



**UNIVERSIDAD  
NACIONAL  
DE LOJA**



*Facultad de la Energía, las Industrias y los Recursos Naturales No Renovables*

---

**CARRERA DE INGENIERÍA EN SISTEMAS**

**“APLICACIÓN MÓVIL PARA EL PROTOTIPADO DE INTERFAZ GRÁFICA DE USUARIO EN LA PLATAFORMA ANDROID MEDIANTE EL RECONOCIMIENTO DE BOCETOS UTILIZANDO VISIÓN ARTIFICIAL”**

*Tesis de grado previa a la obtención del título de Ingeniero en Sistemas.*

**AUTOR:**

*José Luis Aguilar Espinoza*

**DIRECTOR:**

*Ing. Edison Leonardo Coronel Romero, Mg. Sc.*

**LOJA - ECUADOR**

**2019**

## **Certificación**

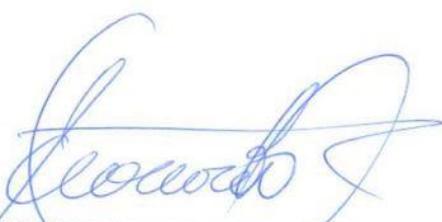
Ing. Edison Leonardo Coronel Romero, Mg. Sc.

**DIRECTOR DE TESIS.**

Certifico:

Que el egresado **José Luis Aguilar Espinoza**, realizó el trabajo de titulación denominado **“APLICACIÓN MÓVIL PARA EL PROTOTIPADO DE INTERFAZ GRÁFICA DE USUARIO EN LA PLATAFORMA ANDROID MEDIANTE EL RECONOCIMIENTO DE BOCETOS UTILIZANDO VISIÓN ARTIFICIAL”** bajo mi dirección y asesoramiento, mismo que fue revisado, enmendado y corregido minuciosamente. En virtud que la Tesis reúne, a satisfacción las cualidades de fondo y forma exigidas para un trabajo de este nivel, autorizo su presentación, sustentación y defensa ante el tribunal respectivo.

Loja, 22 de noviembre del 2018



.....  
Ing. Edison Leonardo Coronel Romero, Mg. Sc.

**DIRECTOR DE TESIS**

## **Autoría**

Yo, **JOSÉ LUIS AGUILAR ESPINOZA**, declaro ser autor del presente trabajo de titulación y eximo expresamente a la Universidad Nacional de Loja y a sus representantes jurídicos de posibles reclamos o acciones legales por el contenido de la misma.

Adicionalmente acepto y autorizo a la Universidad Nacional de Loja, la publicación del trabajo de titulación en el Repositorio Institucional – Biblioteca Virtual.

Firma: .....



Cédula: 0706339884

Fecha: Loja, 27 de junio del 2019

## Carta de autorización

Yo, **JOSÉ LUIS AGUILAR ESPINOZA**, declaro ser autor del trabajo de titulación: **“APLICACIÓN MÓVIL PARA EL PROTOTIPADO DE INTERFAZ GRÁFICA DE USUARIO EN LA PLATAFORMA ANDROID MEDIANTE EL RECONOCIMIENTO DE BOCETOS UTILIZANDO VISIÓN ARTIFICIAL”**, como requisito para optar al grado de: **INGENIERO EN SISTEMAS**; autorizo al Sistema Bibliotecario de la Universidad Nacional de Loja para que con fines académicos, muestre al mundo la producción intelectual de la Universidad, a través de la visibilidad de su contenido de la siguiente manera en el repositorio Digital Institucional:

Los usuarios pueden consultar el contenido de este trabajo en el RDI, en las redes de información del país y del exterior, con las cuales tenga convenio la Universidad.

La Universidad Nacional de Loja, no se responsabiliza por plagio o copia del trabajo de titulación que realice un tercero.

Para constancia de esta autorización, en la ciudad de Loja, el día veintisiete del mes de junio del dos mil diecinueve.

**Firma:**  .....

**Autor:** José Luis Aguilar Espinoza

**Cédula:** 0706339884

**Dirección:** Loja (Avenida Tiwintza y Reinaldo Espinoza)

**Correo Electrónico:** jlaguilare@unl.edu.ec

**Teléfono:** 072974359

**Celular:** 0980858483

### DATOS COMPLEMENTARIOS

**Director de Tesis:** Ing. Edison Leonardo Coronel Romero, Mg. Sc.

**Tribunal de Grado:** Ing. Hernán Leonardo Torres Carrión, Mg. Sc.

Ing. Luis Antonio Chamba Eras, Mg. Sc.

Ing. Roberth Gustavo Figueroa Díaz, Mg. Sc.

## **Dedicatoria**

A mis padres José Aguilar y Josefa Espinoza, quienes son el motivo de mi superación constante.

A mi "ñaño" Luis, quien es mi alegría, luz y fortaleza desde su nacimiento.

A mi hermana Vanessa Aguilar, por su gran apoyo.

# **Agradecimiento**

A Dios, por siempre estar ahí.

A mis padres José Aguilar y Josefa Espinoza, por su gran amor y apoyo incondicional.

A todos mis docentes, por los conocimientos impartidos en mi formación académica.

A mis amigos y personas especiales que siempre estuvieron presentes sin importar las circunstancias.

# Índice de Contenidos

Certificación.....	II
Autoría.....	III
Carta de autorización.....	IV
Dedicatoria.....	V
Agradecimiento.....	VI
Índice de Contenidos.....	VII
Índice de Figuras.....	XI
Índice de Tablas.....	XII
1. Título.....	1
2. Resumen.....	2
2.1 Abstract.....	3
3. Introducción.....	4
4. Revisión de Literatura.....	6
4.1 Interfaz gráfica de usuario.....	6
4.1.1 Interacción Humano - Computadora.....	6
4.1.1.1 Interfaces orientadas a la usabilidad.....	6
4.1.1.2 Herramientas de diseño de interfaces.....	6
4.1.2 Material Design para Android.....	7
4.2 Visión Artificial.....	8
4.2.1 Imagen digital.....	8
4.2.1.1 Matrices de píxeles.....	8
4.2.1.2 Bytes, bits y colores.....	9
4.2.1.3 Frecuencia de imagen.....	9
4.2.1.4 Herramientas.....	9
4.2.2 Librerías de visión artificial.....	9
4.2.2.1 OpenCV.....	9
4.2.2.2 TensorFlow.....	9
4.2.2.3 Comparación entre librerías de Visión Artificial.....	10
4.2.3 Algoritmos.....	11
4.2.3.1 Clasificadores tradicionales.....	11
4.2.3.2 Redes neuronales convolucionales.....	13
4.2.3.3 Comparativa entre algoritmos de detección.....	15
4.2.4 Aprendizaje profundo en la nube.....	17

4.2.4.1	Machine Learning - Engine GCP .....	17
4.2.4.2	IBM Power AI Vision .....	17
4.2.4.3	Comparativa entre servidores de aprendizaje profundo .....	17
4.3	Trabajos Relacionados.....	18
4.3.1	Aplicaciones de prototipado de interfaz de usuario .....	18
4.3.1.1	Marvel – Prototipos de apps .....	18
4.3.1.2	MockApp .....	18
4.3.1.3	Prott Rapid Prototyping Tool.....	18
4.3.1.4	Comparativa de aplicaciones .....	18
4.3.2	Proyectos de visión artificial en Android.....	19
4.3.2.1	Interpretación del lenguaje de señas utilizando redes neuronales .....	19
4.3.2.2	Detección de objetos y su implementación en dispositivos Android .....	19
5.	Materiales y Métodos.....	20
5.1	Contexto .....	20
5.2	Proceso .....	20
5.3	Recursos .....	22
5.3.1	Científicos .....	22
5.3.2	Técnicos.....	22
5.3.2.1	Servidores.....	22
5.3.2.2	Aplicaciones.....	23
5.3.2.3	Metodología .....	23
5.4	Participantes.....	23
6.	Resultados.....	24
6.1	Análisis de los componentes visuales para smartphone de la guía Material Design para Android .....	24
6.1.1	Identificación de los componentes visuales.....	24
6.1.2	Clasificación de los componentes visuales.....	29
6.2	Análisis de algoritmos de detección de objetos en imágenes. ....	30
6.2.1	Búsqueda de algoritmos de detección.....	30
6.2.2	Comparativa entre algoritmos de detección .....	30
6.2.2.1	Recolección de datos.....	31
6.2.2.2	Preparación de datos .....	31
6.2.2.3	Entrenamiento del modelo .....	33
6.2.2.4	Evaluación y comparación del modelo.....	34
6.2.3	Selección de algoritmo de detección .....	38

6.3	Diseño y desarrollo de la aplicación móvil .....	39
6.3.1	Exploración .....	39
6.3.1.1	Establecimiento de stakeholders .....	39
6.3.1.2	Definición del alcance .....	39
6.3.1.3	Alcance .....	39
6.3.1.4	Objetivos .....	39
6.3.1.5	Requerimientos funcionales.....	40
6.3.1.6	Requerimientos no funcionales .....	41
6.3.1.7	Módulos y procesos .....	41
6.3.1.8	Historias de usuario.....	42
6.3.1.9	Establecimiento del proyecto.....	42
6.3.2	Inicialización .....	43
6.3.2.1	Configuración de ambiente de desarrollo.....	43
6.3.2.2	Planificación inicial .....	43
6.3.2.3	Planificación de fases .....	44
6.3.2.4	Diseño del sistema .....	45
6.3.2.5	Diagramas de Clase .....	46
6.3.2.6	Interfaz gráfica de usuario.....	47
6.3.3	Producción.....	47
6.3.3.1	Interfaz gráfica de usuario y lógica de negocio .....	47
6.3.3.2	Comunicación con la nube .....	48
6.3.4	Estabilización y pruebas de aplicación .....	48
6.3.4.1	Pruebas de módulo .....	48
6.3.4.2	Pruebas de integración de módulo .....	48
6.4	Realización de escenario de pruebas .....	48
6.4.1	Prueba de funcionalidad y rendimiento de la aplicación móvil .....	49
6.4.2	Prueba de precisión del modelo Faster R-CNN.....	50
6.4.3	Prueba en escenario real.....	50
7.	Discusión.....	54
7.1	Desarrollo de la propuesta alternativa .....	54
7.1.1	Objetivo Específico 1: Análisis de los componentes visuales para smartphone de la guía Material Design para Android.....	54
7.1.2	Objetivo Específico 2: Análisis de algoritmos de detección de objetos en imágenes	54
7.1.3	Objetivo Específico 3: Diseño y desarrollo de la aplicación móvil .....	55

7.1.4	Objetivo Específico 4: Realización de escenario de pruebas .....	56
7.2	Valoración técnica económica ambiental .....	57
7.2.1	Recursos Humanos .....	57
7.2.2	Técnicos y Tecnológicos .....	57
7.2.3	Materiales y Servicios.....	58
7.2.4	Imprevistos.....	58
8.	Conclusiones .....	60
9.	Recomendaciones .....	62
10.	Bibliografía.....	63
11.	Anexos .....	67

## Índice de Figuras

Figura 1. Ejemplo de componentes de Material Design para Android .....	7
Figura 2. Matriz de píxeles de una imagen [12].....	8
Figura 3. Visualización de TensorBoard [16].....	10
Figura 4. Resultado de la máquina de vectores de soporte [19].....	12
Figura 5. Filtros de extracción Haar [21] .....	12
Figura 6. Detección de Cascada Haar [20] .....	13
Figura 7. Arquitectura SSD [23] .....	14
Figura 8. Arquitectura Faster R-CNN [23] .....	15
Figura 9. Comparativa de presión entre diferentes modelos [23] .....	16
Figura 10. Fotografía de figuras abstractas.....	31
Figura 11. Imagen con técnica Escala de grises .....	32
Figura 12. Imagen con técnica Binarización.....	32
Figura 13. Imagen con técnica Mediana .....	32
Figura 14. Etiquetado de clases en la imagen .....	33
Figura 15. Evaluación del modelo con 5910 iteraciones .....	34
Figura 16. Detección del modelo SSD .....	35
Figura 17. Detección del modelo Faster R-CNN .....	36
Figura 18. Detección del modelo Faster R-CNN con nuevo conjunto de datos .....	37
Figura 19. Modelo implementado en ML – Engine .....	38
Figura 20. Arquitectura del sistema.....	45
Figura 21. Diagrama de clase de aplicación .....	46
Figura 22. Diagrama de clase de detección .....	46
Figura 23. Diagrama de navegación de la aplicación móvil.....	47
Figura 24. Control de versiones en Play Store .....	49
Figura 25. Dispositivos de evaluación de aplicación móvil .....	50
Figura 26. Resultado de pregunta 1 de escenario real.....	51
Figura 27. Resultado de pregunta 2 de escenario real.....	52
Figura 28. Resultado de pregunta 3 de escenario real.....	52
Figura 29. Resultado de pregunta 4 de escenario real.....	53

## Índice de Tablas

TABLA I. COMPARATIVA ENTRE LIBRERÍAS DE VISIÓN ARTIFICIAL .....	11
TABLA II. COMPARATIVA ENTRE SERVIDORES DE APRENDIZAJE PROFUNDO	17
TABLA III. COMPARATIVA ENTRE APLICACIONES DE PROTOTIPADO DE INTERFAZ DE USUARIO .....	18
TABLA IV. COMPONENTES DESTACADOS EN WHATSAPP .....	24
TABLA V. COMPONENTES DESTACADOS EN MESSENGER .....	25
TABLA VI. COMPONENTES DESTACADOS EN FACEBOOK LITE .....	26
TABLA VII. COMPONENTES DESTACADOS EN INSTAGRAM .....	27
TABLA VIII. COMPONENTES DESTACADOS EN WISH .....	27
TABLA IX. CLASES DE INTERFAZ GRÁFICA A DETECTAR .....	28
TABLA X. CLASIFICACIÓN DE COMPONENTES .....	29
TABLA XI. COMPARATIVA DE ALGORITMOS DE DETECCIÓN .....	37
TABLA XII. REQUERIMIENTOS FUNCIONALES.....	40
TABLA XIII. REQUERIMIENTOS NO FUNCIONALES .....	41
TABLA XIV. MÓDULOS Y PROCESOS .....	41
TABLA XV. HISTORIA DE USUARIO: DETECCIÓN DE COMPONENTES.....	42
TABLA XVI. PRESUPUESTO EN TALENTO HUMANO .....	57
TABLA XVII. PRESUPUESTO EN RECURSOS TÉCNICOS Y TECNOLÓGICOS .....	57
TABLA XVIII. PRESUPUESTO EN RECURSOS DE OFICINA.....	58
TABLA XIX. PRESUPUESTO EN SERVICIOS.....	58
TABLA XX. PRESUPUESTO TOTAL DEL PROYECTO .....	59
TABLA XXI. GLOSARIO DE TÉRMINOS.....	67
TABLA XXII. HISTORIA DE USUARIO: CAPTURA DE FOTO .....	74
TABLA XXIII. HISTORIA DE USUARIO: DETECCIÓN DE COMPONENTES.....	74
TABLA XXIV. HISTORIA DE USUARIO: ELIMINACIÓN DE COMPONENTE.....	74
TABLA XXV. HISTORIA DE USUARIO: CAMBIO DE TAMAÑO DE COMPONENTE.	75
TABLA XXVI. HISTORIA DE USUARIO: CAMBIO DE UBICACIÓN DE COMPONENTE .....	75
TABLA XXVII. HISTORIA DE USUARIO: ACTUALIZACIÓN DE CONTENIDO DE COMPONENTE .....	75
TABLA XXVIII. HISTORIA DE USUARIO: COPIADO DE COMPONENTE .....	76
TABLA XXIX. HISTORIA DE USUARIO: GESTIÓN DE BARRAS DE HERRAMIENTAS .....	76
TABLA XXX. HISTORIA DE USUARIO: VISUALIZACIÓN DE PROTOTIPO .....	76
TABLA XXXI. HISTORIA DE USUARIO: COMPARTICIÓN DE CAPTURA DE PANTALLA .....	77
TABLA XXXII. HISTORIA DE USUARIO: GESTIÓN DE PROTOTIPOS.....	77
TABLA XXXIII. PLANIFICACIÓN DE FASES PARA EL DESARROLLO DE LA APLICACIÓN MÓVIL .....	78
TABLA XXXIV. PRUEBA DE MÓDULO: DETECCIÓN DE OBJETOS .....	81
TABLA XXXV. PRUEBA DE MÓDULO: GESTIÓN DE COMPONENTES VISUALES	82
TABLA XXXVI. PRUEBA DE MÓDULO: GESTIÓN DE PROTOTIPOS .....	83
TABLA XXXVII. PRUEBA DE MÓDULO: SECCIÓN DE AYUDA .....	84
TABLA XXXVIII. PRUEBA DE MÓDULO: INICIO DE SESIÓN ANÓNIMA.....	85
TABLA XXXIX. PRUEBA DE INTEGRACIÓN DE MÓDULOS: DETECCIÓN Y GESTIÓN DE COMPONENTES VISUALES.....	86

TABLA XL. PRUEBA DE INTEGRACIÓN DE MÓDULOS: GESTIÓN DE PROTOTIPO Y GESTIÓN DE COMPONENTES VISUALES .....	86
TABLA XLI. EVALUACIÓN DEL MODELO CON TENSORBOARD .....	88

# **1. Título**

**APLICACIÓN MÓVIL PARA EL PROTOTIPADO DE INTERFAZ  
GRÁFICA DE USUARIO EN LA PLATAFORMA ANDROID  
MEDIANTE EL RECONOCIMIENTO DE BOCETOS UTILIZANDO  
VISIÓN ARTIFICIAL**

## **2. Resumen**

Los desarrolladores novatos o iniciales de aplicaciones móviles, afrontan la creación de prototipos de interfaz gráfica de usuario en sus proyectos. Para ejecutar este proceso, se ven obligados a la búsqueda y capacitación de herramientas que están dirigidas a profesionales del Diseño Gráfico; desviando el tiempo destinado para el análisis y desarrollo de Software.

El objetivo de este estudio es determinar si la visión artificial facilita la creación de prototipos de interfaz gráfica de usuario, a través de una aplicación, la cual se compone de un servidor de aprendizaje profundo (Machine Learning - Engine), un servidor de base de datos y funciones en la nube (Firebase) y una aplicación móvil para teléfonos inteligentes con sistema operativo Android. Dicha aplicación permite capturar una foto a un boceto dibujado por el usuario; luego es enviada al servidor de aprendizaje profundo para detectar los posibles componentes que hayan sido dibujados. Según el resultado, es generado el prototipo de interfaz gráfica en la ubicación y clase detectada por la visión artificial, para luego ser gestionado con las opciones disponibles dentro de la aplicación móvil.

Para la visión artificial se utilizó un modelo pre-entrenado de red neuronal convolucional Faster R-CNN, entrenado y evaluado con la librería TensorFlow y desplegado en los servidores IBM Power AI Vision y Machine Learning - Engine. Para el entrenamiento se utilizó un conjunto de trazos a mano recolectados a 19 estudiantes, las cuales fueron tratados mediante técnicas de distorsión utilizando OpenCV, con el fin de aumentar los datos a un total de 862 imágenes, y de esta manera mejorar la calidad de detección del modelo.

Finalmente se desarrolló un escenario de prueba con 19 estudiantes de la Universidad Nacional de Loja de la Carrera de Ingeniería en Sistemas, en donde el 95% afirmó positivamente el funcionamiento y objetivo de la aplicación, validando el correcto funcionamiento del sistema y del presente trabajo de titulación; sin embargo, la aplicación puede ser mejorada aumentando la cantidad y calidad de imágenes en el conjunto de datos, y añadiendo nuevos componentes gráficos de la guía Material Design para Android; de esta forma se mejoraría la calidad de detección y diversidad de componentes por parte del modelo de visión artificial.

## **2.1 Abstract**

Amateur or beginner developers of mobile applications face the creation of prototypes of graphical user interface in their projects. In order to run this process, they are forced to search and get training about the tools that are aimed to Graphic Design professionals; distracting the time allocated for the analysis and development of Software.

The objective of this study is to determine whether artificial vision facilitates the creation of prototypes of graphical user interface, through an application, which is composed of a deep learning server (Machine Learning - Engine), a database server and functions on the cloud (Firebase) and a mobile app for smartphones with Android operating system. Such application allows to capture a photo of a sketch drawn by the user; then, it is sent to the deep learning server to detect the possible components that have been drawn. Depending on the result, it is generated the graphical interface prototype at the location and type detected by the artificial vision; then, it will be managed with the available options within the mobile app.

For the artificial vision it was used a pre-trained model of convolutional neural network Faster R-CNN, trained and evaluated with TensorFlow library and deployed on IBM Power AI Vision and Machine Learning - Engine servers. For the training, a set of hand strokes collected from 19 students was used, which were treated using distortion techniques using OpenCV, in order to increase the data to a total of 862 images, and thus improve the detection quality of the model.

Finally, a test scenario was developed with 19 students from the Universidad Nacional de Loja of the System Engineering Career, where 95% positively affirmed the operation and purpose of the application, validating the correct running of the system and of this degree work; however, the application can be improved by increasing the quantity and quality of images in the data set, and adding new graphic components to the Design Material guide for Android; this would improve the quality of detection and diversity of components by the artificial vision model.

### **3. Introducción**

En el diseño y construcción de aplicaciones, sin importar la metodología utilizada, existe un proceso que hace referencia a la creación de prototipos visuales en donde se forman conceptos generales de la posible interfaz de usuario que presentará la aplicación móvil. Dicha tarea es realizada generalmente por un diseñador gráfico o una persona conocedora de programas de diseño, sin embargo, en equipos de trabajo muy reducidos o desarrolladores iniciales e independientes deben acoger dicho proceso, adentrándose en la búsqueda de herramientas de diseño las cuales requieren capacitación, desviando tiempo y recursos utilizados en ingeniería de software, lo que puede conllevar a atrasos en su calendario de actividades.

El avance de la tecnología, ha permitido que la visión artificial se encuentra presente en muchos procesos de diferente índole, permitiendo la automatización y control de estos, además de facilitar su implementación en proyectos que basen sus procesos en el reconocimiento o detección de objetos.

Por ello, con el desarrollo del presente trabajo de titulación se desea identificar si es posible reducir el tiempo invertido en la creación de prototipos y brindar facilidad al usuario a través de un breve dibujo de su idea, mediante una interfaz (aplicación móvil Android) que permita capturar una fotografía al boceto dibujado, para ser procesada mediante visión artificial, detectando los posibles componentes trazados, y convirtiéndolos en componentes gráficos reales utilizando la guía oficial de diseño (Material Design para Android).

Para ello se planteó el objetivo principal “Desarrollar una aplicación móvil para el prototipado de interfaz gráfica de usuario en la plataforma Android mediante el reconocimiento de bocetos utilizando visión artificial”, y para cumplirlo se definieron 4 objetivos específicos, los cuales fueron: “Analizar los componentes visuales más utilizados para smartphone de la guía Material Design para Android”, “Analizar los algoritmos de detección de objetos en imágenes”, “Diseñar y desarrollar la aplicación móvil”, “Realizar un escenario de pruebas de software que permita evaluar el correcto funcionamiento de la aplicación”.

Por otra parte, la estructura del presente trabajo de titulación es la siguiente:

En la sección revisión de literatura, se encuentra información bibliográfica sobre las temáticas que abarcan el objeto de estudio, como lo es la interfaz gráfica de usuario, en

donde se describe la interacción humana – computadora y Material Design para Android. Además, se explica la visión artificial, sus herramientas y algoritmos destacados; y por último se abordan los trabajos relacionados referentes a aplicaciones para el prototipado gráfico de interfaz de usuario y la visión artificial aplicada a proyectos Android.

En la sección de materiales y métodos se describe el contexto en donde se desarrolló el proyecto de titulación y se detallan los procesos referentes al cumplimiento por cada objetivo. Además, se describen los recursos científicos, técnicos, materiales y participantes involucrados en el desarrollo.

En la sección de resultados se presenta la documentación generada durante el desarrollo de cada objetivo de manera detallada. Después se encuentra la sección discusión, en donde se analizan los resultados obtenidos frente a la información bibliográfica. Finalmente, en las secciones conclusiones y recomendaciones se encuentran el cierre, mediante un análisis sobre cada objetivo realizado, y posibles mejoras futuras.

## **4. Revisión de Literatura**

En esta sección se exponen los temas destacados y necesarios para abordar el objeto de estudio del presente trabajo de titulación, tales como: Interfaz Gráfica de Usuario en donde se describen subsecciones como la interacción entre el humano – computadora y componentes gráficos Material Design; Visión Artificial en donde se describen subsecciones como imagen digital, librerías, algoritmos y aprendizaje profundo en la nube; finalmente, Trabajos Relacionados en donde se describen proyectos similares que relacionan al objeto de estudio.

### **4.1 Interfaz gráfica de usuario**

La interfaz de usuario es todo aquello que el usuario puede ver y todo aquello con lo que este puede interactuar. Android ofrece un conjunto de componentes de interfaz de usuario previamente compilados, como objetos de diseño estructurados que permiten estructurar la interfaz gráfica de usuario de la aplicación [1] [2].

#### **4.1.1 Interacción Humano - Computadora**

La Interacción Humano-Computadora (HCI), es el estudio de la interacción entre el ser humano, las computadoras y los procesos que se desarrollan. Principalmente se enfoca a conocer cómo personas y computadoras pueden interactuar entre sí, para llevar a cabo tareas utilizando software como interfaz de comunicación [3].

##### **4.1.1.1 Interfaces orientadas a la usabilidad**

La usabilidad se basa en conceptos de facilidad, productividad y satisfacción del usuario respecto al sistema y va más allá de aquella idea inicial de interfaz gráfica amigable [4]. Actualmente el sistema operativo Android cuenta con un amplio arsenal de componentes gráficos y reglas de diseño para la elaboración de interfaces de usuario que abarcan gran usabilidad. Todos estos componentes fueron pensados y creados por Google, los cuales se centran en facilidad de uso, acceso rápido y productividad [5].

##### **4.1.1.2 Herramientas de diseño de interfaces**

Una herramienta de diseño de interfaces de usuario es una aplicación que ayuda a los diseñadores de la interfaz de usuario o a los programadores a su diseño y prueba de forma interactiva. Aproximadamente el 50 % del esfuerzo de programación de una aplicación tiene relación con la interfaz de usuario, debido a los requerimientos de

diseño y objetivos del proyecto, lo que justifica el esfuerzo dedicado a producir mejores herramientas para el diseño de interfaces [4].

#### 4.1.2 Material Design para Android

Material Design es una guía creada por Google, para el diseño visual y de interacción, que posee todos los componentes disponibles en la plataforma Android para sus versiones más actuales [6], mientras que los dispositivos iPhone de Apple, utilizan la guía UIKit [7], que es un estándar de diseño completamente diferente orientado a otro entorno. En la Figura 1, se puede observar un conjunto de componentes Material Design.

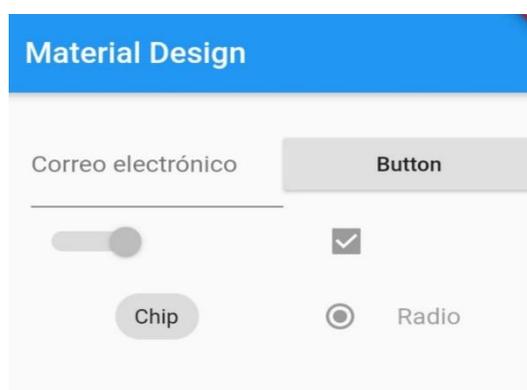


Figura 1. Ejemplo de componentes de Material Design para Android

A continuación, se describe un conjunto de componentes Material Design, los cuales son combinados para la construcción de interfaces en aplicaciones [5], estos son:

- **Raddio Button.** Permite seleccionar una opción de un conjunto. Use los botones de radio cuando el usuario necesite ver todas las opciones disponibles.
- **Floating Action Button.** Muestra la acción principal en una aplicación. Es un botón de icono redondo que se eleva por encima del contenido de otra página.
- **Switch.** Activan o desactivan el estado de un solo ajuste. Son la forma preferida de ajustar la configuración en el móvil.
- **Check Box.** Permite seleccionar uno o más elementos de un conjunto. Las casillas de verificación se pueden usar para activar o desactivar una opción.
- **Edit Text.** Permite ingresar caracteres a través del teclado.
- **Text View.** Permite presentar información a través de texto.
- **Recycler View.** Son índices continuos, verticales de texto o imágenes, es decir una lista de ítems.
- **Chip.** Permite presentar información o realizar selección de opciones.

- **Image View.** Permite visualizar contenido a través de una imagen.
- **Button.** Permite realizar, elegir o desencadenar acciones.
- **Tool Bar.** Muestra información y acciones relacionadas con la pantalla actual.
- **Tab Layout.** Proporciona un diseño horizontal para mostrar varias pestañas dentro de la misma pantalla.
- **Bottom Navigation.** Permite movimiento horizontal entre las pantallas principales.

## 4.2 Visión Artificial

La visión artificial es la capacidad que tienen las máquinas de procesar imágenes en busca de un patrón o estructura para ejecutar una acción determinada [8]. La visión artificial se realiza de manera semejante al proceso de visión humana, siendo su entrada una imagen obtenida mediante algún dispositivo, en donde la imagen es una matriz que representa la intensidad o brillo de la imagen [9] [10].

### 4.2.1 Imagen digital

Las imágenes digitales, generadas por ordenador o creadas por medio de un aparato de captura, suponen la traducción de valores de luminosidad y color a un lenguaje que pueda entender el ordenador [11] [12].

#### 4.2.1.1 Matrices de píxeles

En el mundo digital, las imágenes se representan como una matriz bidimensional de píxeles en la que cada píxel puede adquirir valores de color codificados con tres parámetros R: red (rojo), G: green (verde), B: blue (azul), o valores binarios 0 (negro) y 1 (blanco) como se observa en la Figura 2 [12].

1	1	1	1	1	1	1	1	1	1
1	0	0	0	1	1	0	0	0	1
1	1	0	1	1	1	1	0	1	1
1	1	0	1	1	1	1	0	1	1
1	1	0	1	1	1	1	0	1	1
1	1	0	0	0	0	0	0	1	1
1	1	0	1	1	1	1	0	1	1
1	1	0	1	1	1	1	0	1	1
1	1	0	1	1	1	1	0	1	1
1	0	0	0	1	1	0	0	0	1
1	1	1	1	1	1	1	1	1	1

Figura 2. Matriz de píxeles de una imagen [12]

Por lo general, se utilizan imágenes de baja resolución (entre 24 por 24 y 160 por 120 píxeles) como fuente de análisis en los procesos de visión artificial, ya que en algunos casos no es necesario excesivo detalle en las imágenes, todo dependerá del objetivo a cumplir [13].

#### **4.2.1.2 Bytes, bits y colores**

Un píxel se expresa mediante tres números enteros que representan los componentes rojo, verde y azul. Estos valores de representados por las letras R, G y B se suelen expresar en un rango de 8 bits, o sea, de valores entre 0 y 255 [13].

#### **4.2.1.3 Frecuencia de imagen**

La frecuencia de imagen (frame rate) hace referencia al número de imágenes por segundo que un reproductor de imagen o video es capaz de presentar, por lo general se mide en FPS (frames per second) [13].

#### **4.2.1.4 Herramientas**

La visión artificial necesita de un dispositivo sensible a la luz que permita almacenar las imágenes en formato digital, como: cámara web, cámara de vídeo o cámara analógica [13].

### **4.2.2 Librerías de visión artificial**

En esta sección se describen dos librerías de gran importancia OpenCV y TensorFlow, gracias a su gran documentación y facilidad de implementación, además se encuentran respaldadas por grandes comunidades.

