



# ***Desarrollo de aplicación de reconocimiento de voz cognitiva***

Base de datos multimedia - sonido

Trabajo práctico de Gestión de sistemas de información

Diego Molero Marín  
Carlos Sobrino Pérez

# Índice

<b>Introducción</b>	<b>3</b>
<b>Desarrollo</b>	<b>3</b>
1º Paso:	3
2º Paso:	4
3º Paso:	4
<b>Explicación del Wit.ia</b>	<b>6</b>

# Introducción

Hoy en día ya son muchas las aplicaciones que permiten al usuario poder controlar su entorno mediante la voz. Tras investigar cómo podríamos desarrollar nuestra propia aplicación descubrimos que existen una gran cantidad de empresas que ofrecen SDKs y herramientas para poder realizar el reconocimiento de voz de forma cognitiva. Estas herramientas reciben de entrada el texto en plano. Para poder obtener el texto en plano android ofrece el servicio de google speech-to-text el cual permite obtener texto a través de la voz.

## Desarrollo

### 1º Paso:

Obtención del texto a través de la voz. Como ya se ha mencionado usamos el servicio de google nativo de android speech-to-text para ello usamos este código.

```
public void onClickImgBtnHablar(View v) {
    Intent intentActionRecognizeSpeech = new Intent(
        RecognizerIntent.ACTION_RECOGNIZE_SPEECH);

    intentActionRecognizeSpeech.putExtra(
        RecognizerIntent.EXTRA_LANGUAGE_MODEL, "es-ES");
    try {
        startActivityForResult(intentActionRecognizeSpeech,
            RECOGNIZE_SPEECH_ACTIVITY);
    } catch (ActivityNotFoundException a) {
        Toast.makeText(getApplicationContext(),
            "Tú dispositivo no soporta el reconocimiento por voz",
            Toast.LENGTH_SHORT).show();
    }
}
```

## 2º Paso:

Para procesar el texto de salida del paso uno, es decir el reconocimiento de voz cognitivo, nos decantamos por el uso de Wit.ia, esta es una herramienta gratuita y que ofrece una SDK para Android desarrollada por la comunidad que permite una comunicación directa con los servidores y fácil de usar. Así es como se refleja el uso de Wit en nuestro código tras haber importado la librería:

Creamos el objeto del tipo Wit con la clave de acceso y la clase Main del proyecto que implementa de la clase IWitListener.

```
String accessToken = "DPWQHQRACLK3QSMCWH6L4FWNXELQPJRP";  
_wit = new Wit(accessToken, this);
```

Con esta función enviamos el texto que recibimos del google speech-to-text

```
_wit.captureTextIntent(strSpeech2Text);
```

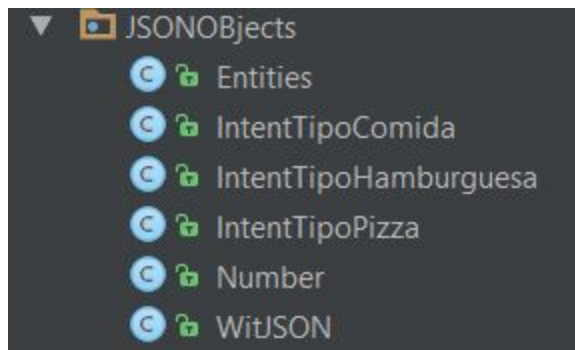
Cuando el servidor responde, el programa llama al método cuya entrada es un mensaje estructurado en JSON.

```
@Override  
public void witDidGraspIntent(ArrayList<WitOutcome> arrayList, String s, Error error) {  
  
    Gson gson = new GsonBuilder().setPrettyPrinting().create();  
    String jsonOutput = gson.toJson(arrayList);  
    Log.i("JSON OUTPUT ->", jsonOutput);  
    ((TextView) findViewById(R.id.txtText)).setText("Done!");  
    WitJSON[] response = FromJSON(jsonOutput);  
    Comida comida = analyzeData(response);  
    if (comida != null) {  
        pedido.addComida(comida);  
        updateList();  
    }  
}
```

## 3º Paso:

Para poder manejar los mensajes de entrada que recibimos del servidor de Wit usamos una librería llamada Gson la cual permite de forma rápida transformando el objeto JSon en un objeto Java.

Para poder hacer esto debemos crear las clases en función de cómo esté estructurado el JSON y la librería Gson se encarga de crear el objeto solo.  
Esta son las clases que tenemos:



Cada clase se compone de los atributos que estructuran la clase JSON.

Este es un ejemplo de una respuesta del servidor a la frase “hola amigos quiero tres hamburguesas de pollo”.

```
{
  "msg_id": "07kO3NYm1aR8A1oVT",
  "_text": "hola amigos quiero tres hamburguesas de pollo",
  "entities": {
    "number": [{
      "confidence": 1,
      "value": 3,
      "type": "value"
    }],
    "intentTipoComida": [{
      "confidence": 1,
      "value": "hamburguesa",
      "type": "value"
    }],
    "intentTipoHamburguesa": [{
      "confidence": 1,
      "value": "Pollo",
      "type": "value"
    }]
  }
}
```

# Explicación del Wit.ai

Consiste en ofrecer el **SDK de forma gratuita**, disponible para todos aquellos desarrolladores que deseen utilizando.

