

**CENTRO ESTADUAL DE EDUCAÇÃO TECNOLÓGICA PAULA
SOUZA**

ETEC ZONA LESTE

NOVOTEC DESENVOLVIMENTO DE SISTEMAS

Alan de Lima Silva

Bruno Costa Rezende

Edward Mevis da Silva

Gustavo Dias da Silva Cruz

AR-PIN: Ferramenta para aprendizado de sistemas embarcados

São Paulo

2023

Alan de Lima Silva

Bruno Costa Rezende

Edward Mevis da Silva

Gustavo Dias da Silva Cruz

AR-PIN: Ferramenta para aprendizado de sistemas embarcados

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

São Paulo

2023

DEDICATÓRIA

Dedicamos esta monografia a todos que estiveram presentes durante a realização deste trabalho. Aos meus colegas de equipe, que compartilhamos não só desafios e adversidades ao longo deste processo, mas também momentos de alegria e descoberta.

AGRADECIMENTOS

Gostaria de expressar meu profundo agradecimento ao orientador Jeferson Roberto de Lima, ao Professor Carlos, Eliel Andrade Matos Godoy e à incrível equipe que se esforçou diligentemente na realização do nosso trabalho de conclusão de curso. Sem o comprometimento e a dedicação de cada um, esse projeto não teria alcançado o sucesso que alcançou.

Alan de Lima Silva, Bruno Costa Rezende, Edward Mevis da Silva e Gustavo Dias da Silva Cruz, vocês foram fundamentais nessa jornada. Cada um de vocês trouxe habilidades únicas e contribuições valiosas, tornando nosso trabalho um verdadeiro exemplo de colaboração e empenho coletivo.

Agradeço pelo compromisso, pelo tempo e pelos esforços dedicados a este projeto. Vocês demonstraram profissionalismo exemplar, paixão pela pesquisa e um trabalho em equipe notável. Sem dúvida, essa experiência foi enriquecedora e marcante em minha formação acadêmica.

Carlos, seu apoio e orientação foram inestimáveis ao longo deste processo. Sua expertise, paciência e vontade de compartilhar conhecimento foram cruciais para o sucesso do nosso trabalho. Sou grato pela sua disponibilidade constante, pelos valiosos insights e pela motivação contínua que você proporcionou.

A todos vocês, meus sinceros agradecimentos. Espero que possamos continuar colaborando e alcançando conquistas ainda maiores em nossas futuras empreitadas acadêmicas e profissionais. Vocês são uma equipe verdadeiramente excepcional, e tenho a sorte de ter tido a oportunidade de trabalhar com todos vocês.

EPÍGRAFE

“Um homem não pode entender a arte que está estudando se apenas procura o resultado final sem ter tempo para se aprofundar no raciocínio do estudo”

Miyamoto Musashi

Resumo

Com a crescente dependência da tecnologia e a necessidade de preparar os estudantes para o mundo digital, torna-se evidente a importância da disponibilização de recursos e utilização de ferramentas digitais. A falta de recursos eletrônicos para o aprendizado nas escolas é uma realidade enfrentada por muitas instituições de ensino no Brasil. Um dos recursos tecnológicos mais promissores para o aprendizado é o Arduino, um sistema embarcado que possuí uma plataforma de prototipagem eletrônica de código aberto que consiste em uma placa de circuito e um microcontrolador programável, contendo uma variedade de pinos de entrada e saída que permitem a conexão de diferentes componentes eletrônicos. O presente trabalho, portanto, foi elaborado a partir do citado problema, trazendo como resultado um aplicativo de realidade aumentada para explorar os conceitos de eletrônica, Arduino Uno R3 e sistemas embarcados. O aplicativo oferece recursos interativos, como, materiais didáticos, questionários e a possibilidade de visualização em tempo real de projetos com Arduino Uno R3 utilizando a realidade aumentada, permitindo que os estudantes aprendam de forma imersiva.

Palavras-Chave: Realidade. Aumentada. Arduino UNO R3. Sistemas. Embarcados.

Abstract

With the growing dependence on technology and the need to prepare students for the digital world, the importance of providing resources and using digital tools becomes evident, the lack of electronic resources for learning in schools is a reality faced by many educational institutions in Brazil. One of the most promising technological resources for learning is Arduino, open-source electronics prototyping platform consisting of a circuit board and a programmable microcontroller, containing a variety of input and output pins that allow the connection of different electronic components. The present work, therefore, was elaborated from the mentioned problem, resulting in an augmented reality application to explore the concepts of electronics and Arduino Uno R3, the application offers interactive resources, such as didactic materials, questionnaires, and the possibility of real-time visualization of projects with Arduino Uno R3 using augmented reality, allowing students to learn in an immersive way.

Keywords: Augmented. Reality. Arduino. UNO. R3. Embedded. Systems.

LISTA DE FIGURAS

Figura 1 - Exemplo de Diagrama de Caso de Uso	17
Figura 2 - Exemplo de Diagrama de Atividade	18
Figura 3 - Exemplo de Diagrama de Sequência	19
Figura 4 - Exemplo de Diagrama de Classe.....	20
Figura 5 - Exemplo de Realidade Aumentada	22
Figura 6 - Exemplo de Eletrônica	24
Figura 7 - Exemplo de Código em Arduino Parte 1	25
Figura 8 - Exemplo de Código em Arduino Parte 2	26
Figura 9 - Resultado da Codificação em Arduino	26
Figura 10 - Exemplo de Engine Unity.....	27
Figura 11 - Resultado do Projeto Unity.....	28
Figura 12 – Exemplo de WireFrame.....	30
Figura 13 - Primeiro Exemplo de Código em C#	31
Figura 14 - Segundo Exemplo de Código em C#	32
Figura 15 - Terceiro Exemplo de Código em C#.....	33
Figura 16 - Resultado da Codificação em C#.....	34
Figura 17 - Exemplo de Imagem No Blender	35
Figura 18 - Primeiro Exemplo de Código com Flutter.....	36
Figura 19 - Segundo Exemplo de Código com Flutter.....	37
Figura 20 – Resultado da Codificação com Flutter.....	38
Figura 21 - Exemplo de Banco Não Relacional: Firebase	39
Figura 22 - Exemplo de Grafo	40
Figura 23 - Diagrama de Caso de Uso AR-PIN	41
Figura 24 - Diagrama de Classe AR-PIN.....	52
Figura 25 - Diagrama de Atividade Realizar Cadastro	54
Figura 26 - Diagrama de Atividade Realizar Login	55
Figura 27 - Diagrama de Atividade Alterar Foto.....	56
Figura 28 - Diagrama de Atividade Deletar Conta	57
Figura 29 - Diagrama de Atividade Realizar LogOut	58
Figura 30 - Diagrama de Atividade Selecionar Tutorial	59
Figura 31 - Diagrama de Atividade Visualizar Exemplo de Código.....	60
Figura 32 - Diagrama de Atividade Visualizar Material Didático	61

Figura 33 - Diagrama de Atividade Realizar Questionário.....	62
Figura 34 - Diagrama de Atividade Visualizar Modelo 3D em RA.....	63
Figura 35 - Diagrama de Sequência Cadastrar Desenvolvedor	64
Figura 36 - Diagrama de Sequência Visualizar Conta.....	65
Figura 37 - Diagrama de Sequência Deletar Conta.....	66
Figura 38 - Diagrama de Sequência Realizar Questionário	67
Figura 39 - Diagrama de Sequência Realizar Login.....	68
Figura 40 - Diagrama de Sequência Visualizar Código.....	69
Figura 41 - Diagrama de Sequência Visualizar Material.....	70
Figura 42 - Diagrama de Sequência Visualizar Modelo.....	71
Figura 43 - Diagrama de Sequência Visualizar Questionário	72
Figura 44 - Diagrama de Sequência Visualizar Tutorial.....	73
Figura 45 - Grafo AR-PIN	74
Figura 46 - Wireframe de Baixa/Alta Fidelidade "Realizar Login"	75
Figura 47 - Wireframe Baixa/Alta Fidelidade "Realizar Cadastro"	76
Figura 48 - Wireframe de Baixa/Alta Fidelidade "Perfil"	77
Figura 49 - Wireframe de Baixa/Alta Fidelidade "Página Inicial"	78
Figura 50 - Wireframe de Baixa/Alta Fidelidade "Tutorial Aberto"	79
Figura 51 - Wireframe de Baixa/Alta Fidelidade "Material Didático"	80
Figura 52 - Wireframe de Baixa/Alta Fidelidade "Questionário"	81
Figura 53 - Wireframe de Baixa/Alta Fidelidade "Questionário Concluído"	82
Figura 54 - Wireframe de Baixa/Alta Fidelidade "Código"	83
Figura 55 - Wireframe de Baixa/Alta Fidelidade "Baixar Modelos 3D"	84
Figura 56 - Wireframe de Baixa e Alta Fidelidade "Modelo 3D"	85

LISTA DE QUADROS

Quadro 1 - Descrição do caso de uso “Realizar Cadastro” (desenvolvedor).....	44
Quadro 2 - Descrição do caso de uso “Realizar Login” (desenvolvedor)	45
Quadro 3 - Descrição do Caso de Uso “Visualizar Tutorial”	46
Quadro 4 - Descrição do caso de uso "Visualizar Modelo 3D"	47
Quadro 5 - Descrição do Caso de Uso “Visualizar Questionário”.....	48
Quadro 6 - Descrição do caso de uso "Visualizar Perfil" (desenvolvedor).....	49
Quadro 7 - Descrição do caso de uso "Editar Foto de Perfil" (desenvolvedor).....	50
Quadro 8 - Descrição do caso de uso "Deletar Perfil" (desenvolvedor)	51

LISTA DE ABREVIATURAS E SIGLAS

Ampères (A)

Bidimensional (2d)

C Sharp (C#)

Common Language Runtime (CLR)

Corrente (I)

Network (.NET)

No Structered Query Language (NoSQL)

Realidade Aumentada (RA)

Resistência (R)

Resistência Equivalente (REQ)

Tecnologias da Informação e Comunicação (TiCs)

Tridimensional (3D)

Unified Modeling Language (UML)

Universal Serial Bus (USB)

Volts (V)

LISTA DE SÍMBOLOS

Densidade (ρ)

Ohms (Ω)

SUMÁRIO

1	INTRODUÇÃO.....	14
2	REFERENCIAL TEÓRICO.....	15
2.1	AR-PIN: Ferramenta de aprendizado de sistemas embarcados	15
2.2	Tecnologias utilizadas	16
2.2.1	UML.....	16
2.2.2	Tecnologias da Informação e Comunicação.....	20
2.2.3	Realidade Aumentada	21
2.2.4	Eletrônica	22
2.2.5	Arduíno.....	24
2.2.6	Unity	27
2.2.7	Wireframes	28
2.2.8	C#.....	30
2.2.9	Blender.....	34
2.2.10	Flutter.....	35
2.2.11	Banco de Dados Não Relacional.....	38
3	DESENVOLVIMENTO	40
3.1	Diagrama de Casos de Uso AR-PIN	40
3.1.1	Documentação dos Casos de Uso AR-PIN	41
3.2	Diagrama de Classe AR-PIN	52
3.3	Diagrama de Atividade AR-PIN.....	53
3.4	Diagramas de Sequência AR-PIN.....	64
3.5	Representação em grafo do banco de dados AR-PIN	74
3.6	Prototipação das telas	74
4	Considerações Finais	86
	REFERÊNCIAS	87

1 INTRODUÇÃO

A respeito do tema educação Tardif (2014) disserta sobre a importância da fusão dos saberes, nas quais sua pluralidade torna o conhecimento gerado rico. Assim, não somente a teoria é necessária, bem como também formas de estimular e praticar o conhecimento. Realizar exercícios e projetos ajuda a aplicar o conhecimento adquirido e identificar lacunas de compreensão, desta forma, complementando a teoria.

E quando se retrata as ferramentas que estimulam o processo educacional, existem meios de inserirmos novas ferramentas em quaisquer esquemas pedagógicos. De acordo com Christensen et al. (2013), quando é introduzida uma nova tecnologia que complementa à antiga, ocorre uma inovação sustentada, o que tem sido historicamente positivo.

O uso de ferramentas visuais e táteis pode ser de grande ajuda para alunos visuais e cinestésicos, recursos como mapas conceituais, diagramas, vídeos, simulações, experimentos práticos e réplicas físicas de conceitos abstratos podem facilitar o aprendizado desse tipo de aluno. De acordo com Barbosa (2018), ferramentas visuais são representações que ajudam a simplificar e transmitir ideias passíveis de modificação.

No que diz respeito à assimilação dos conteúdos acadêmicos, a instituição de ensino deve diversificar as práticas utilizadas nas salas de aula para atender às diversas necessidades dos alunos. Para Saviani (1991), a escola, como representação da instituição de ensino, também deve fornecer as ferramentas necessárias para a prática da ciência.

Não há dúvidas de que as tecnologias da informação e comunicação (TICs), provocaram uma mudança nos métodos tradicionais de ensino, Kenski (2007) atesta que tais tecnologias trouxeram mudanças consideráveis e positivas para a educação. As TICs fornecem acesso ao conhecimento, tornando o aprendizado complexo mais acessível aos alunos dentro e fora do ambiente escolar.

Os computadores tornaram-se uma ferramenta comum na sociedade, servindo como um método para processar informações de diversas formas. Consequentemente, são muito procurados como ferramentas educacionais, uma vez que a informação é parte integrante da aprendizagem (Valente et al, 1999). A educação e a informática tornam-

se assim aliadas com o objetivo de fornecer, capacitar e partilhar informação e conhecimento. Com a devida introdução a este tema, a sua aplicação torna-se essencial, sendo crucial que tais ferramentas sejam acessíveis a todos. Para validar e demonstrar a utilidade de tais ferramentas e tecnologias, este trabalho segue uma pesquisa exploratória com método bibliográfico, que, como demonstra Gil (1991), é realizada por meio de acompanhamento de livros, artigos, revistas, todos demonstrações de outros autores sobre o tema explicado.

2 REFERENCIAL TEÓRICO

Neste capítulo, seguindo o propósito de exploração do tema, fornecemos a fundamentação teórica, abordando tópicos envolvidos, desde o tema do trabalho bem como as tecnologias e métodos utilizados para elaboração e projeção do mesmo.

2.1 AR-PIN: Ferramenta de aprendizado de sistemas embarcados

O aplicativo AR-Pin é uma ferramenta para o aprendizado e exploração dos conceitos de eletrônica, Arduino Uno R3 e sistemas embarcados, o aplicativo oferece recursos interativos, sendo eles, materiais didáticos, questionários e a possibilidade de visualização em tempo real do Arduino Uno R3 e alguns componentes eletrônicos, utilizando a realidade aumentada, permitindo que os estudantes aprendam de forma imersiva. Um dos objetivos do projeto é tornar o ensino acessível para todos, pensando nisso utilizamos em nosso aplicativo uma paleta de cores simples e com alto contraste, para que seja possível usufruir facilmente do conteúdo disponível no aplicativo, fazendo o uso das cores preto, branco e vermelho como cores principais. Optamos por criar um design minimalista com bordas arredondadas combinadas com ícones e linhas simples, para tornar a experiência do usuário mais amigável e fluida.

2.2 Tecnologias utilizadas

Com o propósito de projetar uma aplicação para aprendizado de sistemas embarcados, segue as tecnologias utilizadas para o desenvolvimento da ferramenta AR-PIN.

2.2.1 UML

Considerada uma linguagem de marcação, a Unified Modeling Language (UML), é a linguagem-padrão de modelagem para o desenvolvimento de softwares. De acordo com Guedes (2018), a UML é uma linguagem utilizada para a modelagem de softwares que utilizam a orientação de objetos.

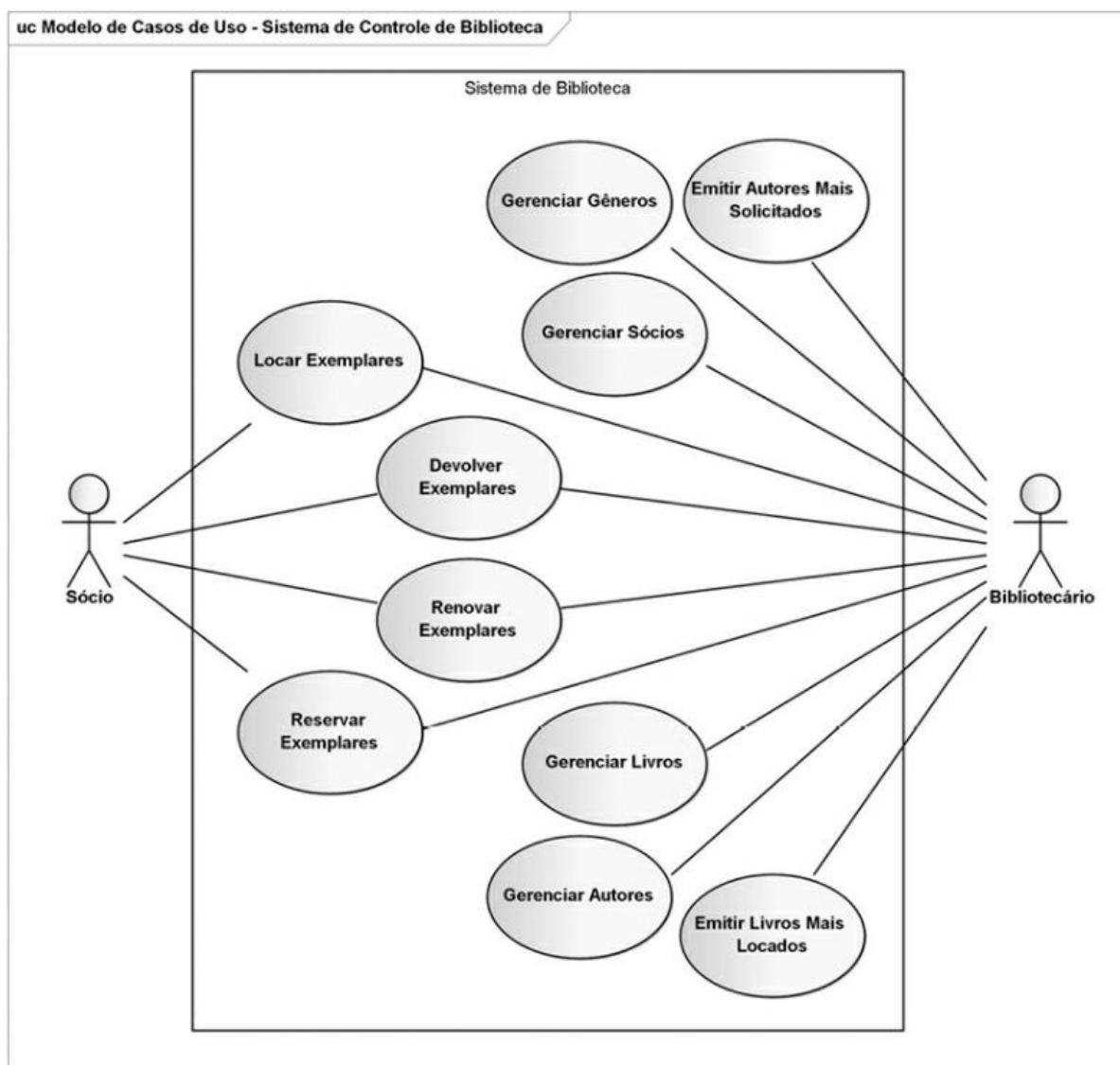
A UML é uma linguagem para visualização, especificação e documentação, contendo diferentes diagramas que exercem inúmeros papéis. Para entender a importância dessa linguagem dentro de um projeto, é necessário conhecer alguns desses diagramas e suas funções.

2.2.1.1 Diagrama de Casos de Uso

O diagrama de caso de uso é uma ferramenta essencial na modelagem de requisitos de um sistema de software. Ele representa as funcionalidades que o sistema entrega para os atores que interagem com o sistema em questão.

O Diagrama de Caso de Uso não só apresenta o sistema de uma forma generalista, como também ajuda na elaboração dos requisitos. Conforme Guedes (2018), os diagramas de casos de uso têm por objetivo mostrar quais funcionalidades o sistema deverá providenciar aos atores, de uma forma generalista.

Figura 1 - Exemplo de Diagrama de Caso de Uso

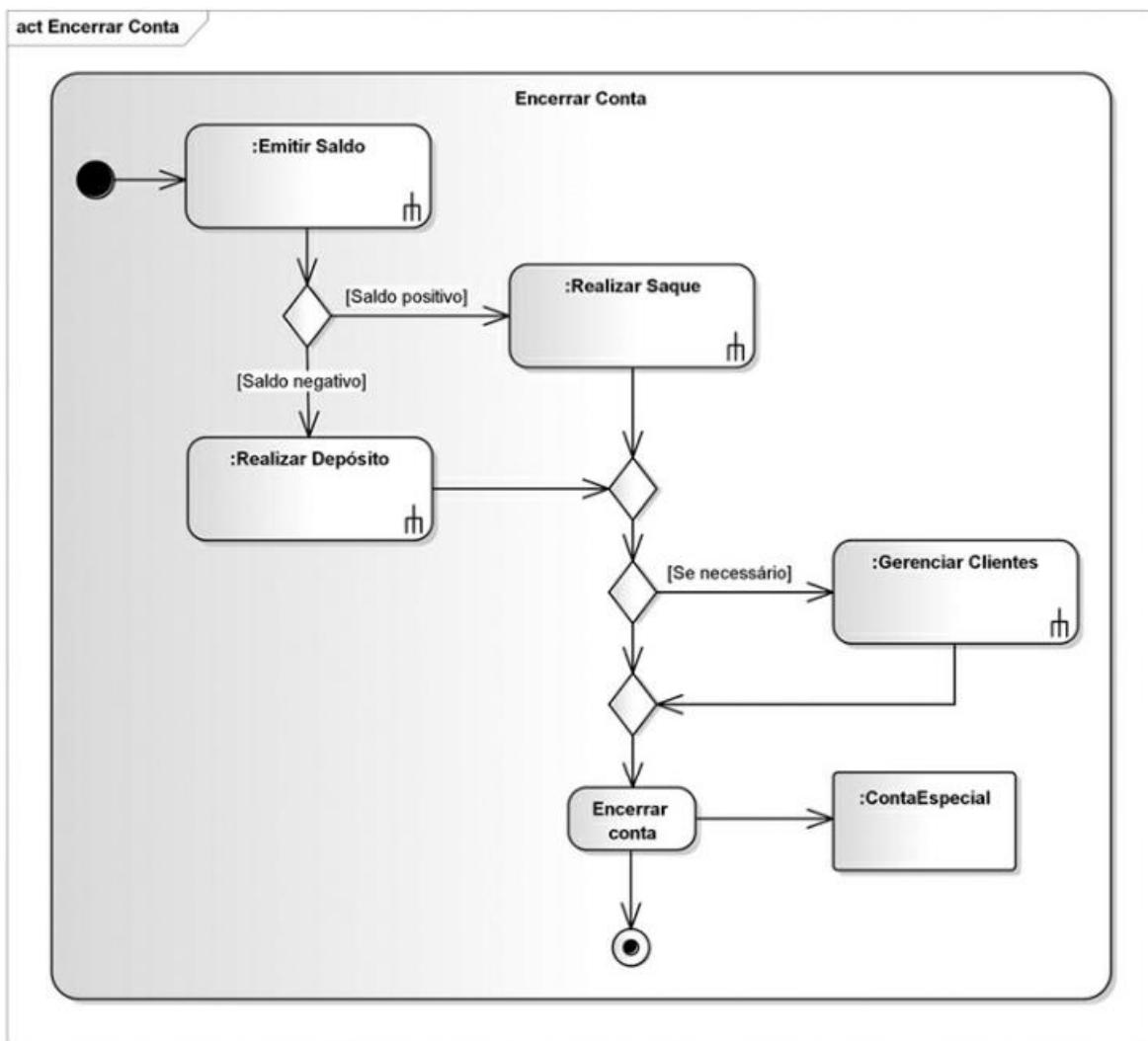


Fonte: (Guedes, 2018).

2.2.1.2 Diagrama de Atividade

O diagrama de atividade, como aponta Guedes (2018), é um diagrama de caráter comportamental que descreve os passos a serem percorridos para a conclusão de uma atividade específica. É fundamental para descrever o funcionamento de atividades do sistema.

Figura 2 - Exemplo de Diagrama de Atividade

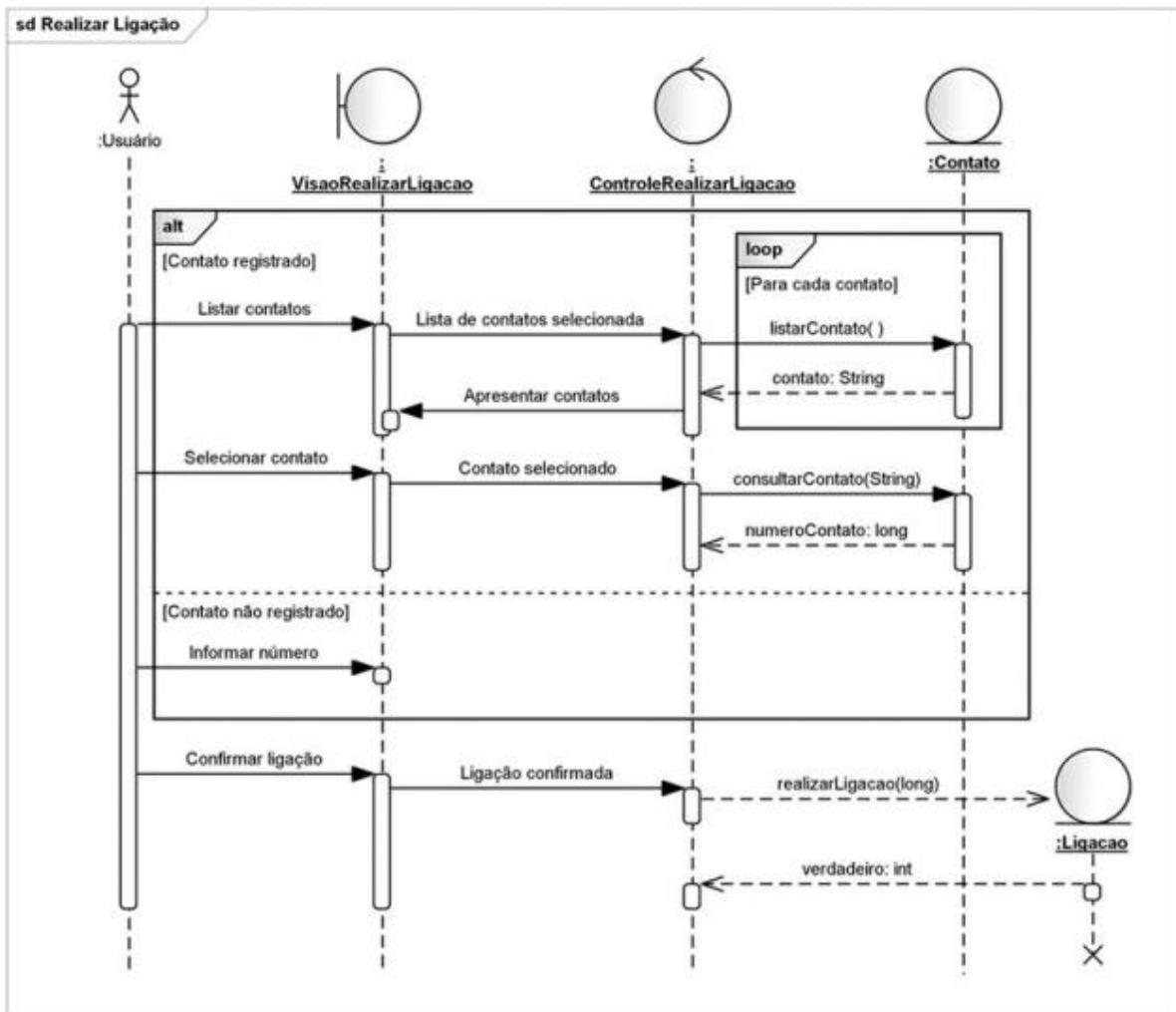


Fonte: (Guedes, 2018).

