UNIVERSIDAD NACIONAL DE CHIMBORAZO
FACULTAD DE INGENIERÍA
CARRERA DE INGENIERÍA EN TECNOLOGÍAS DE LA INFORMACIÓN

Aplicación móvil basada en Computer Vision para promocionar artesanías en la ciudad de Riobamba

Trabajo de Titulación para optar al título de
Ingeniero en Tecnologías de la Información

Autor:
Tierra Gusqui Wilson Armando

Tutor:
Ing. Miryan Estela Narváez, PhD

Riobamba, Ecuador. 2026

DECLARATORIA DE AUTORÍA

Yo, Wilson Armando Tierra Gusqui, con cédula de ciudadanía 0605995125, autor del trabajo de investigación titulado: Aplicación móvil basada en Computer Vision para promocionar artesanías en la ciudad de Riobamba, certifico que la producción, ideas, opiniones, criterios, contenidos y conclusiones expuestas son de mí exclusiva responsabilidad.

Asimismo, cedo a la Universidad Nacional de Chimborazo, en forma no exclusiva, los derechos para su uso, comunicación pública, distribución, divulgación y/o reproducción total o parcial, por medio físico o digital; en esta cesión se entiende que el cesionario no podrá obtener beneficios económicos. La posible reclamación de terceros respecto de los derechos de autor de la obra referida será de mi entera responsabilidad; librando a la Universidad Nacional de Chimborazo de posibles obligaciones.

En Riobamba, 10 de abril de 2026.

Wilson Armando Tierra Gusqui

C.I: 0605995125

DICTAMEN FAVORABLE DEL PROFESOR TUTOR

Quien suscribe, Miryan Estela Narváez Vilema catedrático adscrito a la Facultad de Ingeniería, por medio del presente documento certifico haber asesorado y revisado el desarrollo del trabajo de investigación titulado: Desarrollo de prototipo móvil-web para gestionar disponibilidad y alertar intrusiones mediante autenticación facial en parqueaderos Cooperativa "Señor del Buen Suceso", bajo la autoría de John Jairo Guaman Pineda; por lo que se autoriza ejecutar los trámites legales para su sustentación.

Es todo cuanto informar en honor a la verdad; en Riobamba, a los 27 días del mes de abril de 2026.

Validar únicamente en FirmaEC.
Firmado electrónicamente por:
MIRYAN ESTELA
NARVAEZ VILEMA

Miryan Estela Narváez Vilema

C.I: 0603576778

CERTIFICADO DE LOS MIEMBROS DEL TRIBUNAL

Quienes suscribimos, catedráticos designados Miembros del Tribunal de Grado para la evaluación del trabajo de investigación titulado: Aplicación móvil basada en Computer Vision para promocionar artesanías en la ciudad de Riobamba, presentado por Wilson Armando Tierra Gusqui, con cédula de identidad número 0605995125, bajo la tutoría de PhD. Miryan Estela Narváez Vilema; certificamos que recomendamos la APROBACIÓN de este con fines de titulación. Previamente se ha evaluado el trabajo de investigación y escuchada la sustentación por parte de su autor; no teniendo más nada que observar.

De conformidad a la normativa aplicable firmamos, en Riobamba 22 de junio de 2026.

Fernando Molina, PhD.
PRESIDENTE DEL TRIBUNAL DE GRADO

Diego Reina, Mgs.
MIEMBRO DEL TRIBUNAL DE GRADO

Pamela Buñay, PhD.
MIEMBRO DEL TRIBUNAL DE GRADO

CERTIFICACIÓN

Que, **TIERRA GUSQUI WILSON ARMANDO** con CC: **0605995125**, estudiante de la carrera **INGENIERÍA EN TECNOLOGÍAS DE LA INFORMACIÓN**, Facultad de **INGENIERÍA**; ha trabajado bajo mi tutoría el trabajo de investigación titulado "**APLICACIÓN MÓVIL BASADA EN COMPUTER VISION PARA PROMOCIONAR ARTESANÍAS EN LA CIUDAD DE RIOBAMBA**", cumple con el 6 %, de acuerdo al reporte del sistema Anti plagio **COMPILATIO**, porcentaje aceptado de acuerdo a la reglamentación institucional, por consiguiente autorizo continuar con el proceso.

Riobamba, 24 de abril de 2026

Firmado electrónicamente por:
MIRYAN ESTELA
NARVAEZ VILEMA

PhD. Miryan Narváez
TUTORA

DEDICATORIA

Dedico este trabajo de investigación especialmente a mis padres, por ser un gran ejemplo de perseverancia y esfuerzo constante, gracias a su esfuerzo, apoyo y respaldo en cada una de mis decisiones para poder alcanzar esta meta.

A mis hermanos/as que supieron brindar consejos y recomendaciones para mi superación y bienestar, además del aliento para no rendirme en este camino.

A todos ustedes, con todo mi corazón les dedico este logro.

Wilson Armando Tierra Gusqui

AGRADECIMIENTO

Agradezco a mi familia que a pesar de buenos o malos momentos siempre me alentaron, apoyaron y confiaron en mí, a amigos que fueron un gran apoyo durante este trayecto, dándome un consejo o simplemente motivándome a seguir adelante y no rendirme, a todas y cada una de las personas que fueron parte de este trayecto les agradezco por hacer de esta etapa algo inolvidable, llena de experiencias maravillosas y momentos únicos.

Wilson Armando Tierra Gusqui

ÍNDICE GENERAL

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

RESUMEN

El objetivo de esta investigación fue desarrollar una aplicación móvil basada en Computer Vision para apoyar la promoción de artesanías en la ciudad de Riobamba, mediante el reconocimiento visual de productos y la presentación de información asociada para potenciales compradores.

Para alcanzar este objetivo, se aplicó una metodología de investigación basada en un diseño experimental de pruebas técnicas controladas, complementada con un proceso de desarrollo iterativo para la construcción de un prototipo funcional. La investigación incluyó el análisis de requerimientos, el diseño de la arquitectura y la implementación de un módulo de Computer Vision conectado a una base de datos de artesanías. El diseño experimental se enfocó en la fase de evaluación del sistema, donde se realizaron pruebas técnicas bajo condiciones controladas en dispositivos móviles para medir el rendimiento, la precisión y los tiempos de respuesta del modelo frente a distintos escenarios de uso. Esto permitió analizar el comportamiento del sistema y validar su funcionamiento en comparación con los métodos tradicionales de búsqueda de productos artesanales.

Los resultados obtenidos evidenciaron que la aplicación permitió identificar artesanías de forma automática y mostrar información relevante del producto reconocido, reduciendo el tiempo de búsqueda manual por parte del usuario. Además, se comprobó la correcta integración entre el módulo de reconocimiento visual y el sistema de gestión de información. En cuanto al rendimiento, se evaluaron métricas relacionadas con el tiempo de respuesta de reconocimiento y la estabilidad de la aplicación durante su ejecución, observándose un comportamiento aceptable para un entorno móvil en condiciones de prueba controladas. Los ensayos realizados confirmaron que el sistema respondió de manera consistente sin interrupciones significativas durante el proceso de detección y consulta de datos.

Se concluyó que la solución aportó una alternativa innovadora para difundir productos artesanales y que la evaluación del desempeño permitió validar su funcionamiento, recomendándose como trabajo futuro la optimización del modelo de reconocimiento y la ampliación de las pruebas de rendimiento en escenarios reales.

Palabras claves: Computer Vision, artesanías, evaluación de rendimiento, reconocimiento de imágenes.

ABSTRACT

The objective of this research was to develop a computer vision-based mobile application to support the promotion of handicrafts in the city of Riobamba, through visual product recognition and the presentation of associated information to potential buyers.

To achieve this objective, a research methodology based on an experimental design of controlled technical tests was applied, complemented by an iterative development process for building a functional prototype. The research included requirements analysis, architecture design, and the implementation of a computer vision module connected to a handicrafts database. The experimental design focused on the system evaluation phase, where technical tests were performed under controlled conditions on mobile devices to measure the model's performance, accuracy, and response times under different usage scenarios. This allowed for the analysis of the system's behavior and the validation of its functionality compared to traditional methods for searching for handicrafts.

The results obtained showed that the application automatically identified handicrafts and displayed relevant information about the recognized product, reducing the time spent manually searching by the user. Furthermore, the successful integration between the visual recognition module and the information management system was verified. Regarding performance, metrics related to recognition response time and application stability during execution were evaluated, revealing acceptable behavior for a mobile environment under controlled test conditions. The trials confirmed that the system responded consistently without significant interruptions during the data detection and query process.

It was concluded that the solution provided an innovative alternative for promoting handcrafted products and that the performance evaluation validated its functionality. Future work recommended includes optimizing the recognition model and expanding performance testing to real-world scenarios.

Keywords: Computer vision, crafts, performance evaluation, image recognition.

Reviewed and approved by Jacqueline Armijos

CAPÍTULO I. INTRODUCCIÓN

La artesanía constituyó una manifestación cultural, social y económica que refleja la identidad de los pueblos. En ciudades con una rica tradición como Riobamba, las expresiones artesanales representan no solo un patrimonio valioso, sino también una fuente importante de ingresos para muchas familias. Sin embargo, los artesanos enfrentan dificultades para posicionar sus productos en un entorno cada vez más competitivo y digitalizado. La falta de visibilidad, la escasa presencia en plataformas tecnológicas y la carencia de herramientas innovadoras limitan significativamente su capacidad para acceder a nuevos mercados y atraer a potenciales compradores.

En este contexto, la transformación digital, apoyada por tecnologías emergentes como Computer Vision, representa una oportunidad estratégica para dinamizar el sector artesanal. Esta tecnología permitirá que una aplicación móvil pueda interpretar imágenes, reconocer patrones visuales, clasificar productos y facilitar búsquedas visuales, mejorando la experiencia de compra y el proceso de promoción artesanal. Mediante el uso de la cámara del dispositivo móvil, los usuarios podrán descubrir artesanías similares a partir de una fotografía, mientras que los artesanos podrán clasificar automáticamente sus productos al subir una imagen.

Esta propuesta de investigación plantea el desarrollo de una aplicación móvil basada en Computer Vision como estrategia para potenciar la promoción y comercialización de artesanías en Riobamba. La app no solo ofrecerá un espacio virtual para la exhibición de productos, sino que integrará funcionalidades como catálogos inteligentes, identificación visual de productos y una interfaz intuitiva. Estas características permitirán conectar de forma eficiente a los artesanos con sus clientes, impulsando el comercio justo y fomentando la valoración cultural de sus productos.

De esta manera, el proyecto buscó aportar a la reactivación económica del sector artesanal a través de la innovación tecnológica. Al integrar herramientas de inteligencia artificial como la Computer Vision en un entorno accesible y funcional, se pretende fortalecer la identidad cultural, aumentar la competitividad del sector artesanal y posicionar a Riobamba como referente en el uso de tecnologías avanzadas para la promoción del patrimonio cultural.

1.1 Planteamiento del Problema

En la ciudad de Riobamba, el comercio artesanal representa una actividad económica de valor cultural significativo, sustentada por familias que mantienen viva la tradición a través de la elaboración de productos únicos. Sin embargo, este sector enfrenta una serie de limitaciones para adaptarse a los entornos digitales actuales, lo que restringe su capacidad de expansión y visibilidad en mercados más amplios. Muchos artesanos carecen de acceso a plataformas de venta modernas, recursos tecnológicos o conocimientos necesarios para promocionar sus productos de forma efectiva, lo cual los coloca en desventaja frente a otras formas de comercio que sí aprovechan las ventajas de la digitalización.

A pesar del auge del comercio electrónico y del uso masivo de dispositivos móviles, existe una brecha tecnológica que impide que los productores artesanales puedan integrarse plenamente a estas dinámicas. Además, los canales tradicionales de comercialización como ferias, mercados locales o ventas informales presentan limitaciones en alcance y continuidad, reduciendo las posibilidades de crecimiento económico y desarrollo sostenible para los artesanos.

Debido a esto surge la necesidad de desarrollar soluciones digitales que permitan publicar productos y que optimicen la forma en que los usuarios los descubren y acceden a ellos. Una de las barreras más comunes es la dificultad para encontrar productos artesanales específicos, debido a la falta de sistemas eficientes de categorización, búsqueda o filtrado. El uso de tecnologías como Computer Vision puede representar una innovación clave al permitir que los usuarios identifiquen productos mediante imágenes, y que los artesanos clasifiquen automáticamente sus productos al momento de subirlos.

Ante esta realidad, se plantea la necesidad de una aplicación móvil que integre técnicas de Computer Vision para facilitar la identificación, clasificación visual y promoción de productos artesanales, ampliando su alcance comercial y fortaleciendo la visibilidad del sector artesanal en Riobamba.

1.2 Justificación

La artesanía es una manifestación tangible de la identidad cultural de los pueblos. En Riobamba, representa no solo una tradición profundamente arraigada, sino también una fuente significativa de ingresos para muchas familias. Sin embargo, los canales tradicionales de promoción y venta artesanal son insuficientes frente a las exigencias del mercado actual, que demanda procesos más ágiles, visuales e interactivos. La escasa presencia de los artesanos en entornos digitales limita su visibilidad, reduce su alcance comercial y obstaculiza la expansión de sus productos hacia nuevos públicos.

En este contexto, la incorporación de tecnologías emergentes como Computer Vision puede transformar la forma en que se promocionan y acceden los productos artesanales. Esta tecnología, integrada en una aplicación móvil, permitirá que los usuarios puedan buscar artesanías utilizando imágenes capturadas desde sus dispositivos, accediendo de forma rápida a productos visualmente similares o relacionados. Además, facilitará a los artesanos la clasificación automática de sus creaciones, reduciendo barreras técnicas y mejorando la organización de catálogos digitales.

