

**Construcción de una aplicación web y aplicación móvil que administre los datos
públicos de las farmacias Marthita de la ciudad de Quito**

Caicedo Olaya, Leonardo Iván, Dávalos Guamán, Marco Joel y Maldonado Cedeño, Victoria

Samira

Departamento de Eléctrica y Electrónica

Carrera de Redes y Telecomunicaciones

Monografía, previo a la obtención del título de Tecnología Superior en Redes y
Telecomunicaciones

Ing. Caiza Caizabuano, José Rubén. Mgtr

8 de agosto del 2022

Latacunga

Reporte de Verificacion de Contenido



ProyectoDatosPublicosTecnologia.pdf

Scanned on: 12:42 July 27, 2022 UTC



Overall Similarity Score



Results Found



Total Words in Text

Identical Words	179
Words with Minor Changes	0
Paraphrased Words	0
Omitted Words	1715



Website | Education | Businesses

Firma:

.....
Ing. Caiza Caizabuan, José Rubén Mgtr

Director



ESPE

UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

3

Departamento de Eléctrica y Electrónica

Carrera de Redes y Telecomunicaciones

Certificación

Certifico que la monografía: **“Construcción de una aplicación web y aplicación móvil que administre los datos públicos de las farmacias Marthita de la ciudad de Quito”** fue realizada por los señores **Caicedo Olaya, Leonardo Iván; Dávalos Guamán, Marco Joel; Maldonado Cedeño, Victoria Samira**, la misma que cumple con los requisitos legales, teóricos, científicos, técnicos y metodológicos establecidos por la Universidad de las Fuerzas Armadas ESPE, además fue revisada y analizada en su totalidad por la herramienta de prevención y/o verificación de similitud de contenidos; razón por la cual me permito acreditar y autorizar para que se la sustente públicamente.

Latacunga, 8 de agosto del 2022.

Firma:

.....
Ing. Caiza Caisabuano, José Rubén. Mgtr

C. C.: 059265429-6

Departamento de Eléctrica y Electrónica

Carrera de Redes y Telecomunicaciones

Responsabilidad de Autoría

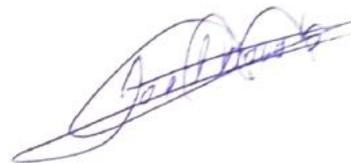
Nosotros, **Caicedo Olaya, Leonardo Iván; Dávalos Guamán, Marco Joel; Maldonado Cedeño, Victoria Samira** con cédulas de ciudadanía n° 172797182-0; 175152780-3; 175274979-4, declaramos que el contenido, ideas y criterios de la monografía: **Construcción de una aplicación web y aplicación móvil que administre los datos públicos de las farmacias Marthita de la ciudad de Quito** es de nuestra autoría y responsabilidad, cumpliendo con los requisitos legales, teóricos, científicos, técnicos, y metodológicos establecidos por la Universidad de las Fuerzas Armadas ESPE, respetando los derechos intelectuales de terceros y referenciando las citas bibliográficas.

Latacunga, 8 de agosto del 2022.



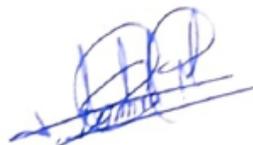
Caicedo Olaya, Leonardo Iván

C. C.: 172797182-0



Dávalos Guamán, Marco Joel

C. C.: 175152780-3



Maldonado Cedeño, Victoria Samira

C. C.: 175274979-4



ESPE

UNIVERSIDAD DE LAS FUERZAS ARMADAS

INNOVACIÓN PARA LA EXCELENCIA

5

Departamento de Eléctrica y Electrónica

Carrera de Redes y Telecomunicaciones

Autorización de Publicación

Nosotros **Caicedo Olaya, Leonardo Iván; Dávalos Guamán, Marco Joel; Maldonado Cedeño, Victoria Samira** con cédulas de ciudadanía n° 172797182-0; 175152780-3; 175274979-4, autorizamos a la Universidad de las Fuerzas Armadas ESPE publicar la monografía: **Construcción de una aplicación web y aplicación móvil que administre los datos públicos de las farmacias Marthita de la ciudad de Quito** en el Repositorio Institucional, cuyo contenido, ideas y criterios son de nuestra responsabilidad.

Latacunga, 8 de agosto del 2022.