2.2.1.3 Diagrama de Sequência

Tratando-se também de um diagrama comportamental, o diagrama de sequência, de acordo com Guedes (2018), é um diagrama que se preocupa com a ordem temporal em que as mensagens são trocadas entre os objetos envolvidos em um determinado processo.

Figura 3 - Exemplo de Diagrama de Sequência

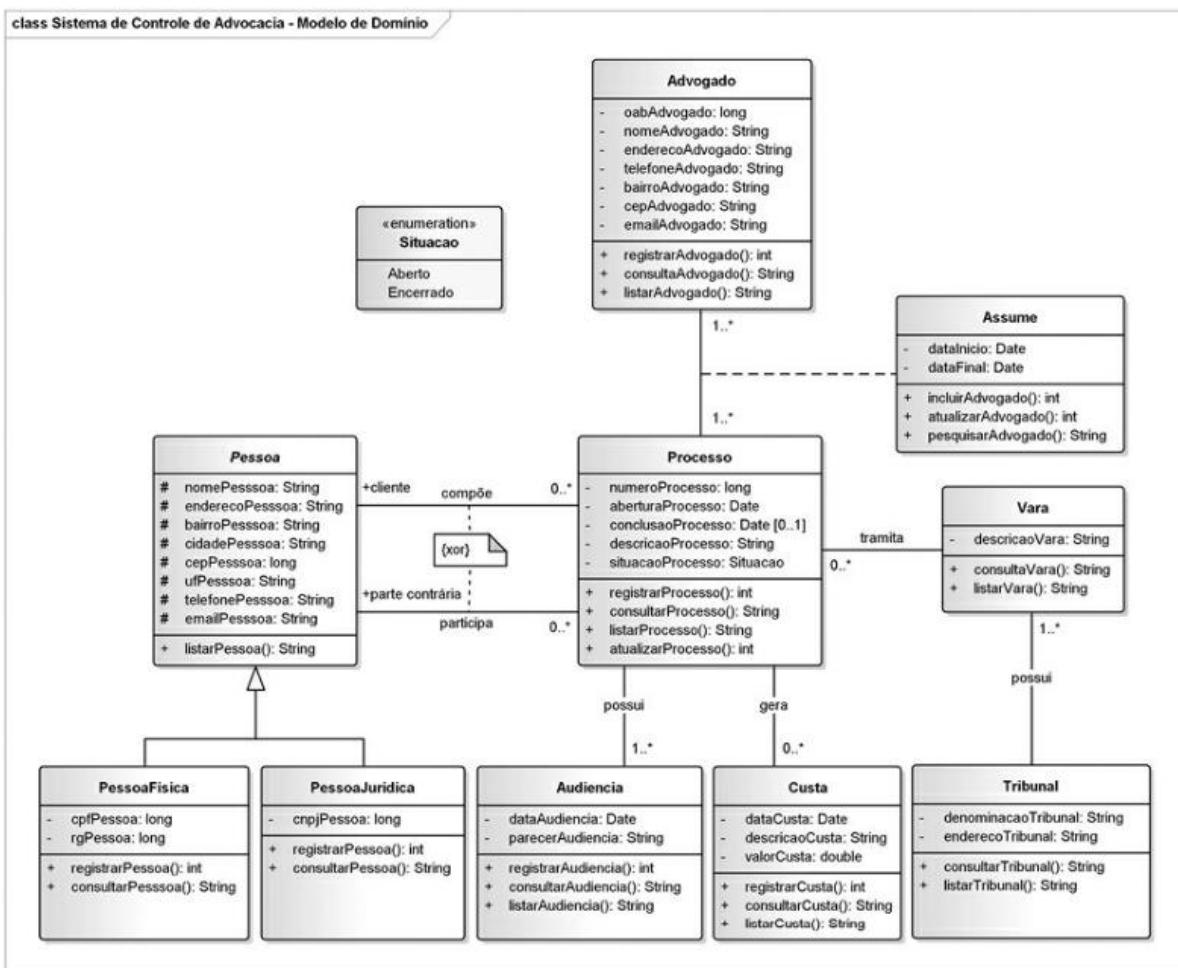


Fonte: (Guedes, 2018).

2.2.1.4 Diagrama de Classe

O diagrama de classes permite mostrar a visualização dos atributos e métodos do sistema, em forma de classes. Para Booch et al. (2006), os diagramas de classes também são importantes para construir sistemas por meio de engenharia de produção e reversa. Esses diagramas, portanto, apresentam uma visão de como as classes estão organizadas, apresentando-os de forma mais concreta a estrutura.

Figura 4 - Exemplo de Diagrama de Classe



Fonte: (Guedes, 2018).

2.2.2 Tecnologias da Informação e Comunicação

As Tecnologias de Informação e Comunicação (TICs) trouxeram mudanças de grande impacto nos métodos de ensino atuais, proporcionando uma aprendizagem mais significativa e interativa Mattar (2010 apud LOPES et.al., 2019), argumenta que as TICs podem facilitar o ensino e a aprendizagem, permitindo que os professores compartilhem o conhecimento de forma dinâmica. Abrangendo as diversas TICs, a tecnologia que mais se destacou na utilização acadêmica foi a realidade aumentada.

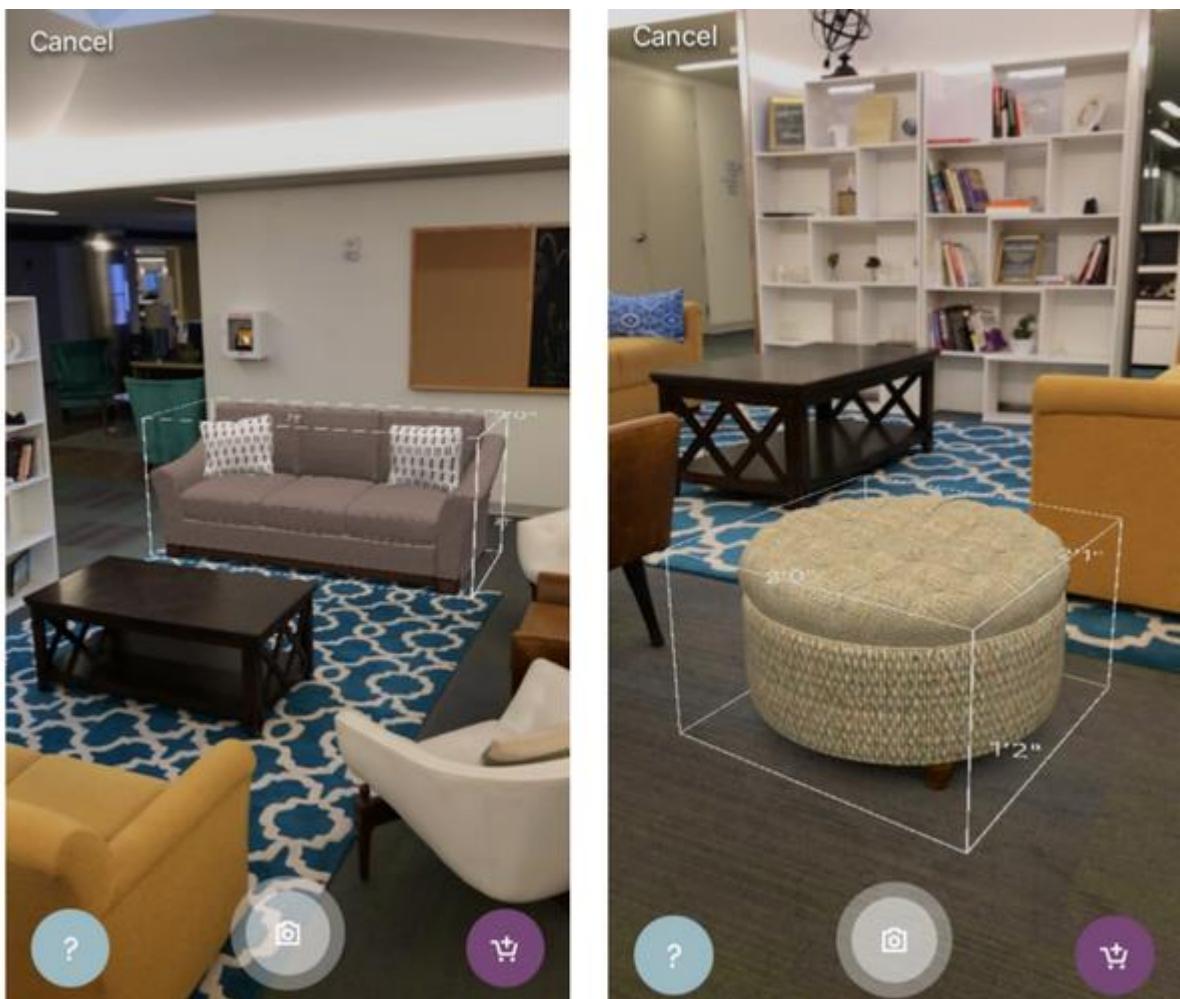
Todavia, este avanço enfrenta algumas barreiras, como por exemplo o atraso das instituições de ensino à adesão destas tecnologias, os autores Veen e Vrakking (2009 apud LOPES et al., 2019) destacam que, apesar dos avanços tecnológicos, as escolas muitas vezes não conseguem acompanhar a fluência tecnológica dos alunos,

desconectando-se do contexto tecnológico destes estudantes. A utilização das Tecnologias de Informação e Comunicação inovam e auxiliam o aprendizado, Prensky (2012 apud LOPES et al., 2019) enfatiza que devido à diferença entre a atualidade e a época em que o modelo escolar foi desenvolvido, o uso das TICs inova e aprimora a experiência de aprendizagem dos alunos de hoje.

2.2.3 Realidade Aumentada

A tecnologia de Realidade Aumentada (RA), nos permite sobrepor o mundo real com elementos virtuais. Os autores Tori, Kirner e Siscouto (2006), definem a realidade aumentada como um ambiente virtual para a interação com o mundo real do usuário. O termo Realidade Aumentada foi criado nos Estados Unidos, por Tom Caudell e David Mizell, a proposta da RA é complementar o mundo ao nosso redor com objetos 3D, Ferreira (2014) disserta que a RA está em constante evolução, e que atualmente os sistemas existentes precisam ser adaptados ao ambiente e o utilizador.

Figura 5 - Exemplo de Realidade Aumentada



Fonte:(Wayfair, 2018).

2.2.4 Eletrônica

O campo da ciência e da engenharia que estuda como controlar a energia elétrica por meio de dispositivos e meios condutivos ou semicondutores é chamado eletrônica, de acordo com os autores (BARBOSA; CANDIOTO, 2022), A eletrônica, utiliza energia elétrica em baixas correntes por meio de dispositivos que aproveitam o fluxo de elétrons, permitindo a criação de circuitos com diversas aplicações.

A tensão elétrica é medida em Volts (V), e representa a diferença de potencial elétrico entre dois pontos, é responsável também pelo deslocamento das cargas elétricas, para Miranda (2006), tensão elétrica é a força capaz de movimentar os elétrons de maneira ordenada para e em um condutor.

Todo material possuí uma resistividade elétrica, que é representado por ρ (densidade), a resistência é medida em Ohms (Ω), os autores (BARBOSA; CANDIOTO, 2022), destacam que os Ohms são a capacidade de um corpo em dificultar a passagem de corrente elétrica.

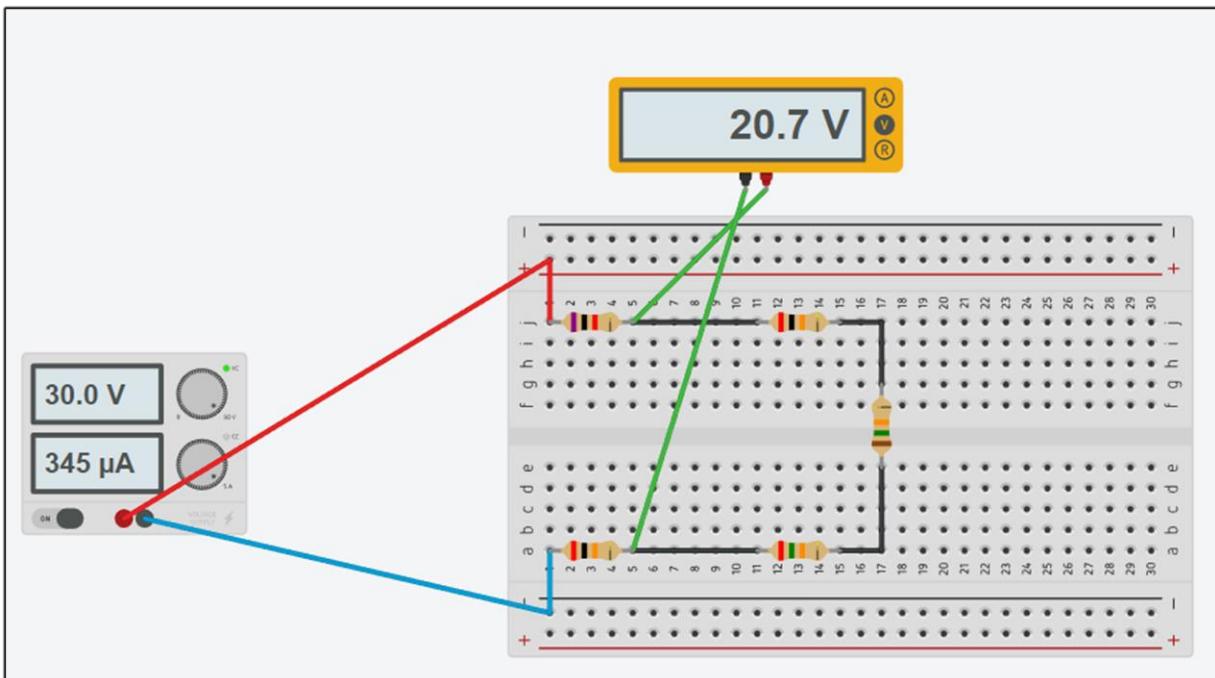
Para medirmos as diferenças de potencial entre os polos, utilizamos a grandeza Ampere (A), para (BARBOSA; CANDIOTO, 2022), os Amperes representam o movimento ordenado de cargas elétricas em um circuito.

A lei de Ohm estabelece uma relação entre corrente elétrica, diferença de tensão e resistência de um condutor, expressa pela **fórmula $I = V/R$** , onde I é a corrente, V é a tensão e R é a resistência (MAGON, 2018).

A resistência equivalente é igual ao resultado do produto pela soma dos respectivos resistores, quando os resistores são conectados em paralelo, a resistência equivalente é calculada usando a **fórmula $1/Req = 1/R1 + 1/R2 + \dots + 1/Rn$, onde Req é a resistência equivalente e R1, R2, ..., Rn são as resistências individuais** (MAGON, 2018).

Para realizar a prototipagem eletrônica com Arduino, é necessário compreender os conceitos básicos de tensão, corrente e resistência, bem como as relações estabelecidas pela lei de Ohm e as associações de resistores em série e em paralelo (BARBOSA; CANDIOTO, 2022; MAGON, 2018).

Figura 6 - Exemplo de Eletrônica



Fonte: Do Próprio Autor, 2023.

2.2.5 Arduíno

O Arduíno, conforme destacado por Souza et al. (2011), é um pequeno computador programável capaz de processar entradas e saídas entre o dispositivo e componentes externos conectados a ele. Esta plataforma de tecnologia e código aberto utiliza sensores de luz, som, temperatura e diversos dispositivos de saída como entrada. Conforme McRoberts (2011), o Arduíno proporciona a oportunidade de desenvolver objetos interativos de forma independente ou conectá-lo a um computador, rede ou à Internet para recuperar e enviar dados, promovendo a interatividade da rede com dispositivos.

A expansão das funcionalidades do Arduíno é viabilizada por meio de shields, conforme explicado por Barbosa e Candioto (2022). Essas placas de circuito contêm outros dispositivos e podem ser conectadas à placa principal para adicionar recursos.

Evans, Noble e Hochenbaum (2013) ressaltam que o Arduíno UNO se diferencia de seus antecessores pela inclusão de um microcontrolador que adapta a conexão serial

para USB, ampliando as opções de uso, como a integração de teclado e outros dispositivos. É fundamental, conforme alertado por Barbosa e Candioto (2022), garantir a alimentação adequada da placa, evitando tensões muito baixas ou altas, que podem comprometer o seu funcionamento.

Figura 7 - Exemplo de Código em Arduino Parte 1

```
1 #include <Servo.h> // Inclui a biblioteca Servo para controlar o motor servo
2 Servo motor;        // Cria um objeto motor do tipo Servo
3
4 int pot;            // Variável para armazenar o valor lido do potenciômetro
5 int LED = 4;         // Pino do LED
6 const int pinBuzzer = 7; // Pino do buzzer
7
8 void setup()
9 {
10    Serial.begin(9600);      // Inicia a comunicação serial
11    motor.attach(9);        // Anexa o motor servo ao pino 9
12    pinMode(LED, OUTPUT);   // Define o pino do LED como saída
13    pinMode(pinBuzzer, OUTPUT); // Define o pino do buzzer como saída
14 }
15
16 void loop()
17 [
18    pot = map(analogRead(A1), 0, 1023, 0, 180); // Lê o valor do potenciômetro e mapeia para o intervalo de 0 a 180
19    Serial.println(pot); // Imprime o valor do potenciômetro na porta serial
20    delay(400); // Aguarda 400 milissegundos
21
22    motor.write(pot); // Define o ângulo do motor servo com base no valor do potenciômetro
23
24    if (pot >= 0 && pot <= 70)
25    {
26        digitalWrite(pinBuzzer, LOW); // Desliga o buzzer
27        digitalWrite(LED, HIGH); // Liga o LED
28        delay(200); // Aguarda 200 milissegundos
29        digitalWrite(LED, LOW); // Desliga o LED
30        delay(200); // Aguarda 200 milissegundos
31    }
]
```

Fonte: Do Próprio Autor, 2023.

Na imagem acima, nota-se componentes presentes que se repetem por outras vezes na linguagem de Arduíno como por exemplo: na linha 1, `#include` irá incluir a biblioteca Servo (`<Servo.h>`) para controlar o motor servo, e na linha abaixo cria-se um objeto do tipo importado. Nas linhas 4 a 7, inicializa-se variáveis utilizadas outrora. Nas linhas 8 e 16 há a chamada dos métodos: `setup`, na qual instancia certas configurações de entrada e saída, e `loop`, na qual tudo que está dentro desse bloco, será repetido conforme as instruções nele passadas.

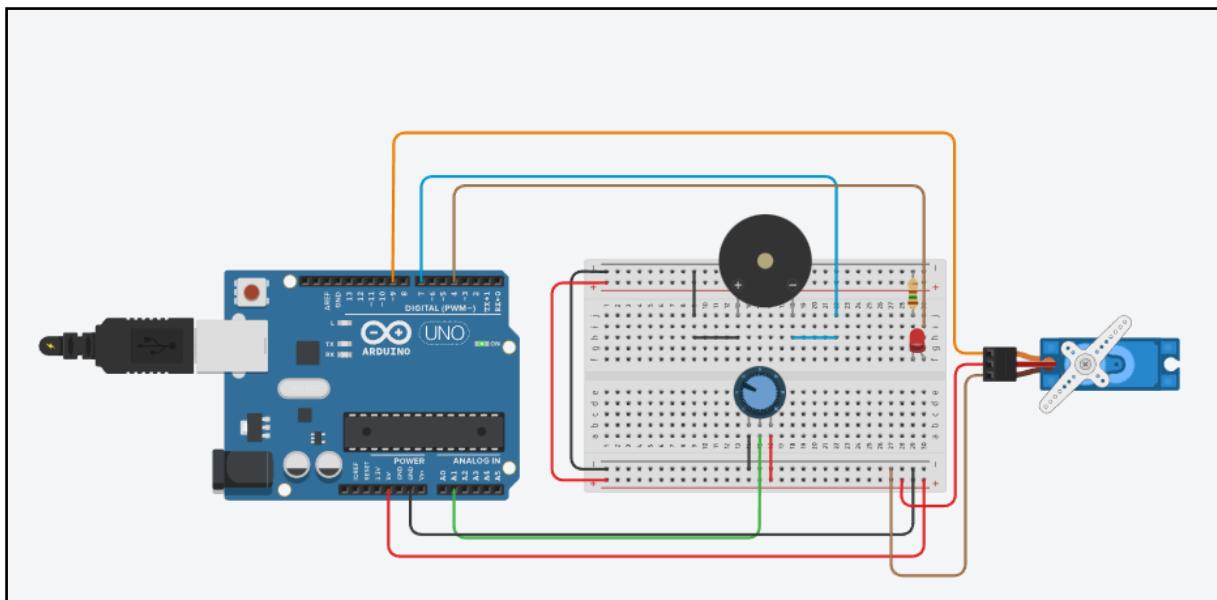
Figura 8 - Exemplo de Código em Arduino Parte 2

```
32
33 if (pot > 90 && pot <= 1023)
34 {
35     digitalWrite(LED, LOW); // Desliga o LED
36     digitalWrite(pinBuzzer, HIGH); // Liga o buzzer
37
38     tone(pinBuzzer, 440); // Gera um tom de 440 Hz no buzzer
39     delay(800); // Mantém o tom por 800 milissegundos
40     noTone(pinBuzzer); // Desliga o tom do buzzer
41     delay(500); // Aguarda 500 milissegundos
42
43     tone(pinBuzzer, 530, 300); // Gera um tom de 530 Hz por 300 milissegundos
44     delay(500); // Aguarda 500 milissegundos
45 }
46 }
```

Fonte: Do Próprio Autor, 2023.

O código de exemplo descreve a ação de acionamento de um led através do movimento realizado no potenciômetro, que ao acionar o led, ativa o buzzer e o servo motor, a seguir o exemplo prático deste código é apresentado.

Figura 9 - Resultado da Codificação em Arduino



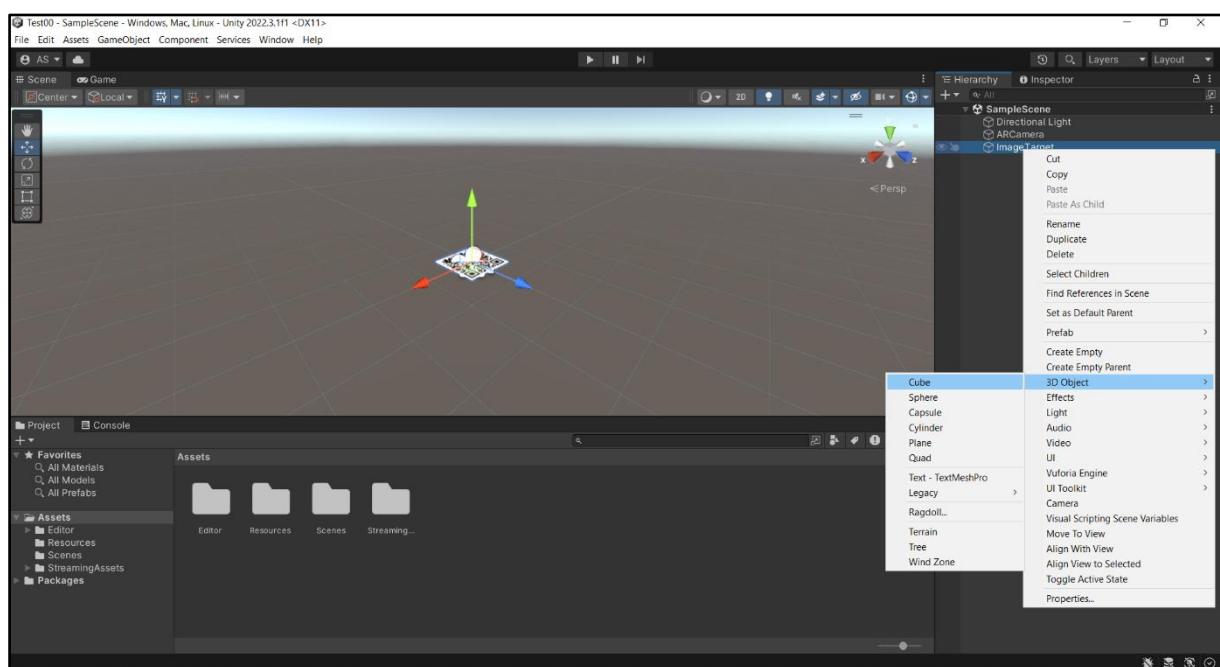
Fonte: Do Próprio Autor, 2023.

2.2.6 Unity

A Unity é uma plataforma de desenvolvimento de aplicativos 2D e 3D, adequada para criação tanto de jogos quanto de sistemas de Realidade Virtual ou Aumentada.

