

Desarrollo de Aplicaciones Multiplataforma

IES Nervión,
Braille Writer



Rebeca Saraí González Guerra,
Curso 2017-2018

Tabla de Contenido

- [1. Estudio de Viabilidad del Sistema \(EVS\)](#)
 - [1.1. Objetivos del EVS \(Catálogo\) \(EVS 1\)](#)
 - [1.2. Establecimiento del alcance del sistema](#)
 - [1.2.1 Descripción General del sistema \(EVS 1\)](#)
 - [1.2.3 Estructura Organizativa](#)
 - [1.3. Estudio de la Situación Actual](#)
 - [1.3.1. Descripción de la situación actual](#)
 - [1.3.2. Descripción de los sistemas de Información Actuales](#)
 - [1.3.3 Diagnóstico de la Situación Actual](#)
 - [Problemas](#)
 - [Deficiencias](#)
 - [Mejoras](#)
 - [1.4. Catálogo de usuarios \(EVS 1, 2\)](#)
- [1.5. Definición de requisitos previos](#)
 - [1.5.1. Catálogo de requisitos previos](#)
- [Requisitos de Interfaz](#)
- [Requisitos Funcionales](#)
- [1.6. Estudio de Alternativas de construcción](#)
 - [1.6.1. Alternativa Primera](#)
 - [1.6.1.1. Requisitos](#)
 - [1.6.1.2. Tecnologías](#)
 - [1.6.1.2. Alternativa Segunda](#)
 - [1.6.1.2.1. Requisitos](#)
 - [1.6.1.2.2. Tecnologías \(S. O. Lenguajes, BD, redes,...\)](#)
 - [1.6.2. Análisis coste-beneficio \(incluyendo intangibles\)](#)
 - [1.6.3. Estudio de Riesgos](#)
 - [1.6.3.1 Estudio de riesgos de Alternativa primera](#)
 - [1.6.3.2. Estudio de riesgos de alternativa segunda](#)
- [1.7. Solución propuesta](#)
 - [1.7.1. Justificación](#)

2. Gestión de Proyectos

Gantt del desarrollo completo

Gantt EVS

Gantt del Análisis de Sistemas de información

Gantt del Diseño de Sistemas de información

Gantt de Construcción del sistema Informático

2.1.1. Hitos

3. Análisis de Sistemas de Información

3.1. Descripción General del Entorno Tecnológico del Sistema (ASI 1.2)

3.2. Catálogo de Usuarios (ASI 1.4)

3.3. Modelo de Negocio (ASI 1)

3.4. Modelo de Casos de Uso (ASI 2).

3.5. Especificación de Casos de Uso

3.6. Modelo de Clases de Análisis (ASI 4.1, ASI 5)

3.7. Interfaces de usuario (ASI 8)

3.7.1. Aspectos comunes de la interfaz de usuario (plantilla)

Aplicación móvil Android:

Aplicación web monitor en tiempo real:

3.7.2. Perfiles de usuarios

3.8. Especificación de pantallas/ventanas

3.8. Especificación de Requisitos Software ERS (ASI 9)

3.8.1. Introducción

3.8.2. Catálogo de requisitos de software

Requisitos de Interfaz

Requisitos Funcionales

4. Diseño de Sistemas de Información (DSI)

4.1 DSI 1: Definición de la arquitectura del sistema

4.1.1 Particionamiento Físico del Sistema de Información y especificación del entorno tecnológico

4.1.2 Catálogo de requisitos de Diseño (lenguajes que se van a usar, criterio de ubicación de módulos y datos, o sea un árbol de directorios donde muestre se va a cómo organizar la información de nuestra aplicación)

Requisitos de móvil

Requisitos de Base de datos

Requisitos servidor web para monitor

Requisitos cliente web para monitor

Requisitos de Machine Learning

4.1.3 Catálogo de Excepciones. (Rangos o valores no válidos en la entrada de datos, atributos obligatorios, formatos, específicos, etc

4.2 DSI 3: Diseño de la interfaz de usuario

4.2.1 Modelo de navegación de interfaz de Pantalla

Ventana de reconocimiento de rostros

4.3 DSI 6: Modelo físico de datos

4.4 DSI 8 y 4: Especificaciones detallada de componentes

4.5 DSI 9: Diseño de migración y carga inicial de datos

4.6 DSI 10: Plan de prueba

4.6.1 Especificación del Entorno de Pruebas

4.6.1 Especificación Técnica de Niveles de Prueba

4.7 DSI 11: Establecimiento de requisitos de implantación

4.7.1 Especificación de Requisitos de implantación

1. Estudio de Viabilidad del Sistema (EVS)

1.1. Objetivos del EVS (Catálogo) (EVS 1)

Con el presente trabajo de Estudio de Viabilidad del Sistema, se busca entender de manera más completa los puntos más relevantes del desarrollo del proyecto Braille Writer, como son el estudio del mercado, alternativas, herramientas y el desarrollo técnico.

1.2. Establecimiento del alcance del sistema

1.2.1 Descripción General del sistema (EVS 1)

El sistema consiste en una aplicación móvil para dispositivos android que usa inteligencia artificial para resolver problemas que las personas con deficiencia visual enfrentan. Como el no poder acceder al texto que se encuentra a su alrededor o leer una carta que les llega por correo, lo que los hace tardar mucho más tiempo en realizar las tareas comunes, afectándoles a lo largo de toda su vida.

El sistema permite al usuario, quienes son las personas invidentes, funcionalidades de reconocimiento de imágenes, identificación de alrededores y lectura de texto, todo a través de la cámara en tiempo real. Si llega una carta que no incorpora braille, pueden saber a quién va dirigida y qué dice. **Tan sólo sosteniendo el móvil son capaces de escuchar, acceder y gestionar la información del mundo que**

les rodea.

Además de la aplicación móvil, se hace uso de una base de datos en tiempo real y una aplicación web auxiliar con funcionalidades de administrador del sistema.

Siendo el objetivo principal del sistema informático mejorar la calidad de vida e inserción social de toda esta población, facilitando el acceso, modificación y gestión de la información a través de las nuevas tecnologías.

1.2.2 Diagrama de contexto del sistema

En el presente diagrama del contexto del sistema se representarán las funciones que los distintos tipos de usuarios pueden realizar en el mismo. El usuario, una vez inicia sesión en la aplicación móvil android, puede reconocer múltiples tipos de imágenes en tiempo real, incluyendo:

- Reconocimiento de texto que luego puede escuchar y traducir al braille haciendo uso de Optical Character Recognition.
- Selección de un archivo para traducir a braille
- Reconocimiento de objetos para escuchar el objeto con mayor porcentaje de predicción.
- Detección de rostros para identificar emociones, alegría, tristeza, etc.

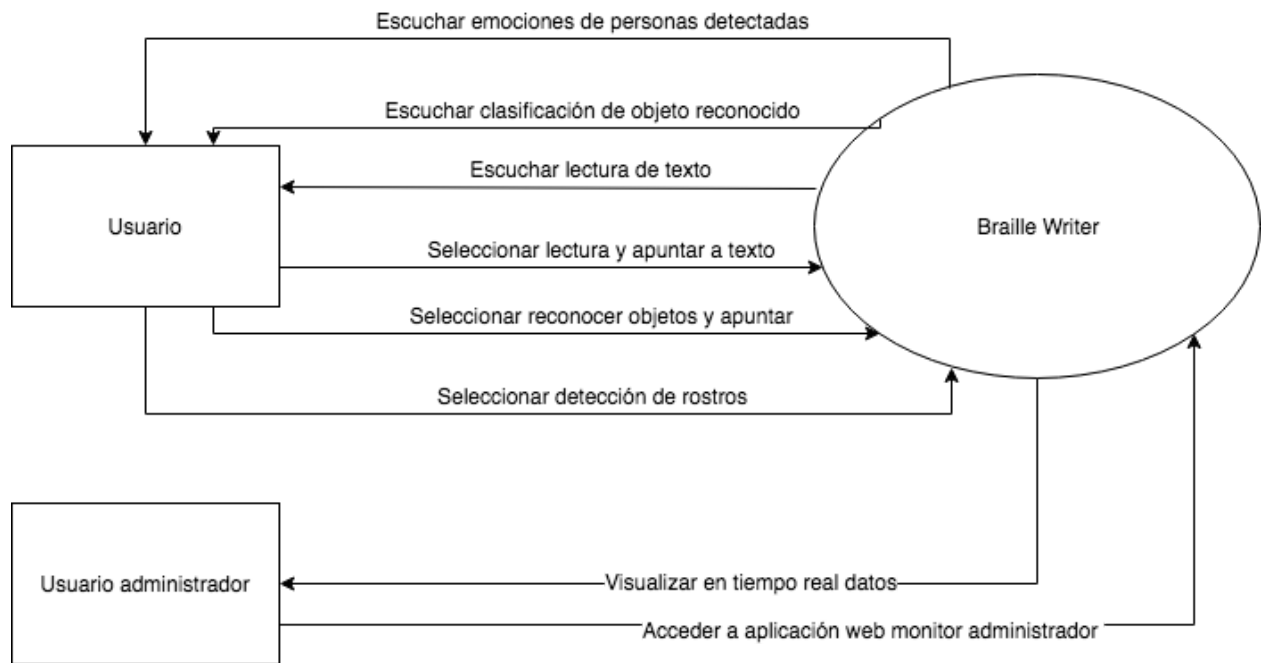


Diagrama de contexto del sistema

1.2.3 Estructura Organizativa

El sistema tiene dos actores principales: el usuario final, consumidor de la aplicación móvil, el cual puede ser suscrito o no suscrito; y el administrador que tiene acceso a un panel de control web en el cual se pueden visualizar estadísticas relevantes del uso de la misma en tiempo real.

1.3. Estudio de la Situación Actual

1.3.1. Descripción de la situación actual

Este proyecto parte de una idea original propia ya planificada con anterioridad.

Se llevaron a cabo validaciones de la misma mediante entrevistas a posibles usuarios finales y mapas de empatía. De las anteriores se identificó que tenía prioridad y sería más beneficioso para los usuarios el desarrollo de una aplicación móvil con visión artificial para clasificar imágenes.

También se llevaron a cabo estudios de la experiencia de usuario y de la interfaz de la aplicación que dieron lugar a diversos diseños hasta dar con el más adecuado.

Antes de desarrollar el sistema informático, existe el requisito de integrar la visión artificial en una aplicación para dispositivos móviles, esto es, que los modelos de redes neuronales usados no se ejecuten en la nube sino en el móvil del usuario final. Esto supone un reto por las capacidades de procesamiento de la CPU de estos dispositivos. Afortunadamente, durante el tiempo en que se realiza el desarrollo de este proyecto, múltiples grandes marcas han lanzado al mercado móviles con chips dedicados a resolver cálculos empleando aprendizaje automático y redes neuronales, estos realizan esas tareas con hasta 25 veces más velocidad que las CPU tradicionales, a la vez que son hasta 25 veces más eficientes desde el punto de vista energético. Encontramos ejemplos en Huawei o Qualcomm, que ya

ofrecen optimizaciones para la conocida librería de deep learning, TensorFlow, usada en este proyecto.

1.3.2. Descripción de los sistemas de Información Actuales

Se dispone a desarrollar el proyecto a nivel técnico, desde el principio. Por lo que no se parte de un sistema actual de la aplicación Braille Writer.

En cuanto a aplicaciones similares, no existe en el mercado algo que cubra todas las funciones del sistema Braille Writer. De esta manera, existen pocas herramientas o sistemas muy parecidos al propuesto en el presente proyecto, y que compartan también el mismo público objetivo, las personas con agudeza visual baja. Entre estas se encuentran disponible dos aplicaciones: Braigo y Seeing AI.

La aplicación móvil Seeing AI sólo está disponible para IOS, es decir móviles iPhone. La cual permite reconocer imágenes y texto, sólo en idioma inglés.

De la misma manera, consideramos a la aplicación de Braigo. Esta requiere subir una foto o imagen a un servidor para que sea procesada, y tras un tiempo, indicar en pantalla si se encuentra texto en la imagen o no, u objetos.

Con Braille Writer, no es necesario esperar a que se suba la imagen al servidor para procesar, se hace en tiempo real.

1.3.3 Diagnóstico de la Situación Actual

- Problemas

No existen suficientes opciones en aplicaciones en idioma español para el reconocimiento de imágenes en tiempo real.

- Deficiencias

Las opciones disponibles

- Mejoras

Se permitirá acceder a las diversas actividades y funcionalidades del software a través de una interfaz accesible. Se permitirá leer el texto en el entorno de la persona invidente o con agudeza visual baja, así como acceder al texto en documentos enteros y gestionarlos para tratarlos como sea conveniente. Reconociendo imágenes e identificando alrededores, facilitando su interacción con su entorno.

- Reconocer texto a través de la cámara en tiempo real, leer documentos enteros, cartas, facturas u otros.
- Detectar rostros humanos y sus emociones
- Reconocer más de 1000 objetos en tiempo real
- Generar un archivo de formato braille unicode para imprimir en una impresora braille, a partir de un archivo de texto plano, doc o pdf

- Suscribirse para usar todas las funcionalidades de manera ilimitada

1.4. Catálogo de usuarios (EVS 1, 2)

Código	Nombre	Descripción
USU1	Usuario sin suscripción	La aplicación permitirá a cualquiera que la use sin suscribirse acceder a algunas funcionalidades.
USU2	Usuario con suscripción	Este usuario tiene acceso ilimitado a todas las funcionalidades de la aplicación móvil.
USU 3	Usuario administrador	Este usuario tiene acceso al panel de control y visualización de datos en tiempo real desde una aplicación web, Single Page Application (SPA)

1.5. Definición de requisitos previos

1.5.1. Catálogo de requisitos previos

Requisitos de Interfaz

Requisitos de accesibilidad

Requisito	Nombre	Prioridad	Descripción
Req_1	Guía auditiva inicial	Alta	El sistema indica al usuario cuando a entrado en la aplicación y la actividad inicial.
Req_2	Guía auditiva entre las actividade s	Alta	El sistema informático indica al usuario cuando intenta entrar a una actividad en la que ya se encuentra.
Req_3	Interfaz con alto contraste	Alta	La interfaz debe de tener una paleta de colores con alto contraste.
Req_4	Accesibilid ad	Alta	Los elementos de la interfaz deben ser accesible para el lector de pantalla.
Req_5	Reconoci miento de voz	Alta	La aplicación podrá realizar reconocimiento de voz de los usuarios.

Req_6	Comandos de voz	Alta	La aplicación, a través del reconocimiento de voz deberá de reconocer las palabras dichas del usuario para navegar entre las actividades y funcionalidades
-------	-----------------	------	--

Requisitos Funcionales

Reconocimiento de Texto a través de la cámara y lectura en altavoz

Requisito	Nombre	Prioridad	Descripción
Req_7	OCR (Optical Character Recognition)	Alta	Con la integración de Inteligencia Artificial, Machine Learning, a través de la cámara se reconozca el texto visible en el entorno. Carteles, facturas, documentos, informes, etc
Req_8	Guardarlo como cadena de caracteres	Alta	Una vez delimitado el texto, guardarlo de manera temporal para luego procesarlo.

Req_9	Audio de ese texto en tiempo real	Alta	Procesar el texto a través de la cámara de forma inmediata para leerlo en altavoz, también llamado Text to Speech.
-------	-----------------------------------	------	--

Traducción de archivos a braille y guardado

Requisito	Nombre	Prioridad	Descripción
Req_10	Acceso a archivos de texto .txt, .pdf y .doc	Alta	La aplicación, con permiso del usuario, accede a los archivos del dispositivo y permite seleccionar uno de formato .txt, .pdf y .doc
Req_11	Conversión a braille	Alta	Una vez seleccionado el documento, se hace una conversión en base ASCII para hacerlo compatible con impresoras braille.
Req_12	Guardar como archivo braille unicode	Alta	Guardar el texto reconocido y obtenido a través de la cámara como archivo en la memoria interna del dispositivo.

