



ANTONIO MENEGHETTI FACULDADE
Curso de Graduação de Sistemas de Informação

HENRIQUE ROSA

DESENVOLVIMENTO DE UM APLICATIVO MOBILE PARA
ATIVIDADES VOLTADAS AO PENSAMENTO COMPUTACIONAL E
COMPUTAÇÃO DESPLUGADA

Restinga Seca

2020



HENRIQUE ROSA

**DESENVOLVIMENTO DE UM APLICATIVO MOBILE PARA
ATIVIDADES VOLTADAS AO PENSAMENTO COMPUTACIONAL E
COMPUTAÇÃO DESPLUGADA**

Trabalho de Conclusão do Curso apresentado como requisito para obtenção do grau em Bacharel em Sistemas de Informação da Antonio Meneghetti Faculdade.

Orientador: Prof. Dr. Felipe Becker Nunes

HENRIQUE ROSA

**DESENVOLVIMENTO DE UM APLICATIVO MOBILE PARA
ATIVIDADES VOLTADAS AO PENSAMENTO COMPUTACIONAL E
COMPUTAÇÃO DESPLUGADA**

Trabalho de Conclusão do Curso apresentado como requisito parcial para obtenção do grau em Bacharel em Sistemas de Informação da Antonio Meneghetti Faculdade.

Banca Examinadora:

Orientador (a): _____

Prof. Dr. Felipe Becker Nunes
Antonio Meneghetti Faculdade - AMF

Membro: _____

Prof. Dr. Jonas Bulegon Gassen
Antonio Meneghetti Faculdade - AMF

Membro: _____

Prof. Ms. Samuel Vizzotto
Antonio Meneghetti Faculdade - AMF

RESTINGA SECA/RS
2020

RESUMO

Diversas iniciativas e pesquisas estão sendo abordadas para a adoção de práticas que possibilitem o ensino de programação e lógica para estimular o aprendizado do Pensamento Computacional. Adjunto ao uso desta abordagem está a Computação Desplugada, que permite a criação de atividades offline e sua integração com demais recursos on-line posteriormente. A partir destes pontos de pesquisa, o presente trabalho buscou propor uma aplicação que proporcione um ambiente de ensino na área do Pensamento Computacional, tendo o intuito de usar a Computação Desplugada, onde o professor possa disponibilizar um aplicativo que reconheça imagens de algoritmos montados de forma física, aprimorando a experiência no ensino e aprendizado de algoritmos. Diante desse objetivo foram elencadas especificações para o desenvolvimento da aplicação, além de testes dos algoritmos de extração de texto de imagem. O resultado desse trabalho consiste no desenvolvimento de uma aplicação mobile, o qual foi apresentado suas telas e funcionalidades principais da extração de texto das imagens.

Palavras-chave: Pensamento Computacional, Computação Desplugada, Ensino, Programação, Algoritmos.

ABSTRACT

Several initiatives and research are being approached for the adoption of practices that enable the teaching of programming and logic, in order to stimulate Computational Thinking. Along with the use of this approach, also is Unplugged Computing, which allows the creation of offline activities and their integration with other online resources. From these research points, the present work sought to propose an application that provides a teaching environment in the area of Computational Thinking, with the intention of using Unplugged Computing, where the teacher can provide an application that recognizes images of algorithms developed in a physical environment, improving the experience in teaching and learning of algorithms. In view of this objective, specifications for the development of the application were listed, in addition to testing the text image extraction algorithms. The result of this work is the development of a mobile application, which were presented its main screens and features for extracting text from images.

Keywords: Computational Thinking, Unplugged Computing, Teaching, Programming, Algorithms.

LISTA DE FIGURAS

Figura 1: Estágios Pensamento Computacional	15
Figura 2: Eixos da Computação	18
Figura 3: Exemplo do Teste de Pensamento Computacional	23
Figura 4: Visão Geral do Projeto	26
Figura 5: Diagrama de Atividade para o Fluxo Base	36
Figura 6: Diagrama de Atividade para Cadastrar Turma	37
Figura 7: Diagrama de Atividade Usuário de Vincular a Turma Através do Código	38
Figura 8: Modelo de dados do aplicativo	39
Figura 9: Tela Login	44
Figura 10: Tela Cadastro de Usuário	45
Figura 11: Tela Listagem de Atividades	46
Figura 12: Tela Listagem de Turmas.....	47
Figura 13: Tela Castro de Turma.....	48
Figura 14: Tela Listagem de Atividades Vinculadas a Turma	49
Figura 15: Tela Cadastro de Atividade.....	50
Figura 16: Exemplo para Teste de Imagem com Muito Texto.....	52
Figura 17: Exemplo para Teste de Imagem com Pouco Texto.....	55
Figura 18: Exemplo para Teste de Imagem com Texto Escrito a Mão	56

LISTA DE TABELAS

Tabela 1: Resultados dos Testes de Imagem com Muito Texto	52
Tabela 2: Resultados dos Testes de Imagem com Pouca Texto	55
Tabela 3: Resultados dos Testes de Imagem com Texto Escrito a Mão	56

LISTA DE ABREVIATURAS

- PC – Pensamento Computacional;
- CD – Computação Desplugada;
- SBC - Sociedade Brasileira de Computação;
- PBL - Problem Based Learning;
- OCR - Optical Character Recognition.

SUMÁRIO

RESUMO	4
ABSTRACT	5
LISTA DE FIGURAS	6
LISTA DE TABELAS	7
LISTA DE ABREVIATURAS	8
1 INTRODUÇÃO	14
1.1 JUSTIFICATIVA	17
1.2 OBJETIVOS.....	19
2 REFERENCIAL TEÓRICO	20
2.1 PENSAMENTO COMPUTACIONAL.....	20
2.2 COMPUTAÇÃO DESPLUGADA	22
2.3 TRABALHO RELACIONADOS	23
3 METODOLOGIA	26
3.1 ALGORITMOS DE EXTRAÇÃO DE TEXTO DE IMAGEM	28
3.2 ALGORITMOS DE EXTRAÇÃO DE TEXTO DE IMAGEM	28
4 DESENVOLVIMENTO	30
4.1 REQUISITOS FUNCIONAIS E NÃO FUNCIONAIS	30
4.2 REGRAS DE NEGÓCIO	30
4.3 USER STORY	31
4.3.1 CADASTRAR TURMA	31
4.3.2 VINCULAR USUÁRIO POR E-MAIL	33
4.3.3 RESPONDER ATIVIDADE.....	34
4.4 DIAGRAMA DE ATIVIDADES.....	35
4.4.1 FLUXO BASE.....	36
4.4.2 CADASTRAR TURMA	37
4.4.3 USUÁRIO SE VINCULAR A TURMA ATRAVÉS DO CÓDIGO	38
4.5 MODELO LÓGICO	39
4.6 FERRAMENTAS	41
4.6.1 SERVER-SIDE.....	41
4.6.2 CLIENT-SIDE.....	42
5 RESULTADOS E DISCUSSÕES	43
5.1 TELAS.....	43
5.1.1 LOGIN.....	43

5.1.2	CADASTRO DE USUÁRIO	44
5.1.3	LISTAGEM DE ATIVIDADES	45
5.1.4	LISTAGEM DE TURMAS	46
5.1.5	CADASTRO DE TURMA	47
5.1.6	LISTAGEM DE ATIVIDADES RELACIONADAS A TURMA	48
5.1.7	CADASTRO DE ATIVIDADE	49
5.2	TESTES ALGORITMOS EXTRAÇÃO DE TEXTO DE IMAGENS	50
5.2.1	RESULTADOS DA IMAGEM COM MUITO TEXTO	51
5.2.2	RESULTADOS TESTE DA IMAGEM COM POUCO TEXTO	54
5.2.3	RESULTADOS TESTE DA IMAGEM COM TEXTO ESCRITO A MÃO.....	56
6	CONSIDERAÇÕES FINAIS.....	58
6.1	TRABALHOS FUTUROS	58
	REFERÊNCIAS	60

1 INTRODUÇÃO

Os algoritmos são métodos computacionais de uso frequente no cotidiano, tratando-se de um conjunto de instruções utilizadas para resolver uma tarefa. As pessoas, independente de faixa etária, trabalham com algoritmos no seu dia-a-dia, por exemplo, ao delinear um caminho para se chegar à escola e, por vezes, não são exercitadas a pensar dessa forma, assim, acabam fazendo seus algoritmos cotidianos sem procurar refletir sobre o que está realizando, nem buscar melhores maneiras de concluir. A Computação trabalha com algoritmos e procura, na medida do possível, por bons métodos para realizar tarefas. Essa área apresenta princípios e habilidades que, se trabalhada com os estudantes desde cedo, podem contribuir para o exercício da lógica e resolução de problemas (COSTA, 2012).

O uso de tecnologias para resolver problemas cotidianos está cada vez mais enraizado no nosso dia-a-dia, um exemplo é o uso de uma calculadora para resolver uma equação matemática. Diante disso, diversas técnicas estão sendo estudadas para implementação do ensino da computação no ensino médio e fundamental.

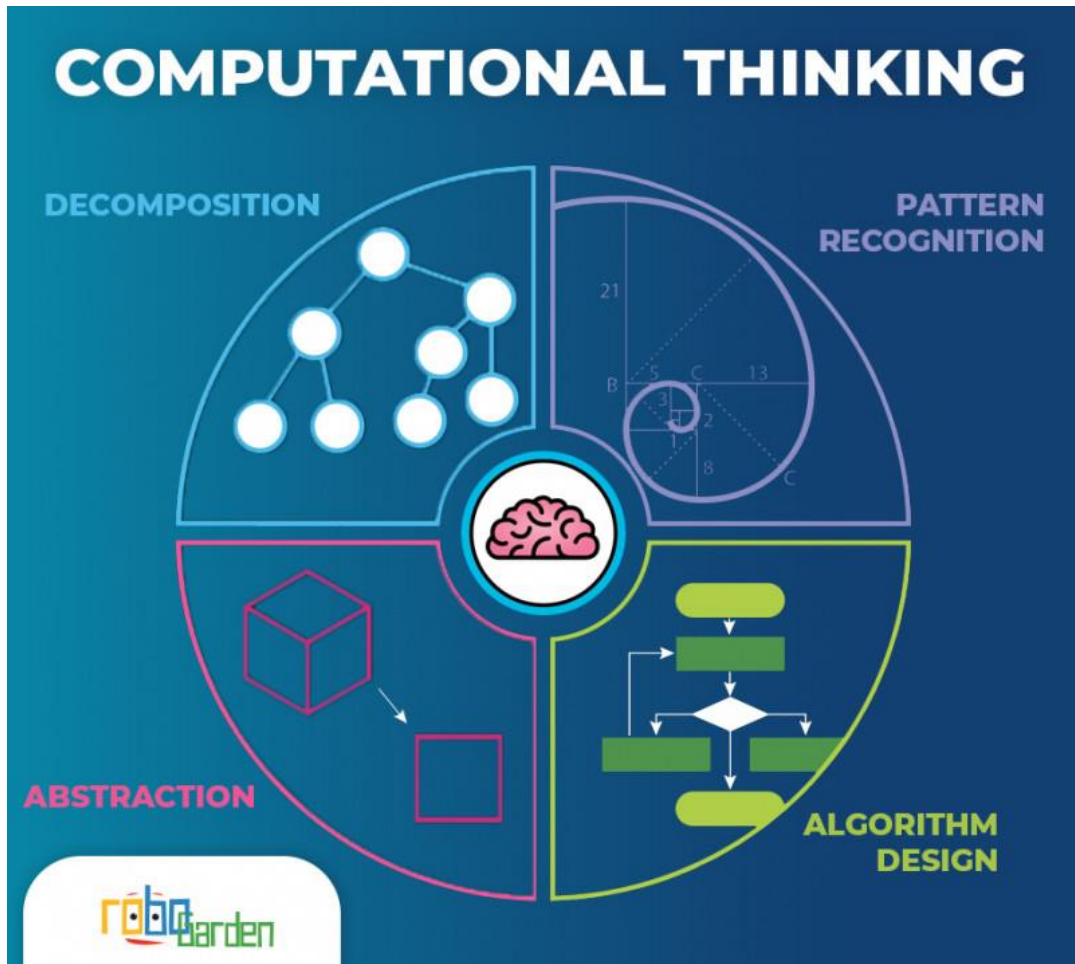
A introdução de conceitos de Computação, enquanto ciência, na educação básica é de fundamental importância e relevância (CSTA, 2011). No entanto, seu ensino, no Brasil, acontece apenas nos âmbitos da graduação e pós-graduação. O ACM Model Curriculum for K-12 Computer Science (CSTA, 2011) defende que é necessário o desenvolvimento de habilidades computacionais na Educação Básica, no sentido da Ciência da Computação ser importante intelectualmente, promover múltiplos caminhos profissionais futuros, desenvolver a capacidade de resolver problemas, apoiar e relacionar-se com outras ciências e motivar os estudantes (FRANÇA & BEZERRA, 2013).

O assunto está cada vez mais recorrente e atual, visto que são vários os benefícios de alunos aprenderem tal ciência, tais como ter uma capacidade de resolver problemas com mais facilidade, levando tal prática para outros segmentos do estudo e vida. Ainda, segundo Freire, Coutinho, Silva e Lima (2019), dentre diversos benefícios, está a leitura, interpretação de texto e matemática, dentre outros.

É diante deste contexto que destaca-se o Pensamento computacional (PC), que vem sendo considerado a habilidade do século XXI, pois atualmente estamos fortemente influenciados pela computação (Yadav et al., 2017), e essa habilidade revela-se como instrumento para favorecer a compreensão do mundo, refletindo naturalmente na produtividade e na criatividade humana. Nesse sentido, muitas iniciativas estão sendo realizadas para ensinar PC para diversos públicos, muitas vezes sendo conduzidas em espaços formais de aprendizagem como escolas e universidades (ORTIZ, MOREIRA & PEREIRA, 2018).