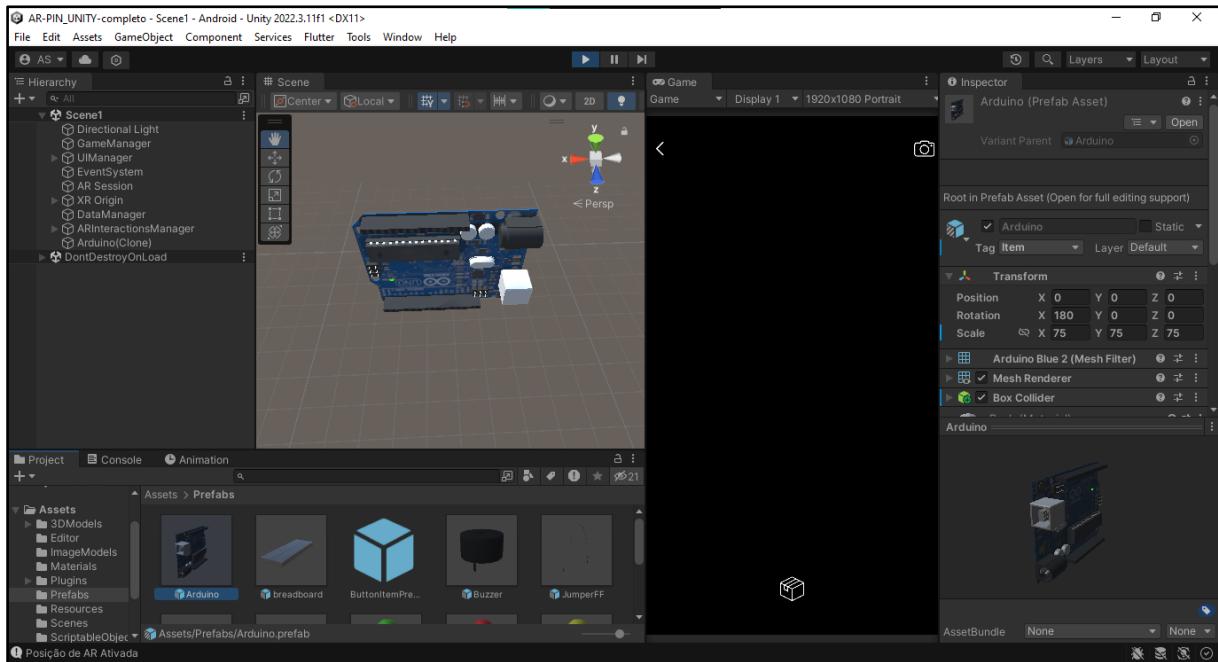
Uma das vantagens do Unity é a sua capacidade de integração com bibliotecas externas por meio de plugins (assets), o que torna a plataforma altamente personalizável Omaia & Machado (2020) mencionam a existência da Asset Store do Unity, onde os desenvolvedores podem baixar e instalar assets diretamente na plataforma.

Figura 10 - Exemplo de Engine Unity



Fonte: Do Próprio Autor, 2023.

Figura 11 - Resultado do Projeto Unity



Fonte: Do Próprio Autor, 2023.

2.2.7 Wireframes

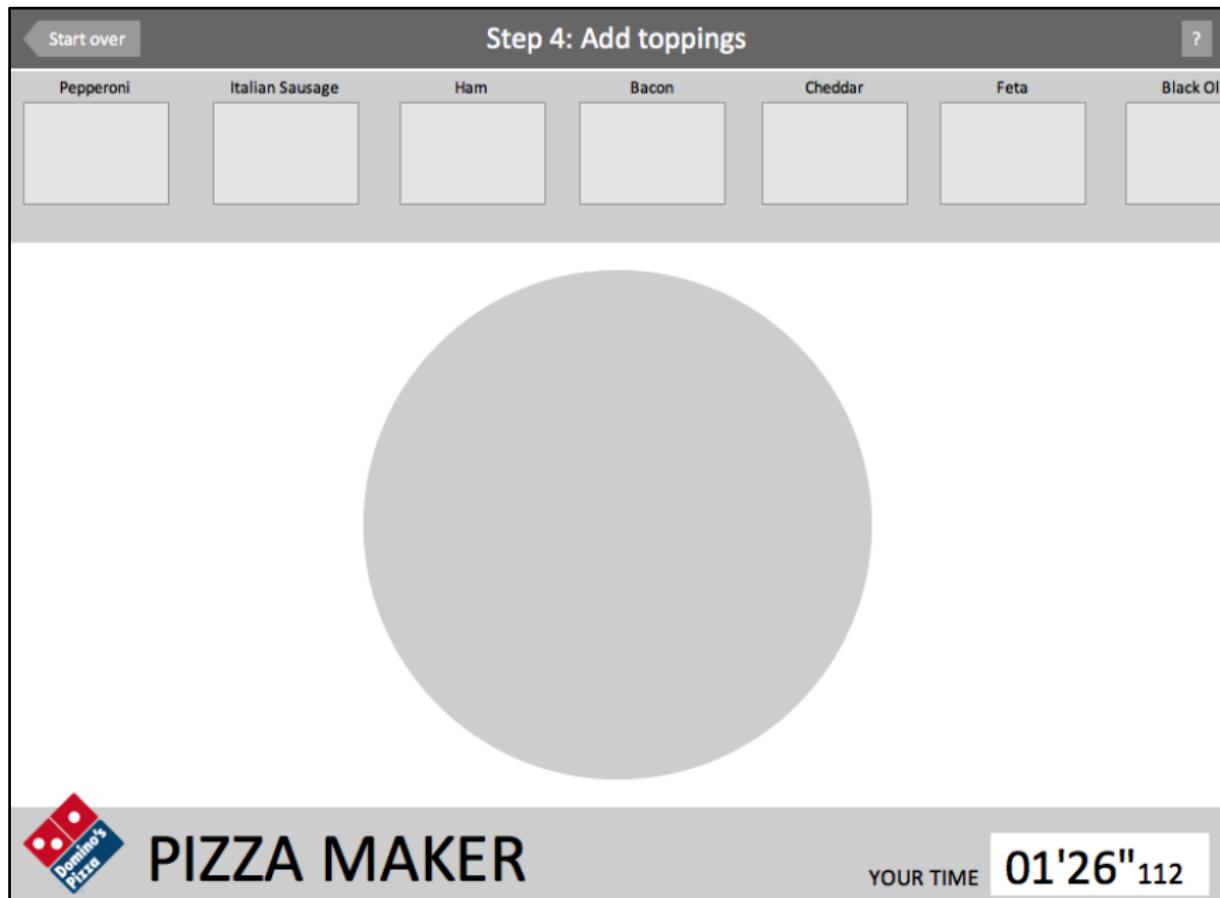
Wireframe é um rascunho ou desenho monocromático com o fim de representar páginas a serem acessadas pelos usuários, o processo de design de uma interface geralmente começa com a realização de sketches manuais ou mockups, como é ressaltado por Moran et al. (2018 apud Baulé, 2020).

De acordo com Huang et al., (2019 apud KREUCH, 2021), os sketches são desenhos simples que ajudam a transmitir a ideia geral da interface. Para Moran et al., (2018 apud Baulé, 2020), os mockups são modelos mais complexos que mostram a interface em detalhes.

O desenvolvimento de um aplicativo é um processo complexo que envolve muitas etapas diferentes, os autores Rogers; Sharp; Preece (2013 apud Baulé 2020), destacam que uma das etapas mais importantes é o design de sua interface, a interface é a maneira como o usuário interage com o aplicativo e, portanto, é essencial que seja fácil de usar e agradável.

Depois que os sketches ou mockups são criados, eles são testados para obter feedback, de acordo com (Baulé, 2020), o feedback é usado para melhorar a interface até que ela atenda às necessidades dos usuários, e uma vez que a interface está pronta, ela é implementada pelo desenvolvedor.

Figura 12 – Exemplo de WireFrame



Fonte: (Teixeira, 2015)

2.2.8 C#

C# (se pronuncia “C-Sharp”) é uma linguagem de programação moderna criada pela Microsoft, ela é uma linguagem de programação orientada a objetos, fortemente tipada e versátil. De acordo com a Microsoft (2023), o objetivo do C# é possibilitar que os desenvolvedores criem uma ampla variedade de sistemas, sejam eles jogos 2D e 3D, programas web e ou mobile e aplicativos para análise de dados. Para (Carlucci et al., 2021; Saade, 2010, o C# é familiar para programadores com experiência em linguagens como C, C++ e Java

A arquitetura .NET, na qual os programas C# são executados, inclui o Common Language Runtime (CLR) e uma extensa biblioteca de classes que suporta uma variedade de funcionalidades como é ressaltado pela Microsoft (2022).

Figura 13 - Primeiro Exemplo de Código em C#

```
1  using System;
2
3  namespace Calculadora
4  {
5      internal static class Program
6      {
7          static void Main()
8          {
9              Console.WriteLine("Qual operação deseja fazer:");
10             Console.WriteLine("1- Adição");
11             Console.WriteLine("2- Subtração");
12             Console.WriteLine("3- Multiplicação");
13             Console.WriteLine("4- Divisão \n");
14
15             int operacao;
16
17             // Loop até que uma operação válida seja digitada
18             while (true)
19             {
20                 string operacaoInput = Console.ReadLine();
21
22                 // Tenta converter a entrada para um número inteiro
23                 if (int.TryParse(operacaoInput, out operacao))
24                 {
25                     break; // Sai do loop se a conversão for bem-sucedida
26                 }
27
28                 Console.WriteLine("Entrada inválida para a operação. Certifique-se de digitar um número válido.");
29             }
30
31             Console.WriteLine("Digite o primeiro número: ");
32             int num1;
33
34             // Loop até que um primeiro número válido seja digitado
35             while (true)
36             {
37                 string num1Input = Console.ReadLine();
38
39                 // Tenta converter a entrada para um número inteiro
40                 if (int.TryParse(num1Input, out num1))
41                 {
42                     break; // Sai do loop se a conversão for bem-sucedida
43                 }
44
45                 Console.WriteLine("Entrada inválida para o primeiro número. Certifique-se de digitar um número válido.");
46             }
47
48             Console.WriteLine("Digite o segundo número: ");
49             int num2;
```

Fonte: Do Próprio Autor, 2023.

Na Figura 13, são apresentadas algumas características do C#, tais como:

- Métodos: o C# possui métodos nativos que manipulam e modificam dados, como textos, datas, números, entre outros.
- O método WriteLine grava os dados especificados, seguidos pelo terminador de linha, no fluxo de saída padrão. Nas linhas 8 a 13, assim como nas linhas 28, 31, 45 e 48, os dados passados como parâmetros são direcionados para a saída padrão do sistema, que é o console.

Figura 14 - Segundo Exemplo de Código em C#

```
46 }  
47  
48 Console.WriteLine("Digite o segundo número: ");  
49 int num2;  
50  
51 // Loop até que um segundo número válido seja digitado  
52 while (true)  
53 {  
54     string num2Input = Console.ReadLine();  
55  
56     // Tenta converter a entrada para um número inteiro  
57     if (int.TryParse(num2Input, out num2))  
58     {  
59         break; // Sai do loop se a conversão for bem-sucedida  
60     }  
61  
62     Console.WriteLine("Entrada inválida para o segundo número. Certifique-se de digitar um número válido.");  
63 }  
64  
65 int resultado = 0;  
66  
67 // Executa a operação selecionada com os números fornecidos  
68 switch (operacao)  
69 {  
70     case 1:  
71         resultado = Adicao(num1, num2);  
72         break;  
73     case 2:  
74         resultado = Subtracao(num1, num2);  
75         break;  
76     case 3:  
77         resultado = Multiplicacao(num1, num2);  
78         break;  
79     case 4:  
80         resultado = Divisao(num1, num2);  
81         break;  
82     default:  
83         Console.WriteLine("Número inválido, digite outro número.");  
84         return;  
85     }  
86  
87     Console.WriteLine("O resultado da operação é: {0}", resultado);  
88 }  
89  
90 // Função para realizar a adição  
91 public static int Adicao(int num1, int num2)  
92 {  
93     int resultado = num1 + num2;
```

Fonte: Do Próprio Autor, 2023.

Na figura 14, há alguns métodos adicionais, mencionados junto com a Figura 13, mas que apresentam outras estruturas. Na linha 52, encontra-se a estrutura While, que consiste em um bloco de repetição onde a condição fornecida entre parênteses é verificada a cada iteração, executando o conteúdo dentro das chaves.

Figura 15 - Terceiro Exemplo de Código em C#

```
88     }
89
90     // Função para realizar a adição
91     public static int Adicao(int num1, int num2)
92     {
93         int resultado = num1 + num2;
94         return resultado;
95     }
96
97     // Função para realizar a subtração
98     public static int Subtracao(int num1, int num2)
99     {
100        int resultado = num1 - num2;
101        return resultado;
102    }
103
104    // Função para realizar a multiplicação
105    public static int Multiplicacao(int num1, int num2)
106    {
107        int resultado = num1 * num2;
108        return resultado;
109    }
110
111    // Função para realizar a divisão
112    public static int Divisao(int num1, int num2)
113    {
114        // Verifica se o divisor é zero para evitar uma exceção de divisão por zero
115        if (num2 == 0)
116        {
117            Console.WriteLine("Erro: Divisão por zero não é permitida.");
118            return 0;
119        }
120
121        int resultado = num1 / num2;
122        return resultado;
123    }
124}
125}
```

Fonte: Do Próprio Autor, 2023.

Na figura 15, existem outras funções que também têm associado a elas o tipo de valor a ser retornado. Essas funções estão localizadas nas linhas 91, 98, 105 e 112 e são responsáveis por executar as operações matemáticas no exemplo da calculadora.

Figura 16 - Resultado da Codificação em C#

```
Qual operação deseja fazer:  
1- Adição  
2- Subtração  
3- Multiplicação  
4- Divisão  
  
1  
Digite o primeiro número:  
5  
Digite o segundo número:  
5  
O resultado da operação é: 10
```

Fonte: Do Próprio Autor, 2023.

2.2.9 Blender

O software Blender, um recurso fundamental para a modelagem tridimensional (3D), é destacado por Novelli (2015) como uma ferramenta de código aberto, que possibilita a criação, animação, texturização, renderização e integração com outras ferramentas de desenvolvimento.

A compreensão da criação e interação com objetos tridimensionais tornou-se fundamental com o crescimento da modelagem 3D. Sendo essencial tanto para profissionais quanto para estudantes envolvidos no desenvolvimento de projetos de comunicação e interação humano-computador, como é ressaltado por Nunes et al. (2015).

Figura 17 - Exemplo de Imagem No Blender



Fonte: (Blender, 2023).

2.2.10 Flutter

Desenvolvido e desenvolvido pelo Google, o Flutter é um framework de código aberto que faz uso da linguagem de programação Dart, também criada pelo Google, segundo (Venteu; Pinto, 2018), framework são conjuntos de códigos prontos com funções já implementadas para aplicações específicas. propósitos.

O Flutter se destaca por ser híbrido e versátil, para Budiu (2013 apud MARTINS et al., 2020), as aplicações híbridas tornam-se uma ferramenta ideal para redução de custos, devido à necessidade de construir o código apenas uma vez.

Figura 18 - Primeiro Exemplo de Código com Flutter

```
1 import 'package:flutter/material.dart';
2
3   Run | Debug | Profile
4 void main() {
5   runApp(CounterApp());
6 }
7
8 class CounterApp extends StatefulWidget {
9   @override
10  _CounterAppState createState() => _CounterAppState();
11 }
12
13 class _CounterAppState extends State<CounterApp> {
14   int _counter = 0; // Variável que armazena o valor do contador
15
16   void _incrementCounter() {
17     setState(() {
18       _counter++; // Incrementa o valor do contador em 1
19     });
20
21   void _decrementCounter() {
22     setState(() {
23       _counter--; // Decrementa o valor do contador em 1
24     });
25 }
```

Fonte: Do Próprio Autor, 2023.

Na Figura 18, está apresentado o primeiro trecho de exemplo de código utilizando o Flutter. Na linha 3, está o método Main, responsável por iniciar o código através do runApp, que recebe como parâmetro o Widget, representando a página em questão no código mencionado. Na linha 7, é criado uma classe que herda o StatelessWidget, classe essa que cria um Widget com mudança de estado, portanto, tudo o que estender essa classe poderá usar eventos que mudam o estado de uma tela. Na linha 12, há uma classe que herda o State do tipo CounterApp. Dessa forma, o que está dentro dessa segunda classe está dentro do ciclo de vida iniciado da classe CounterApp inicializados nas linhas 7 a 9.

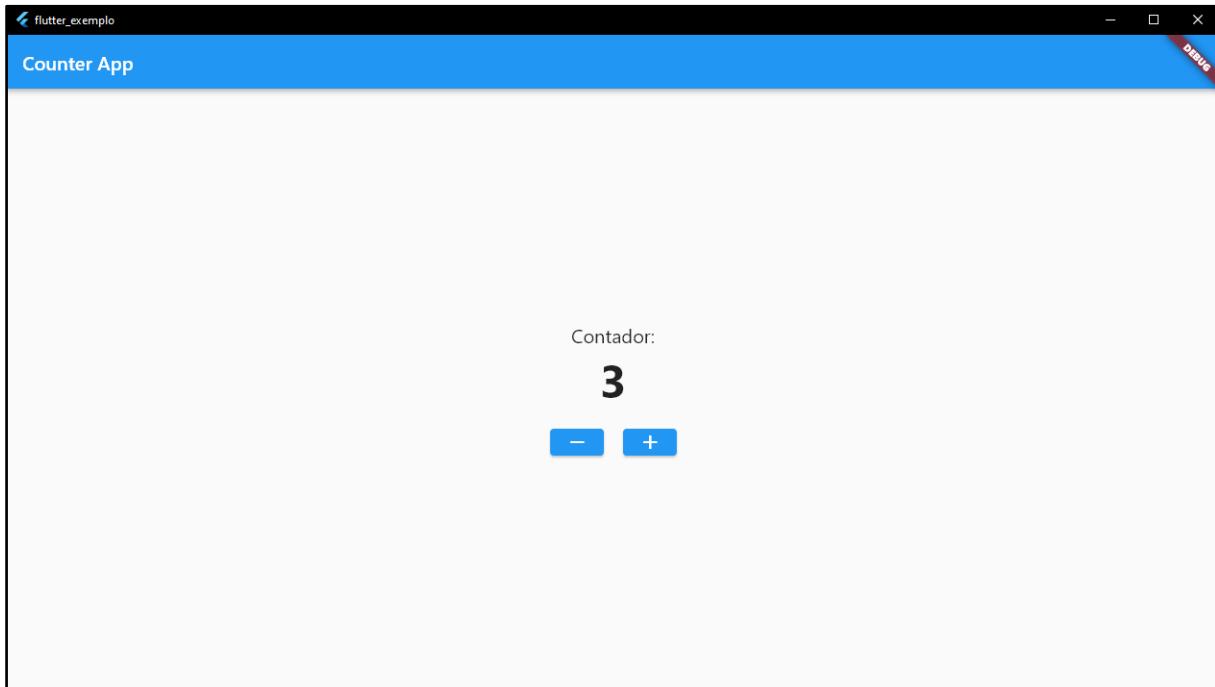
Figura 19 - Segundo Exemplo de Código com Flutter

```
26
27 @override
28 Widget build(BuildContext context) {
29   return MaterialApp(
30     home: Scaffold(
31       appBar: AppBar(
32         title: Text('Counter App'),
33       ), // AppBar
34       body: Center(
35         child: Column(
36           mainAxisAlignment: MainAxisAlignment.center,
37           children: <Widget>[
38             Text(
39               'Contador!', // Título do contador
40               style: TextStyle(fontSize: 20),
41             ), // Text
42             Text(
43               '_counter', // Valor atual do contador
44               style: TextStyle(fontSize: 48, fontWeight: FontWeight.bold),
45             ), // Text
46             SizedBox(height: 20),
47             Row(
48               mainAxisAlignment: MainAxisAlignment.center,
49               children: <Widget>[
50                 ElevatedButton(
51                   onPressed:
52                     _decrementCounter, // Chama a função _decrementCounter ao ser pressionado
53                     child: Icon(Icons.remove), // ícone de menos no botão
54                 ), // ElevatedButton
55                 SizedBox(width: 20),
56                 ElevatedButton(
57                   onPressed:
58                     _incrementCounter, // Chama a função _incrementCounter ao ser pressionado
59                     child: Icon(Icons.add), // ícone de mais no botão
60                 ), // ElevatedButton
61               ], // <Widget>[]
62             ), // Row
63           ], // <Widget>[]
64         ), // Column
65       ), // Center
66     ), // Scaffold
67   ); // MaterialApp
68 }
```

Fonte: Do Próprio Autor, 2023.

Na continuação do código registrado na imagem 18, temos um método build com o retorno do tipo Widget, na qual o Widget do Flutter dentro dele, no caso o MaterialApp inicializado na linha 29, será retornado após sua chamada. Essa imagem é um exemplo também de como funciona a formação em cascata de widgets no Flutter, que possibilita a construção de páginas da aplicação.

Figura 20 – Resultado da Codificação com Flutter



Fonte: Do Próprio Autor, 2023.

2.2.11 Banco de Dados Não Relacional

Com o avanço e a disseminação da internet e a grande demanda para armazenamento de dados, David Paniz (2016) observa que surgiram novas demandas em termos de desempenho, volume de dados e complexidade de implementação, acompanhadas por inovações relacionadas à persistência de dados.

Os bancos de dados NoSQL surgiram como uma resposta à necessidade de escalabilidade no armazenamento e processamento de grandes volumes de dados na Web, conforme abordado por Mauricio de Diana e Marco Aurélio Gerosa (2010). Os autores Mauricio de Diana e Marco Aurélio Gerosa (2010), ressaltam que apesar da escassez de recomendações e estudos comparativos sobre o uso desses bancos de dados, é necessário explorá-los.

No contexto Web, Vinícius Salles Garcia e Eder Carlos Salazar Sotto (2019) destacam que os bancos de dados NoSQL surgiram como alternativas para lidar com a escalabilidade.

Figura 21 - Exemplo de Banco Não Relacional: Firebase

The screenshot shows the Firebase Realtime Database interface. On the left, there's a sidebar with navigation links like 'Visão geral do projeto', 'Realtime Database', 'Criação', 'Liberar e monitorar', 'Analytics', 'Engajamento', 'Todos os produtos', and 'Personalize sua navegação.' Below these, it says 'Spark' and 'Sem custos financeiros US\$ 0/mês'. At the bottom of the sidebar, there's a 'Faz...' button. The main area is titled 'Realtime Database' and has tabs for 'Dados', 'Regras', 'Backups', 'Uso', and 'Extensões NOVO'. A banner at the top says 'Proteja os recursos do Realtime Database de abusos, como fraude de faturamento ou phishing' with a 'Configurar o App Check' button. The main content area shows a single node with the URL 'https://exemplo-951b1-default-rtbd.firebaseio.com/' followed by ':null'. At the bottom, it says 'Local do banco de dados: Estados Unidos (us-central1)'.

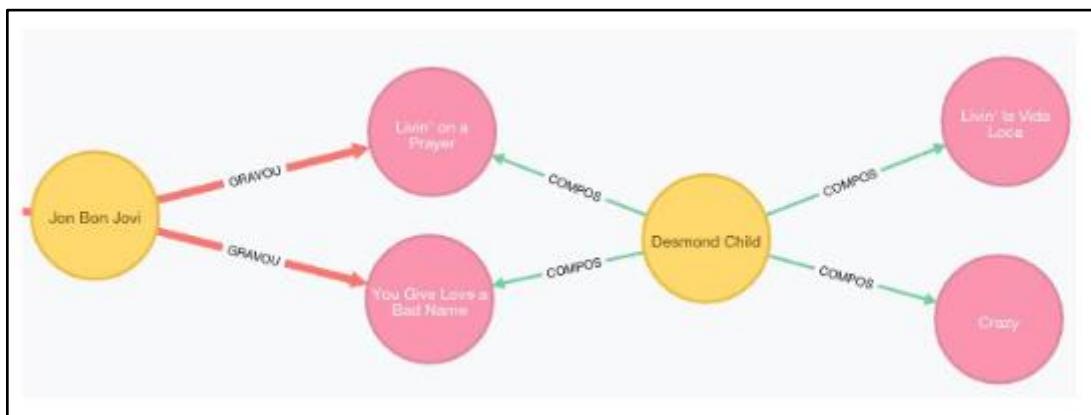
Fonte: Do Próprio Autor, 2023.

2.2.11.1 Banco de Dados Orientado a Grafo

A concepção de banco de dados de grafo tem suas origens na teoria dos grafos, um ramo da matemática dedicado à representação de relações entre objetos. A aplicação desse conceito em bancos de dados surgiu da necessidade de modelar e armazenar dados complexos, especialmente aqueles que envolvem relações e interconexões significativas. Tais bancos viabilizam o armazenamento de várias entidades e seus relacionamentos, representados por nodos e arestas, sendo o nodo associado a uma entidade e a aresta a um relacionamento.

Os autores SADALAGE e FOWLER (2015, apud Rabelo, 2017), destacam que as entidades são consideradas instâncias de objetos do aplicativo, exibindo propriedades próprias e os vínculos entre essas entidades são orientados por arestas, que podem conter propriedades específicas. A estrutura do grafo simplifica a identificação de padrões entre os nodos, facilitando o armazenamento eficiente de dados e permitindo diversas interpretações com base nos relacionamentos estabelecidos.

Figura 22 - Exemplo de Grafo



Fonte: (David Paniz, 2016)

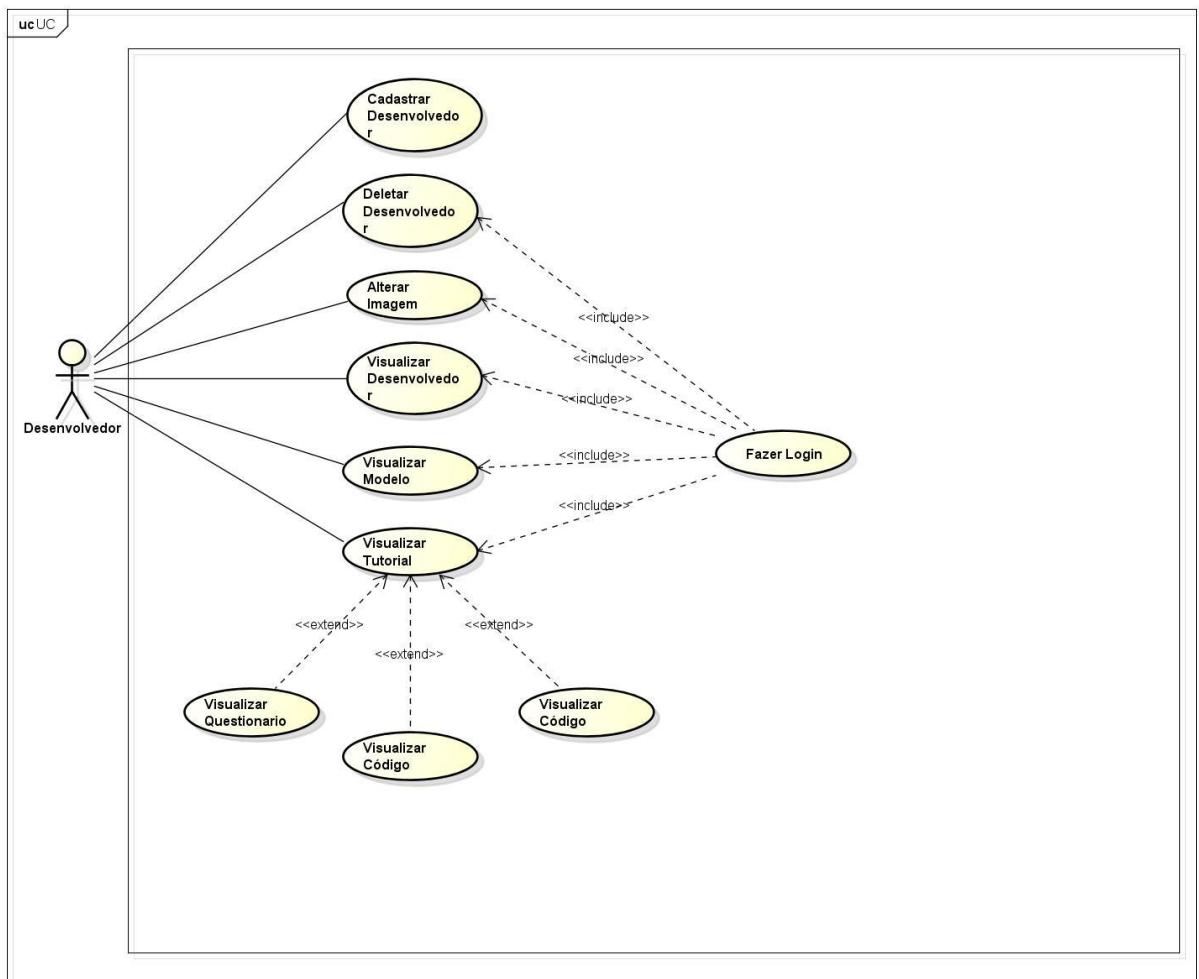
3 DESENVOLVIMENTO

A seção demonstrará o processo de criação da plataforma AR-PIN, tomando como base os métodos de documentação UML, contendo as imagens das telas do sistema e descrições dos diagramas quando necessário.

