

UNIVERSIDAD POLITÉCNICA SALESIANA

SEDE QUITO

CARRERA:

COMPUTACIÓN

Trabajo de titulación previo a la obtención del título de:

Ingenieros en Ciencias de la Computación

TEMA:

**“CONSTRUCCIÓN DE UNA APLICACIÓN MÓVIL CON REDES
NEURONALES CONVOLUCIONALES PARA RECONOCER IMÁGENES DE
PRENDAS DE VESTIR DE TIENDAS EN LÍNEA EN ECUADOR”**

AUTORES:

DANIELA MHAITE GONZÁLEZ COLLAGUAZO

JULIO CÉSAR SARÁUZ RIVADENEIRA

TUTOR:

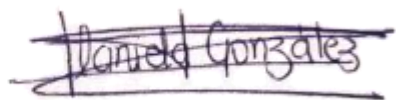
DANIEL GIOVANNY DÍAZ ORTIZ

Quito, noviembre del 2021

CESIÓN DE DERECHOS DE AUTOR

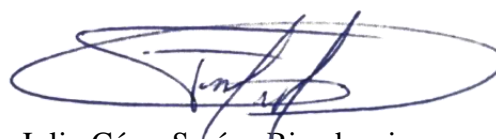
Nosotros, Daniela Mhaite González Collaguazo, Julio César Saráuz Rivadeneira, con documento de identificación N° 1726428723 y N° 1750051318, manifestamos nuestra voluntad y cedemos a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que somos autores del trabajo de titulación con el tema: “Construcción de una aplicación móvil con redes neuronales convolucionales para reconocer imágenes de prendas de vestir de tiendas en línea en Ecuador”, mismo que ha sido desarrollado para optar por el título de INGENIEROS EN CIENCIAS DE LA COMPUTACIÓN en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En aplicación a lo determinado en la Ley de Propiedad Intelectual, en nuestra condición de autores nos reservamos los derechos morales de la obra antes citada. En concordancia, suscribimos este documento en el momento que hacemos la entrega del trabajo final en formato digital a la Biblioteca de la Universidad Politécnica Salesiana.



Daniela Mhaite González Collaguazo

1726428723



Julio César Saráuz Rivadeneira

1750051318

Quito, noviembre 2021

DECLARATORIA DE COAUTORÍA DEL DOCENTE TUTOR

Yo declaro que bajo mi dirección y asesoría fue desarrollado el Proyecto Técnico, con el tema: “CONSTRUCCIÓN DE UNA APLICACIÓN MÓVIL CON REDES NEURONALES CONVOLUCIONALES PARA RECONOCER IMÁGENES DE PRENDAS DE VESTIR DE TIENDAS EN LÍNEA EN ECUADOR”., realizado por Daniela Mhaite González Collaguazo y Julio César Saráuz Rivadeneira, obteniendo un producto que cumple con todos los requisitos por la Universidad Politécnica Salesiana para ser considerado como trabajo final de titulación.

Quito, noviembre 2021



Daniel Giovanni Díaz Ortiz

CI: 1716975501

DEDICATORIA

De Daniela González

Dedico este trabajo de titulación: A mis padres Jorge y Edith quien con su apoyo incondicional, amor y esfuerzo me han permitido cumplir una meta más en mi vida, por enseñarme a nunca decaer y continuar hasta el final; a mi hermano que me ha dado esa fortaleza que la necesite en todo momento. Y a toda mi familia, amigos por sus ánimos y apoyo a lo largo de estos años de la universidad.

De Julio Saráuz

Dedico con todo mi corazón y esfuerzo mi tesis a mi madre, pues sin ella no habría tenido la oportunidad de hoy presentar este logro y a mi hermana, base fundamental para que haya llegado hasta este punto. Gracias a su apoyo emocional y financiero que supieron entregarme sin ningún interés mayor. Por eso les doy mi trabajo en ofrenda por su paciencia y amor de madre y fraternal respectivamente, las aprecio y amo madre, hermana mía.

AGRADECIMIENTO

Agradecemos a la Universidad Politécnica Salesiana y a todos los docentes por brindarnos la oportunidad de culminar con nuestra carrera, por ser esa guía de enseñanzas y compartir sus conocimientos con todos los estudiantes, por contribuir en nuestra formación profesional y personal.

También a nuestros familiares y amigos quienes nos ayudaron y apoyaron en esta etapa de crecimiento profesional.

Y un agradecimiento especial a nuestro tutor el ingeniero Daniel Díaz por su apoyo, para la culminación de este presente trabajo de titulación.

Daniela Mhaite González Collaguazo

Julio César Saráuz Rivadeneira

ÍNDICE

Antecedentes	3
Planteamiento del problema	4
Objetivos	7
Objetivo General	7
Objetivos específicos.....	7
Marco metodológico	7
CAPÍTULO 1	9
Estado del arte	9
1. MARCO TEÓRICO	9
1.1 Redes Neuronales Convulcionales	9
1.2.1 Visual Code	11
1.3 Lenguajes de programación.....	12
1.3.1 Angular.....	12
1.3.2 Python.....	13
1.3.3 Node Js	14
1.3.4 Ionic.....	15
1.3.5 Nest Js	16
1.4 Plataformas móviles	17
1.4.1 Android.....	17
- Mediante el uso de Android se graba la aplicación en un archivo apk, que brinde la oportunidad de instalar en todos los dispositivos Android compatibles. Es una plataforma	

abierta está basado en Linux, ya que ayuda con varios servicios incorporados como la localización GPS, bases de datos, sensores.	17
- Android tiene varios componentes que nos ayudan para solicitar la función de la cámara del usuario, lo que nos va servir para guardar en nuestra base de datos y poder buscar prendas similares.....	17
1.4.2 IOS	17
1.5 Metodologías de desarrollo	18
1.5.1 Metodología Scrum	18
La metodología Scrum es un marco de trabajo o framework que se utiliza dentro de equipos que desean dar origen a grandes y complejos proyectos. Esto quiere decir, que se trata de una metodología ágil que siempre busca entregar en periodos muy cortos de tiempo sus proyectos basándose en los principios de transparencia, inspección y adaptación. Esto abre paso a que el cliente puede interactuar con el equipo, además de ingresar el producto al mercado lo más pronto posible y comenzar a generar ventas (WAM, 2021).	18
1.5.2 Roles.....	19
1.5.3 Cronograma de Actividades realizadas en el proyecto	20
1.5.4 Sprints.....	22
1.5.4.1 Sprint 1	22
CAPÍTULO 2	27
2. REQUERIMIENTOS FUNCIONALES Y NO FUNCIONALES.....	27
2.1 Requerimientos funcionales	27
2.2 Requerimientos no funcionales	28

2.3 ARTEFACTOS UML	29
2.3.1 Historias de usuario	29
2.3.2 Diagrama de caso de uso	34
2.3.3 Diagrama Secuencias	37
2.3.4 Prototipo Inicial	42
2.3.5 Prototipo Final	44
2.3.4 Diagrama conceptual de la base de datos	46
CAPÍTULO 3	47
3.1 ARQUITECTURA	47
3.2 DIAGRAMA DE DESPLIEGUE	48
3.3 PRUEBAS: CAJA NEGRA, CARGA Y STRESS	48
3.4.1 Pruebas de Caja Negra	48
3.4.2 Prueba De Carga.....	55
3.4.3 Prueba De Stress.....	58
3.5 ANÁLISIS DE RESULTADOS	61
3.5.1 Prueba de Carga	61
3.5.2 Prueba de Stress	61
3.5.3 Pruebas de Aceptación de la aplicación	61
CONCLUSIONES	69
RECOMENDACIONES	70
REFERENCIAS	71

ÍNDICE DE TABLAS

Tabla 1.....	8
Tabla 2.....	19
Tabla 3.....	19
Tabla 4.....	20
Tabla 5.....	20
Tabla 6.....	27
Tabla 7.....	28
Tabla 8.....	29
Tabla 9.....	30
Tabla 10.....	31
Tabla 11.....	32
Tabla 12.....	33
Tabla 13.....	49
Tabla 14.....	56
Tabla 15.....	57
Tabla 16.....	58
Tabla 17.....	59
Tabla 18.....	60

ÍNDICE DE FIGURAS

Figura 1	22
Figura 2	23
Figura 3	24
Figura 4	24
Figura 5	25
Figura 6	26
Figura 7	26
Figura 8	34
Figura 9	34
Figura 10	35
Figura 11	36
Figura 12	36
Figura 13	37
Figura 14	38
Figura 15	39
Figura 16	40
Figura 17	41
Figura 18	42
Figura 19	43
Figura 20	43
Figura 21	44
Figura 22	45

Figura 23	45
Figura 24	46
Figura 25	47
Figura 26	48
Figura 27	55
Figura 28	56
Figura 29	57
Figura 30	59
Figura 31	59
Figura 32	62
Figura 33	62
Figura 34	63
Figura 35	63
Figura 36	64
Figura 37	64
Figura 38	65
Figura 39	65
Figura 40	66
Figura 41	66
Figura 42	67
Figura 43	67
Figura 44	68

RESUMEN

Este proyecto de titulación tiene como objetivo la construcción de una aplicación móvil con redes neuronales convolucionales para reconocer imágenes de prendas de vestir de tiendas en línea en Ecuador, utilizando la metodología de desarrollo ágil SCRUM.

Por este motivo, se desarrolló una aplicación con redes neuronales del dataset FASHION MNIST, con los servicios de Flask y Nest Js, el modelo de una base no relacional en Mongo DB y el framework en Ionic.

El producto final son dos APK del sistema operativo Android e iOS, que permite el registro de un administrador para realizar el ingreso de urls que se desea scrapear las imágenes que se encuentran en las páginas web de las tiendas de Ecuador y guardarlas en una base de datos. Y para los otros usuarios tomar y subir foto para poder predecir prendas similares de las que desean buscar.

En la sección de pruebas de caja negra se comprueba que se llegó a cumplir con los requerimientos funcionales en cada una de las fases planificadas, con las pruebas de carga se identificó la cantidad de peticiones que la aplicación puede soportar se hizo desde el mínimo 100 que nos indica 0% de errores y como máximo 10000 con un 4,13% de error y la prueba stress consiste en probar los límites que la aplicación soporta probando como máximo 100000 peticiones la aplicación no funciona de manera adecuada las solicitudes enviadas.

ABSTRACT

The objective of this degree project is to build a mobile application with convolutional neural networks to recognize images of clothing from online stores in Ecuador, using the agile development methodology SCRUM.

For this reason, an application was developed with neural networks of the FASHION MNIST dataset, with the services of Flask and Nest Js, the model of a non-relational base in Mongo DB and the framework in Ionic.

The final product are two APK of the Android and iOS operating system, which allows the registration of an administrator to make the entry of urls that you want to scrape the images found on the web pages of the stores in Ecuador and save them in a database. And for other users to take and upload photo to be able to predict similar garments of the ones they want to search.

In the black box testing section it is verified that the functional requirements were met in each of the planned phases, with the load tests the amount of requests that the application can support was identified from a minimum of 100 which indicates 0% errors and a maximum of 10000 with a 4.13% error rate and the stress test consists of testing the limits that the application supports by testing a maximum of 100000 requests the application does not work properly the requests sent.

INTRODUCCIÓN

Antecedentes

(García, 2012) su estudio tuvo como objetivo general el desarrollo de un sistema de reconocimiento de imágenes para dispositivos móviles, en modo local, y se adapta a cualquier tipo de imagen que se quiera identificar. Para realizar el estudio usó 25 imágenes de hojas de 12 tipos de plantas distintas. De ellas escogió 18 fotos, el 80% para crear el grupo de entrenamiento de la red neuronal, y las otras 7 que forman el 20% para el conjunto de test. En los resultados se demostró que, si funcionaba el programa, pero los momentos de Hu no dieron buenos resultados para una correcta clasificación de las plantas, además que demostraron que la infraestructura no era adecuada y que no se tuvo una base de datos con muchas imágenes de la misma planta, para que pueda así reconocer las diferentes características de cada tipo de planta.

(Asensio, 2016) su objetivo general es la evaluación del desempeño de una Red Neuronal Convolutiva. Se implementó varios flujos de trabajo, usando redes neuronales convolucionales, usaron la arquitectura de la red AlexNet descargada de internet y pre entrenada con 23 capas y la otra es implementada desde cero con 25 capas. Como resultados se observó que AlexNet produce mejores mapas de características. En conclusión, mientras más imágenes tengan la red para entrenarse, más características van a aprender. De modo que, el conjunto de imágenes de entrenamiento son importantes para el correcto funcionamiento de la red ante un problema de aprendizaje supervisado.

(Suárez, 2017) el objetivo de este proyecto es el uso de redes neuronales convolucionales para el desarrollo de un algoritmo en el lenguaje de programación R que posteriormente al ser entrenado con una base de datos robusta, es capaz de reconocer caracteres escritos a mano, usaron números del 0 al 9. Se buscó conseguir un porcentaje superior al 95%, con un conjunto de 10000

imágenes de comprobación. En conclusión, se demostró la prontitud con la que la red es entrenada y testeada, tardando una 1 hora en un computador de gama media, la red ha obtenido un 96% de acierto con un set de entrenamiento de 10.000 imágenes, de las cuales 5.000 estaban distorsionadas. Lo cual evidencia un buen comportamiento del lenguaje de programación escogido, R.

Planteamiento del problema

En base al portal del Grupo Bit hoy en día más de 4 millones de personas se conectan a Internet formando así el 53% de la población mundial. El (96.6%) lo hace mediante los dispositivos móviles. Y un 85% se conectan diariamente, invirtiendo en promedio 6 horas y 30 minutos conectados. Actualmente, ha cambiado la dinámica de vida de las personas, una de ellas comprar. Gracias al internet podemos realizar compras en la comodidad de nuestro hogar o en cualquier parte del mundo. Según el estudio sobre e-commerce del IAB Spain de 2016, casi 16 millones de usuarios de 16 a 60 años compran online (Grupo Bit, 2018).

Según Datta Business Innovation(2020), “Un estudio realizado por Mercado Libre, reveló que, desde la llegada del coronavirus a la región, 1,7 millones de personas se han convertido en nuevos consumidores digitales”. El reporte fue realizado con información de Argentina, Brasil, Chile, Colombia y México; con la llegada del Covid-19 cambió los hábitos del consumo en las personas, ya que los usuarios prefieren comprar online. “Hemos visto un cambio significativo en el comportamiento de los consumidores latinoamericanos durante este periodo de aislamiento social. Las categorías que antes eran demandadas como los celulares, han perdido parte de su relevancia y los usuarios están volcando sus compras a ítems más significativos en esta pandemia, como productos de salud, alimentos y artículos de entretenimiento”, lo afirmó el Director de Mercado Libre, Alan Meyer (Cornieles, 2020).

En el Ecuador algunos negocios han implementado su tienda en línea como son OLX, MercadoLibre, Yaesta.com. Pero no obtuvieron el impulso requerido ya que hay una gran cantidad de usuarios que no disponen de internet en sus celulares. El perfil de los usuarios compradores en línea se encuentran en promedio de edades entre los 18 y 45 años. Uno de los portales que ha tenido éxito es De Prati que comercializa ropa, artículos para el hogar y productos de tecnología. Los obstáculos que tiene Ecuador en el comercio electrónico es lograr una mayor penetración de internet a nivel nacional. Nuestro país se encuentra en las etapas iniciales del comercio electrónico, es importante explotar esta herramienta informática para internacionalizar la producción ecuatoriana y mejorar las exportaciones (Cruz, 2017).

Según una encuesta global sobre el comercio conectado de Nielsen en 2018, las categorías más destacadas en las compras en línea son moda 61%, viajes 59%, libros/música 49%, tecnología 47% y boletos para eventos 45%. Pero el problema es cómo llegar a los consumidores no necesariamente para realizar una compra, sino como descubrir nuevas tiendas o artículos que le interesen (THE NIELSEN COMPANY, 2018). Por esta razón es importante conocer cierta información sobre los usuarios para saber cómo atender de manera más efectiva sus necesidades, ya que hay ciertos productos que se les hace más rápido comprar como son productos de belleza, para mascotas o accesorios de hogar, y los productos que son más lentos en buscar son productos para bebés, moda infantil, jardín, moda, zapatos (E-SHOW MAGAZINE, 2019).

