

ESTUDO E IMPLEMENTAÇÃO DE UM APLICATIVO EM FLUTTER QUE REALIZE A INTERPRETAÇÃO DE IMAGENS USANDO MACHINE LEARNING

PIOVEZAN, Guilherme Aparecido Ferreira,
PONTE, Vitor Kemuel Carreiro
DEZANI, Henrique

e-mail:

guilherme.piovezan@fatec.sp.gov.br; vitor.ponte@fatec.sp.gov.br;
henrique.dezani@fatec.sp.gov.br

Resumo: Uma das dificuldades encontradas atualmente é a procura e consulta de livros à distância, bem como a construção de referências. Para facilitar isto pensou-se em um aplicativo que, usando *Machine Learning* auxiliasse esse processo. O sistema visa identificar livros por meio de uma foto tirada de sua capa e por este meio o usuário poderá ter acesso aos seus dados (autor, editora, data e local de publicação, resumo, entre outros). O projeto consiste na criação de uma aplicação, utilizando duas tecnologias disponibilizadas pelo Google, o Flutter (utilizada na criação de aplicativos móveis), e a plataforma Firebase, que apresenta ferramentas como *Machine Learning*, capaz de reconhecer textos em imagens. Seu público-alvo será as livrarias, além de que poderá ser ofertado para pessoas que gostam da leitura em seu dia a dia (consumidor final), criando seu repositório de livros no aplicativo.

Palavras-chave: Aplicativo móvel. Reconhecimento de Textos em imagens. *Machine Learning*. Busca por livros. Livrarias.

Abstract: One of the difficulties encountered today is the search and consultation of books at a distance, as well as the construction of references. To facilitate this, we thought of an application that, using *Machine Learning*, would help this process. The system aims to identify books by means of a photo taken from their cover and through this means the user can have access to their data (author, publisher, date and place of publication, abstract, among others). The project consists of creating an application, using two technologies made available by Google, Flutter (used in the creation of mobile applications), and the Firebase platform, which features tools such as *Machine Learning*, capable of recognizing texts in images. Its target audience will be bookstores, and it can be offered to people who enjoy reading in their daily lives (end consumers), creating their book repository in the application.

Keywords: Mobile app. Recognition of Texts in Images. *Machine Learning*. Search for books. Bookstores.

1 Introdução

O termo *Machine Learning*, que traduzindo diretamente para o português significa aprendizado de máquina, foi criado por Samuel (2000) que descreve o conceito como “um campo de estudo que concede aos computadores a habilidade de aprender sem terem sido programados para tal”. Diante disso podemos entender o termo como uma área da ciência que desenvolve sistemas para treinar algoritmos com o intuito de realizar tarefas com precisão e eficiência com base no raciocínio lógico humano. Tal estudo distingue o termo em duas vertentes, do que é ser uma *Machine Learning*, onde alguns especialistas afirmam: fazer com que as máquinas “pensem” da forma como humano e outros como ferramenta de tomada de decisão a partir de algoritmos que identificam padrões de comportamento assemelhando-se ao raciocínio lógico.

Turim (1950) trouxe uma pergunta à tona: ao invés de perguntar se elas podem pensar, por que não perguntamos se elas podem passar por um teste de comportamento. Então ele criou um teste onde o examinador e o examinado conversam através de um teclado. Neste experimento, o examinador é um humano, porém não sabemos se o examinado é uma máquina ou um humano, o algoritmo passa no teste se aquele que está interrogando-o não tem a capacidade de dizer através das respostas se é um ser humano ou um computador.

Nos dias atuais todos andamos com um *smartphones* que nos possibilitam navegar na internet, escrever documentos, realizar vídeo chamada, fazer compras por aplicativos, navegar nas redes sociais entre outras atividades rotineiras de usuários. O que torna esse tipo de ferramenta mais presente na vida dos usuários, analisando seus comportamentos, como: de compras, de consumos, redes sociais, jogos e aplicativos, além de traçar um tipo de perfil singular para todas as pessoas, gerando uma melhor experiência de usuário.

