



UNIVERSIDAD POLITÉCNICA SALESIANA
SEDE CUENCA
CARRERA DE INGENIERÍA DE SISTEMAS

DISEÑO Y DESARROLLO DE UNA APLICACIÓN PWA PARA UN E-COMMERCE QUE PERMITA LA RECOMENDACIÓN DE PRODUCTOS BASADO EN ALGORITMOS INTELIGENTES

Trabajo de titulación previo a la obtención
del título de Ingeniero de Sistemas

AUTOR: ÁNGEL MESIAS JADAN CORTE

TUTOR: ING. REMIGIO ISMAEL HURTADO ORTIZ, PhD.

Cuenca - Ecuador

2024

**CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA DEL TRABAJO DE
TITULACIÓN**

Yo, Ángel Mesias Jadan Corte con documento de identificación N° 0106405236,
manifiesto que:

Soy el autor y responsable del presente trabajo; y, autorizo a que sin fines de lucro la
Universidad Politécnica Salesiana pueda usar, difundir, reproducir o publicar de manera
total o parcial el presente trabajo de titulación.

Cuenca, 09 de enero de 2024

Atentamente,

Ángel Mesias Jadan Corte

0106405236

**CERTIFICADO DE CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE
TITULACIÓN A LA UNIVERSIDAD POLITÉCNICA SALESIANA**

Yo, Ángel Mesias Jadan Corte con documento de identificación N° 0106405236, expreso mi voluntad y por medio del presente documento cedo a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que soy autor del Proyecto técnico: “Diseño y desarrollo de una aplicación PWA para un E-Commerce que permita la recomendación de productos basado en algoritmos inteligentes”, el cual ha sido desarrollado para optar por el título de: Ingeniero de Sistemas, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En concordancia con lo manifestado, suscribo este documento en el momento que hago la entrega del trabajo final en formato digital a la Biblioteca de la Universidad Politécnica Salesiana.

Cuenca, 09 de enero de 2024

Atentamente,

Ángel Mesias Jadan Corte

0106405236

CERTIFICADO DE DIRECCIÓN DEL TRABAJO DE TITULACIÓN

Yo, Remigio Ismael Hurtado Ortiz con documento de identificación N° 0104621388, docente de la Universidad Politécnica Salesiana declaro que bajo mi tutoría fue desarrollado el trabajo de titulación: DISEÑO Y DESARROLLO DE UNA APLICACIÓN PWA PARA UN E-COMMERCE QUE PERMITA LA RECOMENDACIÓN DE PRODUCTOS BASADO EN ALGORITMOS INTELIGENTES, realizado por Ángel Mesias Jadan Corte con documento de identificación N° 0106405236, obteniendo como resultado final el trabajo de titulación bajo la opción Proyecto técnico que cumple con todos los requisitos determinados por la Universidad Politécnica Salesiana.

Cuenca, 09 de enero de 2024

Atentamente,

Ing. Remigio Ismael Hurtado Ortiz, PhD.

0104621388

DEDICATORIA

La siguiente tesis está dedicada primeramente a Dios, que gracias a él que me ha brindado la salud y las ganas de estudiar, a mis padres Manuel Jesús Jadan Zhispón y Ana María Corte Fajardo, gracias a su apoyo incondicional durante toda la carrera de estudiante, que pese a todas las dificultades que se han presentado, siempre han estado apoyándome, mismos que me han guiado con los valores, para ser una persona de bien.

También a mis hermanas, quienes han sido de apoyo en los momentos más difíciles y han creído en mí, también a las todas las personas que alguna vez, de alguna forma me han dado su apoyo y consejo.

Ángel Mesias Jadan Corte

AGRADECIMIENTOS

Primeramente, quiero agradecer a Dios y la Virgen María, por la salud y la vida, quienes han sido mi compañía durante todo el tiempo, sobre todo en los momentos más difíciles se ha sentido su presencia. Agradezco a la Universidad Politécnica Salesiana, quien me ha abierto las puertas, para poder obtener los conocimientos para ser un buen profesional y sobre todo ser una buena persona en la sociedad.

Agradezco de manera muy enorme a mis padres quienes han confiado firmemente en mi estar siempre ahí brindándome su amor y apoyo, pues gracias a ellos que han hecho todo lo posible llegar a esta etapa de mi vida, ya que sin ellos no sería posible haber alcanzado esta meta en mi vida.

Agradezco a mis hermanas, que siempre me han dicho sentirse orgullosas de mí, que, en situaciones económicas muy malas, han sido quienes me han levantado de una u otra manera.

Agradezco a mis profesores por su paciencia y dedicación para transmitir todo su conocimiento y experiencia profesional.

Ángel Mesías Jádan Córte

RESUMEN

Hoy en día existen muchas tiendas virtuales, y a su vez una cantidad significativa de información en internet, esto es una ventaja en ciertas ocasiones, pero a su vez una desventaja ya que esta sobrecarga de información puede llevar a perder el interés de los usuarios y a retrasar la búsqueda de información. Esto de igual manera afecta a las tiendas virtuales, pues estas al contar con una gran cantidad de productos, de igual manera se vuelve difícil para el cliente a la hora de elegir un producto que le sea necesario y de utilidad, esto hace que el propósito que es la comodidad y facilidad que deben brindar queden olvidadas, esto hace que los clientes no se sientan atraídos a realizar sus compras.

A diferencia de una tienda física la recomendación por parte de la tienda no es posible, puesto que por la gran cantidad de concurrencia, no es posible tener un contacto directo entre cada cliente y la tienda, pues cada cliente varia sus gustos y necesidades, una solución a corto plazo seria el contratar a más personal para atención al cliente, pero este lleva a nuevos problemas, como por ejemplo el incremento en los costos del negocio, otro problema es que en cuanto más crezca la tienda a nivel mundial, se requeriría una mayor cantidad de personal. En la actualidad ya existe una gran cantidad de tiendas virtuales, pero un problema para los desarrolladores de software es que surge la necesidad de desarrollar aplicaciones compatibles con distintas plataformas de desarrollo.

Para solucionar estos problemas se ha desarrollado un sistema que incorpora uno de los algoritmos inteligentes para realizar un sistema de recomendación de los productos, uno de los más rápidos y sencillos de implementar que es el algoritmo KNN. Este algoritmo permite realizar una recomendación de una cantidad de productos, en base a los atributos que se requieran. Estos algoritmos son de gran ayuda para que un cliente pueda encontrar los productos que requieran al realizar una compra, ya que con base a los datos de entrada realizará la recomendación. Estos datos pueden ser gustos de usuarios, historial de compras, en este caso el aprendizaje se da en función de la información publicada de los productos en la red social Facebook. De esta manera, se toman los comentarios que reciban los productos y con base a la similitud con respecto a otros comentarios, se realiza las recomendaciones. En cuanto al problema con el desarrollo de aplicaciones para las diferentes plataformas, existen varias tecnologías para el desarrollo de las plataformas, se ha optado por la implementación con el framework Flutter, para realizar la aplicación, puesto que la aplicación puede ser implementada para ios, Android y web.

Como trabajo futuro se pueden incorporar otros algoritmos de recomendación que puedan soportar la gran cantidad de información, puesto que el KNN, a medida que aumenta la cantidad de productos, su proceso se vuelve mucho más lento, es decir, no es un algoritmo escalable. Además, aumentar la funcionalidad en la tienda virtual.

Palabras clave: Asistentes virtuales, algoritmos inteligentes, tienda virtual, plataforma.

ABSTRACT

Nowadays, there are many online stores, along with a significant amount of information on the internet. This is an advantage in certain situations, but it can also be a disadvantage since the overload of information can lead to a loss of user interest and delay the search for information. This also affects online stores, as having a large number of products makes it difficult for customers to choose a product that is necessary and useful. This situation causes the intended convenience and ease that online stores should provide to be forgotten, making customers less inclined to make purchases.

In contrast to a physical store, providing recommendations from the store is not feasible due to the large number of customers, making direct contact between each customer and the store impractical. Each customer has varying tastes and needs. A short-term solution would be to hire more customer service personnel, but this leads to new problems, such as increased business costs. Another problem is that as the store grows globally, more personnel would be required. Currently, there are numerous online stores, but a challenge for software developers is the need to develop applications compatible with different development platforms.

To address these issues, a system has been developed that incorporates one of the intelligent algorithms to create a product recommendation system. One of the fastest and simplest algorithms to implement is the KNN (k-nearest neighbors) algorithm. This algorithm allows for the recommendation a number of the products based on the required attributes. These algorithms are a great support for customers when making a purchase, as they make recommendations based on input data such as user preferences and purchase history. In this case, learning is based on the information published about products on the social network Facebook. Product recommendations are made by considering the comments received and finding similarities with other comments.

Regarding the problem of developing applications for different platforms, there are various technologies available. The implementation has chosen the Flutter framework for developing the mobile application, as it can be deployed for iOS, Android, and web.

As future work, other recommendation algorithms that can handle large amounts of information could be incorporated. The KNN algorithm becomes slower as the number of products increases, making it less scalable. Additionally, enhancing the functionality of the virtual store is a potential avenue for future development.

Keywords: virtual assistants, intelligent algorithms, virtual store, platform.

INDICE DE CONTENIDO

1. INTRODUCCIÓN	13
2. PROBLEMA	13
3. OBJETIVOS GENERALES Y ESPECÍFICOS	15
3.1. OBJETIVO GENERAL	15
3.2. OBJETIVOS ESPECÍFICOS	15
4. REVISIÓN DE LA LITERATURA O FUNDAMENTOS TEÓRICOS	15
4.1. Aplicaciones híbridas	16
4.2. PWA (Progressive Web App)	16
4.3. FLUTTER	17
4.4. DEFINIENDO WIDGETS Y ELEMENTOS	17
4.5. MODELS	17
4.6. MODULS	18
4.7. SERVICES	18
4.8. DART	18
4.8.1. GETX	18
4.9. ARQUITECTURA DE MICROSERVICIOS.....	20
4.9.4. Aplicar CI/CD por servicio	21
4.9.5. Aplicar alta disponibilidad por servicio (HA)/decisiones sobre clústeres.....	22
4.10. DJANGO.....	23
4.10.1. DJANGO REST-FRAMEWORK.....	23
4.10.2. HTTP.....	23
4.10.3. BASES DE DATOS SOPORTADAS.....	24
4.11. PROCESOS DE PRUEBAS	25
4.11.1. Modelo en V.....	25
4.11.2. Integración entre desarrollo y verificación.....	26
4.11.3. Pruebas unitarias	27
4.11.4. Pruebas de integración.	27
4.11.5. Prueba del sistema.....	27
4.11.6. Validación del producto.	27
4.11.7. Modelo V	27
4.11.8. Aplicación del modelo	28
4.12. Algoritmos inteligentes	29
4.12.1. Historia.....	30
4.12.2. Popularización del término IA y su impacto en la cultura.	31

4.12.3. K-Nearest Neighbors Algorithm (KNN).....	34
5. MARCO METODOLÓGICO.....	35
5.1. METODOLOGÍA DE ANÁLISIS DE DATOS	35
5.2. METODOLOGIA SCRUM	36
6. RESULTADOS.....	37
6.1. Introducción	38
6.2. Propósito	38
6.3. Alcance.....	38
6.4. Visión general del producto.	38
6.5. Referencias	39
6.6. Especificación de requerimientos.....	39
6.2. Arquitectura.....	41
6.2.1. Vista de desarrollo.....	41
6.2.2. Vista física.....	42
6.3. Vista API.....	42
6.3. Fases de desarrollo	42
6.3.1. Primera fase, Sprint 1	43
6.3.2. Segunda fase, Sprint 2.....	46
7. CRONOGRAMA	56
8. PRESUPUESTO.....	58
9. CONCLUSIONES.....	59
10. Recomendaciones	60
Referencias Bibliográficas	60

Índice de tablas

TABLA 1. RF – MODULO PARA COMPRA DE PRODUCTOS.....	39
TABLA 2. RF – MODULO PARA RECOMENDACIÓN DE PRODUCTOS.....	40
TABLA 3. DISEÑO DE SPRINT DE FORMA GENERAL CON SUS ACTIVIDADES.....	43
TABLA 4. ACTIVIDADES DE LA FASE DE TRANSICIÓN.....	43
TABLA 5. ACTIVIDADES REALIZADAS PARA CORRECCIÓN DE ERRORES DE LA APLICACIÓN.....	46
TABLA 6. DATOS DE PARÁMETROS PARA ALGORITMO DE RECOMENDACIÓN.....	54

Índice de figuras

FIGURA 1. COMPARACIÓN DE APLICACIÓN MONOLÍTICA ANTE UNA DE MICROSERVICIOS	20
FIGURA 2. DIFERENCIA ENTRE UNA API Y SERVICIO MONOLÍTICO.	22
FIGURA 3. MODELO EN V DE PROCESO DE PRUEBA DE UNA APLICACIÓN	29
FIGURA 4. HISTORIA DE LA INTELIGENCIA ARTIFICIAL. (ABELIUK & GUTIÉRREZ, 2021).....	33
FIGURA 5. DIAGRAMA DE VENN QUE MUESTRA LA RELACIÓN ENTRE DISTINTAS SUBÁREAS DE LA INTELIGENCIA ARTIFICIAL. (ABELIUK & GUTIÉRREZ, 2021).....	34
FIGURA 6 ARQUITECTURA GENERAL DE LA APLICACIÓN	35
FIGURA 7. ESTRUCTURA ESPECIFICA DE REQUERIMIENTOS.....	38
FIGURA 8. ARQUITECTURA GENERAL DE LA APLICACIÓN.	39
FIGURA 9. ARQUITECTURA EN 3 CAPAS.	41
FIGURA 10. DIAGRAMA DE VISTA FÍSICA DE DESPLIEGUE DE APLICACIÓN.	42
FIGURA 11. ARQUITECTURA DE APLICACIÓN.....	42
FIGURA 12. ESPECIFICACIONES DEL SERVIDOR.....	44
FIGURA 13. CONSUMO GLOBAL DEL INTERNET. (PORTALANZA, Y OTROS, 2022).....	45
FIGURA 14. DIAGRAMA DE ACTIVIDADES DEL USUARIO CLIENTE, PARA REALIZAR COMPRA	45
FIGURA 15. VENTANA INICIAL SIN INICIO DE SESIÓN	46
FIGURA 16. VENTANA DE REGISTRO.	47
FIGURA 17. FORMULARIO DE INICIO DE SESION.	47
FIGURA 18. VENTANA CON SESIÓN INICIADO	48
FIGURA 19. VISTA DE PRODUCTO CON LISTA DE PRODUCTOS RECOMENDADOS.	48
FIGURA 20. COMENTARIO DE SEGUIDOR DE LA PÁGINA.....	49
FIGURA 21. COMENTARIO A PUBLICACIÓN.....	49
FIGURA 22. VISTA DE PRODUCTO CON SUS RECOMENDACIONES.....	50
FIGURA 23. NUEVO COMENTARIO A OTRA PUBLICACIÓN DE PRODUCTO.....	51
FIGURA 24. VISTA DE PRODUCTO CON NUEVAS RECOMENDACIONES.	51
FIGURA 25. VISTA DE PRODUCTO AGREGADO AL CARRITO DE COMPRAS.	52
FIGURA 26. LISTADO DE PRODUCTOS PARA COMPRAR Y PROCEDER CON EL PAGO.....	52
FIGURA 27. VISTA DE BOTÓN DE PAGOS.	53
FIGURA 28. VENTANA DE REGISTRO DE DATOS DE TARJETA DE CRÉDITO.	53
FIGURA 29. ARQUITECTURA DE MODELO DE RECOMENDACIÓN.....	54
FIGURA 30. GRAFICA QUE PUBLICACIONES Y SUS COMENTARIOS.	55
FIGURA 31. TABLA DE RESULTADOS DE API DE FACEBOOK, PROCESADO	55
FIGURA 32. RESULTADO DE RECOMENDACIONES	55