Los motivos por lo que nuestra propuesta es necesaria son porque la búsqueda de Google con palabras no ayuda a reconocer el tipo de prenda o color que deseamos comprar. Google con su herramienta Google Lens, ha desarrollado algoritmos basados en el contenido de formas y analizando productos similares, con tomar una foto se puede tener más información sobre ese objeto. Esta aplicación sirve para usuarios Android, y no es compatible en todos los dispositivos

móviles, ya que requiere Android 6.0 o superior para que funcione. Mientras tanto con los dispositivos iOS se debe usar como una función integrada dentro de la aplicación principal de Google para iOS, ya que no dispone de una aplicación independiente. Por lo tanto los usuarios quedan insatisfechos, en la búsqueda de las prendas que desean comprar, prefiriendo así ir a un lugar físico; ya que no existe una aplicación accesible para todos los dispositivos móviles con un motor de búsqueda de imágenes.

Dada la problemática citada anteriormente, el avance de las tecnologías de la información y el acceso más frecuente a internet facilitaron el crecimiento de los comercios electrónicos. Por esto es la necesidad de implementar un sistema que facilite la búsqueda de prendas de vestir en las diferentes tiendas en línea que hay en el Ecuador.

El sistema propuesto realizará la extracción de imágenes de distintos sitios web aplicando librerías de NodeJs, con la implementación de redes neuronales convolucionales para el reconocimiento de imágenes, de esta forma identificando las prendas de vestir y su color.

Buscamos mejorar la experiencia de los usuarios al momento de querer adquirir un producto de manera online, ya que actualmente el procedimiento que realizamos al buscar un producto es escribiendo lo que necesitamos encontrar y muchas veces aparecen en el navegador productos que no buscamos o de tiendas de otros países. Y de esta manera ocasionando pérdida de interés en los usuarios al comprar en los diferentes eCommerce que hay en Ecuador.

Uno de los principales factores principales nace de las necesidades del usuario que pretende realizar una búsqueda en internet sin tener que escribir específicamente lo que desea, sino en su lugar proponemos la búsqueda por imágenes que permitan facilitar la tarea de consultar y extraer

la información de las páginas indexadas a los portales específicos que cumplan el criterio enviado por el cliente.

Los beneficiarios son los usuarios que desean adquirir una prenda de vestir en las tiendas en línea de Ecuador.

Objetivos

Objetivo General

Construir una aplicación móvil con redes neuronales convolucionales para reconocer imágenes de prendas de vestir de tiendas en línea en Ecuador.

Objetivos específicos

- Construir un algoritmo que permita realizar la extracción de imágenes de distintos sitios web con el uso de web scraping.
- Diseñar una interfaz amigable al usuario que permite ingresar imágenes y extraer resultados esperados a partir de los parámetros de entrada.
- Construir una aplicación móvil híbrida para las plataformas Android e iOS.
- Realizar pruebas de funcionalidad y de experiencia de usuario.

Marco metodológico

Para la selección de la metodología de desarrollo que usaremos en este proyecto las tres opciones son: XP, SCRUM, KANABN, y nos guiaremos por cuatro criterios que son:

- Tipo de Framework

- Tipo de Revisión

- Objetivos

- Tipo de Desarrollo

- Tipo de Proyecto

Tabla 1

Comparación de metodologías de desarrollo

Criterios	XP	SCRUM	KANBAN
Tipo de Framework	Es más flexible, dinámico y funcional.	Gestiona el desarrollo de software, se basa en un proceso incremental e iterativo.	Tarjeta Visual útil para los responsables de proyectos.
Objetivos	Se basa en satisfacer al cliente, trabajo en equipo, garantizar software con buena calidad, coste, tiempo y alcance.	Controla y planifica los proyectos con requisitos cambiantes, se trabaja en base a sprints, el proyecto requiere una planificación previa.	Visualización del proyecto, Minimizar tiempos muertos, y/o acumulación de tareas. Mejora de flujo de trabajo grupal, reducir la duración de un proyecto.
Tipo de Revisión	Se integra una vez al día y se realiza pruebas al momento que se termine cada proceso.	Revisiones del Sprint para verificar las tareas cumplidas, colaboran el equipo scrum y los colaboradores.	La manera de revisar es una fuente de información, que demuestra dónde están los cuellos de botella en el proceso y qué es lo que impide que el flujo de trabajo sea continuo.
Tipo de Desarrollo	Desarrollo por fases: Exploración Planificación del proyecto Diseño	Control de forma periódica y adaptable la evolución del proyecto.	Sirve para visualizar el trabajo, y así evitar la acumulación de trabajo

Nota. Se toma en cuenta cinco aspectos para escoger la mejor metodología.

	Codificación Pruebas		pendiente y maximizar la eficiencia.
Tipo de Proyecto	Se usa para proyectos pequeños, que sean flexibles a los contratiempos y cambios.	Para proyectos con equipos pequeños, flexibilidad a cambios a mitad del desarrollo, en cada fase se replantea las tareas y objetivos.	

Elaborado por: Daniela González y Julio Saráuz

La metodología a escoger es SCRUM, porque nos indica paso a paso las etapas para establecer de manera exitosa y organizada nuestro proyecto. Ya que podemos organizar las tareas y contar con una planificación del tiempo estimado. Reuniones diarias para asegurar los avances del proyecto, lo cual fomenta el trabajo en equipo.

CAPÍTULO 1

Estado del arte

1. MARCO TEÓRICO

1.1 Redes Neuronales Convulcionales

Dentro del Deep Learning podemos encontrar un algoritmo denominado red neuronal convolucional, mismo que nos permite modificar fotos, pues estas serán las entradas y serán procesadas para poder asignarles pesos, termino definido para valores numéricos que permiten un fin establecido, una vez se aplican los pesos, se podrá llegar a reconocer elementos pertenecientes a la fotografía ingresada. Esta técnica se encuentra como una de las mejores en cuestión de algoritmos que permiten el crecimiento y mejoramiento de la Visión por computador.

En este tipo de algoritmo se pueden localizar algunas capas ocultas, en las cuales todas las de primera instancia pueden reconocer líneas, curvas y de esta manera pueden ir mejorando hasta el punto de poder conocer una forma completa y compleja, como puede ser el caso de una cara o siluetas. Todos estos trabajos son genéricos en esta clase de red.

El reconocimiento o los tipos de clases para el objeto, se crean por categorías de escenas y por un determinado grupo a la cual la fotografía se generaliza.

La red toma como entrada los pixeles de una imagen. Si tenemos una imagen con apenas 28×28 pixeles de alto y ancho, eso equivale a 784 neuronas. Y eso es si sólo tenemos 1 color (escala de grises). Si tuviéramos una imagen a color, necesitaríamos 3 canales (red, green, blue) y entonces usamos $28 \times 28 \times 3 = 2352$ neuronas de entrada. Esa es nuestra capa de entrada (Bootcamp AI, 2019).

Características:

- Este algoritmo permite generar un modelo de clasificación con el cual mediante el uso de fashion-mnist el cual se trata de un grupo de datos de imágenes de ropa, tenemos la posibilidad de identificar 10 tipos de prendas de ropa. Que sirve para realizar una evaluación comparativa de los algoritmos de aprendizaje automático.

- Este conjunto de datos Fashion-MNIST contiene un grupo para el entrenamiento en los 60 000 tipos de ejemplares que posee y otro grupo de 10 000 ejemplos para las pruebas. En cada caso, la fotografía se encuentra en formato B/N con una dimensión de 28 de alto y 28 de ancho.

1.2 Herramientas de desarrollo

1.2.1 Visual Code

Visual Studio Code es un editor de código fuente que permite trabajar con diversas sintaxis, pertenecientes a lenguajes diferentes, además de que podemos encontrar unos atajos únicos del aplicativo que son de gran ayuda para el usuario, esto con el fin de permitir generar código más pronto y ordenado. Visual es una aplicación gratis que la podemos adquirir, así como su código se encuentra totalmente abierto al público, lo cual nos brinda una utilidad para descargar complementos, mismo que nos ayudan a personalizar y fortalecer a nuestra herramienta.

En los complementos que podemos encontrar de esta herramienta, tenemos opciones infinitas, desde el color del texto, hasta etiquetar todo lo que deseemos en el transcurso de nuestro trabajo. Una de las mayores ventajas que nos ofrece, es la capacidad de autocompletar estructuras, pero aumentar la velocidad de desarrollo. Incluso podemos encontrar complementos que son compatibles con lenguajes como Python, C, Java y todos los que podamos conocer. (AITANA SOLUCIONES ERP Y CRM, 2018)

Características:

- Nos permite generar la codificación en typescript; que es el lenguaje base de Nest js, IONIC y Angular, facilitando la depuración del código fuente utilizando para generar el backend y frontend de la aplicación híbrida.
- Este editor de código fuente compatible con varios lenguajes de programación, permite el desarrollo de las aplicaciones con el framework, permitiendo crear una sencilla y económica codificación. Nuestro desarrollo podrá ser desplegado en

múltiples dispositivos con sistemas operativos diferente pues no necesitará de complementos externos, solo requerirán el archivo correspondiente a su plataforma.

1.3 Lenguajes de programación

1.3.1 Angular

Conocemos angular como uno de los nuevos frameworks más potentes originados por javascript, el cual presenta un gran potencial y muy amigable con los desarrolladores para su cometido, además de ofrecer la oportunidad de dar origen a aplicativos de frontend muy actuales. El poder desarrollar aquí se requiere de un conocimiento medio a experto por el tipo de lógicas que maneja. Dentro de las clases de aplicaciones que se pueden generar aquí, encontramos con portales web SPA o también conocidos como PWA.

En esta herramienta podemos encontrar un sinfín de posibilidades y una plataforma de desarrollo de aplicaciones muy completas, además de ser escalables con el tiempo y brindan la oportunidad de optimizarlas. El uso de angular promueve a una mejor práctica del ambiente de desarrollo y una codificación más homogénea con gran capacidad de modular.

Angular utiliza el lenguaje denominado TypeScript (Es posible usar Javascript puro también de ser el caso), este es conocido como uno de los súper sets del lenguaje Javascript donde se nos promociona un numero de herramientas muy elevado que son compatibles con el lenguaje y sin olvidar de la codificación estática y con el uso de los decoradores.

Característica:

- Es el lenguaje base elegido para el desarrollo en Ionic, utilizando Typescript que nos sirve para potenciar las características de Angular y mejorar la escabilidad.

- Angular nos facilita la manera de compartir el estado de la aplicación entre lo que hay en la parte del servidor y lo que se encuentra en la parte del cliente de la aplicación que estamos se está construyendo.

1.3.2 Python

Este es uno de los lenguajes más populares de los últimos años, Python ha tenido gran acogida por parte de los desarrolladores al incorporar una nueva filosofía, en la cual es muy llamativa por todos, cualquier persona con conocimientos básicos en desarrollo puede ser capaz de aprender este lenguaje. Este se trata de un software totalmente gratuito y trata de incorporar en si partes de códigos open source originado de sus colaboradores y así mejorar su rendimiento en el mercado. No cuenta con algún tipo de licencia que tenga que ser cancelada monetariamente para empezar a darle uso.

Es posible dar origen a aplicaciones de múltiples paradigmas, esto quiere decir que se puede mezclar ciertas propiedades de otros paradigmas de desarrollo. Esto con el fin de que estemos frente a un lenguaje flexible y muy sencillo de adquirir conocimiento. Es muy común observar a personas que conozcan por su propia cuenta este lenguaje.

Para muchos Python está al alcance de todo sistema operativo conocido. Se puede adquirir y ejecutar en cada uno de ellos o podemos optar por el uso únicamente del interpretador.

Características:

- Mediante este lenguaje de programación se desarrolló los servicios externos en Flask que permiten desarrollar todo tipo de aplicaciones complejas y sencillas, ya que es muy versátil. Al tener la posibilidad de estar frente a un lenguaje muy flexible, con una disponibilidad muy alta para los nuevos desarrolladores. Generar un aplicativo web o un servicio rest con

este framework se torna una tarea mucho más sencilla pues no es necesario generar una estructura muy compleja que fuerce a los conocimientos del usuario.

- Se usó la librería OpenCV para el procesamiento de imágenes, ya que esta librería sirve para poder reconocer el movimiento, los objetos y muchas construcciones en tres dimensiones que parten de las fotografías. Open CV es una biblioteca libre, que permite ser utilizada para varios fines ya sean estos de investigación o también con remuneración monetaria en distintas plataformas existentes en el mercado, estas pueden ser Linux, Mac OS, Microsoft y Android por enumerar algunas.

1.3.3 Node Js

Node.js es un entorno de tiempo de ejecución el cual se encuentra codificado bajo JavaScript. Con el cual genera un lapso de compilación en un momento real, además de incluir lo que podemos encontrar necesario para la correcta complicación del programa que hayamos escrito en el lenguaje base de Node. Es posible además de encontrar algunos pros de este lenguaje y sobre todo encontrar muchas soluciones a una diversidad de problemas, esto debido a que es más óptimo mediante el uso del curso de Node, con el cual se puede adquirir el conocimiento suficiente y desarrollar una habilidad que abra paso a querer indagar y llegar a lo más profundo de lo que utilizamos actualmente.

Node nace bajo a necesidad de los desarrolladores originales que dieron origen a JavaScript. La gran diferencia es que mutaron a algo que solo pueda ser ejecutado en un navegador o en algunos ordenadores que soporten estas aplicaciones independientes. Gracias a este lenguaje se puede abrir oportunidades en la programación de JavaScript no solo generando sitios web interactivos, también dando la capacidad de generar cosas que para otros lenguajes competentes de secuencia no lo puedan realizar (OpenWebinars, 2019).

Característica:

- Es el lenguaje base desarrollado sobre las bases del motor Chrome V8, es un entorno de código abierto para hacer posible el desarrollo de aplicaciones JavaScript del lado del servidor. Lo que permite que una vez desarrolladas puedan ejecutarse dentro de diferentes sistemas como OS X, Microsoft Windows y Linux. Ha permitido el desarrollo de Angular y Nest Js, usando en el frontend y el backend.

1.3.4 Ionic

Esta plataforma que nos permite dar origen a nuevos aplicativos que puedan ser ejecutados en dispositivos móviles, siendo estos de la categoría de Android o de IOS, es de código abierto y que incorpora en su estructura varias plataformas ya conocidas como HTML5, y los estilos de CSS, además de lógicas generadas en JavaScript, permitiendo crear todas las interfaces de usuario amigables para los futuros usuarios.

Toda la estructura inicial de Ionic se conoce que esta sobre angular y cordova, esto se dio origen en el año 2013 con el fin de crear aplicaciones móviles que sean de plataformas híbridas, es decir que puedan ser ejecutadas en los teléfonos con distintos sistemas operativos y la construcción sea fácilmente realizada como si se tratase de un portal web.

La gran utilidad que nos presenta este framework radica en el uso de todos los complementos de hardware y software que consta un dispositivo electrónico móvil, gracias al uso de cordova o de capacitor. (QUALITY DEVS, 2019).

Característica:

- Con el uso de Ionic podemos mencionar que usar esta librería nos dará la oportunidad de generar una aplicación destinada a los móviles y probada en el escritorio de un computador,

todo gracias a la integración de los tres pilares de los portales web: CSS, HTML y JavaScript. Ayudando al fácil manejo de los componentes e implementaciones que se harán en dicha aplicación. Además, al utilizar capacitor permite a un desarrollador generar aquella funcionalidad nativa y adaptada al teléfono, sea este Android o IOS.

1.3.5 Nest Js

Con este framework que está totalmente desarrollado en TypeScript podemos combinar con Node y permitir una nueva visión en desarrollo del lado del servidor. El origen de Nest parte de copiar muchas ideas que fueron dadas por Spring, pero la diferencia o el aporte extra que da Nest es que al momento de desarrollar y permitir que un programa madure se puedan realizar directamente con Node. El futuro que tiene esta herramienta es muy prometedor, pues para presentar soluciones sólidas y extensibles, todas las llamadas que se hacen en TypeScript como lenguaje base fomenta a generar un salto muy importante en comparación con C# o Java. (Caules, 2019).

Característica:

- Es un framework progresivo de Node.js para la creación de aplicaciones eficientes, confiables y escalables del lado del servidor, el cual está constituido y es compatible con TypeScript. Es el encargado de brindar todos los servicios que permite manipular en la base de datos de Mongo db y realizar acciones para funcionamiento de la aplicación. Usando algunas librerías existentes como es Express que es un framework web rápido, el cual proporciona varias utilidades HTTP para crear API REST robustas de manera fácil y rápida y TypeORM para los lenguajes TypeScript y JavaScript moderno.

1.4 Plataformas móviles

1.4.1 Android

Es un sistema operativo móvil diseñado para dispositivos móviles con pantalla táctil como los Smartphone conocidos hoy en día, los cuales cuenta con memorias internas y hardware tecnológico que los convierte en dispositivos inteligentes, también es el caso para los televisores y algunos sistemas de multimedia del hogar, es posible también encontrar este sistema operativo en autos que tengan incorporado dispositivos compatibles. Este es uno de los sistemas operativos desarrollado por Google y se encuentra basado en el Kernel de Linux con otros de código abierto. Es posible realizar el diseño en esta plataforma de manera sencilla y en distintas plataformas que ofertan el poder generar nuevos aplicativos para Android (AZ, 2021).

