Android e DialogFlow

Curso de ChatBots

Android e DialogFlow

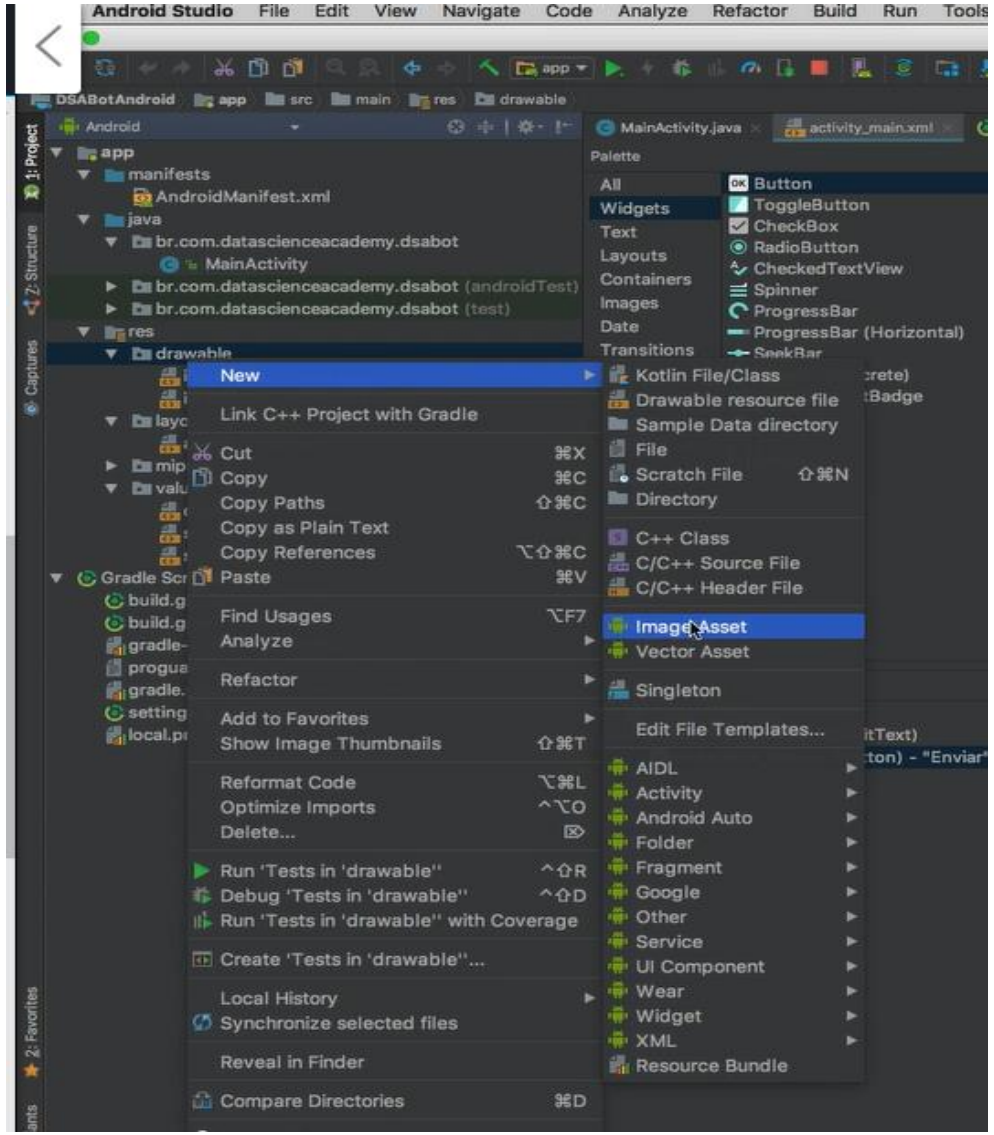
Desenhando o Layout do aplicativo

Componente	Propriedade	Valor
Plain Text	**Name**	consulta_edittext
	Text	
	Hint	Digite algo aqui...
Button	**Name**	consultar_button
	Text	Enviar
	onClick	consultar
imageButton	**Name**	mic_imageButton
	Icone	
TextView	**Name**	resultado_textview
	TextSize	18

