

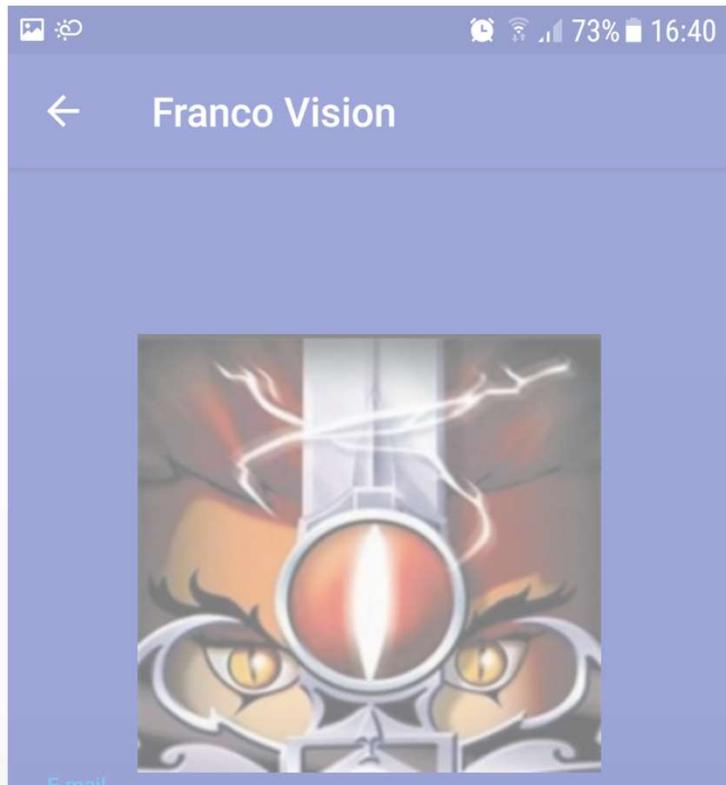
Aplicativo para reconhecimento de objetos em imagens destinado a pessoas portadoras de deficiência visual

Aluno(a): Jean Carlos Franco

Orientador: Roberto Heinzle

Roteiro

- Introdução
- Objetivos
- Fundamentação Teórica
- Trabalhos Correlatos
- Requisitos e especificação
- Implementação
- Operacionalidade
- Resultados e discussões
- Conclusões e sugestões

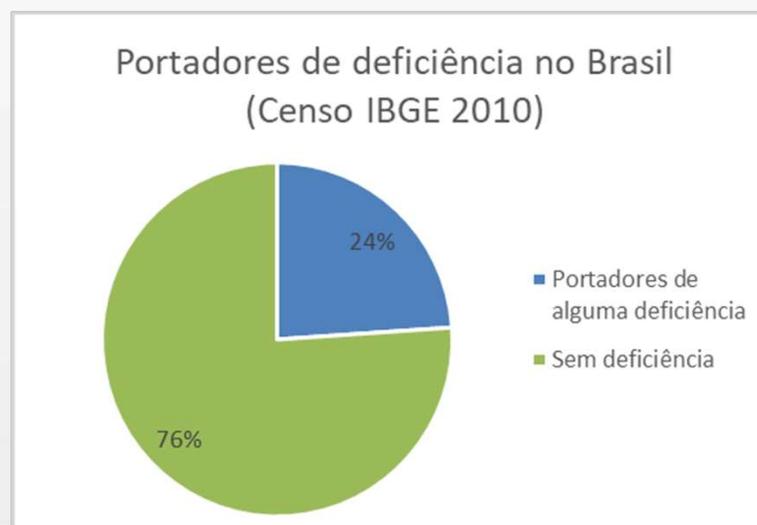


Introdução e objetivos

ACESSAR OU REGISTRAR

Introdução

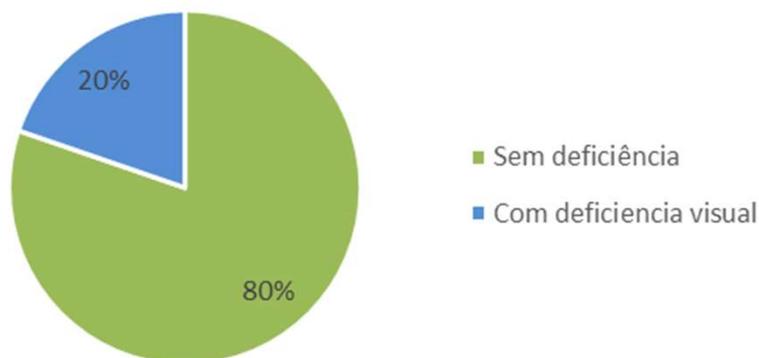
- Portador de deficiência
 - Pessoa que tem algum impedimento de longo prazo de natureza física, mental, intelectual ou sensorial
- Censo IBGE 2010
 - Existem cerca de 45 milhões de pessoas portadoras de alguma deficiência no Brasil



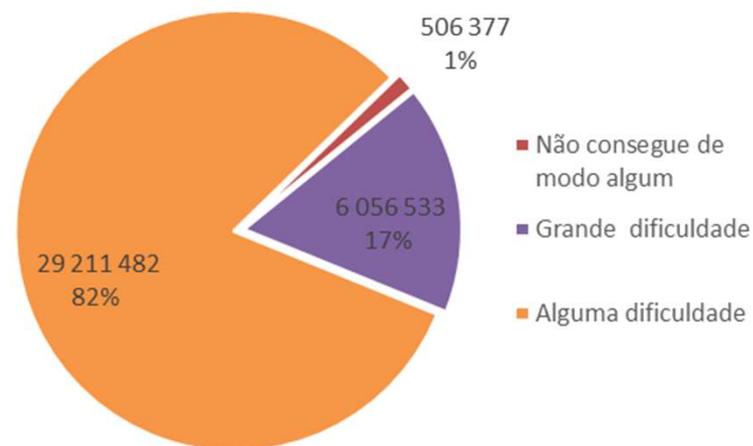
Introdução

- Existem mais de 35 milhões de pessoas com algum nível de deficiência visual, destas cerca de 500 mil não conseguem enxergar nada

Deficientes visuais no Brasil
(Censo IBGE 2010)



