



# COMPARACIÓN **APLICACIONES TTS**

## ANDROID IOS



# INDICE

**1-REQUISITOS DEL PROTOTIPO A IMPLEMENTAR**

**2-CRITERIOS DE COMPARACION EN LA IMPLEMENTACION**

**3-PROYECTO IMPLEMENTACIÓN ANDROID**

**4-PROYECTO IMPLEMENTACION IOS**

**5-COMPARACIÓN DE LAS DOS IMPLEMENTACIONES**

**6-COMPARACIÓN DE LAS TECNOLOGÍAS**

**7-CONCLUSIONES**



# 1. REQUISITOS DEL PROTOTIPO



## REQUISITOS FUNCIONALES



RF01 Introducir texto

RF02 Seleccionar idioma (Español , Inglés , Francés, Alemán)

RF03 Botón de escucha para reproducir el habla

## OTROS REQUISITOS



R01 Velocidad del habla.

R02 verificación del texto introducido.

R03 Implementación de interfaz.

# 2.CRITERIOS DE COMPARACION



## HORAS INVERTIDAS EN LA CREACIÓN

### Descripción

Horas que se han tardado en implementar la app

### Tipo

N Numérico (horas)

## CLASES UTILIZADAS EN LA APP

### Descripción

Número de clases utilizadas en la creación de la app

### Tipo

N Numérico

## MULTIPLATAFORMA

### Descripción

Si la aplicación es válida en varias plataformas

### Tipo

Booleano (si o no)

## INTRODUCCIÓN VARIOS LENGUAJES

### Descripción

Si se pueden utilizar o introducir nuevos lenguajes

### Tipo

Booleano (si o no)

## FACILIDAD DE IMPLEMENTACIÓN

### Descripción

Facilidad con la que se ha realizado la implementación

### Tipo

N Numérico (1-10)



# 2.CRITERIOS DE COMPARACION



## VELOCIDAD DE USO

### Descripción

Cuanto tarda en reproducir el texto

### Tipo

Numerico (1-10)

## NATURALIDAD DEL LENGUAJE

### Descripción

Naturalidad con la que la voz sintética reproduce el texto

### Tipo

Numerico (1-10)

## INFORMACIÓN PARA IMPLEMENTAR LA APP

### Descripción

Información ofrecida en internet como ayudas o guías para la implementación

### Tipo

Numerico (1-10)

## SENCILLEZ INTERFAZ

### Descripción

Si es fácil entender y usar la interfaz

### Tipo

Numerico (1-10)

## COSTE DE IMPLEMENTACIÓN

### Descripción

Coste final de la implementación de la app

### Tipo

Numerico (euros)

# 3 PROYECTO IMPLEMENTACIÓN ANDROID

## 3.1 Documentación diseño



# 3 PROYECTO IMPLEMENTACIÓN ANDROID



## 3.2 Documentación construcción

```
lista.setOnItemClickListener(new AdapterView.OnItemClickListener() {  
    @Override  
    //DEPENDIENDO DE LA SELECCION LA LISTA TOMARA LA POSICION PARA CONSEGUIR EL VALOR  
    public void onItemClick(AdapterView<?> parent, View view, int i, long id) {  
        switch (i){  
            case 0:  
                valor = idiomas[i];  
                tl.setLanguage(Locale.getDefault());  
                break;  
            case 1:  
                valor = idiomas[i];  
                tl.setLanguage(Locale.ENGLISH);  
                break;  
            case 2:  
                valor = idiomas[i];  
                tl.setLanguage(Locale.FRANCE);  
                break;  
            case 3:  
                valor = idiomas[i];  
                tl.setLanguage(Locale.GERMAN);  
                break;  
        }  
    }  
}
```

```
<!-- BOTON PARA ESCUCHAR -->  
<Button  
    android:id="@+id/button"  
    style="@style/Widget.AppCompat.Button.Colored"  
    android:layout_width="80dp"  
    android:layout_height="80dp"  
    android:layout_marginBottom="27dp"  
    android:layout_marginEnd="55dp"  
    android:layout_marginRight="55dp"  
    android:autoText="false"  
    android:background="@drawable/boton"  
    android:elevation="0dp"  
    android:fontFamily="casual"  
    android:layout_alignParentBottom="true"  
    android:layout_alignRight="@+id/lista1"  
    android:layout_alignEnd="@+id/lista1" />
```