Android e DialogFlow

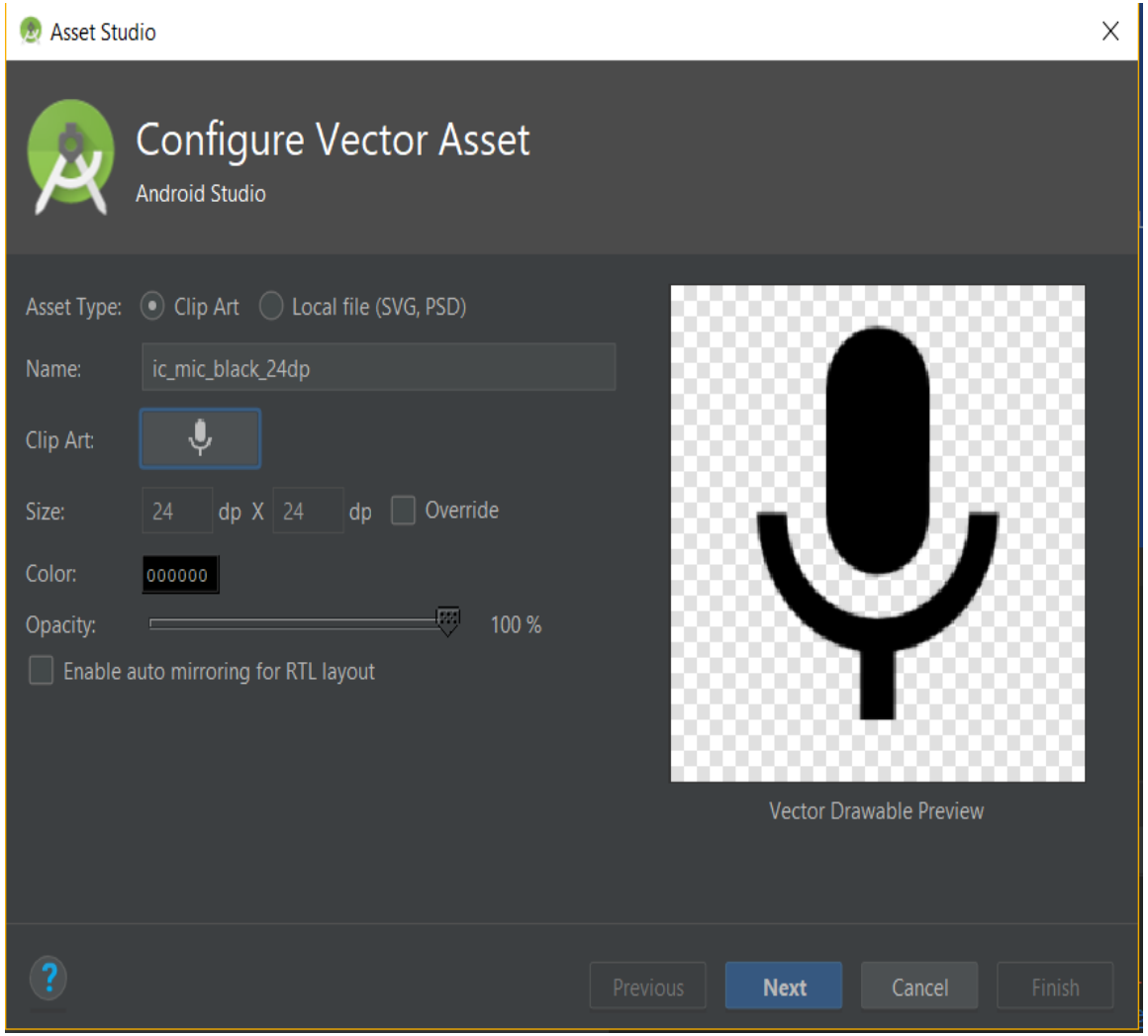
Criando o ícone

Selecione Vector Asset, como mostra a figura



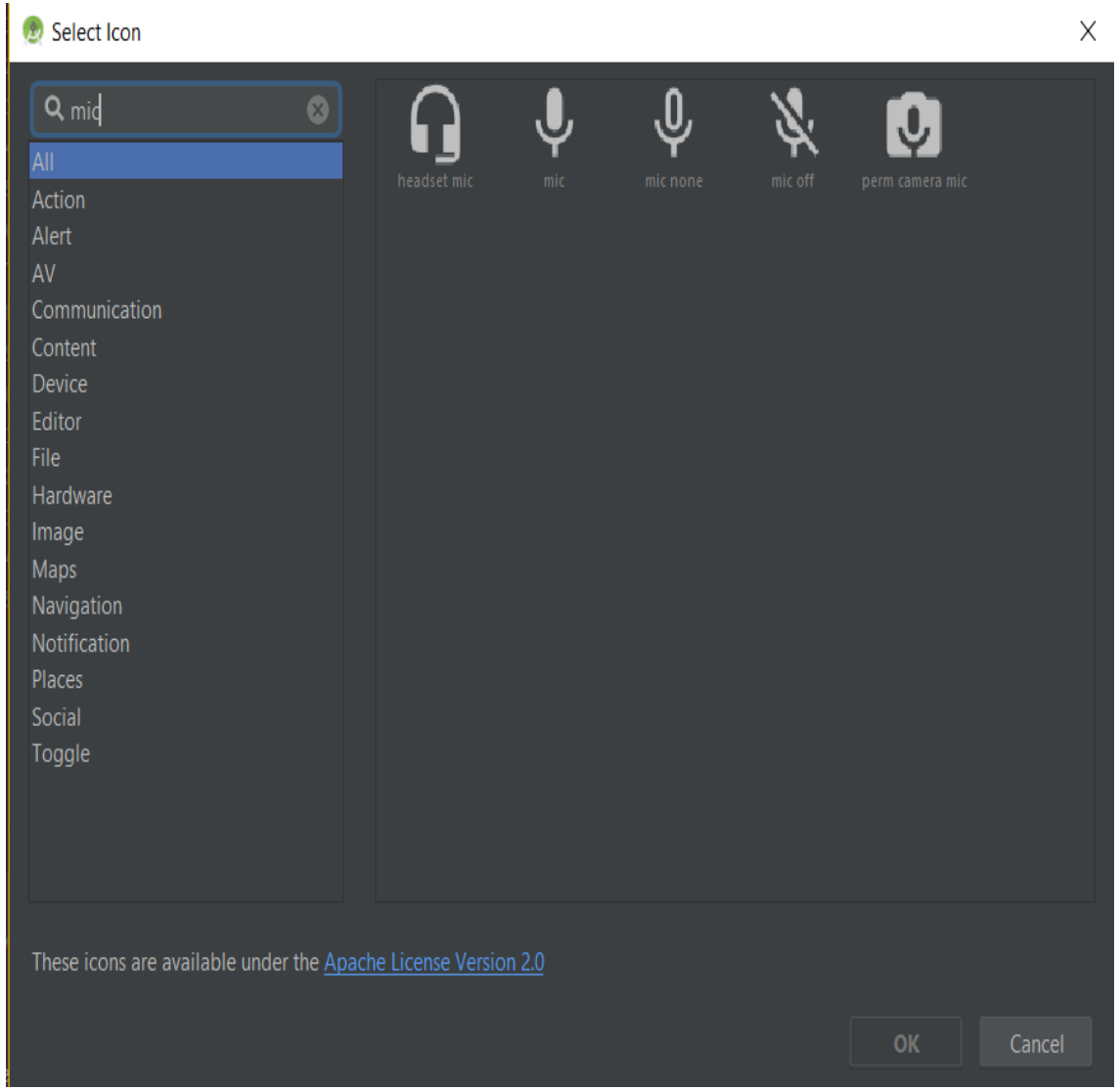
Android e DialogFlow