Níveis da deficiência visual



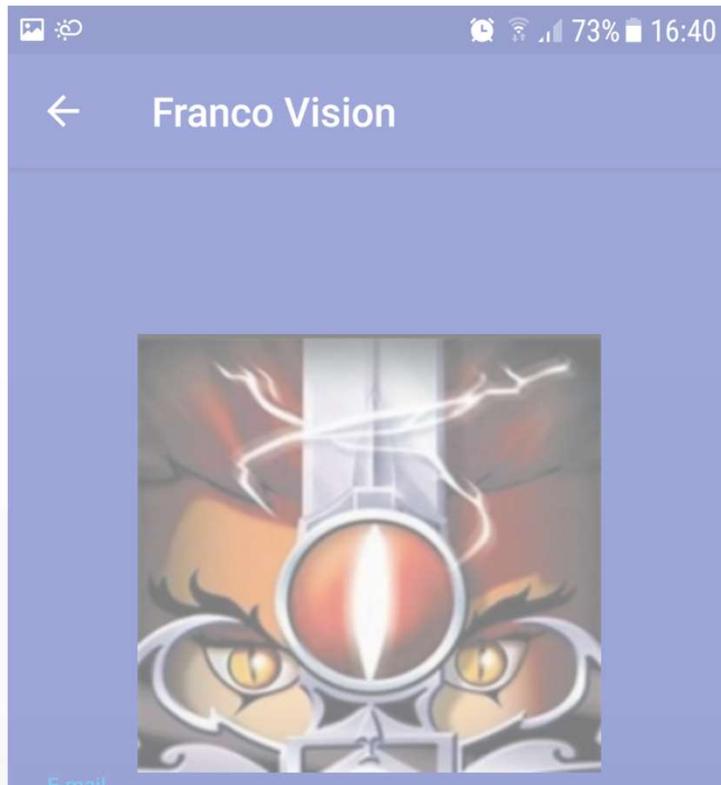
Introdução

- Tecnologia assistiva
 - Próteses
 - Cadeiras de roda
 - Veículos adaptados
 - Softwares como o Virtual Vision, que permitem pessoas com deficiência a utilizar computadores com o sistema operacional Windows
 - Aplicativos para transcrição de áudio em texto

Objetivos

Desenvolver um aplicativo móvel para a plataforma Android que permita realizar o reconhecimento de objetos em imagens

- a) disponibilizar uma interface que permita a captura, através da câmera do celular, da imagem a ser analisada;
- b) oferecer um mecanismo para reconhecer os objetos presentes na imagem e apresentar o seu conteúdo ao usuário;
- c) fornecer uma interface que permita a utilização do aplicativo por deficientes visuais através de gestos e com menus descritos por voz.



Fundamentação Teórica

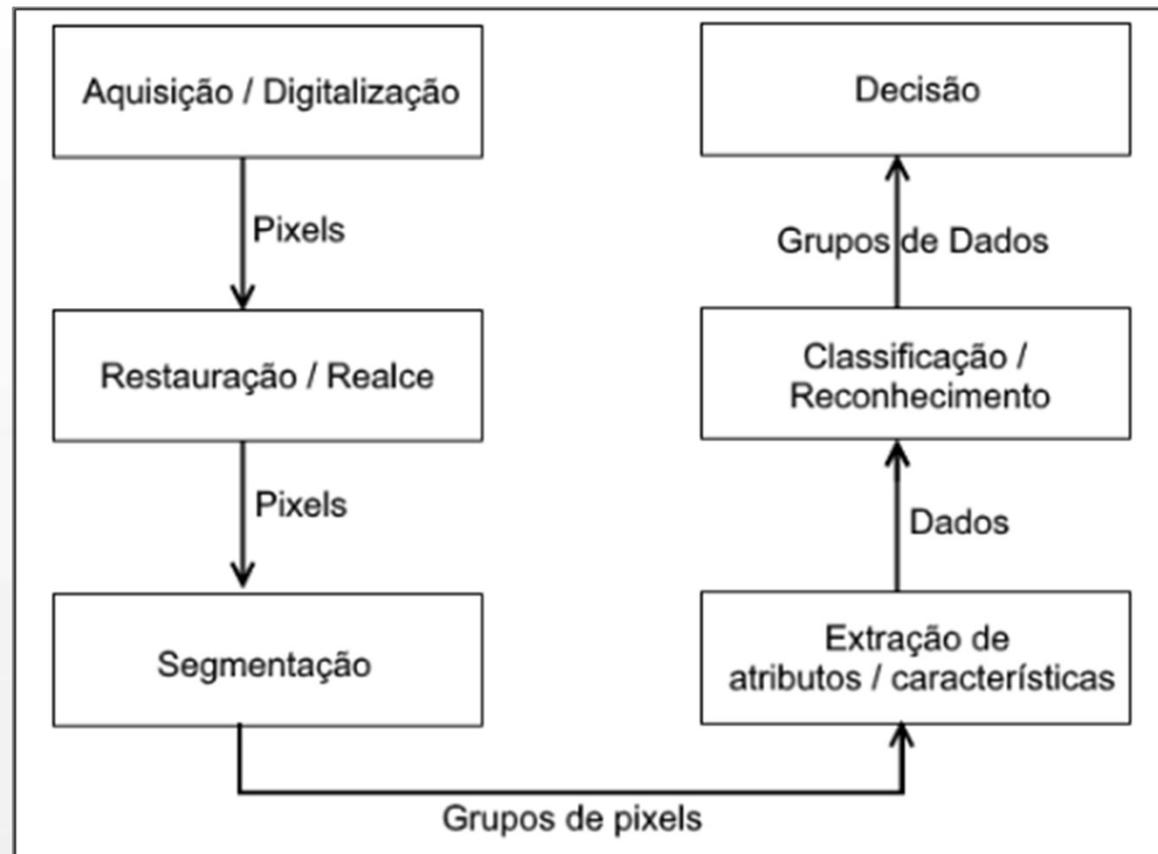
ACESSAR OU REGISTRAR

Computação gráfica e o reconhecimento de imagens

- Áreas da computação gráfica
 - processamento de imagens: manipulação de imagens já existentes
 - síntese de imagens: geração de imagens a partir de dados
 - análise de imagens: mapeamento de parâmetros descritivos (em geral numéricos) que identifiquem informações importantes da imagem
- Visão computacional
 - Divisão da área de análise de imagens que utiliza métodos e técnicas para que sistemas computacionais sejam capazes de interpretar imagens

Computação gráfica e o reconhecimento de imagens

Etapas de um sistema de VC genérico



Síntese de voz

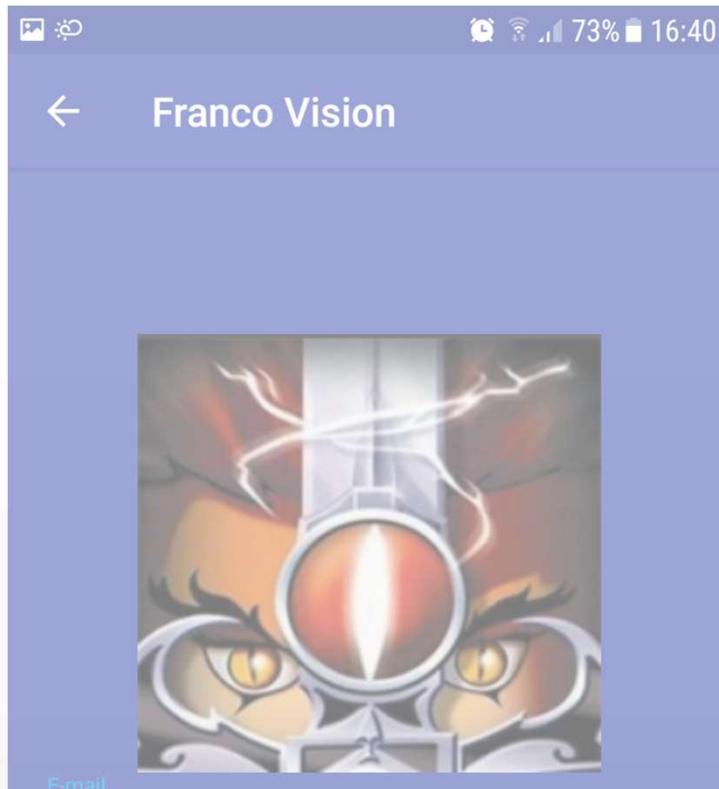
- A fala é o meio de comunicação mais poderoso que o ser humano possui
- A síntese de voz permite que a interação com computadores seja feita de uma maneira mais natural e intuitiva
- Tecnologia: Text-To-Speech (TTS): conversão de textos em síntese de voz