Característica:

- Mediante el uso de Android se graba la aplicación en un archivo apk, que brinde la oportunidad de instalar en todos los dispositivos Android compatibles. Es una plataforma abierta está basado en Linux, ya que ayuda con varios servicios incorporados como la localización GPS, bases de datos, sensores.
- Android tiene varios componentes que nos ayudan para solicitar la función de la cámara del usuario, lo que nos va servir para guardar en nuestra base de datos y poder buscar prendas similares

1.4.2 IOS

Este es un sistema operativo muy restrictivo, el cual fue lanzado por Apple. El nombre que hoy lo conocemos nace de iPhoneOs, porque Iphone Operative System utiliza las siglas de IOS. Este sistema fue lanzado principalmente para su teléfono que pertenece a la marca, a pesar de ser utilizado en años posteriores para otros de la misma compañía, como es el caso de los Ipod.

Este es un sistema operativo privativo que no puede ser utilizado por ningún otro dispositivo que no sea de marca Apple. Este es una de las grandes diferencias que posee ante Android, pues en el caso de este otro sistema operativo, puede ser implementado en cualquier dispositivo, pero en el caso de IOS es un poco más especial, pues no puede ser instanciado en ningún modelo de Smartphone que no sea de marca Apple. Y en cuestión de instalar aplicaciones dentro de este sistema operativo no puede ser de cualquier fuente desconocida, solamente pueden ser instaladas directamente de la App Store, la cual contiene apps que han pasado varias fases de revisión. (García, 2021).

Característica:

- Mediante el uso de iOS se graba la aplicación con un archivo apk, que brinde la oportunidad de instalar en todos los dispositivos iOS compatibles.

1.5 Metodologías de desarrollo

1.5.1 Metodología Scrum

La metodología Scrum es un marco de trabajo o framework que se utiliza dentro de equipos que desean dar origen a grandes y complejos proyectos. Esto quiere decir, que se trata de una metodología ágil que siempre busca entregar en periodos muy cortos de tiempo sus proyectos basándose en los principios de transparencia, inspección y adaptación. Esto abre paso a que el cliente puede interactuar con el equipo, además de ingresar el producto al mercado lo más pronto posible y comenzar a generar ventas (WAM, 2021).

1.5.2 Roles

Patrocinador

Tabla 2

Patrocinador

Nombre y Apellido	Cargo en el equipo	Función en el equipo
Daniel Díaz	Patrocinador	Se encarga de que se cumpla los objetivos y requerimientos del proyecto.

Nota. Información del patrocinador del proyecto.

Elaborado por: Daniela González y Julio Saraúz

Scrum Master

Tabla 3

Scrum Master

Nombre y Apellido	Cargo en el equipo	Función en el equipo
Julio Saraúz	Estudiante	Se encarga de ser intermediario entre el patrocinador y los desarrolladores.

Nota. Información del Scrum Master del proyecto.

Elaborado por: Daniela González y Julio Saraúz

Equipo de desarrollo

Tabla 4

Desarrolladores

Nombre y Apellido	Cargo en el equipo	Función en el equipo
Julio Saráuz	Estudiante	Se encargan de la implementación del software.
Daniela González	Estudiante	Se encargan de la implementación del software.

Nota. Información de los desarrolladores del proyecto.

Elaborado por: Daniela González y Julio Saráuz

1.5.3 Cronograma de Actividades realizadas en el proyecto

Son todas las actividades que se cumplieron en el tiempo determinado para el avance del proyecto, previa a una planificación.

Tabla 5

Cronograma de Actividades realizadas

Nombre de tarea	Duración en días	Comienzo de la tarea	Fin de la tarea
Proyecto 1	78 días	jue 15/4/21	Mié 18/8/21
Fase 1	9 días	jue 15/4/21	sáb 24/4/21
Análisis de requerimientos funcionales y no funcionales	3 días	jue 15/4/21	sáb 17/4/21

Modelado de diagramas de caso de uso, secuencia, conceptual	2 días	lun 19/4/21	mar 20/4/21
Crear mockups para la app final	4 días	mié 21/4/21	sáb 24/4/21
Fase 2	6 días	lun 26/4/21	lun 3/5/21
Módulo de inicio de sesión	4 días	lun 26/4/21	jue 29/4/21
Módulo de registro de usuarios	2 días	vie 30/4/21	lun 3/5/21
Fase 3	12 días	lun 3/5/21	mar 18/5/21
Creación del Web Service para Scrapeo de la Web	7 días	lun 3/5/21	mar 11/5/21
Generar repositorio para control de versiones del Web Scraping	5 días	mié 12/5/21	mar 18/5/21
Fase 4	28 días	mié 19/5/21	jue 24/6/21
Crear servidor con NEST para consumir servicios	6 días	mié 19/5/21	mié 26/5/21
Crear modelo de predicción con Python - reconocimiento de prendas	7 días	vie 28/5/21	sáb 5/6/21
Crear servicios en FLASK para consumir servicio de predicción	4 días	lun 7/6/21	jue 10/6/21
Consumir servicios de Flask desde Nest js	4 días	vie 11/6/21	mié 16/6/21
Fase 5	5 días	jue 17/6/21	mié 23/6/21
Generar versión 1 de app para consumir servicios en IONIC	5 días	jue 17/6/21	mié 23/6/21

Fase 6	10 días	jue 24/6/21	mar 6/7/21
Generar .apk para Android	3 días	jue 24/6/21	dom 27/6/21
Generar .apk para iOS	5 días	lun 28/6/21	vie 2/7/21
Fase Final de Pruebas y Mejoras	8 días	vie 2/7/21	mar 13/7/21
Pruebas en las fases	2 días	vie 2/7/21	lun 5/7/21
Pruebas en los dispositivos físicos	4 días	mar 6/7/21	vie 9/7/21
Mejorar la interfaz gráfica	4 días	lun 12/7/21	jue 15/7/21

Nota. Información de las actividades a realizarse en el proyecto.

Elaborado por: Daniela González y Julio Saráuz

1.5.4 Sprints

Mediante el cronograma de actividades se realizó los siguientes entregables con fechas señaladas. Clasificado por fases del proyecto, los sprints se detallan a continuación:

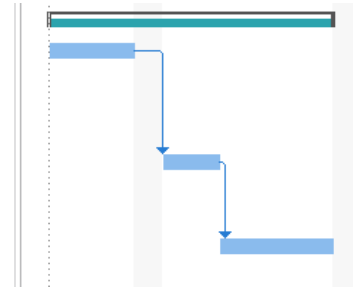
1.5.4.1 Sprint 1

En la Figura 1 se puede revisar la planificación del primer sprint aquí se observa los tiempos estimados del trabajo en días y horas que se asignan por cada desarrollador, según la complejidad de cada tarea.

Figura 1

Fase 1 de desarrollo de la aplicación

▲ Fase 1	9 días	jue 15/4/21	sáb 24/4/21
Análisis de requerimientos funcionales y no funcionales	3 días	jue 15/4/21	sáb 17/4/21
Modelado de diagramas de caso de uso, secuencia, conceptual	2 días	lun 19/4/21	mar 20/4/21
Crear mockups para la app final	4 días	mié 21/4/21	sáb 24/4/21



Nota. Spring 1 realizado en 9 días desde el jueves 15 de mayo al sábado 24 de mayo del 2021, empezando con la fase inicial del análisis de la aplicación.

Elaborado por los autores: Daniela González y Julio Saráuz

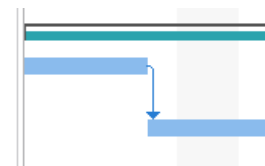
1.5.4.2 Sprint 2

En la Figura 2 se puede revisar la planificación del primer sprint aquí se observa los tiempos estimados del trabajo en días y horas que se asignan por cada desarrollador, según la complejidad de cada tarea.

Figura 2

Fase 2 de desarrollo de la aplicación

▲ Fase 2	6 días	lun 26/4/21	lun 3/5/21
Módulo de inicio de sesión	4 días	lun 26/4/21	jue 29/4/21
Módulo de registro de usuarios	2 días	vie 30/4/21	lun 3/5/21



Nota. Spring 2 realizado en 6 días desde el lunes 26 de mayo al lunes 03 de mayo del 2021, continuando con la fase 2 de los primeros módulos de la aplicación.

Elaborado por los autores: Daniela González y Julio Saráuz

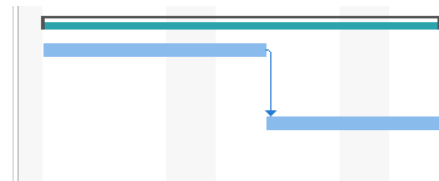
1.5.4.3 Sprint 3

En la Figura 3 se puede revisar la planificación del primer sprint aquí se observa los tiempos estimados del trabajo en días y horas que se asignan por cada desarrollador, según la complejidad de cada tarea.

Figura 3

Fase 3 de desarrollo de la aplicación

▲ Fase 3	12 días	lun 3/5/21	mar 18/5/21
Creación del Web Service para Scrapeo de la Web	7 días	lun 3/5/21	mar 11/5/21
Generar repositorio para control de versiones del Web Scraping	5 días	mié 12/5/21	mar 18/5/21



Nota. Spring 3 realizado en 12 días desde el lunes 03 de mayo al martes 18 de mayo del 2021, continuando con la fase 3 de la aplicación.

Elaborado por los autores: Daniela González y Julio Saráuz

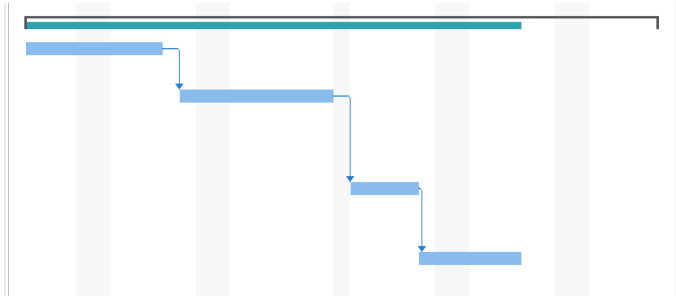
1.5.4.4 Sprint 4

En la Figura 4 se puede revisar la planificación del primer sprint aquí se observa los tiempos estimados del trabajo en días y horas que se asignan por cada desarrollador, según la complejidad de cada tarea.

Figura 4

Fase 4 de desarrollo de la aplicación

▲ Fase 4	28 días	mié 19/5/21	jue 24/6/21
Crear servidor con NEST para consumir servicios	6 días	mié 19/5/21	mié 26/5/21
Crear modelo de predicción con Python - reconocimiento de prendas	7 días	vie 28/5/21	sáb 5/6/21
Crear servicios en FLASK para consumir servicio de predicción	4 días	lun 7/6/21	jue 10/6/21
Consumir servicios de Flask desde Nest js	4 días	vie 11/6/21	mié 16/6/21



Nota. Spring 4 realizado en 28 días desde el miércoles 19 de mayo al jueves 24 de junio del 2021, continuando con la fase 4 de la aplicación.

Elaborado por los autores: Daniela González y Julio Saráuz

1.5.4.5 Sprint 5

En la Figura 5 se puede revisar la planificación del primer sprint aquí se observa los tiempos estimados del trabajo en días y horas que se asignan por cada desarrollador, según la complejidad de cada tarea.

Figura 5

Fase 5 de desarrollo de la aplicación

▲ Fase 5	5 días	jue 17/6/21	mié 23/6/21
Generar version 1 de app para consumir servicios en IONIC	5 días	jue 17/6/21	mié 23/6/21



Nota. Spring 5 realizado en 5 días desde el jueves 17 de junio al miércoles 23 de junio del 2021, continuando con la fase 5 de la aplicación.

Elaborado por los autores: Daniela González y Julio Saráuz

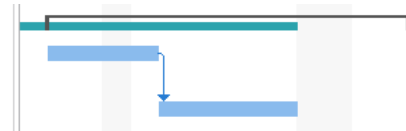
1.5.4.6 Sprint 6

En la Figura 6 se puede revisar la planificación del primer sprint aquí se observa los tiempos estimados del trabajo en días y horas que se asignan por cada desarrollador, según la complejidad de cada tarea.

Figura 6

Fase 6 de desarrollo de la aplicación

▲ Fase 6	10 días	jue 24/6/21	mar 6/7/21
Generar .apk para Android	3 días	jue 24/6/21	dom 27/6/21
Generar .apk para los	5 días	lun 28/6/21	vie 2/7/21



Nota. Spring 6 realizado en 10 días desde el jueves 24 de junio al martes 06 de julio del 2021, continuando con la fase 6 de la aplicación.

Elaborado por los autores: Daniela González y Julio Saráuz

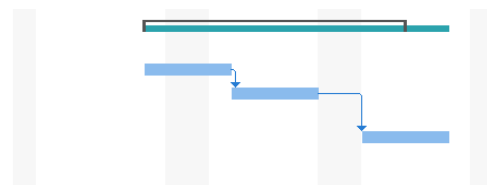
1.5.4.7 Sprint 7

En la Figura 7 se puede revisar la planificación del primer sprint aquí se observa los tiempos estimados del trabajo en días y horas que se asignan por cada desarrollador, según la complejidad de cada tarea.

Figura 7

Fase 7 de desarrollo de la aplicación

▲ Fase Final de Pruebas y Mejoras	8 días	vie 2/7/21	mar 13/7/21
Pruebas en las fases	2 días	vie 2/7/21	lun 5/7/21
Pruebas en los dispositivos físicos	4 días	mar 6/7/21	vie 9/7/21
Mejorar la interfaz gráfica	4 días	lun 12/7/21	jue 15/7/21



Nota. Spring 7 realizado en 8 días desde el viernes 02 de julio al martes 13 de julio del 2021, finalizando con la fase de pruebas y mejoras.

Elaborado por los autores: Daniela González y Julio Saráuz

CAPÍTULO 2

2. REQUERIMIENTOS FUNCIONALES Y NO FUNCIONALES

2.1 Requerimientos funcionales

Tabla 6

Requerimientos Funcionales de la aplicación

CODIGO	REQUERIMIENTO
RF1	Inicio de sesión de usuarios
RF2	Registro de usuarios
RF3	Ingreso de iniciar sesión con clave temporal
RF4	La aplicación permitirá al usuario capturar una imagen con su celular en tiempo real y guardar en la base de datos.
RF5	La aplicación debe permitir subir una foto de la galería de cada usuario que tiene en su celular y guardar en la base de datos MongoDB.

RF6	La aplicación debe permitir ingresar un url de las tiendas que se desea scrapear y guardar en la base de datos MongoDB.
------------	---

Nota. Esta tabla contiene la especificación de los requerimientos funcionales de la aplicación.

Elaborado por los autores: Daniela González y Julio Saráuz

2.2 Requerimientos no funcionales

Tabla 7

Requerimientos No Funcionales de la aplicación

Código	Requerimiento
RNF1	La aplicación debe tener un logo y nombre.
RNF2	La aplicación debe admitir formatos jpg y png para la foto que desean subir.
RNF3	La aplicación debe utilizar un gestor de base de datos.
RNF4	La aplicación debe ser compatible el sistema operativo Android.
RNF5	La aplicación debe ser compatible el sistema operativo iOS.

RNF6	La aplicación mostrará mensajes de error e información en caso de ser necesario.
RBF7	La aplicación debe tener una interfaz gráfica amigable para el usuario.

Nota. Esta tabla contiene la especificación de los requerimientos no funcionales de la aplicación.

Elaborado por los autores: Daniela González y Julio Saráuz

2.3 ARTEFACTOS UML

2.3.1 Historias de usuario

Las historias de usuario de cada módulo se presentan a continuación:

Tabla 8

Historia de usuario del requerimiento 1

Requerimiento	RF1
Actor	Administrador
Flujo Normal	<ol style="list-style-type: none"> 1. El usuario ingresa a iniciar sesión de la aplicación 2. Despliega una ventana con los controles para el inicio de sesión. 3. El usuario ingresa sus credenciales de acceso 4. La aplicación valida los datos ingresados 5. La aplicación permite ingresar al usuario a su sesión.

	6. La aplicación notifica que las credenciales son correctas.
Flujo Alternativo	<p>3. a El sistema presenta un cuadro de diálogo si no encuentra valores ingresados.</p> <p>4. a El sistema presenta un cuadro de diálogo de error si existen campos vacíos.</p> <p>6. a El sistema presenta un cuadro de diálogo si las credenciales son incorrectas</p>

Nota. Esta tabla contiene la historia de usuario N°1.

