

Reconhecimento de Fala

Speech Recognition

Prof. Me. Hélio Esperidião

Speech Recognition

- Reconhecimento de fala é uma área interdisciplinar originária da linguística computacional cujo objetivo é desenvolver métodos e tecnologias que permitam o reconhecimento e a transcrição de linguagem falada de maneira automático

Permissões

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

Layout

```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:layout_editor_absoluteX="136dp"
    tools:layout_editor_absoluteY="91dp">

    <TextView
        android:id="@+id/lblTextoReconhecido"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:gravity="center"
        android:hint=""
    />
</LinearLayout>
```

Oncreate e atributos

```
class MainActivity : AppCompatActivity() {  
    private var codRetorno: Int = 10  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        setContentView(R.layout.activity_main)  
  
        if (ContextCompat.checkSelfPermission(this, Manifest.permission.RECORD_AUDIO) != PackageManager.PERMISSION_GRANTED) {  
            val RECORD_AUDIO_REQUEST_CODE = 1  
            ActivityCompat.requestPermissions(this, arrayOf(Manifest.permission.RECORD_AUDIO), RECORD_AUDIO_REQUEST_CODE)  
        }else{  
            getSpeechInput()  
        }  
  
    }  
}
```

onRequestPermissionsResult

```
override fun onRequestPermissionsResult(requestCode: Int, permissions: Array<String>, grantResults: IntArray) {  
    super.onRequestPermissionsResult(requestCode, permissions, grantResults)  
  
    if ((grantResults.isNotEmpty() && grantResults[0] == PackageManager.PERMISSION_GRANTED)) {  
        getSpeechInput()  
    } else {  
        Toast.makeText(this, "Não suportado", Toast.LENGTH_SHORT).show()  
    }  
}
```

getSpeechInput

```
private fun getSpeechInput()
{
    val intent = Intent(RecognizerIntent.ACTION_RECOGNIZE_SPEECH)
    intent.putExtra(RecognizerIntent.EXTRA_LANGUAGE_MODEL, RecognizerIntent.LANGUAGE_MODEL_FREE_FORM)
    intent.putExtra(RecognizerIntent.EXTRA_LANGUAGE, Locale.getDefault())
    if (intent.resolveActivity(packageManager) != null){
        startActivityForResult(intent, codRetorno)
    } else{
        Toast.makeText(this, "Não suportado", Toast.LENGTH_SHORT).show()
    }
}
```

onActivityResult

```
override fun onActivityResult(requestCode: Int, resultCode: Int, data: Intent?) {
    super.onActivityResult(requestCode, resultCode, data)
    if (resultCode == RESULT_OK && data != null) {
        if (codRetorno == 10) {
            val result = data.getStringArrayListExtra(RecognizerIntent.EXTRA_RESULTS)
            val lblTextoReconhecido: TextView = findViewById(R.id.lblTextoReconhecido)
            lblTextoReconhecido.text = result[0]
        }
    }
}
```