3.1 Diagrama de Casos de Uso AR-PIN

Neste tópico, segue-se a figura do diagrama de Caso de Uso do sistema, onde é possível analisá-la, a figura demonstra as interações dos atores com o sistema, incluindo as ações que eles podem realizar.

Figura 23 - Diagrama de Caso de Uso AR-PIN



Fonte: Do Próprio Autor, 2023.

3.1.1 Documentação dos Casos de Uso AR-PIN

Baseando-se na revisão bibliográfica a respeito de diagramas UML, este módulo contém a documentação dos Casos de Uso.

Requisitos Funcionais do Desenvolvedor:

RF01 – O desenvolvedor poderá criar uma conta no sistema.

RF02 – O desenvolvedor poderá fazer login na sua conta.

RF03 – O desenvolvedor poderá atualizar a foto de perfil da sua conta

RF04 – O desenvolvedor poderá deletar a sua conta.

RF05 – O desenvolvedor poderá visualizar exemplos de código de Arduino

RF06 – O aplicativo deverá fornecer uma funcionalidade de simulação de circuito que permita que os desenvolvedores utilizem a câmera do dispositivo móvel para visualizar e interagir com circuitos virtuais em tempo real.

RF07 – O aplicativo deverá disponibilizar tutoriais que abordem os conceitos de eletrônica e de Arduino UNO R3

RF08 – O aplicativo deverá disponibilizar materiais didáticos a respeito de eletrônica e de Arduino UNO R3.

RF09 – O aplicativo deverá disponibilizar questionários sobre os conteúdos apresentados.

Requisitos Não Funcionais:

RNF01 – A interface do aplicativo deve ser intuitiva e de fácil utilização, proporcionando uma experiência agradável para os desenvolvedores.

RNF02 – O aplicativo deve ser compatível com diferentes dispositivos móveis, como smartphones e tablets, e ser executado em diferentes sistemas operacionais, como Android.

RNF03 – O aplicativo deve implementar medidas de segurança para proteger as informações dos desenvolvedores, incluindo dados de login e informações de conta.

RNF04 – O aplicativo deve ser confiável, evitando falhas frequentes ou travamentos inesperados. Além disso, os tutoriais e os circuitos virtuais devem ser exibidos corretamente e sem erros.

RNF06 – O aplicativo deve ser projetado para operar com altas demandas. Ele deve ser capaz de gerenciar eficientemente o armazenamento de dados e garantir um desempenho consistente, mesmo com um aumento na demanda.

RNF07 – O código-Fonte e a arquitetura do aplicativo devem ser projetados de forma modular e organizada, facilitando a manutenção e a introdução de novos recursos no futuro.

RNF08 – Os questionários disponibilizados devem fornecer feedback imediato aos desenvolvedores, indicando o número de acertos.

Quadros descritivos dos casos de uso

Documentação dos casos de uso do Desenvolvedor

Quadro 1 - Descrição do caso de uso “Realizar Cadastro” (desenvolvedor)

Nome do Caso de Uso	Cadastrar Desenvolvedor
Autor Principal	Desenvolvedor
Resumo	Este caso de uso descreve o processo para a realização do cadastro do Desenvolvedor
Cenário Principal	
Ações do Ator	Ações do Sistema
	1. Solicitar dados do desenvolvedor.
2. Inserir os dados requeridos.	
	3. Verificar dados.
	4. Realizar cadastro.
Cenário de Exceção - Dados inválidos	
Ações do Ator	Ações do Sistema
	1. Comunicar que os dados estão incorretos.
	2. Recusar requisição.

Fonte: Do Próprio Autor, 2023.

Quadro 2 - Descrição do caso de uso “Realizar Login” (desenvolvedor)

Nome do Caso de Uso	Fazer Login
Autor Principal	Desenvolvedor
Resumo	Este caso de uso descreve o processo para a realização do login do Desenvolvedor
Pré-condições	O desenvolvedor já deve estar cadastrado no sistema
Cenário Principal	
Ações do Ator	Ações do Sistema
	1. Solicitar o usuário e senha.
2. Inserir os dados requeridos.	
	3. Verificar dados.
	4. Realizar login.
Cenário de Exceção - Dados inválidos	
Ações do Ator	Ações do Sistema
	1. Comunicar que os dados estão incorretos.
	2. Recusar requisição.

Fonte: Do Próprio Autor, 2023.

Quadro 3 - Descrição do Caso de Uso “Visualizar Tutorial”

Nome do Caso de Uso	Visualizar Tutorial
Autor Principal	Desenvolvedor
Resumo	Este caso de uso descreve o processo da visualização dos tutoriais
Pré-condições	O desenvolvedor deve ter realizado o login previamente.
Cenário Principal	
Ações do Ator	Ações do Sistema
	1. Apresentar os tutoriais
2. Selecionar o tutorial	
	3. Executar o tutorial
Cenário de Exceção – Pular etapas	
Ações do Ator	Ações do Sistema
1.Pular etapa desejada.	
	2.Pula a etapa selecionada.

Fonte: Do Próprio Autor, 2023.

Quadro 4 - Descrição do caso de uso "Visualizar Modelo 3D"

Nome do Caso de Uso	Visualizar Modelo 3D
Autor Principal	Desenvolvedor
Resumo	Este caso de uso descreve o processo de visualização de um modelo 3D de componentes relacionados à Eletrônica.
Pré-condições	O Desenvolvedor deve ter realizado o login previamente.
Cenário Principal	
Ações do Ator	Ações do Sistema
1. Baixar modelos 3D.	
2. Selecionar o modelo 3D.	
	3. Apresenta o modelo 3D em RA.
4. Visualiza o modelo 3D em RA.	
	5. Finaliza a visualização do projeto.
Cenário de Exceção – Pular etapa	
Ações do Ator	Ações do Sistema

Fonte: Do Próprio Autor, 2023.

Quadro 5 - Descrição do Caso de Uso “Visualizar Questionário”

Nome do Caso de Uso	Realizar Questionário
Autor Principal	Desenvolvedor
Resumo	Este caso de uso descreve o processo da visualização e realização dos questionários.
Pré-condições	O desenvolvedor deve ter realizado o login previamente e selecionado a guia de tutoriais.
Cenário Principal	
Ações do Ator	Ações do Sistema
1. Iniciar a lição.	
	2. Verifica a associação de um questionário para a lição.
	3. Retorna questionário.
4. Inicia questionário.	
	5. Retorna as questões.
6. Insere as alternativas.	
	7. Valida as respostas com gabarito.
	8. Retorna erros e acertos.
9. Progride tutorial.	
Cenário de Exceção – Respostas não-aprovadas	
Ações do Ator	Ações do Sistema
	1. Retorna não-aprovação.
2. Encerra a lição e interrompe progresso do tutorial.	

Fonte: Do Próprio Autor, 2023.

Quadro 6 - Descrição do caso de uso "Visualizar Perfil" (desenvolvedor)

Nome do Caso de Uso	Visualizar Conta
Autor Principal	Desenvolvedor
Resumo	Este caso de uso descreve o processo de visualização da conta do desenvolvedor
Pré-condições	O Desenvolvedor deve ter realizado o login previamente.
Cenário Principal	
Ações do Ator	Ações do Sistema
	1. Apresentar a conta do desenvolvedor.
2. Selecionar a aba perfil.	
	3. Apresentar o perfil.
Cenário de Exceção - Dados inválidos	
Ações do Ator	Ações do Sistema

Fonte: Do Próprio Autor, 2023.

Quadro 7 - Descrição do caso de uso "Editar Foto de Perfil" (desenvolvedor)

Nome do Caso de Uso	Editar Perfil
Autor Principal	Desenvolvedor
Resumo	Este caso de uso descreve o processo de edição do perfil do desenvolvedor
Pré-condições	O desenvolvedor deve ter realizado o login previamente
Cenário Principal	
Ações do Ator	Ações do Sistema
	1. Apresentar os dados da conta.
2. Selecionar a foto de perfil.	
	3. Apresentar a tela de edição.
4. Editar a foto da conta.	
5. Salvar os dados novamente.	
	6. Atualizar a conta.
Cenário de Exceção - Dados inválidos	
Ações do Ator	Ações do Sistema
	1. Notificar caso ocorra algum erro em relação a edição dos dados.
	2. Não atualizar o perfil.

Fonte: Do Próprio Autor, 2023.

Quadro 8 - Descrição do caso de uso "Deletar Perfil" (desenvolvedor)

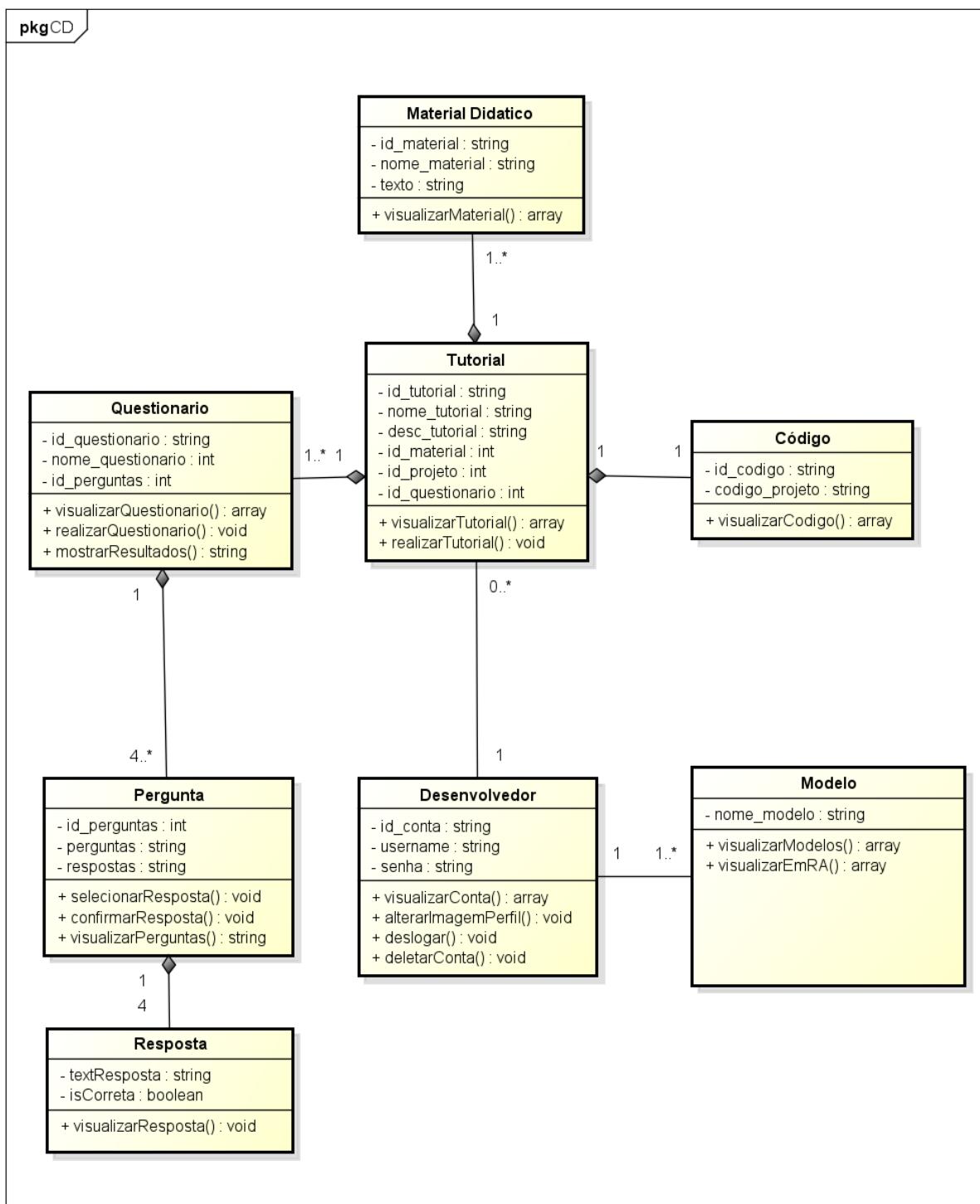
Nome do Caso de Uso	Deletar Perfil
Autor Principal	Desenvolvedor
Resumo	Este caso de uso descreve o processo de exclusão do perfil do desenvolvedor
Pré-condições	O desenvolvedor deve ter realizado o login previamente.
Cenário Principal	
Ações do Ator	Ações do Sistema
	1. Apresentar os dados da conta.
2. Selecionar a opção de exclusão de conta.	
	3. Apresentar aviso de exclusão.
4. Confirmar exclusão da conta.	
	5. Excluir conta.
Cenário de Exceção - Dados inválidos	
Ações do Ator	Ações do Sistema
	1 Notificar caso ocorra algum erro em relação a exclusão da conta.
	2. Não excluir a conta.

Fonte: Do Próprio Autor, 2023.

3.2 Diagrama de Classe AR-PIN

Neste tópico, segue a figura do diagrama de Classe do sistema, por meio desse diagrama podemos compreender melhor como as classes dos sistemas se relacionam entre si.

Figura 24 - Diagrama de Classe AR-PIN

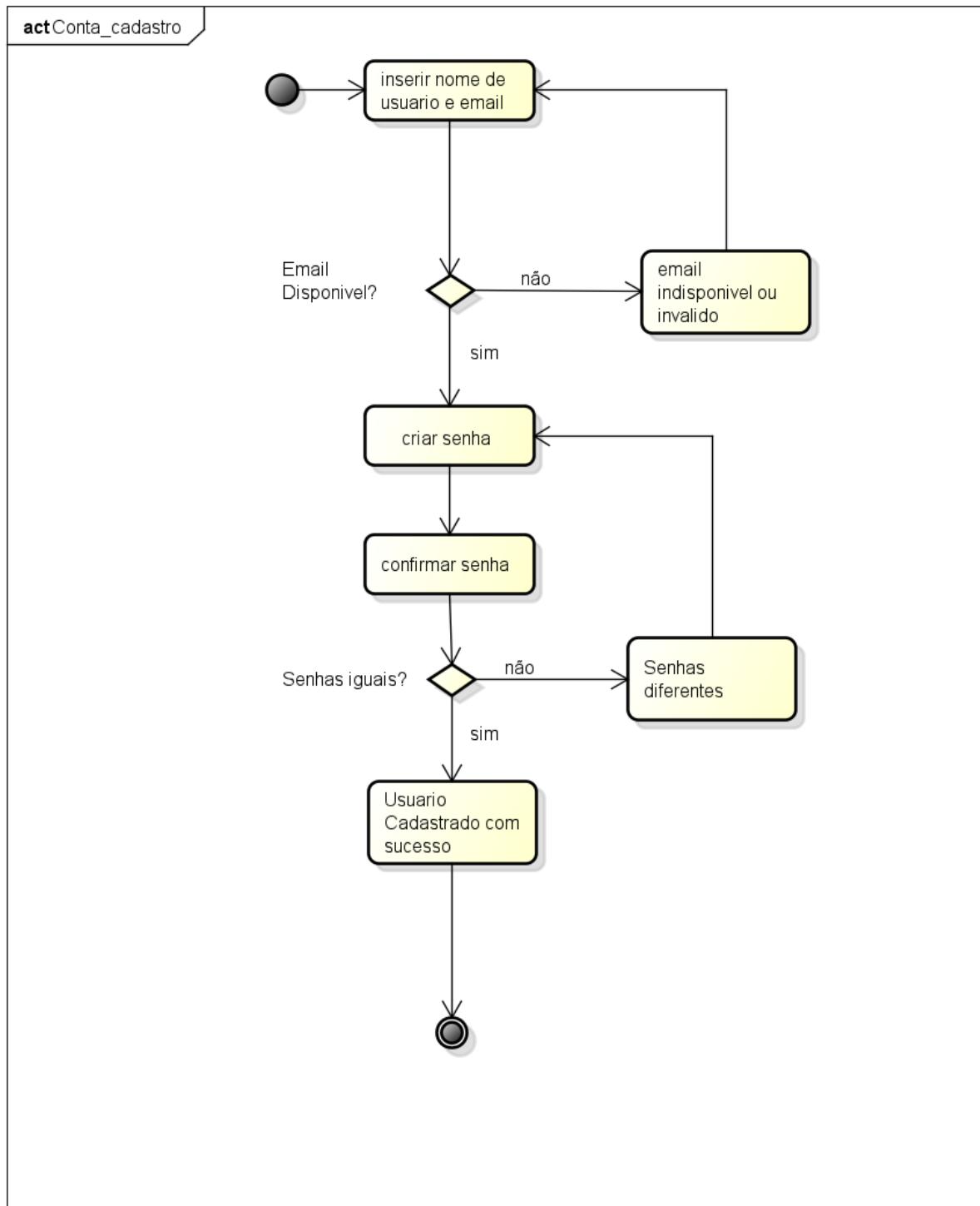


Fonte: Do Próprio Autor, 2023.

3.3 Diagrama de Atividade AR-PIN

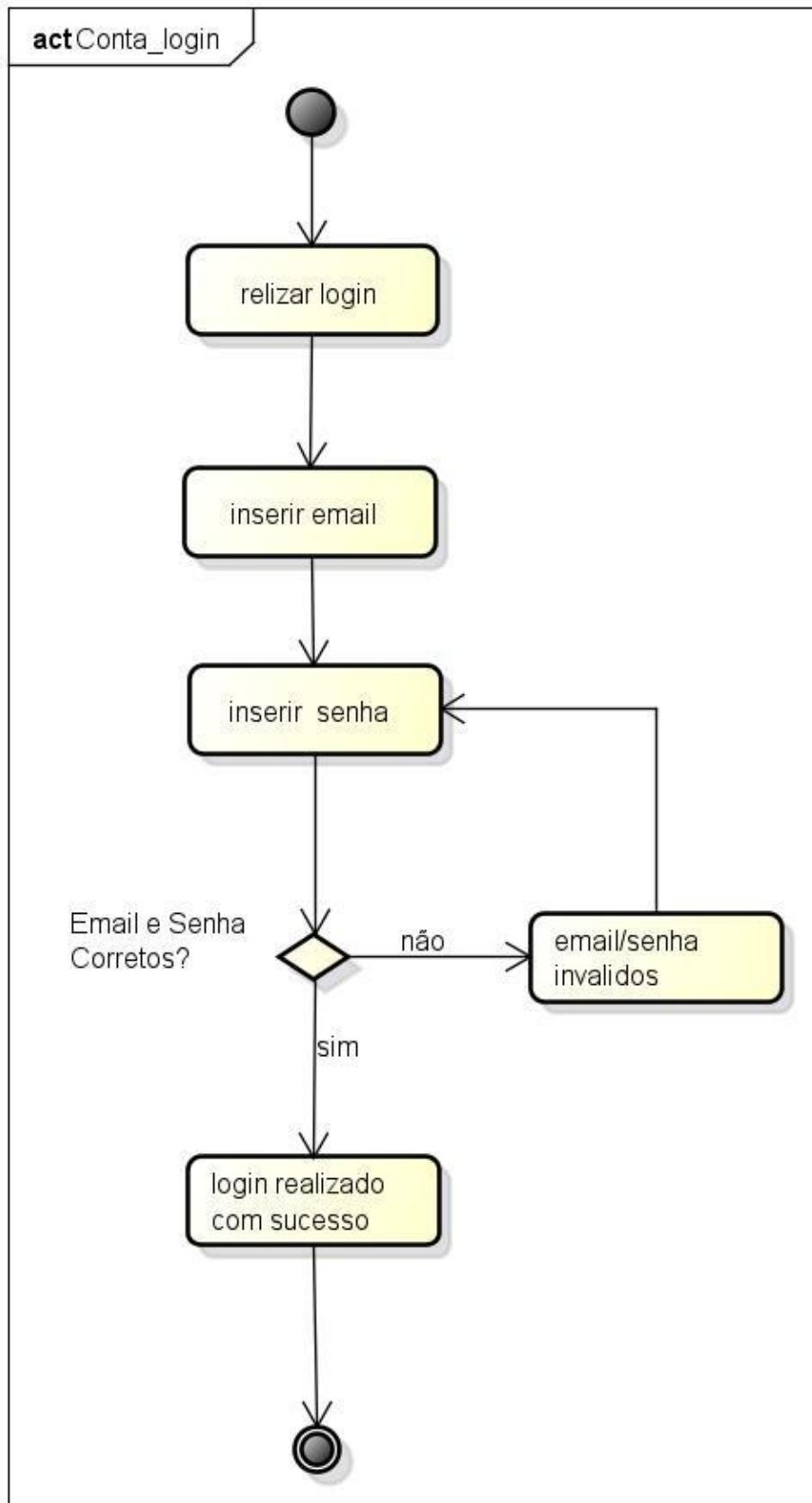
Essa seção contém os diagramas de Atividade do trabalho. Com os Diagramas de Atividade do sistema, é possível descrever seu funcionamento, demonstrando os detalhes das etapas sequenciais de cada processo

Figura 25 - Diagrama de Atividade Realizar Cadastro



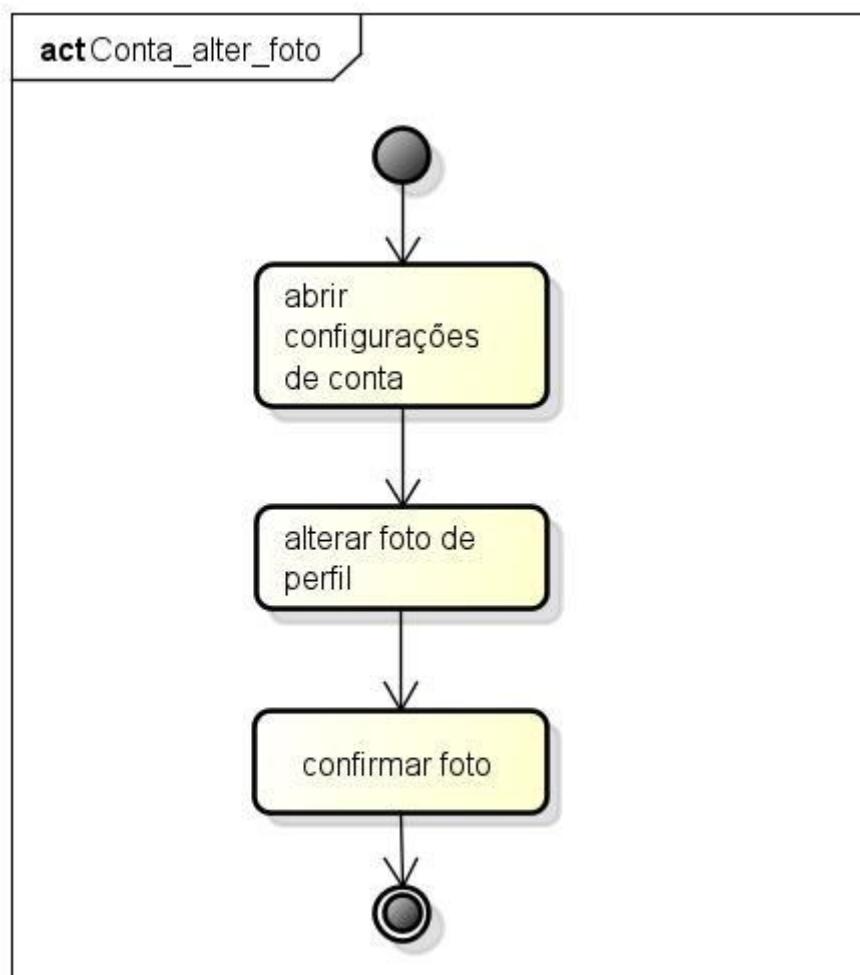
Fonte: Do Próprio Autor, 2023.

