



COLÉGIO TÉCNICO DE CAMPINAS  
INFORMÁTICA INTEGRADO AO ENSINO MÉDIO



**Apprender:** aplicativo para auxílio no uso do celular

DANIEL HENRY MATHEUS IMAMURA  
FÁBIO ALVES DOS SANTOS  
VITOR OSAMU SHOJI

Campinas  
2022

DANIEL HENRY MATHEUS IMAMURA

FÁBIO ALVES DOS SANTOS

VITOR OSAMU SHOJI

**Apprender:** aplicativo para auxílio no uso do celular

**Versão Original**

Trabalho de Conclusão do Curso de  
Informática no Colégio Técnico de  
Campinas para a obtenção do título  
de Técnico em Informática.

Área de concentração: Inteligência  
artificial.

Orientadora: Simone Pierini Facini  
Rocha

Co-orientador: Sérgio Luis Moral  
Marques

Campinas

2022

## RESUMO

A pesquisa do aplicativo “Apprender” consiste em uma tentativa de introduzir o básico do uso de *smartphones* para a população que não é tão familiarizada com essa tecnologia. Para isso, foi estudado a causa do problema, segundo a Revista Ceuma Perspectivas, em 2017, mais de 87,5% dos idosos utilizam telefones alfanuméricos, pois não se adequaram aos *smartphones*. Esse problema pode ter sido causado pela dificuldade da população idosa a se adequar à Era da Tecnologia. Um aplicativo que ensina a utilizar aplicativos. Como seria possível fazer essa síntese funcionar? Depois de muitos estudos, concluímos que uma interface comunicativa e de fácil entendimento pode superar alguns estranhamentos do público.

## **SUMÁRIO**

<b>1 INTRODUÇÃO</b>	<b>4</b>
1.1 TEMA	4
1.2 JUSTIFICATIVA	4
1.3 OBJETIVOS	5
1.3.1 Objetivos específicos	5
<b>2 REFERENCIAL TEÓRICO</b>	<b>6</b>
<b>3 METODOLOGIA</b>	<b>7</b>
3.1 PROCESSAMENTO DOS DADOS	7
3.2 CLIENTE	9
3.3 BACK-END	10
<b>4 RESULTADOS</b>	<b>11</b>
<b>5 CONCLUSÃO</b>	<b>13</b>
<b>6 REFERÊNCIAS</b>	<b>14</b>

## 1 INTRODUÇÃO

O presente trabalho busca relatar o desenvolvimento do projeto.

### 1.1 TEMA

O desenvolvimento de um aplicativo para introduzir e ensinar o usuário quanto ao uso de smartphones utilizando inteligências artificiais de variados tipos e uma interface interativa.

### 1.2 JUSTIFICATIVA

Esse projeto procura solucionar os problemas que muitos idosos enfrentam atualmente relacionado ao uso dos smartphones. Por conta da falta de informação sobre esse tema, muitos idosos acabam desistindo de usar um smartphone por não entender seu funcionamento e não possuir expectativa de melhora no aprendizado.

O aplicativo pretende utilizar diversos algoritmos de inteligência artificial de detecção de objetos, classificação, reconhecimento de caracteres ópticos e processamento de linguagem natural para, respectivamente, localizar e classificar elementos de interface visual, classificar o tipo de tela (login, cadastro, chat, vídeo, dashboard, etc), o tipo de ícone e a ação que deve ser tomada, ler textos nos elementos da interface detectados e processar pesquisas por texto.

Uma das ferramentas únicas do Aprender é o assistente flutuante, que acompanha o usuário durante todo o uso do celular, ela está lá para quando necessário, uma vez que sentir uma dificuldade no uso de algum aplicativo, basta acionar o assistente que será direcionado para a tela com o tutorial para o aplicativo em específico.

Hoje, não há nenhum app que faça o que o Aprender se propõe a fazer, pois o que temos hoje são alguns apps que mudam a interface do

celular para facilitar o uso para idosos ou pessoas com problemas de visão, e também um app chamado iDosos que ensina por meio de tutoriais algumas funcionalidades básicas do smartphone, como ligações, editar contatos na agenda e mandar mensagens de texto.

### 1.3 OBJETIVOS

O objetivo dessa pesquisa é desenvolver um aplicativo baseado em inteligência artificial para celulares Android que ensine a utilização de diversas funcionalidades do celular e de aplicativos de terceiros a usuários não familiarizados com as novas tecnologias.