# 3 PROYECTO IMPLEMENTACIÓN ANDROID

## 3.3 Documentación de pruebas

Intentar que la aplicación, pronunciase al español. Nos dimos cuenta que cuando intentábamos establecer el idioma como español no reconocía las siglas.

```
public void onInit(int status) {  
    if(status != TextToSpeech.ERROR) {  
        tl.setLanguage(Locale.getDefault());  
    }  
}
```

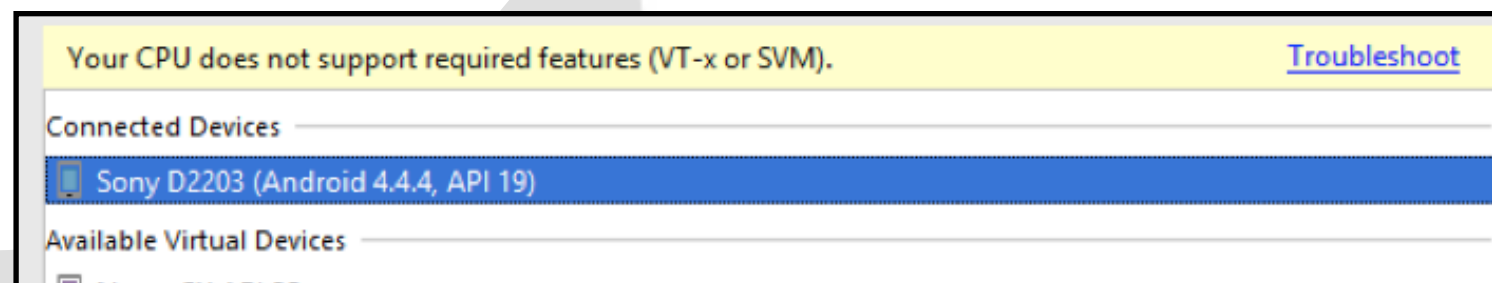
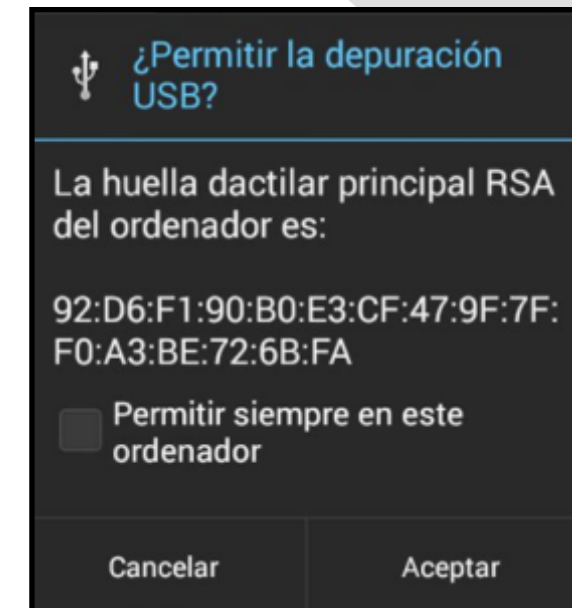
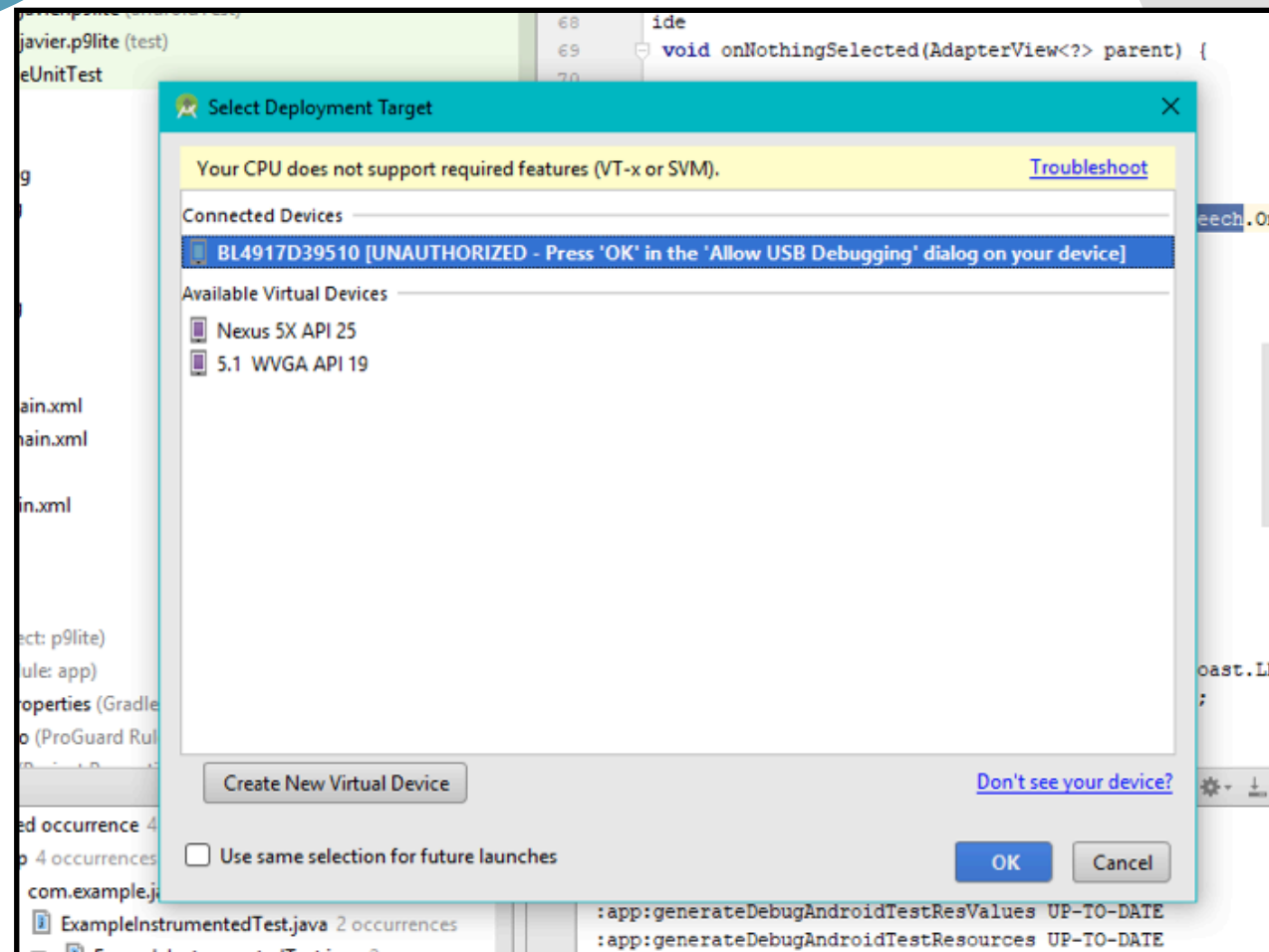
Establecer distintos idiomas. Nuestro pensamiento era hacerlo de manera dinámica. No obstante, esto no nos fue posible.

```
public void onItemSelected(AdapterView<?> parent, View view, int i, long id) {  
    switch (i) {  
        case 0:  
            tl.setLanguage(Locale.getDefault());  
            break;  
        case 1:  
            tl.setLanguage(Locale.ENGLISH);  
            break;  
        case 2:  
            tl.setLanguage(Locale.FRANCE);  
            break;  
        case 3:  
            tl.setLanguage(Locale.GERMAN);  
            break;  
    }  
}
```



# 3 PROYECTO IMPLEMENTACIÓN ANDROID

## 3.4 Documentación de instalación



# 3 PROYECTO IMPLEMENTACIÓN ANDROID

## 3.5 Manual de usuario

La App es simple, el usuario la abre y le aparecerá una interfaz donde podrá escribir, seleccionar el idioma y escuchar.

