

INSTITUTO FEDERAL DO ESPÍRITO SANTO  
CURSO SUPERIOR DE TECNOLOGIA EM ANÁLISE E  
DESENVOLVIMENTO DE SISTEMAS

**GABRIEL DO NASCIMENTO FERREIRA**

**EVOLOGIC:  
APLICATIVO PARA ENSINAR LOGICA DE PROGRAMAÇÃO**

ALEGRE-ES  
2022

**GABRIEL DO NASCIMENTO FERREIRA**

**EVOLOGIC:  
APLICATIVO PARA ENSINAR LOGICA DE PROGRAMAÇÃO**

Trabalho de Conclusão de Curso apresentado à Coordenadoria do Curso de Análise e Desenvolvimento de Sistemas do Instituto Federal do Espírito Santo, como requisito parcial para a obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas.

Orientador: Flávio Pavesi Simão

ALEGRE-ES

2022

GABRIEL DO NASCIMENTO FERREIRA

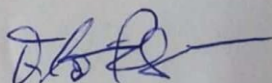
EVOLOGIC:

APLICATIVO PARA ENSINAR LOGICA DE PROGRAMAÇÃO

Trabalho de Conclusão de Curso apresentado à Coordenadoria do Curso de Análise e Desenvolvimento de Sistemas do Instituto Federal do Espírito Santo, como requisito parcial para a obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas.

Aprovado em 30 de novembro de 2022.

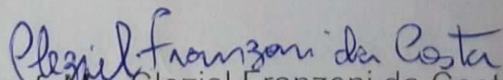
COMISSÃO EXAMINADORA



Prof. Dr. Flávio Pavesi Simão

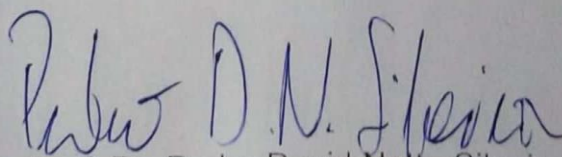
Instituto Federal do Espírito Santo

Orientador



Prof. Me. Cleziel Franzoni da Costa

Instituto Federal do Espírito Santo



Prof. Dr. Pedro David Netto Silveira

Instituto Federal do Espírito Santo

Dados Internacionais de Catalogação-na-Publicação (CIP)  
Biblioteca Monsenhor José Bellotti – IFES campus de Alegre

---

F383e Ferreira, Gabriel do Nascimento

Evologic: aplicativo para ensinar Lógica de Programação / Gabriel do Nascimento Ferreira – 2022.

40 f. : il

Orientador: Flávio Pavesi Simão

Monografia (Graduação) – Instituto Federal de Educação, Ciência e Tecnologia do Espírito Santo. Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas, 2022.

1. Aplicativos móveis. 2. Programação lógica (Computação). 3. Ensino auxiliado por computador. I. Simão, Flávio Pavesi. II. Título. III. Instituto Federal do Espírito Santo.

CDD 23: 005.1

---

Elaborado por Felipe Fernandes Klajn – CRB6-ES 984

## **AGRADECIMENTOS**

Eu agradeço a todos os meus familiares, aos meus amigos, colegas de faculdade, a todos os professores que fizeram parte da minha formação e principalmente ao meu orientador por todos os aprendizados passados.

## RESUMO

Esse trabalho surgiu da percepção sobre a dificuldade de aprendizagem dos alunos do curso de Análise e Desenvolvimento de Sistemas do IFES – *Campus* de Alegre, com relação a disciplina de programação. Neste trabalho será apresentado o EVOLOGIC, um aplicativo digital que tem como objetivo geral auxiliar, de forma dinâmica, alunos com dificuldade na aprendizagem de lógica de programação. Para tanto, foi necessário previamente conhecer os conceitos relacionados à lógica de programação, verificando também a contribuição dos aplicativos e videoaulas no processo de aprendizado. Para o desenvolvimento do referencial teórico foram realizadas leituras de artigos, livros e sites que abordavam sobre lógica de programação e aplicativos educacionais. Por meio das pesquisas, foi possível verificar uma contribuição positiva do uso dos aplicativos e videoaulas como uma ferramenta de apoio no desenvolvimento no estudo de alunos, desde que professores e responsáveis estejam preparados para utilizá-los. Ao final deste trabalho, foi possível desenvolver e disponibilizar o EVOLOGIC em sua versão inicial.

**Palavras-chaves:** Lógica de programação. Aprendizagem. Aplicativos. Videoaulas. Educação.

## **ABSTRACT**

This work arose from the perception of the learning difficulty of the students of the Systems Analysis and Development course at IFES – Campus de Alegre, regarding the programming discipline. In this work, EVOLOGIC will be presented, a digital application whose general objective is to help, in a dynamic way, students with difficulties in learning programming logic. Therefore, it was necessary to previously know the concepts related to programming logic, also verifying the contribution of applications and video lessons in the learning process. For the development of the theoretical framework, readings of articles, books and websites that addressed programming logic and educational applications were carried out. Through research, it was possible to verify a positive contribution of the use of applications and video lessons as a support tool in the development of students' study, provided that teachers and guardians are prepared to use them. At the end of this work, it was possible to develop and make available the EVOLOGIC in its initial version.

**Keywords:** Programming logic. Learning. applications. Video classes. Education.

## LISTA DE ILUSTRAÇÕES

Figura 1 - Diagrama de Casos de Uso.....	18
Figura 2 - Modelo Lógico do Banco de dados.....	25
Figura 3 - Diagrama de Classes.....	24
Figura 4 – Diagrama de Atividades.....	26
Figura 5 – Visualização do projeto em android.....	27
Figura 6 – Ambiente do Figma.....	28
Figura 7 – Implementação da Interface no android studio.....	29
Figura 8 – Tela de Login.....	29
Figura 9 – Tela de Registrar novo usuário.....	30
Figura 10 – Tela de inicial.....	31
Figura 11 – Tela de como funciona?.....	32
Figura 12 – Tela da vídeoaula.....	33
Figura 13 – Tela de Material da aula.....	33
Figura 14 – Tela de Pergunta.....	34
Figura 15 – Tela final da aula.....	35



## LISTA DE TABELAS

Tabela 1 - Requisitos Funcionais .....	17
Tabela 2 - Requisitos não Funcionais .....	17
Tabela 3 - Descrição dos atores .....	19
Tabela 4 - Especificação do caso de uso "Criar Usuário" .....	19
Tabela 5 - Especificação do caso de uso "Realizar Login" .....	19
Tabela 6 - Especificação do caso de uso "Assistir Vídeos" .....	20
Tabela 7 - Especificação do caso de uso "Visualizar material" .....	20
Tabela 8 - Especificação do caso de uso "Responder perguntas" .....	20
Tabela 9 - Especificação do caso de uso "Visualizar respostas" .....	21

# SUMÁRIO

<b>1. INTRODUÇÃO.....</b>	<b>8</b>
<b>1.1 OBJETIVO GERAL .....</b>	<b>9</b>
<b>1.2 OBJETIVO ESPECÍFICO.....</b>	<b>10</b>
<b>2. DESENVOLVIMENTO .....</b>	<b>11</b>
<b>2.1 REFERENCIAL TEÓRICO.....</b>	<b>11</b>
<b>2.1.1 LÓGICA DE PROGRAMAÇÃO.....</b>	<b>11</b>
<b>2.1.2 DIFICULDADES DE ESTUDO DE LÓGICA DE PROGRAMAÇÃO.....</b>	<b>11</b>
<b>2.1.3 APLICATIVOS EDUCACIONAIS E SEUS BENEFÍCIOS.....</b>	<b>12</b>
<b>2.1.4 VÍDEOS EDUCACIONAIS.....</b>	<b>13</b>
<b>2.1.5 APLICAÇÕES RELACIONADAS.....</b>	<b>14</b>
<b>2.2 METODOLOGIA.....</b>	<b>15</b>
<b>2.2.1 FERRAMENTAS TECNOLÓGICAS.....</b>	<b>15</b>
<b>2.3 RESULTADOS E DISCUSSÃO.....</b>	<b>17</b>
<b>2.3.1 O PROJETO.....</b>	<b>17</b>
<b>2.3.2 DOCUMENTAÇÃO DO SOFTWARE.....</b>	<b>18</b>
<b>2.3.3 REQUISITOS FUNCIONAIS.....</b>	<b>18</b>
<b>2.3.4 REQUISITOS NÃO FUNCIONAIS.....</b>	<b>18</b>
<b>2.3.5 DIAGRAMA DE CASOS DE USO.....</b>	<b>19</b>
<b>2.3.5.1 DESCRIÇÃO DOS ATORES.....</b>	<b>20</b>
<b>2.3.5.3 DESCRIÇÃO DOS CASOS DE USO.....</b>	<b>20</b>
<b>2.3.6 MODELO LÓGICO DO BANCO DE DADOS.....</b>	<b>22</b>
<b>2.3.7 DIAGRAMA DE CLASSES.....</b>	<b>24</b>
<b>2.3.8 DIAGRAMA DE ATIVIDADE.....</b>	<b>27</b>
<b>2.3.9 IMPLEMENTAÇÃO DO APLICATIVO.....</b>	<b>28</b>
<b>2.3.10 INTERFACES.....</b>	<b>29</b>
<b>2.3.10.1 APRESENTAÇÃO DAS INTERFACES.....</b>	<b>30</b>
<b>3. CONCLUSÃO.....</b>	<b>37</b>
<b>REFERENCIAS.....</b>	<b>38</b>

## 1.INTRODUÇÃO

A Educação vem sendo inserida em novos espaços, trazendo uma mudança de assuntos e de valores, sempre pensando em todas os movimentos da sociedade. As universidades e escolas, são como organizações pertencentes e ativas da nossa sociedade, e estimuladoras do conhecimento sistematizado, não poderiam deixar de se adequar a esse novo dinamismo (MARIA, 2006).

