



UNIVERSIDAD PONTIFICIA DE SALAMANCA  
FACULTAD DE INFORMÁTICA  
Grado en Informática

Trabajo Fin de Grado

**Estudio, desarrollo e implementación de  
una aplicación de ayuda a personas con  
discapacidad, basada en el  
reconocimiento de imágenes.**

Bogdán Gaytán Martín

Dra. Ana María Feroso García  
Director

Salamanca, julio de 2021





UNIVERSIDAD PONTIFICIA DE SALAMANCA  
FACULTAD DE INFORMÁTICA  
Grado en Informática

Trabajo Fin de Grado

# **Estudio, desarrollo e implementación de una aplicación de ayuda a personas con discapacidad, basada en el reconocimiento de imágenes.**

Bogdán Gaytán Martín

Dra. Ana María Feroso García  
Director

Salamanca, julio de 2021



## Resumen

Mediante este trabajo se pretende implementar, mediante técnicas de inteligencia artificial aplicadas al reconocimiento de imágenes, una aplicación móvil de ayuda y apoyo a personas con discapacidad intelectual, con el objetivo de aumentar su autonomía en el hogar, especialmente en el desempeño de sus tareas domésticas y vida cotidiana. El desarrollo ha implicado el aprendizaje de técnicas de visión artificial e interfaz multicanal, así como el análisis y estudio de la problemática de las personas con discapacidad intelectual para adaptar el recurso tecnológico implementado, a sus necesidades.

La aplicación móvil se ha desarrollado con sistema operativo Android. El funcionamiento de esta aplicación consiste en primer lugar en el análisis de una imagen. Esta puede ser tanto de un electrodoméstico como de cualquier objeto de la vida cotidiana. El análisis se realiza mediante la división de la imagen en un mapa de bits en dos dimensiones. Para obtener la información de la imagen, se compara este mapa de bits con la información almacenada en el modelo de datos. Una vez realizada la comparación, se muestra el resultado que incluye la imagen realizada y una descripción de su funcionamiento. La aplicación incluye la funcionalidad de transformar el texto de la descripción en voz.

La finalidad de este proyecto consiste en el diseño de una aplicación que permita el aprendizaje a personas con una discapacidad intelectual. De esta manera, se facilita una mayor autonomía personal en el desempeño de las labores del hogar.

## Abstract

This work aims to implement, through artificial intelligence techniques applied to image recognition, a mobile application to help and support people with intellectual disabilities, with the aim of increasing their autonomy at home, especially in the performance of their domestic tasks and everyday life. The development has involved learning artificial vision and multichannel interface techniques, as well as the analysis and study of the problems of people with intellectual disabilities to adapt the implemented technological resource to their needs.

The mobile application has been developed with the Android operating system. The operation of this application consists in the first place in the analysis of an image. This can be both an electrical appliance and any object of everyday life. Analysis is done by dividing the image into a two-dimensional bitmap. To obtain the information from the image, this bitmap is compared with the information stored in the data model. Once the comparison is made, the result is displayed, including the image made and a description of its operation. The application includes the functionality to transform the text of the description into speech.

The purpose of this project consists of the design of an application that allows people with an intellectual disability to learn. In this way, greater personal autonomy is facilitated in the performance of household chores.

## Descriptores

Reconocimiento de Imágenes, Visión artificial, Inteligencia Artificial, Android







## Índice

1. Introducción.....	9
1.1 Motivación.....	9
1.2 Estructura de la memoria.....	9
2. Objetivos del proyecto.....	11
2.1 Objetivos técnicos.....	11
2.2 Objetivos personales.....	11
3. Antecedentes.....	13
4. Bases teóricas.....	15
4.1 Inteligencia artificial.....	15
4.1.1 Visión Artificial.....	15
4.1.2 Microsoft Azure Custom Vision.....	16
4.2 Intérprete.....	20
4.3 Conjunto de datos.....	21
4.4 Entorno de desarrollo.....	21
4.5 Text To Speech.....	21
5. Desarrollo de aplicación móvil.....	23
5.1 Especificación de requisitos.....	23
5.1.1 Requisitos generales.....	23
5.1.2 Requisitos funcionales.....	23
5.2 Casos de uso.....	23
5.3 Modelo de datos.....	24
5.4 Implementación.....	24
5.4.1 Interfaz.....	24
5.4.2 Funciones.....	27
6. Resultado.....	31
6.1 Instalación.....	31
6.2 Funcionamiento.....	32
7. Conclusiones.....	38
8. Líneas futuras.....	40
9. Bibliografía.....	42



## Tabla de Figuras

Figura 1: New Project.....	17
Figura 2: Cargar Imágenes. ....	18
Figura 3: Etiquetado.....	18
Figura 4: Entrenar. ....	19
Figura 5: Resultados entrenamiento.....	19
Figura 6: Exportar modelo de datos ....	20
Figura 7- Interfaz Main Activity.....	25
Figura 8: Interfaz Adver ....	26
Figura 9: Interfaz Menu main ....	26
Figura 10: Interfaz info.....	27
Figura 11: Funciones.....	28
Figura 12: Drive apk. ....	31
Figura 13: Descarga.....	32
Figura 14: Inicio.....	33
Figura 15: Permisos.....	33
Figura 16: Pantalla principal. ....	34
Figura 17: Selección de imagen. ....	34
Figura 18: Resultado. ....	35
Figura 19: Camera.....	35
Figura 20: Tomar foto. ....	36
Figura 21: Resultado camera. ....	37



# 1. Introducción

---

En este primer capítulo se hablará de algunos factores que me han impulsado a escoger este tema, así como la distribución del contenido desarrollado a lo largo de esta memoria.

## 1.1 Motivación.

La elección de este trabajo se debe al interés de demostrar como la inteligencia artificial es capaz de no tan solo solucionar grandes problemas matemáticos, sino también de facilitar y ayudar a las personas que más lo necesitan a ser más autónomas. La inteligencia artificial ya no es algo del futuro, es del presente que vivimos. Está presente en muchas herramientas que utilizamos, como es el asistente de voz o incluso las recomendaciones. Sin embargo, en este proyecto nos centraremos en una parte más concreta de esta tecnología, la visión artificial.

La escasez de conocimientos previos en este campo me supone un reto personal y una motivación para investigar, desarrollar y aprender más sobre inteligencia artificial. Ha sido esto lo que me ha llevado a elegir este proyecto.

## 1.2 Estructura de la memoria

El contenido de la memoria se estructura de la siguiente manera:

En el capítulo dos se explican los objetivos a nivel técnico y personal. Los técnicos son aquellos que marcan la estructura funcional del proyecto. Los objetivos personales son los que se reúnen con el fin de aplicar los conocimientos adquiridos.

En el capítulo tres se exponen los antecedentes de este proyecto. En él se describe la necesidad de desarrollar una herramienta que posibilite un problema no resuelto.

En el capítulo cuatro se explican en profundidad las bases teóricas. Se desarrollan las herramientas empleadas y los conocimientos necesarios para este proyecto.

En el quinto capítulo se explica el desarrollo de la aplicación. Indicando los pasos necesarios para su implementación.

El sexto capítulo muestra una breve guía de instalación de la aplicación móvil y los resultados obtenidos.

El capítulo séptimo refleja las conclusiones obtenidas una vez finalizado el proyecto.

En el octavo capítulo se indican posibles mejoras de cara a versiones futuras.



## 2. Objetivos del proyecto

---

El segundo capítulo trata a cerca de la finalidad tanto a nivel personal como a nivel técnico de este proyecto.

El objetivo principal de este proyecto consiste en facilitar la labor cotidiana a personas con una discapacidad intelectual. Para ello se han definido una serie de objetivos a cumplir durante la realización de la aplicación.

### 2.1 Objetivos técnicos

El objetivo principal de este proyecto es el estudio y su posterior aplicación práctica de técnicas de visión artificial aplicadas al reconocimiento de imágenes. El objetivo de la aplicación práctica es a su vez poner estas técnicas al servicio de las personas con discapacidad intelectual para apoyarlas en la mejora de su autonomía personal.

Para ello se precisa cumplir con objetivos secundarios como son:

- El estudio de diferentes técnicas y herramientas de inteligencia artificial en el ámbito de visión artificial para el reconocimiento de imágenes, así como técnicas de conversión de interfaz multicanal (texto y voz).
- Reconocimiento eficiente de imágenes: Será necesario lograr una correcta recopilación de datos en forma de imagen para obtener un buen resultado. Esto implica una mayor precisión a la hora del procesamiento de imágenes.
- Diseñar una aplicación móvil sencilla, utilizable y adaptada a los usuarios destinatarios, las personas con discapacidad intelectual, en el ámbito de la mejorar de su autonomía en el hogar.