Elaborado por los autores: Daniela González y Julio Saráuz

Tabla 9

Historia de usuario del requerimiento 2 y 3

Requerimiento	RF2, RF3
Actor	Administrador
Flujo Normal	<ol style="list-style-type: none"> 1. El usuario ingresa al registro de la aplicación 2. La aplicación despliega una ventana con los controles para el registro del usuario. 3. El usuario debe ingresar los datos: usuario y correo electrónico. 4. La aplicación guarda los datos ingresados. 5. La aplicación genera una contraseña temporal, enviada al correo ingresado por el usuario.

	<p>6. La aplicación indica un mensaje de revisar el correo electrónico.</p> <p>7. La aplicación solicita la clave temporal.</p> <p>8. La aplicación solicita el cambio de clave.</p> <p>9. La aplicación guarda la nueva contraseña.</p>
Flujo Alternativo	<p>3. a El sistema presenta un cuadro de diálogo si no encuentra valores ingresados.</p> <p>4. a El sistema presenta un cuadro de diálogo de error si existen campos vacíos.</p>

Nota. Esta tabla contiene las historias de usuario N° 2 y N°3

Elaborado por los autores: Daniela González y Julio Saráuz

Tabla 10

Historia de usuario del requerimiento 4

Requerimiento	RF4
Actor	Usuario
Flujo Normal	<p>1. El usuario ingresa a la aplicación a capturar una foto.</p> <p>2. La aplicación despliega una ventana con los controles para capturar una foto en tiempo real.</p> <p>3. El usuario envía la foto capturada.</p> <p>4. La base de datos guarda la foto enviada por el usuario.</p>

	5. El usuario podrá visualizar los resultados de las prendas encontradas similares y los url de las páginas encontradas.
Flujo Alternativo	3. a El sistema presenta un cuadro de diálogo si no encuentra valores ingresados. 4. a El sistema presenta un cuadro de diálogo de error si existen campos vacíos.

Nota. Esta tabla contiene la historia de usuario N°4.

Elaborado por los autores: Daniela González y Julio Saráuz

Tabla 11

Historia de usuario del requerimiento 5

Requerimiento	RF5
Actor	Usuario
Flujo Normal	1. El usuario ingresa a subir imagen de la aplicación. 2. Se despliega una ventana con la carpeta que desea subir la imagen que tenga en el almacenamiento del celular. 3. El usuario sube la imagen seleccionada. 4. La base de datos guarda la foto enviada por el usuario. 5. El usuario podrá visualizar los resultados de las prendas encontradas similares y los url de las páginas encontradas.

Flujo Alternativo	<p>3. a El sistema presenta un cuadro de diálogo si no encuentra valores ingresados.</p> <p>4. a El sistema presenta un cuadro de diálogo de error si existen campos vacíos.</p>
-------------------	--

Nota. Esta tabla contiene la historia de usuario N°5

Elaborado por los autores: Daniela González y Julio Saráuz

Tabla 12

Historia de usuario del requerimiento 6

Requerimiento	RF6
Actor	Administrador
Flujo Normal	<ol style="list-style-type: none"> 1. El administrador ingresa a ingresar página de la aplicación. 2. La aplicación despliega una ventana, donde pide la URL de la página web. 3. El usuario ingresa una URL. 4. El usuario envía la URL. 5. La base de datos guarda la URL ingresada.
Flujo Alternativo	<p>3. a El sistema presenta un cuadro de diálogo si no encuentra valores ingresados.</p> <p>4. a El sistema presenta un cuadro de diálogo de error si existen campos vacíos.</p>

Nota. Esta tabla contiene la historia de usuario N°6.

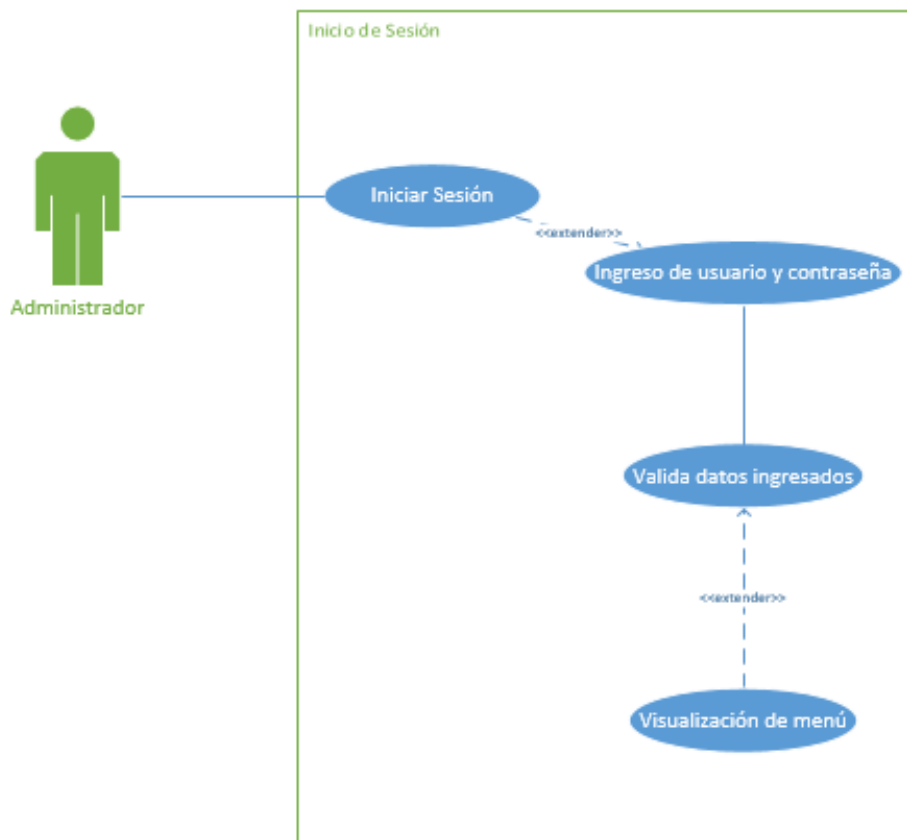
Elaborado por los autores: Daniela González y Julio Saráuz

2.3.2 Diagrama de caso de uso

En la figura 8, se define el funcionamiento del requerimiento N°1 mediante el diagrama de caso de uso, donde se visualiza las actividades que tiene que realizar el administrador cuando inicia sesión en la aplicación.

Figura 8

Diagrama de caso de uso N°1

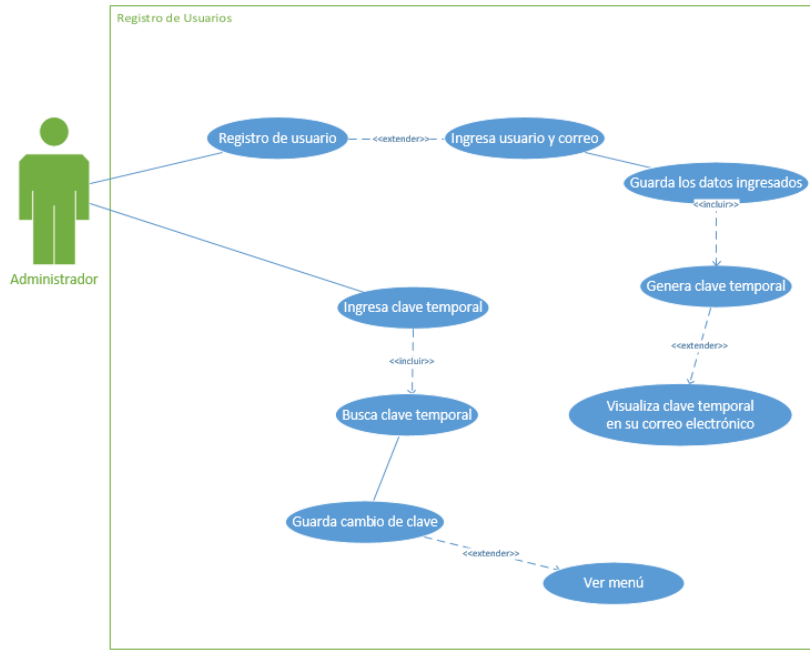


Nota. Inicio de Sesión en la aplicación.