La propuesta de desarrollar una aplicación móvil con capacidades de visión por computadora se justifica por su potencial para modernizar la experiencia de compra, facilitar el acceso al mercado digital, y reforzar la conexión entre el valor cultural del producto artesanal y el consumidor moderno. Al incorporar funcionalidades como reconocimiento de imágenes y clasificación inteligente, esta herramienta tecnológica promueve la inclusión digital, fortalece la economía local y contribuye a la preservación activa del patrimonio cultural de Riobamba.

El proyecto tiene también un impacto social relevante, ya que proporciona a los artesanos una solución accesible y adaptada a las tendencias tecnológicas actuales. No se trata solo de desarrollar una plataforma técnica, sino de generar una herramienta de transformación digital orientada al desarrollo sostenible, donde la tecnología se convierte en un puente entre la tradición artesanal y el comercio inteligente.

1.3 Formulación del Problema

¿Cómo influye una aplicación móvil basada en Computer Vision en el rendimiento de identificación, clasificación y búsqueda visual de artesanías en la ciudad de Riobamba?

1.4 Objetivos

Objetivo General

Implementar una aplicación móvil basada en Computer Vision para promocionar artesanías en la ciudad de Riobamba.

Objetivos Específicos

- Investigar las tecnologías móviles y herramientas de Computer Vision adecuadas para el desarrollo de una aplicación destinada a la promoción de artesanías.
- Desarrollar una aplicación móvil que utilice Computer Vision para identificar, clasificar y promocionar productos artesanales.
- Evaluar el rendimiento de la aplicación móvil utilizando la herramienta Flutter DevTools.

CAPÍTULO II. MARCO TEÓRICO

2.1 Fundamentos de Computer Vision

2.1.1 Computer Vision

En la actualidad, la integración de Computer Vision en dispositivos móviles ha trascendido el ámbito recreativo para convertirse en un motor de desarrollo económico en sectores artesanales y culturales. Investigaciones recientes han explorado cómo la inteligencia artificial puede reducir la brecha digital en comunidades con limitaciones tecnológicas.

El uso de modelos de clasificación de imágenes basados en Deep Learning permite que productos con alta variabilidad visual, como las artesanías tradicionales, sean categorizados con precisiones superiores al 90%, facilitando su inserción en catálogos digitales automáticos. Por otro lado, sostiene que la adopción de arquitecturas ligeras como YOLO en entornos móviles es fundamental para garantizar el rendimiento en dispositivos de gama media, lo cual es directamente aplicable a la realidad tecnológica de los comerciantes en regiones en vías de desarrollo [1].

En el contexto de la gestión del patrimonio y comercio local, destacan que las aplicaciones móviles no solo sirven como vitrinas de venta, sino como herramientas de preservación cultural que aumentan la visibilidad de productores minoritarios frente a mercados globalizados [2]. Finalmente, estudios sistemáticos en el área de humanidades digitales en Ecuador confirman que la principal barrera para los artesanos no es la falta de producto, sino la falta de canales técnicos eficientes, validando la necesidad de soluciones basadas en reconocimiento visual para optimizar la búsqueda y promoción.

Computer Vision es un campo de la inteligencia artificial que permite a las máquinas interpretar imágenes y videos, generando información procesable para la toma de decisiones [3].

A través de la mezcla de la inteligencia de machine learning y los modelos de deep learning, esta herramienta es capaz de analizar imágenes y descifrar lo que contienen: desde simples objetos hasta rostros, emociones e incluso patrones sutiles y difíciles de detectar.

2.1.2 Etapas del procesamiento de imágenes

El procesamiento de imágenes representó una de las etapas fundamentales dentro de Computer Vision, porque permitió transformar una imagen digital en información útil, para su posterior análisis e interpretación por parte de un sistema inteligente. A lo largo del tiempo, esta disciplina evolucionó desde simples técnicas de mejora visual hasta complejos algoritmos de aprendizaje profundo, capaces de identificar patrones, formas y texturas con alta precisión. En la Figura 1, se muestra el procesamiento de imágenes.

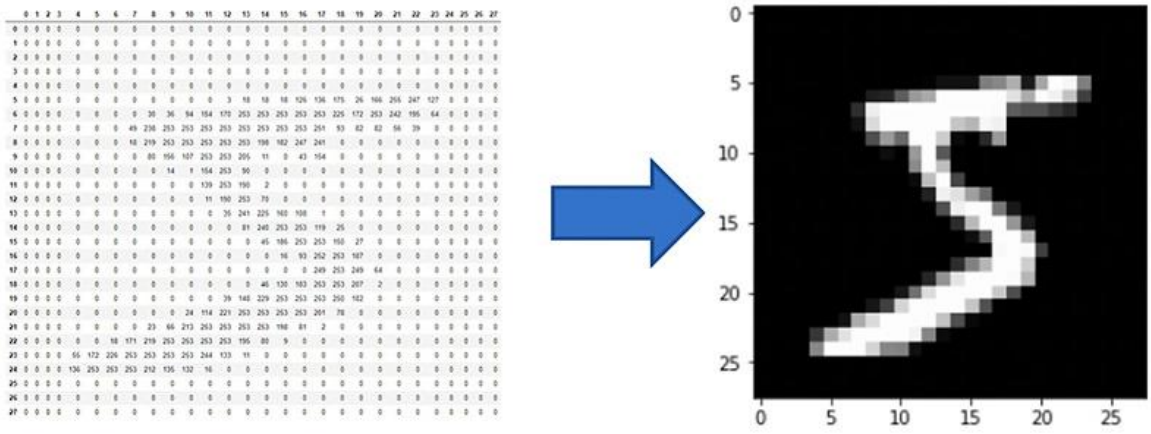


Figura 1. Procesamiento de imágenes
Fuente: *[4]*

2.1.3 Técnicas y algoritmos aplicados a la clasificación de objetos

Los enfoques Deep Learning en visión artificial son útiles para la detección de objetos, reconocimiento de objetos, enfoque de imágenes borrosas y segmentación de escenas. Deep Learning implica entrenar redes neuronales convolucionales (CNN), que aprenden directamente de los datos con patrones a diferentes escalas. Entrenar CNN requiere un conjunto de gran tamaño de imágenes de entrenamiento o de nubes de puntos etiquetadas. Es posible considerar dos tipos de reconocimiento [5] sí en la imagen se busca un objeto particular o conocido se realiza una detección del objeto, sin embargo, si se busca reconocer diferentes instancias de una categoría genérica se realiza el reconocimiento de la instancia. El último tipo corresponde al proceso que clasifica los objetos en diferentes clases, por ejemplo, reconocer dos marcas y modelos de vehículos diferentes, y según ciertas características compartidas ubicarlos en la clase autos. El reconocimiento de objetos está basado en asignar dichas clases a los diferentes objetos, y la herramienta que realiza este proceso se denomina clasificador. La transferencia del aprendizaje utiliza redes previamente entrenadas para acelerar este proceso con menos datos de entrenamiento [6]. La Figura 2, muestra la clasificación de objetos.

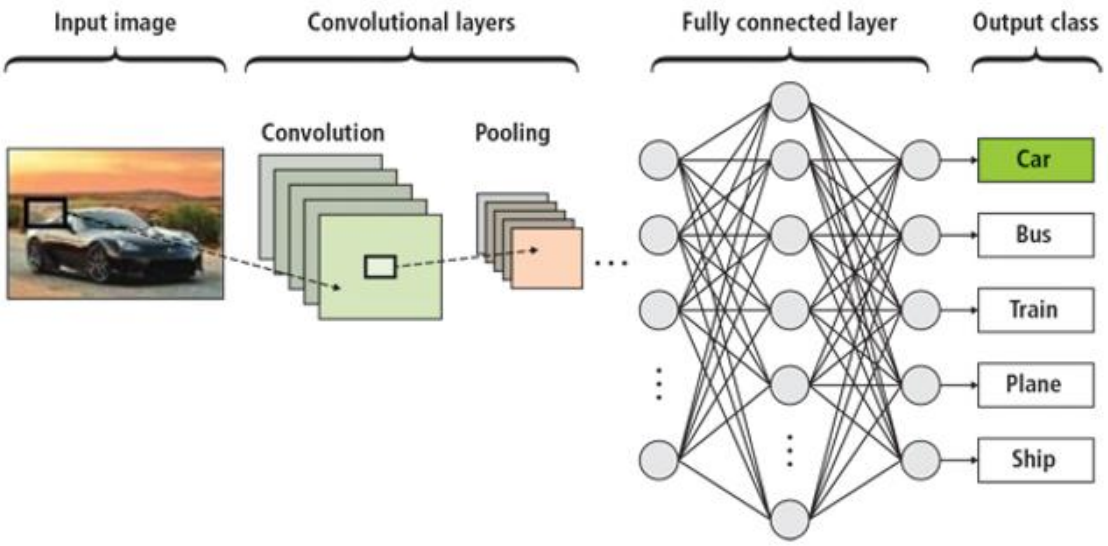


Figura 2. Clasificación de objetos
Fuente: [7]

2.1.4 Aplicaciones de Computer Vision en entornos móviles

Las técnicas de Computer Vision se está integrando cada vez más en la vida diaria gracias al avance de los smartphones. Los nuevos smartphones, cada vez más potentes, permiten procesar algoritmos complejos que antes solo podían ejecutarse en dispositivos de alto rendimiento. Esto ha dado paso a múltiples aplicaciones, siendo la realidad aumentada una de las más destacadas, ya que combina de manera visual el mundo real con elementos virtuales, creando experiencias más interactivas para el usuario.

2.2 Tecnologías y herramientas para el desarrollo de aplicaciones móviles

2.2.1 OpenCV y su integración con Flutter

OpenCV (Open Source Computer Vision Library) representa una de las bibliotecas más reconocidas para el procesamiento digital de imágenes y visión por computador. Su estructura modular permite aplicar técnicas de detección, segmentación, reconocimiento facial, análisis de movimiento y reconstrucción tridimensional en múltiples plataformas [8]. (Ver Figura 3)

Figura 3. Funciones de OpenCV
Fuente: [9]

2.2.2 TensorFlow Lite y modelos de aprendizaje automático

Es un conjunto de herramientas que ayuda a los desarrolladores a ejecutar sus modelos en dispositivos incorporados, móviles o de IoT, permite implementar el aprendizaje automático integrado en el dispositivo. Optimizado para el aprendizaje automático integrado en el dispositivo, ya que aborda 5 limitaciones clave: latencia (no hay ida y vuelta con un servidor), privacidad (ningún dato personal sale del dispositivo), conectividad (no es necesaria una conexión a Internet), tamaño (tamaño reducido del modelo y de los objetos binarios) y consumo de energía (inferencia de alta eficiencia sin necesidad de conexiones de red) [10].

2.2.3 Selección de librerías y frameworks móviles

Los frameworks móviles multiplataforma se utilizan para generar una aplicación a la que se puede acceder a través de una gran cantidad de dispositivos, independientemente del sistema operativo, Android o iOS. El objetivo es utilizar un único código en múltiples plataformas para facilitar la portabilidad y, al final, el coste del desarrollo de software se mantiene bajo. Con los frameworks móviles multiplataforma, existe la posibilidad de una exposición máxima al público objetivo. Por ejemplo, una sola aplicación puede apuntar a plataformas iOS y Android, lo que maximiza el alcance de la aplicación [11].

2.3 Framework Flutter para el desarrollo

Flutter se constituyó como un framework de desarrollo multiplataforma de código abierto creado por Google con el propósito de simplificar la creación de aplicaciones nativas desde una única base de código. Este entorno permitió el desarrollo simultáneo para Android, iOS, Web y escritorio, reduciendo los costos de mantenimiento y asegurando una experiencia de usuario uniforme entre plataformas [12].

2.3.1 Características de Flutter

Flutter, desarrollado por Google, es un framework de código abierto que se destaca por permitir el desarrollo de aplicaciones multiplataforma con un rendimiento similar al nativo, utilizando una única base de código. La Figura 4 presenta el diseño adaptativo y responsivo en Flutter.

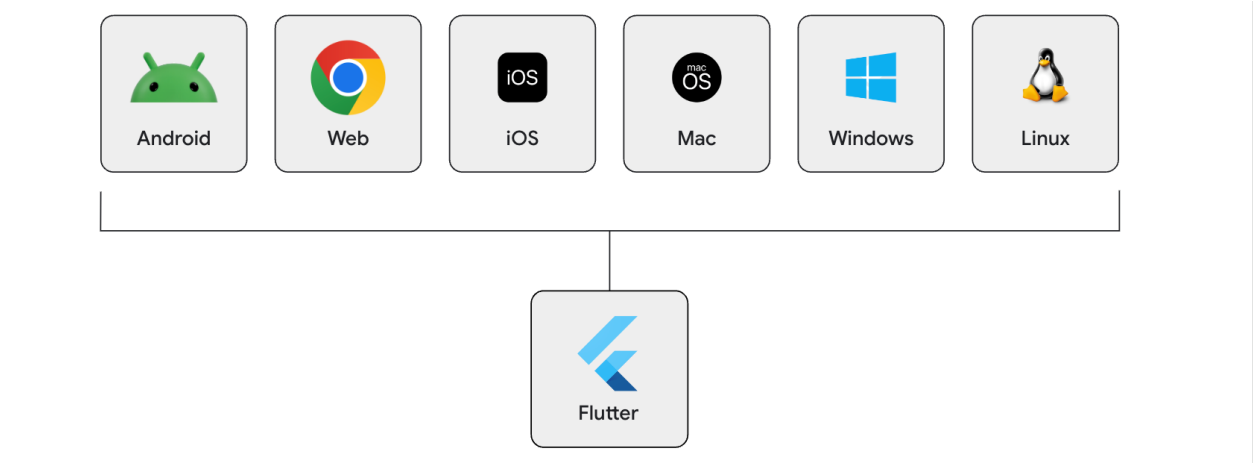


Figura 4. Diseño adaptativo y responsivo en Flutter
Fuente: [13]

La Tabla 1 presenta características principales de Flutter.