### 2.2 Objetivos personales

Gracias a la implementación de esta aplicación se pone en práctica lo aprendido a lo largo del grado. Además de adquirir nuevos conocimientos tanto de programación como de herramientas.

- Puesta en práctica de los conocimientos adquiridos durante el curso académico: en el desarrollo de esta aplicación se ponen también en práctica todos los conocimientos aprendidos durante la carrera. Entre ellos se encuentra el lenguaje de programación de Java.
- Aprendizaje de nuevas herramientas: Para la implementación de la aplicación ha sido necesario la previa adquisición de nuevas herramientas. Estas abarcan la Api de Tensor Flow y la interfaz de aplicación de programa (API) de Text to Speech. Además, el aprendizaje del funcionamiento de herramientas dedicadas a la visión artificial, correspondiente al campo de la inteligencia artificial desarrollada durante este proyecto.

- Familiarización con el usuario final: Para lograr una correcta implementación de la aplicación y conseguir satisfacer a los usuarios finales que la utilizarán, se llevarán a cabo una serie de entrevistas. La finalidad de esto consiste en entender mejor las necesidades y dificultades del día a día de las personas con discapacidades intelectuales.



## 3. Antecedentes

---

A lo largo de este capítulo se hablará acerca de la necesidad de esta aplicación, algunas de las herramientas similares y la relevancia de esta.

La autonomía personal es uno de los aspectos fundamentales en el desarrollo de las personas. Esto hace que el proceso de enseñanza-aprendizaje de las habilidades de autonomía para la vida diaria sea clave, y este proceso de aprendizaje que se desarrolla a lo largo de las distintas etapas de la vida está influenciado por muchos factores, entre ellos el escolar, el familiar, etc.

Entre las habilidades de autonomía personal incluye actividades básicas relacionadas con el cuidado personal o con el funcionamiento físico y mental, pero también otras más complejas, aunque igualmente relevantes como llamar por teléfono, fregar, hacer la comida, lavar la ropa, manejar distintos electrodomésticos, etc. La relevancia de estas habilidades es indudable, puesto que hacen posible que las personas se desenvuelven de forma independiente tanto en el cuidado personal como en la vida diaria.

Algunas de las herramientas tecnológicas ya existentes y destacables para la ayuda de las personas con discapacidad intelectual son: ABAPLANET, basada en el análisis aplicado al comportamiento; SOY CAPAZ, la cual ayuda a organizar las actividades; PICAA, esta aplicación incluye actividades de matemáticas, lenguaje, conocimiento del entorno, autonomía y habilidades sociales. [1]

No obstante, debido a la situación generada por la COVID-19, muchas de estas personas han tenido dificultades a la hora de realizar labores domésticas. Con el propósito de facilitar a los usuarios los conocimientos para poder desempeñar estas tareas surge la aplicación móvil VIA FACIL. Esta aplicación móvil resulta innovadora, pues emplea la visión artificial como herramienta para la ayuda de las personas con discapacidad intelectual. Este concepto de visión artificial, un campo concreto de la inteligencia artificial (IA), se desarrollará en capítulos posteriores.

La aplicación destaca por su sencillez y su funcionamiento intuitivo para los usuarios. Para desarrollar la aplicación se ha contado con la ayuda del equipo docente de ASPRODES, quienes han proporcionado información de calidad para la recopilación de datos. Gracias a esto, la aplicación posee una mayor calidad a la hora de reconocer las imágenes y una guía sencilla que describe el funcionamiento de los objetos reconocidos. Los usuarios finales de esta herramienta, tras haber realizado pruebas, han declarado que les es útil y muy práctica en muchas de las tareas del hogar. Por ello, es grato saber que el proyecto resulta de utilidad porque ayuda a las personas a llevar una vida más autónoma y lograr así mayor estabilidad emocional.



## 4. Bases teóricas

---

En este capítulo se desarrollará más en profundidad qué son y cómo funcionan las tecnologías empleadas en la implementación de este proyecto. También se hablará de las alternativas que se han planteado y el por qué se han escogido o rechazado. Así como las diferentes partes que son requeridas para el correcto funcionamiento de la aplicación.

Las herramientas necesarias son:

- Un modelo de datos basado en inteligencia artificial, capaz de reconocer y clasificar las imágenes.
- Un intérprete capaz de transformar los datos de la inteligencia artificial en información reconocible por el usuario final.
- Conjunto de datos donde se especifiquen las instrucciones de uso, adaptadas a su comprensión por parte de las personas con discapacidad intelectual, de cada objeto doméstico a reconocer.
- Un entorno de desarrollo para programar la aplicación móvil.
- Una herramienta que permita transformar el texto en audio, Text To Speech, que facilite la comprensión a sus usuarios, de las instrucciones recibidas para utilizar cada objeto considerado.

### 4.1 Inteligencia artificial

Podemos definir la inteligencia artificial como un programa de computación diseñado para realizar determinadas operaciones que se consideran propias de la inteligencia humana, como el autoaprendizaje [2]. No obstante, tan solo precisamos para este trabajo, la capacidad de procesar imágenes. Esta parte de la inteligencia artificial o IA es conocida como visión artificial.

#### 4.1.1 Visión Artificial

Para implementar la visión artificial en una aplicación móvil, se precisa de un conjunto de datos que contenga un esquema genérico de información. Al proceso de recopilación y procesamiento de dichos datos se le denomina entrenamiento de la inteligencia artificial. Sin embargo, dicho entrenamiento a menudo puede suponer un gran costo, tanto en forma de tiempo como de infraestructura. Las grandes inteligencias artificial se entrenan empleado súper ordenadores o equipos destinados a ello. Debido a esto, si deseamos crear una inteligencia artificial funcional, es decir, con un conjunto de datos eficientes, es preferible emplear herramientas destinadas a ello como son Google Cloud, Techable Machine o Microsoft Azure Custom Vision, ya que realizar este proceso en nuestro propio equipo, si este no posee los componentes necesarios, podría suponer mucho tiempo y no se garantiza conseguir resultados adecuados.

A la hora de emplear la visión artificial en un dispositivo móvil, el conjunto de datos, o modelo de datos, puede ser tratado de forma remota, alojando dicha información en la nube y accediendo a esta a través del Smartphone. Aunque esto puede suponer un inconveniente, ya que la aplicación móvil dejaría de funcionar durante el periodo de tiempo en el que el usuario no disponga de conexión a internet. Otro inconveniente relacionado con almacenar el modelo de datos de datos en la nube, es la limitación de velocidad, es decir, la rapidez para obtener la

respuesta que se desea se encuentra en función a la velocidad y calidad de la conexión a internet.

Debido a esto, se ha optado por almacenar dicho conjunto de datos previamente entrenado por la inteligencia artificial en un fichero, almacenando éste en el dispositivo móvil del usuario. Gracias a esto, se permite al usuario disponer de la aplicación móvil en todo momento y lugar sin necesidad de una conexión a internet. Mejorando así su eficiencia y velocidad de procesado.

#### **4.1.1.1 Generación de modelo de datos**

Como se ha mencionado en el apartado anterior, el conjunto de datos necesario para la implementación de la visión artificial, debe ser entrenado. Este proceso consiste en recopilar diferentes imágenes de un objeto en concreto, e indicar a la inteligencia artificial el nombre de dicho objeto. Cuanto mayor sea el numero de imágenes, mejor resultado se obtendrá.

La inteligencia artificial, procesa todas las imágenes de un mismo objeto, es decir, con la misma etiqueta, en este caso el nombre del objeto. Este proceso consiste en separar cada imagen en un mapa de bits, obteniendo un patrón común en todas las imágenes bajo una misma etiqueta. Una vez procesadas todas las etiquetas con sus respectivas imágenes, se crea un esquema donde se indica la etiqueta y su correspondiente mapa de bits, dicho mapa contiene el patrón en forma de bits.

Por ultimo se comprime el esquema, en un fichero o modelo de datos.

