



UNIVERSIDAD DE MÁLAGA



Graduada en Ingeniería de la Salud

Lector móvil de señales

Mobile sign reader

Realizado por  
Inés Díaz del Rey

Tutorizado por  
José Luis Pastrana Brincones

Departamento  
Lenguajes y Ciencias de la Computación

MÁLAGA, junio de 2022



UNIVERSIDAD  
DE MÁLAGA



ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INFORMÁTICA  
GRADUADA EN INGENIERÍA DE LA SALUD

**Lector móvil de señales**

**Mobile sign reader**

Realizado por  
**Inés Díaz del Rey**

Tutorizado por  
**José Luis Pastrana Brincones**

Departamento  
**Lenguajes y Ciencias de la Computación**

UNIVERSIDAD DE MÁLAGA  
MÁLAGA, JUNIO DE 2022

Fecha defensa: julio de 2022

# Resumen

Actualmente, son muchas las personas afectadas por el deterioro de su visión, provocando que tengan una deficiencia visual. Con el paso de los años el número incrementa, por lo que la creación de herramientas tecnológicas destinadas a personas con esta discapacidad resulta una necesidad para la sociedad, además de ayudar a facilitar su vida diaria.

Este proyecto se ha destinado al desarrollo de una aplicación para el teléfono móvil, tanto para Android como para iOS, de reconocimiento de textos y señales por medio de un sintetizador de voz, con el fin de prestar apoyo a las personas con deficiencia visual en su día a día.

Tras una breve explicación de la metodología y las tecnologías usadas, se ha realizado el análisis, diseño e implementación de la aplicación, terminando con una fase de pruebas y depuración.

Para la implementación de esta aplicación se ha hecho uso de la plataforma Xamarin, con su marco de interfaz de usuario Xamarin.Forms, el cual permite desarrollar aplicaciones móviles compartiendo el mismo código entre las diferentes plataformas.

La aplicación móvil que se ha diseñado proporcionará cierta autonomía a estas personas, ya que le permitirá enfrentarse a situaciones como identificar una parada de autobús o un paso de peatones, al igual que la distinción de textos.

**Palabras clave: Deficiencia visual, Aplicación móvil, Reconocimiento de textos, Reconocimientos de señales, Xamarin.Forms**

# Abstract

Currently, there are many people affected by the deterioration of their vision, causing them to have a visual impairment. Over the years the number increases, so the creation of technological tools for people with this disability is a necessity for society, as well as helping to facilitate their daily lives.

This project has been aimed at the development of a cell phone application, both for Android and iOS, for text and signal recognition by means of a voice synthesizer, in order to provide support to people with visual impairment in their day-to-day.

After a brief explanation of the methodology and technologies used, the analysis, design and implementation of the application was carried out, ending with a testing and debugging phase.

For the implementation of this application we have made use of the Xamarin platform, with its user interface framework Xamarin.Forms, which allows the development of mobile applications sharing the same code between different platforms.

The mobile application that has been designed will provide some autonomy to these people, as it will allow them to face situations such as identifying a bus stop or a crosswalk, as well as the distinction of texts.

**Keywords: Visual Impairment, Mobile Application, Text Recognition, Sign Recognition, Xamarin.Forms**



# Índice

<b>1. Introducción</b>	<b>11</b>
1.1. Motivación . . . . .	11
1.2. Objetivos . . . . .	11
1.3. Estructura del documento . . . . .	12
<b>2. Estado del arte</b>	<b>15</b>
<b>3. Metodología y Tecnologías usadas</b>	<b>17</b>
3.1. Metodología . . . . .	17
3.2. Lenguaje de programación C# . . . . .	18
3.3. Lenguaje de programación XAML . . . . .	19
3.4. Xamarin . . . . .	19
3.4.1. Xamarin.Forms . . . . .	19
3.5. Android . . . . .	20
3.6. iOS . . . . .	20
3.7. Línea temporal . . . . .	21
<b>4. Análisis de la aplicación</b>	<b>23</b>
4.1. Requisitos funcionales . . . . .	23
4.2. Requisitos no funcionales . . . . .	24
4.3. Casos de uso . . . . .	25
<b>5. Diseño de la aplicación</b>	<b>33</b>
5.1. Interfaz de usuario . . . . .	33
5.2. Arquitectura del sistema . . . . .	39
<b>6. Implementación de la aplicación</b>	<b>41</b>
6.1. Interfaz de usuario . . . . .	41
6.1.1. Pantalla de bienvenida . . . . .	41
6.1.2. Pantalla principal . . . . .	42

6.1.3.	Pantalla principal de configuración . . . . .	44
6.1.4.	Pantalla de configuración de idioma . . . . .	47
6.2.	Navegación de menús . . . . .	49
6.3.	Módulo de voz . . . . .	51
6.4.	Base de datos . . . . .	57
6.5.	Reconocimiento de textos y señales . . . . .	65
6.5.1.	Reconocimiento de textos . . . . .	65
6.5.2.	Reconocimiento de señales . . . . .	66
<b>7.</b>	<b>Fase de pruebas y depuración</b>	<b>69</b>
7.1.	Pruebas unitarias . . . . .	69
7.2.	Pruebas de integración . . . . .	71
7.2.1.	Completar la configuración inicial . . . . .	71
7.2.2.	Cambiar la configuración de idioma desde la página principal . . . . .	71
7.2.3.	Cambiar la configuración de voz desde la página principal . . . . .	72
7.2.4.	Cambiar la configuración de descripción desde la página principal . . . . .	72
7.2.5.	Realizar un reconocimiento de texto desde la página principal y escuchar de nuevo la descripción del texto . . . . .	73
7.2.6.	Realizar un reconocimiento de señal desde la página principal y seleccionar la opción de realizar una nueva captura . . . . .	74
7.3.	Pruebas de despliegue . . . . .	74
<b>8.</b>	<b>Conclusiones y Líneas Futuras</b>	<b>77</b>
8.1.	Conclusiones . . . . .	77
8.2.	Líneas Futuras . . . . .	78
	<b>Bibliografía</b>	<b>81</b>
	<b>Apéndice A. Manual de</b>	
	<b>Usuario</b>	<b>83</b>
A.1.	Configuración inicial . . . . .	84
A.2.	Configuración general . . . . .	86
A.3.	Reconocimiento de texto o señal . . . . .	87

# Índice de Figuras

1.	Diagrama de casos de uso . . . . .	25
2.	Pantalla Inicial . . . . .	33
3.	Configuración Inicial Idioma . . . . .	34
4.	Configuración Inicial Voz . . . . .	34
5.	Configuración Inicial Descripción . . . . .	35
6.	Pantalla Principal . . . . .	35
7.	Elección Texto o Señal . . . . .	36
8.	Elección Oír de nuevo o Recapturar . . . . .	36
9.	Pantalla principal de Configuración . . . . .	37
10.	Configuración Idioma . . . . .	37
11.	Configuración Voz . . . . .	38
12.	Configuración Descripción . . . . .	38
13.	Arquitectura del sistema . . . . .	39
14.	Pantalla de Bienvenida . . . . .	83
15.	Pantalla Principal . . . . .	84
16.	Pantalla de Configuración Inicial de Idioma . . . . .	84
17.	Pantalla de Configuración Inicial de Voz . . . . .	85
18.	Pantalla de Configuración Inicial de Descripción . . . . .	85
19.	Pantalla Principal de Configuración . . . . .	86
20.	Pantalla de Configuración de Idioma . . . . .	86
21.	Pantalla de Configuración de Voz . . . . .	87
22.	Pantalla de Configuración de Descripción . . . . .	87
23.	Pantalla de Elección de Texto o Señal . . . . .	88
24.	Acceso Cámara Teléfono Móvil . . . . .	88
25.	Aceptación o Cancelación de Foto . . . . .	89
26.	Pantalla de Nueva Escucha o Captura . . . . .	89





# Índice de Tablas

1.	Caso de uso Configuración . . . . .	26
2.	Caso de uso Descripción . . . . .	26
3.	Caso de uso Idioma . . . . .	27
4.	Caso de uso Voz . . . . .	27
5.	Caso de uso Captura Texto . . . . .	28
6.	Caso de uso Texto . . . . .	28
7.	Caso de uso Oír de nuevo Texto . . . . .	29
8.	Caso de uso Recapturar Texto . . . . .	29
9.	Caso de uso Captura Señal . . . . .	30
10.	Caso de uso Señal . . . . .	30
11.	Caso de uso Oír de nuevo Señal . . . . .	31
12.	Caso de uso Recapturar Señal . . . . .	31
13.	Resultados pruebas unitarias . . . . .	70
14.	Resultados pruebas de despliegue . . . . .	75



# 1

# Introducción

## 1.1. Motivación

A nivel mundial se estima que unos 1.300 millones de personas sufren algún tipo de deficiencia visual. Las causas de esta son diversas, siendo las más frecuentes las cataratas y los errores de refracción no corregidos.

Se considera que una persona tiene déficit visual cuando en al menos uno de sus ojos existe una reducción considerable de la agudeza visual afectada incluso con uso de lentes, o bien, cuando existe una destacable disminución del campo visual del ojo.

La mayoría de las personas con deficiencia visual supera los 50 años, aunque todo el mundo está expuesto a padecerla, bien por causa hereditaria, bien por mal control de la salud ocular.

En este contexto, la tecnología aplicada a este campo puede ser de gran ayuda para las personas afectadas.

En este trabajo se presenta el desarrollo de una aplicación para el teléfono móvil capaz de identificar señales y textos de nuestro entorno, facilitando así la vida cotidiana de las personas con deficiencia visual.

## 1.2. Objetivos

En este trabajo se van a considerar principalmente dos objetivos, uno de carácter formativo y otro de carácter técnico y social.

El objetivo formativo consiste en el aprendizaje de la tecnología Xamarin, que bajo Visual Studio, permite el desarrollo de aplicaciones móviles tanto para IOS como para Android. Dicho aprendizaje se enfocará primordialmente en Xamarin.Forms y los lenguajes asociados XAML y C#.

El segundo objetivo, de carácter técnico y social, se basa en el desarrollo de una herramienta para la mejora de la vida cotidiana de las personas con deficiencia visual. Para ello, se va a desarrollar una aplicación móvil usando Xamarin.Forms, pudiendo así generarla tanto para Android como para IOS, con la que capturar imágenes utilizando la cámara del teléfono móvil para luego analizarlas. Si la imagen capturada se realiza a un texto, el análisis se hará por medio de una librería OCR (reconocimiento óptico de caracteres) a fin de poder informar con un sintetizador de voz del contenido de dicho texto. Si la imagen capturada se realiza a una señal, el análisis se hará por medio de un algoritmo de cálculo de similitud entre imágenes para poder informar por medio del sintetizador de voz de la descripción de dicha señal.

### **1.3. Estructura del documento**

En esta primera parte del trabajo, se describe una breve introducción de la motivación del proyecto y los objetivos a seguir.

A continuación, se desarrollará el estado del arte, en el que comentarán las diferentes aplicaciones enfocadas a facilitar la vida diaria de las personas con deficiencia visual y la aportación en esta que tendrá la aplicación que se desarrollará en el proyecto.

Tras esto, se realizará una descripción de la metodología y las tecnologías usadas. Dichas tecnologías son el lenguaje de programación C#, el lenguaje de programación XAML, Xamarin junto con Xamarin.Forms, Android e iOS.

Posteriormente, se llevará a cabo el desarrollo del análisis de la aplicación. En dicho análisis se describirán los requisitos funcionales, los requisitos no funcionales y los casos de uso. Para una mejor comprensión de los casos de uso se incluirán las plantillas de los mismos.

Después, se comentará el diseño de la aplicación. En esta sección, se hará una descripción de la interfaz de la aplicación, mostrando capturas del diseño de dicha interfaz, y se detallará la arquitectura del sistema que se basará en la arquitectura en capas.

Seguidamente, se pasará a desarrollar la implementación de la aplicación, en la cual se tratará la interfaz de usuario, la navegación de menús, el módulo de voz, la base de datos y el reconocimiento de textos y señales. En cada una de estas secciones, se detallarán las partes del código con funcionalidades más relevantes.

Tras la implementación, se llevará a cabo la fase de pruebas y depuración. En esta fase se desarrollarán pruebas unitarias, pruebas de integración y pruebas de despliegue.

Para finalizar, se comentarán las conclusiones y líneas futuras, en las que se valorará el desarrollo del proyecto y sus limitaciones.

Como documentación final, en el proyecto se incluirá un manual de usuario de la aplicación, donde se detallará el funcionamiento de la misma con los diferentes aspectos a tener en cuenta.



# 2

## Estado del arte

Gracias al avance tecnológico actual existen diferentes aplicaciones útiles para personas con deficiencia visual. Entre ellas se encuentran aplicaciones móviles que ayudan a las personas con problemas de visión a facilitar su vida diaria.

*Be My Eyes* (BeMyEyes, [s.f.](#)) se trata de una aplicación gratuita tanto para Android como iOS, que conecta a personas con discapacidad visual con personas voluntarias a través de una videollamada para que dichos voluntarios les ayuden a realizar tareas sencillas o les indiquen lo que ven a través de la cámara.

*Aira* (Aira, [s.f.](#)) es una aplicación para Android e iOS de pago similar a la anterior, pero en este caso a los usuarios no se les conecta con voluntarios, sino con agentes altamente capacitados para prestar su ayuda en las tareas diarias.

*LookTel* (LookTel, [s.f.](#)) es una aplicación para iOS de pago, utiliza la cámara del teléfono móvil para identificar y transmitir por medio de voz la denominación de un billete.

*TapTapSee* (TapTapSee, [s.f.](#)) se trata de una aplicación gratuita para Android e iOS que permite fotografiar objetos e identificarlos por medio de voz.

*Light Detector* (LightDetector, [s.f.](#)) es una aplicación gratuita para iOS que ayuda a identificar el encendido de luces, donde están las ventanas y si las cortinas están cerradas o abiertas. La aplicación funciona transformando la luz natural o artificial que identifique en sonido.

*KNFB Reader* (KNFBReader, [s.f.](#)) se trata de una aplicación de pago para Android e iOS de conversión de texto a voz.

La aplicación móvil que se va a desarrollar en este proyecto tiene la misma finalidad que las anteriores, facilitar la vida diaria de las personas con deficiencia visual. Será compatible tanto para Android como iOS y sencilla de usar. En todo momento mantendrá al usuario informado



de las decisiones que ha tomado para que no le dificulte la navegación por ella y dispondrá de la posibilidad de oír una descripción detallada de la interfaz de cada pantalla.

El objetivo principal de la aplicación será facilitar al usuario la identificación de textos y de señales tan cotidianas como la parada del bus o del tranvía, el paso de peatones, la salida de emergencia, la parada del taxi, el aseo, el prohibido fumar, el peligro de muerte, flechas de indicación, entre otras. Todo ello descrito por medio de un sintetizador de voz.

Tras la identificación de un texto o una señal, se podrá volver a oír la descripción obtenida o si se desea realizar una nueva captura de cualquier texto o señal.