Na Figura 1 podemos ver que o Pensamento Computacional é dividido em 4 partes, são elas: Decomposição (*Decomposition*), Reconhecimento de Padrões (*Pattern Recognition*), Abstração (*Abstraction*) e Algoritmos (*Algorithm Design*).

Figura 1: Estágios Pensamento Computacional



Fonte: <https://robogarden.ca/blog/the-possibilities-are-endless-with-computational-thinking>

A Decomposição trata de pegar um problema e, de certa maneira, quebrá-lo em problemas menores, ou tarefas, para uma resolução mais fácil e assertiva. Após é feito o Reconhecimento de Padrões, que trata de reconhecer os padrões para as resoluções das tarefas. Depois vem a Abstração, que nada mais é que ignorar certos detalhes irrelevantes para focar no problema em si, assim procurando uma solução que se aplique a vários problemas. Por último é desenvolvida uma solução, usando uma série de etapas para a resolução das tarefas, isso é chamado de Algoritmo.

“Também no contexto do estímulo ao Pensamento Computacional, a utilização da metodologia de Computação Desplugada (CD) para adaptação no ensino de conteúdo tem sido relatada como uma metodologia com bastante aceitação por parte dos alunos, por ser uma estratégia lúdica se caracterizando como um método de ensino de computação sem o uso do computador” (BELL et al., 2011).

“Assim, mesmo em escolas que não dispõe de equipamentos, o Pensamento Computacional pode ser estimulado através do uso dessa técnica” (SOUZA & LOPES, 2018). A Computação Desplugada utiliza conceitos das Ciências da Computação sem o uso de computadores ou outro meio digital. Assim, não é preciso ter uma infraestrutura preparada para a aplicação de tal conceito.

Com base na realidade de escolas brasileiras, onde não há um computador para cada aluno, surge a necessidade de sistemas mais inclusivos, no sentido de que possibilita que qualquer professor possa trabalhar PC com seus alunos. Assim, podemos destacar aplicativos de celular ou aplicativos usados de forma offline em computadores, em que é aplicada a CD, ou seja, não necessariamente precise estar conectado à internet para que o aluno consiga trabalhar conceitos básicos de programação.

Um exemplo que podemos abordar é o uso do Scratch em sua versão offline¹, que é um software desenvolvido para ensinar programação para crianças. Segundo Araujo, Rodrigues, Silva e Soares (2015), o Scratch foi criado pelo Media Lab do Instituto de Tecnologia de Massachusetts (MIT) e tem como público alvo as crianças com faixa etária a partir de oito anos. Trata-se de um software interativo que permite a criação de histórias, jogos e animações por meio da manipulação de blocos dinâmicos. O Scratch foi desenvolvido para ensinar programação para quem não tem familiaridade com ela. É um exemplo de uso de PC enquanto online, porém offline é uma lacuna que vem sendo explorada por diversas pesquisas.

O Scratch é um exemplo de uma ferramenta para o ensino de programação, porém não há muitas ferramentas similares disponíveis. Um ponto a ser explorado, também, é a construção de algoritmos de forma física, por meio de peças, cartas, etc. Esse ponto pode ser relacionado com o uso de um software que possa trabalhar de maneira online e offline para captura das atividades realizadas pelos alunos, o que pode facilitar o ensino e correção dessas atividades aplicadas pelo docente.

Com a necessidade de sistemas mais inclusivos cada vez mais presente e necessária nas escolas brasileiras, foi, então, elaborado esse projeto, que se tornou um aplicativo mobile, que

¹ Scratch Offline. Disponível em: <https://scratch.mit.edu/download>

visa, tanto facilitar o trabalho do professor com a gestão de sua turma e atividades relacionadas a programação.

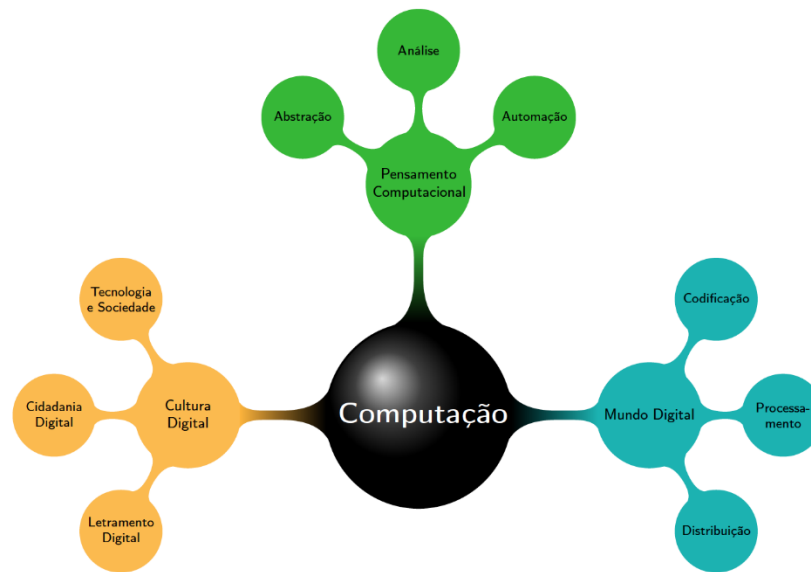
1.1 JUSTIFICATIVA

“O Pensamento Computacional é compreendido como um processo de resolução de problemas, projeto de sistemas e compreensão do comportamento humano norteados por conceitos fundamentais da Ciência da Computação” (Wing, 2006). No decorrer dessa última década, surgiram iniciativas para promover o PC, essas iniciativas partem, principalmente, através de trabalhos de pesquisa e extensão das universidades e faculdades (ARAÚJO, ANDRADE & GUERRERO, 2016). A SBC (Sociedade brasileira de Computação) entende como crucial o ensino de computação na educação básica e busca meios de inclusão e ensino dessa ciência para os alunos.

O desenvolvimento do PC em alunos por meio do ensino de programação demonstra resultados positivos. Brennan e Resnick (2012) mostram que uma abordagem baseada em PC pode ser utilizada no aprendizado de programação, principalmente para jovens e usando ferramentas que motivem e estimulem a criatividade. Para Wing (2006), “PC é baseado em abstração e decomposição de problemas ou sistemas complexos, habilidades altamente recomendadas para alunos de programação” (ZANETTI, BORGES & RICARTE, 2016).

Pensando na realidade das escolas brasileiras, há uma necessidade da aplicação do PC de forma a não utilizar aparelhos eletrônicos, desplugada, tomando como parâmetro a pouca, ou nenhuma, infraestrutura existente, principalmente em escolas públicas e de regiões menos favorecidas (BRASIL, 2017). Assim levando o conhecimento sobre Ciência da Computação para locais onde essa realidade ainda não existe (SANTOS et al., 2016).

“Também há relatos por docentes uma preferência por atividades desplugadas por se sentirem mais confiantes em sua execução” (Rodrigues 2018), já que a sua aplicação plugada demanda do educador conhecimentos técnicos referentes ao uso dos dispositivos eletrônicos (SOUZA & NUNES, 2018). Ainda no contexto da SBC, ela destaca o PC como um dos eixos para ensino de computação na educação básica, conforme visto na Figura 2.

Figura 2: Eixos da Computação

Fonte: SBC (2019).

A SBC (2019) destaca o PC como à capacidade de compreender, definir, modelar, comparar, solucionar, automatizar e analisar problemas (e soluções) de forma metódica e sistemática, através da construção de algoritmos. Apesar de ser um termo recente, vem sendo considerado como um dos pilares fundamentais do intelecto humano, junto com a leitura, a escrita e a aritmética pois, como estas, serve para descrever, explicar e modelar o universo e seus processos complexos.

Tanto o uso de ferramentas online, quanto offline, são lacunas a serem exploradas, tanto pela parte de pesquisa, quanto de aplicação de tais metodologias em sala de aula. Vale ressaltar que uma das lacunas se encontra no uso de ferramentas offline.

Oliveira & Junior & Filho (2019) citam que estratégias têm sido direcionadas ao contato mais prematuro com os conceitos e a lógica de programação, o que tem proporcionado um ambiente de proliferação para o desenvolvimento de ferramentas e metodologias de ensino de programação, porém esse foco tem se voltado mais para jogos sérios e a construção de ambientes de gamificação.

Atualmente existem soluções offline, o Scratch² por exemplo, já citado anteriormente, ainda no ponto de offline existem ferramentas que não necessitam de uso de computadores, como jogos de cartas e quebra cabeças, por exemplo, porém, também existem soluções online, como o Portugol Webstudio³. Entretanto, há uma lacuna de ferramentas, tais como, que possam integrar esses dois lados, online e offline, além de ferramentas que possam reconhecer algoritmos montados de forma física, estando offline.

“A revisão de estudos que fornecem evidências sobre a utilidade da Computação Desplugada para desenvolver habilidades do PC mostra que ainda existe espaço para pesquisar essa área e existe também uma necessidade de mais pesquisas empíricas sobre esse assunto” (BRACKMANN, 2017). Desta forma, a solução proposta neste trabalho é o desenvolvimento de um aplicativo que reconheça imagens de algoritmos montados de forma física, aprimorando o ensino e melhorando a experiência no ensino e aprendizado de algoritmos.

1.2 OBJETIVOS

1.2.1 Objetivo principal

Prover uma aplicação que proporcione um ambiente de ensino na área do Pensamento Computacional, tendo o intuito de usar a Computação Desplugada, onde o professor possa disponibilizar um aplicativo que reconheça imagens de algoritmos montados de forma física, aprimorando a experiência no ensino e aprendizado de algoritmos.

1.2.2 Objetivos específicos

- Definição do problema e objetivo do projeto;
- Modelagem, estruturação e construção da aplicação;
- Implementação da aplicação;
- Testes da aplicação;
- Análise das limitações e benefícios do aplicativo.

² Scratch Offline. Disponível em: <https://scratch.mit.edu/download>

³ Portugol Webstudio. Disponível em: <https://portugol-webstudio.cubos.io/>

2 REFERENCIAL TEÓRICO

Nesta seção serão abordados, mais detalhadamente, os conceitos de Pensamento Computacional e Computação Desplugada, e suas aplicações. Ainda, serão abordados alguns trabalhos e pesquisas relacionados ao tema deste trabalho.

2.1 PENSAMENTO COMPUTACIONAL

Há dez anos, Jannette Wing publicou o artigo “Computational Thinking” (WING, 2006). Esse artigo alavancou discussões sobre a educação em computação nas escolas de todo o mundo. A partir dele, fomentou-se a reflexão das razões porque computação deveria ser ensinada nas escolas, posicionando-a como uma ciência, e não somente como um conhecimento voltado ao manuseio de computadores ou específico a profissionais da computação (ARAUJO, ANDRADE & GUERRERO, 2016). O conceito em si do PC pode remeter-se a algo mais antigo, por abordar temas que já são de conhecimento mais geral, porém, foi somente a partir de 2006 que o tema ganhou destaque e começou a ser abordado por diversos estudos.

O termo Pensamento Computacional é relativamente novo na Ciência da Computação, existindo diferentes definições, mas que convergem no potencial que essa habilidade pode oferecer. Por exemplo: PC é saber utilizar o computador como um instrumento de aumento do poder cognitivo e operacional humano, ou seja, é servir-se desses recursos para aumentar a produtividade, inventividade, e criatividade (Blikstein, 2008), e PC envolve resolver problemas, projetar sistemas e compreender o comportamento humano baseando se nos conceitos fundamentais da Ciência da Computação (WING, 2006) (ORTIZ, MOREIRA & PEREIRA, 2018).

Mesmo sendo um termo relativamente novo e não havendo uma definição única e objetiva acerca do assunto, há diversos estudos sobre formas de aplicação de PC em diversas áreas, dessas áreas destaca-se o ensino, em que há diversas alternativas para implementação e uso de tais conceitos em escolas.

“As habilidades constituintes do PC ainda não são consenso na literatura e dependem da abordagem teórica empregada em cada estudo” (ARAUJO et al., 2016). “A *Computing At School* é uma organização educacional britânica que considera PC como um processo cognitivo que envolve raciocínio lógico na resolução de problemas e pode ainda empregar mais cinco habilidades” (CSIZMADIA et al., 2015).

Pensamento algorítmico faz parte do PC e podemos entender pensamento algorítmico como o processo de desenvolver um algoritmo para resolução de problemas. Com o uso dele, facilita a definição de uma série de etapas a serem seguidas para resolução do problema. O pensamento algorítmico é uma estratégia para se chegar na solução por meio de passos

sequenciais bem definidos e ordenados. “A decomposição é uma maneira de dividir o problema central em problemas menores, mais fáceis de gerenciar e resolver” (ORTIZ, ANDRADE, GUERRERO, MELO & SOUZA, 2018).

Essas são as definições mais aceitas hoje, sobre PC, pois a partir dela podemos destacar as competências do PC e o processo de resolução de problemas a qual ele pode tratar. Ainda podemos incluir nesses conceitos o Reconhecimento de Padrões, que trata de ter uma visão mais ampla, depois de quebrado os problemas em problemas menores, no que há em comum entre esses problemas e, futuramente, aproveitar a mesma resolução.

Pensar computacionalmente não é sinônimo de “programar”. No entanto, Blikstein (2008) aponta que umas das etapas fundamentais do PC é saber programar um computador para realizar tarefas cognitivas e de maneira automatizada, para que este conhecimento seja um suporte ao raciocínio humano (ZANETTI, BORGES & RICARTE, 2016). Podemos definir o ato de “programar” como criar soluções para resolver diversos problemas. Tendo em visto esse conceito, o computador somente seria um aliado, tendo em vista sua capacidade de processamento de dados e informações.