Para este proceso se han estudiado diversas herramientas como son Google Cloud, Techable Machine y Microsoft Azure Custom Vision. También se ha planteado la opción de tomar un modelo previamente entrenado, no obstante, se ha descartado esta opción ya que muchos objetos que se precisaban no estaban incluidos y a demás contenían muchos objetos innecesarios para el desarrollo de la aplicación móvil, ralentizando así la aplicación.

Se ha descartado la herramienta de Google ya que no dispone de software libre. Esto supone un gran inconveniente teniendo en cuenta que la aplicación móvil pretende ser gratuita sin cargos.[3]

Techable Machine fue descartada ya que, al realizar las primeras pruebas, estas habían resultado correctas, sin embargo, al realizar las pruebas en un entorno más cerrado, las pruebas no habían sido satisfactorias. [4]

La opción elegida ha resultado ser Microsoft Azure Custom Vision, la cual arrojaba buenos resultados además de ser gratuita.

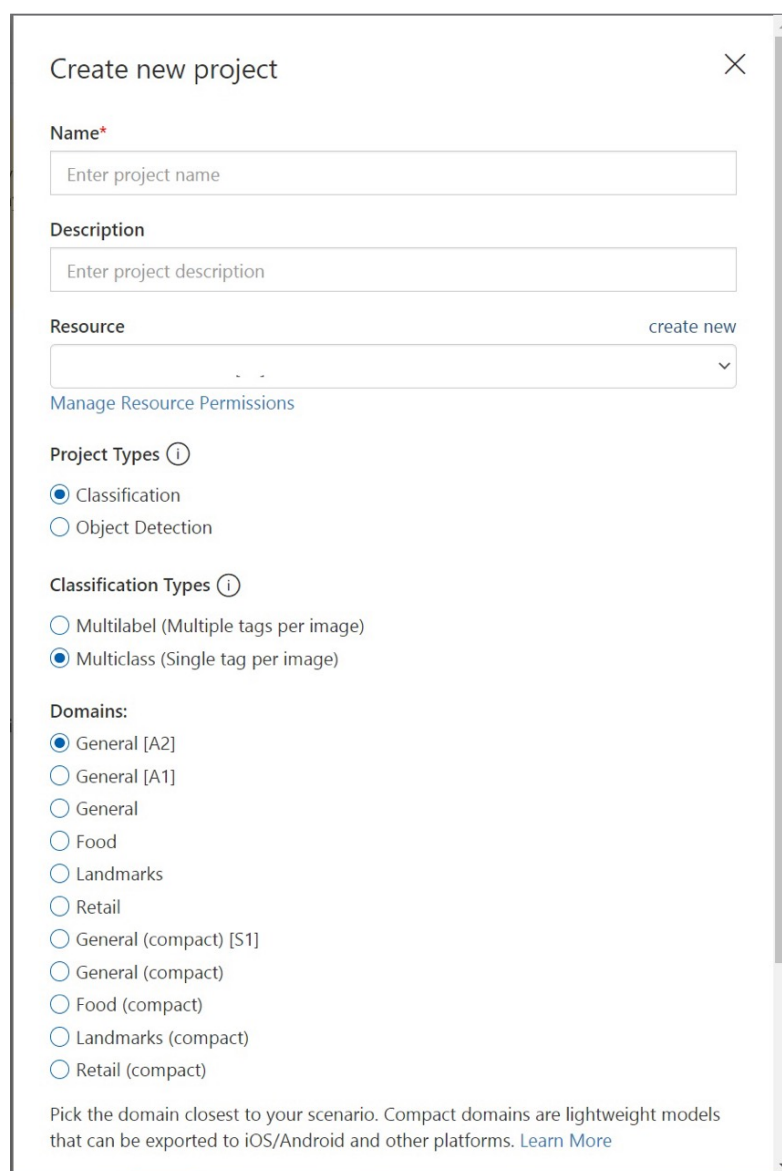
#### **4.1.2 Microsoft Azure Custom Vision**

En este apartado se explicará el proceso de entrenamiento de un modelo de datos empleando la herramienta de Microsoft Azure Custom Vision.

#### 4.1.2.1 Conjunto de imágenes

Previamente a utilizar la herramienta de entrenamiento de una inteligencia artificial de Microsoft, debemos crear una cuenta que nos identifique. Microsoft proporciona diferentes tipos de suscripciones para poder emplear su herramienta. Dicha suscripción define el número total de imágenes y proyectos que se desean realizar, para este proyecto se ha seleccionado la suscripción gratuita, mediante ésta, tenemos la opción de crear un proyecto con un número ilimitado de etiquetas y un máximo de cinco mil imágenes. [10]

Una vez tengamos una cuenta con una suscripción, creamos un nuevo proyecto rellenando la información necesaria como se muestra en la figura 1.



Create new project

Name\*

Enter project name

Description

Enter project description

Resource [create new](#)

[Manage Resource Permissions](#)

Project Types ⓘ

☒ Classification

☐ Object Detection

Classification Types ⓘ

☐ Multilabel (Multiple tags per image)

☒ Multiclass (Single tag per image)

Domains:

☒ General [A2]

☐ General [A1]

☐ General

☐ Food

☐ Landmarks

☐ Retail

☐ General (compact) [S1]

☐ General (compact)

☐ Food (compact)

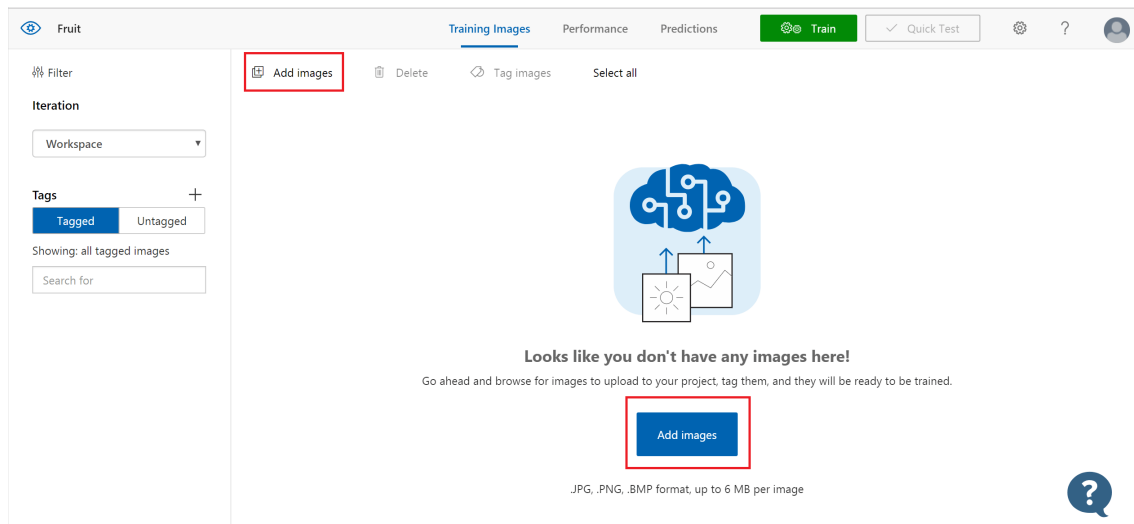
☐ Landmarks (compact)

☐ Retail (compact)

Pick the domain closest to your scenario. Compact domains are lightweight models that can be exported to iOS/Android and other platforms. [Learn More](#)

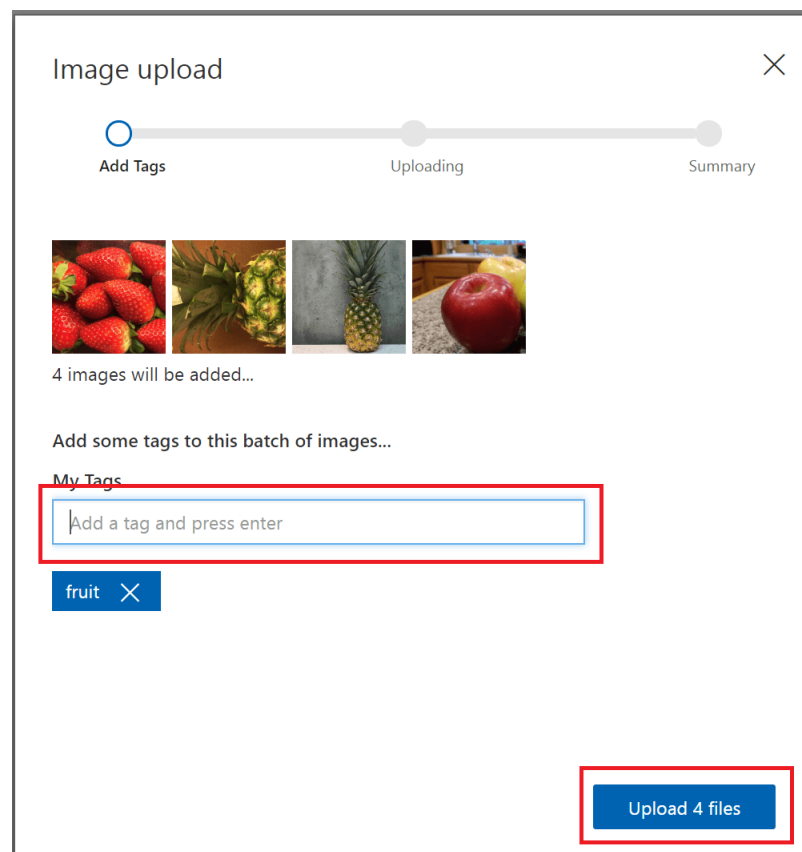
Figura 1: New Project

Una vez creado el proyecto procederemos a recopilar y cargar las imágenes deseadas.



