

Contenido

1. Autores del trabajo, planificación y entrega.....	2
1.1 Autores.....	2
1.2 Planificación	2
1.3 Entrega	2
2. Requisitos del prototipo a implementar	3
2.1 Requisitos funcionales.....	3
2.2 Otros requisitos	3
3. Criterios de comparación en la implementación	4
3.1 Criterio 1: Horas invertidas en la creación de la aplicación	4
3.2 Criterio 1: Clases utilizadas en la aplicación.....	4
3.3 Criterio 1: Multiplataforma	4
3.4 Criterio 1: Introducción de varios lenguajes	4
3.5 Criterio 1: Facilidad de implementación	4
3.6 Criterio 1: Velocidad de uso	4
3.7 Criterio 1: Naturalidad del lenguaje	5
3.8 Criterio 1: Información para implementar la aplicación	5
3.9 Criterio 1: Sencillez interfaz	5
3.10 Criterio 1: Coste de implementación	5
4. Proyecto de implementación de un prototipo del sistema utilizando la tecnología Android	6
4.1 Documentación de diseño	6
4.2 Documentación de construcción.....	7
4.3 Documentación de pruebas.....	7
4.4 Documentación de instalación.....	8
4.5 Manual de usuario.....	9
5. Proyecto de implementación de un prototipo del sistema utilizando la tecnología iOS	10
5.1 Documentación de diseño	10
5.2 Documentación de construcción.....	10
5.3 Documentación de pruebas.....	12
5.4 Documentación de instalación.....	13
5.5 Manual de usuario.....	13
6. Comparación de las dos implementaciones	14
7. Comparación de la implementación de las tecnologías.....	16
8. Conclusiones	18

1. Autores del trabajo, planificación y entrega

1.1 Autores

Somos el grupo 3 del turno de tarde, formado por:

Sandra Félix Blázquez

Javier González Iglesias

Javier Rodríguez Merchante

1.2 Planificación

La planificación es la siguiente:

<https://app.ganttpro.com/shared/token/acd1bf8a6b8e4beaf04b1e28cd521c037ce3a189bba1f53ab68fa8d9d03c6bde/190866>

1.3 Entrega

El repositorio del TG3 en GitHub es el siguiente:

<https://github.com/Javierrodriguezmerchante/TG3>

2. Requisitos del prototipo a implementar

2.1 Requisitos funcionales

REQ.	DESCRIPCIÓN
RF01	Introducir texto
RF02	Seleccionar idioma (español, inglés, francés, alemán)
RF03	Botón de escucha para reproducir el habla

2.2 Otros requisitos

REQ.	DESCRIPCIÓN
R01	Velocidad del habla.
R02	Verificación del texto introducido.
R03	Implementación de interfaz.

3. Criterios de comparación en la implementación

3.1 Criterio 1: Horas invertidas en la creación de la aplicación

Nombre del criterio: Horas invertidas en la creación de la aplicación

Descripción: Horas que se tarda en crear la aplicación, programando y la interfaz

Tipo de valor: Numérico (horas)

3.2 Criterio 2: Clases utilizadas en la aplicación

Nombre del criterio: Clases utilizadas en la aplicación

Descripción: Clases necesarias para la realización de la aplicación final, incluyendo el Text-to-speech y la interfaz gráfica

Tipo de valor: Numérico

3.3 Criterio 3: Multiplataforma

Nombre del criterio: Multiplataforma

Descripción: Si la aplicación es apta para varias plataformas

Tipo de valor: Booleano (si o no)

3.4 Criterio 4: Introducción de varios lenguajes

Nombre del criterio: Introducción de varios lenguajes

Descripción: Si se pueden introducir varios lenguajes, o solo está creada para uno

Tipo de valor: Booleano (si o no)

3.5 Criterio 5: Facilidad de implementación

Nombre del criterio: Facilidad de implementación

Descripción: Facilidad para implementar la aplicación TEXT-TO-SPEECH

Tipo de valor: Numérico (1-10)