1. INTRODUCCIÓN

Al día de hoy las nuevas tecnologías han ocupado el espacio de la cotidianidad de las personas, estas cada vez han sido de apoyo para la humanidad, en ciertas épocas han existido los llamados revoluciones tecnológicas, con el descubrimiento de la electricidad la tecnología se ha podido ver un gran salto, pues con ello se han podido crear una serie de inventos, entre la que destaca la computadora, que, a partir de los transistores, no ha parado de tener una evolución.

El comercio y la comunicación han sido entre las necesidades que la humanidad siempre ha requerido mejorar, para ello con el tiempo se ha ido evolucionando de diferentes maneras, en la antigüedad se tenía que atravesar un

2. PROBLEMA

Uno de los problemas principales al desarrollar aplicaciones móviles, es su capacidad en ciertos dispositivos, pues no cuentan con la cantidad de almacenamiento disponible, por este motivo al requerir instalar otras aplicaciones, la solución es eliminar videos, fotos, audios o desinstalar otras aplicaciones. (Linares, 2020)

Otras aplicaciones son las llamadas web e híbridas, pero a diferencia de las aplicaciones nativas, estas no permiten que el dispositivo sea utilizado al máximo, pues existen ciertos recursos que no pueden ser accedidos por estas aplicaciones. (El financiero, 2012)

Al utilizar aplicaciones web, es que estas solo permiten su uso al tener una conexión a internet, pese a que no necesitan ser instaladas en el dispositivo, además de ser multiplataforma.

Otro tipo de aplicaciones son las aplicaciones híbridas, este tipo de aplicaciones tienen la ventaja de verse y ejecutarse como las aplicaciones nativas, pero su desarrollo es mucho más rápido ya que utilizan el mismo código para dispositivos móviles como Android e

IOS, esto hace que su desarrollo sea mucho más económico que una aplicación nativa, pero la desventaja de estas es que su velocidad y potencia no es la misma que una nativa. (García C., 2021)

Últimamente con la evolución de las aplicaciones móviles, se han buscado diferentes tipos de aplicaciones, una de ellas, son las aplicaciones PWA, que busca abordar este desafío, de tener la velocidad y potencia de una aplicación nativa y el ágil desarrollo de una aplicación híbrida, pese a que esta tecnología no es nueva, en los últimos años se ha vuelto más popular. Entre sus características son:

- Mejor rendimiento.
- Tiempos de carga muy reducidos.
- Interfaz similar a la de una aplicación nativa.
- Actualización automática.
- No ocupan espacio en la memoria.
- Diseño responsive.
- Mejor experiencia de usuario.
- Notificación push.
- Protocolo seguro TLS.
- Sin conexión a Internet.
- Menor coste de desarrollo.

En el Ecuador en los últimos años previo a la pandemia actual que se encuentra a nivel mundial, estaba en constante crecimiento, y de esta forma surge la necesidad de gestionar, recomendar y brindar seguridad en los diferentes servicios y rutas turísticas, por medio de tickets de viajes, este incremento se da luego de ver las caídas del precio del petróleo que se había dado. (Romero Fernández, Álvarez Gómez, & Santiago Chavéz, 2017)

3. OBJETIVOS GENERALES Y ESPECÍFICOS

3.1. OBJETIVO GENERAL

Diseñar y desarrollar una aplicación PWA para un E-Commerce que permita la recomendación de productos basado perfiles de compradores a través del uso de técnicas inteligentes.

3.2. OBJETIVOS ESPECÍFICOS

OE1. Revisar el estado del arte sobre PWA y recomendación de productos

basados en el perfil y contenido

OE2. Definir una arquitectura basada en microservicios para el consumo y

publicación de productos.

OE3. Diseñar e implementar un algoritmo inteligente que permita recomendar productos en base a los perfiles o contenidos de redes sociales.

OE4. Validar y probar la aplicación con métricas de rendimiento y confiabilidad, además del algoritmo inteligente con medidas de precisión y la matriz de confusión.

4. REVISIÓN DE LA LITERATURA O FUNDAMENTOS TEÓRICOS

En esta parte se estudiará y conocerá los principales conceptos para el desarrollo de la aplicación, el estado de arte, conceptos y tecnologías que se utilizaran como los frameworks de desarrollo, servidor web, hipervisor de aplicaciones, los tipos de procesos de prueba, el proceso que conlleva los algoritmos inteligentes.

4.1. Aplicaciones nativas

Una aplicación nativa, son aplicaciones que son construidas específicamente para un dispositivo con un sistema operativo en específico, una ventaja para estas aplicaciones de este tipo, que, al ser desarrollada de forma nativa, estas aplicaciones pueden utilizar todos los recursos de cada dispositivo, además de contar con la interfaz gráfica nativa del dispositivo.

4.1. Aplicaciones híbridas

Una característica que es de la reutilización del código para las diferentes plataformas, generalmente utilizando tecnologías web como HTML, JavaScript y CSS, con la diferencia que se puede utilizar los recursos del dispositivo a diferencia de la aplicación web, que limita este acceso, desde una aplicación híbrida se puede acceder de igual manera a recursos como son los sensores y cámaras.

Entre las desventajas más importantes es la interfaz gráfica que se ofrece pues al ser de forma general no se tiene las interfaces nativas del dispositivo, haciendo un poco confuso ante el usuario, otra de las desventajas es que en dispositivos con recursos limitados estas aplicaciones pueden hacer que el dispositivo sea más lento y tenga problemas de rendimiento, por el uso de contenedores web de estas aplicaciones.

4.2. PWA (Progressive Web App)

Las aplicaciones PWA, es una metodología de desarrollo rápida, esta técnica de desarrollo busca aprovechar al máximo las capacidades y recursos de las diferentes plataformas de desarrollo, por lo que no se toma en cuenta los requerimientos rígidos, por decir así.

Hoy en día hay muchos framework que están utilizando aplicaciones pwa, para desarrollar aplicaciones que funcionen como nativas, pero para diferentes plataformas, en especial para dispositivos móviles y aplicaciones web, pero también como aplicaciones para escritorio.

4.3. FLUTTER

Flutter es un framework de frontend para desarrollar modernas, nativas y reactivas aplicaciones, para Android y IOS, y ya en sus últimas versiones para aplicaciones web y escritorio, y para dispositivos embebidos como Raspberry Pi, home, automóviles y otros. (Napoli, 2020)

Flutter es un proyecto alojado en Github, de código abierto, con contribuciones de Google y la comunidad. Este framework tiene como motor de programación Dart, que es un moderno "lenguaje de programación orientado a objetos", para compilar en nativo código ARM y código JavaScript.

4.4. DEFINIENDO WIDGETS Y ELEMENTOS

El UI de Flutter está basado en el uso de widgets, que son pequeños fragmentos de gráficos, esto es una gran ventaja ante otros frameworks, ya que a medida que la aplicación crece se vuelve complejo al realizar widgets que cambien de acuerdo a sus estados, pues se puede optimizar actualizando ciertos widgets que deben de cambiar y no actualizar toda la vista, es aquí donde Flutter implementa un gestor de estados, para controlar el estado de los widgets.

4.5. MODELS

En Flutter que es un framework que utilizar el lenguaje de programación Dart, y este a su vez está orientado a objetos, esta característica es facilita la programación de software de esta manera, en este directorio podemos colocar todos los modelos para nuestro sistema y sea mucho más fácil y escalable. Además, al programar utilizando modelos, será mucho más fácil realizar la comunicación entre la aplicación REST para realizar las diferentes peticiones http.

4.6. MODULS

Flutter al contar con widgets para su manejo, es importante distribuir y dividir por partes, en moduls, podemos realizar este control, pues se puede dividir por directorios, para los componentes a mostrar y este a su vez estar ligado a un controlador que será el encargado de manejar los estados, la dinámica y comportamientos del widget de cada vista.

4.7. SERVICES

El services, nos permite realizar una conexión entre nuestros modelos a un backend, para realizar las diferentes operaciones http, y desde del controlador que asignemos a cada services, nos brindara una mejor comprensión, de lo que se realiza.

4.8. DART

“Dart es un lenguaje de programación desarrollado por Google”, mismo que es el motor del framework de flutter, pero Dart, también puede ser utilizado como herramienta para desarrollar aplicaciones para plataformas de escritorio, web, para lado de servidores y sobre todo para el desarrollo de aplicaciones móviles. Este último nos permite tener una experiencia muy buena, al ser una herramienta que nos permite crear aplicaciones de alto nivel. (Napoli, 2020)

4.8.1. GETX

GetX, es una librería de Flutter, para manejar los estados de la aplicación, de alto rendimiento, con la característica de inyección inteligente, también con el manejo de rutas de forma rápida y ligera. (Pub.dev, s.f.)

En Flutter es importante contar con un gestor de estados para controlar el comportamiento de los widgets, como son Getx BLOC, setState, Provider, que son los más conocidos e implementados. (Docusaurus, 2023)

Gestión de estados y programación reactiva.

A diferencia de otros gestores de estados, GetX no utiliza Streams o ChangeNotifier, esto debido a que con la misma sintaxis se puede crear aplicaciones de servidores, para reducir el rendimiento y la latencia, existe GetValue y GetStream, estos ofrecen un mejor rendimiento a un bajo costo operativo.

Complejidad: Algunos administradores de estados son muy complejos, con texto y código que a menudo resulta repetitivo, en GetX no se tiene que definir una clase para cada evento, a lo que se puede reducir el código y mantener mucho más limpio y comprensible el código y su lógica.

Con GetX se respetan los widgets anidados, si un widget se mantiene vigilado por un Obx, y este a su vez contiene otro widget con otro Obx, solo se actualizará al Obx que envíe el cambio.

También otra ventaja es que, si por ejemplo se envía un valor que ya está en el widget este no vuelve a actualizar, debido a que ya se sabe que esta con este valor y no es necesario cambiar.

Administrador de estado reactivo.

En muchas ocasiones se puede decir que la programación reactiva es compleja haciendo que se deje de implementar en la programación, en GetX, se vuelve mucho más sencilla de implementar. (Docusaurus, 2023)

Ventajas

- No se necesita crear StreamControllers
- No se necesita crear un StreamBuilder para cada variable.
- No se necesita crear una clase para cada estado.

- No se necesita crear un get para cada valor inicial.

4.9. ARQUITECTURA DE MICROSERVICIOS

“La arquitectura de software define guías generales que indican la estructura, funcionamiento e interacción, entre las partes del software.”

Una arquitectura de microservicios se compone de módulos independientes, donde cada uno se encarga de realizar una tarea en específica, y este a su vez está conectada por medio de una red, esta arquitectura es una evolución de la arquitectura SOA (Arquitectura Orientada a Servicios). Estas aplicaciones están pensadas para solucionar problemas de integración en diferentes tipos de plataformas como móviles, y web, por medio de interfaces, llamada API (Application Programming Interface), a diferencia de los microservicios que son implementados con enfoques comerciales. (Red Hat, 2022)

Una de las diferencias más grande es la manera en cómo se despliegan, pues por años se han ido empaquetando de una forma monolítica, mediante un equipo de desarrollo construye la aplicación de gran tamaño que cumple con todas las necesidades comerciales que se hayan requerido.

Por el otro lado el estilo arquitectónico de microservicios se puede ir desarrollando con el tiempo y desplegando de forma independiente en aplicaciones más pequeñas, donde cada una cumple una función del sistema.

Simplistically, microservices architecture is about breaking down large silo applications into more manageable, fully decoupled pieces

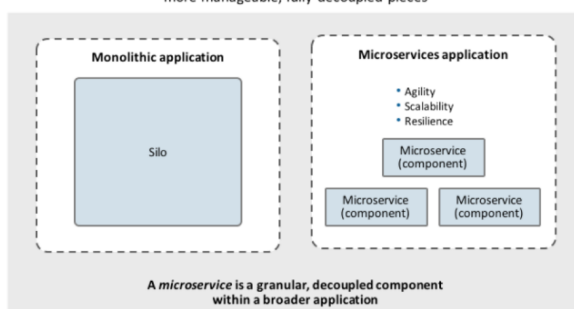


Figura 1. Comparación de aplicación monolítica ante una de microservicios

Para el desarrollo de aplicaciones de una arquitectura de microservicios existen 5 reglas básicas para el mejor control de la implementación de la aplicación.

- Dividir un sistema de arquitectura monolítico a muchos servicios pequeños.

En un servicio que su acceso es mediante la red, es una unidad pequeña que donde cada microservicio cuenta con su proceso independiente de los demás servicios, se le conoce como “servicio por contenedor”, en general estos servicios son llevados mediante contenedores, un contenedor Docker es un mecanismo de despliegue ligero. (James Davis)

- Optimizar servicios para una sola función.

