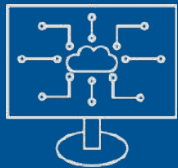




insidergic



Somos uma empresa de desenvolvimento de softwares, tecnologias e inteligência artificial.



Temos a missão de entregar soluções robustas e ao mesmo tempo de fácil acesso para nossos clientes



Nossos dois principais valores são:  
Ser orientado a dados;  
Apaixonados pelo problema e não pela solução.



Antônio Ramos

- Engenheiro de Produção
- Mestrando em Computação Inteligente
- Cientista de Dados
- Atuação em projeto envolvendo IA na área da saúde



Nathan Feitoza

- Engenheiro de Software
- Pós graduando em arquitetura de software
- Programador desde os 13 anos
- Trabalhou em startups de CRM, ERP, e-commerce. Além de uma consultoria global

# **Projeto Beira Rio**

# **IA para Reconhecimento de**

# **Imagens**



**CALÇADOS BEIRA RIO<sup>®</sup> S.A.**  
Produzindo valores

insidergic

O projeto trata-se do desenvolvimento de um algoritmo para reconhecimento de imagens utilizando técnicas de tratamento, padronização de imagens, modelos de machine learning e deep learning (aprendizado de máquina / aprendizado profundo).



# Etapas do Projeto

<b>Conhecimento e tratamento da banco de imagens</b>	<b>Limpeza do banco de imagens</b>	<b>Desenvolvimento do algoritmo de extração de características e clusterização</b>	<b>Implantação e testes</b>
--	--	--	---------------------------------

# Etapas do Projeto

**Conhecimento  
e tratamento  
da banco de  
imagens**

Limpeza do banco de  
imagens

Desenvolvimento do algoritmo  
de extração de características  
e clusterização

Implantação e  
testes

- **Acessos ao bucket no S3;**
- **Conhecimento dos arquivos disponibilizados;**
- **Preparação do ambiente de trabalho para processar arquivos;**
- **Montagem do algoritmo para pré-processamento. (Detecção de objetos)**



# Tarefas da Etapa de Tratamento

## Conhecimento dos arquivos disponibilizados

**Objetos (24)**

Os objetos são as entidades fundamentais armazenadas no Amazon S3. Você pode usar o inventário do Amazon S3 para obter uma lista de todos os objetos em seu bucket. Para outras pessoas acessarem seus objetos, você precisará conceder permissões explicitamente a eles. Saiba mais

[Copiar URI do S3](#) [Copiar URL](#) [Fazer download](#) [Abrir](#) [Excluir](#) [Ações](#) [Criar pasta](#)

[Carregar](#)

Localizar objetos por prefixo

	Nome	Tipo	Última modificação	Tamanho	Classe de armazenamento
<input type="checkbox"/>	_BOTAS 2023/	Pasta	-	-	-
<input type="checkbox"/>	_INVERNO 2023/	Pasta	-	-	-
<input type="checkbox"/>	europa/	Pasta	-	-	-
<input type="checkbox"/>	Fotos Lojas Bogotá/	Pasta	-	-	-
<input type="checkbox"/>	Fotos Originais NY Miami 1/	Pasta	-	-	-
<input type="checkbox"/>	Fotos SP agosto 2022/	Pasta	-	-	-
<input type="checkbox"/>	FOTOS SP e Salvador BH/	Pasta	-	-	-
<input type="checkbox"/>	FOTOS SP JUNHO.22/	Pasta	-	-	-
<input type="checkbox"/>	ITALIA 05_03/	Pasta	-	-	-
<input type="checkbox"/>	ITALIA 24_02/	Pasta	-	-	-