3.6 Criterio 6: Velocidad de uso

Nombre del criterio: Velocidad de uso

Descripción: Velocidad con la que funciona la aplicación desde que se introduce el texto hasta que se reproduce

Tipo de valor: Numérico (1-10)

3.7 Criterio 7: Naturalidad del lenguaje

Nombre del criterio: Naturalidad del lenguaje

Descripción: Naturalidad con la que el lenguaje habla si se parece a una voz real o es muy robótica

Tipo de valor: Numérico (1-10)

3.8 Criterio 8: Información para implementar la aplicación

Nombre del criterio: Información para implementar la aplicación

Descripción: Si se ha encontrado información suficiente en internet, manuales o guías para proceder a implementar la misma

Tipo de valor: Numérico (1-10)

3.9 Criterio 9: Sencillez interfaz

Nombre del criterio: Sencillez interfaz

Descripción: Si la interfaz es sencilla y se puede usar fácilmente

Tipo de valor: Numérico (1-10)

3.10 Criterio 10: Coste de implementación

Nombre del criterio: Coste de implementación

Descripción: El coste que ha llevado implementar la aplicación (programas para implementarla, manuales, códigos)

Tipo de valor: Numérico (euros)

4. Proyecto de implementación de un prototipo del sistema utilizando la tecnología Android

4.1 Documentación de diseño

El primer diseño fue el presentado durante la revisión del avance del prototipo con el profesor. Esta interfaz era simple con una de las funcionalidades anterior mente descritas. Dicha interfaz era un modelo propio de Android estudio con un encabezado personalizable

El segundo diseño, está un poco más elaborado, in la inclusión de varias imágenes, suprimiendo la interfaz proporcionada por Android y la eliminación del encabezado incorporando una imagen personalizada como encabezado. También se cambió el tipo de letra, la forma y el color del botón, y por último la incorporación de otra funcionalidad más como es la selección de idioma

Finalmente se creó un último diseño, proporcionado por Javier González para la obtención de una interfaz más vistosa y llamativa. Las interfaces anteriores fueron eliminadas incorporando los elementos proporcionados por mi compañero cambiando así el encabezado y el botón al igual que el fondo de la aplicación



4.2 Documentación de construcción

El proyecto se ha realizado en java por medio de la utilización de Android studio. La realización del proyecto se basa principalmente dos archivos.

El primero es el archivo MainActivity.java en el cual se encuentran todos los métodos y clases utilizados, que dan funcionalidad a cada uno de los elementos que se encuentran en la interfaz. En concreto este fragmento se asocia con el spinner o lista desplegable

```
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;
        }
    }
});
```

Y el segundo archivo es el activity_main.xml, el cual es donde se almacenan todos los contenedores, o elementos que contiene la interfaz, así como cada una de sus características asociadas, de nuestra aplicación.

```
<!-- 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" />
```

4.3 Documentación de pruebas

Los casos de prueba, se han basado principalmente en escribir en la aplicación y que esta nos lo emitiese en forma de voz, en diferentes lenguajes.

Durante el desarrollo han ocurrido una serie de problemas:

- El primero surgió cuando se realizó la primera versión, a la hora de intentar que la aplicación, pronunciase al español. Nos dimos cuenta que cuando intentábamos establecer el idioma como español no reconocía las siglas, que más tarde vimos que el idioma no estaba implementado en Android studio. Para solucionar este problema se puede obtener directamente por medio del uso del método `getDefault` cogiendo por defecto el idioma del terminal

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

- El siguiente problema surgió a la hora de establecer distintos idiomas. Nuestro pensamiento era hacerlo de manera dinámica. No obstante, esto no nos fue posible. Inicialmente creamos un array con las diferentes idiomas y un bucle for que recorriese el array, así de esta forma compararía el valor seleccionado con el valor del array y si eran iguales entraría dentro de un if y hablaría en el idioma seleccionado. Pero el problema surgió cuando le intentamos especificar el lenguaje, el cual no reconocía la expresión `tl.setLanguage(Locale.valor)`; Puesto que no reconocía el valor de la variable `valor`. Esto se solucionó por

medio del uso de la herramienta case y dependiendo del valor que tomase de la posición de array asignarle directamente un idioma en concreto

```
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;  
    }  
}
```

4.4 Documentación de instalación

Descripción suficiente para que una persona que no ha participado en el proyecto pueda instalar el prototipo.

