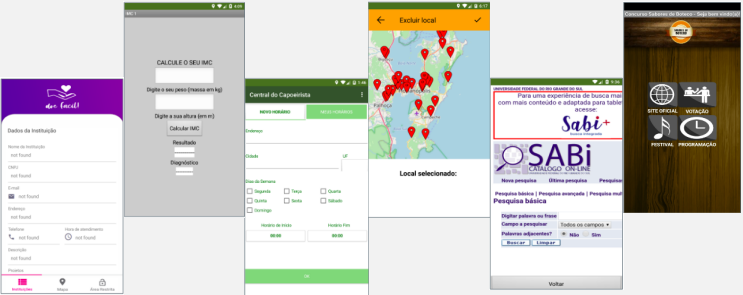




# VISÃO COMPUTACIONAL

## Alunos: Angélica Siqueira de Souza Marcelo Fernando Rauber

<https://github.com/magcid/visaoComputacional>  
Data: 23/03/2022



### OBJETIVO:

Detectar automaticamente os componentes de UI para identificar a posição, tamanho e o tipo do componente de UI nos screenshots de apps.

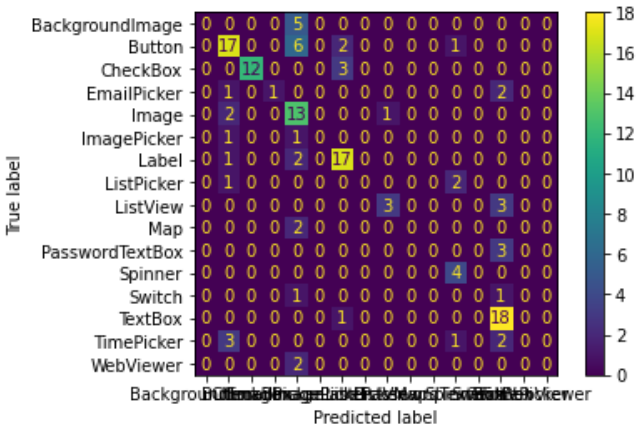
Button CheckBox DatePicker Image BackgroundImage Label ListPicker	ListView PasswordTextBox Slider Spinner Switch TextBox	Notifier TimePicker WebView Map ImagePicker VideoPlayer
---	---	--

### METODOLOGIA:

Para o Dataset foram geradas e utilizadas imagens de capturas de tela. Estas posteriormente foram rotuladas com o objetivo identificar onde e quais os componentes estão presentes nas telas para serem utilizados no treinamento.

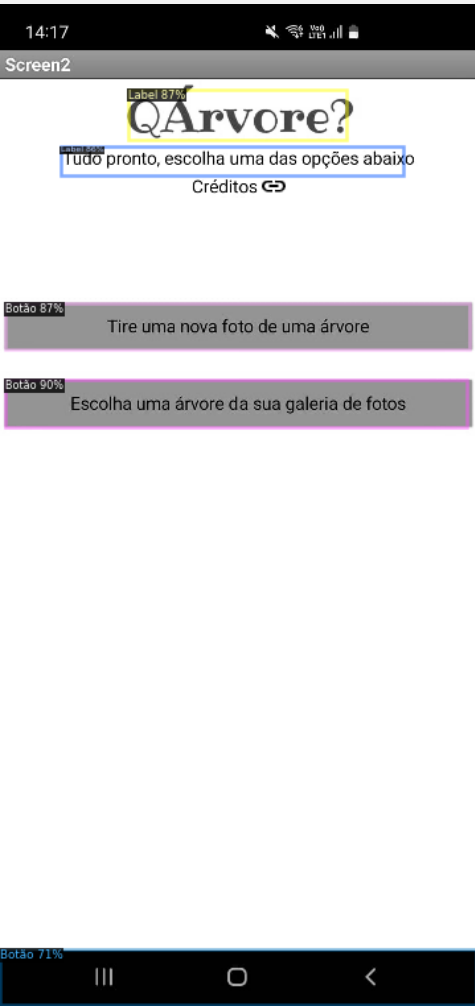
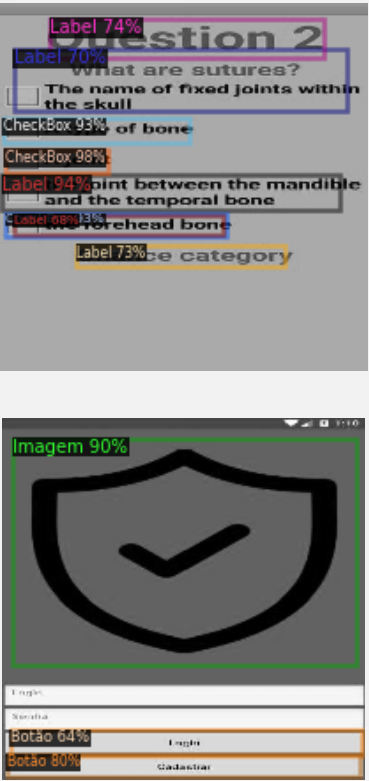
### Visão Computacional Clássica

Buscando uma abordagem ligada a Visão Computacional Clássica, no primeiro experimento gerou-se o Histograma de Gradientes Orientados (HoG) dos componentes individualmente recortando as imagens conforme marcação prévia com LabelMe. Com base no HoG, foi treinada uma rede neural usando Suport Vector Machine (SVM) usando a implementação do Sklearn.



### Deep Learning

Para a utilização de CNN, optamos por utilizar o Detectron2. Após a rotulagem das interfaces identificando os componentes de interesse com a ferramenta LabelME, esse Dataset foi importado para a ferramenta Roboflow pois facilita a manipulação e exportação em diferentes formatos de dados. Escolhemos o formato COCO por ser facilmente importado para o Detectron2. O Dataset de 364 imagens (3065 rótulos) está dividido em 80% para treino e 20% validação. Para o treinamento da rede do Detectron2 utilizamos o modelo Faster R-CNN. Após 600 épocas de treinamento, alcançamos 93% de acurácia. Passando para a análise do Dataset de validação obtivemos precisão média de 30,37% (seguindo o padrão CocoEvaluator).



### Conclusão

- A utilização de Faster r-CNN levou a resultados satisfatórios com Deep Learning, sendo que o conjunto de treinamento atingiu 93% de acurácia.
- O conjunto de validação de dados mostrou uma precisão média de 30,37% seguindo o padrão CocoEvaluator;
- Nos experimentos a utilização da abordagem clássica mostrou-se viável para classificar os componentes individuais, mas é difícil identificação dos componentes (potencial trabalho futuro);