Con respecto a la configuración, se dispondrá de la posibilidad de elegir el idioma, ya sea español o inglés, implantándose el elegido tanto en la interfaz como en el sintetizador de voz. También se podrá elegir entre un tono de voz agudo o un tono de voz grave, según las preferencias del usuario. Y por último, existirá la posibilidad de elegir si se desea oír una descripción detallada de la interfaz de la aplicación durante la navegación del usuario en ella o no.

La aplicación no necesitará acceso a Internet, por lo que se podrá utilizar en todo momento. Además, con el fin de no ocupar más espacio en el dispositivo tras su instalación y configuración, las imágenes tomadas con la cámara no se almacenarán en la galería.

# 3

## Metodología y Tecnologías usadas

### 3.1. Metodología

La aplicación móvil se ha desarrollado haciendo uso de Xamarin, una plataforma de código abierto que se utiliza para compilar aplicaciones con mejor rendimiento para iOS y Android. Específicamente, se ha utilizado Xamarin.Forms. Se trata de un marco de interfaz de usuario de código abierto, en el cual se crean interfaces de usuario en lenguaje XAML y se desarrolla código en lenguaje C#.

Para una mejor comprensión del análisis del sistema, en concreto de los casos de uso, se ha utilizado el lenguaje de especificación UML.

En la arquitectura del sistema se ha utilizado la arquitectura en capas, la cual se realiza dividiendo la aplicación en capas, de manera que cada una de ellas tenga un papel específico.

El desarrollo de la aplicación se ha llevado a cabo mediante una metodología iterativa incremental. En cada iteración se ha añadido más funcionalidad, creando así versiones más complejas, obteniendo retroalimentación que orientase el desarrollo según las necesidades que aparecían y realizando adaptaciones con el fin de cumplir los objetivos planteados.

Para el proceso de lectura de textos se ha utilizado la librería de OCR *Tesseract* de Xamarin, basada en el reconocimiento óptico de caracteres, facilitando la identificación de los caracteres visualizados en las imágenes tomadas con la cámara del teléfono móvil. Por otro lado, para la

identificación de señales se ha hecho uso de un conjunto de algoritmos capaces de encontrar similitud entre imágenes, obteniendo así la descripción de la señal capturada con la cámara.

Con el fin de facilitar el uso de la aplicación móvil para personas con deficiencia visual, se ha incorporado el sintetizador de voz *Text to Speech* de Xamarin, el cual informa de la descripción de las páginas de la aplicación, de la navegación que el usuario realice y, evidentemente, de la identificación obtenida del texto o señal capturados.

### **3.2. Lenguaje de programación C#**

El lenguaje de programación C# tiene sus raíces en la familia de lenguajes C. Este es un lenguaje de programación orientado a objetos y orientado a componentes. C# facilita a los desarrolladores la creación de diferentes aplicaciones sólidas y seguras que se ejecutan en .NET.

Desde su origen, C# ha introducido diversas características con el fin de aceptar nuevas cargas de trabajo y prácticas emergentes de diseño de software. Actualmente, se encuentra en estado de prueba la versión 11 de C#.

Cabe destacar en este lenguaje su compatibilidad con las operaciones asincrónicas, las cuales permiten retrasar la ejecución de una función a la espera de completarse otra. Todos los tipos de C# se heredan de un mismo y único tipo raíz *object*. Los valores de cualquier tipo pueden ser almacenados, transportados y operados de manera coherente.

Además, C# hace énfasis en el control de versiones para asegurarse de la compatibilidad de los programas y bibliotecas que evolucionan con el tiempo.

En C# hay dos clases de tipos: tipos de valor y tipos de referencia. Las variables de tipos de valor contienen directamente sus datos y las variables de tipos de referencia almacenan referencias a sus datos. Los tipos de valor de C# se dividen en tipos simples, de estructura, de enumeración, de valor de tupla y de valor que aceptan valores NULL. Los tipos de referencia de C# se dividen en tipos de clase, de matriz, de interfaz y delegados.

### 3.3. Lenguaje de programación XAML

XAML viene determinado por lenguaje de marcado de aplicaciones extensible y se trata de un lenguaje declarativo basado en XML. XAML es el principal formato para declarar la interfaz de usuario de una aplicación y los elementos de dicha interfaz de usuario si se programa con C#.

En las aplicaciones de Xamarin.Forms se usa XAML para definir el contenido visual de una página y funciona junto con un archivo de código subyacente de C#. Las propiedades que se establezcan en los elementos creados en el archivo XAML pueden ser modificadas en el archivo de código subyacente de C#, siempre y cuando se le haya designado un valor al atributo *x:Name* del elemento, ya que, en el archivo de código C# este valor pasará a ser un objeto del tipo que sea el elemento.

### 3.4. Xamarin

Xamarin es una plataforma de código abierto para compilar aplicaciones modernas que ofrecen un mejor rendimiento para Android, iOS y Windows con .NET. A pesar de que Xamarin permita desarrollar aplicaciones en C# y compartir el mismo código en las plataformas mencionadas, la implementación real en cada sistema es diferente.

#### 3.4.1. Xamarin.Forms

Xamarin.Forms es un marco de interfaz de usuario de código abierto. Por medio de Xamarin.Forms, los desarrolladores tienen la posibilidad de compilar aplicaciones en Xamarin.Android, Xamarin.iOS y Windows desde un único código base compartido. Además, los desarrolladores pueden crear interfaces de usuario en XAML junto al código subyacente en C#. Estas interfaces se representan en cada plataforma como controles nativos con mejor rendimiento, por lo que se obtiene una apariencia y un funcionamiento nativos.

Xamarin.Forms tiene una amplia variedad de bibliotecas que agregan diversas funciones a las aplicaciones. Entre las más destacadas se encuentra Xamarin.Essentials, la cual simplifica el proceso de acceso a utilidades nativas, ya que proporciona varias.

### 3.5. Android

Android corresponde a un sistema operativo móvil basado en el núcleo Linux y otros software de código abierto. Este sistema fue diseñado para dispositivos móviles con pantalla táctil y se trata del sistema operativo móvil más utilizado del mundo, muy por encima de iOS.

Inicialmente, la empresa Android Inc. desarrolló dicho sistema, pero en 2005 fue adquirida por Google. En 2007, Android fue presentado junto con la fundación del Open Handset Alliance para progresar en los estándares abiertos de los dispositivos móviles. El 5 de noviembre de 2007 se anuncia la primera versión del sistema operativo denominada *Android 1.0 Apple Pie*, pero hasta el año 2008 no estuvieron disponibles los terminales con Android. El código fuente principal de Android se conoce como Android Open Source Project (AOSP), el cual está licenciado principalmente bajo la Licencia Apache.

Android se ha sometido a numerosas actualizaciones desde su aparición. Estas actualizaciones suelen arreglar fallos y agregar nuevas funciones. Generalmente, cada versión de Android tomaba el nombre de un elemento relacionado con dulces en orden alfabético, pero a partir de la versión 10 esto dejó de ser así. La última versión, lanzada el 4 de octubre de 2021, ha sido Android 12, la cual proporciona un sistema más eficiente y con mejor rendimiento.

### 3.6. iOS

iOS es un sistema operativo móvil de la empresa multinacional Apple Inc. Originalmente se desarrolló para el teléfono iPhone, aunque después se ha usado en otros tipos de dispositivos Apple. La empresa no permite la instalación de iOS en hardware de terceros.

El 9 de enero de 2007, Apple reveló la existencia de iPhone OS (iOS) en la Macworld Conference Expo, aunque no se le designó un nombre oficial al sistema hasta la salida de la primera versión beta del iPhone SDK (kit de desarrollo software) el 6 de marzo de 2008. La primera versión de iOS apareció con el lanzamiento del iPhone el 17 de junio de 2007. En sus inicios se le asignó el nombre iPhone OS, pero debido a su incorporación en tres dispositivos diferentes, el 7 de junio de 2010 se renombró a iOS.

Apple provee de actualizaciones del sistema operativo de los diferentes dispositivos asociados por medio de iTunes. El 20 de septiembre de 2021 se lanzó la última versión de este software denominada iOS 15, con el objetivo principal de mejorar la experiencia del usuario.

### **3.7. Línea temporal**

La planificación de este trabajo se basará en las siguientes fases y se estimará el tiempo asignado para cada una de ellas, sumando un total de 296 horas.

1. Estudio y aprendizaje de las herramientas y del estado del arte: 40 horas
  - 1.1. Estudio de Xamarin, Xamarin.Forms: 20 horas
  - 1.2. Estudio de C# y el estado del arte: 20 horas
2. Estudio y obtención de requisitos: 20 horas
3. Diseño y arquitectura: 40 horas
  - 3.1. Diseño de la interfaz de usuario: 20 horas
  - 3.2. Diseño de la arquitectura del sistema: 20 horas
4. Implementación: 120 horas
  - 4.1. Implementación de la interfaz de usuario: 20 horas
  - 4.2. Implementación del reconocimiento de textos y señales: 20 horas
    - 4.2.1. Preprocesamiento de imágenes: 20 horas
    - 4.2.2. Base de datos de símbolos conocidos: 20 horas
  - 4.3. Implementación de la navegación de menús: 20 horas
  - 4.4. Implementación del módulo de voz: 20 horas
5. Fase de pruebas y depuración: 60 horas
  - 5.1. Pruebas unitarias: 20 horas
  - 5.2. Pruebas de integración: 20 horas
  - 5.3. Pruebas de despliegue: 20 horas
6. Desarrollo del manual de uso de la aplicación: 16 horas



# 4

## Análisis de la aplicación

### 4.1. Requisitos funcionales

En esta sección, se van a detallar los requisitos funcionales del sistema, es decir, el comportamiento que dicho sistema debe tener cuando se cumplen ciertas condiciones.

**RF1:** Cada vez que se acceda a la aplicación aparecerá una pantalla de bienvenida.

**RF2:** Las acciones que se realicen serán comentadas por medio de un sintetizador de voz. Y si se desea, los detalles de la aplicación también se comentarán.

**RF3:** Al entrar en la aplicación por primera vez, se mostrarán las configuraciones de “Idioma”, “Voz” y “Descripción”.

**RF4:** En la página principal, se mostrarán las opciones de “Configuración” y “Pulsar para iniciar”.

**RF5:** Al seleccionar “Configuración”, se indicarán las diferentes opciones a configurar disponibles “Idioma”, “Voz”, “Descripción” y “Salir”.

**RF5.1:** Al seleccionar “Idioma”, se permitirá elegir entre español o inglés. Además, se mostrará la opción “Salir” para regresar a la página de configuración.

**RF5.2:** Al seleccionar “Voz”, se permitirá elegir entre un tono de voz agudo o un tono de voz grave. Además, se mostrará la opción “Salir” para regresar a la página de configuración.

**RF5.3:** Al seleccionar “Descripción”, se permitirá elegir entre obtener una descripción detallada (“Sí”) o no obtenerla (“No”). Además, se mostrará la opción “Salir” para regresar a la página de configuración.



**RF5.4:** Al seleccionar “Salir”, se regresará a la página principal.

**RF6:** Al seleccionar “Pulsar para iniciar”, se indicará las dos posibles opciones de captura a realizar “Texto” y “Señal”.

**RF6.1:** Al seleccionar “Texto”, se accederá a la cámara del teléfono móvil para realizar una foto al texto que se desee identificar.

**RF6.2:** Al seleccionar “Señal”, se accederá a la cámara del teléfono móvil para realizar una foto a la señal que se desee identificar.

**RF7:** Tras realizar la foto, se transmitirá por medio del sintetizador de voz lo que se ha identificado en esta.

**RF8:** Tras la identificación de la foto tomada, se mostrarán las opciones de “Oír de nuevo”, “Realizar otra captura” y “Salir”.

**RF8.1:** Al seleccionar “Oír de nuevo”, se volverá a transmitir por medio del sintetizador de voz la información obtenida de la foto.

**RF8.2:** Al seleccionar “Realizar otra captura”, se volverán a mostrar las opciones de captura a realizar “Texto” y “Señal”. Y se procederá a *RF6.1* y *RF6.2*.

**RF8.3:** Al seleccionar “Salir”, se regresará a la página principal.

## **4.2. Requisitos no funcionales**

A continuación, se detallarán los requisitos no funcionales del sistema, los cuales representan las características generales que debe tener dicho sistema.

**RNF1:** El sistema debe presentar una interfaz de usuario sencilla con el fin de facilitar el manejo de la aplicación a los usuarios.

**RNF2:** La navegación entre páginas debe ser fluida, proporcionando tiempos de respuesta rápidos.

**RNF3:** La base de datos de la aplicación debe ser local para facilitar la disponibilidad de uso de la aplicación sin conexión a internet.

**RNF4:** La aplicación necesita ser instalada en dispositivos Android 8.0 o superior e iOS 7 o superior.

**RNF5:** El dispositivo de instalación debe tener mínimo 300 MB disponibles para instalar la aplicación y poderse usar.

**RNF6:** La aplicación debe ser fácil de analizar y modificar para corregir posibles errores.

### 4.3. Casos de uso

En esta sección, se van a describir los casos de uso, los cuales definen las acciones que el usuario llevará a cabo durante el manejo de la aplicación.

Para una previa visión de estos, se mostrará a continuación un diagrama de los casos de uso de la aplicación usando el Lenguaje de Modelo Unificado (UML).

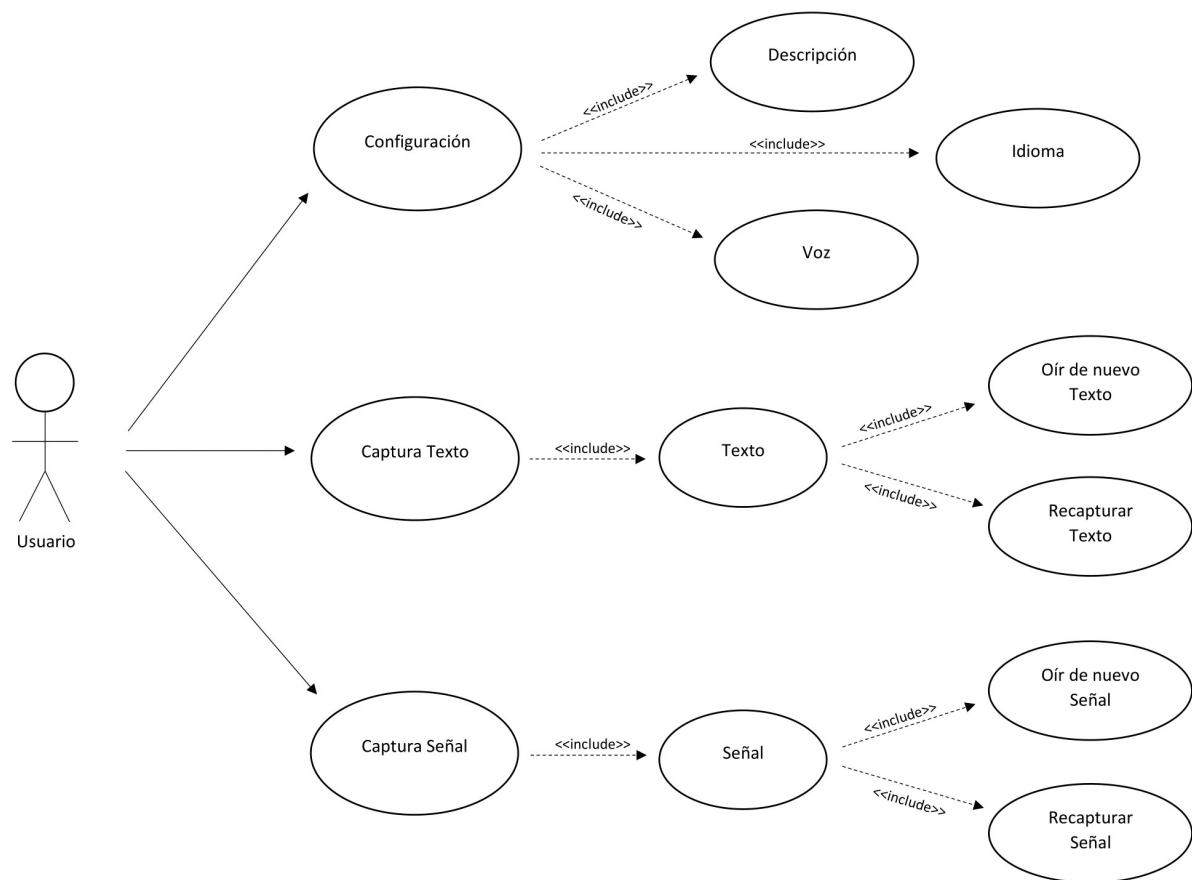


Figura 1: Diagrama de casos de uso

Con el fin de facilitar la comprensión del diagrama de casos de uso, se van a detallar seguidamente las plantillas de los diferentes casos de uso.