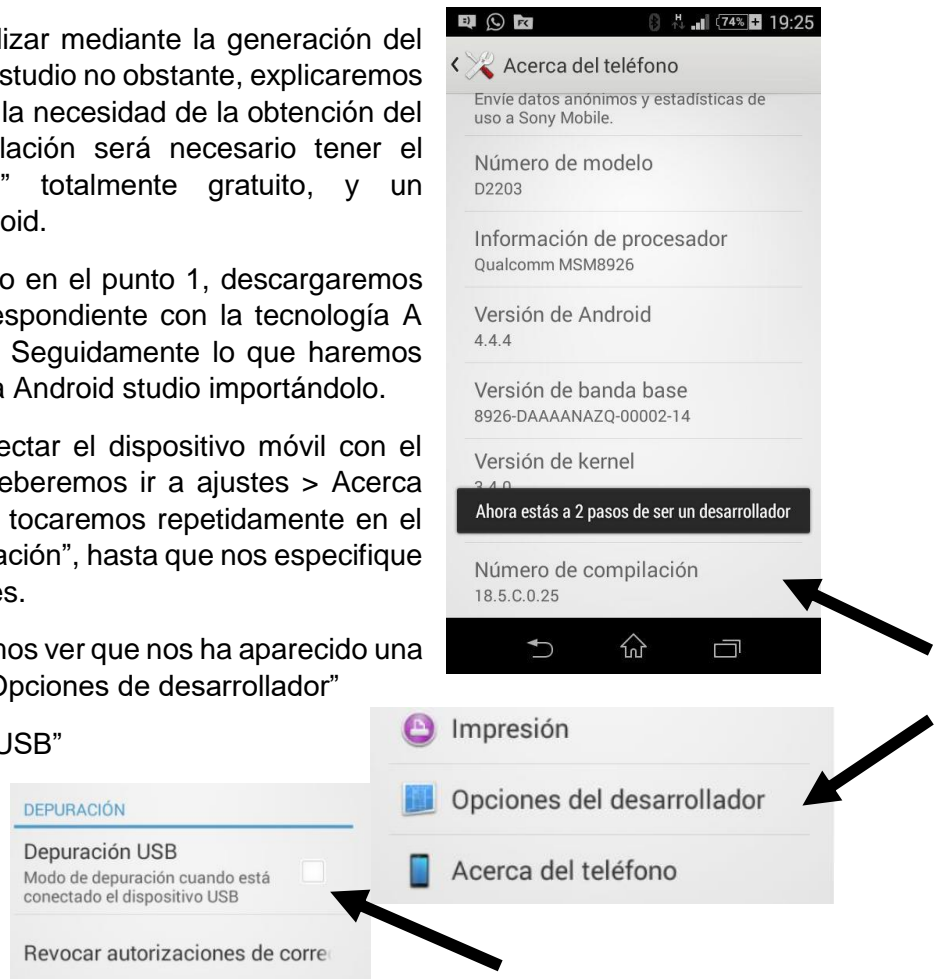
La instalación se puede realizar mediante la generación del .apk desde el propio Android studio no obstante, explicaremos otra forma de instalación sin la necesidad de la obtención del .apk. En la siguiente instalación será necesario tener el programa “Android studio” totalmente gratuito, y un Smartphone con el S.O Android.

En el repositorio especificado en el punto 1, descargaremos el código del prototipo correspondiente con la tecnología A que corresponde a Android. Seguidamente lo que haremos será incorporar este código a Android studio importándolo.