Figura 26 - Diagrama de Atividade Realizar Login



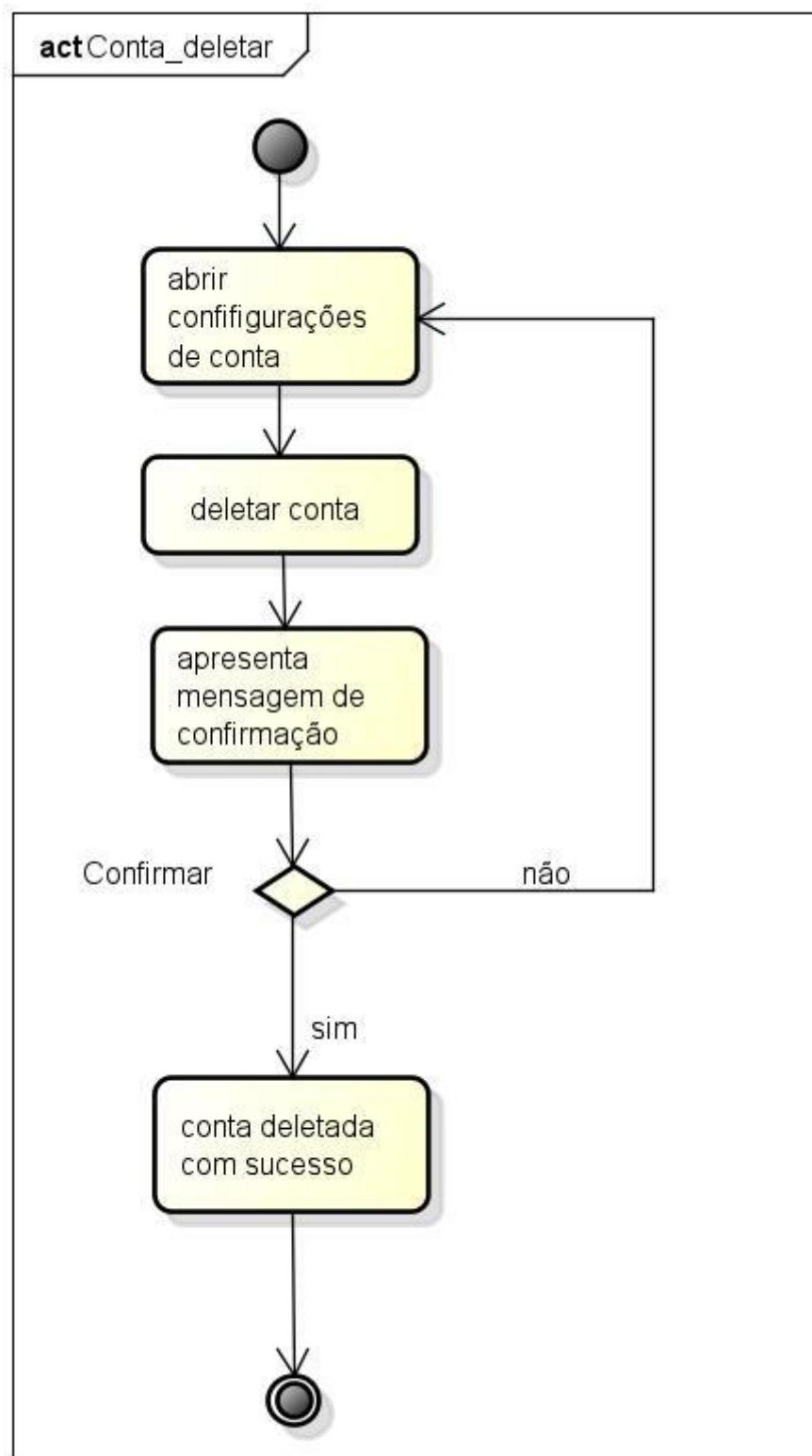
Fonte: Do Próprio Autor, 2023.

Figura 27 - Diagrama de Atividade Alterar Foto



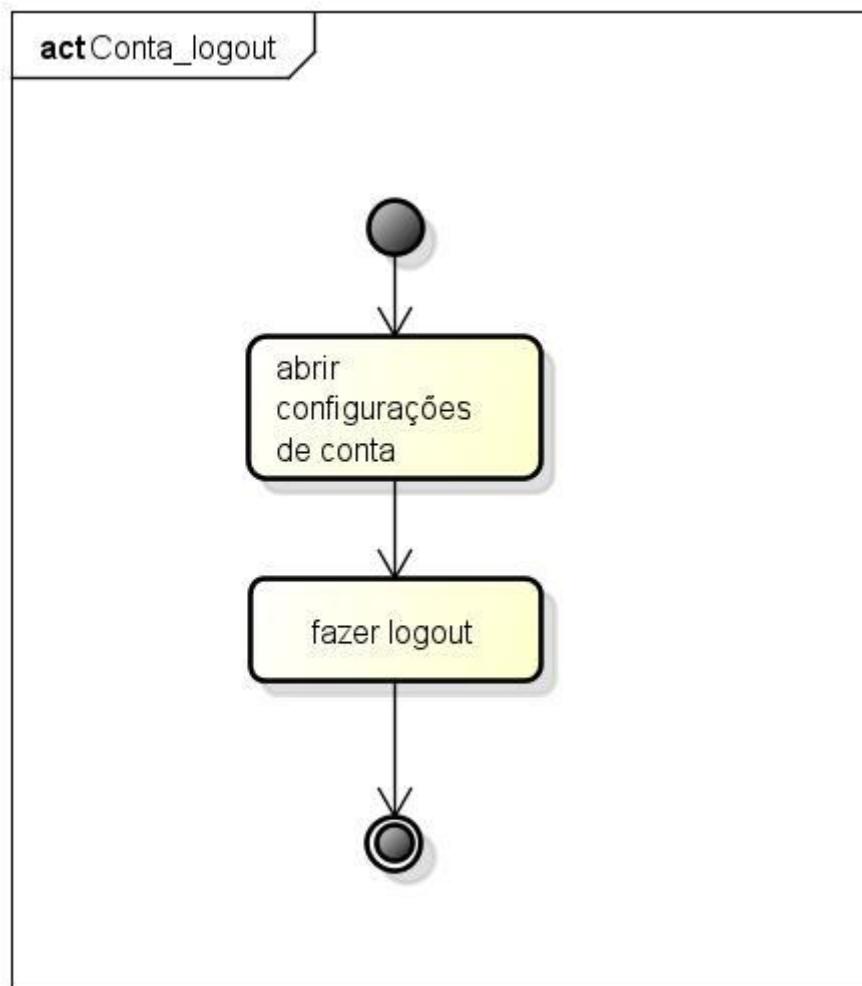
Fonte: Do Próprio Autor, 2023.

Figura 28 - Diagrama de Atividade Deletar Conta



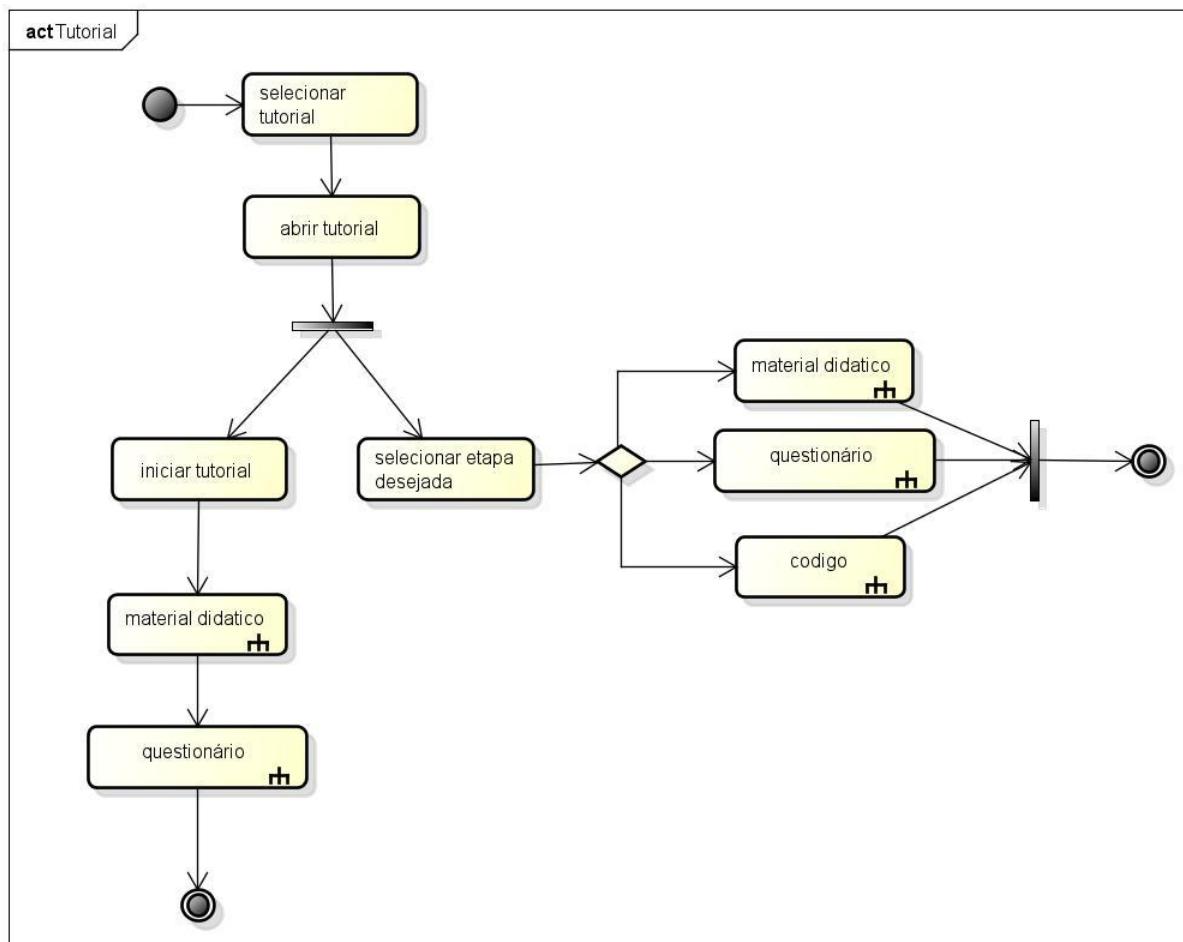
Fonte: Do Próprio Autor, 2023.

Figura 29 - Diagrama de Atividade Realizar LogOut



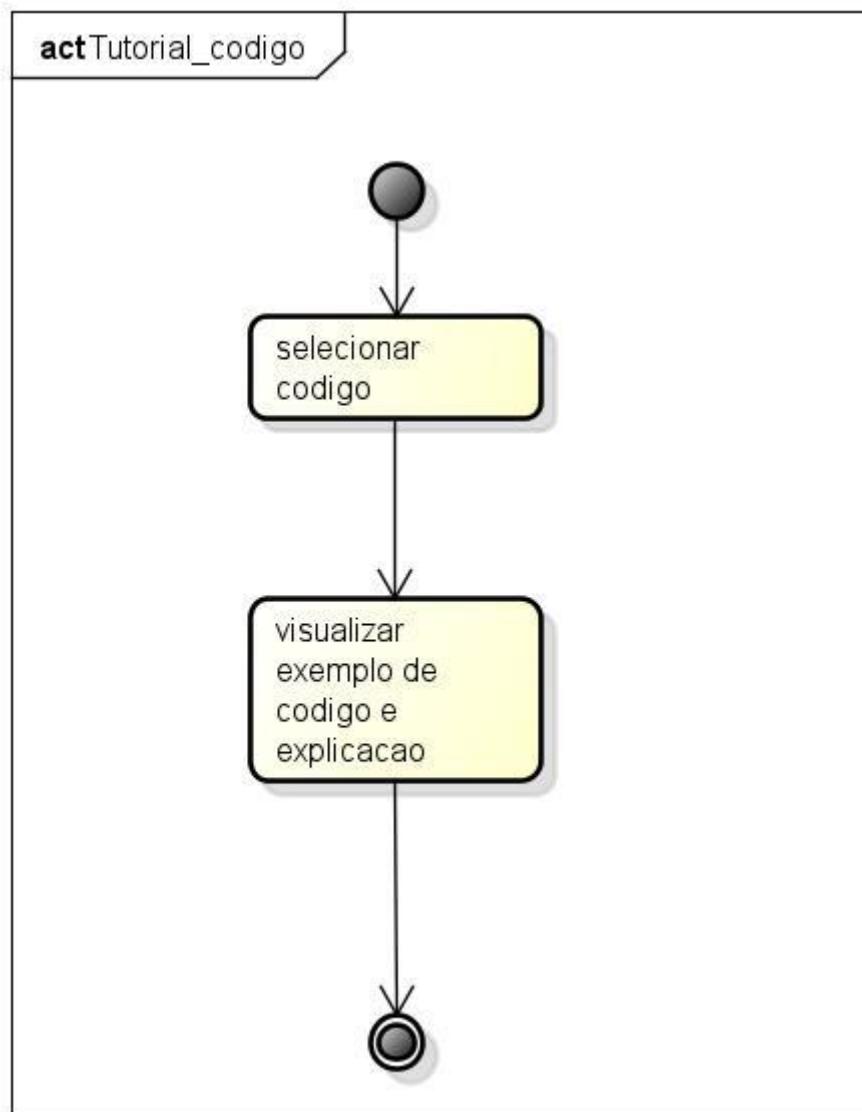
Fonte: Do Próprio Autor, 2023.

Figura 30 - Diagrama de Atividade Selecionar Tutorial



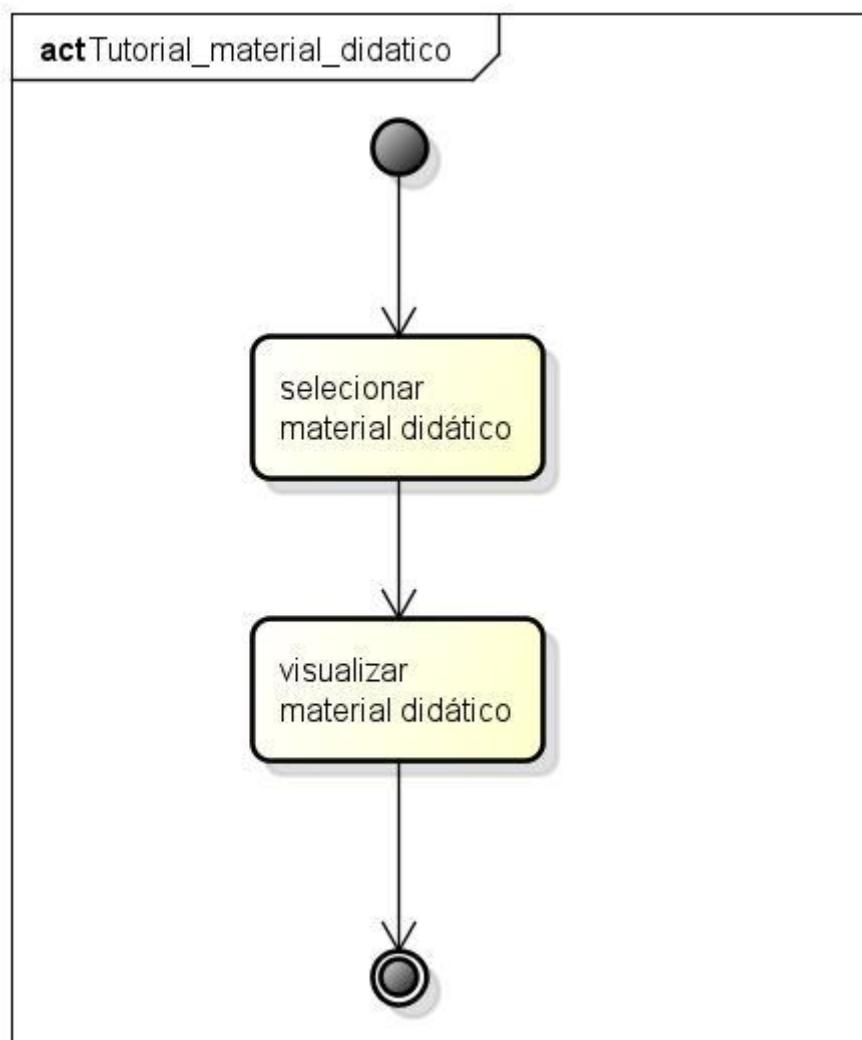
Fonte: Do Próprio Autor, 2023.

Figura 31 - Diagrama de Atividade Visualizar Exemplo de Código



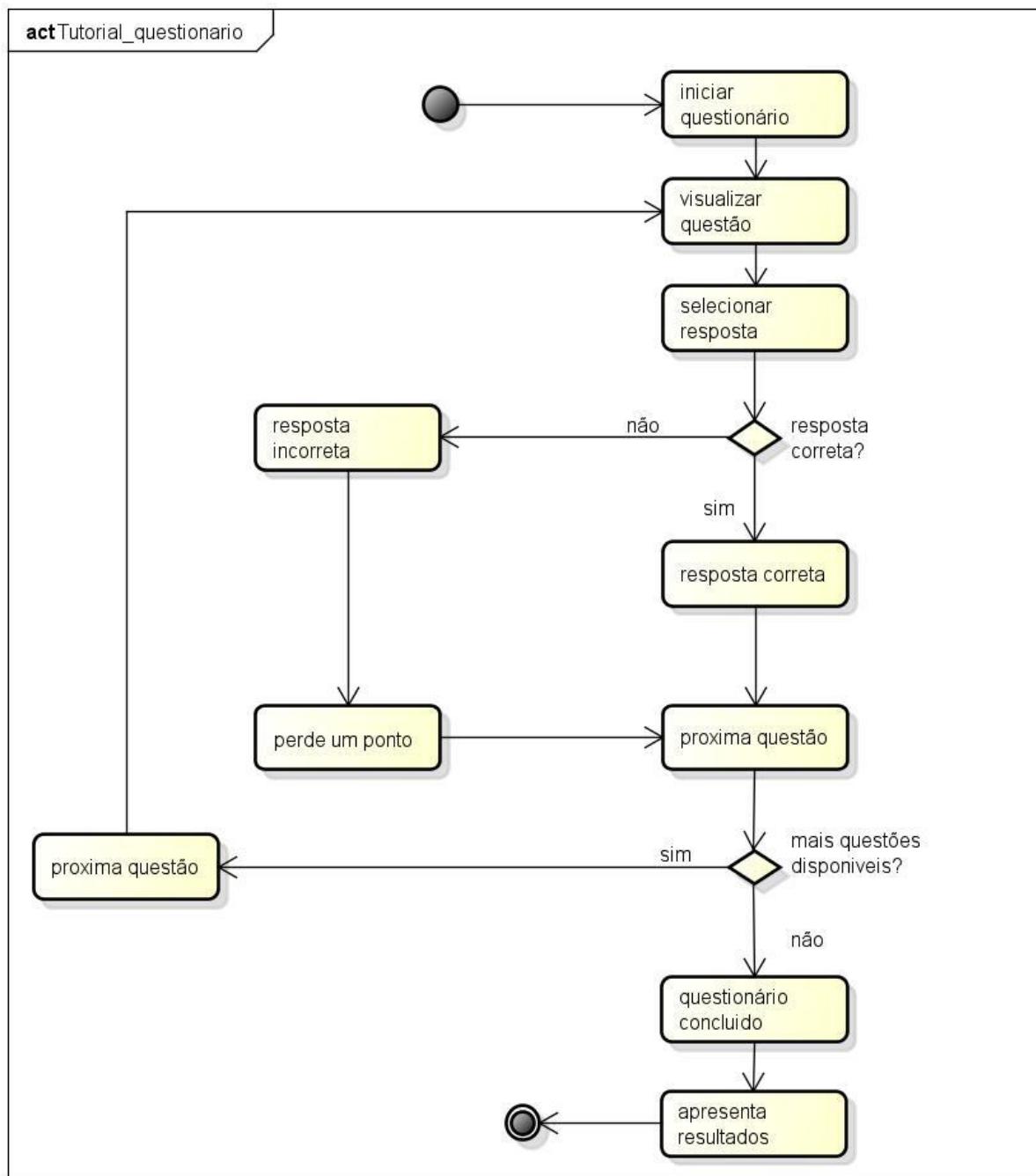
Fonte: Do Próprio Autor, 2023.

Figura 32 - Diagrama de Atividade Visualizar Material Didático



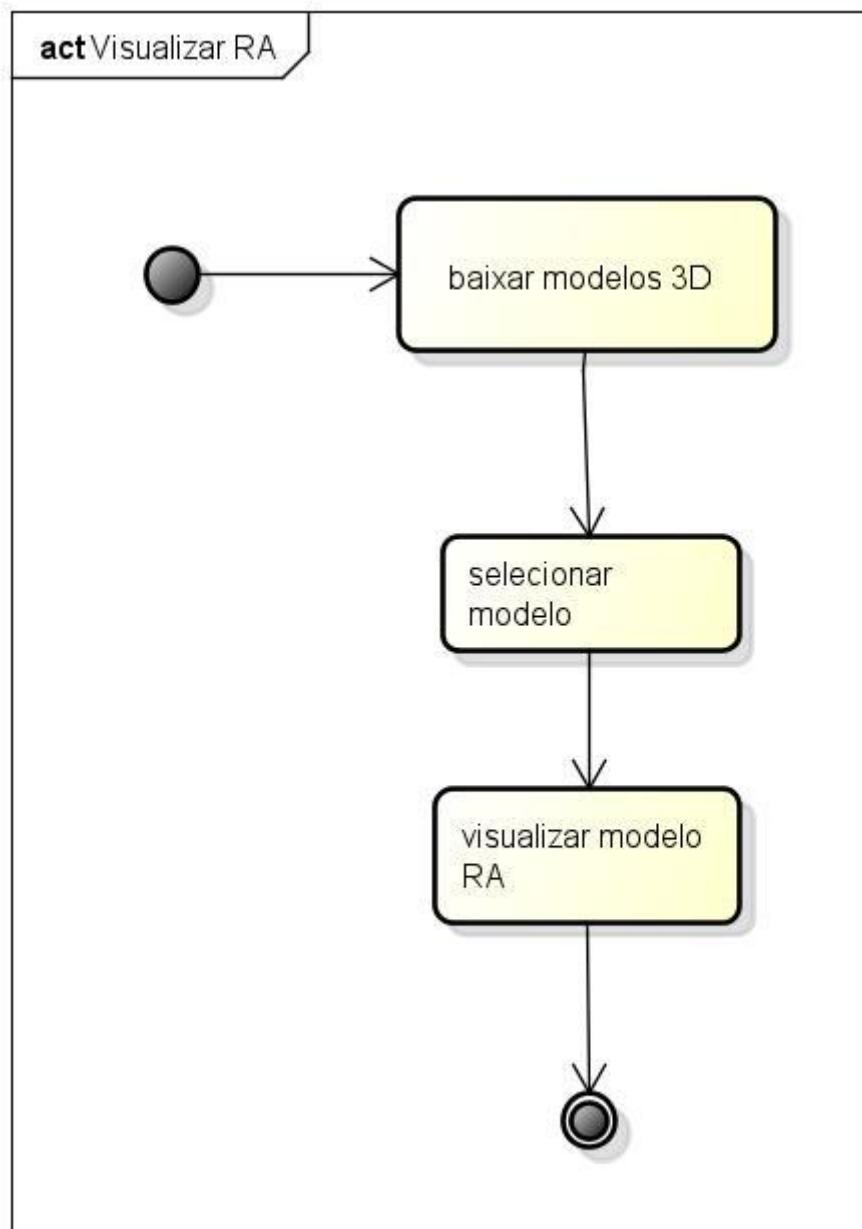
Fonte: Do Próprio Autor, 2023.

Figura 33 - Diagrama de Atividade Realizar Questionário



Fonte: Do Próprio Autor, 2023.

Figura 34 - Diagrama de Atividade Visualizar Modelo 3D em RA

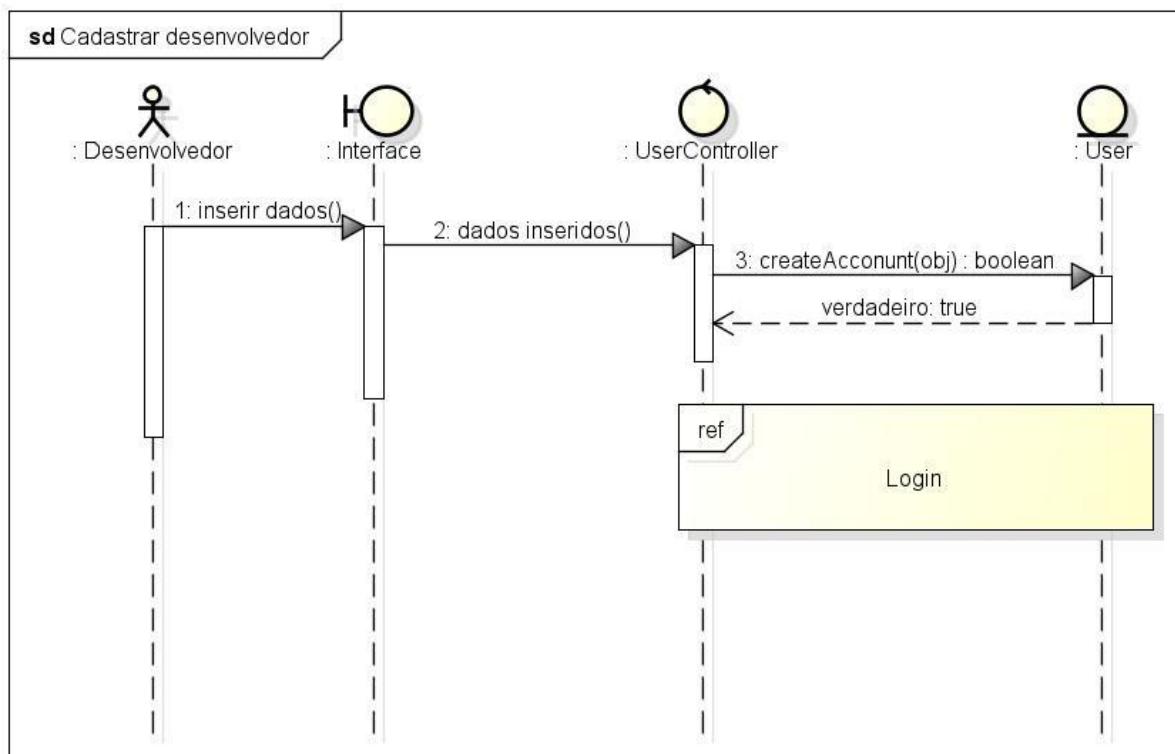


Fonte: Do Próprio Autor, 2023.