Cursos de programação ainda necessitam se adequar a essa nova realidade, já que o aprendizado de linguagens de programação necessita do entendimento de um grupo de conceito e técnicas, bem como o pensamento a respeito de normas, sintaxes e sentidos das linguagens de programação, capacidade de argumento lógico, exato e abstrato e de pensamento algorítmico (PIEIDADE et al., 2019). Os aspectos tornam o estudo um procedimento complicado e árduo para muitos alunos e acadêmicos (MARTINS; MENDES; FIGUEIREDO, 2013; JENKINS, 2002).

Segundo Robins et al. (2003), citado por Cheng (2019) “[...] um dos maiores desafios enfrentados pelos alunos iniciantes em programação era que eles conheciam a sintaxe e a semântica de cada instrução na linguagem de programação, mas não entendiam como combinar as instruções para criar um programa de computador válido. [...]”. Além disso, Deters et al. (2008) dizem que as disciplinas de linguagem de programação são vistas como desafiadoras pelos estudantes, pois são aquelas em que a reprovação e a desistência estão muito presentes.

Os maiores problemas relatados em pesquisas pelos alunos, falam sobre a carência de conhecimento dos estudantes relativo à resolução de problemas, ao pensamento matemático e lógico, ao período aplicado a aprendizagem e dizem, do mesmo modo, a idade dos estudantes nos cursos iniciais (FERREIRA et al., 2017; SILVA et al., 2018).

Salcedo e Idobro (2011) indicam, ainda, o fato de os alunos e acadêmicos usarem em exagero a técnica de decorar os conteúdos que estão sendo estudados, o que no ponto de vista dos escritores essa técnica é fraca para o ensino de lógica de programação, pois necessita de um grande esforço prático e dinâmico. A melhor técnica de estudo para a lógica de programação é feita de forma demorada, progressiva e precisa um treinamento intenso e sempre

utilizando técnicas de concentração (HOLANDA et al., 2018; SILVA et al., 2018).

Programar significa muito mais do que escrever linhas de códigos, com uma coleção de orientações ou códigos de programação soltos, por isso o primeiro obstáculo dos estudantes é entender o que está sendo escrito e qual a função de cada comando na montagem do software (BOSSE; GEROSA, 2017).

Além das adversidades informadas pelos estudantes e acadêmicos no parágrafo anterior, também é possível encontrar pesquisas apontando problemas associados as metodologias didáticas e pedagógicas, bem como relacionados às concepções e conhecimentos dos docentes (GOMES et al., 2008; MOONS; BRACKER, 2013).

Atualmente existe um crescimento na criação de vídeos feitos principalmente para apoiar e acrescentar a função do educador em classe. Esse auxílio é tão significativo que não se entende a recusa do uso desse método por alguns educadores (FIGUEIREDO et al, 1999).

Outro enorme benefício do vídeo é que, um meio audiovisual, necessita de pouquíssimo empenho e afincos por parte do aluno (MORAN, 1995). Ou seja, o aluno obtém os ensinamentos de modo intuitivo.

O trabalho justifica-se por ressaltar a importância de se falar sobre o estudo de lógica de programação e como utilizar ferramentas tecnológicas a favor disso. Os acadêmicos e responsáveis devem estar prontos para utilização de novos métodos de ensino. Assim sendo, o trabalho traz informações sobre as dificuldades relacionadas à aprendizagem de programação, informações que foram trazidas por meio de uma revisão bibliográfica, buscando trazer clareza sobre o assunto com o objetivo de ajudar no estudo e encontrar didáticas diferenciadas junto com a tecnologia para auxiliar e estimular o indivíduo.

## 1.1 OBJETIVO GERAL

O presente trabalho tem como objetivo geral auxiliar as pessoas no ensino de programação de forma dinâmica através do EvoLogic, um protótipo de aplicativo digital desenvolvido e voltado para alunos do curso de informática.

### 1.1.2 OBJETIVO ESPECÍFICO

- Explicar o que é lógica de programação e entender de que maneira é possível auxiliar na aprendizagem.
- Analisar como aplicativos e meios audiovisuais podem ajudar na educação.
- Fazer o levantamento de requisitos para o desenvolvimento do software.
- Documentar a modelagem do software.
- Desenvolver o protótipo do software

## **2.DESENVOLVIMENTO**

### **2.1 REFERENCIAL TEÓRICO**

#### **2.1.1 LÓGICA DE PROGRAMAÇÃO**

A lógica matemática ou figurada surgiu através de Frege, no final do século XX e foi aprimorada no século atual por Whitehead e Bertrand Russell. Ela mirava ultrapassar os problemas de ambiguidades de várias linguagens, por causa do ambiente vago e equivocado das expressões utilizadas e do caráter metafórico que seria capaz de perturbar o rigor lógico do raciocínio. Esses problemas foram solucionados com a criação da linguagem simbólica artificial que usava variáveis assertivas e conectivas (CHAUÍ, 1984).

No ano de 1957 Skinner publica a obra *Verbal Behavior*, com pesquisas minuciosas a respeito do desempenho comportamental no estudo de diversas linguagens. Uma atualização dessa obra, feita por Noam Chomsky, chamada *Syntactic Structures* esclareceu que o é necessário ter conhecimento sobre criatividade na linguagem, embasada em padrões sintáticos. O formalismo dessa gramática possibilitava, no início, que ela fosse utilizada para programar (RUSSELL e NORVIG, 1995).

De acordo com Santos (2006), “O desenvolvimento de algoritmos e o estudo de estruturas de dados devem receber especial atenção na abordagem do tema programação”, mostrando assim a relevância de manter a concentração voltada à métodos ou ferramentas de aprendizagem de programação e de construção de dados.

#### **2.1.2 DIFICULDADES DE ESTUDO DE LÓGICA DE PROGRAMAÇÃO**

Atualmente vivemos em mundo onde tudo está conectado a tecnologia, no qual a conhecimento pode ser obtido por qualquer pessoa e em qualquer localização. Esse projeto, analisa os métodos de aprendizagem tradicionais, em que as Tecnologias da Informação e Comunicação (TIC) são necessários para aumentar o nível da educação. A usabilidade das TIC para aprimorar a aprendizagem dentro das instituições tem sido foco de muitas pesquisas, porque algumas dessas tecnologias têm como objetivo primordial ajudar na capacitação

de alunos (IAHNKE, 2014).

São diversos os motivos que atrapalham no desenvolvimento do pensamento lógico computacional e como resultado o ensino de Lógica de Programação, alguns deles são: ausência de motivação, pois o docente pensa que a matéria é um empecilho complicado de ser superado; o ensino através de instruções, que não instaura uma ligação por entre a teoria e a prática dos conceitos; o aperfeiçoamento do raciocínio lógico (HINTERHOLZ, 2009). Esses fatores têm feito com que os docentes enfrentem desmotivação e acabem sendo reprovados e talvez isso seja um dos motivos para a evasão dos alunos dos cursos direcionados a essa área.

Visando encontrar maneiras novas de se passar os conteúdos sobre lógica de programação, diversos pesquisadores têm mostrado que a utilização de conteúdos audiovisuais no momento de aprendizagem, podem transformar alguns conceitos abstratos em algo de fácil compreensão e são ótimos para atrair o foco do aluno. A utilização de outros métodos de Aprendizagem para ajudar no ensino, possuem as seguintes vantagens: a) As atividades são realizadas de forma mais prazerosa; b) são excelentes desafios; c) detêm dimensão simbólica d) não existe um limite de oportunidades; e) falam de maneira construtiva e relacional. (MACEDO et al. 2005).

Se o conteúdo audiovisual é capaz de ajudar alunos no ensino de lógica de programação, será que um aplicativo eletrônico tem essa aptidão? Esse será o assunto do próximo capítulo.

### 2.1.3 APLICATIVOS EDUCACIONAIS E SEUS BENEFÍCIOS

Pesquisas feitas por Tarouco (2004), Traxler (2005) e Valentim (2009) têm mostrado que as oportunidades e vantagens na utilização das tecnologias em dispositivos móveis (Smartphones, Celulares, E-readers e Tablets) para a aplicação e visualização de tópicos educacionais a qualquer momento e de qualquer lugar.

Segundo as Diretrizes de Políticas de Aprendizagem Móvel (UNESCO, 2013), a comodidade e facilidade de acesso a dispositivos móveis e a alta propagação do uso na comunidade fez com que houvesse um aumento de pessoas que possuem

dispositivos a sua disposição e que sabem como utilizá-lo. Os dispositivos móveis eventualmente são utensílios significativos para contribuir com o crescimento e amplificação do estudo, em especial para docentes com dificuldades de acesso à educação de melhor qualidade, em razão dos motivos geográficos, financeiros e sociais. Segundo a UNESCO (2013, p.9) “a tecnologia móvel não é e nunca será uma solução definitiva no campo da educação, embora seja uma ferramenta poderosa, entre muitos outros, que muitas vezes não são levados em consideração e que podem fornecer suporte pedagógico de formas até então insuspeitadas”.

Algumas das fundamentais vantagens dos dispositivos móveis são muito mais do que disponibilidade, conforto e comunicação (SCHEPMAN et al., 2012). Com dispositivos móveis os estudantes conseguem utilizar aplicativos, meios exclusivos de estudo, possibilidade acessar o navegador, instrumentos de cooperação, redes sociais, entre outros (DUARTE-FILHO; BARBOSA, 2012).

Como visto aplicativos são benéficos no ensino, e é por isso que esse projeto tem como foco a construção de um aplicativo educacional.

#### 2.1.4 VÍDEOS EDUCACIONAIS

A televisão, o computador, o vídeo e o cinema atuam direta e indiretamente tem uma função educacional importante. Transmitem constantemente conhecimentos interpretados, mostram métodos de conduta, demonstram linguagens coloquiais e multimídia, beneficiam alguns fatores em desvantagem de outros (MACHADO, 1988), enfim indicam declarações e mantem comunicações entre diversa partes da sociedade.