Clique no Clip Art do microfone na figura ao lado



Android e DialogFlow

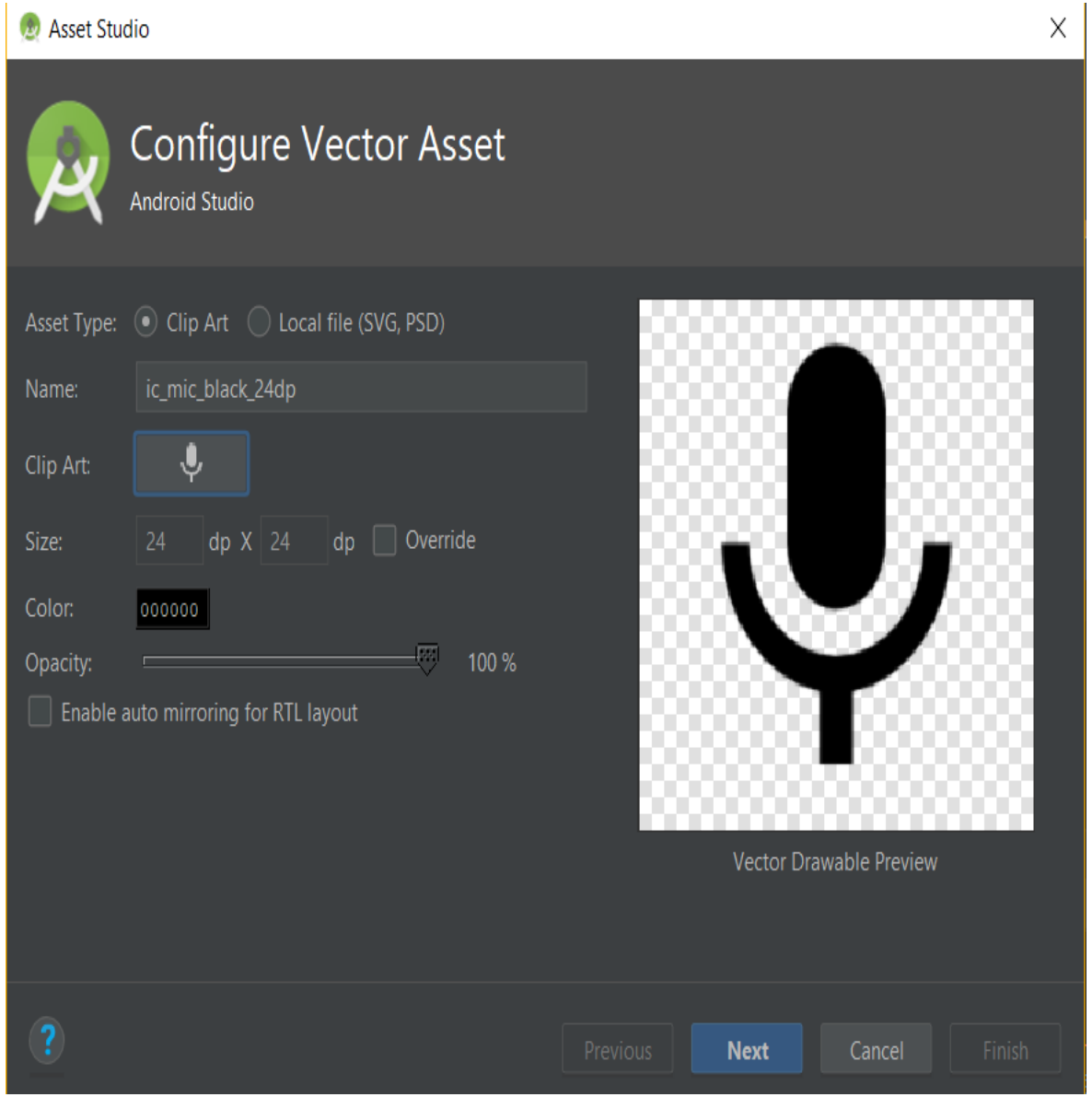
Digite mic na busca e clique no microfone na figura ao lado e depois OK.