En la selección de idiomas podrá elegir entre 4 diferentes: Español, Inglés, Francés y Alemán y por defecto la aplicación viene con el español seleccionado.

Una vez se haya escrito algo y seleccionado el idioma que se quiera, se pulsa el botón con el icono del megáfono y reproducirá lo que este escrito



# 4 PROYECTO IMPLEMENTACIÓN IOS

## 4.1 Documentación diseño



Pantalla inicial



Pantalla carga



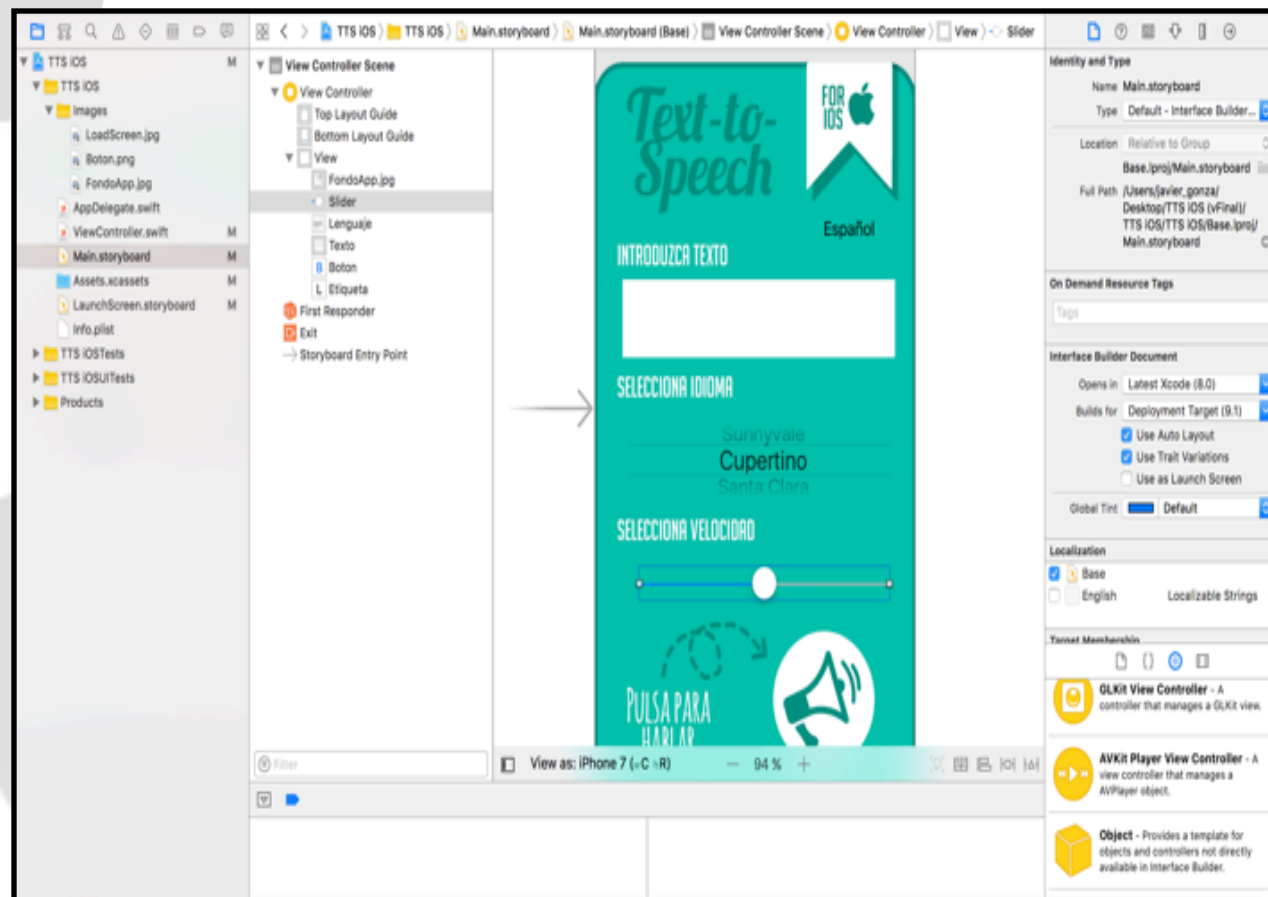
Logo app

# 4 PROYECTO IMPLEMENTACIÓN IOS

View Controller.swift,  
Ej. Declaración de elementos

## 4.2 Documentación construcción

Main.storyboard (Interfaz grafica)



```
@IBOutlet var Etiqueta: UILabel!  
//Etiqueta donde se almacena la selección del Picker  
  
@IBOutlet var Texto: UITextView!  
//Texto donde el usuario escribe  
  
@IBOutlet var Lenguaje: UIPickerView!  
var array_lenguajes = ["Español", "Inglés", "Francés", "Alemán"]  
//Picker que almacena los idiomas y array de idiomas que se muestran  
  
@IBOutlet var Slider: UISlider!  
//Slider de selección de velocidad  
  
@IBOutlet var Boton: UIButton!  
//Boton de Hablar.
```

Ej. Código Boton Actuación Hablar

```
@IBAction func hablar(_ sender: Any) {  
    //Acción pulsar boton - Hablar  
  
    let texto = AVSpeechUtterance(string: Texto.text!)  
    //Cogemos texto del cuadro de texto  
  
    let idioma = AVSpeechUtterance(string: Etiqueta.text!)  
    //Cogemos de la etiqueta el idioma seleccionado  
  
    if (idioma.speechString == "Español"){  
        texto.voice = AVSpeechSynthesisVoice (language: "es-ES")  
    }  
    if (idioma.speechString == "Inglés"){  