Tabla 1. Características de Flutter

Característica	Descripción
Desarrollo Multiplataforma	Permite crear aplicaciones para iOS, Android, Web, Windows, macOS y Linux desde una única base de código.
Rendimiento Nativo	Las aplicaciones se compilan a código de máquina nativo (código ARM), lo que asegura una ejecución rápida y fluida, comparable al rendimiento de las aplicaciones nativas.
Hot Reload con Estado	Esta funcionalidad aumenta enormemente la productividad, ya que permite ver los cambios en el código reflejados en la aplicación en segundos sin perder el estado actual.
Motor de Renderizado Propio	Utiliza el motor gráfico Skia para dibujar cada píxel, garantizando una interfaz de usuario consistente y controlada en todas las plataformas.
Lenguaje Dart	Utiliza Dart, un lenguaje de programación optimizado por Google que se compila a código nativo y está diseñado específicamente para el desarrollo de aplicaciones cliente.

2.3.2 Beneficios de Flutter en rendimiento y usabilidad

Flutter es uno de los frameworks más populares para crear aplicaciones móviles multiplataforma. Esto debido a su arquitectura que utiliza compilación nativa, su motor gráfico propio y un entorno de desarrollo que está bastante optimizado.

La evaluación del rendimiento en Flutter se llevó a cabo mediante el uso de Flutter DevTools como herramienta principal para monitorear, analizar y optimizar diversos aspectos del comportamiento de la aplicación, entre ellos la fluidez de la interfaz, la carga del CPU, el consumo de memoria y la eficiencia del hilo de renderizado. Para garantizar la validez de las mediciones, la aplicación se ejecutó en modo profile o release sobre un dispositivo real, debido a que el modo debug introduce procesos adicionales que generan sobrecargas y pueden alterar los resultados obtenidos durante el análisis de rendimiento.

a) Alto rendimiento gracias a la compilación nativa

Flutter se compila directamente a código máquina nativo (ARM o x64). Esto elimina la capa intermedia y reduce significativamente la latencia en la ejecución, logrando una experiencia fluida y estable comparable a las aplicaciones desarrolladas de forma nativa en Android o iOS. El uso del motor Skia permite renderizar cada elemento visual directamente sobre la pantalla, manteniendo un rendimiento constante incluso a 60 fps o 120 fps en dispositivos modernos [14].

b) Usabilidad mejorada mediante interfaces coherentes

El sistema de widgets declarativos de Flutter permite construir interfaces visuales coherentes entre plataformas, evitando discrepancias entre versiones de Android o iOS. Esto mejora la experiencia del usuario (UX) al ofrecer consistencia visual y comportamientos uniformes en todos los dispositivos [15].

2.3.3 Paquetes y plugins para la integración con Computer Vision

En el desarrollo de la aplicación móvil basada en Computer Vision, se consideró importante identificar los paquetes y complementos más adecuados que facilitan la integración de funcionalidades de procesamiento de imágenes, detección de objetos y aprendizaje automático dentro del entorno de Flutter.

La Tabla 2, presenta principales paquetes o plugins de Computer Vision.

Tabla 2. Plugins Computer Vision

Paquete / Plugin	Descripción
google_ml_kit	Proporciona acceso a las APIs de Google ML Kit para tareas en dispositivo como detección de texto, reconocimiento facial, objetos y etiquetas de imagen.
tflite_flutter	Permite cargar y ejecutar modelos de TensorFlow Lite directamente en el dispositivo, realizando inferencias de aprendizaje automático en tiempo real.
opencv_dart	Vinculación de OpenCV con Dart mediante FFI, posibilitando operaciones clásicas de procesamiento de imágenes.
flutter_vision	Plugin especializado para detección y segmentación de objetos utilizando modelos YOLOv5, YOLOv8 y YOLOv11, integrando TensorFlow Lite para la inferencia.
hybrid_vision	ML Kit y Apple Vision ofrece resultados multiplataforma.

2.4 Inteligencia artificial y aprendizaje en la clasificación de imágenes

La clasificación de imágenes es una tarea importante en Computer Vision, ya que se encarga de entender automáticamente lo que hay en una imagen. Esto implica asignar una categoría o etiqueta a la imagen completa, basándose en lo que se puede ver en ella.

2.4.1 Fundamentos de la inteligencia artificial

La Inteligencia Artificial representa una disciplina dentro de la informática que busca construir máquinas capaces de funcionar de manera autónoma en entornos complejos y en constante cambio, creando sistemas que sean capaces de adaptarse y responder de manera efectiva a situaciones desafiantes [16].

2.4.2 Modelos de IA y su aprendizaje en la aplicación visual

La aplicación visual de la IA Computer Vision, se basa principalmente en las Redes Neuronales Convolucionales (CNNs). Estos modelos de Deep Learning aprenden a "ver" una imagen a través de un proceso de extracción jerárquica de características. Primeramente, las capas internas detectan patrones simples como bordes y texturas. Mientras que la información avanza a través de la red, el modelo combina estos patrones para identificar formas complejas y, finalmente, reconocer objetos completos como rostros, vehículos y más con gran precisión.

2.4.3 Modelos ligeros para dispositivos móviles (YOLO)

El modelo YOLO se destaca por su enfoque de detección de un solo disparo (single-shot), donde procesa la imagen una sola vez para predecir simultáneamente las ubicaciones y clases de todos los objetos. Esto elimina cuellos de botella y garantiza una baja latencia, resultando crucial para aplicaciones en tiempo real en móviles.

2.5 Metodología de desarrollo Mobile-D

Esta metodología se centra especialmente en las pequeñas empresas de desarrollo, debido a los tiempos cortos de desarrollo lo que produce como resultado la minimización de costes de producción, lo cual hace que esta metodología se convierta en una buena opción para pequeñas organizaciones que se limitan a tener poco personal y recursos.

2.5.1 Fases del proceso Mobile-D

La metodología Mobile-D se divide en cinco fases principales, cada una con un conjunto de etapas iterativas que garantizan la entrega continua de valor y la estabilidad del producto final. Este enfoque permite mantener un control riguroso sobre la planificación, el desarrollo y la calidad del software móvil [17].

La Tabla 3, resume las fases del proceso de la metodología Mobile-D

Tabla 3. Fases del proceso Mobile-D

Fase	Descripción
Exploración	Se analizan las necesidades del cliente, se define el alcance del proyecto, los requerimientos funcionales y no funcionales, y se evalúa la viabilidad técnica. Permite establecer la base conceptual y organizativa del desarrollo.
Inicialización	Se configura el entorno de desarrollo (hardware, software y repositorios), se conforma el equipo de trabajo y se planifican las primeras iteraciones.
Producción	Corresponde al desarrollo iterativo del sistema. Se implementan las funcionalidades principales, aplicando prácticas ágiles como integración continua y revisión constante con el cliente.
Estabilización	Se centra en la optimización del código, la corrección de errores, la mejora de la experiencia de usuario y la documentación técnica del sistema.
Pruebas del sistema	Se realizan pruebas integrales de funcionamiento, rendimiento y aceptación. Tras la validación final, se publica o entrega oficialmente la aplicación móvil.

CAPÍTULO III. METODOLOGÍA

3.1 Diseño de la investigación

Se adoptó un enfoque experimental de pruebas técnicas controladas donde se midieron indicadores objetivos de rendimiento de la aplicación (tiempo de respuesta de la búsqueda por imagen, precisión del modelo de clasificación, métricas de rendimiento).

3.2 Tipo de investigación

La presente investigación fue de tipo aplicada, pues buscó resolver un problema real como la baja visibilidad y digitalización del comercio artesanal en Riobamba, a través del desarrollo e implementación de una aplicación móvil basada en Computer Vision. El propósito fue diseñar, construir y evaluar una herramienta tecnológica que facilitó la promoción y búsqueda de artesanías mediante imágenes, contribuyendo a mejorar los procesos de comercialización de los artesanos locales.

3.3 Población de estudio y tamaño muestra

No aplica población ni muestra estadística debido a que la investigación corresponde a pruebas técnicas controladas sobre un prototipo de software, pues el enfoque radicó en el desarrollo y evaluación técnica de un prototipo de aplicación móvil de Computer Vision para artesanías. Cada acción relevante como la captura de imagen, la búsqueda por foto o la clasificación automática de un producto constituyó un "evento" que se reproduce de manera continua e ilimitada en condiciones reales de uso. El rendimiento del sistema se evaluó midiendo métricas como tiempos de inferencia, uso de recursos y precisión de la búsqueda a lo largo de un flujo constante de datos, esto prescindió de la necesidad de definir un tamaño de muestra fijo y permitió validar la escalabilidad y robustez del prototipo en un entorno de uso abierto.

3.4 Técnicas de recolección de datos

Pruebas de rendimiento del sistema: A través de herramientas de profiling de Flutter (DevTools), se registraron métricas automáticas de la aplicación como tiempo medio de inferencia del modelo de Computer Vision, uso de CPU/memoria durante la clasificación y latencia de la cámara. Estos datos cuantitativos sirvieron para validar la viabilidad de la solución en dispositivos reales.

3.5 Métodos de análisis y procesamiento de datos

El análisis de los datos se realizó mediante técnicas estadísticas y de visualización adaptadas a los distintos orígenes de información. Para los datos cuantitativos se aplicó un análisis descriptivo que incluyó frecuencias, medias y desviaciones estándar. Los resultados se presentaron en gráficos de barras, líneas de tendencia y evolución de las métricas.

Para evaluar el rendimiento de la aplicación se midieron tres indicadores clave: el tiempo de respuesta, entendido como el lapso medio (en milisegundos) que transcurrió desde

que el usuario envió una imagen hasta que recibió los resultados (idealmente < 500 ms para asegurar fluidez); el uso de recursos, que cuantificó el porcentaje medio y los picos máximos de CPU y memoria consumidos durante la inferencia del modelo de Computer Vision (objetivo < 70 % para prevenir ralentizaciones y sobrecalentamiento) y la escalabilidad, analizada registrando cómo variaron el tiempo de respuesta y el consumo de recursos al incrementar las búsquedas simultáneas de 1 a 5, de modo que un sistema bien escalable mostró solo un incremento moderado en ambos parámetros.

3.6 Identificación de variables

3.6.1 Variable dependiente

Rendimiento de la aplicación móvil.

3.6.2 Variable independiente

Aplicación móvil basada en Computer Vision para promocionar artesanías en la ciudad de Riobamba.

3.7 Operacionalización de variables

La Tabla 4, detalla la operacionalización de variables.

Tabla 4. Operacionalización de Variables

PROBLEMA	TEMA	OBJETIVOS	VARIABLES	CONCEPTUALIZACIÓN	DIMENSIÓN	INDICADORES
¿Cómo influirá el rendimiento de la aplicación móvil basada en Computer Vision para la identificación, clasificación y promoción de artesanías mediante búsqueda visual, en la visibilidad digital de los productos artesanales y en la experiencia de compra de artesanías en la ciudad de Riobamba?	Aplicación móvil basada en Computer Vision para promocionar artesanías en la ciudad de Riobamba.	GENERAL Implementar una aplicación móvil basada en Computer Vision para promocionar artesanías en la ciudad de Riobamba.	INDEPENDIENTE Aplicación móvil basada en Computer Vision para promocionar artesanías en la ciudad de Riobamba.	Se refiere al proceso de diseño, construcción e implementación del prototipo de una aplicación móvil en Flutter que integra módulos de Computer Vision (TensorFlow Lite) para capturar imágenes de artesanías, preprocesarlas y ejecutar la inferencia del modelo de clasificación. Este desarrollo incluye tanto la composición de la interfaz de usuario como la lógica de negocio necesaria para presentar de forma interactiva los resultados de la búsqueda visual de productos artesanales.	Aplicación móvil	•Módulos desarrollados •Líneas de código implementadas •Componentes funcionales integrados
		ESPECIFICOS •Investigar las tecnologías móviles y herramientas de Computer Vision adecuadas para el desarrollo de una aplicación destinada a la promoción de artesanías. •Desarrollar una aplicación móvil que utilice Computer Vision para identificar, clasificar y promocionar productos artesanales. •Evaluar el rendimiento de la aplicación móvil utilizando la herramienta Flutter DevTools.	DEPENDIENTE Rendimiento de la aplicación móvil.	El rendimiento de la aplicación móvil se entiende como la capacidad del prototipo para procesar búsquedas por imagen con rapidez, estabilidad y precisión. Esto implica medir el tiempo medio que tarda en mostrar resultados, el consumo de recursos (CPU y memoria) durante la inferencia y la exactitud del modelo de clasificación (accuracy, recall y F1-score). Un alto rendimiento se traduce en una experiencia fluida para el usuario y en la correcta identificación de las artesanías objetivo bajo diversas condiciones de uso.	Rendimiento de la aplicación móvil	•Tiempo de respuesta •Recursos del dispositivo •Navegabilidad

3.8 Metodología de desarrollo

El desarrollo de este prototipo móvil se basa en la metodología Mobile-D, un enfoque ágil que busca crear aplicaciones móviles de manera rápida y eficiente. Este modelo surge como respuesta a las necesidades de equipos de trabajo pequeños que enfrentan plazos ajustados y recursos limitados, enfocándose en la entrega continua de valor, la calidad del producto y la capacidad de adaptarse a los cambios.