Android e DialogFlow

O ícone foi criad e o nome é ic_mic_black_24dp.

Atribua esse ícone ao componente imageButton



Android e DialogFlow

Importando a biblioteca do DialogFlow

Selecione o menu Docs → SDKS → Android SDK

Platform	Docs
Android SDK	Android SDK Docs
Botkit SDK	Botkit SDK Docs
C++	C++ Docs
Cordova SDK	Cordova SDK Docs
HTML + JS Example	
iOS SDK	iOS SDK Docs
Java SDK	Java SDK Docs
JavaScript SDK	JavaScript SDK Docs
.NET (WP8, W10)	.NET SDK Docs
Node.js SDK	Node.js SDK Docs
Python SDK	Python SDK Docs
Ruby SDK	Ruby SDK Docs
Unity SDK	Unity SDK Docs
Xamarin SDK	Xamarin SDK Docs

Android e DialogFlow

Adicione as seguintes bibliotecas as dependências do seu app.

```
21  dependencies {
22      implementation fileTree(dir: 'libs', include: ['*.jar'])
23      implementation 'com.android.support:appcompat-v7:27.1.1'
24      implementation 'com.android.support.constraint:constraint-layout:1.1.3'
25      testImplementation 'junit:junit:4.12'
26      androidTestImplementation 'com.android.support.test:runner:1.0.2'
27      androidTestImplementation 'com.android.support.test.espresso:espresso-core:3.0.2'
28      compile 'ai.api:sdk:2.0.7@aar'
29      compile 'ai.api:libai:1.6.12'
30  }
```

Android e DialogFlow

Declarando permissões de acesso

```xml
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="br.edu.ifrn.myappbot">

    <uses-permission android:name="android.permission.INTERNET"/>
    <uses-permission android:name="android.permission.RECORD_AUDIO"/>

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>

</manifest>
```

manifest

Android e DialogFlow

Configurações do DialogFlow

Criar uma instância de AIConfiguration, especificando o token de acesso, localidade e mecanismo de reconhecimento.

```
final AIConfiguration config = new AIConfiguration("CLIENT_ACCESS_TOKEN",
            AIConfiguration.SupportedLanguages.English,
            AIConfiguration.RecognitionEngine.System);
```

Precisamos conectar nossa aplicação ao bot no DialogFlow. Isso é feito através do parâmetro CLIENT_ACCESS_TOKEN.