3.4 Diagramas de Sequência AR-PIN

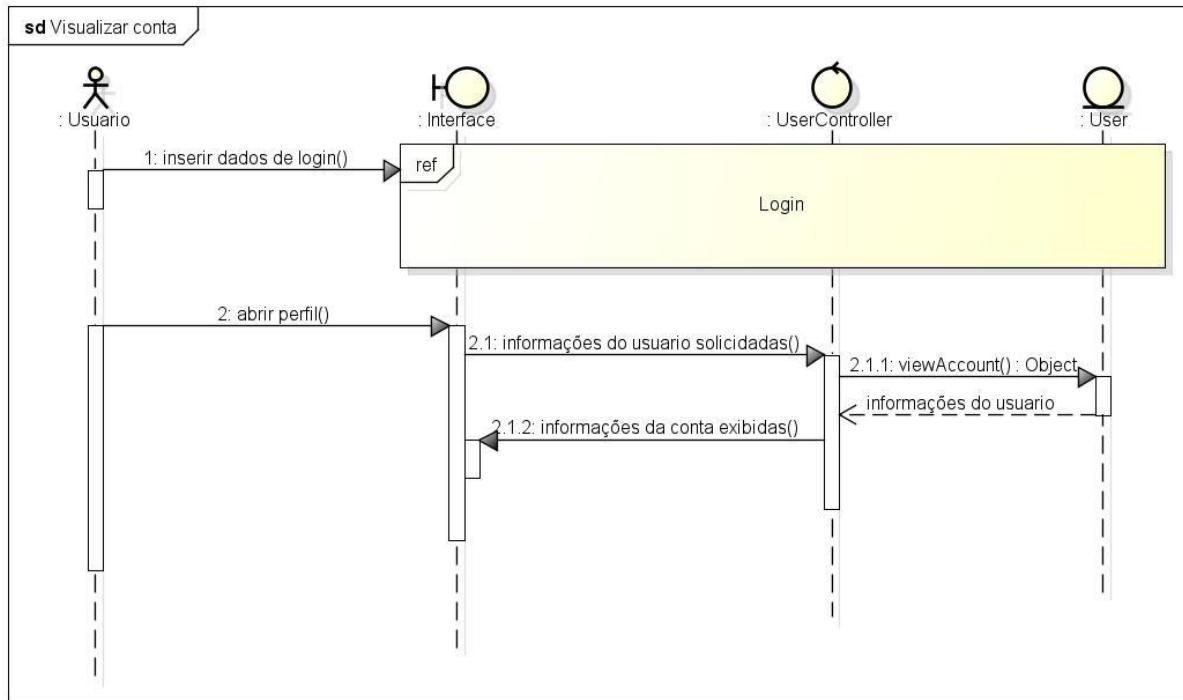
Essa seção contém os Diagramas de Sequência do sistema. Por meio destes diagramas, é possível capturar o comportamento de um único cenário, seguindo a ordem temporal e as respectivas interações na execução de uma determinada função.

Figura 35 - Diagrama de Sequência Cadastrar Desenvolvedor



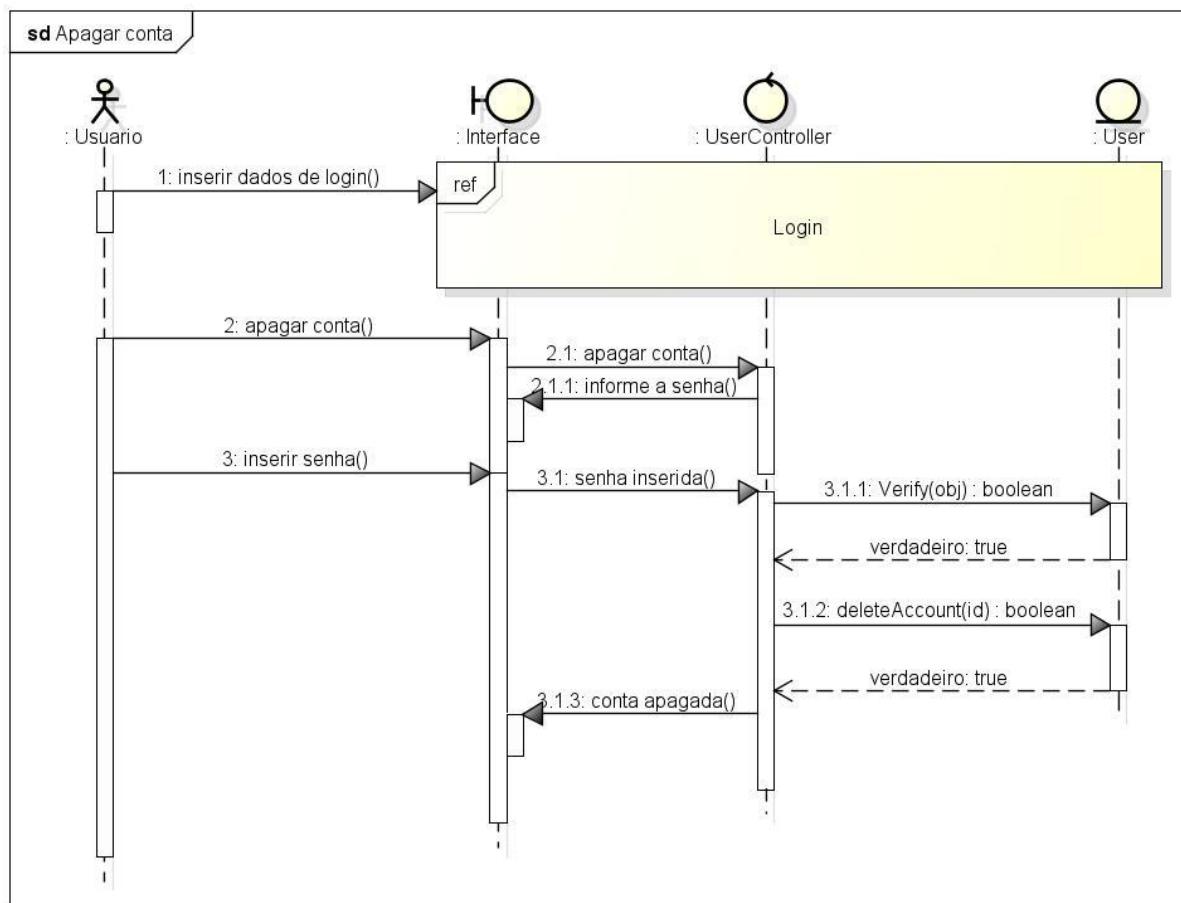
Fonte: Do Próprio Autor, 2023.

Figura 36 - Diagrama de Sequência Visualizar Conta



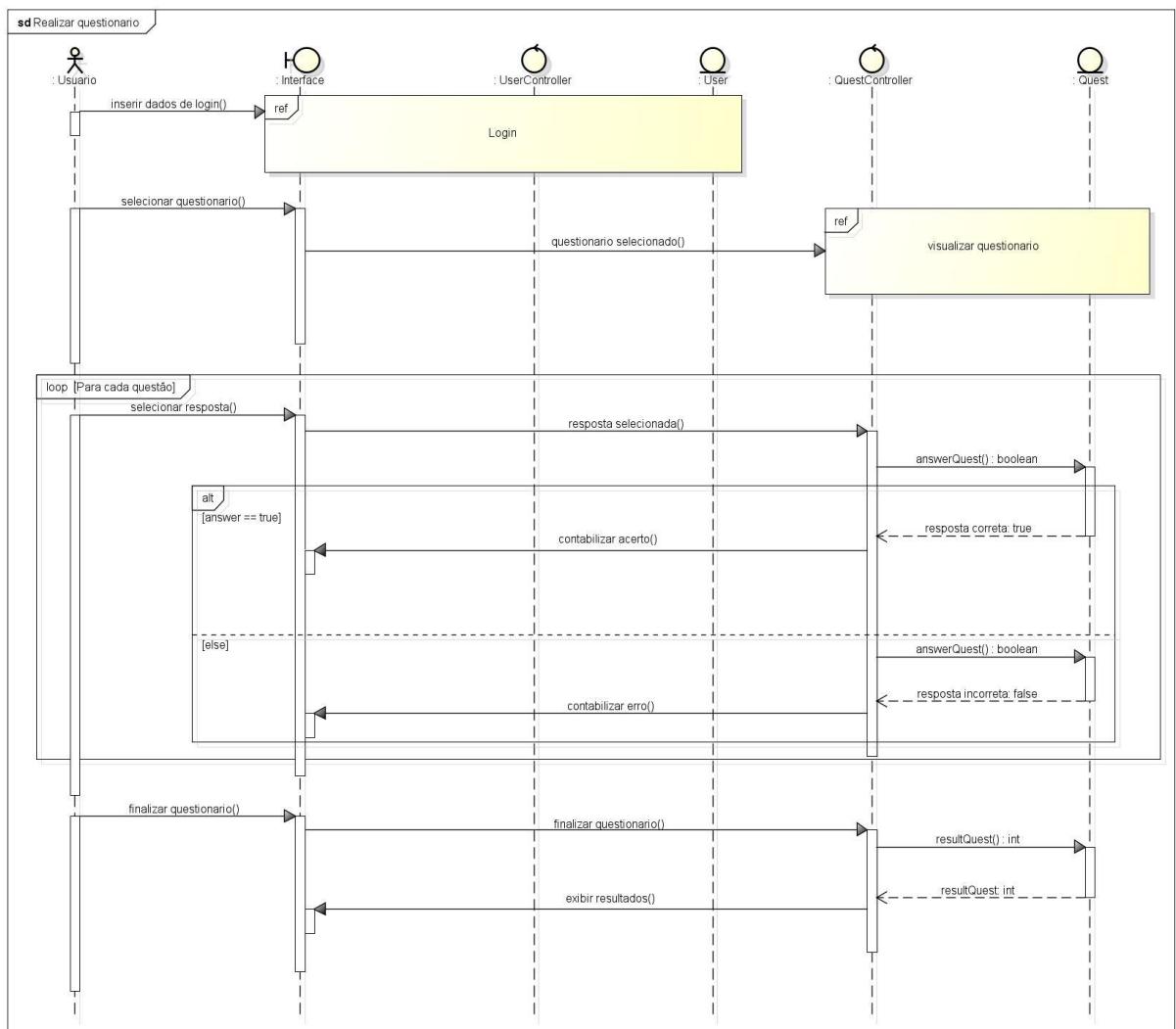
Fonte: Do Próprio Autor, 2023.

Figura 37 - Diagrama de Sequência Deletar Conta



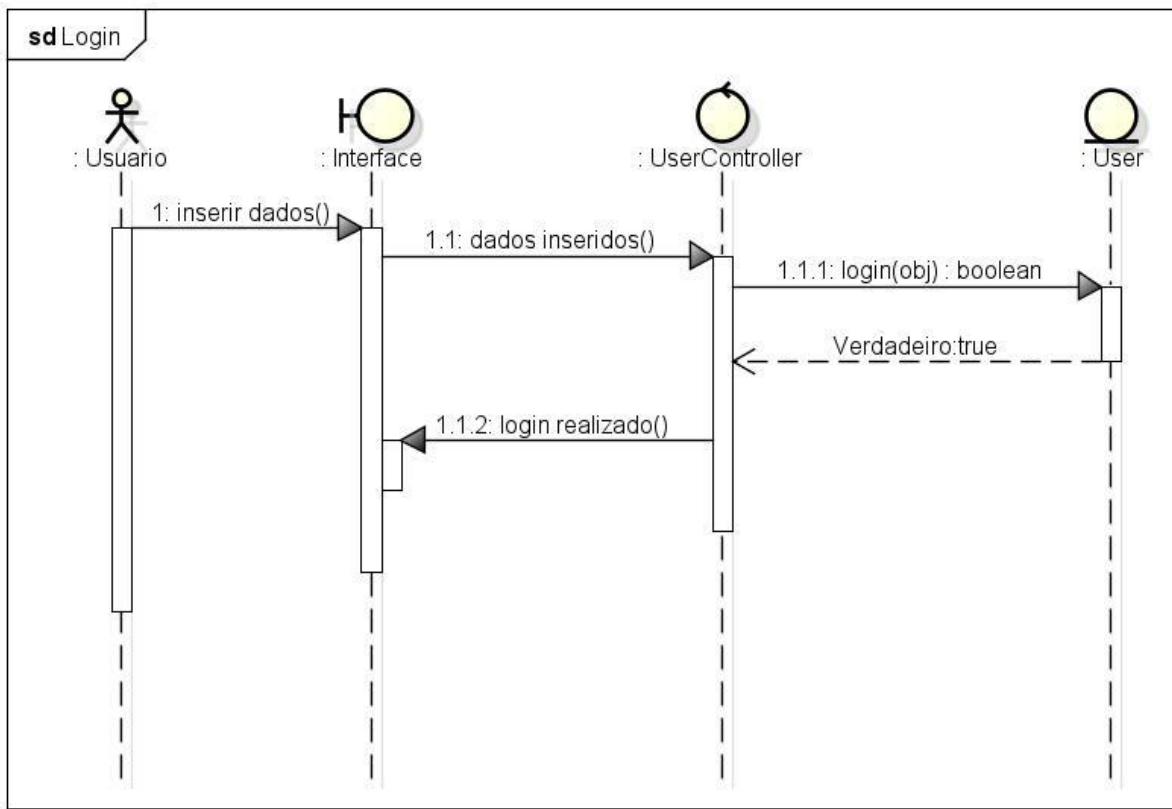
Fonte: Do Próprio Autor, 2023.

Figura 38 - Diagrama de Sequência Realizar Questionário



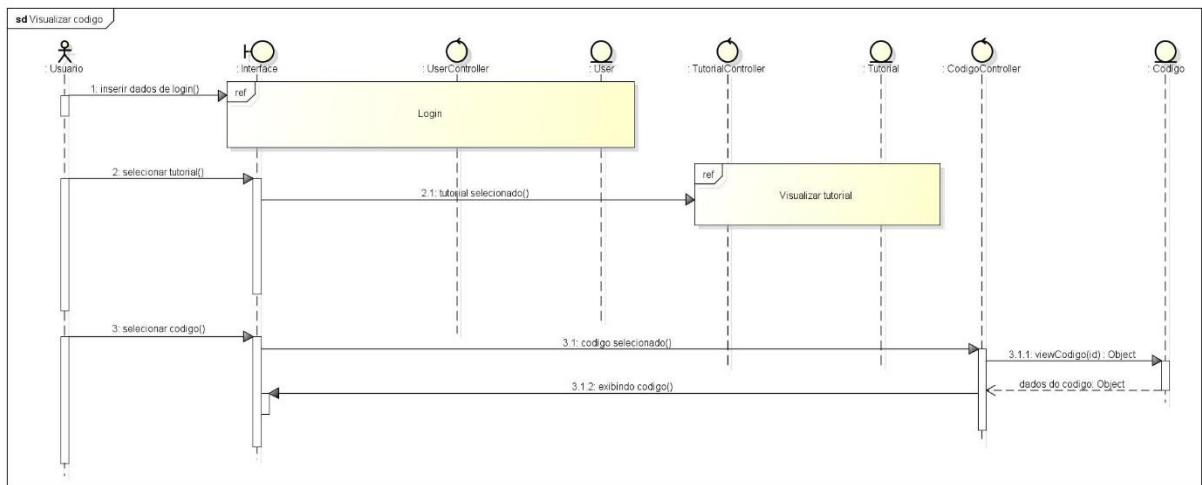
Fonte: Do Próprio Autor, 2023.

Figura 39 - Diagrama de Sequência Realizar Login



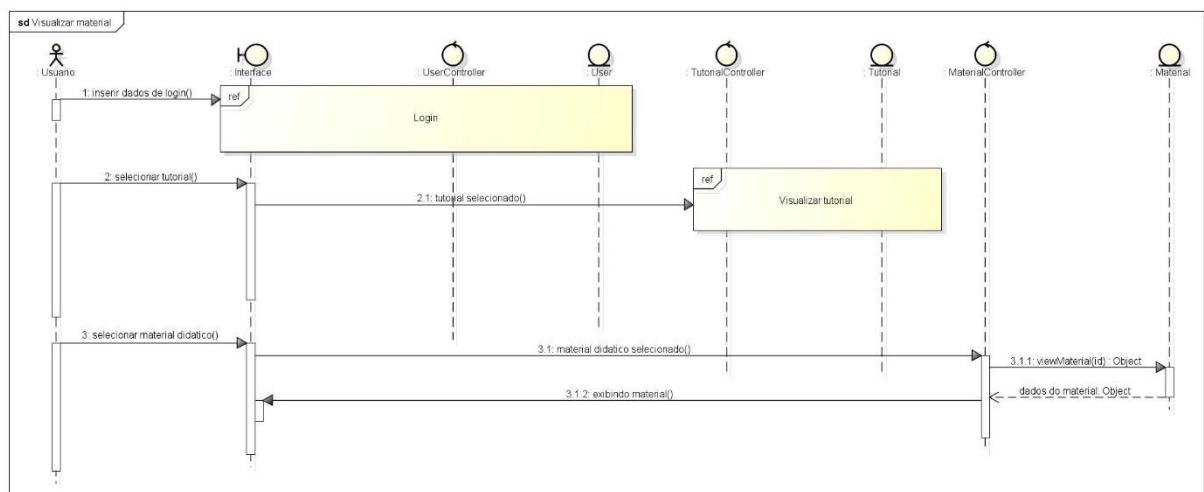
Fonte: Do Próprio Autor, 2023.

Figura 40 - Diagrama de Sequência Visualizar Código



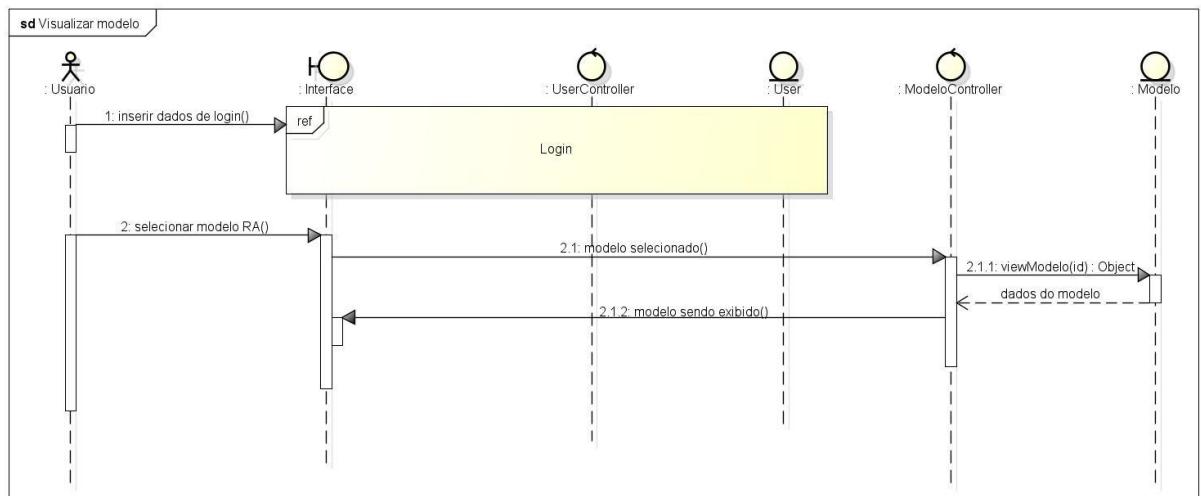
Fonte: Do Próprio Autor, 2023.

Figura 41 - Diagrama de Sequência Visualizar Material



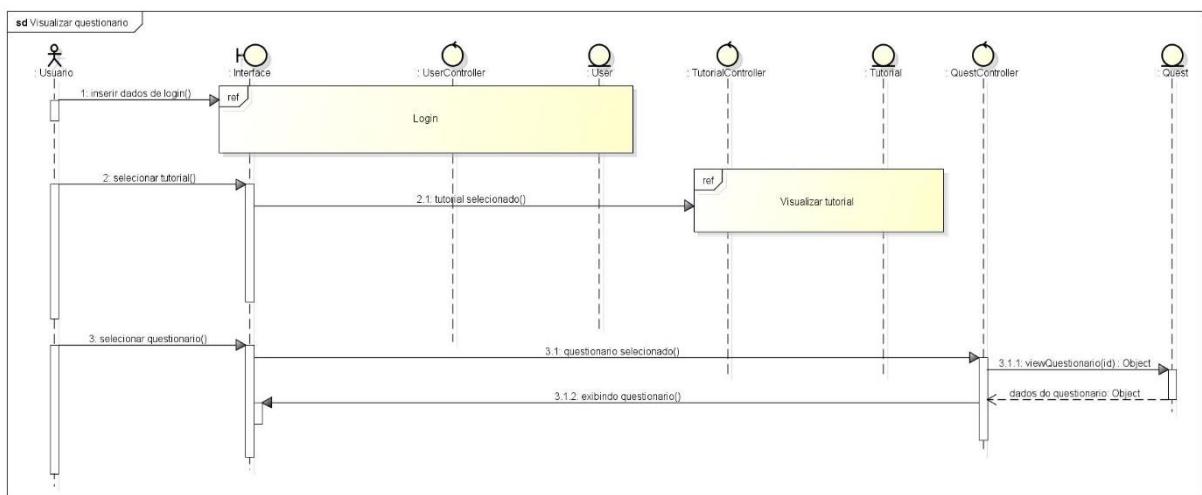
Fonte: Do Próprio Autor, 2023.

Figura 42 - Diagrama de Sequência Visualizar Modelo



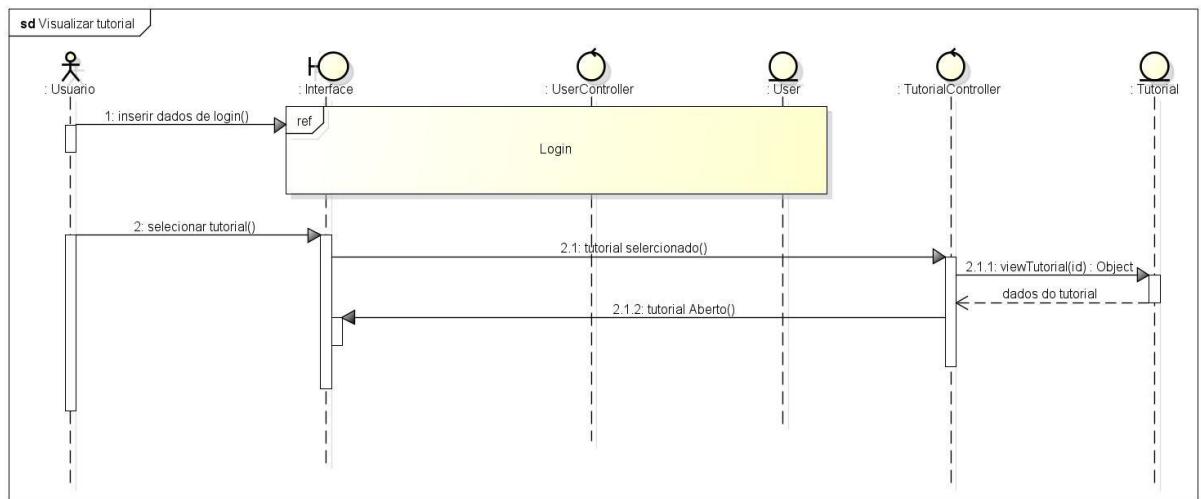
Fonte: Do Próprio Autor, 2023.

Figura 43 - Diagrama de Sequência Visualizar Questionário



Fonte: Do Próprio Autor, 2023.

Figura 44 - Diagrama de Sequência Visualizar Tutorial

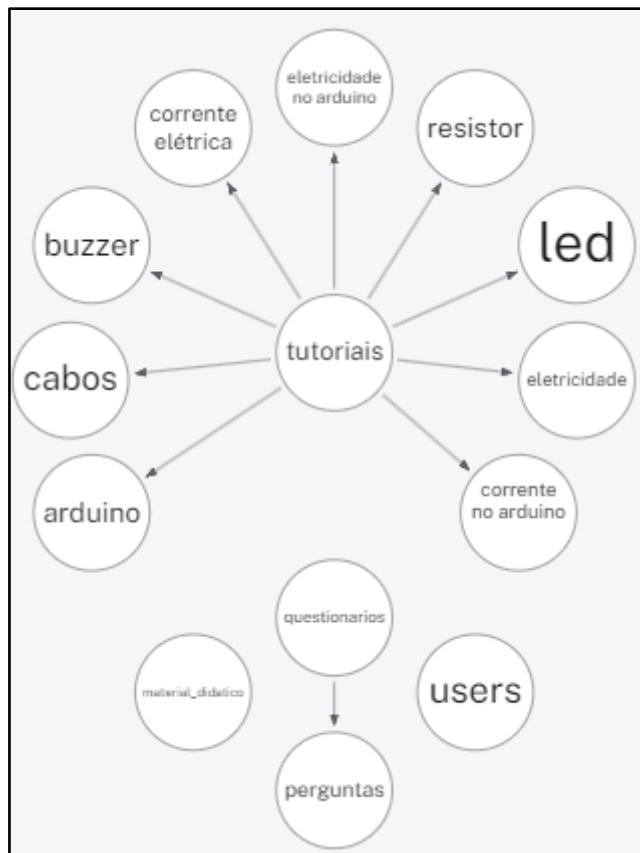


Fonte: Do Próprio Autor, 2023.

3.5 Representação em grafo do banco de dados AR-PIN

Neste tópico será apresentado a representação gráfica do banco de dados não relacional do projeto AR-PIN

Figura 45 - Grafo AR-PIN

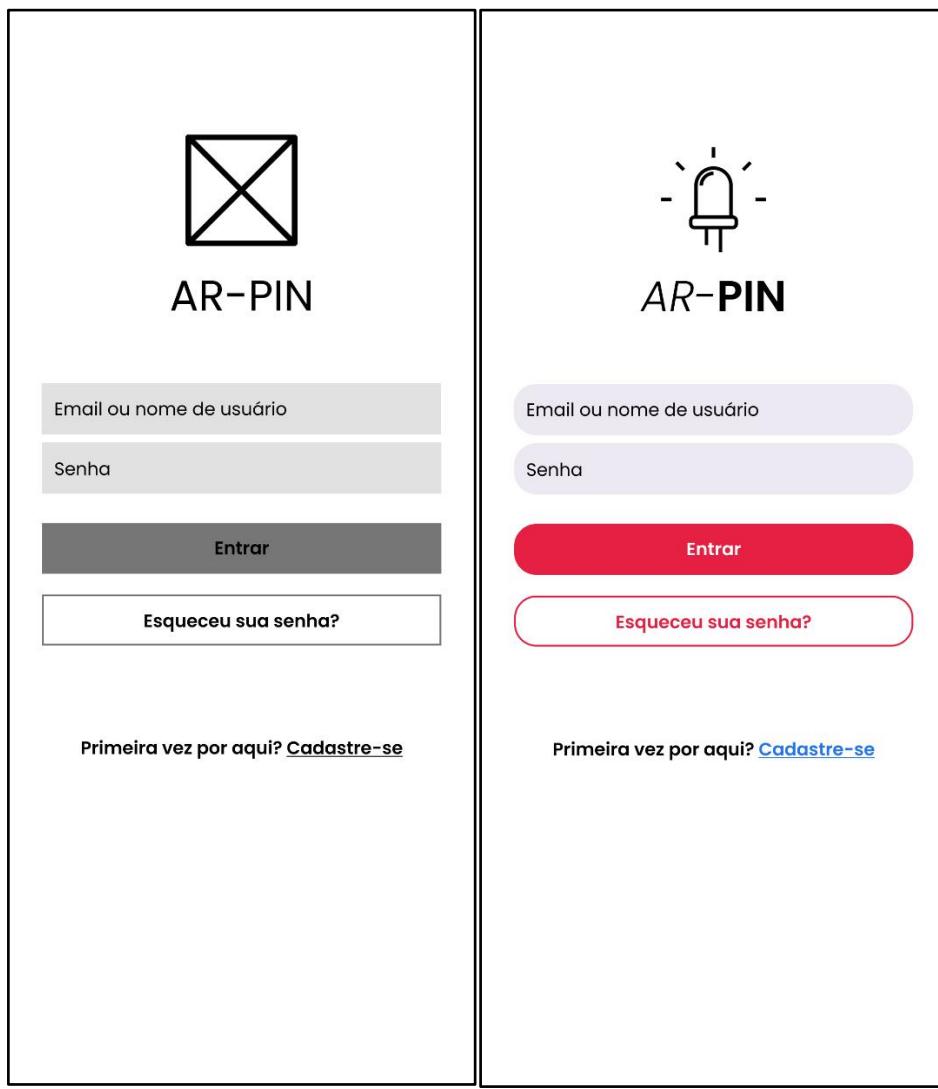


Fonte: Do Próprio Autor, 2023.