API Cloud Vision

- Utilização de Machine learning (ML)
- Plataforma Google Cloud
 - máquinas virtuais
 - ferramentas inteligentes de IOT (internet of things – internet das coisas)
 - serviços de machine learning
 - armazenamento na nuvem
 - visão computacional

API Cloud Vision do Google

- Cloud Vision
 - análise avançada de imagens
 - oferece modelos pré-treinados por meio de uma API
 - capacidade de criar modelos personalizados
 - classifica as imagens em milhares de categorias
 - detecta objetos e rostos individuais
 - extração de palavras contidas nas imagens



Trabalhos Correlatos

ACESSAR OU REGISTRAR

Munchkin Kramer (2017)

Reconhecimento dos textos em cartas de jogo para deficientes visuais

- Visão computacional
- Reconhecimento de texto
- Aplicativo para Android/IOS
- Tecnologia assistiva
- Sintetização de voz



Cogncook Heinzle (2017)

Reconhecimento de alimentos e sugestão de receitas

- Visão computacional
- Reconhecimento objetos
- Não é tecnologia assistiva
- Aplicativo para Android/IOS



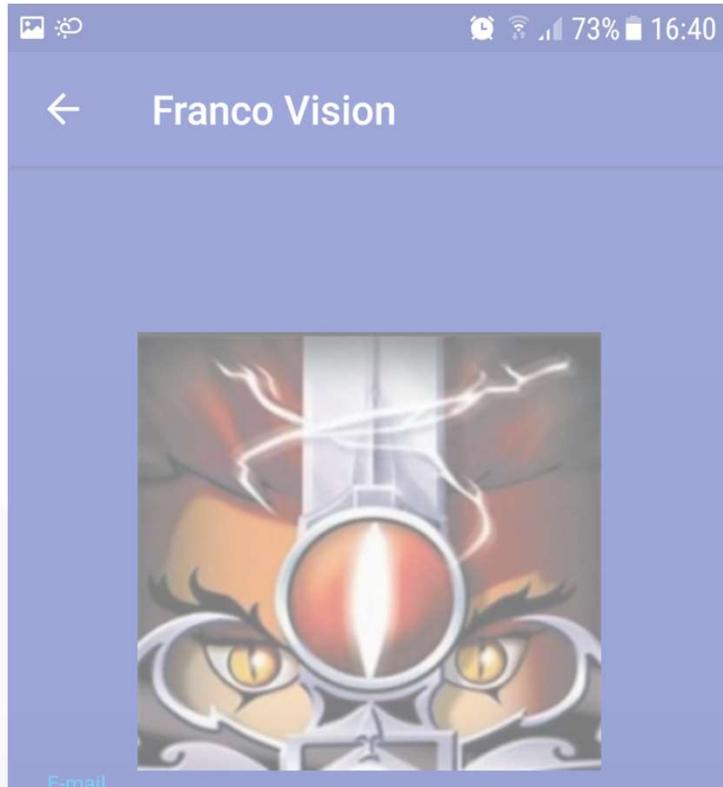
E-Motiv

Seefeld (2016)

Protótipo para o reconhecimento de expressões faciais universais

- Visão computacional
- Reconhecimento de expressões faciais
- Não é uma tecnologia assistiva
- Aplicação para plataforma Windows





Requisitos

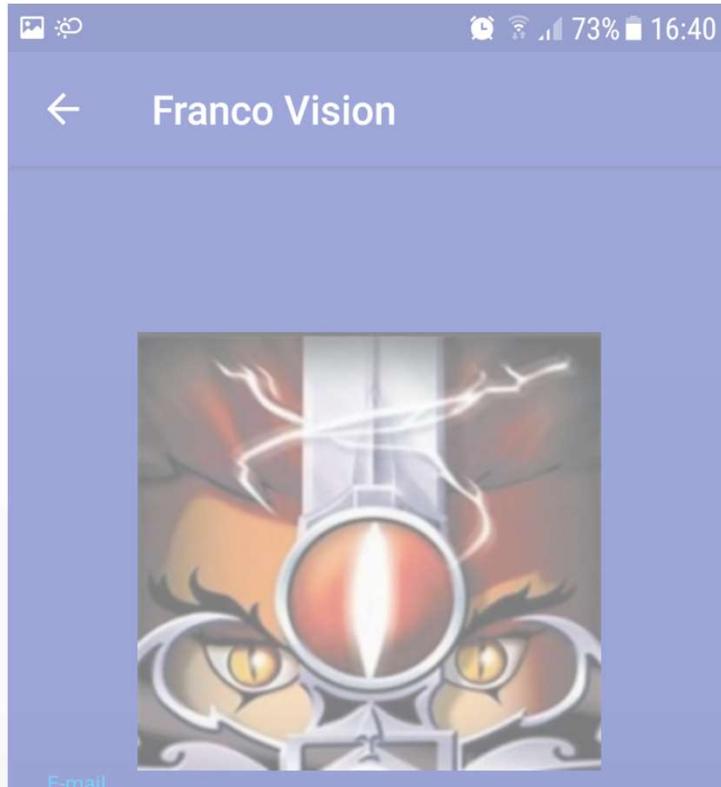
ACESSAR OU REGISTRAR

Requisitos funcionais (RF)

- Permitir o cadastro de um novo usuário através de um e-mail e senha
- Permitir capturar uma imagem do ambiente
- Permitir selecionar uma imagem salva no dispositivo
- Transmitir e processar a imagem selecionada utilizando a API Cloud Vision
- Exibir o resultado da análise da imagem ao usuário
- Converter para comandos de voz o nome dos objetos identificados na análise da imagem
- Descrever através de comandos de voz as instruções para que o usuário possa utilizar o aplicativo

Requisitos não funcionais (RNF)

- Permitir o acionamento da câmera através de gestos na tela, utilizando a opção de duplo clique
- Ser implementado para a plataforma Android utilizando a linguagem de programação Java
- Utilizar a API Cloud Vision do Google para o reconhecimento de objetos na imagem
- Utilizar a plataforma Firebase do Google para realizar a autenticação e cadastro do usuário