3.8.1 Fases de la metodología

Fase de exploración

Consiste en definir el concepto general del proyecto, estableciendo su propósito, alcance, objetivos y requerimientos.
Durante esta etapa se identifican las necesidades del usuario, los posibles riesgos técnicos y las restricciones del sistema. También se realiza un análisis de factibilidad técnica y económica que permite determinar la viabilidad del desarrollo.

El resultado de esta fase es una base sólida de planificación que orienta las etapas siguientes, garantizando que el desarrollo responda a los objetivos planteados por el proyecto. A partir de la información recopilada en la fase de exploración, se elaboró la Tabla 5 de requerimientos funcionales del sistema.

Tabla 5. Requerimientos Funcionales

Tipo de Requerimiento	Descripción
Funcional	El sistema permitió el registro e inicio de sesión de los artesanos y usuarios interesados en productos artesanales. El sistema utilizó técnicas de Computer Vision para reconocer y clasificar las artesanías a partir de las imágenes cargadas por los usuarios. La aplicación fue capaz de sugerir productos similares en función de las coincidencias visuales o categorías identificadas. La aplicación permitió consultar información detallada de los artesanos. Se generaron estadísticas de interacción, tales como visualizaciones, visitas y productos más consultados por los usuarios. Se implementó un catálogo interactivo de artesanías que permitió la búsqueda por tipo, material, ubicación y rango de precios.

A partir de la información recopilada en la fase de exploración, se elaboró la Tabla 6 de requerimientos no funcionales del sistema.

Tabla 6. Requerimientos no funcionales.

Tipo de Requerimiento	Descripción
No Funcional	La aplicación presentó una interfaz visual intuitiva, estética y accesible, adaptada a distintos tamaños de pantalla. El tiempo de procesamiento de las imágenes no es alto.

	La interfaz mantuvo una identidad visual coherente con la cultura artesanal de Riobamba, utilizando paletas cromáticas y tipografías representativas. El sistema demostró un rendimiento estable, evitando sobrecargas de CPU o consumo excesivo de memoria. La base de datos admitió consultas simultáneas sin pérdida significativa de rendimiento. La arquitectura del sistema fue modular y escalable, lo que facilitó su mantenimiento y actualización futura.

Fase de diseño

En esta etapa se presentan los diferentes diagramas de ingeniería de software que permiten organizar y estructurar la solución propuesta de acuerdo con los requerimientos.

Diagrama de casos de uso

Los casos de uso se detallan en la Figura 5.

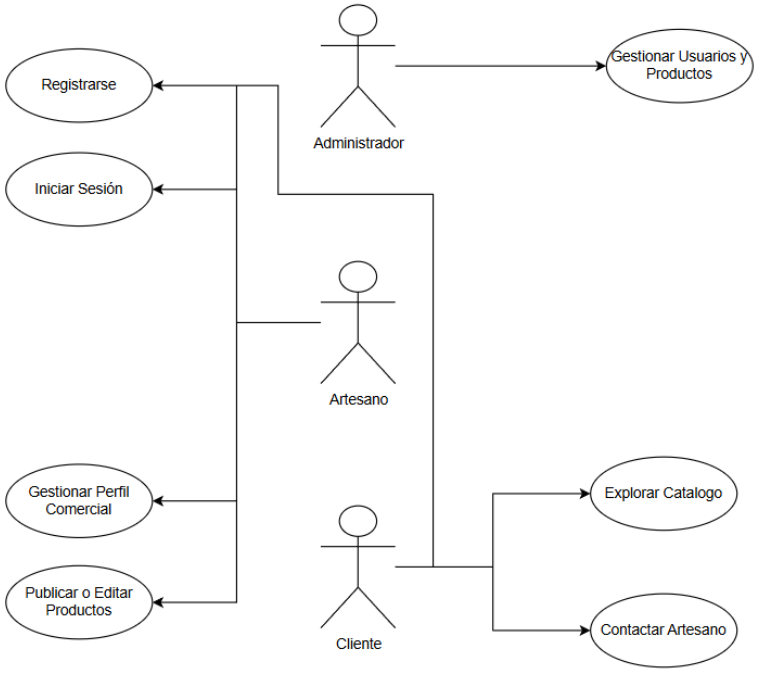


Figura 5. Diagrama de Casos de Uso

Diagrama de actividades

El diagrama de actividades se detalla en la Figura 6.

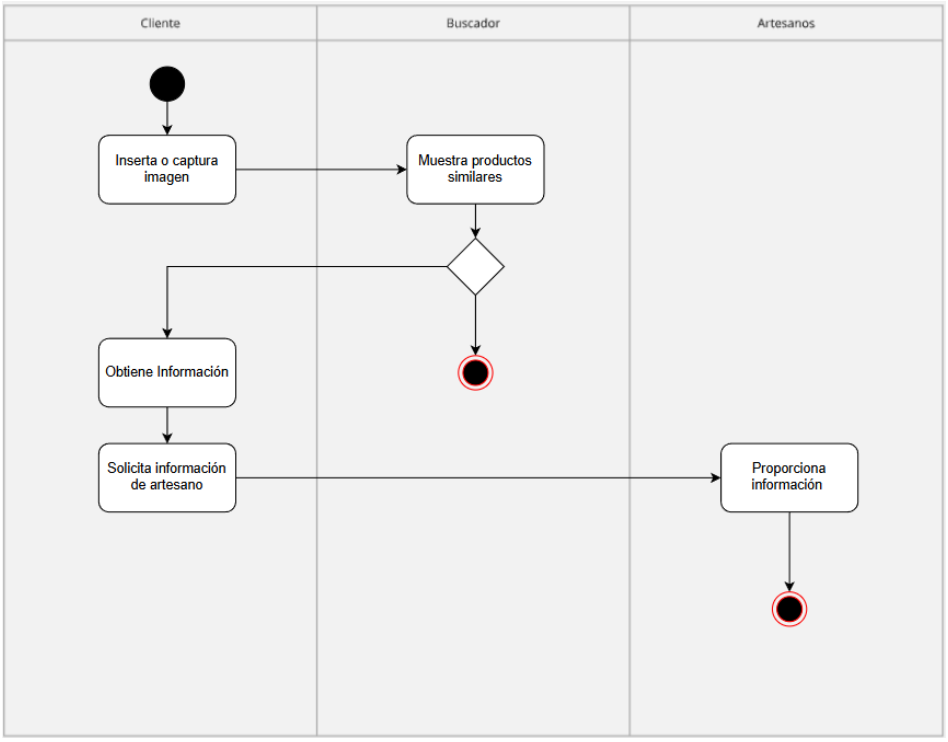


Figura 6. Diagrama de actividades

Diagrama de despliegue o arquitectura

El diagrama de arquitectura se detalla en la Figura 7.

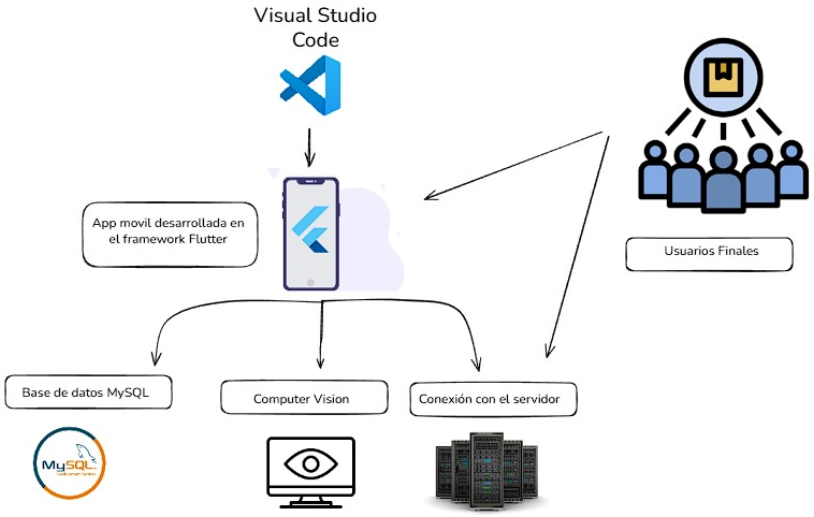


Figura 7. Diagrama de Arquitectura

Diagrama físico de la base de datos y script

El diagrama físico de la base de datos y script se detalla en la Figura 8 y 9.

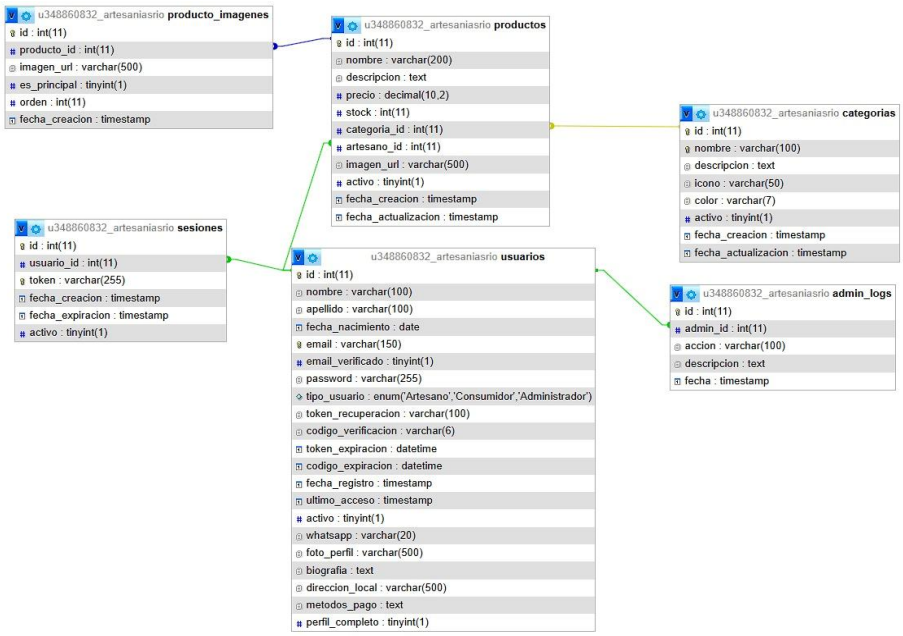


Figura 8. Diagrama Físico de Base de Datos

```
Base de datos: `app`
-- Estructura de tabla para la tabla `productos`
CREATE TABLE `productos` (
  `id` int(11) NOT NULL,
  `Nombre` varchar(255) NOT NULL COMMENT 'Nombre o Título del Producto',
  `Descripcion` text DEFAULT NULL COMMENT 'Descripción detallada del producto',
  `Tipo_de_producto` varchar(100) DEFAULT NULL COMMENT 'Categoría o Tipo de Producto',
  `Material` varchar(100) DEFAULT NULL COMMENT 'Material principal de fabricación',
  `Dimensiones` varchar(100) DEFAULT NULL COMMENT 'Medidas o Talla del Producto',
  `Precio` decimal(10,2) NOT NULL COMMENT 'Precio de venta del producto'
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
-- Estructura de tabla para la tabla `usuarios`
CREATE TABLE `usuarios` (
  `id` int(10) UNSIGNED NOT NULL,
  `nombre` varchar(100) NOT NULL,
  `apellido` varchar(100) NOT NULL,
  `correo_electronico` varchar(150) NOT NULL,
  `contrasena_hash` varchar(255) NOT NULL,
  `fecha_nacimiento` date DEFAULT NULL,
  `fecha_registro` timestamp NOT NULL DEFAULT current_timestamp(),
  `ultima_conexion` timestamp NULL DEFAULT NULL,
  `activo` tinyint(1) DEFAULT 1,
  `nivel` int(11) NOT NULL DEFAULT 3
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
-- Indices de la tabla `productos`
ALTER TABLE `productos`
  ADD PRIMARY KEY (`id`),
  ADD UNIQUE KEY `Nombre` (`Nombre`);
-- Indices de la tabla `usuarios`
ALTER TABLE `usuarios`
  ADD PRIMARY KEY (`id`),
  ADD UNIQUE KEY `correo_electronico` (`correo_electronico`);
-- AUTO_INCREMENT de la tabla `productos`
ALTER TABLE `productos`
  MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=4;
-- AUTO_INCREMENT de la tabla `usuarios`
ALTER TABLE `usuarios`
  MODIFY `id` int(10) UNSIGNED NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=12;
COMMIT;
```

Figura 9. Script Base de Datos

Diccionario de datos de la base de datos

La Tabla 7, ilustra el Admin_logs.

Tabla 7. Admin_logs

Campo	Tipo	Descripción
id	INT (PK)	Identificador único del log
admin_id	INT(11)	Identificador único de admin
accion	VARCHAR(100)	Que hace
descripcion	TEXT	Describe el error
fecha	TIMESTAMP	Fecha del log

En la Tabla 8, se observan las categorías.

Tabla 8. Categorías

Campo	Tipo	Descripción
id	INT (PK)	Identificador único del usuario
nombre	VARCHAR(100)	Nombre de la categoría
descripcion	TEXT	Descripción de la categoría
icono	VARCHAR(50)	Icono de la categoría
color	VARCHAR(7)	Color que se le asigna a la categoría
activo	TINYINT	Estado de la categoría
fecha_creacion	TIMESTAMP	Fecha de la creacion de categoría
fecha_actualizacion	TIMESTAMP	Fecha que se actualiza la categoría

La Tabla 9 resume los productos.