En una aplicación de arquitectura monolítica tradicional en cada función toma su tiempo para ejecutar las tareas, en una de microservicios esto es completamente diferente, ya que hay una función comercial por servicio, de esta manera el sistema se hace mucho más fácil de escribir y dar mantenimiento al código. A esto se le conoce como “Principio de Responsabilidad Única” (SRP)._(Microsoft, s.f.)

- Comunicación mediante REST API y message brokers.

Una de las desventajas ante la arquitectura SOA es su simplicidad pues, no cuenta con la gran cantidad de estándares y opciones para implementar, esta manera los microservicios tienen a evitar el fuerte acoplamiento, ya que toda la comunicación debe ser mediante el servicio de API o algún mecanismo de comunicación como es el patrón de Claim Check. (Microsoft, s.f.)

4.9.4. Aplicar CI/CD por servicio

Dentro de una aplicación de gran tamaño compuesta por muchos servicios diferentes, avanzan de forma diferente, en donde cada servicio tiene una interconexión continua y única de integración/entrega que permite que proceda a un paso natural, en un enfoque

monolítico no es posible, donde el sistema es libreado a la fuerza de la velocidad del componente mucho más lento. (Dinsing, s.f.)

4.9.5. Alta disponibilidad.

En sistemas de gran tamaño, contar con un alto grado de disponibilidad es importante, una de las mejores opciones es el uso de clusters, para tener distribuido el sistema. En lugar de escalar todos los servicios de manera uniforme, se reconoce que un enfoque monolítico puede causar sobrecarga en algunos servidores y desperdicio en otros. Esto puede llevar a que algunos servicios acaparen todos los recursos compartidos, dejando a otros sin suficientes recursos. En sistemas grandes, no es necesario escalar todos los servicios por igual; algunos pueden funcionar en pocos servidores para ahorrar recursos, mientras que otros necesitan escalar significativamente en número.

De manera mucho más resumida se puede decir que una arquitectura de microservicios se compone de subservicios, donde cada uno se ejecuta en su propio proceso y esta destinado a realizar cierta tarea y realizando una comunicación ligera mediante APIs, por el protocolo HTTP. Estos servicios son creados con fines comerciales y desplegados de forma independiente, de forma automatizada, esto hace que los servicios no sean centralizados o cuenten con una cantidad mínima, además estos servicios pueden ser desarrollados en diferentes plataforma y lenguajes de desarrollo, para su almacenamiento se puede utilizar diferentes tecnologías de almacenamiento.

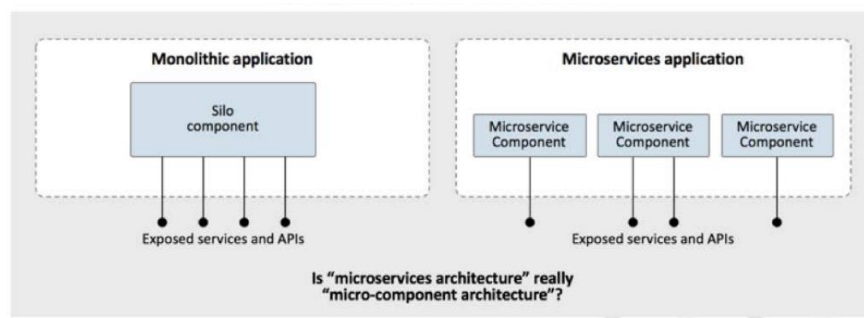


Figura 2. Diferencia entre una API y servicio monolítico.

Patrones de desarrollo de microservicios.

- Patrón de fachada.

Es una API de un negocio destinada a brindar un servicio en específico, que se encuentra fuera del sistema.

- Patrón de entidad y agregados.

Estos son útiles para aquellos equipos que no están acostumbrados a conceptos de diseños comerciales.

- Patrón de servicios.

Brindan correlación en las operaciones para las diversas entidades, mediante servicios.

- Patrón de microservicios adaptadores.

- Patrón de aplicación Strangler.

4.10. DJANGO

“Django es un framework de alto nivel de Python” creado por desarrolladores con experiencia, donde se encargan de una gran parte de las molestias que se dan al desarrollar una aplicación web, para que los desarrolladores se puedan concentrar en el código que se necesite sin tener que reinventar lo que ya existe. Es gratuito y de código abierto. (Django, 2023)

4.10.1. DJANGO REST-FRAMEWORK

Django REST Framework tiene una combinación para construir y mantener aplicaciones web muy robustas,

4.10.2. HTTP

Hypertext Transfer Protocol, es un protocolo de comunicación mediante el modelo cliente servidor, donde el cliente solicita cierta información y el servidor responde a la petición,

mediante internet por los diferentes métodos de solicitud disponibles, los métodos que existen son GET, POST, PUT, PATCH Y DELETE.

4.10.3. BASES DE DATOS SOPORTADAS

Método HTTP	ACCION
GET	Leer
POST	Crear
PUT	Actualizar
PATCH	Actualizar (Parcial)
DELETE	Eliminar

Este protocolo proporciona una respuesta a una solicitud para las comunicaciones entre 2 aplicaciones, donde la aplicación que solicita se le conoce como cliente y a la que devuelve se le conoce como servidor. Si la solicitud se procede exitosamente, el cliente recibe un código de respuesta y los datos en un formato JSON o HTML, esto dependerá del servidor.

HTTP, tiene muchos códigos de respuesta, pero existen cuatro resultados básicos para una solicitud.

- 200: Funciono.
- 300: Fue redirigido.
- 400: Error en la solicitud.
- 500: Error con el servidor.

A este protocolo se puede decir que tiene una ventaja en la interacción de las solicitudes, ya que cada respuesta es independiente de la anterior, por lo que esto hace que sea muy resistente por que el paquete no depende de los paquetes anteriores. (James Davis)

Bases de datos soportadas

Postgres

MS SQL Server

BigQuery

MySQL

4.11. PROCESOS DE PRUEBAS

El proceso de prueba es muy importante, ya que en esta fase se puede detectar los bugs y errores que se generen en una aplicación, además de tener la facilidad de detectar funcionalidades que no se hayan implementado y funciones que estén demás. (Álvaro, 2017)

4.11.1. Modelo en V.

El Modelo V, es un modelo que surge en los años 90 con su primera versión, y al pasar de los tiempos ha ido empleando nuevos procesos para el desarrollo uno en el desarrollo de software, conocido como modelo de 4 niveles, también define los procesos de gestión de la calidad, que van descritos como interactuar estas fases individuales entre cada una de ellas, de aquí su nombre por su forma de la letra V.

Fases

El modelo en V, su principal función es trazar el avance de un proyecto en cada fase individuales de forma detallada, para tener conocimiento de su avance y desarrollo correcto, y tener un buen control de este.

A las fases se las puede dividir en: (Álvaro, 2017)

- **Fase de especificaciones:** Al inicio del proyecto se prevé un análisis cada una de las especificaciones que requiere el sistema, a esta fase se la puede llamar como la fase de especificaciones. (Álvaro, 2017)
- **Fase funcional:** Para completar el proyecto es importante revisar los requerimientos, tanto los funcionales como los no funcionales, de esta forma tener una prioridad en los requerimientos. (Álvaro, 2017)
- **Fase de diseño:** Luego de las fases anteriores, viene la fase de diseño, donde se van a llevar la planificación de los componentes y las interfaces. (Álvaro, 2017)
- **Codificación:** Finalmente con las fases anteriores podemos llegar a la fase de desarrollo o codificación que sería en software. (Álvaro, 2017)

Seguido de las fases mencionadas es importante seguir con la fase de control de calidad, para su control y verificación donde siempre están relacionadas con cada una de las fases de desarrollo, donde dentro del método están las siguientes tareas. (Álvaro, 2017)

- Pruebas de unidad.
- Pruebas de integración.
- Integración del sistema.
- Validación.

4.11.2. Integración entre desarrollo y verificación

El modelo V, hace mucha referencia a la forma de cómo lleva las fases, donde la fase de desarrollo va igual con la fase de control de calidad, en la parte izquierda están las tareas de diseño y desarrollo del sistema, en la parte derecha esta las medidas del control de calidad. (Álvaro, 2017)

4.11.3. Pruebas unitarias

Una medida muy útil para el control del software es el uso de pruebas unitarias, aquí es donde se verifican los módulos de forma detallada cada uno de los módulos de forma individual, para evitar los errores que se puedan dar, pero también cumpla con las funciones y operaciones que deben de realizar, por tanto, es muy importante ir desarrollando las pruebas unitarias en paralelo con el desarrollo. (Álvaro, 2017)

4.11.4. Pruebas de integración.

Es aquí donde se realizan las pruebas y verificaciones que cada uno de los componentes del sistema interactúe con el resto, sería un gran problema en tener componentes demás o que se requieran, pero no existan, pues se puede perder tiempo y retrasar el proyecto. Ya en este punto un resultado que sea incorrecto y que no cumpla lo planificado indicaría un problema serio de interfaz. (Álvaro, 2017)

4.11.5. Prueba del sistema.

En esta etapa se lleva la verificación del sistema que haya cumplido con los requisitos generales del sistema definidos al diseñar la arquitectura del sistema, ya en esta etapa se simula un entorno de condiciones muy similares con el entorno del usuario a utilizar el sistema, para tener una mayor precisión posible, en cuanto al correcto funcionamiento. (Álvaro, 2017)

4.11.6. Validación del producto.

Ya en esta etapa final, el encargado de dar su validación es el cliente o usuario final, pues es quien comprueba si se cumplen las especificaciones en el funcionamiento del sistema, también vale indicar que el sistema es comprobado de una forma superficial, por lo que es muy importante realizar el seguimiento y asesoría al cliente o usuario del sistema. (Álvaro, 2017)

4.11.7. Modelo V

En el 2006, para reflejar nuevos principios como el desarrollo ágil, surgió el modelo V XT, que hace referencia a Extreme Tailoring donde describe la nueva posibilidad de

adaptar el modelo al requisito de cada proyecto. (Gutierrez, Escalona, Mejías, & Reina, 2006)

4.11.8. Aplicación del modelo

El modelo V XT, es más utilizado en el ámbito de la industria ya que se puede encontrar de manera pública en la gran mayoría de ofertas de proyectos de software y en otros casos puede ser obligatorio su uso, en especial para las empresas que se dedican a desarrollar software para empresas públicas, este modelo puede ser empleado en proyectos de cualquier tamaño, para proyectos de empresas públicas y empresas militares. Pero pueden existir ciertas variaciones donde reflejen los pasos del proceso típico de la disciplina. (IONOS, 2020)

- **Ventas.**

- Optimización de la comunicación entre las partes involucradas, por medio de términos de responsabilidades claramente definidos. (IONOS, 2020)
- Minimización de riesgos y mejor planificación a través de roles, estructuras y resultados fijos y predeterminados. (IONOS, 2020)
- Mejora de la calidad del producto gracias a medidas de control de la calidad firmemente integradas. (IONOS, 2020)
- Ahorro de costes gracias al procesamiento transparente a lo largo de todo el ciclo de vida del producto (IONOS, 2020)

- **Desventajas.**

- El modelo puede ser demasiado sencillo para mapear el proceso completo del desarrollo desde el punto de vista del desarrollador, puesto que esta centralizado en gestión de proyectos cercanos, además que la estructura puede parecer regidos que no permiten una respuesta flexible a los cambios durante el desarrollo, por tanto, lo que lleva un curso lineal del proyecto. (IONOS, 2020)

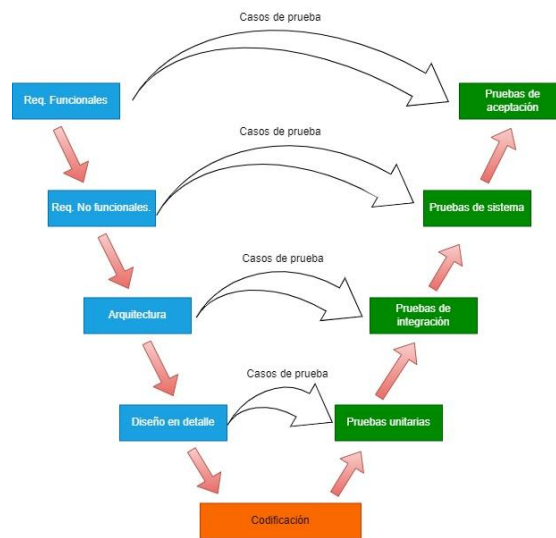


Figura 3. Modelo en V de proceso de prueba de una aplicación

- Requerimientos Funcionales.

Estos requerimientos son las funcionalidades que deben ser cumplidos de forma obligatoria, pues el no cumplir, no se habrá realizado con el objetivo.

- Requerimientos No Funcionales.

A estos requerimientos se los podemos decir que son requerimientos extras que muchas de las veces no están dentro de los requerimientos solicitados por el cliente, pero en ocasiones son muy importantes, pues muchos de ellos nos ayudan a mejorar el sistema en su rendimiento.