2 Justificativa

Segundo Melo, o Brasil demonstra dificuldades em relação à formação e educação de qualidade, influenciando muito pouco o jovem à prática da leitura, o presente trabalho justifica-se pela necessidade de desenvolver um aplicativo com ênfase na consulta de livros visando agilidade e autonomia na busca de obras, fomentando a cultura despertando o interesse na leitura por parte do público-alvo interessado.

3 Objetivos

O algoritmo de *Machine Learning* será um dos principais objetos de estudo proporcionando a elaboração de uma ferramenta para auxiliar na identificação de textos em imagens, por exemplo. Assim, o objetivo geral do estudo é desenvolver um aplicativo para dispositivos móveis, capaz de detectar livros a partir de uma fotografia tirada da capa da obra. Dispondo de ferramentas de *Machine Learning*, para favorecer o público de leitores, trazendo características e avaliações, como: autor(res), editora, data da publicação, resumo, valor de compra, livros semelhantes, entre outros atributos importantes.

4 Fundamentação Teórica

1. Flutter

Flutter (2022) é um *framework* criado inicialmente para o desenvolvimento de aplicações móveis – para *web*, *smartwatch*, *desktop*, televisão – de forma “híbrida”, isto é, podem ser gerados códigos tanto para dispositivos com sistema *iOS* quanto para *Android*, fazendo uso da linguagem de programação denominada *Dart*, criada também pela Google e que possui uma sintaxe muito similar ao Java, C# e *TypeScript*.

2. Firebase

Firebase (2022), é uma plataforma criada sobre a infraestrutura do Google, com base em um conceito chamado *Backend as a Service* (Baas), isto é, se consolida como um serviço de computação em nuvem que atua como um *middleware*, disponibilizando um conjunto de serviços para aplicativos móveis e *web* prontos, podendo consumir o serviço de autenticação, armazenamento de arquivos, banco de dados em tempo real, *functions* (*serverless*) e *Machine Learning* que será utilizado neste trabalho.

3. Machine Learning

Segundo Silveira e Bullock (2017), *Machine Learning*, é um conceito associado à Inteligência Artificial, razão por estar em alta nos últimos anos, consistindo em mecanismos computacionais que se baseiam no comportamento humano para resolver problemas. Tal ferramenta se resume à um algoritmo que será capaz de tomar decisões com base em análise de dados e estatística, seja o problema, para classificar algo, validar um modelo, entre outras possibilidades.

4. Application Programming Interface

De acordo com Pereira (2016), *Application Programming Interface* (API), que pode ser compreendida como uma interface de programação de aplicação, é um conjunto de normas e padrões que possibilita a comunicação entre plataformas através de uma série de regras estabelecidas por uma equipe de desenvolvedores, ou até por um padrão já existente na comunidade. Como dito anteriormente, as APIs permitem que desenvolvedores criem *Softwares* (aplicativos, programas e plataformas diversas) de forma mais simples e prática, que, basicamente, tem a função de facilitar e simplificar o trabalho da equipe de desenvolvimento, além de fornecerem um papel importante na segurança, já que são capazes de bloquear acessos e permissões a dados de *software* e *hardware*.

5. Google Books API

Segundo o Google, a APIs de livros, é um repositório de obras mantido pela empresa, podendo pesquisar exemplares e recuperar informações, visibilidade e disponibilidade de e-books. A API integra aplicativos desenvolvidos a seu banco de dados, proporcionando aos desenvolvedores de *software* disponibilizarem as aplicações tanto em ambiente *web* quanto *mobile*, permitindo ‘incorporar’ a visualização de livros em sites ou aplicativos.