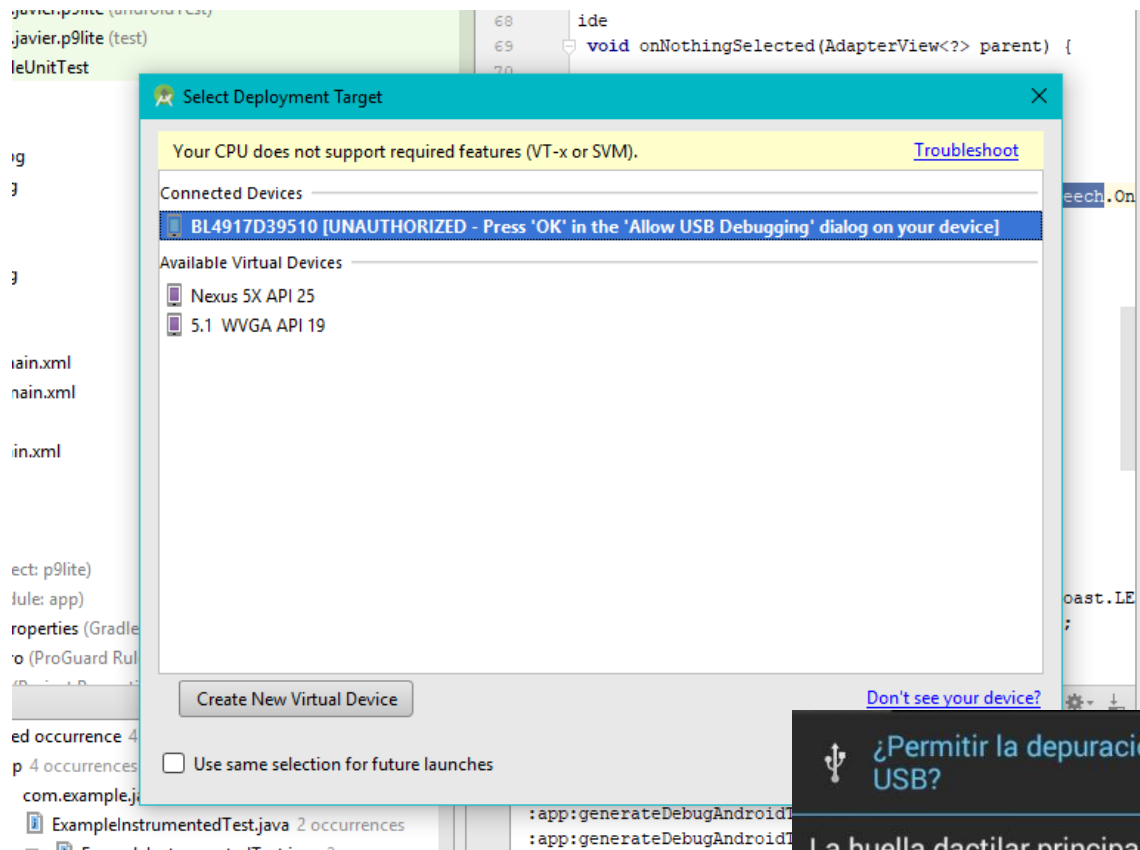
El siguiente paso será, conectar el dispositivo móvil con el ordenador. Seguidamente deberemos ir a ajustes > Acerca del teléfono, y una vez aquí tocaremos repetidamente en el apartado “Numero de compilación”, hasta que nos especifique que ya somos desarrolladores.

Volviendo hacia atrás podremos ver que nos ha aparecido una nueva opción que es la de “Opciones de desarrollador”

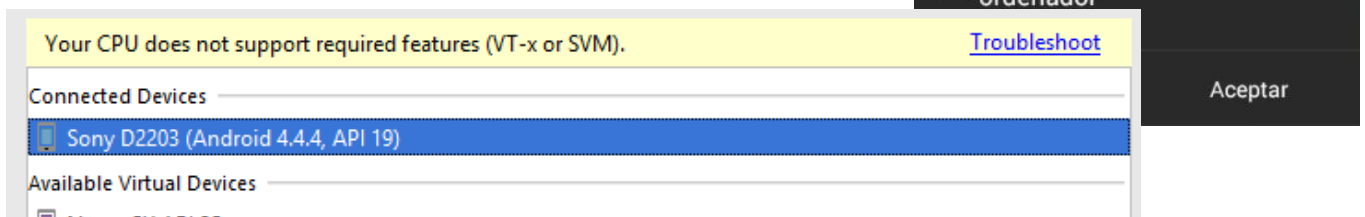
Y activar la “depuración por USB”



Seguidamente iremos a Android estudio y le daremos al “Play” y nos saldrá esta imagen, especificando que no lo reconoce.