## Algoritmo para pré-processamento (IA para detecção de objetos)

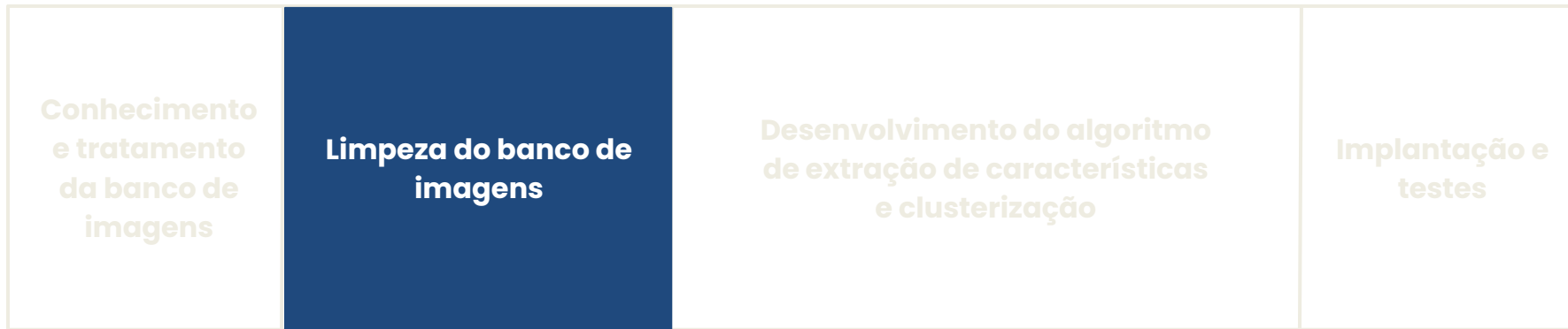
### Prepare Model Inference

[ ] (3 células ocultas)