#### **4.2.2.1 OpenCV**

OpenCV u Open Computer Vision es un conjunto de herramientas de código abierto disponibles en varios lenguajes de programación como C++, C, Python, Java. Además, es compatible con distintas plataformas como Windows, Linux, MacOS, Android. Permite llevar a cabo el proceso de visión artificial, desde la adquisición de la imagen hasta la extracción de la información, también cuenta con filtros o herramientas para procesar imágenes [13].

#### **4.2.2.2 TensorFlow**

TensorFlow es una biblioteca de software de código abierto para el cálculo numérico utilizando gráficos de flujo de datos disponible para distintos lenguajes de programación

como C++, Python y Java. TensorFlow fue desarrollado originalmente por investigadores e ingenieros de Google dentro de la organización de investigación de Inteligencia Artificial, para realizar investigaciones de aprendizaje automático y redes neuronales profundas [14]. Es compatible con una gran variedad de aplicaciones, con un enfoque en entrenamiento e inferencia en redes neuronales profundas. Varios servicios de Google usan TensorFlow en producción, además permite a los desarrolladores experimentar con nuevas optimizaciones y algoritmos de entrenamiento [15].

Los cálculos que utilizará TensorFlow para, como entrenar una red neuronal profunda masiva, pueden ser complejos y confusos. Para facilitar la comprensión, la depuración y la optimización de los programas TensorFlow, se ha incluido un conjunto de herramientas de visualización llamado TensorBoard, que permite visualizar el gráfico de entrenamiento, trazar métricas cuantitativas sobre su ejecución y presentar datos adicionales como imágenes que interfieren en sus procesos (ver Figura 3) [16]:

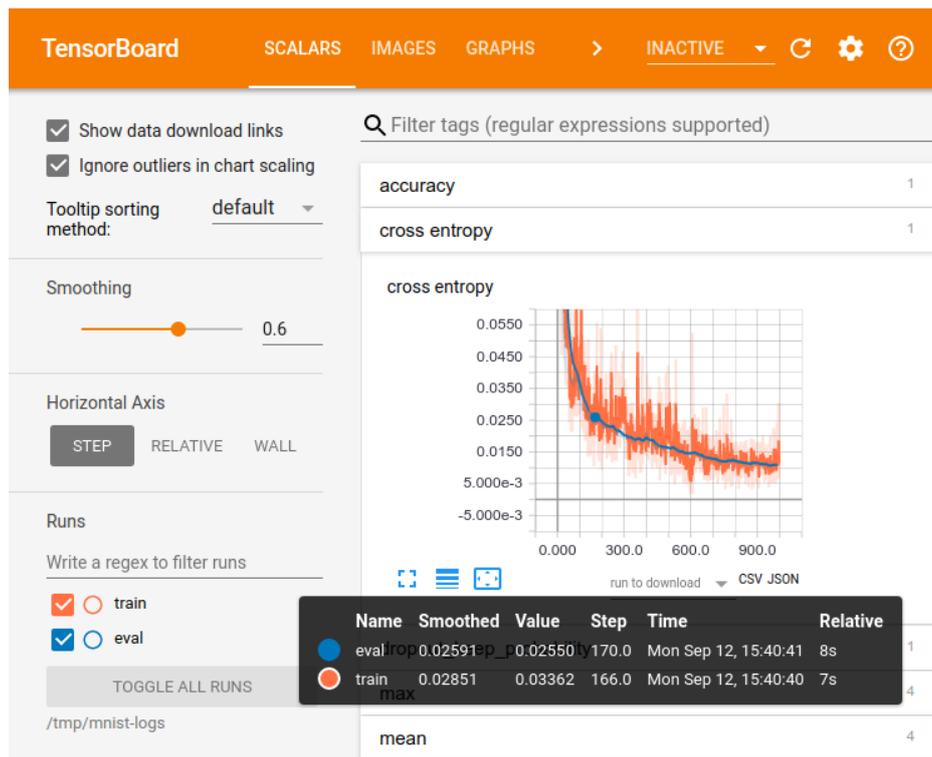


Figura 3. Visualización de TensorBoard [16]

#### 4.2.2.3 Comparación entre librerías de Visión Artificial

OpenCV y TensorFlow apuntan a diferentes tipos de algoritmos de visión artificial y poseen características únicas, la TABLA I presenta una comparativa de estas herramientas [17] [18].

TABLA I. COMPARATIVA ENTRE LIBRERÍAS DE VISIÓN ARTIFICIAL

PARÁMETRO	LIBRERÍA	
	OpenCV	TensorFlow
Tipo de algoritmos	Visión por computadora tradicional	Redes neuronales profundas
Manipulación de imágenes (filtros, escalas, etc.)	Si	No
Disponible en Android	Si	Si
Entrenamiento	Si	Si
Evaluación	Si	Si
Despliegue (servidor)	No	Si

#### 4.2.3 Algoritmos

En esta sección se describen diferentes algoritmos de visión artificial, de los cuales se dividen en clasificadores tradicionales y algoritmos de redes neuronales.

##### 4.2.3.1 Clasificadores tradicionales

- **Máquina de vectores de soporte (SVM)**

La máquina de vectores de soporte pertenece a la categoría de los clasificadores lineales, puesto que inducen separadores lineales o hiperplanos, sin embargo, se puede utilizar para la clasificación no lineal. La idea es seleccionar un hiperplano de separación que equidista de los ejemplos más cercanos de cada clase para, de esta forma, conseguir lo que se denomina un margen máximo a cada lado del hiperplano. En la Figura 4, se observa que para definir el hiperplano, solo se consideran los ejemplos de entrenamiento de cada clase que caen justo en la frontera de dichos márgenes. Estos ejemplos reciben el nombre de vectores soporte [19].

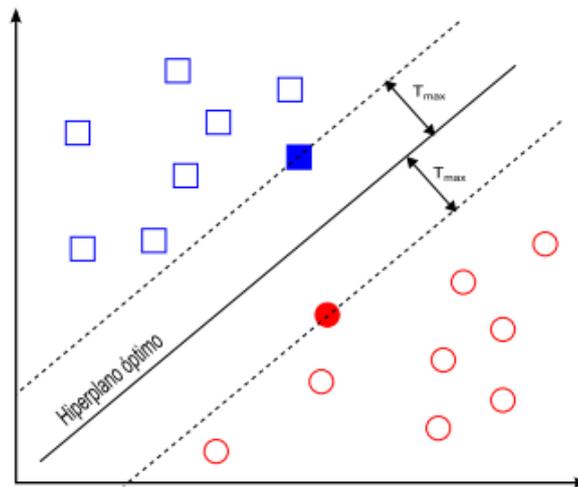


Figura 4. Resultado de la máquina de vectores de soporte [19]

La máquina de vectores de soporte se puede emplear para los siguientes objetivos [19]:

- Clasificación binaria de ejemplos separables linealmente.
- Clasificación binaria de ejemplos cuasiseparables linealmente (presencia de ruido).
- Clasificación binaria de ejemplos no separables linealmente.
- Regresión.

• **Clasificador en cascada Haar**

Haar es un extractor de características que actúa sobre regiones rectangulares de una imagen en escala de grises [20]. Una característica está formada por un número finito de rectángulos o filtros. Estos filtros pueden ser calculados sobre la imagen, actúan en orientación espacial y frecuencia [21]. En la Figura 5, se presentan los filtros usados para la extracción de características.

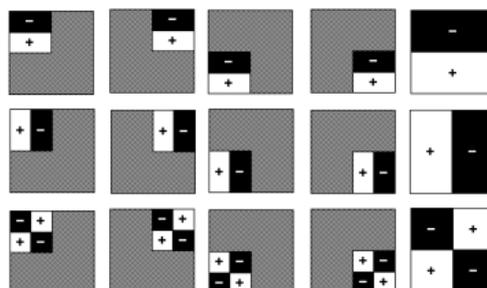


Figura 5. Filtros de extracción Haar [21]

El clasificador en cascada, consiste en una serie de clasificadores débiles que conforman un clasificador fuerte. Como se observa en la Figura 6, una subventana clasificará con la primera etapa, si el objeto a detectar supera esta etapa, seguirá con las restantes. En cambio, si el objeto a detectar no supera alguna de las etapas, se descarta todo el proceso [20].

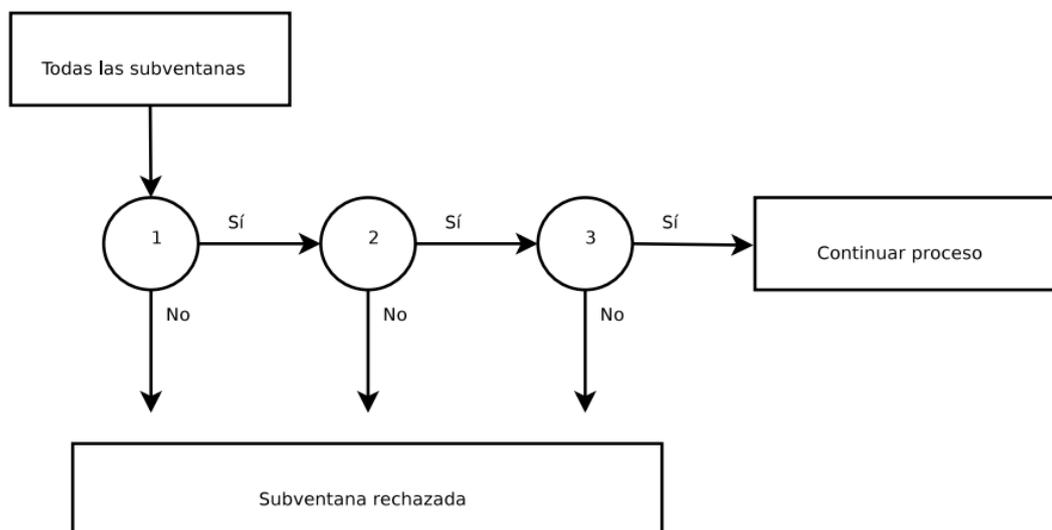


Figura 6. Detección de Cascada Haar [20]

Por lo general este tipo de clasificadores se usan específicamente en la detección del rostro humano, pero se puede entrenar para inferir otros objetos de acuerdo con los datos de entrenamiento.

#### 4.2.3.2 Redes neuronales convolucionales

- **Single Shot Detector (SSD)**

El enfoque SSD se basa en una red neuronal convolucional hacia adelante que produce una colección de tamaño fijo de recuadros delimitadores y puntajes para la presencia de objetos en esos cuadros, seguido de un paso de supresión no máxima para producir las detecciones finales. Las primeras capas de red se basan en una arquitectura estándar utilizada para la clasificación de imagen de alta calidad, que llamará a la siguiente red. Luego se agrega una estructura auxiliar a la red para producir detecciones con las siguientes características [22] (ver Figura 7):

- **Mapas de características de escala múltiple para detección.** Se agregan capas de características convolucionales hasta el final de la red. Estas capas disminuyen

de tamaño progresivamente y permiten predicciones de detecciones en escalas múltiples [22].

- **Predictores convolucionales para la detección.** Cada capa de características agregada puede producir un conjunto fijo de predicciones de detección utilizando un conjunto de filtros convolucionales [22].
- **Cajas predeterminadas y relaciones de aspecto.** Se asocian un conjunto de cajas delimitadoras por defecto con cada celda de mapa de características, para múltiples mapas de características. En cada celda de mapa de características, se predicen las compensaciones relativas a las formas de la caja predeterminada en la celda, así como los puntajes por clase que indican la presencia de una instancia de clase en cada uno de esos cuadros [22].

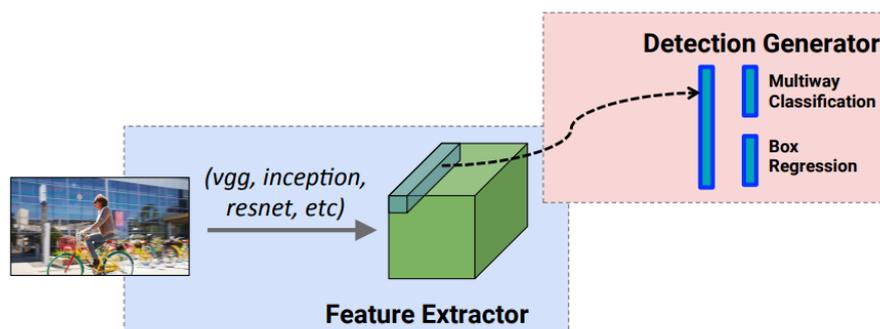


Figura 7. Arquitectura SSD [23]

### ● **Faster R-CNN**

El modelo Faster R-CNN es una mejora del modelo Fast R-CNN. Este modelo se basa en propuestas de región, posee dos redes: la red de propuesta de región (RPN) para generar propuestas de región y una red que usa estas propuestas para detectar objetos. La principal diferencia aquí con Fast R-CNN es que el último usa la búsqueda selectiva para generar propuestas de región. El costo de tiempo de generar propuestas de región es mucho menor en RPN que la búsqueda selectiva, cuando RPN comparte la mayor parte de la computación con la red de detección de objetos. Brevemente, RPN clasifica los recuadros de región y propone los que con mayor probabilidad contienen objetos. El modelo posee las siguientes características [24] (ver Figura 8):

- **Mapas de características de escala múltiple para detección.** Se agregan capas de características convolucionales hasta el final de la red. Estas capas disminuyen

de tamaño progresivamente y permiten predicciones de detecciones en escalas múltiples [24] [25].

- **Redes de propuesta de región.** Una Red de propuesta de región toma una imagen de cualquier tamaño como entrada y emite un conjunto de propuestas de objetos rectangulares, cada una con un puntaje. El objetivo de esta red es compartir computación con una red de detección [24].
- **Transición de anclas.** En cada ubicación de ventana deslizante, se predicen simultáneamente las propuestas de región. Las k propuestas están parametrizadas en relación con k cuadros de referencia, llamados anclajes. Cada ancla se centra en la ventana deslizante en cuestión, y es asociado con una escala y relación de aspecto. Se hace uso de tres escalas y tres proporciones de aspecto, produciendo nueve anclas en cada posición deslizante [24].

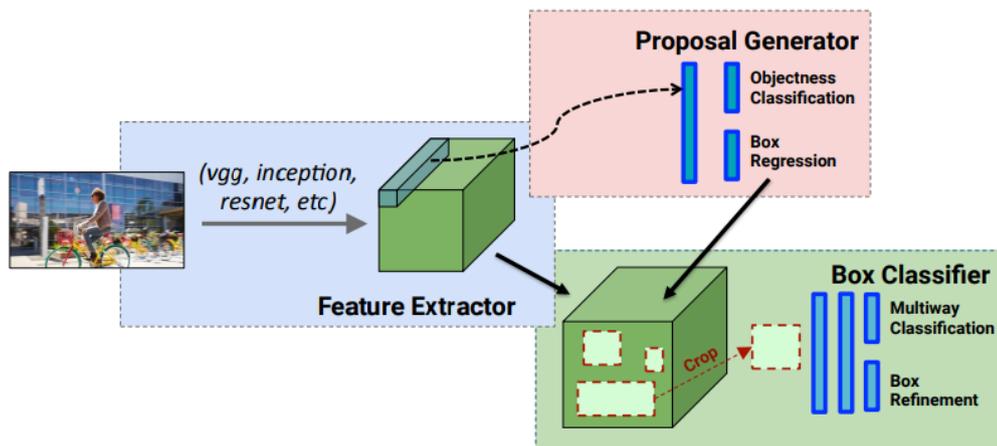


Figura 8. Arquitectura Faster R-CNN [23]

#### 4.2.3.3 Comparativa entre algoritmos de detección

Como se observa en la Figura 9, los modelos con arquitectura o extractor de características Resnet 101 e Inception Resnet v2, poseen mayor precisión (mAP) en la detección de objetos de mediana y pequeña escala [23].

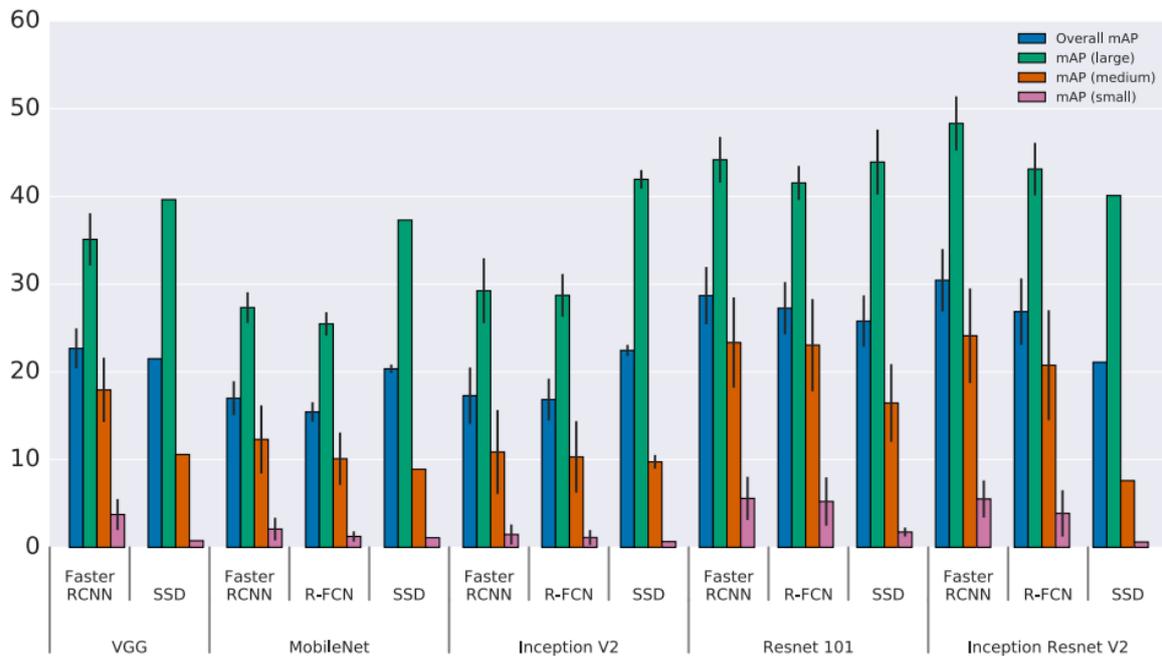


Figura 9. Comparativa de presión entre diferentes modelos [23]

- Procediendo a la comparación entre el clasificador en cascada Haar y una red neuronal convolucional, según lo concluido en [26], el método basado en redes neuronales convolucionales posee mejores resultados para encontrar rostros en ubicaciones u orientaciones complejas [26].
- Para la comparación entre SVM y una red neuronal convolucional, según lo concluido en [27], la CNN permiten una disminución en la tasa de clasificación errónea que en SVM. Además, el tiempo de entrenamiento se reduce en gran medida cuando se usa aprendizaje profundo basado en GPU en comparación con algoritmos tradicionales que utilizan CPU, esto posee un gran impacto cuando el dataset tiende a escalar.

Por lo tanto, en el presente proyecto se buscará una solución adecuada que utilice modelos de redes neuronales convolucionales como SSD o Faster R-CNN, con el fin de obtener mejores resultados en los procesos de detección de objetos, debido a que los componentes serán dibujados a mano alzada por diferentes usuarios, es decir, el dibujo tendrá determinadas variaciones de forma y orientación, por lo que se requiere la mayor precisión posible.

#### 4.2.4 Aprendizaje profundo en la nube

En esta sección se describen los servicios de aprendizaje profundo en la nube, los cuales permiten desplegar y monitorear modelos de visión artificial para su consumo.

##### 4.2.4.1 Machine Learning - Engine GCP

Machine Learning o ML - Engine es un servicio brindado por Google, que permite crear modelos superiores de aprendizaje automático y llevarlos a producción mediante herramientas de preparación y predicción que pueden ser usados juntos o por separado [28].

##### 4.2.4.2 IBM Power AI Vision

Power AI Vision es una herramienta que proporciona un sólido soporte de flujo de trabajo para modelos de aprendizaje profundo relacionados con la visión artificial. Este software de nivel empresarial proporciona un ecosistema completo para capacitación, creación y despliegue de modelos basados en aprendizaje profundo [29].

##### 4.2.4.3 Comparativa entre servidores de aprendizaje profundo

En la TABLA II se realiza una comparativa entre las características más importantes de estos servicios [29] [28].

TABLA II. COMPARATIVA ENTRE SERVIDORES DE APRENDIZAJE PROFUNDO

PARÁMETRO	SERVIDOR	
	Machine Learning - Engine	IBM Power AI Vision
Compatibilidad de librerías	TensorFlow	TensorFlow, PyTorch
Tiempo de prueba de herramientas	365 días (crédito de \$300)	3-7 días
Entrenamiento	Si (GPU)	Si (GPU)
Evaluación	Si (GPU)	Si (GPU)
Despliegue	Si (Cuenta de prueba CPU, Cuenta de paga acceso a GPU)	Si (Cuenta de prueba CPU, Cuenta de paga acceso a GPU)
Consumo mediante	Https, SDK	Https

### 4.3 Trabajos Relacionados

En este apartado se describen algunos trabajos relacionados al objeto de estudio; tales como: aplicaciones de prototipado de interfaz gráfica de usuario, y la aplicación de visión artificial utilizando redes neuronales en proyectos Android.

#### 4.3.1 Aplicaciones de prototipado de interfaz de usuario

##### 4.3.1.1 Marvel – Prototipos de apps

Marvel es una herramienta diseñada para prototipar aplicaciones. Se trata de una aplicación web. Esto significa que para poder usar el programa tendremos que tener una conexión a internet. Marvel no crea prototipos en sí, sino que sube a la plataforma las pantallas capturadas y se diseñan con otro programa de prototipado [30].

##### 4.3.1.2 MockApp

MockApp es una aplicación que permite prototipar interfaz gráfica de usuario en la plataforma Android, mediante la selección de componentes preestablecidos, posee una gran cantidad de opciones para su diseño.

##### 4.3.1.3 Prott Rapid Prototyping Tool

Prott es una aplicación que permite capturar con la cámara los prototipos dibujados en papel, para luego asignar eventos a las imágenes mediante la selección de áreas, permitiendo la sensación de botones e interacción [31].

##### 4.3.1.4 Comparativa de aplicaciones

En la TABLA III se realiza una comparativa entre las características más importantes de estas aplicaciones.

TABLA III. COMPARATIVA ENTRE APLICACIONES DE PROTOTIPADO DE INTERFAZ DE USUARIO

	APLICACIÓN		
PARÁMETRO	Prott Rapid Prototyping Tool	Mockapp	Marvel – Prototipos De Apps
Componentes	Canvas	Material Design	Material Design, canvas
Dependencia de plataforma web	No	No	Si

<b>Complejidad</b>	Baja (prototipos simple)	Alta (gran cantidad de opciones para editar)	Media (el prototipo se edita en la web)
<b>Método de diseño</b>	Captura de imagen, arrastrar y soltar	Edición nativa, arrastrar y soltar	Captura de imagen, arrastrar y soltar
<b>Requiere internet</b>	No	No	Si

#### **4.3.2 Proyectos de visión artificial en Android**

##### **4.3.2.1 Interpretación del lenguaje de señas utilizando redes neuronales**

Este proyecto consiste en el entrenamiento de una red neuronal convolucional en una aplicación Android, para poder interpretar el lenguaje de señas. La red neuronal convolucional utiliza una arquitectura extractora de características pre entrenada llamada Inception v3. En dicho proyecto se implementa la red sobre TensorFlow Lite, es decir, no necesita conexión a internet, sino que infiere desde el hardware del dispositivo móvil, proceso que reduce la carga de la batería. Para el entrenamiento y evaluación del modelo, se emplea el uso de TensorFlow sobre una GPU para aumentar la velocidad de entrenamiento [32].

##### **4.3.2.2 Detección de objetos y su implementación en dispositivos Android**

Este proyecto consiste en la implementación de una red neuronal convolucional en una aplicación Android, para detectar objetos del dataset KITTI. Se utiliza una red neuronal convolucional pre entrenada llamada SqueezeDet, que es muy económica computacionalmente y en almacenamiento, por ende, muy eficiente al momento de inferir. Utiliza TensorFlow Lite e infiere desde el hardware del dispositivo [33].

## **5. Materiales y Métodos**

En esta sección se describe el contexto en donde se desarrolló el presente trabajo de titulación, además se detallan los procesos y recursos que fueron necesarios para su correcta finalización. Por último, se describen los participantes que intervinieron en el desarrollo del mismo.

### **5.1 Contexto**

El proyecto se desarrolló y evaluó en la Universidad Nacional de Loja, en la Carrera de Ingeniería en Sistemas. Mientras que el entorno virtual del sistema, se desplegó y ejecutó en los servidores de aprendizaje profundo IBM Vision Power AI y Machine Learning - Engine (ver sección 6.2.2.3, 6.2.2.4, 6.2.3) y servidor de base de datos y funciones Firebase (ver sección 6.3.2.1).

### **5.2 Proceso**

Para cumplir el objetivo general (desarrollar una aplicación móvil para el prototipado de interfaz gráfica de usuario en la plataforma Android mediante el reconocimiento de bocetos utilizando visión artificial), se ejecutaron los procesos que se describen a continuación.

#### **Fase 1. Analizar los componentes visuales más utilizados para Smartphone de la guía Material Design para Android.**

1. Se buscó las aplicaciones más populares o mejor calificadas en Play Store (ver sección 6.1.1).
2. Se seleccionó los componentes de Material Design a utilizar (ver sección 6.1.1 TABLA IX), de acuerdo a la constante aparición de estos en las aplicaciones analizadas anteriormente (ver sección 6.1.1 TABLA IV TABLA V TABLA VI TABLA VII TABLA VIII).
3. Se estableció identificadores a cada componente seleccionado (ver sección 6.1.2 TABLA X) para ser reconocidos por el usuario y visión artificial.

#### **Fase 2. Analizar los algoritmos de detección de objetos en imágenes.**

1. Se buscó herramientas y algoritmos de visión artificial en fuentes bibliográficas confiables (ver sección 6.2.1).
2. Se seleccionó las herramientas a utilizar en base a comparaciones en fuentes bibliográficas confiables (ver sección 6.2.1).

3. Se recolectó y refinó muestras de dibujo de varios estudiantes para crear un conjunto de datos (ver sección 6.2.2.1, 6.2.2.2).
4. Se compararon los modelos Single Shot Detection y Faster R-CNN mediante su entrenamiento y evaluación (ver sección 6.2.2.3, 6.2.2.4 TABLA XI).
5. Se seleccionó el modelo con mejores resultados y fue desplegado en el servidor ML – Engine (ver sección 6.2.3).

### **Fase 3. Diseñar y desarrollar la aplicación móvil.**

1. Se estableció los involucrados en la aplicación (ver sección 6.3.1.1).
2. Se estableció el alcance de la aplicación (ver sección 6.3.1.3).
3. Se estableció los requerimientos funcionales y no funcionales (ver sección 6.3.1.5, 6.3.1.6).
4. Se estableció los módulos, procesos e historias de usuario (ver sección 6.3.1.7, 6.3.1.8).
5. Se configuró el ambiente de desarrollo físico y virtual (ver sección 6.3.2.1).
6. Se planificó el desarrollo por fases (ver sección 6.3.2.3) acorde a los módulos y procesos obtenidos.
7. Se realizó el diseño del sistema junto a sus artefactos (ver sección 6.3.2.4, 6.3.2.5).
8. Se diseñó la interfaz gráfica de usuario y su navegación (ver sección 6.3.2.6).
9. Se desarrolló las funcionalidades (ver sección 6.3.3) acorde a los módulos y procesos obtenidos.
10. Se ejecutaron pruebas de módulo y funcionalidades de la aplicación móvil (ver sección 6.3.4.1).
11. Se ejecutaron pruebas de integración entre los módulos desarrollados (ver sección 6.3.4.2).

### **Fase 4. Realizar un escenario de pruebas de software que permita evaluar el correcto funcionamiento de la aplicación.**

1. Se seleccionó las pruebas utilizadas para evaluar el sistema (ver sección 6.4).
2. Se ejecutó las pruebas de funcionalidad y rendimiento a la aplicación móvil (ver sección 6.4.1).
3. Se ejecutó la prueba de precisión del modelo de detección de objetos (ver sección 6.4.2).
4. Se ejecutó una prueba del sistema en un escenario real en donde intervino un conjunto de usuarios finales (ver sección 6.4.3).

## 5.3 Recursos

Para lograr el completo desarrollo del trabajo de titulación se emplearon los recursos descritos a continuación.

### 5.3.1 Científicos

- **Bibliográfica.** Permitió obtener información sobre las temáticas relacionadas con el objeto de estudio en las siguientes fases [34] [35]:
  - En la Fase 1, permitió reconocer los componentes de la guía Material Design para Android (ver sección 5.2 Fase 1).
  - En la Fase 2, permitió la búsqueda y comparación de herramientas y algoritmos de visión artificial (ver sección 5.2 Fase 2).
  - En la Fase 3, ayudó proporcionando información sobre la arquitectura Mobile-D (ver sección 5.2 Fase 3).
  - En la Fase 4, ayudó proporcionando información sobre las pruebas para evaluar el sistema (ver sección 5.2 Fase 4).
- **Encuesta.** Permitió recolectar información sobre el funcionamiento del sistema en un escenario real, lo que permitió concluir la Fase 4 (ver sección 5.2 Fase 4). Además, permitió recolectar imágenes de componentes abstractos para la elaboración del conjunto de datos que sirvió de entrenamiento para el modelo de visión artificial (ver sección 5.2 Fase 1.3 y Fase 2.4) [36].
- **Muestreo no probabilístico por conveniencia.** Permitió seleccionar una muestra de estudiantes (ver sección 5.4), los cuales evaluaron el sistema (ver sección 5.2 Fase 4 y Anexo 13, estudiantes de 6to “A” de la carrera de Ingeniería en Sistemas de la Universidad Nacional de Loja) [37].
- **Observación directa.** Permitió seleccionar los componentes de interfaz gráfica de usuario que se utilizaron en la aplicación móvil (ver sección 5.2 Fase 1) [38] [39].

### 5.3.2 Técnicos

#### 5.3.2.1 Servidores

- **IBM Power AI Vision.** Servidor de visión artificial para el entrenamiento de modelos de redes neuronales (ver sección 5.2 Fase 2.4).
- **Machine Learning – Engine.** Servidor como servicio de visión artificial para el entrenamiento, evaluación y despliegue a producción de modelos de redes neuronales (ver sección 5.2 Fase 2.5).

- **Firestore.** Servidor como servicio para la gestión de información en la nube de la aplicación móvil (ver sección 5.2 Fase 3.5).

#### 5.3.2.2 Aplicaciones

- **Android Studio.** Entorno de desarrollo integrado, empleado para programar la aplicación móvil (ver sección 5.2 Fase 3.5).
- **Firestore SDK.** Conjunto de herramientas de desarrollo que permitieron implementar las funcionalidades de Firestore en la aplicación móvil (ver sección 5.2 Fase 3.5).
- **Google Play Console.** Herramienta que permitió evaluar la funcionalidad y rendimiento de la aplicación móvil (ver sección 5.2 Fase 4.1).
- **TensorFlow.** Librería de visión artificial que permitió entrenar y evaluar modelos de redes neuronales (ver sección 5.2 Fase 2.4).
- **OpenCV.** Librería de visión artificial empleada en el uso de técnicas de distorsión a imágenes (ver sección 5.2 Fase 2.3).
- **LabelImg.** Herramienta de etiquetado de objetos en imágenes, utilizada para seleccionar los objetos a detectar en el conjunto de datos (ver sección 5.2 Fase 2.3).

#### 5.3.2.3 Metodología

- **Metodología Mobile-D.** Metodología ágil utilizada para el diseño y desarrollo de la aplicación móvil (ver sección 5.2 Fase 3).