Por ello iremos a nuestro teléfono y veremos que ha salido la siguiente imagen y le damos a aceptar y seguidamente Android studio nos reconocerá el dispositivo y la versión.



Aceptamos y se lanzará la aplicación en nuestro dispositivo, de no ser así el propio Android studio buscará la versión del S.O y nos permitirá instalarla.

4.5 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.

5. Proyecto de implementación de un prototipo del sistema utilizando la tecnología iOS

5.1 Documentación de diseño

El diseño completo de la App se compone de la interfaz de usuario que se trata de una sola pantalla donde se realizan todas las acciones disponibles y el propio icono de la aplicación. Además, se ha diseñado una pantalla de carga, mientras el usuario espera a que se lance la App. Se ha optado por un diseño simple y a la vez atractivo para el usuario basado en las líneas modernas de diseño vectorizado actual con colores pálidos (en este caso, un verde azulado como tono predominante combinado con el blanco) y fuentes de letra modernas.



(Diseño Pantalla Inicial)



(Diseño Pantalla de Carga)



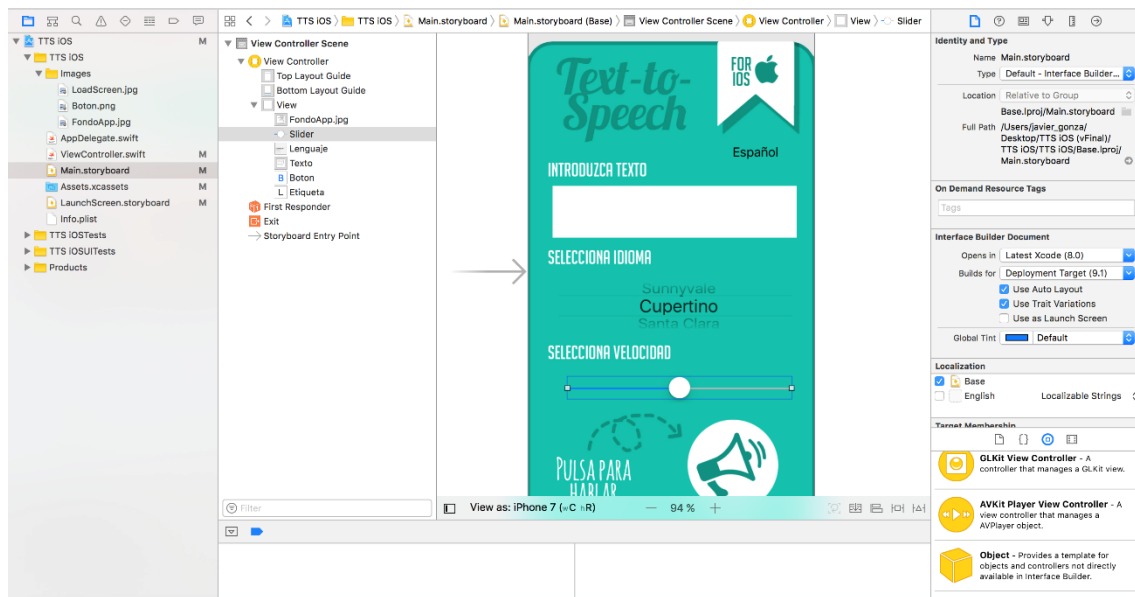
(Diseño Logo App)

5.2 Documentación de construcción

Antes de comenzar con el código, se ha creado un proyecto de lenguaje de programación SWIFT 3 (ya que el entorno Xcode obliga a trabajar con la versión más nueva de éste) para iPhone, configurado un perfil de desarrollador asociado al ID de Apple de Javier González e instalado en su iPhone el certificado de desarrollador correspondiente para poder realizar las pruebas e instalaciones en el mismo.