Caicedo Olaya, Leonardo Iván

C. C.: 172797182-0

Dávalos Guamán, Marco Joel

C. C.: 175152780-3

Maldonado Cedeño, Victoria Samira

C. C.: 175274979-4

Agradecimientos y Dedicatorias

Agradecimientos

En el planteamiento y ejecución de este proyecto, participaron personas que de manera directa o indirecta influyeron en el resultado del mismo. El cual puedo decir con júbilo, que estoy más que satisfecho por haber llegado hasta la resolución final que se plantea en el presente documento.

Mis padres, que me concedieron con la oportunidad de estudiar la carrera de Redes y Telecomunicaciones, y me apoyaron con soluciones acertadas en ciertos argumentos. Docentes de la Universidad de las Fuerzas Armadas, quienes me permitieron desarrollar este proyecto y su aporte en conocimiento adicional. Compañeros, que compartieron junto a mí el esfuerzo en la elaboración y su proactividad de trabajo en equipo. La franquicia Farmacias Marthita, la cual nos facilitó todos los recursos necesarios para poder llevar a cabo este designio y su pronta retroalimentación de sus requerimientos.

Autores de las fuentes de información que me permitieron obtener el conocimiento necesario para llevar a cabo la ejecución e implementación. La comunidad de código abierto, por el desarrollo libre de sus tecnologías que facilitaron la realización y mejoraron en gran medida los resultados obtenidos.

Expreso mi más grande sentimiento de gratitud a todas las personas mencionadas con las cuales pude llegar a esta solución en este proyecto.

Caicedo Olaya, Leonardo Iván

Dedicatoria

“Una red está compuesta por varios dispositivos que se comunican entre sí, con el objetivo de obtener un resultado más grande”. Es la manera en la que se desarrolló este proyecto, por una red de maestros, amigos y familia que estuvieron acompañándome en esta aventura en busca de un aporte más a la sociedad.

Es por ello que dedico este trabajo a mis padres, Lupe e Iván como pilares de amor incondicional que me inculcaron sus valores y conocimiento en cada etapa de mi vida. Todo para formar un ser humano amante del crecimiento de la humanidad, que vive en la constante búsqueda de soluciones efectivas que aporten un valor más al mundo.

A toda mi familia porque con sus oraciones, consejos y palabras de aliento hicieron de mí una mejor persona y de una u otra forma me acompañan en todos mis sueños y metas. Una dedicación especial a mi abuelita Carmen, que dejó su energía y fuerza en mi corazón después de su partida, por ella se queda en mí, una razón más para esforzarme por mis metas.

Caicedo Olaya, Leonardo Iván

Agradecimientos

Al concluir con el presente trabajo quiero agradecer a las personas que se involucraron en el proyecto, a la franquicia de farmacias “Farmacias Marthita” por permitimos desarrollar el proyecto, así como también su apoyo económico, por otra parte, quiero agradecer a mis padres que me apoyaron durante todo el periodo de desarrollo, agradezco también la ayuda brindada por mis docentes a cargo por guiarme en el proceso de investigación, agradezco a mis hermanos que fueron los encargados de darme ánimos para seguir desarrollando la tesis, agradezco a mis familiares que siempre creyeron en mí y también me dieron el impulso para seguir con el presente proyecto, agradezco a mis compañeros de tesis que juntos pudimos ayudar a crear el proyecto.

También un agradecimiento a todas las personas que se involucraron en el proyecto por su guía y su disposición a ayudarnos a nuestros profesores, así como también a nuestro director de carrera el encargado de que este proyecto se desarrollara.

Dávalos Guamán, Marco Joel

Dedicatoria

El proyecto está dedicado a mi padre que no está con nosotros y espero que donde quiera que este se sienta orgulloso, así como también, a mi madre que es la que siempre me apoya en todas mis decisiones y la que está conmigo, así como también mi abuelita que me guio para poder realizar la tesis, a mis hermanos siempre estuvieron ahí ayudándome a desarrollar la tesis. Este proyecto va dedicado a toda mi familia que son el motor que me impulsa a seguir adelante, también mi enamorada la que me ayuda en todo y esta para mí, a mi abuelita Encarna que es la persona más buena y siempre me da ánimos para sobresalir. También a todas las personas que de alguna forma me supieron guiar en el camino, así como también a las que me apoyaron en el desarrollo del presente proyecto.