Impresión en formato Braille y estándar

Requisito	Nombre	Prioridad	Descripción
Req_13	Imprimir braille	Medio	Las personas invidentes podrán imprimir en una impresora de tipo braille.
Req_14	Imprimir estándar	Medio	Poder imprimir un documento desde la aplicación.

Compartir archivos

Requisito	Nombre	Prioridad	Descripción
Req_15	Compartir archivos	Bajo	Se podrá compartir mediante redes sociales y correo electrónico los archivos, a través de la aplicación.

Suscripción mensual de usuarios premium

Requisito	Nombre	Prioridad	Descripción
Req_16	Suscripción de usuarios	Alta	Los usuarios se podrán optar a una suscripción mensual de forma sencilla para usar todas las funcionalidades.

Monitor para visualización de datos en tiempo real

Requisito	Nombre	Prioridad	Descripción
Req_17	Visualización de datos	Baja	El sistema contará también con una aplicación web que permitirá la monitorización de datos relacionados al uso de los usuarios finales de la aplicación móvil.
Req_18	Usuario Administrador	Media	El sistema contará con la posibilidad de tener al menos un usuario administrador registrado en la base de datos
Req_19	Acceso de administrador	Media	Sólo el usuario administrador podrá acceder al monitor, a través de un sistema de autenticación y previo registro en la base de datos

1.6. Estudio de Alternativas de construcción

1.6.1. Alternativa Primera

La primera alternativa es desarrollar una aplicación Android, utilizando la librería de Machine Learning Tensorflow, con Tensorflow Mobile el cual está enfocado para poder acceder a modelos de redes neuronales desde dispositivos móviles. De esta manera, constando de un modelo de redes neuronales convolucionales entrenado, re-entrenarlo para añadir más clases de imágenes a reconocer y hacerlo más preciso.

Para utilizar Tensorflow Mobile, es necesario usar Android NDK, el cual permite ejecutar e implementar código completamente nativo, usando C++.

En esta primera alternativa se propone usar el modelo de redes neuronales convolucionales Inception V3, el cual es capaz de clasificar hasta 1000 clases de imágenes distintas. También se plantea la posibilidad de re-entrenarlo con Python, a través de la API Keras y sobre un servidor dedicado con tarjetas gráficas que soportan CUDA, como lo son los containers de AWS con tarjetas NVIDIA.

Junto a lo anterior también se haría uso de la API de Google Vision para el reconocimiento óptico de caracteres (Optical Character Recognition). En conjunto

con una base de datos nosql en tiempo real, en la nube de Firebase. La cual ofrece escalabilidad, amplia documentación y comunidad.

Se precisa, como una parte complementaria del sistema informático, una aplicación web con un panel de control que permita monitorizar los datos en tiempo real provenientes de la base de datos, a través de gráficas interactivas, con el principal objetivo de analizar estos datos de forma rápida y agradable. En primera instancia el número de usuarios suscritos y los objetos más reconocidos por la funcionalidad de reconocimiento de la aplicación Android. De esta manera, se pueden tratar los datos de uso de los usuarios y sacar conclusiones e incluso predicción en el comportamiento de los mismos.

Se propone Node.js para la realización del mismo, el cual es un entorno de ejecución para JavaScript construido con el motor de JavaScript V8 de Chrome, para el servidor (Backend), deployed con la plataforma de Heroku. A la par que Vue.js como framework de la parte cliente (Frontend).

1.6.1.1. Requisitos

Las tecnologías deben permitir el posible uso comercial de la aplicación, siguiendo un modelo de negocios de suscripción por el producto. De esta manera, las herramientas aunque sean de pago, deben no suponer un problema para ello.

La aplicación supone un coste debido a los servicios de Firebase y servidores, los cuales son de pago por uso, por lo que se dispondrá de un presupuesto estimado para cubrir el desarrollo del primer prototipo.

La aplicación web que será monitor de los datos, debe de ser dinámica y reactiva a los datos de la base de datos. Esto quiere decir que debe de actualizarse en tiempo real, se sugiere un framework para frontend que facilite el desarrollo y simplifique un coste de proceso bajo.

1.6.1.2. Tecnologías

- Android Studio 3.0, Android NDK y Gradle
- Lenguaje de programación Java
- Lenguaje de programación Python
- AWS
- API Google Vision
- Computación en nube
- Firebase Real Time Database
- Firebase Authentication
- Tensorflow Mobile
- Keras
- Heroku
- Node.js
- Vue.js

1.6.1.2. Alternativa Segunda

La segunda opción es desarrollar la aplicación móvil, también nativa desde el IDE oficial de Android, con varios de los servicios de Inteligencia artificial de Google Cloud Platform.

Utilizando la Vision API - Image Content Analysis, para reconocer las imágenes a través de peticiones al servidor de la nube y con respuesta JSON de las predicciones.

Con un cliente de la api, capturar las imágenes, enviarlas y dar una respuesta a los usuarios luego de un tiempo necesario de procesamiento.

Con uso de API Google Vision para el Optical Character Recognition (OCR), para la lectura de texto.

1.6.1.2.1. Requisitos

Para el desarrollo de la aplicación con estas tecnologías y servicios, se debe de dar de alta en la plataforma de Google Cloud, iniciando con una inversión financiera para cubrir el costo por uso del Machine Learning y crear un proyecto. Luego habilitar las API necesarias.

Al añadir las dependencias al IDE de desarrollo se debe de estar al tanto de la compatibilidad con la API de Android y los demás componentes.

Se debe de crear una cuenta en Firebase y crear un proyecto.

1.6.1.2.2. Tecnologías (S. O. Lenguajes, BD, redes,..)

- Computación en la nube
- Lenguaje de programación Java
- Android Studio 3.0
- API Google Vision
- Google Cloud Platform Datastore y Vision API
- Firebase Authentication
- Heroku
- Node.js
- Vue.js

1.6.2. Análisis coste-beneficio (incluyendo intangibles)

A	B	C	D	E	F	G
	Año	1	2	3	4	5
Ingresos	Ahorro gestoría	15.000,00 €	15.000,00 €	15.000,00 €	15.000,00 €	15.000,00 €
	Clasificación y archivo	4.347,20 €	4.347,20 €	4.347,20 €	4.347,20 €	4.347,20 €
	Total ingresos	19.347,20 €	19.347,20 €	19.347,20 €	19.347,20 €	19.347,20 €
Costes	Paquete CompuPYME	500,00 €				
	Equipos	2.350,00 €				
	Formación	2.200,00 €				
	Adaptación	850,00 €				
	Mantenimiento	975,00 €	975,00 €	975,00 €	975,00 €	975,00 €
	Operación	10.868,00 €	10.868,00 €	10.868,00 €	10.868,00 €	10.868,00 €
	Consumibles	1.020,00 €	1.020,00 €	1.020,00 €	1.020,00 €	1.020,00 €
	Total costes	18.763,00 €	12.863,00 €	12.863,00 €	12.863,00 €	12.863,00 €
	Beneficio neto	584,20 €	6.484,20 €	6.484,20 €	6.484,20 €	6.484,20 €
	Beneficio acumulado	584,20 €	7.068,40 €	13.552,60 €	20.036,80 €	26.521,00 €
	Hora operación:	10,45 €		Consumibles por mes	85	
	Horas operación:	20		Meses/año	12	
	Horas ahorradas:	8				
		52				

1.6.3. Estudio de Riesgos

1.6.3.1 Estudio de riesgos de Alternativa primera

Riesgo	Porcentaje	Importancia
Poco espacio para descargar dependencias de la app	30%	1
Inexactitud del reconocimiento de objetos	25%	2
Inexactitud en reconocimiento de texto	25%	2
Inexactitud en detección de emociones y rostros	10%-15%	2
Alto coste de procesamiento en el monitor web por parte del frontend	10%	3

Mayor consumo energético de la aplicación en base al modelo móvil que se use al ejecutar tensorflow	15%	3
Usuarios no instalen la aplicación por tamaño muy grande de los modelos.	10%	4
Usuarios no instalen la aplicación por mayor consumo de energía	10%	4

1.6.3.2. Estudio de riesgos de alternativa segunda

Riesgo	Porcentaje	Importancia
Altos costes	80%	1
Largos tiempos de espera de respuesta	40%	2
Inexactitud del reconocimiento de objetos	25%	3
Inexactitud en reconocimiento de texto	25%	3
Inexactitud en detección de emociones y rostros	10%-15%	3
Alto coste de procesamiento en el monitor web por parte del frontend	10%	3

1.7. Solución propuesta

1.7.1. Justificación

Entre las diferentes alternativas y opciones para abordar el desarrollo del proyecto, la más conveniente es la primera, esto se debe a lo siguiente:

La aplicación debe de ser lo más veloz posible, permitiendo al usuario no tener que esperar intervalos de tiempo molestos para saber lo que tiene en frente o el texto de una carta. Por lo que el hacer peticiones constantemente a la nube para cada una de los reconocimientos, supone un gran riesgo de espera a los usuarios que no tengan una conexión a red lo suficientemente rápida.

Los costes mensuales de los servicios de Inteligencia Artificial de la nube de Google serían cada vez más elevados a medida que más usuarios usen la aplicación o aumenten su frecuencia de uso. Sin embargo, con Tensorflow y Keras, supone un coste sólo al comienzo durante el tiempo del entrenamiento del modelo. Ofreciendo también total autonomía y control sobre el modelo en sí, su peso, optimización, número de capas, estructura, clases de imágenes capaz de identificar y mucho más.

Como parte del sistema se usarán últimas tecnologías de desarrollo web para la construcción del monitor. Las tecnologías propuestas para esta, han sido aprendidas durante la realización de las prácticas.

Node.js usa un modelo de operaciones Entrada/Salida sin bloqueo y orientado a eventos, que lo hace liviano y eficiente. Además de esto, el gestor de paquetes de Node.js, npm, es el ecosistema más grande de librerías de código abierto en el mundo.

2. Gestión de Proyectos

2.1. Planificación en MS-Project, asignando tiempos a las tareas (Gantt)

Gantt del desarrollo completo

Braille Writer			2017																				2018											
Entregables	Duración		Septiembre				Octubre				Noviembre				Diciembre				Enero				Febrero				Marzo				Abril			
			W1	W2	W3	W4	W1	W2	W3	W4	W5	W1	W2	W3	W4	W1	W2	W3	W4	W1	W2	W3	W4	W5	W1	W2	W3	W4	W1	W2	W3	W4		
Evaluación de Viabilidad del Sistema																																		
EVS I	5 w																																	
EVS II	2 w																																	
EVS III	3 w																																	
EVS III	2 w																																	
Análisis de Sistemas de Información																																		
ASI I	3 w																																	
ASI II	2 w																																	
ASI III	2 w																																	
ASI IV	2w																																	
ASI IV	1 w																																	
Diseño de sistemas de información																																		
DSI I	6 w																																	
DSI I	3 w																																	
DSI II	2 w																																	
DSI III	3 w																																	
DSI IV	3 w																																	
DSI V	2 w																																	
Construcción del Sistema Informático																																		
CSI I	14 w																																	
CSI I	2 w																																	
CSI II	4 w																																	
CSI III	5 w																																	
CSI IV	3 w																																	
CSI VI	4 w																																	
CSI VI	3 w																																	
Revisión																																		
Documentación de Código	4 w																																	
Actualización de documentación II	4 w																																	

Gantt EVS

Braille Writer				2017							
Entregables			Duración	Septiembre				Octubre			
				W1	W2	W3	W4	W1	W2	W3	W4
Evaluación de Viabilidad del Sistema			5 w								
EVS I			2 w								
EVS II			3 w								
EVS III			2 w								

Gantt del Análisis de Sistemas de información

Braille Writer				2017									
Entregables		Duración		Octubre				Noviembre					
				W1	W2	W3	W4	W1	W2	W3	W4	W5	
Análisis de Sistemas de Información		3 w											
ASI I		2 w											
ASI II		2 w											
ASI III		2w											
ASI IV		1 w											

Gantt del Diseño de Sistemas de información

Braille Writer				2017													
Entregables		Duración		Noviembre					Diciembre				Enero				
				W1	W2	W3	W4	W5	W1	W2	W3	W4	W1	W2	W3	W4	
Diseño de sistemas de información		6 w															
DSI I		3 w															
DSI II		2 w															
DSI III		3 w															
DSI IV		3 w															
DSI V		2 w															

Gantt de Construcción del sistema Informático

Braille Writer																			2018							
			Diciembre				Enero				Febrero					Marzo				Abril						
			W1	W2	W3	W4	W1	W2	W3	W4	W1	W2	W3	W4	W5	W1	W2	W3	W4	W1	W2	W3	W4			
Entregables		Duración																								
Construcción del Sistema Informático																										
	CSI I	14 w																								
	CSI II	2 w																								
	CSI III	4 w																								
	CSI III	5 w																								
	CSI IV	3 w																								
	CSI VI	4 w																								
	CSI VI	3 w																								
Revisión																										
	Documentación de Código	4 w																								
	Actualización de documentación II	4 w																								
		4 w																								

2.1.1. Hitos

Código	Nombre de tarea	Fecha finalización
EVS	Comienzo de EVS	30/10/2017
EVS	Establecimiento del alcance	20/09/2017
EVS	Identificación del Alcance	20/09/2017
EVS	Estudio del diagrama de contexto	20/09/2017
EVS	Estudio de la situación Actual	20/09/2017
EVS	Descripción de los sistemas de información existentes	20/09/2017
EVS	Estudio de la estructura organizativa	20/09/2017
EVS	Definición de Requisitos del Sistema	20/09/2017
EVS	Catalogación de usuarios	20/09/2017
EVS	Identificación de Requisitos	20/09/2017

EVS	Catalogación de Requisitos	20/09/2017
EVS	Estudio de alternativas de la solución	20/09/2017
EVS	Descripción de las alternativas	20/09/2017
EVS	Estudio del hardware alternativas	20/09/2017
EVS	Estudio de las tecnologías de las alternativas	20/09/2017
EVS	Estudio de los requisitos de las alternativas	20/09/2017
EVS	Valoración de las alternativas	20/09/2017
EVS	Estudio de los riesgos	20/09/2017
EVS	Selección de la solución	30/10/2017
EVS	Evaluación de las alternativas y selección	30/10/2017
EVS	EVS Completo	30/10/2017
GP	Gestión de proyecto	20/11/2017
GP	Selección de la estrategia de desarrollo	2/11/2017
GP	Lista de actividades e hitos	15/11/2017
	Planificación MS-Proyect	20/11/2017

GP	Gestión proyecto Completo	20/11/2017
ASI	Análisis de Sistemas de información	29/01/2018
ASI	Determinación del alcance del sistema	28/12/2017
ASI	Identificación de los usuarios participantes y finales	28/12/2017
ASI	Estudio del entorno tecnológico general	27/01/2018
ASI	Establecimiento de Requisitos	29/01/2018
ASI	Obtención de requisitos	29/01/2018
ASI	Especificación de casos de Uso	29/01/2018
ASI	Análisis de requisitos	29/01/2018
ASI	Catálogo de Usuarios	29/01/2018
ASI	Análisis de Casos de Uso	29/01/2018
ASI	Especificación de Casos de	29/01/2018

	Uso	
ASI	Análisis de Clases	10/02/2018
ASI	Identificación de responsabilidades y atributos	10/02/2018
ASI	Identificación de asociaciones y agregaciones	10/02/2018
ASI	Identificación de Generalizaciones	10/02/2018
ASI	Modelos de clases de análisis	12/02/2018
ASI	Definición de Interfaces de Usuario	20/02/2018
ASI	Especificación de principios generales de la interfaz	20/02/2018
ASI	Identificación de perfiles y diálogos	20/02/2018
ASI	Especificación de formatos individuales de la interfaz de pantalla	20/02/2018
ASI	Especificación de comportamiento dinámico de la interfaz	20/02/2018
ASI	Estudio Aspectos comunes de la interfaz	22/02/2018
ASI	Estudio Perfiles de usuario	23/02/2018
ASI	Estudio Mapa de pantallas/ventanas	24/02/2018
ASI	Estudio de especificación de pantallas/ventanas	24/02/2018
ASI	Estudio Especificación Informes	24/02/2018

ASI	Análisis de consistencia y especificación de requisitos	26/02/2018
ASI	Elaboración de la Especificación de Requisitos Software (ERS)	11/03/2018
ASI	Estudio Propósito y ámbito	11/03/2018
ASI	Estudio descripción general del producto y funciones	11/03/2018
ASI	Estudio características de los usuarios	11/03/2018
ASI	Estudio requisitos específicos	11/03/2018
ASI	Interfaces Externas	12/03/2018
ASI	Diagrama de descomposición de requisitos	16/03/2018
ASI	Estudio de entradas, salidas procesos y requisitos no funcionales	18/03/2018
ASI	Restricciones de diseño	20/03/2018
ASI	Análisis de Sistemas de información completo	21/03/2018
DSI	Diseño de sistemas de información	21/03/2018
DSI	Identificación de requisitos de diseño	21/03/2018
DSI	Especificación de estándares y normas	21/03/2018
DSI	Identificación de Subsistemas	22/03/2018
DSI	Identificación del entorno tecnológico	22/03/2018

DSI	Diseño de la arquitectura	02/04/2018
DSI	Particionamiento físico del sistema	02/04/2018
DSI	Catálogo de requisitos de diseño	02/04/2018
DSI	Catálogo de excepciones	08/04/2018
DSI	Diseño de casos de uso reales	08/04/2018
DSI	Diseño de clases	08/04/2018
DSI	Diseño físico de datos	12/04/2018
DSI	Diseño de la interfaz de usuario	12/04/2018
DSI	Modelo de navegación	12/04/2018
DSI	Formatos individuales de interfaz	12/04/2018
DSI	Modelo de clases de diseño	12/04/2018
DSI	Modelo físico de datos	14/04/2018
DSI	Generación de especificaciones de construcción	14/04/2018
DSI	Especificación detallada de componentes	14/04/2018
DSI	Diseño de procedimientos de migración y carga actual	14/04/2018
DSI	Plan de migración y carga inicial de datos	14/04/2018

DSI	Especificaciones técnicas del plan de pruebas	15/05/2018
DSI	Catálogo de requisitos de implantación	15/05/2018

3. Análisis de Sistemas de Información

3.1. Descripción General del Entorno Tecnológico del Sistema (ASI 1.2)

El programa consta de desarrollo nativo para Android, con lenguaje de programación Java, usando principalmente como IDE Android Studio.