La App consta de dos archivos importantes sobre los que ha recaído el mayor peso de desarrollo de código:

- 1) Main.storyboard, que almacena la interfaz gráfica, además de los elementos de interacción con el usuario. En este caso, un cuadro de texto, un “picker” de selección de idioma, un slider de velocidad y un botón para hablar.



- 2) ViewController.swift, que almacena la clase principal creada de nuestra aplicación con todas las variables, objetos relacionados con la interfaz y funciones. Importante importar la librería AVFoundation correspondiente a la tecnología TTS de Apple.

En resumidas cuentas, se ha implementado los diseños de la interfaz, creado los elementos de interacción de usuario, declarado dichos elementos en nuestra clase principal de trabajo (main) y se han incluido las diferentes funciones y lógicas de funcionamiento interno.

A continuación, se muestran imágenes de nuestro código fuente donde se ven ejemplos de ello como la declaración de elementos (Primera imagen) y el código del botón de actuación “hablar” (Segunda imagen) con sus comentarios correspondientes y aclarativos.

```
@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.
```

```

@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")
    }
    //Sentencias IF correspondientes a decir el Texto con el idioma seleccionado

    texto.rate = Slider.value;
    //A Texto se le da la velocidad indicada en el Slider, en .rate

    let sintetizador = AVSpeechSynthesizer()
    //Creacion de sintetizador

    sintetizador.speak(texto)
    //El sintetizador dice ese texto con las características que le hemos dado
}

```

5.3 Documentación de pruebas

Los casos de pruebas siempre han estado enfocados a realizar la función principal de nuestra aplicación: hablar determinado texto, habiendo escogido unas determinadas características y siempre se han enfocado pruebas a que esto se lleve a cabo correctamente.

A pesar de ello, se pueden destacar dos casos de pruebas problemáticos específicos en este proceso como son:

- 1) El manejo del teclado de la aplicación. En un principio, tras escribir la información en el cuadro de texto, no se podía pulsar el botón de “Hablar” porque éste se encontraba debajo del teclado que no se ocultaba. La solución fueron un par de líneas para llevar nosotros el manejo de esconder o no el teclado, y siempre que se tocara cualquier lugar de la pantalla, éste se escondería.

```

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)
}

```

- 2) La selección escogida del Picker. En nuestro Picker, tenemos almacenados los idiomas y siempre que se toque el botón “Hablar” se escogería el seleccionado para llevar a cabo dicha acción. El problema se encontraba en que la selección no se podía pasar directamente a la función de acción del Botón, teniendo que añadir una variable intermedia donde se almacenara dicho resultado. La solución fue la creación de una etiqueta oculta donde se almacena el idioma seleccionado en el Picker, y ésta se traspasa a la función del Botón Hablar.

```
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.
```

5.4 Documentación de instalación

Sobre la instalación gratuita del IDE de Apple Xcode, se puede realizar a través del apartado que tienen dedicado en su página web: <https://developer.apple.com/xcode/> o a través de la Mac App Store y siempre desde un Mac con sistema operativo MacOS Sierra (10.12) o superior.

Para la instalación de la App en sí, en estos momentos solo se puede realizar en iPhones que tengan el certificado de desarrollador de Javier González instalado. Esto se puede hacer conectando el iPhone al Mac de Javier con el Xcode desde donde tiene la App para testearla y se darían los permisos.

En un futuro, y si se publicase la App en el AppStore (ya que Apple no admite la instalación de Apps de fuentes externas) se podría descargar a través de ella.

5.5 Manual de usuario

El uso de esta App es sencillo, 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”. Los pasos de selección de idioma, velocidad e introducción de texto pueden realizarse en un orden cualquiera, siempre que finalmente se pulse el botón de hablar.

Los idiomas disponibles son 4: español, inglés, francés y alemán (español viene por defecto, si no escogemos ninguno) y la velocidad escogida puede ser cualquiera entre lo que acota la barra de selección que es 0 y 1, siendo 0,5 el valor por defecto.