Como estudado por Ferrés (1996), um vídeo bem-feito é capaz de ser utilizado para apresentar um novo tópico, além de conseguir estimular a curiosidade e o acesso a novos tópicos. Isso ajuda no desejo de busca dos docentes, para examinar o tema do vídeo e do assunto programático.

A Videoaula, que é uma categoria de exibição de tópicos de forma organizada, exige uma dedicação especial. Essa categoria, que reúne a maior parte dos vídeos didáticos ou educativos, de acordo Moran (1991), a falta uso de conteúdo audiovisual por meio de professores, pode transformar a aprendizagem cansativa e pouco produtiva, pois é apontado que a utilização desses podem fazer com que os docentes



desempenhem funções mais ativas.

Após a apresentação desses pontos é possível perceber que vídeos auxiliam docentes na sua jornada de ensino, e por isso o EvoLogic utilizará isso na sua construção.

### 2.1.5 APLICAÇÕES RELACIONADAS

Foi realizada uma pesquisa com o propósito de descobrir aplicações que contêm similaridade com o aplicativo desenvolvido. Foram usados como base para a pesquisa os aplicativos: DuinoBlocks, que foi criado tendo como finalidade propor e desenvolver um ambiente de programação que utiliza uma linguagem visual, permitindo aos usuários iniciantes na programação desenvolverem suas habilidades programando um dispositivo robótico baseado no hardware Arduino. Segundo o estudo de ALVES (2013), os testes realizados no ambiente mostraram que professores e alunos se sentirão mais confiantes ao trabalhar com a linguagem visual do ambiente ao invés da linguagem textual Wiring, padrão do Arduino (ALVES, 2013).

Há também o Tynker, criado em 2013, é uma criativa plataforma online para ensinar crianças a como programar através de atividades que elas adoram, como jogos e histórias. A plataforma possui mecanismos pré-definidos que são basicamente blocos que representam comandos ou propriedades, que são montados para criar um código. Desse modo, as crianças aprendem o fundamental da programação e construção de código através de uma linguagem visual simples e intuitiva e, conforme vão evoluindo no aprendizado, são apresentadas a desafios mais complexos e até mesmo são introduzidas às linguagens textuais JavaScript e Python. O objetivo desta plataforma é prover para as crianças uma fundação sólida para o modo de pensar computacionalmente, para prepará-las para o futuro (TYNKER, 2017).

## 2.2 METODOLOGIA

Para realizar o levantamento de informações e busca de soluções para a criação do projeto, foi usado como recurso a consulta bibliográfica. O escritor Amaral (2007, p.1), afirma que:

A pesquisa bibliográfica é uma etapa fundamental em todo trabalho científico que influenciará todas as etapas de uma pesquisa, na medida em que der o embasamento teórico em que se baseará o trabalho.

No levantamento da revisão da bibliografia, fez-se uma busca por artigos publicados em periódicos e anais de eventos, através das palavras chaves: lógica de programação, audiovisual na aprendizagem e aplicativos educacionais na plataforma Google acadêmico e na base Scielo, além de pesquisas em livros.

Foi

utilizado metodologias ativas como a aprendizagem entre pares (ou times), também conhecida como *peer instruction* ou *team based learning*, é uma metodologia ativa que incentiva o debate e a reflexão em conjunto.

O desenvolvimento prático do projeto foi realizado após este levantamento de informações e contou com um conjunto de ferramentas, como destacadas a seguir.

### 2.2.1 FERRAMENTAS TECNOLÓGICAS

- Figma: Site e aplicativo para desenvolver e modificar imagens. Utilizado para desenvolver a Interface do Usuário do aplicativo.
- Android Studio: IDE (*Integrated Development Environment*), para aplicação mobile. Foi utilizado no desenvolvimento e integração do projeto Evologic. Foi utilizado pois é uma ferramenta que facilita a criação de aplicativos e de suas interfaces.
- Draw.io: Esse aplicativo foi utilizado para a criação dos diagramas do projeto. Foi escolhido por suas praticidades em criar todos os tipos de diagrama UML.
- Java: Linguagem Orientada a objetos que possibilita o desenvolvimento e execução das operações para o desenvolvimento de

softwares. Foi utilizado dentro do Android Studio com linguagem para programar o aplicativo, e integrar as funções. Essa linguagem foi escolhida por sua facilidade e familiaridade com a mesma pois foi vista durante o curso.

- SQLite: Biblioteca em linguagem C que implementa um banco de dados SQL embutido. Foi utilizado dentro do Android Studio para fazer a criação do banco de dados do projeto e realizar cadastro e busca de informações. Foi utilizado pois já existe uma integração direta junto com o Android Studio e facilita na comunicação entre o código e o banco de dados.

## 2.3 RESULTADOS E DISCUSSÃO

### 2.3.1 O PROJETO

O aplicativo foi desenvolvido para dispositivos móveis devido à popularidade e seu alcance maior a todo o público. Com o objetivo de registrar e desenvolver o projeto, foi empregue a UML (Linguagem de Modelagem Unificada), com o propósito de produzir o diagrama de casos de uso. Os diagramas mostrados, tem o conceito de desenhar e organizar os procedimentos do aplicativo. Depois de realizar o registro, iniciou-se a montagem da UI (Interface do Usuário) feita pelo Photoshop, o local onde foi desenvolvido o estilo das interfaces. Em seguida, iniciou-se a execução do aplicativo na área de desenvolvimento, o Android Studio.

### 2.3.2 DOCUMENTAÇÃO DO SOFTWARE

Segundo Sérgio Amadeu da Silveira (2004, p. 11), a documentação de software é feita pela união de diversos dados técnicos que são capazes de ser retratados e ordenados no formato de diagramas, textos, fluxogramas, entre outros. Segundo o escritor, a sua relevância se dá, pois, a documentação se torna uma prioridade em todos os procedimentos das tarefas que serão executadas no software, transportando consigo uma visibilidade clara e complexa de como o software irá funcionar.

### 2.3.3 REQUISITOS FUNCIONAIS

Tabela 1 - Requisitos Funcionais

Identificador	Nome	Descrição	Prioridade	Dependência
RF01	Criar Usuário	O aluno deve conseguir criar um usuário através de um cadastro simples.	Alta	
RF02	Realizar Login	O sistema deve permitir que o aluno realize o login.	Alta	RF01
RF03	Acessar o conteúdo audiovisual	O aluno deve poder acessar o vídeo integrado ao aplicativo normalmente.	Alta	RF02
RF04	Acessar tela de material necessário para aula	O aluno deve poder acessar a tela que foi integrada para mostrar quais são as ferramentas que serão usadas durante o vídeo.	Médio	RF02
RF05	Responder perguntas	O aluno deve poder responder todas as perguntas.	Alta	RF03
RF06	Acessar respostas das perguntas	O aluno deve conseguir visualizar as respostas corretas para as perguntas respondidas, validadas pelo código do aplicativo.	Médio	RF05

Fonte: O autor (2022)

### 2.3.4 REQUISITOS NÃO FUNCIONAIS

Tabela 2 - Requisitos não Funcionais

Identificador	Descrição
RNF01	O sistema deve ser desenvolvido para dispositivos Android.
RNF02	O sistema deve ter conexão com banco de dados.
RNF03	O sistema deve ter um design simples e de fácil utilização.

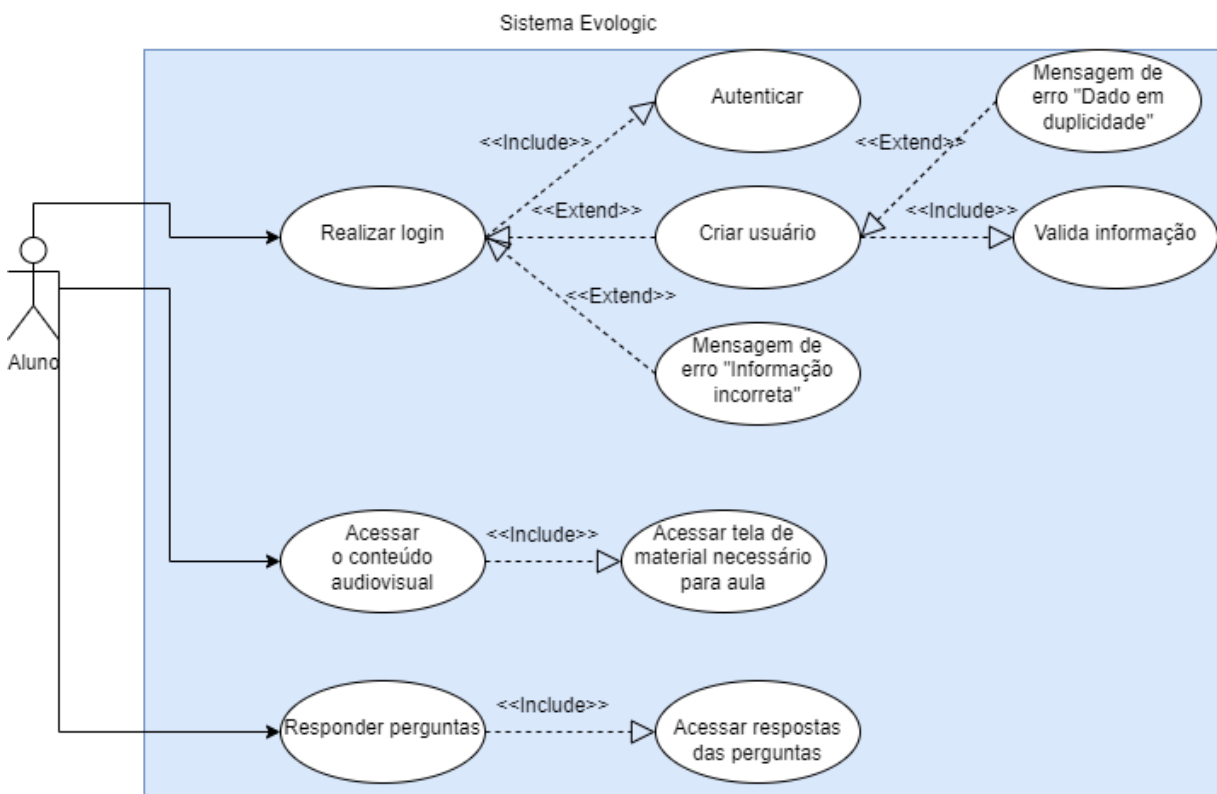
RNF04	Uso de Design responsivo nas interfaces gráficas.
-------	---

Fonte: O autor (2022)

### 2.3.5 DIAGRAMA DE CASOS DE USO

Diagrama de casoS de uso é o retrato das funções externas e observáveis do sistema e dos componentes externos ao sistema e, como eles se relacionam (Bezerra, 2007). O registro UML para relacionamentos de desse tipo é um segmento de reta conectando o ator e o caso de uso (Stadzisz, 2002).