Especificação

ACESSAR OU REGISTRAR

Diagrama de caso de uso

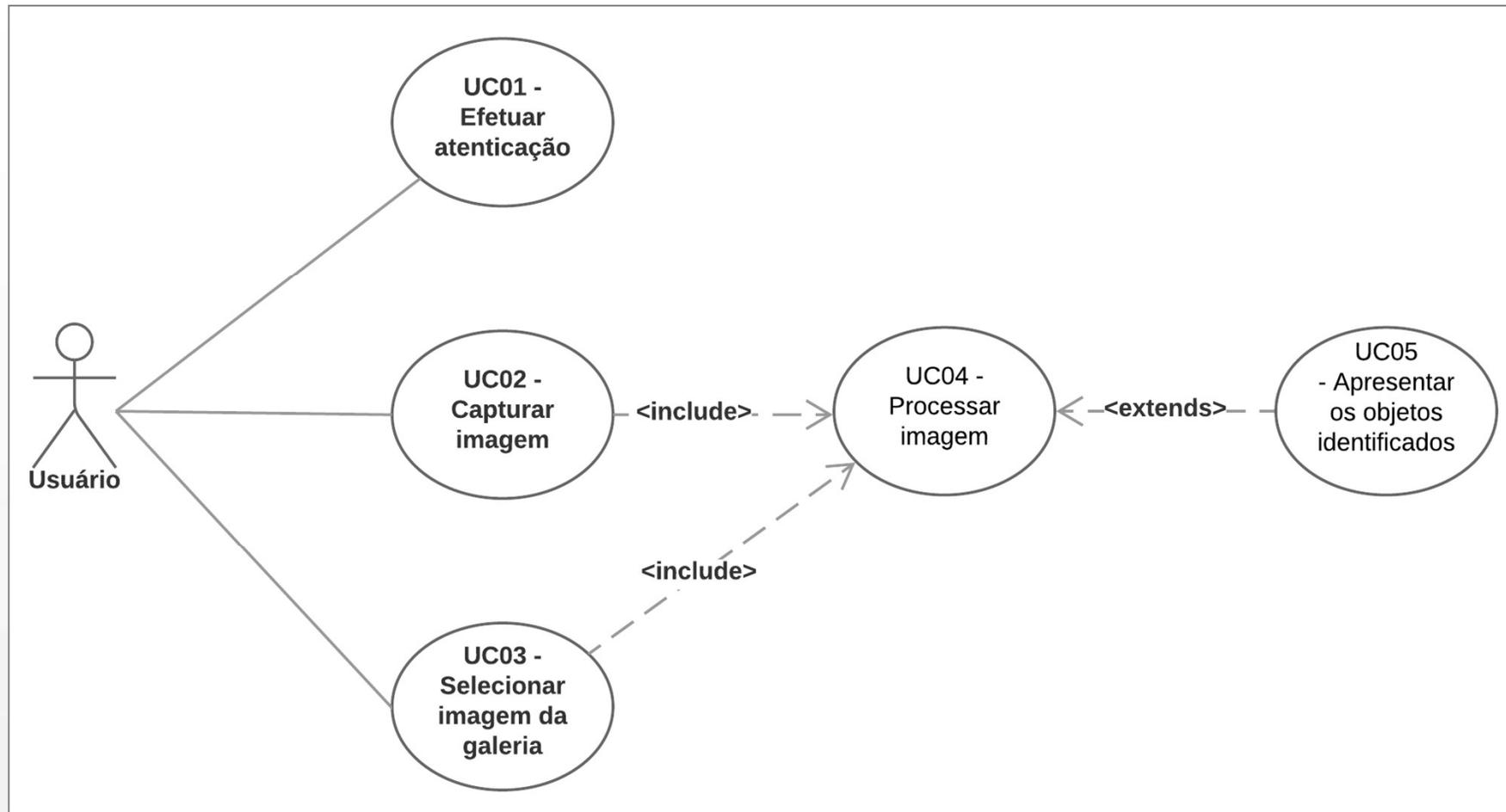
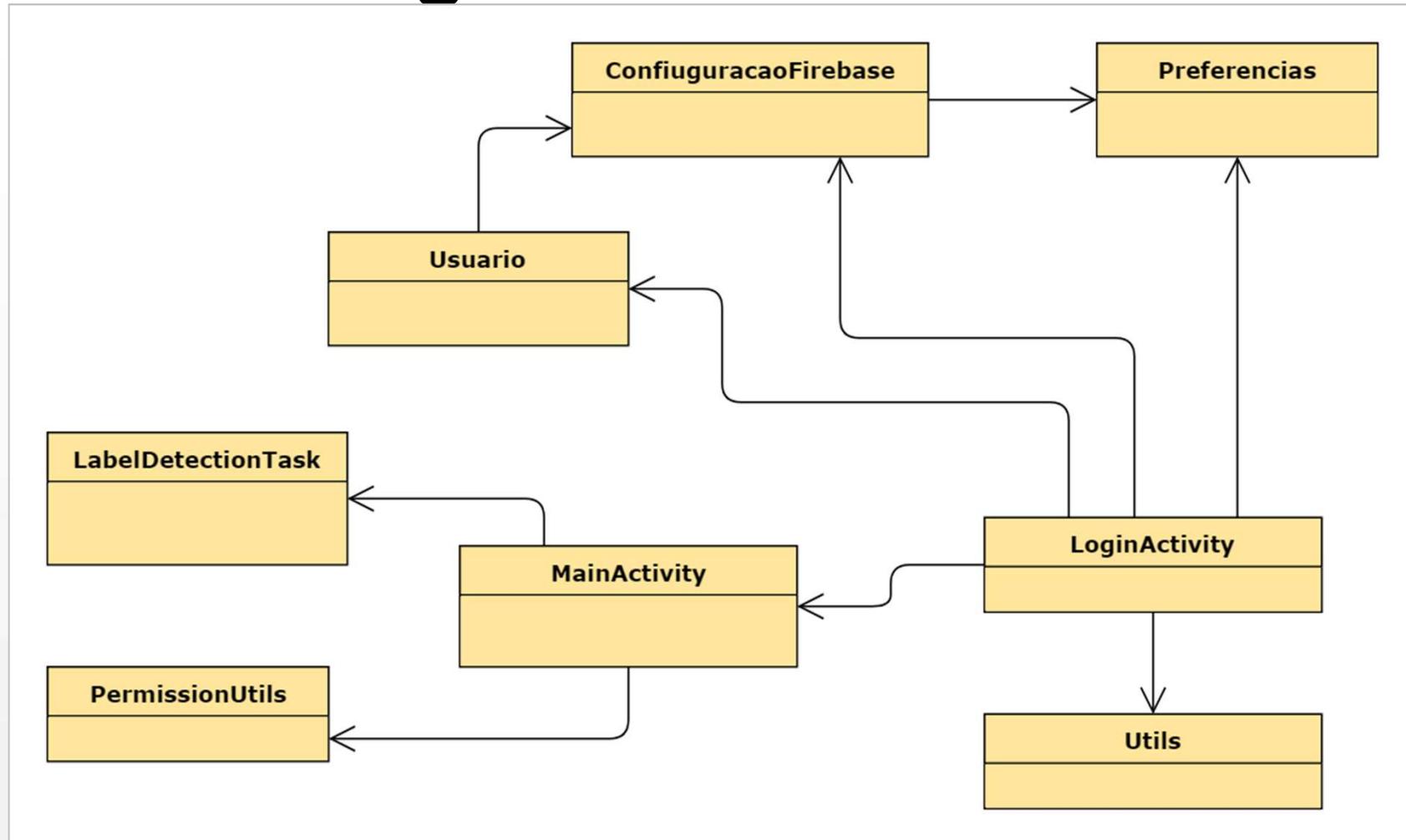
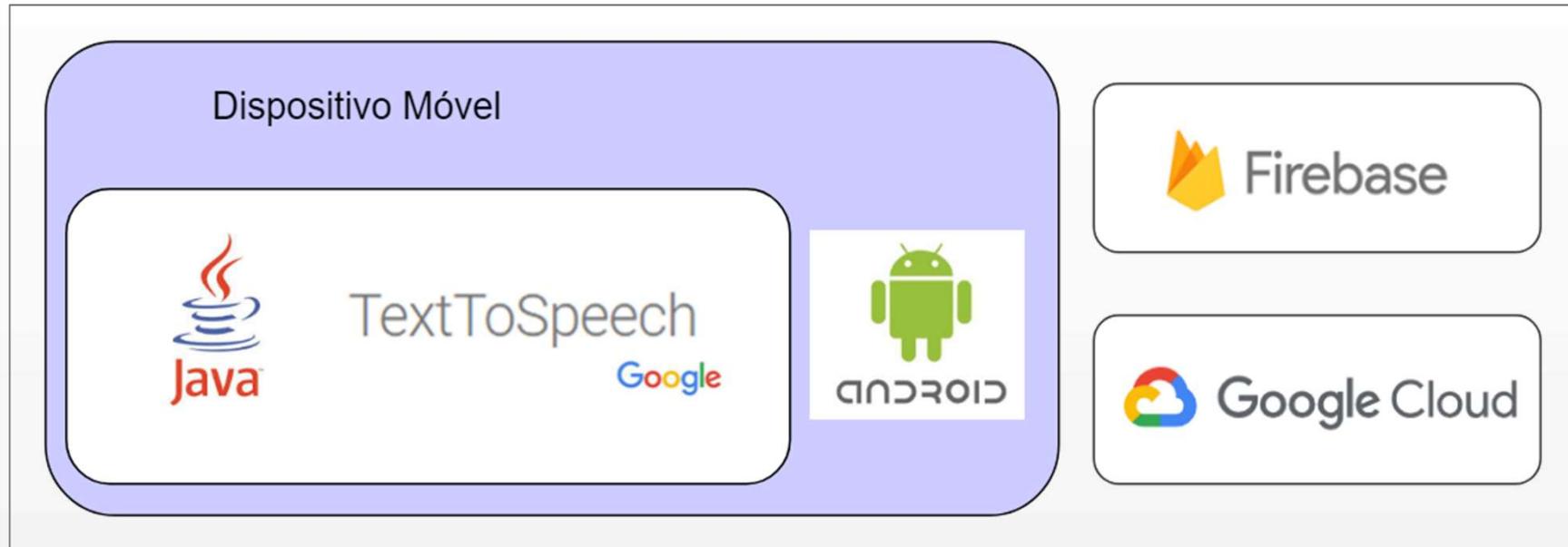
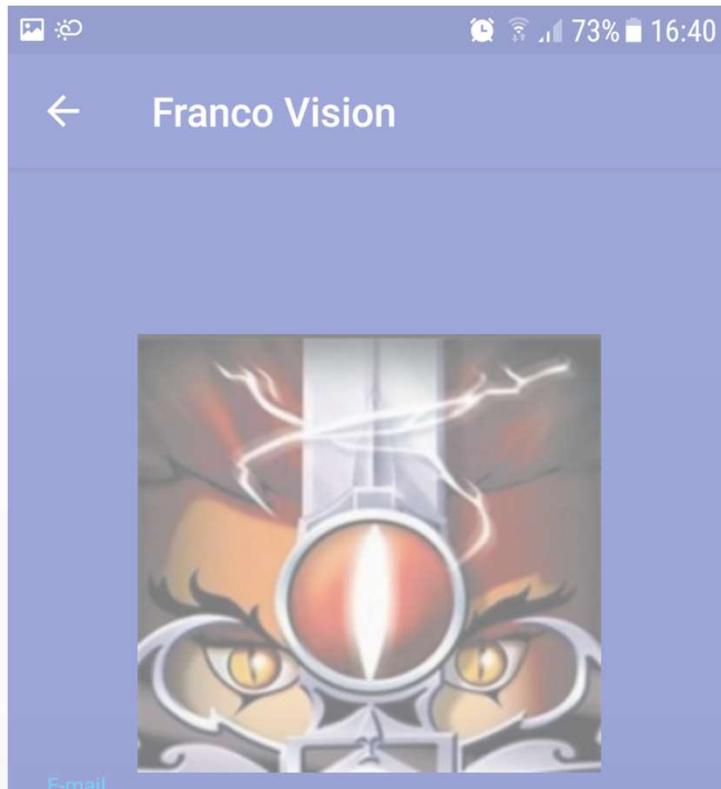