### 5.4 Participantes

El presente trabajo de titulación fue desarrollado por José Luis Aguilar Espinoza, con la guía de los docentes tutores Ing. Manuel Alberto Córdova Neira (Fase 1 y 2) e Ing. Edison Leonardo Coronel Romero (Fase 3 y 4). Además, participaron en la evaluación del sistema 19 alumnos de 6to "A" de la carrera de Ingeniería en Sistemas de la Universidad Nacional de Loja, que aprobaron recientemente la materia de Programación Avanzada (ver sección 5.2 Fase 2.3 y 5.2 Fase 4.4).

## 6. Resultados

### 6.1 Análisis de los componentes visuales para smartphone de la guía Material Design para Android

La ejecución de este objetivo comprendió el reconocimiento, selección y delimitación de componentes de interfaz gráfica de usuario de la guía Material Design para Android [5] que fueron utilizados en el trabajo de titulación. Para cumplirlo se desarrollaron las siguientes actividades.

#### 6.1.1 Identificación de los componentes visuales

En este apartado se utiliza la técnica de observación directa (ver sección 5.3.1) para seleccionar los componentes Material Design presentados en las aplicaciones más populares y mejor calificadas de Play Store (ver Anexo 2, Top 10 de aplicaciones mejor calificadas y más populares en la tienda oficial de Android Google Play), con el fin de delimitar la cantidad de componentes que serán detectados mediante visión artificial. Estos componentes fueron escogidos de acuerdo a la continua o repetitiva aparición en las diferentes pantallas de interfaz de usuario. Las aplicaciones analizadas junto a sus componentes destacados o más repetidos fueron los siguientes (descripciones recogidas de la tienda oficial Play Store):

- **WhatsApp Messenger**

WhatsApp Messenger es una aplicación de mensajería gratis, disponible para Android y otros teléfonos inteligentes. WhatsApp puede enviar y recibir mensajes, fotos, videos, documentos, mensajes de voz, y para realizar y recibir llamadas (ver TABLA IV).

TABLA IV. COMPONENTES DESTACADOS EN WHATSAPP

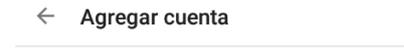
CLASE	REPRESENTACIÓN
Tool Bar	
Switch	Activa esta opción para recibir notificaciones cuando el código de 
Check Box	Reproducir sonidos para los mensajes entrantes y salientes. 
Tab Layout	

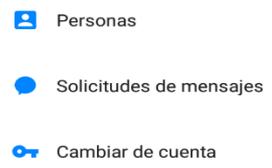
<b>Image View</b>	
<b>Text View</b>	Llegando a sauces
<b>Recycler View</b>	<div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 5px;"> <p><b>Hugo</b> 7:39 PM ✓/Oka</p> </div> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 5px;"> <p><b>Pichulo Recargado</b> 7:04 PM Andres: De ley 🔊</p> </div> <div style="border: 1px solid #ccc; padding: 5px;"> <p><b>+593 98 618 6227</b> 3:16 PM Ok</p> </div>
<b>Floating Action Button</b>	
<b>Edit Text</b>	<input type="text" value="Escribir mensaje"/>

- **Messenger Lite**

Messenger Lite es una versión ligera de la aplicación Messenger, esta permite enviar contenido a todos los contactos de tu agenda de teléfonos o amigos de Facebook e introducir un número de teléfono para añadir un contacto nuevo (ver TABLA V).

TABLA V. COMPONENTES DESTACADOS EN MESSENGER

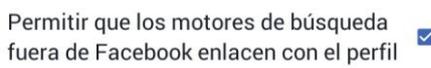
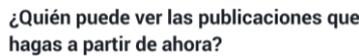
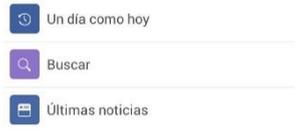
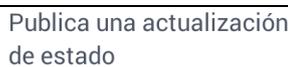
CLASE	REPRESENTACIÓN
<b>Tool Bar</b>	
<b>Switch</b>	
<b>Check Box</b>	<input checked="" type="checkbox"/> Solicitar contraseña al cambiar a esta cuenta desde este dispositivo
<b>Tab Layout</b>	
<b>Image View</b>	
<b>Text View</b>	Inicia sesión en tu cuenta de Facebook. Podrás cambiar de cuenta para ver los mensajes nuevos.

<b>Recycler View</b>	
<b>Floating Action Button</b>	
<b>Edit Text</b>	

- **Facebook Lite**

Facebook Lite es la versión rápida y ligera de la popular plataforma Facebook, compatible para la mayoría de teléfonos inteligentes (ver TABLA VI).

TABLA VI. COMPONENTES DESTACADOS EN FACEBOOK LITE

CLASE	REPRESENTACIÓN
<b>Tool Bar</b>	
<b>Switch</b>	
<b>Radio Button</b>	
<b>Check Box</b>	
<b>Tab Layout</b>	
<b>Image View</b>	
<b>Text View</b>	
<b>Recycler View</b>	
<b>Floating Action Button</b>	
<b>Edit Text</b>	

- **Instagram**

Instagram es una aplicación que permite capturar y compartir imágenes o videos con otros usuarios de su plataforma (ver TABLA VII).

TABLA VII. COMPONENTES DESTACADOS EN INSTAGRAM

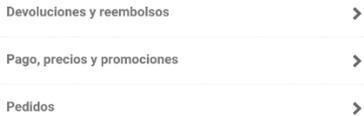
CLASE	REPRESENTACIÓN
Tool Bar	
Switch	Cuenta privada
Check Box	Tus seguidores
Tab Layout	
Bottom Navigation	
Image View	
Text View	seguidores seguidos
Recycler View	
Button	
Edit Text	liftedstorm

- **Wish**

Wish es una aplicación que permite comprar y vender gran variedad de productos en diferentes partes del mundo (ver TABLA VIII).

TABLA VIII. COMPONENTES DESTACADOS EN WISH

CLASE	REPRESENTACIÓN
Tool Bar	
Switch	Facebook
Check Box	Tus pedidos Avísame del estado de mis pedidos

<b>Tab Layout</b>	
<b>Image View</b>	
<b>Text View</b>	
<b>List View</b>	
<b>Button</b>	
<b>Edit Text</b>	

Luego del análisis de las diferentes aplicaciones, se define el modo de disponibilidad de cada componente descrito anteriormente, por el cual será llamado o ejecutado dentro de la aplicación. Como se muestra en la TABLA IX (para conocer la descripción de estos componentes ver la sección 4.1.2), consta de una columna Clases, en donde se encuentran componentes según la guía Material Design para Android, y la columna Disponibilidad, en donde se establece el método de llamado a dicho componente dentro de la aplicación, los cuales pueden ser:

- **Mediante dibujo por el usuario.** Este método presentará los componentes que se encontraron repetidamente en cada pantalla dentro de las aplicaciones analizadas.
- **Mediante opción dentro de la aplicación.** Este método presentará los componentes que se encontraron una vez por pantalla, ya que son barras de herramientas limitadas a una por cada pantalla.

TABLA IX. CLASES DE INTERFAZ GRÁFICA A DETECTAR

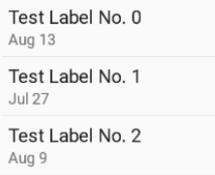
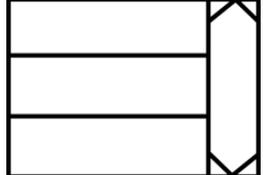
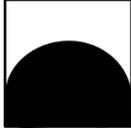
CLASES	DISPONIBILIDAD
Check Box	<ul style="list-style-type: none"> <li>- Encontrados varias veces por pantalla.</li> <li>- Ejecución o llamada mediante dibujo por el usuario.</li> </ul>
Radio Button	
Switch	
Button	
Floating Action Button	
Chip	
Recycler View	
Text View	
Edit Text	

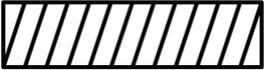
<b>Image View</b>	
<b>Tool Bar</b>	
<b>Bottom Navigation</b>	- Encontrados una sola vez por pantalla.
<b>Tab Layout</b>	- Ejecución mediante opción dentro de la aplicación.

### 6.1.2 Clasificación de los componentes visuales

Una vez seleccionados los componentes visuales, se procede a la asignación gráfica única en base a una aproximación abstracta de su representación real, tomando en cuenta las características gráficas representativas de cada clase, además se encierran sobre un contenedor para contemplar el alto y ancho del componente. Esto se realiza a cada clase que será identificada por el usuario y la visión artificial, como se muestra en la TABLA X:

TABLA X. CLASIFICACIÓN DE COMPONENTES

CLASE	REPRESENTACIÓN REAL	REPRESENTACIÓN ABSTRACTA
<b>Radio Button</b>	 Radio	
<b>Chip</b>	 Chip	
<b>Check Box</b>	 CheckBox	
<b>Edit Text</b>	 Name	
<b>Recycler View</b>		
<b>Switch</b>	 Switch	
<b>Image View</b>		
<b>Floating Action Button</b>		

Button		
Text View		

## 6.2 Análisis de algoritmos de detección de objetos en imágenes.

La ejecución de este objetivo comprendió la búsqueda, comparación y selección de algoritmos y herramientas para la aplicación de visión artificial, basadas en fuentes bibliográficas confiables citadas en la sección 4.2. Para cumplirlo se desarrollaron las siguientes actividades.

### 6.2.1 Búsqueda de algoritmos de detección

Inicialmente se realizó la búsqueda de herramientas para la aplicación de visión artificial que fuesen compatibles con la plataforma Android. Los resultados fueron las librerías OpenCV y TensorFlow (ver sección 4.2.2). Partiendo de dicho software se analizaron sus algoritmos más destacados; por parte de OpenCV se encontraron los clasificadores tradicionales como clasificador en cascada HAAR o Máquina de vectores de soporte SVM (ver sección 4.2.3.1). Mientras que TensorFlow se especializa en redes neuronales convolucionales como Single Shot Detector o Faster R-CNN (ver sección 4.2.3.2).

Sin embargo, para el presente trabajo de titulación se determinó el uso de TensorFlow gracias a la comparativa realizada (ver sección 4.2.2.3), ya que junto a ML – Engine permiten el entrenamiento, evaluación y despliegue de un modelo de red neuronal (ver sección 4.2.4), el cual posee un mejor rendimiento que los algoritmos tradicionales según la comparativa obtenida en la sección 4.2.3.3.

### 6.2.2 Comparativa entre algoritmos de detección

La comparativa entre algoritmos o modelos de redes neuronales permite determinar cuál obtiene mejores resultados de inferencia, para ello es necesario entrenar y evaluar un modelo; en este caso se utilizaron dos modelos pre-entrenados Single Shot Detector y Faster R-CNN (ver sección 4.2.3.3) que varían en cuanto a velocidad y calidad de inferencia.

El objetivo del modelo es determinar a qué clase pertenece cada figura dibujada en base a una fotografía como dato de entrada, tomando como referencia las figuras abstractas obtenidas en la sección 6.1.2. Por ende, es necesario entrenar y evaluar un modelo de

red neuronal con imágenes propias de acuerdo a las figuras abstractas, lo que conlleva a un conjunto de pasos detallados a continuación [17].

### 6.2.2.1 Recolección de datos

Este paso consiste en recolectar un conjunto de imágenes o fotografías representativas en donde se encuentren los objetos a detectar por la visión artificial, en este caso las figuras abstractas pertenecientes a cada clase como se muestra en la Figura 10. En total se recolectaron 862 imágenes, 86 para cada clase.

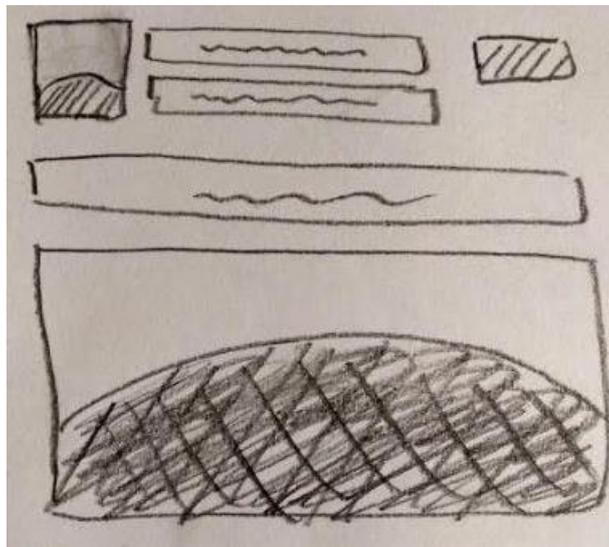


Figura 10. Fotografía de figuras abstractas

### 6.2.2.2 Preparación de datos

Una vez recolectadas las fotografías, es necesario refinarlas para que el proceso de extracción de datos sea más sencillo y eficiente; entre mejor sean los datos representativos del objeto a detectar, mejor será la inferencia del modelo. Para lograr un refinamiento adecuado se usan las siguientes técnicas [13]:

- **Escala de grises.** Cuando el color de la imagen no aporta ninguna información relevante para interactuar se acostumbra a descartar la información de color de la imagen transformándola en una imagen de tonos de grises como se presenta en la Figura 11.

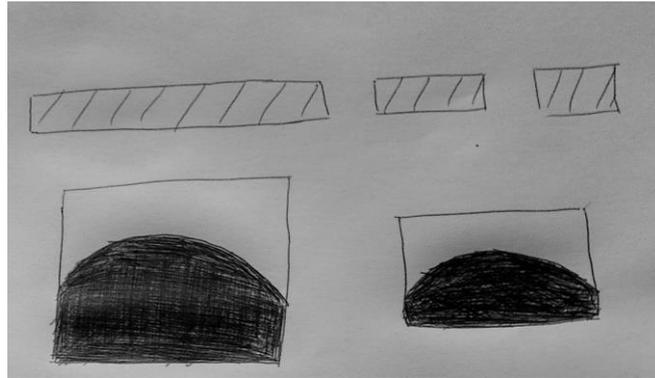


Figura 11. Imagen con técnica Escala de grises

- **Binarización por umbral adaptativo.** El proceso de binarización por umbral parte de una imagen en tonos de grises y, a partir de un valor definible de umbral de intensidad de luz llamado threshold, la imagen se transforma en una imagen binaria, es decir, con píxeles blancos o píxeles negros como se presenta en la Figura 12.

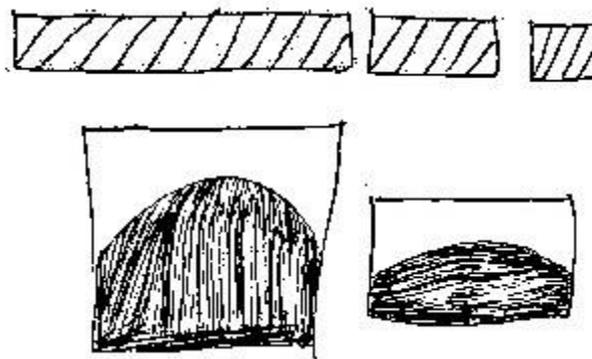


Figura 12. Imagen con técnica Binarización

- **Mediana.** Esta técnica se aplica sobre las imágenes en escala de grises y provoca una cierta suavización de la imagen útil a la hora de simplificar contornos y áreas irregulares como se presenta en la Figura 13.

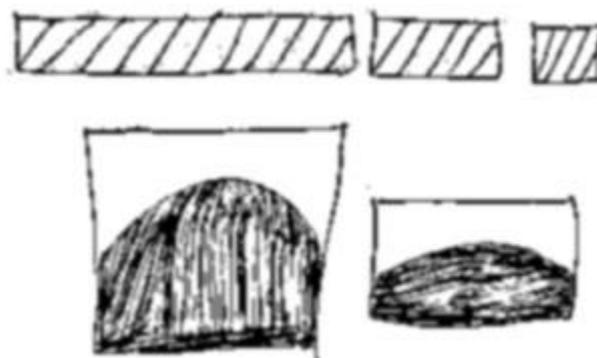


Figura 13. Imagen con técnica Mediana

Para aplicar estas técnicas a las fotografías recolectadas se utilizó la librería OpenCV, bajo el lenguaje de programación Python; en el Anexo 4 se presenta el código de programación desarrollado.

Una vez aplicadas las técnicas anteriormente descritas, se procede a enmarcar o establecer cuadros de delimitación con la clase respectiva (ver sección 6.1.2) a que pertenece cada objeto de interés dentro de las imágenes como se muestra en la Figura 14. Este proceso se realizó mediante la aplicación Labelling que es un software de código abierto para etiquetar imágenes [40].

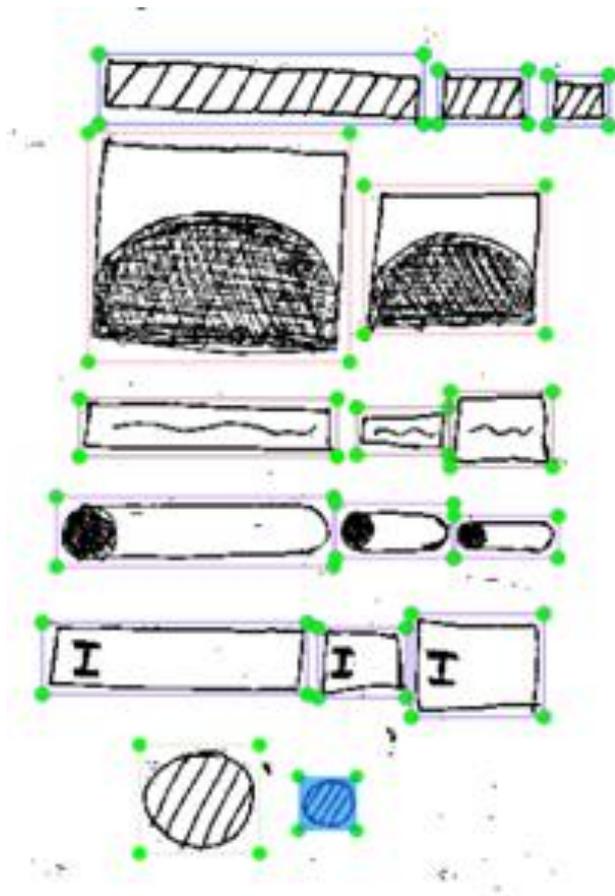


Figura 14. Etiquetado de clases en la imagen

### 6.2.2.3 Entrenamiento del modelo

Para que TensorFlow pueda utilizar el conjunto de datos obtenido en los anteriores pasos (ver sección 6.2.2.1, 6.2.2.2), fue necesario transformarlo a un formato llamado TFRecord (TensorFlow Record) en donde se agrupan las etiquetas e imágenes para ser leídos y entrenados [17]; en el Anexo 5 se adjunta el código de programación desarrollado para esta conversión.

Una vez iniciado el entrenamiento, TensorFlow enviará los datos a la red neuronal (en este caso Single Shot Detector y Faster R-CNN) para su aprendizaje. Ambos modelos se entrenaron a la cantidad de 500K iteraciones, es decir, los datos entraron dicha cantidad a la red neuronal para su configuración. La ejecución del entrenamiento se realizó en los servidores de IBM Power AI Vision.

#### 6.2.2.4 Evaluación y comparación del modelo

La evaluación del modelo consiste en obtener la calidad de inferencia en determinada iteración, como se puede observar en la Figura 15, TensorBoard (ver sección 4.2.2.2) encierra el objeto detectado y presenta el nombre de la clase junto a la precisión del modelo para dicha clase.

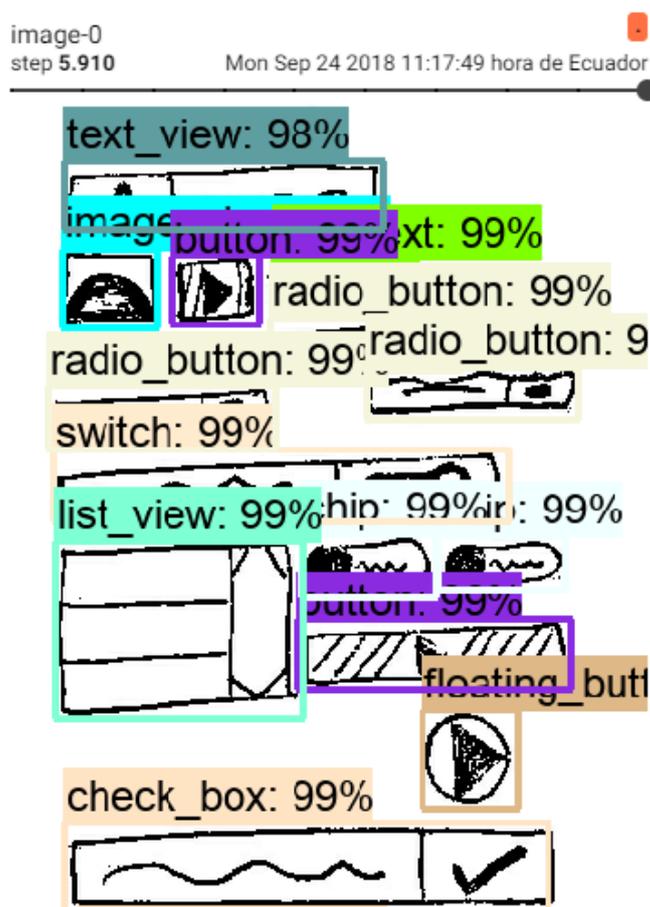


Figura 15. Evaluación del modelo con 5910 iteraciones

Además, se realizaron comparaciones de cada modelo mediante el dibujo a mano sobre una hoja en blanco; los resultados fueron los siguientes:

- **Single Shot Detector.** Como se puede observar en la Figura 16, este modelo no detecta objetos pequeños o de baja resolución (16 píxeles o menos), pero funciona bien con los objetos de resolución mediana y grande. El tiempo de inferencia fue de 10 a 15 segundos.

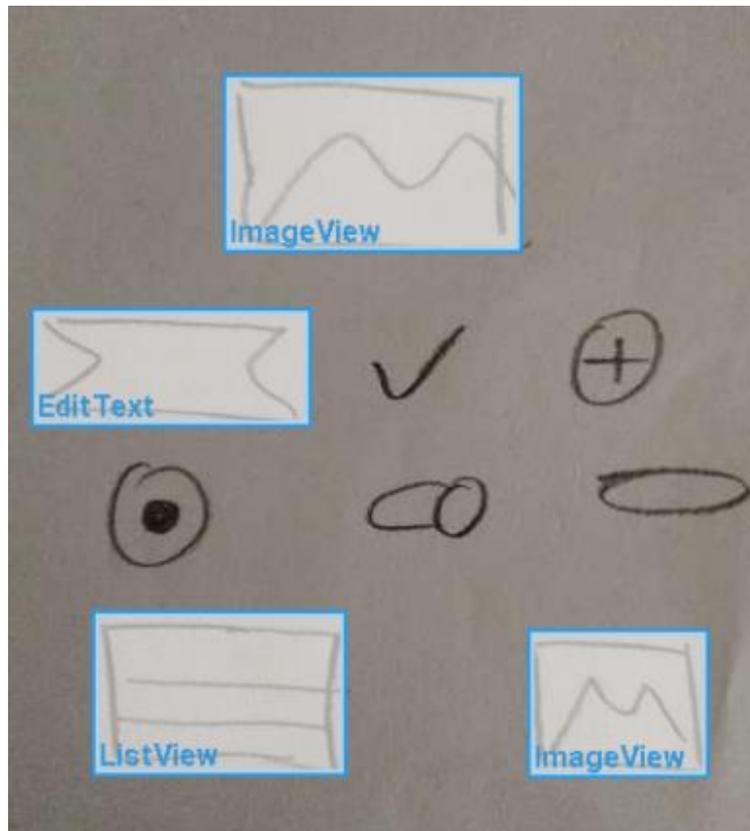


Figura 16. Detección del modelo SSD

- **Faster R-CNN.** Como se puede observar en la Figura 17, el modelo detecta objetos de resolución pequeña, mediana y grande. Sin embargo, el tiempo de inferencia fue de 40 a 50 segundos.

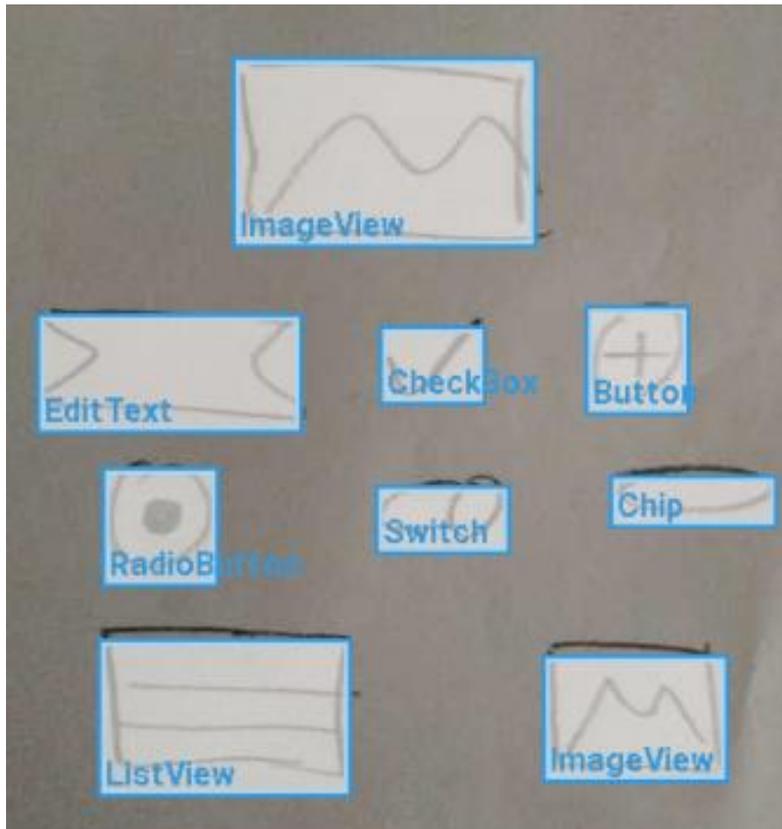


Figura 17. Detección del modelo Faster R-CNN

Por objeto de experimentación, se creó un nuevo conjunto de datos (versión 2) con características que representan mejor a cada componente y se evaluó a 170K iteraciones. Como se puede observar en la Figura 18, el modelo Faster R-CNN con el nuevo conjunto de datos, infiere adecuadamente a un número bajo de iteraciones, además, este modelo se ejecuta sobre un procesador de cuatro núcleos, por lo tanto, el tiempo de inferencia se reduce a 15 segundos.

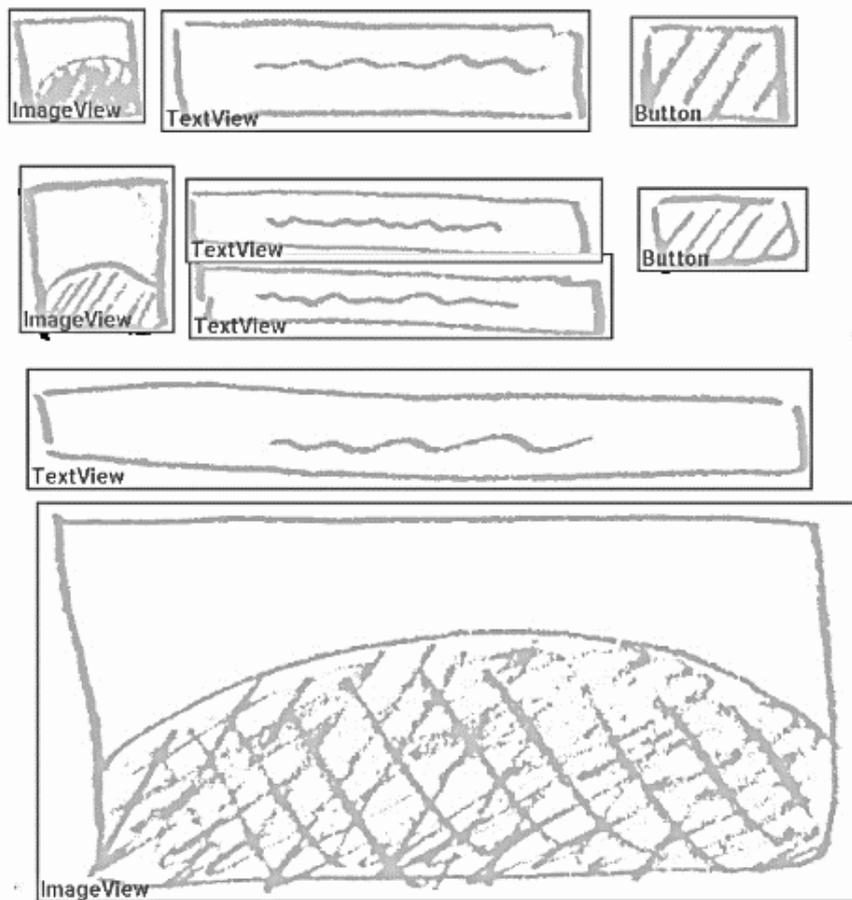


Figura 18. Detección del modelo Faster R-CNN con nuevo conjunto de datos

La TABLA XI muestra la comparativa entre los modelos antes mencionados, en donde el modelo Faster R-CNN junto al nuevo conjunto de datos, obtiene mejores resultados.

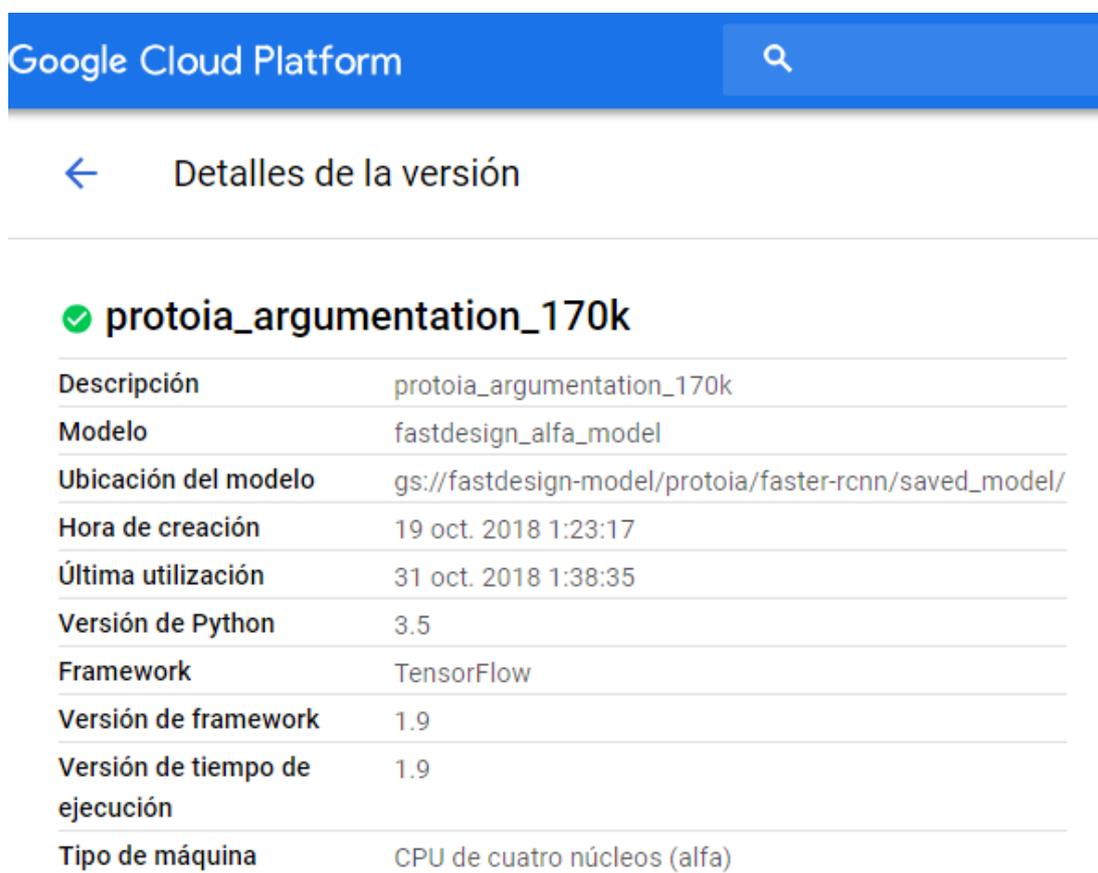
TABLA XI. COMPARATIVA DE ALGORITMOS DE DETECCIÓN

PARÁMETROS	ALGORITMOS		
	SSD	Faster R-CNN	
Versión de dataset	1	1	2
Inferencia en objetos pequeños	Bajo	Alto	Alto
Inferencia en objetos medianos	Medio	Alto	Alto
Inferencia en objetos grandes	Alto	Alto	Alto
Núcleos de CPU	1	1	4
Tiempo de inferencia	6-10 seg.	40-50 seg.	13-15 seg.
Falsos - positivos	Medio	Medio	Bajo
Positivos - positivos	Alto	Alto	Alto

### 6.2.3 Selección de algoritmo de detección

El dibujo de componentes puede presentarse a diferentes escalas, por lo cual la aplicación deberá ser capaz de detectar objetos a diferentes tamaños. Por lo tanto, se determina que la mejor técnica para la detección de objetos en el presente trabajo de titulación es el modelo Faster R-CNN mediante el servidor Google Cloud Platform – ML Engine (ver sección 6.2.2.4 TABLA XI).