<b>Nombre:</b>	Configuración
<b>Descripción:</b>	Permite al usuario seleccionar las diferentes opciones de configuración de la aplicación.
<b>Actores:</b>	Usuario
<b>Precondición:</b>	-
<b>Postcondición:</b>	El usuario ha elegido una opción a configurar
<b>Flujo principal:</b>	<ol style="list-style-type: none"> <li>1. El usuario solicita configurar la aplicación</li> <li>2. El sistema muestra las diferentes opciones de configuración que existen</li> <li>3. El usuario elige una opción de configuración</li> <li>4. El sistema redirige a la página de configuración de la opción seleccionada</li> <li>5. El caso de uso finaliza</li> </ol>
<b>Flujo alternativo:</b>	<ol style="list-style-type: none"> <li>3a. El usuario elige la opción de salir de la configuración <ol style="list-style-type: none"> <li>3a.1. El sistema redirige a la página principal</li> </ol> </li> </ol>

Tabla 1: Caso de uso Configuración

<b>Nombre:</b>	Descripción
<b>Descripción:</b>	Permite al usuario modificar la configuración de la descripción
<b>Actores:</b>	Usuario
<b>Precondición:</b>	El usuario ha elegido la opción de configuración “Descripción”
<b>Postcondición:</b>	El usuario ha modificado la configuración de “Descripción”
<b>Flujo principal:</b>	<ol style="list-style-type: none"> <li>1. El sistema muestra las dos opciones disponibles a elegir para la configuración de “Descripción” y se encuentra marcada la opción previamente elegida</li> <li>2. El usuario elige la opción de descripción que desea</li> <li>3. El sistema almacena la nueva información en la base de datos y recarga la página con la selección elegida</li> <li>4. El caso de uso finaliza</li> </ol>
<b>Flujo alternativo:</b>	<ol style="list-style-type: none"> <li>2a. El usuario elige la opción de salir de la configuración de descripción <ol style="list-style-type: none"> <li>2a.1. El sistema redirige a la página de configuración principal</li> </ol> </li> </ol>

Tabla 2: Caso de uso Descripción

<b>Nombre:</b>	Idioma
<b>Descripción:</b>	Permite al usuario modificar la configuración del idioma
<b>Actores:</b>	Usuario
<b>Precondición:</b>	El usuario ha elegido la opción de configuración “Idioma”
<b>Postcondición:</b>	El usuario ha modificado la configuración de “Idioma”
<b>Flujo principal:</b>	<ol style="list-style-type: none"> <li>1. El sistema muestra las dos opciones disponibles a elegir para la configuración de “Idioma” y se encuentra marcada la opción previamente elegida</li> <li>2. El usuario elige la opción de idioma que desea</li> <li>3. El sistema almacena la nueva información en la base de datos y recarga la página con la selección elegida</li> <li>4. El caso de uso finaliza</li> </ol>
<b>Flujo alternativo:</b>	<ol style="list-style-type: none"> <li>2a. El usuario elige la opción de salir de la configuración de idioma <ol style="list-style-type: none"> <li>2a.1. El sistema redirige a la página de configuración principal</li> </ol> </li> </ol>

Tabla 3: Caso de uso Idioma

<b>Nombre:</b>	Voz
<b>Descripción:</b>	Permite al usuario modificar la configuración del tono de voz
<b>Actores:</b>	Usuario
<b>Precondición:</b>	El usuario ha elegido la opción de configuración “Voz”
<b>Postcondición:</b>	El usuario ha modificado la configuración de “Voz”
<b>Flujo principal:</b>	<ol style="list-style-type: none"> <li>1. El sistema muestra las dos opciones disponibles a elegir para la configuración de “Voz” y se encuentra marcada la opción previamente elegida</li> <li>2. El usuario elige la opción de tono de voz que desea</li> <li>3. El sistema almacena la nueva información en la base de datos y recarga la página con la selección elegida</li> <li>4. El caso de uso finaliza</li> </ol>
<b>Flujo alternativo:</b>	<ol style="list-style-type: none"> <li>2a. El usuario elige la opción de salir de la configuración de voz <ol style="list-style-type: none"> <li>2a.1. El sistema redirige a la página de configuración principal</li> </ol> </li> </ol>

Tabla 4: Caso de uso Voz

<b>Nombre:</b>	Captura Texto
<b>Descripción:</b>	Permite al usuario realizar una foto con la cámara del teléfono móvil al texto que desee
<b>Actores:</b>	Usuario
<b>Precondición:</b>	El usuario ha pulsado en el botón “Pulsa para iniciar” de la página principal
<b>Postcondición:</b>	El usuario ha capturado el texto deseado para analizar
<b>Flujo principal:</b>	<ol style="list-style-type: none"> <li>1. El usuario elige la opción de capturar un texto</li> <li>2. El sistema accede a la cámara del teléfono móvil</li> <li>3. El usuario realiza la foto al texto que desee</li> <li>4. El usuario acepta la foto realizada</li> <li>5. El caso de uso finaliza</li> </ol>
<b>Flujo alternativo:</b>	<ol style="list-style-type: none"> <li>4a. El usuario deniega la foto realizada <ol style="list-style-type: none"> <li>4a.1 Regreso al paso 2</li> </ol> </li> </ol>

Tabla 5: Caso de uso Captura Texto

<b>Nombre:</b>	Texto
<b>Descripción:</b>	Permite al usuario escuchar la descripción del texto capturado
<b>Actores:</b>	Usuario
<b>Precondición:</b>	El usuario ha realizado la captura al texto deseado
<b>Postcondición:</b>	El sistema ha informado de la descripción obtenida del texto capturado
<b>Flujo principal:</b>	<ol style="list-style-type: none"> <li>1. El sistema registra la imagen capturada y la analiza</li> <li>2. El sistema devuelve la descripción obtenida del texto capturado</li> <li>3. El caso de uso finaliza</li> </ol>
<b>Flujo alternativo:</b>	<ol style="list-style-type: none"> <li>2a. El sistema no reconoce el texto capturado y no devuelve una descripción de este</li> </ol>

Tabla 6: Caso de uso Texto

<b>Nombre:</b>	Oír de nuevo Texto
<b>Descripción:</b>	Permite al usuario volver a escuchar la descripción del texto capturado
<b>Actores:</b>	Usuario
<b>Precondición:</b>	El sistema ha informado previamente de la descripción obtenida del texto capturado
<b>Postcondición:</b>	El sistema ha vuelto a informar de la descripción obtenida del texto capturado
<b>Flujo principal:</b>	<ol style="list-style-type: none"> <li>1. El usuario ha seleccionado la opción de oír de nuevo el texto capturado</li> <li>2. El sistema vuelve a informar de la descripción obtenida del texto</li> <li>3. El caso de uso finaliza</li> </ol>
<b>Flujo alternativo:</b>	-

Tabla 7: Caso de uso Oír de nuevo Texto

<b>Nombre:</b>	Recapturar Texto
<b>Descripción:</b>	Permite al usuario realizar una nueva foto con la cámara del teléfono móvil a otro texto que desee
<b>Actores:</b>	Usuario
<b>Precondición:</b>	El sistema ha informado previamente de la descripción obtenida del texto capturado
<b>Postcondición:</b>	El sistema informado de la descripción obtenida del nuevo texto capturado
<b>Flujo principal:</b>	<ol style="list-style-type: none"> <li>1. El sistema accede a la cámara del teléfono móvil</li> <li>2. El usuario realiza la foto al texto que desee</li> <li>3. El usuario acepta la foto realizada</li> <li>4. El sistema registra la imagen capturada y la analiza</li> <li>5. El sistema devuelve la descripción obtenida del texto capturado</li> <li>6. El caso de uso finaliza</li> </ol>
<b>Flujo alternativo:</b>	<ol style="list-style-type: none"> <li>3a. El usuario deniega la foto realizada <ol style="list-style-type: none"> <li>3a.1 Regreso al paso 1</li> </ol> </li> <li>5a. El sistema no reconoce el texto capturado y no devuelve una descripción de este</li> </ol>

Tabla 8: Caso de uso Recapturar Texto

<b>Nombre:</b>	Captura Señal
<b>Descripción:</b>	Permite al usuario realizar una foto con la cámara del teléfono móvil a la señal que desee
<b>Actores:</b>	Usuario
<b>Precondición:</b>	El usuario ha pulsado en el botón “Pulsa para iniciar” de la página principal
<b>Postcondición:</b>	El usuario ha capturado la señal deseada para analizar
<b>Flujo principal:</b>	<ol style="list-style-type: none"> <li>1. El usuario elige la opción de capturar una señal</li> <li>2. El sistema accede a la cámara del teléfono móvil</li> <li>3. El usuario realiza la foto a la señal que desee</li> <li>4. El usuario acepta la foto realizada</li> <li>5. El caso de uso finaliza</li> </ol>
<b>Flujo alternativo:</b>	<ol style="list-style-type: none"> <li>4a. El usuario deniega la foto realizada <ol style="list-style-type: none"> <li>4a.1 Regreso al paso 2</li> </ol> </li> </ol>

Tabla 9: Caso de uso Captura Señal

<b>Nombre:</b>	Señal
<b>Descripción:</b>	Permite al usuario escuchar la descripción de la señal capturada
<b>Actores:</b>	Usuario
<b>Precondición:</b>	El usuario ha realizado la captura a la señal deseada
<b>Postcondición:</b>	El sistema ha informado de la descripción obtenida de la señal capturada
<b>Flujo principal:</b>	<ol style="list-style-type: none"> <li>1. El sistema registra la imagen capturada y la analiza</li> <li>2. El sistema devuelve la descripción obtenida de la señal capturada</li> <li>3. El caso de uso finaliza</li> </ol>
<b>Flujo alternativo:</b>	<ol style="list-style-type: none"> <li>2a. El sistema no reconoce la señal capturada y no devuelve una descripción de esta</li> </ol>

Tabla 10: Caso de uso Señal

<b>Nombre:</b>	Oír de nuevo Señal
<b>Descripción:</b>	Permite al usuario volver a escuchar la descripción de la señal capturada
<b>Actores:</b>	Usuario
<b>Precondición:</b>	El sistema ha informado previamente de la descripción obtenida de la señal capturada
<b>Postcondición:</b>	El sistema ha vuelto a informar de la descripción obtenida de la señal capturada
<b>Flujo principal:</b>	<ol style="list-style-type: none"> <li>1. El usuario ha seleccionado la opción de oír de nuevo la señal capturada</li> <li>2. El sistema vuelve a informar de la descripción obtenida de la señal</li> <li>3. El caso de uso finaliza</li> </ol>
<b>Flujo alternativo:</b>	-

Tabla 11: Caso de uso Oír de nuevo Señal

<b>Nombre:</b>	Recapturar Señal
<b>Descripción:</b>	Permite al usuario realizar una nueva foto con la cámara del teléfono móvil a otra señal que desee
<b>Actores:</b>	Usuario
<b>Precondición:</b>	El sistema ha informado previamente de la descripción obtenida de la señal capturada
<b>Postcondición:</b>	El sistema informado de la descripción obtenida de la nueva señal capturada
<b>Flujo principal:</b>	<ol style="list-style-type: none"> <li>1. El sistema accede a la cámara del teléfono móvil</li> <li>2. El usuario realiza la foto a la señal que desee</li> <li>3. El usuario acepta la foto realizada</li> <li>4. El sistema registra la imagen capturada y la analiza</li> <li>5. El sistema devuelve la descripción obtenida de la señal capturada</li> <li>6. El caso de uso finaliza</li> </ol>
<b>Flujo alternativo:</b>	<ol style="list-style-type: none"> <li>3a. El usuario deniega la foto realizada <ol style="list-style-type: none"> <li>3a.1 Regreso al paso 1</li> </ol> </li> <li>5a. El sistema no reconoce la señal capturada y no devuelve una descripción de esta</li> </ol>

Tabla 12: Caso de uso Recapturar Señal





# 5

## Diseño de la aplicación

### 5.1. Interfaz de usuario

En esta sección, se mostrará el diseño de las pantallas que componen la aplicación y se hará una breve explicación de ellas.

La pantalla inicial tendrá el siguiente diseño y se tratará de una *splash screen*.



Figura 2: Pantalla Inicial

Tras la pantalla inicial, aparecerá una pantalla de configuración inicial de idioma en la que existirán las opciones de español o inglés.



Figura 3: Configuración Inicial Idioma

A continuación, por cada tipo de pantalla se mostrarán dos versiones, la versión con la configuración de idioma en español y la versión con la configuración de idioma en inglés.

Tras la pantalla de configuración de idioma inicial, se mostrará la pantalla de configuración inicial de voz en la que podrá elegir entre un tono de voz agudo y un tono de voz grave.

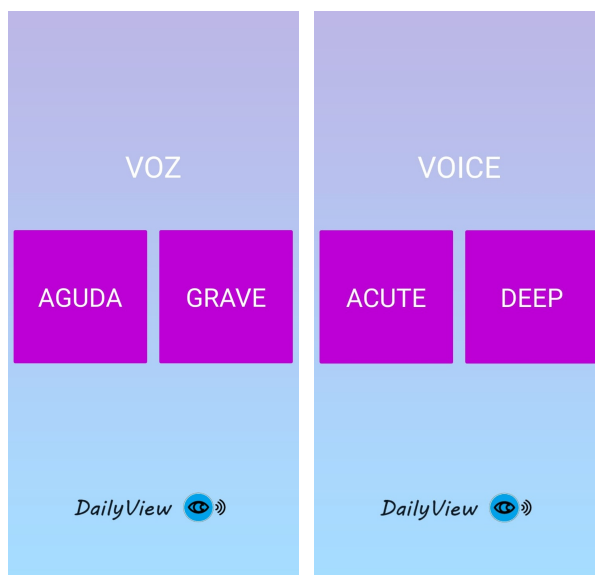


Figura 4: Configuración Inicial Voz

Tras la pantalla anterior, se mostrará la pantalla de configuración inicial de descripción en la que podrá elegir entre obtener una descripción detallada de la aplicación o no.