Figura 1 - Diagrama de Casos de Uso



Fonte: O autor (2022)

Só existe o usuário aluno pois, a adição de conteúdo é feita através de versionamento, quando temos novos conteúdos lançamos uma nova versão do aplicativo. Está escolha foi feita por se tratar de um protótipo.

### 2.3.5.1 DESCRIÇÃO DOS ATORES

Tabela 3 - Descrição dos atores

Atores	Descrição
Aluno	Usuário que vai iniciar o aplicativo e assistir as videoaulas.

Fonte: O Autor (2022)

### 2.3.5.3 DESCRIÇÃO DOS CASOS DE USO

Tabela 4 - Especificação do caso de uso "Criar Usuário"

Criar Usuário	
Descrição	Função que realiza a construção do aluno ao sistema.
Atores	Aluno
Pré-condição	1. Pegar as informações necessárias para criação dos usuários
Pós-condição	1. O aluno deve fornecer todas as informações corretamente
Fluxo Principal	1. Na tela inicial o usuário clica em Criar conta, e surgirá um formulário de cadastro. 2. O formulário é preenchido. 3. O sistema valida as informações
Fluxo de Exceção	1. Caso alguma informação esteja em branco o sistema exibe um aviso e o usuário precisa preencher todos os campos para que o caso de uso seja encerrado. 2. Caso o sistema verifique a duplicidade no banco de dados, é exibido um aviso de cadastro já existente.

Fonte: O Autor (2022)

Tabela 5 - Especificação do caso de uso "Realizar Login"

Realizar Login	
Descrição	Ação que permite o acesso ao sistema
Atores	Aluno
Pré-condição	1. Estar cadastrado no sistema
Pós-condição	1. O usuário deve estar apto para utilizar o sistema
Fluxo Principal	1. Na tela inicial o usuário clica em realizar login, o sistema oferece os campos de login e senha. 2. O aluno informa os valores de login e senha. 3. O sistema faz a autenticação dos dados.
Fluxo de Exceção	1. Se o usuário não tiver login cadastrado é necessário criar um login. 2. Caso a informação de login inserida não existir na base de dados ou estiver errada é exibida uma mensagem para o usuário, fazendo com que ele tente inserir novamente

Requisitos relacionados	Criar Usuário
-------------------------	---------------

Fonte: O Autor (2022)

Tabela 6 - Especificação do caso de uso “Assistir Vídeos”

<b>Acessar o conteúdo audiovisual</b>	
Descrição	Função que permite que o aluno assista a videoaula
Atores	Aluno
Pré-condição	1. Iniciar aplicativo 2. Estar conectado à internet.
Pós-condição	Não existe pós-condição
Fluxo Principal	1. Após iniciar o aplicativo, você é levado para uma tela com a videoaula. 2. O aluno pode iniciar a videoaula. 3. Depois de assistir a videoaula, o usuário clica em “concluir aula” e essa informação é salva no banco de dados.
Requisitos relacionados	Realizar Login

Fonte: O Autor (2022)

Tabela 7 - Especificação do caso de uso “Visualizar material necessário para aula”

<b>Acessar tela de material necessário para aula</b>	
Descrição	Função que permite que o aluno visualize quais vão ser as ferramentas utilizadas naquela aula.
Atores	Aluno
Pré-condição	1. Iniciar aplicativo
Pós-condição	Não existe pós-condição
Fluxo Principal	1. Após iniciar o aplicativo, você é levado para uma tela com a videoaula. 2. Na tela da videoaula tem um botão chamado Material da aula, e ele abre uma tela com o material daquela aula
Requisitos relacionados	Realizar Login

Fonte: O Autor (2022)

Tabela 8 - Especificação do caso de uso “Responder perguntas

<b>Responder perguntas</b>	
Descrição	Função que permite que o aluno responda as perguntas sobre a videoaula assistida.
Atores	Aluno
Pré-condição	1. Assistir a videoaula



Pós-condição	Não existe pós-condição
Fluxo Principal	1. Após selecionar a opção de “concluir aula”, você é levado para uma tela que apresenta algumas perguntas. 2. O usuário deve responder todas as perguntas. 3. Após isso clicar em “Finalizar perguntas”
Fluxo de Exceção	1. Se não responder as perguntas aparecerá uma mensagem de erro na tela.
Requisitos relacionados	Assistir vídeos

Fonte: O Autor (2022)

Tabela 9 - Especificação do caso de uso “Visualizar respostas da pergunta”

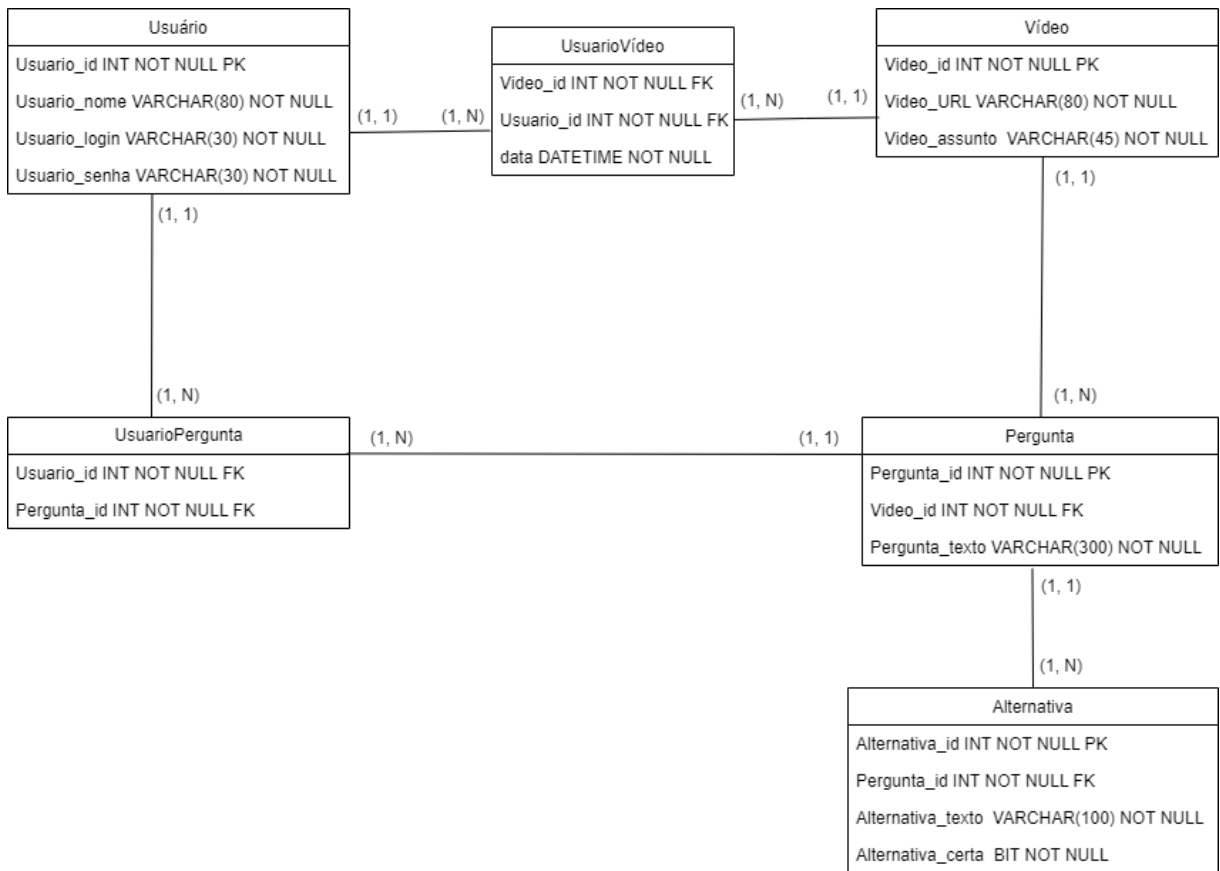
<b>Acessar respostas das perguntas</b>	
Descrição	Função que permite que o aluno visualize quais vão ser as ferramentas utilizadas naquela aula.
Atores	Aluno
Pré-condição	1. Responder perguntas
Pós-condição	Não existe pós-condição
Fluxo Principal	1. Após responder as perguntas, o aluno é levado para uma tela com a quantidade de acertos e as respostas das perguntas com uma breve explicação.
Requisitos relacionados	Responder perguntas

Fonte: O Autor (2022)

### 2.3.6 MODELO LÓGICO DO BANCO DE DADOS

De acordo com o autor MACHADO (2020), o modelo lógico expõe a infraestrutura do banco de dados conforme as possibilidades permitidas pelo modelo conceitual, mas sem considerar, características específicas de um Sistema Gerenciador de Banco de Dados (SGBD). Resultando assim um esquema lógico de dados sob a óptica do modelo conceitual.

Figura 2 - Modelo Lógico do Banco de dados



Fonte: O Autor (2022)

#### Tabela Usuário:

- `usuario_nome`: Armazena o nome do aluno.
- `usuario_login`: Armazena o nome do login do aluno.
- `usuario_senha`: Armazena a senha do usuário para que ele possa se autenticar.