Es imperante el uso de la librería de Machine Learning de código abierto Tensorflow con su api de alto nivel Keras y Python como lenguaje para el re-entrenamiento de los modelos de Deep Learning. Esta librería y sus dependencias son cargadas mediante Gradle, una herramienta de construcción oficial de android. Siendo posible también de ser generado un proyecto integrado con esta libreria usando CMake o Docker.

Como parte del entorno también se encuentran servicios de computación en nube, como la base de datos en tiempo real de Firebase.

3.2. Catálogo de Usuarios (ASI 1.4)

Código	Nombre	Descripción
USU1	Usuario sin suscripción	La aplicación permitirá a cualquiera que la use sin suscribirse, de hacer uso de algunas funcionalidades, permitiendo interactuar de

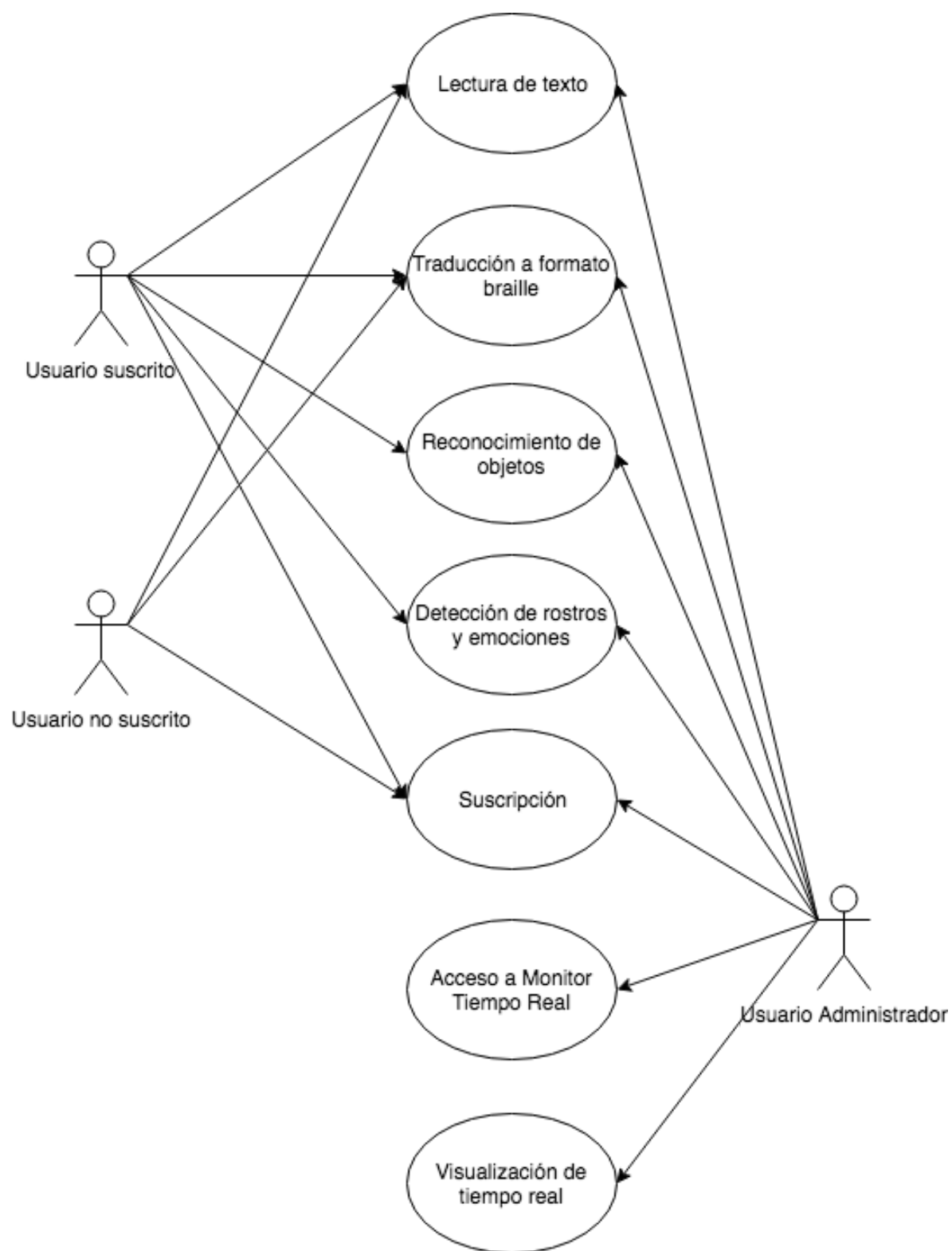
		forma parcial.
USU2	Usuario con suscripción	Este usuario tiene acceso ilimitado a todas las funcionalidades de forma completa.
USU 3	Usuario administrador	Accede al panel de control y monitor de datos en tiempo real, una aplicación web sincronizada a la base de datos y podrá interactuar con esta herramienta para tratar los datos y profundizar en su significado.

3.3. Modelo de Negocio (ASI 1)

El proyecto contará de un modelo de negocios de suscripción, en el cual los usuarios podrán hacer uso de varias funcionalidades y tendrán la opción de usar todas de forma ilimitada por un modesto precio mensual. Esto con la opción de suscripción de Google Services.

3.4. Modelo de Casos de Uso (ASI 2).

En el presente diagrama se muestran los posibles casos de uso de la aplicación, teniendo en cuenta el perfil del usuario: suscrito, no suscrito o administrador.



3.5. Especificación de Casos de Uso

UC1	Lectura de texto a través de cámara	
Requisito	Req_5	
Versión	1.0	
Descripción	El usuario podrá acceder al texto a través de la cámara para escucharlo	
Precondiciones	Permitir acceso a la cámara	
Postcondiciones	Se escuchará en un sonido, el texto identificado	
Secuencia normal		
	Paso	Acción
	1	El usuario selecciona la función de lectura
	2	Apunta con la cámara al posible texto
	3	Toca cualquier parte de la pantalla para escuchar el contenido del texto reconocido
	4	Se produce un audio del texto detectado

Excepciones		
	Paso	Acción
	1	Si el usuario no apunta correctamente al texto, es posible que se detecte de forma incompleta.
	2	Si el usuario posee una cámara de baja calidad con poca resolución puede tardar más tiempo en detectar el texto de forma correcta.

UC2	Traducción de archivo a formato braille
Requisito	
Versión	1.0
Descripción	El usuario podrá seleccionar archivos y traducirlos a formato braille unicode compatible con impresoras braille y dispositivos de braille refrescable.
Precondiciones	Permitir acceso a archivos
Postcondiciones	Se generará y guardará un archivo en formato braille unicode en el dispositivo

Secuencia normal		
	Paso	Acción
	1	El usuario selecciona el archivo de texto, PDF o doc.
	2	Confirma que es el que desea traducir.
	3	Genera un archivo de formato braille y automáticamente guarda en el sistema de archivos interno.
	4	Permite compartir archivo con otras aplicaciones.
Excepciones		
	Paso	Acción
	1	Sólo es posible seleccionar un archivo del tipo .txt o .doc. El resto de archivos son desactivados, se muestran con fondo y letra gris para indicar que no se pueden seleccionar.
	2	Si la memoria interna disponible del dispositivo es menor a la necesaria, el archivo no es

		guardado en el dispositivo.

UC3	Reconocimiento de objetos					
Requisito						
Versión	1.0					
Descripción	El usuario podrá reconocer objetos que estén al alcance de la cámara					
Precondiciones	Permitir acceso a la cámara					
Postcondiciones	Escuchar el nombre del objeto reconocido que obtiene mayor índice de precisión					
Secuencia normal	<table><tr><td>Paso</td><td>Acción</td></tr><tr><td>1</td><td>El usuario luego de seleccionar a reconocimiento de objetos, apunta la cámara al área u objeto que desea reconocer</td></tr></table>		Paso	Acción	1	El usuario luego de seleccionar a reconocimiento de objetos, apunta la cámara al área u objeto que desea reconocer
Paso	Acción					
1	El usuario luego de seleccionar a reconocimiento de objetos, apunta la cámara al área u objeto que desea reconocer					

	2	Toca cualquier parte de la pantalla
	3	Escucha la predicción con porcentaje más alto del objeto, proveniente de la salida del modelo de deep learning
	4	
Excepciones		
	Paso	Acción
	1	Si el usuario no apunta a ningún objeto con suficiente luz, en ese caso no escuchará nada.

UC4	Detección de rostros y emociones
Requisito	
Versión	1.0
Descripción	El usuario podrá detectar rostros al alcance de la cámara y reconocer sus emociones

Precondiciones	Ser usuario suscrito y permitir acceso a la cámara										
Postcondiciones	Se genera un audio describiendo la emoción de los rostros humanos										
Secuencia normal	<table border="1"> <thead> <tr> <th>Paso</th><th>Acción</th></tr> </thead> <tbody> <tr> <td>1</td><td>El usuario selecciona la actividad de reconocer rostros.</td></tr> <tr> <td>2</td><td>Usa su cámara para apuntar a la posición en donde puede encontrarse la o las personas.</td></tr> <tr> <td>3</td><td>Toca cualquier parte de la pantalla para escuchar.</td></tr> <tr> <td>4</td><td>Se reproduce un audio con la emoción de esa persona en caso de ser solo una. En caso de ser varias, describe cuántas personas se encuentran y si la mayoría tiene una emoción determinada.</td></tr> </tbody> </table>	Paso	Acción	1	El usuario selecciona la actividad de reconocer rostros.	2	Usa su cámara para apuntar a la posición en donde puede encontrarse la o las personas.	3	Toca cualquier parte de la pantalla para escuchar.	4	Se reproduce un audio con la emoción de esa persona en caso de ser solo una. En caso de ser varias, describe cuántas personas se encuentran y si la mayoría tiene una emoción determinada.
Paso	Acción										
1	El usuario selecciona la actividad de reconocer rostros.										
2	Usa su cámara para apuntar a la posición en donde puede encontrarse la o las personas.										
3	Toca cualquier parte de la pantalla para escuchar.										
4	Se reproduce un audio con la emoción de esa persona en caso de ser solo una. En caso de ser varias, describe cuántas personas se encuentran y si la mayoría tiene una emoción determinada.										
Excepciones	<table border="1"> <thead> <tr> <th>Paso</th><th>Acción</th></tr> </thead> <tbody> </tbody> </table>	Paso	Acción								
Paso	Acción										

	1	Si no detecta ningún rostro, al tocar la pantalla, escuchará un audio que le indicará que ningún rostro ha sido reconocido.
	2	Si los rostros están a una gran distancia en la que la cámara no puede enfocar, no detectará los rostros.

UC5	Suscripción de usuarios	
Requisito		
Versión	1.0	
Descripción	Acción que permite a los usuarios suscribirse para usar todas las funcionalidades de forma ilimitada.	
Precondiciones	No ser un usuario suscrito aún.	
Postcondiciones	Se genera un nuevo usuario.	
Secuencia normal		
	Paso	Acción

	1	El usuario selecciona la opción de suscribirse.
	2	Selecciona y verifica los datos de cuenta de Google Play.
	3	Confirma suscripción.
	4	Se le añade dicho coste mensual.

UC6	Acceso a monitor en tiempo real
Requisito	
Versión	1.0
Descripción	El usuario administrador podrá acceder a una aplicación web Single Page Application a través de un sistema de autenticación.
Precondiciones	Ser usuario administrador antes registrado en la base de datos.
Postcondiciones	Se accede a la aplicación web SPA

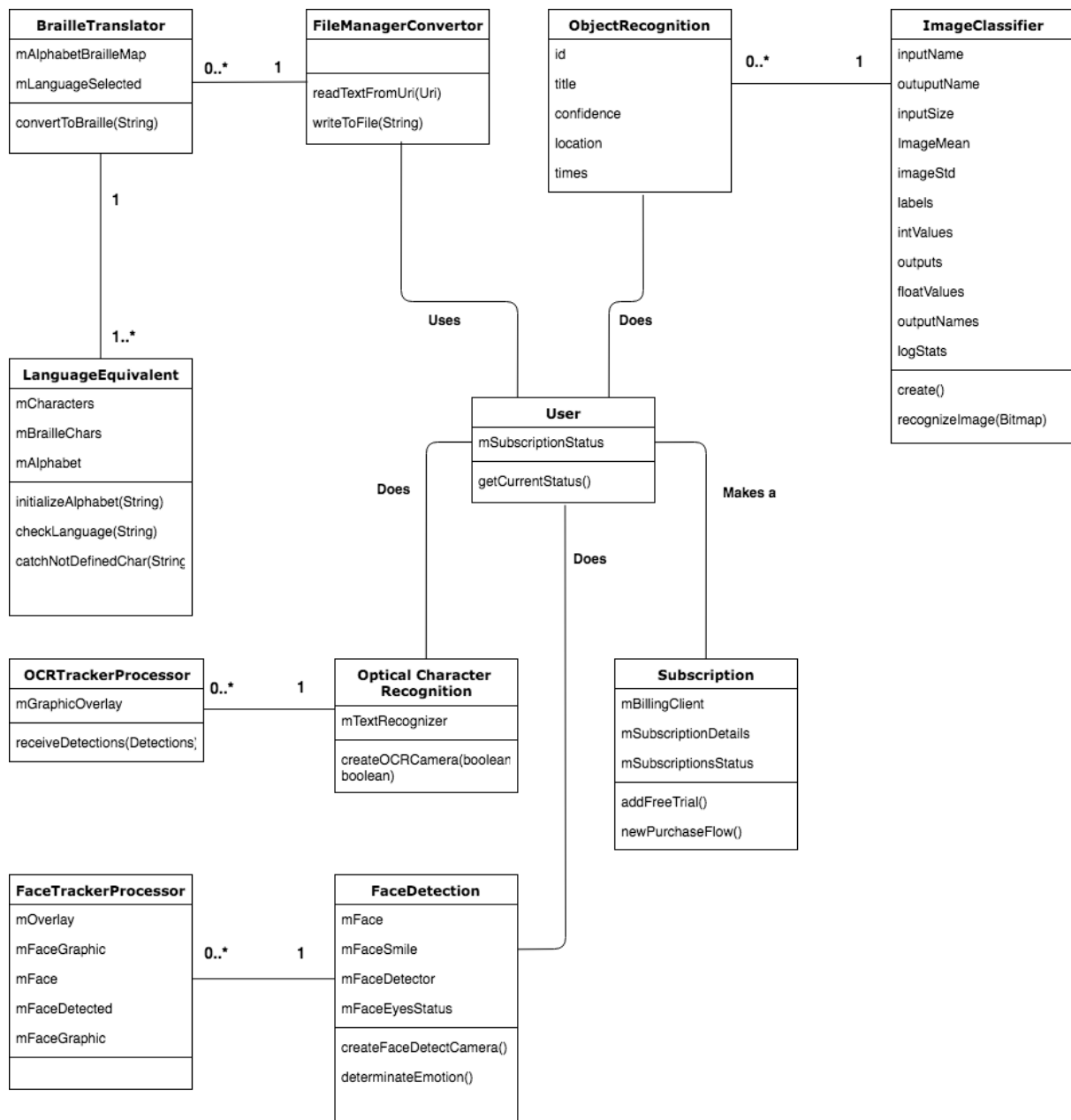
Secuencia normal		
	Paso	Acción
	1	El usuario ingresa a la dirección web de la aplicación y se encuentra en el Landing Page.
	2	Selecciona desde el menú la opción “Iniciar sesión”
	3	Ingresa sus datos de correo y contraseñas.
	4	El sistema de autenticación valida estos datos con la base de datos
	5	Siendo correctos, inicia sesión exitosamente. De lo contrario se le es indicado en pantalla.
Excepciones		
	Paso	Acción
	1	En el caso de olvidar la contraseña, selecciona la opción de “He olvidado mi contraseña” y un link será enviado a su correo asociado para re-establecerla.
	2	En el caso de que ingrese datos incorrectos, se

	<table> <tr> <td></td><td>le será notificado en pantalla.</td></tr> </table>		le será notificado en pantalla.
	le será notificado en pantalla.		