El servidor ML – Engine es de paga, sin embargo, Google brinda un crédito de \$300 durante un año para poder evaluar su plataforma con algunas restricciones. Como se observa en la Figura 19, el modelo entrenado Faster R-CNN, ya se encuentra desplegado en ML – Engine y listo para ser consumido.



The screenshot shows the Google Cloud Platform interface. At the top, there is a blue header with the text 'Google Cloud Platform' and a search icon. Below the header, there is a navigation bar with a back arrow and the text 'Detalles de la versión'. The main content area displays the details for a model named 'protoia\_argumentation\_170k', which is marked with a green checkmark. The details are presented in a table format with the following rows:

Descripción	protoia_argumentation_170k
Modelo	fastdesign_alfa_model
Ubicación del modelo	gs://fastdesign-model/protoia/faster-rcnn/saved_model/
Hora de creación	19 oct. 2018 1:23:17
Última utilización	31 oct. 2018 1:38:35
Versión de Python	3.5
Framework	TensorFlow
Versión de framework	1.9
Versión de tiempo de ejecución	1.9
Tipo de máquina	CPU de cuatro núcleos (alfa)

Figura 19. Modelo implementado en ML – Engine

## **6.3 Diseño y desarrollo de la aplicación móvil**

La ejecución de este objetivo comprendió el diseño y desarrollo de la aplicación móvil utilizando como guía la metodología ágil denominada Mobile-D [41], la cual consta de las siguientes etapas: Exploración, Inicialización, Producción, Estabilización y Pruebas. A continuación, se detalla el desarrollo de cada una de ellas [42].

### **6.3.1 Exploración**

En esta etapa se planifican y extraen los requerimientos del proyecto con el fin de obtener una visión completa de su alcance junto a todas sus funcionalidades. Esta etapa se divide en diversas actividades que se detallan a continuación.

#### **6.3.1.1 Establecimiento de stakeholders**

En esta actividad se definieron los involucrados, así como sus roles, actividades y responsabilidades que se describen a continuación:

- **Desarrollador:** Persona encargada de realizar el análisis, desarrollo, pruebas de la aplicación.
- **Usuarios:** Personas quienes usarán la aplicación. Este rol lo puede desempeñar el desarrollador o el diseñador de interfaz gráfica de usuario.

#### **6.3.1.2 Definición del alcance**

En esta actividad se definió el alcance, objetivos y requerimientos iniciales del proyecto de acuerdo al tiempo destinado

#### **6.3.1.3 Alcance**

Aplicación móvil que permita diseñar prototipos básicos de interfaz gráfica de usuario con los componentes de Material Design para Android definidos anteriormente (ver sección 6.1.2), que se ejecutará en teléfonos inteligentes con dicho sistema operativo, utilizando el reconocimiento de bocetos a través de Google Cloud Platform - ML Engine.

#### **6.3.1.4 Objetivos**

- Tomar foto al boceto dibujado por el usuario.
- Detectar los componentes dibujados en Google Cloud Platform.

- Diseñar el prototipo con los componentes previamente definidos de Material Design para Android.
- Permitir cambios de edición, tales como: tamaño, ubicación, contenido, hasta donde el componente gráfico lo permita.
- Ver, guardar, editar y eliminar el prototipo.

### 6.3.1.5 Requerimientos funcionales

En la TABLA XII se describen los requerimientos funcionales.

TABLA XII. REQUERIMIENTOS FUNCIONALES

CÓDIGO	DESCRIPCIÓN
RF-001	El usuario podrá capturar una foto al boceto
RF-002	El usuario podrá enviar una foto para detectar el prototipo
RF-003	El usuario podrá ver el resultado de detección
RF-004	El usuario podrá eliminar un componente del prototipo
RF-005	El usuario podrá actualizar el tamaño de un componente del prototipo
RF-006	El usuario podrá actualizar la ubicación de un componente del prototipo
RF-007	El usuario podrá actualizar el texto de un componente del prototipo que extienda de Text View
RF-008	El usuario podrá actualizar la imagen de un componente del prototipo que extienda de Image View
RF-009	El usuario podrá copiar un componente del prototipo
RF-010	El usuario podrá añadir barras de herramientas y navegación al prototipo (Tool bar, Botton Navigation, Tab layout)
RF-011	El usuario podrá quitar barras de herramientas y navegación al prototipo (Tool bar, Botton Navigation, Tab layout)
RF-012	El usuario podrá visualizar solo el prototipo (sin las opciones de prototipado)
RF-013	El usuario podrá compartir externamente una captura de pantalla del prototipo
RF-014	El usuario podrá guardar el prototipo
RF-015	El usuario podrá descartar el prototipo
RF-016	El usuario podrá visualizar una lista de sus prototipos
RF-017	El usuario podrá eliminar un prototipo de la lista
RF-018	El usuario podrá editar un prototipo
RF-019	El usuario podrá recortar la foto en la zona donde se encuentre el prototipo

### 6.3.1.6 Requerimientos no funcionales

En la TABLA XIII se describen los requerimientos no funcionales.

TABLA XIII. REQUERIMIENTOS NO FUNCIONALES

CÓDIGO	DESCRIPCIÓN
RNF-001	La aplicación requiere smartphones con las siguientes especificaciones: Sistema operativo oficial Android 5.0 o superior, 2 o más Gb de RAM, cámara de 2 o más megapíxeles, conexión estable a internet.
RNF-002	La aplicación dependerá del servidor para la detección de objetos.
RNF-003	La aplicación requiere un mínimo de 10 MB aproximadamente de almacenamiento interno del Smartphone.
RNF-004	La aplicación permitirá la detección de un boceto por petición al servidor.
RNF-005	La aplicación iniciara sesión automáticamente (usuario anónimo).
RNF-006	La aplicación contará con sección de ayuda.

### 6.3.1.7 Módulos y procesos

En la TABLA XIV se organizan los módulos y procesos con sus respectivos requerimientos.

TABLA XIV. MÓDULOS Y PROCESOS

MÓDULO	CÓDIGO	PROCESO	REQUERIMIENTOS
Detección de Objetos	M-001	Detectar componentes del boceto	RF-001, RF-002, RF-003, RF-019
Gestión de Componentes Visuales	M-002	Modificar los componentes detectados	RF-004, RF-005, RF-006, RF-007, RF-008, RF-009, RF-010, RF-011, RF-012, RF-013
Gestión de Prototipos	M-003	Gestionar prototipos	RF-014, RF-015, RF-016, RF-017, RF-018
Sección de ayuda	M-004	Ver la sección de ayuda	RNF-006
Inicio de sesión anónima	M-005	Inicio automático de la aplicación como usuario anónimo	RNF-005

### 6.3.1.8 Historias de usuario

La TABLA XV detalla la historia de usuario más destacada de la aplicación. En el Anexo 6 se encuentran en su totalidad.

TABLA XV. HISTORIA DE USUARIO: DETECCIÓN DE COMPONENTES

HISTORIA DE USUARIO: DETECCIÓN DE COMPONENTES		
<b>Identificador:</b> HU-002	<b>Rol:</b> Usuario	
<b>Entrada de datos:</b> Fotografía	<b>Salida de datos:</b> Detección	
<b>Descripción:</b> Necesito enviar una foto del prototipo con la finalidad de detectar los componentes gráficos		
<b>Criterio de aceptación:</b> <ul style="list-style-type: none"><li>• <b>Éxito</b><ul style="list-style-type: none"><li>○ Envío de fotografía al servidor y visualización del resultado de detección</li></ul></li><li>• <b>Fracaso</b><ul style="list-style-type: none"><li>○ Mensaje advirtiendo al usuario sobre el problema</li></ul></li></ul>		
<b>Dificultad:</b> Alta	<b>Esfuerzo:</b> Alto	<b>Prioridad:</b> Alta
<b>Observaciones:</b> El usuario puede o no estar dentro de la aplicación durante el proceso de detección		

### 6.3.1.9 Establecimiento del proyecto

En esta actividad se definieron los recursos físicos y técnicos necesarios para el desarrollo del proyecto, los cuales son:

- IDE Android Studio
- Lenguaje de programación Kotlin
- Arquitectura de desarrollo MVP
- Arquitectura de desarrollo Clean
- Google Cloud Platform - ML Engine (detección de objetos)
- Firebase (versión gratuita)
- Smartphone con procesador ARM doble núcleo a 1 GHZ o superior, 1.5GB RAM o superior, cámara 1MP o superior, Android 5.0 o superior
- Computador CPU 4 núcleos y 8GB de RAM (para el desarrollo)
- Bitbucket (gestión de versiones)

### 6.3.2 Inicialización

En esta etapa se planifican las actividades de desarrollo junto a sus artefactos, además, se contemplan los recursos físicos o virtuales que se necesitarán para la correcta ejecución del proyecto. Esta etapa se divide en diversas actividades que se detallan a continuación.

#### 6.3.2.1 Configuración de ambiente de desarrollo

En esta actividad se realiza la configuración de herramientas necesarias para desarrollar el proyecto, en este caso, la preparación de las siguientes herramientas:

- **Firebase.** Configuración inicial del proyecto para acceder a los servicios de Firebase, además descargar el SDK Firebase para establecer comunicación con el cliente junto a un archivo de configuración del proyecto [43].
- **Google Cloud Platform - ML Engine.** El modelo de detección de objetos Faster R-CNN ya fue implementado en el anterior objetivo (ver sección 2.3).
- **IDE Android Studio.** La creación de la aplicación móvil, así como la configuración del lenguaje Kotlin, el repositorio Bitbucket, arquitecturas de desarrollo y todas las librerías necesarias para el desarrollo de la aplicación se realizan en Android Studio, además dispone de una interfaz amigable y avanzada de monitoreo de software y hardware ideal para evaluar la aplicación en un dispositivo real o simulado.

Para obtener mayor información sobre configuración y estructura de herramientas y servicios empleados para el desarrollo de la aplicación, revisar el manual de programador adjunto en el Anexo 15. Así también se adjunta el manual de usuario en el Anexo 16, donde se detallan las funcionalidades de la aplicación móvil junto a su interfaz de usuario.

#### 6.3.2.2 Planificación inicial

- **P-001: Detectar componentes del boceto**
  - El usuario podrá capturar una foto al boceto.
  - El usuario podrá enviar una foto para detectar el prototipo.
  - El usuario podrá ver el resultado de detección.

- **P-002: Modificar los componentes detectados**
  - El usuario podrá eliminar un componente del prototipo.
  - El usuario podrá actualizar el tamaño de un componente del prototipo.
  - El usuario podrá actualizar la ubicación de un componente del prototipo.
  - El usuario podrá actualizar el texto de un componente del prototipo que extienda de Text View.
  - El usuario podrá actualizar la imagen de un componente del prototipo que extienda de Image View.
  - El usuario podrá copiar un componente del prototipo.
  - El usuario podrá añadir barras de herramientas y navegación al prototipo (Tool bar, Botton Navigation, Tab layout).
  - El usuario podrá quitar barras de herramientas y navegación al prototipo (Tool bar, Botton Navigation, Tab layout).
  - El usuario podrá visualizar solo el prototipo (sin las opciones de prototipado).
  - El usuario podrá compartir externamente una captura de pantalla del prototipo.
  
- **P-003: Gestionar prototipos**
  - El usuario podrá guardar el prototipo.
  - El usuario podrá descartar el prototipo.
  - El usuario podrá visualizar una lista de sus prototipos.
  - El usuario podrá eliminar un prototipo de la lista.
  - El usuario podrá editar un prototipo.
  
- **P-004: Ver la sección de ayuda**
  - La aplicación contará con sección de ayuda.
  
- **P-005: Inicio automático de la aplicación como usuario anónimo**
  - La aplicación iniciara sesión automáticamente (usuario anónimo).

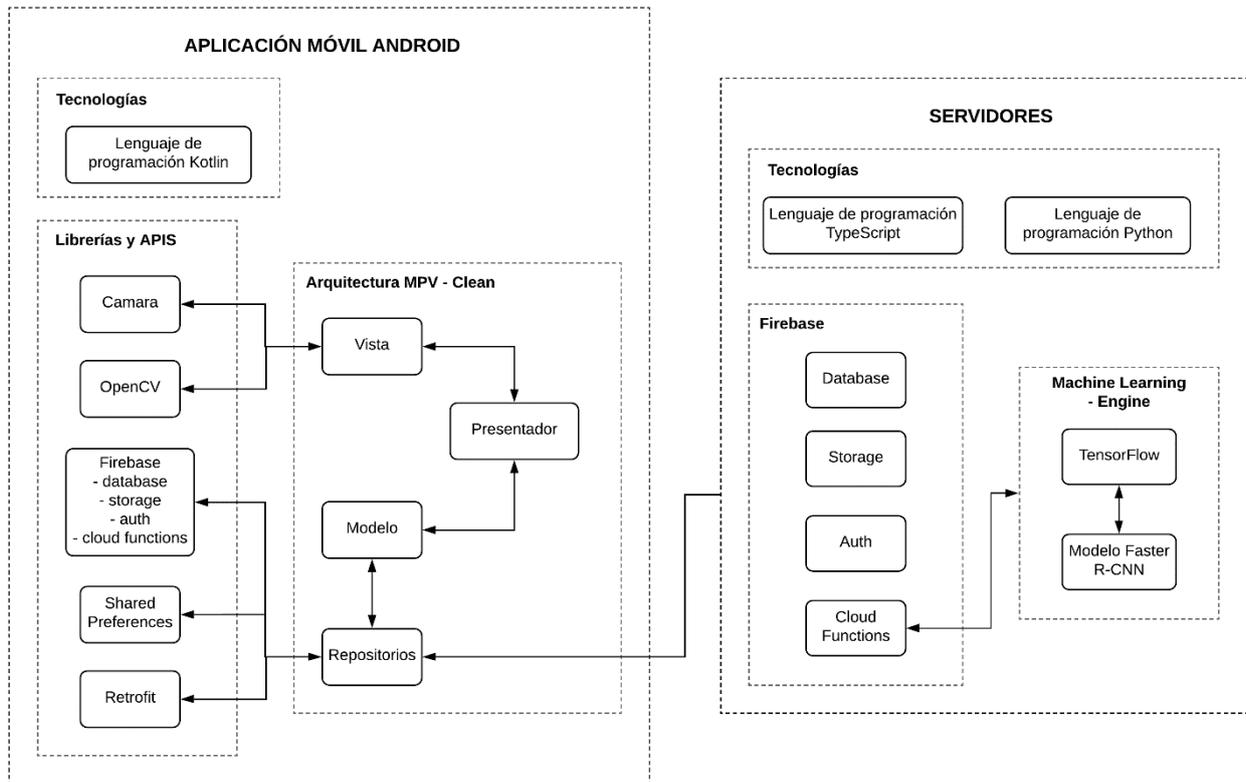
### 6.3.2.3 Planificación de fases

En el Anexo 7 se describe las fases e iteraciones ejecutadas para el desarrollo de la aplicación basadas en la metodología Mobile-D [44].

### 6.3.2.4 Diseño del sistema

La arquitectura del sistema propuesta para esta aplicación se presenta en la

Figura 20, en donde se encuentran las tecnologías, librerías, APIS y servidores



utilizados.

Figura 20. Arquitectura del sistema

- **Aplicación Android.** Aplicación cliente desarrollada en el lenguaje de programación Kotlin, en base a las arquitecturas de desarrollo MVP y Clean [45] para su construcción ordenada y escalable, además se integran librerías y APIS como la cámara y OpenCV [18] con el fin de gestionar las fotografías capturadas por el usuario. Por último, se utilizan las librerías Firebase y Retrofit, que permiten establecer la comunicación con los servidores.
- **Firebase.** Servidor como servicio de la aplicación, en donde se encuentra la base de datos no relacional en formato JSON y el almacenamiento de archivos, además dispone de funcionalidades para la autenticación de usuarios, y el desarrollo de funciones para la comunicación con el servidor Machine Learning - Engine.
- **Machine Learning – Engine.** Servidor como servicio de inteligencia artificial basado en TensorFlow, en donde se encuentra el modelo de detección de objetos Faster R-

CNN, este recibe como entrada una imagen y retorna un objeto JSON de los posibles objetos detectados.

### 6.3.2.5 Diagramas de Clase

A continuación, se describe el diagrama que conforma el sistema, las clases User, Prototype y ViewWidget son la base de la aplicación (ver Figura 21), mientras que las clases Detection y Box forman parte del diagrama usado para recibir e instanciar el resultado de inferencia que retorna el servidor (ver Figura 22).

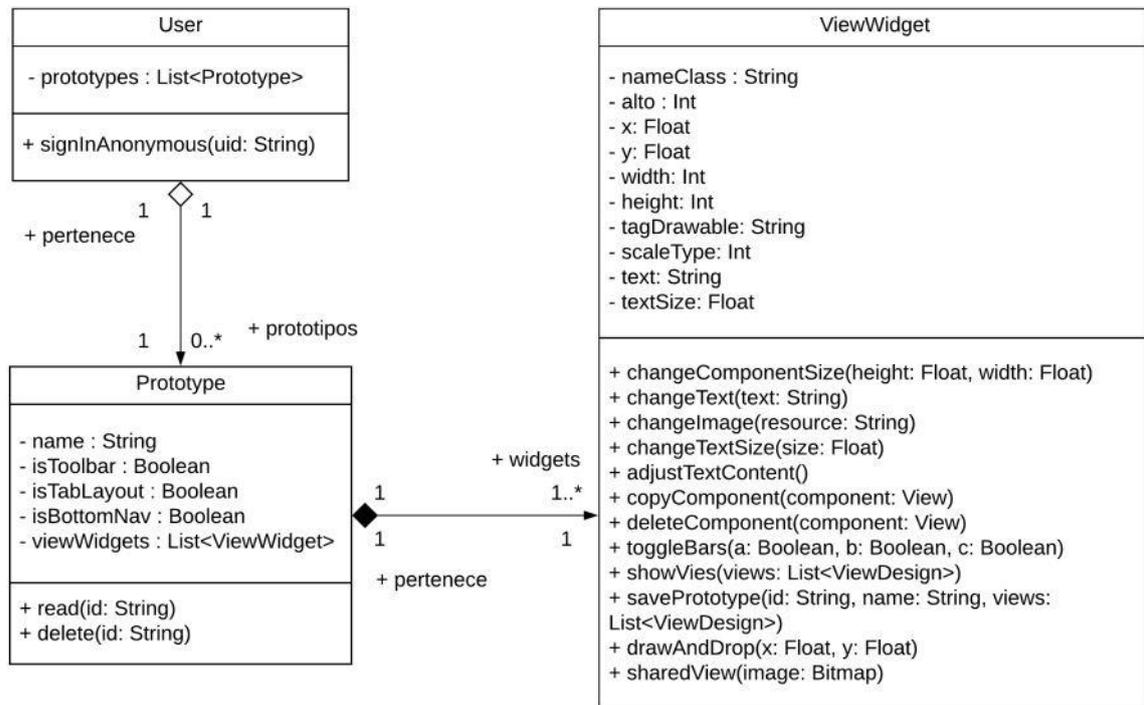


Figura 21. Diagrama de clase de aplicación

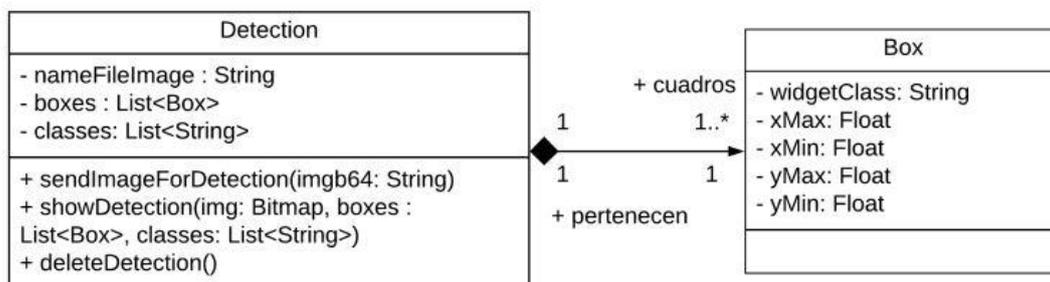


Figura 22. Diagrama de clase de detección

La base de datos NoSQL posee una estructura JSON la cual está compuesta por usuarios, prototipos, vistas o widgets. Para observar su estructura véase el Anexo 8.

### 6.3.2.6 Interfaz gráfica de usuario

La Figura 23 describe el diagrama de navegación de la aplicación móvil. Para observar la interfaz gráfica de la aplicación véase el Anexo 16.

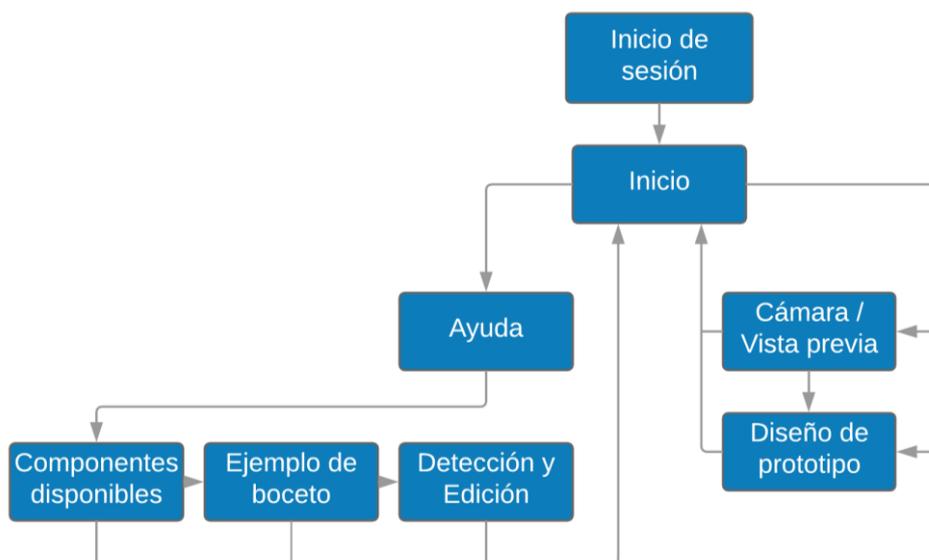


Figura 23. Diagrama de navegación de la aplicación móvil

### 6.3.3 Producción

En esta etapa inició la programación de la aplicación de acuerdo a los módulos, historias de usuario, prototipos de interfaz gráfica y planificación obtenidas en etapas anteriores. A continuación, se describe el desarrollo de la aplicación cliente que abarca la interfaz gráfica de usuario y lógica de negocio, así como la comunicación con la nube.

#### 6.3.3.1 Interfaz gráfica de usuario y lógica de negocio

- La programación de interfaz gráfica de usuario se realizó acorde a los prototipos de manera responsiva. Se hizo uso de librerías actualizadas y oficiales de Material Design para Android de permitiendo una correcta compatibilidad y ejecución en versiones de Android 5 o superior.
- La lógica de negocio de desarrollo acorde a las historias de usuario y módulos, además se aplicó la arquitectura MVP para dividir y desacoplar la vista de la lógica de negocio, permitiendo un flujo limpio y de fácil mantenimiento.

### **6.3.3.2 Comunicación con la nube**

Para el desarrollo y consumo de datos en la nube se utilizó Firebase, servidor en donde están alojados los datos de usuarios, además se implementaron funciones desplegadas en Firebase para el consumo del modelo de detección de objetos en ML - Engine. Toda esta comunicación fue organizada bajo la arquitectura Clean [45], la cual permitió disponer todas las funciones en el momento que se necesiten, además de separar la comunicación con la nube de la lógica de negocio, esto conduce a una estructura ordenada y escalable.

### **6.3.4 Estabilización y pruebas de aplicación**

En esta etapa se realizan las pruebas de módulo e integración de módulos, de tal manera que se obtengan los resultados esperados acordes a los requerimientos (ver sección 6.3.1.6), caso contrario se procede a reparar la funcionalidad. A continuación, se detallan cada una de estas pruebas.

#### **6.3.4.1 Pruebas de módulo**

Las pruebas de módulo evalúan que los resultados obtenidos sean los esperados, de acuerdo a un conjunto de pasos. En el Anexo 9 se encuentran las tablas que detallan dichas pruebas.

#### **6.3.4.2 Pruebas de integración de módulo**

Las pruebas de integración de módulo validan la información requerida entre ellos, para buscar estabilidad y correcto funcionamiento. En el Anexo 10 se encuentran las tablas que detallan dichas pruebas.

## **6.4 Realización de escenario de pruebas**

En esta sección se ejecutó un conjunto de pruebas las cuales permitieron evaluar la calidad y correcto funcionamiento del sistema.

Para la aplicación móvil se ejecutaron pruebas de funcionalidad y rendimiento en un conjunto de celulares inteligentes virtuales a través de Google Play Console, herramienta oficial para el despliegue, control y monitoreo de aplicaciones móviles Android [46].

Por otra parte, el modelo de detección de objetos se evaluó con TensorBoard, herramienta que mide la calidad de inferencia del modelo (ver sección 4.2.2.2). Por

último, se expuso el sistema en un escenario real, en donde interactuó con el usuario final (ver sección 5.3.1) proporcionando validación y retroalimentación del mismo.

#### 6.4.1 Prueba de funcionalidad y rendimiento de la aplicación móvil

La Figura 24 presenta la evolución de versiones de la aplicación (eje x) y el número de dispositivos probados (eje y). Además, se observan errores de funcionalidad y rendimiento en versiones anteriores a 10, debido a la falta de permisos en servicios, notificaciones y compatibilidad de cámara. Sin embargo, estos errores fueron resueltos con éxito, como se muestra en la versión 10.



Figura 24. Control de versiones en Play Store

La Figura 25 detalla los diferentes dispositivos móviles en los que fue evaluada la aplicación, junto a su respectiva aprobación, modelo y versión de sistema operativo. Además, se observa que la aplicación superó las pruebas con éxito en los diferentes dispositivos virtuales. Cabe resaltar que, estos dispositivos son establecidos de acuerdo a una clasificación de mayor uso por usuarios según Google Play Console [46].

	Nombre del modelo	Versión de Android	Configuración regional
✓	Pixel 2	Android 8.1	Español (Estados Unidos)
✓	Xperia XZ1 Compact	Android 8.0	Español
✓	Moto G4 Play	Android 6.0	Español
✓	Mate 9	Android 7.0	Español
✓	LG G6	Android 7.0	Español (Estados Unidos)
✓	Pixel 2	Android 9	Español (Estados Unidos)
✓	Galaxy S9	Android 8.0	Español
✓	P8 Lite	Android 5.0	Español (Estados Unidos)
✓	K3 2017	Android 6.0	Español (Estados Unidos)
✓	Galaxy J7(2016)	Android 6.0	Español (Estados Unidos)
✓	Galaxy J1 Ace	Android 5.1	Español

Figura 25. Dispositivos de evaluación de aplicación móvil

#### 6.4.2 Prueba de precisión del modelo Faster R-CNN.

En el Anexo 12 se observa la clase evaluada junto a la precisión del modelo (eje y) y el número de iteraciones (eje x). Los resultados contemplan valores de precisión entre 0.97 ( $0.97 * 100 = 97\%$ ) y 1 mAP ( $1 * 100 = 100\%$ ), lo que significa que el modelo posee una tasa de precisión aceptable con los datos entrenados en todas las clases. Sin embargo, su comportamiento puede ser diferente cuando tenga que inferir datos nunca antes vistos en el entrenamiento.

#### 6.4.3 Prueba en escenario real

La prueba de escenario real fue realizada en los laboratorios informáticos de la carrera de Ingeniería en Sistemas de la Universidad Nacional de Loja, en donde intervinieron 19 alumnos del ciclo 6to "A" que fueron seleccionados a través de la técnica de muestro no probabilístico por conveniencia, ya que aprobaron la materia de Programación Avanzada en donde estudiaron el desarrollo de aplicaciones móviles (ver sección 5.4).

La prueba en escenario real consistió en la ejecución de los siguientes pasos por parte de los estudiantes:

- Exposición a los estudiantes sobre los objetivos, funcionalidades y procesos que involucra la aplicación.
- Descarga de la aplicación desde Play Store (tienda oficial de Google).
- Dibujado del boceto sobre una hoja con los componentes disponibles.
- Captura del boceto a través de la cámara dentro de la aplicación.
- Gestionar el prototipo resultante con las funcionalidades disponibles.
- Desarrollo de una encuesta por objeto de validación.

Los resultados a las interrogantes planteadas en la encuesta fueron los siguientes:

- **¿El uso de la aplicación le facilitó o agilizó al crear un prototipo?**

El 95% de encuestados afirmó que el proyecto les ayudaba a crear prototipos de manera rápida o fácil, ya que solo tenían que dibujar, mientras que el 5% respondió que no (ver Figura 26).



Figura 26. Resultado de pregunta 1 de escenario real

- **¿Utilizaría la aplicación para crear prototipos de interfaz de usuario en sus proyectos?**

El 84% de encuestados afirmó que utilizarían la aplicación para crear prototipos en sus proyectos debido a su sencillez y facilidad de uso, mientras que el 16% respondió negativamente (ver Figura 27).