### Pre Train

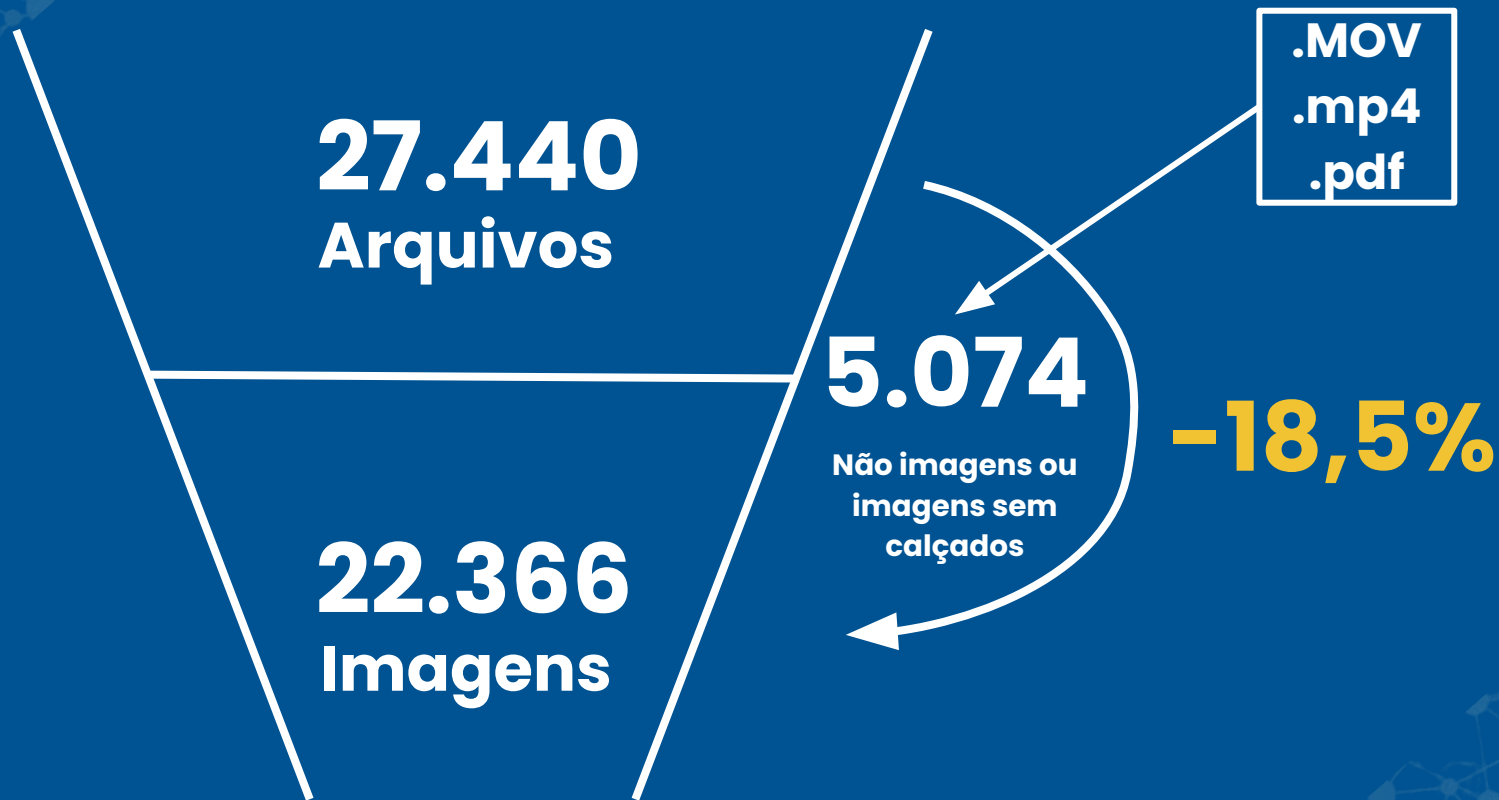
```
[ ] def trat_detect_objet (PATHs):  
  
    for idx, line in PATHs.iterrows():  
  
        # Define input and output tensors (i.e. data) for the object detection classifier  
        # Input tensor is the image  
        image_tensor = detection_graph.get_tensor_by_name('image_tensor:0')  
  
        # Output tensors are the detection boxes, scores, and classes  
        # Each box represents a part of the image where a particular object was detected  
        detection_boxes = detection_graph.get_tensor_by_name('detection_boxes:0')  
  
        # Each score represents level of confidence for each of the objects.  
        # The score is shown on the result image, together with the class label.  
        detection_scores = detection_graph.get_tensor_by_name('detection_scores:0')  
        detection_classes = detection_graph.get_tensor_by_name('detection_classes:0')  
  
        # Number of objects detected  
        num_detections = detection_graph.get_tensor_by_name('num_detections:0')  
  
        # Load image using OpenCV and  
        # expand image dimensions to have shape: [1, None, None, 3]  
        # i.e. a single-column array, where each item in the column has the pixel BGR value  
  
    try:  
        tratamento_detect = pd.read_csv(f'/content/drive/MyDrive/Notebooks/Bairro Rio/tratamentos/{line.folder_name}.csv')  
        if len(tratamento_detect) < 0:  
            tratamento_detect = pd.DataFrame(None, columns=['nome', 'itens_detectados', 'detect_seguros', 'excluido'])  
    except:  
        tratamento_detect = pd.DataFrame(None, columns=['nome', 'itens_detectados', 'detect_seguros', 'excluido'])
```

# Etapas do Projeto



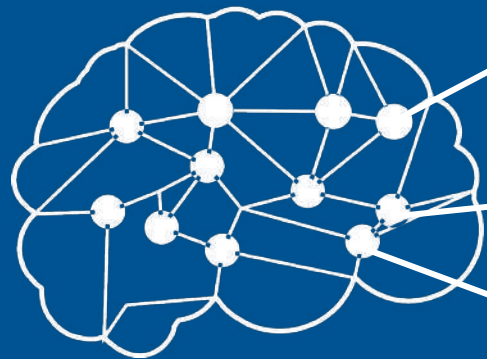
- **Análise de tipos de arquivos;**
- **Pré-processamento através da detecção de calçados;**
- **Análise de qualidade das detecções;**
- **Padronização dos tamanhos das imagens.**