UC7	Visualización de objetos más reconocidos por usuario				
Requisito					
Versión	1.0				
Descripción	El administrador o persona con acceso a panel de base de datos de Firebase, podrá visualizar los objetos que más son reconocidos por los usuarios.				
Precondiciones	Haber iniciado sesión como usuario administrador en la aplicación web				
Postcondiciones					
Secuencia normal	<table> <tr> <th>Paso</th><th>Acción</th></tr> <tr> <td>1</td><td>El usuario navega entre el componente principal de la web.</td></tr> </table>	Paso	Acción	1	El usuario navega entre el componente principal de la web.
Paso	Acción				
1	El usuario navega entre el componente principal de la web.				

	2	Visualiza e interactúa con gráfica mostrando e interpretando datos de objetos más reconocidos por los usuarios hasta el momento.

3.6. Modelo de Clases de Análisis (ASI 4.1,ASI 5)

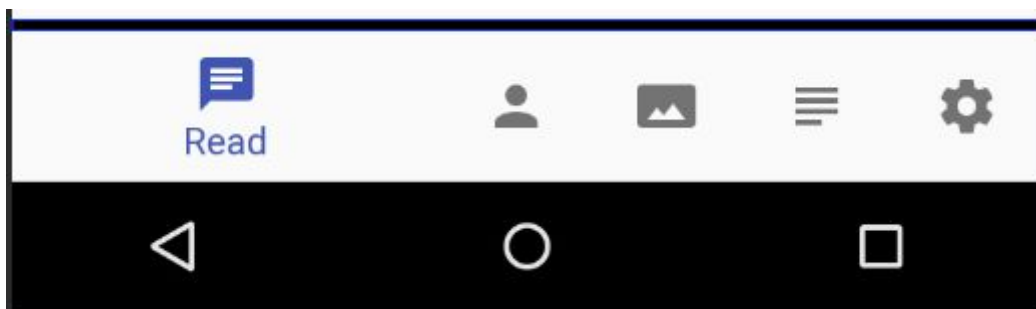


3.7. Interfaces de usuario (ASI 8)

3.7.1. Aspectos comunes de la interfaz de usuario (plantilla)

Aplicación móvil Android:

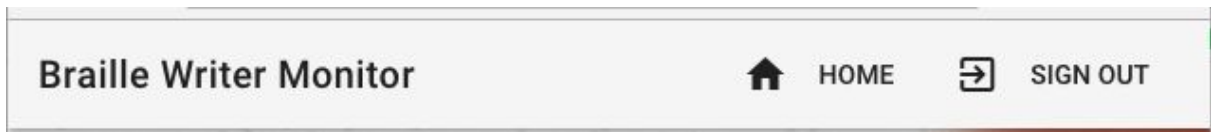
Se tendrá como aspecto común entre todas las pantallas de la aplicación móvil android un menú de navegación inferior que permitirá la navegación entre las funcionalidades principales, de forma más específica los fragmentos.



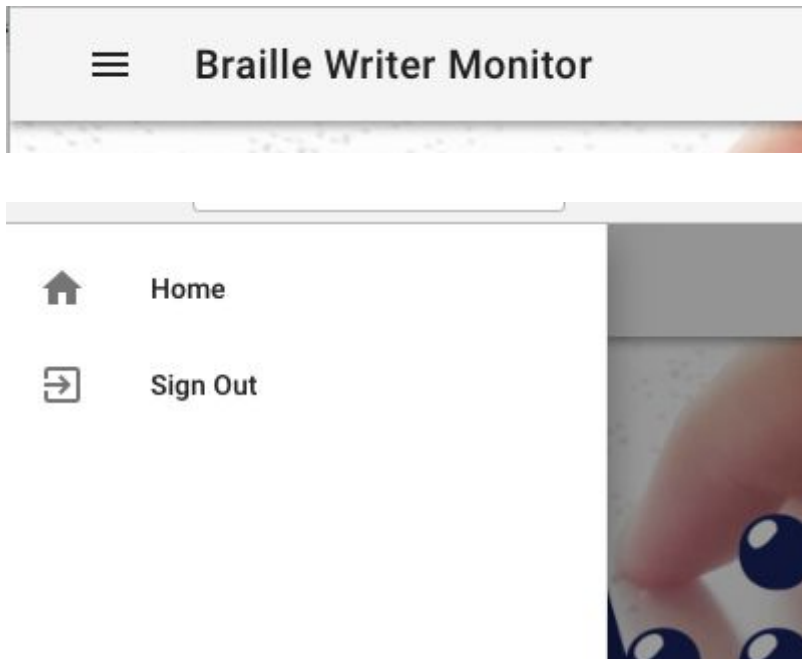
Aplicación web monitor en tiempo real:

Se tendrá como aspecto común un menú en la parte superior derecha y otro menú lateral a la izquierda cuando la pantalla supere una anchura de 960px. Respetando así buenas normas del diseño responsive.

Menú para pantallas con anchura superior a 769 píxeles. En tablets, laptops y ordenadores de escritorio.



Menú para pantallas móviles. Un menú hamburguesa el cual se muestra con una animación de transición en un menú de lateral vertical.



3.7.2. Perfiles de usuarios

- Aplicación móvil

Los perfiles de usuario serán usuario suscrito y no suscrito. El no suscrito solo podrá acceder a algunas funcionalidades de la aplicación mientras que el suscrito podrá acceder a todas.

- Aplicación web monitor

Los perfiles de usuario activo serán de usuario administrador no registrado y usuario administrador registrado, el usuario no registrado solo podrá acceder a la página principal Landing page y al login para identificarse.

El usuario registrado tendrá acceso a la página de monitorización principal.

3.8. Especificación de pantallas/ventanas

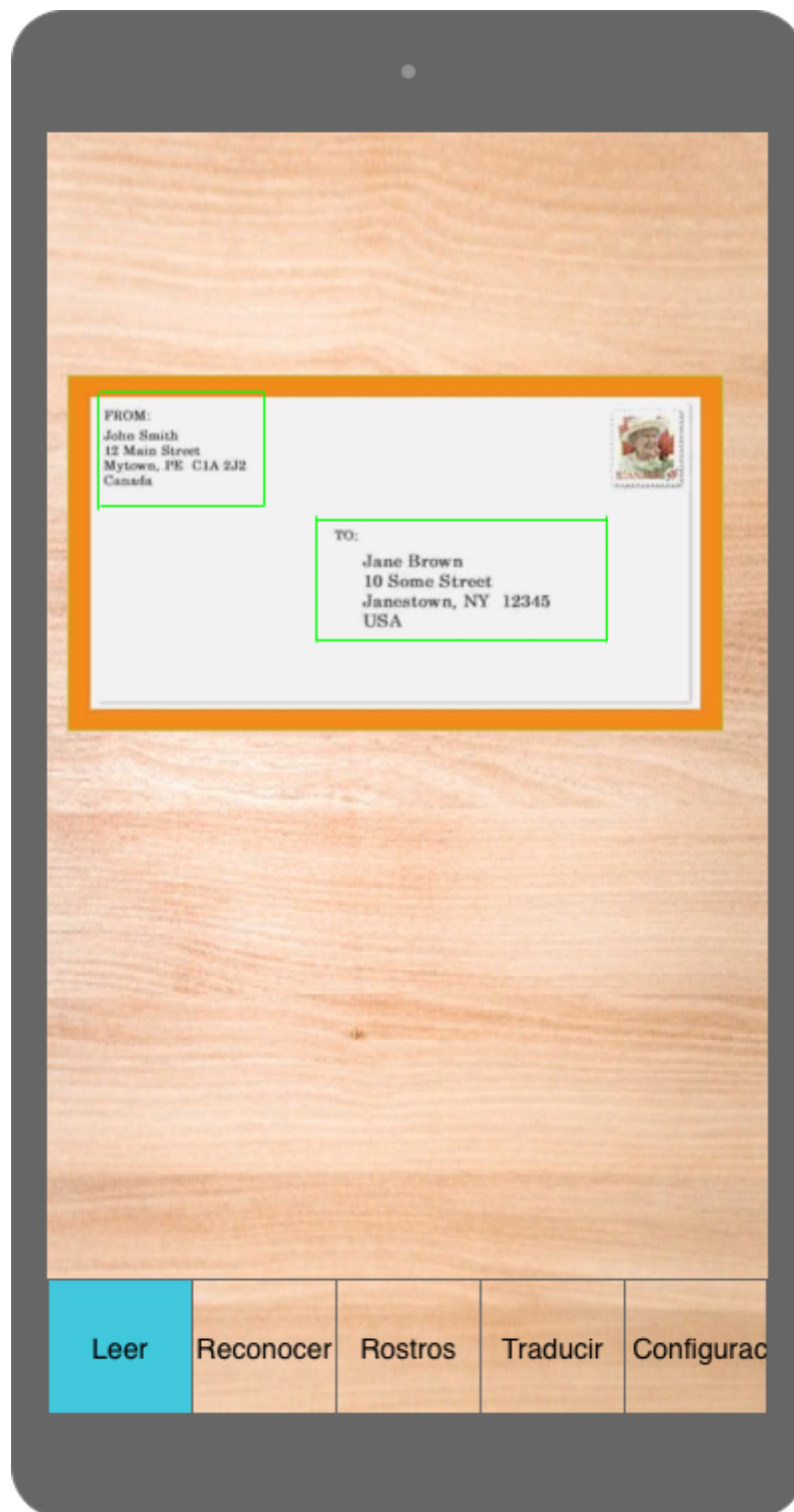
Para cada pantalla/ventana:

- Definición gráfica de la pantalla/ventana (sólo elementos particulares de la pantalla/ventana)
- Descripción de la funcionalidad de la pantalla.
- Hoja de descripción de campos



Ventana de leer.

Interfaz con la que se puede reconocer texto en tiempo real usando la cámara del móvil y al presionar cualquier parte de la pantalla se escuchará el texto que ha detectado en el idioma por defecto configurado en el dispositivo.



Ventana de leer.

Se muestra otro ejemplo de la utilidad de esta funcionalidad, al poder leer la carta, que dice y a quién va dirigida.



Ventana de reconocimiento de objetos.

En esta se realiza reconocimiento de objetos en tiempo real, mostrando en pantalla una lista de objetos clasificados por el modelo por orden de precisión. Se realiza un texto a audio del último objeto cuando el usuario toca cualquier parte de la pantalla.



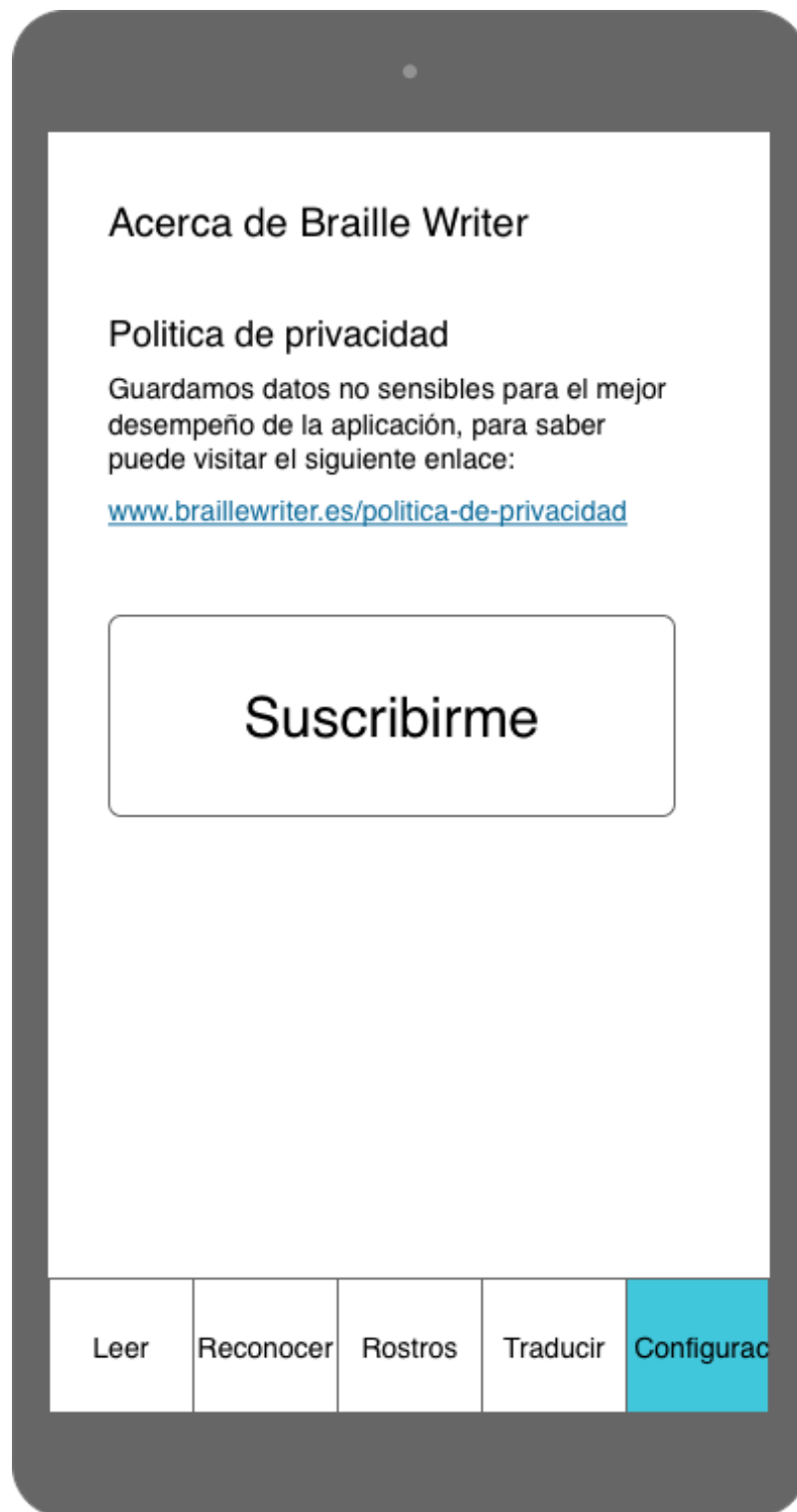
Ventana de reconocimiento de rostros

En esta se accede a la detección de rostros en tiempo real y reconocimiento de sus emociones, al tocar cualquier parte de la pantalla.