#### 1.3.1 Objetivos específicos

- Treinar um modelo de inteligência artificial para detectar elementos de interface visual em fotos da tela do usuário.
- Treinar um modelo de inteligência artificial para classificar o tipo de tela a partir dos elementos visuais detectados.
- Utilizar reconhecimento de caracteres ópticos para detectar as etiquetas (labels) dos elementos visuais detectados que possuem texto.
- Utilizar processamento de linguagem natural para processar e classificar pesquisas por texto.
- Fazer uma interface amigável e intuitiva para os usuários, de forma que não precisem se esforçar para entender suas funcionalidades;
- Fazer um aplicativo que pareça simples mas dê todo o suporte necessário ao usuário;
- Criar um sistema de caixa de texto, em que o usuário possa fazer uma pesquisa específica de uma funcionalidade que possui dificuldade;

## 2 REFERENCIAL TEÓRICO

A pesquisa foi direcionada para o público idoso e moradores de zonas rurais onde geralmente há menos informação sobre os smartphones.

Segundo a estimativa da FGVcia<sup>1</sup> (Centro de Tecnologia de Informação Aplicada da Escola de Administração de Empresas de São Paulo da Fundação Getúlio Vargas), realizada em junho de 2021 no Brasil, há 242 milhões de smartphones em uso no Brasil. E 97% dos idosos acessam a internet segundo a CNDL (Confederação Nacional de Dirigentes Lojistas) em 2021, sendo que 84% desses idosos acessam pelo smartphone.

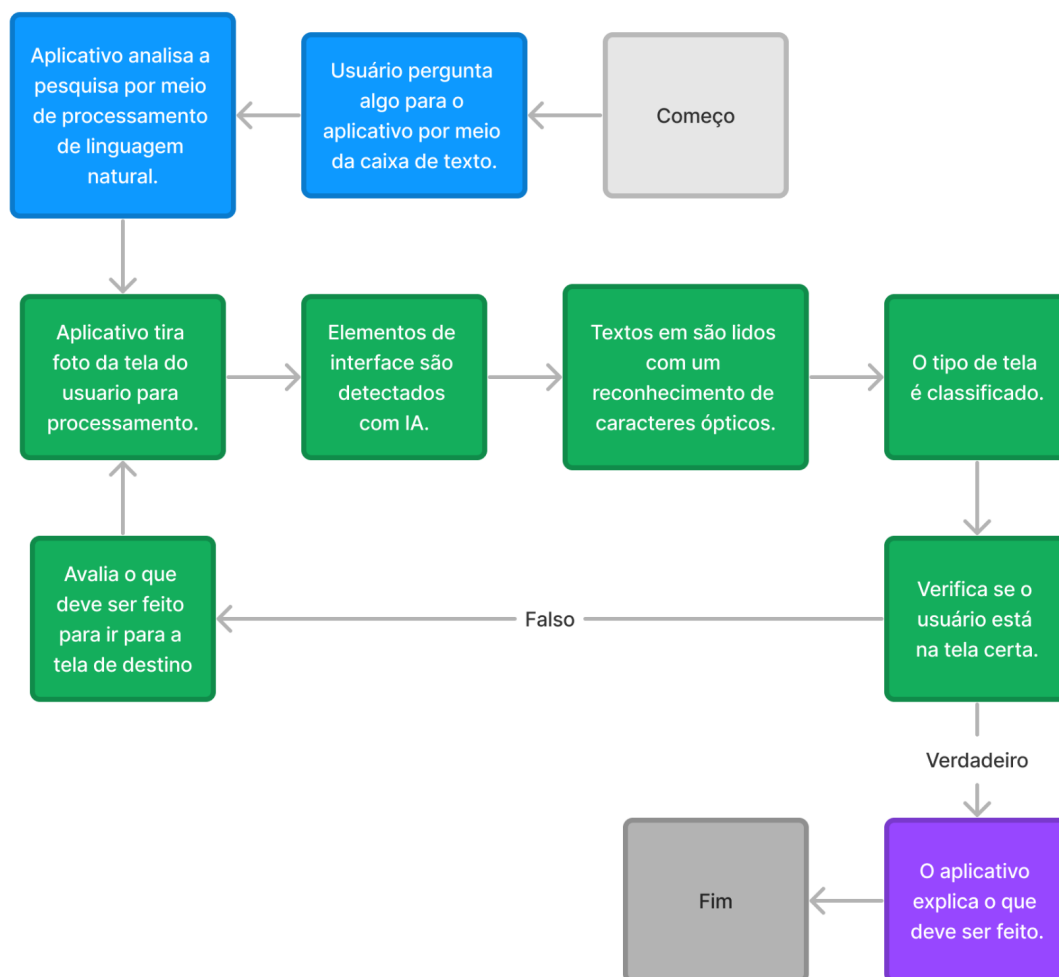
Dito isso, adequamos ao máximo nosso aplicativo para ser o mais simples e intuitivo possível, para que a experiência do usuário novo à tecnologia seja a melhor possível.