Para a SBC, o PC não é abordado como a programação por si só, mas sim, é a habilidade de compreender, definir, modelar, comparar, solucionar, automatizar e analisar problemas (e soluções) de forma metódica e sistemática (SBC, 2019). Como diversas iniciativas estão sendo desenvolvida, destacarei o caso levantado por Oliveira, Junior & Filho (2019), que utilizaram do uso de um jogo sério.

“Os jogos sérios permitem a transmissão do conhecimento de forma lúdica, prática e imersiva aos alunos” (MICHAEL e CHEN, 2005). A ludicidade é um fator importante quando se trata da necessidade de criar engajamento com o público, ou seja, o divertimento pode ter um papel influente na aquisição de conhecimento pelo estudante e pode ser utilizado para capturar a atenção de crianças para o estudo do Pensamento Computacional (GIANNAKOS, 2013) (OLIVEIRA, JUNIOR & FILHO, 2019).

O jogo sério desenvolvido foi baseado no clássico Bomberman, um jogo 2d que apresenta características similares aos jogos utilizados para a extração de características. O Bomberman é um jogo de estratégia em forma de labirinto, com o objetivo de destruir os obstáculos. Nesse contexto, a adaptação realizada no jogo foi permitir que o personagem interaja com o ambiente por comandos programáveis. Ressalta-se também que as fases do jogo foram elaboradas com base no método PBL, ou seja, pela proposta de apresentar desafios (problemas) contextuais ao ensino do pensamento computacional a serem realizados durante o jogo (OLIVEIRA, JUNIOR & FILHO, 2019).

O conceito de PC que será abordado neste trabalho está relacionado às atividades que são realizadas pelos alunos em sala de aula de forma offline. Também, a aplicação proposta neste projeto visa auxiliar no processo de ensino contido nesta área.

2.2 COMPUTAÇÃO DESPLUGADA

“A Computação Desplugada pode ser compreendida como um método de ensino com o objetivo de explicar os fundamentos da computação sem a necessidade de um computador, esse método foi criado com o propósito de possibilitar que indivíduos que não possuem acesso ao computador tenham a oportunidade de entender o seu funcionamento e seus fundamentos” (Bell et al., 2009) (SILVA & GUARDA, 2019).

Um espaço em que a infraestrutura tecnológica é precária e/ou ausente, esse é o cenário de diversas escolas no Brasil, onde não há a possibilidade de prover computadores para todos alunos, nesse contexto a CD facilitaria o ensino de computação para os alunos. Um levantamento realizado por Avila et al. (2016) descreve os seus desdobramentos no Brasil e sugere a Computação Desplugada (Bell et al., 2011), dentre as diversas metodologias citadas para a disseminação das habilidades do PC, como uma abordagem apropriada para a introdução de conhecimentos relacionados à programação de computadores, devido ao alto índice de evasão nas disciplinas de programação de cursos de Licenciatura em Computação (RODRIGUES, ARANHA & SILVA, 2018).

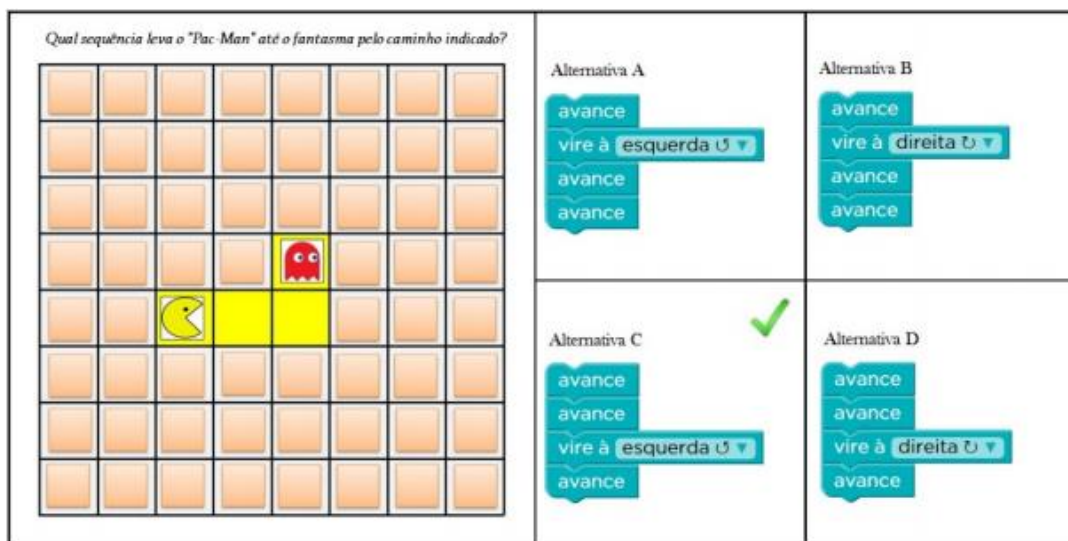
Rompida a questão de infraestrutura, ainda tem a questão de evasão das disciplinas de programação, seja devido a forma que são ministradas as disciplinas ou a real complexidade da mesma, também, nesse contexto, a CD facilitaria por não ter uma única prática de ensino e uso. Um ponto importante sobre a CD é que pode ser aplicada de forma descontraída e divertida, não sendo restringida a questão de idade e ensino, podendo ser aplicada do ensino fundamental até o ensino superior. Segundo SOUZA & LOPES (2018), a utilização da metodologia de Computação Desplugada para adaptação no ensino de conteúdo tem sido relatada como uma metodologia com bastante aceitação por parte dos alunos, por ser uma estratégia lúdica se caracterizando como um método de ensino de computação sem o uso do computador.

A Computação Desplugada é uma opção para se ensinar conceitos de Algoritmos e Programação sem o uso de tecnologia em que várias dinâmicas podem ser construídas. Um exemplo é o “Contando os Pontos”, do livro “*Computer Science Unplugged*” (BELL et al., 2009). As Dinâmicas de Grupo também podem auxiliar, tendo em vista que envolvem os alunos em atividades lúdicas em sala. Um exemplo é os alunos lerem

QR Codes em sala e nos corredores de forma a resolver determinado exercício (CASTRO & SIQUEIRA, 2019).

Outra abordagem que está sendo usada para o ensino de algoritmos utilizando CD são as cartas ou blocos, que quando montados, formam um algoritmo. Um exemplo é proposto por Brackmann (2017), onde se pergunta quais são os comandos que levam o 'Pac-Man' até o fantasma pelo caminho assinalado. Mas neste caso, as opções de resposta, ao invés de flechas, são blocos que encaixam uns nos outros. A alternativa correta neste exemplo é a C, conforme Figura 3.

Figura 3: Exemplo do Teste de Pensamento Computacional



Fonte: Brackmann (2017).

A partir do material levantado surgiu a ideia de um aplicativo mobile que possibilita a extração de texto de imagens onde há um algoritmo, para assim fazer a análise do texto extraído por parte do docente, sendo o escopo definido para este projeto.

2.3 TRABALHO RELACIONADOS

No trabalho “Metodologias, Técnicas, Ambientes e Tecnologias Alternativas utilizadas no Ensino de Algoritmos e Programação no Ensino Superior no Brasil”, dos autores Ronney Moreira de Castro e Sean Wolfgang Matsui Siqueira (2019), é realizado um mapeamento para identificar metodologias, técnicas, ambientes e tecnologias alternativas para o ensino de algoritmos.

No trabalho, o autor elenca algumas metodologias e técnicas alternativas encontradas neste levantamento realizado, além de suas respectivas especificações e exemplos. Também, durante esse levantamento, o autor aborda questões de ambientes para ensino. Como resultado, de forma geral os autores expuseram formas alternativas para o ensino de algoritmos e estratégias para difusão dessas práticas.

Já no trabalho “Estímulo ao Pensamento Computacional a partir da Computação Desplugada: uma proposta para Educação Infantil”, dos autores Elisângela Ribas dos Santos, Graciele Soares, Guilherme Dal Bianco, João Bernardes da Rocha Filho e Regis Alexandre Lahm (2016), é realizado um estudo sobre as possibilidades de construção de estratégias didáticas para difusão de PC. Também é abordado questões de infraestrutura de escolas e problemas de acesso à internet.

Os autores realizaram um estudo sobre possibilidade de construção de estratégias didáticas para difusão do PC, para tais estudos foi implementada uma oficina teórico-prática utilizando CD para levantamentos dos resultados. Como resultados foi constatado que parte dos professores não havia tido conhecimento sobre tais práticas e a maioria não se sente à vontade para aplicá-las.

E no trabalho “Estimulando o Pensamento Computacional e o raciocínio lógico no ensino fundamental por meio da OBI e computação desplugada”, das autoras Jessica Silva de Souza e Alba Sandrya Bezerra Lopes (2018), estas aplicaram desafios propostos pela Olimpíada Brasileira de Informática em uma escola com o objetivo de auxiliar no desenvolvimento do raciocínio lógico e Pensamento Computacional por meio da Computação Desplugada. Além de um maior envolvimento dos alunos nas atividades, foi notado uma melhora na compreensão de questões que envolvem lógica, além de uma maior compreensão dos enunciados das questões.

Após as análises dos trabalhos citados anteriormente notou-se que aplicar atividades utilizando PC e CD pode ser mais interativo para os alunos, além de serem métodos eficazes para o ensino de algoritmos. Essas pesquisas ajudaram a ver que existem soluções online e offline sendo aplicadas e esta foi a motivação do trabalho.

Desta forma, o diferencial deste trabalho é o desenvolvimento de um aplicativo utilizando Computação Desplugada, em que os alunos possam trabalhar em sala de aula tanto online quanto offline. Com o aplicativo, o aluno poderá responder atividades cadastradas pelos docentes por meio da extração de textos em imagens capturadas dos algoritmos construídos de

forma física, assim como, o docente poderá visualizar essas respostas, desde que esteja online, e dar feedbacks sobre essas respostas, proporcionando uma maior interatividade entre aluno e docente, através da aplicação.

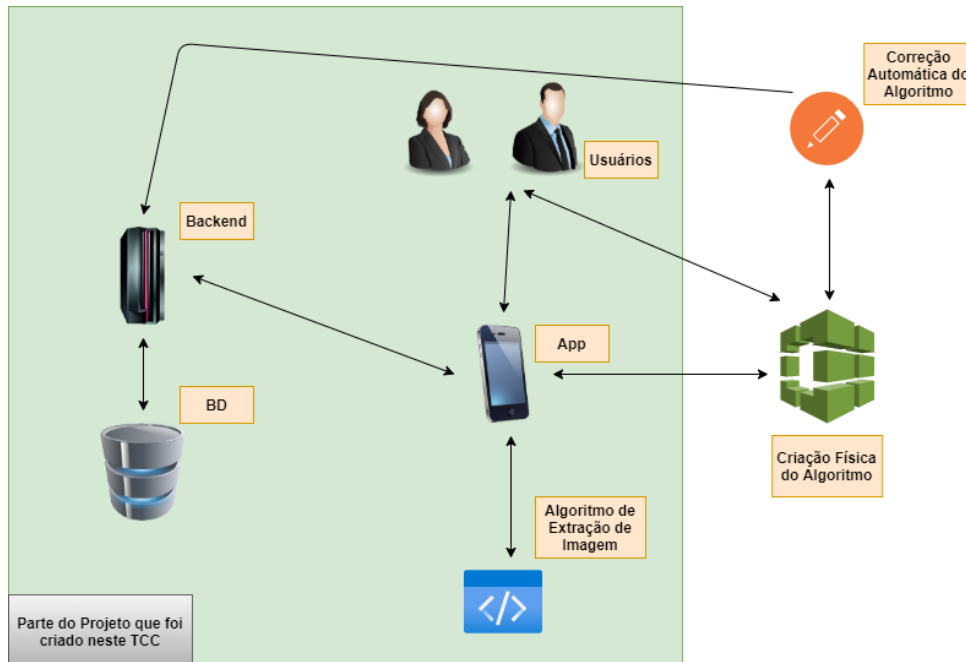
3 METODOLOGIA

O ponto inicial deste trabalho foi o levantamento do problema de pesquisa e dos trabalhos que abordam o conceito de Pensamento Computacional em sua didática e o uso de Computação Desplugada em sua prática. Juntamente com a pesquisa, foi elencado o valor que o uso desses conceitos pode agregar.

Com isso, foi identificada uma lacuna envolvendo a aplicação de Pensamento Computacional, juntamente com a Computação Desplugada, a ser explorada. Com a computação cada vez mais presente no nosso dia-a-dia, também com a atenção aplicada sobre ela em questões de ensino, há poucos trabalhos envolvendo a aplicabilidade desses conceitos em sala de aula.

A partir deste ponto, foi construído um esboço de um projeto mais amplo e com a execução a longo prazo de suas tarefas, conforme pode ser visto na Figura 4, em que a pesquisa apresentada neste trabalho é um eixo do projeto como um todo.

Figura 4: Visão Geral do Projeto



Fonte: Autor(2020)

A imagem anterior apresenta a ideia geral e ampla do projeto como um todo, em que a parte visual na cor verde representa o que foi realizado neste trabalho de conclusão de curso. Desta forma, este trabalho abrangeu dentro do projeto a criação do backend e seu respectivo

banco de dados para armazenar as informações oriundas do aplicativo mobile, assim como, houve a seleção e escolha das melhores alternativas algorítmicas para ser realizada a extração do texto das imagens.