Tabla 9. Productos

Campo	Tipo	Descripción
id	INT (PK)	Identificador único del producto
nombre	VARCHAR(200)	Nombre del producto
descripcion	TEXT	Descripción del producto
precio	DECIMAL(10,2)	Precio del producto
stock	INT(11)	Unidades disponibles
categoria_id	INT(11)	Identificador de la categoría
artesano_id	INT(11)	Identificador del artesano
imagen_url	VARCHAR(500)	Url de las imagenes
activo	TINYINT(1)	Si el producto está activo
fecha_creacion	TIMESTAMP	Fecha de creación del producto
fecha_actualizacion	TIMESTAMP	Fecha de actualización del producto

La Tabla 10, detalla el producto imágenes.

Tabla 10. Producto Imágenes

Campo	Tipo	Descripción
id	INT (PK)	Identificador único de la imagen
producto_id	INT(11)	Identificador único del producto
imagen_url	VARCHAR(500)	Url de las imagenes
es_principal	TINYINT(1)	Elige la imagen principal
orden	INT(11)	Orden en el que se muestra la imagen
fecha_creacion	TIMESTAMP	Fecha de creación de la imagen

En la Tabla 11, se aprecian las sesiones.

Tabla 11. Sesiones

Campo	Tipo	Descripción
id	INT (PK)	Identificador único de la sesión
usuario_id	INT(11)	Identificador único del usuario
token	VARCHAR(255)	Token de usuario
fecha_creacion	TIMESTAMP	Fecha de la creacion de la sesión
fecha_expiracion	TIMESTAMP	Fecha que se expira la sesión
activo	TINYINT(1)	Estado de la sesión

En la Tabla 12, se aprecia datos de los usuarios.

Tabla 12. Usuarios

Campo	Tipo	Descripción
id	INT (PK)	Identificador único del usuario
nombre	VARCHAR(100)	Nombre de usuario
apellido	VARCHAR(100)	Apellido de usuario
fecha_nacimiento	DATE	Fecha de nacimiento
email	VARCHAR(150)	Email del usuario
email_verificado	TINYINT(1)	Estado de verificación
password	VARCHAR(255)	Contraseña del usuario
tipo_usuario	ENUM	Tipo de usuario
token_recuperacion	VARCHAR(100)	Token de recuperación de contraseña
codigo_verificacion	VARCHAR(6)	Código de recuperación de contraseña
token_expiracion	DATETIME	Expira el token
codigo_expiracion	DATETIME	Expira el código
fecha_registro	TIMESTAMP	Fecha de creación del usuario
ultimo_acceso	TIMESTAMP	Fecha del último acceso
activo	TINYINT(1)	Estado del usuario
whatsapp	VARCHAR(20)	Número de Whatsapp
foto_perfil	VARCHAR(500)	Foto de perfil del usuario
biografia	TEXT	Biografía del usuario
direccion_local	VARCHAR(500)	Dirección del artesano
metodos_pago	TEXT	Métodos de pago
perfil_completo	TINYINT(1)	Estado del perfil

Modelado de interfaz

El modelado de interfaz se muestra en la Figura 10.

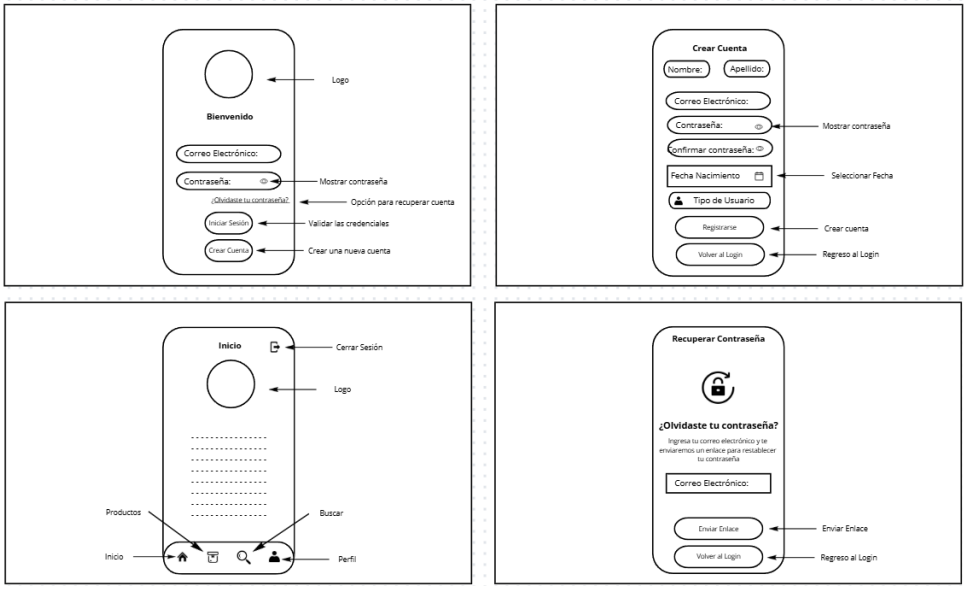


Figura 10. Modelado de Interfaz

Fase de implementación

Arquitectura

El sistema desarrollado se estructuró bajo una arquitectura modular, la cual permitió organizar sus componentes de manera clara, reducir el acoplamiento entre funcionalidades y facilitar tanto el mantenimiento como la escalabilidad del prototipo. Esta arquitectura se orientó al desarrollo de una aplicación para la gestión y detección de artesanías, integrando procesamiento inteligente de imágenes y almacenamiento estructurado de información.

Módulo 1: Aplicación cliente (Frontend – Flutter)

Este módulo correspondió a la aplicación móvil utilizada por los usuarios finales. Se implementó para permitir la visualización del catálogo de artesanías, la navegación por categorías, la consulta de información descriptiva y la interacción con contenidos visuales. La interfaz fue desarrollada con un enfoque intuitivo y responsivo, orientado a mejorar la experiencia del usuario y facilitar el acceso a la información cultural y artesanal.

Módulo 2: Backend de servicios (API)

Este módulo actuó como intermediario entre la aplicación móvil, la base de datos y los módulos de procesamiento inteligente. Se encargó de gestionar las solicitudes provenientes del frontend, validar la información, administrar la lógica de negocio y exponer servicios para el registro, consulta y actualización de los datos relacionados con las artesanías, garantizando la integridad y consistencia de la información.

Módulo 3: Módulo de detección de artesanías (YOLO – Computer Vision)

Este componente se desarrolló para el análisis automático de imágenes, utilizando técnicas de visión por computador y el modelo YOLO (You Only Look Once) para la detección de artesanías. El módulo permitió identificar objetos artesanales dentro de imágenes o flujos visuales, generando resultados estructurados que fueron enviados al backend para su almacenamiento y posterior visualización. Su implementación como módulo independiente facilitó la optimización del rendimiento y la actualización del modelo de detección.

Módulo 4: Gestión del catálogo artesanal

Este módulo se destinó a la administración de la información asociada a cada artesanía, incluyendo nombre, descripción, categoría, origen cultural, imágenes y metadatos relevantes. Su función principal fue mantener organizado el catálogo digital y permitir consultas eficientes desde la aplicación móvil.

Módulo 5: Base de datos (MySQL)

Este módulo se utilizó para el almacenamiento persistente de la información del sistema. Se diseñó bajo un enfoque relacional, permitiendo gestionar entidades como

usuarios, artesanías, categorías, resultados de detección y registros de interacción. La base de datos garantizó integridad referencial, consistencia y disponibilidad de los datos.

Lenguaje de desarrollo

Para el desarrollo del prototipo se emplearon los siguientes lenguajes y tecnologías, seleccionados por su compatibilidad, eficiencia y adecuación al contexto del proyecto:

- Dart (Flutter): fue utilizado para el desarrollo de la aplicación móvil, permitiendo la creación de interfaces multiplataforma, modernas y de alto rendimiento. Flutter facilitó la construcción de componentes reutilizables y una navegación fluida para la visualización del catálogo de artesanías.
- YOLO (You Only Look Once): fue empleado como modelo de detección de objetos para identificar artesanías en imágenes. Su arquitectura permitió realizar detecciones en tiempo real con alta precisión, siendo adecuado para aplicaciones que requieren rapidez y eficiencia en el procesamiento visual.
- MySQL: fue utilizado como sistema gestor de base de datos relacional para el almacenamiento de la información generada por la aplicación. Su uso permitió una organización estructurada de los datos y un acceso eficiente a los registros del sistema.

Patrón de diseño

MVVM (Model–View–ViewModel) es un enfoque arquitectónico que se utilizó para lograr una separación clara entre la interfaz de usuario y la lógica de la aplicación. Su aplicación permitió reducir el acoplamiento entre componentes, facilitando el mantenimiento del sistema, la realización de pruebas y la escalabilidad del proyecto a futuro.

Este patrón se compone de tres elementos fundamentales que definen su estructura y funcionamiento:

Model (Modelo): corresponde a la capa encargada de representar los datos del sistema. Contiene la información estructurada que utiliza la aplicación, sin incluir la lógica relacionada con su presentación o manipulación visual.

View (Vista): representa la interfaz de usuario. Incluye todos los elementos visuales y de interacción, como pantallas, controles, animaciones y eventos, siendo responsable de mostrar la información y captar las acciones del usuario.

ViewModel (VistaModelo): actúa como un intermediario entre la Vista y el Modelo. En esta capa se concentra la lógica de presentación, el procesamiento de datos y la comunicación con los modelos, preparando la información antes de que sea mostrada en la interfaz.

El patrón MVVM se rige por principios que garantizan una correcta separación de responsabilidades. Cada vista se encuentra asociada a un único ViewModel, el cual gestiona su comportamiento. Además, la Vista no interactúa directamente con el Modelo, sino que lo

hace exclusivamente a través del ViewModel, asegurando un flujo de información ordenado y controlado.

La implementación de este patrón permitió estructurar la aplicación de forma más clara y organizada, haciendo que el sistema resulte más comprensible, flexible y adecuado para el desarrollo de aplicaciones modernas.

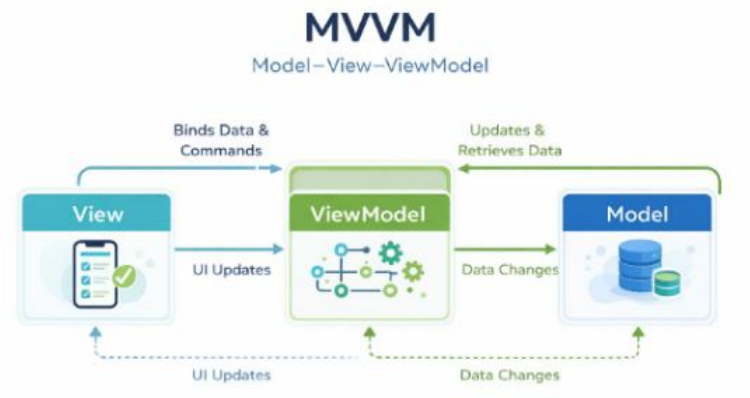


Figura 11. Modelo MVVM

Código (Backend & FrontEnd)

BackEnd

Para el backend se usó librerías de uso gratuito como son: OpenCV y FastApi. (Ver Figura 12)

```dart
// Importaciones necesarias para la pantalla de búsqueda de productos
import 'package:flutter/material.dart';
import 'package:image_picker/image_picker.dart';

// Widget de pantalla de búsqueda de productos
class SearchProductsScreen extends StatefulWidget {
  const SearchProductsScreen({super.key});
  static const String routeName = '/search-products';

  @override
  State<SearchProductsScreen> createState() => _SearchProductsScreenState();
}

class _SearchProductsScreenState extends State<SearchProductsScreen> {
  final ImagePicker _picker = ImagePicker();

  // Función para tomar foto con la cámara
  Future<void> _takePicture() async {
    try {
      final XFile? image = await _picker.pickImage(
        source: ImageSource.camera,
        imageQuality: 80,
      );
      if (image != null) {
        ScaffoldMessenger.of(context).showSnackBar(
          const SnackBar(
            content: Text('Cámara abierta correctamente'),
          ), // SnackBar
        );
      }
    } catch (e) {
      if (mounted) {
        ScaffoldMessenger.of(context).showSnackBar(
          SnackBar(content: Text('Error al tomar foto: $e')),
        );
      }
    }
```

Figura 12. Código Backend

FrontEnd

Para el frontend se utilizó el lenguaje Dart mediante el framework Flutter. (Ver Figura 13)

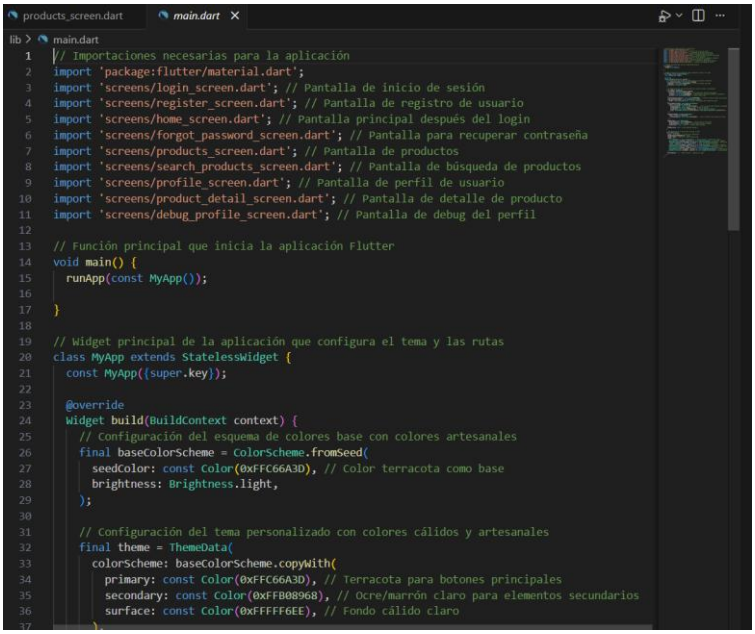


Figura 13. Frontend

Pantallas

La interfaz de usuario fue implementada mediante el kit de desarrollo Flutter, utilizando el motor de renderizado de Dart. Esto permitió la creación de vistas personalizadas y componentes visuales. La Figura 14 muestra la interfaz de login, registro e interfaz final dirigida al usuario.