Além dos idosos, um dos fatores mais estimulou a ideia do projeto foi uma pesquisa da empresa PwC e do Instituto Locomotivas<sup>2</sup>, esse identificou que mais de 33,9 milhões de brasileiros não estão conectadas na internet, e como os celulares modernos andam de mãos dadas com a conexão com as redes, decidimos criar esse trabalho, que tem tudo para ser relevante no cenário nacional.

### 3 METODOLOGIA

Foi elaborado um diagrama de funcionamento do aplicativo para um melhor entendimento do algoritmo:

Figura 1 - Diagrama de funcionamento do aplicativo



Fonte: Figura do autor

#### 3.1 PROCESSAMENTO DOS DADOS

O foco do desenvolvimento do projeto foi o estágio de detecção de elementos de interface visual, visto que essa seria uma das partes fundamentais do funcionamento do aplicativo.



O desenvolvimento do algoritmo foi feito em máquinas com sistemas operacionais Windows 10, Windows 11, macOS e Ubuntu 22.04.1 LTS e na linguagem Python 3.10. A escolha da linguagem foi feita com base no ecossistema oferecido para o desenvolvimento de modelos de inteligência artificial.

Inicialmente o algoritmo de detecção seria feito com o UIED<sup>3</sup>, porém o algoritmo se provou ser impreciso, levando a necessidade de criar um modelo de inteligência artificial específico para o projeto.

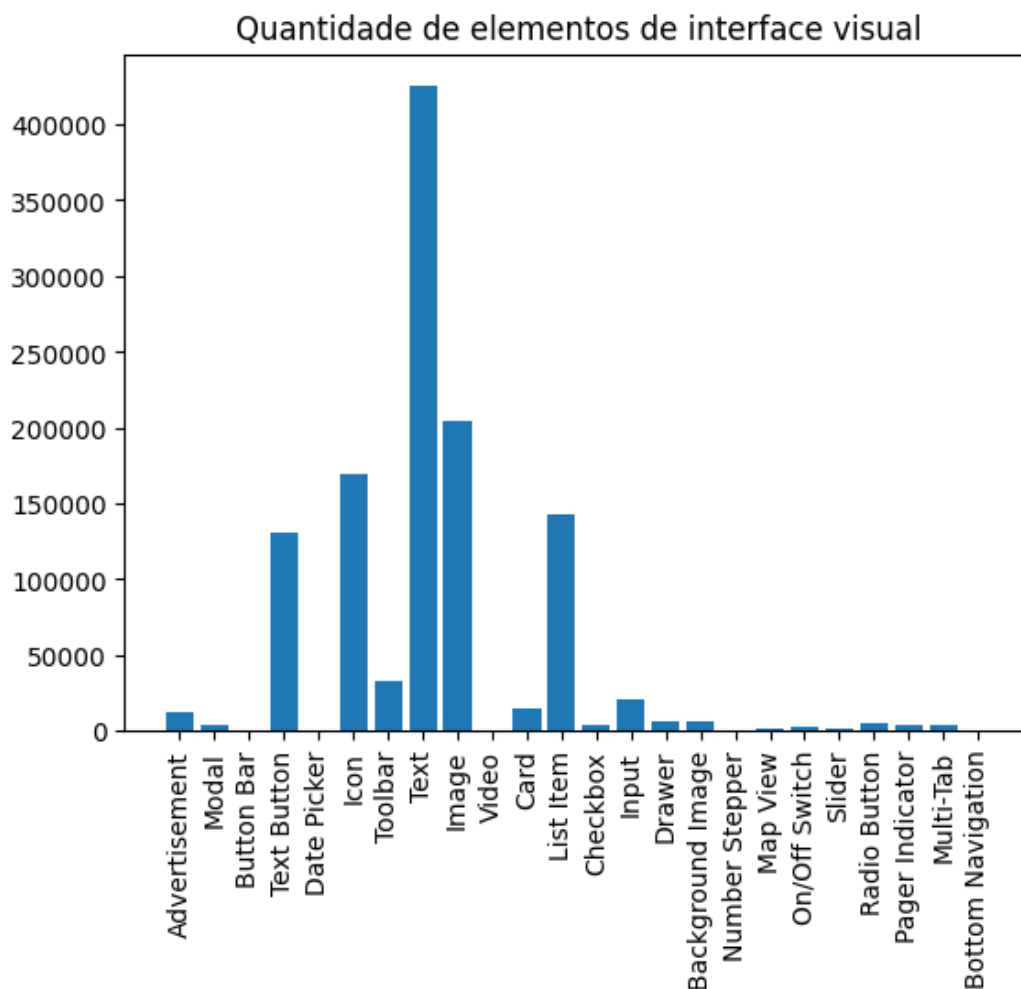
A ideia inicial para o desenvolvimento desse modelo foi utilizar uma biblioteca chamada OpenCV<sup>4</sup> para treinar um algoritmo de detecção de objetos. O OpenCV logo foi substituído por outra biblioteca chamada PyTorch<sup>5</sup>, isso porque o segundo mostrou-se muito mais fácil de utilizar e disponibiliza algoritmos de detecção de objetos por padrão.