        texto.voice = AVSpeechSynthesisVoice (language: "en-US")  
    }  
    if (idioma.speechString == "Francés"){  
        texto.voice = AVSpeechSynthesisVoice (language: "fr-FR")  
    }  
    if (idioma.speechString == "Alemán"){  
        texto.voice = AVSpeechSynthesisVoice (language: "de-DE")  
    }  
}
```



# 4 PROYECTO IMPLEMENTACIÓN IOS



## 4.3 Documentación de pruebas

Prueba1 - Manejo del teclado de la aplicación.

Problema: imposibilidad pulsar botón hablar por teclado

Solución: código para manejar la visibilidad del teclado

```
let tap: UITapGestureRecognizer = UITapGestureRecognizer(target: self, action: #selector(ViewController.dismissKeyboard))

view.addGestureRecognizer(tap)
//Ver y almacenar si en algun momento se toca la pantalla para ocultar el teclado

//Funcion que oculta el teclado y nos muestra la vista
func dismissKeyboard() {
    view.endEditing(true)
}
```

Prueba2 - Selección escogida del Picker.

Problema: : imposibilidad de sacar la selección del Picker al pulsar el botón “Hablar”.

Solución: creación de etiqueta donde se almacena el idioma seleccionado.

```
func pickerView(_ pickerView: UIPickerView, didSelectRow row: Int, inComponent component: Int){
    Etiqueta.text = array_lenguajes [row]
}
//Funcion que almacena en una etiqueta el lenguaje seleccionado
```

```
Etiqueta.isHidden = true //Oculta la etiqueta donde se escribe el texto del picker
//para posteriormente almacenarlo como variable.
```

# 4 PROYECTO IMPLEMENTACIÓN IOS



## 4.4 Documentación de instalación

### Entorno Apple Xcode:

Instalación gratuita desde la web (o Mac AppStore), desde un Mac.