Ventana de traducir documento a formato Braille Unicode Grado 1

En esta ventana se permite elegir un archivo de formato .txt o .doc para traducir y ser guardado en el dispositivo.



Ventana de configuración

En esta ventana se podrá acceder a información relacionada con la política de privacidad de datos

3.8. Especificación de Requisitos Software ERS (ASI 9)

3.8.1. Introducción

Luego de iteraciones en el desarrollo técnico del proyecto, evolucionaron varios de los requisitos especificados en EVS, al principio del mismo. A continuación se presentan los requisitos de software.

3.8.2. Catálogo de requisitos de software

Requisitos de Interfaz

Requisitos de accesibilidad

Requisito	Nombre	Prioridad	Descripción
Req_1	Guía auditiva inicial	Alta	El sistema indica al usuario cuando a entrado en la aplicación y la actividad inicial.
Req_2	Guía auditiva entre las actividades	Alta	El sistema informático indica al usuario cuando intenta entrar a una actividad en la que ya se encuentra.

Req_3	Interfaz con alto contraste	Alta	La interfaz debe de tener una paleta de colores con alto contraste.
Req_4	Accesibilidad	Alta	Los elementos de la interfaz deben ser accesible para el lector de pantalla.
	Estructura MVVM	Media	Implementar la estructura recomendada por Android Developers

Requisitos Funcionales

Reconocimiento de Texto a través de la cámara y lectura en altavoz

Requisito	Nombre	Prioridad	Descripción
Req_5	OCR (Optical Character Recognition)	Alta	Con la integración de Inteligencia Artificial, Machine Learning, a través de la cámara se reconozca el texto visible en el entorno. Carteles, facturas, documentos, informes, etc
Req_6	Guardarlo como	Alta	Una vez delimitado el texto, guardarlo de manera temporal para luego procesarlo.

	cadena de caracteres		
Req_7	Audio de ese texto en tiempo real	Alta	Procesar el texto a través de la cámara de forma inmediata para leerlo en altavoz, también llamado Text to Speech.

Traducción de archivos a braille y guardado

Requisito	Nombre	Prioridad	Descripción
Req_8	Acceso a archivos de texto	Alta	La aplicación, con permiso del usuario, accede a los archivos del dispositivo y permite seleccionar uno.
Req_9	Conversión a braille	Alta	Una vez seleccionado el documento, se hace una conversión en base ASCII para hacerlo compatible con impresoras braille.
Req_10	Guardar como archivo	Alta	Guardar el texto reconocido y obtenido a través de la cámara como archivo en la memoria interna del dispositivo.

	braille unicode		
--	--------------------	--	--

Impresión en formato Braille y estándar

Requisito	Nombre	Prioridad	Descripción
Req_11	Imprimir braille	Medio	Las personas invidentes podrán imprimir en una impresora de tipo braille.
Req_12	Imprimir estándar	Medio	Poder imprimir un documento desde la aplicación.

Compartir archivos

Requisito	Nombre	Prioridad	Descripción
Req_13	Compartir archivos en braille	Bajo	Se podrá compartir mediante redes sociales y correo electrónico los archivos, a través de la aplicación.

Suscripción mensual de usuarios premium

Requisito	Nombre	Prioridad	Descripción
Req_14	Suscripción de usuarios	Alta	Los usuarios se podrán optar a una suscripción mensual de forma sencilla para usar todas las funcionalidades.

Monitor para visualización de datos en tiempo real

Requisito	Nombre	Prioridad	Descripción
Req_15	Visualización de datos	Baja	El sistema contará también con una aplicación web que permitirá la monitorización en tiempo real
Req_16	Usuario Administrador	Media	El sistema contará con la posibilidad de tener al menos un usuario administrador registrado en la base de datos
Req_17	Acceso de administrador	Media	Sólo el usuario administrador podrá acceder al monitor, a través de un sistema de autenticación y previo registro en la base de

			datos
--	--	--	-------

Facturación de productos y suscripciones

Requisito	Nombre	Prioridad	Descripción
	Permita adquirir productos y suscripción mensual		La aplicación debe de permitir comprar una suscripción de tipo mensual, pagando dentro de la aplicación
	API Facturación Oficial	Alta	Utilización de la API oficial de facturación de Android (In App Billing API V3) más reciente requerida para integrar facturación y compras en la aplicación.
	Servicio y conexión del servicio enlazado	Alta	Para conectar con la API, crear un servicio en background que efectúe todas las operaciones relacionadas. Enlazar este a la actividad para conectar con la In App Billing API Versión 3

4. Diseño de Sistemas de Información (DSI)

4.1 DSI 1: Definición de la arquitectura del sistema

El sistema se compone por una aplicación android que usa servicio de base de datos NoSQL en tiempo real de Firebase y librerías manejadas por el sistema de automatización de dependencias Gradle, de las cuales algunas de las más notorias se explican en detalle a continuación:

Tensorflow Mobile: Una adaptación para dispositivos móviles de Tensorflow. Permite usar la capacidad de proceso del mismo dispositivo móvil para realizar los inputs al modelo y recibir los outputs del mismo. Con modelos de redes neuronales pre entrenadas y re-entrenadas a lo largo del desarrollo con Python. El modelo pre-entrenado usado para el reconocimiento de objetos es Inception V3.

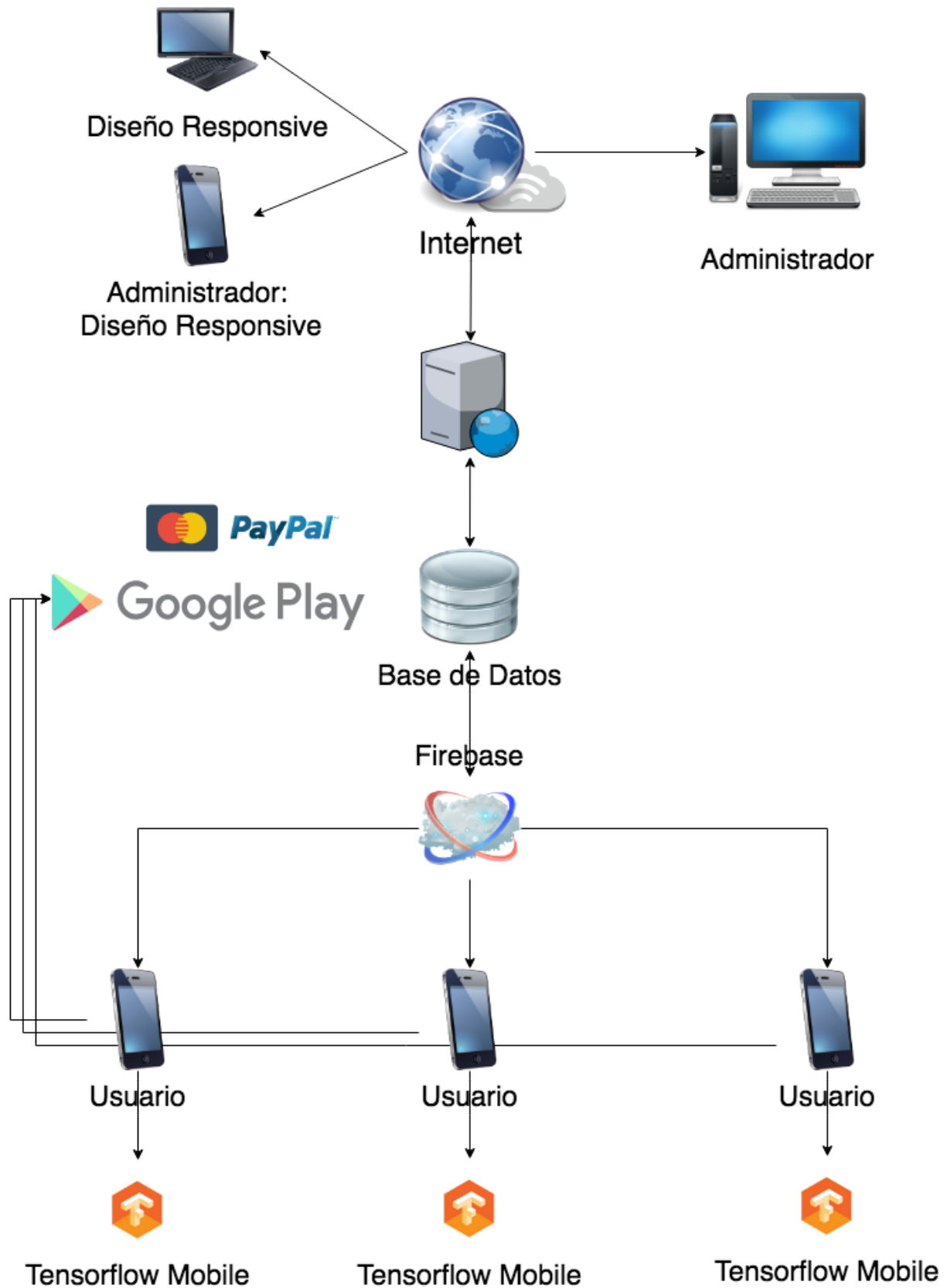
Mobile Vision: Librería para Optical Character Recognition, cuenta con modelos propios avanzados de Google y con múltiples versiones. Eficaz y precisa. Cubre la principal funcionalidad de extraer texto en tiempo real de imágenes a través de la cámara del móvil.

Firebase Auth y Database: Una plataforma que ofrece servicios en la nube para aplicaciones web y móviles. En el presente sistema se usa en ambos, una aplicación Single Page Application con Node.js y para la aplicación android.

Por parte de la plataforma web, se encuentra compuesta por node.js como parte del backend y Vue.js como frontend. Con un hosting en el servicio de computación en nube de Heroku. Se puede visualizar en dispositivos móviles y tablets gracias al diseño responsive.

Android InApp Billing API V3: La API oficial de android más reciente para la integración y el manejo de transacciones y operaciones relacionadas a la facturación. Permiten un flujo sincrónico de compra mejorado, las API para que se pueden rastrear fácilmente la propiedad de bienes consumibles o no, y el almacenamiento de datos de compras directas desde la aplicación en caché local.

4.1.1 Particionamiento Físico del Sistema de Información y especificación del entorno tecnológico



4.1.2 Catálogo de requisitos de Diseño (lenguajes que se van a usar, criterio de ubicación de módulos y datos, o sea un árbol de directorios donde muestre se va a cómo organizar la información de nuestra aplicación)

El diseño anteriormente ilustrado y especificado será abordado dividiendolo en los siguientes niveles de arquitectura:

Requisitos de móvil

Código	Requisito	Descripción
RDM1	Lenguaje Java	
RDM2	Android SDK 26 o mayor	Mínima versión para el IDE
RDM3	Android NDK 14 o mayor	Requerido en el IDE para construir en nativo (C/C++)
RDM4	Android API 23 o mayor	El dispositivo Android debe de ser mínimo versión 23.
RDM5	Modelo en estado congelado (frozen) en carpeta assets	Modelo clasificador adjuntado en la carpeta “assets” de src de proyecto y con formato .pb.
RDM6	Archivo de etiquetas (labels) del modelo en carpeta assets	Archivo con formato .txt que conjunto al modelo, permiten clasificar una clase dentro de una imagen.
RDM7	Espacio disponible	Espacio de memoria mínimo de 100mb en el dispositivo móvil android.
	Utilización de la API oficial de facturación de Android (In App Billing API V3)	Utilización de la api más reciente requerida para integrar facturación y compras en la aplicación.
	Enlazar un servicio compartido a la actividad para conectar con la In App Billing API V3	Para conectar con la API, crear un servicio en background que efectúe todas las operaciones relacionadas.

	Estructura MVVM de Android Components Architecture.	Implementar la estructura recomendada por Android Developers
--	---	--

Requisitos de Base de datos

Código	Requisito	Descripción
RDBD1	Base de datos no relacional	Debe de seguir una arquitectura noSQL.
RDBD2	Escalable	Escala a alrededor de 100.000 conexiones simultáneas y 1.000 escrituras/segundo en una instancia.
RDBD3	Tiempo real	Procesamiento en tiempo real para manejar cargas de trabajo cuyo estado cambia constantemente.

Requisitos servidor web para monitor

Código	Requisito	Descripción
RDS1	Hosting Node js	Servidor con entorno de ejecución para JavaScript basado en el lenguaje de programación ECMAScript Node js.
RDS2	Lenguaje ECMAScript 6 y Javascript	El desarrollo del nivel de servidor será en el lenguaje de programación ECMAScript 6 y Javascript.
RDS3	npm cli	Manejador de paquetes por defecto para Node.js.

Requisitos cliente web para monitor

Código	Requisito	Descripción
RDC1	Vue.js y Javascript	El desarrollo de la parte cliente del monitor web, es en el framework de frontend Vue.js, el cual tiene su propia sintaxis y lenguaje de programación para el renderizado.
RDC2	Diseño responsive	Debe de tener un diseño responsive que permite interactuar de forma cómoda y completa desde cualquier tamaño de pantalla: móviles, tablets y ordenador.
RDC3	256MB de memoria RAM o superior	El dispositivo de parte del cliente debe se tener una memoria ram de 256MB mínima.
RDC4	Conexión a internet	El dispositivo debe de poseer conexión a internet en aras de ver los cambios en tiempo real.
RDC5	Sistema Operativo cualquiera	La aplicación web puede ser accesada desde el navegador y visualizar los datos y sus cambios en tiempo real usando cualquier sistema operativo.

Requisitos de Machine Learning

Código	Requisito	Descripción
RML1	Lenguaje Python	Se utilizará Python para la modificación en los nodos y capas de los modelos.
RML2	Tensorflow	Se usará tensorflow como backend para el re-entrenamiento y cambios de los modelos clasificadores de deep Learning.
RML3	Modelos pre entrenados	Se usará el modelo clasificador de imágenes Inception-V3 para el reconocimiento de imágenes.