# Análise de Tipos de Arquivos



# Detecção de Calçados

mobilenetv2



**22.366** Imagens com  
calçados detectados

**Média de 3 calçados  
detectados**

**Média de 1 calçado  
detectado com mais  
que 50% de confiança**

# Detecção de Calçados

fotos\_emillyIMG\_2503.JPG



20220603\_093539.jpg



# Análise de Qualidade das Detecções

**22.366 Imagens**

**9.993 Imagens**

com qualidade de detecção  
superior a 50% de confiança

# Análise de Qualidade das Detecções

Imagem com detecção superior a 50% de confiança



Imagem com detecção inferior a 50% de confiança





# Padronização dos Tamanhos das Imagens



Tamanho: 4032 x 3024 x 3

Padronizando tamanho em  
500 x 500 x 3





# Etapas do Projeto

Conhecimento  
e tratamento  
da banco de  
imagens

Limpeza do banco de  
imagens

**Desenvolvimento do algoritmo  
de extração de características  
e clusterização**

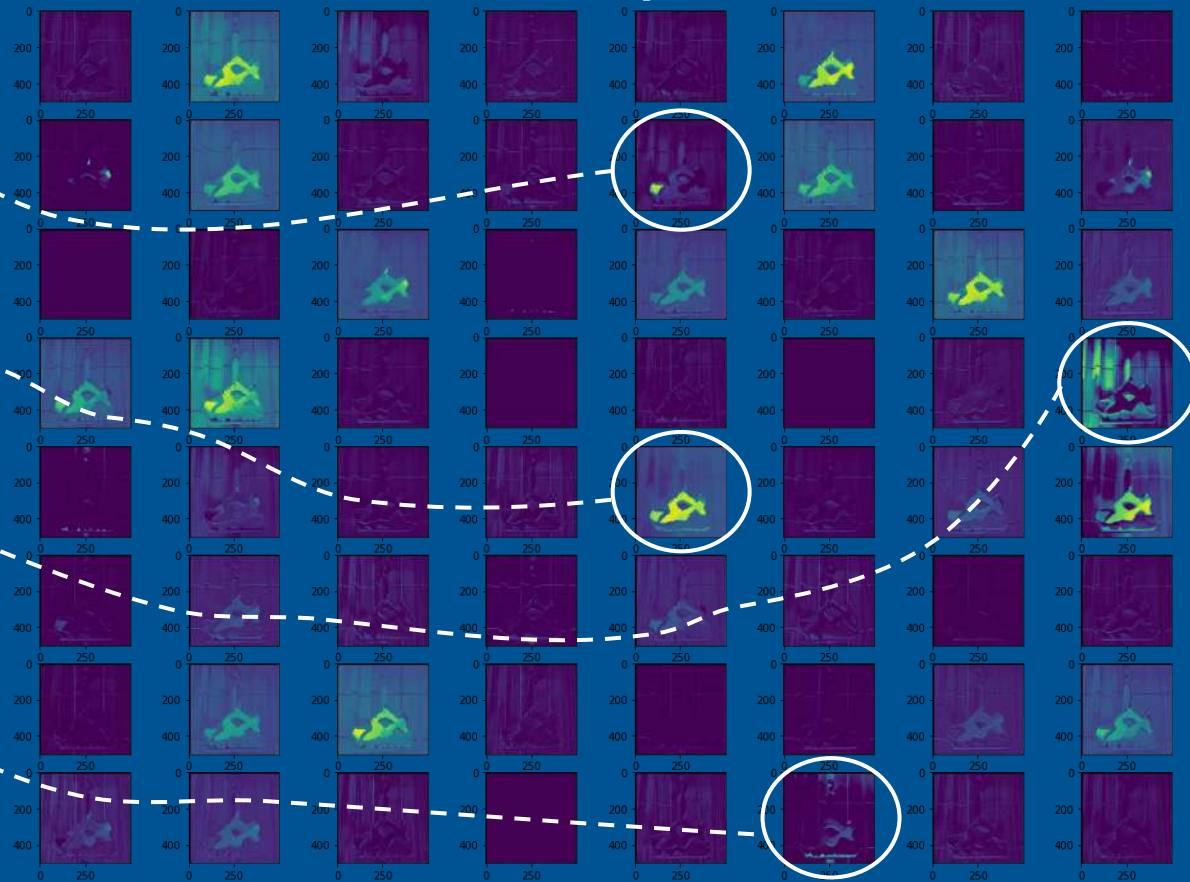
Implantação e  
testes

- **Preparação da rede neural;**
- **Extração dos mapas de características das imagens (Feature Maps);**
- **Implementação da estrutura de vizinhança (Agrupamento);**
- **Refinamento das estruturas de vizinhança.**