### App:

iPhones con Certificado de Desarrollador, a través Xcode. En un futuro y si se quisiera, Apple Store.

# 4 PROYECTO IMPLEMENTACIÓN IOS

## 4.5 Manual de usuario

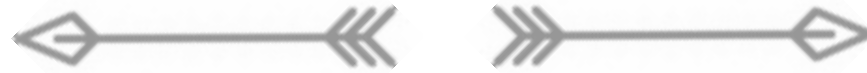
App simple: se abre, se escribe un texto, se selecciona el idioma con el que se quiera que sea dicho, se escoge una velocidad y finalmente se pulsa el botón de “Hablar”.  
(en cualquier orden).

Idiomas disponibles: Español (por defecto), Inglés, Francés y Alemán.

Velocidad de idioma: cualquiera entre los valores 0 y 1 (0.5 por defecto)



# 5 COMPARACIÓN IMPLEMENTACIONES



## ANDROID

**HORAS INVERTIDAS EN  
LA CREACIÓN**

**8**

**VELOCIDAD DE USO**

**7**

**CLASES UTILIZADAS  
EN LA APP**

**6**

**NATURALIDAD DEL  
LENGUAJE**

**7**

**MULTIPLATAFORMA**

**NO**

**INFORMACIÓN PARA  
IMPLEMENTAR LA APP**

**10**

**INTRODUCCIÓN  
VARIOS LENGUAJES**

**SI**

**SENCILLEZ INTERFAZ**

**9**

**FACILIDAD DE  
IMPLEMENTACIÓN**

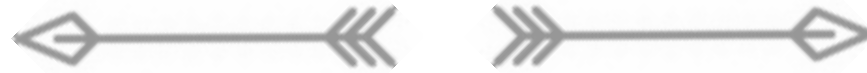
**9**

**COSTE DE  
IMPLEMENTACIÓN**

**0**



# 5 COMPARACIÓN IMPLEMENTACIONES



**IOS**

**HORAS INVERTIDAS EN  
LA CREACIÓN**

**9**

**VELOCIDAD DE USO**

**8**

**CLASES UTILIZADAS  
EN LA APP**

**7**

**NATURALIDAD DEL  
LENGUAJE**

**9**

**MULTIPLATAFORMA**

**NO**

**INFORMACIÓN PARA  
IMPLEMENTAR LA APP**

**8**

**INTRODUCCIÓN  
VARIOS LENGUAJES**

**SI**

**SENCILLEZ INTERFAZ**

**9**

**FACILIDAD DE  
IMPLEMENTACIÓN**

**7**

**COSTE DE  
IMPLEMENTACIÓN**

**0**

# 6 COMPARACIÓN DE LAS TECNOLOGÍAS

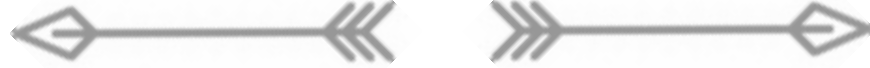
	ANDROID	IOS
<b>Horas invertidas en la creación</b>	<b>8</b>	<b>9</b>
<b>Clases utilizadas en la app</b>	<b>6</b>	<b>7</b>
<b>Multiplataforma</b>	<b>NO</b>	<b>NO</b>
<b>Introducción de varios lenguajes</b>	<b>SI</b>	<b>SI</b>
<b>Facilidad de implementación</b>	<b>9</b>	<b>7</b>

# 6 COMPARACIÓN DE LAS TECNOLOGÍAS

	ANDROID	IOS
Velocidad de uso	7	8
Naturalidad del lenguaje	7	9
Información para implementar la app	10	8
Sencillez interfaz	9	9
Coste de implementación	0	0



# 7 CONCLUSIONES



**ANDROID**



**IOS**







**¡GRACIAS!**

**¿ALGUNA  
PREGUNTA?**