5 Trabalhos Similares

No século passado (Século XX) vários pesquisadores e entusiastas da área tinham trazido à tona vários temas e artigos interessantes sobre como e onde utilizar a tecnologia. Na atualidade podemos ver vários lugares onde é utilizada a Inteligência Artificial, para os mais distintos objetivos.

O estudo feito por Bruner e Falco (2018) trata de um ambiente *mobile* onde se tem a possibilidade de cadastrar novas turmas de alunos, cada aluno tem pelo menos uma foto sua salva na hora de ser registrado na turma e por meio da aplicação de Inteligência Artificial se tem a possibilidade de realizar a chamada de presença de toda a turma através de uma foto tirada da sala de aula. Vale ressaltar a experiência que tiveram em sua fase de teste, em uma sala onde havia duas irmãs gêmeas. O sistema foi capaz de diferenciá-las e com isso ambas tiveram suas presenças registradas através da imagem.

O artigo escrito por Oliveira Sobrinho, Menezes, Nascimento e Júnior (2016), descreve os desafios que o grupo teve na criação de um sistema que conhece placas de trânsito. Após testes e ajustes no treinamento que foi feito com imagens criadas artificialmente, conseguiram alcançar com seu modelo uma acurácia de 99,98%.

Schneider (2020) em seu projeto, nos traz um pouco da sua experiência desenvolvendo um sistema de aprendizado de máquina. Seu relato nos mostra como ele fez para criar um sistema que reconhece construções em uma imagem proveniente de um mapa, o algoritmo tem o papel de transformar a foto recebida em uma imagem em preto e branco, onde cada pedaço branco na imagem se trata de um edifício.

O relato descreve os métodos utilizados ao longo do processo de desenvolvimento, houve algumas trocas nos algoritmos utilizados e o modelo final chegou a uma acurácia de 89% de acerto. Tendo em vista as dificuldades passadas e a resolução das imagens utilizadas, que foram de 5000x5000 o autor desenvolveu um ótimo trabalho mostrando como é importante testar as várias formas de chegar ao resultado.

Podemos ver com as obras citadas como é possível ter várias aplicações da tecnologia em nosso dia a dia, ainda é importante citar que todos foram utilizando o meio de imagem para resolver seus dilemas.

6 Metodologia

O trabalho desenvolvido consiste em uma pesquisa prática aplicada ao desenvolvimento de um sistema computacional, especificamente para dispositivos móveis, utilizando a tecnologia Flutter e o algoritmo de Machine Learning, disponibilizado pela plataforma Firebase.

Para o desenvolvimento do aplicativo utilizou-se da linguagem de programação Dart junto ao framework Flutter. Para o backend da aplicação utilizou os serviços em cloud do Firebase, como o banco de dados Firestore, Machine Learning para captura de textos em imagens. O código foi editado a partir de uma IDE (Integrated Development Environment), IntelliJ IDEA Community Edition, Visual Studio Code ou Android Studio.

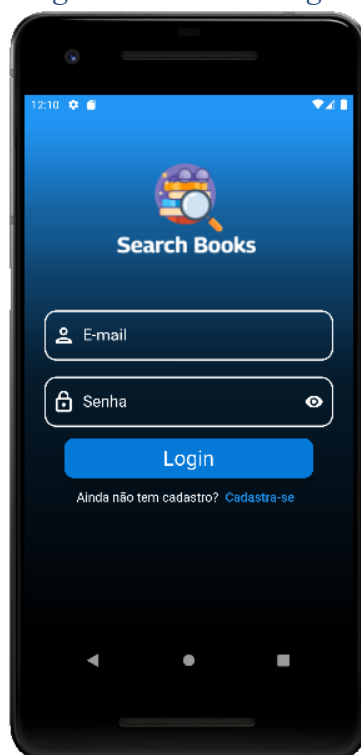
7 Desenvolvimento

O desenvolvimento do projeto iniciou-se a partir da criação da aplicação em *Flutter*, no qual será demonstrada os estudos realizados. O aplicativo desenvolvido tem como objetivo auxiliar leitores na consulta de livros a partir de uma fotografia tirada de sua capa, obtendo a foto a partir do aplicativo, utiliza-se do serviço de *Machine Learning* disponibilizado pelo *Firebase*, para realizar o reconhecimento de textos na imagem, partindo para a busca em uma

API (Application Programming Interface) – será utilizado o *Google Books API* – exibindo uma lista de livros semelhantes, ao consultado, dando oportunidade ao usuário fazer a escolha do exemplar.