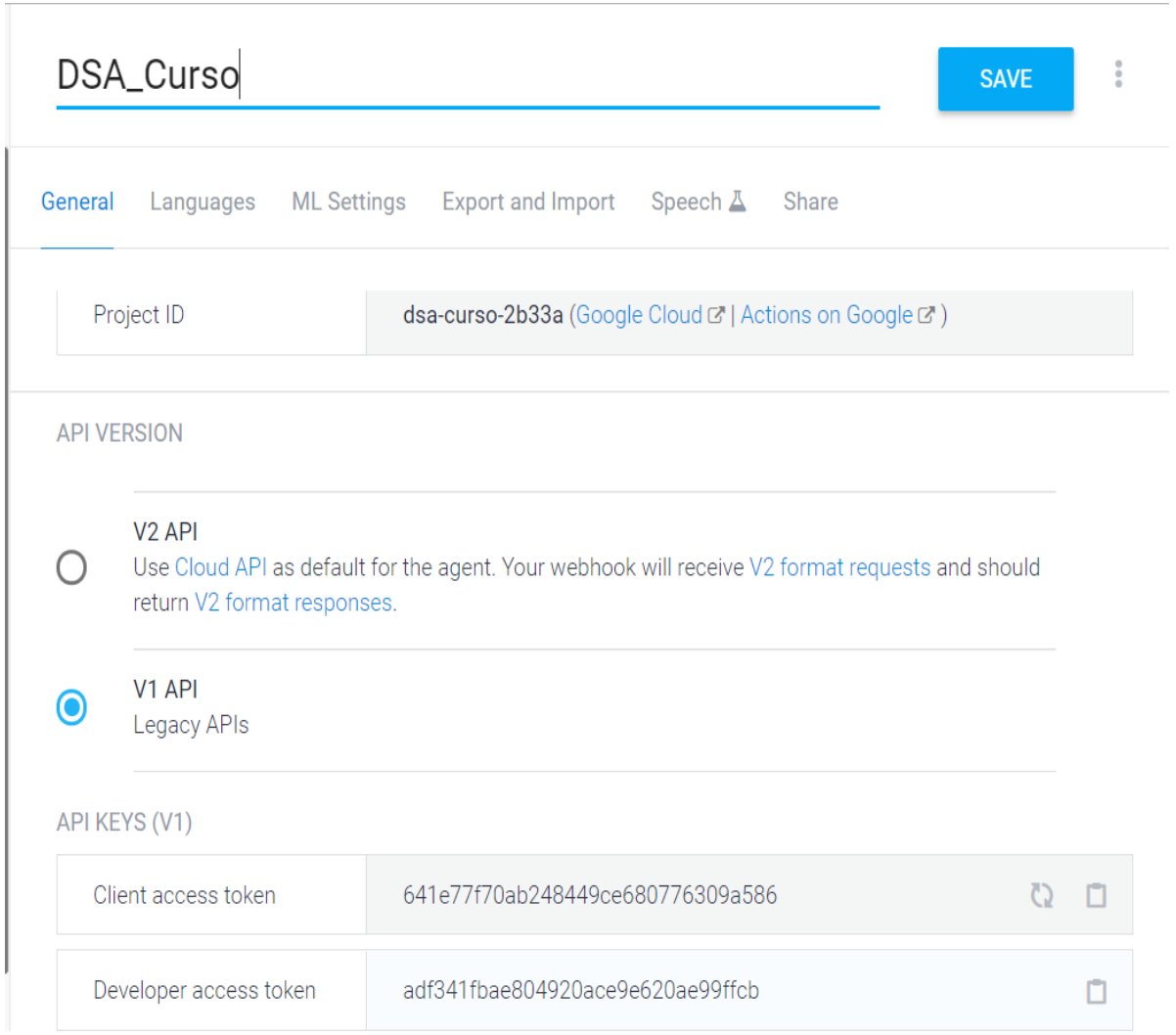
Crie a constante a seguir no arquivo MainActivity.java

```
private static final String CLIENT_ACCESS_TOKEN = "";
```

Android e DialogFlow

Obtendo o token do bot no DialogFlow.

Copie o token da chave Client access token e cole no valor da constante criar anteriormente.




```
private static final String CLIENT_ACCESS_TOKEN = "641e77f70ab248449ce680776309a586";
```

Android e DialogFlow

Agora substitua “CLIENT_ACCESS_TOKEN” pela constante

```
26
27          final AIConfiguration config = new AIConfiguration (CLIENT_ACCESS_TOKEN,
28                  AIConfiguration.SupportedLanguages.English,
29                  AIConfiguration.RecognitionEngine.System);
30
31      }
32
```

Android e DialogFlow

Requisições para o DialogFlow

Use o objeto AIConfiguration para obter uma referência ao AIService, que fará as solicitações de consulta.

```
AIService aiService = AIService.getService(context, config);
```

Copie esse código e cole no app

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    consultaEditText = findViewById(R.id.consulta_editext);
    resultadoTextView = findViewById(R.id.resultado_textView);

    final AIConfiguration config = new AIConfiguration (CLIENT_ACCESS_TOKEN,
            AIConfiguration.SupportedLanguages.English,
            AIConfiguration.RecognitionEngine.System);

    aiService = AIService.getService( context: this, config);


}
```

Android e DialogFlow

Fazendo uma requisição, ou seja, quando o usuário clicar em um botão, será feita uma requisição.

```
new AsyncTask<AIRequest, Void, AIResponse>() {

    @Override
    protected void onPreExecute() {        super.onPreExecute();
        resultadoTextView.setText("Processando...");
    }

    @Override
    protected AIResponse doInBackground(AIRequest... requests) {
        final AIRequest request = requests[0];
        try {
            final AIResponse response = aiService.textRequest(aiRequest);
            return response;
        } catch (AIServiceException e) {
        }
        return null;
    }
    @Override
    protected void onPostExecute(AIResponse aiResponse) {
        if (aiResponse != null) {
            resultadoTextView.setText(aiResponse.getResult().getFulfillment().getSpeech());
        }
        else {
            resultadoTextView.setText("Ocorreu um erro.");
        }
    }
}.execute(aiRequest);
```

Android e DialogFlow

Código completo, Parte 1