Desta forma, este trabalho contempla a criação do aplicativo e seu respectivo backend com banco de dados, assim como a funcionalidade de extração de imagens dos algoritmos a serem criados no aplicativo mobile. As etapas não abrangidas/futuras neste projeto estão centralizadas na criação de soluções para construção de algoritmos físicos pelos alunos, de forma que eles possam tirar fotos das soluções construídas e enviar via aplicativo mobile para um servidor que irá realizar a correção automática deles e fornecer um feedback ao aluno e professor.

É importante ressaltar que o projeto contempla a ideia de que tanto o professor como o aluno podem enviar respostas, evitando assim obrigar que o aluno precise ter um aparelho celular a sua disposição para o envio das atividades propostas pelo docente. Além disso, o envio das respostas poderá ser realizado sem conexão com a internet, em que posteriormente haverá a sincronização dos dados quando esta conexão for estabelecida. Isso reflete na apresentação de soluções alternativas para que os conceitos de Computação Desplugada aliado ao estímulo da construção do Pensamento Computacional possam ser colocados em prática.

Após levantamento das necessidades desta pesquisa específica para o projeto, iniciou-se a modelagem da aplicação, como: modelagem das telas, requisitos funcionais e não funcionais, diagrama de atividade, modelo lógico de dados e *user story*. Após foram levantadas questões sobre as tecnologias a serem usadas para atingir o propósito do trabalho. Todos os levantamentos abordados tiveram como objetivo a construção do projeto do aplicativo, tanto frontend quanto backend, para que pudesse ser feita a codificação dos mesmos. Definidos os levantamentos, e antes da codificação, foram levantadas questões de arquiteturas de software a serem empregadas, e que serão abordadas em detalhes na seção de desenvolvimento.

Para o desenvolvimento no backend foram utilizados o NodeJS e o framework Express para a base da aplicação e o TypeORM para conexão com o banco de dados Postgres. Para os testes no mobile foram utilizados Flutter, React Native e o Ionic Framework.

3.1 ALGORITMOS DE EXTRAÇÃO DE TEXTO DE IMAGEM

Em um primeiro momento foram levantadas 3 alternativas para a extração de texto de imagens, o Tesseract OCR⁴, o Reconhecimento de texto do Firebase⁵ e o OCR, plugin do Ionic Framework⁶.

O Tesseract OCR foi escolhido para testes pelo fato de a responsabilidade de extração de texto ser feito no server-side, eliminando a necessidade do processamento das imagens no aparelho celular do usuário.

O reconhecimento de texto do Firebase e o OCR já eliminam a possibilidade de ter um backend para fazer o processamento dessas imagens, visto que o reconhecimento de texto do Firebase é feito diretamente com os servidores do mesmo. Já o OCR é feito diretamente no aparelho celular do usuário, eliminando qualquer necessidade de uma conexão com a internet para o processamento das imagens.

3.2 ALGORITMOS DE EXTRAÇÃO DE TEXTO DE IMAGEM

Após selecionados os algoritmos, foram realizados os testes. Para os testes dos algoritmos, a aplicação backend foi hospedada em um servidor free da Heroku⁷ com 512 MB RAM. Já para os testes com os algoritmos no aparelho celular foi utilizado um emulador Android, com a versão 29, tela de 5” e com 4 GB RAM.

Em um primeiro momento de testes, o Tesseract OCR já foi descartado por exigir um alto processamento do servidor. Em testes simulando requisições, o tempo para análise superou os 30 segundos, não proporcionando uma boa experiência para o usuário. Desta forma, os demais testes realizados foram conduzidos durante o desenvolvimento deste projeto e são apresentados em detalhes na seção dos Resultados.

Como etapas futuras no desenvolvimento metodológico deste projeto, serão realizados testes do aplicativo com usuários. Primeiro para testar questões de usabilidade e fluxos da aplicação. Ainda serão adicionadas algumas funcionalidades no aplicativo, em implementações

⁴ Tesseract OCR. Disponível em: <https://github.com/tesseract-ocr/tesseract>

⁵ Reconhecimento de texto do Firebase. Disponível em: <https://firebase.google.com/docs/ml-kit/recognize-text>

⁶ OCR. Disponível em: <https://ionicframework.com/docs/native/ocr>

⁷ Heroku. Disponível em: <https://www.heroku.com/home>

futuras, uma delas é a implementação de correção automática dos algoritmos. Essa implementação ainda possibilitará que o usuário tenha um feedback *realtime* do algoritmo extraído da imagem.

4 DESENVOLVIMENTO

Para atingir a finalidade do projeto, as seguintes etapas foram seguidas.

4.1 REQUISITOS FUNCIONAIS E NÃO FUNCIONAIS

Podemos dividir os requisitos de um software como Requisitos Funcionais e Requisitos Não Funcionais. Os Requisitos Funcionais podem ser abordados como tudo aquilo que o software precisa ter para seu funcionamento correto. Sommerville (2012) descreve os Requisitos Funcionais como tudo o que ele deve fazer. Desta forma, os seguintes requisitos funcionais foram definidos para este projeto:

- Cadastro de usuários, turmas, atividades e respostas para essas atividades;
- Alteração dos dados de turmas e atividades;
- Usuários poderão se vincular a uma turma através do código gerado pelo sistema;
- O usuário que criar a turma poderá vincular outros usuários através do email informado no cadastro;
- O usuário que criar a turma poderá visualizar as respostas de atividades que ele cadastrou e dar um feedback para o aluno.

Os Requisitos Não Funcionais não são, necessariamente, necessidades ou funcionalidades do software. Sommerville (2012), cita que podem estar relacionados às propriedades emergentes do sistema, como confiabilidade, tempo de resposta e ocupação de área. Desta forma, os seguintes requisitos não funcionais foram definidos para este projeto:

- Cadastro e login do usuário deve ser através de sua conta junto ao Google;
- Aplicativo deve funcionar tanto online, quanto offline.

4.2 REGRAS DE NEGÓCIO

Outros requisitos do sistema podemos chamar de regras de negócio. Diferente dos Requisitos Funcionais e Não Funcionais, que descrevem “o que” o software deve fazer, as regras de negócio referem-se ao “como” ele deve fazer. As seguintes regras de negócio foram definidas para este projeto:

- Ao criar uma turma, deverá ser gerado, automaticamente, um código único, para identificar a turma.
- O usuário que criar a turma, será considerado o professor da mesma.
- As formas de vincular usuários a turma serão as seguintes:
 - Caso seja o professor da turma, poderá vincular outros usuários através do e-mail;
 - Caso seja outro usuário, poderá ingressar na turma através do código gerado.

- Os demais usuários que forem vinculados a turma serão considerados alunos da mesma.
- Um usuário poderá ser vinculado uma única vez com determinada turma.

4.3 USER STORY

Para a elaboração desta seção é usado o conceito de Histórias de Usuário (*User Story*), que é muito utilizado no desenvolvimento de softwares, principalmente junto com a metodologia ágil, e é usada para descrever uma necessidade do usuário de forma simples.

Uma história de usuário pode ser caracterizada como uma curta e simples descrição da necessidade do cliente (BERNARDO, 2014). Para escrever boas histórias de usuário, o foco deve estar totalmente no cliente (BRASILEIRO, 2017). Uma boa história de usuário deve atender o conceito INVEST: *independent* (independente), *negotiable* (negociável), *valuable* (de grande valor), *estimable* (estimável), *small* (pequena), *testable* (testável) (Bernardo, 2014) (CORREA, 2018).

As Histórias de Usuário serão abordadas da seguinte maneira: História, Contexto, Campos e Comportamentos e Cenários. A História descreve a funcionalidade em si. O Contexto descreve o caminho ou os pré-requisitos para determinado usuário poder tomar aquela ação. Campos e Comportamentos é mais voltado para a parte visual do software, nele são descritos os campos que aquela tela tem, assim como alguma eventual regra ou limitação.

As pessoas geralmente acham mais fácil se relacionar com exemplos da vida real do que com descrições abstratas. Elas podem compreender e criticar um cenário de como elas podem interagir com um sistema de software. Engenheiros de requisitos podem usar a informação obtida a partir deste debate para formular os requisitos do sistema atual (SOMMERVILLE, 2012).

Dessa forma são abordados os Cenários, que buscam simular todos os comportamentos e interações que o usuário terá naquela tela.

4.3.1 CADASTRAR TURMA

Essa *User Story* tem como objetivo documentar elementos, padrões e regras para um usuário poder cadastrar uma turma.

- História: Eu como usuário do sistema, devo poder criar uma turma.

- Contexto: Dado que estou logado no sistema e na tela de listagem de turmas, clico em adicionar turma.
- Campos e Comportamentos:
 - Nome da Turma[nome]: Obrigatório, mínimo de 3 e máximo de 50 caracteres.
 - Descrição da Turma[nome]: Obrigatório, mínimo de 4 e máximo de 500 caracteres.
 - Devo ser vinculado a ela automaticamente como professor e assim considerado pelo sistema;
- Deve ser gerado automaticamente um código único para identificação da turma;
- Cenário: Cadastrar turma com sucesso
 - Informo o nome da turma;
 - Informo a descrição da turma;
 - E clico no botão salvar;
 - Então deve apresentar a mensagem de turma cadastrada com sucesso;
 - E após clicar em Ok devo ser redirecionado para a tela de informações da turma.
 - Cenário: Cadastrar turma sem sucesso, nome obrigatório
 - Quando não informo um nome de turma ou informo os valores “”, “ “;
 - Informo a descrição da turma;
 - E clico no botão salvar;
 - Então deve apresentar a mensagem de campo obrigatório;
 - E permanecer na tela de cadastro de turma.
- Cenário: Cadastrar turma sem sucesso, descrição obrigatória
 - Informo o nome da turma;
 - Quando não informo um nome de turma ou informo os valores “”, “ “;
 - E clico no botão salvar;
 - Então deve apresentar a mensagem de campo obrigatório;
 - E permanecer na tela de cadastro de turma.
 - Cenário: Cadastrar turma sem sucesso, tamanho do nome
 - Quando informo um nome de turma com menos de 3 ou mais de 50 caracteres;
 - Informo a descrição da turma;
 - E clico no botão salvar;

- Então deve apresentar a mensagem de campo deve ter entre 3 a 50 caracteres;
- E permanecer na tela de cadastro de turma.
- Cenário: Cadastrar turma sem sucesso, tamanho da descrição
 - Informo o nome da turma;
 - Quando informo um nome de turma com menos de 4 ou mais de 500 caracteres;
 - E clico no botão salvar;
 - Então deve apresentar a mensagem de campo deve ter entre 4 a 500 caracteres;
- E permanecer na tela de cadastro de turma.

4.3.2 VINCULAR USUÁRIO POR E-MAIL

Essa *User Story* tem como objetivo documentar elementos, padrões e regras para um usuário professor de determinada turma poder vincular um aluno a essa turma através do e-mail.

- História: Eu como usuário professor de determinada turma, devo poder vincular alunos através do e-mail fornecido no cadastro dos mesmos.
- Contexto: Dado que estou logado no sistema e acesso mais informações de alguma turma que sou professor, clico em ‘Mais ações’ e ‘Vincular Aluno’.
- Campos e Comportamentos:
 - E-mail do aluno[e-mail]: Obrigatório, deve ser um e-mail válido;
 - Um usuário só pode ser vinculado uma única vez junto a determinada turma, seja como aluno ou professor.
- Cenário: Vincular aluno com sucesso
 - Informo e-mail do aluno;
 - E clico no botão vincular;
 - Então deve apresentar a mensagem de aluno vinculado com sucesso;
 - E após clicar em Ok devo ser redirecionado para a tela de informações da turma.
 - Cenário: Vincular aluno sem sucesso, e-mail obrigatório
 - Quando não informo um e-mail ou informo os valores “”, “ “;
 - E clico no botão vincular;
 - Então deve apresentar a mensagem de campo obrigatório;

- E permanecer na tela de vinculação de aluno.
- Cenário: Vincular aluno sem sucesso, e-mail inválido
 - Quando não informo um e-mail válido, como “abc13”;
 - E clico no botão vincular;
 - Então deve apresentar a mensagem de deve ser um e-mail válido;
 - E permanecer na tela de vinculação de aluno.
- Cenário: Vincular aluno sem sucesso, usuário já vinculado
 - Informo e-mail do aluno;
 - E clico no botão vincular;
 - Então deve apresentar a mensagem de aluno já vinculado;
 - E permanecer na tela de vinculação de aluno.

4.3.3 RESPONDER ATIVIDADE

Essa *User Story* tem como objetivo documentar elementos, padrões e regras para um usuário aluno de uma determinada turma poder responder uma atividade cadastrado pelo professor desta turma.

- História: Eu como usuário aluno de determinada turma, devo poder responder as atividades cadastradas pelo professor da turma.
- Contexto: Dado que estou logado no sistema e acesso mais informações de alguma turma que sou aluno, clico em determinada atividade e logo após clico em responder.
- Campos e Comportamentos:
 - Parecer do aluno[*note*]: Obrigatório, mínimo de 3 e máximo de 50 caracteres;
 - Imagem[*image*]: Obrigatório, imagem com dados da atividade;
 - Texto extraído[*report*]: Obrigatório, mínimo de 4 caracteres;
- Cenário: Responder atividade com sucesso
 - Informo o parecer;
 - Anexo uma imagem;
 - Extraio o texto da imagem;
 - E clico no botão responder;
 - Então deve apresentar a mensagem de atividade respondida com sucesso;
 - E após clicar em Ok devo ser redirecionado para a tela de informações da turma.

- Cenário: Responder atividade sem sucesso, parecer obrigatório
- Quando não informo um parecer ou informo os valores “”, “ “;
- Anexo uma imagem;
- Extraio o texto da imagem;
- E clico no botão responder;
- Então deve apresentar a mensagem de campo obrigatório;
- E permanecer na tela de responder atividade.
- Cenário: Responder atividade sem sucesso, imagem obrigatória
 - Informo o parecer;
 - Quando não anexo uma imagem;
 - Extraio o text da imagem;
 - E clico no botão responder;
 - Então deve apresentar a mensagem de campo obrigatório;
 - E permanecer na tela de responder atividade.