Na Figura 1 é apresentada a tela onde o usuário irá realizar o *login*, se houver cadastro, no aplicativo. Caso o usuário não possua *login*, ele poderá clicar em ‘Cadastra-se’ destacado em azul para colocar suas informações de cadastro.

Figura 1 – Tela de Login



Fonte: Dos autores.

A autenticação de usuário fica responsável por um serviço do *Firebase*, para usufruir, precisa-se criar um projeto na plataforma (não será detalhado o passo-a-passo para a criação de um projeto no *Firebase*) e vinculá-lo ao aplicativo, além de instalar algumas dependências como mostra a Figura 2, as principais sendo ‘*firebase_core*’ e ‘*firebase_auth*’ para que o serviço de autenticação passa a ser funcional.

Figura 2 – Dependências do projeto

```
dependencies:
  flutter:
    sdk: flutter

  cupertino_icons:
  firebase_core: ^1.22.0
  firebase_auth:
  cloud_firestore:
  form_field_validator: ^1.1.0
  image_picker:
  camera_camera: ^2.0.1
  flutter_firebase_ml_vision: ^0.0.1
  google_ml_kit: ^0.12.0
  google_mlkit_text_recognition: ^0.4.0
  http: ^0.13.4
```

Fonte: Dos autores.

Com as dependências instaladas e o vínculo do aplicativo com a plataforma do *Firebase*, pode-se desenvolver o método para autenticação evidenciada na Figura 3.

Figura 3 – Método de login

```

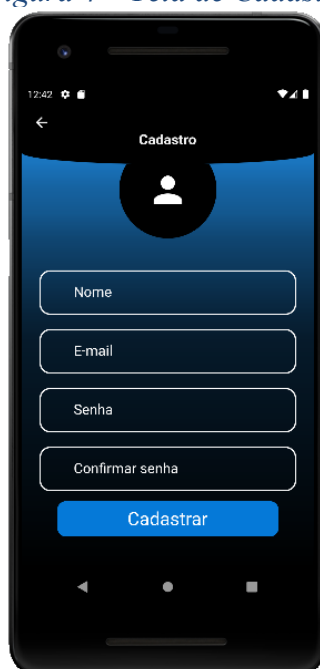
1 login(BuildContext context, String email, String password) async {
2   try {
3     await _auth.signInWithEmailAndPassword(
4       email: email.toString(),
5       password: password.toString(),
6     );
7   } on FirebaseAuthException catch (e) {
8     if (e.code == 'user-not-found') {
9       throw AuthenticationException('Usuário não encontrado.');
```

Fonte: Dos autores.

O método recebe os dados de acesso a partir de argumentos (*e-mail*, *password*), provenientes dos campos de textos na tela de *login*. Na linha 2, chama-se a função do *Firebase* responsável por fazer o *login*, caso o usuário forneça dados incorretos, sendo e-mail ou senha, o aplicativo retorna uma *pop-up* informando ao usuário do motivo do problema.

Na Figura 4, é apresentada a tela de cadastro de usuário novo. O cadastro de um usuário se deve ao fato de que o aplicativo precisará de seu identificador, para realizar a filtragem de livros já consultados por ele na tela Home (que ficará o histórico de consultas). Para realizar o cadastro basta informar o nome, e-mail, senha e confirmar a senha, logo em seguida clicar no botão ‘Cadastrar’ destacado em background azul.

Figura 4 – Tela de Cadastro



Fonte: Dos autores.