Dávalos Guamán, Marco Joel

Agradecimientos

Dios es bondad y amor, el me permite mirar lo bueno de la vida y las personas. Esta etapa nueva de mi vida trae muchas emociones, pero nunca lo hecho sola, gracias a mis padres quienes me brindaron la opción de tener una buena educación y haberme formado como una buena mujer. Mi abuelita ha sido mi pilar fundamental y la que me apoyo en todas mis decisiones a mis hermanas quien trato de ser dar el mejor ejemplo de constancia y trabajo arduo en cada cosa que realicen. Quiero exaltar el trabajo de mis compañeros de tesis quien aprendí mucho más al trabajar en equipo a pesar que lo largo de estos dos años hemos pasado procesos muy fuertes en nuestras vidas, pero eso no impidió que culminemos la meta que tanto anhelamos. Agradezco a la universidad quien forma jóvenes talentosos, a cada profesor quien con esfuerzo daban todo de ellos.

Para finalizar agradecer a cada persona que hizo esto posible indirecta o directamente. A toda mi familia quien son el motor de cada cosa que haga, esto es el comienzo de muchos triunfos más.

Maldonado Cedeño, Victoria Samira

Dedicatoria

Mirar el pasado al ver un obstáculo, solo me levanté con la frente en alto y seguí luchando por mis sueños “Cada etapa de nuestra vida tiene un propósito, donde conocemos personas que nos dan empujones para no rendirnos y culminar la meta.

Dedicare este proyecto a cada persona indirectamente que fue parte de mi proceso de crecimiento a lo largo de mi vida. Miro al cielo y veo a mi angelito “mi hermanito” quien me dejó muy tempranamente y al recordar me da las fuerzas y no desmayar, nada fue fácil en el camino, pero Matías para ti va este proyecto. Como olvidar a mi Bertita y Cintia quienes me enseñó la importancia del amor, a pesar que los días de soledad al estar en una ciudad donde no conoces a nadie sin tu familia me dieron ánimo para no rendirme, ellas me hicieron una mujer fuerte y luchadora. Esto va para mis tres angelitos que tengo en el cielo. Como olvidar a mis padres quien por ellos soy quien soy ahora. “El éxito no se logra con cualidades especiales, es sobre todo el trabajo de constancia, de esfuerzo y organización

Maldonado Cedeño, Victoria Samira

ÍNDICE DE CONTENIDO

Carátula	1
Reporte de Verificación de Contenido.....	2
Certificación	3
Responsabilidad de Autoría.....	4
Autorización de Publicación	5
Agradecimientos y Dedicatorias.....	6
Índice de tablas	20
Índice de figuras	22
Resumen.....	24
Abstract	25
Capítulo I Introducción	26
Antecedentes.....	26
Planteamiento del Problema.....	27
Justificación	27
Objetivos.....	28
<i>Objetivo General:</i>	28
<i>Objetivos Específicos:</i>	28
Alcance	28
Capítulo II Fundamentación Teórica y Referencial.....	30
Software de Aplicación	30
<i>Aplicaciones Web</i>	30
Definición de Aplicaciones Web.....	30
Estructura de Aplicaciones Web	30
Tipos de Aplicaciones Web	30
<i>Aplicaciones Web Estáticas</i>	31

<i>Aplicaciones Web Dinámicas</i>	31
<i>Portales para Aplicaciones Web</i>	31
<i>Aplicación Web con “Gestor de Contenidos”</i>	31
<i>Aplicaciones Web Animadas</i>	31
Uso de Aplicaciones Web.....	32
<i>Ejemplos de Aplicaciones Web Estáticas</i>	32
<i>Ejemplos de Aplicaciones Web Dinámicas</i>	32
<i>Ejemplos de Portales para Aplicaciones Web</i>	32
<i>Ejemplos de Aplicaciones Web con “Gestor de Contenidos”</i>	32
<i>Ejemplos de Aplicaciones Web Animadas</i>	32
<i>Aplicaciones móviles</i>	32
Definición de Aplicaciones Móviles	32
Estructura de Aplicaciones Móviles.....	33
Tipos de Aplicaciones Móviles.....	33
<i>Aplicaciones Nativas</i>	33
<i>Aplicaciones Web o Web App</i>	33
<i>Aplicaciones Híbridas</i>	33
Uso de Aplicaciones Móviles.....	33
<i>Ejemplo Aplicaciones Nativas</i>	33
<i>Ejemplo Aplicaciones web o web app</i>	33
<i>Ejemplo Aplicaciones Híbridas</i>	34
Aplicación Web Progresiva (PWA).....	34
Aplicaciones Web Móviles vs. PWA.....	34
Tecnologías Web.....	35
<i>Interfaz de aplicaciones para transferir datos (API REST)</i>	35
Interfaz de Programación de Aplicaciones (API).....	35