4.1.3 Catálogo de Excepciones. (Rangos o valores no válidos en la entrada de

datos, atributos obligatorios, formatos, específicos, etc

Código	Descripción	Fecha
EX1	Error de archivo elegido no guardado en el dispositivo. Se producirá cuando el dispositivo disponga de memoria disponible insuficiente para ser almacenado el documento traducido a formato braille.	10/12/2017
EX2	Error de formato seleccionado no posible de convertir. Se controla que sólo se elijan archivos de formato .txt y .doc, sin embargo este error se produce si antes de comenzar a traducir los caracteres, estos no pueden ser codificados a unicode braille.	16/12/2017
EX3	Cuando ningún objeto es reconocido, no se emite ningún audio.	16/12/2018
EX4	Cuando ninguna cara es detectada no se emite ningún audio.	16/12/2018
EX5	Cuando ningún texto es detectado, no se emite ningún sonido.	16/12/2018
EX6	Cuando un fragmento se destruye (Se navega de una vista de android a otra nueva) se detiene y apaga el audio Text To Speech de la misma. Si se está, por ejemplo, leyendo un documento muy largo que ha detectado, al cambiar de fragmento este audio se detiene automáticamente para no interferir con los demás.	02/02/2018

4.2 DSI 3: Diseño de la interfaz de usuario

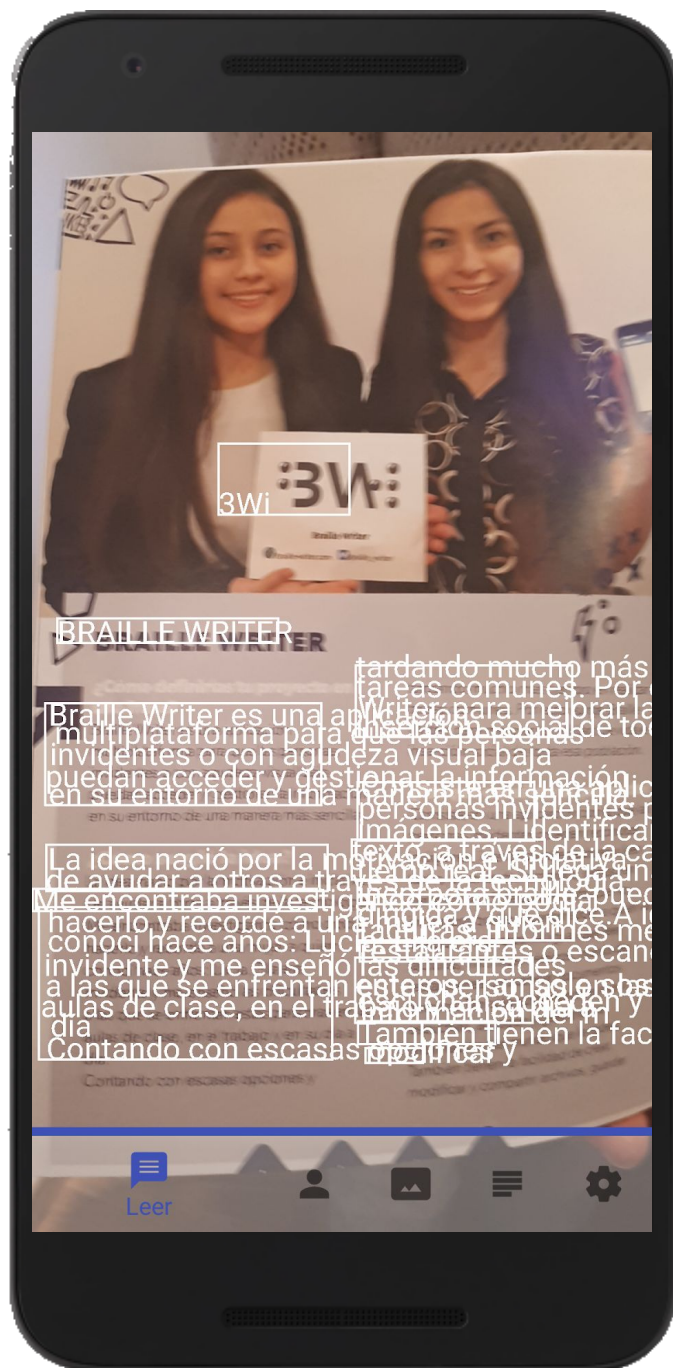
4.2.1 Modelo de navegación de interfaz de Pantalla

La aplicación android permite la navegación entre funcionalidades ofrecidas a través de fragmentos que cambian y se reemplazan con la interacción de un menú inferior.

Se sigue una estructura MVVM, también llamada Android Components Architecture, reciente y recomendada actualmente por la comunidad.

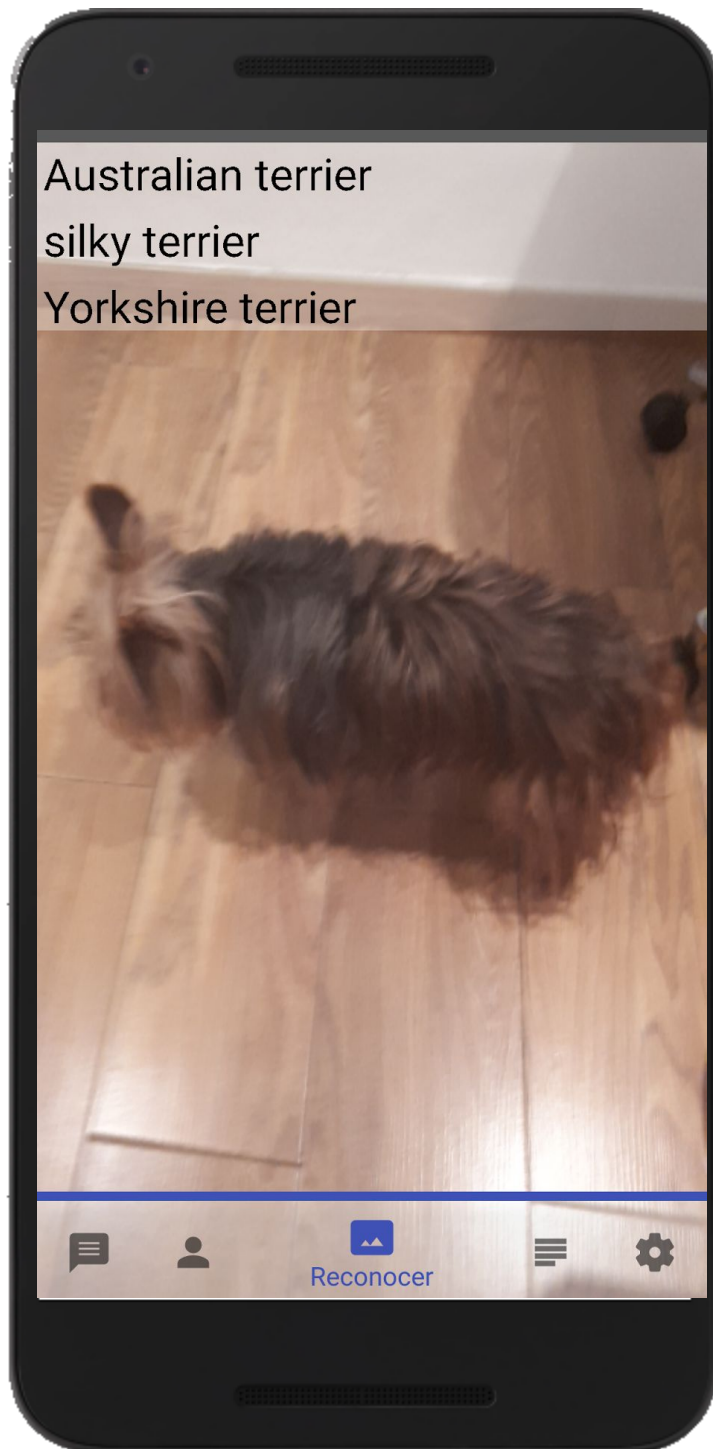
Se tiene una actividad `homeActivity` y en que con un `FragmentManager` se permite el cambio dinámico de Fragmentos como cada vista individual desplegada. Se crea un método static para crear nuevas instancias de estos y permitir tener un sólo fragmento seleccionado bajo el nombre de variable “`mFragment`”.

Cabe destacar, que al crearse por primera vez la actividad, se crea y bindea un servicio para la InApp Billing (Facturación de Android). El cual permite actualizar y establecer el estatus actual del usuario (suscrito o no suscrito). Al suceder cambios de configuración como la rotación de pantalla esta actividad se vuelve a crear y por ende, el servicio y la `serviceConnection` pierden ya que la actividad a la que están bindeados no existe. Se debe entonces, añadir al `ViewModel` un `LiveData` de ese Servicio, para que sobreviva a los cambios, además de bindearlo a la Aplicación como tal y no solo a la actividad.



Ventana de leer.

Interfaz con la que se puede reconocer texto en tiempo real usando la cámara del móvil y al presionar cualquier parte de la pantalla se escuchará el texto que ha detectado en el idioma por defecto configurado en el dispositivo.



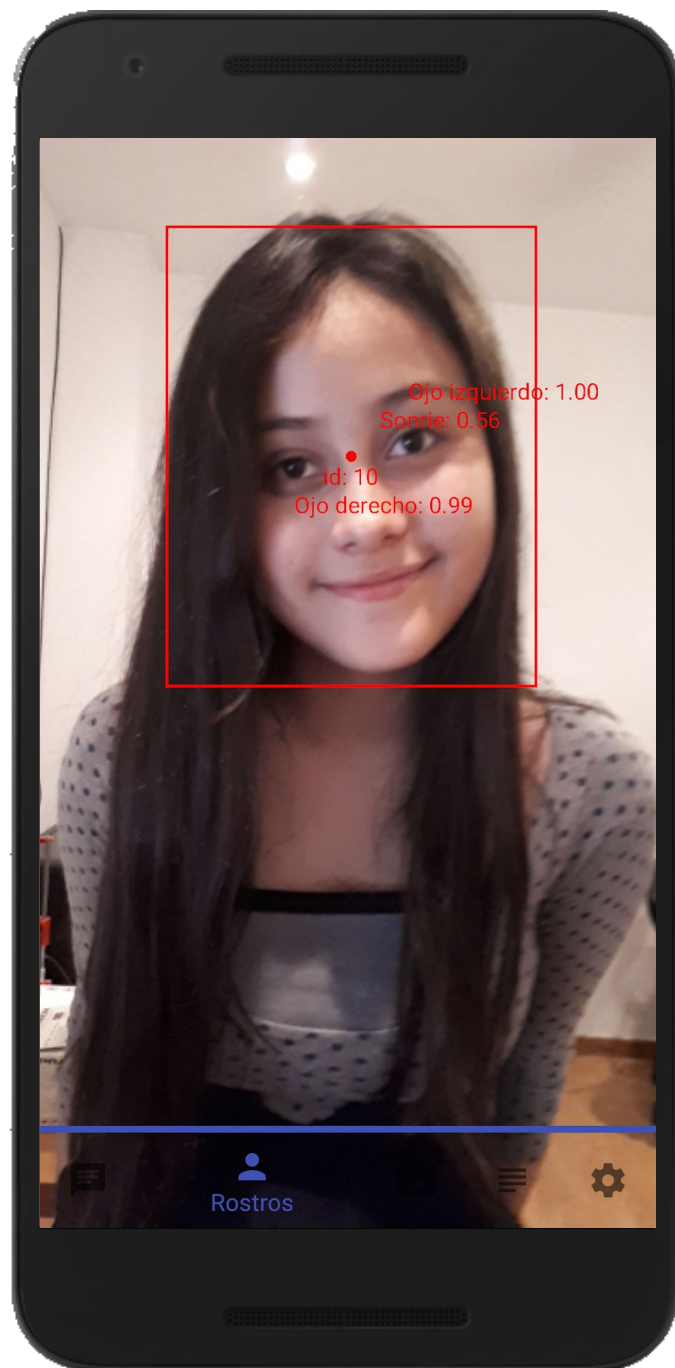
Ventana de reconocimiento de objetos.

En esta se realiza reconocimiento de objetos en tiempo real, mostrando en pantalla una lista de objetos clasificados por el modelo por orden de precisión.



Ventana de reconocimiento de rostros sin estar suscrito

Se realizan peticiones a la api para verificar el estatus del usuario, en caso de no estar suscrito, al entrar al fragmento se muestra esta pantalla invitando a suscribirse.



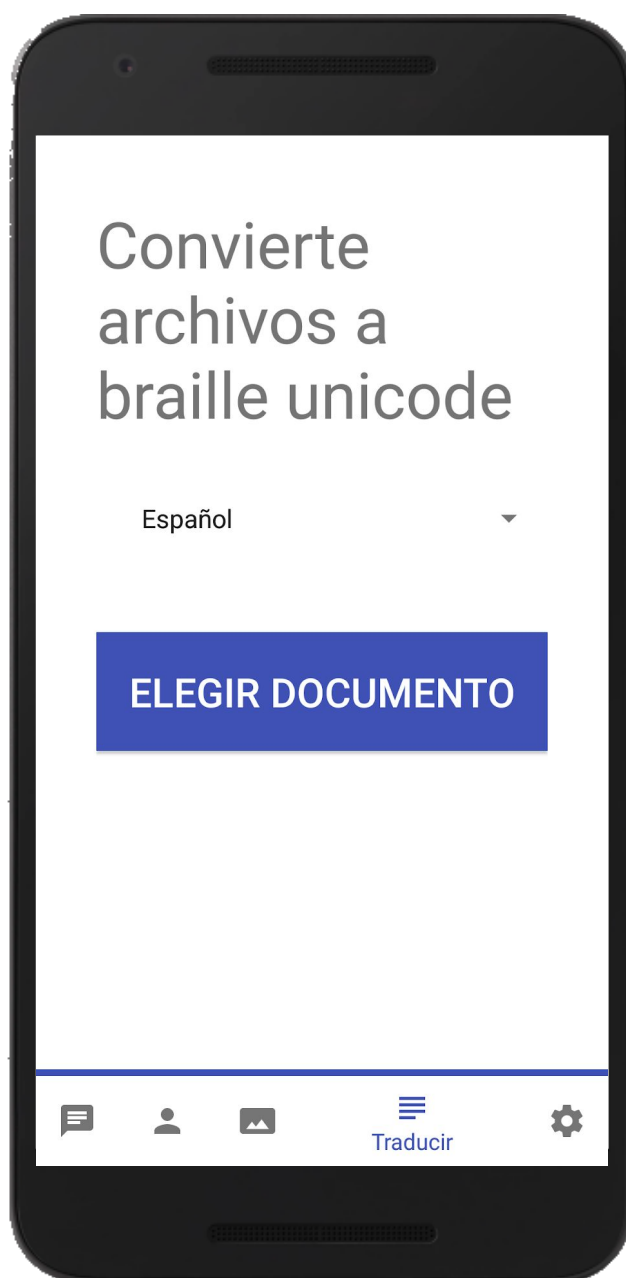
Ventana de reconocimiento de rostros

En esta se accede a la detección de rostros en tiempo real y reconocimiento de sus emociones, al tocar cualquier parte de la pantalla.



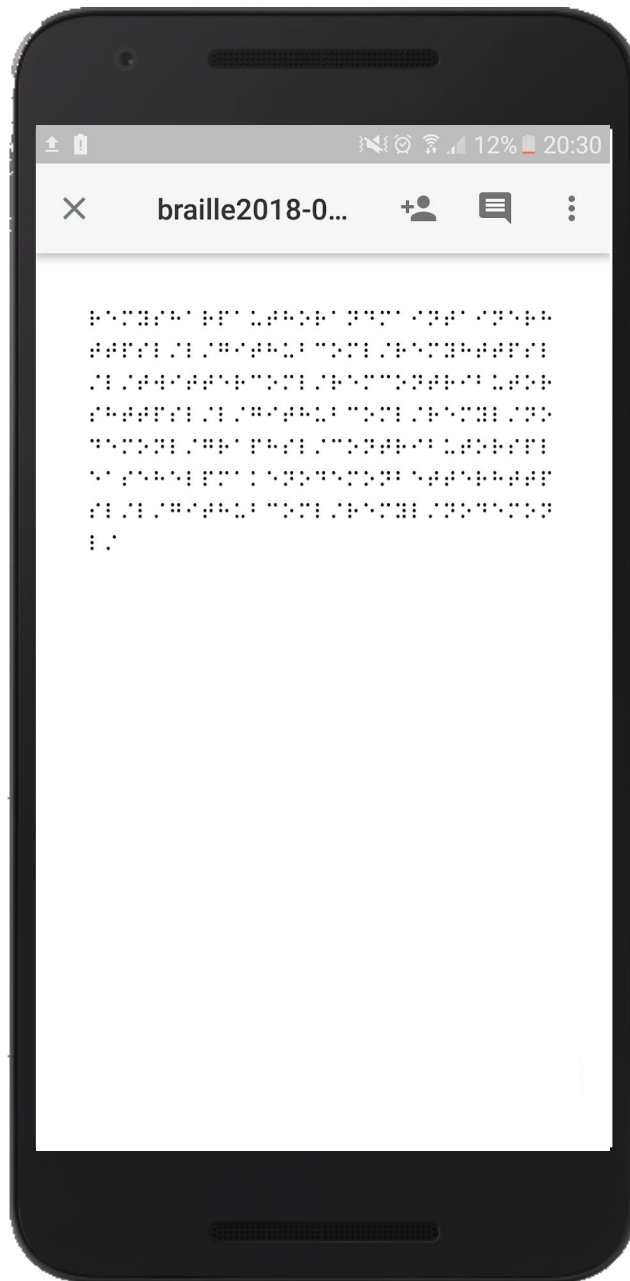
Ventana de configuración

En esta ventana se podrá acceder a información relacionada con la política de privacidad de datos



Ventana de traducir documento a formato Braille Unicode Grado 1

En esta ventana se permite elegir un archivo de formato .txt o .doc para traducir y ser guardado en el dispositivo.