**Figura 2: Cargar Imágenes.**

Completado el proceso de carga de las imágenes al proyecto, debemos etiquetarlas, es decir, identificar mediante un nombre el objeto. Ver figura 3.



**Figura 3: Etiquetado.**

#### 4.1.2.2 Entrenamiento

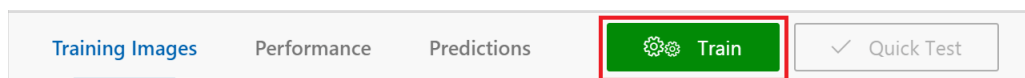


Figura 4: Entrenar.

Para finalizar debemos presionar el botón Train o “Entrenar” como se indica en la figura 19. Cuando finalice el proceso de entrenamiento, se mostrarán los resultados obtenidos en función del conjunto de imágenes que se han proporcionado.

La figura 20 muestra los resultados. En ellos se indica precisión y la coincidencia. La precisión indica la fracción de las clasificaciones identificadas que fueron correctas. Por otro lado, el valor de la coincidencia (Recall), indica la fracción de las clasificaciones reales que se identificaron correctamente.

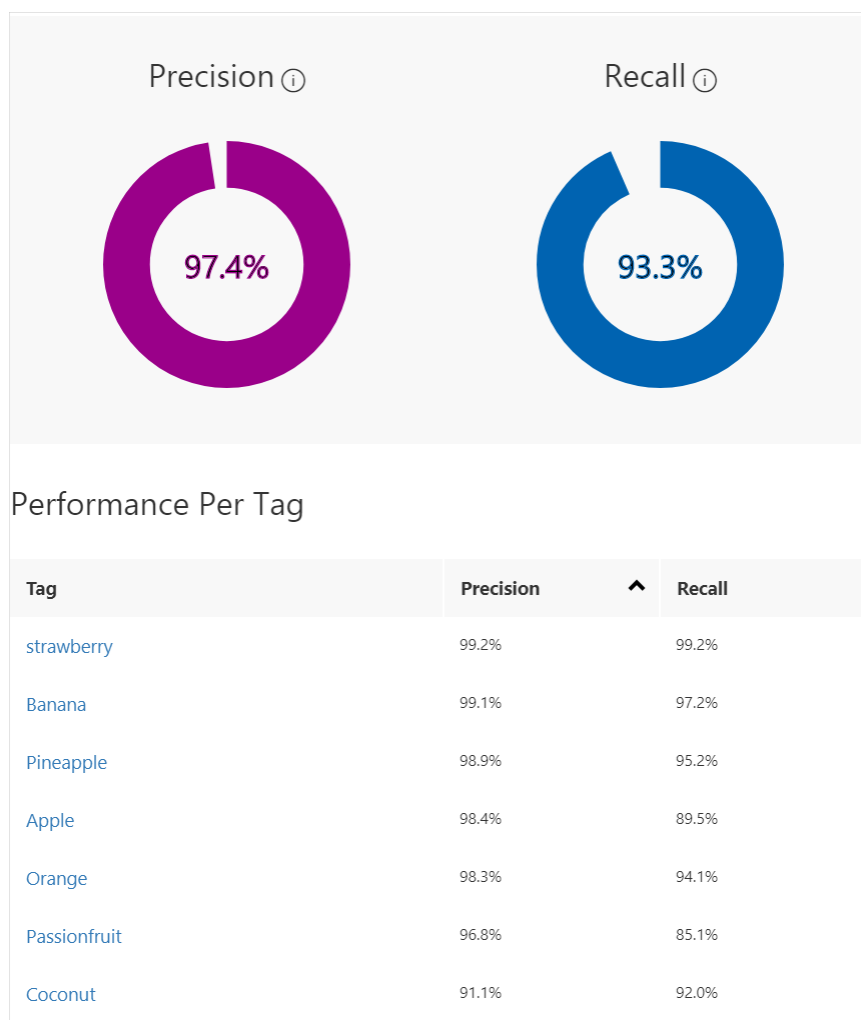


Figura 5: Resultados entrenamiento

### 4.1.2.3 Exportar Modelo de Datos

Una vez entrenado el modelo, debemos exportar dicho modelo. Gracias a este proceso, podremos utilizar el reconocimiento de imágenes sin necesidad de una conexión a Internet.

Este proceso es muy sencillo, consiste en presionar el botón de exportar y seleccionar el formato deseado, en nuestro caso Tensorflow Lite. Este proceso se muestra en la figura 6. [11]

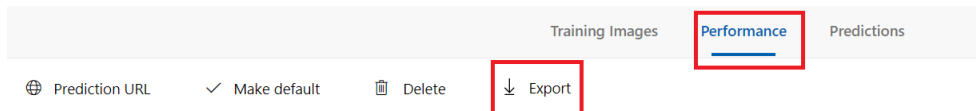


Figura 6: Exportar modelo de datos

## 4.2 Intérprete

El intérprete que se ha utilizado es una api o “interfaz de programación de aplicación” [8]. Dicha API será la encargada de procesar las imágenes empleando el modelo de datos del apartado anterior y obtener el resultado.

La api seleccionada para esta tarea es “org.tensorflow.lite”. Esta interfaz de programación de aplicación, será la encargada de obtener el resultado a partir de una imagen de entrada.

El término “inferencia” se refiere al proceso de ejecutar un modelo de Tensor Flow Lite en el dispositivo para hacer predicciones basadas en datos de entrada. Para realizar una inferencia con un modelo de Tensor Flow Lite, debes ejecutarlo a través de un intérprete. El intérprete de Tensor Flow Lite está diseñado para ser ágil y rápido. El intérprete utiliza un orden de gráficos estáticos y un asignador de memoria personalizado (menos dinámico) para garantizar una carga mínima, inicialización y latencia de ejecución.

La interpretación de una imagen sigue los siguientes pasos:

- Carga de modelo. Debemos cargar el modelo de datos que hemos entrenado previamente. Este proceso se explica en el apartado 4.1.2 .
- Transferencia de datos y ejecución de inferencia. A menudo la imagen de datos no es compatible con el formato de datos esperado por el modelo, por ello la api se encarga de transformar la imagen, forma, color o incluso rotación.
- Interpretación de salida. En este ultimo paso, el interprete se encarga de comparar la imagen de entrada con los diferentes valores del modelo de datos que previamente hemos cargado. En el proceso de comparación se asigna un valor decimal (Float), el cual describe la probabilidad de que la imagen de entrada sea la misma a la de un objeto del modelo.



### 4.3 Conjunto de datos

Con el objetivo de ayudar a las personas con discapacidad intelectual, la aplicación no muestra tan solo el nombre del objeto que se desea identificar, sino también una descripción detallada acerca de su uso. La recopilación de dicha información se ha llevado a cabo con la ayuda del equipo docente de ASPRODES, facilitando los conocimientos imprescindibles para generar un conjunto de datos que les ayuda a desempeñar con autonomía y autosuficiencia las labores del hogar.

La información previamente mencionada se almacena en un fichero “values” con extensión “JSON”. Gracias a esta estructura el programa es capaz de acceder de forma rápida, sencilla y eficaz a los datos que se deben mostrar.