Diagrama de classes



Arquitetura da aplicação





Implementação

ACESSAR OU REGISTRAR

Análise de imagem

```
1 //Realiza a chamada da API para reconhecimento da imagem
2 private void callCloudVision(final Bitmap) {
3     // Altera o texto para processando
4     mImageDetails.setText(R.string.loading_message);
5
6     playInstruction(INSTRUCAO_ANALISE_EM_PROCESSAMENTO);
7
8     // Executa a tarefa de analise da imagem em background
9     try {
10        AsyncTask<Object, Void, String> labelDetectionTask = new
11        LableDetectionTask(this, prepareAnnotationRequest(bitmap));
12        labelDetectionTask.execute();
13    } catch (IOException e) {
14        Log.d(TAG, "failed to make API request because of other IOException " +
15            e.getMessage());
16    }
17 }
```

Análise de imagem

```
1 private Vision.Images.Annotate prepareAnnotationRequest(Bitmap) throws IOException
2 {
3     HttpTransport httpTransport = AndroidHttp.newCompatibleTransport();
4     JsonFactory jsonFactory = GsonFactory.getDefaultInstance();
5
6     VisionRequestInitializer requestInitializer =
7         new VisionRequestInitializer(CLOUD_VISION_API_KEY) {
8             //...
9             }
10        };
11
12    Vision.Builder builder = new Vision.Builder(httpTransport, jsonFactory, null);
13    builder.setVisionRequestInitializer(requestInitializer);
14    Vision vision = builder.build();
15    BatchAnnotateImagesRequest batchAnnotateImagesRequest =
16        new BatchAnnotateImagesRequest();
17    batchAnnotateImagesRequest.setRequests(new ArrayList<AnnotateImageRequest>()
18    {{
19        AnnotateImageRequest annotateImageRequest = new AnnotateImageRequest();
20
21        // Add the image
22        Image base64EncodedImage = new Image();
23        // Convert the bitmap to a JPEG
24        ByteArrayOutputStream byteArrayOutputStream = new ByteArrayOutputStream();
25        bitmap.compress(Bitmap.CompressFormat.JPEG, 90, byteArrayOutputStream);
26        byte[] imageBytes = byteArrayOutputStream.toByteArray();
27
```