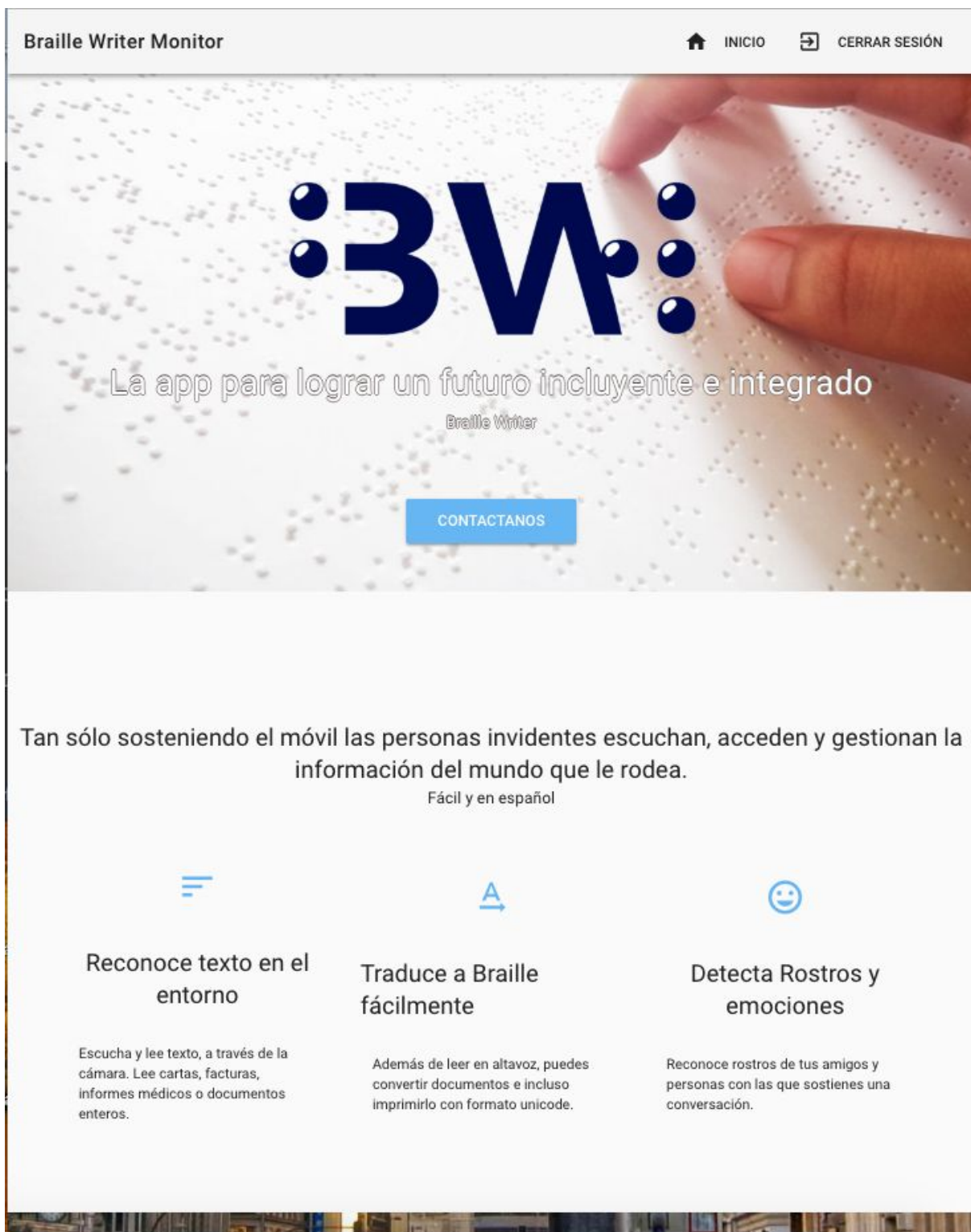
Ventana luego de archivo traducido a braille

Ventana emergente para mostrar el archivo traducido a braille unicode.



Página Home, mostrada por el componente Home de la SPA con Vue y Node.

En esta se muestran gráficas acerca del uso de la aplicación android por parte de los usuarios. Como cuales son los objetos que han sido más reconocidos. La gráfica está hecha con Echarts, conjunto a la reactividad profunda de Vue y firebase, se actualiza en tiempo real. Ofrece un



Página Landing que se muestra como portada, independientemente si has iniciado sesión o no.



Iniciar Sesión

Email*

becahere@gmail.com

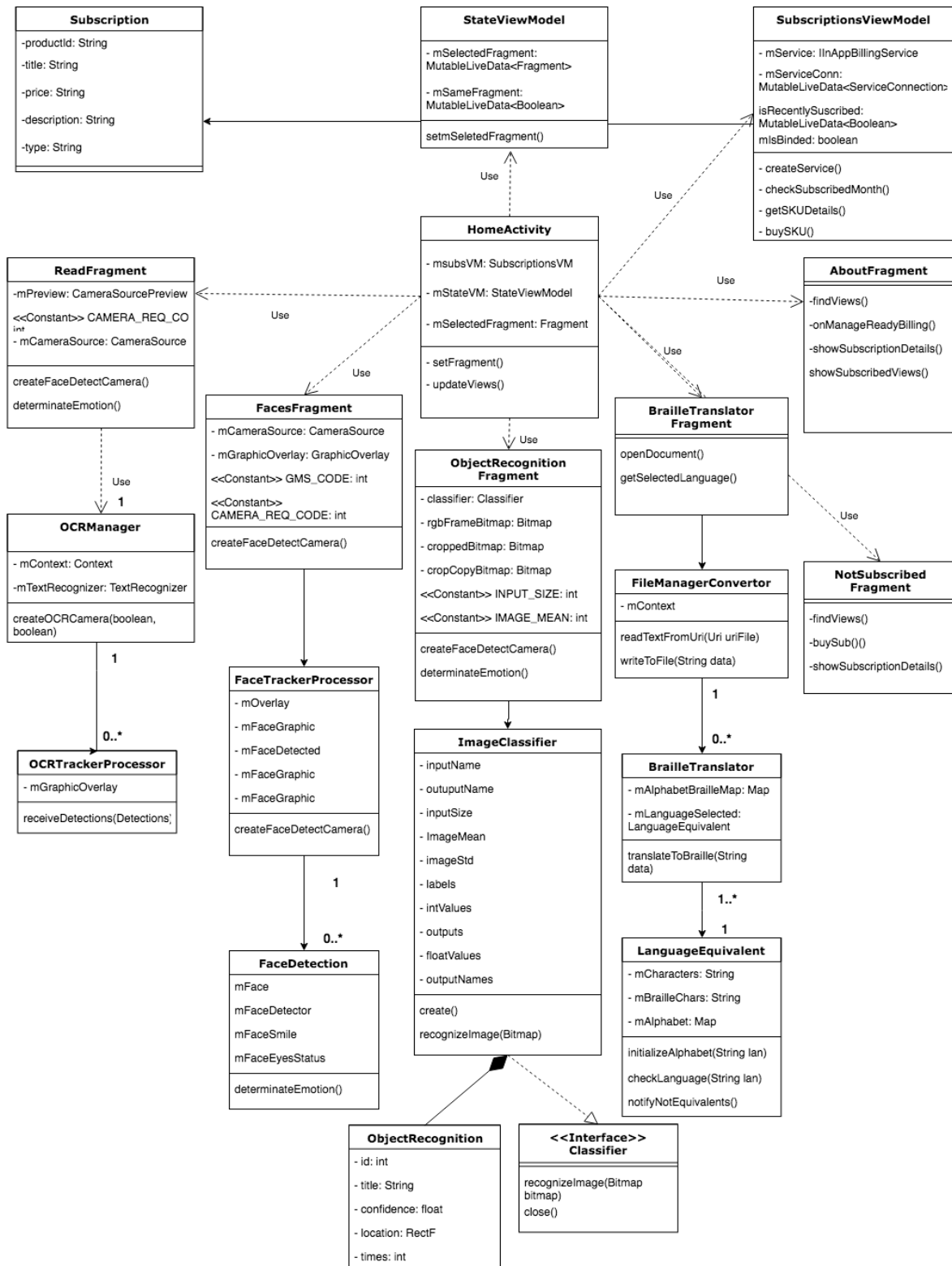
Password*

.....

INICIAR SESIÓN

Página de inicio de sesión, en forma responsive, para pantallas de anchura menores a 765px

4.2 DSI 4: Modelo de clases de diseño



4.4 DSI 6: Modelo físico de datos

La aplicación utiliza una base de datos Firebase para almacenar información de los objetos más reconocidos y el número de veces que usuarios han iniciado el periodo de prueba gratis o iniciado la suscripción mensual. Por lo tanto, se conecta a la base de datos tanto en la aplicación android como en el monitor web en tiempo real.

A su vez, se tiene una tabla para usuarios administradores que pueden acceder al monitor en tiempo real de la aplicación. Se almacena el correo electrónico, la contraseña encriptada y la fecha en que se efectuó el registro.

Firebase Realtime es una base de datos no relacional, alojada en la nube con la que Los datos se almacenan como JSON y se sincronizan en tiempo real con cada cliente conectado. Se pueden crear tablas con JSON, con una interfaz gráfica o de programáticamente desde clientes en aplicaciones web o móviles.

Las tablas tienen configuradas las llamadas “rules”, las cuales son reglas que seguir al escribir o leer de estas. En las reglas se pueden implementar condiciones como sólo permitir acceso si se encuentra autenticado o no.

Las contraseñas de los usuarios administradores registrados para acceder al monitor web en tiempo real, se les puede configurar parámetros de hash para establecer el algoritmo de encriptación e incluso la sal.

```
hash_config {  
  algorithm: SCRYPT,  
  base64_signer_key:  
    LGTuu2vg4ZQ17EDfQTYvfPMYN2dKt8Sclyt+V5CwzCxiG1moUdyKj/tM5+aISi4PhBAsBpNRBsp  
    umdXXkitcYA==,  
  base64_salt_separator: Bw==,  
  rounds: 8,  
  mem_cost: 14,  
}
```

Los datos relevantes a los usuarios que hayan iniciado periodo de prueba gratuita de suscripción o ya estén suscritos son accedidos y almacenados con la Android Inapp Billing API version 3. Con esta y a través del servicio creado y bindeado a la actividad por ViewModel, se actualizan los datos y se guarda un caché en el dispositivo local móvil durante 1 hora antes de volver a hacer una petición con la intención de cerciorarse si está o sigue suscrito o no.

Usando esta api oficial de Android se sigue rastro del número de personas que han iniciado alguno de los estados de compra: usuarios suscritos de pago, usuarios de prueba gratuita temporal o usuarios que hayan comprado otros productos, sin necesidad de almacenar datos sensibles más que la cuenta de Google.

Los productos integrados en la aplicación gestionados son artículos cuya información de propiedad se rastrea y administra a través de Google Play. Cuando un usuario adquiere un artículo integrado en la aplicación gestionado, Google Play

almacena la información de la compra de cada artículo por cada usuario. Esto permite realizar posteriormente, y en cualquier momento, consultas a Google Play para recuperar el estado de los artículos que un usuario en particular haya adquirido. Esta información se conserva en los servidores de Google Play incluso si el usuario desinstala la aplicación o cambia de dispositivo.

Se puede monitorizar y visualizar estos datos desde el monitor web, sólo quienes sean administradores.

braille-writer-f585e

objects

 Australian terrier

 Scotch terrier


 acoustic guitar

 bannister

 binder

 bookcase

 briard

 cairn

 chest

 computer keyboard

 container ship

desk

confidence: 0.41572701930999756

id: "313"

 location

times: 10

title: "desk"

 desktop computer

 electric fan

 home theater

 iron

 laptop

 liner

 menu

 monitor

 nematode

 notebook

 piggy bank

4.5 DSI 8 y 4: Especificaciones detallada de componentes

Se realizaron varios planteamientos y pseudocódigo para el desarrollo de la aplicación. Entre los métodos más importantes, se encuentran:

```
startTranslationToBraille(file){  
    obtener lenguaje seleccionado  
    Leer archivo y traducirlo  
    Si (Se ha traducido){  
        Escribir traducción Braille a un nuevo archivo y notificar al usuario  
    } Sí no{  
        No escribir un nuevo archivo y notificar a usuario  
    }  
}  
  
readTextFromUri(file, language){  
    Creamos traductor a Braille del lenguaje de documento seleccionado  
    Lee linea por linea y mientras esta no sea nula{  
        traducimos y añadimos a conjunto traducido  
    }  
}  
  
translateToBraille(linea){  
    Transformamos la línea a mayúscula
```



```

    Por cada carácter en la línea {
        buscar el equivalente a braille
        añadir a línea traducida
    }
}

checkSubscribedMonth(){
Si (está bindeado el InAppBillingService a la actividad y existe conexión al servicio
de Billing){
    Realizar petición de suscripciones activas del usuario en la app
    Si (petición correcta){
        Si (se encuentra la suscripción mensual activa) Suscrito a verdadero
    }
}
}

getMostConfidentObject(){
    Recorrer los últimos objetos reconocidos
    Se obtiene su certeza de clasificación
    Si (el en el actual, el modelo tiene más certeza){
        es el con seguro hasta ahora y se obtiene su nombre para audio
    }
    se actualiza o crea en base de datos
}

```

4.6 DSI 9: Diseño de migración y carga inicial de datos

La aplicación no importa datos de alguna aplicación anterior. Sin embargo, es necesario que se carguen algunas dependencias antes de ser usada y estos son descargados automáticamente al abrirse por primera vez.

En la aplicación web, una vez desplegada y por consiguiente, ya haber implementado todas las dependencias, se cargarán los datos de la base de datos a los gráficos de la ventana de inicio. Estos datos se actualizarán cada vez que se realicen cambios y actualicen con el uso de la aplicación móvil, en tiempo real.

4.7 DSI 10: Plan de prueba

4.7.1 Especificación del Entorno de Pruebas

Al realizar el estudio inicial de un subsistema de información de nuestra aplicación centraremos el plan de pruebas en modelo de pruebas unitarias, ya que no se realizará integración aun.

Aplicación móvil

Se usará Gradle y pruebas de unidades instrumentadas de Android Test.

Aplicación web

Se usará Mocha js.