### 4.4 Entorno de desarrollo

La programación de la aplicación móvil, implementada actualmente solo para Android, se ha realizado en Android Studio, ya que este lenguaje proporciona mayor facilidad y compatibilidad con las librerías que se utilizan en este proyecto y explicadas en capítulos posteriores. La otra opción para el desarrollo de la aplicación fue Xamarin Forms, la cual permite desarrollar la aplicación para diferentes tipos de dispositivos, tanto IOS como Android. [5]

Por otro lado, Android Studio, otra razón para su elección, permite la programación tanto en Kotlin como en Java. He escogido Java como lenguaje de programación, debido a los conocimientos adquiridos a lo largo de la carrera siendo para mí un lenguaje de programación más familiar. [6]

### 4.5 Text To Speech

Debido a que muchos de los usuarios finales muestran dificultades a la hora de leer, se ha decidido implementar una función que transforme el texto en audio, proporcionando al usuario una ayuda de doble canal, voz y texto. Esta funcionalidad se implementa mediante una librería Text To Speech, que es una API propia del entorno de desarrollo de Android Studio.[9]



# 5. Desarrollo de aplicación móvil.

---

En este capítulo del corpus se explicará el desarrollo de la aplicación, los requisitos de esta, los casos de uso, el modelo de datos y la implementación de la aplicación a sus diferentes niveles como son la interfaz y las funcionalidades.

## 5.1 Especificación de requisitos

En este apartado se redactarán los requisitos necesarios para el desarrollo de la herramienta en cuestión, como son, los requisitos generales donde se especifican las características más genéricas y los requisitos funcionales, en los que se definen las características intrínsecas al funcionamiento de la aplicación móvil.

### 5.1.1 Requisitos generales

La aplicación está diseñada y desarrollada para dispositivos móviles con sistema operativo Android a partir de la versión 5.1 Lollipop.

### 5.1.2 Requisitos funcionales

Los requisitos funcionales son aquellos que definen una función del software a desarrollar y se explicarán con más detalle a través de los casos de uso.

Para el correcto funcionamiento de la aplicación móvil, se precisan permisos que autoricen a la aplicación a utilizar componentes tanto software como hardware del dispositivo que use la app.

## 5.2 Casos de uso

Los casos de uso indican las diferentes funciones que permite desempeñar la aplicación.

- El usuario podrá seleccionar una imagen desde su memoria interna, pulsando el icono de la galería.
- El usuario podrá tomar una imagen al momento con su cámara, pulsando el icono de la cámara.
- La aplicación procesa la imagen ya sea seleccionada vía cámara o galería, y muestra el resultado con la mayor probabilidad.
- La aplicación muestra una alerta si la probabilidad es menor del setenta por ciento, para advertir la posibilidad de error.
- El usuario puede cancelar/reproducir el dictado del texto de la descripción al presionar el botón de stop/play.

- La aplicación comienza a dictar automáticamente la descripción del objeto detectado salvo si el reconocimiento no se considera válido (probabilidad de similitud con objetos del modelo es menor del setenta por ciento).
- En el caso de que la probabilidad se encuentre por encima del setenta por ciento, es decir, que el objeto se pueda considerar como reconocido, será el usuario, el encargado de reproducir el dictado presionando el botón de play.

## 5.3 Modelo de datos

En cuanto a los datos de la aplicación, la descripción de cada objeto a detectar y la ayuda asociada a cada objeto, se encuentra en un fichero de texto en formato JSON. Este posee como clave el nombre del objeto y como atributo la descripción o instrucciones de ayuda para el usuario acerca del uso del objeto.

Se realiza una llamada a este fichero y se busca la clave que coincida con el objeto detectado con mayor probabilidad, tomando su atributo y mostrando dicho atributo como descripción.

Los datos donde se encuentra la información de cada objeto se almacenan en un archivo con nombre "model" y extensión "tflite". Este es el fichero que almacena los valores de cada imagen, que posteriormente son comparados y generando así la probabilidad de similitud de la imagen tomada con todos los objetos de la lista de "labels.txt".

El fichero "model. tflite" se ha generado empleando la herramienta de *Microsoft Azure Custom Vision* que hemos mencionado anteriormente. [7]

## 5.4 Implementación

En este apartado se aborda la fase de implementación del proyecto que se dividen en dos, implementación de interfaz e implementación de funciones. Se explicarán algunos detalles relevantes, así como las diferentes dificultades que se encontraron en el proceso.

Las clases necesarias para la implementación de la aplicación móvil se organizan en un apartado estético o interfaz y un apartado de funciones donde se ejecuta el programa.

### 5.4.1 Interfaz

#### 5.4.1.1 Activity Main

La interfaz principal o main, tan solo contiene 6 elementos, los cuales son:

- ImageView, este será el encargado de mostrar al usuario la imagen que se ha seleccionado o tomado con la cámara del dispositivo.
- Botón Cámara, al presionar dicho botón, se accede a la cámara del dispositivo para tomar la imagen.
- Botón galería, al presionarlo permite seleccionar una imagen del almacenamiento interno del dispositivo.
- Botón pause/play, este botón detiene y reproduce el dictado por voz del texto de la descripción.

- TextView, se dispone de dos elementos TextView uno de ellos para el nombre y el otro para descripción.



**Figura 7- Interfaz Main Activity**

#### 5.4.1.2 Activity\_splash

La clase Activity splash contiene un ImageView donde se muestra el icono de la aplicación. Dicha Activity se muestra durante un breve periodo de tiempo al comienzo de la ejecución de la aplicación.

#### 5.4.1.3 Adver

Se realiza la llamada a esta Activity en forma de Pop-Up, superponiéndose a la clase activity\_main para informar al usuario de que la probabilidad de acierto del objeto a reconocer es menor del 70%. Se oculta al presionar en la pantalla.

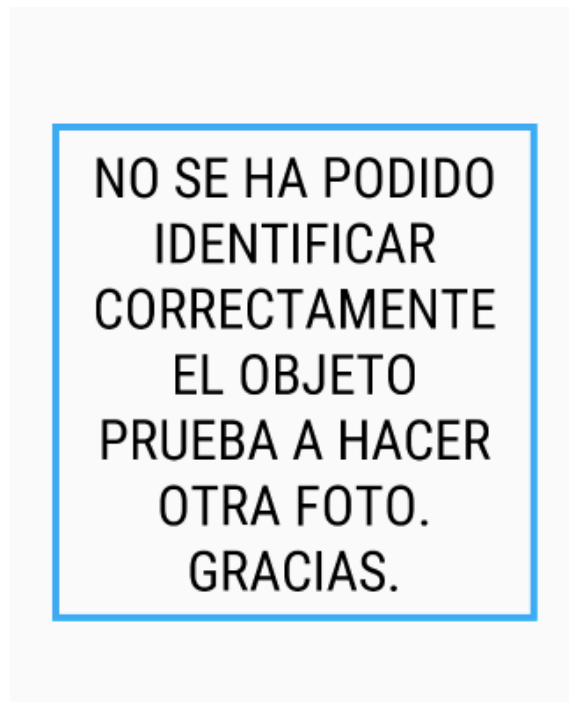


Figura 8: Interfaz Adver

#### 5.4.1.4 *Menu\_main*



Figura 9: Interfaz Menu main

Este layout se encuentra en la parte superior de la pantalla a la hora de ejecutar la aplicación. En él se indica el nombre de la aplicación y un icono el cual al ser pulsado muestra la información como se indica en la Figura 10.



Figura 10: Interfaz info

## 5.4.2 Funciones

### 5.4.2.1 Librerías

Podemos definir una librería como un conjunto de líneas de programa previamente definidas que facilitan operaciones y funcionalidades ya implementadas y listas para ser usada través de una Interfaz de Programación de Aplicaciones. [8]

Las librerías más importantes que debemos destacar son las de Tensor Flow. Estas nos permiten trabajar con el fichero “model.tflite” el cual contiene la información recopilada por la inteligencia artificial, como se indica en el apartado 4.2

Otras de las librerías relevantes son aquellas que nos permiten controlar las interfaces para modificarlas. Éstas se implementan de forma automática al proyecto cuando se genera el documento. Se encuentran intrínsecas al entorno de desarrollo, por lo que no es necesario importarlas.