Figura 5: Configuración Inicial Descripción

Tras la configuración inicial, se muestra la pantalla principal de la aplicación donde se podrá elegir entre acceder a la configuración o iniciar la captura.

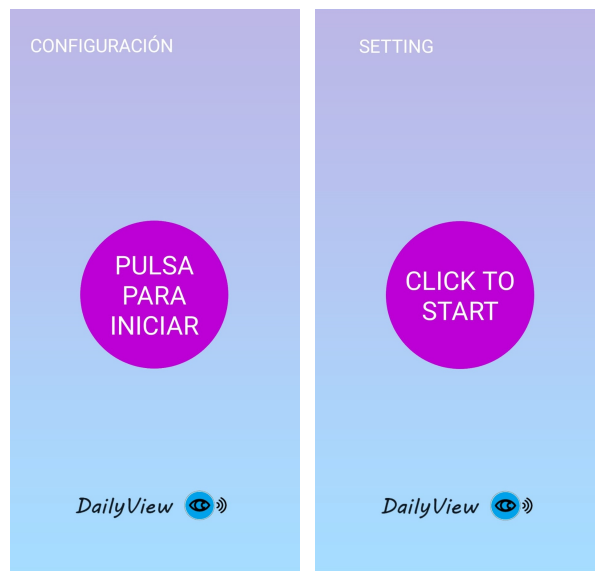


Figura 6: Pantalla Principal

Si en la pantalla principal se ha elegido la opción “Pulsar para iniciar”, se mostrará la pantalla siguiente en la que habrá que elegir para capturar un texto o una señal.

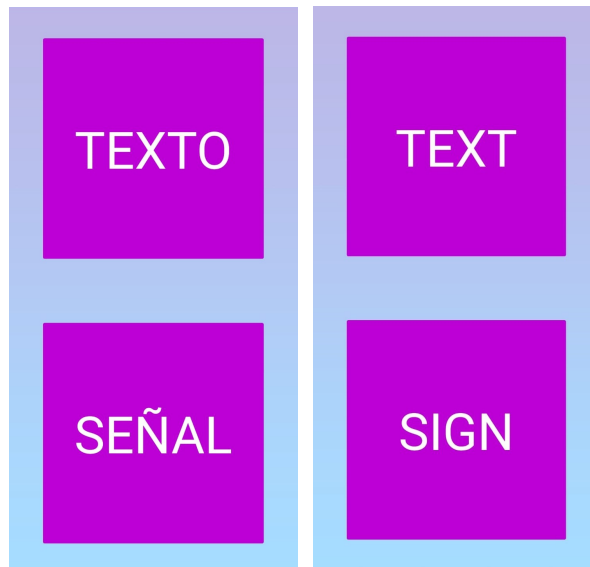


Figura 7: Elección Texto o Señal

Tras elegir entre texto o señal, realizar la captura y que el sistema la analice y comente la descripción de esta, se mostrará la siguiente pantalla en la que se podrá elegir las opciones de oír de nuevo la descripción o realizar una nueva captura.

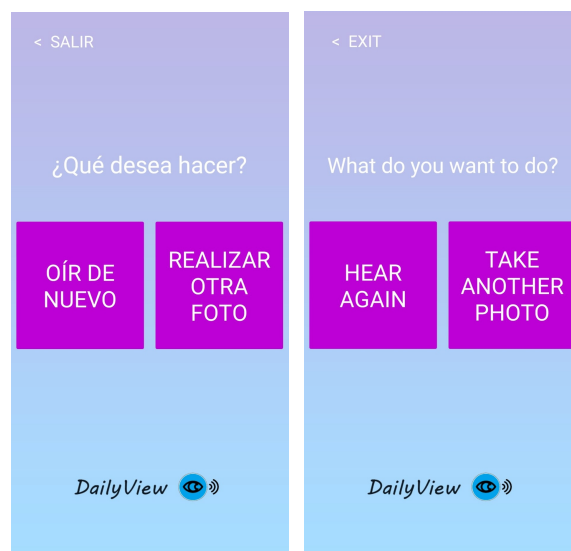


Figura 8: Elección Oír de nuevo o Recapturar

Si en la pantalla principal se ha elegido la opción “Configuración”, se mostrará la pantalla de configuración principal con las tres opciones a configurar: “Descripción”, “Idioma” y “Voz”.



Figura 9: Pantalla principal de Configuración

Si se ha elegido configurar el idioma, se mostrará una pantalla en la que elegir como idioma español o inglés y además, estará la selección previa marcada.



Figura 10: Configuración Idioma

Si se ha elegido configurar la voz, se mostrará una pantalla en la que elegir como tono de voz agudo o grave y además, estará la selección previa marcada.

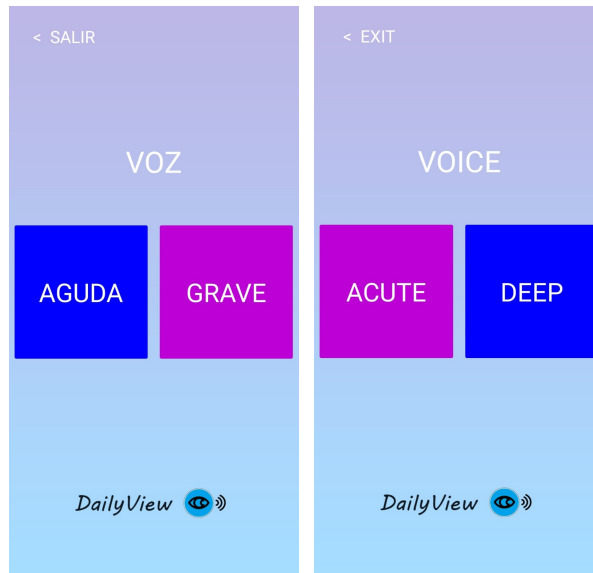


Figura 11: Configuración Voz

Si se ha elegido configurar la descripción, se mostrará una pantalla en la que elegir si se desea una descripción detallada o no y además, estará la selección previa marcada.



Figura 12: Configuración Descripción

## 5.2. Arquitectura del sistema

La arquitectura del sistema que se va a utilizar es la arquitectura en capas, es una de las más utilizadas y se realiza dividiendo la aplicación en capas, de manera que cada capa tenga un papel específico. En esta arquitectura, todas las capas se disponen de forma horizontal, de esta forma cada capa solo podrá comunicarse con la que le sigue.

Las capas que comprenderán mi sistema serán la capa de presentación, la capa de negocio y la capa de acceso a datos.

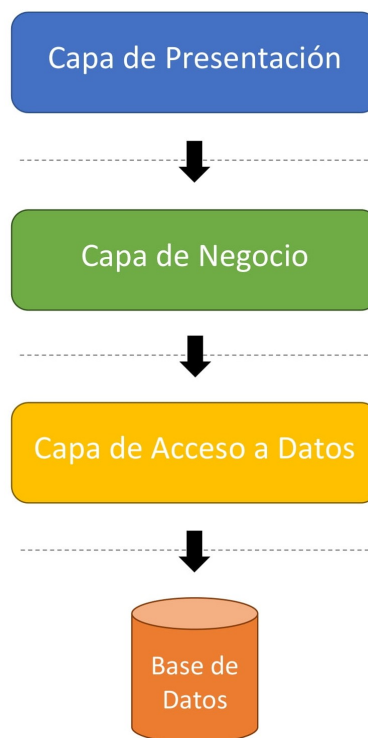


Figura 13: Arquitectura del sistema

- *Capa de presentación:* Es el nivel superior del sistema. Esta capa presenta la interfaz de usuario de la aplicación, que comunica la información y captura la que proporcione el usuario. Además, está en comunicación con la capa de negocio.
- *Capa de negocio:* Es el nivel intermedio del sistema. Esta capa contiene los procesos a realizar tras recibir la información de la capa de presentación, se ocupa de las peticiones que el usuario ha solicitado y es responsable de que se envíen respuestas adecuadas a la capa de presentación. Además, está en comunicación tanto con la capa de presentación



como con la capa de acceso a datos, por lo que también procesa la información que provenga de la capa de acceso a datos.

- *Capa de acceso a datos*: Es el último nivel del sistema. En esta capa se obtienen los datos almacenados en la base de datos. Se comunica con la capa de negocio con el fin de facilitar datos o recibirlos, ya que puede acceder a los datos por medios de instrucciones para su consulta, almacenaje, actualización o eliminación.

# 6

## Implementación de la aplicación

La implementación de la aplicación se ha desarrollado en el entorno de desarrollo Visual Studio 2019 con la creación de un proyecto de aplicación móvil en Xamarin.Forms.

### 6.1. Interfaz de usuario

Para la implementación de la interfaz de usuario se ha utilizado el lenguaje de programación XAML. La aplicación tiene un total de once pantallas diferentes y para el diseño de cada una de ellas se ha desarrollado un archivo XAML.

En esta sección, se van a mostrar los archivos XAML más relevantes para la visualización de la creación de la interfaz de usuario. Entre estos archivos están los asociados a la pantalla de bienvenida (splash screen), a la pantalla principal, a la pantalla principal de configuración y a la pantalla de configuración de idioma, obviando el resto de pantallas por similitud a las anteriores. Las pantallas restantes serían las correspondientes a las de configuración de voz, configuración de descripción, recaptura y nueva escucha, elección de texto o señal, configuración inicial de idioma, configuración inicial de voz y configuración inicial de descripción.

#### 6.1.1. Pantalla de bienvenida

##### *SplashPage.xaml*

```
<?xml version="1.0" encoding="utf-8" ?>
<ContentPage xmlns="http://xamarin.com/schemas/2014/forms"
              xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
              x:Class="DailyView.SplashPage"
```

```

        BackgroundImageSource="splash_background.png"
        NavigationPage.HasNavigationBar="False">
<ContentPage.Content>
    <StackLayout>
        <Image x:Name="img" />
    </StackLayout>
</ContentPage.Content>
</ContentPage>

```

En esta pantalla se usa de fondo de la imagen vista anteriormente en la sección de diseño de la interfaz, guardada en el archivo *splash\_background.png*, y se usa un elemento de tipo *Image* para designarle duración de aparición por medio del código C# asociado.

### 6.1.2. Pantalla principal

#### *MainPage.xaml*

```

<?xml version="1.0" encoding="utf-8" ?>
<ContentPage xmlns="http://xamarin.com/schemas/2014/forms"
    xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
    x:Class="DailyView.MainPage"
    BackgroundImageSource="background.png"
    NavigationPage.HasNavigationBar="False">

    <ContentPage.Content>
        <Grid RowSpacing="0"
            ColumnSpacing="0">
            <Grid.RowDefinitions>
                <RowDefinition />
                <RowDefinition />
                <RowDefinition />
            </Grid.RowDefinitions>
            <Grid.ColumnDefinitions>
                <ColumnDefinition />
                <ColumnDefinition />
            </Grid.ColumnDefinitions>

```

```

        <ColumnDefinition />
    </Grid.ColumnDefinitions>

    <Button x:Name="Setting"
        Grid.Column="0"
        Grid.ColumnSpan="2"
        Grid.Row="0"
        Text="Configuración"
        WidthRequest="250"
        HeightRequest="100"
        VerticalOptions="Start"
        HorizontalOptions="Start"
        BackgroundColor="Color.Transparent"
        FontSize="25"
        Clicked="OnButtonClicked_Setting" />

    <Button x:Name="Start"
        Grid.Column="0"
        Grid.ColumnSpan="3"
        Grid.Row="1"
        Text="Pulsa para iniciar"
        WidthRequest="200"
        HeightRequest="200"
        CornerRadius="100"
        VerticalOptions="CenterAndExpand"
        HorizontalOptions="Center"
        BackgroundColor="#ea8bff"
        FontSize="35"
        TextColor="White"
        Clicked="OnButtonClicked_Start" />

</Grid>
</ContentPage.Content>
</ContentPage>

```

En esta pantalla se usa de fondo una imagen genérica para el resto de pantallas, como se pudo ver en la sección de diseño de la interfaz. Dicha imagen se guarda en el archivo *background.png*. Además, se muestran dos elementos de tipo *Button*, el botón de configuración redirigirá a la página principal de configuración y el botón de inicio redirigirá a la página de elección de texto o señal.

### 6.1.3. Pantalla principal de configuración

#### *SettingPage.xaml*

```
<?xml version="1.0" encoding="utf-8" ?>
<ContentPage xmlns="http://xamarin.com/schemas/2014/forms"
  xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
  x:Class="DailyView.SettingPage"
  BackgroundImageSource="background.png"
  NavigationPage.HasNavigationBar="False">

  <ContentPage.Content>
    <Grid RowSpacing="0"
      ColumnSpacing="0">
      <Grid.RowDefinitions>
        <RowDefinition />
        <RowDefinition />
        <RowDefinition />
      </Grid.RowDefinitions>
      <Grid.ColumnDefinitions>
        <ColumnDefinition />
        <ColumnDefinition />
      </Grid.ColumnDefinitions>

      <Button x:Name="Exit"
        Grid.Column="0"
        Grid.Row="0"
        Text="&lt; Salir">
```

```

WidthRequest="200"
HeightRequest="100"
VerticalOptions="Start"
HorizontalOptions="Start"
BackgroundColor="Color.Transparent"
FontSize="23"
Padding="0,0,50,10"
TextColor="White"
Clicked="OnButtonClicked_Exit" />

<Button x:Name="Explanation"
Grid.Column="1"
Grid.Row="0"
Text="DESCRIPCIÓN"
WidthRequest="300"
HeightRequest="100"
VerticalOptions="Start"
HorizontalOptions="Start"
BackgroundColor="Color.Transparent"
FontSize="23"
Padding="0,0,10,10"
TextColor="White"
Clicked="OnButtonClicked_Explanation" />

<Label x:Name="Setting"
Grid.Column="0"
Grid.ColumnSpan="2"
Grid.Row="0"
VerticalOptions="End"
HorizontalOptions="Center"
Padding="0,0,0,20"
Text="CONFIGURACIÓN"
FontSize="40"

```

```

        TextColor="White" />

    <Button x:Name="Language"
        Grid.Column="0"
        Grid.Row="1"
        Text="Idioma"
        WidthRequest="180"
        HeightRequest="180"
        VerticalOptions="CenterAndExpand"
        HorizontalOptions="Center"
        BackgroundColor="#ea8bff"
        FontSize="35"
        TextColor="White"
        Clicked="OnButtonClicked_Language" />

    <Button x:Name="Voice"
        Grid.Column="1"
        Grid.Row="1"
        Text="Voz"
        WidthRequest="180"
        HeightRequest="180"
        VerticalOptions="CenterAndExpand"
        HorizontalOptions="Center"
        BackgroundColor="#ea8bff"
        FontSize="35"
        TextColor="White"
        Clicked="OnButtonClicked_Voice" />

</Grid>
</ContentPage.Content>
</ContentPage>

```

En esta pantalla se usa de fondo la imagen genérica guardada en el archivo *background.png*. Se compone de cinco elementos, cuatro de tipo *Button* y uno de tipo *Label*. El último contiene el

nombre asignado a la pantalla “Configuración”. Con respecto a los botones, el botón “Idioma” redirige a la pantalla de configuración de idioma, el botón “Voz” a la pantalla de configuración de voz, el botón “Descripción” a la pantalla de configuración de descripción y el botón “Salir” a la pantalla principal.