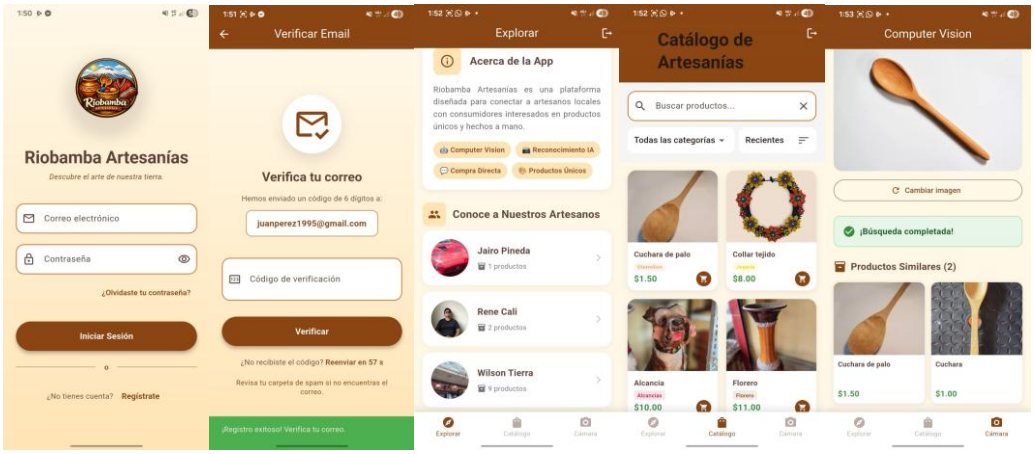


Figura 14. Aplicación Móvil

Fase de Pruebas

¿Qué se va a hacer?

En esta fase se realizarán pruebas funcionales y de compatibilidad de la aplicación móvil desarrollada, con el propósito de verificar que las funcionalidades implementadas operen correctamente conforme a los requisitos definidos. Asimismo, se evaluará el comportamiento general de la aplicación durante su ejecución, identificando posibles errores, fallos de estabilidad o inconsistencias en la interacción del usuario.

Las actividades de prueba incluirán corridas controladas de la aplicación, ejecución de casos de prueba definidos previamente y la recopilación de resultados mediante registros automáticos y observación directa del funcionamiento del sistema.

¿Cómo se va a hacer?

Las pruebas se ejecutarán mediante el uso de herramientas de software orientadas a aplicaciones móviles, principalmente Firebase Test Lab, la cual permitirá realizar pruebas automatizadas en diferentes dispositivos Android, tanto físicos como virtuales.

Los resultados se complementarán con registros generados automáticamente por la herramienta de pruebas, tales como logs, reportes de ejecución y capturas del comportamiento de la aplicación.

¿Quiénes participan?

En el proceso de pruebas participarán usuarios participantes en pruebas funcionales controladas, quienes interactuarán con la aplicación en escenarios reales de uso.

Los datos obtenidos durante esta fase corresponderán a registros automáticos de ejecución, reportes generados por la herramienta de pruebas, los cuales servirán como base para el análisis de resultados.

Pruebas unitarias

Las pruebas unitarias tuvieron como objetivo verificar el correcto funcionamiento de los componentes internos del sistema, garantizando que cada módulo opere de manera independiente antes de su integración con la aplicación móvil.

Para su implementación se utilizó el framework unittest de Python, empleado en el backend del sistema encargado del procesamiento de imágenes, reconocimiento visual y gestión de datos.

Tabla 13. Componentes evaluados en pruebas unitarias

Código	Componente probado	Descripción
PU01	procesar_imagen(imagen)	Verifica el correcto preprocesamiento de imágenes capturadas desde la app móvil

Código	Componente probado	Descripción
PU02	detectar_artesania(imagen)	Comprueba la identificación de artesanías mediante Computer Vision
PU03	clasificar_artesania(características)	Valida la clasificación correcta del tipo de artesanía
PU04	guardar_resultado(usuario, artesania)	Verifica el registro del reconocimiento en la base de datos
PU05	consultar_artesania(id_artesania)	Comprueba que la información retornada corresponda a la artesanía detectada

Estas pruebas permitieron detectar errores de lógica y validación durante el desarrollo incremental, como lo establece Mobile-D.

Pruebas funcionales

Las pruebas funcionales se orientaron a validar que la aplicación móvil cumpla correctamente con los requisitos funcionales definidos, evaluando el flujo completo desde la captura de imágenes hasta la visualización de la información de las artesanías.

Las pruebas se realizaron utilizando dispositivos móviles reales y emuladores, simulando escenarios de uso cotidiano por parte de turistas y ciudadanos. La Tabla 14, muestra las pruebas funcionales del sistema.

Tabla 14. Pruebas funcionales del sistema

Código	Funcionalidad	Resultado esperado
PF01	Captura de imagen desde la cámara	La aplicación obtiene imágenes correctamente
PF02	Reconocimiento de artesanías	El sistema identifica la artesanía detectada
PF03	Visualización de información	Se muestra descripción, origen y datos culturales
PF04	Navegación en la aplicación	La interfaz responde sin errores
PF05	Consulta de catálogo	El usuario puede explorar artesanías disponibles

Pruebas de rendimiento y eficiencia

Considerando que la aplicación está orientada a dispositivos móviles, se realizaron pruebas de rendimiento para asegurar una experiencia de usuario fluida, evaluando tiempos críticos del sistema. Las pruebas de rendimiento se resumen en la Tabla 15.

Tabla 15. Métricas para pruebas de rendimiento

Código	Métrica	Descripción	Valor esperado
PE01	Tiempo de procesamiento (ms)	Tiempo para analizar una imagen	≤ 200 ms
PE02	Latencia de respuesta (ms)	Tiempo entre captura y resultado	≤ 400 ms
PE03	Tiempo de consulta (ms)	Consulta de información de artesanías	≤ 250 ms
PE04	Tiempo de carga (s)	Carga del modelo de Computer Vision	≤ 3 s

Estas métricas se evaluaron bajo condiciones controladas, simulando el uso normal de la aplicación.

CAPÍTULO IV. RESULTADOS Y DISCUSIÓN

4.1 Resultados

Los resultados obtenidos mediante Flutter Devtools evidencian que la aplicación móvil presenta un desempeño estable y un comportamiento adecuado durante su ejecución en un dispositivo Android. No se identificaron errores críticos ni fallos de rendimiento, confirmando que el sistema cumple con los criterios básicos de estabilidad y funcionamiento definidos para la fase de pruebas de la metodología Mobile-D.

En la Figura 15, se muestra el resultado de las pruebas de rendimiento.

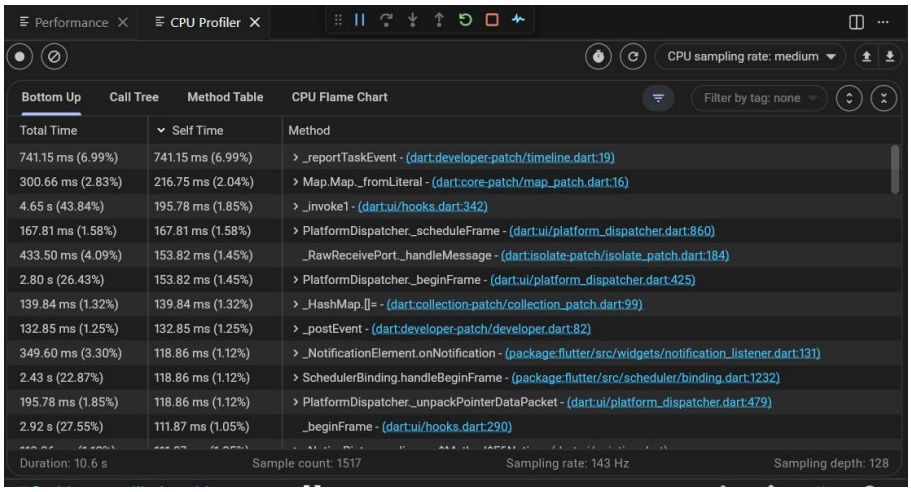


Figura 15. Prueba de rendimiento

Rendimiento del modelo

En la prueba del rendimiento se utilizaron 10 muestras. Se observa un tiempo promedio de 523.4 ms, con una mediana de 500.2 ms, demostrando que la mayoría de las ejecuciones se concentraron alrededor de los 500 ms, reflejando un comportamiento estable en el procesamiento de inferencias del modelo. El mejor caso registrado fue de 312 ms (tiempo mínimo) y el peor caso alcanzó los 980 ms (tiempo máximo), evidenciando que, aunque existen ciertas variaciones entre ejecuciones, los tiempos de respuesta se mantienen dentro de un rango aceptable. Además, la desviación estándar reportada es de 185.6 ms, valor que indica una dispersión moderada de los datos respecto al promedio, permitiendo concluir que el modelo presenta un rendimiento consistente y adecuado para su implementación en dispositivos móviles.

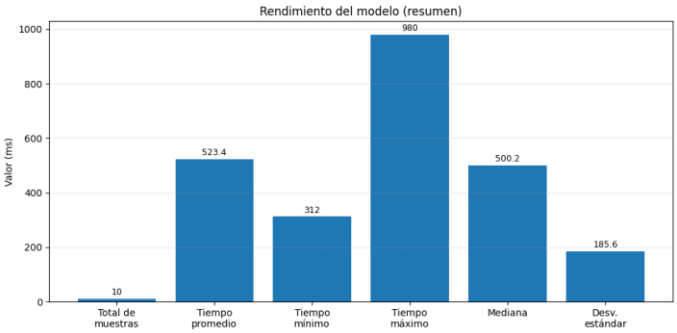


Figura 16. Rendimiento del modelo.

Desde el punto de vista del rendimiento, un tiempo de inicio inferior a 1 segundo evidencia una respuesta rápida del sistema, mejorando la experiencia del usuario y cumple con criterios de eficiencia establecidos para aplicaciones móviles modernas.

La Figura 17, presenta el tiempo de inicio de aplicación.

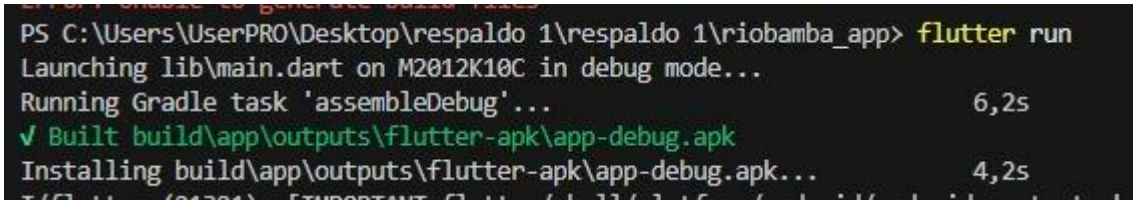


Figura 17. Tiempo de inicio de aplicación

Pruebas de efectividad del modelo de Computer Vision

Para validar el desempeño del núcleo tecnológico, se contrastaron los valores esperados definidos en la metodología con los resultados obtenidos tras el procesamiento del conjunto de imágenes de artesanías locales.

Las métricas empleadas corresponden a estándares utilizados en sistemas de reconocimiento visual. (Ver Tabla 16)

Tabla 16. Métricas de evaluación del modelo de Computer Vision

Código	Métrica	Descripción	Valor esperado	Valor obtenido
ML01	Accuracy (%)	Proporción total de clasificaciones correctas	≥ 85 %	88.5%
ML02	Precision (%)	Exactitud en la identificación de artesanías	≥ 80 %	83.2%
ML03	Recall (%)	Capacidad de detectar correctamente artesanías	≥ 80 %	81%
ML04	F1-score (%)	Equilibrio entre precision y recall	≥ 80 %	82.1%

El modelo alcanzó una exactitud del 88.5%, superando el umbral mínimo establecido. Esto indica que, en condiciones de iluminación controlada, el sistema identifica correctamente las categorías artesanales en la mayoría de las capturas.

Desempeño del modelo CLIP

En la Tabla 17, se describe el resultado de las pruebas de desempeño del modelo clip.

Tabla 17. Desempeño del modelo CLIP

Indicadores de desempeño	Valor	Unidad
Total de muestras	10	muestras
Tiempo promedio	723.8	ms
Tiempo mínimo	332	ms
Tiempo máximo	2500	ms
Mediana	522.5	ms
Desv. estándar	375.76	ms

En el desempeño del proceso de inferencia del modelo CLIP durante la búsqueda de similitudes, a partir de 10 ejecuciones, se obtuvo un tiempo promedio de 723.8 ms, con un mínimo de 332 ms y un máximo de 2 500 ms, indicando que la mayor parte de las consultas se resuelven por debajo de 1 segundo, aunque existen casos puntuales donde la inferencia se incrementa notablemente. La mediana de 522.5 ms sugiere que el comportamiento típico se mantiene cercano al medio segundo, mientras que los valores máximos reflejan picos de procesamiento que pueden depender de la carga del dispositivo, la disponibilidad de recursos y el estado de la aplicación en ese momento.