```
1  package br.edu.ifrn.myappbot;
2
3  import ...
16
17 public class MainActivity extends AppCompatActivity {
18
19     private static final String CLIENT_ACCESS_TOKEN = "641e77f70ab248449ce680776309a586";
20
21     private EditText consultaEditText;
22     private TextView resultadoTextView;
23
24     private AIService aiService;
25
26     @Override
27     protected void onCreate(Bundle savedInstanceState) {
28         super.onCreate(savedInstanceState);
29         setContentView(R.layout.activity_main);
30
31         consultaEditText = findViewById(R.id.consulta_editext);
32         resultadoTextView = findViewById(R.id.resultado_textView);
33
34         final AIConfiguration config = new AIConfiguration (CLIENT_ACCESS_TOKEN,
35                 AIConfiguration.SupportedLanguages.English,
36                 AIConfiguration.RecognitionEngine.System);
37
38         aiService = AIService.getService( context: this, config);
39
40
41     }
```

Android e DialogFlow

Código completo, Parte 2

```
46
47          if(consultaEditText.getText().toString().trim().equals("")){
48              Toast.makeText(this, "Digite algo...", Toast.LENGTH_SHORT).show();
49              return;
50          }
51          final AIRequest aiRequest = new AIRequest();
52          aiRequest.setQuery(consultaEditText.getText().toString());
53          //Agora faço a requisição
```

Android e DialogFlow

Código completo, Parte 2

```
        new AsyncTask<AIRequest, Void, AIResponse>() {

            @Override
            protected void onPreExecute() {
                super.onPreExecute();
                resultadoTextView.setText("Processando...");
            }

            @Override
            protected AIResponse doInBackground(AIRequest... requests) {
                final AIRequest request = requests[0];
                try {
                    final AIResponse response = aiService.textRequest(aiRequest);
                    return response;
                } catch (AIServiceException e) {
                }
                return null;
            }
            @Override
            protected void onPostExecute(AIResponse aiResponse) {
                if (aiResponse != null) {
                    resultadoTextView.setText(aiResponse.getResult().getFulfillment().getSpeech());
                }
                else {
                    resultadoTextView.setText("Ocorreu um erro.");
                }
            }
        }.execute(aiRequest);
    }
}
```

Android e DialogFlow

Manifesto

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="br.edu.ifrn.myappbot">

    <uses-permission android:name="android.permission.INTERNET"/>
    <uses-permission android:name="android.permission.RECORD_AUDIO"/>

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="MyAppBot"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>

</manifest>
```

Android e DialogFlow

app

```
apply plugin: 'com.android.application'

android {
    compileSdkVersion 27
    defaultConfig {
        applicationId "br.edu.ifrn.myappbot"
        minSdkVersion 17
        targetSdkVersion 27
        versionCode 1
        versionName "1.0"
        testInstrumentationRunner "android.support.test.runner.AndroidJUnitRunner"
    }
    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-rules.pro'
        }
    }
}

dependencies {
    implementation fileTree(dir: 'libs', include: ['*.jar'])
    implementation 'com.android.support:appcompat-v7:27.1.1'
    implementation 'com.android.support.constraint:constraint-layout:1.1.3'
    testImplementation 'junit:junit:4.12'
    androidTestImplementation 'com.android.support.test:runner:1.0.2'
    androidTestImplementation 'com.android.support.test.espresso:espresso-core:3.0.2'
    compile 'ai.api:sdk:2.0.7@aar'
    compile 'ai.api:libai:1.6.12'
}
```

Android e DialogFlow

Integrando DialogFlow via Voz

Existe uma interface já declarada dentro da biblioteca do DialogFlow que e esta

```java
public interface AIListener {
    void onResult(AIResponse result); // here process response
    void onError(AIError error); // here process error
    void onAudioLevel(float level); // callback for sound level
visualization
    void onListeningStarted(); // indicate start listening here
    void onListeningCanceled(); // indicate stop listening here
    void onListeningFinished(); // indicate stop listening here
}
```

Ela específica algumas assinaturas de métodos, como pode-se ver acima, que são contratos, ou seja, a classe que implementar essa interface terá que implementar esses métodos.

Android e DialogFlow

Vamos implementar a interface no nosso aplicativo, ou seja, a classe **MainActivity** deve implementar a interface **AIListiner**, veja o código a seguir

```
public class MainActivity extends AppCompatActivity implements AIListener {
```

como assinamos um contrato, temos que implementar todos os métodos declarados na interface *AIListiner*.

```
87      @Override
88      public void onResult(AIResponse result) {
89          
90      }
91
92      @Override
93      public void onError(AIError error) {
94
95      }
96
97      @Override
98      public void onAudioLevel(float level) {
99
100     }
101
102     @Override
103     public void onListeningStarted() {
104
105     }
106
107     @Override
108     public void onListeningCanceled() {
109
110     }
111
112     @Override
113     public void onListeningFinished() {
114
115     }
```

Android e DialogFlow

Antes de adicionar código aos métodos implementados, temos que mudar o idioma de inglês para português do Brasil, uma vez que iremos falar ao invés de digitar a solicitação. Veja a alteração feita

