



Prof. Me. Hélio Esperidião

Speech Recognition – Kotlin

Reconhecimento de fala - Kotlin

# 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.
- Programas que reconhecem fala são baseados em processamento digital de áudio. Uma área das ciências da computação que envolve métodos matemáticos elaborados como Cálculo superior e Transformadas.
- A google disponibiliza uma biblioteca pronta para processamento e reconhecimento de fala.

# Interface do usuário

```
<LinearLayout
  android:layout_height="match_parent"
  android:layout_width="match_parent"
  android:orientation="vertical">
  <Button
    android:id="@+id/button"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Falar" />

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

# imports

```
import android.Manifest
import android.content.Intent
import android.content.pm.PackageManager
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.speech.RecognizerIntent
import android.widget.Button
import android.widget.TextView
import android.widget.Toast
import androidx.core.app.ActivityCompat
import java.util.Locale
```

# onCreate()

```
override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    setContentView(R.layout.activity_main)
    //Carrega a interface pedindo ao usuário permissão
    permissoesUsuario();

    //instancia do botão
    var capturarAudio: Button = findViewById(R.id.button)

    capturarAudio!!.setOnClickListener {

        capturarAudioMicrofone()
    }
}
```

# permissoesUsuario()

```
fun permissoesUsuario(){  
    val codigoidentificacao = 201;  
    ActivityCompat.requestPermissions(  
        this,  
        arrayOf<String>(Manifest.permission.RECORD_AUDIO,  
            Manifest.permission.INTERNET  
        ),  
        codigoidentificacao  
    )  
}
```

# onRequestPermissionsResult()

```
override fun onRequestPermissionsResult(codigoidentificacao: Int, permissions: Array<out String>, grantResults: IntArray) {
    super.onRequestPermissionsResult(codigoidentificacao, permissions, grantResults)

    if(codigoidentificacao==201){
        if (ActivityCompat.checkSelfPermission(this,
            Manifest.permission.RECORD_AUDIO
        ) == PackageManager.PERMISSION_GRANTED){
            if(ActivityCompat.checkSelfPermission(this,
                Manifest.permission.INTERNET
            ) == PackageManager.PERMISSION_GRANTED){

            }
        }
    }
}
```

# capturarAudioMicrofone()

```
private fun capturarAudioMicrofone(){
    //cria um intente com a interface padrão do android para capturar audio.
    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())
    //Verifica se o sistema operacional oferece o intent de captura de audio
    if (intent.resolveActivity(packageManager) != null){
        startActivityForResult(intent, 202)
    } else{
        Toast.makeText(this,"Não suportado", Toast.LENGTH_SHORT).show()
    }
}
```



# onActivityResult()

```
//é executado automaticamente após o fechamento da interface de recuperar audio.  
override fun onActivityResult(requestCode: Int, resultCode: Int, data: Intent?) {  
    super.onActivityResult(requestCode, resultCode, data)  
  
    if (resultCode == RESULT_OK && data != null) {  
        val result = data.getStringArrayListExtra(RecognizerIntent.EXTRA_RESULTS)  
        val lblTextoReconhecido: TextView = findViewById(R.id.lblTextoReconhecido)  
        lblTextoReconhecido.text = result?.get(0) ?: ""  
    }  
}
```