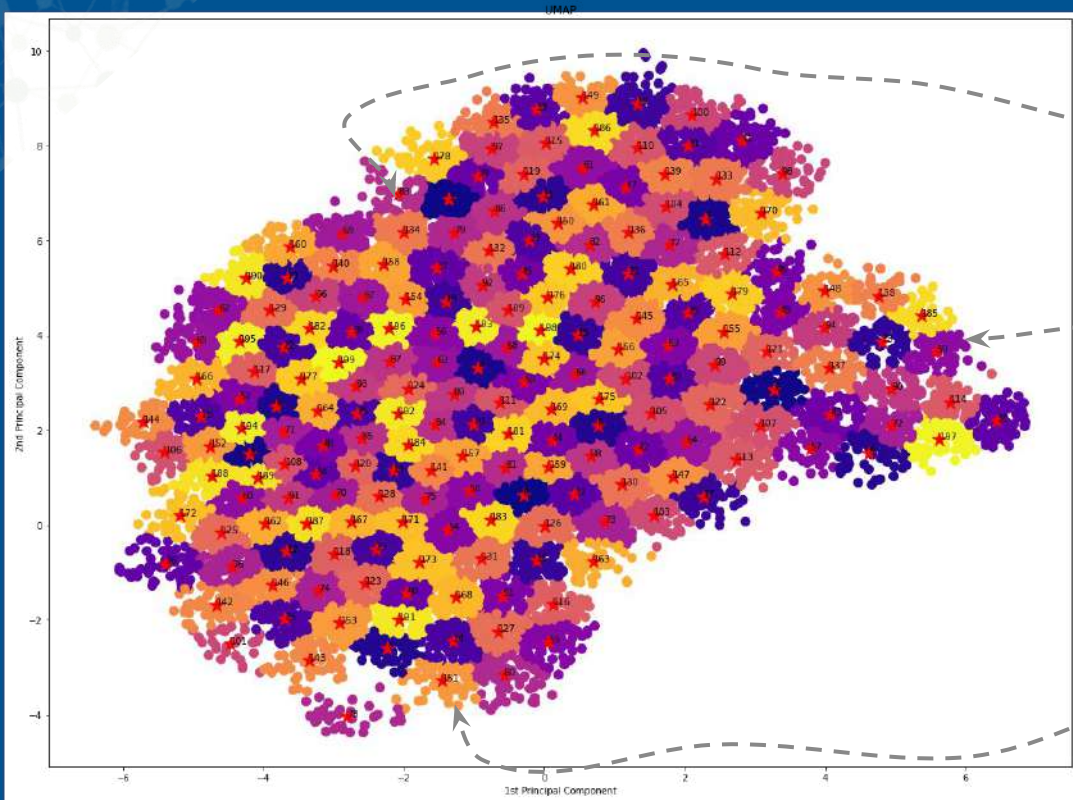
# Feature Maps



## Feature Maps

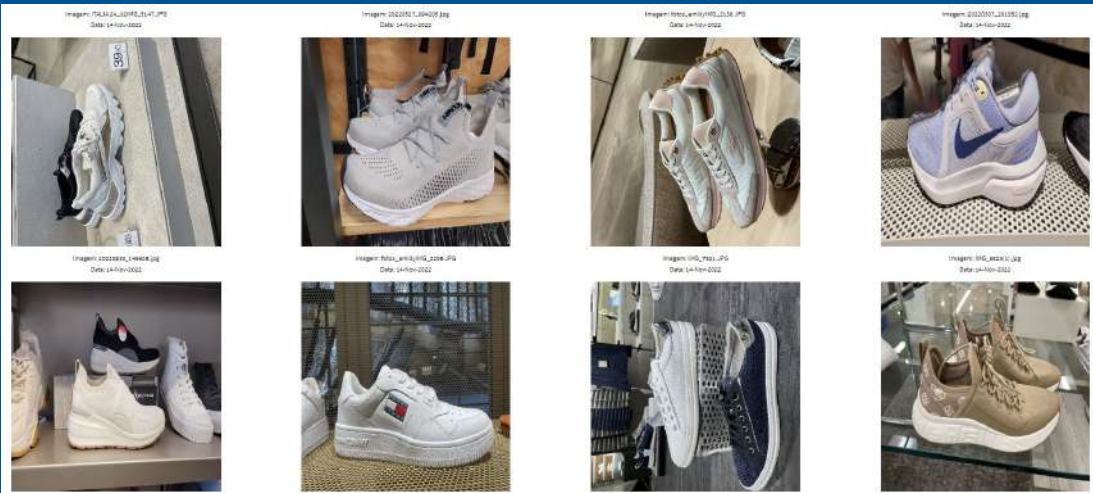


# Estrutura de Vizinhaça



# Estrutura de Vizinhança

insidergic





# Estrutura de Vizinhança



Imagem: IFG\_8322.JPG  