Antes

```
final AIConfiguration config = new AIConfiguration
(CLIENT_ACCESS_TOKEN,
        AIConfiguration.SupportedLanguages.English,
        AIConfiguration.RecognitionEngine.System);
```

Depois

```
final AIConfiguration config = new AIConfiguration
(CLIENT_ACCESS_TOKEN,

AIConfiguration.SupportedLanguages.PortugueseBrazil,
        AIConfiguration.RecognitionEngine.System);
```

Android e DialogFlow

precisamos ainda, definir a classe que irá receber as notificações. Então acrescente o seguinte código abaixo de **final...**

```
aiService.setListener(this);
```

Veja o resultado ao lado →

```
final AIConfiguration config = new AIConfiguration
(CLIENT_ACCESS_TOKEN,

AIConfiguration.SupportedLanguages.PortugueseBrazil,
        AIConfiguration.RecognitionEngine.System);

aiService = AIService.getService(this, config);

aiService.setListener(this);
```

Android e DialogFlow

Sintetização de voz

Na classe MainActivity declara a variável

```
private TextToSpeech tts;
```

E no evento **onCreate**, instancie o objeto tts da seguinte forma

```
tts = new TextToSpeech(this, new
TextToSpeech.OnInitListener() {
    @Override
    public void onInit(int i) {
        //Setar o idioma
        tts.setLanguage(new Locale("pt-BR"));
    }
});
```

Android e DialogFlow

Permissão em tempo de execução

O código a seguir verifica se o aplicativo tem a permissão para ler os contatos do usuário e solicita a permissão, se necessário:

```
// Here, thisActivity is the current activity
if (ContextCompat.checkSelfPermission(thisActivity,
                Manifest.permission.READ_CONTACTS)
        != PackageManager.PERMISSION_GRANTED) {

    // Should we show an explanation?
    if (ActivityCompat.shouldShowRequestPermissionRationale(thisActivity,
            Manifest.permission.READ_CONTACTS)) {

        // Show an expanation to the user *asynchronously* -- don't block
        // this thread waiting for the user's response! After the user
        // sees the explanation, try again to request the permission.

    } else {

        // No explanation needed, we can request the permission.

        ActivityCompat.requestPermissions(thisActivity,
                new String[]{Manifest.permission.READ_CONTACTS},
                MY_PERMISSIONS_REQUEST_READ_CONTACTS);

        // MY_PERMISSIONS_REQUEST_READ_CONTACTS is an
        // app-defined int constant. The callback method gets the
        // result of the request.
    }
}
```

https://developer.android.com/training/permissions/requesting?hl=pt-br

Android e DialogFlow

O código anterior deve ser colado dentro do evento **consultarVoz**, que é evocado quando o usuário clica no microfone. Veja a implementação a seguir

```
public void consultarVoz(View v){

    if (ContextCompat.checkSelfPermission(this, Manifest.permission.RECORD_AUDIO)!=
PackageManager.PERMISSION_GRANTED) {

        // Should we show an explanation?
        if (ActivityCompat.shouldShowRequestPermissionRationale(this, Manifest.permission.RECORD_AUDIO)) {


        } else {

            ActivityCompat.requestPermissions(this, new String[]{Manifest.permission.RECORD_AUDIO},
                    MY_PERMISSIONS_REQUEST_CODE_RECORD_AUDIO);

        }
    } else {
        //Escutar o microfone
        aiService.startListening();
    }
}
```

Android e DialogFlow

```
@Override
public void onRequestPermissionsResult(int requestCode, @NonNull String[] permissions,
@NonNull int[] grantResults) {
    super.onRequestPermissionsResult(requestCode, permissions, grantResults);

    if(requestCode == MY_PERMISSIONS_REQUEST_CODE_RECORD_AUDIO) {
        if(grantResults[0] == PackageManager.PERMISSION_GRANTED) {
            aiService.startListening();
        }
    }
}
```

Android e DialogFlow

Os próximos slides contém o código completo do evento **MainActivity.java**

```
1   package br.edu.ifrn.myappbot;
2
3   import ...
29
30  public class MainActivity extends AppCompatActivity implements AIListener {
31
32      private static final String CLIENT_ACCESS_TOKEN = "641e77f70ab248449ce680776309a586";
33      private static final int MY_PERMISSIONS_REQUEST_CODE_RECORD_AUDIO = 10000;
34
35      private EditText consultaEditText;
36      private TextView resultadoTextView;
37
38      private AIService aiService;
39
40      private TextToSpeech tts;
```

Android e DialogFlow

```
41
42      @Override
43      protected void onCreate(Bundle savedInstanceState) {
44          super.onCreate(savedInstanceState);
45          setContentView(R.layout.activity_main);
46
47          consultaEditText = findViewById(R.id.consulta_editext);
48          resultadoTextView = findViewById(R.id.resultado_textView);
49
50          final AIConfiguration config = new AIConfiguration (CLIENT_ACCESS_TOKEN,
51                  AIConfiguration.SupportedLanguages.PortugueseBrazil,
52                  AIConfiguration.RecognitionEngine.System);
53
54          aiService = AIService.getService( context: this, config);
55
56          aiService.setListener(this);
57
58          tts = new TextToSpeech( context: this, (i) → {
61                  //Setar o idioma
62                  tts.setLanguage(new Locale( language: "pt-BR"));
63          });
65      }
```