Elaborado por los autores: Daniela González y Julio Saráuz

Figura 9

Diagrama de caso de uso N°2 y N°3

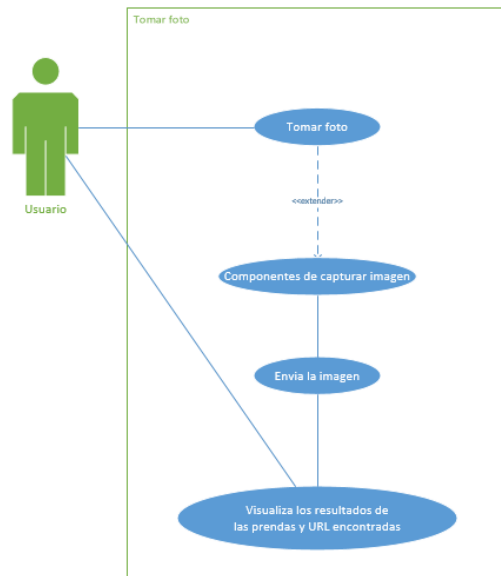


Nota. Registro de Usuarios

Elaborado por los autores: Daniela González y Julio Saráuz

Figura 10

Diagrama de caso de uso N°4

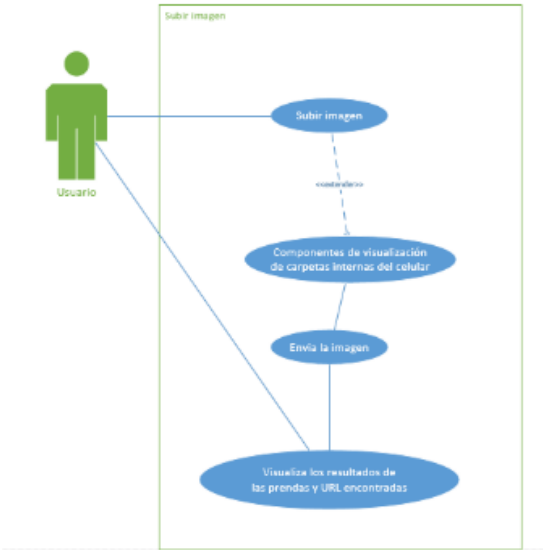


Nota. Tomar Foto

Elaborado por los autores: Daniela González y Julio Saráuz

Figura 11

Diagrama de caso de uso N°5

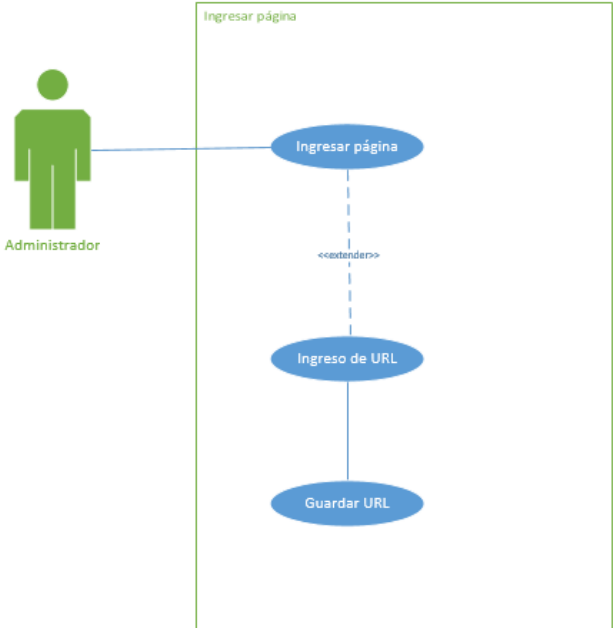


Nota. Subir imagen

Elaborado por los autores: Daniela González y Julio Saráuz

Figura 12

Diagrama de caso de uso N°6



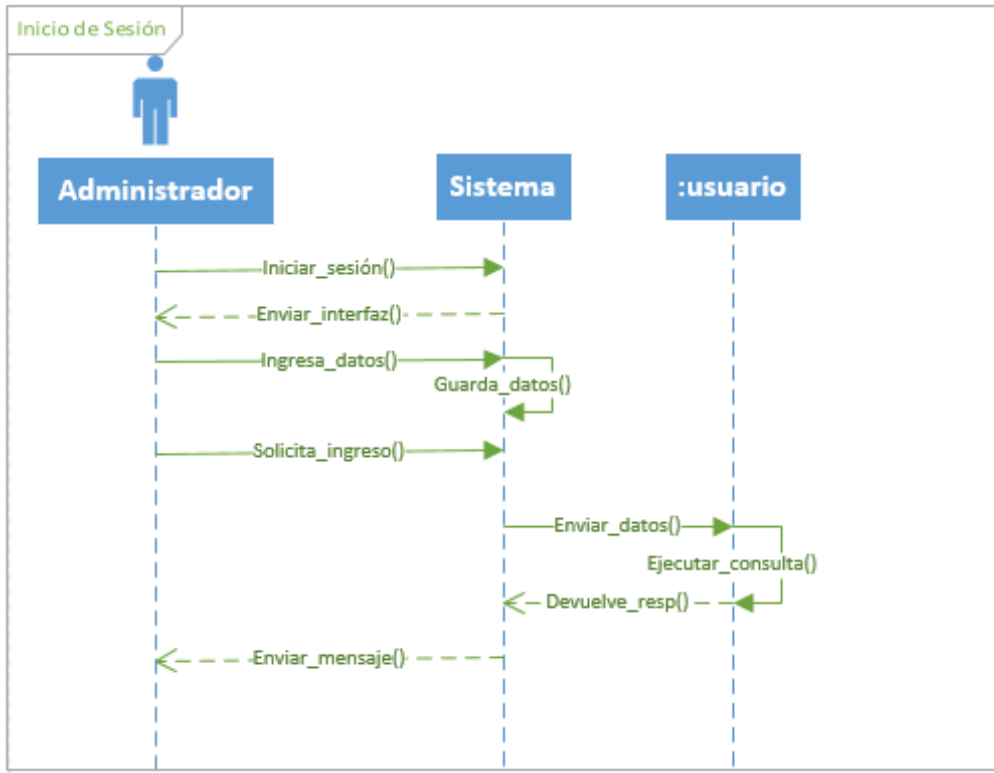
Nota. Ingresar página

Elaborado por los autores: Daniela González y Julio Saráuz

2.3.3 Diagrama Secuencias

Figura 13

Diagrama de secuencias – Inicio de Sesión

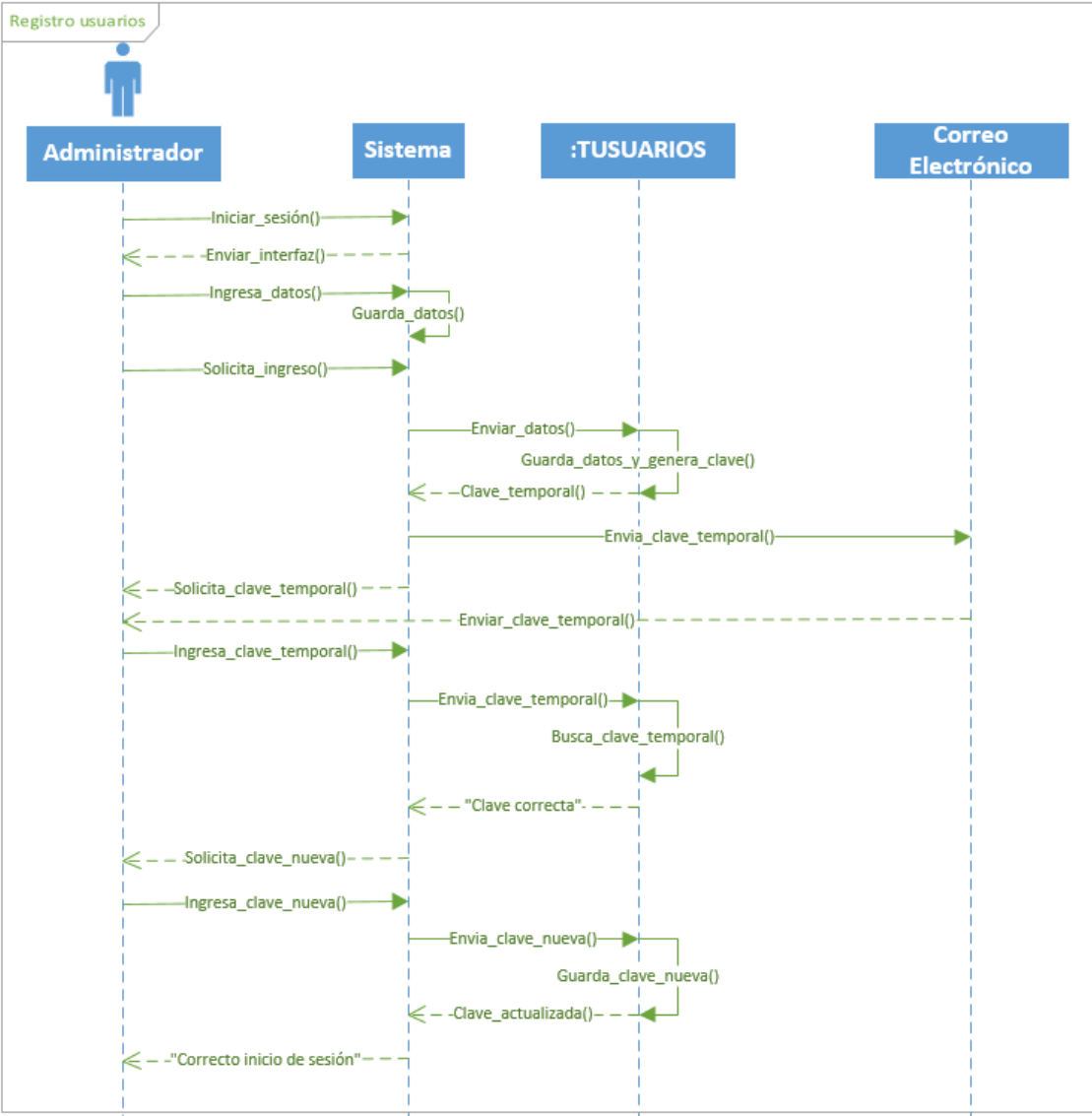


Nota. Inicio de Sesión

Elaborado por los autores: Daniela González y Julio Saráuz

Figura 14

Diagrama de secuencias – Registro usuarios

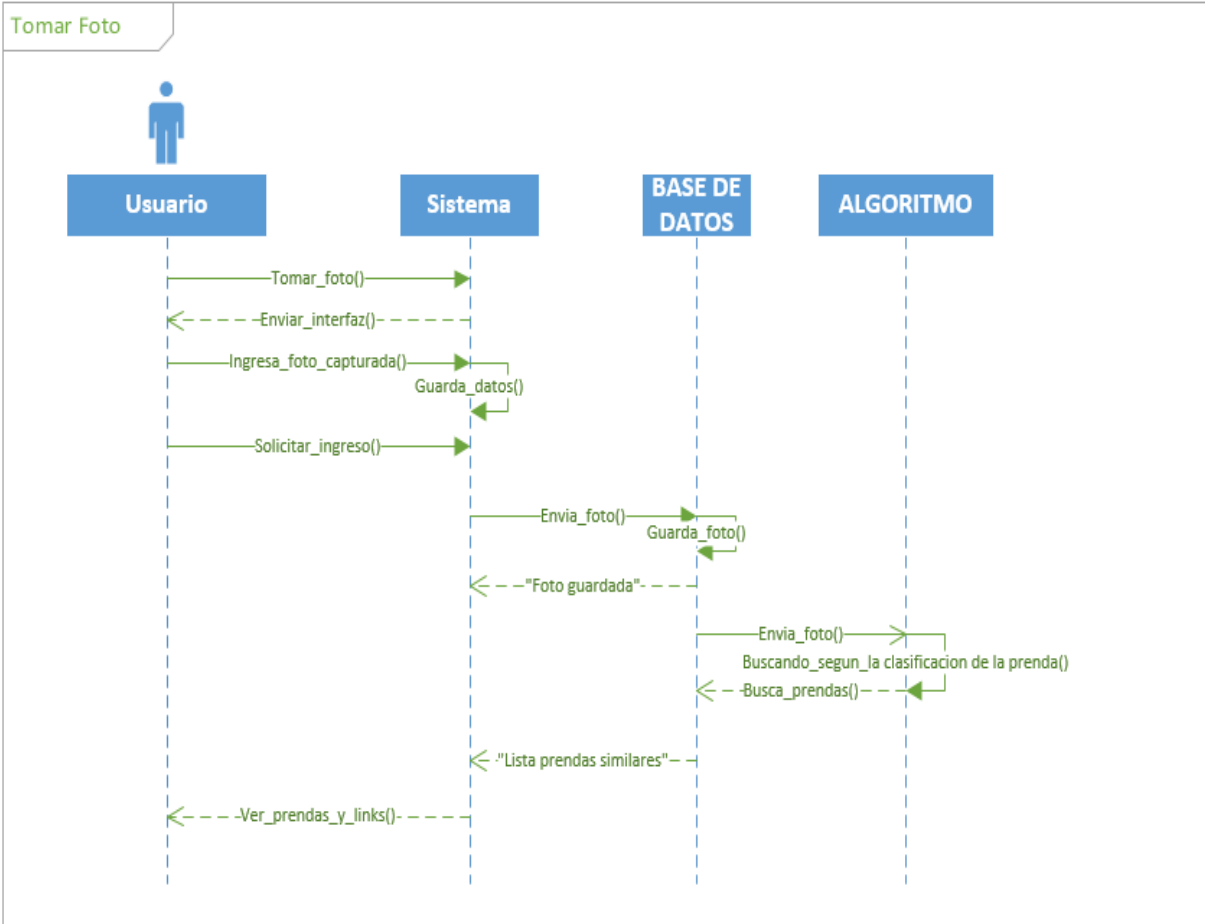


Nota. Registro usuarios

Elaborado por los autores: Daniela González y Julio Saráuz

Figura 15

Diagrama de secuencias – Tomar Foto

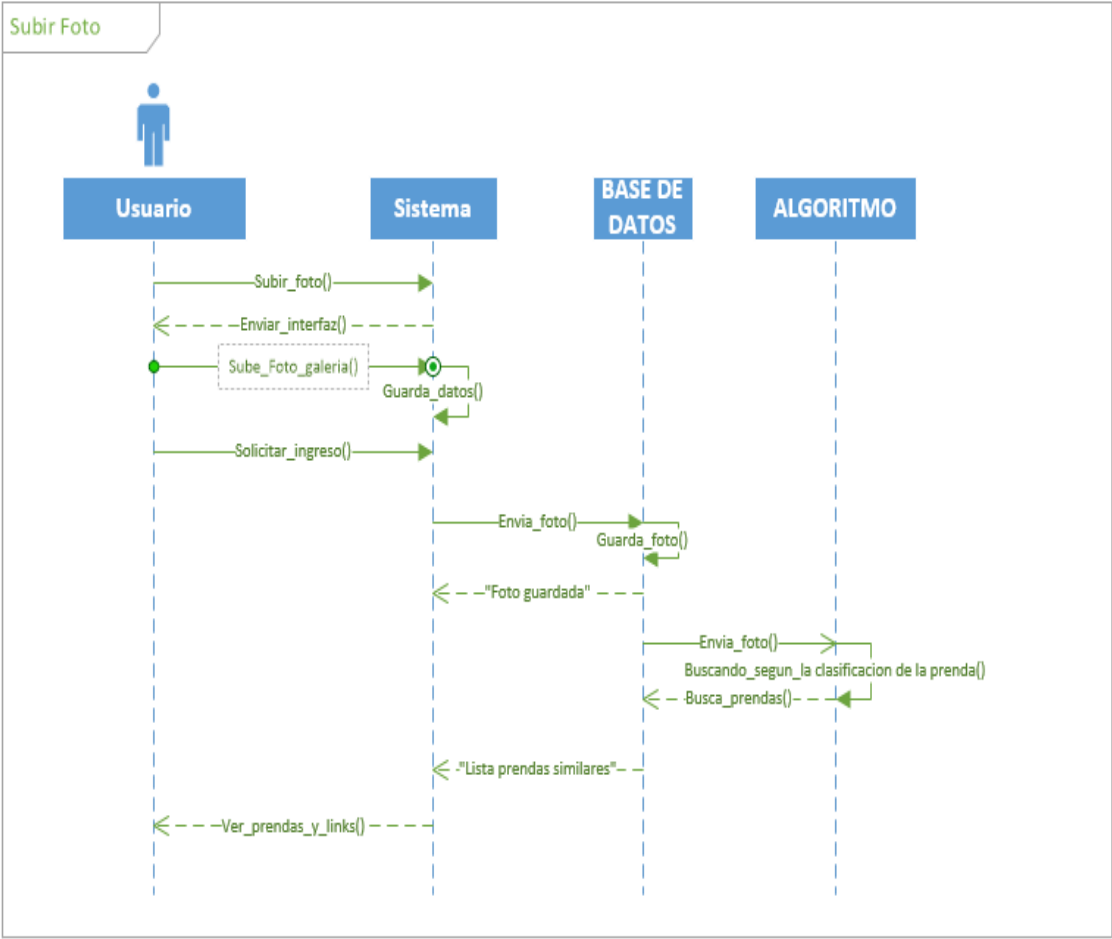


Nota. Tomar Foto

Elaborado por los autores: Daniela González y Julio Saráuz

Figura 16

Diagrama de secuencias – Subir Foto

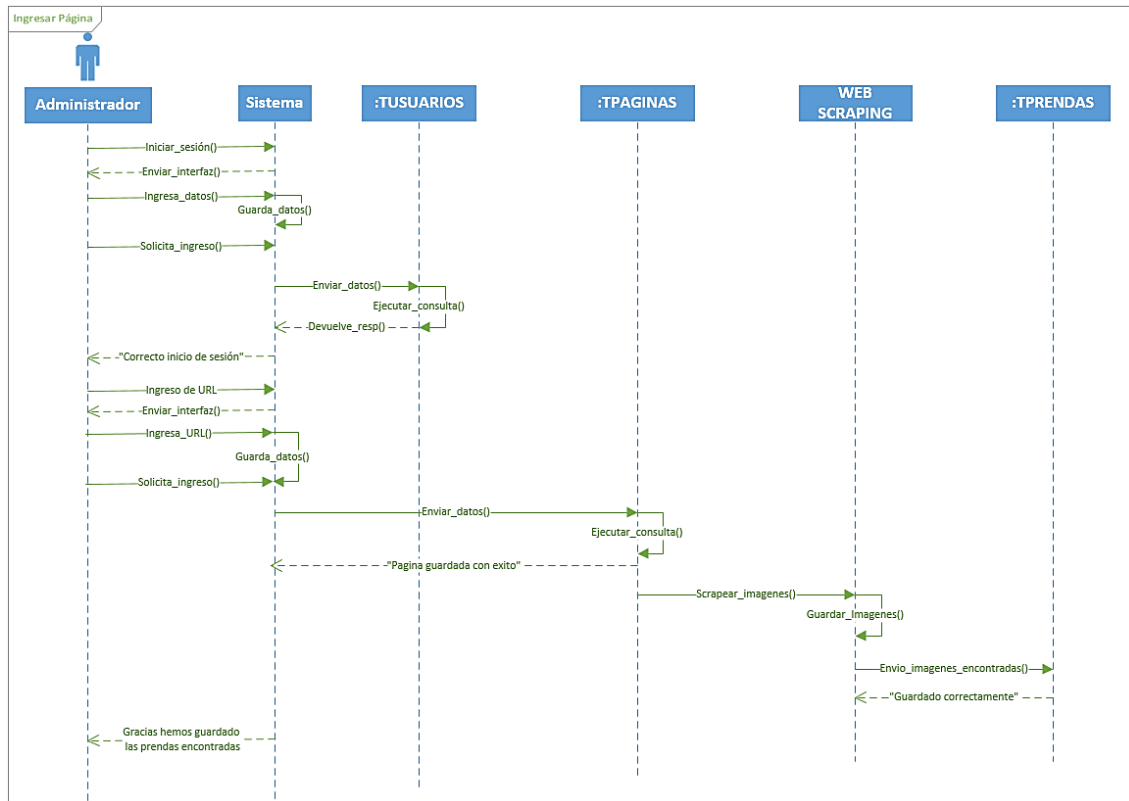


Nota. Subir Foto

Elaborado por los autores: Daniela González y Julio Saráuz

Figura 17

Diagrama de secuencias – Ingresar Página



Nota. Ingresar Página

Elaborado por los autores: Daniela González y Julio Saráuz

2.3.4 Prototipo Inicial

En el prototipo inicial escogimos el color: #EF6C0D y el color: #E5E5E5, el nombre inicial era FindmyStyle.

Figura 18

Prototipo Inicial de la pantalla de inicio y el menú.

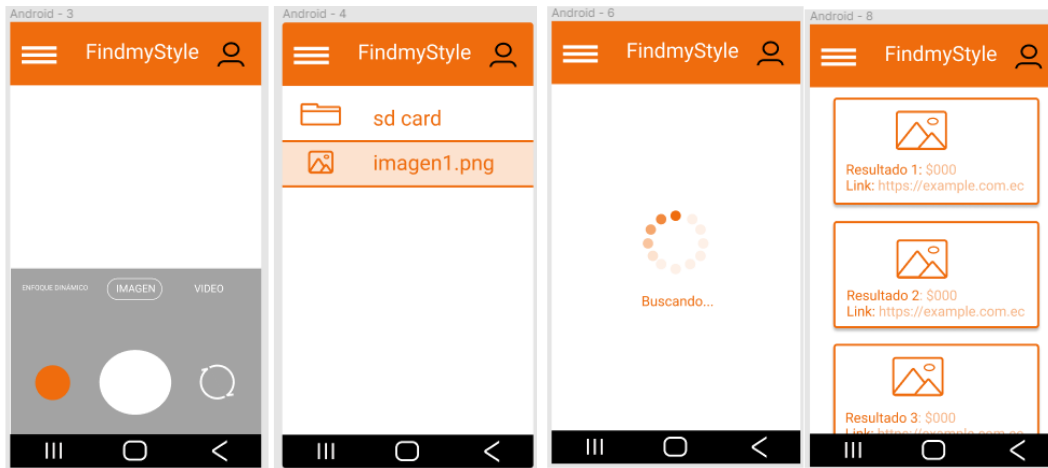


Nota. Prototipo Inicial de la pantalla de inicio y el menú.

Elaborado por los autores: Daniela González y Julio Saráuz

Figura 19

Prototipo de Subir Foto, ir a galería, y buscar prendas similares.

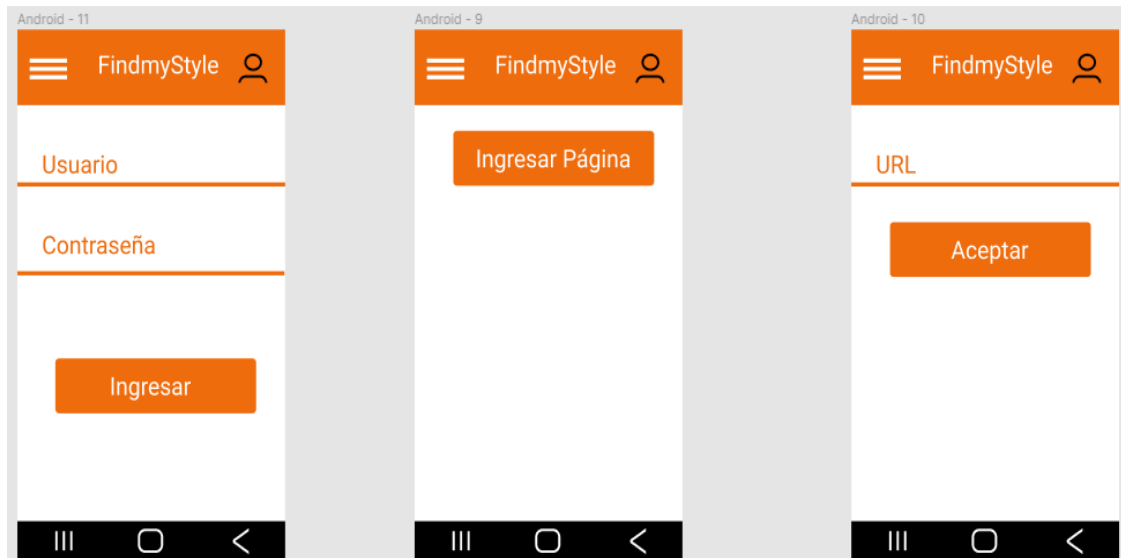


Nota. Prototipo de Subir Foto

Elaborado por los autores: Daniela González y Julio Saráuz

Figura 20

Prototipo de Registro del Administrador, ingresar página y sección de ingresar el URL



Nota. Prototipo de Registro del Administrador, ingresar página y sección de ingresar el URL

Elaborado por los autores: Daniela González y Julio Saráuz

2.3.5 Prototipo Final

En el prototipo final se utilizó colores azules, el nombre de la aplicación tiene de nombre DayJu, hemos añadido ayudas de la aplicación de cómo usar ciertas secciones y también se añadió mensajes de error en el ingreso de los datos.