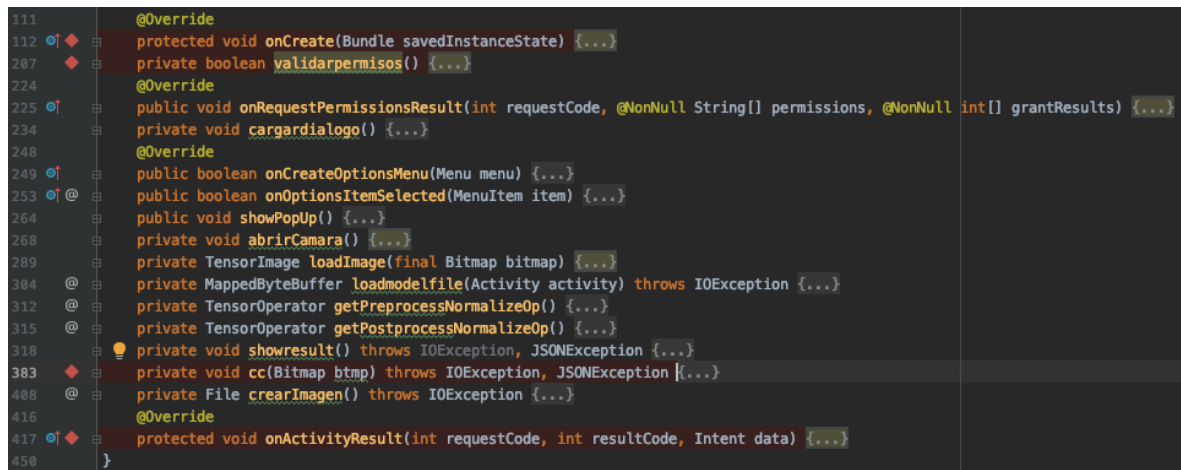
Las librerías que se emplean para transformar el texto en voz son las propias del entorno de desarrollo de Android Studio, “android.speech.tts” [9]. Una posible alternativa a esta librería ha sido el uso de una API proporcionada por Google, no obstante, esta segunda opción, se ha descartado debido a que requiere de conexión a internet para funcionar correctamente.

Por último, debemos importar las librerías “org.json.simple.JSONArray”, “org.json.simple.JSONObject ” y “ org.json.simple.parser.JSONParser ” para posibilitar el tratamiento del fichero “values.Json”. Este archivo contiene la información correspondiente a la descripción de los objetos que se desean identificar.

### 5.4.2.2 Funciones

A lo largo de este apartado se mostrarán las funciones necesarias para el desempeño de la aplicación móvil mediante líneas de código y su razón de ser.

El funcionamiento del reconocimiento de imagen consiste en tomar una imagen mediante la cámara o la galería del dispositivo móvil, dicha imagen se procesa mediante las librerías de Tensor Flow y por último se toma la descripción del objeto resultante mediante el archivo “values.Json” mostrando en la interfaz el resultado, junto a la descripción.



```
111 @Override
112 protected void onCreate(Bundle savedInstanceState) {...}
207 private boolean validarpermisos() {...}
224 @Override
225 public void onRequestPermissionsResult(int requestCode, @NonNull String[] permissions, @NonNull int[] grantResults) {...}
234 private void cargardialogo() {...}
248 @Override
249 public boolean onCreateOptionsMenu(Menu menu) {...}
253 public boolean onOptionsItemSelected(MenuItem item) {...}
264 public void showPopUp() {...}
268 private void abrirCamara() {...}
289 private TensorImage loadImage(final Bitmap bitmap) {...}
304 private MappedByteBuffer loadmodelfile(Activity activity) throws IOException {...}
312 private TensorOperator getPreprocessNormalizeOp() {...}
315 private TensorOperator getPostprocessNormalizeOp() {...}
318 private void showresult() throws IOException, JSONException {...}
383 private void cc(Bitmap bmp) throws IOException, JSONException {...}
408 private File crearImagen() throws IOException {...}
416 @Override
417 protected void onActivityResult(int requestCode, int resultCode, Intent data) {...}
450 }
```

Figura 11: Funciones

Todas las funciones que resultan relevantes para este proyecto se encuentran la clase MainActivity.java.

En la figura 11 podemos observar todas las funciones que se emplean durante el desarrollo de la aplicación. Se desarrollarán más en profundidad aquellas señalizadas con una marca roja en el margen izquierdo.

```
protected void onCreate(Bundle savedInstanceState)
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
```

*onCreate* es una función recursiva que se ejecuta constantemente. El primer comando que ejecuta esta función es seleccionar el layout *activity\_main* de la lista de layout como la pantalla principal.

Tras esto se ejecuta la función *validarpermisos* mediante la cual, se comprueban los permisos de ejecución necesarios tales como la camera y el acceso a memoria interna.

En las siguientes líneas de código se muestran las comprobaciones necesarias y en el caso de que alguno de los permisos no haya sido concedido, esta función se encarga de solicitarlos nuevamente.

```
private boolean validarpermisos() {
    if (Build.VERSION.SDK_INT < Build.VERSION_CODES.M) {
        return true;
    }
}
```



```

        if ((checkSelfPermission(CAMERA) == PackageManager.PERMISSION_GRANTED) &&
            (checkSelfPermission(WRITE_EXTERNAL_STORAGE) ==
             PackageManager.PERMISSION_GRANTED)) {
            return true;
        }
        if ((shouldShowRequestPermissionRationale(CAMERA)) ||
            (shouldShowRequestPermissionRationale(WRITE_EXTERNAL_STORAGE))) {
            cargardialogo();
        } else {
            requestPermissions(new String[]{WRITE_EXTERNAL_STORAGE, CAMERA}, 100);
        }
        return false;
    }
}

```

*btnG* y *btnC* hacen referencia a los botones de galería y camera respectivamente.

Se establece qué al pulsar el botón de galería, el usuario selecciona una imagen del dispositivo que se establece en el *ImageView* para ser procesada.

De igual forma, al presionar el botón de cámara, la aplicación permite tomar una imagen al momento. Dicha imagen se muestra en el *ImageView*, para ser posteriormente procesada.

A continuación, se muestran las líneas de código necesarias para configurar la herramienta Text To Speech. Esta parte del código se encuentra también dentro de la función *onCreate*.

```

t1 = new TextToSpeech(getApplicationContext(), new
TextToSpeech.OnInitListener() {
    @Override
    public void onInit(int status) {
        if (status != TextToSpeech.ERROR) {
            t1.setLanguage(new Locale("spa", "ESP"));
        }
    }
});

```

Las últimas líneas de código de esta función son las encargadas de abrir el fichero *values.Json*, el cual contiene la descripción de los objetos. Almacenando toda la información en una variable *String* llamada *js*.

A continuación, se muestran las líneas de programa que corresponden a la función *onActivityResult*. Esta función, recursiva también, es la encargada de tomar la imagen seleccionada o capturada y la muestra en el *ImageView*.

```

@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data)
{
    super.onActivityResult(requestCode, resultCode, data);
    if (requestCode == 12 && resultCode == RESULT_OK && data != null) {

        imageuri = data.getData();
        try {
            bitmap = MediaStore.Images.Media.getBitmap(getContentResolver(), imageuri);
            imageView.setImageBitmap(bitmap);

            cc(bitmap);
        } catch (IOException | JSONException e) {
            e.printStackTrace();
        }
    }
    if (requestCode == 1 && resultCode == RESULT_OK) {
        //Bundle extras = data.getExtras();
        Bitmap imgBitmap = BitmapFactory.decodeFile(ruta);
    }
}

```

```

        imageView.setImageBitmap(imgBitmap);
    try {
        cc(imgBitmap);
    } catch (IOException exception) {
        exception.printStackTrace();
    } catch (JSONException e) {
        e.printStackTrace();
    }
}

```

La imagen se descompone en un mapa de bits que se pasa por referencia a la función *cc()* donde será procesada dicha foto.

Esta función, recibe la imagen en forma de un mapa de bits que se descompone en los ejes X e Y. A continuación, las librerías de tensor Flow toma el modelo de datos de reconocimiento de imágenes previamente entrenado, comparando el mapa de bits actual con los almacenados en el modelo propiamente dicho. Por cada comprobación se genera una probabilidad que se almacenan en un HashMap donde la clave es el nombre del objeto y el valor es la probabilidad, además la mayor probabilidad se almacena en una variable global que ya hemos declarado previamente. La función finaliza llamando a *showresult()* donde se muestran los resultados.

La siguiente función toma el valor más alto de las probabilidades y obteniendo su respectivo objeto. A continuación, busca en el fichero values.json la clave que coincida con dicho objeto y muestra en el TextView correspondiente a la descripción la información extraída.