4.12. Algoritmos inteligentes

Primeramente, a algo inteligente se lo conoce aquello que es capaz de realizar ciertas actividades por sí mismo, además de tomar decisiones por sí solo, entonces para llegar a tener una inteligencia artificial se compone de un gran grupo de tecnologías donde engloba el Machine Learning, en donde un algoritmo lo que hace es iteraciones a la que se le conoce como aprendizaje por refuerzo, donde realiza una constatación prueba y error, hoy en día con las máquinas se es capaz de realizar estas iteraciones en tiempo récord, pues pese a que en décadas anteriores ya se venían desarrollando, no se contaba con la tecnología necesaria para poner en práctica. (Fernández & Fernández, 2019)

Uno de los retos que se plantea es los algoritmos nos permiten potentes máquinas, pero la falta de neutralidad y transparencia plantea nuevos retos. (Álvaro, 2017)

Con los masivos datos que hoy en día se encuentra en la red de internet, se ha despegado un despertar de la inteligencia artificial, en lo que ahora los algoritmos ya son capaces de aprender de nosotros, a base de nuestros comportamientos e imágenes, la inteligencia artificial es capaz de predecir nuestra conducta. (Elternativa)

Ya en este escenario surgen nuevos retos donde se ve intervenida por la ética y los retos sociales que esto conlleva a la convivencia de la sociedad, ya que, al aprender de nuestros datos, la inteligencia artificial es capaz de reproducir nuestras acciones. (Elternativa)

4.12.1. Historia

A la inteligencia artificial se ha puesto cada vez más popular, en ámbitos como el comercial, con los sistemas de recomendación, hablar sobre la inteligencia artificial, ya sea como ciencia ficción o tecnología. Pero la inteligencia artificial no solo es un algoritmo, pues este solo es una parte de la inteligencia artificial, con una figura muy relevante en el desarrollo de las que se conoce hoy en día, Alan Turing, pues con su máquina es quien se pone como el padre no solo de la inteligencia artificial, sino también con la informática moderna. (Elternativa)

Pese a muchos de los casos de innovación y nuevos inventos que han revolucionado no ha sido hasta 1956, donde se empleó el termino inteligencia artificial, cuando “John McCarthy, Marvin Minsky y Claude Shannon, durante la conferencia de Dartmouth Summer Research Project on Artificial Intelligence” (Proyecto de investigación de verano de Dartmouth sobre inteligencia artificial). En tal conferencia, se partía de la premisa de que “la conjetura de que cada aspecto del aprendizaje o cualquier otra característica de la inteligencia puede, en principio, ser descrito con tanta precisión que puede fabricarse una máquina para simularlo. (Abeliuk & Gutiérrez, 2021)

4.12.2. IA y su impacto en la cultura.

Pese a los avances en los años 30 y 50, donde su investigación ya planteo bases para su desarrollo, la falta de los recursos en tecnología en esa época se disminuye su popularidad no siendo hasta los 80 y 90, donde vuelve nuevamente a surgir, pero ya en los 2000, es donde de verdad empieza a tomar fuerza.

En 1987, Martin Fischles y Oscar Firschein describieron los 12 atributos con los que debe de contar un agente inteligente. (Elternativa)

- “Tener actitudes mentales”.
- “Tener capacidad de obtener conocimiento”.
- “Poder resolver problemas, incluso descomponiendo problemas complejos a otros más simples”.
- “Capaz de realizar operaciones más complejas”.
- “Poseer la capacidad de dar sentido”.
- Planificar, predecir consecuencias, evaluar alternativas.
- “Conocer los límites de sus habilidades y conocimientos”.
- Poder ser original, creando incluso nuevos conceptos o ideas, y hasta utilizando analogías.
- “Poder generalizar”.
- “Poder percibir y modelar el mundo exterior”.
- “Poder entender y utilizar el lenguaje y sus símbolos”.

Se puede llamar la edad de oro de la inteligencia artificial con la llegada del nuevo siglo, en donde conceptos como los chatbots, sistemas de recomendación, clasificadores, etc., ya empiezan a sonar como una realidad donde la principal característica es que pueden mantenerse disponibles las 24 horas, donde los puestos como soporte, están disponibles

sin necesidad de que haya una persona al otro lado de la comunicación. (Abeliuk & Gutiérrez, 2021)

No obstante, a todo lo mencionado la mejor constatación que se puede mencionar es la máquina de IBM, su supercomputador Watson, siendo en el 2011 la gran noticia, al haber podido ganar en el concurso “Jeopardy”, donde ya fue capaz de interactuar con los demás. (Abeliuk & Gutiérrez, 2021)

Una de las principales características que le hacen diferente a su antecesor Deep Blue, es su capacidad de aprender e improvisar, en base a lo que interactúa con el lenguaje humano. Una de las claves del desarrollo son la teorización del Deep Learning, con lo que se da un acercamiento masivo a la población, pues con aplicaciones como SIRI de Apple, empezó la experiencia del aprendizaje automático, y al tener acceso a la información de la población global, como que el término de Chatbot, ya no es muy sonado, pues este ha evolucionado a Asistentes Virtuales, quien no solo puede mantener una conversación, sino que también puede interactuar con actividades como poner música o dar una respuesta a una consulta, además en el sector automotriz ya se están desarrollando asistentes para los vehículos autónomos, quienes prometen disminuir los accidentes de tránsito provocados por los humanos. (Abeliuk & Gutiérrez, 2021)

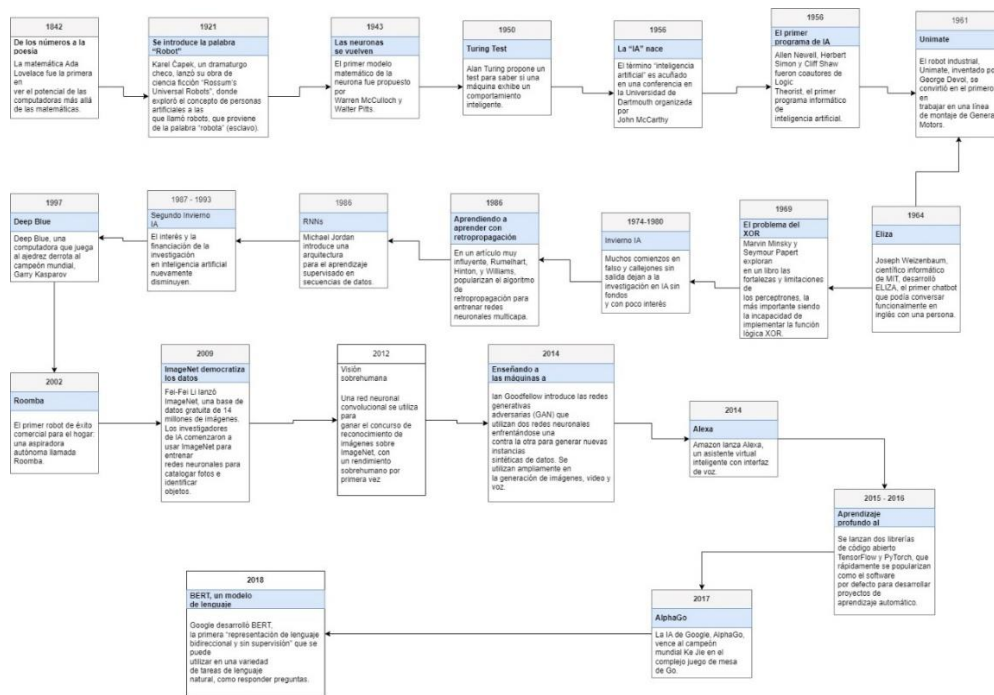


Figura 4. Historia de la inteligencia artificial. (Abeliuk & Gutiérrez, 2021)

Tipos de algoritmos

- Algoritmo de aprendizaje por refuerzo.

Este tipo de algoritmo está basado en una constante de iteración que consiste en prueba y error, pues lo que hace es realizar una serie de pasos, para luego calcular su error, y de esta manera volver a probar y mejorar el error generado. (Fernández & Fernández, 2019)

- Algoritmo de aprendizaje supervisado.

Este tipo de algoritmo de aprendizaje supervisado se basa en modelos de predicción, por lo que ya cuenta definidas ciertas salidas, por lo que debe de realizar un entrenamiento hasta lograr el resultado que se espera, entre los principales algoritmos supervisados encontramos los árboles de decisión, clasificación y regresión ordinaria por mínimos cuadrados y regresión logística. (Fernández & Fernández, 2019)

- Aprendizaje no supervisado.

Este tipo de aprendizaje es similar al de aprendizaje supervisado, pero la diferencia, su modelo se ajustan en función de los datos de entrada, este algoritmo lo que hace es realizar un entrenamiento propio sin necesidad de que sea intervenido por alguien externo, los algoritmos de este tipo son; algoritmos de agrupamiento (clustering), análisis de componentes principales (PCA), Singular Value Decomposition (SVD), análisis de componentes independientes (ICA). (Fernández & Fernández, 2019)

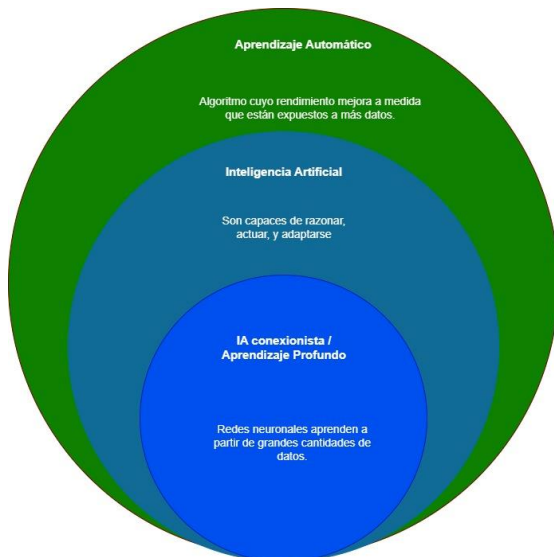


Figura 5. Diagrama de Venn que muestra la relación entre distintas subáreas de la inteligencia artificial. (Abeliuk & Gutiérrez, 2021)

4.12.3. K-Nearest Neighbors Algorithm (KNN)

El algoritmo, de vecinos más cercanos, llamado KNN, es un “algoritmo supervisado sin parámetros”, que utiliza la distancia más cercana entre puntos, para buscar similitudes para realizar predicciones grupales de un dato.

Si bien para asignar un punto a los vecinos más cercanos se dice que debe de constar con un 50% de voto mayoritario, esto en cuanto se consta con 2 clases, pero al tener una cantidad de 4, solo se requerirá el 25%, para tener una conclusión.

En la regresión se utiliza un concepto similar al de clasificación, con la diferencia en este que se toma en cuenta el promedio de los vecinos más cercanos, para realizar una predicción dentro de una clasificación, la diferencia más clara es que en la regresión se utilizan valores continuos, a diferencia de la clasificación que se utilizan valores discretos.

Pero antes de realizar cualquier clasificación es necesario realizar la definición de la distancia, la euclidiana es la más utilizada, es importante mencionar que el algoritmo de KNN, es parte de una familia que se le puede decir que es de “aprendizaje perezoso” pues, a diferencia de otros que pasan por una etapa de entrenamiento, esta solo realiza cálculos de los puntos cercanos sin entrenamiento previo. (IBM, s.f.)

Esto quiere decir que el cálculo que se realiza cuando se realiza la predicción, esto hace que dependa principalmente de la memoria para almacenar sus datos, por lo que se le denomina método de aprendizaje basado en instancias o memoria.

Este tipo de algoritmo no es recomendable si los recursos computacionales con los que se cuentan son bajos esto debido a lo ya antes mencionado, que requiere aun algo grado de memoria y procesamiento.

5. MARCO METODOLÓGICO

En esta parte se presenta la metodología que se utiliza para desarrollar la aplicación móvil y web, además de presentar la propuesta para la solución, las actividades a realizar y los requerimientos desde donde se toma como partida para el desarrollo de la aplicación.

5.1. METODOLOGÍA DE ANÁLISIS DE DATOS

De acuerdo con los requerimientos y los objetivos del proyecto, en base a los diferentes tipos de metodologías ágiles, se ve a la metodología SCRUM, para iniciar con la metodología iniciamos con la organización de los Sprints en base a los requerimientos establecidos. Luego en base a ello podemos trazar un cronograma de trabajo detallando las actividades a realizarse.

Propuesta de solución

De una forma general se presenta la solución en la siguiente imagen, de una manera general como es su funcionamiento, al ser realizado.

En la imagen se puede observar a cada una de las capas que componen la arquitectura de la aplicación desarrollada, para la parte lógica “capa de negocios”, la capa de presentación y para manejar los datos “la capa de datos”.

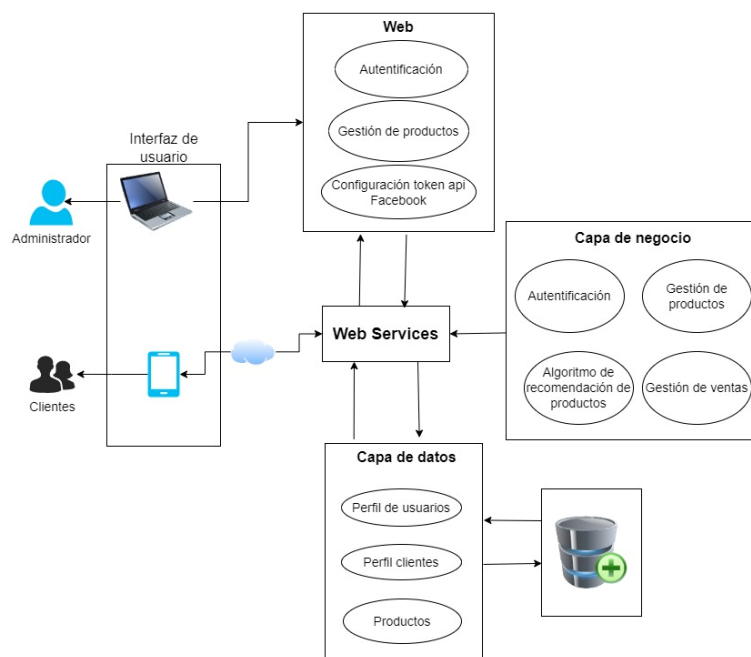


Figura 6 “Arquitectura general de la aplicación”

Para la “lógica del negocio”, “se considera los datos que se requieren para la recomendación de los productos”, por lo que se tomarán los comentarios realizados en las publicaciones de las imágenes en Facebook, para alimentar el algoritmo de KNN, para obtener estos datos se utilizará la API de GraphQL de Facebook, donde luego con la librería de Sklearn, se pone a correr el algoritmo donde recibe los productos con el id de la publicación, luego realiza una comparación de comentarios, y devuelve los productos con los comentarios similares a los realizados en la publicación del producto.

Finalmente, en la capa de presentación que vendría a ser la aplicación móvil, es donde se presentara el producto y los productos recomendados, donde los usuarios son los que intervienen para correr el algoritmo.