Figura 21

Prototipo Final de la pantalla de inicio, con las ayudas e información de la utilización de la aplicación.

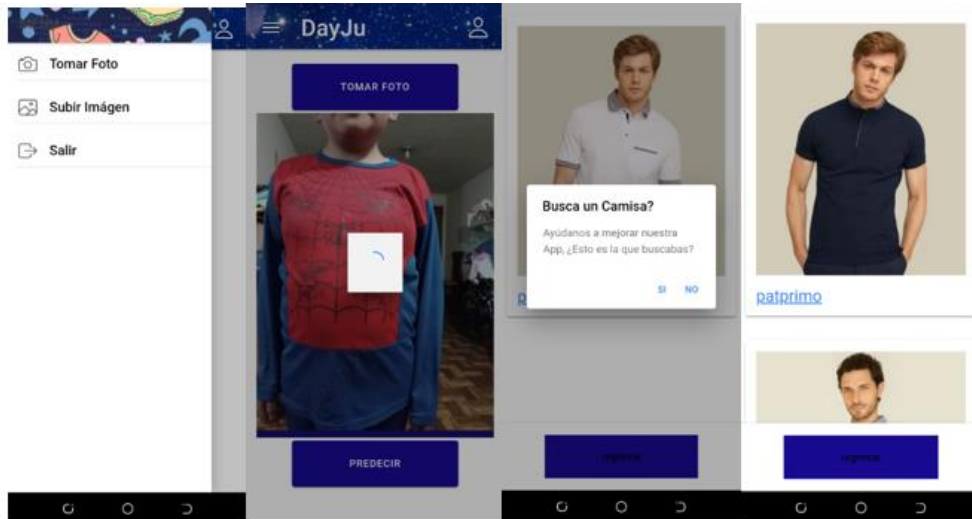


Nota. Prototipo Final de la pantalla de inicio

Elaborado por los autores: Daniela González y Julio Saráuz

Figura 22

Prototipo de Subir Foto, ir a galería, y buscar prendas similares.

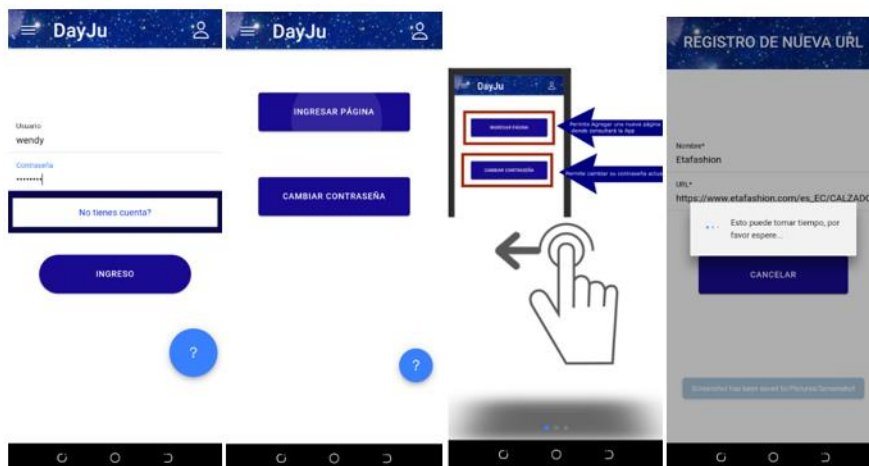


Nota. Prototipo de Subir Foto

Elaborado por los autores: Daniela González y Julio Saráuz

Figura 23

Prototipo Final de Registro del Administrador, ingresar página, sección de ingresar el URL y ayudas de explicación.



Nota. Prototipo de Registro del Administrador, ingresar página y sección de ingresar el URL

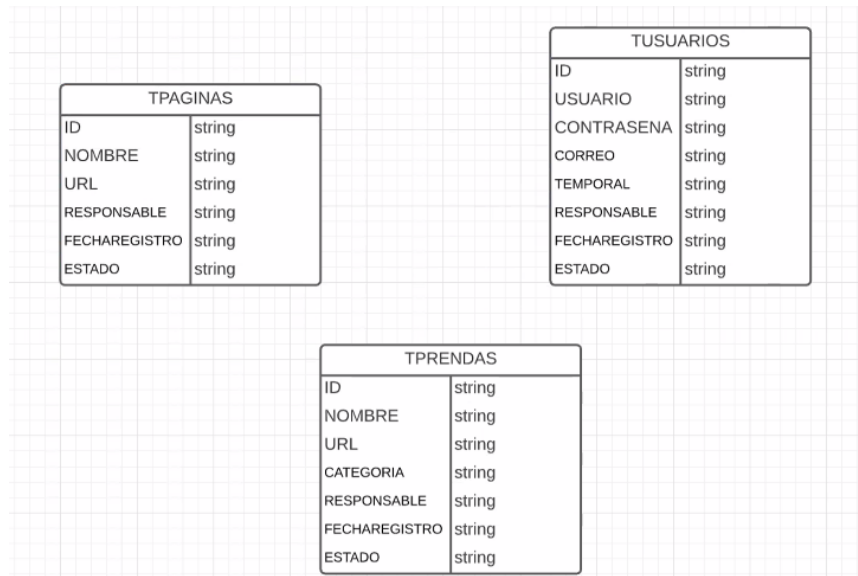
Elaborado por los autores: Daniela González y Julio Saráuz

2.3.4 Diagrama conceptual de la base de datos

La base de datos es no relacional, estas son las tablas creadas para el ingreso de las páginas, usuarios, y prendas.

Figura 24

Diagrama conceptual de la base de datos



Nota. Base no relacional

Elaborado por los autores: Daniela González y Julio Saráuz

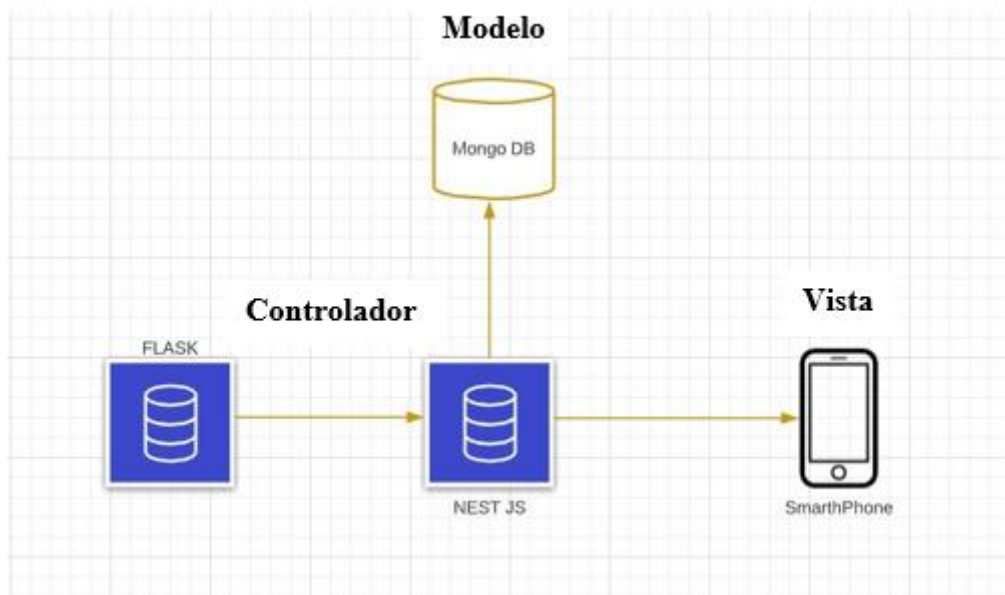
CAPÍTULO 3

3.1 ARQUITECTURA

En la figura No.25 se observa la arquitectura general de la aplicación con los servicios que utiliza. Con esto detallamos que se está usando la arquitectura cliente-servidor y la arquitectura MVC.

Figura 25

Arquitectura de la aplicación



Nota. Arquitectura de la aplicación

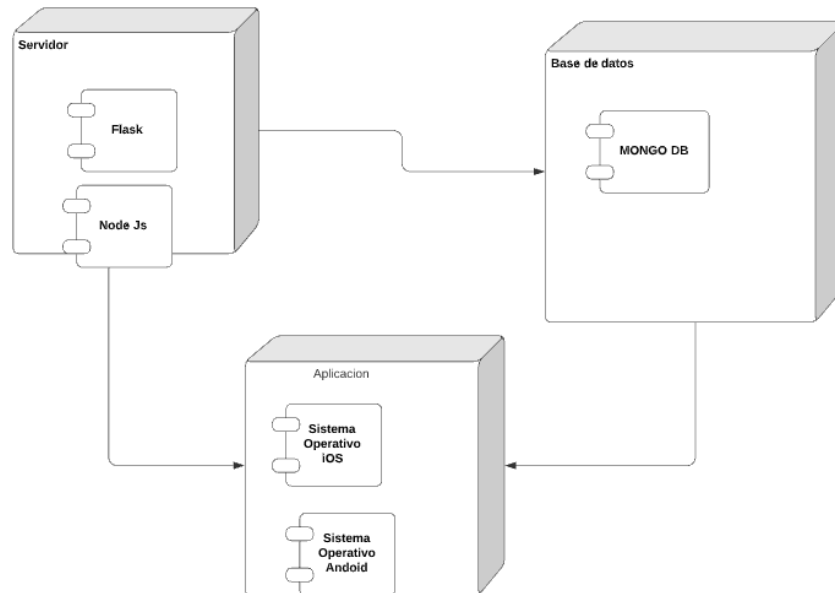
Elaborado por los autores: Daniela González y Julio Saráuz

3.2 DIAGRAMA DE DESPLIEGUE

En el siguiente diagrama de despliegue, se representa la estructura de la aplicación

Figura 26

Diagrama de Despliegue de la Aplicación



Nota. Diagrama de Despliegue de la Aplicación

Elaborado por los autores: Daniela González y Julio Saráuz

3.3 PRUEBAS: CAJA NEGRA, CARGA Y STRESS

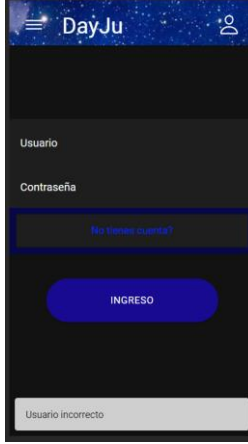

3.4.1 Pruebas de Caja Negra

Las pruebas de caja negra abordan la observación de entradas y salidas del sistema, es decir evalúa los requerimientos funcionales.

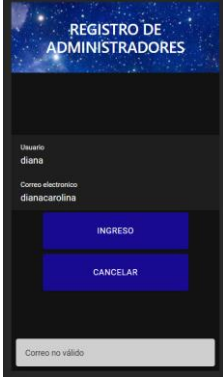

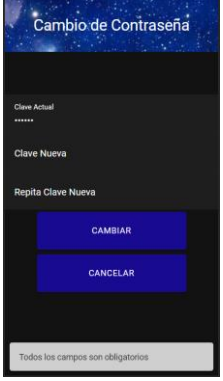
Ambiente de pruebas: La pruebas de caja negra se evalúa en un celular Xiaomi Mi A3, con 4GB de RAM y Android 10.

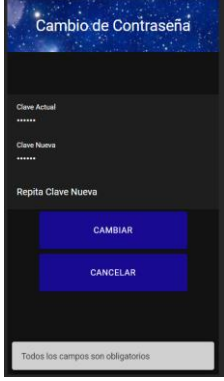
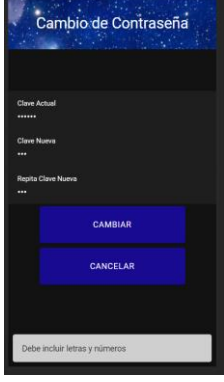
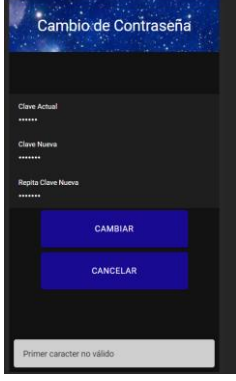
Tabla 13

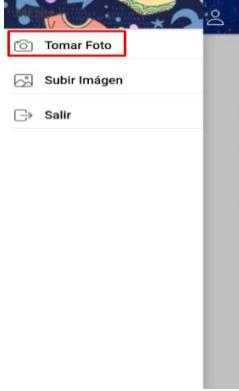


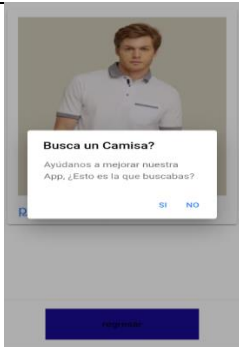
Resultados de pruebas de caja negra de la aplicación

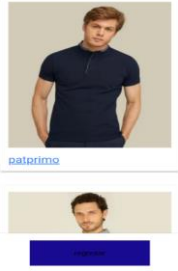


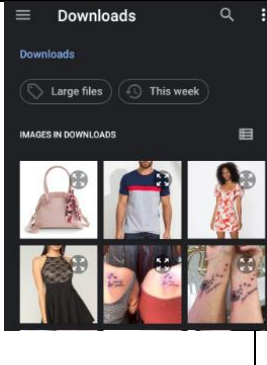
Requerimiento Funcional	Acción para analizar	Resultado	Descripción
RF1	Inicio de sesión de usuarios		<p>Intento de ingreso sin usuarios y contraseñas en el formulario.</p> <p>Mensaje de error: Usuario incorrecto.</p>
			<p>Intento de Ingreso solo introduciendo usuario y no contraseña o ingresa contraseña y no usuario.</p> <p>Mensaje de error: Usuario incorrecto.</p>



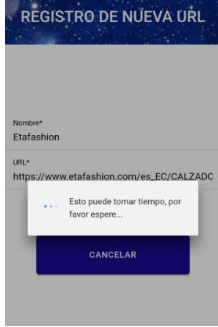
<p>RF2</p>	<p>Registro de usuarios</p>		<p>Intento de registro de un nuevo administrador sin campos.</p> <p>Mensaje de error:</p> <p>Todos los campos son obligatorios.</p>
			<p>Intento de Ingreso solo introduciendo usuario y no correo electrónico o ingresa correo electrónico y no usuario.</p> <p>Mensaje de error: Todos los campos son obligatorios.</p>

			<p>Intento de registro de administrador con un correo invalido.</p> <p>Mensaje de error: Correo no válido.</p>
			<p>Intento de verificación de código con campo vacío.</p> <p>Mensaje de error: Código incorrecto, si no ingresa el código enviado al correo electrónico.</p>
<p>RF3</p>	<p>Ingreso de iniciar sesión con clave temporal</p>		<p>Intento de cambio de contraseña con campos vacíos.</p> <p>Mensaje de error: Todos los campos son obligatorios.</p>

			<p>Intento de cambio de contraseña con 1 campo vacío.</p> <p>Mensaje de error:</p> <p>Todos los campos son obligatorios.</p>
			<p>Intento de cambio de contraseña con caracteres cortos.</p> <p>Mensaje de error:</p> <p>Debe incluir letras y números.</p>
			<p>Intento de cambio de contraseña con primer carácter de la nueva contraseña especial.</p> <p>Mensaje de error:</p> <p>Primer carácter no válido.</p>

<p>RF4</p>	<p>La aplicación permitirá al usuario capturar una imagen con su celular en tiempo real y guardar en la base de datos.</p>		<p>Dispone de un menú en esta ocasión: Tomar Foto.</p>
			<p>Se despliega un botón para poder tomar la foto en ese momento.</p>
			<p>Toma la foto y después en el botón predecir.</p>
			<p>Nos sale un alert indicándonos si lo que buscaba es una camiseta, depende la prenda que ingreso.</p>

			<p>Se visualiza las prendas similares y las tiendas donde se puede comprar.</p>
<p>RF5</p>	<p>La aplicación debe permitir subir una foto de la galería de cada usuario que tiene en su celular y guardar en la base de datos.</p>		<p>Dispone de un menú en esta ocasión: Subir Imágen.</p>
			<p>Se despliega un botón para poder abrir la galería de imágenes del almacenamiento del celular.</p>
			<p>Se selecciona la foto que se desea predecir.</p>

			<p>Nos sale un alert indicándonos si lo que buscaba es una abrigo, depende la prenda que ingreso.</p>
RF6	<p>La aplicación debe permitir ingresar un url de las tiendas que se desea scrapear y guardar en la base de datos.</p>		<p>Tenemos el botón para poder ingresar página.</p>
			<p>Ingresamos el nombre de la empresa y la URL.</p>

Nota. Análisis de cada uno de los requerimientos.