Transferencia de Estado Representacional (REST).....	36
<i>API Gateway</i>	37
Kong.....	37
<i>Konga</i>	38
<i>Lenguajes de programación</i>	38
<i>Lenguaje de Merca de Hipertexto (HTML)</i>	38
Hojas de Estilo en Cascada (CSS).....	38
JavaScript.....	39
Python.....	39
NodeJS.....	39
ReactJS.....	40
JSX	40
Servidor web y servicios en la nube.....	41
<i>Servicio Web</i>	41
<i>Modelo Cliente Servidor</i>	41
Servidor.....	41
Cliente	41
<i>Protocolo TCP/IP</i>	42
Protocolo de Transferencia de Hipertexto.....	42
<i>¿Qué significa HTTP?</i>	42
<i>¿Cómo funciona?</i>	42
<i>Servidor Node.js</i>	42
<i>Amazon Web Services</i>	43
<i>Google Cloud</i>	44
Qué es Google Cloud Platform.....	44
Farmacias	45

<i>Historia de las Farmacias</i>	45
Farmacia Arcaica.....	45
Tecnología y Farmacia.....	46
Comercio y Farmacia	46
<i>Farmacias del Ecuador</i>	46
Metodología de Desarrollo de Aplicaciones.....	47
<i>Metodología Scrum</i>	47
Antecedentes.....	47
Definición.....	47
Desarrollo de Software Agile.....	48
Perfiles de la Metodología Scrum	48
<i>Product Owner</i>	48
<i>Scrum Master</i>	48
<i>Scrum Team</i>	48
<i>Stakeholder</i>	48
Etapas de la Metodología Scrum.....	49
<i>Etapa 1: Product Backlog</i>	49
<i>Etapa 2: Sprint Backlog</i>	49
<i>Etapa 3: Sprint Review</i>	50
<i>Etapa 4: Sprint Planning Meeting</i>	51
Ciclo de vida de un proyecto de Software.....	51
<i>Recopilación de los Requerimientos</i>	51
<i>Análisis</i>	52
<i>Limitaciones</i>	52
<i>Especificación</i>	52
<i>Diseño y Arquitectura</i>	52

<i>Programación</i>	52
<i>Pruebas de software</i>	52
<i>Implementación</i>	53
<i>Mantenimiento</i>	53
Diseño para arquitecturas para Aplicaciones	53
<i>Microservicios</i>	53
<i>Monolítico</i>	53
Devops	54
<i>Qué es Devops</i>	54
<i>Docker</i>	55
<i>Kubernetes</i>	55
<i>Git</i>	55
Gitlab.....	55
Motor de búsqueda	55
<i>Elasticsearch</i>	56
<i>Monstache</i>	57
<i>Kibana</i>	57
Big Data	57
Capítulo III Desarrollo	59
Metodología SCRUM	59
Análisis de Encuestas.....	62
<i>Encuestas Personal de las Sucursales</i>	62
Pregunta 1 ¿Conoce sobre aplicaciones web o móviles?	62
Pregunta 2 ¿Considera usted que se deberían administrar los datos de las farmacias con aplicaciones web o móviles?	62

Pregunta 3. ¿Considera necesario que se implemente una aplicación para consultar los productos de la farmacia?	63
Pregunta 4 ¿Considera usted que los clientes utilizarían la aplicación web o móvil de las farmacias Marthita?.....	63
Pregunta 5 ¿Considera necesario que se le facilita un sistema de facturación para las farmacias?	64
Pregunta 6 ¿Considera usted que la aplicación web y móvil facilita a los empleados realizar consultas sobre recetas médicas sin necesidad de visualizar el producto?	64
Pregunta 7 ¿Considera usted que este proyecto de implementación de una aplicación web y móvil es una herramienta útil ?	65
Pregunta 8 ¿Considera que con esta implementación aumentará las ventas de sus farmacias?	65
Pregunta 9 ¿Cree usted que las farmacias deben implementar tecnología para brindar mejores servicios de atención a los clientes?	66
Requerimiento	66
<i>Requerimientos de negocio</i>	66
<i>Análisis del mercado</i>	66
<i>Funcionalidades</i>	67
Análisis	67
<i>Análisis de procesos de negocios</i>	67
<i>Desarrollar casos de uso</i>	68
Diseño	70
<i>Investigación de la Oferta</i>	70
Servidores Web	70
Bases de Datos.....	71