5.2. METODOLOGIA SCRUM

Para el desarrollo de tecnología de software con un enfoque ágil, empleamos la metodología SCRUM, iniciando con la organización de “Sprints en base a los requerimientos establecidos, con un backlog, esto teniendo muy en cuenta las condiciones del caso impuestas”, para poder cumplir con los requerimientos en los en los plazos establecidos desarrollamos un cronograma, donde se detalla, con fechas establecidas de inicio y fin de cada sprint, donde cada sprint debe ser desarrollado dentro de este rango de fecha, “donde se analice y luego avance para determinar el próximo sprint”. En caso de que el producto requiera más cambios o mejoras. O también se puedan requerir agregar nuevas funcionalidades, por lo que se conduce a nuevos requerimientos.

Roles de asignación.

“De acuerdo con SCRUM, para avanzar con el progreso del desarrollo del proyecto se deben asignar a los integrantes del proyecto y su rol”.

Product Owner. Ing. Remigio Hurtado, “se delega de describir todas las funciones con la que debe contar el proyecto”.

Scrum Mater. Ángel Mesías Jadan Corte, responsable y encargado del desarrollo del proyecto.

Development Team: El estudiante Ángel Mesías Jadan Corte, encargado de codificar, probar e implementar la aplicación en el proyecto.

Roles Scrum

Product Owner: “Ing. Remigio Hurtado (RH)”

Scrum Master: Ángel Mesías Jadan Corte (AJ)

Development Team: Ángel Mesías Jadan Corte (AJ)

Descripción de Actividades.

- OE1. “Revisar el estado del arte” sobre PWA y recomendación de productos basados en el perfil y contenido.

No.	Actividad
1	Investigar y estudiar sobre el desarrollo de aplicaciones pwa.
2	Investigar y estudiar sobre los diferentes procesos de contenido de datos de perfiles de las diferentes redes sociales.
3	Definir los contenidos útiles del perfil.

- OE2. Definir una arquitectura basada en microservicios para el consumo y publicación de productos.

No.	Actividad
1	Investigación sobre arquitecturas de microservicios.
2	Diseño de arquitectura de microservicios.
3	Desarrollo de microservicios sobre la arquitectura diseñada.

- OE3. Diseñar e implementar un algoritmo inteligente que permita recomendar productos en base a los perfiles o contenidos de redes sociales.

No.	Actividad
1	Investigar sobre los diferentes tipos de algoritmos y sus aplicaciones.
2	Evaluar los diferentes algoritmos en base a las necesidades que requeridas y sus características.
3	Implementar el o los algoritmos definidos para “la recomendación de los productos en base a los datos de contenidos de perfiles de las redes sociales”.

- OE4. “Validar y probar la aplicación con métricas de rendimiento y confiabilidad”, además del algoritmo inteligente con medidas de precisión y la matriz de confusión.

No.	Actividad
1	Definir las métricas de evaluación, de rendimiento y confiabilidad.
2	Definir las herramientas de validación de laboratorio.
3	Validación y experimentación de evaluación, rendimiento y confiabilidad.
4	Desarrollar informe de resultados en base a la experimentación de laboratorio.

6. RESULTADOS

“En esta sección se va a presentar todos los resultados que se ha obtenido en el desarrollo del proyecto de la aplicación”, tanto en la parte del algoritmo implementado el Backend y el Frontend, por lo que para iniciar se toma los requerimientos, para en la fase desarrollo donde cada sprint, se hace una muestra detallada de todo el proyecto, donde se presentan graficas evidenciales, arquitecturas y “el proceso del despliegue de la app en la plataforma”.

Especificación de requerimientos.

“En esta sección se especifica los requerimientos funcionales y no funcionales del proyecto. En general este capítulo está organizado en torno a ISO/IEC/IEEE”

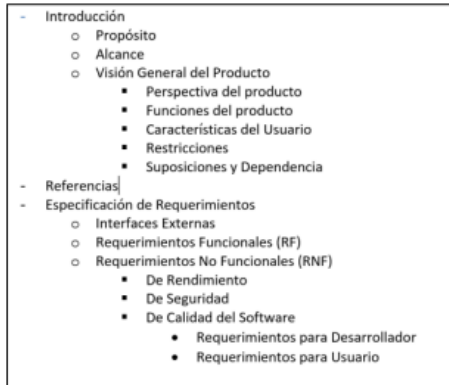


Figura 7. “Estructura específica de requerimientos”.

6.1. Introducción

“Para esta sección se ha tomado en cuenta las directrices y dadas por el estándar 29148 de la IEEE. Donde se definen los requerimientos funcionales y no funcionales para el desarrollo del proyecto”.

6.2. Propósito

“El propósito del sistema es brindar una facilidad de publicar los sus productos de la aplicación e-commerce”, donde puede publicar en la red social Facebook y en base a los comentarios, pueda realizar una recomendación a los diferentes clientes.

6.3. Alcance

“El alcance del sistema es permitir a los clientes que puedan realizar una compra de los diferentes productos más comentados”, por sus consultas de costos, colores y disponibilidad, y estos para adquirir puedan ser cancelados mediante pagos electrónicos como es el botón de pago. Donde el administrador pueda constatar el pago para el posterior envío.

6.4. Visión general del producto.

Perspectiva del producto

“Esta aplicación puede ser accedida por el administrador cuenta con un panel para registrar los productos”, publicar en la red social, y a su vez los clientes puedan acceder a este servicio ya sea por navegadores web o la aplicación móvil, donde cada vez que revise un producto podrá visualizar el producto con los diferentes productos recomendados.

Esto se va a realizar por medio de la aplicación REST, que va a ser la encargada de realizar la lógica de negocios, “con la conexión a la base de datos y la API de Facebook, para obtener los datos en tiempo real”.

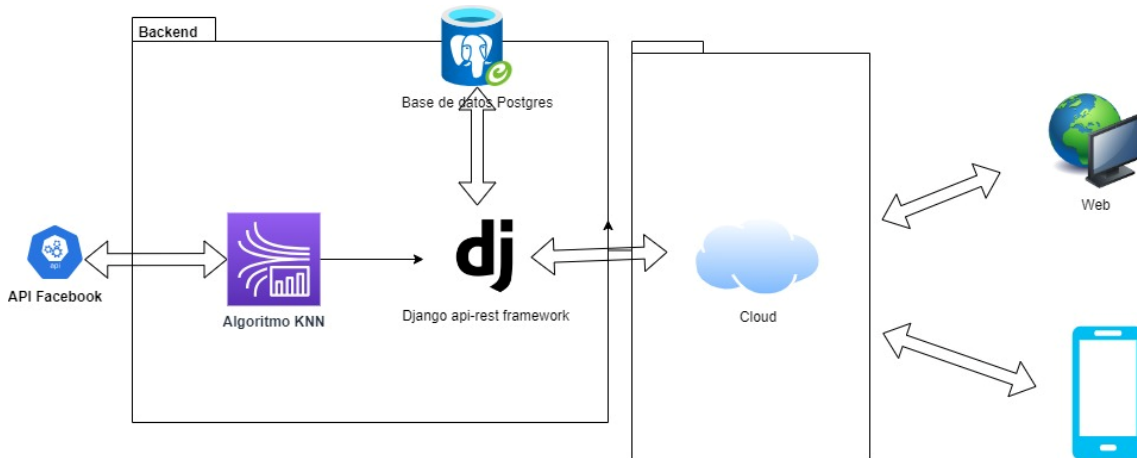


Figura 8. Arquitectura general de la aplicación.

Funciones del Producto

Función de registro de productos: Esta función es para el administrador encargado de cargar los productos, con su imagen, precio y descripción.

Función de publicación de producto en la red social Facebook: Esta función es donde el administrador pueda publicar la imagen con la descripción del producto a la red social Facebook, y pueda ser vista y comentada por los clientes o seguidores de la página.

Función de recomendación: Para acceder esta función el cliente debe acceder a la aplicación mediante el dispositivo móvil o web, para revisar los diferentes productos disponibles y con los productos recomendados de cada producto, donde el cliente puede realizar su compra y pagar con el botón de pagos.

6.5. Referencias

“Todos los requerimientos están basados en los estándares ISO/IEC/IEEE 29148”

6.6. Especificación de requerimientos

Tabla 1. RF – “Modulo para compra de productos”.

Identificador:	Nombre:	Ingreso a aplicación	Funcional
RF01	móvil/web		
			Categoría (Visible/No visible)

Descripción: Incorporar un módulo de ventas en donde el cliente pueda acceder para realizar compras y pagos con un botón de pagos, también el cliente pueda acceder al historial de sus compras, realizado el módulo, se realice el respectivo despliegue.	Visible
Objetivo: Recomendar al cliente los productos que más comentarios han recibido.	
Datos de entrada:	Datos de salida:
<ul style="list-style-type: none"> • “Login de inicio de sesión en aplicación”. 	<ul style="list-style-type: none"> • Login correcto como cliente.
<ul style="list-style-type: none"> • Vista de todos los productos disponibles. 	<ul style="list-style-type: none"> • Historial de compras realizadas. • Opción de comprar y pagar.
Criterios de aceptación: El cliente pueda revisar los productos y sus recomendaciones, pero no pueda realizar la compra si no está iniciada la sesión.	
Prioridad: Alta	

Tabla 2. RF – Módulo para recomendación de productos.

Identificador: RF02	Nombre: recomendación de productos	Proceso de	Funcional
Descripción: Incorporar al módulo la función de recomendar los productos que tengan similares comentarios, al producto que se visualice.		Categoría (Visible/No visible)	
		Visible	
Proceso:			
<ul style="list-style-type: none"> • Obtención de datos. • Limpieza de datos. • “Relación de id de publicaciones con productos de la base de datos”. • Entrenamiento. • Optimización de proceso. • Recomendación. 			
Datos de entrada:		Datos de salida:	
<ul style="list-style-type: none"> • Producto visualizando. 		<ul style="list-style-type: none"> • Lista de productos con comentarios similares. 	
Criterios de aceptación: Incorporar un módulo para la aplicación donde permita realizar una recomendación de productos en base a los comentarios			

recibidos, donde cada vez que ingrese a visualizar el producto también salgan los productos con comentarios similares.

Prioridad: Alta

6.2. Arquitectura

Como objetivo principal en la arquitectura es detallar la solución de cómo está compuesta la aplicación y su relación un componente con el otro, además de ofrecer una mejor comprensión del funcionamiento a nivel técnico.

6.2.1. Vista de desarrollo

“La propuesta de solución se detalle en 3 capas de presentación que se resume como frontend, backend y la base de datos”. “Pese a que en la capa del Backend, se encuentra la mayor cantidad de procesos y cálculos”, es importante que en cada capa también debe de contar con su propia lógica, para que puedan comunicarse una con otra, pues son necesarias y no se pueden realizar si no se cuenta con ciertas lógicas.

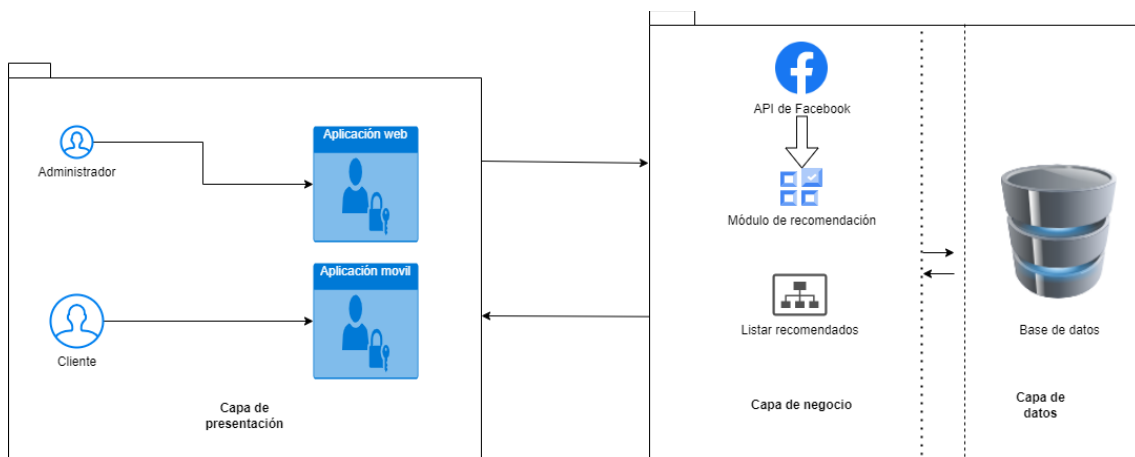


Figura 9. Arquitectura en 3 capas.

En “la capa de datos” se trabaja con “la base de datos postgres”, “para almacenar los datos de los clientes para iniciar sesión, también almacenar los diferentes productos. En la capa de negocio, se manejan los diferentes procesos como son la conexión al api de Facebook que se utiliza la librería Facebook-py, misma que nos sirve para realizar las publicaciones de los productos y obtener los comentarios de los productos, que será alimentada al algoritmo de KNN, para realizar la recomendación de los productos, además para la gestión de control de los productos se utiliza el framework de django para administrar los productos y servir los servicios con la librería django-rest-framework.

Para la capa de presentación se utiliza el panel de administración de django para crear los productos, esto para el administrador, y en cuanto para los clientes se utiliza el framework de Flutter, para realizar “el proceso de compra y visualización de los productos recomendados, de los diferentes productos”.

6.2.2. Vista física

En esta vista podemos ver “la comunicación entre los diferentes componentes que son parte de la propuesta de solución”.

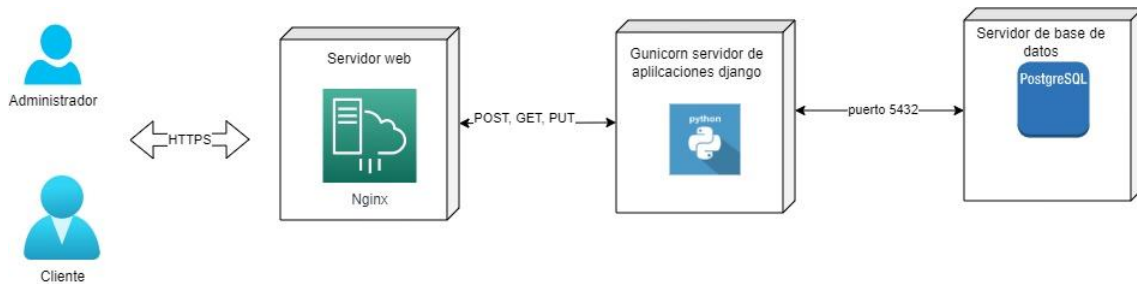


Figura 10. Diagrama de vista física de despliegue de aplicación.