Elaborado por los autores: Daniela González y Julio Saráuz

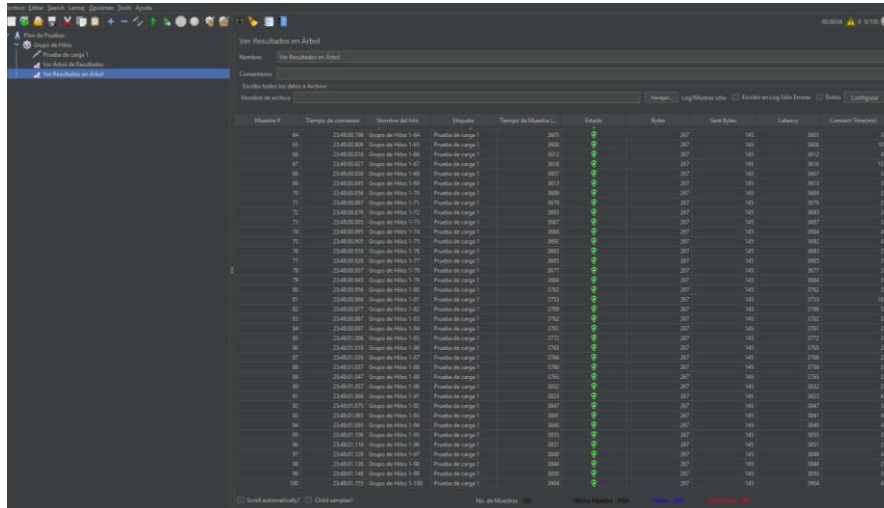
3.4.2 Prueba De Carga

Las pruebas de carga se realizaron con el programa Apache JMeter. Se consideró los procesos que más demandan mayor cantidad de usuarios recurrentes.

Prueba 1: De 100 usuarios en 1 segundo por 1 ciclo

Figura 27

Resultados en Árbol de 100 usuarios.



Nota. Se verifica que en las 100 muestras tomadas el estado indica satisfactorio.

Elaborado por los autores: Daniela González y Julio Saráuz con la herramienta Apache JMeter.

Tabla 14

Reporte final de la prueba de carga de 100 usuarios.

Etiqueta	#Muestra	Media	Min	Max	Desv.Est	%Error	Rendimiento
Prueba de carga 1	100	543	179	935	227.50	0%	52.2/sec
Total	100	543	179	935	227.50	0%	52.2/sec

Nota. Reporte final de 100 muestras tomadas.

Elaborado por los autores: Daniela González y Julio Saráuz con la herramienta Apache JMeter.

Prueba 2: De 1000 usuarios en 1 segundo por 1 ciclo

Figura 28

Resultados en Árbol de 1000 usuarios.

Muestra #	Tiempo de comienzo	Nombre del hilo	Etiqueta	Tiempo de Muestra (ms)	Estado	Bytes	Sent Bytes	Latency	Connect (ms)
964	23.54.34.934	Grupo de hilos 1-962	Prueba de carga 1	17461	OK	257	145	17461	475
965	23.54.34.937	Grupo de hilos 1-962	Prueba de carga 1	17465	OK	257	145	17465	472
966	23.54.34.938	Grupo de hilos 1-967	Prueba de carga 1	17466	OK	257	145	17466	473
967	23.54.34.937	Grupo de hilos 1-969	Prueba de carga 1	17468	OK	257	145	17468	472
968	23.54.34.934	Grupo de hilos 1-981	Prueba de carga 1	17467	OK	257	145	17467	476
969	23.54.34.941	Grupo de hilos 1-970	Prueba de carga 1	17523	OK	257	145	17523	466
970	23.54.34.934	Grupo de hilos 1-964	Prueba de carga 1	17530	OK	257	145	17530	475
971	23.54.34.944	Grupo de hilos 1-975	Prueba de carga 1	17520	OK	257	145	17520	465
972	23.54.34.941	Grupo de hilos 1-971	Prueba de carga 1	17583	OK	257	145	17583	468
973	23.54.34.943	Grupo de hilos 1-972	Prueba de carga 1	17626	OK	257	145	17626	467
974	23.54.34.943	Grupo de hilos 1-974	Prueba de carga 1	17639	OK	257	145	17639	466
975	23.54.34.947	Grupo de hilos 1-976	Prueba de carga 1	17654	OK	257	145	17654	462
976	23.54.34.944	Grupo de hilos 1-976	Prueba de carga 1	17657	OK	257	145	17657	465
977	23.54.34.942	Grupo de hilos 1-972	Prueba de carga 1	17659	OK	257	145	17659	467
978	23.54.34.946	Grupo de hilos 1-977	Prueba de carga 1	17665	OK	257	145	17665	463
979	23.54.34.948	Grupo de hilos 1-979	Prueba de carga 1	17688	OK	257	145	17688	461
980	23.54.34.952	Grupo de hilos 1-982	Prueba de carga 1	17710	OK	257	145	17710	459
981	23.54.34.950	Grupo de hilos 1-982	Prueba de carga 1	17712	OK	257	145	17712	461
982	23.54.34.953	Grupo de hilos 1-984	Prueba de carga 1	17756	OK	257	145	17756	458
983	23.54.34.950	Grupo de hilos 1-981	Prueba de carga 1	17783	OK	257	145	17783	461
984	23.54.34.953	Grupo de hilos 1-983	Prueba de carga 1	17807	OK	257	145	17807	458
985	23.54.34.950	Grupo de hilos 1-980	Prueba de carga 1	17815	OK	257	145	17815	461
986	23.54.34.954	Grupo de hilos 1-986	Prueba de carga 1	17819	OK	257	145	17819	457
987	23.54.34.953	Grupo de hilos 1-987	Prueba de carga 1	17819	OK	257	145	17819	457
988	23.54.34.959	Grupo de hilos 1-990	Prueba de carga 1	17840	OK	257	145	17840	453
989	23.54.34.958	Grupo de hilos 1-989	Prueba de carga 1	17850	OK	257	145	17850	454
990	23.54.34.959	Grupo de hilos 1-991	Prueba de carga 1	17872	OK	257	145	17872	453
991	23.54.34.961	Grupo de hilos 1-992	Prueba de carga 1	17870	OK	257	145	17870	451
992	23.54.34.962	Grupo de hilos 1-993	Prueba de carga 1	17905	OK	257	145	17905	450
993	23.54.34.959	Grupo de hilos 1-991	Prueba de carga 1	17953	OK	257	145	17953	453
994	23.54.34.962	Grupo de hilos 1-994	Prueba de carga 1	17979	OK	257	145	17979	450
995	23.54.34.964	Grupo de hilos 1-998	Prueba de carga 1	17975	OK	257	145	17975	444
996	23.54.34.967	Grupo de hilos 1-999	Prueba de carga 1	17992	OK	257	145	17992	445
997	23.54.34.967	Grupo de hilos 1-997	Prueba de carga 1	17994	OK	257	145	17994	445
998	23.54.34.965	Grupo de hilos 1-996	Prueba de carga 1	18015	OK	257	145	18015	447
999	23.54.34.967	Grupo de hilos 1-995	Prueba de carga 1	18015	OK	257	145	18015	445
1000	23.54.34.964	Grupo de hilos 1-995	Prueba de carga 1	18049	OK	257	145	18049	446

Nota. Se verifica que en las 1000 muestras tomadas el estado indica satisfactorio.

Elaborado por los autores: Daniela González y Julio Saráuz con la herramienta Apache JMeter.

Tabla 15

Reporte final de la prueba de carga de 1000 usuarios.

Etiqueta	#Muestra	Media	Min	Max	Desv.Est	%Error	Rendimiento
Prueba de carga 1	1000	9542	274	18049	4952.40	0%	52.2/sec
Total	1000	9542	274	18049	4952.40	0%	52.2/sec

Nota. Reporte final de 1000 muestras tomadas.

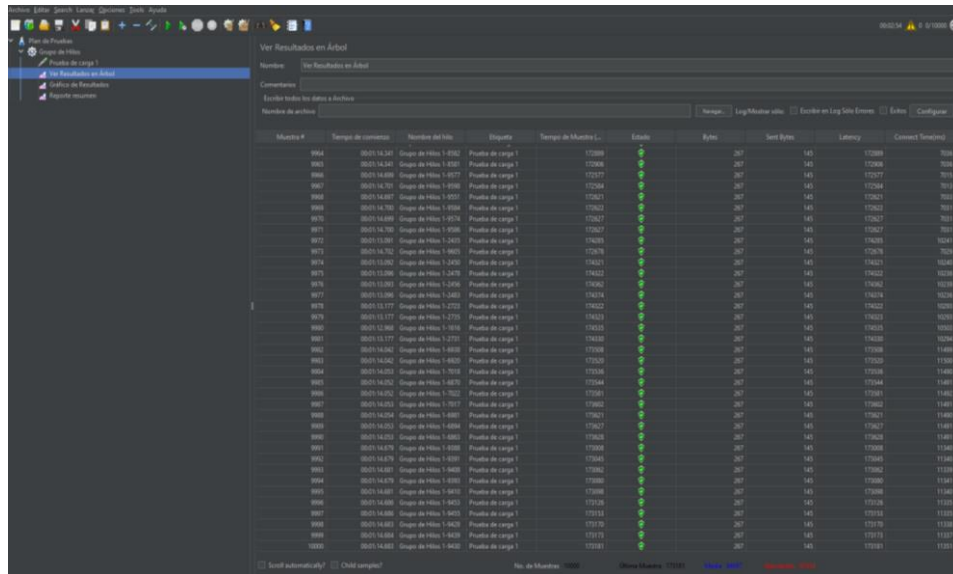
Elaborado por los autores: Daniela González y Julio Saráuz con la herramienta Apache JMeter.

Con 100 usuarios se comprueba que el rendimiento es de 52.2 solicitudes por segundo. En todos los procesos no existen solicitudes con errores.

Prueba 3: De 10000 usuarios en 1 segundo por 1 ciclo

Figura 29

Reporte final de la prueba de carga de 10000 usuarios.



Nota. Se verifica que en las 10000 muestras tomadas el estado indica satisfactorio.

Elaborado por los autores: Daniela González y Julio Saráuz con la herramienta Apache JMeter.

Tabla 16

Reporte final de la prueba de carga de 10000 usuarios.

Etiqueta	#Muestra	Media	Min	Max	Desv.Est	%Error	Rendimiento
Prueba de carga 1	10000	84697	550	174535	51514.12	4.13%	57.1/sec
Total	10000	84697	550	174535	51514.12	4.13%	57.1/sec

Nota. Reporte final de 10000 muestras tomadas.

Elaborado por los autores: Daniela González y Julio Saráuz con la herramienta Apache JMeter.

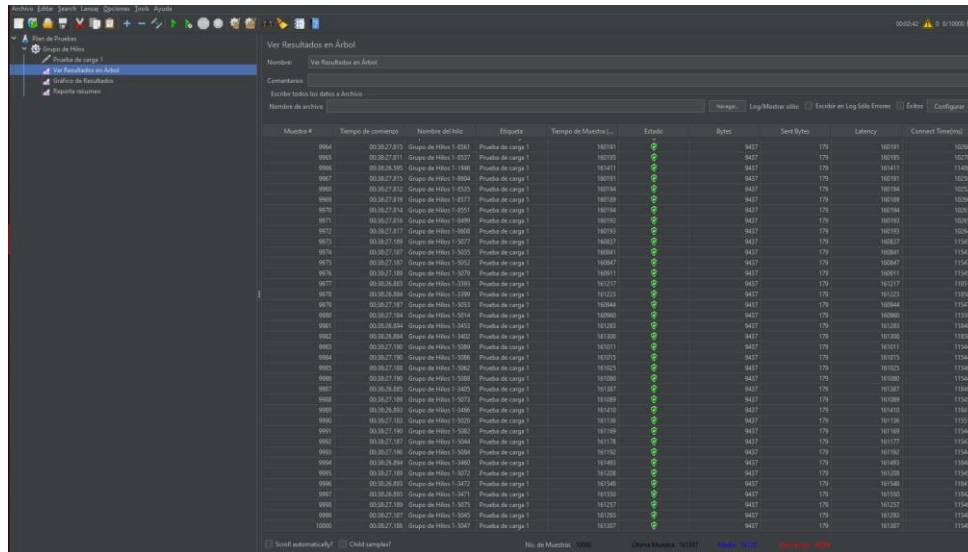
3.4.3 Prueba De Stress

Las pruebas de stress se realizaron con el programa Apache JMeter. Se consideró los procesos que más demandan mayor cantidad de usuarios recurrentes.

1. Prueba con 10000 usuarios

Figura 30

Resultados en Árbol de pruebas de stress



Nota. Se verifica que en las 10000 muestras tomadas el estado indica satisfactorio.

Elaborado por los autores: Daniela González y Julio Saráuz con la herramienta Apache JMeter.

Tabla 17

Resultado de prueba de stress con 10000 peticiones

Etiqueta	#Muestra	Media	Mín	Max	Desv.Est	%Error	Rendimiento
Prueba de carga 1	10000	74129	487	161550	49306.21	8.83%	531.05/sec
Total	10000	74129	487	161550	49306.21	8.83%	531.05/sec

Nota. Resultado de prueba de stress con 10000 peticiones

Elaborado por los autores: Daniela González y Julio Saráuz con la herramienta Apache JMeter.

2. Prueba con 100 000 usuarios

Figura 31

Resultados en Árbol de pruebas de stress

The screenshot displays the Apache JMeter results tree for a stress test. The main window shows a list of 10000 samples, each with a unique ID (e.g., 99964, 99965, etc.), a start time, a thread name (e.g., Grupo de Hilos 1-27621), a label (e.g., Prueba de carga 1), a sample time, a status (all green), and performance metrics like Bytes, Sent Bytes, Latency, and Connect (ms). The status column indicates that all samples were successful.

Nota. Se verifica que en las 10000 muestras tomadas el estado indica satisfactorio.

Elaborado por los autores: Daniela González y Julio Saráuz con la herramienta Apache JMeter.

Tabla 18

Resultado de prueba de stress con 100000 peticiones

Etiqueta	#Muestra	Media	Min	Max	Desv.Est	%Error	Rendimiento
Prueba de carga 1	100000	23594	0	290430	61840.74	83.73%	343.8/sec
Total	100000	23594	0	290430	61840.74	83.73%	343.8/sec

Nota. Resultado de prueba de stress con 100000 peticiones

Elaborado por los autores: Daniela González y Julio Saráuz con la herramienta Apache JMeter.

3.5 ANÁLISIS DE RESULTADOS

3.5.1 Prueba de Carga

Según los resultados mostrados en la Tabla 14, con 100 usuarios se comprueba que el rendimiento es de 52.2 solicitudes por segundo. En todos los procesos no existen solicitudes con errores. En la tabla 15, con 1000 usuarios se comprueba que el rendimiento es de 52.2 solicitudes por segundo. En todos los procesos no existen solicitudes con errores. En la tabla 16, con 10000 usuarios se comprueba que el rendimiento es de 57.1 solicitudes por segundo. En todos los procesos existe un 4.13% de solicitudes con errores.

3.5.2 Prueba de Stress

Según los resultados mostrados en la Tabla 17, con 10000 peticiones por segundo, existen el 8.83% de solicitudes de peticiones enviadas. En la Tabla 18, con 100000 peticiones por segundo, existe un 83.73% de peticiones enviadas, con un rendimiento de 343.8 solicitudes por minuto. Entonces se puede finalizar indicando que con 100000 peticiones la aplicación no funciona de manera adecuada las solicitudes enviadas. Lo cual afecta negativamente a la experiencia del usuario.

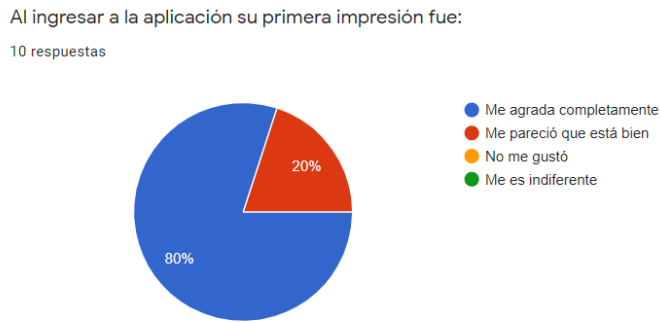
3.5.3 Pruebas de Aceptación de la aplicación

Para las pruebas de aceptación solicitamos 10 usuarios de prueba con dispositivos Android. Los usuarios experimentaron con la aplicación, y nos indicaron los siguientes resultados.