#### 6.1.4. Pantalla de configuración de idioma

##### *LanguagePage.xaml*

```
<?xml version="1.0" encoding="utf-8" ?>
<ContentPage xmlns="http://xamarin.com/schemas/2014/forms"
             xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
             x:Class="DailyView.LanguagePage"
             BackgroundImageSource="background.png"
             NavigationPage.HasNavigationBar="False">

    <ContentPage.Content>
        <Grid RowSpacing="0"
            ColumnSpacing="0">
            <Grid.RowDefinitions>
                <RowDefinition />
                <RowDefinition />
                <RowDefinition />
            </Grid.RowDefinitions>
            <Grid.ColumnDefinitions>
                <ColumnDefinition />
                <ColumnDefinition />
            </Grid.ColumnDefinitions>

            <Button x:Name="Exit"
                  Grid.Column="0"
                  Grid.ColumnSpan="2"
                  Grid.Row="0"
                  Text="&lt; Salir" />
```



```

        WidthRequest="250"
        HeightRequest="100"
        VerticalOptions="Start"
        HorizontalOptions="Start"
        BackgroundColor="Color.Transparent"
        FontSize="23"
        Padding="0,0,100,15"
        TextColor="White"
        Clicked="OnButtonClicked_Exit" />

<Label x:Name="Language"
        Grid.Column="0"
        Grid.ColumnSpan="2"
        Grid.Row="0"
        VerticalOptions="End"
        HorizontalOptions="Center"
        Padding="0,0,0,20"
        Text="IDIOMA"
        FontSize="40"
        TextColor="White" />

<ImageButton x:Name="Spain"
        Grid.Column="0"
        Grid.Row="1"
        Source="spain.png"
        WidthRequest="180"
        HeightRequest="180"
        VerticalOptions="CenterAndExpand"
        HorizontalOptions="Center"
        BackgroundColor="Color.Transparent"
        Clicked="OnButtonClicked_Spain" />

<ImageButton x:Name="US"

```

```

        Grid.Column=" 1 "
        Grid.Row=" 1 "
        Source=" us . png "
        WidthRequest=" 180 "
        HeightRequest=" 180 "
        VerticalOptions=" CenterAndExpand "
        HorizontalOptions=" Center "
        BackgroundColor=" Color . Transparent "
        Clicked=" OnButtonClicked_US " />
    </Grid>
</ContentPage.Content>
</ContentPage>

```

En esta última pantalla se usa de fondo la imagen genérica guardada en el archivo *background.png*. Se compone de cuatro elementos, tres de tipo *Button* y uno de tipo *Label*. El último contiene el nombre asignado a la pantalla “Idioma”. Con respecto a los botones, el botón con la bandera de España almacena en la base de datos la elección del idioma en español, el botón con la bandera de Estados Unidos almacena en la base de datos la elección del idioma en inglés y el botón “Salir” a la pantalla principal de configuración.

En todas las pantallas se tendrá en cuenta la elección del idioma. Por defecto, el diseño se creó en español, pero en los archivos de código C# asociados se considera si el lenguaje elegido es inglés para cambiar el texto de los elementos.

## 6.2. Navegación de menús

La navegación entre las diferentes pantallas se ha realizado mediante navegación jerárquica por medio de la clase *NavegationPage*. Con esta clase, el usuario puede navegar por las pantallas hacia adelante y hacia atrás como desee. La clase implementa la navegación como una pila de objetos en la que la última pantalla en entrar es la primera en salir.

El inicio de la navegación debe realizarse en el archivo *App.xaml.cs*. La primera página que se va a agregar a la pila de navegación es la página de bienvenida, y esto se realiza de la

siguiente forma.

```
public App()
{
    InitializeComponent();

    MainPage = new NavigationPage(new SplashPage());
}
```

Por tanto, al iniciar la aplicación, aparecerá la página de bienvenida, la cual, como se mencionó anteriormente, es temporal, y este límite de duración de aparición se ha implementado por medio de la función *img.FadeTo(1,1000)*, donde *img* corresponde al elemento *Image* de la pantalla de bienvenida que se comentó en la sección anterior.

Para navegar por el resto de pantallas, se hace uso del método *PushAsync* en la propiedad *Navigation* de cada página. Se trata de un método asíncrono que añadirá la pantalla asignada a la pila de navegación. Por ejemplo, para navegar de la página de bienvenida a la página de configuración inicial de idioma, se haría por medio de la siguiente línea de código.

```
await Navigation.PushAsync(new IniLanguagePage(settingsES_A ,
settingsES_D , settingsUS_A , settingsUS_D));
```

Los argumentos asociados a la clase *IniLanguagePage* están relacionados con el sintetizador de voz, por lo que se explicarán en la sección “Módulo de voz”.

El método a seguir en el resto de páginas es idéntico. Por ejemplo, en la pantalla principal, al pulsar el botón de configuración debe redirigirse a la pantalla de configuración inicial, por tanto, en el método *OnButtonClicked\_Setting* asociado a la acción de pulsar el botón de configuración se añade la siguiente línea de código para la redirección.

```
await Navigation.PushAsync(new SettingPage(setting , settingsES_A ,
settingsES_D , settingsUS_A , settingsUS_D));
```

El atributo *setting* será comentado en la sección “Base de datos”.

De igual forma se realiza para aquellos botones que deban redireccionar a otras páginas,

como son los botones asociados a salir de la página actual. Por ejemplo, si se pulsa el botón “Salir” de la página de configuración de descripción, esta acción deberá redirigir a la pantalla de configuración principal. A continuación, se muestra la implementación del método de la clase *ExplanationPage* que realizará la acción.

```
async void OnButtonClicked_Exit(object sender, EventArgs e)
{
    await semaphoreSlim.WaitAsync();

    await Navigation.PushAsync(new SettingPage(setting,
settingsES_A, settingsES_D, settingsUS_A, settingsUS_D));

    ...

    semaphoreSlim.Dispose();
    semaphoreSlim = new SemaphoreSlim(1, 1);
}
```

Los puntos suspensivos corresponden a una parte del código del método irrelevante para esta explicación. El uso de la clase *SemaphoreSlim* se destina únicamente a controlar la ejecución de los procesos.

Por tanto, se ha podido observar que la navegación jerárquica tiene una implementación sencilla y de igual uso para todas las pantallas de la aplicación. En el método asociado a la acción de pulsar cualquier botón que se redireccione a otra página, simplemente habría que llamar al método *PushAsync* y pasar como parámetro la página a la que se vaya a redirigir.

### 6.3. Módulo de voz

En esta sección, se hablará sobre la implementación del sintetizador de voz *Text to Speech* de Xamarin.

En primer lugar, es necesario agregar la referencia *Xamarin.Essentials* al documento donde se vaya a utilizar la clase *TextToSpeech*. Dicha clase funciona por medio de la llamada al método

*SpeakAsync*, la cual tiene como parámetros el texto a escuchar y las opciones de lenguaje y voz.

Se han implementado cuatro objetos de la clase *SpeechOptions* para poder disponer de diferentes combinaciones de opciones de lenguaje y voz. Esto es así debido a la configuración que elige el usuario, la cual puede ser idioma español o inglés y tono de voz agudo o grave. Por tanto, las cuatro opciones disponibles son español y tono agudo, español y tono grave, inglés y tono agudo e inglés y tono grave. A continuación se muestra la creación de estos cuatro objetos.

```
var locales = await TextToSpeech.GetLocalesAsync();

foreach (Locale locale in locales)
{
    if (locale.Language.ToString().Equals("en") && locale.Country
        .ToString().Equals("US"))
    {
        localeUS = locale;
    }
    else if (locale.Language.ToString().Equals("es") && locale
        .Country.ToString().Equals("ES"))
    {
        localeES = locale;
    }
}

var settingsUS_A = new SpeechOptions()
{
    Volume = 1.0f,
    Pitch = 1.0f,
    Locale = localeUS
};

var settingsES_A = new SpeechOptions()
```

```

{
    Volume = 1.0f,
    Pitch = 1.0f,
    Locale = localeES
};

var settingsUS_D = new SpeechOptions()
{
    Volume = 1.0f,
    Pitch = 0.5f,
    Locale = localeUS
};

var settingsES_D = new SpeechOptions()
{
    Volume = 1.0f,
    Pitch = 0.5f,
    Locale = localeES
};

```

El bucle *foreach* simplemente fue un mero trámite para obtener ambos idiomas como objetos del tipo *Locale*, ya que así se configuran en las opciones. El parámetro *Pitch* determina el tono de la voz y el parámetro *Volume* el volumen que tendrá la voz, el cual se dejó a un valor apropiado para no ser molesto.

A continuación, se mostrará un ejemplo de la implementación del sintetizador de voz cuando se pulsa el botón de configuración de la página principal y se redirige a la página principal de configuración.

```

async void OnButtonClicked_Setting(object sender, EventArgs e)
{
    await semaphoreSlim.WaitAsync();

    if (setting.language == "Spain" && setting.voice == "Acute")

```

```

    {
        await TextToSpeech.SpeakAsync("Ha pulsado el botón de
configuración.", settingsES_A);
    }
    else if (setting.language == "US" && setting.voice == "Acute"
)
    {
        await TextToSpeech.SpeakAsync("You have clicked the
settings button.", settingsUS_A);
    }
    else if (setting.language == "Spain" && setting.voice == "
Deep")
    {
        await TextToSpeech.SpeakAsync("Ha pulsado el botón de
configuración.", settingsES_D);
    }
    else if (setting.language == "US" && setting.voice == "Deep")
    {
        await TextToSpeech.SpeakAsync("You have clicked the
settings button.", settingsUS_D);
    }

    await Navigation.PushAsync(new SettingPage(setting ,
settingsES_A , settingsES_D , settingsUS_A , settingsUS_D));

    if (setting.language == "Spain" && setting.voice == "Acute"
&& setting.explanation == "yes")
    {
        await TextToSpeech.SpeakAsync("Se encuentra en la página
de configuración. " +
            "Pulse el botón central izquierdo si desea cambiar el
idioma o pulse el botón central derecho si desea modificar el
tono de voz. " +

```

```

        "Para modificar la configuración de descripción ,
pulse el botón situado en la esquina superior derecha. " +
        "Para salir de la página de configuración pulse el
botón situado en la esquina superior izquierda.", settingsES_A);
    }
    else if (setting.language == "US" && setting.voice == "Acute"
&& setting.explanation == "yes")
    {
        await TextToSpeech.SpeakAsync("You are on the settings
page. " +
        "Click the central left button if you want to change
the language or click the central right button if you want to
change the tone of voice. " +
        "To modify the description settings , click the button
located in the upper right corner. " +
        "To exit the settings page click the button located
in the upper left corner.", settingsUS_A);
    }
    else if (setting.language == "Spain" && setting.voice == "
Deep" && setting.explanation == "yes")
    {
        await TextToSpeech.SpeakAsync("Se encuentra en la página
de configuración. " +
        "Pulse el botón central izquierdo si desea cambiar el
idioma o pulse el botón central derecho si desea modificar el
tono de voz. " +
        "Para modificar la configuración de descripción ,
pulse el botón situado en la esquina superior derecha. " +
        "Para salir de la página de configuración pulse el
botón situado en la esquina superior izquierda.", settingsES_D);
    }
    else if (setting.language == "US" && setting.voice == "Deep"
&& setting.explanation == "yes")

```



```

    {
        await TextToSpeech.SpeakAsync("You are on the settings
page. " +
            "Click the central left button if you want to change
the language or click the central right button if you want to
change the tone of voice. " +
            "To modify the description settings , click the button
located in the upper right corner. " +
            "To exit the settings page click the button located
in the upper left corner.", settingsUS_D);
    }
    else if (setting.language == "Spain" && setting.voice == "
Acute" && setting.explanation == "no")
    {
        await TextToSpeech.SpeakAsync("Se encuentra en la página
de configuración.", settingsES_A);
    }
    else if (setting.language == "US" && setting.voice == "Acute"
&& setting.explanation == "no")
    {
        await TextToSpeech.SpeakAsync("You are on the settings
page.", settingsUS_A);
    }
    else if (setting.language == "Spain" && setting.voice == "
Deep" && setting.explanation == "no")
    {
        await TextToSpeech.SpeakAsync("Se encuentra en la página
de configuración.", settingsES_D);
    }
    else if (setting.language == "US" && setting.voice == "Deep"
&& setting.explanation == "no")
    {
        await TextToSpeech.SpeakAsync("You are on the settings

```

```

page.", settingsUS_D);
    }

    semaphoreSlim.Dispose();
    semaphoreSlim = new SemaphoreSlim(1, 1);
}

```

En la variable *setting* se hará más hincapié en la sección “Base de datos”, pero se toman en cuenta las configuraciones del usuario, si ha elegido el idioma español o inglés, el tono de voz agudo o grave o si oír la descripción detallada o no. Y en función de estas configuraciones, se introducen como parámetros el texto y las opciones correspondientes.

## 6.4. Base de datos

Para la implementación de la base de datos se hizo uso de la base de datos *SQLite* local, por lo que fue necesario incluir la librería *SQLite.NET* en el proyecto por medio del paquete de NuGet *sqlite-net-pcl*.

El siguiente paso es crear dos carpetas, una nombrada como “Data” y otra nombrada como “Models”. En esta última, se guardan los ficheros “DVSetting.cs” y “Sign.cs”, cada uno de ellos representa una clase de la base de datos. En “DVSetting.cs” se almacenarán los datos pertenecientes a la configuración del usuario y en “Sign.cs” se almacenarán los datos pertenecientes a los archivos de las señales que se van a considerar. Por otro lado, en la carpeta “Data” se guarda el fichero “Database.cs” en el que se establece la base de datos y se crean los métodos de obtención, actualización, almacenamiento y eliminación de los datos para las dos clases.

### *DVSetting.cs*

```

using SQLite;

namespace DailyView.Models
{
    public class DVSetting
    {

```

```

        [PrimaryKey]
        public int id { get; set; }
        public string language { get; set; }
        public string voice { get; set; }
        public int clickStart { get; set; }
        public string explanation { get; set; }
    }
}

```

En esta entidad, existen cuatro atributos: *language* que almacena el idioma elegido, *voice* que almacena el tono de voz elegido, *explanation* que almacena si se ha elegido la descripción detallada o no, y *clickStart* que almacena simplemente si se pulsó el botón de texto o señal por primera vez para saber si el sistema ha solicitado permisos de acceso al almacenamiento interno del dispositivo y avisar sobre ello.