Análise de imagem

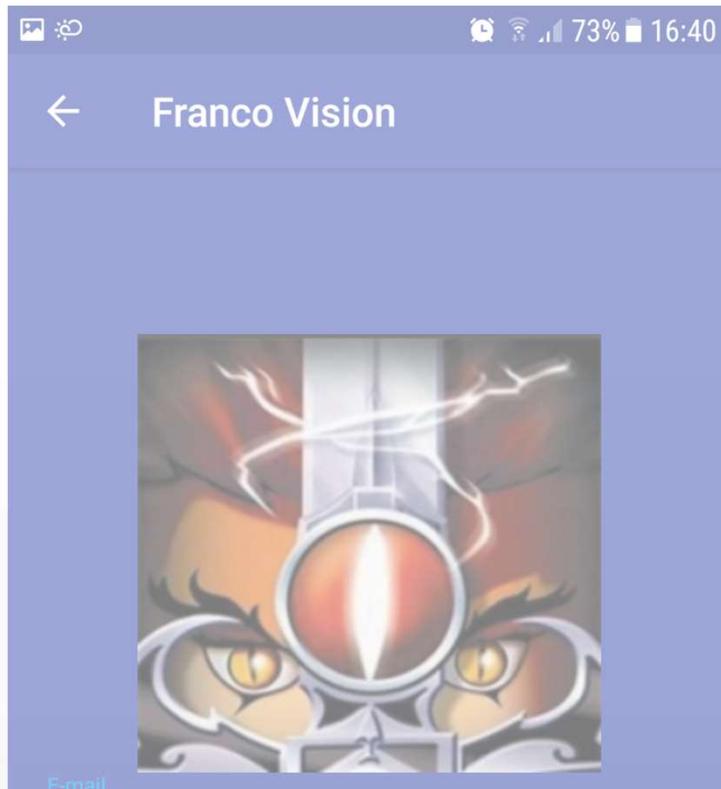
```
28 // Base64 encode the JPEG
29 base64EncodedImage.encodeContent(imageBytes);
30 annotateImageRequest.setImage(base64EncodedImage);
31
32 // add the features
33 annotateImageRequest.setFeatures(new ArrayList<Feature>() {{
34     Feature labelDetection = new Feature();
35     labelDetection.setType("LABEL_DETECTION");
36     labelDetection.setMaxResults(MAX_LABEL_RESULTS);
37     add(labelDetection);
38 }});
39
40
41 add(annotateImageRequest);
42 }});
43
44 Vision.Images.Annotate annotateRequest =
45     vision.images().annotate(batchAnnotateImagesRequest);
46 // ...
47 return annotateRequest;
48 }
49 }
```

Apresentação do resultado da análise

```
1 //conversão do resultado para texto e fala
2 private static String convertResponseToStringAndSpeech(BatchAnnotateImagesResponse response)
3 {
4     StringBuilder message = new StringBuilder("Itens encontrados na imagem:\n\n");
5     String resultadoAnalise;
6
7     List<EntityAnnotation> labels = response.getResponses().get(0).getLabelAnnotations();
8     if (labels != null) {
9
10        resultadoAnalise = "Items found in image: ";
11        tts.speak(resultadoAnalise, TextToSpeech.QUEUE_ADD, null, "TESTE");
12
13        for (EntityAnnotation label : labels) {
14            //Formata a string com os itens encontrados na imagem
15            message.append(String.format(Locale.US, "%.3f: %s", label.getScore(),
16 label.getDescription() ));
17            message.append("\n");
18            //Texto a ser sintetizado
19            tts.speak(label.getDescription(), TextToSpeech.QUEUE_ADD, null, "TESTE");
20        }
21    } else {
22        message.append("Nenhum objeto encontrado.");
23        resultadoAnalise = "No objects found.";
24        tts.speak(resultadoAnalise, TextToSpeech.QUEUE_ADD, null, "TESTE");
25    }
26    return message.toString();
27 }
```

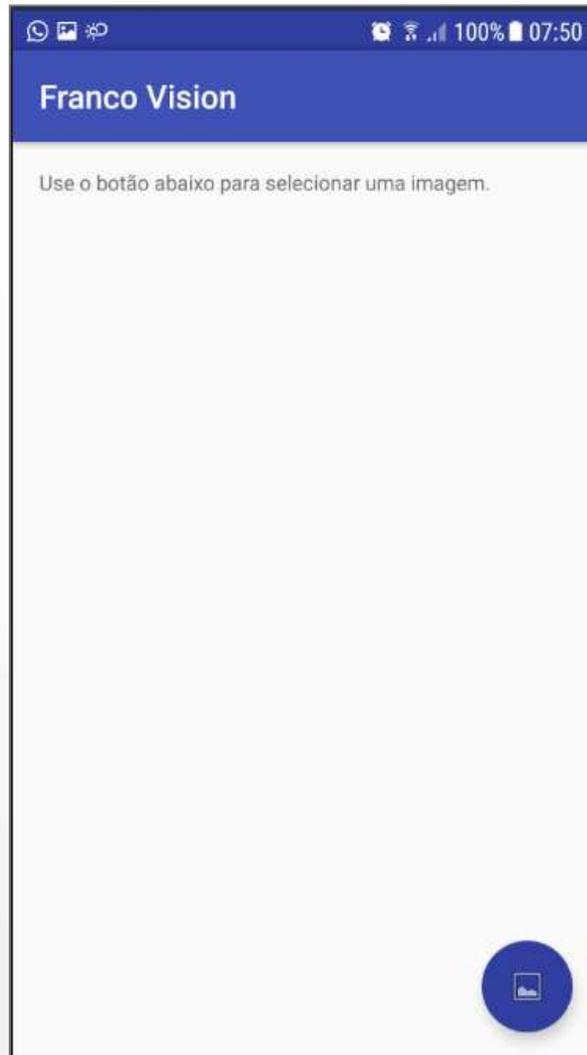
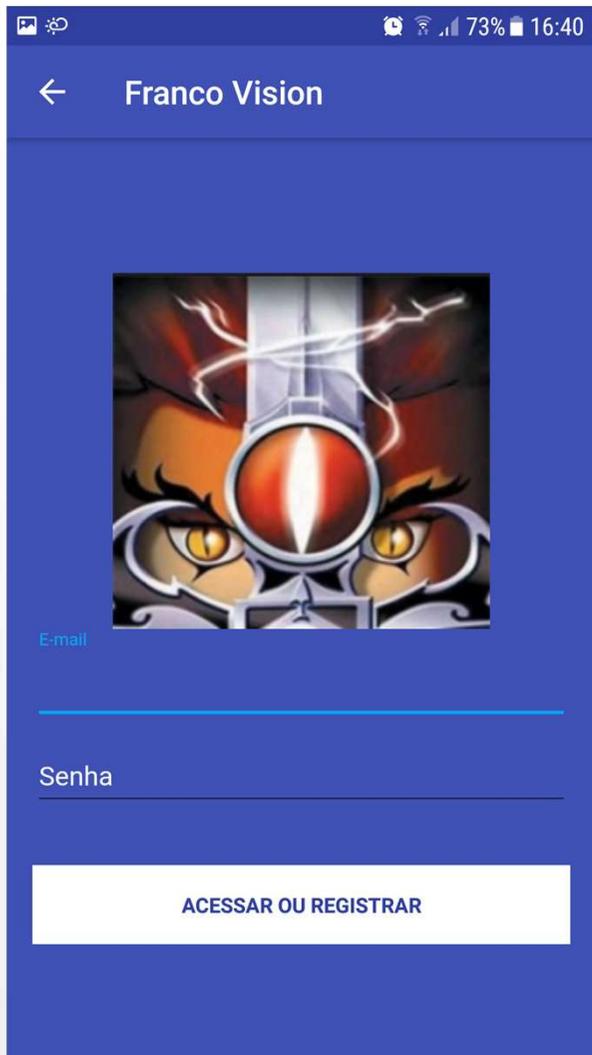
Instruções de voz ao usuário