A tabela Usuário representa dentro do aplicativo, o local onde é salvo os dados do usuário.

#### Tabela Vídeo:

- `video_URL`: Armazena a URL do vídeo no youtube.
- `video_assunto`: Armazena o assunto do vídeo.

A tabela Vídeo representa dentro do aplicativo, o local onde é salvo os dados do Vídeo.

#### Tabela UsuarioVideo:

- video\_id: Armazena o último vídeo que o usuário assistiu.
- Usuário\_id: Armazena o usuário.

A tabela UsuarioVideo representa dentro do aplicativo, o local onde é salvo o último vídeo assistido e a referência do usuário que está naquela aula.

#### Tabela Pergunta

- vídeo\_id: Armazena a qual vídeo/assunto aquela pergunta tem relação.
- pergunta\_texto: Armazena o enunciado da pergunta.

A tabela Pergunta representa dentro do aplicativo, o local onde é salvo as informações sobre as perguntas.

#### Tabela alternativa

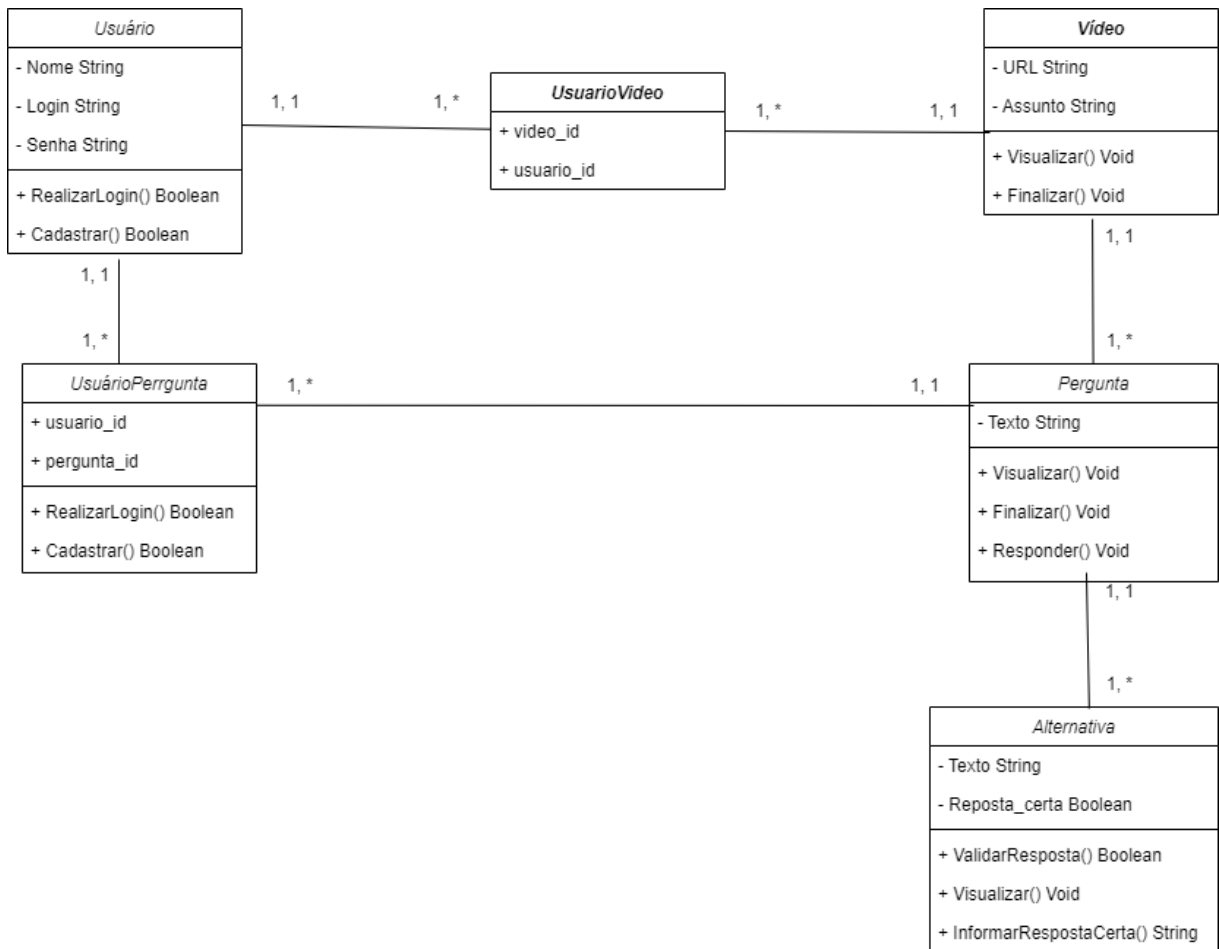
- pergunta\_id: Armazena a qual pergunta aquela alternativa tem relação.
- alternativa\_texto: Armazena o texto das alternativas.
- Alternativa\_certa: Armazena um boolean que se tiver false a resposta está incorreta e se estiver true a resposta está certa.

A tabela alternativa representa dentro do aplicativo, o local onde é salvo as informações sobre as alternativas.

### 2.3.7 DIAGRAMA DE CLASSES

Segundo a abordagem orientada a objetos, os diagramas de classes são os artefatos responsáveis por apresentar a modelação de toda a estrutura estática do sistema. Estes diagramas descrevem a estrutura interna das classes, assim como são estabelecidas as relações existentes entre as diferentes classes (MACHADO, 2020).

Figura 3 - Diagrama de Classe



Fonte: O Autor (2022)

#### Tabela usuário:

- Nome: Armazena o nome do aluno.
- Login: Armazena o nome do login do aluno.
- Senha: Armazena a senha do usuário para que ele possa se autenticar.
- RealizarLogin(): Função que possibilita que o aluno realize o login. Retorna um boolean que se tiver false as informações de login estão incorretas e se estiver true o Login foi realizado com sucesso.
- Cadastrar(): Função que possibilita que o aluno realize o cadastro do usuário dele. Retorna um boolean que se tiver false as informações de login estão incorretas e se estiver true o Login foi realizado com sucesso.

#### Tabela Vídeo:

- video\_URL: Armazena a URL do vídeo no youtube.

- video\_assunto: Armazena o assunto do vídeo.
- Visualizar(): Função desenvolvida para permitir a visualização do vídeo.
- Finalizar(): Função desenvolvida para permitir que o aluno possa finalizar o vídeo.

#### Tabela UsuarioVideo:

- video\_id: Armazena o último vídeo que o usuário assistiu.
- Usuário\_id: Armazena o usuário.

#### Tabela Pergunta

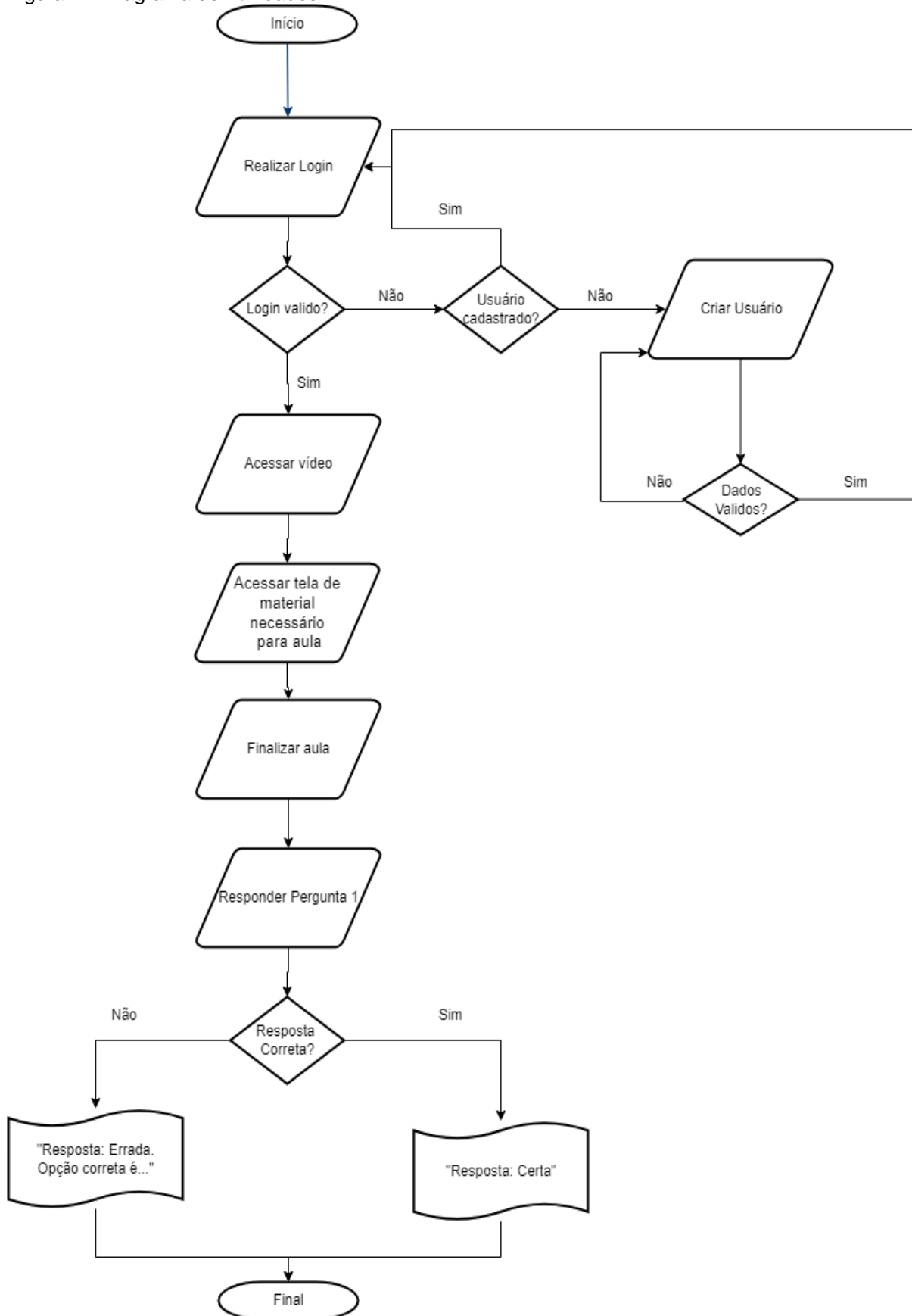
- pergunta\_texto: Armazena o enunciado da pergunta.
- Visualizar(): Função desenvolvida para permitir a visualização da pergunta.
- Finalizar(): Função desenvolvida para permitir que o aluno possa finalizar a pergunta.
- Responder(): Função desenvolvida para permitir que o aluno possa responder a pergunta.