### ***Sign.cs***

```

using SQLite;

namespace DailyView.Models
{
    public class Sign
    {
        [PrimaryKey]
        public string imageSource { get; set; }
        public string description { get; set; }
    }
}

```

En esta otra entidad, existen dos atributos: *imageSource* que almacena el nombre del archivo donde se guarda la imagen de la señal y *description* que almacena la descripción de la señal.

### ***Database.cs***

```

using System.Threading.Tasks;

```

```

using System.Collections.Generic;
using SQLite;
using DailyView.Models;

namespace DailyView.Data
{
    public class Database
    {
        readonly SQLiteAsyncConnection database;

        public Database(string dbPath)
        {
            // Establishing the connection
            database = new SQLiteAsyncConnection(dbPath);
            database.CreateTableAsync<DVSetting>().Wait();
            database.CreateTableAsync<Sign>().Wait();
        }

        public Task<List<DVSetting>> GetSettingsAsync()
        {
            // Get all settings.
            return database.Table<DVSetting>().ToListAsync();
        }

        public Task<DVSetting> GetSettingAsync(int id)
        {
            // Get a specific setting.
            return database.Table<DVSetting>()
                .Where(i => i.id == id)
                .FirstOrDefaultAsync();
        }

        public Task<int> UpdateSettingAsync(DVSetting setting)

```

```

    {
        // Update an existing setting.
        return database.UpdateAsync(setting);
    }

    public Task<int> SaveSettingAsync(DVSetting setting)
    {
        // Save a new setting.
        return database.InsertAsync(setting);
    }

    public Task<int> DeleteSettingAsync(DVSetting setting)
    {
        // Delete a setting.
        return database.DeleteAsync(setting);
    }

    public Task<List<Sign>> GetSignsAsync()
    {
        //Get all signs.
        return database.Table<Sign>().ToListAsync();
    }

    public Task<Sign> GetSignAsync(string imgS)
    {
        // Get a specific sign.
        return database.Table<Sign>()
            .Where(i => i.imageSource == imgS)
            .FirstOrDefaultAsync();
    }

    public Task<int> UpdateSignAsync(Sign sign)

```

```

        {
            // Update an existing sign.
            return database.UpdateAsync(sign);
        }

        public Task<int> SaveSignAsync(Sign sign)
        {
            // Save a new sign.
            return database.InsertAsync(sign);
        }

        public Task<int> DeleteSignAsync(Sign sign)
        {
            // Delete a sign.
            return database.DeleteAsync(sign);
        }
    }
}

```

En este archivo, primeramente, en el constructor de la clase se establece la conexión con la base de datos, y dicha clase se compone de diez métodos. Por un lado, están los métodos asociados a la clase *DVSetting*: *GetSettingsAsync()*, devuelve todos los registros de la clase, *GetSettingAsync(int id)* devuelve el registro cuyo *id* sea el que se pasa como parámetro, *UpdateSettingAsync(DVSetting setting)* actualiza el registro introducido con los nuevos parámetros, *SaveSettingAsync(DVSetting setting)* inserta el registro pasado como parámetro, y *DeleteSettingAsync(DVSetting setting)* elimina el registro pasado como parámetro. Por el otro lado, están los métodos asociados a la clase *Sign*: *GetSignsAsync()*, *GetSignAsync(string imgS)*, *UpdateSignAsync(Sign sign)*, *SaveSignAsync(Sign sign)* y *DeleteSignAsync(Sign sign)*, que se puede concluir la funcionalidad por lógica y similitud a los anteriores mencionados.

En la clase principal *App*, se crea un objeto *Database* por medio del patrón *Singleton*. El propósito de este patrón es asegurarse de que una clase, en este caso *Database*, tenga una única instancia. Esto se logra creando la instancia en una clase, la clase *App*, y recuperándolo

como una instancia estática. A continuación, se mostrará la creación de dicho objeto.

```
mstatic Database database;

public static Database Database
{
    get
    {
        if (database == null)
        {
            database = new Database(Path.Combine(Environment.
GetFolderPath(
                Environment.SpecialFolder.LocalApplicationData),
"DailyViewDB.db3"));

            string [] files = new string [] {"AseoHombreFigura.jpg"
, "AseoMinusvalidosFigura.jpg", "AseoMujerFigura.jpg",
                "FlechaAbajo.jpg", "FlechaArriba.jpg", "
FlechaDerecha.jpg", "FlechaIzquierda.jpg",
                "Senal_CaminoReservadoParaPeatones.jpg", "
Senal_Extintor.jpg", "Senal_Hospital.jpg",
                "Senal_InformacionTuristica.jpg", "Senal_Manguera
.jpg", "Senal_MonumentoNacional.jpg",
                "Senal_ParadaDeAutobus.jpg", "
Senal_ParadaDeTranvias.jpg", "Senal_PasoDePeatones.jpg",
                "Senal_PasoInferiorParaPeatones.jpg", "
Senal_PasoSuperiorParaPeatones.jpg", "Senal_PeligroDeMuerte.jpg",
                "Senal_PeligroDeRadiacion.jpg", "
Senal_PeligroIndeterminado.jpg", "Senal_PeligroRiesgoDeIncendio.
jpg",
                "Senal_PeligroRiesgoElectrico.jpg", "
Senal_PeligroSueloResbaladizo.jpg", "Senal_ProhibidoFumar.jpg",
                "Senal_ProhibidoPasoPeatones.jpg", "
Senal_ProhibidoPerros.jpg", "Senal_SalidaDeEmergencia.jpg", "
```

```

Senal_Taxi.jpg"};

        string[] descriptions = new string[] {"Figura de
hombre.", "Figura de minusválido.", "Figura de mujer.",
        "Flecha hacia abajo.", "Flecha hacia arriba.", "
Flecha a la derecha.", "Flecha a la izquierda.",
        "Señal de camino reservado para peatones.", "
Señal de extintor.", "Señal de hospital.",
        "Señal de información turística.", "Señal de
manguera.", "Señal de monumento nacional.",
        "Señal de parada de autobús.", "Señal de parada
de tranvías.", "Señal de paso de peatones.",
        "Señal de paso inferior para peatones.", "Señal
de paso superior para peatones.", "Señal de peligro de muerte.",
        "Señal de peligro de radiación.", "Señal de
peligro de indeterminado.", "Señal de peligro de riesgo de
incendio.",
        "Señal de peligro de riesgo eléctrico.", "Señal
de peligro de suelo resbaladizo.", "Señal de prohibido fumar.",
        "Señal de prohibido el paso de peatones.", "Señal
de prohibido perros.", "Señal de salida de emergencia.", "Señal
de taxi."};

        for (int i = 0; i < files.Length; i++)
        {
            Sign sign = new Sign { imageSource = "DailyView."
+ files[i], description = descriptions[i] };
            App.Database.SaveSignAsync(sign);
        }
    }
    return database;
}

```



```
}  
}
```

En el código anterior, se puede observar que si no existe la base de datos, se crea y se realiza el almacenaje de las señales en ella. Se definen dos *arrays* de caracteres, uno guarda los nombres de los archivos y el otro las descripciones, que se pasan a un bucle *for* para ir almacenando cada señal en la base de datos.

Para la creación del primer y único registro que tendrá el usuario en la tabla de la clase *DVSetting* de la base de datos, se irá recogiendo la información de la configuración inicial, tanto del idioma como del tono de voz, los cuales pasarán como parámetros en la navegación hacia una nueva página de la configuración inicial de descripción. Por tanto, el constructor de la página de configuración inicial de descripción será de la siguiente forma: *IniExplanationPage(string language, string voice, SpeechOptions setES\_A, SpeechOptions setES\_D, SpeechOptions setUS\_A, SpeechOptions setUS\_D)*. Siendo *language* el idioma elegido, *voice* el tono de voz y los objetos de tipo *SpeechOptions* las configuraciones de lenguaje y voz mencionadas anteriormente. El parámetro *language* se guarda en la variable *lang* y el parámetro *voice* en la variable *voi*, con el fin de facilitar la creación del registro en la tabla de la entidad.

Si el usuario decidiera pulsar el botón “Sí” asociado a oír la descripción detallada, el registro se almacenaría de la siguiente forma.

```
DVSetting setting = new DVSetting { id = 0, language = lang ,  
voice = voi, clickStart = 0, explanation = "yes" };  
await App.Database.SaveSettingAsync( setting );
```

Si, por el contrario, decidiera pulsar el botón “No”, cambiaría el valor del atributo *explanation* por “no”.

Una vez completada la configuración inicial y creado el registro, se irá pasando como parámetro en la creación de las diferentes páginas un objeto de tipo *DVSetting* que corresponderá al registro creado, para así facilitar el acceso a su información, que será necesaria en todas las páginas, y poder modificar la información en las páginas asociadas a la configuración.

Por ejemplo, si el usuario deseara cambiar el idioma de inglés a español, las líneas de código correspondientes a este cambio de idioma en el registro de la tabla serían las siguientes.

```
setting.language = "Spain";  
await App.Database.UpdateSettingAsync(setting);
```

Siendo *setting* una variable interna de la página que guarda la información del objeto *DV-Setting* pasado como parámetro, el cual se menciona anteriormente.

## 6.5. Reconocimiento de textos y señales

En esta sección, se tratará la implementación del reconocimiento de textos por medio de la librería OCR *Tesseract* y del reconocimiento de señales por medio de un conjunto de algoritmos capaces de encontrar similitud entre imágenes.

En ambos reconocimientos se requiere el acceso a la cámara del teléfono móvil y para ello se ha instalado el paquete de Nuget *Xamarin.Plugin.Media*.

### 6.5.1. Reconocimiento de textos

Para hacer uso de la librería OCR *Tesseract* es necesario instalar el paquete de NuGet *Xamarin.Tesseract* en el proyecto, y añadir la referencia *using Tesseract*; en la página que se vaya a emplear.

En primer lugar, se accede a la cámara del teléfono móvil para tomar la foto.

```
await CrossMedia.Current.Initialize();  
var photo = await CrossMedia.Current.TakePhotoAsync(new  
StoreCameraMediaOptions()  
{  
    PhotoSize = PhotoSize.MaxWidthHeight,  
    MaxWidthHeight = 2000  
});
```

Si se toma la foto correctamente, se considerará que idioma se eligió en la configuración para valorar si la foto se tomó a un texto en español o a un texto en inglés. Al ser el código similar y solo cambiar en el parámetro que se introduce en el método *Init*, se mostrará únicamente el uso de la librería OCR para reconocimiento de texto en español.

```
if (!tesseractApi.Initialized)
{
    var init = await tesseractApi.Init("spa");
}

SKBitmap bitmap = SKBitmap.Decode(photo.GetStream());
SKBitmap target = BinaryImage(bitmap, 100);
SKImage image = SKImage.FromBitmap(target);
SKData encoded = image.Encode();
Stream stream = encoded.AsStream();

await tesseractApi.SetImage(stream);

var text = tesseractApi.Text;
```

El método *BinaryImage* se trata de un método creado en la misma clase, el cual se encarga de preprocesar la imagen para un mejor reconocimiento del texto, pasándola a imagen binaria, es decir, a blanco y negro.

Posteriormente, el texto obtenido se transmitirá por medio del sintetizador de voz. En el caso de no haber identificado ningún texto, se comunicará de ello al usuario.

### 6.5.2. Reconocimiento de señales

Dentro del conjunto de algoritmos de similitud de imágenes creados existen dos clases a considerar para esta implementación, *ImageSearchAlgorithm* y *SearchImage*. Por lo que en el archivo donde se desarrolla el reconocimiento de señales, se va a establecer una variable del primer tipo mencionado, *algorithm*, y otra del segundo tipo, *searchImage*.

Primeramente, se accede a la cámara del teléfono móvil para tomar la foto.

```
await CrossMedia.Current.Initialize();
var photo = await CrossMedia.Current.TakePhotoAsync(new
StoreCameraMediaOptions());
```

Una vez tomada la foto correctamente, se realiza el reconocimiento.

```
SKBitmap bitmap = SKBitmap.Decode(photo.GetStream());
var dstInfo = new SKImageInfo(400, 400);
bmpSearchImage = bitmap.Resize(dstInfo, SKFilterQuality.High);

centroidColor.Add(new SKColor(0xFFFF0000));
centroidColor.Add(new SKColor(0xFF0000FF));

algorithm.RunAlgorithm(bmpSearchImage, centroidColor.Count);
bmpSearchImageProcessed = algorithm.ProcessImage(bmpSearchImage,
centroidColor);

searchImage = new SearchImage(algorithm);

var items = searchImage.GetMostSimilarImage(
bmpSearchImageProcessed, centroidColor);

string mostSimilarImage = items.Item1;
double similarity = items.Item2;

Sign sign = await App.Database.GetSignAsync(mostSimilarImage);
```

Con relación al método *RunAlgorithm* de la variable *algorithm* mencionada anteriormente, se le pasa como parámetros la imagen tomada transformada a un objeto de la clase *SKBitmap* y el número de centroides asignados. Este método obtiene el valor de color RGB de cada pixel de la imagen, para posteriormente ejecutar un algoritmo de *Kmeans* que considere un array con todos los valores y el número de centroides.

En lo que se refiere al método *ProcessImage* se le pasa como parámetros la imagen tomada

transformada a un objeto de la clase *SKBitmap* y el conjunto de centroides. Este método se encarga de preprocesar la imagen en función del centroide más cercano al valor de color RGB de cada pixel.

El siguiente paso es obtener la señal de mayor similitud con la imagen, esto se consigue mediante la aplicación del método *GetMostSimilarImage* a la variable *searchImage*. A este método se le pasa como variables la imagen procesada y el conjunto de centroides. Realiza una comparación de las imágenes almacenadas en el proyecto y registradas en la base de datos con la imagen procesada y considerando los centroides, con el fin de obtener la señal que más se asemeja a la imagen y el porcentaje de similitud.

Posteriormente, se realizará la búsqueda de dicha señal en la base de datos para obtener su descripción y transmitirla por medio del sintetizador de voz. En el caso de no haber cumplido con un porcentaje mínimo de similitud, se considerará la señal capturada no registrada en la base de datos y se comunicará al usuario.

# 7

## Fase de pruebas y depuración

Una prueba se puede definir como una actividad en la que un sistema o un componente de este se ejecuta en ciertas circunstancias registrando los resultados obtenidos. Tras esto se realiza un proceso de evaluación en el que los resultados obtenidos se comparan con los resultados esperados para localizar así los fallos en el sistema. Estos fallos conducen al proceso de depuración en el que se tiene que identificar el error y corregirlo, pudiendo dar lugar a una nueva prueba.

El objetivo de las pruebas no es asegurar la ausencia de errores en un sistema, sino demostrar que existen defectos en él. En esta sección se desarrollarán tres niveles diferentes de pruebas: pruebas unitarias, pruebas de integración y pruebas de despliegue.

### 7.1. Pruebas unitarias

Como se indicó con anterioridad, la metodología utilizada es una metodología iterativa incremental, por lo que en cada iteración se ha ido creando una versión más compleja y con mayor funcionalidad. Es por ello que se han ido realizando pruebas unitarias a los métodos que se han ido creando en cada una de ellas, y así poder observar errores en estos para corregirlos posteriormente.