6. Comparación de las dos implementaciones

6. Comparación de las dos implementaciones

6.1 Evaluación de los criterios en la implementación usando la tecnología Android

CRITERIO	EVALUACIÓN	COMENTARIOS
Horas invertidas en la creación	8	Las horas en las que más se ha tardado es en la creación de la interfaz
Clases utilizadas en la creación	6	1 para el TTS y 5 para otras funciones
Multiplataforma	NO	Solo se puede utilizar en dispositivos Android
Introducción de varios lenguajes	SI	
Facilidad de implementación	9	Es muy fácil la implementación
Velocidad de uso	7	
Naturalidad de lenguaje	7	El lenguaje es un poco robotizado
Información para implementar la aplicación	10	Se ha podido encontrar mucha información para crear la aplicación
Sencillez interfaz	9	La interfaz es muy sencilla e intuitiva
Coste de implementación	0	

6.2 Evaluación de los criterios en la implementación usando la tecnología IOS

CRITERIO	EVALUACIÓN	COMENTARIOS
Horas invertidas en la creación	9	Las horas en las que más se ha tardado es en la creación de la interfaz
Clases utilizadas en la creación	7	2 para el TTS y 5 para otras funciones
Multiplataforma	NO	Solo se puede en plataformas de Apple
Introducción de varios lenguajes	SI	
Facilidad de implementación	7	
Velocidad de uso	8	

CRITERIO	EVALUACIÓN	COMENTARIOS
Naturalidad de lenguaje	9	Es un lenguaje muy natural en cada idioma
Información para implementar la aplicación	8	Había bastante documentación para ayudarnos a crear la app
Sencillez interfaz	9	La interfaz es muy sencilla e intuitiva
Coste de implementación	0	El coste ha sido 0, ya que no nos hemos dado de alta como programadores

7. Comparación de la implementación de las tecnologías

CRITERIOS	TECNOLOGÍA ANDROID	TECNOLOGÍA IOS	COMENTARIOS
Horas invertidas en la creación	12	13	Más o menos se han utilizado las mismas horas en la creación de las aplicaciones, el mayor tiempo se ha dedicado a mejorar la interfaz gráfica
Clases utilizadas en la creación	6	7	
Multiplataforma		NO	La tecnología implementada en Android solo permite utilizarlo en dispositivos Android y la de iOS solo en dispositivos de Apple
Introducción de varios lenguajes	SI	SI	En ambas se han podido introducir los mismos lenguajes
Facilidad de implementación	9	7	Ha sido un poco más sencillo implementar la tecnología Android de acuerdo a que había mucha más información al respecto
Velocidad de uso	7	8	En Android la aplicación sufre cierto “laggeo” con comparación con iOS
Naturalidad de lenguaje	7	9	Es más natural la voz en iOS, como vimos anteriormente y lo hemos comprobado
Información para implementar la aplicación	10	8	Había más información en Android de acuerdo a que es un Mercado más amplio
Sencillez interfaz	9	9	Ambas interfaces son parecidas y sencillas de acuerdo a los requisitos que implantamos
Coste de implementación	0	0	Los costes han sido 0, hemos buscado todos los recursos gratuitos para realizar las aplicaciones

8. Conclusiones

Las conclusiones, obtenidas después de la realización de los dos prototipos, así como su análisis, es la facilidad de implementación de este tipo de aplicaciones, gracias a los entornos, así como a la documentación disponible en cursos, blogs, videos, documentación oficial, ejemplos GitHub, etc...

También, aun habiendo realizado dos prototipos similares, pero en diferentes plataformas a ambos nos han surgido las mismas dudas o problemas a la hora de realizar una misma solución (en lo referente a la implementación de código), que más tarde, dicho problema se ha solucionado de maneras diferentes igual de válidas.

Por último, y como ya habíamos señalado en la documentación anterior, tras la implementación, nos reafirmamos en que el habla en Android, es más robotizado, y en algunos casos y a veces se “laggea” junto con una especie de latencia cuando se intenta cambiar de idioma rápido y ejecutar el programa.