Foi criada uma primeira rede neural de modelo Faster R-CNN<sup>6</sup> treinada com o dataset ReDraw<sup>7</sup>. Esse primeiro modelo se provou inadequado, isso porque todas as imagens do dataset são apenas do elemento e não inclui o resto da tela.

Isso resultou em um modelo que apenas analisava a tela como um todo e classificava a tela inteira, muitas vezes como uma imagem.

Uma segunda tentativa de criar a rede neural foi feita desta vez com o modelo YOLOv7<sup>8</sup> treinado com o dataset RICO<sup>9</sup>. Foi necessária a elaboração de um script de conversão do dataset para um formato que o algoritmo de treinamento do YOLOv7 suportasse. Esse script está disponível em um repositório público do Github<sup>10</sup>.

Figura 2 - Quantidade de elementos de interface visual no dataset RICO



Fonte: Figura do autor

Essa nova rede neural se mostrou ser muito mais precisa, sendo capaz de, por exemplo, detectar elementos de listas e textos e diferenciar algumas imagens de ícones.

### 3.2 CLIENTE

O desenvolvimento da aplicação *mobile* foi feito em máquinas MAC/IOS. A tecnologia utilizada foi a Android, na linguagem Java. A escolha da linguagem foi baseada na familiaridade com ela, visto que ela já foi estudada em outras oportunidades.

O foco dessa área do projeto foi o desenvolvimento da interface que o usuário interage.

Para a criação do assistente flutuante foram utilizadas várias bibliotecas, a mais importante sendo a *Service* que permitiu o assistente funcionar perfeitamente.

### 3.3 BACK-END

O desenvolvimento da API foi feita em ASP.NET em conjunto com banco de dados em MySQL, na API foi utilizado o Entity Framework Core, para usar os Models para as tabelas e Data Annotations para assegurar que os dados seriam passados na formatação correta.

## 4 RESULTADOS

Foi possível elaborar um algoritmo de funcionamento do projeto e a criação de um modelo de inteligência artificial para a detecção dos elementos de interface visual.

O algoritmo se mostrou capaz de detectar diversos componentes visuais e alguns componentes abstratos como listas. O algoritmo também distingue certos elementos que visualmente são parecidos (e.g. ícones e imagens).

Figura 3 - Detecção de elementos visuais



Fonte: Figura do autor

Foi observado que ainda é possível otimizar o treinamento do modelo de diversas maneiras para a obtenção de resultados mais precisos.

Foi também possível implementar uma interface simples que entrega ao usuário um rápido tutorial amigável aos olhos do aplicativo desejado.

Na questão do desenvolvimento *mobile*, o principal desafio foi superado: a implementação de um *background service*, um serviço que mantém a funcionalidade do aplicativo, esteja ele aberto, minimizado ou completamente fechado. Além disso, colocamos um sistema de popup para que o usuário possa ver o tutorial com o foco e tamanho necessários.

A API foi desenvolvida e concluída em ASP.NET, com um banco de dados em MySQL onde é possível utilizar quase todos os métodos de requisição http, e os utilizando para fazer um cadastro e login do usuário no app. Um Model que não foi conectado com o app mobile é os guias linkados no banco de dados para serem demonstrados na tela quando requisitados pelo usuário.