Los errores observados por cada método de cada página a lo largo del desarrollo de la aplicación se han ido corrigiendo con el fin de no encontrar defectos en el sistema. Finalmente se ha conseguido obtener un funcionamiento correcto de los mismos. A continuación se puede observar una tabla a modo de resumen de la ejecución correcta de los métodos de cada página.

<b>Página</b>	<b>Método</b>	<b>Salida</b>
SplashPage	Animate()	OK
IniLanguagePage	OnButtonClicked_Spain()	OK
	OnButtonClicked_US()	OK
IniVoicePage	OnButtonClicked_Acute()	OK
	OnButtonClicked_Deep()	OK
IniExplanationPage	OnButtonClicked_Yes()	OK
	OnButtonClicked_No()	OK
MainPage	OnButtonClicked_Start()	OK
	OnButtonClicked_Setting()	OK
SettingPage	OnButtonClicked_Language()	OK
	OnButtonClicked_Voice()	OK
	OnButtonClicked_Explanation()	OK
	OnButtonClicked_Exit()	OK
LanguagePage	OnButtonClicked_Spain()	OK
	OnButtonClicked_US()	OK
	OnButtonClicked_Exit()	OK
VoicePage	OnButtonClicked_Acute()	OK
	OnButtonClicked_Deep()	OK
	OnButtonClicked_Exit()	OK
ExplanationPage	OnButtonClicked_Yes()	OK
	OnButtonClicked_No()	OK
	OnButtonClicked_Exit()	OK
TextOrSignPage	OnButtonClicked_Text()	OK
	OnButtonClicked_Sign()	OK
RecapturePage	OnButtonClicked_Hear()	OK
	OnButtonClicked_Capture()	OK
	OnButtonClicked_Exit()	OK

Tabla 13: Resultados pruebas unitarias

## 7.2. Pruebas de integración

En esta sección, se van a realizar pruebas para las funcionalidades que se ofrecen en la aplicación. En estas pruebas se va a involucrar a varios componentes de la aplicación, pudiendo observar así que los componentes interactúan entre sí correctamente.

### 7.2.1. Completar la configuración inicial

En esta prueba se tendrá que acceder por primera vez a la aplicación y se comprobará que tras aparecer la pantalla de bienvenida comience la configuración inicial mostrándose la pantalla de configuración inicial de idioma, se seleccionará un idioma (p. ej., español), después se mostrará la pantalla de configuración inicial de voz, se seleccionará un tono de voz (p. ej., agudo), tras esto se mostrará la pantalla de configuración inicial de descripción, se seleccionará si se desea oír la descripción (p. ej., sí) y para finalizar, se mostrará la pantalla principal. Considerando, además, que se comente adecuadamente la información que debe transmitir el sintetizador de voz.

A modo de esquema, los pasos a seguir en esta prueba serán los siguientes:

Pantalla de bienvenida → Pantalla de configuración inicial de idioma → Selección de español → Pantalla de configuración inicial de voz → Selección de aguda → Pantalla de configuración inicial de descripción → Selección de oír descripción detallada → Pantalla principal

Tras la realización de la prueba, se pudo comprobar que se ejecutó correctamente y que los componentes que interactúan entre sí funcionaron adecuadamente.

### 7.2.2. Cambiar la configuración de idioma desde la página principal

En esta prueba se comprobará que, partiendo de la pantalla principal, al pulsar el botón de configuración se mostrará la pantalla principal de configuración, después se pulsará el botón de idioma, se mostrará la pantalla de configuración de idioma, se cambiará el idioma seleccionado y se mostrará la nueva selección del idioma. Considerando, además, que se comente adecuadamente la información que debe transmitir el sintetizador de voz.



A modo de esquema, los pasos a seguir en esta prueba serán los siguientes:

Pantalla principal → Pulsar botón de configuración → Pantalla principal de configuración → Pulsar botón de idioma → Pantalla de configuración de idioma → Cambiar selección de idioma → Nuevo idioma seleccionado marcado

Tras la realización de la prueba, se pudo comprobar que se ejecutó correctamente y que los componentes que interactúan entre sí funcionaron adecuadamente.

### **7.2.3. Cambiar la configuración de voz desde la página principal**

En esta prueba se comprobará que, partiendo de la pantalla principal, al pulsar el botón de configuración se mostrará la pantalla principal de configuración, después se pulsará el botón de voz, se mostrará la pantalla de configuración de voz, se cambiará el tono de voz seleccionado y se mostrará la nueva selección de voz. Considerando, además, que se comente adecuadamente la información que debe transmitir el sintetizador de voz.

A modo de esquema, los pasos a seguir en esta prueba serán los siguientes:

Pantalla principal → Pulsar botón de configuración → Pantalla principal de configuración → Pulsar botón de voz → Pantalla de configuración de voz → Cambiar selección de tono de voz → Nuevo tono de voz seleccionado marcado

Tras la realización de la prueba, se pudo comprobar que se ejecutó correctamente y que los componentes que interactúan entre sí funcionaron adecuadamente.

### **7.2.4. Cambiar la configuración de descripción desde la página principal**

En esta prueba se comprobará que, partiendo de la pantalla principal, al pulsar el botón de configuración se mostrará la pantalla principal de configuración, después se pulsará el botón de descripción, se mostrará la pantalla de configuración de descripción, se cambiará la selección anterior y se mostrará la nueva selección de la escucha de la descripción. Considerando, además, que se comente adecuadamente la información que debe transmitir el sintetizador de voz.

A modo de esquema, los pasos a seguir en esta prueba serán los siguientes:

Pantalla principal → Pulsar botón de configuración → Pantalla principal de configuración → Pulsar botón de descripción → Pantalla de configuración de descripción → Cambiar selección → Nueva selección marcada

Tras la realización de la prueba, se pudo comprobar que se ejecutó correctamente y que los componentes que interactúan entre sí funcionaron adecuadamente.

#### **7.2.5. Realizar un reconocimiento de texto desde la página principal y escuchar de nuevo la descripción del texto**

En esta prueba se comprobará que, partiendo de la pantalla principal, al pulsar el botón de inicio se mostrará la pantalla de elección de texto o señal, después se pulsará el botón de texto, se accederá a la cámara del teléfono móvil, después se realizará la foto al texto deseado y se aceptará la imagen capturada, tras esto se realizará el reconocimiento y se comentará la descripción del texto, seguidamente, se mostrará la pantalla de elección entre oír de nuevo la descripción o realizar una nueva captura, se seleccionará la opción de oír de nuevo y se escuchará nuevamente la descripción del texto. Considerando, además, que se comente adecuadamente la información que debe transmitir el sintetizador de voz.

A modo de esquema, los pasos a seguir en esta prueba serán los siguientes:

Pantalla principal → Pulsar botón de inicio → Pantalla de elección de texto o señal → Pulsar botón de texto → Acceso a la cámara del teléfono móvil → Capturar el texto deseado → Aceptación de la foto → Transmisión de la descripción del texto → Pantalla de elección entre oír de nuevo la descripción o realizar una nueva captura → Pulsar botón de oír de nuevo → Transmisión de la descripción del texto

Tras la realización de la prueba, se pudo comprobar que se ejecutó correctamente y que los componentes que interactúan entre sí funcionaron adecuadamente.

### **7.2.6. Realizar un reconocimiento de señal desde la página principal y seleccionar la opción de realizar una nueva captura**

En esta prueba se comprobará que, partiendo de la pantalla principal, al pulsar el botón de inicio se mostrará la pantalla de elección de texto o señal, después se pulsará el botón de señal, se accederá a la cámara del teléfono móvil, después se realizará la foto a la señal deseada y se aceptará la imagen capturada, tras esto se realizará el reconocimiento y se comentará la descripción de la señal, seguidamente, se mostrará la pantalla de elección entre oír de nuevo la descripción o realizar una nueva captura, se seleccionará la opción de realizar una nueva captura y se redireccionará de nuevo a la pantalla de elección de texto o señal para poder realizar una nueva captura. Considerando, además, que se comente adecuadamente la información que debe transmitir el sintetizador de voz.

A modo de esquema, los pasos a seguir en esta prueba serán los siguientes:

Pantalla principal → Pulsar botón de inicio → Pantalla de elección de texto o señal → Pulsar botón de señal → Acceso a la cámara del teléfono móvil → Capturar la señal deseada → Aceptación de la foto → Transmisión de la descripción de la señal → Pantalla de elección entre oír de nuevo la descripción o realizar una nueva captura → Pulsar botón de realizar una nueva captura → Pantalla de elección de texto o señal

Tras la realización de la prueba, se pudo comprobar que se ejecutó correctamente y que los componentes que interactúan entre sí funcionaron adecuadamente.

## **7.3. Pruebas de despliegue**

Para comprobar que la aplicación funciona correctamente se va a instalar en un dispositivo Android. Por medio de este, se van a realizar cada una de las pruebas ejecutadas anteriormente, tanto las pruebas unitarias como las pruebas de integración.

Con el fin de evitar comentar nuevamente la información acerca de las pruebas, se mostrará una tabla recogiéndolas y observando la ejecución correcta tanto de los métodos de cada página de la aplicación como de las acciones consideradas en las pruebas de integración.

<b>Prueba</b>	<b>Salida</b>
SplashPage.Animate()	OK
IniLanguagePage.OnButtonClicked_Spain()	OK
IniLanguagePage.OnButtonClicked_US()	OK
IniVoicePage.OnButtonClicked_Acute()	OK
IniVoicePage.OnButtonClicked_Deep()	OK
IniExplanationPage.OnButtonClicked_Yes()	OK
IniExplanationPage.OnButtonClicked_No()	OK
MainPage.OnButtonClicked_Start()	OK
MainPage.OnButtonClicked_Setting()	OK
SettingPage.OnButtonClicked_Language()	OK
SettingPage.OnButtonClicked_Voice()	OK
SettingPage.OnButtonClicked_Explanation()	OK
SettingPage.OnButtonClicked_Exit()	OK
LanguagePage.OnButtonClicked_Spain()	OK
LanguagePage.OnButtonClicked_US()	OK
LanguagePage.OnButtonClicked_Exit()	OK
VoicePage.VoicePageOnButtonClicked_Acute()	OK
VoicePage.OnButtonClicked_Deep()	OK
VoicePage.OnButtonClicked_Exit()	OK
ExplanationPage.OnButtonClicked_Yes()	OK
ExplanationPage.OnButtonClicked_No()	OK
ExplanationPage.OnButtonClicked_Exit()	OK
TextOrSignPage.OnButtonClicked_Text()	OK
TextOrSignPage.OnButtonClicked_Sign()	OK
RecapturePage.OnButtonClicked_Hear()	OK
RecapturePage.OnButtonClicked_Capture()	OK
RecapturePage.OnButtonClicked_Exit()	OK
Configuración inicial	OK
Cambio de idioma	OK
Cambio de tono de voz	OK
Cambio de escucha de descripción	OK
Reconocimiento de texto y oír de nuevo	OK
Reconocimiento de señal y recapturar	OK

Tabla 14: Resultados pruebas de despliegue



# 8

## Conclusiones y Líneas Futuras

### 8.1. Conclusiones

La realización de este trabajo ha permitido adquirir nuevos conocimientos sobre la tecnología de Xamarin y la aplicación de los lenguajes de programación C# y XAML.

Previamente, se tenían conocimientos del entorno de desarrollo Visual Studio, pero no de la creación de proyectos en Xamarin.Forms. Por lo que ha resultado un reto desarrollar la aplicación móvil en este marco de interfaz de usuario. Se ha tenido que profundizar en su estudio de forma teórica, comprendiendo su uso y aprendiendo sus lenguajes asociados. Es cierto que se conocían ambos lenguajes, C# y XAML, pero superficialmente.

No se había desarrollado antes ninguna aplicación móvil, ni para Android, ni para iOS, por lo que se tuvo que comprender la forma en la que Xamarin.Forms compartía un código común en estas plataformas para su implementación. Y en la instalación de paquetes, considerar las diferentes configuraciones que se tenían que realizar tanto al proyecto de Xamarin.Android como al de Xamarin.iOS.

Se han recuperado conceptos anteriormente estudiados de la ingeniería del software. Tras elegir la metodología a aplicar, se ha procedido al desarrollo del análisis de la aplicación, en el que se considera los requisitos y los casos de uso, el diseño de esta, en el que se muestra el diseño de la interfaz y la arquitectura del sistema, y la implementación de toda la aplicación.

La implementación de la librería OCR *Tesseract* para el reconocimiento de textos no ha resultado complicada, pero existen librerías OCR para C# más precisas que, por desgracia, no son compatibles con Xamarin.Forms, por lo que se tuvo que implementar un preprocesado de la imagen capturada para mejorar la eficacia de esta librería.

Con respecto al reconocimiento de señales, ha sido un verdadero reto, puesto que Xamarin.Forms no aportaba ninguna herramienta que facilitara dicho reconocimiento. Se tuvo que hacer una búsqueda y aprendizaje profundos en algoritmos que permitieran la comparación de imágenes y conseguir dar con las clases de Xamarin adecuadas para que fuese compatible cualquier implementación de estos.

Como se ha podido observar en lo comentado anteriormente, Xamarin.Forms no ha facilitado demasiado el principal objetivo de este proyecto de implementar en la aplicación el reconocimiento de textos y señales, ya que escasea en herramientas eficientes para ello por falta de compatibilidad. A pesar de esto, se han conseguido aplicar soluciones a estos inconvenientes.

Al desarrollar una aplicación enfocada a personas con deficiencia visual, se ha indagado en este problema actual y en las aplicaciones móviles que tienen disponibles para ayudarlas en su vida diaria. Con el avance tecnológico existente, es interesante el desarrollo de aplicaciones que puedan facilitar su día a día, ya que esta discapacidad afecta a un elevado número de personas y este aumenta con los años.

## **8.2. Líneas Futuras**

En vista a un trabajo futuro, se podría considerar la adición de nuevas señales a la base de datos de las cuales fuera interesante su reconocimiento para facilitar el día a día de los usuarios. Para ello, habría que considerar la optimización de los algoritmos de reconocimiento de imágenes, con el fin de reducir el tiempo de procesado y obtener resultados correctos.

Con respecto al reconocimiento de texto, podría trabajarse en mejorar el preprocesado de la imagen con el fin de conseguir mejores resultados considerando las limitaciones de *Tesseract*.

Como ampliación, sería interesante incorporar a la aplicación un reconocimiento de voz, para que efectuase la acción de pulsar botones cuando el usuario se lo transmitiese verbalmente, y ser así más favorable para las personas con deficiencia visual.