Frameworks o Librerías para Maquetado Frontend	72
Aplicaciones Móviles Multiplataforma	72
Servicios en la Nube	73
Servidor Web para Implementación de APIs	74
Sistema de Integración Continua y Despliegue Continuo	74
Representación Gráfica de Datos para su Análisis	75
Autenticación de Usuarios para API	75
<i>Evaluación de la oferta</i>	76
NodeJS: Servidor Web	76
MongoDB Atlas: Base de Datos	76
Elasticsearch: Motor de Búsqueda y Base de Datos	77
ReactJS: Librería para Maquetado Frontend.....	77
Aplicaciones Web Progresivas: Aplicación web multiplataforma	77
Google Cloud: Servicio en la Nube	78
Flask: Servidor Web para la Implementación de APIs	78
GitLab CI/CD: Sistema de Implementación Continua y Despliegue Continuo	78
Kibana: Representación Gráfica de Datos para Análisis.....	79
JSON Web Token (JWT): Autenticación de Usuarios para API.....	79
<i>Arquitectura de servicios</i>	79
Arquitectura Básica de la Aplicación.....	79
Arquitectura del Frontend.....	80
Arquitectura del Backend	81
Arquitectura para Datos Públicos	82
Arquitectura para Conexión de Clientes a Kubernetes	83
<i>Ingeniería de procesos de negocio</i>	83
<i>Documentación tradicional</i>	85

<i>Diseñar casos de prueba</i>	88
<i>Diseño de Interfaz de Usuario</i>	90
Diseño gráfico conceptual y experiencia de usuario.....	90
Diseño gráfico digital	90
Maqueta.....	92
Implementación	93
<i>Desarrollo de servicio de negocio</i>	93
<i>Integración con la plataforma</i>	96
<i>Implementación</i>	99
Aplicación Móvil	99
Base de Datos.....	100
GitLab CI/CD	101
Kubernetes: Google Cloud	102
<i>Pruebas Unitarias</i>	104
Aplicación web	104
Microservicios	105
<i>Pruebas de Integración</i>	106
<i>Pruebas de Rendimiento</i>	108
Presupuesto	109
Capítulo IV	111
Conclusiones.....	111
Recomendaciones.....	111
Glosario de Términos.....	112
Bibliografía	114
Anexos.....	118

ÍNDICE DE TABLAS

Tabla 1 <i>Comparación entre aplicaciones web móviles y aplicaciones web progresivas.....</i>	34
Tabla 2 <i>Comparación de frameworks</i>	40
Tabla 3 <i>Metodología Scrum.....</i>	60
Tabla 4 <i>Caso de estudio para usuario administrador.....</i>	68
Tabla 5 <i>Caso de estudio para usuario supervisor.....</i>	69
Tabla 6 <i>Caso de estudio de usuario vendedor.....</i>	69
Tabla 7 <i>Comparativa de ofertas en servidores web.....</i>	70
Tabla 8 <i>Comparativa de ofertas en bases de datos.....</i>	71
Tabla 9 <i>Comparativa de ofertas en librerías para maquetado frontend.....</i>	72
Tabla 10 <i>Comparativa de ofertas en aplicaciones multiplataforma</i>	73
Tabla 11 <i>Comparativa de ofertas en servicios en la nube.....</i>	73
Tabla 12 <i>Comparativa de ofertas en de servidores web.....</i>	74
Tabla 13 <i>Comparativa de ofertas en sistema de integracion continua y despliege continuo.....</i>	74
Tabla 14 <i>Comparativa de ofertas en representación gráfica de datos para su análisis.....</i>	75
Tabla 15 <i>Comparativa de ofertas en Autenticación de usuarios</i>	76
Tabla 16 <i>Casos de prueba.....</i>	88
Tabla 17 <i>Resultados obtenidos para la carga de la aplicación en navegadores y carga de componentes.....</i>	104
Tabla 18 <i>Resultados obtenidos en los microservicios para su tiempo de consulta optimizado.....</i>	106
Tabla 19 <i>Resultados obtenidos en pruebas para la integración entre microservicios y aplicación web.....</i>	107