4.4 DIAGRAMA DE ATIVIDADES

O diagrama de atividade mostra o comportamento dinâmico de um sistema ou parte de um sistema através do fluxo de controle entre ações que o sistema executa. Ele é similar a um fluxograma exceto que pode mostrar fluxos concorrentes (PRESSMAN, 2011).

Um diagrama de atividades demonstra o fluxo de controle entre uma determinada atividade e outra. Também é definido como um diagrama que especifica o comportamento do software. O componente principal de um diagrama de atividade é um nó ação, representado por um retângulo arredondado, que corresponde a uma tarefa executada por um sistema de software.

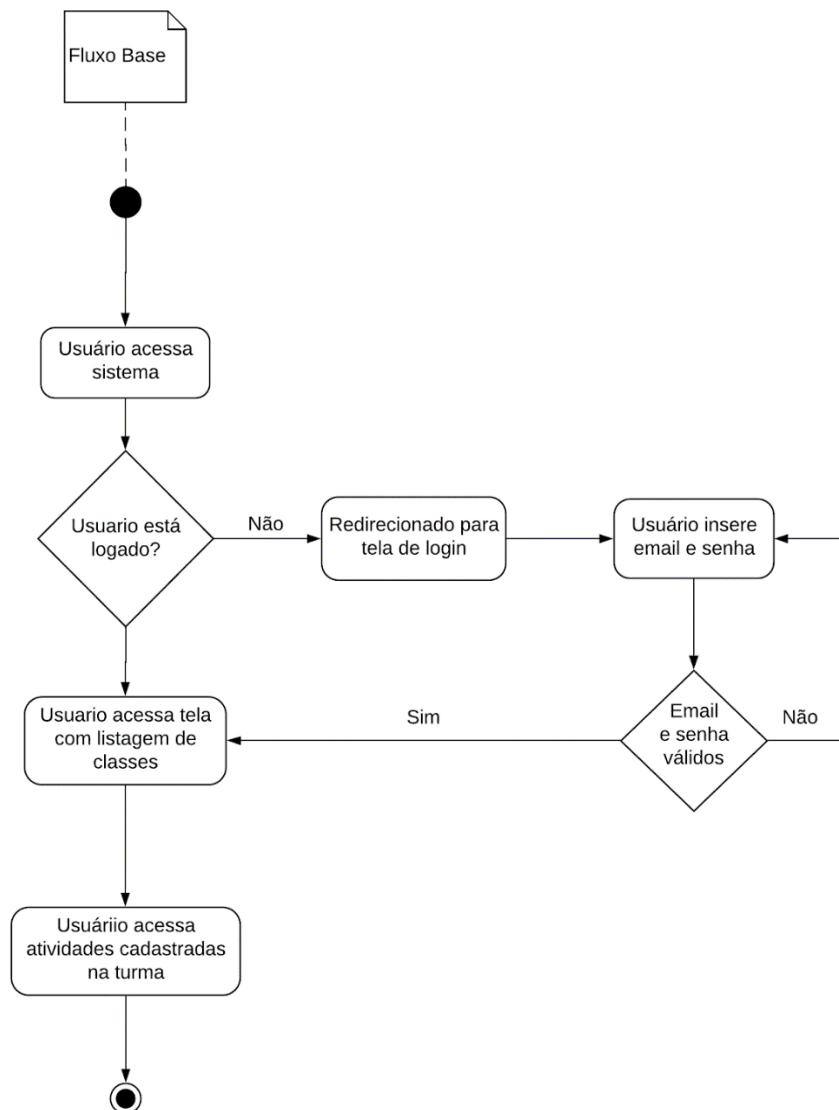
Setas que vão de um nó ação para outro indicam o fluxo de controle. Isto é, uma seta entre dois nós ação significa que depois que a primeira ação é completada, a segunda ação começa. Um ponto preto sólido forma o nó inicial que representa o ponto inicial da atividade. Um ponto preto envolvido por um círculo preto é o nó final indicando o fim da atividade (PRESSMAN, 2011).

Com base nessa descrição, foram elaborados diversos fluxos exemplificando o funcionamento do sistema. Esses fluxos serão explicados mais detalhadamente a seguir:

4.4.1 FLUXO BASE

A figura 5 exemplifica como seria o fluxo do usuário acessando o sistema desde o acesso do aplicativo, passando pelo login, listagem das turmas até a listagem das atividades.

Figura 5: Diagrama de Atividade para o Fluxo Base



Fonte: Autor (2020).

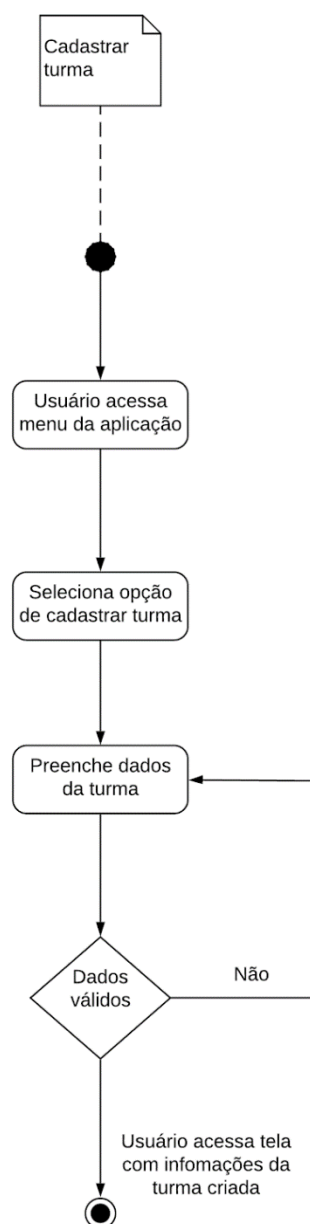
O fluxo acima representa um usuário acessando as listagens de atividades e turmas vinculadas a ele no sistema, a partir dele que se distinguem e executem as demais atividades. Como não a distinção de nível de usuários até a criação de turma, o fluxo se aplica a qualquer usuário. Pode-se observar que será iniciado sempre que o usuário acessar o sistema, a partir

desse momento será validado se ele está logado ou não no sistema, caso não esteja é redirecionado para a tela de login, na tela de login o usuário deverá inserir e-mail e senha, caso estejam corretos e já estejam cadastrados, usuário é redirecionado para a com a listagem de turmas e, a partir dela, poderá acessar a listagem de atividades e se encerra o fluxo.

4.4.2 CADASTRAR TURMA

A figura 6 exemplifica o fluxo do usuário até a criação de uma turma.

Figura 6: Diagrama de Atividade para Cadastrar Turma



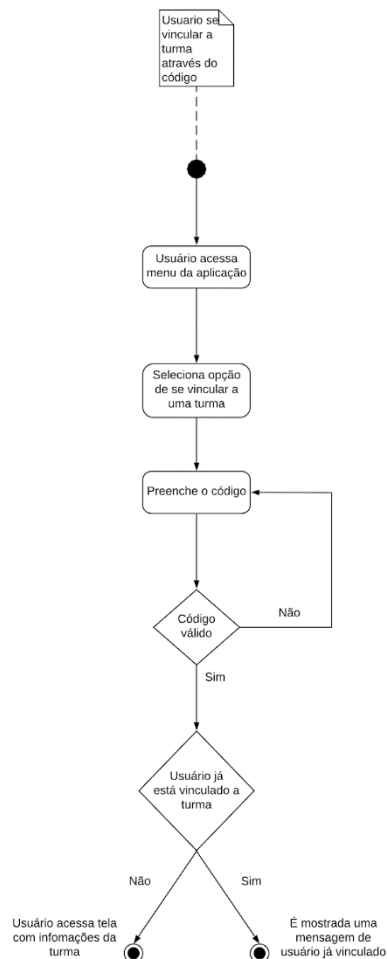
Fonte: Autor (2020).

A figura acima mostra o fluxo para o usuário criar uma turma. Como não há distinção de usuários, até o momento, qualquer usuário pode criar uma turma. Fluxo parte que o usuário já esteja logado no sistema. Usuário acessa o menu da aplicação e terá a opção para criar turma, selecionando essa opção, ele é redirecionado para a tela de cadastro de turmas, dado que preencha os dados corretamente ele é redirecionado para a tela com informações da turma e se encerra o fluxo.

4.4.3 USUÁRIO SE VINCULAR A TURMA ATRAVÉS DO CÓDIGO

A figura 7 exemplifica o fluxo do usuário para se vincular a turma através do código da mesma.

Figura 7: Diagrama de Atividade Usuário de Vincular a Turma Através do Código



Fonte: Autor (2020).

A figura acima mostra o fluxo para o usuário se vincular a uma turma através do código da mesma. Fluxo parte que o usuário já esteja logado no sistema. Usuário acessa o menu da aplicação e terá a opção para se vincular a uma turma, selecionando essa opção, ele é redirecionado para a tela de para se vincular a turma, dado que o código corretamente é validado se ele já está vinculado a essa turma, seja como aluno ou como professor, caso não ele é redirecionado para a tela com informações da turma a qual ele se vinculou e se encerra o fluxo, caso ele já esteja vinculado é mostrado a mensagem “Usuário já vinculado a turma” e se encerra o fluxo.

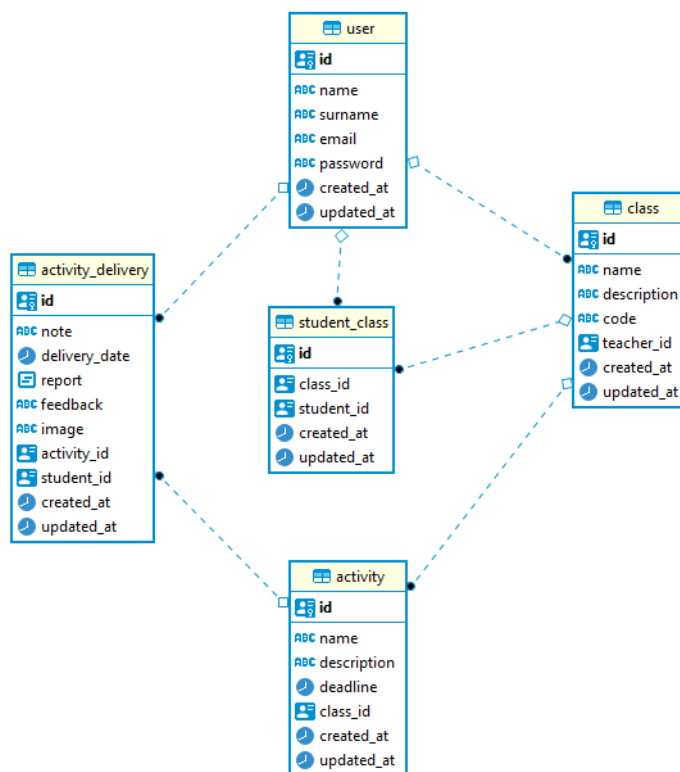
4.5 MODELO LÓGICO

Se entre os requisitos de software tivermos a necessidade de criar, estender ou habilitar uma interface com um banco de dados ou se as estruturas de dados complexas tiverem de ser construídas e manipuladas, a equipe de software poderá optar por criar um modelo de dados como parte da modelagem de requisitos. Um analista ou engenheiro de software define todos os objetos de dados processados no sistema, os relacionamentos entre os objetos de dados e outras informações pertinentes aos relacionamentos. O diagrama entidade-relacionamento (entity-relationship diagram) trata das questões e representa todos os objetos de dados introduzidos, armazenados, transformados e produzidos em uma aplicação (PRESSMAN, 2011).

Objeto de dados é uma representação das informações compostas que devem ser compreendidas pelo software. Por informação composta, queremos dizer algo que tenha uma série de propriedades ou atributos diferentes. Conseqüentemente, a largura (um único valor) não seria um objeto de dados válido, mas dimensões (incorporando altura, largura e profundidade) poderia ser definido como um objeto (PRESSMAN, 2011).

Pode-se definir o Modelo Lógico como algo que representa os dados no banco de dados. Essa representação pode ser altamente fiel ou não. A seguir está ilustrado o modelo lógico construído para este projeto na Figura 8.

Figura 8: Modelo de dados do aplicativo



Fonte: Autor (2020).

A tabela, ou objeto de dados, *User* representa onde serão armazenados os dados dos usuários que usarão o aplicativo. A tabela *Class* representa onde serão armazenados os dados das turmas criadas no aplicativo, ela pode se relacionar com a tabela *User* de duas maneiras, a primeira pelo campo *teacher_id* da própria tabela que vincula um usuário a ela, o tornando professor da mesma, a segunda é através da tabela intermediária *Student_Class*, todos os usuários que estiverem ligados a turma através desta tabela serão os alunos daquela turma.

A tabela *Activity* é a tabela onde serão armazenados os dados das atividades cadastradas, toda atividade deverá estar relacionada a uma turma, dessa forma a ligação está através do campo *class_id*. A tabela *Activity_Delivery* é a tabela onde serão armazenados os dados das respostas dessas atividades, conseqüentemente ela deve estar relacionada a uma atividade, esse relacionamento se dá pelo atributo *activity_id*, por se tratar das respostas das atividades, também deverá estar ligada a tabela de usuários, esse relacionamento se dá pelo campo *student_id*.

4.6 FERRAMENTAS

Nesta seção serão abordadas as tecnologias usadas para armazenamento, tratamento e amostragem dos dados, assim como testes para a mesma. Para desenvolvimento no backend foi utilizado o Express para a base da aplicação e o TypeORM para conexão com o banco de dados. Para os testes no mobile foram utilizados o Flutter, React Native e o Ionic Framework.