En la figura 32 se aprecia que un 20% le parece que está bien la aplicación y el 80% que le agrada completamente.

Figura 32

Primera impresión de la aplicación móvil



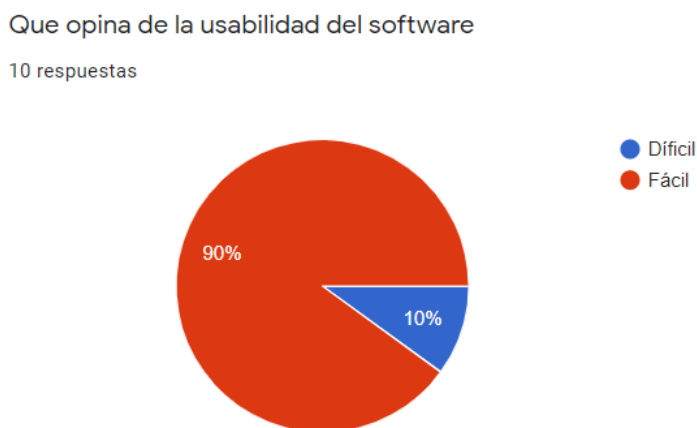
Nota. Porcentajes de la primera impresión en la aplicación móvil.

Elaborado por los autores: Daniela González y Julio Saráuz.

En la figura 33 se aprecia que un 10% opina que es difícil y un 90% que es fácil.

Figura 33

Usabilidad de la aplicación



Nota. Porcentajes de la opinión de los usuarios en la usabilidad de la aplicación.

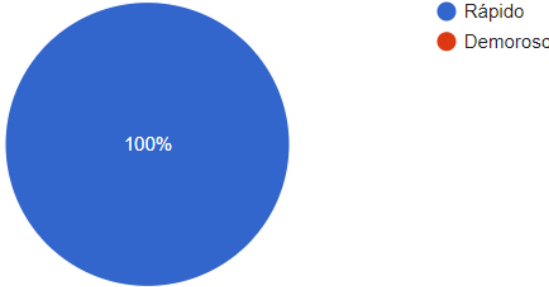
Elaborado por los autores: Daniela González y Julio Saráuz.

En la figura 34 se aprecia que el 100% opina que en el proceso de obtener una respuesta es rápido.

Figura 34

El proceso de obtener una respuesta

En el proceso de obtener una respuesta es:
10 respuestas



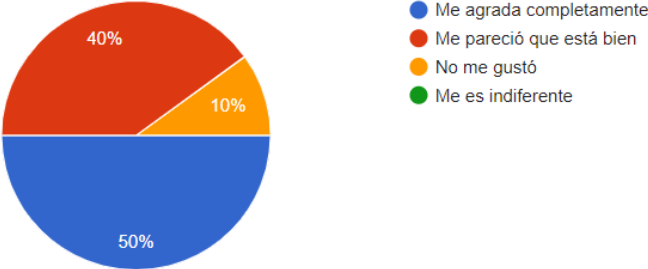
Nota. Porcentajes de la opinión de los usuarios en el proceso de obtener una respuesta. Elaborado por los autores: Daniela González y Julio Saráuz.

En la figura 35 el 10% no le gusto los colores usados en la aplicación, un 40% indica que le parece que está bien y un 50% indica que le agrada completamente.

Figura 35

Colores usados en la aplicación

Mi opinión sobre los colores usados en la aplicación es:
10 respuestas



Nota. Porcentajes de la opinión de los usuarios en los colores usados en la aplicación. Elaborado por los autores: Daniela González y Julio Saráuz.

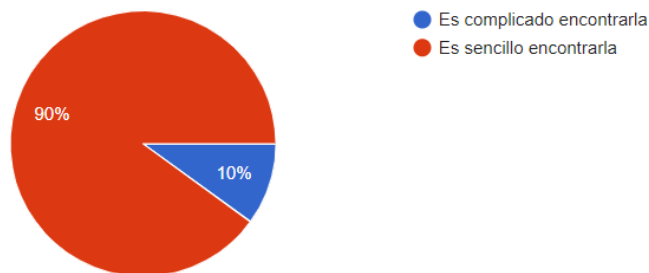
En la figura 36 el 10% de los usuarios opina que es complicado encontrar el menú y el 90% opina que es sencillo encontrar el menú.

Figura 36

Visualización del menú de la aplicación

Para visualizar el menú de la aplicación:

10 respuestas



Nota. Porcentajes de la opinión de los usuarios en la usabilidad de la aplicación.

Elaborado por los autores: Daniela González y Julio Saráuz.

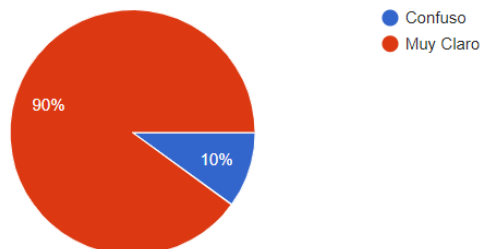
En la figura 37 el 10% de los usuarios opina que es confuso la organización de las pantallas, mientras que el 90% opina que es muy claro.

Figura 37

Organización de la información

Con respecto a la pantalla, ¿ que opina de la organización de la información?

10 respuestas



Nota. Organización de la información en las pantallas

Elaborado por los autores: Daniela González y Julio Saráuz.

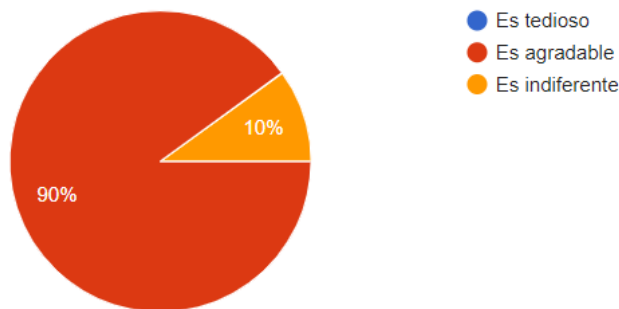
En la figura 38 el 10% de los usuarios opina que le es indiferente y el 90% opina que es agradable al interactuar con la aplicación.

Figura 38

Experiencia en la interacción con la aplicación

En general, su experiencia al interactuar con la aplicación:

10 respuestas



Nota. Experiencia en la interacción con la aplicación

Elaborado por los autores: Daniela González y Julio Saráuz.

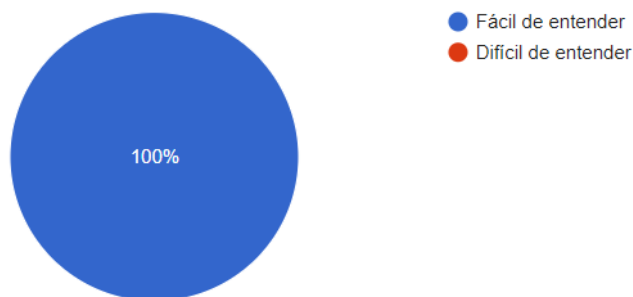
En la figura 39 el 100% de usuarios opinan que los términos son fáciles de entender.

Figura 39

Uso de los términos en toda la aplicación

El uso de términos en toda la aplicación fueron:

10 respuestas



Nota. Uso de los términos en toda la aplicación

Elaborado por los autores: Daniela González y Julio Saráuz.

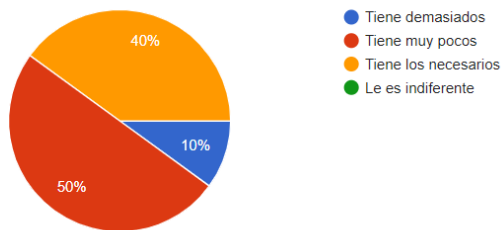
En la figura 40 el 10% indica que tiene demasiados elementos multimedia, un 40% indica que tiene los necesarios y un 50% indica que tiene muy pocos.

Figura 40

Elementos multimedia (fotos, gifs) son abundantes

¿Cree que los elementos multimedia (como fotos y gifs) utilizados en esta aplicación móvil son abundantes?

10 respuestas



Nota. Elementos multimedia (fotos, gifs) son abundantes

Elaborado por los autores: Daniela González y Julio Saráuz.

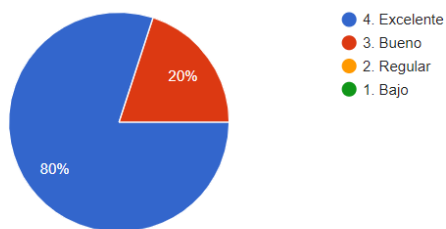
En la figura 41 el 20% de los usuarios indica que la importancia de las ayudas es Buena, y el 80% indica que es Excelente.

Figura 41

Importancia de las ayudas

Califique de 1 a 4 la importancia de las ayudas dispuestas en la aplicación. Siendo 4 la más importante y 1 la menos importante.

10 respuestas



Nota. Importancia de las ayudas en la aplicación.

Elaborado por los autores: Daniela González y Julio Saráuz.

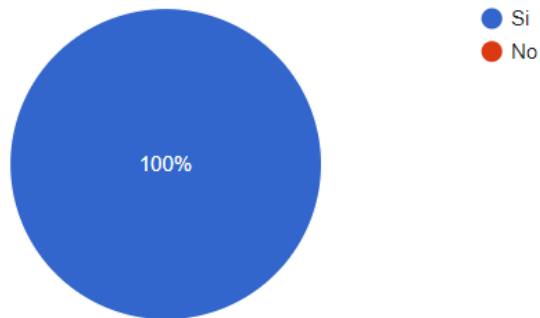
En la figura 42 el 100% indica que es fácil repetir tareas sobre la aplicación.

Figura 42

Facilidad de repetir tareas y/o actividades

Le es fácil repetir tareas y/o actividades sobre la aplicación:

10 respuestas



Nota. Facilidad de repetir tareas y/o actividades

Elaborado por los autores: Daniela González y Julio Saráuz.

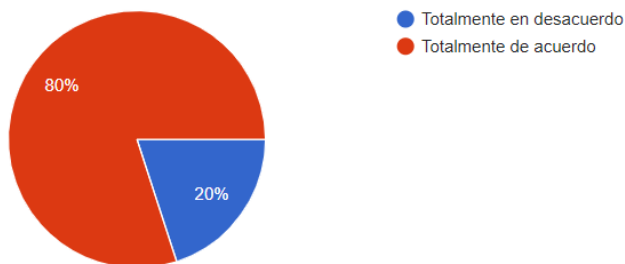
En la figura 43 el 20% indica que está totalmente en desacuerdo con lo fácil que es usar la aplicación y el 80% indica que está totalmente de acuerdo.

Figura 43

Facilidad de usar la aplicación

En general , estoy satisfecho con lo fácil que es usar esta aplicación:

10 respuestas



Nota. Facilidad de usar la aplicación

Elaborado por los autores: Daniela González y Julio Saráuz.

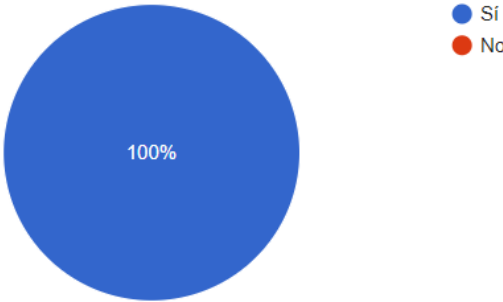
En la figura 44 el 100% de los usuarios indica que la interfaz es agradable.

Figura 44

Interfaz de la aplicación

La interfaz de esta aplicación es agradable

10 respuestas



Nota. Interfaz de la aplicación

Elaborado por los autores: Daniela González y Julio Saráuz.

CONCLUSIONES

Se cumplió el objetivo general que es construir una aplicación móvil con redes neuronales convolucionales para reconocer imágenes de prendas de vestir de tiendas en línea en Ecuador y la utilización de la metodología SCRUM fue importante para el desarrollo del proyecto.

El modelo MVC nos permitió identificar los requerimientos, necesidades y alcances de la aplicación móvil.

El uso de las redes neuronales combinado con diferentes web services permite la creación de una arquitectura para un software adaptable que abre paso a la inteligencia artificial en los dispositivos móviles.

La réplica de sistemas que solo se usaban en dispositivos de escritorio a los dispositivos móviles, mejorando la disponibilidad y accesibilidad de los mismos a los usuarios, por tratarse de una herramienta al alcance del usuario promedio.

MongoDB Atlas logró de ser de mucha utilidad en la fase de autenticación de usuarios, en la fase de guardar las fotos que se va a predecir con los servicios de Flask y Nest Js, y en guardar las imágenes encontradas en las páginas ingresadas de las tiendas en línea de Ecuador.

Se realizó una prueba de aceptación con una muestra de 10 usuarios en la fase final de la aplicación. Los resultados nos muestran que el 90% de usuarios opina que es fácil usar la aplicación, el 90% indica que la experiencia al interactuar con la aplicación es agradable y un 100% indica que es fácil repetir tareas y/o actividades.

RECOMENDACIONES

Implementar un módulo dentro del reconocimiento de imágenes para poder identificar los colores de las prendas de vestir, de las imágenes proporcionadas por cada usuario.

Para trabajos futuros se recomienda hacer un despliegue de las aplicaciones para las tiendas de Play Store de Android y App Store de iOS.

Mejorar el dataset de Fashion Mnist haciendo más robusto con imágenes que los usuarios ingresen de más prendas y almacenar en un repositorio e ir etiquetando para que reconozca la red neuronal, así de esta forma mejorar la precisión al momento de predecir las prendas de vestir.

Se puede cambiar el modelo de predicción por otro de cualquier producto como: carros, vegetales, útiles escolares, etc., Se tendría que buscar un dataset y hacer ciertas configuraciones en la red convolucional.

Realizar una campaña de publicidad a través de Facebook e Instagram para que más tiendas en línea del Ecuador se registren y los usuarios puedan conocer de los beneficios que obtendrá nuestra aplicación.

REFERENCIAS

- A., D. (9 de septiembre de 2020). *HOSTINGER TUTORIALES*. Obtenido de <https://www.hostinger.es/tutoriales/que-es-angular>
- A., R. Q., & I., C. M. (2011). Redes neuronales artificiales para el procesamiento de imágenes, una revisión de la última década. *REVISTA DE INGENIERÍA ELÉCTRICA, ELECTRÓNICA Y COMPUTACIÓN*, 11.
- Abellán, E. (05 de marzo de 2020). *WAM*. Obtenido de <https://www.wearemarketing.com/es/blog/metodologia-scrum-que-es-y-como-funciona.html>
- Adeva, R. (03 de marzo de 2021). *AZ adsl zone*. Obtenido de <https://www.adslzone.net/reportajes/software/que-es-android/>
- Asensio, E. O. (2016). Sistema de reconocimiento de gestos de la mano basado en procesamiento de imagen y Redes Neuronales Convolucionales. 69.
- Cornieles, P. (29 de abril de 2020). IA LATAM. Obtenido de <https://ia-latam.com/2020/04/29/la-pandemia-genero-17-millones-nuevos-compradores-online-en-america-latina/>
- CRM, A. S. (16 de octubre de 2018). *AITANA*. Obtenido de <https://blog.aitana.es/2018/10/16/visual-studio-code/>
- Cruz, N. K. (2017). El Comercio Electrónico en el Ecuador. 4.
- Datta Business Innovation. (18 de enero de 2020). Obtenido de <https://datta.com.ec/articulo/datos-curiosos-sobre-el-consumo-de-los-servicios-digitales->

Portaltic/EP. (10 de agosto de 2019). *E&N*. Obtenido de

<https://www.estrategiaynegocios.net/tecnologia/1309087-330/cu%C3%A1ntas-b%C3%BAsquedas-se-hacen-en-google-en-un-d%C3%ADa#:~:text=El%2056%2C1%25%20de%20la,env%C3%ADo%20de%2018.100.000%20mensajes.>

Robledano, Á. (23 de septiembre de 2019). *OpenWebinars*. Obtenido de

<https://openwebinars.net/blog/que-es-python/>

Silva, S., & Freire, E. (23 de noviembre de 2019). *Bootcamp AI*. Obtenido de

<https://bootcampai.medium.com/redes-neuronales-convolucionales-5e0ce960caf8#:~:text=Las%20redes%20neuronales%20convolucionales%20es,poder%20diferenciar%20unos%20de%20otros.&text=La%20red%20toma%20como%20entrada%20los%20pixeles%20de%20una%20imagen.>

Simón, J. N. (2012). RECONOCIMIENTO DE OBJETOS MEDIANTE SENSOR 3D KINECT. 109.

Suárez, J.D. (2017). Redes Convolucionales en R. 78.

THE NIELSEN COMPANY. (2018). COMERCIO CONECTADO. NIELSEN, 19.