Finalmente, el TextView que contiene la descripción es dictado por el TextToSpeech que se crea en las últimas líneas de este código.

```

        mydialog.setContentview(R.layout.adver);
        mydialog.show();

        fab.setElevation(90);
        fab1.setElevation(99);
    }else{
        t1.speak(descr.getText().toString(),TextToSpeech.QUEUE_ADD, null);
    }
}

```

## 6. Resultado

En este capítulo se expondrán en detalle el principal resultado, la aplicación móvil de ayuda en el hogar al usuario con discapacidad intelectual. Se describirá el proceso de instalación necesaria para el uso de la aplicación móvil en un dispositivo Android y una guía de uso de ésta.

### 6.1 Instalación

Como se ha indicado en el capítulo 4.1.1, necesitaremos un dispositivo móvil Android con una versión superior o igual a 5.1 Lollipop para poder utilizar esta aplicación.

La aplicación actualmente se encuentra almacenada en Google Drive, siendo posible acceder a ella mediante el siguiente enlace.

<https://drive.google.com/file/d/1nXUqxdHF1GdQS5BPdLGWEyGZhOgCq5y8/view?usp=sharing>

Una vez accedido a dicho enlace desde el teléfono móvil, seleccionamos la opción de descargar. Ejecutamos el fichero app-debug.apk una vez haya terminado la descarga para finalizar la instalación. Este proceso se ve reflejado en las figuras 12 y 13.

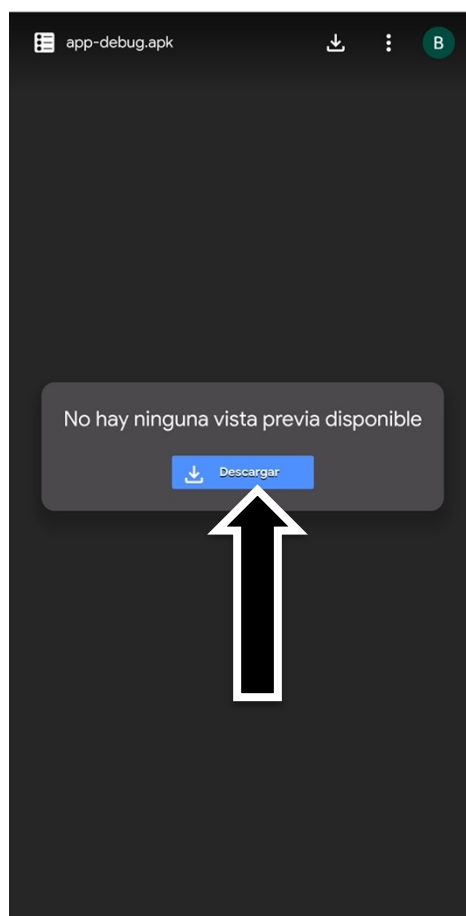


Figura 12: Drive apk.



**Figura 13: Descarga.**

Una vez finalice la instalación, la aplicación aparecerá en el cajón de aplicaciones de nuestro dispositivo.

## 6.2 Funcionamiento

Una vez finalizada la instalación como se menciona en el apartado anterior. Podemos ejecutar la aplicación seleccionándola del cajón de aplicaciones de nuestro móvil.

La primera vez que se ejecute la aplicación, se nos solicita los permisos que debemos aceptar como se muestra en la figura 15.



Figura 14: Inicio.

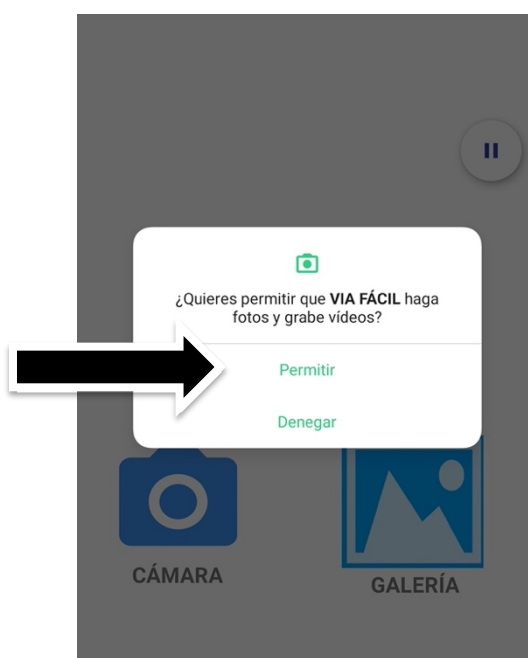
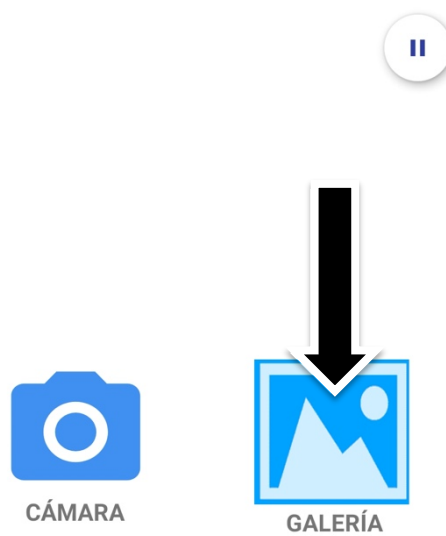


Figura 15: Permisos.

Una vez aceptados los permisos que se nos soliciten, podemos utilizar la aplicación.

Si queremos seleccionar una imagen que ya hemos realizado previamente o que tengamos en nuestro dispositivo, seleccionamos el botón con el icono de GALERIA, al pulsarlo se mostraran nuestras fotografías. Seleccionamos aquella que nos interese. Este proceso se encuentra reflejado en las figuras 16 y 17.



**Figura 16: Pantalla principal.**



**Figura 17: Selección de imagen.**

Tras seleccionar la imagen deseada, la aplicación nos muestra la pantalla principal, con el nombre y la descripción del objeto en cuestión.



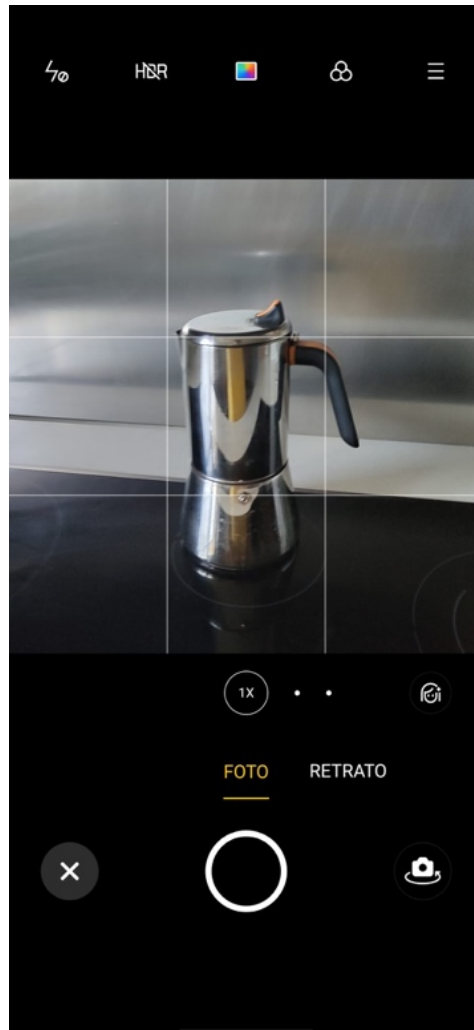
**Figura 18: Resultado.**

Si se desea pausar el dictado de la descripción tan solo debemos presionar el botón de pause, tal como se muestra en la figura 18.

Por otro lado, si queremos realizar una foto al momento, tan solo debemos presionar el icono de camera, el cual abrirá la interfaz propia de cada dispositivo para tomar la imagen. Ver figura 19 y 20.



