raspberry Pi**. 1. ed. São Paulo, SP: Novatec Editora, 2017.

Lei 10.436/2002

PAGANI, Talita. **Design Thinking: como usar a criatividade para resolver problemas complexos**. São Paulo: Senac São Paulo, 2018.

PYTHON PACKAGING USER GUIDE. **Guia de Usuário para Empacotamento de Python**. 2025. Disponível em: <https://packaging.python.org/pt-br/latest/>. Acesso em: 21 junho. 2025.

RAMALHO, Luciano. **Python fluente: programação clara, concisa e eficaz**. 1. ed. São Paulo: Novatec, 2015.

RECK, Marcelo Sala. **Beacons BLE - Bluetooth Low Energy - design e análise de um sistema de localização indoor**. 2016. Trabalho de Conclusão de Curso (Bacharelado em Engenharia de Controle e Automação) – Universidade de Caxias do Sul, Caxias do Sul, RS, 2016.

ROBOCORE. RoboCore Tecnologia LTDA, c2008. Raspberry Pi 3 - Model B+ Anatel – Placa Raspberry Pi - RoboCore. Disponível em: <<https://www.robocore.net/placa-raspberry-pi/raspberry-pi-3-model-b-plus>>. Acesso em: 21 de maio. de 2025.

RODRIGUES, Amanda Anjolin. **Desenvolvimento de aplicativo Android utilizando os serviços do Firebase**. 2021. Trabalho de Conclusão de Curso (Tecnólogo em Análise e Desenvolvimento de Sistemas) – Faculdade de Tecnologia de São Paulo, São Paulo, 2021.

RODRIGUES, Patrícia Rocha; ALVES, Lynn Rosalina Gama. **Tecnologia assistiva - uma revisão do tema**. Holos, Natal, v. 6, p. 170-180, 2013.

RONALDO, Franklin. **Desenvolvimento de um sistema IoT para medição de nível de rios**. 2022. Trabalho de Conclusão de Curso (Engenharia Eletrônica) – Universidade Tecnológica Federal do Paraná, Paraná, 2022.

SANTOS, Renata Ferreira dos; SAMPAIO, Priscila Yukari Sewo; SAMPAIO, Ricardo Aurélio Carvalho; GUTIERREZ, Gustavo Luis; ALMEIDA, Marco Antonio Bettine de. **Tecnologia assistiva e suas relações com a qualidade de vida de pessoas com deficiência**. Revista de Terapia Ocupacional da Universidade de São Paulo, São Paulo, Brasil, v. 28, n. 1, p. 54–62, 2017

SANTOS, Sandro. **Introdução à IoT: Desvendando a Internet das Coisas**. 1. ed. Joinville, SC: Clube de Autores, 2018.

SECRETARIA DE ESTADO DA EDUCAÇÃO DO PARANÁ. Superintendência da Educação. Departamento de políticas e Tecnologias Educacionais. **Impressora 3D: imaginar, planejar e materializar**. 2018.

SILVA, Davidson Felliipe. **SISTEMA DE COMUNICAÇÃO BLUETOOTH UTILIZANDO MICROCONTROLADOR**. 2009. Trabalho de Conclusão de Curso (Bacharel em Engenharia da Computação) – Escola Politécnica de Pernambuco – Universidade de Pernambuco, Recife, PE, 2009.

SILVA, J. C. da; HORA, H. R. da; CARVALHO, R. A. de. **Prospecção tecnológica para a comunicação imediata entre surdos e ouvintes**. Revista Sinalizar, Goiânia, v. 4, 2019. DOI: 10.5216/rs.v4.57913.

SILVA, Gabriel Luiz; SILVA, Luiz Gustavo Pereira. **ÓCULOS DE ASSISTÊNCIA NA LOCOMOÇÃO PARA DEFICIENTES VISUAIS**. 2025. Artigo – Centro Universitário Mário Palmério, Monte Carmelo, MG, 2025.

SOARES, F. C.; BARBOSA, M. P. **Metodologia utilizada para calcular o bloco óptico de um sistema de triagem da acuidade visual utilizando tela LCD**. ForScience, v. 6, n. 1, 16 maio 2018.

STOREK, Marco. **MELHORIA DA QUALIDADE NA ENGENHARIA DO PRODUTO: UMA APLICAÇÃO EM CAPACITORES PARA UPS**. 2007. Dissertação (Mestrado em Engenharia de Produção) – Universidade Federal do Rio Grande do Sul Escolha de Engenharia, Porto Alegre, RS, 2007.

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

TOLEDO, João. **Fundamentos de experiência do usuário – UX design e user interface – UI**. 1. ed. São Paulo, SP: Editora Senac São Paulo, 2024.

TOSO, C.; CERUTTI, E.; GRANDI, S.; SANTOS CRUZ, J. A. A tecnologia assistiva no ensino superior: reflexões sobre seu uso para alunos ouvintes e surdos. **Revista on line de Política e Gestão Educacional**, Araraquara, v. 22, n. 3, p. 1065–1080, 2018. DOI: 10.22633/rpge.v22i3.11874.

UNIVERSIDADE FEDERAL DE MINAS GERAIS – UFMG; SILVA, Bárbara Vitor et al. **Setembro Azul: A Libras como Língua e a História da Libras**. Espaço do Conhecimento UFMG, 13 set. 2022. Disponível em: <https://www.ufmg.br/espacodoconhecimento/setembro-azul-a-libras-como-lingua-e-a-historia-da-libras/> . Acesso em: 6 set. 2025.

VERETA, C.; ... STREIECHEN, E.; **O surdo no mercado de trabalho**. Revista Processando o Saber, v. 16, n. 01, 13-24, 4 jun. 2024. DOI 10.5281/zenodo.14219063. Disponível em: <https://www.fatecpg.edu.br/revista/index.php/ps/article/view/333>. Acesso em: 6 set. 2025.

LOR, Elen. **Apostila CEG225 2018**. Paraná: Universidade Federal do Paraná, 2018.

ZAMMETTI, Frank. **Flutter na prática**: Melhore seu desenvolvimento mobile com o SDK open source mais recente do Google. 1. ed. São Paulo: Novatec, 2020.