# Bibliografía

- BeMyEyes. (s.f.). *Be My Eyes - See the world together*. <https://www.bemyeyes.com/>
- Aira. (s.f.). *Download the Aira App, Try it for Free - Aira*. <https://aira.io/download-and-try-it/>
- LookTel. (s.f.). *LookTel – Instant Recognition Apps for Persons with Low Vision or Blindness*. <http://www.looktel.com/>
- TapTapSee. (s.f.). *TapTapSee - Blind and Visually Impaired Assistive Technology - powered by CloudSight.ai Image Recognition API*. <http://taptapseeapp.com/>
- LightDetector. (s.f.). *Light Detector Blind Accessibility App - iAccessibility Solutions for iOS Communications*. <https://www.iaccessibility.com/apps/blind/index.cgi/product?ID=42>
- KNFBReader. (s.f.). *OneStep Reader App features the best OCR. Turn print into speech or Braille instantly. iOS 3 now available. | KNFB Reader*. <https://knfbreader.com/>
- Tutorial de diagramas de casos de uso ( Guía con ejemplos ). (s.f.). <https://creately.com/blog/es/diagramas/tutorial-diagrama-caso-de-uso/>
- Déficit o deficiencia visual, tipos, limitaciones y tratamiento. (s.f.). <https://opticasanpedro.com/deficit-o-deficiencia-visual-tipos-limitaciones-y-tratamiento/>
- WebAIM: Screen Reader User Survey 8 Results. (s.f.). <https://webaim.org/projects/screenreadersurvey8/>
- Allen, J., Hunnicutt, M. S., Klatt, D. H., Robert C., c. A. & David, c. P. (1987). *From Text to Speech: The MITalk system* (C. C. U. Press, Ed.).
- Wayback Machine. (s.f.). [https://web.archive.org/web/20070323064223/http://www.adobe.com/resources/accessibility/best\\_practices/best\\_practices\\_acc\\_flash.pdf](https://web.archive.org/web/20070323064223/http://www.adobe.com/resources/accessibility/best_practices/best_practices_acc_flash.pdf)
- Hermes, D. (2015). *Xamarin Mobile Application Development: Cross-Platform C and Xamarin.Forms Fundamentals*.
- Hindriks, D. & Karlsson, J. (2020). *Xamarin.Forms Projects: Build multiplatform mobile apps and a game from scratch using C and Visual Studio 2019, 2nd Edition*.
- Herramientas de desarrollo, documentación técnica y ejemplos de código | Microsoft Docs. (s.f.). <https://docs.microsoft.com/es-es/>
- Wikipedia. (s.f.). <https://www.wikipedia.org/>

*30 aplicaciones, dispositivos y tecnologías para personas con discapacidad visual* - American Academy of Ophthalmology. (s.f.). <https://www.aaopt.org/salud-ocular/consejos/30-aplicaciones-dispositivos-y-tecnolog%C3%ADas-para-pe>

*Guía de arquitectura de apps | Desarrolladores de Android | Android Developers.* (s.f.). <https://developer.android.com/topic/architecture>

*Arquitectura en Capas.* (s.f.). <https://reactiveprogramming.io/blog/es/estilos-arquitectonicos/capas>

*Curso Básico de Xamarin: Implementando bases de datos locales con SQLite* – Luis Beltran. (s.f.). <https://www.luisbeltran.mx/2019/12/04/curso-basico-de-xamarin-implementando-bases-de-datos-locales-con-sqlite/>

*Catálogo | BOR.* (s.f.). <http://bor.es/catalogo/>

*OCR Using Tesseract In C.* (s.f.). <https://www.c-sharpcorner.com/article/ocr-using-tesseract-in-C-Sharp/>

*halkar/Tesseract.Xamarin: Tesseract OCR wrapper for Xamarin.* (s.f.). <https://github.com/halkar/Tesseract.Xamarin>

*The step by step set up of Media Plugin for Xamarin.Forms!* | ÇøñfuzëD SørucëÇødë. (s.f.). <https://theconfuzedsourcocode.wordpress.com/2020/01/28/the-step-by-step-set-up-of-media-plugin-for-xamarin-forms/>

*Media Plugin Usage.* (s.f.). <https://gist.github.com/ericbrunner/5a4c76cb5cbabc3a992c5a97a216c81c>

*Image Search using C with K-Means algorithm - CODE-AI.* (s.f.). <https://devindeep.com/image-search/>

*Microsoft Word - Dinámicas.doc - get.php.* (s.f.). <http://www.lsi.us.es/docencia/get.php?id=361>

# Apéndice A

# Manual de Usuario

Al entrar en la aplicación, se mostrará la pantalla de bienvenida durante unos segundos antes de redireccionarse a la siguiente pantalla.



Figura 14: Pantalla de Bienvenida

Si no es la primera vez que se accede a la aplicación, la siguiente pantalla que se mostrará será la pantalla principal, la cual contiene las opciones de acceder a la configuración general y de iniciar el reconocimiento de textos y señales, por medio de los botones “Configuración” y “Pulsar para iniciar” respectivamente.

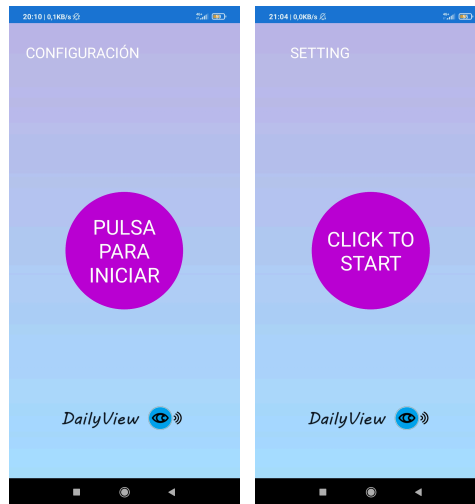


Figura 15: Pantalla Principal

## A.1. Configuración inicial

Si es la primera vez que se accede a la aplicación tras su instalación, habrá que establecer una configuración inicial, la cual se mostrará después de la pantalla de bienvenida.

En primer lugar, se mostrará la pantalla de configuración inicial de idioma, en la que se tendrá que elegir el idioma deseado entre español o inglés. Se comunicará por medio de voz las opciones disponibles y el botón a pulsar para cada una de ellas.



Figura 16: Pantalla de Configuración Inicial de Idioma

Después, se mostrará la pantalla de configuración inicial de voz, en la que se tendrá que elegir el tono de voz deseado entre agudo o grave. Se comunicará por medio de voz las opciones disponibles y el botón a pulsar para cada una de ellas.

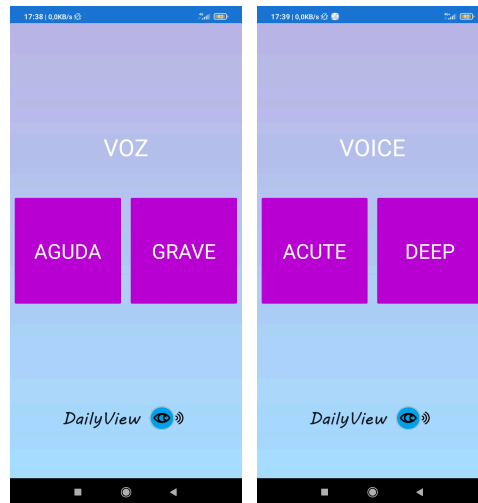


Figura 17: Pantalla de Configuración Inicial de Voz

Para finalizar la configuración inicial, se mostrará la pantalla de configuración inicial de descripción, en la que se tendrá que elegir entre escuchar una descripción detallada de la interfaz de la aplicación o no. Se comunicará por medio de voz las opciones disponibles y el botón a pulsar para cada una de ellas. Hasta familiarizarse con la aplicación es aconsejable elegir la escucha de la descripción.

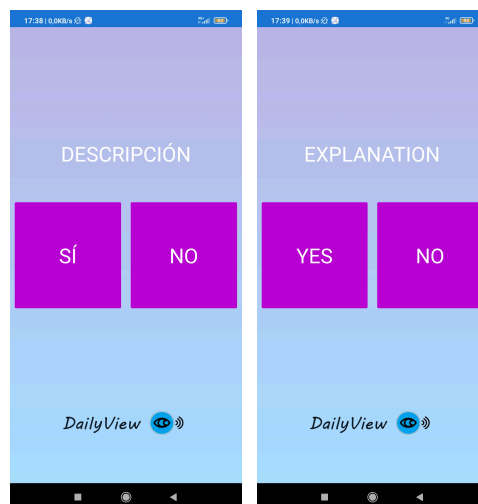


Figura 18: Pantalla de Configuración Inicial de Descripción

Tras la configuración inicial, se redireccionará a la pantalla principal.

## A.2. Configuración general

Si en la pantalla principal se pulsa en el botón “Configuración”, se mostrará la pantalla principal de configuración. En ella se puede acceder a las configuraciones de idioma, voz y descripción.

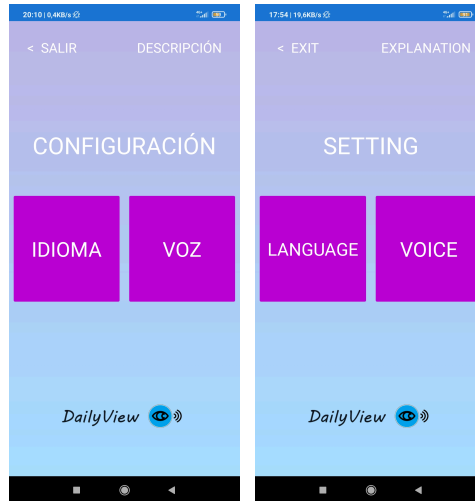


Figura 19: Pantalla Principal de Configuración

Si se pulsa el botón “Idioma”, se mostrará la pantalla de configuración de idioma, en la que se podrá cambiar el idioma seleccionado pulsando el botón deseado.

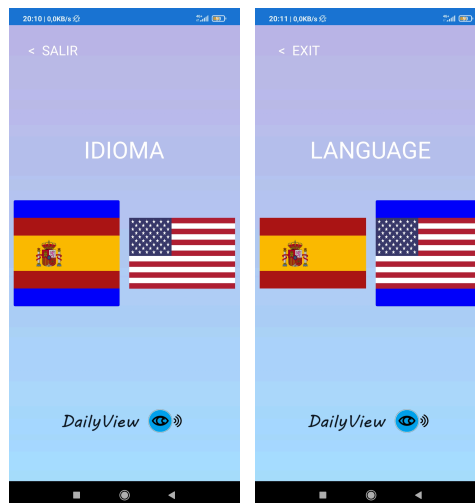


Figura 20: Pantalla de Configuración de Idioma

Si se pulsa el botón “Voz”, se mostrará la pantalla de configuración de voz, en la que se podrá cambiar el tono de voz seleccionado pulsando el botón deseado.

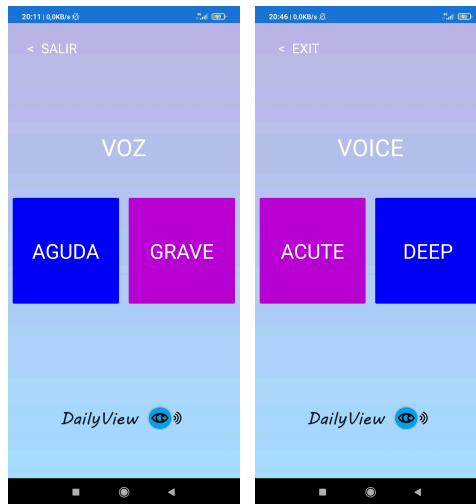


Figura 21: Pantalla de Configuración de Voz

Si se pulsa el botón “Descripción”, se mostrará la pantalla de configuración de descripción, en la que se podrá cambiar la selección de escucha de descripción pulsando el botón deseado.

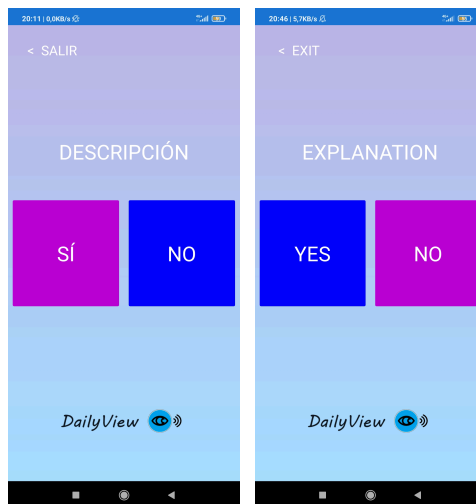


Figura 22: Pantalla de Configuración de Descripción

### A.3. Reconocimiento de texto o señal

Si en la pantalla principal se pulsa en el botón “Pulsar para inicial”, se mostrará la pantalla de elección de texto o señal. En ella se elegirá el reconocimiento que se desea realizar.



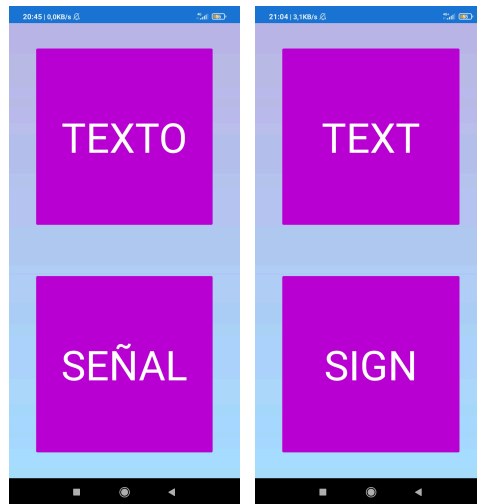


Figura 23: Pantalla de Elección de Texto o Señal

Tras pulsar cualquiera de las dos opciones, se accederá a la cámara del teléfono móvil para realizar la captura al texto o la señal que se desee. Una vez realizada la foto, se tendrá que aceptar y se procederá al reconocimiento. Aunque también se podrá denegar la foto realizada y volver a capturar una nueva.

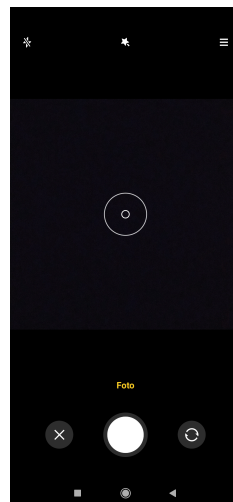


Figura 24: Acceso Cámara Teléfono Móvil



Figura 25: Aceptación o Cancelación de Foto

Tras comunicarse por medio de voz la descripción del texto o la señal capturados, se mostrará una pantalla en la que se encontrarán dos opciones, oír de nuevo la descripción o realizar una nueva captura.

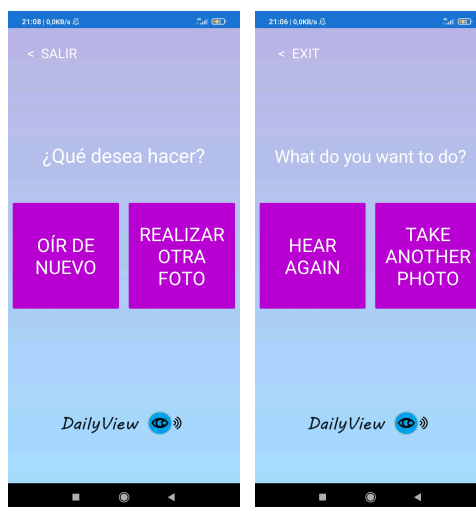


Figura 26: Pantalla de Nueva Escucha o Captura

Si se pulsa el botón de realizar una nueva captura, se podrá volver a elegir entre texto o señal.



UNIVERSIDAD  
DE MÁLAGA

| [uma.es](http://uma.es)

E.T.S de Ingeniería Informática  
Bulevar Louis Pasteur, 35  
Campus de Teatinos  
29071 Málaga

E.T.S. DE INGENIERÍA INFORMÁTICA