Seguindo a mesma lógica de autenticação, o *Firebase* possui um serviço para realizar o cadastro de usuário, informando o e-mail e senha a plataforma realizar o cadastro, logando automaticamente, no entanto, para guardar informações do usuário como o nome, criamos uma coleção de dados *'users'* para armazenar os atributos, no método escrito exibido na Figura 5.

Figura 5 - Método de registro de usuário

```

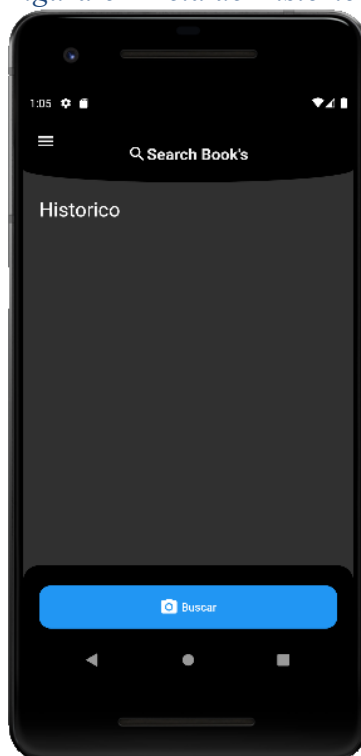
1  register(
2      String email, String senha, UserModel model, BuildContext context) async {
3      try {
4          await _auth.createUserWithEmailAndPassword(email: email, password: senha);
5      } on FirebaseAuthException catch (e) {
6          if (e.code == 'weak-password') {
7              throw AuthenticationException('A senha é muito fraca!');
8          } else if (e.code == 'email-already-in-use') {
9              throw AuthenticationException('Este email já está cadastrado');
10         }
11     }
12     firestore.collection('users').doc(userIsLogaded().uid).set({
13         "nome": model.nome,
14         "uid": userIsLogaded().uid,
15     });
16     Navigator.of(context).push(
17         MaterialPageRoute(
18             builder: (context) => const HomeView(),
19         ),
20     );
21 }

```

Fonte: Dos Autores.

Logo em seguida, o usuário será redirecionado para a tela *Home*, como mostrado na Figura 6.

Figura 6 – Tela de Histórico



Fonte: Dos autores.

Ao *'logar'*, o usuário será redirecionado para a tela de histórico de consultas de livros já realizados por ele. Nesta tela serão listados todos os livros consultados, permitindo o usuário

visualizar a obra novamente, além de realizar novas consultas, clicando no botão ‘Buscar’ em *background* azul.

Ao clicar no botão ‘Buscar’ na tela de histórico de livros, o usuário será redirecionado para a câmera nativa do dispositivo, podendo registrar a fotografia da capa do livro que se deseja consultar como mostrado na Figura 7.

Figura 7 - Fotografando a capa do livro



Fonte: Dos Autores

Na figura 8, observa-se a implementação do método para abrir a câmera do dispositivo nativo para capturar a fotografia.

Figura 8 – Método para abrir câmera nativa

```
1 Future<XFile?> pickImage() async =>
2   await ImagePicker().pickImage(source: ImageSource.camera);
```

Fonte: Dos autores.

O próximo passo é interpretar textos na imagem captura pela câmera do dispositivo, para tanto precisa-se instalar algumas dependências (como mostrado na Figura 9), sendo elas ‘*google_ml_kit*’ e ‘*google_mlkit_text_recognition*’. Com as dependências instaladas, a figura 10, exibirá o método de reconhecimento de textos desenvolvido para realizar a interpretação da imagem e a captura dos textos na imagem.

Figura 9 - Dependências necessárias para o desenvolvimento

```
google_ml_kit: ^0.12.0
google_mlkit_text_recognition: ^0.4.0
```

Fonte: Dos autores.