La herramienta funciona de la siguiente manera:

Los comandos de voz del usuario son enviadas a los servidores de la compañía, quien lo procesa y devuelve un resultado.

[DiegoMolero](#) / [VoiceRecog](#) / ...

Nosotros hemos creado un Proyecto,

Para configurar dicho proyecto realizamos las siguientes acciones:

1. Primero creamos las entidades:

Las entidades se colocan las palabras claves que quieres en una frases y el valor que tienen que devolver dichas palabras.

Las entidades que hemos creados son las siguientes:

Your app uses 4 entities

Name	Search Strategy ?
<b>intentTipoComida</b> → User-defined entity	<input type="radio"/> trait <input checked="" type="radio"/> free-text <input type="radio"/> keywords
<b>intentTipoPizza</b> → User-defined entity	<input type="radio"/> trait <input checked="" type="radio"/> free-text <input type="radio"/> keywords
<b>intentTipoHamburguesa</b> → User-defined entity	<input checked="" type="radio"/> trait <input type="radio"/> free-text <input type="radio"/> keywords
<b>wit/number</b> → Captures number either in written form or numeral, like 'six' or '6'	—

- **Tipo de comida**

Se compone de dos tipos que son Pizzas y Hamburguesas:

## Keywords

Keyword ?	Synonyms ?
hamburguesa	<div>Hamburguesas</div> <div>+ Add synonym...</div>
pizza	<div>pizzas</div> <div>+ Add synonym...</div>

Los valores devueltos a nuestra aplicación es el elemento en **Keyword**. Los sinónimos son de las maneras que también puede llamarse las palabras claves.

Expressions Filter by: all values

“ Search through your expressions.
Text
“ dame tres hamburguesas de ternera x
“ dame tres hamburguesas de pollo x
“ quiero una pizza barbacoa x
“ Quiero tres pizzas x

Estas con unas cuantas expresiones usadas en el entidad de Tipo de comida, es decir, que tienen relación con alguna palabra clave de dicha entidad.

#### - Tipo de Pizza.

Esta entidad se componen de los tipos de pizzas que va a tener nuestra aplicación. Qué son las que muestro en la imagen siguiente:

Keywords

Keyword ?	Synonyms ?
Bacon	bacon + Add synonym...
Barbacoa	barbacoa + Add synonym...
Margarita	+ Add synonym...
Vegetal	Vegetales vegetales + Add synonym...
Cuatro quesos	4 quesos cuatro quesos + Add synonym...

Como vemos se compone de 5 valores que son: Bacon, Barbacoa, Margarita, Vegetal y Cuatro quesos.

Un ejemplo de una expresión relacionada con esta entidad es:

Expressions Filter by: all values

“ Search through your expressions.
Text
“ quiero una pizza barbacoa x

- **Tipo de hamburguesas.**

Estás entidad está compuesta por dos valores: **pollo y ternera**.

En la siguiente imagen se muestran los valores:

Keywords

Keyword ?	Synonyms ?
Pollo	pollo + Add synonym...
Ternera	carne vacuno + Add synonym...

Como vemos los valores son Keywords, con sinónimos en Ternera como vacuno y carne.

Unos ejemplos de expresión donde interviene esta entidad son:

Expressions

Filter by: all values

“ Search through your expressions.
Text
“ dame tres hamburguesas de ternera x
“ dame tres hamburguesas de pollo x

- **Wit/Number**

Esta entidad es el que detecta los números tanto escritos con letras o como número.

wit/number



Captures number either in written form or numeral, like `six` or `6`

Y ahora muestro una imagen con unas cuantas expresiones donde aparecen elementos de esta entidad.

Expressions

“ Search through your expressions.
Text
“ dame tres hamburguesas de ternera x
“ dame tres hamburguesas de pollo x
“ quiero una pizza barbacoa x
“ Quiero tres pizzas x
“ Quiero una pizza barbacoa x
“ Quiero tres hamburguesas de pollo x
“ Quiero una hamburguesa de carne x



Un ejemplo de cómo funciona esta herramienta es así:

### Test how your app understands a sentence

You can train your app by adding more examples

quiero **siete** hamburguesas de pollo

<input type="radio"/> intentTipoComida	<input type="text" value="hamburguesa"/>
<input type="radio"/> intentTipoHamburguesa	<input type="text" value="Pollo"/>
<input type="radio"/> wit/number	7
<a href="#">Add a new entity</a>	

Nosotros le hemos introducido la frase “*quiero siete hamburguesas de pollo*” y nos ha devuelto los siguientes valores de siete que sale remarcado el valor **7**, hamburguesas nos devuelve ***hamburguesa***, y pollo el valor que nos devuelve es ***Pollo***.

## Bibliografía

-<https://wit.ai/DiegoMolero/VoiceRecog> - *Lugar de desarrollo de los datos de Wit ia para el reconocimiento de voz*

-<https://github.com/DiegoMolero/VoiceRecog> - *Repositorio de la aplicación y toda la documentación necesaria*

-<https://github.com/wit-ai/wit-android-sdk> - *Repositorio de la apk para la conectividad con Wit ia*