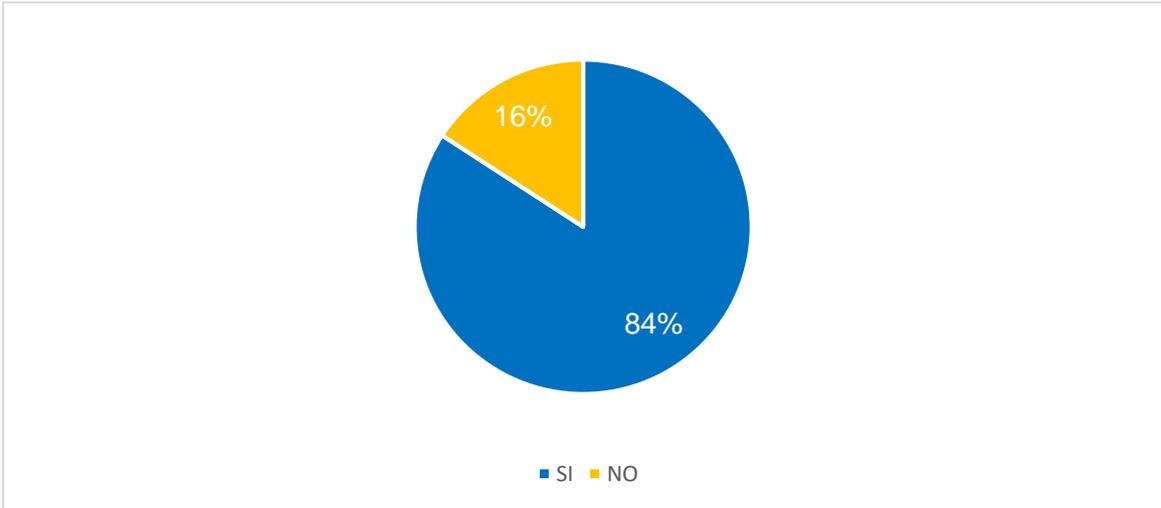


Figura 27. Resultado de pregunta 2 de escenario real

- **La detección de componentes fue:**

El 68% de los encuestados afirmó que la detección por parte del modelo era buena y aceptable, el 26% afirmó que era excelente ya que detectaba todo lo que dibujaban, mientras que, el 6% respondió que el modelo no detectaba su dibujo, esto se debe a la forma de trazo que posee cada individuo (ver Figura 28).

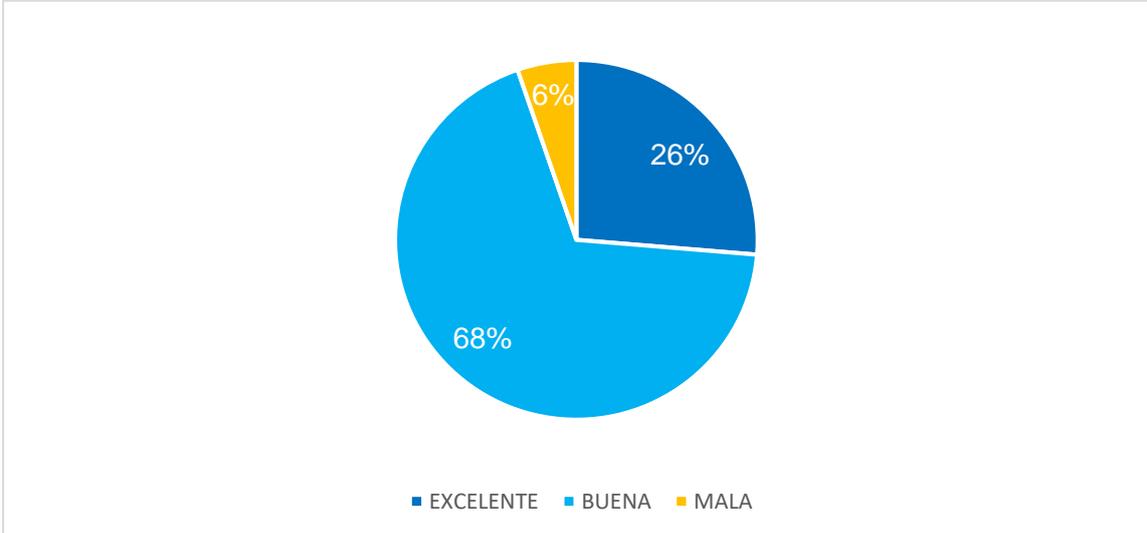


Figura 28. Resultado de pregunta 3 de escenario real

- **¿Encontró problemas en el funcionamiento de la aplicación?**

Con respecto a la pregunta de la Figura 29, el error se debía problemas en algunos dispositivos Samsung antiguos, en donde la orientación de la foto capturada era errónea. Este comportamiento no se predijo durante el desarrollo, sin embargo, se

corrigió y se publicó nuevamente en Play Store, la validación de esto se muestra en la Figura 24, en donde la versión de código 10 de la aplicación no presenta errores.

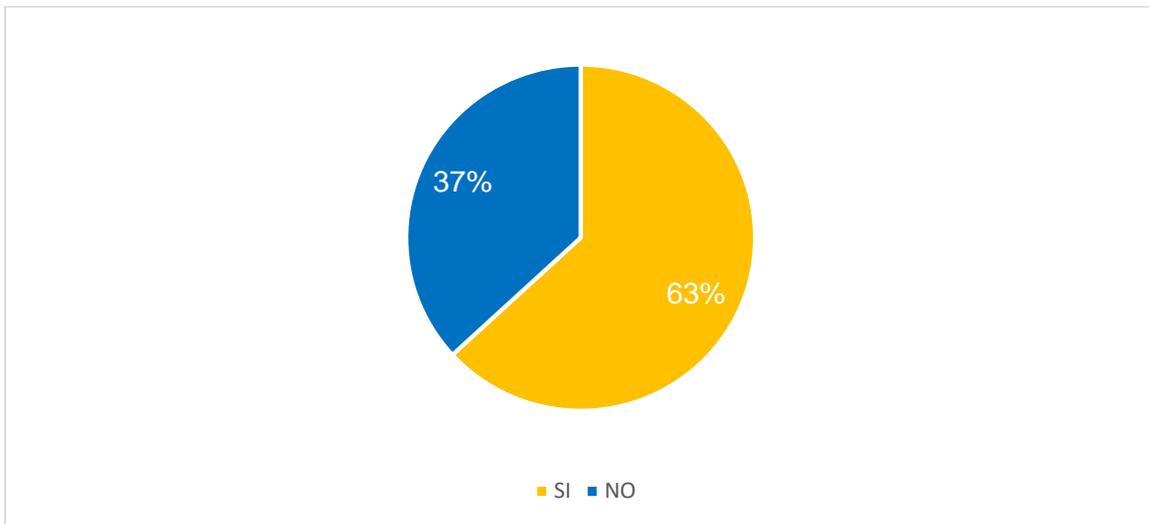


Figura 29. Resultado de pregunta 4 de escenario real

- **Recomendaciones**

Además, los evaluadores sugirieron recomendaciones para mejorar la aplicación, las cuales fueron:

- Opción que permita seleccionar el área de interés a detectar en la fotografía.
- Opción que facilite el desplazamiento y alineación de los componentes en el diseño.

Todas estas recomendaciones se tomaron en cuenta y se añadieron en la aplicación móvil.

## **7. Discusión**

### **7.1 Desarrollo de la propuesta alternativa**

A continuación, se presenta un análisis de los resultados obtenidos según el cumplimiento de objetivos del presente trabajo de titulación.

#### **7.1.1 Objetivo Específico 1: Análisis de los componentes visuales para smartphone de la guía Material Design para Android**

El cumplimiento de este objetivo consistió en la selección de componentes de interfaz gráfica de usuario de la guía oficial Material Design para Android (ver sección 6.1.1 TABLA IX); los cuales fueron utilizados en el sistema. Esto se logró aplicando la observación directa sobre cinco aplicaciones de la tienda oficial para Android Play Store, en la categoría más popular (ver sección 6.1.1 TABLA IV TABLA V TABLA VI TABLA VII TABLA VIII).

El análisis determinó que, la cantidad de componentes más utilizados fueron diez, de acuerdo a su insistente presentación en las diferentes pantallas de las aplicaciones analizadas (ver sección 6.1.1 TABLA IX). Para que el usuario y la visión artificial sepan reconocer un componente para ser dibujado o detectado, fue necesario la asignación de un identificador único, en este caso, una figura que represente a dicho componente dentro de la aplicación. Esta asignación se realizó basándose en las representaciones gráficas de los componentes de la guía Material Design para Android (ver sección 6.1.2 TABLA X).

#### **7.1.2 Objetivo Específico 2: Análisis de algoritmos de detección de objetos en imágenes**

El cumplimiento de este objetivo consistió en la comparación bibliográfica en donde se enfrentaron los algoritmos de visión artificial tradicionales frente al estado de arte (ver sección 4.2.3.3). Mediante la comparativa se demuestra que, las redes neuronales convolucionales superan en precisión, velocidad de entrenamiento e inferencia cuando se ejecutan en GPU, mientras que los algoritmos tradicionales son compatibles únicamente con CPU. Además, la precisión o calidad de inferencia con técnicas redes neuronales poseen un mejor resultado en objetos ubicados en orientaciones levemente diferentes a las muestras de entrenamiento.

La comparativa bibliográfica determinó el uso de una red neuronal convolucional. Ahora, para determinar que red se adapta a los requerimientos del proyecto, se basó en comparativas de fuentes bibliográficas confiables, en donde se enfrentan diferentes modelos pre-entrenados (ver sección 4.2.3.3 Figura 9). Además, se determinó realizar una propia comparativa entre dos modelos de estructuras diferentes; por un lado, SSD, modelo de tiempo corto de inferencia, pero poco preciso en objetos de pequeña escala, frente a Faster R-CNN, un modelo con tiempo de inferencia mayor, pero con alta precisión (ver sección 6.2.2).

Al realizar la comparativa experimental en este proyecto, se confirma la comparativa bibliográfica. El modelo con mayor precisión es Faster R-CNN, ya que obtuvo mejores resultados en la comparativa de detección de objetos dibujados por el usuario (ver sección 6.2.2.4 TABLA XI). Sin embargo, el tiempo que ocupa en realizar la inferencia es de aproximadamente 10 segundos más que SSD, pero se decidió sacrificar tiempo, con el fin de detectar diferentes tamaños de componentes dibujados.

El despliegue del modelo resultante se realizó en un servidor como servicio llamado Machine Learning – Engine, ya que requiere un hardware de GPU o CPU capaz de procesarlo en el menor tiempo posible, en este caso en 15 segundos aproximadamente. Además, se aprovechó el crédito brindado por dicha plataforma con el fin de realizar pruebas experimentales de forma gratuita (ver sección 6.2.2.4 Figura 18 y 6.2.3).

Los trabajos relacionados de redes neuronales convolucionales en Android utilizando TensorFlow (ver sección 4.3.2), basan su proceso en modelos pre entrenados económicos computacionalmente que se ejecutan directamente en el hardware del dispositivo móvil, aumentando el tiempo de inferencia y reduciendo la carga eléctrica de la batería. Mientras que, en el presente trabajo de titulación, libera al cliente móvil de dicho procesamiento y efectos secundarios, mediante su ejecución en la nube.

### **7.1.3 Objetivo Específico 3: Diseño y desarrollo de la aplicación móvil**

El resultado de este objetivo comprendió el diseño y desarrollo de una aplicación móvil Android, en donde se aplicó la metodología ágil Mobile-D permitiendo realizar una ingeniería de software clara y con documentación necesaria (ver sección 6.3).

La aplicación desarrollada permite al usuario crear prototipos básicos de interfaz gráfica de usuario mediante la captura de una fotografía al boceto dibujado sobre una hoja. Además, permite editar los componentes del prototipo con opciones básicas de diseño (ver sección 6.3.1.4). También se integran funciones de gestión de prototipos para que

el usuario pueda administrarlos. Todas estas funciones fueron agregadas de acuerdo a los requerimientos obtenidos en la ingeniería de software (ver sección 6.3.1.7).

Los trabajos relacionados que permiten crear prototipos de interfaz gráfica (ver sección 4.3.1), implementan sus componentes a través de diseños pre establecidos, por lo que no brindan realidad en sus prototipos, además quedan obsoletos con el tiempo. Mientras que, en el presente trabajo de titulación se crea una instancia del componente gráfico nativo y real del sistema operativo Android en el que se esté ejecutando la aplicación, presentando un componente gráfico actualizado y existente en Material Design. Además, se innova en el diseño de prototipos a través de la aplicación de visión artificial a un boceto, frente al diseño tradicional de arrastrar y soltar, liberando al usuario de interfaces de diseño y simplificando dicho proceso.

#### **7.1.4 Objetivo Específico 4: Realización de escenario de pruebas**

El cumplimiento de este objetivo consistió en la ejecución de un conjunto de pruebas con el fin de evaluar el correcto funcionamiento del sistema. La aplicación móvil se validó a través de Google Play Console, en donde se evaluaron términos de funcionalidad y rendimiento dentro de un conjunto de teléfonos inteligentes virtuales más utilizados en el mercado (ver sección 6.4.1 Figura 25). Los resultados presentaron errores debido a la falta de permisos en servicios, notificaciones y compatibilidad de cámara, los cuales fueron reparados satisfactoriamente como lo indica la evolución de la aplicación en la sección 6.4.1 Figura 24, en donde la versión 10 de compilación de la aplicación se encuentra estable y aprobada en los diferentes celulares inteligentes (ver Figura 25).

Por otro lado, la evaluación del modelo de detección de objetos fue realizada a través de TensorBoard (ver sección 4.2 TABLA XLI) en donde los resultados de precisión fueron entre 97% y 100%, lo que estima una inferencia prometedora con los datos de entrenamiento.

Por último, se realizó una evaluación del sistema en un entorno real, con alumnos de la carrera de Ingeniería en Sistemas de la Universidad Nacional de Loja que aprobaron recientemente la materia de Programación Avanzada. El 95% de los alumnos afirmaron que la aplicación les permitió crear prototipos básicos de manera fácil y rápida (ver sección 6.4.3 Figura 26), además, el 94% determinó que la precisión del modelo de detección era muy buena, sin embargo, en algunos casos la detección de componentes fue errónea debido a la manera de dibujar de algunos estudiantes.

## 7.2 Valoración técnica económica ambiental

El presente trabajo de titulación está realizado en base a los conocimientos adquiridos durante el transcurso de la carrera de Ingeniería en Sistemas, enfocados en el desarrollo de un sistema que permita crear prototipos básicos de interfaz gráfica de usuario, de manera fácil y rápida utilizando visión artificial. El mismo que finalizó de manera satisfactoria, gracias a la inversión en recursos humanos, técnicos y tecnológicos, materiales y servicios e imprevistos; los cuales se describen a continuación.

### 7.2.1 Recursos Humanos

En la TABLA XVI se describe el presupuesto utilizado en talento humano.

TABLA XVI. PRESUPUESTO EN TALENTO HUMANO

ROL	HORAS	VALOR UNITARIO (\$)	VALOR TOTAL (\$)
Responsable	1000	5,00	5.000,00
Director	50	20,00	1.000,00
Estudiantes	19	0,00	0,00
<b>Total</b>			<b>6.000,00</b>

### 7.2.2 Técnicos y Tecnológicos

En la TABLA XVII se describe el presupuesto utilizado en recursos técnicos y tecnológicos.

TABLA XVII. PRESUPUESTO EN RECURSOS TÉCNICOS Y TECNOLÓGICOS

RECURSO	CANTIDAD	VALOR UNITARIO (\$)	VALOR TOTAL (\$)
<b>Hardware</b>			
Computador portátil	1	900,00	900,00
Smartphone	1	200,00	200,00
Memoria Flash	1	10,00	10,00
<b>Subtotal</b>			<b>1.110,00</b>
<b>Software</b>			
Android Studio	1	00,00	00,00
Sublime Text	1	00,00	00,00
TensorFlow	1	00,00	00,00
OpenCV	1	00,00	00,00
Lenguaje Java Script	1	00,00	00,00
Google Play Console	1	25,00	25,00
Lenguaje Kotlin	1	00,00	00,00
Lenguaje Python	1	00,00	00,00
Lucidchart	1	00,00	00,00

Windows 10 Pro	1	15,00	15,00
Office 2016	1	15,00	15,00
<b>Subtotal</b>			55,00
<b>Total</b>			<b>1.195,00</b>

### 7.2.3 Materiales y Servicios

En la TABLA XVII se describe el presupuesto utilizado en recursos de oficina; mientras que en la TABLA XIX, se describe el presupuesto utilizado en servicios.

TABLA XVIII. PRESUPUESTO EN RECURSOS DE OFICINA

RECURSO	CANTIDAD	VALOR UNITARIO (\$)	VALOR TOTAL (\$)
Cartuchos	3	25,00	75,00
Resma de papel	2	4,00	8,00
Esferos	10	0,45	4,50
Lápices	10	0,30	3,00
<b>Total</b>			<b>90,50</b>

TABLA XIX. PRESUPUESTO EN SERVICIOS

SERVICIO	CANTIDAD	VALOR UNITARIO (\$)	VALOR TOTAL (\$)
<b>Servidor – Nube</b>			
Firestore plan Spark	1	00,00	00,00
Predicción en línea (ML – Engine)	5000	0,056	280,00
IBM Power AI Vision	1	00,00	00,00
<b>Subtotal</b>			280,00
<b>Otros</b>			
Transporte	25	0,30	7,50
Copias	100	0,02	2,00
Horas de internet	100	0,75	75,00
Anillado de documento	5	1,00	5,00
<b>Subtotal</b>			89,50
<b>Total</b>			<b>369,50</b>

### 7.2.4 Imprevistos

Para los imprevistos, se utilizó el 10% del presupuesto total, sumado al valor total del trabajo de titulación, como se describe en la TABLA XX.

TABLA XX. PRESUPUESTO TOTAL DEL PROYECTO

RECURSO	TOTAL (\$)
Talento Humano	6.000,00
Recursos Técnicos y Tecnológicos	1.195,00
Recursos de Oficina	90,50
Servicios	369,50
Imprevistos	765,50
<b>Total</b>	<b>8.420,50</b>

Aquellos servicios que no se predijo utilizar, como IBM Power AI Vision y Machine Learning – Engine, marcaron el buen camino del trabajo de titulación, ya que, permitieron el entrenamiento y despliegue en la nube del modelo de detección de objetos para su consumo desde la aplicación, liberando al cliente móvil de esta carga de trabajo.

La aplicación proporciona opciones de diseño simples y de fácil uso, con el objetivo de reducir el tiempo invertido en dichas tareas de prototipado de interfaz gráfica de usuario, permitiendo enfocar los recursos en actividades de diseño y desarrollo de software. Esto ayudará a aquellos programadores novatos que están empezando en el desarrollo de proyectos Android, a sobrellevar de mejor manera las tareas de prototipado gráfico, para enfocarse en la ingeniería de software.

## 8. Conclusiones

Dado por concluido el presente trabajo de titulación, se dan a conocer las siguientes conclusiones:

- El análisis de componentes Material Design sobre 5 aplicaciones Android (WhatsApp Messenger, Messenger Lite, Facebook Lite, Instagram y Wish) encontradas en el top 10 de popularidad en Play Store, ayudó a la delimitación de los 10 componentes gráficos más usados (Check Box, Radio Button, Switch, Button, Floating Action Button, Chip, Recycler View, Text View, Edit Text, Image View, Tool Bar, Bottom Navigation y Tab Layout), además, facilitó la asignación de formas o figuras únicas para su identificación por parte del modelo de detección de objetos y el usuario.
- La recolección de datos para el entrenamiento del modelo de detección de objetos realizada a 19 estudiantes de la carrera de Ingeniería en Sistemas de la Universidad Nacional de Loja, permitió la generación de un conjunto de datos con 862 imágenes representativas, debido a la variación en el trazo único de cada individuo.
- La implementación de TensorFlow en la nube (IBM Power AI Vision), permitió entrenar los modelos SSD y Faster R-CNN hasta 500K iteraciones en 6 días.
- El uso de servicios Firebase y ML – Engine permitieron realizar el despliegue, evaluación y control de procesos de aplicación y aprendizaje profundo, proveyendo recursos de hardware y software escalable (se adapta a demanda), facilitando la concentración de esfuerzos en el desarrollo de procesos de aplicación móvil y visión artificial.
- El uso de Google Play Console y la evaluación del sistema en un escenario real (aplicada a desarrolladores iniciales), permitieron corregir y estabilizar la aplicación, con el fin de alcanzar su correcto funcionamiento, basándose en diferentes modelos de teléfonos inteligentes (Pixel 2, Xperia XZ1, Moto G4 Play, Mate 9, LG G6, Galaxy S9, P8 Lite, K3 2018, Galaxy J7, Galaxy J1 Ace).
- La visión artificial facilita el proceso de creación de prototipos de interfaz gráfica de usuario, ya que mediante un breve boceto dibujado, se puede obtener una idea

gráfica y real del prototipo, en este caso, utilizando los componentes Material Design propios del sistema operativo Android.

## 9. Recomendaciones

Dado por concluido el presente trabajo de titulación, se plantean las siguientes recomendaciones:

- El uso de una unidad gráfica de procesamiento dedicada (GPU) NVIDIA GTX o superior [47], para reducir el tiempo de entrenamiento e inferencia en proyectos que empleen técnicas de aprendizaje profundo en visión artificial.
- Aumentar la cantidad de imágenes representativas y nuevos componentes Material Design al modelo de visión artificial, para mejorar su calidad y diversidad de inferencia.
- Extender la aplicación a otras plataformas incorporando sus componentes gráficos nativos, como por ejemplo UIKit para iOS.
- Convertir el prototipo detectado a un lenguaje de diseño, como por ejemplo XML u otros. Para evitar su reconstrucción y agilizar su implementación en la aplicación real.
- El usuario debe evitar borrones o rayones excesivos en su dibujo, para mejorar la calidad de inferencia y evitar detecciones erróneas (falsos positivos) en las fotografías capturadas.

## 10. Bibliografía

- [1] Android Developers, "Interfaz de usuario," *Android Developers*, 2014. [Online]. Available: <https://developer.android.com/guide/topics/ui/index.html?hl=es-419>. [Accessed: 02-Dec-2017].
- [2] L. González, "El diseño de interfaz gráfica para publicaciones digitales," *DGSCA-UNAM*, México, p. 12, Aug-2004.
- [3] M. De Teja and G. Manuel, "Ergonomía e interfases de interacción humano-computadora," *IX Congr. Int. Ergon.*, pp. 1–8, 2007.
- [4] J. Abascal and R. Morillón, "Tendencias en Interacción Persona Computador," *Intel. Artif.*, vol. 6, no. 16, 2010.
- [5] Google, "Develop for Android - Material Design." [Online]. Available: <https://material.io/develop/android/>. [Accessed: 31-Oct-2018].
- [6] "Material Design para Android | Android Developers." [Online]. Available: <https://developer.android.com/design/material/index.html?hl=es-419>. [Accessed: 24-Aug-2017].
- [7] Apple, "UIKit | Documentación para desarrolladores de Apple." [Online]. Available: <https://developer.apple.com/documentation/uikit>. [Accessed: 31-Oct-2018].
- [8] R. C. Choque, "Areas de aplicación de la Inteligencia Artificial," *Rev. Boliv.*, p. 17, 2013.
- [9] J. F. SPARC (Organization), John J.; Archila D. and Universidad Tecnológica de Pereira., "Detección y análisis de movimiento usando visión artificial.," *Scientia Et Technica*, 1995. [Online]. Available: <http://www.redalyc.org:9081/html/849/84922625031/>. [Accessed: 02-Dec-2017].
- [10] P. Infaimon, "Visión artificial aplicada a la industria," p. 28, 2010.
- [11] C. Andrés and O. Santiago, "Formatos De Imagen Digital," *Rev. Digit. Univ.*, vol. 5, p. 10, 2005.
- [12] S. L. Mora, "Imagen digital Índice de contenidos Intro Plataformas y hardware," pp. 1–48, 2008.
- [13] E. García, "Visión Artificial," *Fund. para la Univ. Oberta Catalunya*, p. 115, 2002.

- [14] Nishant Shukla, "Machine Learning with TensorFlow," p. 325, 2017.
- [15] M. Abadi *et al.*, "TensorFlow: A System for Large-Scale Machine Learning TensorFlow: A system for large-scale machine learning," *12th USENIX Symp. Oper. Syst. Des. Implement. (OSDI '16)*, pp. 265–284, 2016.
- [16] TensorFlow, "TensorBoard: Visualizing Learning | TensorFlow." [Online]. Available: [https://www.tensorflow.org/guide/summaries\\_and\\_tensorboard](https://www.tensorflow.org/guide/summaries_and_tensorboard). [Accessed: 31-Oct-2018].
- [17] TensorFlow, "TensorFlow Guide | TensorFlow." [Online]. Available: <https://www.tensorflow.org/guide/>. [Accessed: 12-Nov-2018].
- [18] OpenCV, "OpenCV: OpenCV modules." [Online]. Available: <https://docs.opencv.org/master/>. [Accessed: 12-Nov-2018].
- [19] E. J. Carmona Suárez, "Máquinas de Vectores Soporte (SVM)," pp. 1–25, 2014.
- [20] R. P. Palacios, "Implementación del algoritmo de detección facial de Viola-Jones," 2010.
- [21] L. M. Guevara, J. D. Echeverry, and W. Ardila, "Faces Detection in Digital Images Using Cascade Classifiers," *Sci. Tech.*, vol. XIV, no. 38, pp. 1–6, 2008.
- [22] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, and S. E. Reed, "{SSD:} Single Shot MultiBox Detector," *CoRR*, vol. abs/1512.0, 2015.
- [23] A. Cvpr and P. Id, "Speed / accuracy trade-offs for modern convolutional object detectors - 3562," *Cvpr*, pp. 7310–7319, 2017.
- [24] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, 2017.
- [25] J. Dai, Y. Li, K. He, and J. Sun, "R-FCN: Object Detection via Region-based Fully Convolutional Networks," no. Nips, 2016.
- [26] E. Cengil, "Comparison Of Hog ( Histogram of Oriented Gradients ) and Haar Cascade Algorithms with a Convolutional Neural Network Based Face Detection Approaches," vol. 3, pp. 244–255, 2017.
- [27] R. T. Kelly, "Critical Comparison of the Classification Ability of Deep Convolutional Neural Network Frameworks with Support Vector Machine Techniques in the

Image Classification Process Critical Comparison of the Classification Ability of Deep Convolutional Neural Net,” 2017.

- [28] “Análisis predictivo: Cloud Machine Learning Engine | Cloud Machine Learning Engine (Cloud ML Engine) | Google Cloud.” [Online]. Available: <https://cloud.google.com/ml-engine/>. [Accessed: 04-Jul-2018].
- [29] IBM, “IBM PowerAI Vision - Linux on Power Developer Portal.” [Online]. Available: <https://developer.ibm.com/linuxonpower/deep-learning-powerai/technology-previews/powerai-vision/>. [Accessed: 12-Nov-2018].
- [30] Marvel, “Apps | Marvel.” [Online]. Available: <https://marvelapp.com/apps/>. [Accessed: 12-Nov-2018].
- [31] “Prott - Prototyping tool for Web iOS Android apps.” [Online]. Available: <https://prottapp.com/>. [Accessed: 12-Nov-2018].
- [32] C. Alberto, M. Zabaleta, M. I. N. Elva, and C. Rodríguez, “Interpretación del lenguaje de señas utilizando redes neuronales,” 2017.
- [33] J. R. Lenz, J. R. Lenz, and E. A. Cooksey, “Application of Computational Fluid Dynamics to Compressor Efficiency Improvement,” 2015.
- [34] R. Aleixandre-Benavent, G. G. Alcaide, J. G. de Dios, and A. Alonso-Arroyo, “Fuentes de información bibliográfica (I). Fundamentos para la realización de búsquedas bibliográficas/Sources of bibliographic information. Rationale for conducting a literature search,” *Acta Pediatr. Esp.*, vol. 69, no. 3, pp. 131–136, 2011.
- [35] L. B. López and O. Arguedas-Arguedas, “La búsqueda bibliográfica,” *DIAETA (B.Aires)*, vol. 24, no. 3, pp. 31–37, 2009.
- [36] M. Torres, K. Paz, and F. G. Salazar, “Metodos De Recoleccion De Datos Para Una Investigación,” *Fac. Ing. Univ. Rafael Landívar*, vol. 03, pp. 1–21, 2014.
- [37] T. Otzen and C. Manterola, “Técnicas de Muestreo sobre una Población a Estudio Sampling Techniques on a Population Study,” *Int. J. Morphol.*, vol. 35, no. 1, pp. 227–232, 2017.
- [38] A. O. Neill, “Métodos De Investigación Cualitativa,” *Silogismo*, vol. 8, no. 1, pp. 1–11, 2018.

- [39] J. Luis, “El Método de la Investigación Research Method,” *Daena Int. J. Good Conscienc.*, vol. 9, no. 3, pp. 195–204, 2014.
- [40] Tzutalin, “Labellmg.” [Online]. Available: <https://github.com/tzutalin/labellmg>.
- [41] Universidad Santiago de Cali *et al.*, “Evolution of Mobile Software Development from Platform-Specific to Web-Based Multiplatform Paradigm,” *Rev. Tecnol. | J. Technol.*, vol. 12 número, no. 4, pp. 181–183, 2013.
- [42] R. R. Vique, “Métodos para el desarrollo de aplicaciones móviles,” *Univ. Oberta Catalunya*, p. 66, 2010.
- [43] B. Nicolas, “Real-Time database : Firebase INFO-H-415 : Advanced database,” pp. 1–23, 2018.
- [44] P. R. P. B. J. C. A. F. Adam Warterski, “Metodología de desarrollo ágil para sistemas móviles Introducción al desarrollo con Android y el iPhone,” *Rev. Tecnol. | J. Technol.*, vol. 12 número, no. 4, pp. 181–183, 2013.
- [45] M. González, D. Director, A. Pardo, and F. Realitzat, “Clean Architecture Y Rxjava En Android.”
- [46] Google, “Google Play Console.” [Online]. Available: <https://developer.android.com/distribute/console?hl=es>. [Accessed: 11-Apr-2019].
- [47] D. Strigl, K. Kofler, and S. Podlipnig, “Performance and scalability of GPU-based convolutional neural networks,” *Proc. 18th Euromicro Conf. Parallel, Distrib. Network-Based Process. PDP 2010*, pp. 317–324, 2010.