O Javascript⁸ foi desenvolvido originalmente como uma linguagem para atuar no *client-side*, ou seja, atuar no lado do cliente, ou navegadores. Com ele é possível alterar o conteúdo HTML e o CSS apresentados para o usuário.

O Typescript⁹ é uma linguagem de programação, desenvolvida pela Microsoft. Com ele é possível desenvolver aplicações em Javascript, porém com algumas funcionalidades a mais no desenvolvimento, como tipagem dinâmica de dados, classes e interfaces. Ele foi usado como linguagem base para desenvolvimento das aplicações que utilizam Javascript.

NodeJS¹⁰ é uma plataforma de código aberto para construir aplicações, atuando tanto no lado *client-side*, como no *server-side*, que utiliza o Javascript como sintaxe. Os motivos por ele ser escolhido como principal ferramenta nesse trabalho é fato de ser uma plataforma leve e facilitar na produtividade, visto que sua curva de aprendizagem é pequena e pode-se desenvolver com muita rapidez.

4.6.1 SERVER-SIDE

Para armazenamento dos dados foi usado o banco de dados relacional de código aberto PostgreSQL¹¹. Pode-se observar o modelo lógico montado na seção 4.5 montado no PostgreSQL. Ele foi escolhido por ser um banco escalável, apoio da comunidade e suporte a diversos tipos de dados.

Express¹² é um framework NodeJS, de código aberto, para desenvolvimento de aplicações no lado do servidor. Com ele é possível a abstrações de rotas, middlewares e muitas outras funções para facilitar a criação tanto de API's.

⁸ Javascript: Disponível em: <https://developer.mozilla.org/pt-BR/docs/Web/JavaScript>

⁹ Typescript. Disponível em: <https://www.typescriptlang.org/>

¹⁰ NodeJS. Disponível em: <https://nodejs.org/pt-br/>

¹¹ PostgreSQL. Disponível em: <https://www.postgresql.org>

¹² Express. Disponível em: <https://expressjs.com/pt-br/>

Assim como o Express nos fornece abstrações para criar a API, o TypeORM¹³ nos fornece abstrações para comunicação com o banco de dados. Ele foi usado como a forma de comunicação com o banco de dados PostgreSQL, através dele foram construídas consultas, inserções e alterações nos dados do banco de dados.

4.6.2 CLIENT-SIDE

React Native¹⁴ é um framework Javascript de código aberto, baseado em ReactJS, que possibilita o desenvolvimento de aplicações mobile multiplataforma, ou seja, tanto para Android, como para iOS. Diferente de outros frameworks que possuem a mesma finalidade, todo código desenvolvido é convertido para linguagem nativa do sistema operacional, o que torna o app muito mais fluido. Ele foi usado para construção do aplicativo para testes.

Ionic Framework¹⁵ é um framework de código aberto que possibilita desenvolvimento de aplicações mobile híbridas. Diferente do React Native, que pode ser usado somente com ReactJS, o Ionic pode ser usado com Angular, Vue JS, além do próprio ReactJS. Ele foi usado para construção do aplicativo para testes.

Assim como o React Native e o Ionic, o Flutter¹⁶ é um framework de código aberto para desenvolvimento de aplicações multiplataforma. Desenvolvida pela Google, a linguagem base para desenvolvimento em Flutter é o Dart¹⁷. Ele foi usado para construção do aplicativo para testes.

¹³ TypeORM. Disponível em: <https://typeorm.io/#/>

¹⁴ React Native. Disponível em: <https://reactnative.dev/>

¹⁵ Ionic Framework. Disponível em: <https://ionicframework.com/>

¹⁶ Flutter. Disponível em: <https://flutter.dev/>

¹⁷ Dart. Disponível em: <https://dart.dev/>

5 RESULTADOS E DISCUSSÕES

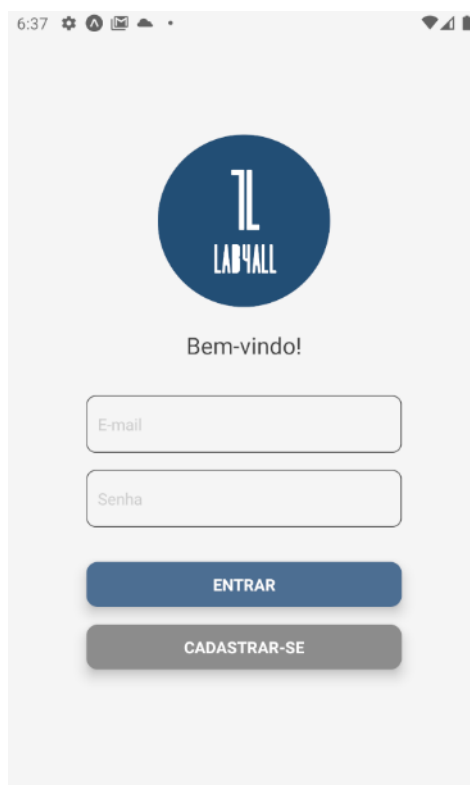
A fim de apresentar os resultados, os mesmos foram divididos em duas seções, uma apresentando as telas desenvolvidas e suas descrições e a outra seção onde será abordado sobre os testes realizados com algoritmos de extração de texto de imagens, como testes realizados e resultados dos mesmos.

5.1 TELAS

Nesta seção serão abordadas as tecnologias usadas para armazenamento, tratamento e amostragem dos dados, assim como testes para a mesma. Para desenvolvimento no backend foi utilizado o Express para a base da aplicação e o TypeORM para conexão com o banco de dados. Para os testes no mobile foram utilizados o Flutter, React Native e o Ionic Framework.

5.1.1 LOGIN


A Figura 9 representa tela de login, ela é a tela inicial do aplicativo, é para onde um usuário não autenticado será redirecionado e através dela o usuário poderá se autenticar no aplicativo. Seu conteúdo implica no logo do projeto, um formulário com dois campos, um para inserir o e-mail e outro para inserir a senha do usuário, a tela também possui dois botões, um para poder logar, dado que os dados estejam corretos, e outro para redirecionar o usuário para a tela de Cadastro de Usuário.

Figura 9: Tela Login

Fonte: Autor (2020).

5.1.2 CADASTRO DE USUÁRIO

A Figura 10 representa a tela de Cadastro de Usuário, ela é tela onde os usuários poderão criar suas contas. Seu conteúdo consiste em um formulário com os campos de e-mail, nome, sobrenome, senha e confirmação de senha, a tela também possui dois botões, um confirmar seus dados e submetê-los, dado que os dados estejam corretos, e outro para redirecionar o usuário para a tela de Login.

Figura 10: Tela Cadastro de Usuário

A tela de cadastro de usuário apresenta o logo 'TL LAB4ALL' no topo. Abaixo, há o texto 'Ainda não possui uma conta?' e a instrução 'Preencha corretamente os campos abaixo e clique em "Concluir"'. O formulário contém cinco campos de entrada: 'E-mail', 'Nome', 'Sobrenome', 'Senha' e 'Confirme sua Senha'. Na base, há dois botões: 'CONCLUIR' (em azul) e 'CANCELAR' (em cinza).

Fonte: Autor (2020).

5.1.3 LISTAGEM DE ATIVIDADES

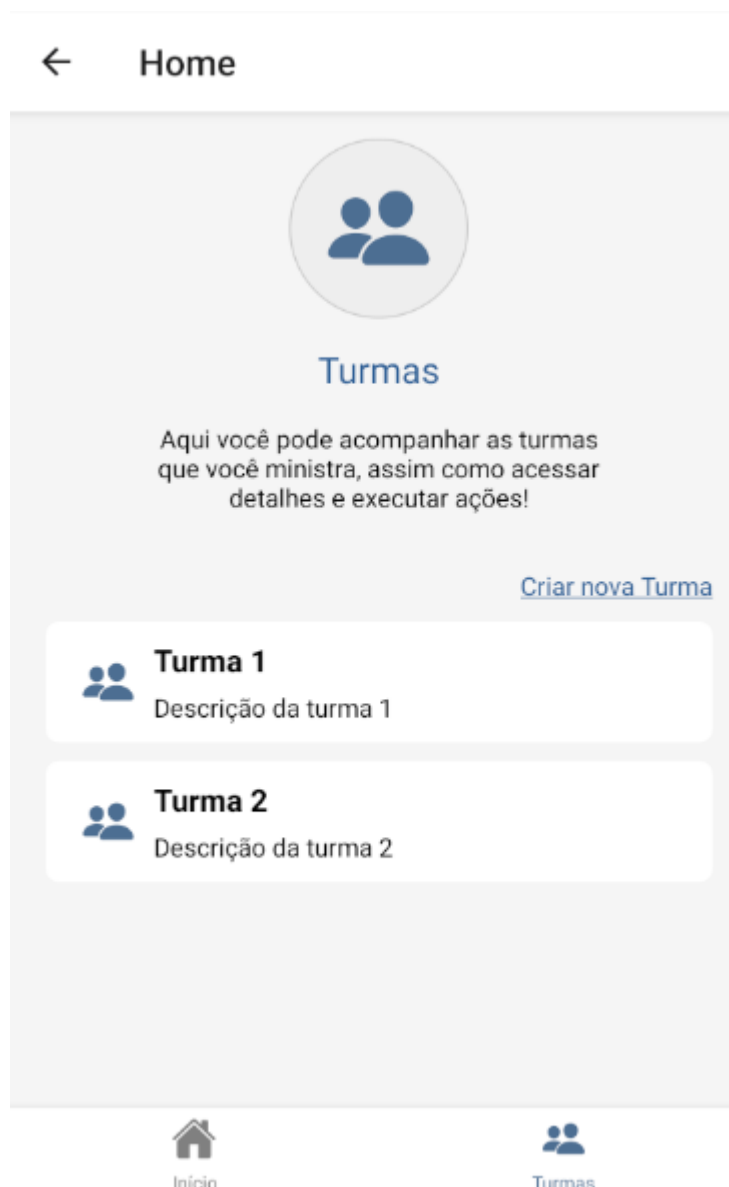
A Figura 11 representa a tela de Listagem de Atividade, ela é tela inicial do sistema, após o login, onde os usuários poderão ver todas as atividades que eles devem responder ou já foram respondidas. Seu conteúdo consiste em uma listagem com as informações das atividades relacionadas ao usuário, como: nome da atividade, o nome da turma a qual a atividade foi cadastrada e a data de entrega. No rodapé da tela, ainda está disponível um Barra de Abas onde ele pode navegar entre a tela atual e a tela de Listagem de Turmas.

Figura 11: Tela Listagem de Atividades

Fonte: Autor (2020).

5.1.4 LISTAGEM DE TURMAS

A Figura 12 representa a tela de Listagem de Turmas, ela é tela onde os usuários poderão visualizar todas as turmas a qual estão relacionados. Seu conteúdo consiste em um link, que redirecionará o usuário para a tela de Cadastro de Turma e uma listagem com as informações das turmas relacionadas ao usuário, como: nome da turma e descrição da turma. No rodapé da tela, ainda está disponível um Barra de Abas onde ele pode navegar entre a tela atual e a tela de Listagem de Atividades

Figura 12: Tela Listagem de Turmas

Fonte: Autor (2020).

5.1.5 CADASTRO DE TURMA

A Figura 13 representa a tela de Listagem de Turmas, ela é tela onde os usuários poderão visualizar todas as turmas a qual estão relacionados. Seu conteúdo consiste em um link, que redirecionará o usuário para a tela de Cadastro de Turma e uma listagem com as informações das turmas relacionadas ao usuário, como: nome da turma e descrição da turma. No rodapé da tela, ainda está disponível um Barra de Abas onde ele pode navegar entre a tela atual e a tela de Listagem de Atividades.

Figura 13: Tela Castro de Turma

← Criar Turma

Criar nova turma

Preencha corretamente os campos abaixo e clique em "Concluir"

Nome

Descrição

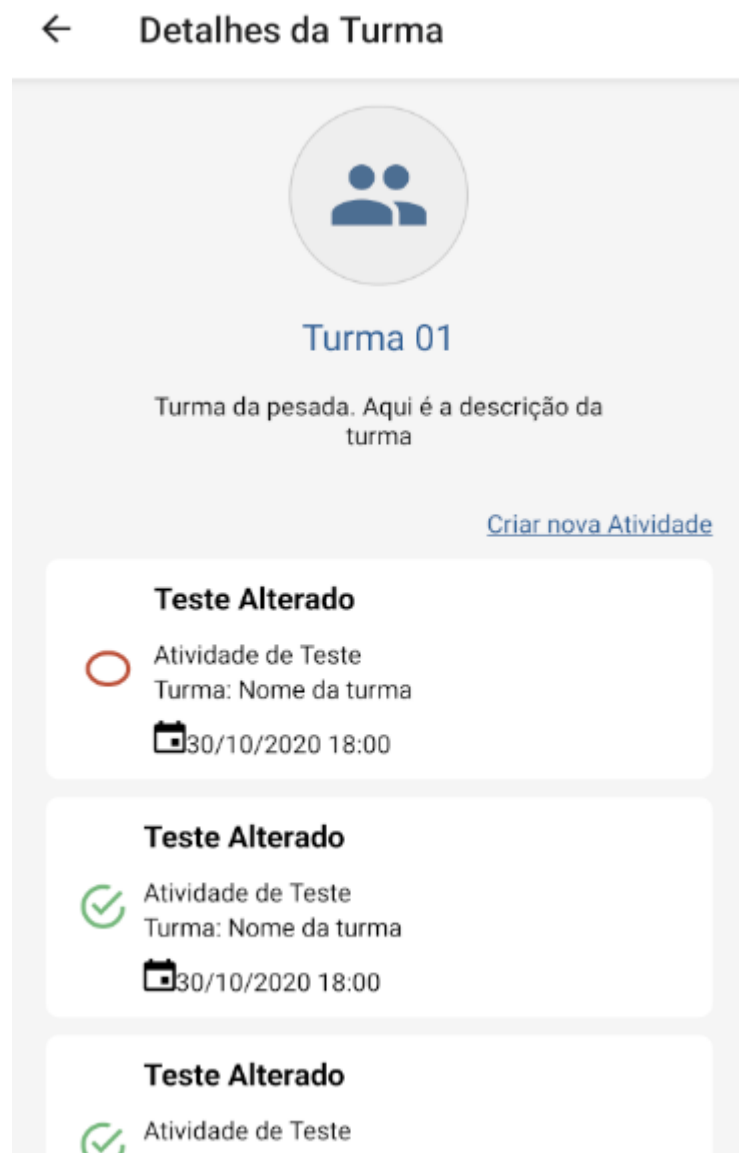
CONCLUIR

CANCELAR

Fonte: Autor (2020).