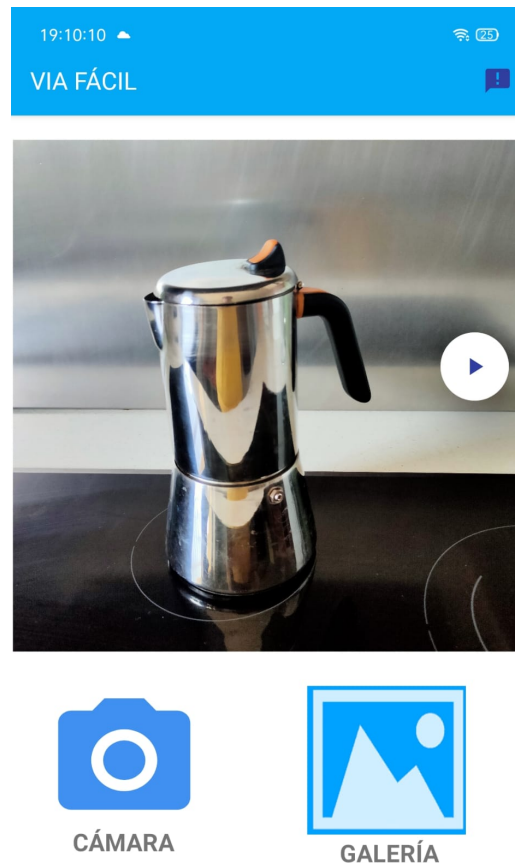
**Figura 19: Camera.**



**Figura 20: Tomar foto.**

Si estamos conformes con la imagen tomada, presionamos aceptar y nuevamente, la aplicación nos regresa a la pantalla principal. Mostrando el resultado con su correspondiente descripción como se indica en la figura 21.





### CAFETERA

Sirve para hacer café.

Lleno de agua la parte inferior  
de la cafetera.

Figura 21: Resultado camera.

# 7. Conclusiones

---

La inteligencia artificial es uno de los campos de la tecnología que más interés está generando. El reconocimiento de imágenes es parte de esta tecnología, que día a día está resultando ser más precisa y segura. Cabe mencionar que existen muchas formas y herramientas de conseguir un mismo fin. Sin embargo, el problema resulta a la hora de seleccionar correctamente las herramientas para dicha labor teniendo en cuenta el propósito de la aplicación.

Se ha programado una herramienta destinada a las personas con discapacidad intelectual. Dicha aplicación es capaz de reconocer cualquier objeto para el que ha sido entrenada mediante la inteligencia artificial que *Microsoft Azure Custom Vision* ha proporcionado. No obstante, se ha percatado de que la aplicación móvil VIA Fácil, puedes ser mejorada en varios aspectos como indicaremos en el siguiente apartado, y por tanto se puede seguir trabajando para poder ofrecer a las personas que lo necesiten la mejor ayuda posible.

Por último, cabe mencionar que el proyecto se ha llevado a cabo con éxito. Prueba de ello es la aplicación móvil, la cual se ha implementado tras un estudio de las técnicas y herramientas de visión artificial. Dicha aplicación móvil, capaz de reconocer con precisión muchos de los objetos para los que ha sido entrenada, proporcionando la información necesaria tanto por el canal de voz como el de texto. Cumpliendo así los objetivos inicialmente planteados, este proyecto ha contribuido al crecimiento a nivel personal, y aportando nuevas habilidades y nuevos conocimientos que serán de utilidad para mejorar esta aplicación. Evidencia también del valor aportado del proyecto es que de hecho ya se ha probado su uso y beneficio para las personas con discapacidad intelectual con usuarios reales proporcionados por una asociación como ASPRODES, que trabaja por la integración y ayuda a las personas con discapacidad intelectual, esto también supone una satisfacción personal por la ayuda facilitada a quienes más lo necesitan.



## 8. Líneas futuras

---

Para realizar este trabajo, se ha creado una aplicación móvil totalmente funcional, aunque como un prototipo. Por lo tanto, esta aplicación puede ser mejorada considerablemente en desarrollos posteriores. Aspectos que no se han considerado prioritarios, como los que se mencionan a continuación, se implementarán en las versiones futuras de la aplicación.

El número de usuarios de Android es mayor, sin embargo, creemos que implementar esta aplicación para dispositivos móviles con sistema operativos iOS, puede favorecer a su difusión.

También se ha considerado ampliar el número de objetos que es capaz de detectar. Para solucionar esto, se quiere diseñar una segunda aplicación móvil auxiliar que permita añadir objetos al modelo de entrenamiento utilizado, así como mejorar aquellos que ya están etiquetados. Una posible solución sería añadir más imágenes sobre los mismos que ayuden a su reconocimiento con mayor precisión.



# 9. Bibliografía

---

- [1] APPS DISCAPACIDAD INTELECTUAL. (2015, diciembre 9). Recuperado el 9 de Julio de 2021, de Fun4us.org website: <https://fun4us.org/apps/apps-discapacidad-intelectual/>
- [2] ¿Qué es la inteligencia artificial (IA)? (s/f). Recuperado el 9 de Julio de 2021, de Oracle.com website: <https://www.oracle.com/mx/artificial-intelligence/what-is-ai/>
- [3] Vision AI. (s/f). Recuperado el 9 de Julio de 2021, de Google.com website: [https://cloud.google.com/vision/?hl=es&utm\\_source=google&utm\\_medium=cpc&utm\\_campaign=emea-es-all-es-dr-bkws-all-all-trial-e-gcp-1010042&utm\\_content=text-ad-none-any-DEV\\_c-CRE\\_170507686544-ADGP\\_Hybrid%20%7C%20BKWS%20-%20EXA%20%7C%20Tt%20~%20AI%20%26%20ML%20~%20Vision%20AI%23v4-KWID\\_43700053288209441-aud-304040939401%3Akwd-15026601681-userloc\\_1005527&utm\\_term=KW\\_google%20vision-NET\\_g-PLAC\\_&ds\\_rl=1242853&ds\\_rl=1245734&ds\\_rl=1242853&ds\\_rl=1245734&gclid=CjwKCAjw55-HBhAEiwARMCszglU9VnTFCa7Ro82l4OlfSc9Z3bnAoUD\\_-IB7\\_ChCY6eZME9mcgN7hoCY4MQAvD\\_BwE&gclsrc=aw.ds](https://cloud.google.com/vision/?hl=es&utm_source=google&utm_medium=cpc&utm_campaign=emea-es-all-es-dr-bkws-all-all-trial-e-gcp-1010042&utm_content=text-ad-none-any-DEV_c-CRE_170507686544-ADGP_Hybrid%20%7C%20BKWS%20-%20EXA%20%7C%20Tt%20~%20AI%20%26%20ML%20~%20Vision%20AI%23v4-KWID_43700053288209441-aud-304040939401%3Akwd-15026601681-userloc_1005527&utm_term=KW_google%20vision-NET_g-PLAC_&ds_rl=1242853&ds_rl=1245734&ds_rl=1242853&ds_rl=1245734&gclid=CjwKCAjw55-HBhAEiwARMCszglU9VnTFCa7Ro82l4OlfSc9Z3bnAoUD_-IB7_ChCY6eZME9mcgN7hoCY4MQAvD_BwE&gclsrc=aw.ds)
- [4] (S/f). Recuperado el 9 de Julio de 2021, de Withgoogle.com website: <https://teachablemachine.withgoogle.com/>
- [5] profexorgeek. (s/f). ¿Qué es Xamarin? - Xamarin. Recuperado el 9 de Julio de 2021, de Microsoft.com website: <https://docs.microsoft.com/es-es/xamarin/get-started/what-is-xamarin>
- [6] Meet Android Studio. (s/f). Recuperado el 9 de Julio de 2021, de Android.com website: <https://developer.android.com/studio/intro>
- [7] Get to know Azure. (s/f). Recuperado el 9 de Julio de 2021, de Microsoft.com website: <https://azure.microsoft.com/en-us/overview/>
- [8] Anyelguti. (s/f). Librerías para la programación web. Recuperado el 9 de Julio de 2021, de Aprende-web.net website: <http://aprende-web.net/librerias/>
- [9] TextToSpeech. (s/f). Recuperado el 9 de Julio de 2021, de Android.com website: <https://developer.android.com/reference/android/speech/tts/TextToSpeech>
- [10] Patrick Farley. (s/f-b). Inicio rápido: Creación de un clasificador con el sitio web de Custom Vision. Recuperado el 13 de Julio de 2021, de Microsoft.com website: <https://docs.microsoft.com/es-es/azure/cognitive-services/custom-vision-service/getting-started-build-a-classifier>
- [11] Patrick Farley. (s/f). Exportación del modelo para su uso con dispositivos móviles. Recuperado el 13 de Julio de 2021, de Microsoft.com website: <https://docs.microsoft.com/es-es/azure/cognitive-services/custom-vision-service/export-your-model>