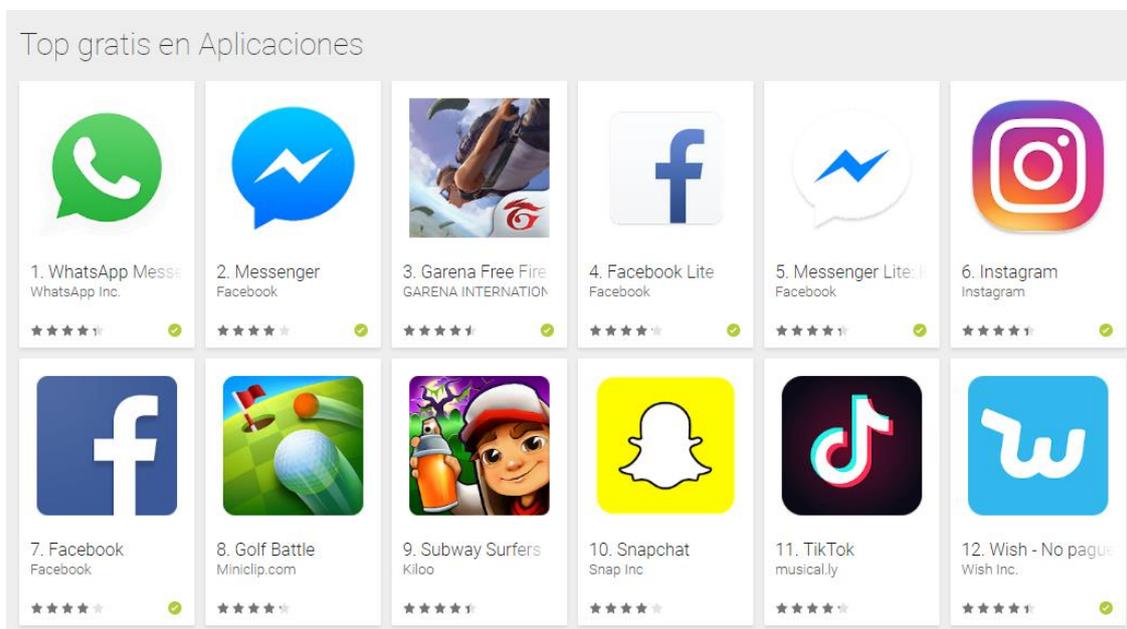
# 11. Anexos

## Anexo 1. Glosario de términos.

TABLA XXI. GLOSARIO DE TÉRMINOS

<b>Término</b>	<b>Descripción</b>
<b>Componente Material Design</b>	Elemento utilizado para la interacción entre el usuario y la aplicación.
<b>Gradle</b>	Paquete de herramientas de compilación avanzadas, para automatizar y administrar el proceso de compilación.
<b>Instant Run</b>	Función para los comandos ejecutar y depurar en Android Studio que reduce de manera significativa el tiempo entre las actualizaciones de tu app.
<b>NDK</b>	Conjunto de herramientas que le permite implementar partes de su aplicación en código nativo, utilizando lenguajes como C y C ++.
<b>Inferencia</b>	Deducir una clase o extraer un resultado de una imagen.
<b>Píxel</b>	Menor unidad que forma parte de una imagen digital.
<b>GPU</b>	Unidad gráfica de procesamiento.
<b>Regresión</b>	Proceso estadístico para estimar las relaciones entre variables.
<b>Búsqueda selectica</b>	Algoritmo que genera propuestas de región al recorrer una imagen.
<b>Red Propuesta de región</b>	Algoritmo que genera propuestas de región a partir de características establecidas.
<b>Propuesta de región</b>	Cuadro delimitador que marca un objeto de interés.
<b>Supresión no máxima</b>	Algoritmo que elimina los cuadros de calificación más baja generados en una detección sobre un objeto de interés, y elige al de mayor puntaje.
<b>mAP</b>	Media de precisión media, es la métrica para medir la precisión de detección de un modelo.
<b>APK</b>	Formato de empaquetado de aplicaciones Android para su despliegue e instalación en dispositivos móviles.
<b>Google Play Console</b>	Plataforma oficial de despliegue y control de aplicaciones Android.
<b>TFRecord</b>	Nombre del formato de almacenamiento binario de TensorFlow.

## Anexo 2. Top 10 de aplicaciones mejor calificadas y más populares en la tienda oficial de Android Google Play.



**Anexo 3. Foto capturada a estudiantes en la recolección de datos para el modelo de detección.**



#### Anexo 4. Código de programación de técnicas de distorsión en imagen.

```
""  
# Uso:  
# Configurar imread, imágenes de entrada  
# Configurar imwrite directorio de salida  
""  
  
import numpy as np  
import cv2  
  
for index in range(1, 14):  
    gray = cv2.imread('C:/images/' + str(index), cv2.IMREAD_GRAYSCALE)  
    adaptative = cv2.adaptiveThreshold(gray, 255,  
cv2.ADAPTIVE_THRESH_GAUSSIAN_C, cv2.THRESH_BINARY, 51, 8)  
    cv2.imwrite( 'C:/images/'+ str(index) +'adaptative.jpg', adaptative)  
    cv2.imwrite( 'C:/images/'+ str(index) +'gray.jpg', gray)
```

## Anexo 5. Código de programación para conversión a formato TFRecord.

```
"""
Uso:
# Desde el directorio tensorflow/models/
# Para crear los datos de entrenamiento:
python generate_tfrecord.py --csv_input=data/train_labels.csv --
output_path=train.record
# Para crear los datos de evaluación:
python generate_tfrecord.py --csv_input=data/test_labels.csv --
output_path=test.record
"""

from __future__ import division
from __future__ import print_function
from __future__ import absolute_import

import os
import io
import pandas as pd
import tensorflow as tf

from PIL import Image
from object_detection.utils import dataset_util
from collections import namedtuple, OrderedDict

flags = tf.app.flags
flags.DEFINE_string('csv_input',
'C:/Users/Yavac/Desktop/images/detection/protoia_labels_eval.csv', '')
flags.DEFINE_string('output_path',
'C:/Users/Yavac/Desktop/images/detection/eval.record', '')
FLAGS = flags.FLAGS

# TO-DO replace this with label map
def class_text_to_int(row_label):
    if row_label == 'edit_text':
        return 1
    if row_label == 'image_view':
        return 2
    if row_label == 'list_view':
        return 3
    if row_label == 'chip':
        return 4
    if row_label == 'radio_button':
        return 5
    if row_label == 'check_box':
        return 6
    if row_label == 'switch':
        return 7
    if row_label == 'button':
        return 8
    if row_label == 'floating_button':
```

```

    return 9
if row_label == 'text_view':
    return 10
else:
    None

def split(df, group):
    data = namedtuple('data', ['filename', 'object'])
    gb = df.groupby(group)
    return [data(filename, gb.get_group(x)) for filename, x in zip(gb.groups.keys(),
gb.groups)]

def create_tf_example(group, path):
    with tf.gfile.GFile(os.path.join(path, '{}'.format(group.filename)), 'rb') as fid:
        encoded_jpg = fid.read()
    encoded_jpg_io = io.BytesIO(encoded_jpg)
    image = Image.open(encoded_jpg_io)
    width, height = image.size

    filename = group.filename.encode('utf8')
    image_format = b'jpg'
    xmin = []
    xmax = []
    ymin = []
    ymax = []
    classes_text = []
    classes = []

    for index, row in group.object.iterrows():
        xmin.append(row['xmin'] / width)
        xmax.append(row['xmax'] / width)
        ymin.append(row['ymin'] / height)
        ymax.append(row['ymax'] / height)
        classes_text.append(row['class'].encode('utf8'))
        classes.append(class_text_to_int(row['class']))

    tf_example = tf.train.Example(features=tf.train.Features(feature={
        'image/height': dataset_util.int64_feature(height),
        'image/width': dataset_util.int64_feature(width),
        'image/filename': dataset_util.bytes_feature(filename),
        'image/source_id': dataset_util.bytes_feature(filename),
        'image/encoded': dataset_util.bytes_feature(encoded_jpg),
        'image/format': dataset_util.bytes_feature(image_format),
        'image/object/bbox/xmin': dataset_util.float_list_feature(xmin),
        'image/object/bbox/xmax': dataset_util.float_list_feature(xmax),
        'image/object/bbox/ymin': dataset_util.float_list_feature(ymin),
        'image/object/bbox/ymax': dataset_util.float_list_feature(ymax),
        'image/object/class/text': dataset_util.bytes_list_feature(classes_text),
        'image/object/class/label': dataset_util.int64_list_feature(classes),
    }))
    return tf_example

```

```
def main(_):
    writer = tf.python_io.TFRecordWriter(FLAGS.output_path)
    path = 'C:/Users/Yavac/Desktop/images/detection/images/'
    examples = pd.read_csv(FLAGS.csv_input)
    grouped = split(examples, 'filename')
    for group in grouped:
        tf_example = create_tf_example(group, path)
        writer.write(tf_example.SerializeToString())

    writer.close()
    output_path = os.path.join(os.getcwd(), FLAGS.output_path)
    print('Successfully created the TFRecords: {}'.format(output_path))

if __name__ == '__main__':
    tf.app.run()
```

## Anexo 6. Historias de Usuario de la aplicación móvil.

TABLA XXII. HISTORIA DE USUARIO: CAPTURA DE FOTO

Historia de Usuario: Captura de Foto		
<b>Identificador:</b> HU-001	<b>Rol:</b> Usuario	
<b>Entrada de datos:</b> Ninguno	<b>Salida de datos:</b> Foto	
<b>Descripción:</b> Necesito capturar y recortar foto con la finalidad de obtener una imagen digital del prototipo		
<b>Criterio de aceptación:</b>		
<ul style="list-style-type: none"> <li>• <b>Éxito</b> <ul style="list-style-type: none"> <li>○ Activación y vista previa de la cámara, captura de foto</li> </ul> </li> <li>• <b>Fracaso</b> <ul style="list-style-type: none"> <li>○ Mensaje advirtiendo al usuario sobre el problema</li> </ul> </li> </ul>		
<b>Dificultad:</b> Media	<b>Esfuerzo:</b> Medio	<b>Prioridad:</b> Alta
<b>Observaciones:</b> El usuario podrá activar o desactivar el flash LED de su cámara		

TABLA XXIII. HISTORIA DE USUARIO: DETECCIÓN DE COMPONENTES

Historia de Usuario: Detección de Componentes		
<b>Identificador:</b> HU-002	<b>Rol:</b> Usuario	
<b>Entrada de datos:</b> Ninguno	<b>Salida de datos:</b> Detección	
<b>Descripción:</b> Necesito enviar una foto del prototipo con la finalidad de detectar los componentes gráficos		
<b>Criterio de aceptación:</b>		
<ul style="list-style-type: none"> <li>• <b>Éxito</b> <ul style="list-style-type: none"> <li>○ Envío de foto al servidor y visualización de resultado</li> </ul> </li> <li>• <b>Fracaso</b> <ul style="list-style-type: none"> <li>○ Mensaje advirtiendo al usuario sobre el problema</li> </ul> </li> </ul>		
<b>Dificultad:</b> Alta	<b>Esfuerzo:</b> Alto	<b>Prioridad:</b> Alta
<b>Observaciones:</b> El usuario puede o no estar dentro de la aplicación durante el proceso de detección		

TABLA XXIV. HISTORIA DE USUARIO: ELIMINACIÓN DE COMPONENTE

Historia de Usuario: Eliminación de Componente		
<b>Identificador:</b> HU-003	<b>Rol:</b> Usuario	
<b>Entrada de datos:</b> Ninguno	<b>Salida de datos:</b> Ninguno	
<b>Descripción:</b> Necesito eliminar un componente con la finalidad de modificar el prototipo		
<b>Criterio de aceptación:</b>		
<ul style="list-style-type: none"> <li>• <b>Éxito</b> <ul style="list-style-type: none"> <li>○ Eliminación de componente</li> </ul> </li> </ul>		
<b>Dificultad:</b> Baja	<b>Esfuerzo:</b> Bajo	<b>Prioridad:</b> Media
<b>Observaciones:</b> Ninguna		

TABLA XXV. HISTORIA DE USUARIO: CAMBIO DE TAMAÑO DE COMPONENTE

Historia de Usuario: Cambio de Tamaño de Componente		
<b>Identificador:</b> HU-004	<b>Rol:</b> Usuario	
<b>Entrada de datos:</b> Tamaño	<b>Salida de datos:</b> Ninguno	
<b>Descripción:</b> Necesito actualizar el tamaño al componente con la finalidad de modificar el prototipo		
<b>Criterio de aceptación:</b>		
<ul style="list-style-type: none"> <li>• <b>Éxito</b> <ul style="list-style-type: none"> <li>○ Actualización de alto o ancho de componente</li> </ul> </li> </ul>		
<b>Dificultad:</b> Baja	<b>Esfuerzo:</b> Bajo	<b>Prioridad:</b> Alta
<b>Observaciones:</b> Restringir el tamaño al límite de la pantalla		

TABLA XXVI. HISTORIA DE USUARIO: CAMBIO DE UBICACIÓN DE COMPONENTE

Historia de Usuario: Cambio de Ubicación de Componente		
<b>Identificador:</b> HU-005	<b>Rol:</b> Usuario	
<b>Entrada de datos:</b> Coordenadas	<b>Salida de datos:</b> Ninguno	
<b>Descripción:</b> Necesito actualizar la ubicación de un componente con la finalidad de modificar el prototipo		
<b>Criterio de aceptación:</b>		
<ul style="list-style-type: none"> <li>• <b>Éxito</b> <ul style="list-style-type: none"> <li>○ Actualización de ubicación de componente</li> </ul> </li> </ul>		
<b>Dificultad:</b> Media	<b>Esfuerzo:</b> Bajo	<b>Prioridad:</b> Alta
<b>Observaciones:</b> Ninguna		

TABLA XXVII. HISTORIA DE USUARIO: ACTUALIZACIÓN DE CONTENIDO DE COMPONENTE

Historia De Usuario: Actualización De Contenido De Componente		
<b>Identificador:</b> HU-006	<b>Rol:</b> Usuario	
<b>Entrada de datos:</b> Contenido	<b>Salida de datos:</b> Ninguno	
<b>Descripción:</b> Necesito actualizar el contenido de un componente con la finalidad de modificar el prototipo		
<b>Criterio de aceptación:</b>		
<ul style="list-style-type: none"> <li>• <b>Éxito</b> <ul style="list-style-type: none"> <li>○ Actualización de texto o imagen de componente según sea el caso</li> </ul> </li> </ul>		
<b>Dificultad:</b> Media	<b>Esfuerzo:</b> Medio	<b>Prioridad:</b> Alta
<b>Observaciones:</b> En componentes que extiendan de TextView cambiar texto y tamaño. Componentes que extiendan de ImageView cambiar la imagen		

TABLA XXVIII. HISTORIA DE USUARIO: COPIADO DE COMPONENTE

Historia de Usuario: Copiado de Componente		
<b>Identificador:</b> HU-007	<b>Rol:</b> Usuario	
<b>Entrada de datos:</b> Ninguno	<b>Salida de datos:</b> Ninguno	
<b>Descripción:</b> Necesito copiar un componente con la finalidad de obtener otro de igual clase y contenido		
<b>Criterio de aceptación:</b>		
<ul style="list-style-type: none"> <li>• <b>Éxito</b> <ul style="list-style-type: none"> <li>○ Duplicado del componente</li> </ul> </li> </ul>		
<b>Dificultad:</b> Baja	<b>Esfuerzo:</b> Medio	<b>Prioridad:</b> Alta
<b>Observaciones:</b> Ninguna		

TABLA XXIX. HISTORIA DE USUARIO: GESTIÓN DE BARRAS DE HERRAMIENTAS

Historia de Usuario: Gestión de Barras de Herramientas		
<b>Identificador:</b> HU-008	<b>Rol:</b> Usuario	
<b>Entrada de datos:</b> Ninguno	<b>Salida de datos:</b> Ninguno	
<b>Descripción:</b> Necesito añadir o remover barras de herramienta o navegación con la finalidad de modificar el prototipo		
<b>Criterio de aceptación:</b>		
<ul style="list-style-type: none"> <li>• <b>Éxito</b> <ul style="list-style-type: none"> <li>○ Se muestra u oculta la barra seleccionada</li> </ul> </li> </ul>		
<b>Dificultad:</b> Media	<b>Esfuerzo:</b> Medio	<b>Prioridad:</b> Alta
<b>Observaciones:</b> Barras disponibles: Toolbar, Tablayout, Bottomnavigation		

TABLA XXX. HISTORIA DE USUARIO: VISUALIZACIÓN DE PROTOTIPO

Historia de Usuario: Visualización de Prototipo		
<b>Identificador:</b> HU-009	<b>Rol:</b> Usuario	
<b>Entrada de datos:</b> Ninguno	<b>Salida de datos:</b> Ninguno	
<b>Descripción:</b> Necesito visualizar solo el marco que contiene al prototipo con la finalidad de una visión específica		
<b>Criterio de aceptación:</b>		
<ul style="list-style-type: none"> <li>• <b>Éxito</b> <ul style="list-style-type: none"> <li>○ Se muestra o esconde la barra de diseño</li> </ul> </li> </ul>		
<b>Dificultad:</b> Baja	<b>Esfuerzo:</b> Bajo	<b>Prioridad:</b> Alta
<b>Observaciones:</b> Usar un botón físico para mostrar de nuevo la barra de opciones		

TABLA XXXI. HISTORIA DE USUARIO: COMPARTICIÓN DE CAPTURA DE PANTALLA

Historia de Usuario: Compartición de Captura de Pantalla		
<b>Identificador:</b> HU-010	<b>Rol:</b> Usuario	
<b>Entrada de datos:</b> Ninguno	<b>Salida de datos:</b> Ninguno	
<b>Descripción:</b> Necesito capturar una imagen del prototipo con la finalidad de compartir el resultado de diseño con aplicaciones externas disponibles en el Smartphone		
<b>Criterio de aceptación:</b>		
<ul style="list-style-type: none"> <li>• <b>Éxito</b> <ul style="list-style-type: none"> <li>○ Se captura una imagen del prototipo y se abren las aplicaciones disponibles para compartir</li> </ul> </li> <li>• <b>Fracaso</b> <ul style="list-style-type: none"> <li>○ Mensaje advirtiendo al usuario sobre el problema</li> </ul> </li> </ul>		
<b>Dificultad:</b> Media	<b>Esfuerzo:</b> Bajo	<b>Prioridad:</b> Alta
<b>Observaciones:</b> Usar un botón físico para compartir, solo se podrá compartir en modo visualización de prototipo		

TABLA XXXII. HISTORIA DE USUARIO: GESTIÓN DE PROTOTIPOS

Historia de Usuario: Gestión de Prototipos		
<b>Identificador:</b> HU-011	<b>Rol:</b> Usuario	
<b>Entrada de datos:</b> Ninguno	<b>Salida de datos:</b> Ninguno	
<b>Descripción:</b> Necesito ver, eliminar, editar y guardar un prototipo con la finalidad de gestionarlos		
<b>Criterio de aceptación:</b>		
<ul style="list-style-type: none"> <li>• <b>Éxito</b> <ul style="list-style-type: none"> <li>○ Se visualiza, elimina, edita y guarda un prototipo</li> </ul> </li> <li>• <b>Fracaso</b> <ul style="list-style-type: none"> <li>○ Mensaje advirtiendo al usuario sobre el problema</li> </ul> </li> </ul>		
<b>Dificultad:</b> Media	<b>Esfuerzo:</b> Medio	<b>Prioridad:</b> Alta
<b>Observaciones:</b> Ninguna		

## Anexo 7. Planificación de fases e iteraciones para el desarrollo de la aplicación móvil.

TABLA XXXIII. PLANIFICACIÓN DE FASES PARA EL DESARROLLO DE LA APLICACIÓN MÓVIL

Fase	Iteración	Descripción
<b>Exploración</b>	Iteración 1	Establecimiento de stakeholders, alcance, objetivos y requerimientos iniciales.
<b>Inicialización</b>	Iteración 1	Configuración del ambiente de desarrollo, planificación inicial, planificación de fases, diseño del sistema.
<b>Producción</b>	Iteración 1: Módulo de Detección de objetos	Programación de interfaz gráfica de usuario.
	Iteración 1: Módulo de Gestión de Componentes Visuales	Programación de interfaz gráfica de usuario.
	Iteración 1: Módulo de Gestión de Prototipos	Programación de interfaz gráfica de usuario.
	Iteración 1: Sección de ayuda	Programación de interfaz gráfica de usuario.
	Iteración 1: Inicio de sesión anónima	Programación de interfaz gráfica de usuario.
<b>Estabilización</b>	Iteración 1: Módulo de Detección de objetos	Controles de interfaz gráfica de usuario.
	Iteración 1: Módulo de Gestión de Componentes Visuales	Controles de interfaz gráfica de usuario.
	Iteración 1: Módulo de Gestión de Prototipos	Controles de interfaz gráfica de usuario.
	Iteración 1: Sección de ayuda	Controles de interfaz gráfica de usuario.
	Iteración 1: Inicio de sesión anónima	Controles de interfaz gráfica de usuario.
<b>Pruebas y reparación</b>	Iteración 1: Pruebas del sistema	Ejecución de pruebas y reparación de interfaz gráfica de usuario.
<b>Producción</b>	Iteración 2: Módulo de Detección de objetos	Lógica de negocio y comunicación con la nube.

	Iteración 2: Módulo de Gestión de Componentes Visuales	Lógica de negocios y edición de componentes visuales.
	Iteración 2: Módulo de Gestión de Prototipos	Lógica de negocio y comunicación con la nube.
	Iteración 2: Sección de ayuda	Lógica de negocio y comunicación con la nube
	Iteración 2: Inicio de sesión anónima	Lógica de negocio y comunicación con la nube
<b>Estabilización</b>	Iteración 2: Módulo de Detección de objetos	Controles de lógica de negocio y comunicación con la nube.
	Iteración 2: Módulo de Gestión de Componentes Visuales	Controles de lógica de negocio y comunicación con la nube.
	Iteración 2: Módulo de Gestión de Prototipos	Controles de lógica de negocio y comunicación con la nube.
	Iteración 2: Sección de ayuda	Controles de lógica de negocio y comunicación con la nube.
	Iteración 2: Inicio de sesión anónima	Controles de lógica de negocio y comunicación con la nube.
<b>Pruebas y reparación</b>	Iteración 2: Pruebas del sistema	Ejecución de pruebas y reparación de lógica de negocio y comunicación con la nube.
<b>Estabilización</b>	Iteración 3: Integración de módulos: Detección y Gestión de Componentes Visuales	Integración de funcionalidad entre los módulos.
	Iteración 3: Integración de módulos: Gestión de Prototipos y Gestión de Componentes Visuales	Integración de funcionalidad entre los módulos.
	Iteración 3: Sección de ayuda	Integración de funcionalidad.
	Iteración 3: Inicio de sesión anónima	Integración de funcionalidad.
<b>Pruebas y reparación</b>	Iteración 3: Pruebas del sistema	Ejecución de pruebas y reparación de integración.

## Anexo 8. Estructura JSON de base de datos NoSQL de la aplicación.

```
"user" : {
  "id_user" : {
    "design" : {
      "id_design" : {
        "bottomNav" : true,
        "id" : "id_design",
        "name" : "johanna",
        "tabLayout" : true,
        "toolbar" : true,
        "views" : [
          {
            "height" : 157,
            "id" : 0,
            "nameClass" : "android.widget.ImageView",
            "scaleType" : 0,
            "tagDrawable" : "2131165327",
            "text" : "",
            "textSize" : 0,
            "width" : 112,
            "x" : 5.318656921386719,
            "y" : 48.326210021972656
          },
          ...
        ]
      },
      ...
    },
    ...
  },
  ...
}
```

## Anexo 9. Pruebas de módulos de la aplicación móvil.

TABLA XXXIV. PRUEBA DE MÓDULO: DETECCIÓN DE OBJETOS

Prueba de Módulo: Detección de Objetos
<b>Identificador:</b> PM-001
<b>Objetivo:</b> Detectar componentes del boceto dibujado por el usuario
<b>Pasos:</b> <ol style="list-style-type: none"><li>1. Ir a la pantalla de inicio</li><li>2. Presionar nuevo prototipo<ol style="list-style-type: none"><li>a. La primera vez presionar continuar y aceptar permisos requeridos</li></ol></li><li>3. Presionar capturar foto</li><li>4. Presionar confirmar foto capturada</li><li>5. Esperar respuesta del servidor</li><li>6. Presionar ver resultado (pantalla de inicio)<ol style="list-style-type: none"><li>a. Presionar notificación. En caso que se encuentre fuera de la pantalla de inicio se presentará una notificación cuando la detección finalice, al presionarla lo llevará a la pantalla de inicio</li></ol></li></ol>
<b>Resultado esperado:</b> <ol style="list-style-type: none"><li>1. Se muestra la pantalla de inicio</li><li>2. La cámara del dispositivo muestra su vista previa<ol style="list-style-type: none"><li>a. Permisos requeridos aceptados y guardados</li></ol></li><li>3. Se captura la foto y se visualiza</li><li>4. La foto es enviada al servidor</li><li>5. Se presenta una notificación en la barra del sistema y un widget en la pantalla de inicio indicando que la detección se encuentra en proceso</li><li>6. Se dibujan los cuadros delimitadores de los elementos detectados sobre la foto capturada<ol style="list-style-type: none"><li>a. La pantalla de inicio se muestra</li></ol></li></ol>
<b>Resultado obtenido:</b> <ol style="list-style-type: none"><li>1. Se muestra la pantalla de inicio</li><li>2. La cámara del dispositivo muestra su vista previa<ol style="list-style-type: none"><li>a. Permisos requeridos aceptados y guardados</li></ol></li><li>3. Se captura la foto y se visualiza</li><li>4. La foto es enviada al servidor</li><li>5. Se presenta una notificación en la barra del sistema y un widget en la pantalla de inicio indicando que la detección se encuentra en proceso</li><li>6. Se dibujan los cuadros delimitadores de los elementos detectados sobre la foto capturada<ol style="list-style-type: none"><li>a. La pantalla de inicio se muestra</li></ol></li></ol>

TABLA XXXV. PRUEBA DE MÓDULO: GESTIÓN DE COMPONENTES VISUALES

Prueba de Módulo: Gestión de Componentes Visuales
<b>Identificador:</b> PM-002
<b>Objetivo:</b> Modificar los componentes del prototipo con las opciones disponibles
<p><b>Pasos:</b></p> <ol style="list-style-type: none"> <li>1. Abrir un diseño o aceptar respuesta de detección</li> <li>2. Seleccionar un componente               <ol style="list-style-type: none"> <li>a. Deslizar cambio de altura</li> <li>b. Deslizar cambio de anchura</li> <li>c. Presionar cambio de contenido                   <ol style="list-style-type: none"> <li>i. Cambiar tamaño o texto o ajustar tamaño del componente al contenido (cuando el componente extiende de la clase TextView) al presionar aplicar</li> <li>ii. Cambiar imagen (cuando el componente extiende de la clase ImageView) al presionar la imagen deseada</li> </ol> </li> <li>d. Presionar copia de componente</li> <li>e. Presionar prolongadamente eliminar componente</li> <li>f. Presionar añadir / quitar barras adicionales                   <ol style="list-style-type: none"> <li>i. Seleccionar la barra deseada, presionar aplicar</li> </ol> </li> <li>g. Presionar ver prototipo                   <ol style="list-style-type: none"> <li>i. Presionar botón volumen arriba para regresar</li> <li>ii. Presionar botón volumen abajo para compartir captura de pantalla</li> </ol> </li> </ol> </li> </ol>
<p><b>Resultado esperado:</b></p> <ol style="list-style-type: none"> <li>1. Se presenta pantalla de diseño de prototipo junto con sus componentes</li> <li>2. El componente cambia de color de fondo               <ol style="list-style-type: none"> <li>a. El componente cambia su altura</li> <li>b. El componente cambia su anchura</li> <li>c. Se presenta un dialogo para cambiar el contenido                   <ol style="list-style-type: none"> <li>i. Se cambia el tamaño o texto o ajusta el tamaño del componente al contenido (cuando el componente extiende de la clase TextView)</li> <li>ii. Se cambia la imagen (cuando el componente extiende de la clase ImageView)</li> </ol> </li> <li>d. El componente de duplica</li> <li>e. El componente se elimina</li> <li>f. Se presenta un diálogo para activar o desactivar barras adicionales                   <ol style="list-style-type: none"> <li>i. Se activa o desactiva la barra seleccionada</li> </ol> </li> <li>g. Se oculta la barra de opciones de diseño                   <ol style="list-style-type: none"> <li>i. Se muestra la barra de opciones de diseño</li> <li>ii. Se abre un diálogo con las aplicaciones disponibles para compartir la captura del prototipo</li> </ol> </li> </ol> </li> </ol>
<p><b>Resultado obtenido:</b></p> <ol style="list-style-type: none"> <li>1. Se presenta pantalla de diseño de prototipo junto con sus componentes</li> <li>2. El componente cambia de color de fondo               <ol style="list-style-type: none"> <li>a. El componente cambia su altura</li> <li>b. El componente cambia su anchura</li> <li>c. Se presenta un dialogo para cambiar el contenido                   <ol style="list-style-type: none"> <li>i. Se cambia el tamaño o texto o ajusta el tamaño del componente al contenido (cuando el componente extiende de la clase TextView)</li> </ol> </li> </ol> </li> </ol>

<ul style="list-style-type: none"> <li>ii. Se cambia la imagen (cuando el componente extiende de la clase ImageView)</li> <li>d. El componente de duplica</li> <li>e. El componente se elimina</li> <li>f. Se presenta un diálogo para activar o desactivar barras adicionales <ul style="list-style-type: none"> <li>i. Se activa o desactiva la barra seleccionada</li> </ul> </li> <li>g. Se oculta la barra de opciones de diseño <ul style="list-style-type: none"> <li>i. Se muestra la barra de opciones de diseño</li> <li>ii. Se abre un diálogo con las aplicaciones disponibles para compartir la captura del prototipo</li> </ul> </li> </ul>
---

TABLA XXXVI. PRUEBA DE MÓDULO: GESTIÓN DE PROTOTIPOS

Prueba de Módulo: Gestión de Prototipos
<b>Identificador:</b> PM-003
<b>Objetivo:</b> Guardar, eliminar, editar y visualizar prototipos
<p><b>Pasos:</b></p> <p><b>Guardar prototipo</b></p> <ol style="list-style-type: none"> <li>1. Abrir un diseño o aceptar la respuesta de detección</li> <li>2. Presionar guardar <ol style="list-style-type: none"> <li>a. Asignar nombre al prototipo, presionar guardar prototipo (cuando abre desde la detección)</li> </ol> </li> </ol> <p><b>Eliminar prototipo</b></p> <ol style="list-style-type: none"> <li>1. Ir a pantalla de inicio</li> <li>2. Presionar eliminar</li> <li>3. Confirmar eliminar prototipo</li> </ol> <p><b>Editar prototipo</b></p> <ol style="list-style-type: none"> <li>1. Abrir un diseño</li> <li>2. Gestionar los componentes</li> </ol> <p><b>Visualizar prototipos</b></p> <ol style="list-style-type: none"> <li>1. Ir a la pantalla de inicio</li> </ol>
<p><b>Resultado esperado:</b></p> <p><b>Guardar prototipo</b></p> <ol style="list-style-type: none"> <li>1. Se presenta la pantalla de diseño con el prototipo y sus componentes</li> <li>2. Se presenta un mensaje indicando que el prototipo se ha guardado. <ol style="list-style-type: none"> <li>a. El prototipo se guarda con el nombre asignado</li> </ol> </li> </ol> <p><b>Eliminar prototipo</b></p> <ol style="list-style-type: none"> <li>1. Se presenta la pantalla de inicio</li> <li>2. Se presenta un dialogo de confirmación</li> <li>3. Eliminar prototipo de la lista</li> </ol> <p><b>Editar prototipo</b></p> <ol style="list-style-type: none"> <li>1. Se presenta la pantalla de diseño con el prototipo y sus componentes</li> <li>2. Se gestionan sus componentes</li> </ol> <p><b>Visualizar prototipos</b></p>

1. Se presenta la pantalla de inicio junto con la lista de prototipos del usuario
<b>Resultado obtenido:</b> <b>Guardar prototipo</b> <ol style="list-style-type: none"> <li>1. Se presenta la pantalla de diseño con el prototipo y sus componentes</li> <li>2. Se presenta un mensaje indicando que el prototipo se ha guardado. <ol style="list-style-type: none"> <li>a. El prototipo se guarda con el nombre asignado</li> </ol> </li> </ol> <b>Eliminar prototipo</b> <ol style="list-style-type: none"> <li>1. Se presenta la pantalla de inicio</li> <li>2. Se presenta un dialogo de confirmación</li> <li>3. Eliminar prototipo de la lista</li> </ol> <b>Editar prototipo</b> <ol style="list-style-type: none"> <li>1. Se presenta la pantalla de diseño con el prototipo y sus componentes</li> <li>2. Se gestionan sus componentes</li> </ol> <b>Visualizar prototipos</b> <ol style="list-style-type: none"> <li>1. Se presenta la pantalla de inicio junto con la lista de prototipos del usuario</li> </ol>