6.3. Vista API

“En esta sección se puede ver la arquitectura de comunicación entre la aplicación, la base de datos y la aplicación PWA”.

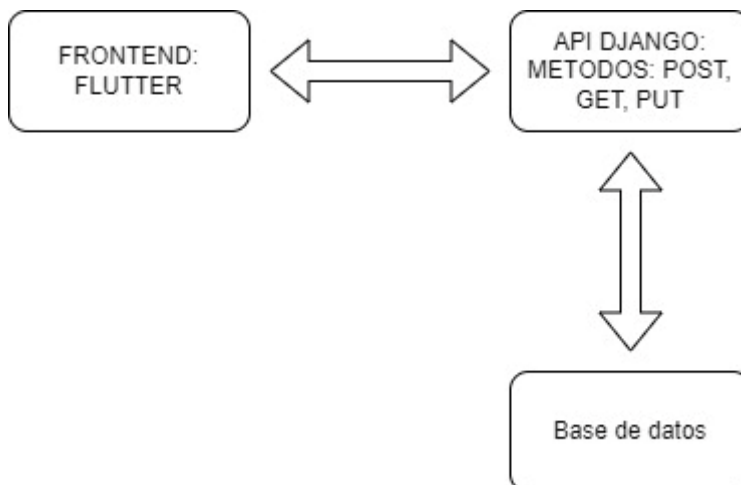


Figura 11. Arquitectura de aplicación

6.3. Fases de desarrollo

Para “el desarrollo del proyecto se utilizó el método de desarrollo ágil SCRUM, pues el objetivo es brindar productos de cierto valor en poco tiempo”, tomando al “Sprint” con los requerimientos ya mencionados donde cada uno tiene una duración de 3 a 4 semanas, esto para tener el contacto más cercano y poder ir cumpliendo con las actividades planificadas.

Tabla 3. Diseño de sprint de forma general “con sus actividades”

Sprint 1	Sprint 2
<ul style="list-style-type: none"> • “Fase de transición” • Integración funcional • Pruebas 	<ul style="list-style-type: none"> • “Corrección de la aplicación” • “Integración y corrección”. • Lanzamiento de nueva versión. • Pruebas unitarias • “Pruebas de integración”
Sprint 3	Sprint 4
<ul style="list-style-type: none"> • “Modelado de modo de recomendación” • Diseño de interfaz • Pruebas 	<ul style="list-style-type: none"> • Corrección de errores Sprint anterior • Resultado final

6.3.1. Primera fase, Sprint 1

En el primer Sprint, se detallan todas las actividades realizadas con el propósito de detallar y verificar la correcta integración, pruebas y despliegue.

Fase de transición:

Para poder sacar a producción un producto de software es muy importante realizar pruebas de forma local, para no perjudicar “la integridad de la base de datos ya en producción”, “por lo que es muy importante que las pruebas se realicen” de forma muy similar al ambiente de producción, puesto que de un buen test dependerá el rendimiento del sistema.

Tabla 4. “Actividades de la fase de transición”.

Actividad	Descripción
“Creación de la base de datos en Postgres”	Con django es muy fácil, la creación de la base de datos, solo tenemos que migrar los modelos a la base de datos que nos servirá para almacenar los datos de los productos y clientes, y sus accesos tanto por dispositivos web como móviles.

Corrección y validación de inicio de sesión	Con django se va a utilizar el manejo de tokens para el inicio de las sesiones, en los diferentes dispositivos, pues con el login por defecto solo se pueda acceder al panel de administración.
--	---

Levantar servicios REST: Para levantar un servicio web REST, es importante tener un servidor para poder desplegar la aplicación, pese a que hay la posibilidad de levantar en un servidor de forma física, hoy en día gracias a la gran cantidad de proveedores de estas infraestructuras, para ello utilizaremos AWS Lightsail, por su fácil despliegue y configuración, con servidor Ubuntu,

Requerimientos mínimos para el servidor web Backend

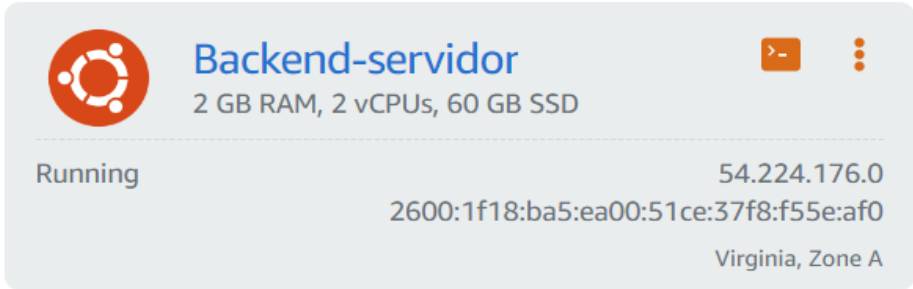


Figura 12. Especificaciones del servidor

Generar APK de aplicación: “En Flutter generar el APK es muy sencillo”, solo debemos de correr el comando flutter build apk, en la consola o terminal, en este caso de VSCode con este comando ya tendremos nuestro apk para Android, es importante revisar que no tenga errores o advertencias, pues esto puede ser un problema con la aplicación a la hora de que entre ya a producción. Una vez generado subiremos a la plataforma de Google Play, donde tomara algunos días, hasta que ya esté disponible para el público en general.

Análisis del impacto de ecommerce en el Ecuador

De acuerdo a una publicación que se ha realizado en la “CECE (Cámara de Comercio Electrónico), en América Latina, ya existen alrededor de 300 millones de personas que son compradores digitales”, de acuerdo a esta publicación se provee que para el 2025, exista un crecimiento del 20%, pese a que a comparación de “otras regiones como Europa y Estados Unidos”, el uso de medios electrónicos para sus compras, “se espera que en los siguientes años se dupliquen las compras esto por parte de los compradores minoristas”.

En el Ecuador, a comparación de otros países es un país relativamente joven ante este medio de transacción comercial, ocupa el puesto 67 entre 196 naciones, en comparación del número de habitantes. De los 18 millones de habitantes, se ha visto que el 40% de la población que son entre 25-54 años son los que más participan en el comercio electrónico. En el 2022 se ha registrado un crecimiento del +1.3% en comparación del año 2021.

Una de las condiciones para que el comercio electrónico no haya sido de gran relevancia exponencial, es por la diferencia socio económica, y la falta de conocimiento de plataformas seguras, para este tipo de transacciones, pues el temor ante estafas es alto.

También se puede decir que anterior mente por la falta de internet era difícil tener conocimiento sobre la compra de productos en internet, hoy en día, una gran parte de la población ya cuenta con este servicio, un motivo para que haya existido la aceleración del crecimiento del internet fue la pandemia que afecto el acceso a la educación presencial, por lo que fue indispensable contar con internet, pero aún se puede decir que el 37% no tiene acceso a internet a nivel mundial, por lo que para esta población, no es posible acceder a los medios electrónicos para sus compras. (Portalanza, y otros, 2022)

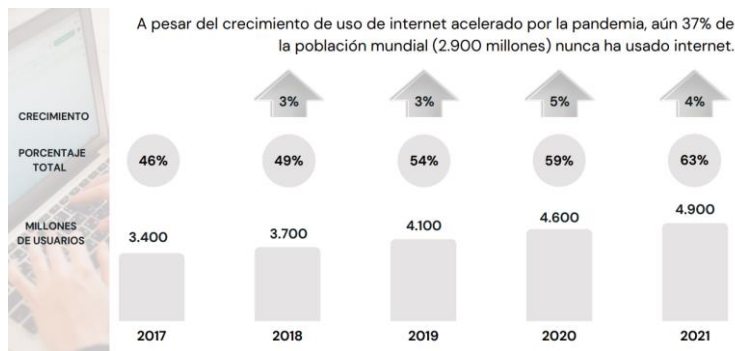


Figura 13. Consumo global del internet. (Portalanza, y otros, 2022)

Modelado: Para generar un modelo de los métodos, es importante tener en cuenta los datos que requerimos y como cada atributo de los diferentes objetos interactúen con los demás componentes.

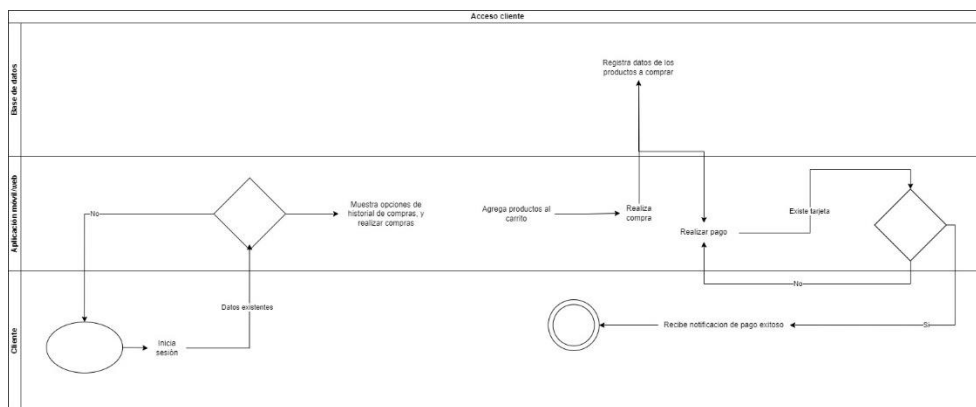


Figura 14. Diagrama de actividades del usuario cliente, para realizar compra

Pruebas unitarias: Para los servicios REST, es muy importante realizar las pruebas utilizando algún software como Postman, que es una herramienta muy útil para probar los servicios web.

6.3.2. Segunda fase, Sprint 2

Corrección de la aplicación: Dentro de este Sprint, tenemos algunos errores que se han generado, por lo que requieren “ser corregidas de manera importante, para el correcto funcionamiento de la aplicación”.

Tabla 5. Actividades realizadas para corrección de errores de la aplicación.

Descripción	Problema	Solución
Publicación de imagen en la página de Facebook	Al estar en el servidor local no es posible publicar las imágenes en la página de Facebook.	Luego de probar diferentes acciones, y leer los requerimientos de la API de Facebook, se pudo ver que el problema es el servidor, pues se debe exponer de forma publica la imagen, por lo que la url del servidor en local no es válido para la API, por lo que al probar cargando a un servidor de pruebas en línea, la publicación fue exitosa.

Diseño descriptivo de interfaz de usuario

Widget principal

Aquí es donde se ha creado las diferentes interfaces graficas para el uso de la aplicación por parte del usuario, donde al inicio de la aplicación ya podemos observar los productos creados, y la opción de iniciar sesión, o crear una cuenta.

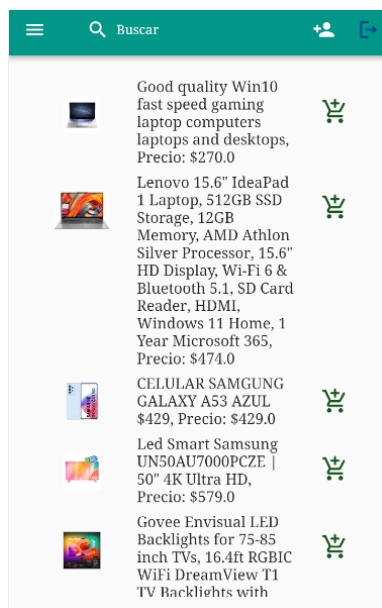


Figura 15. Ventana inicial sin inicio de sesión

Registro de usuario

En caso de no tener una cuenta registrada, la aplicación tiene la opción para registrarse, en donde debe ingresar el nombre, apellido, un correo electrónico y una contraseña, así con estos datos podrá luego ingresar.

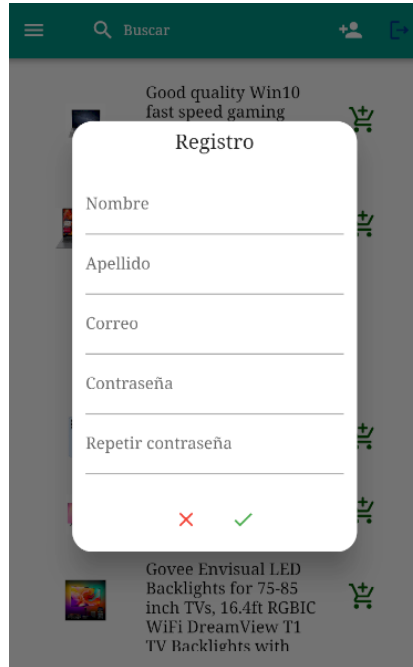
The image shows a mobile application interface with a dark green header containing a search bar labeled 'Buscar' and navigation icons. A white registration modal is centered on the screen. The modal has a title 'Registro' and five input fields: 'Nombre', 'Apellido', 'Correo', 'Contraseña', and 'Repetir contraseña'. At the bottom of the modal are two small icons, a red 'X' and a green checkmark. The background of the app is a product listing for 'Good quality Win10 fast speed gaming' and 'Govee Envisual LED Backlights for 75-85 inch TVs, 16.4ft RGBIC WiFi DreamView T1 TV Backlights with'.

Figura 16. Ventana de registro.

Inicio de sesión

Luego de estar registrado en la aplicación, al dar clic en el botón de iniciar, aparecerá la siguiente ventana, donde debe poner el correo y la contraseña, que se puso al registrarse.

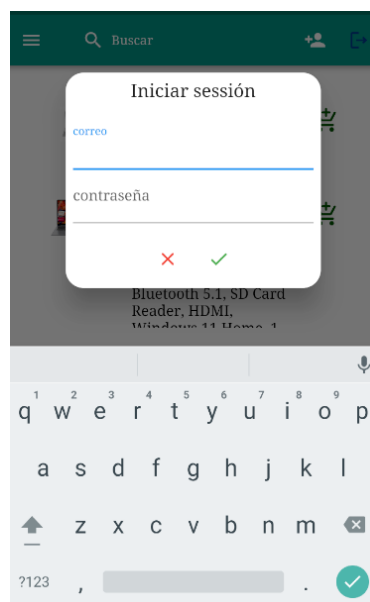
The image shows the same mobile application interface as Figure 16, but with a white login modal titled 'Iniciar sesión'. It features two input fields: 'correo' and 'contraseña'. Below the fields are the same red 'X' and green checkmark icons. A virtual keyboard is visible at the bottom of the screen, showing the letters 'q w e r t y u i o p' on the first row, 'a s d f g h j k l' on the second, and 'z x c v b n m' on the third, along with a spacebar and a green checkmark icon.