Android e DialogFlow

```
67      public void consultar (View v){
68          //Toast.makeText(this, "Olá IFRN", Toast.LENGTH_SHORT).show();
69          resultadoTextView.setText(consultaEditText.getText().toString());
70
71          if(consultaEditText.getText().toString().trim().equals("")){
72              Toast.makeText(this, "Digite algo...", Toast.LENGTH_SHORT).show();
73              return;
74          }
75          final AIRequest aiRequest = new AIRequest();
76          aiRequest.setQuery(consultaEditText.getText().toString());
77          //Agora faço a requisição
78
79          new AsyncTask<AIRequest, Void, AIResponse>() {
80
81              @Override
82              protected void onPreExecute() {
83                  super.onPreExecute();
84                  resultadoTextView.setText("Processando...");
85              }
86
87              @Override
88              protected AIResponse doInBackground(AIRequest... requests) {
89                  final AIRequest request = requests[0];
90                  try {
91                      final AIResponse response = aiService.textRequest(aiRequest);
92                      return response;
93                  } catch (AIServiceException e) {
94                  }
95                  return null;
96              }
97              @Override
```

Android e DialogFlow

```
97              @Override
98              protected void onPostExecute(AIResponse aiResponse) {
99                  if (aiResponse != null) {
100                     resultadoTextView.setText(aiResponse.getResult().getFulfillment().getSpeech());
101                 }
102                 else {
103                     resultadoTextView.setText("Ocorreu um erro.");
104                 }
105             }
106         }.execute(aiRequest);
107     }
```

Android e DialogFlow

```
109        @Override
110        public void onResult(AIResponse result) {
111            if(result != null && !result.isError()){
112                String resposta = result.getResult().getFulfillment().getSpeech();
113
114                resultadoTextView.setText(resposta);
115
116                if (Build.VERSION.SDK_INT >= 21 && tts != null) {
117                    tts.speak(resposta, TextToSpeech.QUEUE_FLUSH,  params: null,  utteranceId: null);
118                }
119            }
120        }
```

Android e DialogFlow

```
121
122         @Override
123         public void onError(AIError error) { resultadoTextView.setText("Erro: " + error.getMessage()); }
126
127         @Override
128         public void onAudioLevel(float level) {
129
130         }
131
132         @Override
133         public void onListeningStarted() {
134
135         }
136
137         @Override
138         public void onListeningCanceled() {
139
140         }
141
142         @Override
143         public void onListeningFinished() {
144
145         }
```

Android e DialogFlow

```
147     //Evento consultarVoz
148     public void consultarVoz(View v){
149
150         if (ContextCompat.checkSelfPermission( context: this,
151                 Manifest.permission.RECORD_AUDIO)
152                 != PackageManager.PERMISSION_GRANTED) {
153
154             // Should we show an explanation?
155             if (ActivityCompat.shouldShowRequestPermissionRationale( activity: this,
156                     Manifest.permission.RECORD_AUDIO)) {
157
158                 AlertDialog.Builder janela = new AlertDialog.Builder( context: MainActivity.this );
159                 janela.setTitle("Ação requerida...");
160                 janela.setMessage("Conceda permissão na próxima tela para que possamos utilizar seu
161                 janela.setPositiveButton( text: "OK", new DialogInterface.OnClickListener() {
162                     @Override
163                     public void onClick(DialogInterface dialogInterface, int i) {
164                         ActivityCompat.requestPermissions( activity: MainActivity.this,
165                                 new String[]{Manifest.permission.RECORD_AUDIO},
166                                 MY_PERMISSIONS_REQUEST_CODE_RECORD_AUDIO);
167                     }
168                 });
169
170                 janela.show();
171
172             } else {
173
```

Android e DialogFlow

```
174                 // No explanation needed, we can request the permission.
175
176                 ActivityCompat.requestPermissions( activity: this,
177                         new String[]{Manifest.permission.RECORD_AUDIO},
178                         MY_PERMISSIONS_REQUEST_CODE_RECORD_AUDIO);
179             }
180         } else {
181             //Escutar o microfone
182             aiService.startListening();
183         }
184     }
185
186     @Override
187     public void onRequestPermissionsResult(int requestCode, @NonNull String[] permissions, @NonNull
188         super.onRequestPermissionsResult(requestCode, permissions, grantResults);
189
190         if(requestCode == MY_PERMISSIONS_REQUEST_CODE_RECORD_AUDIO) {
191             if(grantResults[0] == PackageManager.PERMISSION_GRANTED) {
192                 aiService.startListening();
193             }
194         }
195     }
196 }
```