TABLA XXXVII. PRUEBA DE MÓDULO: SECCIÓN DE AYUDA

Prueba de Módulo: Sección de Ayuda
<b>Identificador:</b> PM-004
<b>Objetivo:</b> Visualizar ayuda de aplicación
<b>Pasos:</b> <ol style="list-style-type: none"> <li>1. Ir a pantalla de inicio</li> <li>2. Presionar ayuda</li> <li>3. Deslizar entre diferentes secciones de ayuda</li> </ol>
<b>Resultado esperado:</b> <ol style="list-style-type: none"> <li>1. Se presenta pantalla de inicio</li> <li>2. Se presenta pantalla de ayuda</li> <li>3. Se presentan diferentes secciones de ayuda</li> </ol>
<b>Resultado obtenido:</b> <ol style="list-style-type: none"> <li>1. Se presenta pantalla de inicio</li> <li>2. Se presenta pantalla de ayuda</li> <li>3. Se presentan diferentes secciones de ayuda</li> </ol>

TABLA XXXVIII. PRUEBA DE MÓDULO: INICIO DE SESIÓN ANÓNIMA

Prueba de Módulo: Inicio de Sesión Anónima
<b>Identificador:</b> PM-005
<b>Objetivo:</b> Inicio de sesión de manera automática, identificador autogenerado
<b>Pasos:</b> <ol style="list-style-type: none"> <li>1. Iniciar aplicación</li> <li>2. Esperar inicio de sesión automático</li> </ol>
<b>Resultado esperado:</b> <ol style="list-style-type: none"> <li>1. Se inicia la aplicación</li> <li>2. Se presenta la pantalla de inicio</li> </ol>
<b>Resultado obtenido:</b> <ol style="list-style-type: none"> <li>1. Se inicia la aplicación</li> <li>2. Se presenta la pantalla de inicio</li> </ol>

## Anexo 10. Pruebas de integración de módulos de la aplicación móvil.

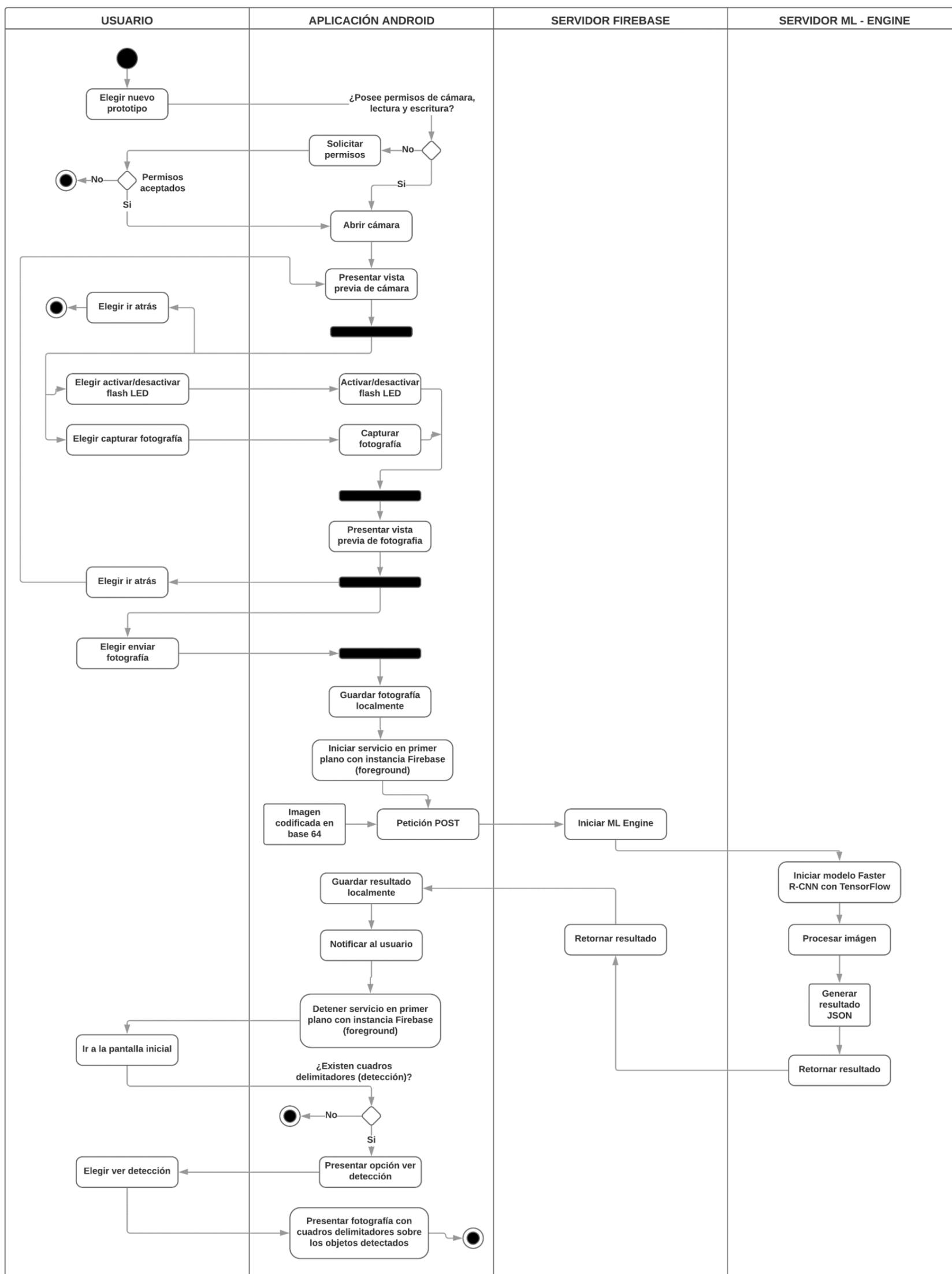
TABLA XXXIX. PRUEBA DE INTEGRACIÓN DE MÓDULOS: DETECCIÓN Y GESTIÓN DE COMPONENTES VISUALES

Prueba de Integración de Módulos: Detección y Gestión de Componentes Visuales
<b>Identificador:</b> PIM-001
<b>Objetivo:</b> Convertir el resultado de detección, en componentes Material Design
<b>Información esperada:</b> <ul style="list-style-type: none"><li>• Objeto Detección<ul style="list-style-type: none"><li>○ Clases: Nombre de componentes detectados</li><li>○ Cajas: Ubicación de componentes detectados</li></ul></li></ul>
<b>Información generada:</b> <ul style="list-style-type: none"><li>• Objeto Detección<ul style="list-style-type: none"><li>○ Clases: Nombre de componentes detectados</li><li>○ Cajas: Ubicación de componentes detectados</li></ul></li></ul>

TABLA XL. PRUEBA DE INTEGRACIÓN DE MÓDULOS: GESTIÓN DE PROTOTIPO Y GESTIÓN DE COMPONENTES VISUALES

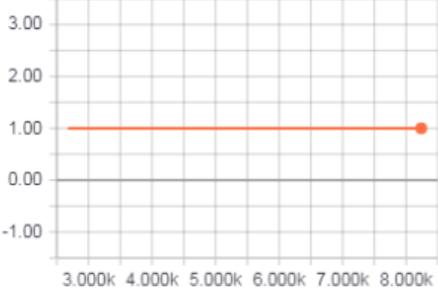
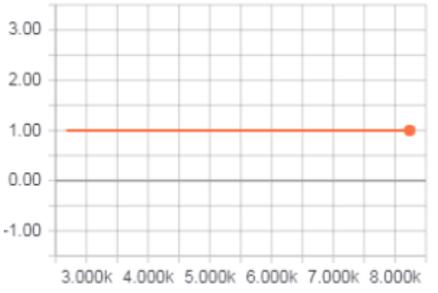
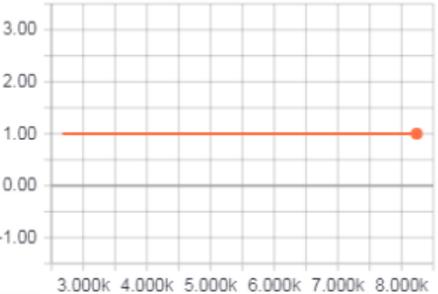
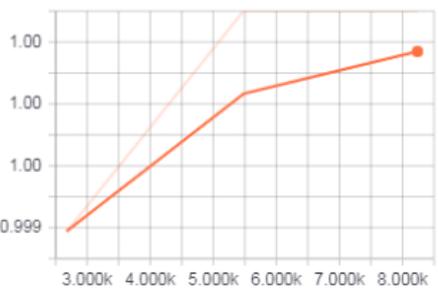
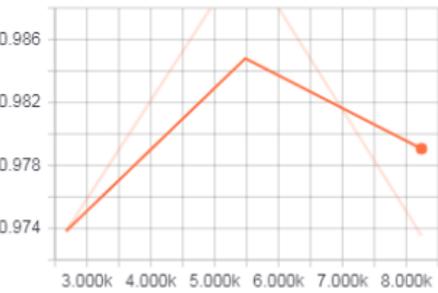
Prueba de Integración de Módulos: Gestión de Prototipo y Gestión de Componentes Visuales
<b>Identificador:</b> PIM-002
<b>Objetivo:</b> Presentar el prototipo seleccionado para su edición
<b>Información esperada:</b> <ul style="list-style-type: none"><li>• Objeto Prototipo<ul style="list-style-type: none"><li>○ Componentes: Tamaño, ubicación, texto, clase.</li><li>○ Información: Identificador, nombre.</li></ul></li></ul>
<b>Información generada:</b> <ul style="list-style-type: none"><li>• Objeto Prototipo<ul style="list-style-type: none"><li>○ Componentes: Tamaño, ubicación, texto, clase.</li><li>○ Información: Identificador, nombre.</li></ul></li></ul>

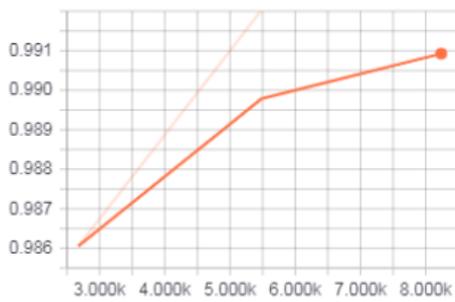
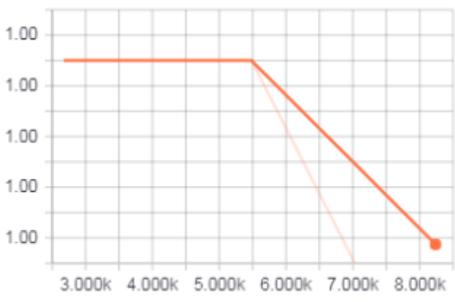
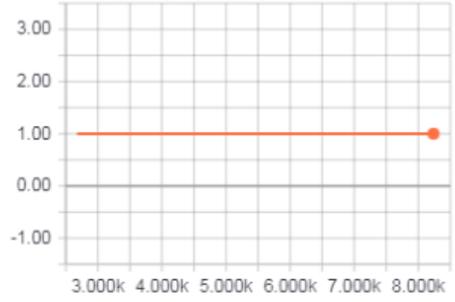
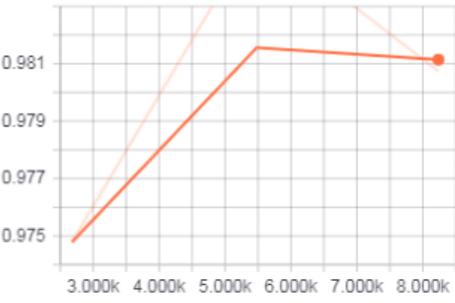
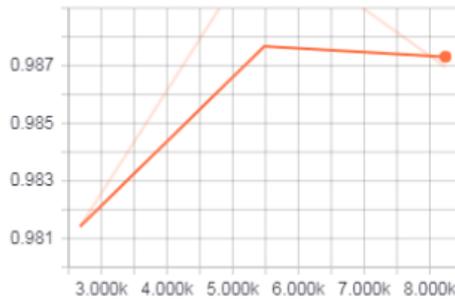
Anexo 11. Diagrama de procesos del módulo de Detección de Objetos.



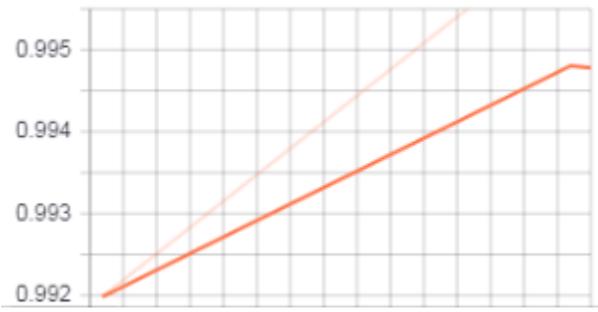
## Anexo 12. Evaluación del modelo de detección con TensorBoard

TABLA XLI. EVALUACIÓN DEL MODELO CON TENSORBOARD

Clase	Resultado								
Floatin Action Button	 <p>The graph shows a constant value of 1.00 across the entire range from 3.000k to 8.000k. The y-axis ranges from -1.00 to 3.00.</p> <table border="1"> <thead> <tr> <th>Iteration (k)</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>3.000k</td> <td>1.00</td> </tr> <tr> <td>8.000k</td> <td>1.00</td> </tr> </tbody> </table>	Iteration (k)	Value	3.000k	1.00	8.000k	1.00		
Iteration (k)	Value								
3.000k	1.00								
8.000k	1.00								
Image View	 <p>The graph shows a constant value of 1.00 across the entire range from 3.000k to 8.000k. The y-axis ranges from -1.00 to 3.00.</p> <table border="1"> <thead> <tr> <th>Iteration (k)</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>3.000k</td> <td>1.00</td> </tr> <tr> <td>8.000k</td> <td>1.00</td> </tr> </tbody> </table>	Iteration (k)	Value	3.000k	1.00	8.000k	1.00		
Iteration (k)	Value								
3.000k	1.00								
8.000k	1.00								
Recycler View	 <p>The graph shows a constant value of 1.00 across the entire range from 3.000k to 8.000k. The y-axis ranges from -1.00 to 3.00.</p> <table border="1"> <thead> <tr> <th>Iteration (k)</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>3.000k</td> <td>1.00</td> </tr> <tr> <td>8.000k</td> <td>1.00</td> </tr> </tbody> </table>	Iteration (k)	Value	3.000k	1.00	8.000k	1.00		
Iteration (k)	Value								
3.000k	1.00								
8.000k	1.00								
Radio Buton	 <p>The graph shows an increasing trend from 0.999 at 3.000k to approximately 1.005 at 8.000k. The y-axis ranges from 0.999 to 1.00.</p> <table border="1"> <thead> <tr> <th>Iteration (k)</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>3.000k</td> <td>0.999</td> </tr> <tr> <td>8.000k</td> <td>~1.005</td> </tr> </tbody> </table>	Iteration (k)	Value	3.000k	0.999	8.000k	~1.005		
Iteration (k)	Value								
3.000k	0.999								
8.000k	~1.005								
Switch	 <p>The graph shows a peak at 5.000k (value ~0.986) and a decline thereafter, ending at approximately 0.979 at 8.000k. The y-axis ranges from 0.974 to 0.986.</p> <table border="1"> <thead> <tr> <th>Iteration (k)</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>3.000k</td> <td>~0.974</td> </tr> <tr> <td>5.000k</td> <td>~0.986</td> </tr> <tr> <td>8.000k</td> <td>~0.979</td> </tr> </tbody> </table>	Iteration (k)	Value	3.000k	~0.974	5.000k	~0.986	8.000k	~0.979
Iteration (k)	Value								
3.000k	~0.974								
5.000k	~0.986								
8.000k	~0.979								

<p><b>Text View</b></p>	 <p>A line graph with a grid. The x-axis is labeled from 3.000k to 8.000k in increments of 1.000k. The y-axis is labeled from 0.986 to 0.991 in increments of 0.001. A solid orange line starts at (3.000k, 0.986), rises to (5.500k, 0.9895), and then continues to (8.000k, 0.991). A dashed orange line shows a linear trend from (3.000k, 0.986) to (5.500k, 0.991).</p>
<p><b>Button</b></p>	 <p>A line graph with a grid. The x-axis is labeled from 3.000k to 8.000k in increments of 1.000k. The y-axis is labeled from 1.00 to 1.00 in increments of 0.00. A solid orange line starts at (3.000k, 1.00), remains constant until 5.500k, then drops to approximately 0.985 at 8.000k. A dashed orange line shows a linear trend from (3.000k, 1.00) to (7.000k, 0.985).</p>
<p><b>Chip</b></p>	 <p>A line graph with a grid. The x-axis is labeled from 3.000k to 8.000k in increments of 1.000k. The y-axis is labeled from -1.00 to 3.00 in increments of 1.00. A solid orange line is constant at y = 1.00 across the entire x-axis range.</p>
<p><b>Edit Text</b></p>	 <p>A line graph with a grid. The x-axis is labeled from 3.000k to 8.000k in increments of 1.000k. The y-axis is labeled from 0.975 to 0.981 in increments of 0.002. A solid orange line starts at (3.000k, 0.975), rises to (6.000k, 0.981), and then slightly decreases to (8.000k, 0.980). A dashed orange line shows a linear trend from (3.000k, 0.975) to (6.000k, 0.981).</p>
<p><b>Check Box</b></p>	 <p>A line graph with a grid. The x-axis is labeled from 3.000k to 8.000k in increments of 1.000k. The y-axis is labeled from 0.981 to 0.987 in increments of 0.002. A solid orange line starts at (3.000k, 0.981), rises to (6.000k, 0.987), and then slightly decreases to (8.000k, 0.986). A dashed orange line shows a linear trend from (3.000k, 0.981) to (6.000k, 0.987).</p>

**General**



**Anexo 13. Estudiantes de 6to "A" de la carrera de Ingeniería en Sistemas de la Universidad Nacional de Loja**

NOMBRE	CÉDULA	FIRMA
Jean Carlos Alarcón O.	1105026114	
Anthony Daniel Ortega Malabando	1106081937	
Byron Bolivar Gonzalez Cueva	1150269494	
GERARDO RAMIREZ MOSIN	0704 062107	
Erick Jara Vega	070488174-7	
Ricardo Javier Espinosa Torres	0706735446	
Brayan Alejandro Cobos Cabrera	1105399065	
Jean Carlos León Mejía	1150846978	
Derwin Alexander Rogel Rivera	0707078465	
Karen Nathali Navea Gonzalez	1105209397	
Gaston Rene Chamba R.	1104308422	



LA RECOLECCIÓN DE DATOS A ESTUDIANTES DETALLADA A CONTINUACIÓN ES DE CARÁCTER ACADÉMICO, CON EL ÚNICO FIN DE VALIDAR LA PRECENCIA EN LA EVALUACIÓN REALIZADA AL PRESENTE TRABAJO DE TITULACIÓN.

NOMBRE	CÉDULA	FIRMA
Samantha González	1105663247	
BRAYAN COLLACUAZO	1105993065	
Sisa Nusta Para Tene	1105194367	
Pavel Rivadeneira Cajus	1724002009	
Diego Merino Volante	1150007873	
Alex John Chumba Mucos	1105643314	
Alexandria Patricia López Romero	115044055-8	
Johnney Javier Bermeo Amijos	1104119860	
Byron Ronaldo Jimenez Borja	1600474868	
Edgar Andrés Soto R.	2150709304	
Robin Lenin Córdoba Alvarado	210091617-6	Córdoba Alvarado
Franklin Stalin Mora Pradoza	0704433754	
Carlos Israel Pucha Castillo	1150106308	



## Anexo 14. Modelo de Evaluación de Proyecto

Encuesta de evaluación de proyecto de titulación:

### APLICACIÓN MÓVIL PARA EL PROTOTIPADO DE INTERFAZ GRÁFICA DE USUARIO EN LA PLATAFORMA ANDROID MEDIANTE EL RECONOCIMIENTO DE BOCETOS UTILIZANDO VISIÓN ARTIFICIAL

- ¿El uso de la aplicación le facilitó o agilizó al crear un prototipo?

SI

NO

¿Por qué?

---

- ¿Utilizaría la aplicación para crear prototipos de interfaz de usuario en sus proyectos?

SI

NO

¿Por qué?

---

- La detección de componentes de la aplicación fue:

Excelente

Buena

Mala

- ¿Encontró problemas en el funcionamiento de la aplicación?

SI

NO

En caso de responder SI, ¿Qué problemas encontró?

---

- ¿Tiene alguna sugerencia o recomendación para mejorar la aplicación?
-

## Anexo 15. Manual del programador

### ÍNDICE DE CONTENIDO

<b>ÍNDICE DE CONTENIDO</b> .....	95
<b>ÍNDICE DE FIGURAS</b> .....	95
1. INTRODUCCIÓN.....	96
2. SERVIDORES .....	96
2.1. Configuración de Firebase.....	96
2.1.1. Firebase Database .....	97
2.1.2. Firebase Cloud Functions .....	97
2.2. Configuración de GCP ML – Engine .....	98
3. APLICACIÓN MÓVIL.....	100
3.1. Configuración de Android Studio .....	100
3.2. Configuración de librerías, dependencias y permisos .....	100
3.3. Preparación de arquitecturas .....	101
3.4. Modelo – Vista – Presentador (MVP).....	102
3.5. Clean .....	102
3.6. Desarrollo y evaluación.....	103
3.6.1. Desarrollo .....	103
3.6.2. Evaluación.....	104
3.7. Publicación en Play Store .....	104

### ÍNDICE DE FIGURAS

Figura 1. Pantalla principal del proyecto en Firebase.....	96
Figura 2. Pantalla de configuración de Database.....	97
Figura 3. Pantalla de configuración de Cloud Functions .....	98
Figura 4. Pantalla principal del proyecto GCP.....	98
Figura 5. Pantalla de gestión de modelos .....	99
Figura 6. Implementación de dependencias en Gradle app .....	101
Figura 7. Permisos agregados en AndroidManifest.xml .....	101
Figura 8. Estructura MVP.....	102
Figura 9. Estructura Clean .....	103
Figura 10. Aplicación Protoia publicada en Play Store .....	105

## 1. INTRODUCCIÓN

El presente manual de programador tiene como objetivo detallar la configuración, estructura y funciones importantes de la aplicación móvil Android. Así también la configuración de servidores de base de datos, funciones y detección de objetos.

## 2. SERVIDORES

En el lado de servidores se han utilizado dos servicios, Firebase para la gestión de base de datos y funciones y GCP ML – Engine para el despliegue del modelo y detección de objetos. A continuación, se procede a explicar la configuración de estas herramientas.

### 2.1. Configuración de Firebase

En este proyecto se utiliza Firebase como servidor de base de datos NoSQL en formato JSON. También es usado como servidor de funciones bajo JavaScript y Node.js 6.0 con el fin de llamar al servicio de ML – Engine.

En primer lugar, se debe registrar y crear un proyecto para dispositivos móviles en el sitio oficial <https://console.firebase.google.com/>, para este proyecto se utilizó la versión gratuita de Firebase “Plan Spark”, para conocer la infraestructura ofrecida en este plan, revisar la siguiente documentación oficial <https://firebase.google.com/pricing/?hl=es-419>. Como se observa en la figura 1, la pantalla de inicio a la izquierda se encuentran varias herramientas, entre ellas Database y Functions son las utilizadas en este proyecto.

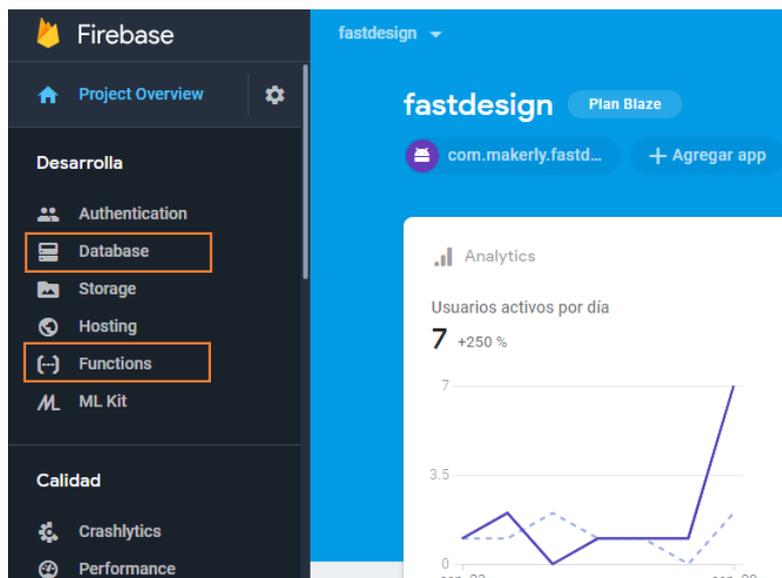


Figura 30. Pantalla principal del proyecto en Firebase

Adicionalmente al crear el proyecto se generará un archivo de configuraciones de Firebase, que tendremos que pegarlo dentro del proyecto Android. Este archivo contiene configuraciones de seguridad e identificación de nuestro proyecto móvil necesarias para el servidor Firebase. Para mayor información sobre Firebase, revisar la siguiente documentación oficial <https://firebase.google.com/docs/?hl=es-419>.

### 2.1.1. Firebase Database

- En el apartado Database podemos configurar la base de datos, las reglas de seguridad de acceso a ella, realizar exportaciones e importaciones en formato JSON y monitorear su actividad (figura 2).
- Para mayor información sobre Firebase Database, revisar la siguiente documentación oficial <https://firebase.google.com/docs/database/?hl=es-419>.

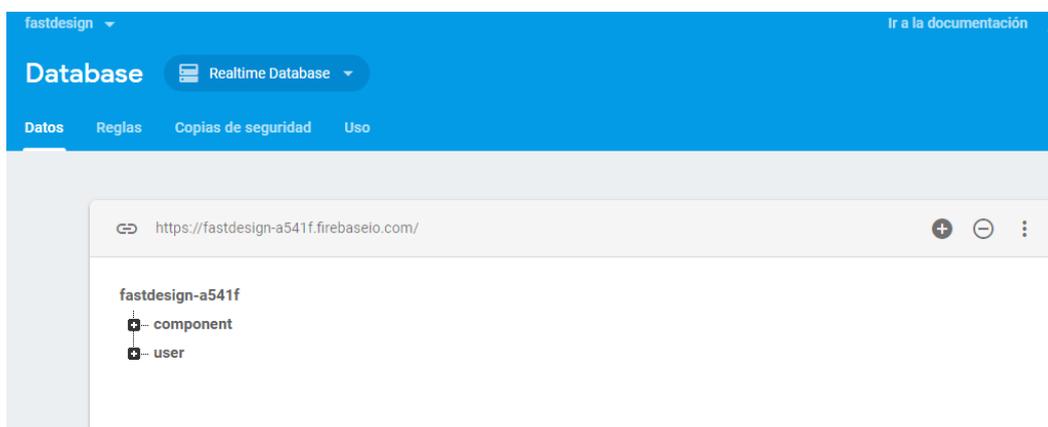
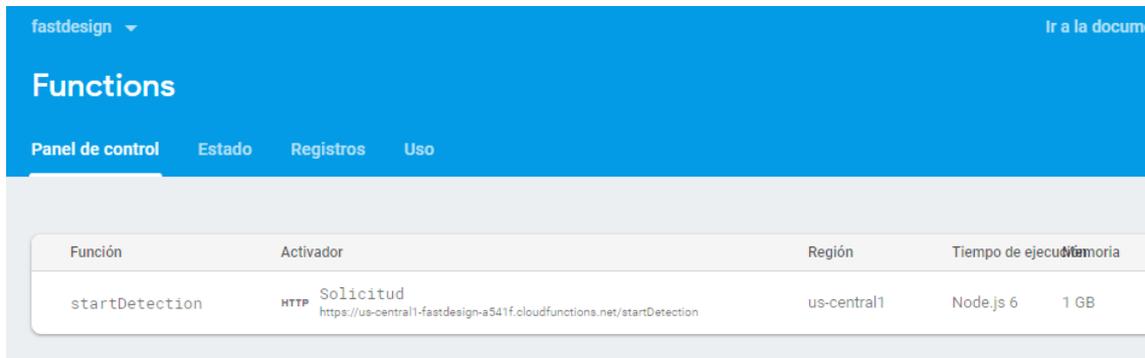


Figura 31. Pantalla de configuración de Database

### 2.1.2. Firebase Cloud Functions

- En el apartado Cloud Functions podemos ver detalles acerca de las funciones que se han implementado, como su estado, memoria asignada, tiempo de espera de respuesta, monitoreo de actividad, entre otras (figura 3).
- Para poder desplegar funciones se debe instalar Node.js 6.0 u 8.0 en el ordenador de trabajo, luego se procede a la instalación del SDK de Firebase y Firebase CLI para poder evaluar y desplegar las funciones desde el computador a la nube.
- Estas funciones pueden ser llamadas a través del SDK de Firebase o alguna herramienta que permita el consumo de API REST, ya que Cloud Functions nos permite generar una ruta de acceso https a dicha función.

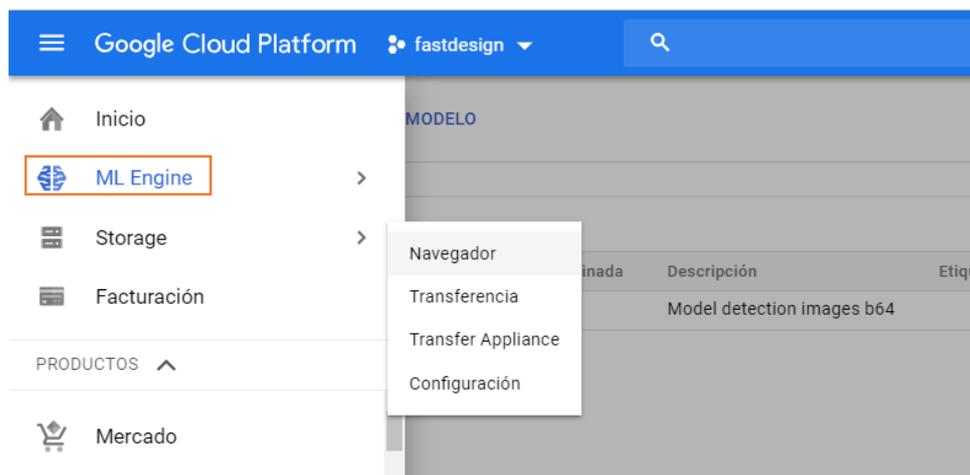
- Para mayor información sobre Firebase Cloud Functions, revisar la siguiente documentación oficial <https://firebase.google.com/docs/functions/?hl=es-419>.



**Figura 32. Pantalla de configuración de Cloud Functions**

## 2.2. Configuración de GCP ML – Engine

Para realizar la detección de objetos, se utiliza el servicio ML – Engine, que permite subir modelos entrenados con TensorFlow para su consumo en evaluación o producción. Para poder acceder a este servicio se debe registrar en Google Cloud Platform <https://cloud.google.com/> y crear un nuevo proyecto. En la figura 4, se observan varias herramientas, para este proyecto se utilizó ML – Engine.