Data: 24-Nov-2022



Imagem: LOCHRES\_28\_048IFG\_8322.JPG  
Data: 24-Nov-2022



Imagem: IFG\_2088.JPG  
Data: 24-Nov-2022



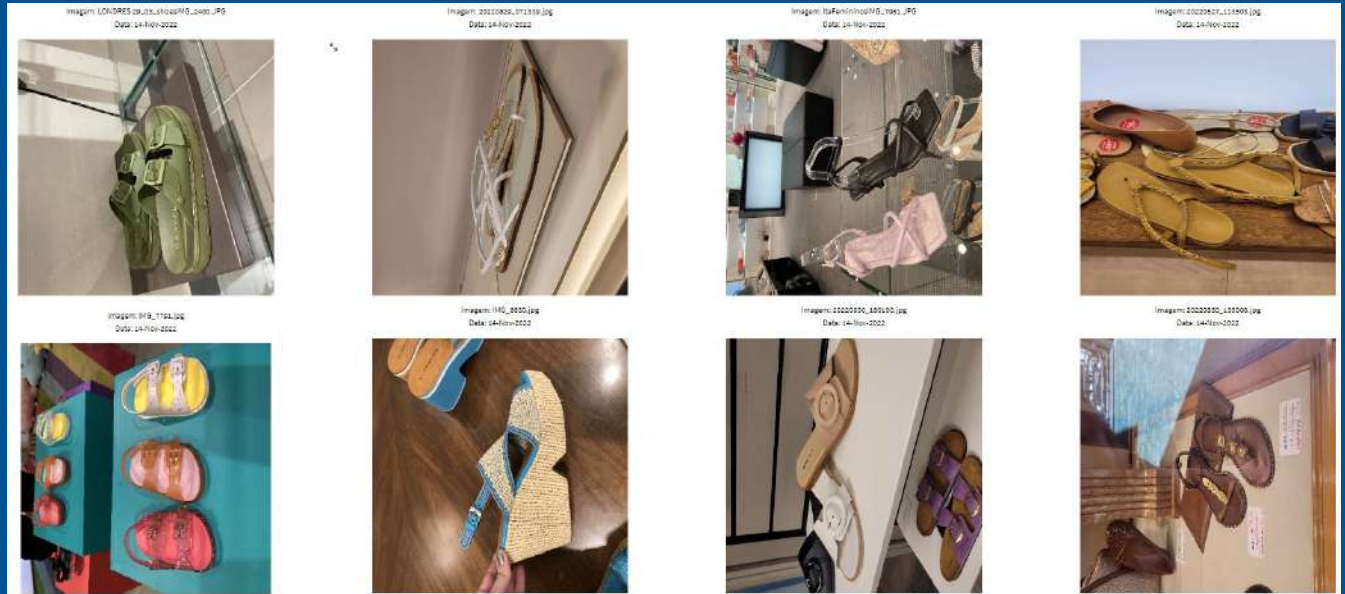
Imagem: LOCHRES\_22\_76mIFG\_2088.JPG  
Data: 24-Nov-2022