3.6 Prototipação das telas

Neste tópico será apresentado os wireframes de baixa fidelidade e os protótipos das telas da aplicação.

Figura 46 - Wireframe de Baixa/Alta Fidelidade “Realizar Login”



Fonte: Do Próprio Autor, 2023.

Neste wireframe, é apresentada a tela de login, onde o usuário poderá entrar em sua conta se ele já possuir uma, caso contrário, ele poderá criar uma conta clicando em “Cadastre-se”.

Figura 47 - Wireframe Baixa/Alta Fidelidade "Realizar Cadastro"

The wireframe illustrates a user registration process across two versions of a form.

Left Column (Low Fidelity):

- Header:** A large square icon containing a black 'X' and the text "AR-PIN".
- Slogan:** A rectangular box containing the text "Slogan".
- Section Title:** "Crie sua Conta" (Create your Account).
- Input Fields:** Four stacked input fields labeled "Nome de usuário" (User Name), "Email", "Senha" (Password), and "Confirmar Senha" (Confirm Password).
- Primary Action:** A dark grey button labeled "Criar Conta" (Create Account).
- Secondary Action:** Text "Já tem uma conta? [Entrar](#)" (Already have an account? [Enter](#)) at the bottom.

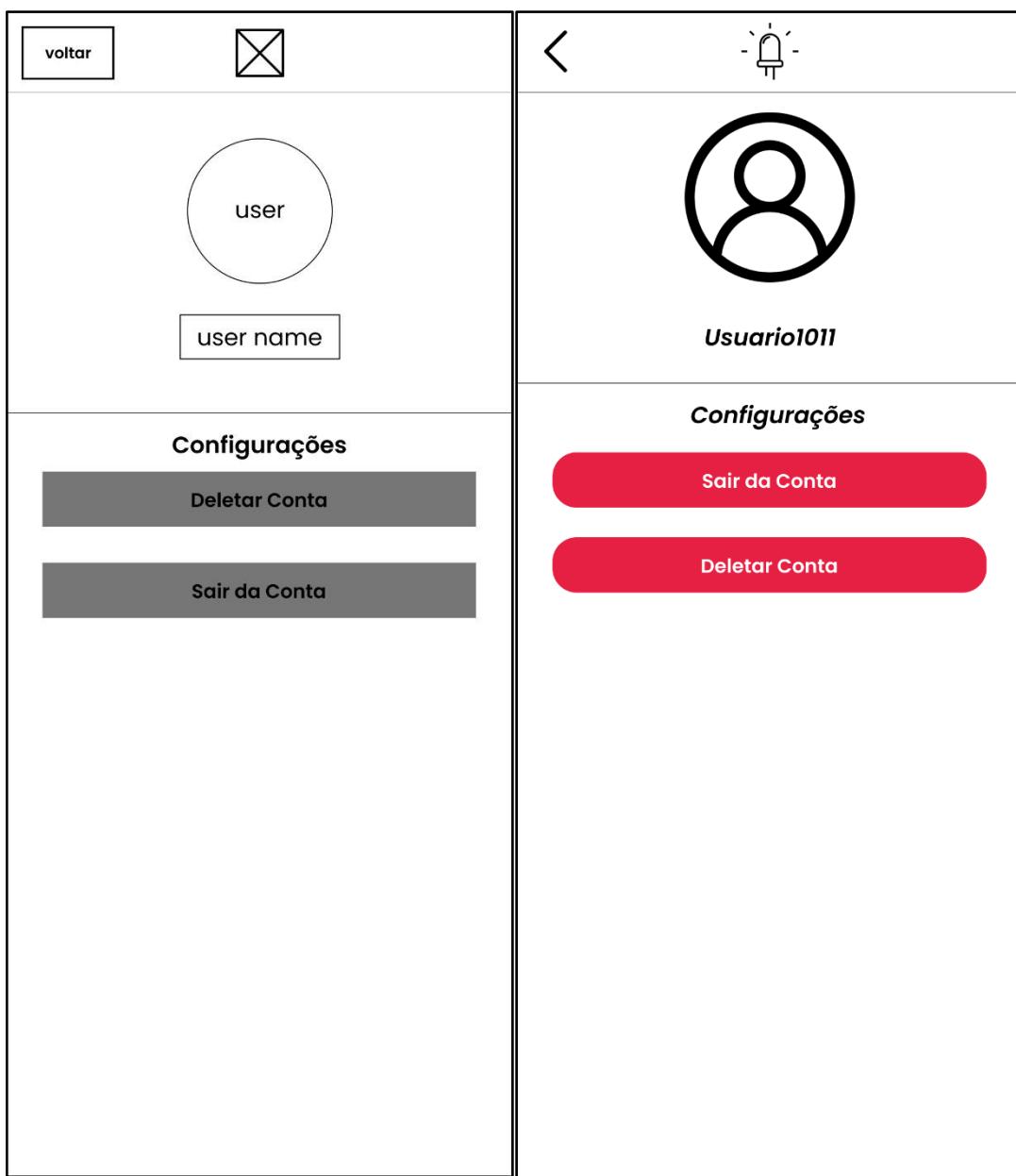
Right Column (High Fidelity):

- Header:** A small icon of a lightbulb with dashed lines above it, followed by the text "AR-PIN".
- Text:** The slogan "Porque ensinar não é apenas transmitir conhecimento!" (Because teaching is not just transmitting knowledge!).
- Section Title:** "Crie sua Conta" (Create your Account).
- Input Fields:** Four rounded input fields labeled "Nome de usuário" (User Name), "Email", "Senha" (Password), and "Confirmar Senha" (Confirm Password).
- Primary Action:** A red button labeled "Criar Conta" (Create Account).
- Secondary Action:** Text "Já tem uma conta? [Entrar](#)" (Already have an account? [Enter](#)) at the bottom.

Fonte: Do Próprio Autor, 2023.

No wireframe acima, é apresentada a tela para a realização do cadastro na plataforma AR-PIN, onde é necessário inserir o E-mail desejado, nome de usuário e uma senha.

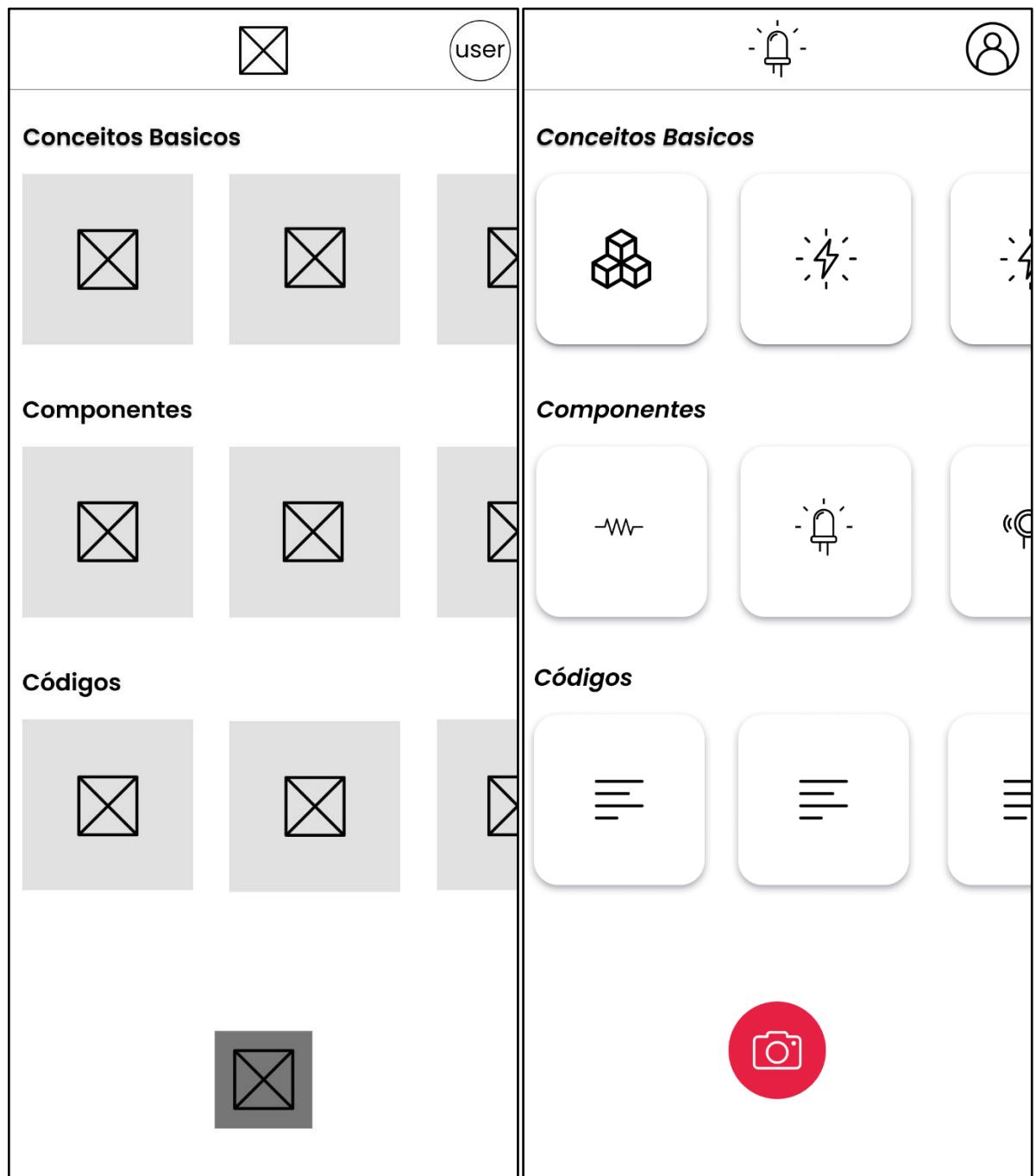
Figura 48 - Wireframe de Baixa/Alta Fidelidade "Perfil"



Fonte: Do Próprio Autor, 2023.

Neste wireframe é apresentada a tela do perfil do usuário, onde é possível visualizar e editar algumas informações da conta, como E-mail, senha e foto de perfil.

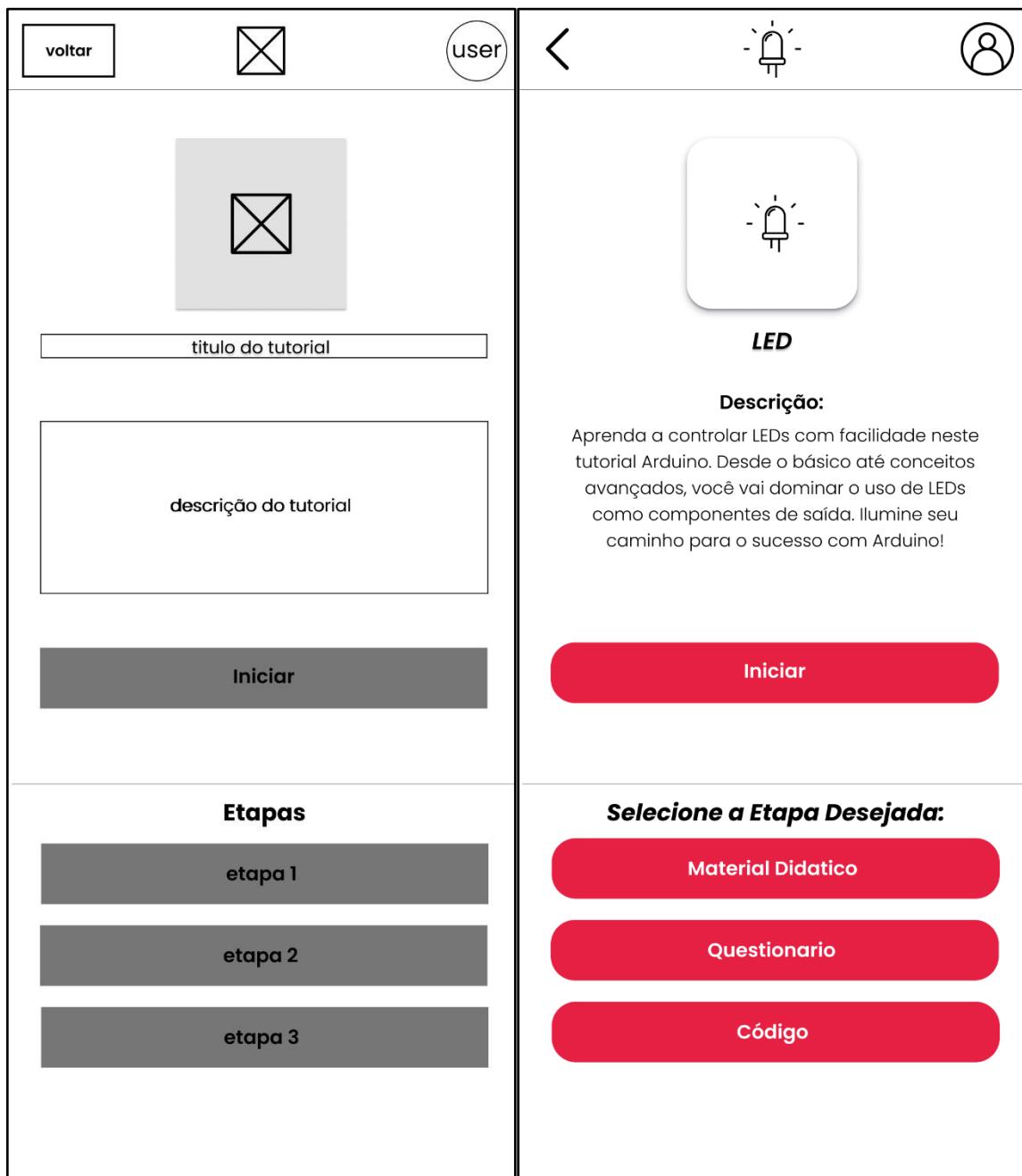
Figura 49 - Wireframe de Baixa/Alta Fidelidade "Página Inicial"



Fonte: Do Próprio Autor, 2023.

O wireframe acima apresenta a página inicial da aplicação, onde é exposto o conteúdo do aplicativo, o usuário poderá escolher qual tutorial deseja realizar, desde conceitos básicos para o entendimento do Arduino, códigos ou componentes do mesmo.

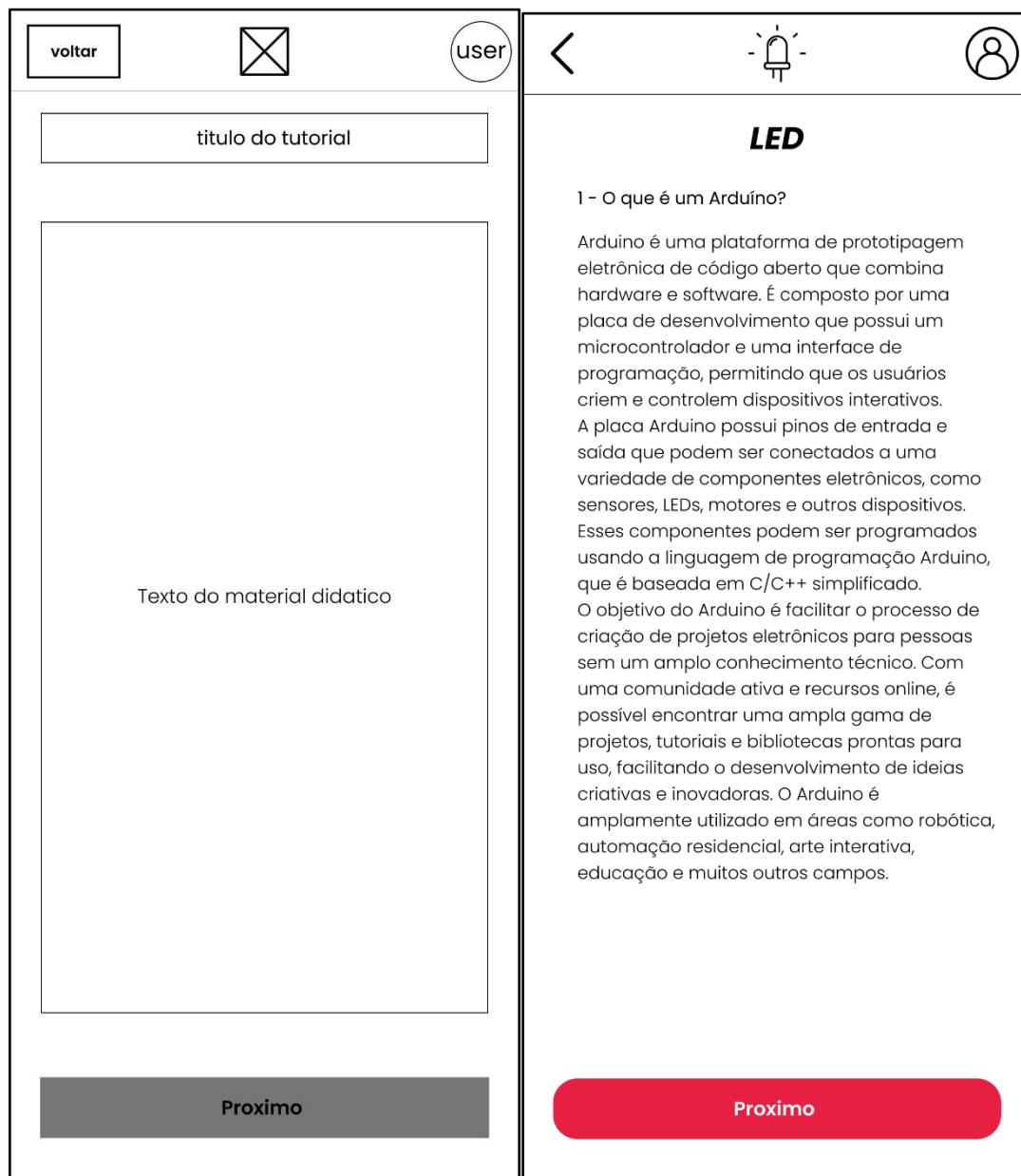
Figura 50 - Wireframe de Baixa/Alta Fidelidade "Tutorial Aberto"



Fonte: Do Próprio Autor, 2023.

A figura acima mostra como é a configuração do tutorial, o usuário recebe a descrição do que irá aprender neste tutorial e lhe é dada a opção de selecionar quais etapas deseja realizar, podendo escolher entre realizar todas elas ou não, as etapas incluem a visualização do projeto, a leitura do material didático e um questionário.

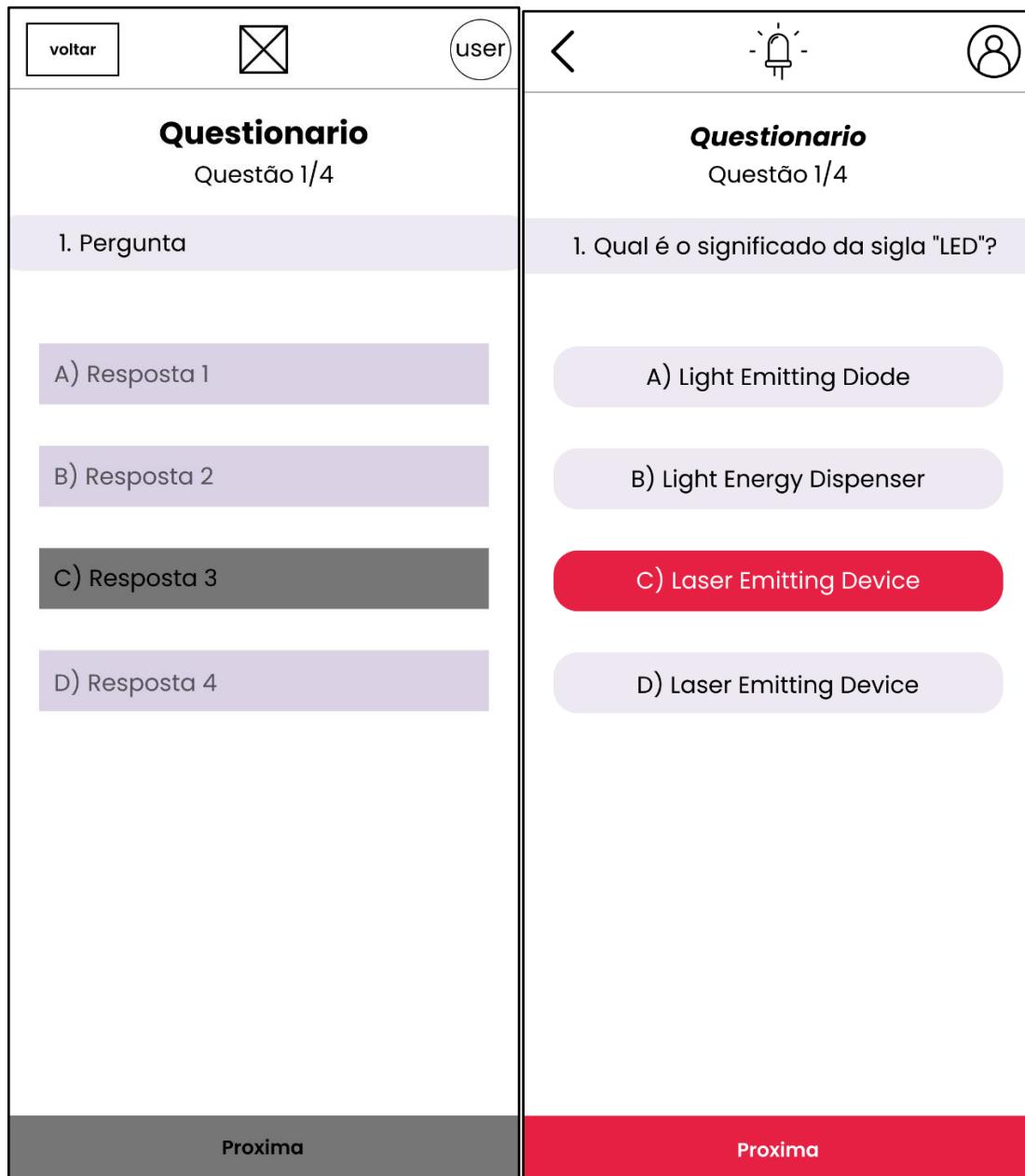
Figura 51 - Wireframe de Baixa/Alta Fidelidade "Material Didático"



Fonte: Do Próprio Autor, 2023.

Na figura acima é apresentada a tela com o conteúdo do material didático, onde é explicado o conteúdo teórico do tutorial.

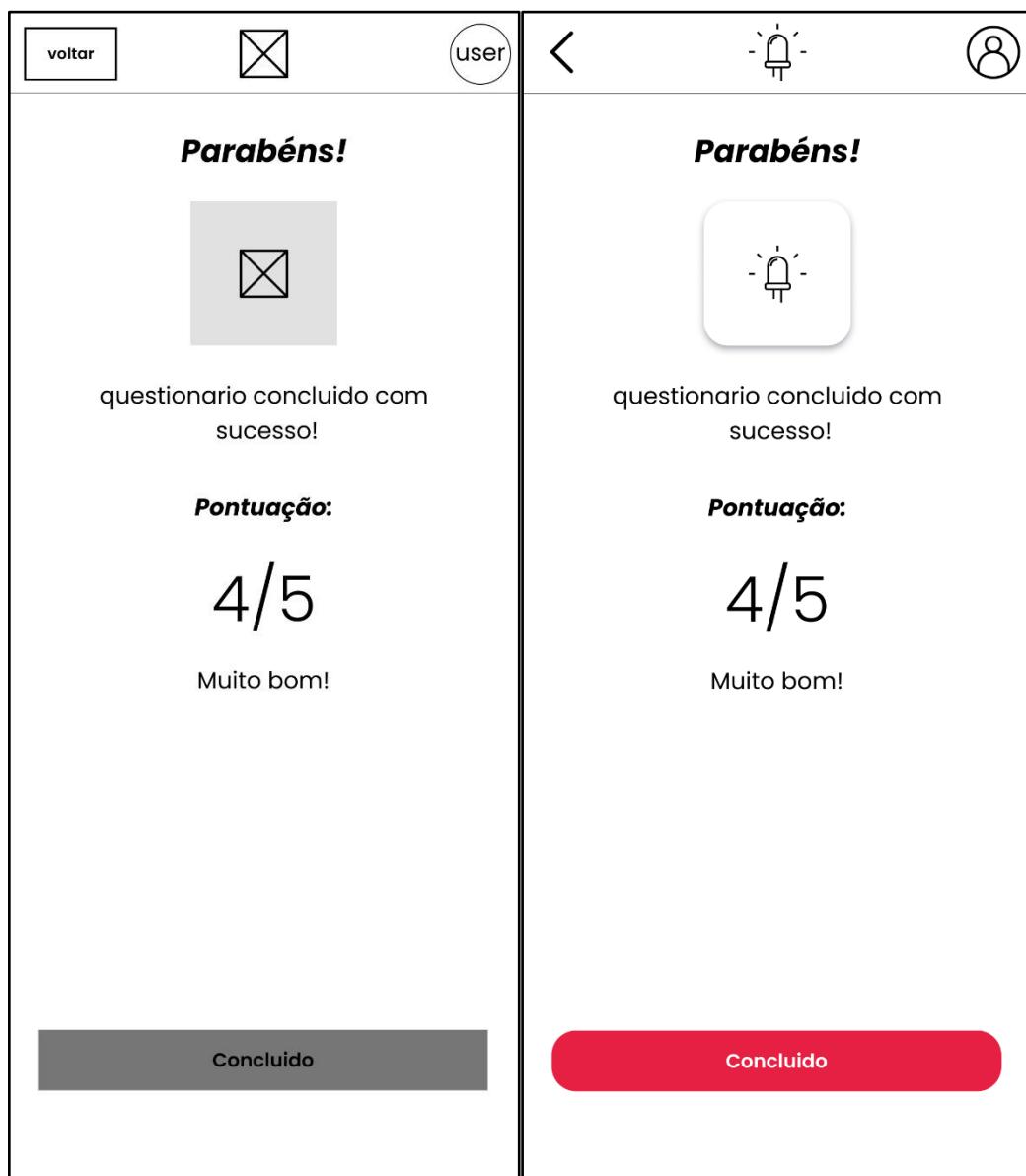
Figura 52 - Wireframe de Baixa/Alta Fidelidade "Questionário"



Fonte: Do Próprio Autor, 2023

Este wireframe é apresentada a tela do questionário, o usuário pode selecionar somente uma alternativa dentre todas.