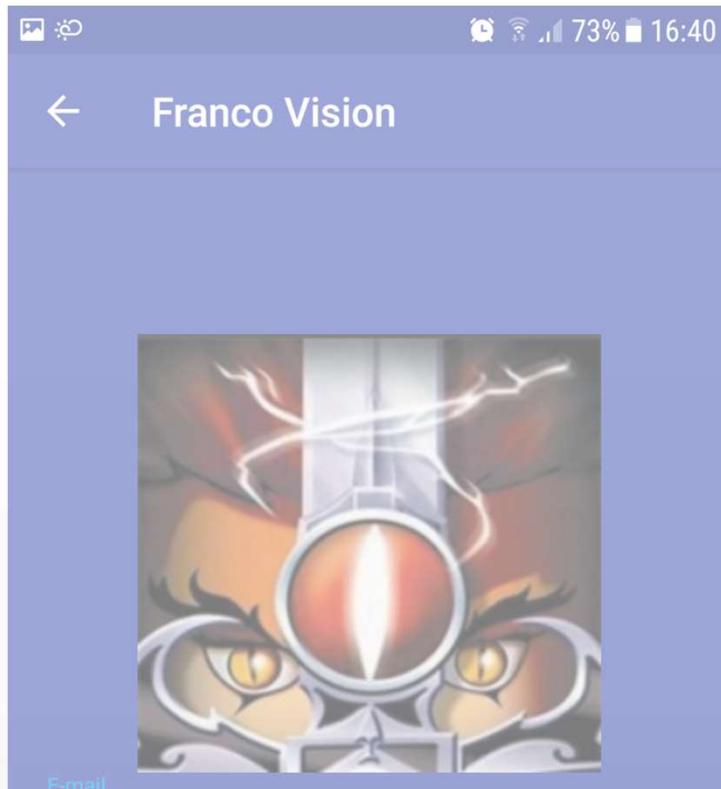
```
1 private void playInstruction (int instruction ) {
2
3     //Trata a instrução recebida e executada o arquivo correspondente (pré-gravado)
4     switch ( instruction ) {
5         case INSTRUCAO_INICIAL:
6             mediaPlayer = MediaPlayer.create(MainActivity.this,
7 R.raw.abertura_de_duplo_clique);
8             break;
9         case INSTRUCAO_CAMERA_ACIONADA:
10            mediaPlayer = MediaPlayer.create(MainActivity.this, R.raw.camera_ativada);
11            break;
12        case INSTRUCAO_ANALISE_EM_PROCESSAMENTO:
13            mediaPlayer = MediaPlayer.create(MainActivity.this,
14 R.raw.analisando_imagem);
15            break;
16        case INSTRUCAO_ANALISE_SUCESSO:
17            mediaPlayer = MediaPlayer.create(MainActivity.this, R.raw.analise_sucesso);
18            break;
19        // ....
20    }
21    // ....
22    //Executa o som
23    mediaPlayer.start();
24 }
```



Operacionalidade

ACESSAR OU REGISTRAR





Resultado e discussões

ACESSAR OU REGISTRAR

Critérios utilizados nos testes

- Fotos de objetos e cenas com paisagens do cotidiano
- Para considerar um resultado relevante foi avaliado o auxílio dado ao usuário em reconhecer o objeto e os detalhes da sua composição
- Consideradas apenas as classificações com pontuação maior do que 0.7
- 67% dos testes realizados foram considerados com resultado relevante

Exemplos de teste

🗨️ 📷 🔍 81% 17:28

Franco Vision

Itens encontrados na imagem:

- 0.947: yellow
- 0.899: flower
- 0.796: cut flowers
- 0.741: floristry
- 0.737: flower arranging
- 0.646: petal
- 0.645: floral design
- 0.558: spring
- 0.526: vase
- 0.515: decor



📷

🗨️ ✎️ ✎️ ... 64% 20:18

Franco Vision

Itens encontrados na imagem:

- 0.959: car
- 0.916: town
- 0.903: landmark
- 0.902: city
- 0.821: sky
- 0.818: street
- 0.769: downtown
- 0.692: town square
- 0.692: building
- 0.642: tourism



📷

Análise resultados

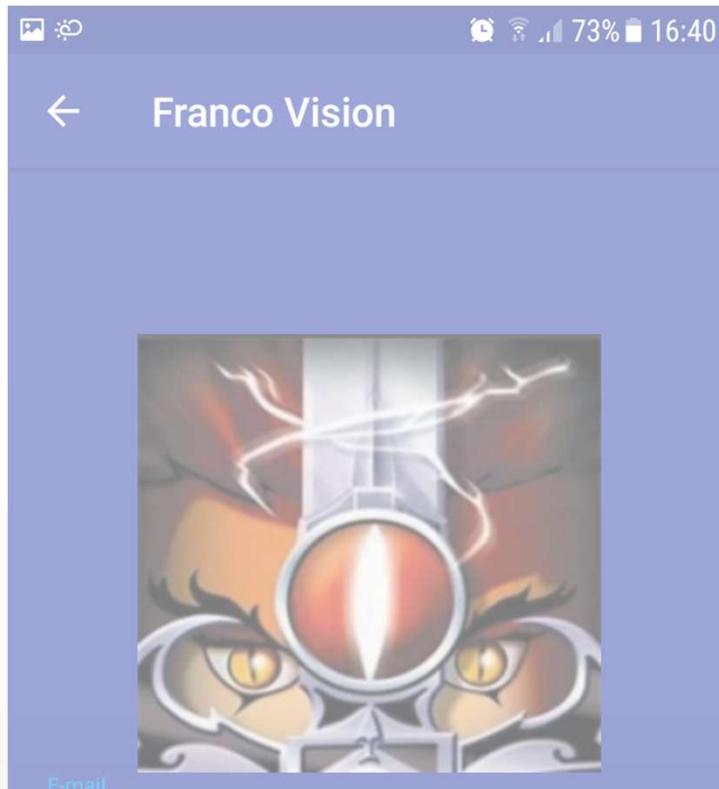
Conteúdo da imagem	Resultado da análise	Resultado relevante	Itens relevantes
Extintor de incêndio	Product, Product	Não	0/2
Motocicleta com um baú de carga	Product, motor vehicle, bicycle accessory, bicycle, vehicle, wheel	Sim	6/7
Rua contendo casas, vegetação rasteira e árvores ao fundo	Tree, sky, building, plant, facade	Sim	5/5
Armário contendo um forno elétrico e um forno de micro-ondas, ambos brancos	Home appliance, kitchen appliance, clothes dryer, major appliance, electronics, microwave oven	Sim	5/7
Maçã vermelha (cerca 80 cm de distância da câmera)	Nenhum item com pontuação acima de 0,7	Não	0/0
Maçã vermelha (cerca 10 cm de distância da câmera)	Fruit, apple, produce	Sim	2/3
Fechada de um prédio contendo outros prédios ao redor e árvores	Metropolitan area, tower block, skyscraper, condominium, urban area, building, city, metropolis, residential área, mixed use	Sim	9/10