#### Tabela alternativa

- alternativa\_texto: Armazena o texto das alternativas.
- Alternativa\_certa: Armazena um boolean que se tiver false a resposta está incorreta e se estiver true a resposta está certa.
- ValidarResposta(): Função desenvolvida para validar a resposta do aluno. Retorna um boolean que se tiver false a resposta está incorreta e se estiver true a resposta está certa.
- Visualizar(): Função desenvolvida para permitir a visualização da alternativa.
- InformarRespostaCerta(): Função desenvolvida para que o sistema retorne a resposta certa se o usuário selecionar a opção errada.

### 2.3.8 DIAGRAMA DE ATIVIDADE

Figura 4 – Diagrama de Atividades



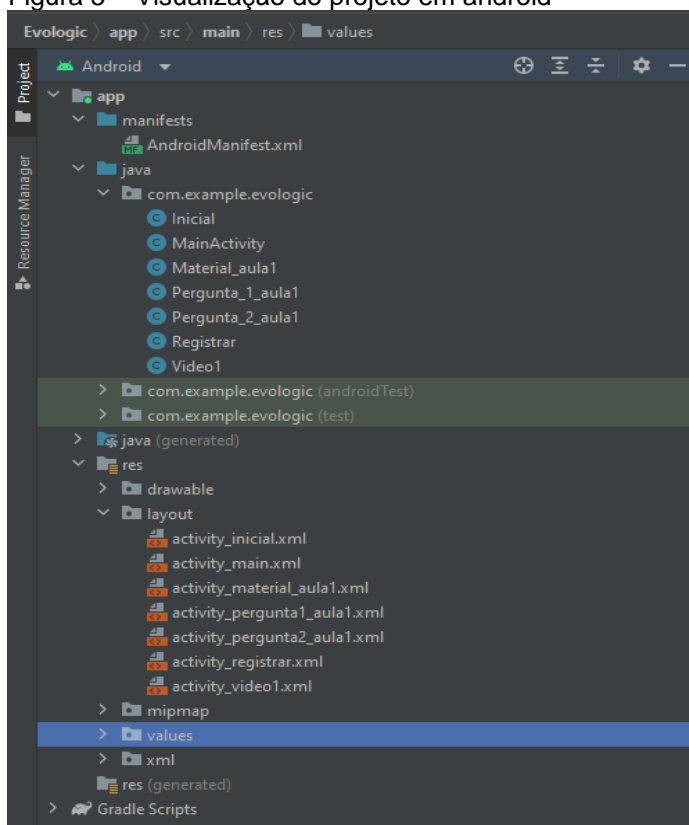
Fonte: O Autor (2022)

O diagrama de atividade foi construído para demonstrar o caminho principal que o aluno irá percorrer. Logo após baixar o aplicativo, o aluno poderá se autenticar, realizando login. Se não autenticado ele pode criar um novo usuário. Se autenticado ele vai poder iniciar o aplicativo e assistir a primeira videoaula, dentro da tela de vídeo aula o sistema permite que ele acesse a tela de material da aula. Após concluir a videoaula o aluno sinaliza que finalizou a aula e é levado as telas de pergunta, o aluno visualizará a pergunta e as alternativas. Após selecionar a pergunta o sistema verifica se a resposta está correta, se estiver ele informa que o aluno acertou, se não estiver ele informa o erro e a alternativa correta.

### 2.3.9 IMPLEMENTAÇÃO DO APLICATIVO

O Evologic foi desenvolvido no Android Studio (versão 2021.3.1), utilizando a linguagem de programação Java e o Banco de Dados SQLite. O Android Studio é uma IDE utilizada para criação de apps Android. Ele exibe os arquivos do projeto na visualização de projetos Android, mostrado na Figura 4. Onde é organizado por módulos para agilizar o acesso aos principais arquivos do projeto.

Figura 5 – Visualização do projeto em android

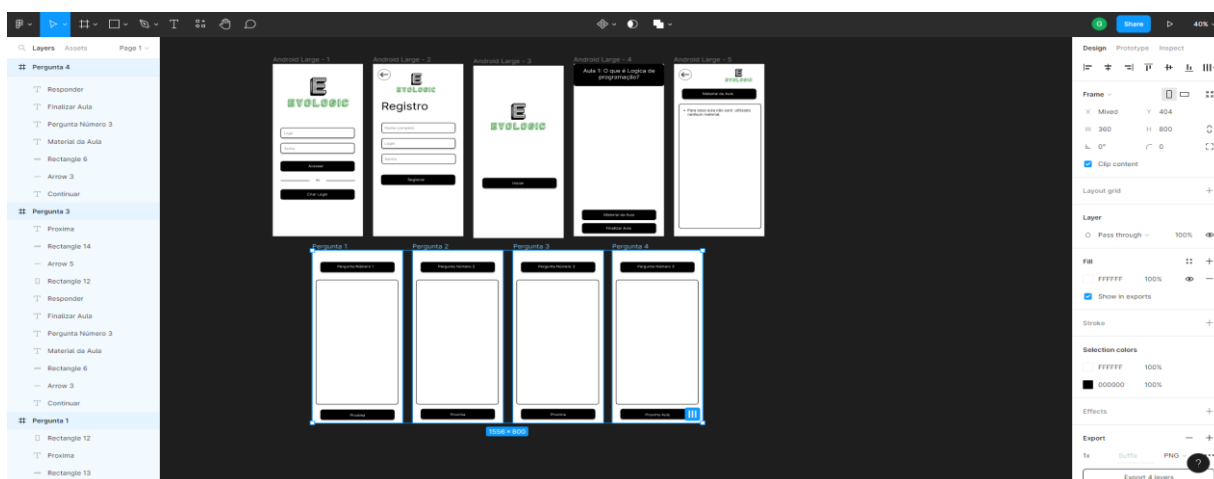


Fonte: O Autor (2022)

## 2.3.10 INTERFACES

O design das interfaces foi criado com o Figma e implementado no Android Studio.

Figura 6 – Ambiente do Figma



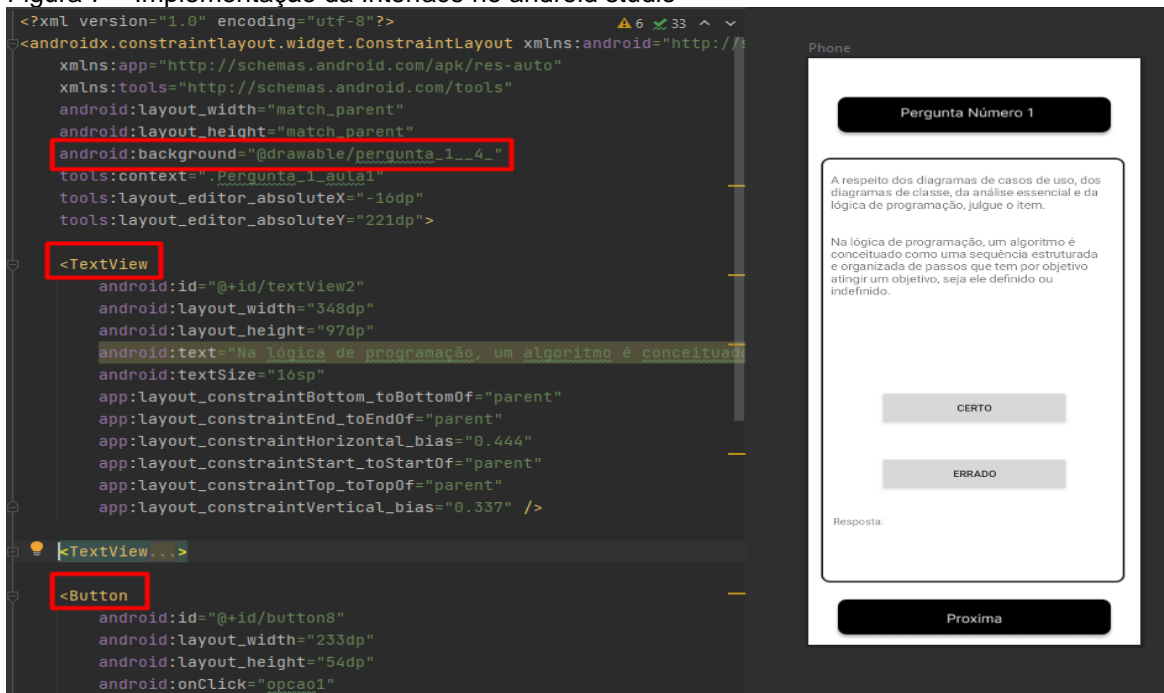
Fonte: O Autor (2022)

No Figma foi construído toda a interface gráfica, segundo as medidas padrões dos smartphones, essa plataforma foi escolhida por sua praticidade em criação de Interfaces de usuários. Após a criação as interfaces foram adicionadas como background (figura 5)

Em seguida foi inserido os campos necessários para a interface criada, como por exemplo: Button, TextView e EditText. Também houve a necessidade de adicionar LinearLayout, para alinhar os campos em uma direção única. Na imagem 5 é possível visualizar a adição dos campos Button e TextView



Figura 7 – Implementação da Interface no android studio

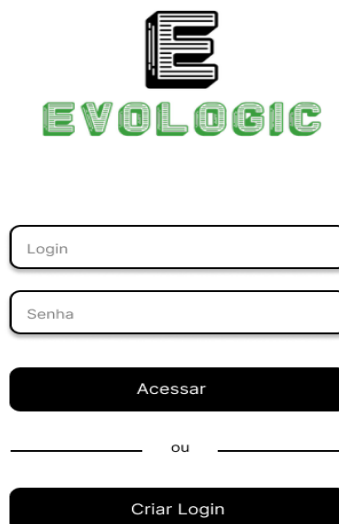


Fonte: O Autor (2022)

### 2.3.10.1 APRESENTAÇÃO DAS INTERFACES

A seguir serão apresentadas todas as telas presentes no aplicativo.