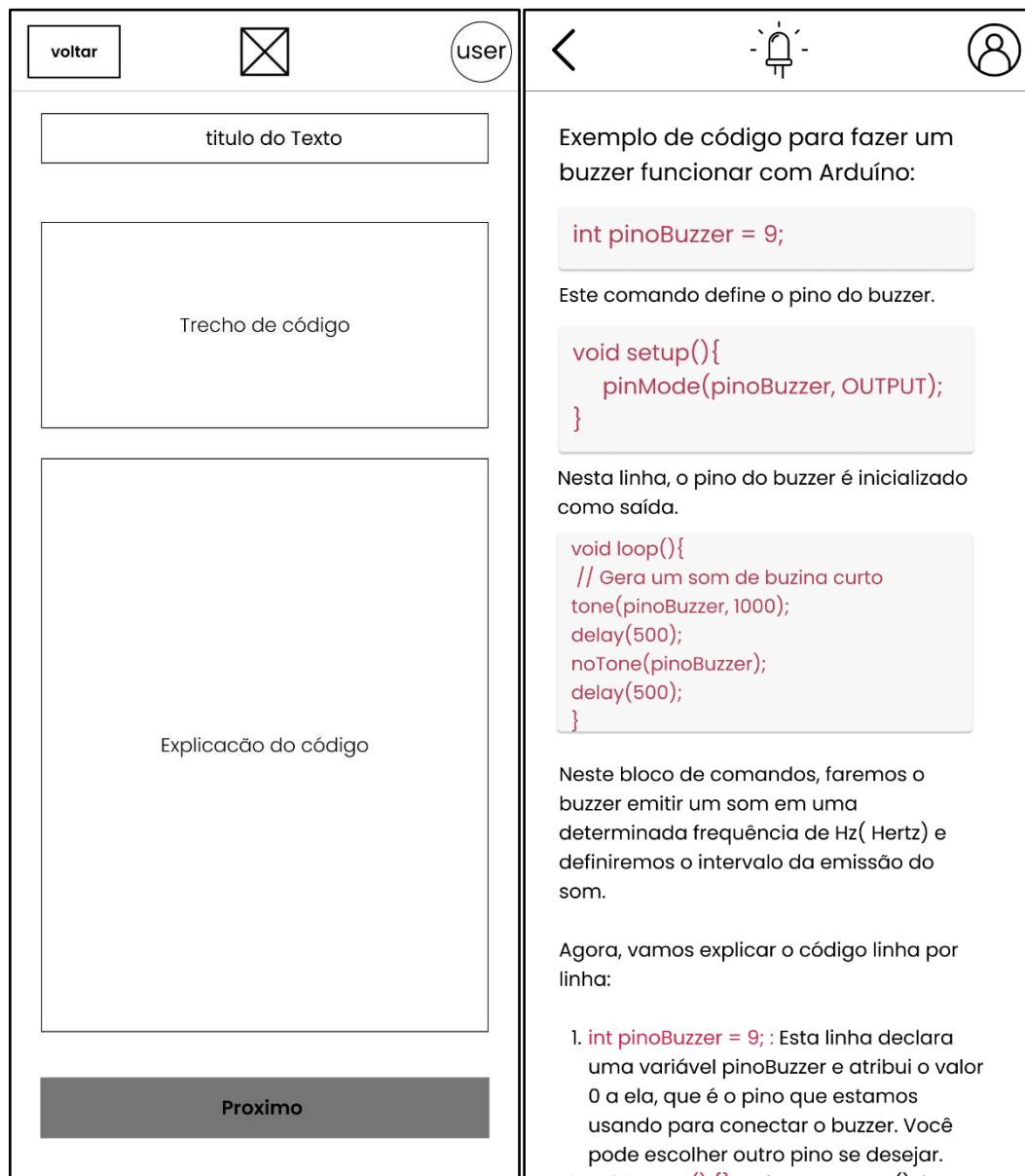
Figura 53 - Wireframe de Baixa/Alta Fidelidade "Questionário Concluído"



Fonte: Do Próprio Autor, 2023.

O wireframe acima apresenta a tela de conclusão do questionário, ela é exibida somente após todas as perguntas do questionário forem respondidas, também é informado a quantidade de acertos.

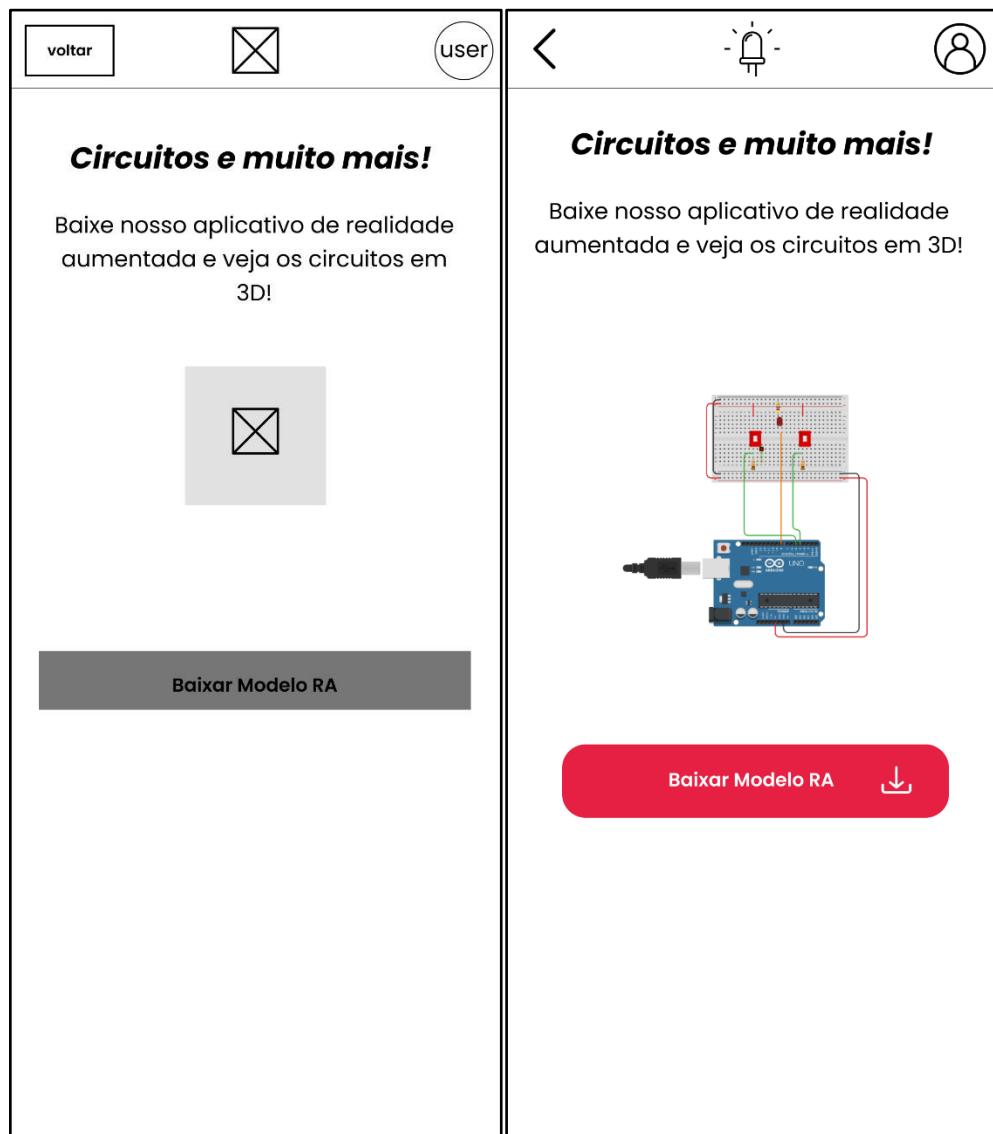
Figura 54 - Wireframe de Baixa/Alta Fidelidade "Código"



Fonte: Do Próprio Autor, 2023.

O wireframe acima apresenta a tela de apresentação de um exemplo de código, na qual, ao ser exibida, mostra ao usuário um código funcional a respeito de tal lição bem como descrição de determinadas funções.

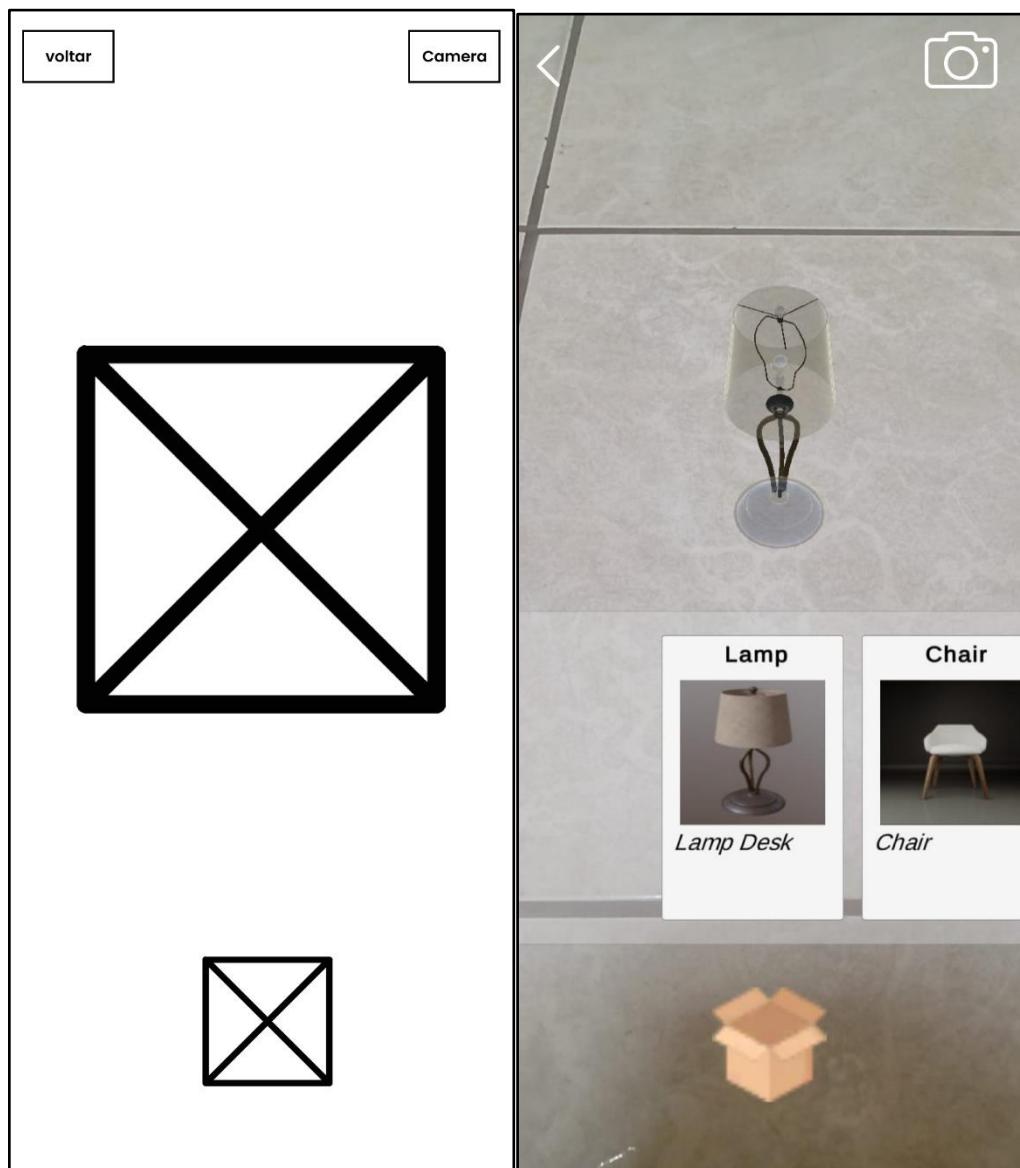
Figura 55 - Wireframe de Baixa/Alta Fidelidade "Baixar Modelos 3D"



Fonte: Do Próprio Autor, 2023.

Este wireframe apresenta a tela de disponibilidade de modelos 3D, dando ao usuário a opção de baixar os modelos.

Figura 56 - Wireframe de Baixa e Alta Fidelidade "Modelo 3D"



Fonte: Do Próprio Autor, 2023.

Neste wireframe é apresentada a tela de exibição do projeto em Realidade Aumentada, o usuário poderá interagir com o modelo em RA, alternar entre a exibição do projeto em RA e o código por trás do funcionamento do projeto.

4 Considerações Finais

A educação híbrida, com diversidade de conhecimentos e ferramentas que permitem aos alunos estimular a sua aprendizagem através do método científico, demonstra inovação e melhor gestão dos talentos de cada indivíduo, permitindo-lhes desenvolver o gosto pela aprendizagem de acordo com as suas capacidades.

O desempenho de um aluno depende de diversas causas e não pode ser atribuído apenas à motivação. Como explica Pereira (2015), o desempenho de um aluno depende de múltiplas variáveis de ordem emocional, motivacional, mental, além de outros fatores internos e externos.

As instituições educativas, entre as suas muitas responsabilidades, também têm a tarefa de fornecer ferramentas para a aprendizagem. Segundo Paulo Freire (1996), “ensinar não é transferência de conhecimento, mas a criação de possibilidades para sua própria produção ou construção”.

O objetivo deste trabalho foi desenvolver hipóteses, baseadas no rigor científico, para a criação de um sistema como ferramenta para proporcionar aprendizagem acessível na área do conhecimento computacional, especificamente em sistemas embarcados, utilizando ferramentas semelhantes às utilizadas neste projeto.

A hipótese é que um sistema de aprendizagem em sistemas embarcados contribuiria para a construção do conhecimento de forma acessível a todos. A sua natureza interativa e fácil de usar pode servir como estimulante para a aprendizagem e inovação contínuas, beneficiando tanto os indivíduos como a sociedade como um todo.

Como desenvolvedores e projetistas desta hipótese para resolver a desafiadora questão da disponibilidade e acessibilidade do conhecimento, juntamente com a educação híbrida, consideramos a ferramenta como uma solução satisfatória para o problema colocado pela falta de componentes eletrônicos para a aprendizagem de alunos de sistemas embarcados .

REFERÊNCIAS

BARBOSA, Derick Mazelli; CANDIOTO, Katia Cristiane Gandolpho. **PROTOTIPAGEM ELETRÔNICA COM ARDUINO**. Jundiaí, Sp: Edições Brasil, 2022. 67 p.

BARBOSA, Fabio José Marques et al. Visualização da Informação e Métodos Visuais como Ferramentas Estratégicas para o Gerenciamento de Projetos. **Revista de Gestão e Projetos, [S.L.], v. 9, n. 1, p. 102-114, 3 maio 2018. University Nove de Julho.** <http://dx.doi.org/10.5585/gep.v9i1.635>. Disponível em: <https://periodicos.uninove.br/gep/article/view/9685/0>. Acesso em: 09 jul. 2023.

BAULÉ, Daniel de Souza. **Desenvolvimento de um modelo de geração automática de wireframes no app inventor a partir de sketches usando deep learning**. 2020. 166 f. TCC (Graduação) - Curso de Ciências da Computação, Informática e Estatística, Universidade Federal de Santa Catarina, Florianópolis, 2020. Disponível em: <https://repositorio.ufsc.br/handle/123456789/218145>. Acesso em: 19 jul. 2023.

BOOCH, Grady; JACOBSON, Ivar; RUMBAUGH, James. **UML: guia do usuário**. 2. ed. Rio de Janeiro: Elsevier, 2006. 473 p. ISBN 9788535217841.

CARLUCCI, André et al. **C# Para Iniciantes**. São Paulo: Agência Hex, 2021. Disponível em: <https://livrocsharp.com.br/wp-content/uploads/dae-uploads/CSharplniciantes.pdf>. Acesso em: 01 jul. 2023.

CHRISTENSEN, Clayton M. et al. Ensino Híbrido: uma inovação disruptiva?. **Clayton Christensen Institute**: for Disruptive Innovation, Boston, v. 4, p. 1-43, 21 maio 2013. Traduzido para o Português por Fundação Lemann e Instituto Península. Disponível em: https://porvir.org/wp-content/uploads/2014/08/PT_Is-K-12-blended-learning-disruptive-Final.pdf. Acesso em: 25 jun. 2023.

DIANA, Mauricio de; GEROUSA, Marco Aurélio. **NOSQL na Web 2.0**: um estudo comparativo de bancos não-relacionais para armazenamento de dados na web 2.0. 2010. 8 f. Tese (Doutorado) - Curso de Ciência da Computação, Universidade de São Paulo (Usp), São Paulo, Sp, 2010. Disponível em: http://200.17.137.109:8081/novobsi/Members/josino/fundamentos-de-banco-de-dados/2012.1/sbbd_wtd_12.pdf. Acesso em: 25 jun. 2023.

EVANS, Martin; NOBLE, Joshua; HOCHENBAUM, Jordan. **Arduino Em Ação**. São Paulo: Novatec, 2013. p. 28.

FERREIRA, Joana Rita Santos. **Realidade Aumentada**: conceito, tecnologia e aplicações. 2014. 89 f. Dissertação (Mestrado) - Curso de Engenharia e Gestão Industrial, Universidade da Beira Interior, Covilhã, 2014. Disponível em: https://ubiblitorum.ubi.pt/bitstream/10400.6/5907/1/3930_7645.pdf. Acesso em: 19 jul. 2023.

FREIRE, Paulo. **Pedagogia da Autonomia**. São Paulo: Paz e Terra, 1996.

GARCIA, Vinícius Salles; SOTTO, Eder Carlos Salazar. Comparativo Entre Os Modelos De Banco De Dados Relacional E Não-Relacional. **Revista Interface Tecnológica**, [S.L.], v. 16, n. 2, p. 12-24, 21 dez. 2019. Interface Tecnológica. <http://dx.doi.org/10.31510/infa.v16i2.673>. Disponível em: <https://revista.fatectq.edu.br/interfacetecnologica/article/view/673>. Acesso em: 25 jun. 2023.

GIL, Antonio Carlos. **Como Elaborar Projetos de Pesquisa**. 3. ed. São Paulo: Atlas S.A, 1991. 101 p. Disponível em: https://sgcd.fc.unesp.br/Home/helber-freitas/tcci/gil_como_elaborar_projetos_de_pesquisa_-anto.pdf. Acesso em: 26 jul. 2023.

GUEDES, Gilleanes T. A. **UML: uma abordagem prática**. 2. ed. São Paulo: Novatec, 2018.

KENSKI, Vani Moreira. **Educação e tecnologias: o novo ritmo da informação**. Campinas: Papirus, 2007. 144 p.

KREUCH, Eduardo Schmitt. **Automação da Identificação de UI Design Patterns a partir de Wireframes utilizando Machine Learning**. 2021. 110 f. TCC (Graduação) - Curso de Sistemas de Informações, Informática e Estatística, Universidade Federal de Santa Catarina, Florianópolis, 2021. Disponível em: <https://repositorio.ufsc.br/handle/123456789/228238>. Acesso em: 19 jul. 2023.

LOPES, Luana Monique Delgado; VIDOTTO, Kajiana Nuernberg Sartor; POZZEBON, Eliane; FERENHOF, Helio Aisenberg. **INOVAÇÕES EDUCACIONAIS COM O USO DA REALIDADE AUMENTADA: UMA REVISÃO SISTEMÁTICA**.

Scielo, 2019 Disponível em:
<https://www.scielo.br/j/edur/a/D8BG7VqVDPmYk3d5xmCJJyF/>. Acesso em: 21 jun. 2023.

MAGON, Claudio José. **Conceitos básicos da Eletrônica**: teoria e prática. São Paulo, Sp: Ifsc, 2018. 339 p.

MARTINS, Alessandro Rocha; CONCEIÇÃO, Lucas da; PEREIRA, Romantiezer Beloni; SILVA, André Bessa da. Aplicativos Híbridos: Desenvolvimento De Aplicativos Utilizando Tecnologias Web. **Revista Científica Ambiente Acadêmico**, [s. l], v. 6, n. 1, p. 26-39, jun. 2020. Disponível em: <https://multivix.edu.br/wp-content/uploads/2020/07/revista-ambiente-academico-v06-n01-artigo02.pdf>. Acesso em: 27 jun. 2023.

MICROBERTS, Michael. **Arduino Básico**. São Paulo, Sp: Novatec, 2011. 456 p.

MICROSOFT. **Documentação do C#**. Disponível em: <https://learn.microsoft.com/pt-br/dotnet/csharp/>. Acesso em: 21 jun. 2023.

MICROSOFT. **Guia de programação em C#**. Disponível em: <https://learn.microsoft.com/pt-br/dotnet/csharp/programming-guide/>. Acesso em 21 jun. 2023.

MICROSOFT. **Introdução ao C#**. Disponível em: <https://learn.microsoft.com/pt-BR/dotnet/csharp/tour-of-csharp/tutorials/>. Acesso em 21 jun. 2023.

MIRANDA JUNIOR, Luiz Carlos de. **Segurança em instalações e serviços em eletricidade**. São Paulo: Cpn, 2006. Disponível em:
https://wiki.sj.ifsc.edu.br/images/0/04/NR10_Manual_Fundacentro.pdf. Acesso em: 01 jul. 2023.

NOVELLI, Renan. **Modelagem e interação em ambiente 3d utilizando Blender e Unity**. 2015. 50 f. TCC (Graduação) - Curso de Tecnologia em Análise e Desenvolvimento de Sistemas, Universidade Tecnológica Federal do Paraná, Curitiba, Pr, 2015. Disponível em: <http://repositorio.utfpr.edu.br/jspui/handle/1/13391>. Acesso em: 25 jun. 2023.

NUNES, Eunice P. dos Santos *et al.* **Tendências e Técnicas em Sistemas Computacionais - Comunicação por campo de proximidade**: tecnologia, aplicações e questões de segurança. Cuiabá, Mato Grosso: Sociedade Brasileira de Computação, 2015. 161 p. Disponível em: https://www.researchgate.net/publication/360614842_Tendencias_e_Tecnicas_em_Sistemas_Computacionais_-Comunicacao_por_campo_de_proximidade_Tecnologia_aplicacoes_e_questoes_de_seguranca#fullTextFileContent. Acesso em: 25 jun. 2023.

OMAIA, Derzu; MACHADO, Liliane. Realidade Aumentada com Vuforia e Unity. **Anais Estendidos do Simpósio de Realidade Virtual e Aumentada (Srv Estendido 2020)**, [S.L.], p. 61-64, 7 nov. 2020. Sociedade Brasileira de Computação. http://dx.doi.org/10.5753/srv_estendido.2020.12958. Disponível em: https://doi.org/10.5753/srv_estendido.2020.12958. Acesso em: 25 jun. 2023.

PANIZ, David. **NoSQL: como armazenar os dados de uma aplicação moderna**. São Paulo, SP: Casa do Código, 2016. 177 p. Acesso em: 24 jun. 2023.

PEREIRA, Fernando Oliveira. **Especificidades do rendimento, aptidão e motivação escolares em alunos com dificuldade de aprendizagem**. 2015. Disponível em: <https://www.scielo.br/j/pee/a/DkkYyDtKZ67dRN RVXK6YH7N/>. Acesso em: 27 jun. 2023.

RABELO, Daniel Ferreira; CÂNDIDO, Marco Vinicius Isecke. **Análise de desempenho de banco de dados nosql em um sistema que utiliza um banco de dados relacional e não relacional para armazenamento de dados**. 2017. 88 f. TCC (Graduação) - Curso de Engenharia de Computação, Centro Universitário de Anápolis Unievangélica, Anápolis, 2017. Disponível em: http://repositorio.aee.edu.br/bitstream/aee/45/1/TCC2_2017_02_DanielFerreiraRabelo_MarcoViniciusIseckeCandido.pdf. Acesso em: 03 nov. 2023.

SAADE, Joel. **C#: guia do programador**. São Paulo: Novatec Editora, 2010.

SAVIANI, Demeval. **Pedagogia Histórico-Crítica: primeiras aproximações**. 2. ed. São Paulo: Cortez / Autores Associados, 1991. (Coleção polêmicas do nosso tempo; v. 5).

SOUZA, Anderson R. de; PAIXÃO, Aleksander C.; UZEDA, Diego D.; DIAS, Marco A.; DUARTE, Sergio; AMORIM, Helio S. de. A placa Arduino: uma opção de baixo custo para experiências de física assistidas pelo pc. **Revista Brasileira de Ensino de Física**, [S.L.], v. 33, n. 1, p. 01-05, mar. 2011. FapUNIFESP (SciELO).
<http://dx.doi.org/10.1590/s1806-11172011000100026>. Acesso em: 25 jul. 2023.

TARDIF, M. **Saberes docentes e formação profissional**. 17.ed. Petrópolis, RJ: Vozes, 2014. Disponível em: https://www.pucpr.br/wp-content/uploads/2017/10/ensino-hibrido_uma-inovacao-disruptiva.pdf. Acesso em: 25 jul. 2023.

TEIXEIRA, Fabricio. **Introdução e boas práticas em UX Design**. São Paulo, SP: Casa do Código, 2015.

TORI, Romero; KIRNER, Claudio; SISCOUTTO, Robson. **Fundamentos e Tecnologia de Realidade Virtual e Aumentada**. Belém Pa: Editora Sbc – Sociedade Brasileira de Computação, Porto Alegre, 2006. 422 p. Disponível em: https://pcs.usp.br/interlab/wp-content/uploads/sites/21/2018/01/Fundamentos_e_Tecnologia_de_Realidade_Virtual_e_Aumentada-v22-11-06.pdf. Acesso em: 23 jun. 2023.

VALENTE, José Armando; FREIRE, Fernanda Maria Pereira; ROCHA, Heloísa Vieira da; D'ABREU, Jose Vilhete; BARANAUSKAS, Maria Cecília Calani; MARTINS, Maria Cecília; PRADO, Maria Elisabete Brisola Brito (org.). **O computador na sociedade do conhecimento**. Campinas: Nied, 1999. 116 p. Disponível em: http://penta3.ufrrgs.br/MEC-CicloAvan/integracao_midias/textos/1pref.pdf. Acesso em: 06 jun. 2023.

VENTEU, Kelly Cristina; PINTO, Giuliano Scombatti. **DESENVOLVIMENTO MÓVEL HÍBRIDO**. Revista Interface Tecnológica, [S.L.], v. 15, n. 1, p. 86-96, 30 jun. 2018. Interface Tecnologica. <http://dx.doi.org/10.31510/infa.v15i1.337>. Acesso em: 29 jul. 2023.