Latencia de cámara

La Tabla 18, ilustra los datos acerca de la latencia de cámara.

Tabla 18. Latencia de cámara

Indicadores de desempeño	Valor	Unidad
Total de capturas	10	capturas
Tiempo promedio	5161.2	ms
Tiempo mínimo	1651	ms
Tiempo máximo	14648	ms

La latencia de la cámara, medidos en 10 capturas. Se obtuvo un tiempo promedio de 5161.2 ms, con un mínimo de 1651 ms y un máximo de 14648 ms, demostrando una variabilidad considerable entre capturas, estos tiempos llegan a ser variables debido a las condiciones del entorno como: iluminación, texturas, movimiento. En términos prácticos, esto significa que el componente más costoso en tiempo dentro del flujo de uso (tomar foto → continuar el proceso) no es la inferencia del modelo, sino la obtención de la imagen desde la cámara, especialmente cuando el dispositivo presenta picos de carga, cambios de enfoque, procesamiento interno de la cámara o condiciones variables de iluminación.

Uso de memoria durante la clasificación

En la Tabla 19, se observa los datos sobre el uso de memoria durante la clasificación de imágenes.

Tabla 19. Uso de memoria durante clasificación

Parámetros	Valor	Unidad
Muestras	20	muestras
Promedio	414.01	MB
Pico máximo	548.97	MB

El consumo de memoria durante la ejecución del proceso de clasificación, con 20 muestras registradas. Se evidencia un promedio de 414.01 MB, alcanzando un pico máximo de 548.97 MB, esto indica que la aplicación mantiene un consumo moderado-alto de memoria en el momento de procesar imágenes y ejecutar el modelo. Este comportamiento es coherente con aplicaciones de visión por computador, ya que tanto el manejo de imágenes como el modelo y sus tensores intermedios incrementan el uso de memoria temporalmente. Además, el valor pico sugiere que en ciertos momentos (por ejemplo, al cargar la imagen completa o al ejecutar la inferencia) la app puede requerir más recursos, lo cual debe considerarse al definir los requisitos mínimos del dispositivo.

La evaluación se realizó mediante Flutter DevTools, empleando los siguientes módulos:

- Network: para medir la duración de peticiones HTTP (latencia de consumo de API).
- Memory: para observar el consumo de memoria (RSS/Heap) y su estabilidad.
- CPU Profiler: para identificar los métodos que concentran mayor tiempo de ejecución.
- Logs generados por la app para ubicar eventos clave (listar categorías/productos, búsqueda de similares y detección).

Las pruebas fueron realizadas con la herramienta Flutter devtools, aplicando un flujo de evaluación previamente definido que permitió observar el comportamiento de la aplicación durante escenarios de uso reales, como uso de CPU, uso de memoria, y la ejecución del proceso de búsqueda/detección de artesanías.

Los resultados se obtuvieron a partir de los módulos de DevTools, principalmente Network, Memory y CPU Profiler. En Network se registraron las duraciones de las solicitudes HTTP hacia el backend, esto permitió medir la latencia de consulta y carga de información. En Memory se monitoreó el consumo de memoria del proceso (RSS) y del heap de Dart/Flutter, verificando su estabilidad durante la ejecución. Finalmente, con CPU Profiler se identificaron las secciones del código y métodos que concentraron mayor tiempo de procesamiento durante la sesión perfilada, especialmente asociados al renderizado y manejo de eventos.

Rendimiento de comunicación

La Tabla 20, resume los tiempos observados en la pestaña Network para peticiones realizadas por la aplicación al servidor.

Tabla 20. Duración de solicitudes (Flutter DevTools – Network)

Tipo	Cantidad observada	Duración mínima (ms)	Duración promedio (ms)	Duración máxima (ms)	Estado
POST (JSON)	3	334	343	358	200
SOCKET (TCP)	3	223	233	249	Closed

Los resultados evidencian que las solicitudes POST se completan en un rango de 334–358 ms, con un promedio aproximado de 343 ms, reflejando una comunicación estable con el backend durante el flujo de consulta de datos.

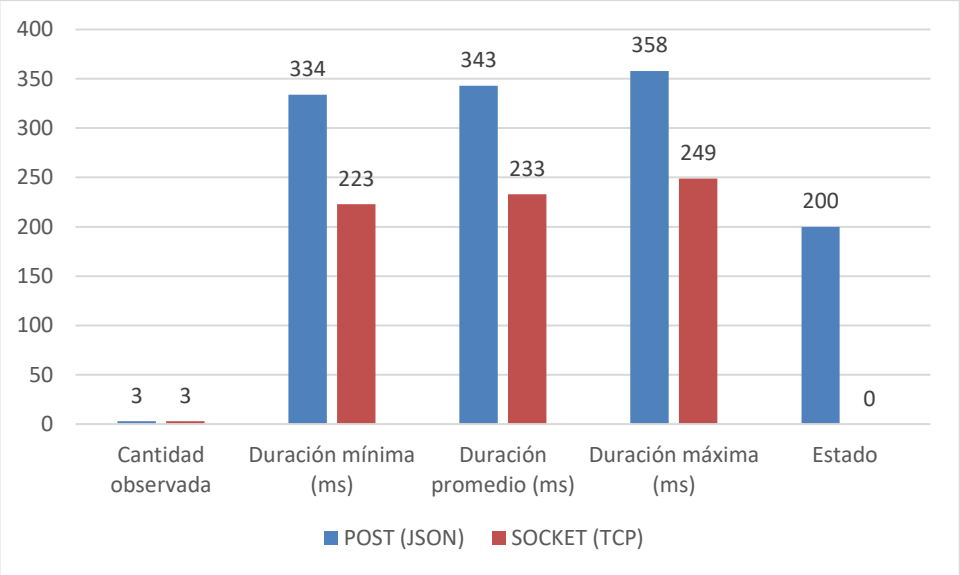


Figura 18. Tiempo de respuesta a solicitudes

Consumo de memoria

En la Tabla 21, se muestran los valores de memoria observados en Memory chart durante la ejecución de la aplicación.

Tabla 21. Consumo de memoria

Indicador de memoria	Valor observado
RSS (memoria total del proceso)	232.66 MB
Allocated	17.69 MB
Dart/Flutter (Heap)	16.03 MB
Raster Picture	0.81 MB
Dart/Flutter Native	226.8 KB

La Figura 19, ilustra los datos del consumo de memoria durante el uso de la aplicación.

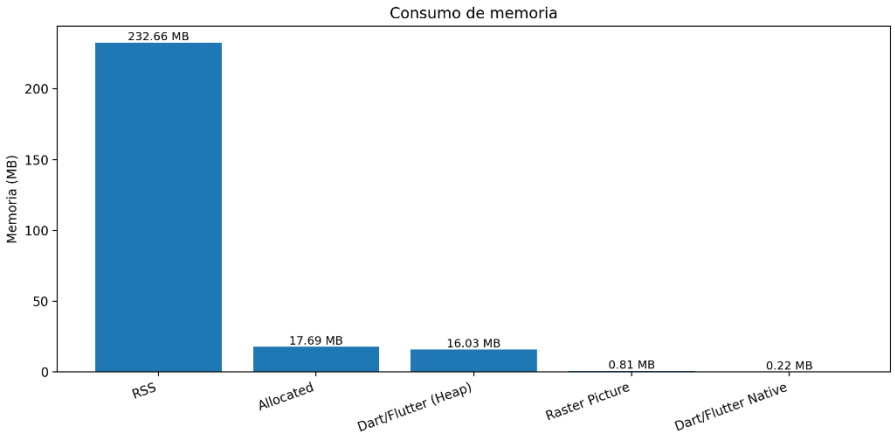


Figura 19. Consumo de memoria

La aplicación móvil registró un consumo total de memoria (RSS) de 232.66 MB, mientras que la memoria gestionada directamente por Dart/Flutter (Allocated) refleja un 17.69 MB, con un Heap de 16.03 MB. Además, la memoria asociada al renderizado (Raster/Picture) fue de 0.81 MB y el uso nativo de Dart/Flutter se mantuvo en 226.8 KB. Estos valores evidencian que la carga de memoria del código Dart se mantiene baja y controlada, demostrando un comportamiento estable durante la ejecución.

Perfil de CPU (CPU Profiler)

En la Tabla 22, se resume la sesión de perfilado capturada en CPU Profiler.

Tabla 22. Resumen de sesión (Flutter DevTools – CPU Profiler)

Indicadores	Valor
Duración perfilada	10.6 s
Sample count	1517
Sampling rate	143
Profundidad	128

Se registró una sesión de perfilado con una duración total de 10.6 segundos, en la que se capturaron 1517 muestras del comportamiento de la aplicación. La captura se realizó con una frecuencia de muestreo de 143 Hz, esto significa que el profiler tomó observaciones de la ejecución aproximadamente 143 veces por segundo, permitiendo identificar con buen detalle qué funciones consumen más tiempo de CPU. Adicionalmente, se configuró una profundidad de 128, indica que el análisis contempló hasta 128 niveles de llamadas en el árbol de ejecución (call stack), proporcionando una trazabilidad amplia para ubicar con precisión los puntos donde se concentran los costos de procesamiento.

4.2 Discusión

En las pruebas realizadas, el sistema demostró coherencia en la recomendación, porque los resultados presentados mantienen relación directa con características visibles del

producto, como forma, color, textura y elementos distintivos, confirmando que el modelo logra extraer rasgos relevantes y compararlos de manera efectiva. Sin embargo, es importante recalcar que estas pruebas se ejecutaron bajo condiciones relativamente controladas, por lo que en un entorno real el desempeño puede variar debido a factores externos como la iluminación, sombras, reflejos, calidad de la cámara, distancia de captura, desenfoque o incluso fondos con mucho “ruido” visual. Aun así, los resultados validan que el enfoque aplicado es funcional y que la aplicación puede aportar valor en la promoción de artesanías, al facilitar al usuario la búsqueda de productos relacionados de forma rápida y visual.

El sistema demostró coherencia en la recomendación mediante la extracción efectiva de rasgos como forma, color y textura. Este comportamiento valida el uso de modelos de aprendizaje profundo que, según la literatura, superan el 90% de precisión en la categorización de productos con alta variabilidad visual, como las artesanías. Sin embargo, el contraste cuantitativo de esta investigación revela que el rendimiento está condicionado por factores técnicos específicos

Los resultados obtenidos coinciden con estudios recientes en el área de Computer Vision. Investigaciones actuales basadas en aprendizaje profundo han demostrado que los modelos son capaces de extraer características visuales relevantes como forma, color y textura, mejorando significativamente el rendimiento en tareas de reconocimiento visual [18]. De igual manera, trabajos recientes evidencian que los sistemas de búsqueda visual permiten generar resultados coherentes basados en similitud entre imágenes mediante representaciones profundas [19]. Asimismo, estudios sobre reconocimiento de patrimonio cultural confirman que estas tecnologías son aplicables en contextos de objetos culturales, facilitando su clasificación y análisis digital [20]. En conjunto, estos antecedentes respaldan la validez del enfoque implementado en esta investigación.

Finalmente, si se desea mejorar aún más la precisión y la estabilidad del sistema, sería necesario ampliar el conjunto de datos con más ejemplos reales, incluir mayor variedad de ángulos y condiciones de captura, y realizar pruebas con diferentes dispositivos, de modo que el modelo se adapte mejor a escenarios cotidianos y mantenga un rendimiento consistente al escalar el uso de la aplicación.

CAPÍTULO V. CONCLUSIONES Y RECOMENDACIONES

5.1 Conclusiones

- Con base en la revisión y comparación realizada, se determinó que fue totalmente viable integrar una aplicación móvil desarrollada en Flutter con técnicas de Computer Vision, orientadas a la identificación y recomendación de productos artesanales. La investigación permitió seleccionar tecnologías que se ajustaron al prototipo por su compatibilidad, facilidad de integración y buen desempeño en dispositivos móviles, dejando claro que el análisis visual puede convertirse en una herramienta útil para apoyar la promoción de artesanías cuando se consideran condiciones básicas de captura, como buena iluminación, enfoque y un ángulo adecuado.
- Se logró desarrollar una aplicación funcional que utilizó Computer Vision para identificar y clasificar productos artesanales a partir de una fotografía y, en base a ello, mostrar opciones similares para facilitar la búsqueda del usuario. En las pruebas realizadas, el flujo principal de uso (captura de imagen, procesamiento, clasificación y visualización de resultados) se ejecutó de forma estable, demostrando que la propuesta ayuda a una búsqueda más rápida y visual.
- El prototipo demostró un rendimiento adecuado y estable, con tiempos de respuesta eficientes en las solicitudes POST, que oscilaron entre 334 y 358 ms, con un promedio de 343 ms. El modelo presentó un tiempo promedio de 723.8 ms y una mediana de 522.5 ms, demostrando que el 50% de las ejecuciones fueron más rápidas que el promedio, aunque se registraron picos de hasta 2500 ms. El consumo de memoria se mantuvo controlado, con un RSS total de 232.66 MB, de los cuales 17.69 MB (7.6%) correspondieron a Dart/Flutter. Estos resultados confirman un desempeño aceptable, aunque las variaciones observadas sugieren la necesidad de optimizar el procesamiento de imágenes y la carga de recursos para lograr mayor consistencia en los tiempos de respuesta.