5.1.6 LISTAGEM DE ATIVIDADES RELACIONADAS A TURMA

A Figura 14 representa a tela de Listagem de Atividade, ela é tela inicial do sistema, após o login, onde os usuários poderão ver todas as atividades que eles devem responder ou já foram respondidas. Seu conteúdo consiste em um link para redirecionar o usuário para a tela de Cadastro de Atividade e uma listagem com as informações das atividades relacionadas ao usuário, como: nome da atividade, o nome da turma a qual a atividade foi cadastrada e a data de entrega.

Figura 14: Tela Listagem de Atividades Vinculadas a Turma

Fonte: Autor (2020).

5.1.7 CADASTRO DE ATIVIDADE

A Figura 15 representa a tela de Cadastro de Atividade, ela é tela onde os usuários poderão Cadastrar Atividades. Seu conteúdo consiste em um formulário com os campos de nome, descrição e data de entrega, a tela também possui dois botões, um confirmar seus dados e submetê-los, dado que os dados estejam corretas, e outro para redirecionar o usuário para a tela de Listagem de Atividades Relacionadas a determinada Turma.

Figura 15: Tela Cadastro de Atividade

← Criar Atividade

Criar nova atividade

Criando nova atividade para turma X.
Preencha corretamente os campos
abaixo e clique em "Concluir"

Nome

Descrição

Data de Entrega

CONCLUIR

CANCELAR

Fonte: Autor (2020).

5.2 TESTES ALGORITMOS EXTRAÇÃO DE TEXTO DE IMAGENS

Para os testes relacionados aos algoritmos de extração de imagens que serão abordados nesta seção, foram escolhidos os seguintes: o reconhecimento de texto do Firebase e o OCR e o plugin do Ionic Framework. Ambos algoritmos escolhidos funcionam no frontend. O plugin do Ionic Framework somente pode ser usado juntamente com o Ionic Framework, já o algoritmo do Firebase pode ser usado com o React Native e o Flutter.

Para um melhor levantamento de qual algoritmo se enquadra melhor no uso do projeto, foram realizados testes com 3 imagens, visando, também, selecionar o algoritmo que forneça

um melhor desempenho, pelo fato de o plugin do Ionic Framework ser executado diretamente no aparelho celular do usuário e o do Firebase ser executado em um servidor externo.

As 3 imagens para teste foram escolhidas de forma aleatória. Para contemplar uma maior abrangência nos testes, essas três imagens visam contemplar um cenário possível. Foi selecionada uma imagem com muito texto visando medir a diferença de tempo e performance, além e garantir que o aplicativo não trave sua execução com o algoritmo local.

Foi selecionada uma imagem com uma quantidade relativamente menor de texto, a fim de medir a diferença de tempo e precisão entre imagens e algoritmos. Por último, foi selecionado uma imagem com texto escrito a mão para analisar como os algoritmos irão se portar nesse cenário, vale ressaltar que já havia sido levantado a hipótese de ser inviável utilizar texto escrito a mão. Todos os testes foram realizados através de um emulador Android.

5.2.1 RESULTADOS DA IMAGEM COM MUITO TEXTO

A Figura 16 foi selecionada para os testes com uma imagem com muito texto.

Figura 16: Exemplo para Teste de Imagem com Muito Texto

"Não quero que nos tornemos estranhos. Não quero fingir que não te vi. Não quero viver como se você não existisse. Não quero ter que desligar o rádio quando a nossa música tocar. Olha pra mim e diz que a gente vai ficar bem. Diz que ainda me ama e também sente minha falta. Queria poder voltar atrás e mudar algumas coisas. Queria ter cuidado melhor do nosso amor, queria saber te mostrar o quanto é amado. É tão difícil acordar e não poder te mandar uma mensagem. Eu sempre via outras pessoas falando sobre ter perdido um amor e o quanto isso doia, e eu pensava "ei, levanta, você pode superar" e hoje eu entendo: só quem já perdeu um grande amor sabe o quanto dói. Esse frio na barriga, sensação de vazio, fadiga e vontade de que o tempo passe rápido, embora ele se arraste lentamente e pareça apertar meu coração contra a parede. Não sei o que dizer. Nem sei como terminar esse texto. Nem sei por que as coisas tem que terminar."

— **Drica Serra; a menina e o violão.**

Fonte: <https://br.pinterest.com/pin/656258976927130281>.

Através dos testes, obtivemos os seguintes resultados:

Tabela 1: Resultados dos Testes de Imagem com Muito Texto

Framework	Algoritmo	Tempo (média)	Resultado
Flutter	Firestore	01.10 segundos	"Não quero que nos tornemos estranhos. Não quero fingir que não te vi. Não quero viver como se você não existisse. Não quero ter que desligar o rádio quando a nossa música tocar. Olha pra mim e diz que a gente vai ficar bem. Diz que ainda me ama e também sente minha falta. Queria poder voltar atrás e mudar algumas coisas. Queria ter cuidado melhor do nosso amor, queria saber te mostrar o quanto é amado. É tão difícil acordar e

			não poder te mandar uma mensagem. Eu sempre via outras pessoas falando sobre ter perdido um amor e o quanto isso doia, e eu pensava "ei, levanta, você pode superar'e hoje eu entendo: só quem já perdeu um grande amor sabe o quanto dói. Esse frio na barriga, sensação de vazio, fadiga e vontade de que o tempo passe rápido, embora ele se arraste lentamente e pareça apertar meu coração contra a parede. Não sei o que dizer. Nem sei como terminar esse texto. Nem sei por que as coisas tem que terminar" Drica Serra; a menina e o violão.
React Native	Firebase	01.20 segundos	"Não quero que nos tornemos estranhos. Não quero fingir que não te vi. Não quero viver como se você não existisse. Não quero ter que desligar o rádio quando a nossa música tocar. Olha pra mim e diz que a gente vai ficar bem. Diz que ainda me amae também sente minha falta. Queria poder voltar atrás e mudar algumas coisas. Queria ter cuidado melhor do nosso amor, queria saber te mostrar o quanto é amado. É tão difícil acordar e não poder te mandar uma mensagem. Eu sempre via outras pessoas falando sobre ter perdido um amor e o quanto isso doia, e eu pensava "ei, levanta, você pode superar'e hoje eu entendo: só quem já perdeu um grande amor sabe o quanto dói. Esse frio na barriga, sensação de vazio, fadiga e vontade de que o tempo passe rápido, embora ele se arraste lentamente e pareça apertar meu coração contra a parede. Não sei o que dizer. Nem sei como terminar esse texto. Nem sei por que as coisas tem que terminar" Drica Serra; a menina e o violão.
Ionic Framework	Ionic	01.50 segundos	"Não quero que nos tornemos estranhos. Não quero fingir que não te vi. Não quero viver como se você não existisse. Não quero ter que desligar o rádio quando a nossa música tocar. Olha pra mim e diz que a gente vai ficar bem. Diz que ainda me amae também sente minha falta. Queria poder voltar atrás e mudar algumas coisas. Queria ter cuidado melhor do nosso amor, queria saber te mostrar o quanto é amado. E tão difícil acordar e não poder te mandar uma mensagem. Eu sempre via outras pessoas falando sobre ter perdido um amor e o quanto isso doia, e eu pensava "ei, levanta, você pode superar'e hoje eu entendo: só quem já perdeu um grande amor sabe o quanto dói. Esse frio na barriga, sensação de

			vazio, fadiga e vontade de que o tempo passe rápido, embora ele se arraste lentamente e pareça apertar meu coração contra a parede. Não sei o que dizer. Nem sei como terminar esse texto. Nem sei por que as coisas tem que terminar." Drica Serra; a menina e o violão.
--	--	--	---

Fonte: Autor (2020)

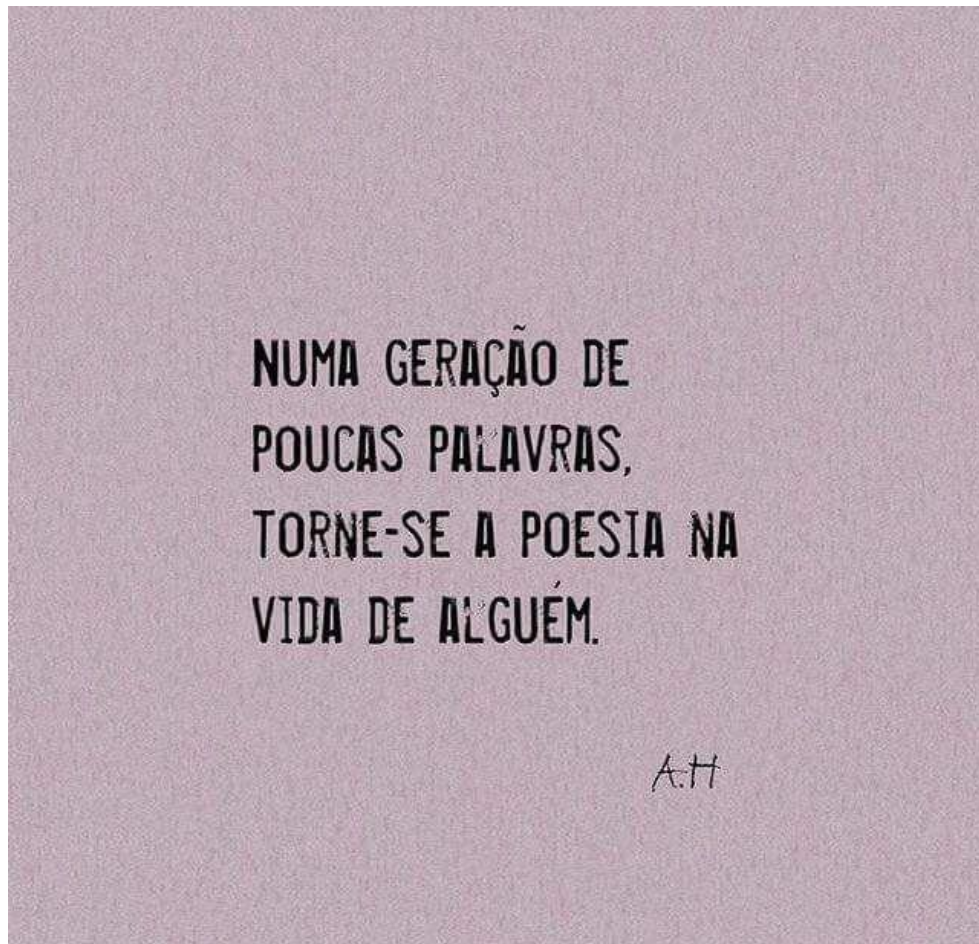
Segundo os resultados obtidos, pode-se observar que independente do algoritmo utilizado, a fidelidade da extração se mantém relativamente alta. Pode-se reparar, também, pontos em que a extração não foi feita corretamente, a maioria por concatenar um espaço em branco. Tais fatores reforçaram a ideia do reconhecimento do texto das imagens com algoritmos de forma impressa.

O tempo para extração também não destoa tanto entre eles, até o caso em que houve mais demora para extração do texto foi em que a extração ocorre no aparelho celular do usuário. Importante ressaltar que a demora é algo que afeta muito a experiência do usuário, podendo causar rejeição do aplicativo por parte deles.

5.2.2 RESULTADOS TESTE DA IMAGEM COM POUCO TEXTO

A Figura 17 foi selecionada para os testes com uma imagem com pouco texto.

Figura 17: Exemplo para Teste de Imagem com Pouco Texto



Fonte: <https://br.pinterest.com/pin/647392515159941689/>.

Tabela 2: Resultados dos Testes de Imagem com Pouca Texto

Framework	Algoritmo	Tempo (média)	Resultado
Flutter	Firebase	0.50 segundos	"NUMA GERAÇÃO DE POUCAS PALAVRAS, TORNE-SE A POESIA NA VIDA DE ALGUEM AH"
React Native	Firebase	0.50 segundos	"NUMA GERAÇÃO DE POUCAS PALAVRAS, TORNE-SE A POESIA NA VIDA DE ALGUEM AH"
Ionic Framework	Ionic	0.60 segundos	NUMA GERAÇÃO DE POUCAS PALAVRAS, TORNE-SE A POESIA NA VIDA DE AGUEM. At

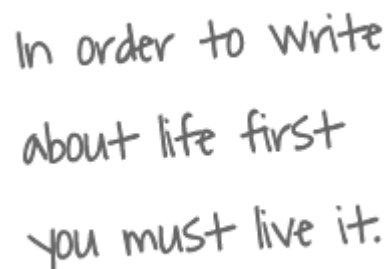
Assim como nos resultados do primeiro teste, a fidelidade da extração se mantém relativamente alta. Pode-se reparar, também, pontos em que a extração não foi feita corretamente, como os casos de acentuação, mas deve-se considerar que as acentuações não estão muito nítidas nas imagens também.