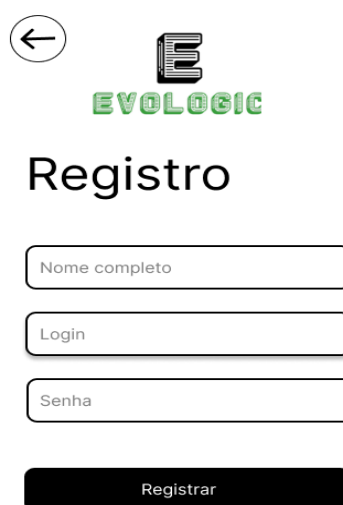
Figura 8 – Tela de Login



Fonte: O Autor (2022)

Ao entrar no aplicativo, a primeira tela a ser apresentada para o usuário é a de login (figura 6). Nela será feita uma verificação para autenticar o usuário. Ao inserir um login e uma senha que não se encontra na base de dados, o sistema irá retornar a mensagem de “Usuário ou senha inválidos” se estiver em branco irá retornar que o campo está vazio. Após inserir um usuário válido o aplicativo redireciona o usuário para a tela de Inicial. Além disso é possível encontrar nessa tela o método de criar usuário.

Figura 9 – Tela de Registrar novo usuário



A tela de registro do aplicativo EVOLOGIC. No topo, há um ícone de seta para trás dentro de um círculo e o logotipo 'EVOLOGIC' em verde. Abaixo, o título 'Registro' é exibido em uma fonte grande e preta. Três campos de entrada de texto são empilhados verticalmente: 'Nome completo', 'Login' e 'Senha'. Cada campo possui uma borda arredondada e um ícone de lupa para pesquisa. Abaixo dos campos, há um botão preto com o texto 'Registrar' em branco.

Fonte: O Autor (2022)

Ao clicar na opção de Criar Login, na tela de login (figura 6) o usuário é redirecionado para a tela “Registrar” (figura 7). Nesta são apresentadas as informações necessárias para o cadastro do aluno no aplicativo. Ao completar os campos e clicar no botão “Registrar” o aplicativo vai retornar a mensagem de “Usuário criado com sucesso”. Caso algum campo ou informação esteja vazio, o aplicativo vai retornar a mensagem “Atenção: Verifique os campos, nenhum pode ficar vazio”. Após a ação de cadastro ser realizada, o usuário será transferido para a tela de Login.

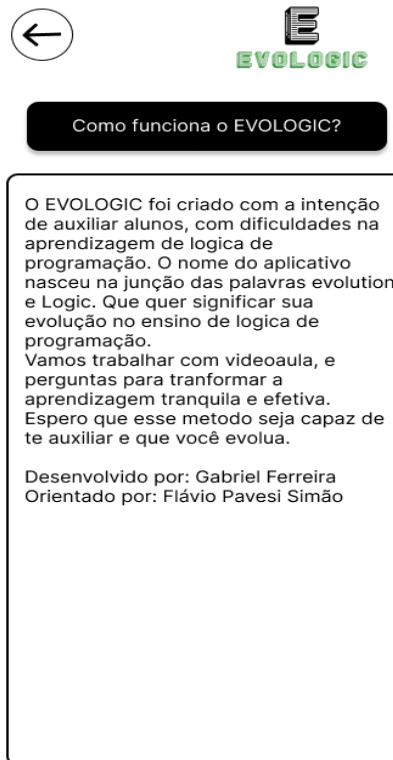
Figura 10 – Tela de inicial



Fonte: O Autor (2022)

Após a autenticação do usuário na tela de login (figura 6), a próxima tela é a do menu. Essa tela é responsável por apresentar o botão de “iniciar” o estudo e ir direto para aula do aluno, através do banco de dados temos a informação de qual aula aquele aluno parou e retornamos a partir dela. Nessa tela também existe o botão de “Como funciona?” que abre uma nova janela que apresenta para o aluno uma breve explicação do funcionamento do aplicativo (figura 9).

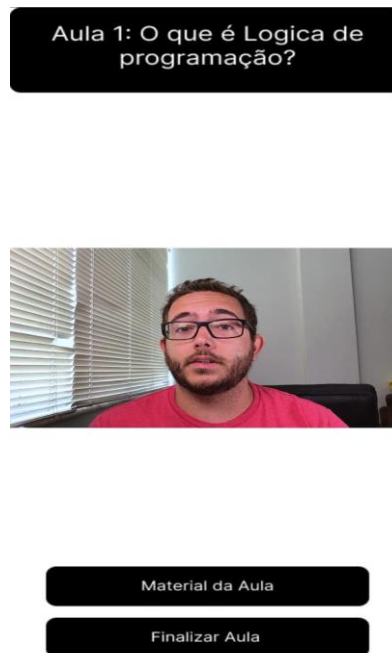
Figura 11 – Tela de como funciona?



Fonte: O Autor (2022)

Para ir à tela de “Como funciona?” sobre o aplicativo basta clicar no botão que se encontra na tela de menu (figura 8). Nela você encontra um resumo sobre o aplicativo, nome do desenvolvedor e orientador.

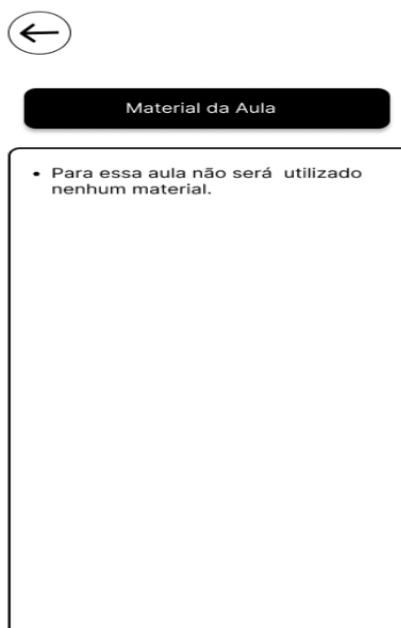
Figura 12 – Tela da vídeoaula



Fonte: O Autor (2022)

Depois de selecionar iniciar na tela de menu (figura 8), você é levado para tela da aula (figura 10), nessa tela você encontrara na parte de cima o tema daquela aula, o vídeo sobre o tema daquela aula, que poderá ser assistido pelo aluno, um botão com o nome “Material da aula” que leva para uma tela que informa os materiais e aplicativos que serão usados naquela aula (figura 11).

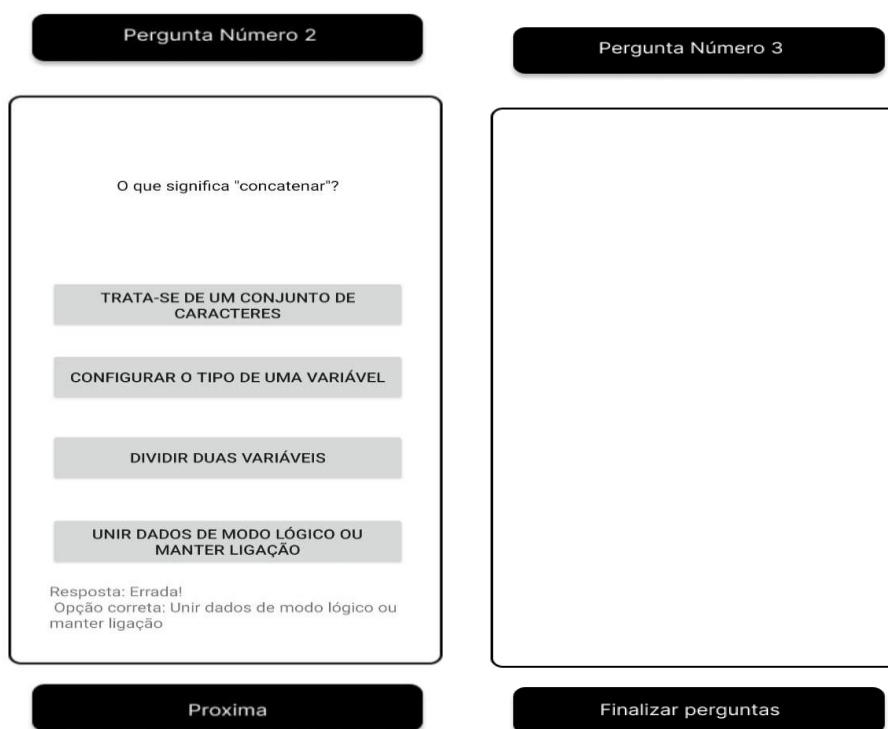
Figura 13 – Tela de Material da aula



Fonte: O Autor (2022)

Para ir à tela de “Material da aula” daquela aula basta clicar no botão que se encontra na tela do vídeo (figura 10). Nela você encontra um resumo sobre o aplicativo, nome da desenvolvedora e orientador.