5.2 Recomendaciones

- Ampliar el estudio con un conjunto de datos más grande y variado de artesanías, incluyendo fotos tomadas en condiciones reales, porque eso habría permitido validar mejor la robustez del modelo y reducir sesgos por escenarios “ideales”.
- Optimizar el flujo de procesamiento de imágenes para evitar cargas innecesarias: comprimir o redimensionar imágenes antes de analizarlas, usar caché para resultados recientes y controlar mejor la carga de listas cuando se muestran muchos productos similares.
- Fortalecer la arquitectura de la aplicación mediante una separación clara entre el módulo de Computer Vision y el módulo de interfaz de usuario, siguiendo principios de modularidad y bajo acoplamiento. Esto permitirá mejorar la escalabilidad, mantenibilidad y facilidad de actualización del sistema.

BIBLIOGRAFÍA

[1] Y. Chen, «Deep Learning-Driven Craft Design: Integrating AI Into Traditional Handicraft Creation,» *IEEE Access,* vol. 13, pp. 111790-111805, 2025.

[2] L. M. L. Hollard, «LeYOLO: New Embedded Architecture for Efficient Object Detection on Mobile Devices,» *Proc. Conf. Robots and Vision (CRV),* pp. 45-53, 2025.

[3] L. Nicolás, «¿Qué es Computer Vision o Visión por Computadora?».

[4] Xeridia, «Xeridia,» 6 Mayo 2019. [En línea]. Available: https://www.xeridia.com/blog/la-vision-artificial-y-el-procesamiento-de-imagenes.

[5] K. y. L. B. Grauman, «Visual Ob-ject Recognition,» *School of Computing* , 2023.

[6] MathWorks, «MathWorks,» [En línea]. Available: https://la.mathworks.com/discovery/computer-vision.html. [Último acceso: 2025].

[7] NVIDIA, «Nvidia,» 2023. [En línea]. Available: https://www.nvidia.com/en-us/glossary/computer-vision/.

[8] G. Bradski, «opencv,» 2025. [En línea]. Available: https://opencv.org/.

[9] F. Alvi, «opencv,» 11 Octubre 2023. [En línea]. Available: https://opencv.org/blog/learning-opencv/.

[10] «TensorFlow,» [En línea]. Available: https://www.tensorflow.org/lite/guide?hl=es-419. [Último acceso: 2025].

[11] S. Vergara, «ItDo,» [En línea]. Available: https://www.itdo.com/blog/frameworks-de-desarrollo-de-aplicaciones-moviles-apps-que-dominaran-el-2024/. [Último acceso: 2025].

[12] Google Developers, «Flutter Docs,» 2024. [En línea]. Available: https://flutter.dev/docs. [Último acceso: 2025].

[13] Flutter Docs, «Flutter Docs,» 28 Octubre 2025. [En línea]. Available: https://docs.flutter.dev/ui/adaptive-responsive.

[14] Flutter Docs, «Flutter Docs,» [En línea]. Available: https://docs.flutter.dev/perf. [Último acceso: 2025].

[15] Flutter, «Flutter,» [En línea]. [Último acceso: 2025].

[16] P. J. Moreno Vallejo, G. K. Bastidas Guacho y P. R. Moreno Costales, Fundamentos de la inteligencia artificial: una vision introductoria, La Plata: Puerto Madero, 2024.

[17] P. Buñay, «APLICACIÓN DE LA METODOLOGÍA MOBILE-D EN EL DESARROLLO DE,» Riobamba, 2020.

[18] A. Dosovitskiy, «An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale,» *International Conference on Learning Representations (ICLR)*, 2021.

[19] M. Radford, «Learning Transferable Visual Models From Natural Language Supervision,» *ICML*, 2021.

[20] S. Fiorucci, «Machine Learning for Cultural Heritage: A Survey,» *Pattern Recognition Letters*, vol. 133, pp. 102-108, 2020.

[21] ISO25000, «Usabilidad,» 2021. [En línea]. Available: https://iso25000.com/index.php/normas-iso-25000/iso-25010/23-usabilidad.

[22] Xeridia, «xeridia,» 6 Mayo 2019. [En línea]. Available: https://www.xeridia.com/blog/la-vision-artificial-y-el-procesamiento-de-imagenes.

ANEXOS

MANUAL DE USUARIO

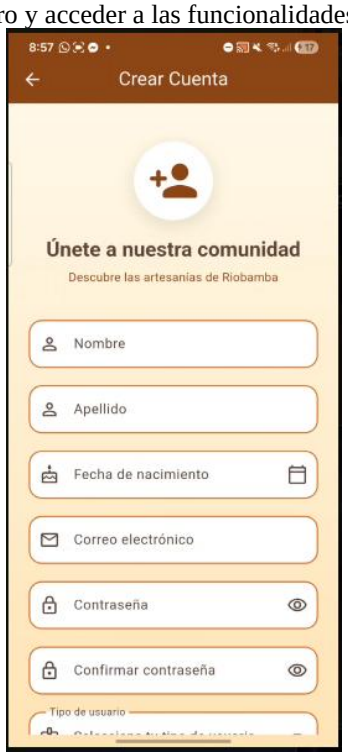
Modulo principal

1. **Pantalla de inicio de sesión**

   Para acceder al sistema, el usuario debe ingresar sus credenciales en el formulario de inicio de sesión, como se muestra en la imagen. Es necesario introducir un correo electrónico válido y la contraseña previamente registrada. En caso de olvidar la contraseña, el sistema ofrece la opción de recuperación mediante el enlace correspondiente. Una vez completados los campos, el usuario debe presionar el botón “Iniciar Sesión” para acceder a la plataforma y gestionar sus productos o información personal.

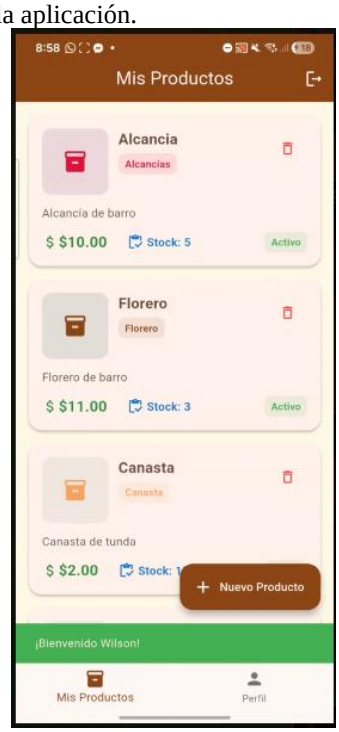
2. **Pantalla de registro (Crear cuenta)**

   Para registrarse en el sistema, el usuario debe completar el formulario de creación de cuenta como se observa en la imagen. Debe ingresar datos personales como nombre, apellido, fecha de nacimiento, correo electrónico y una contraseña segura, la cual debe ser confirmada. Estos campos son obligatorios para garantizar la correcta identificación del usuario. Una vez completada la información requerida, el usuario podrá finalizar el registro y acceder a las funcionalidades del sistema.



Módulo Artesano

3. **Pantalla de "Mis Productos"**
   En esta sección, el usuario puede visualizar el listado de productos registrados en el sistema, como se muestra en la imagen. Cada producto incluye información relevante como nombre, categoría, precio, cantidad en stock y estado (activo o inactivo). Además, el usuario dispone de opciones para eliminar productos o agregar nuevos mediante el botón "Nuevo Producto". Esta interfaz permite una gestión eficiente del inventario dentro de la aplicación.



4. **Pantalla de edición de producto**
   Para modificar la información de un producto, el usuario puede acceder al formulario de edición, tal como se muestra en la imagen. En esta sección es posible actualizar datos como el stock, la categoría y el estado del producto (activo o inactivo). Asimismo, se pueden gestionar las imágenes asociadas, permitiendo agregar o eliminar fotografías del producto. Una vez realizados los cambios necesarios, el usuario debe presionar el botón "Guardar" para actualizar la información en el sistema.

5. **Pantalla de Perfil del Artesano**

Para gestionar su identidad dentro de la plataforma, el artesano cuenta con una sección de perfil personal. En esta pantalla, el usuario puede actualizar su información básica, incluyendo una breve descripción del emprendimiento, su número de contacto de WhatsApp y la dirección física del local o punto de venta. Además, permite configurar los métodos de pago aceptados, asegurando que los clientes tengan claridad sobre las opciones de transacción disponibles.

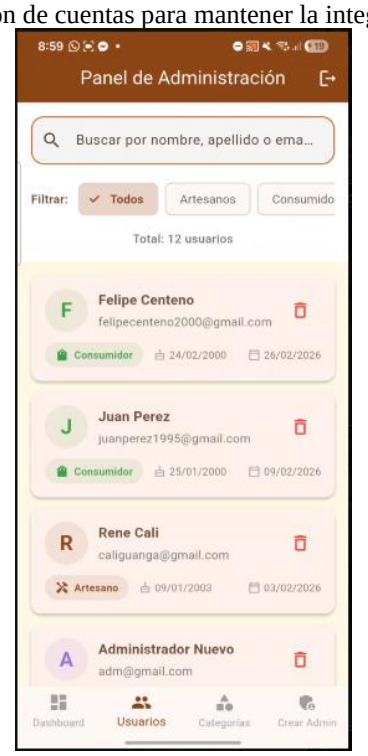
Módulo Administrador

6. **Panel de Administración (Dashboard)**

El sistema cuenta con un centro de control integral para el administrador, donde se visualizan métricas clave del ecosistema en tiempo real. Esta sección presenta indicadores cuantitativos sobre el total de usuarios, registros nuevos en la última semana, usuarios activos en el último mes y la cantidad de artesanos vinculados. Incluye un resumen visual mediante gráficos de barras que facilitan la interpretación del crecimiento y la actividad de la plataforma.

7. **Gestión de Usuarios**

Desde el panel de administración, es posible acceder al listado completo de usuarios registrados, segmentados por roles como "Consumidor" o "Artesano". Esta pantalla ofrece herramientas de búsqueda avanzada por nombre o correo electrónico, así como filtros específicos por categoría. El administrador tiene la facultad de supervisar la fecha de registro, la última actividad y, de ser necesario, gestionar la eliminación de cuentas para mantener la integridad de la base de datos.



8. **Administración de Categorías de Productos**

Para mantener el catálogo organizado, la aplicación dispone de un módulo de gestión de categorías. En esta sección, el administrador puede visualizar las categorías existentes (como Alcancías, Bordados, Cuero, entre otras) representadas por iconos y descripciones breves. Asimismo, permite la creación de nuevas categorías mediante un botón de acción rápida y la eliminación de aquellas que ya no sean requeridas para la clasificación de las artesanías.

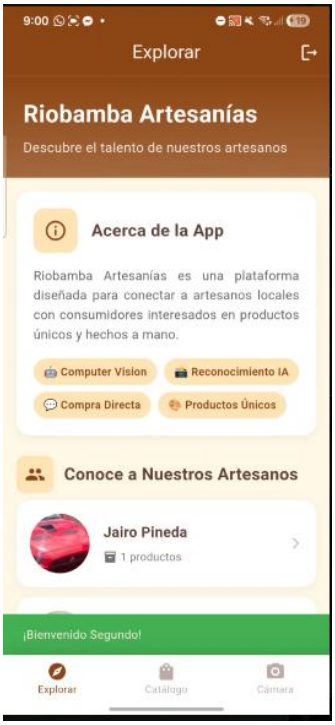
9. **Registro de Nuevos Administradores**

La seguridad y el control de acceso se gestionan mediante una interfaz dedicada a la creación de cuentas administrativas. En este formulario, los administradores actuales pueden dar de alta a nuevos gestores ingresando un correo electrónico institucional y asignando una contraseña segura. Esta funcionalidad garantiza que el acceso al panel de control esté restringido únicamente a personal autorizado.

Módulo de usuario

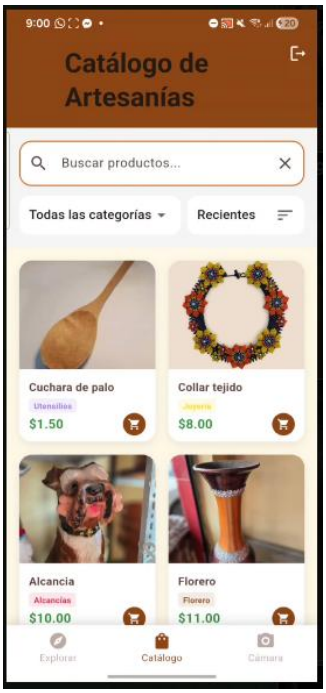
10. **Pantalla Principal Usuario: Explorar**

La interfaz de bienvenida para los usuarios finales está diseñada para resaltar el talento local. Esta sección ofrece una introducción a la misión de la plataforma y destaca las capacidades tecnológicas integradas, como el reconocimiento mediante IA y Computer Vision. Además, presenta una sección de "Artesanos destacados" que permite a los consumidores acceder directamente al catálogo específico de sus creadores favoritos.



11. Catálogo General de Artesanías

El catálogo interactivo permite a los clientes navegar por la diversidad de productos disponibles. Los artículos se presentan en una cuadrícula con imágenes de alta calidad, nombres descriptivos, categorías y precios. La pantalla integra una barra de búsqueda y filtros por "Recientes" o categorías específicas, junto con un acceso rápido al carrito de compras para agilizar el proceso de adquisición directa.



12. Búsqueda por Similitud Visual (Computer Vision)

Una de las innovaciones de la aplicación es el módulo de Computer Vision, que permite encontrar productos mediante el análisis de imágenes. El usuario puede capturar una fotografía en tiempo real o subir un archivo desde la galería. El sistema inicia un proceso de análisis automático para identificar patrones, formas y texturas del objeto fotografiado