Figura 17. Formulario de inicio de sesión.

Vista principal session iniciada

Una vez iniciada sesión se va a mostrar la página con botones como ver el carrito de pedidos, cerrar sesión, y ver el historial de compras.

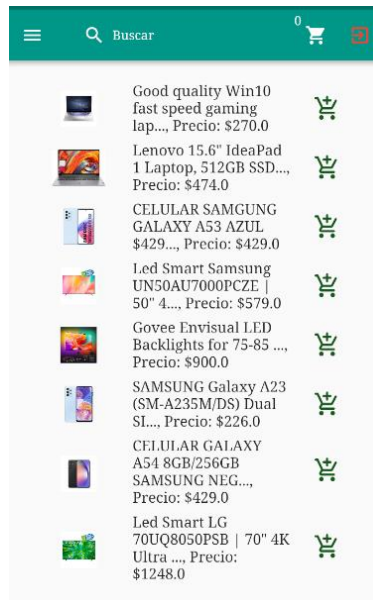


Figura 18. Ventana con sesión iniciado

Vista producto

En la siguiente imagen podemos ver un producto y con sus productos recomendados, con similitud de comentarios.



Figura 19. Vista de producto con lista de productos recomendados.

Publicación en Facebook con su comentario

En la siguiente imagen podemos ver la publicación con un comentario, el algoritmo tomara este comentario, para realizar recomendaciones en base de este comentario.

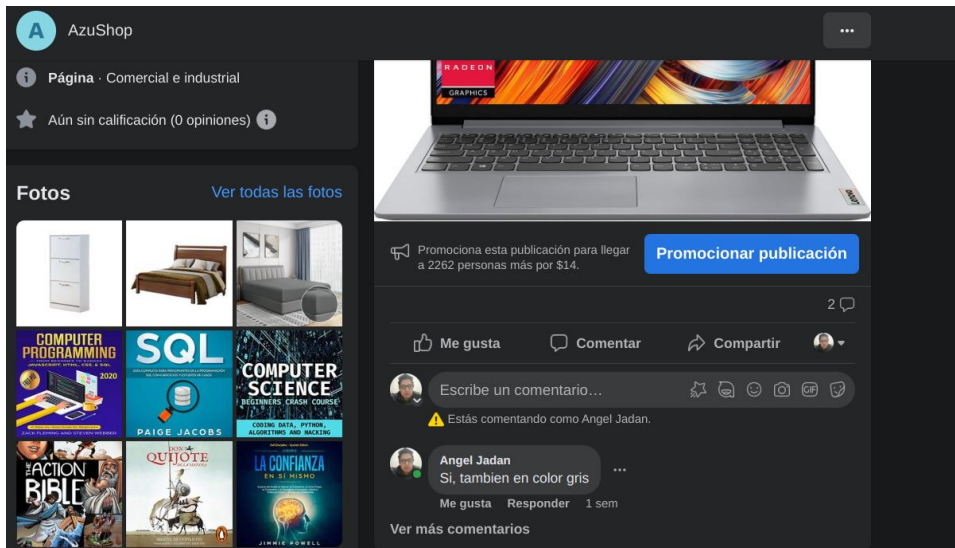


Figura 20. Comentario de seguidor de la página.

Publicación de otro producto

Para comprobar que el algoritmo funciona de forma correcta, a una de las publicaciones le realiza un comentario un seguidor de la página, este comentario hará que la recomendación cambie.

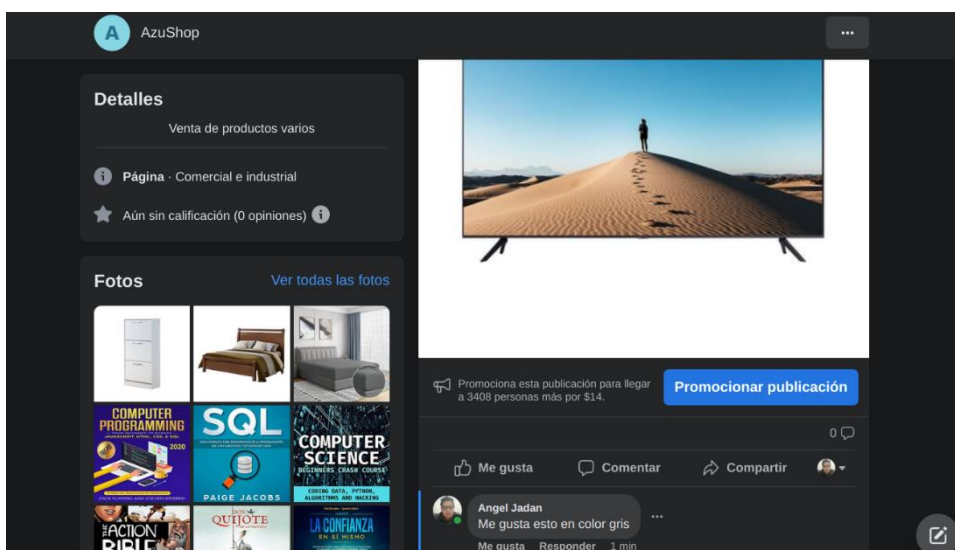


Figura 21. Comentario a publicación

Vista de producto actualizado el algoritmo

Con el comentario realizado, ahora la recomendación sería de la siguiente manera, como se puede ver en la imagen.



Figura 22. Vista de producto con sus recomendaciones

Nuevo comentario a publicación

En cuanto otra publicación reciba otro comentario más similar al comentario de la publicación original, se mostrará el producto de esta publicación.

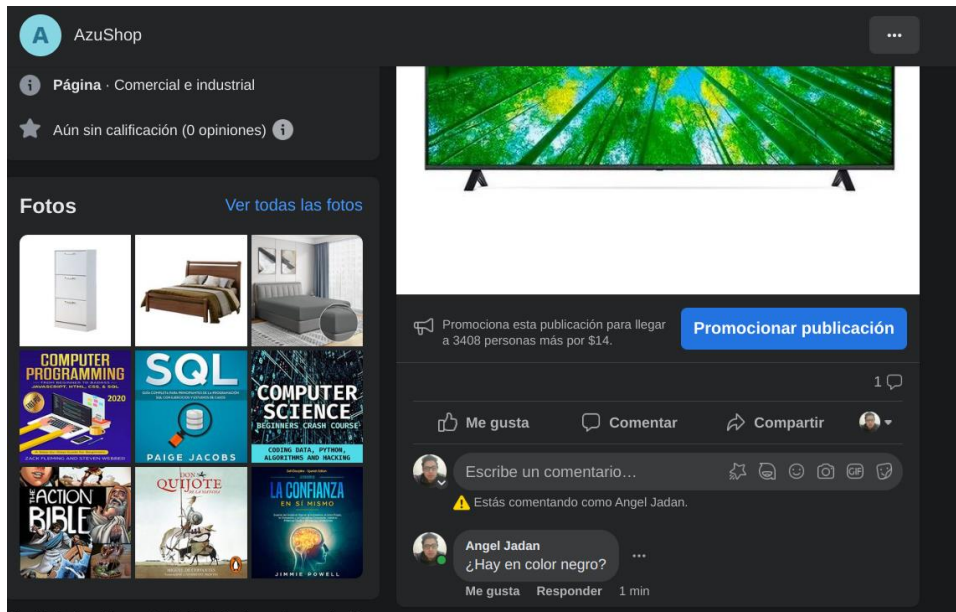


Figura 23. Nuevo comentario a otra publicación de producto.

Vista de producto con recomendados actualizado.

Ahora si nuevamente revisamos el producto, podremos ver que la recomendación nuevamente ha cambiado, puesto que este nuevo comentario es mucho más similar al original, este nuevo producto sale en la lista de recomendados.

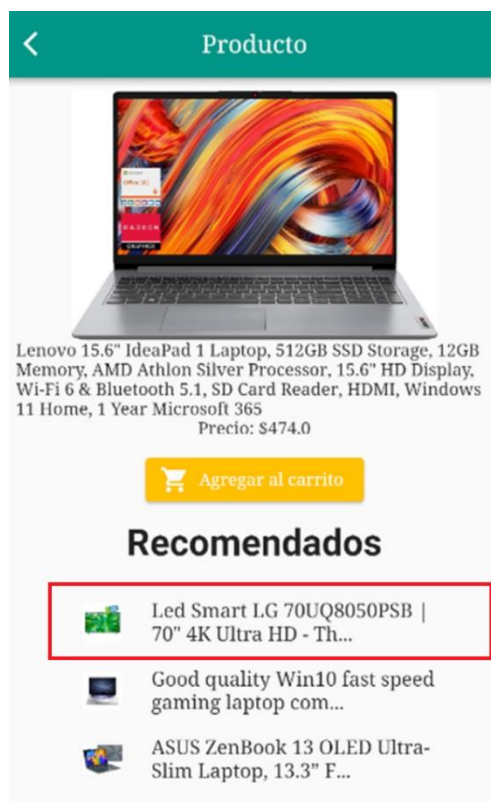


Figura 24. Vista de producto con nuevas recomendaciones.

Agregación de producto

En caso de querer comprar el producto al dar clic en el botón “Agregar al carrito”, saldrá un mensaje que dice “Producto agregado”, para luego poder ir al carrito de compras y realizar el pago.



Figura 25. Vista de producto agregado al carrito de compras.

Listado de productos agregados

En “la siguiente imagen se puede visualizar los productos agregados al carrito de compras con la opción de realizar el pedido”.

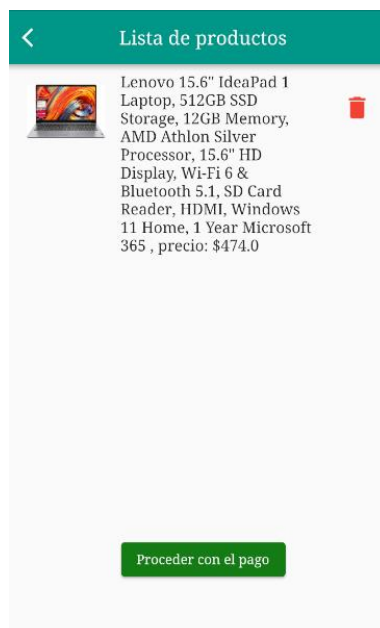


Figura 26. Listado de productos para comprar y proceder con el pago.

Botón de pago

Ahora para finalizar la compra podemos observar la opción para registrar el pago con el botón electrónico de Stripe, esta interfaz ya nos da creado Stripe.



Figura 27. Vista de botón de pagos.

Ingreso de datos de tarjeta

Al poner en “Realizar pago”, nos mostrara la siguiente imagen para llenar “los datos de la tarjeta de crédito o debitó”, en Stripe, “nos dan los datos de una tarjeta de crédito de pruebas, para las pruebas de la aplicación”.

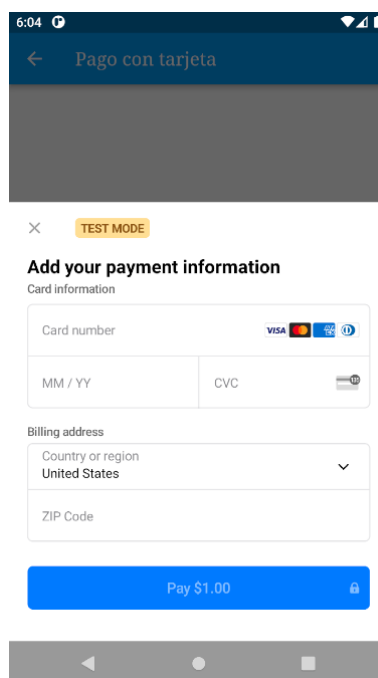


Figura 28. Ventana de registro de datos de tarjeta de crédito.

Módulo de recomendación: Para el desarrollo del modelo de recomendación, se utiliza 2 fases, que consiste en extraer los datos que son los comentarios de cada publicación de Facebook, para luego ser procesados, en forma que quitaremos los datos que nos son útiles.

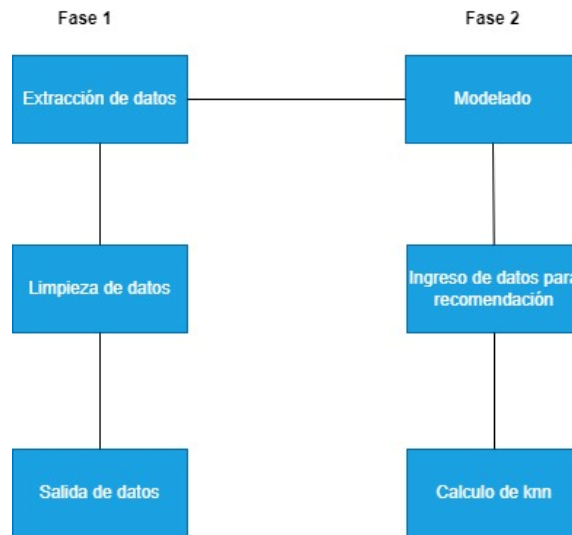


Figura 29. Arquitectura de modelo de recomendación

Preparación de datos: Como ya mencionado anteriormente, primero necesitamos sacar los datos para poder realizar la recomendación de los productos, por lo que extraer y limpiar los datos, para quedarnos con los Ids y los comentarios de cada publicación, el Id de cada publicación nos sirve para identificar a que producto pertenece cada publicación.

Modelado: En este modelo de algoritmo no permite realizar una prueba, esto debido a que trabaja en comparación con las publicaciones y sus comentarios más cercanos, por lo que el producto con su ID de publicación se genera una colección de productos con comentarios similares.

Tabla 6. Datos de parámetros para algoritmo de recomendación

Parámetro	Descripción	Valor
Conjunto de datos	Datos de comentarios, de publicaciones	n
Dato de ingreso	Producto, con su ID, con sus comentarios en las publicaciones.	1
n recomendados		2

Con los datos mostrados en la tabla anterior, podemos realizar la predicción de los datos, en base a los comentarios de las publicaciones de los productos, y el producto que reciben

(ID), se comprara cuáles son los productos más cercanos o similares al que se está ingresando.