4.7.1 Especificación Técnica de Niveles de Prueba

Plan de pruebas unitarias

Código	Descripción de prueba	Tipo	Fecha
PR 1	Lectura de texto (OCR) de un documento impreso a través de la cámara.	F	15/11/2018
PR 2	Lectura de texto en tiempo real de un documento impreso a través de la cámara.	F	22/11/2018
PR 3	Text to speech de lectura de texto de un letrero de la calle.	F	23/11/2018
PR 4	Elegir un documento de texto para convertir a braille unicode utilizando el framework de android.	F	10/11/2018
PR 5	Traducir un documento de texto a braille con codificación UTF-8 y guardarlo.	F	10/11/2018
PR 6	Traducir un documento de texto con codificación UTF-16 a braille, guardarlo y abrirlo con aplicación definida para visualización de archivos.	F	10/11/2018
PR 7	Reconocer un objeto, portátil sobre escritorio.	F	24/11/2018
PR 8	Detectar rostro humano.	F	24/11/2018
PR 9	Interfaz de navegación entre actividades con gestos sobre la pantalla.	I	3/02/2018
PR 10	Reconocimiento de emociones en rostros humanos detectados.	F	10/02/2018
PR 11	Prueba de carga de nuevos modelos re entrenados.	F	10/02/2018
PR 12	Registro de usuario.	F	11/02/2018
PR 13	Interfaz con fragmentos y menú inferior.	I	01/03/2018
PR 14	Prueba de almacenamiento de base de datos en tiempo real.	F	08/03/2018
PR 15	Prueba de inicio de sesión de usuario.	F	08/03/2018

PR 16	Traducción a braille de un documento largo de texto. De peso 1 mb.	F	10/04/2018
PR 17	Visualización en tiempo real de datos en el monitor.	F	10/04/2018
PR 18	Diseño responsive en monitor web.	I	10/04/2018
PR 19	Prueba de gráfica de word cloud.	I	10/04/2018
PR 20	Prueba de gráfica de usuarios.	I	10/04/2018
PR 21	Prueba de navegación entre fragmentos android cuando se procesa la lectura de un documento y se está reproduciendo un audio con razón verificar si se detiene el mismo.	I	15/04/2018
PR 22	Prueba de compra de suscripción mensual bajo condición de tester	F	19/05/2018
PR 23	Prueba de compra de suscripción mensual bajo condición de tester sin dinero en cuenta, transacción fallida a propósito.	F	10/06/2018
PR 24	Prueba de iniciación de tiempo prueba gratis por 12 días de la suscripción mensual bajo condición de tester.	F	1/06/2018
PR 25	Prueba de iniciación de tiempo de prueba gratis por 12 días de la suscripción mensual, sin condición de tester.	F	1/06/2018

4.8 DSI 11: Establecimiento de requisitos de implantación

4.8.1 Especificación de Requisitos de implantación

Aplicación móvil

El proceso de publicación y lanzamiento de la aplicación a la tienda Google Play requiere pasar pasos necesarios en aras habilitar correctamente el proceso de compra y suscripción dentro de la app.

Primero era imperante la publicación de la aplicación sin los permisos de Billing, por lo que es necesario para vender suscripciones correctamente, lanzar una segunda versión. Se debe registrar una cuenta de comerciante y un ID e información de la suscripción. Se crea una APK firmada, se sube y se espera a ser aprobada.

Se debe de tener en el manifest la siguiente versión de la actual, esta debe ser única, antes no publicada ni subida a Google Play de ninguna forma.

Se deben agregar cuentas de gmail a la lista de testers de productos o suscripciones de la aplicación en Google Play console para que estos no paguen dinero real y se pueda acceder al producto de suscripción de forma gratuita.

Se encuentra disponible en:

<https://play.google.com/store/apps/details?id=com.rebecasarai.brailewriter>

Aplicación web

Los requisitos de esta son la instalación de Node.js en el servidor en nube para su despliegue. Usando Heroku , se realiza un push desde Heroku CLI a la cuenta, consecuentemente un build y su despliegue para ser accedido online. Creando un archivo server.js, para servir la aplicación desde node y usando Vue como frontend. Se modificó también el archivo package.json para poder indicar el comando necesario de ser ejecutado para el deploy, node server.js

Comandos utilizados:

```
heroku create brailewriter
```

```
heroku config:set NODE_ENV=production --app brailewriter
```

```
npm install express --save
```

```
npm run build
```

```
git init
```

```
heroku git:remote --app braille writer
```

```
git add . && git commit -a -m "Adding files."
```

```
git push heroku master
```

Debido a que Vue es solo una biblioteca de frontend, la forma más fácil de alojarlo y hacer cosas como servir recursos es crear un sencillo script amigable para Express que Heroku pueda usar para iniciar un mini servidor web. Lea rápidamente en Express si aún no lo hizo.

Esto servirá un directorio dist. dist es un directorio predefinido que Vue.js construye, que es una versión comprimida y minificada de su sitio. Construiremos esto y luego le diremos a Heroku que ejecute server.js para que Heroku albergue este directorio dist.

Se encuentra disponible en: <https://brailewriter.herokuapp.com/>

Productos de uso

1. Código de la aplicación.

Adjunto en carpeta de “Proyecto”

2. Scripts de creación de la base de datos.

Script para crear o importarla:

```
{
  "objects" : {
    "Australian terrier" : {
      "confidence" : 0.6337074041366577,
      "id" : "4",
      "location" : {
        "bottom" : 0,
        "empty" : true,
        "left" : 0,
        "right" : 0,
        "top" : 0
      },
      "times" : 3,
      "title" : "Australian terrier"
    },
    "Scotch terrier" : {
      "confidence" : 0.11409337818622589,
      "id" : "109",
      "location" : {
        "bottom" : 0,
        "empty" : true,
        "left" : 0,
        "right" : 0,
        "top" : 0
      },
      "times" : 1,
      "title" : "Scotch terrier"
    },
    "acoustic guitar" : {
      "confidence" : 0.29870733618736267,
      "id" : "345",
      "location" : {
        "bottom" : 0,
        "empty" : true,
        "left" : 0,
        "right" : 0,
        "top" : 0
      },
      "times" : 1,
      "title" : "acoustic guitar"
    },
    "bannister" : {
      "confidence" : 0.15925894677639008,
      "id" : "718",
      "location" : {
```

```

    "bottom" : 0,
    "empty" : true,
    "left" : 0,
    "right" : 0,
    "top" : 0
  },
  "times" : 1,
  "title" : "bannister"
},
"binder" : {
  "confidence" : 0.16550354659557343,
  "id" : "835",
  "location" : {
    "bottom" : 0,
    "empty" : true,
    "left" : 0,
    "right" : 0,
    "top" : 0
  },
  "times" : 1,
  "title" : "binder"
},
"bookcase" : {
  "confidence" : 0.11079055815935135,
  "id" : "300",
  "location" : {
    "bottom" : 0,
    "empty" : true,
    "left" : 0,
    "right" : 0,
    "top" : 0
  },
  "times" : 6,
  "title" : "bookcase"
},
"briard" : {
  "confidence" : 0.13346028327941895,
  "id" : "208",
  "location" : {
    "bottom" : 0,
    "empty" : true,
    "left" : 0,
    "right" : 0,
    "top" : 0
  },
  "times" : 2,
  "title" : "briard"
},
"cairn" : {
  "confidence" : 0.18675774335861206,
  "id" : "91",
  "location" : {
    "bottom" : 0,
    "empty" : true,
    "left" : 0,
    "right" : 0,
    "top" : 0
  },
  "times" : 1,
  "title" : "cairn"
},
"chest" : {
  "confidence" : 0.19681160151958466,
  "id" : "762",
  "location" : {
    "bottom" : 0,
    "empty" : true,
    "left" : 0,
    "right" : 0,

```



```

    "top" : 0
  },
  "times" : 8,
  "title" : "chest"
},
"computer keyboard" : {
  "confidence" : 0.8303995132446289,
  "id" : "543",
  "location" : {
    "bottom" : 0,
    "empty" : true,
    "left" : 0,
    "right" : 0,
    "top" : 0
  },
  "times" : 12,
  "title" : "computer keyboard"
},
"container ship" : {
  "confidence" : 0.29396453499794006,
  "id" : "243",
  "location" : {
    "bottom" : 0,
    "empty" : true,
    "left" : 0,
    "right" : 0,
    "top" : 0
  },
  "times" : 6,
  "title" : "container ship"
},
"desk" : {
  "confidence" : 0.41572701930999756,
  "id" : "313",
  "location" : {
    "bottom" : 0,
    "empty" : true,
    "left" : 0,
    "right" : 0,
    "top" : 0
  },
  "times" : 10,
  "title" : "desk"
},
"desktop computer" : {
  "confidence" : 0.23816241323947906,
  "id" : "550",
  "location" : {
    "bottom" : 0,
    "empty" : true,
    "left" : 0,
    "right" : 0,
    "top" : 0
  },
  "times" : 14,
  "title" : "desktop computer"
},
"electric fan" : {
  "confidence" : 0.1461353749036789,
  "id" : "512",
  "location" : {
    "bottom" : 0,
    "empty" : true,
    "left" : 0,
    "right" : 0,
    "top" : 0
  },
  "times" : 2,
  "title" : "electric fan"
}

```

```

},
"home theater" : {
  "confidence" : 0.16196076571941376,
  "id" : "696",
  "location" : {
    "bottom" : 0,
    "empty" : true,
    "left" : 0,
    "right" : 0,
    "top" : 0
  },
  "times" : 1,
  "title" : "home theater"
},
"iron" : {
  "confidence" : 0.205108180642128,
  "id" : "659",
  "location" : {
    "bottom" : 0,
    "empty" : true,
    "left" : 0,
    "right" : 0,
    "top" : 0
  },
  "times" : 1,
  "title" : "iron"
},
"laptop" : {
  "confidence" : 0.2039875090122223,
  "id" : "228",
  "location" : {
    "bottom" : 0,
    "empty" : true,
    "left" : 0,
    "right" : 0,
    "top" : 0
  },
  "times" : 1,
  "title" : "laptop"
},
"liner" : {
  "confidence" : 0.1492927521467209,
  "id" : "244",
  "location" : {
    "bottom" : 0,
    "empty" : true,
    "left" : 0,
    "right" : 0,
    "top" : 0
  },
  "times" : 1,
  "title" : "liner"
},
"menu" : {
  "confidence" : 0.28098082542419434,
  "id" : "803",
  "location" : {
    "bottom" : 0,
    "empty" : true,
    "left" : 0,
    "right" : 0,
    "top" : 0
  },
  "times" : 1,
  "title" : "menu"
},
"monitor" : {
  "confidence" : 0.41664132475852966,
  "id" : "869",

```

```

"location" : {
  "bottom" : 0,
  "empty" : true,
  "left" : 0,
  "right" : 0,
  "top" : 0
},
"times" : 18,
"title" : "monitor"
},
"nematode" : {
  "confidence" : 0.1380215436220169,
  "id" : "651",
  "location" : {
    "bottom" : 0,
    "empty" : true,
    "left" : 0,
    "right" : 0,
    "top" : 0
  },
  "times" : 6,
  "title" : "nematode"
},
"notebook" : {
  "confidence" : 0.11526108533143997,
  "id" : "552",
  "location" : {
    "bottom" : 0,
    "empty" : true,
    "left" : 0,
    "right" : 0,
    "top" : 0
  },
  "times" : 1,
  "title" : "notebook"
},
"patio" : {
  "confidence" : 0.31326329708099365,
  "id" : "679",
  "location" : {
    "bottom" : 0,
    "empty" : true,
    "left" : 0,
    "right" : 0,
    "top" : 0
  },
  "times" : 1,
  "title" : "patio"
},
"piggy bank" : {
  "confidence" : 0.1362011432647705,
  "id" : "931",
  "location" : {
    "bottom" : 0,
    "empty" : true,
    "left" : 0,
    "right" : 0,
    "top" : 0
  },
  "times" : 1,
  "title" : "piggy bank"
},
"printer" : {
  "confidence" : 0.22500766813755035,
  "id" : "556",
  "location" : {
    "bottom" : 0,
    "empty" : true,
    "left" : 0,

```

```

    "right" : 0,
    "top" : 0
  },
  "times" : 1,
  "title" : "printer"
},
"refrigerator" : {
  "confidence" : 0.178620845079422,
  "id" : "668",
  "location" : {
    "bottom" : 0,
    "empty" : true,
    "left" : 0,
    "right" : 0,
    "top" : 0
  },
  "times" : 1,
  "title" : "refrigerator"
},
"scale" : {
  "confidence" : 0.16047194600105286,
  "id" : "521",
  "location" : {
    "bottom" : 0,
    "empty" : true,
    "left" : 0,
    "right" : 0,
    "top" : 0
  },
  "times" : 1,
  "title" : "scale"
},
"screen" : {
  "confidence" : 0.43632152676582336,
  "id" : "510",
  "location" : {
    "bottom" : 0,
    "empty" : true,
    "left" : 0,
    "right" : 0,
    "top" : 0
  },
  "times" : 11,
  "title" : "screen"
},
"stretcher" : {
  "confidence" : 0.11464375257492065,
  "id" : "957",
  "location" : {
    "bottom" : 0,
    "empty" : true,
    "left" : 0,
    "right" : 0,
    "top" : 0
  },
  "times" : 1,
  "title" : "stretcher"
},
"syringe" : {
  "confidence" : 0.1089705303311348,
  "id" : "531",
  "location" : {
    "bottom" : 0,
    "empty" : true,
    "left" : 0,
    "right" : 0,
    "top" : 0
  },
  "times" : 7,

```

```

    "title" : "syringe"
  },
  "toilet tissue" : {
    "confidence" : 0.13260331749916077,
    "id" : "889",
    "location" : {
      "bottom" : 0,
      "empty" : true,
      "left" : 0,
      "right" : 0,
      "top" : 0
    },
    "times" : 2,
    "title" : "toilet tissue"
  },
  "wardrobe" : {
    "confidence" : 0.282188355922699,
    "id" : "317",
    "location" : {
      "bottom" : 0,
      "empty" : true,
      "left" : 0,
      "right" : 0,
      "top" : 0
    },
    "times" : 1,
    "title" : "wardrobe"
  },
  "water bottle" : {
    "confidence" : 0.4000895023345947,
    "id" : "958",
    "location" : {
      "bottom" : 0,
      "empty" : true,
      "left" : 0,
      "right" : 0,
      "top" : 0
    },
    "times" : 1,
    "title" : "water bottle"
  },
  "wing" : {
    "confidence" : 0.10437553375959396,
    "id" : "503",
    "location" : {
      "bottom" : 0,
      "empty" : true,
      "left" : 0,
      "right" : 0,
      "top" : 0
    },
    "times" : 2,
    "title" : "wing"
  }
}

```

3. Instrucciones y requisitos de instalación.

También incluidos en carpeta “código” .

Para Braille Writer se ejecute correctamente, es necesario tener lo siguiente:

Monitor en tiempo real (aplicación web)

- node.js - Para el backend y el despliegue del monitor de la aplicación web
- Vue.js - Marco frontend de Javascript, para que la Reactividad muestre los datos en tiempo real en los gráficos del monitor.
- Express.js - marco de aplicación de red node.js rápido para enrutar y servir la aplicación en Heroku, así como la máquina local.

Aplicación de Android (aplicación móvil nativa)

- Android Studio - Android Studio
- Android SDK 23 - Dado que es el SDK de compilación de destino
- Android NDK - Android NDK (para la implementación nativa de C ++ requerida por Tensorflow Mobile)
- Google Services Plugin: para implementar la API de Google Vision más rápido para detección de rostros y OCR
- Firebase Json File - El archivo Json requerido por Firebase para conectarse correctamente a Cloud Real Time Database, ya incluido en el proyecto Android
- Modelo de clasificador convolucional de red neuronal - Gradle ya está configurado para verificar si el modelo ya está en la carpeta de activos, si no, lo descarga nuevamente.

Instalación

Monitor de tiempo real

La aplicación web requiere Node.js (<https://nodejs.org/>) v8 + para ejecutar.

Se debe instalar las dependencias y devDependencies e inicie el servidor:

```
$ cd BrailleWriterMonitor
```

```
$ npm install
```

```
$ node server.js
```

Para el desarrollo en producción:

```
$ npm install --production $ NODE_ENV=production node brailewriter
```

Iniciar sesión en el monitor de tiempo real

Correo electrónico del usuario: becahere@gmail.com

contraseña: brailewriter

Android

Se debe tener instalado Android NDK y Google Services seleccionándolos desde el SDK Manager.

Sincronizar el proyecto gradle en Android Studio. Normalmente se descargan las dependencias. Si aún hay más dependencias por descargar y lo pide, entonces lo mejor es hacerlo.

Se debe comprobar si el modelo clasificador de imágenes está en la carpeta “assets”. Si no estaba allí, gradle-download está a cargo para descargarlo.

Se requiere que se pruebe en un dispositivo real y no en un emulador para que las funciones de reconocimiento a través de la cámara muestren lo que hace efectivamente.

Los tres gmail de los profesores se pueden agregar a la lista de Comprobadores de facturación en la aplicación en Google Play para que cualquiera de ellos pague dinero y pueda acceder al producto de suscripción de forma gratuita. También configuré un período de prueba gratuito para cualquiera antes de suscribirse, de esta manera es absolutamente gratis durante 12 días, y si la suscripción para entonces todavía está activa, entonces se cobran los 0,99 € en la cuenta.