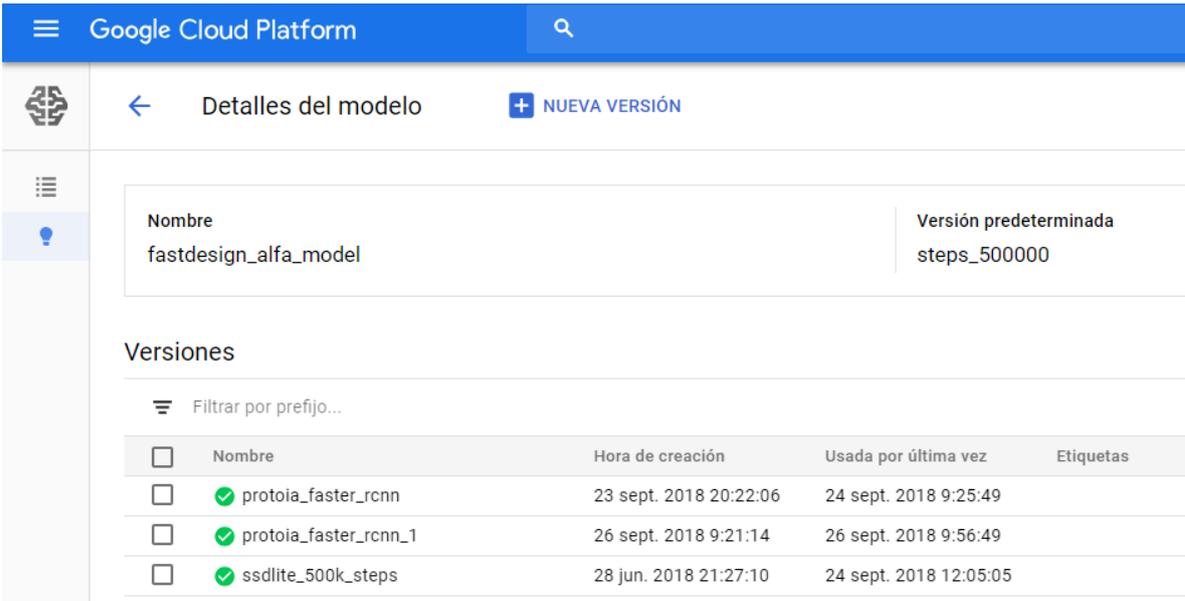


**Figura 33. Pantalla principal del proyecto GCP**

Este servicio posee un precio de \$0,06 por predicción en línea, es decir, por cada vez que la función sea ejecutada. Sin embargo, Google nos facilita un crédito de \$300 durante un año. El computador donde corre este servicio está compuesto por un procesador de 1 núcleo y 4 GB de RAM, por ahora el hardware que realiza la predicción

en línea no puede ser configurable. Para mayor información sobre este servicio, revisar la siguiente documentación oficial <https://cloud.google.com/ml-engine/>.

Para su uso, se debe crear un nuevo modelo, luego una versión y subir el modelo entrenado (figura 5). Para poder consumir este modelo, es necesario utilizar GoogleApis, la cual es llamada y gestionada en la función desarrollada y desplegada por Firebase Functions.



The screenshot shows the Google Cloud Platform interface for managing ML Engine models. The top navigation bar includes the Google Cloud Platform logo and a search icon. The main content area is titled 'Detalles del modelo' and features a '+ NUEVA VERSIÓN' button. Below this, the model name 'fastdesign\_alfa\_model' is displayed alongside its 'Versión predeterminada' (steps\_500000). A section titled 'Versiones' contains a filter 'Filtrar por prefijo...' and a table listing three model versions with their names, creation times, and last usage dates.

<input type="checkbox"/>	Nombre	Hora de creación	Usada por última vez	Etiquetas
<input type="checkbox"/>	✔ protoia_faster_rcnn	23 sept. 2018 20:22:06	24 sept. 2018 9:25:49	
<input type="checkbox"/>	✔ protoia_faster_rcnn_1	26 sept. 2018 9:21:14	26 sept. 2018 9:56:49	
<input type="checkbox"/>	✔ ssdlite_500k_steps	28 jun. 2018 21:27:10	24 sept. 2018 12:05:05	

**Figura 34. Pantalla de gestión de modelos**

GoogleApis es un conjunto de herramientas e interfaces que permiten la gestión entre los servicios de Google Cloud Platform con otros servidores. Para mayor información sobre esta herramienta, revisar la siguiente documentación oficial <https://cloud.google.com/ml-engine/reference/>.

### **3. APLICACIÓN MÓVIL**

La aplicación móvil está desarrollada en el lenguaje de programación Kotlin usando el IDE Android Studio, bajo un conjunto de arquitecturas escalables que permiten un desarrollo con buenas prácticas, limpio y entendible. El proyecto se encuentra almacenado en el sistema de control de versiones Bitbucket, con el fin de llevar un control en el desarrollo. Se puede clonar desde el siguiente enlace <https://Yavac@bitbucket.org/Yavac/protoia.git>.

A continuación, se explicará la configuración de las herramientas necesarias y usadas en este proyecto.

#### **3.1. Configuración de Android Studio**

En este proyecto se utilizaron herramientas con las siguientes versiones:

- Android Studio 3.1.3
- SDK Android con Api 21 o superior
- Java 8
- Kotlin 1.2.3
- Control de versiones Bitbucket - Git

El proyecto se creó para versiones de Android 5.0 o Api 21 o superior, con el fin de acceder la mayor parte de componentes y funciones disponibles en Material Design para Android. La aplicación debe ejecutarse en smartphones con las siguientes características:

- Procesador ARM doble núcleo a 1 GHZ o superior
- Memoria RAM 1.5GB o superior
- Cámara de 1 megapíxeles o superior
- Conexión estable a internet
- Android 5.0 p superior

#### **3.2. Configuración de librerías, dependencias y permisos**

##### **3.2.1. Librerías**

Las librerías usadas como SDK de Firebase, Retrofit, Glide, Dagger2, entre otras, son utilizadas para la gestión de información entre la aplicación y los servidores, gestión de interfaz gráfica, arquitectura, inyección de dependencias, etc. Las librerías junto a sus

versiones se especifican en el archivo Gradle de la aplicación, para ser descargadas e instaladas automáticamente (figura 6).

```
dependencies {
    implementation fileTree(include: ['*.jar'], dir: 'libs')
    implementation "org.jetbrains.kotlin:kotlin-stdlib-jre7:$kotlin_version"
    implementation 'com.android.support:appcompat-v7:28.0.0-rc01'
    implementation 'com.android.support:design:28.0.0-rc01'
    implementation 'com.android.support:support-v4:28.0.0-rc01'
    implementation 'com.android.support.constraint:constraint-layout:1.1.2'
    testImplementation 'junit:junit:4.12'
    implementation 'com.squareup.retrofit2:retrofit:2.4.0'
    implementation 'com.squareup.retrofit2:converter-gson:2.4.0'
    implementation 'com.squareup.retrofit2:converter-scalars:2.4.0'
    implementation 'com.squareup.okhttp3:okhttp:3.11.0'
    kapt 'com.google.dagger:dagger-compiler:2.13'
    implementation 'com.google.dagger:dagger:2.13'
    compileOnly 'javax.annotation:jsr250-api:1.0'
    implementation 'com.google.firebase:firebase-core:16.0.1'
    implementation 'com.google.firebase:firebase-storage:16.0.1'
    implementation 'com.google.firebase:firebase-database:16.0.1'
    implementation 'com.google.firebase:firebase-auth:16.0.3'
    implementation 'com.github.bumptech.glide:glide:4.8.0'
    kapt 'com.github.bumptech.glide:compiler:4.8.0'
    androidTestImplementation 'com.android.support.test:runner:1.0.2'
    androidTestImplementation 'com.android.support.test.espresso:espresso-core:3.0.2'
    implementation 'org.greenrobot:eventbus:3.1.1'
    implementation project(':camera')
}
```

**Figura 35. Implementación de dependencias en Gradle app**

### 3.2.2. Permisos

Para un correcto funcionamiento de la aplicación, es necesario añadir varios tipos de permisos (figura 7), con el fin de obtener acceso a ciertas características y evitar errores o caídas en la aplicación a falta de estos permisos. Para versiones de Android 5.0+, los permisos son programados y mostrados al usuario en tiempo de ejecución.

```
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.CAMERA" />
<uses-permission android:name="android.permission.FLASHLIGHT" />
<uses-permission android:name="android.permission.VIBRATE" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />
```

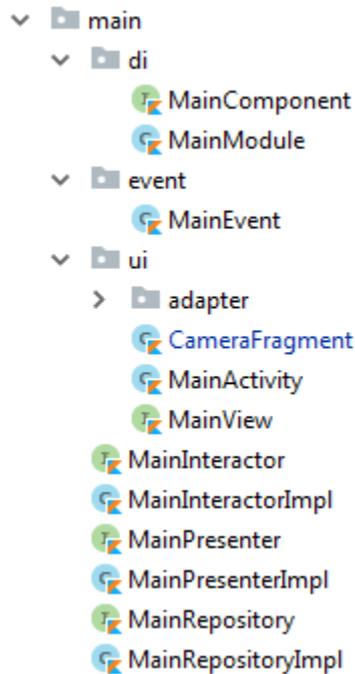
**Figura 36. Permisos agregados en AndroidManifest.xml**

## 3.3. Preparación de arquitecturas

A continuación, se describen las arquitecturas utilizadas en el desarrollo y estructuración de la aplicación.

### 3.4. Modelo – Vista – Presentador (MVP)

El patrón MVP permite la separación de la lógica de interfaz de usuario y lógica de negocios, esto permite una codificación organizada. La figura 8, muestra la estructura aplicada a la pantalla principal de la aplicación.



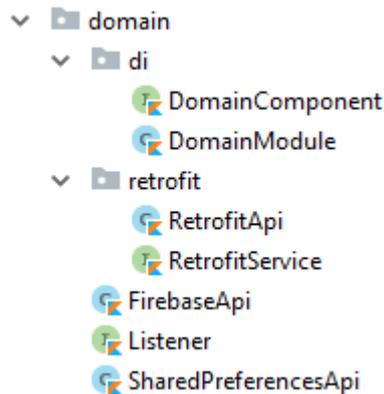
**Figura 37. Estructura MVP**

También podemos observar en la figura 8, una carpeta llamada “di”, esta carpeta contiene la clase e interfaz para la generación de inyección de dependencias para esta sección de la aplicación. La inyección de dependencias se realiza con Dagger2 implementada en el archivo gradle app, y permite “inyectar” o introducir objetos o instancias de otras clases, con el fin de lograr el desacoplamiento entre clases y que estas posean “libertad” o independencia, es decir que no dependan de otras, además esto facilita el testeado unitario de clases.

Esta configuración se encuentra implementada para cada pantalla de la aplicación, con el fin de mantener un ambiente agradable, escalable y mantenible en cuanto al desarrollo de la aplicación.

### 3.5. Clean

La arquitectura Clean permite dividir o desacoplar la lógica de negocios de la lógica de servicios o servidores, permitiendo usarla en otras secciones del proyecto de manera fácil, rápida. La figura 9, muestra la estructura de la arquitectura.



**Figura 38. Estructura Clean**

Aquí se encuentra las rutas y configuraciones de servicios de base de datos y servidores, además posee la configuración para ser inyectada como dependencia. Como se observa en la figura 9, Firebase, Retrofit, y Shared Preferences se encuentran configurados para el consumo de datos a los servidores y almacenamiento local respectivamente.

## 3.6. Desarrollo y evaluación

### 3.6.1. Desarrollo

La estructura de directorios maneja el estándar por defecto de Android Studio, sin embargo, se añadieron nuevos directorios los cuales se detallan a continuación:

- **entity.** Contiene las entidades o modelos de datos de la lógica de negocios de la aplicación.
- **lib.** Contiene la configuración básica de EventBus, que permite implementar un bus o flujo de comunicación de datos entre clases.
- **service.** Contiene la arquitectura Clean del servicio utilizado para mantener en procesamiento de primer plano la petición al servidor de detección de objetos.
- **útil.** Contiene funciones o métodos de carácter genéricos, que pueden ser usados en la mayoría de la aplicación con el fin de disminuir código repetitivo.
- **App, AppModule.** Estas clases son utilizadas para la inicialización e instanciación de varios servicios y configuraciones disponibles para toda la aplicación, como es el caso de Firebase e inyección de dependencias.

Se deben tener en cuenta los siguientes puntos importantes referente a la codificación, estos son:

- Para la integración de la cámara se utilizó una librería desarrollada por Google, que facilita la manipulación básica de la misma, el repositorio oficial se encuentra en <https://github.com/google/cameraview>.
- Para lograr la integración correcta entre la aplicación y el servidor de Firebase, se debe pegar el archivo generado cuando creamos el proyecto en el servidor. Este archivo de configuraciones de identificación y seguridad de Firebase se encuentra en el directorio "protoia\app" con el nombre "google-services.json".
- Debido a que el servidor de detección de objetos tarda varios segundos en completarse, fue necesario cambiar el tiempo de espera de la función a 60 segundos, esto se hizo en la configuración de Retrofit en el directorio "domain".

### 3.6.2. Evaluación

Para la evaluación o testeo de la aplicación se utilizó Firebase Test Lab, que es un simple pero poderoso gestor de pruebas para aplicaciones. La evaluación consiste en subir el apk generador por Android Studio, realizar configuraciones sobre las versiones de Android y dispositivos de ejecución. Para mayor información revisar el siguiente enlace <https://firebase.google.com/docs/test-lab/?hl=es-419>.

También se utilizó la consola de Google Play para desarrolladores, que ofrece informes sobre errores, caídas o bloqueos de la aplicación mientras es usada por los usuarios.

## 3.7. Publicación en Play Store

Para publicar aplicaciones en la tienda oficial de Google o Play Store, es necesario lo siguiente:

- Crear una cuenta de desarrollador, que posee un costo de \$25. La cuenta se puede crear desde el siguiente enlace <https://play.google.com/apps/publish/signup/>.
- En la tienda de aplicaciones solo se suben versiones release del apk, esto se genera en Android Studio.
- Finalmente se procede a subir el apk release y a llenar todas las fichas y datos que nos pide Google acerca de la aplicación. Luego de la publicación, la aplicación tardará un par de horas aparecer en Play Store.

Como se observa en la figura 10, Protoia ya se encuentra publicada (como versión beta o "sin publicar") en Play Store lista para ser descargada.



# Protoia (sin publicar)

Aitec Herramientas

Para todos

- Esta aplicación está en desarrollo y puede ser inestable.
- Esta aplicación es compatible con tu dispositivo.

Añadir a la lista de deseos

Instalar

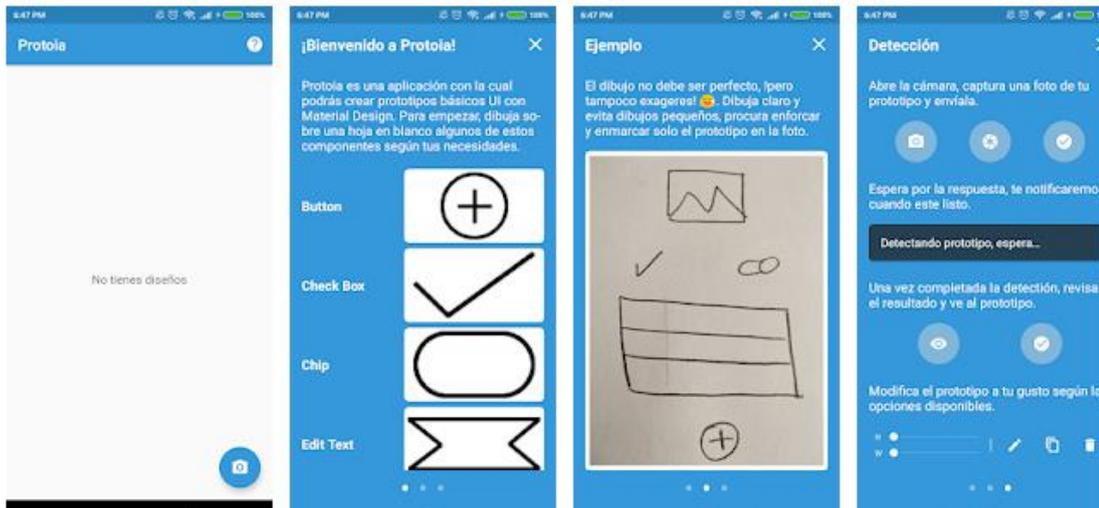


Figura 39. Aplicación Protoia publicada en Play Store

## Anexo 16. Manual de usuario de la aplicación móvil

### ÍNDICE DE CONTENIDO

<b>ÍNDICE DE FIGURAS</b> .....	106
<b>ÍNDICE DE TABLAS</b> .....	106
1. PANTALLAS DE NAVEGACIÓN .....	107
1.1. Inicio de sesión anónimo .....	107
1.2. Principal.....	107
1.3. Cámara – Vista previa .....	110
1.4. Diseño de prototipo.....	112
1.5. Ayuda .....	113

### ÍNDICE DE FIGURAS

Figura 1. Pantalla inicio de sesión anónimo .....	107
Figura 2. Pantalla principal.....	108
Figura 3. Lista de prototipos.....	108
Figura 4. Botón eliminar prototipo .....	108
Figura 5. Ítem de prototipo .....	109
Figura 6. Botón nuevo prototipo .....	109
Figura 7. Botón ayuda de la aplicación .....	109
Figura 8. Sección esperando detección .....	109
Figura 9. Sección Prototipo detectado .....	109
Figura 10. Pantalla Captura de foto .....	110
Figura 11. Botón de flash LED .....	110
Figura 12. Botón retroceder .....	111
Figura 13. Botón capturar foto .....	111
Figura 14. Pantalla Vista previa de foto – ver resultado de detección .....	111
Figura 15. Botón retroceder .....	111
Figura 16. Botón aceptar resultado de detección .....	112
Figura 17. Pantalla de diseño de prototipo.....	112
Figura 18. Secciones de pantalla Ayuda .....	114

### ÍNDICE DE TABLAS

TABLA I. OPCIONES DE EDICIÓN DE PROTOTIPO .....	112
---	-----

## INTRODUCCIÓN

El presente manual de usuario tiene como objetivo detallar la funcionalidad de la aplicación móvil “Protoia”. Esta se encuentra publicada en versión beta, por lo que puede ser inestable, se puede encontrar por medio del siguiente enlace:

<https://play.google.com/store/apps/details?id=com.makerly.fastdesign>.

### 1. PANTALLAS DE NAVEGACIÓN

La aplicación cuenta con varias pantallas de navegación, cada una posee diferentes funcionalidades. A continuación, se analizan cada una de ellas.

#### 1.1. Inicio de sesión anónimo

En esta pantalla se realiza el inicio de sesión anónimo, este proceso es automático y necesita de conexión a internet la primera vez que la aplicación es iniciada, el usuario debe esperar a que la barra de progreso se oculte. En la figura 1, se puede observar esta pantalla.

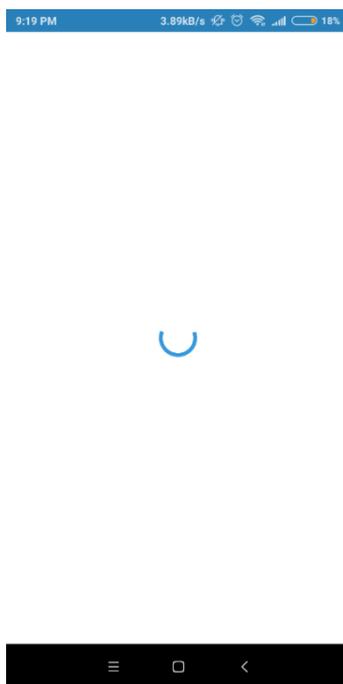
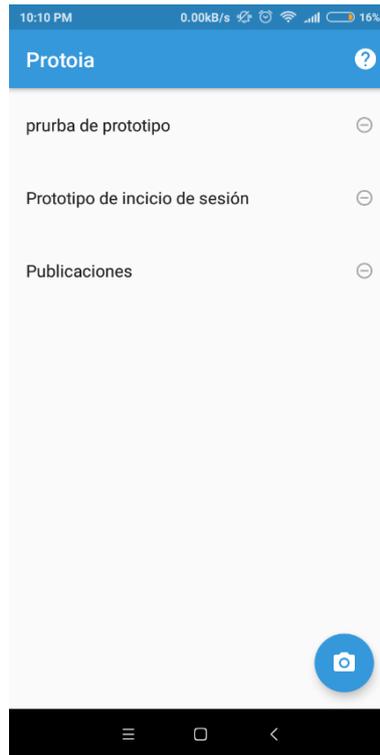


Figura 40. Pantalla inicio de sesión anónimo

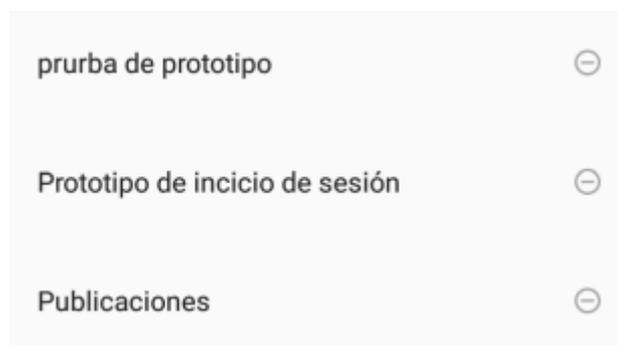
#### 1.2. Principal

Esta pantalla (figura 2), es la que ve el usuario luego del inicio de sesión automático. Posee varias secciones y funciones, a continuación, se detalla cada una de ellas:



**Figura 41. Pantalla principal**

- Lista de prototipos. En esta lista se visualizan todos los prototipos guardados por el usuario (figura 3).



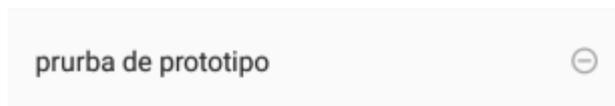
**Figura 42. Lista de prototipos**

- Eliminar prototipo. Cada ítem de la lista de prototipos tiene la funcionalidad de eliminar un prototipo permanentemente, se encuentra ubicada a la derecha del ítem (figura 4).



**Figura 43. Botón eliminar prototipo**

- Acceder al prototipo. Esta opción permite presionar sobre un ítem de la lista, esta acción nos llevara al diseño del prototipo presionado (figura 5).



**Figura 44. Ítem de prototipo**

- Nuevo prototipo. Mediante esta acción podemos iniciar la creación de un nuevo prototipo, al presionar el botón circular situado en la parte inferior a la derecha de la pantalla (figura 6).



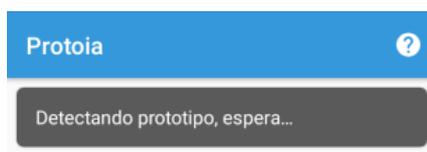
**Figura 45. Botón nuevo prototipo**

- Ayuda. Esta opción nos permite acceder a la ayuda de la aplicación, al presionar sobre el botón ubicado en la parte superior derecha de la pantalla sobre la barra de tareas (figura 7).

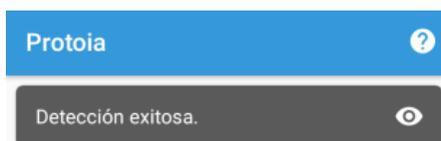


**Figura 46. Botón ayuda de la aplicación**

- Detección. Esta parte de la aplicación solo se observa cuando se está realizando el proceso de detección que se explica más adelante. En ella podemos observar los resultados cuando el proceso haya finalizado (figura 8 y 9).



**Figura 47. Sección esperando detección**

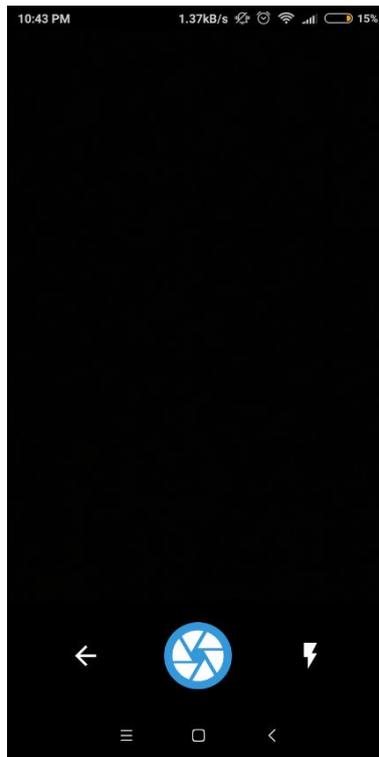


**Figura 48. Sección Prototipo detectado**

En caso de estar fuera de la pantalla principal o fuera de la aplicación, estos avisos se mostraran en la barra de notificaciones, mediante una notificación.

### 1.3. Cámara – Vista previa

La pantalla de cámara (figura 10), se utiliza para capturar una foto del boceto dibujado por el usuario. Para acceder a esta pantalla, es necesario que el usuario acepte unos permisos que serán requeridos por el sistema operativo, en caso de no aceptarse, dicha pantalla no se mostrará. A continuación, se detallan sus funciones:



**Figura 49. Pantalla Captura de foto**

- Activar/Desactivar Flash LED. Este botón (figura 11), permite al usuario encender o apagar el Flash LED.



**Figura 50. Botón de flash LED**

- Retroceder. Este botón (figura 12), permite al usuario retroceder a la pantalla principal.



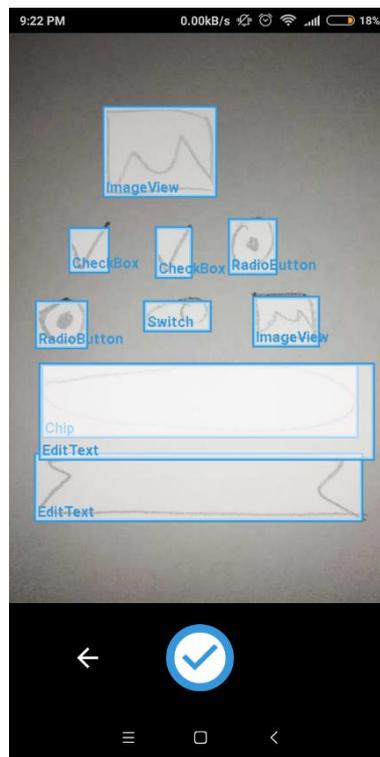
**Figura 51. Botón retroceder**

- Capturar foto. Este botón (figura 13), permite al usuario capturar una foto y presenta su vista previa.



**Figura 52. Botón capturar foto**

La pantalla de vista previa (figura 14), se utiliza para ver la vista previa de la foto capturada y también para ver los resultados de detección de componentes. A continuación, se detallan sus funciones:



**Figura 53. Pantalla Vista previa de foto – ver resultado de detección**

- Retroceder. Este botón (figura 15), permite al usuario retroceder a la pantalla principal.



**Figura 54. Botón retroceder**

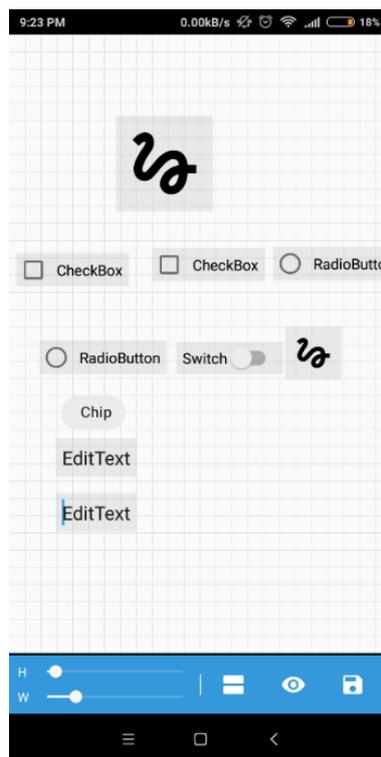
- Aceptar resultado de detección. Este botón (figura 16), posee dos funciones:
  - Permite al usuario aceptar el resultado de detección para continuar con el diseño de prototipo.
  - Permite confirmar la vista previa de la foto capturada para enviarla al servidor. En este caso, se iniciará un servicio en primer plano, mostrando una notificación en la barra de notificaciones del sistema.



**Figura 55. Botón aceptar resultado de detección**

## 1.4. Diseño de prototipo

Esta pantalla (figura 17), permite al usuario editar el prototipo con opciones básicas descritas en la tabla 1.



**Figura 56. Pantalla de diseño de prototipo**

Para poder acceder a la mayoría de las siguientes funciones, es necesario seleccionar un componente. Para cambiar la ubicación a un componente solo debe arrastrarlo.

**TABLA XLII. OPCIONES DE EDICIÓN DE PROTOTIPO**

ICONO	DESCRIPCIÓN
	<p>Permite cambiar el contenido de un componente.</p> <ul style="list-style-type: none"> <li>• En componentes que extiendan de ImageView, permite cambiar la imagen.</li> <li>• En componentes que extiendan de TextView, permite cambiar el texto y tamaño de texto.</li> </ul>
	<p>Permite duplicar un componente.</p>
	<p>Permite mediante pulsación larga, eliminar un componente.</p>
	<p>Permite activar o desactivar barras adicionales como:</p> <ul style="list-style-type: none"> <li>• ToolBar</li> <li>• TabLayout</li> <li>• BottomNavigation</li> </ul>
	<p>Permite ocultar la barra de diseño. En este modo, las teclas de volumen cumplen una función:</p> <ul style="list-style-type: none"> <li>• Botón volumen arriba para mostrar la barra de diseño.</li> <li>• Botón volumen abajo para compartir captura de pantalla del diseño.</li> </ul>
	<p>Permite guardar el prototipo, la primera se debe ingresar un nombre de prototipo.</p>
	<ul style="list-style-type: none"> <li>• H o Height, permite cambiar la altura del componente.</li> <li>• W o Width, permite cambiar el ancho del componente.</li> </ul>

## 1.5. Ayuda

La pantalla de Ayuda (figura 18), permite al usuario conocer acerca de las funciones importantes de la aplicación. No existe mucha iteración, solo el cambio o deslizamiento entre tres secciones. Para salir, debe pulsar el botón “X” ubicado en la parte superior derecha de la pantalla.

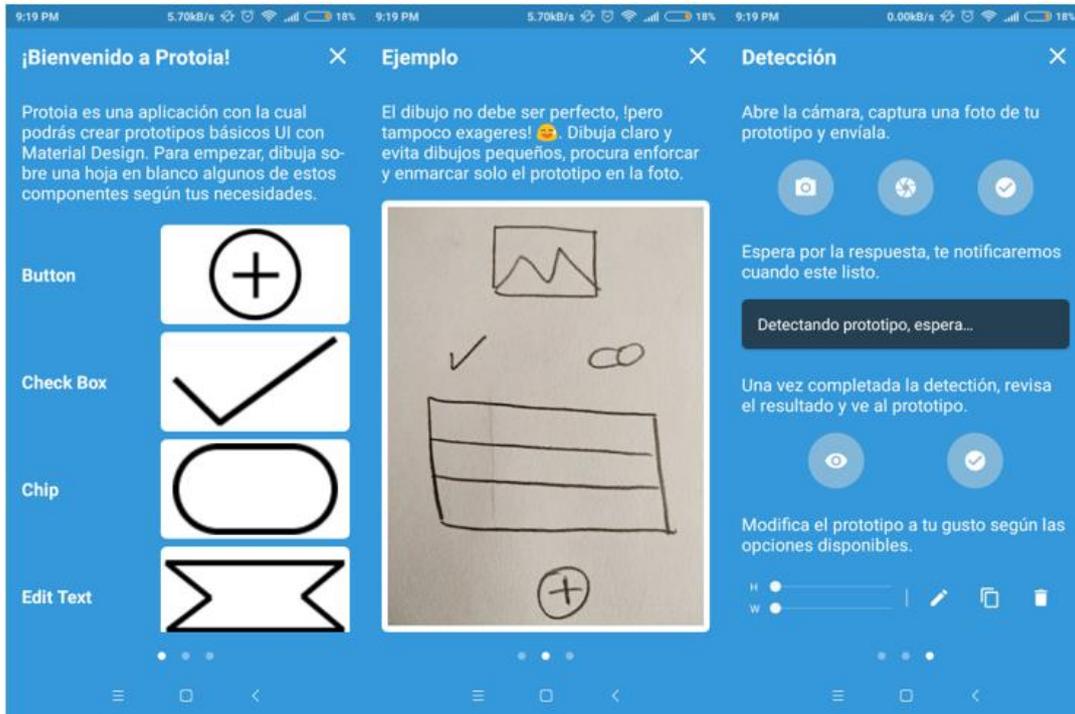


Figura 57. Secciones de pantalla Ayuda