## 5 CONCLUSÃO

O projeto acabou sendo complexo demais para o período de tempo disponível para a realização da pesquisa, mas alguns objetivos foram alcançados.

Os próximos passos no desenvolvimento do algoritmo de processamento dos dados seriam aperfeiçoar o algoritmo de detecção de objetos e elaborar uma maneira de classificar os tipos de tela a partir dos elementos de interface visual detectados com o YOLOv7 para dar continuidade ao processamento dos dados.

No cliente, as próximas ações a serem tomadas seriam: otimizar ainda mais o código, trabalhar em deixar a interface mais perfeita possível e incrementar a função de caixa de texto, permitindo o usuário a fazer uma busca mais específica sobre a funcionalidade de um app.

É necessário um maior investimento de tempo para a continuidade do projeto e a realização de mais pesquisa sobre o assunto.

## 6 REFERÊNCIAS

- 1 FGV: **Retrospectiva 2021: Brasil tem dois dispositivos digitais por habitante, revela pesquisa da FGV.** Disponível em: <https://portal.fgv.br/noticias/retrospectiva-2021-brasil-tem-dois-dispositivos-digitais-habitante-revela-pesquisa-fgv>. Acesso em: 23 de nov. 2022.
- 2 CNDL. **97% dos idosos acessam a internet, aponta pesquisa da CNDL/SPC Brasil.** Disponível em: <https://cndl.org.br/varejosa/numero-de-idosos-que-acessam-a-internet-cresce-de-68-para-97-aponta-pesquisa-cndl-spc-brasil/#:~:text=O%20principal%20meio%20de%20acesso.notebook%20e%2036%25%20computador%20desktop>. Acesso em: 24 de nov. 2022.
- 3 MulongXie. **MulongXie/UIED: An accurate GUI element detection approach based on old-fashioned CV algorithms [Upgraded on 5/July/2021].** Disponível em: <https://github.com/MulongXie/UIED>. Acesso em: 23 de nov. 2022.
- 4 OpenCV. **Home - OpenCV.** Disponível em: <https://opencv.org/>. Acesso em: 24 de nov. 2022.
- 5 PyTorch Foundation. **PyTorch.** Disponível em: <https://pytorch.org/>. Acesso em: 24 de nov. de 2022.
- 6 REN, S.; HE, K.; GIRSHICK, R.; SUN, J. **Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks.** Disponível em: <https://arxiv.org/pdf/1506.01497.pdf>. Acesso em: 24 de nov. 2022.
- 7 MORAN, KEVIN et al. **The ReDraw Dataset: A Set of Android Screenshots, GUI Metadata, and Labeled Images of GUI Components.** Disponível em: <https://zenodo.org/record/2530277#.Y39b0H3MKM8>. Acesso em: 24 de nov. 2022.
- 8 WongKinYiu. WongKinYiu/yolov7: Implementation of paper - YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. Disponível em: <https://github.com/WongKinYiu/yolov7>. Acesso em: 24 de nov. 2022.
- 9 RICO. **Rico: A Mobile App Dataset for Building Data-Driven Design Applications.** Disponível em: <https://interactionmining.org/rico>. Acesso em: 24 de nov. 2022.
- 10 SHOJI, V. **User Interface Element Detection for Phones using YOLO v7.** Disponível em:

<https://github.com/VShoji/mobile-user-interface-detection-model>. Acesso em: 24 de nov. 2022.

11 Documentação sobre Autenticação e Autorização no ASP.NET Web API. Disponível em: <https://learn.microsoft.com/pt-br/aspnet/web-api/overview/security/authentication-and-authorization-in-aspnet-web-api>

12 Documentação sobre o Entity Framework. Disponível em: <https://learn.microsoft.com/en-us/dotnet/framework/data/adonet/ef/overview>

13 Documentação sobre os tipos de retorno de ação do controlador na API Web do ASP.NET Core. Disponível em: <https://learn.microsoft.com/pt-br/aspnet/core/web-api/action-return-types?view=aspnetcore-3.1>

14 A study of smartphone usage and barriers among the elderly. Disponível em: <https://ieeexplore.ieee.org/abstract/document/7002686>

15 Documentação da classe Intent no Android. Disponível em: <https://developer.android.com/guide/components/intents-filters?hl=pt-br>

16 Documentação da biblioteca Service do Android. Disponível em: <https://developer.android.com/guide/components/services?hl=pt-br>