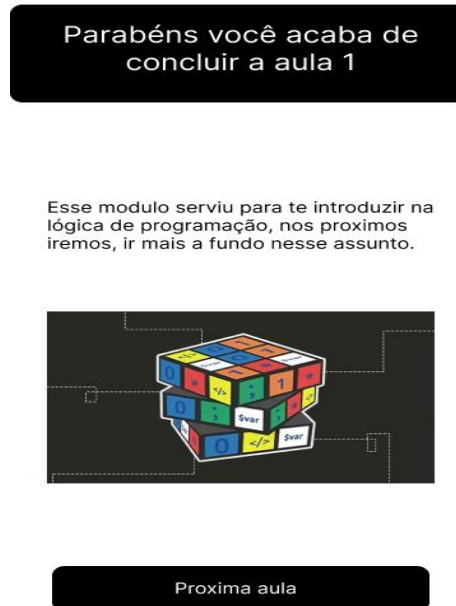
Figura 14 – Tela de Pergunta



Fonte: O Autor (2022)

Ao clicar na opção de finalizar aula, na tela da videoaula (figura 10) o usuário é redirecionado para a tela de Perguntas (figura 12). Nesta tela são apresentadas perguntas sobre o assunto abordado na aula. Ao responder a pergunta ele informa se a resposta está certa ou errada. Se a resposta estiver errada ele informa qual a opção certa. O aplicativo não permite que você passe para a próxima pergunta sem que você responda a atual pergunta. Na última pergunta aparecerá um botão chamado finalizar perguntas, que ao clicar te leva para tela final da aula (Figura 13).

Figura 15 – Tela final da aula



Fonte: O Autor (2022)

Após finalizar as 3 perguntas, e clicar no botão de finalizar perguntas na tela de perguntas (Figura 12). Você é redirecionado para a tela final de aula, onde parabeniza pela conclusão da aula e existe um botão que te leva para a próxima aula.

### 3. CONSIDERAÇÕES FINAIS

O desenvolvimento teórico do presente estudo nos mostrou a importância que os aplicativos têm como ferramenta educacional contribuindo na evolução da aprendizagem do aluno. Trazendo um debate sobre como é importante que os docentes tenham um preparo para utilizar tais ferramentas de forma correta e buscar trazer, sem medo, os aplicativos para o meio educacional com o objetivo de ensinar e ajudar as pessoas com dificuldades de uma forma mais dinâmica e fora dos padrões já estabelecidos, promovendo assim um avanço na área educacional.

Porém antes de desenvolver um aplicativo com esse objetivo é preciso que o criador tenha um conhecimento sobre lógica de programação e como os aplicativos podem ajudar. Para que assim a educação possa utilizar da melhor forma os benefícios que a tecnologia pode trazer, não somente para as salas de aula, mas também em qualquer ambiente.

Foi apresentada a documentação do projeto junto com as telas finais. Todo o projeto encontra-se implementado, porém, faltou ser testado com o público para comprovar de fato a sua eficácia no auxílio ao desenvolvimento de lógica de programação.

Apesar de já existirem aplicativos e pesquisas sobre os benefícios da aprendizagem de lógica de programação, ainda existe a necessidade de se falar sobre o tema, pois muitas pessoas ainda acreditam que essa disciplina é mais complicada do que a realidade. O aplicativo tem potencial para ser utilizado como uma ferramenta educacional, pois ele dá a liberdade de trabalhar com vídeos e atividades diversas.

O sistema apresentado deverá ser testado futuramente para comprovar sua eficácia. O plano futuro para esse projeto é trazer novas funcionalidades como chat de dúvida, geração de resultado dos alunos para o professor, envio de feedbacks das respostas e alterar o formato de novos conteúdos para que um administrador seja capaz de adicionar eles. Para que exista mais opções de trabalhar de formas diferentes a dificuldade do aluno em linguagem de programação e deixar disponível para download no Play Store, loja oficial de aplicativos Android e mais futuramente desenvolver uma versão para IOS.



## REFERENCIAS

ALVES, R. M.; SAMPAIO, F. F. Duinoblocks: desenho e implementação de um local de programação através de imagem para robótica educacional baseado no hardware arduino. In: CONGRESSO DE TECNOLOGIA NA EDUCAÇÃO, 3., 2014. Anais. . . [S.l.: s.n.], 2014. p. 11–20.

BEZERRA, Eduardo. Princípios de análise e Projeto de sistemas com UML. 2. ed., Rio de Janeiro: Elsevier, 2007.

BOSSE, Yorah; GEROSA, Marco. Why is learning programming so complicated: patterns of difficulties related to teaching programming logic at the intermediate level. ACM SIGSOFT Software Engineering Notes, v. 41, n. 6, 2017, p. 1-6.

CHAUÍ, Marilena e outros. Primeira filosofia; lições introdutórias. São Paulo: Brasiliense, 1984.

COELHO, Hilda Simone. Documentação de software: uma necessidade. Texto Livre: linguagem e tecnologia, v. 2, n. 1, p. 17-21, 2009.

DIAS, C. D. Pensamento computacional: uma relação de proximidade com a matemática e o raciocínio lógico. 2016. 25 p. Dissertação (Licenciatura em Computação) - Universidade Federal de Juiz de Fora, 2016.

FERRES, Joan. Vídeo e Educação. 2 a ed., Porto Alegre, Artes Médicas, 1996.

FIGUEIREDO, V. F. et al. Mídia & Educação. Rio de Janeiro: Gryphus, 1999.

GOMES, Anabela; HENRIQUES, Joana; MENDES, António. Proposta para auxiliar alunos com problemas no ensino inicial de programação. Educação & Tecnologias, v. 1, n. 1, 2008, p. 93-103.

HOLANDA, Wallace; COUTINHO, Jarbele; FONTES, Laysa. Metodologia para apoiar o ensino de programação inicial. In: Congresso brasileiro de tecnologia na aprendizagem, 7., 2018. Anais [...] Porto Alegre: SBC, 2018. p. 699-708.

HINTERHOLZ, OSWALDO. (2009). Tepequém: uma nova ferramenta para o ensino de algoritmos nos cursos superiores em computação. In XVII-Anais do Workshop sobre Educação em Informática.

IAHNKE, Silvana Letícia Pires; BOTELHO, Silvia Silva da Costa; FERREIRA, André Luis Andrejew. (2014). COLMEIAS: A Integração das Aprendizagens Móvel e Colaborativa para Potencializar a Aprendizagem Significativa. Renote, Porto Alegre, v.12, n.2, 2014.

MARIA, Ana e Gabriela. "A tecnologia renovando o processo educativo". Núcleo de Tecnologia Educacional de Barreiras, BA, 2006. Disponível em: [http://edutec.net/Textos/Alia/PROINFO/prf\\_txtie16.htm](http://edutec.net/Textos/Alia/PROINFO/prf_txtie16.htm). Acesso em: 10 setembro 2022.

MACHADO, Arlindo. A Arte do vídeo. São Paulo, Brasiliense, 1988

MACHADO, Felipe Nery Rodrigues. Banco de Dados–Projeto e Implementação. Saraiva Educação SA, 2020.

MOONS, Jan; BACKER, Carlos. Pilot work and evaluation of a participatory learning site for early programming using cognitive load theory and constructivism. *technology and education*, v. 60, n. 1, 2013. p. 368-384.

MORAN, José M. O vídeo na sala de aula. *Comunicação e Educação*, São Paulo, (2): 27 a 35. jan./abr, 1995.

MORAN, José M. Como ver Televisão. São Paulo, Paulinas, 1991.

SILVA, Walquiria et al. Dificuldades dos alunos nas aulas de programação na faculdade. *Diversitas Journal*, v. 3, n. 3, 2018, p. 761-770.

RUSSELL Stuart J. e NORVIG Peter. *Artificial Intelligence: A Modern Approach*. New Jersey: Prentice Hall, 1995.

SALCEDO, Sebastian; IDOBRO, Ana. New technologies and ways to teach programming languages using robotics scribbler and Alice. In: *ASEE/IEEE frontiers in education conference*, 41., 2011. Proceedings [...]. IEEE: Rapid City, 2011. p. F4G1-F4G-6.

SCHEPMAN, A.; RODWAY, P.; BEATTIE, C.; LAMBERT, J. An observational study of undergraduate students' adoption of (mobile) note-taking software. *Human Behavior*, v. 27, n. 3, p. 308 – 317, 2012.

SOAD, G.; DUARTE-FILHO, N. F.; BARBOSA, E. F. Uma contribuição ao estabelecimento de características de qualidade para aplicações educacionais móveis. In: *XIV Simpósio Brasileiro de Qualidade de Software (SBQS)*. [S.l.: s.n.], 2015. p. 165– 179.

STADZISZ, Paulo César. Projeto de Software Usando UML. Disponível em: . Acesso em: 19 outubro 2022.

TAROUCO, L. M. R. et al. *Objetos de Aprendizagem para M-Learning*. 2004. Disponível em: [http://www.cinted.ufrgs.br/CESTA/objetosdeaprendizagem\\_sucesu.pdf](http://www.cinted.ufrgs.br/CESTA/objetosdeaprendizagem_sucesu.pdf). Acesso em: 23 de setembro de 2022.

TRAXLER, J. Defining, discussing and evaluating mobile learning: The moving finger writes and having writ. IN: *The International Review of Research in Open and Distance Learning*. 2007 Disponível em: <http://www.irrodl.org/index.php/irrodl/article/view/346>. Acesso em 15 de julho de 2022.

TYNKER. Tynker - coding for kids. Disponível em: <http://www.irrodl.org/index.php/irrodl/article/view/420>. Acesso em: 20 setembro 2022

UNESCO. Policy Guidelines for Mobile Learning. Disponível em: <http://unesdoc.unesco.org/images/0021/002196/219641E.pdf>. Acesso em: 10 de junho de 2022.

VALENTIM, H. Para uma Compreensão do Mobile Learning: Reflexão sobre a utilidade das tecnologias móveis na aprendizagem informal e para a construção de ambientes pessoais de aprendizagem. Lisboa. Dissertação de Mestrado. 2009. Disponível em: [http://www.hugovalentim.com/sites/default/.../Hugo\\_Valentim\\_M-Learning.pdf](http://www.hugovalentim.com/sites/default/.../Hugo_Valentim_M-Learning.pdf). Acesso em: 12 de junho de 2022.