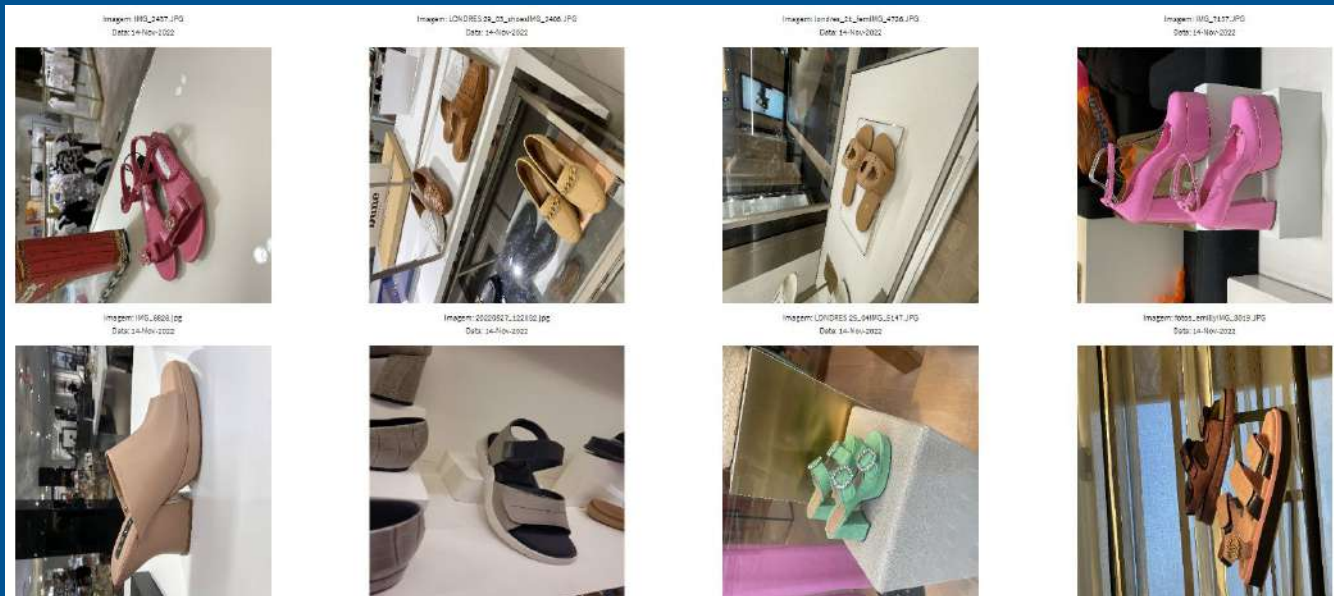
Imagem: IFG\_1451.JPG  
Data: 24-Nov-2022



Imagem: LOCHRES\_22\_76mIFG\_1451.JPG  
Data: 24-Nov-2022



# Estrutura de Vizinhança



# Etapas do Projeto



- **Criação da API para consumo da IA;**
- **Criação do front-end da solução para consumo;**
- **Teste da solução.**



# Front-end

insidergic

×

Entrar

Sobre o Projeto

Recomendações

Obter Imagens Semelhantes

insidergic

Share ☆ ≡

## IA para Obtenção de Calçados Semelhantes

Usuário

Senha

Entrar

insidergic