Como esperado, o tempo se manteve muito parecido também com o exemplo anterior e a extração ocorreu mais rápido que a primeira imagem, por ter menos texto também. Novamente o caso em que houve mais demora para a extração foi onde ela ocorre no aparelho celular do usuário.

5.2.3 RESULTADOS TESTE DA IMAGEM COM TEXTO ESCRITO A MÃO

A Figura 18 foi selecionada para os testes com uma imagem com escrita à mão.

Figura 18: Exemplo para Teste de Imagem com Texto Escrito a Mão



In order to write
about life first
you must live it.

Fonte: <https://www.myscript.com/pt/handwriting-recognition/>.

Tabela 3: Resultados dos Testes de Imagem com Texto Escrito a Mão

Framework	Algoritmo	Tempo (média)	Resultado
Flutter	Firebase	0.50 segundos	"In orde v to Wite aboutlife first YOUu MUST ve it."
React Native	Firebase	0.60 segundos	"In orde v to Wite aboutlife first YOUu MUST ve it."
Ionic Framework	Ionic	0.60 segundos	In order to Wite aboutlife first ou muS+ ive it.

Fonte: Autor (2020)

Como citado anteriormente, já havia sido levantado a hipótese de que a extração de texto escrito a mão seria inviável. Pode-se notar que, embora o tempo de extração se manteve baixo, a fidelidade acabou se tornando o grande ponto negativo. Pensando no uso do projeto, acabaria se tornando muito complexo na extração de texto de imagens por dois fatores, ou o usuário teria de enviar o algoritmo errado, já esperando erro, ou passaria um tempo considerável na correção do que foi extraído.

6 CONSIDERAÇÕES FINAIS

Com a programação e o ensino de algoritmos cada vez mais presente no ensino, e esses conceitos cada vez mais amparados por órgãos como a Sociedade Brasileira de Computação, torna-se cada vez mais viável o uso de metodologias que possam fugir dos conceitos ensinados por uma instituição de ensino superior.

Esta pesquisa apresentou uma aplicação que ajude o ensino de programação em escolas em diversos níveis de ensino. Até buscando atender o atual cenário das escolas brasileiras, onde não há um computador para todos os alunos de uma turma, como já citado no trabalho.

A aplicação visou abordar conceitos Pensamento Computacional utilizando Computação Desplugada e é por utilizar, principalmente, esse segundo conceito que a aplicação acaba se tornando viável nas escolas. Visando esses conceitos foram prototipadas e desenvolvidas as telas. Também foram feitos testes com os algoritmos de extração de imagem para elencar a melhor solução a ser aplicada no projeto, além de selecionar os algoritmos, foram levantados dados para embasamento da melhor alternativa para os textos, descartando, assim, os textos escritos a mão.

Como já abordado no trabalho, há algumas limitações durante a construção da aplicação, como não haver muitos trabalhos que abordem a temática, principalmente trabalhos práticos. Outras limitações estão vinculadas aos algoritmos de extração de texto da imagem. Uma dessas limitações é sobre o custo, tanto financeiro quanto de hardware, pois ou os algoritmos são de terceiros e são cobrados a partir de um certo número de imagens processadas, ou os algoritmos têm de ser hospedados em servidores, o que precisa de um servidor com maior poder computacional. Outro ponto referente aos algoritmos é a falta de precisão na extração de texto escrito a mão, o que gera a necessidade de se trabalhar com peças físicas e/ou impressas.

6.1 TRABALHOS FUTUROS

A primeira fase do projeto contemplou a elaboração de todos os requisitos para o desenvolvimento do aplicativo, listagem dos algoritmos para extração de texto de imagens e os seus respectivos testes e todos esses passos já estão contemplados neste trabalho.

Uma segunda fase do projeto, conforme já mencionado na seção de Metodologia, são os testes em sala de aula utilizando a aplicação e a extração de texto da imagem, ou seja, os algoritmos construídos pelos alunos são reconhecidos via imagem para a extração do texto,

visando atender o máximo possível de cenários que possam ocorrer. Além dos testes e obtenção dos feedbacks por parte dos alunos e professores, uma melhoria em questões de layout e dados também está prevista.

Com as correções da segunda etapa implementadas se iniciará a terceira fase do projeto. Esta etapa consiste na implementação da correção automática dos algoritmos. Assim atendendo as necessidades de criação de turmas e atividades, extração de texto das imagens e obtenção em tempo real do algoritmo extraído com feedback das linhas em que ocorreram acertos e erros para o docente e alunos. Como já abordado anteriormente, há uma lacuna de trabalhos que abordem essa temática, principalmente questão prática, em sala de aula. Isso acaba sendo um ponto a ser trabalhado, também, em trabalhos futuros.

REFERÊNCIAS

- A Model Curriculum for K–12 Computer Science: Report of the ACM K–12 Task Force Computer Science Curriculum Committee**, 2011. Disponível em <http://www.csta.acm.org/Curriculum/sub/CurrFiles/UaurioCSTA_K-12_CSS.pdf>. Acessado em: 23 out. 2020.
- ARAÚJO, D. C.; RODRIGUES, A. N.; SILVA, C. V. A.; SOARE, L. S. **O Ensino de Computação na Educação Básica apoiado por Problemas: Práticas de Licenciandos em Computação**, 2015. Disponível em: <<https://sol.sbc.org.br/index.php/wei/article/view/10229>>. Acessado em: 13 nov. 2020.
- ARAÚJO, A. L. S. O.; ANDRADE, W. L.; GUERRERO, D. D. S. **Um Mapeamento Sistemático sobre a Avaliação do Pensamento Computacional no Brasil**, 2018. Disponível em: <<https://bitly.com/R4wPG>>. Acessado em: 19 out. 2020.
- ARAÚJO, A. L. S. O.; ANDRADE, W. L.; GUERRERO, D. D. S.; MELO, M. R. A.; SOUZA, I. M. L. **Como Identificar Habilidades do Pensamento Computacional? Um Estudo Empregando Análise Fatorial**, 2018. Disponível em: <<http://dx.doi.org/10.5753/cbie.wcbie.2018.530>>. Acessado em: 19 nov. 2020.
- BELL, T., ALEXANDER, J., FREEMAN, I., GRIMLEY, M. (2009). Computer Science Unplugged: School students doing real computing without computers. The New Zealand Journal of Applied Computing and Information Technology, 13(1):20–29. Acessado em: 05 nov. 2020.
- BELL, TIMOTHY C., IARN H. WITTEN, and MIKE FELLOWS. Computer Science Unplugged: Off-line activities and games for all ages. Computer Science Unplugged, 1998. Acessado em: 19 out. 2020.
- BERNARDO, KLEBER. Estória de usuário. Você saberia contar?. 2014. Disponível em: <<https://www.culturaagil.com.br/estoria-de-usuario-voce-saberia-contar/>>. Acessado em: 13 nov. 2020.
- BLIKSTEIN, P. (2008). **O Pensamento Computacional e a Reinvenção do Computador na Educação**. Disponível em: <<http://bit.ly/1IXIbNn>>. Acessado em: 06 nov. 2020.
- BRACKMANN, C. P. **DESENVOLVIMENTO DO PENSAMENTO COMPUTACIONAL ATRAVÉS DE ATIVIDADES DESPLUGADAS NA EDUCAÇÃO BÁSICA**, 2017. Disponível em: <<https://lume.ufrgs.br/handle/10183/172208>>. Acessado em: 18 nov. 2020.
- BRASIL. (2017). INFRAESTRUTURA: Equipamentos, internet e condições gerais. <https://www.todospelaeducacao.org.br/tecnologia/equipamentos-tecnologia-escola/>. Acessado em: 26 out. 2020.
- BRASILEIRO, Roberto. Veja agora 08 dicas para criar excelentes histórias de usuário. 2017. Disponível em: <<http://www.metodoagil.com/historias-de-usuario/>>. Acesso em: 27 nov. 2018.
- BULCÃO, J. S. B.; ALVEZ, C.; NASCIMENTO C J.; MADEIRA, C. A. G.; CAMPOS, A. M. **Computação desplugada alinhada aos descritores de Matemática do SAEB: Um relato de**

experiência, 2019. Disponível em: <<https://www.br-ie.org/pub/index.php/wcbie/article/view/8982>>. Acessado em: 30 out. 2020.

CASTRO, R. M.; SIQUEIRA, S. W. M. **Metodologias, Técnicas, Ambientes e Tecnologias Alternativas utilizadas no Ensino de Algoritmos e Programação no Ensino Superior no Brasil**, 2019. Disponível em: <c>. Acessado em: 15 nov. 2020.

COSTA, T.; BATISTA, A.; MAIA, M.; ALMEIDA, L.; FARIAS, A. **Trabalhando Fundamentos de Computação no Nível Fundamental: experiência de licenciandos em Computação da Universidade Federal da Paraíba**, 2012. Disponível em: <<https://bityli.com/R4P36>>. Acessado em: 01 nov. 2020.

CORREA, J. S. **Implementação da História de Usuário: Experiência na Fábrica de Tecnologias Turing (FTT), 2018**. Disponível em: <<https://bityli.com/wLPnZ>>. Acessado em: 18 nov. 2020.

FRANÇA, R. S.; AMARAL, H. J. C. **Ensino de Computação na Educação Básica no Brasil: Um Mapeamento Sistemático**, jul. 2013. Disponível em: <<https://bityli.com/X22vB>>. Acessado em: 13 out. 2020.

FREIRE, L. P.; COUTINHO, J. C. S.; SILVA, V. M. L.; LIMA, N. C. A. **Uma Proposta de Encontros de Tutoria Baseada em Metodologias Ativas para Disciplinas de Programação Introdutória**, jul. 2019. Disponível em: <<https://bityli.com/koEmF>>. Acessado em: 13 out. 2020.

GIANNAKOS, M. N. (2013). Enjoy and Learn with Educational Games: Examining Factors Affecting Learning Performance, *Computers & Education*, 68, p. 429-439. Acessado em: 13 nov. 2020. . Acessado em: 15 nov. 2020.

MICHAEL, D. R., CHEN, S. L. (2005). *Serious Games: Games That Educate, Train, and Inform*, Cengage Learning PTR. Acessado em: 09 nov. 2020.

OLIVEIRA, V. M.; JUNIOR, J. R. A. A.; FILHO, A. O. B. **Aplicação de Métodos Lúdicos para o Desenvolvimento e Avaliação da Capacidade de Pensamento Algorítmico em Crianças**, 2019. Disponível em: <<https://www.br-ie.org/pub/index.php/wcbie/article/view/8963>>. Acessado em: 15 nov. 2020.

ORTIZ, J. S. B.; MOREIRA, C.; PEREIRA, R. **Aspectos do Contexto Sociocultural dos Alunos estão Presentes nas Pesquisas para Ensinar Pensamento Computacional?**, jul. 2018. Disponível em: <<https://bityli.com/koEmF>>. Acessado em: 13 out. 2020.

PRESSMAN, R. S. *Engenharia de Software: Uma Abordagem Profissional*. Sétima Edição. AMGH Editora Ltda., 2011. . Acessado em: 03 nov. 2020.

RODRIGUES, S. R. C.; ARANHA, E.; SILVA, T. R. **Computação Desplugada no Ensino de Programação: Uma Revisão Sistemática da Literatura**, 2019. Disponível em: <<https://www.br-ie.org/pub/index.php/sbie/article/view/7998>>. Acessado em: 20 out. 2020.

SANTOS, E. R.; SOARES, G.; BIANCO, G. D.; FILHO, J. B. R.; LAHM, R. A. **Estímulo ao Pensamento Computacional a partir da Computação Desplugada: uma proposta para Educação Infantil**, 2016. Disponível em: <<https://relatec.unex.es/article/view/2584/1929>>. Acessado em: 15 nov. 2020.

SBC. **Diretrizes para ensino de Computação na Educação Básica**, 2019 Disponível em <<https://bitly.com/2X7ay>>. Acessado em: 10 nov. 2020.

SILVA, D. J. G. M.; GUARDA, G. F. **CriptoData: Ensino de Criptografia via Computação Desplugada**, 2019. Disponível em: <<https://br-ie.org/pub/index.php/sbie/article/view/8733>>. Acessado em: 19 out. 2020.

SOMMERVILLE, I. Engenharia de Software. Nona Edição. Pearson Education do Brasil, 2012. Acessado em: 13 nov. 2020.

SOUZA, J. S.; LOPES, A. S. B. **Estimulando o pensamento computacional e o raciocínio lógico no ensino fundamental por meio da OBI e computação desplugada**, 2018. Disponível em: <<https://www.br-ie.org/pub/index.php/sbie/article/view/8186>>. Acessado em: 16 nov. 2020.

SOUZA, F. F. S.; NUNES, M. A. S. N. **Práticas e resultados obtidos na aplicação do Pensamento Computacional Desplugado no ensino básico: Um Mapeamento Sistemático**, 2019. Disponível em: <<https://br-ie.org/pub/index.php/sbie/article/view/8733>>. Acessado em: 19 out. 2020.

WING, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3):33–35. Acessado em: 16 nov. 2020.

YADAV, A., STEPHESON, C., and HONG, H. (2017). **Computational thinking for teacher education. Communications of the ACM**, 60(4):55–62. Acessado em: 13 nov. 2020.

ZANETTI, H. A. P.; BORGES, M. A. F.; RICARTE, I. L. M. **Pensamento Computacional no Ensino de Programação: Uma Revisão Sistemática da Literatura Brasileira**, 2016. Disponível em: <<https://www.br-ie.org/pub/index.php/sbie/article/view/6677>>. Acessado em: 19 out. 2020.