No método *readTextFromImage*, apresentado na Figura 10, é utilizado efetivamente a *Machine Learning*, responsável pela interpretação e captura de textos na imagem. A captura do caminho da imagem (linha 4) é feita a partir da classe *InputImage*, para logo em seguida, utilizar o objeto no método *processImage* de *_textRecognizer*, que definitivamente irá retornar o texto capturado.

Figura 10 - Método de reconhecimento de textos em imagens

```

1 Future<String?> readTextFromImage(XFile imageFile) async {
2   InputImage inputImage;
3
4   inputImage = InputImage.fromFilePath(imageFile.path);
5
6   final recognizedText = await _textRecognizer.processImage(inputImage);
7
8   return recognizedText.text;
9 }

```

Fonte: Dos autores.

Na Figura 11, demonstra o método responsável por chamar as funções de capturar um arquivo de imagem e enviar para a função de captura de textos na imagem, chamando-a no ao clicar no botão 'Buscar' na tela Home/Histórico de consulta.

Figura 11 - Método *getTextImage*

```

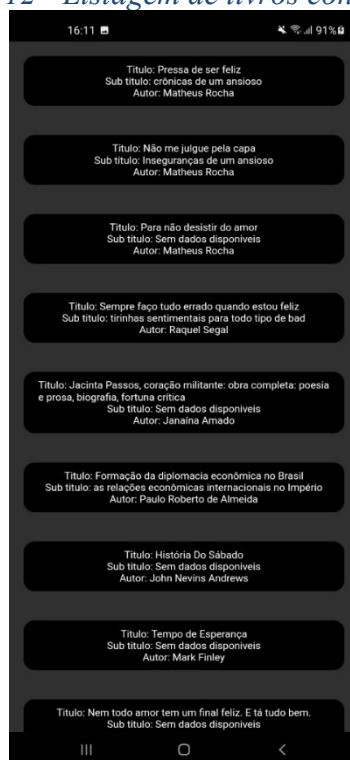
1 void getTextImage() async {
2   var pickedFile = await pickImage();
3   var originalText = await readTextFromImage(pickedFile!);
4
5   setState(() {
6     text = originalText!;
7   });
8
9   searchBooks();
10 }

```

Fonte: Dos autores

Com o texto capturado o aplicativo fica encarregado de consultar o título do livro na *API* do *Google Books*, ficando responsável pelo serviço de consulta. Para realizar a busca utilizamos a seguinte url: 'https://www.googleapis.com/books/v1/volumes?q={titulo_livro}', onde *titulo_livro*, é o parâmetro que será o texto capturado pela *Machine Learning*, então realizando a consulta e retornando uma listagem de livros semelhantes ao consultado, como mostrado na Figura 12.

Figura 12 - Listagem de livros consultados



Fontes: Dos autores.

A partir do momento que a API retorna a lista de livros consultados, pode-se selecionar o item desejado a visualização pelo usuário. Selecionando o primeiro livro da lista consultada, que se assemelha a obra fotografada na Figura 7, o usuário será encaminhado para a tela de detalhes, exibido na Figura 13.

Figura 13 - Tela de detalhes do livro



Fonte: Dos autores.

Como mostrado na Figura 13, pode-se observar detalhes da obra, como o volume que o identifica, o título, subtítulo se houver, ISBN (*International Standard Book Number*, sendo um sistema internacional de identificação de livros), data de publicação, editora, autor ou autores, quantidade de páginas, idioma, sinopse, e um botão, cuja função é redirecionar para o *Google Books* podendo visualizar mais informações, além de que o livro passar a fazer parte do histórico de consultas do usuário como mostra a Figura 14.

Figura 14 - Histórico de consultas



Fontes: Dos autores.