En la siguiente grafica podemos ver los comentarios que ha recibido cada publicación.

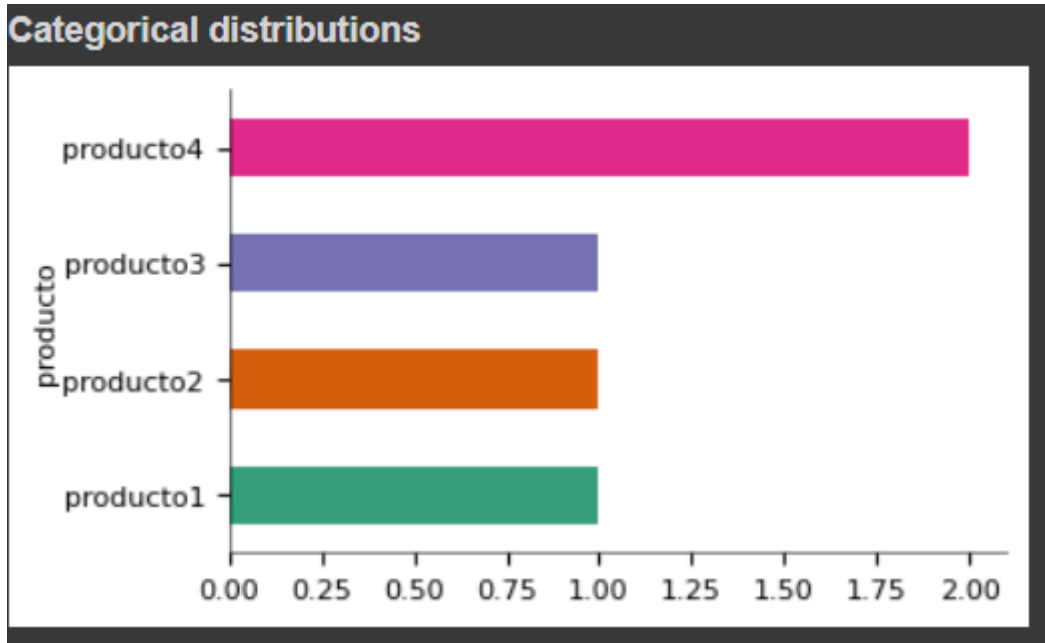


Figura 30. Grafica que publicaciones y sus comentarios.

En la gráfica siguiente podemos ver los productos y sus comentarios, que tiene cada una de ellas.

index	producto	comentarios
0	producto1	¿Sigue disponible?. Esta muy caro. Mas informacion. ¿Sigue disponible?. Muy buen producto. ¿En que color tiene?
1	producto2	... Me gusta. Mas informacion. ¿Sigue disponible?. Muy buen producto. ¿En que color tiene?
2	producto3	Interesante. Esta muy caro. Mas informacion. ¿Sigue disponible?. Muy buen producto. ¿En que color tiene?. ¿Aún sigue disponible?
3	producto4	¡klklj kloo00. ¿En que color tiene?. ¿Aún sigue disponible?
4	producto4	¡klklj kloo00. ¿En que color tiene?. ¿Aún sigue disponible?

Figura 31. Tabla de resultados de API de Facebook, procesado

```
Recomendaciones para el producto4: ['producto4', 'producto3']
```

Figura 32. Resultado de recomendaciones

7. CRONOGRAMA

Actividad Realizada	Duración	Fecha inicio	Fecha fin	Encargado
OE1. ACT 1. Investigar y estudiar sobre el desarrollo de aplicaciones pwa.	15	18/10/2021	24/10/2021	AJ
OE1. ACT 2. Investigar y estudiar sobre los diferentes procesos de contenido de datos de perfiles de las diferentes redes sociales.	15	25/10/2021	31/10/2021	AJ
OE1. ACT 3. Definir los contenidos útiles del perfil.	15	1/11/2021	7/11/2021	AJ
OE1. ACT 4. Documentar información recopilada.	10	8/11/2021	10/11/2021	AJ
OE1. ACT 5. Revisión por parte del tutor.	1	11/11/2021	11/11/2021	RH
OE2. ACT 1. Investigación sobre arquitecturas de microservicios.	10	12/11/2021	31/12/2021	AJ
OE2. ACT 2. Diseño de arquitectura de microservicios.	15	6/1/2022	31/01/2022	AJ
OE2. ACT 3. Desarrollo de microservicios sobre la arquitectura diseñada.	15	2/2/2022	20/2/2022	AJ
OE3. ACT 1. Investigar sobre los diferentes tipos de algoritmos y sus aplicaciones.	15	1/3/2022	15/3/2022	AJ
OE3. ACT 2. Evaluar los diferentes algoritmos en base a las necesidades que requeridas y sus características.	15	16/3/2022	27/6/2022	AJ
OE3. ACT 3. Implementar el o los algoritmos definidos para “la recomendación de los productos en base a los datos de contenidos de perfiles” de las redes sociales.	10	28/6/2023	28/6/2023	AJ
OE3. ACT 4. Documentar información recopilada.	10	1/7/2022	15/8/2022	AJ
OE3. ACT 5. Revisión por parte del tutor.	1	16/7/2023	16/7/2023	RH
OE4. ACT 1. Definir las métricas de evaluación, de rendimiento y confiabilidad.	25	1/9/2022	15/12/2022	AJ
OE4. ACT 2. Definir las herramientas de validación de laboratorio.	15	1/1/2023	31/1/2023	AJ
OE4. ACT 3. Validación y experimentación de evaluación, rendimiento y confiabilidad.	15	15/2/2023	30/6/2023	AJ

OE4. ACT 4. Desarrollar informe de resultados en base a la experimentación de laboratorio.	25	1/7/2023	25/7/2023	AJ
OE4. ACT 5. Documentar información recopilada.	10	26/7/2023	30/11/2023	AJ
OE4. ACT 6. Revisión por parte del tutor	1	01/12/2023	04/01/2024	RH
REVISIÓN/CORRECCIÓN	15	05/01/2024	08/01/2024	AJ

Total, de horas: Desarrollador 700

Fecha de inicio: 18/10/2021

Fecha de finalización: 09/01/2024

8. PRESUPUESTO

DENOMINACIÓN	CANTIDAD	COSTO UNITARIO	COSTO TOTAL
	unidades	dólares	dólares
1. Bienes			
Copias	100	0,05	5
Impresiones	100	0,1	10
Empastado	2	10	20
2. Tecnológico			
PC	1	1100	1100
Smartphone	1	200	200
Servidor en la nube	1	20	20
3. Servicios			
Transporte	200	2,5	500
Internet	1	22,4	22,4
Alimentación	200	2,5	500
4. Personal			
Horas de desarrollo	680	8	5440
Horas de consultoría y capacitación	40	2	80
5. Otros			
Imprevistos	1	\$ 100,00	\$ 100,00
Total	1327	\$ 1.467,55	\$ 7.997,40

9. CONCLUSIONES

Para el desarrollo del proyecto, es crucial poseer conocimientos exhaustivos sobre los conceptos inherentes a cada tecnología a emplear, así como comprender detalladamente sus ventajas y desventajas para su implementación.

“En el caso del desarrollo de la aplicación PWA, se han evaluado diversos frameworks, como Ionic, React Native y Flutter”. Se ha optado por Flutter debido a sus ventajas, especialmente en términos de respaldo proporcionado por Google y una activa comunidad de desarrolladores. Respecto a la obtención de datos para el algoritmo, se exploraron diversas opciones, decidiéndose finalmente por el algoritmo KNN, para recopilar datos de los comentarios en las publicaciones de productos en la red social Facebook y dar como salida 3 productos recomendados.

En lo que respecta a la arquitectura del sistema, para el backend se ha elegido desarrollar una aplicación REST con Django Rest Framework. Esta elección se fundamenta en su facilidad de implementación y en la disponibilidad, por defecto, de un panel administrativo que facilita la gestión de la publicación de productos en la mencionada red social, incluyendo imágenes y descripciones.

La validación de los servicios se lleva a cabo mediante Postman, utilizado como “herramienta de pruebas para garantizar el correcto funcionamiento de los servicios”, tanto para la gestión de productos como para la recomendación de estos.

Trabajo futuro

En esta sección del documento podemos detallar lo que podemos realizar a futuro en base a la información que se ha podido observar con el desarrollo de este sistema, pues el algoritmo puede ser mejorado para su implementación y su escalabilidad, inclusive puede ser combinado con otros algoritmos, como para clasificar los comentarios en base a los estados de los comentarios, como estos pueden ser positivos y negativos, pues de esta manera se puede realizar una recomendación mucho más efectiva, además de permitir que esta mejora pueda permitir trabajar con una gran cantidad de datos debido a que el algoritmo KNN, mientras más crezca la base de datos y la recomendación pueda ser más efectiva y ágil, esto , de los productos. Además, para una siguiente versión se espera realizar una conexión con Firebase, esto para poder realizar el registro de usuarios e inicio de sesión con cuentas externas como Google.

Para las pruebas más efectivas se espera realizar con personas ya utilizando la aplicación, mismas que puedan dar su opinión en cuanto a la efectividad, “el diseño de la interfaz de la aplicación”, pues es mucho más fácil que los usuarios finales nos brinden su opinión y punto de vista, e incluso se espera poder integrar a otras aplicaciones desarrolladas por el autor.

10. Recomendaciones

Primeramente para poder empezar con el desarrollo de sistemas similares es importante tener un conocimiento sobre los diferentes lenguajes de programación, puesto que en este caso, se utilizó principalmente 2 lenguajes de programación como son Python y Dart, también es importante tener un conocimiento sobre servicios REST, e inclusive sobre servicios que están expuestos en Graph, puesto que en este caso Facebook, ya nos brinda el servicio con la información que requerimos, y debemos saber cómo utilizar, para realizar operaciones como son Querys y Mutations.

Para la implementación de los algoritmos inteligentes es muy importante tener conocimientos matemáticos, pues pese a que ya muchos algoritmos ya están listos para ser utilizados, es importante saber sobre cómo se realiza el proceso de cada algoritmo, ya sea este el algoritmo de recomendación o de predicción, esto con el fin de saber que datos vamos a requerir para el algoritmo.

A pesar de que para el desarrollo de las aplicaciones con Flutter, este al compilar ya realiza un código nativo para cada plataforma, es importante tener unos conocimientos básicos de cada plataforma, de los lenguajes de programación en los que se desarrollan para estos dispositivos, ya que existen ciertas configuraciones que debemos tener en cuenta, que está en nativos.

Referencias Bibliográficas

(s.f.).

Abeliuk, A., & Gutiérrez, C. (2021). Historia y evolución de la inteligencia artificial. 8. Santiago de Chile. Obtenido de <https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&cad=rja&uact=8&ved=2ahUKEwj0xvup3v-AAxUFs4QIHZ2DCcEQFnoECBIQAQ&url=https%3A%2F%2Frevistasdex.uchile.cl%2Findex.php%2Fbits%2Farticle%2Fdownload%2F2767%2F2700&usg=AOvVaw2iMjpp-c5A8v96pdRFI9YI&op>

Álvaro, S. (3 de 5 de 2017). *CCCBLAB*. (CCCB) Recuperado el 2023 de 8 de 3, de <https://lab.cccb.org/es/convivir-con-algoritmos-inteligentes/>

Community, jengeb, Nikordaris. (s.f.). *Learning graphql*. Recuperado el 21 de 08 de 2022, de <https://riptutorial.com/Download/graphql.pdf>

Dinsing, T. (s.f.). *Ericsson*. Obtenido de https://www.ericsson.com/en/ci-cd?gad_source=1&gclid=CjwKCAiAkp6tBhB5EiwANTCx1Lmup-awhy5MaoWM5wqb2mJzc2tR7F9BHRIOwy3wjG0MWYTS1M4S5xoCYzQAvD_BwE&gclid=aw.ds#cidintelecom

Django. (1 de 8 de 2023). *Django*. Recuperado el 22 de 8 de 2023, de <https://www.djangoproject.com>

Docusaurus. (2023). *GetX Document*. Obtenido de <https://chornthorn.github.io/getx-docs/docs/state/reactive-state>

- Elternativa. (s.f.). *Elternativa*. Recuperado el 5 de 8 de 2023, de <https://www.elternativa.com/blog-elternativa/historia-inteligencia-artificial/>
- Google. (2023). *Google Privacidad y Términos*. Obtenido de <https://policies.google.com/privacy?hl=es>
- Google. (s.f.). *Google*. Obtenido de <https://support.google.com/googleplay/android-developer/answer/6112435?hl=es-419>
- Gutierrez, J., Escalona, M., Mejías, M., & Reina, A. (2006). *Modelos de Pruebas para Pruebas de Sistemas*. Barcelona: CIMNE.
- IBM. (s.f.). *IBM*. Obtenido de <https://www.ibm.com/topics/knn#:~:text=The%20k-nearest%20neighbors%20algorithm%2C%20also%20known%20as%20KNN%20or,of%20an%20individual%20data%20point.>
- IONOS. (23 de 6 de 2020). *Digital Guide*. Recuperado el 5 de 8 de 2023, de <https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/modelo-v/>
- James Davis, P. (s.f.). *Softcover*. Obtenido de https://www.softcover.io/read/92780ad5/django_book
- Microsoft. (s.f.). *Microsoft*. Obtenido de <https://learn.microsoft.com/en-us/azure/architecture/patterns/claim-check>
- Napoli, M. L. (2020). *Beginning Flutter: A Hands On Guide To App Development*. Indianapolis: Wrox.
- Portalanza, A., Bernal, J. D., Duque, J., Barbery, D., Villacís, G. F., Otatti, L., . . . Rodriguez, H. (2022). *ESTUDIO DE TRANSACCIONES NO PRESENCIALES EN ECUADOR*. Obtenido de UESS.
- Pub.dev. (s.f.). *Pub.dev*. Recuperado el 23 de 8 de 2022, de <https://pub.dev/packages/get#about-get>
- Red Hat. (2 de 6 de 2022). *Red Hat*. Obtenido de <https://www.redhat.com/en/topics/api/what-are-application-programming-interfaces>