Usabilidade

- Instruções por voz
- Acionamento da câmera
- Sintetização por voz do resultado da análise

Correlatos

Características / trabalhos relacionados	Kramer (2017)	Heinzle (2017)	Seefeld (2016)	Franco (2018)
Domínio	Reconhecimento dos textos em cartas de jogo para deficientes visuais	Reconhecimento de alimentos e sugestão de receitas	Reconhecimento de expressões faciais	Reconhecimento de objetos para deficientes visuais
Reconhecimento de imagem	Sim	Sim	Sim	Sim
Reconhecimento de texto	Sim	Não	Não	Não
Sintetização de voz	Sim	Não	Não	Sim
Utiliza Internet	Sim	Sim	Não	Sim
Tecnologia assistiva	Sim	Não	Não	Sim
Plataforma suportada	Android/IOS	Android/IOS	Windows (Desktop)	Android



Conclusões e sugestões

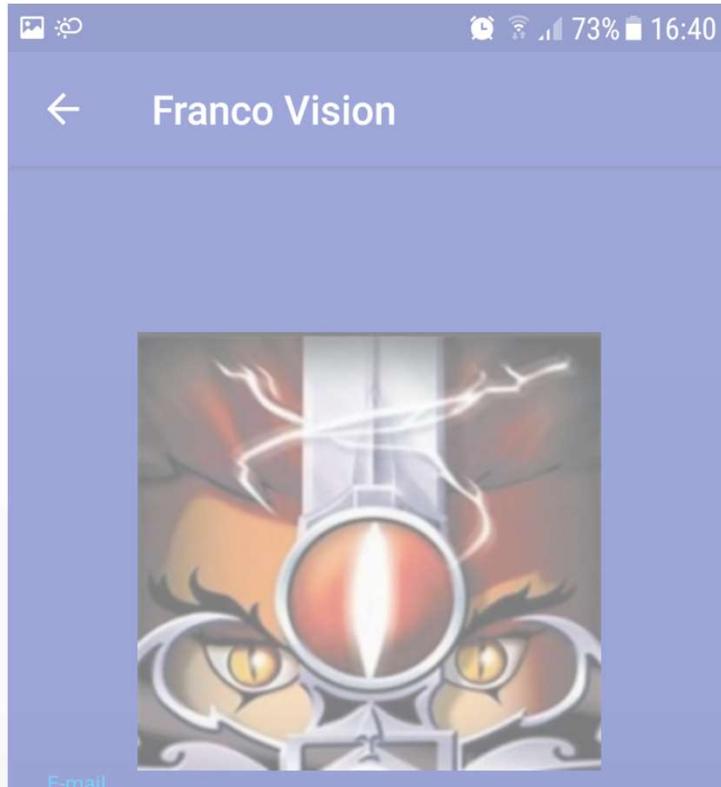
ACESSAR OU REGISTRAR

Conclusões

- O objetivo principal do projeto, que era o de criar um aplicativo que auxilie as pessoas com deficiência visual a identificar objetos em uma paisagem, foi atingido.
- 67% dos testes realizados apresentaram resultado satisfatório, demonstrando detalhes do conteúdo da imagem
- Necessita de aprimoramentos na usabilidade da captura da foto para ser utilizado no dia a dia
- O resultado da análise precisa ser traduzido para português para aumentar o público apto a utilizar o aplicativo
- A utilização da API Cloud Vision possui custos, o que implicaria a cobrança de uma mensalidade dos usuários
- Importância das tecnologias assistivas

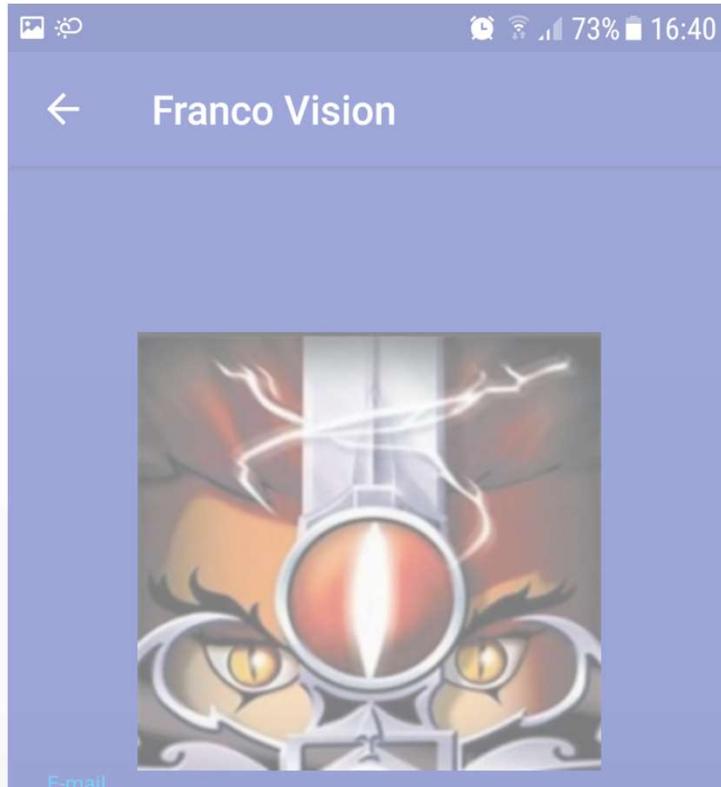
Extensões sugeridas

- Melhorar a utilização da câmera do celular, removendo a confirmação da imagem capturada
- Tradução do resultado da análise da imagem para português
- Evolução do modelo de análise da imagem para permitir que o usuário possa determinar o tipo de análise a ser executada conforme o contexto
- Criação de uma API REST própria que permita realizar a análise da imagem sem utilizar a API Cloud Vision
- Adição de outras funcionalidades voltadas à acessibilidade, como informar a localização atual do usuário, ler o conteúdo de placas ou identificar cores



Demonstração

ACESSAR OU REGISTRAR



Perguntas?

ACESSAR OU REGISTRAR