8 Resultados e Discussões

Com o objeto de estudo abordado, desenvolveu-se um aplicativo para dispositivos móveis, que realiza consultas de livros, através de fotografias tiradas da capa da obra, adendo ao auxílio de um algoritmo de *Machine Learning*, disponibilizada pela plataforma *Firebase*, capaz de identificar textos em imagens, apresentando os dados ao usuário final, como: editora, valor monetário, autor, data de lançamento, local de publicação, entre outras características, sendo um potencial produto tecnológico para a sociedade ou para fins de pesquisas relacionados a área de algoritmo de *Machine Learning*, *Machine Learning* e desenvolvimento de aplicativos com o *Framework Flutter*.

9 Conclusões

Logo o algoritmo de *Machine Learning* é um ramo da ciência da computação que busca, construir mecanismos e/ou dispositivos que simulem a capacidade de pensar e resolver problemas em nosso cotidiano. Com a evolução tecnológica e computacional, o algoritmo de *Machine Learning* ganhou forças, tendo em vista que seu desenvolvimento possibilitou um grande avanço na análise computacional, podendo a máquina chegar a fazer análises e síntese da voz humana.

Dessa maneira, com o presente projeto, conclui-se que o protótipo desenvolvido de um aplicativo capaz de identificar livros através de fotos tiradas da capa, atendeu as expectativas

para o público-alvo disponibilizado, gerando assim resultados satisfatórios e permitindo a comunidade que se baseiem na aplicação, e desenvolvam novas aplicações com base em inteligência artificial.

Referências

BRUNER, Luis Henrique; FALCO, Leonardo Jesus. **Estudo e implementação de um sistema de reconhecimento de faces utilizando os serviços cognitivos da Microsoft**. CONIC, 2018.

GOOGLE. Firebase. [S. l.], 2022. Disponível em: https://firebase.google.com/?gclid=CjwKCAjwTKmaBhBMEiwAyINuwDoLi9z4ObThbEnoSER5WNpJIR_ujDCPCmITTSpuX7NM7FlXFW1h7RoCuK0QAvD_BwE&gclidsrc=aw.ds. Acesso em: 15 out. 2022.

_____. Flutter. [S. l.], 2022. Disponível em: https://flutter.dev/?gclid=CjwKCAjw-rOaBhA9EiwAUkLV4iGRRyt8ztIKflot_B-gx2qxNSgRvJuxkQ66J_QYlJs9q6MMQK9xkRoCZW0QAvD_BwE&gclidsrc=aw.ds. Acesso em: 17 out. 2022.

_____. Google APIs de livros. [S. l.], 2022. Disponível em: <https://developers.google.com/books>. Acesso em: 17 out. 2022.

MELO, J. N do Amaral. **A importância da leitura praticada: Uma Atitude Reflexiva para Formação do Leitor**.

OLIVEIRA SOBRINHO, Marcos Vinícius; MENEZES, Matheus Chaves; NASCIMENTO, Chrystian Gustavo Martins; BRAZ JÚNIOR, Geraldo. **Reconhecimento de Sinais de Trânsito Utilizando Deep Learning**. 2016. 6 f. TCC (Graduação) - Curso de Ciência da Computação, Universidade Federal do Maranhão, São Luís, 2016.

PEREIRA, Caio Ribeiro. **Construindo APIs REST com Node.js**. Rio de Janeiro: Casa do Código, 2016. 193 p.

SAMUEL, A. J. *Some Studies in Machine Learning Using the Game of Checkers*. Ibm Journal. *Ibm Journal of Research and Development*, pp. 206-226. jan. 2000.

SCHNEIDER, Thomas de Queiroz Barros. **Processamento de imagens aéreas para a detecção de construções com machine learning**, São Paulo, 2020.

SILVEIRA, Guilherme; BULLOCK, Bennett. *Machine Learning: Introdução à classificação*. [S. l.]: Casa do Código, 2017. 407 p. ISBN 978-85-94188-18-2.

TURING, A. M. I. Computing *Machinery and Intelligence*. Mind. Westbrook, p. 433-460. 01 out. 1950. Disponível em: <https://academic.oup.com/mind/article/LIX/236/433/986238>. Acesso em: 06 nov. 2022.