Tabla 20 <i>Resultados obtenidos en pruebas de rendimiento</i>	108
Tabla 21 <i>Presupuesto primario</i>	109
Tabla 22 <i>Presupuesto secundario</i>	109
Tabla 23 <i>Costo Total</i>	110

ÍNDICE DE FIGURAS

Figura 1 <i>Aplicaciones web</i>	30
Figura 2 <i>Conexión de la aplicación web con microservicios</i>	37
Figura 3 <i>Google Cloud</i>	44
Figura 4 <i>Microservicios</i>	54
Figura 5 <i>Elasticsearch</i>	56
Figura 6 <i>Encuesta pregunta 1</i>	62
Figura 7 <i>Encuesta pregunta 2</i>	62
Figura 8 <i>Encuesta pregunta 3</i>	63
Figura 9 <i>Encuesta pregunta 4</i>	63
Figura 10 <i>Encuesta pregunta 5</i>	64
Figura 11 <i>Encuesta pregunta 6</i>	64
Figura 12 <i>Encuesta pregunta 7</i>	65
Figura 13 <i>Encuesta pregunta 8</i>	65
Figura 14 <i>Encuesta pregunta 9</i>	66
Figura 15 <i>Arquitectura basica de una aplicacion web</i>	80
Figura 16 <i>Comunicación de la aplicación web con el backend y el usuario</i>	81
Figura 17 <i>Comunicación de la API gateway con la aplicación web</i>	82
Figura 18 <i>Comunicación de la API de los datos públicos con usuarios externos</i>	82
Figura 19 <i>Esquema conceptual del funcionamiento NGINX Ingress Controller</i>	83
Figura 20 <i>Comunicación entre bases de datos</i>	84
Figura 21 <i>Esquema de comunicación de API REST</i>	85
Figura 22 <i>Documentación API</i>	85
Figura 23 <i>Interfaz de documentación de API</i>	86
Figura 24 <i>Manual de Usuario</i>	87
Figura 25 <i>Desarrollo de diseño de interfaz facturación con Adobe Illustrator</i>	90

Figura 26 <i>Interfaz de inicio de sesión</i>	91
Figura 27 <i>Diseño de interfaz de ordenes de compras</i>	91
Figura 28 <i>Diseño de interfaz móvil</i>	92
Figura 29 <i>Desarrollo de diseños de interfaz con Sketch</i>	93
Figura 30 <i>Maquetado de diseños Inicio de sesión</i>	93
Figura 31 <i>Maquetación de interfaz de Facturación</i>	94
Figura 32 <i>Codificación en JavaScript</i>	94
Figura 33 <i>Maquetación mediante JSX</i>	95
Figura 34 <i>Aplicación Web Progresiva PWA</i>	95
Figura 35 <i>Archivo “manifest”</i>	96
Figura 36 <i>Conexiones a las APIs mediante Axios</i>	97
Figura 37 <i>Validación de API</i>	97
Figura 38 <i>Codificaciones de las conexiones</i>	98
Figura 39 <i>Validación mediante API</i>	98
Figura 40 <i>Envío de nuevo usuario</i>	99
Figura 41 <i>Estructura Aplicación Web Progresiva</i>	100
Figura 42 <i>Selección del proveedor de Cloud y ubicación del clúster para MongoDB Atlas</i>	100
Figura 43 <i>Interfaz para revisión del estado del clúster</i>	101
Figura 44 <i>Microservicios alojados en GitLab</i>	101
Figura 45 <i>Interfaz de estado para el clúster de Kubernetes</i>	102
Figura 46 <i>Configuración de la ubicación del clúster de Kubernetes</i>	103
Figura 47 <i>Revisión del estado de los servicios implementados en el clúster</i>	103

Resumen

El desarrollo de una arquitectura de software se enfoca principalmente en la mejora continua de los componentes que permita adaptarse a las necesidades de la franquicia que pueden presentarse en ciertos puntos de crecimiento. El sistema va permitir que se muestre datos reales de los productos, ventas, proveedores y ganancias de la empresa gestionando la empresa a nivel del mercado para ello vio necesario cumplir con una alta disponibilidad y escalabilidad a largo plazo debido a la cantidad en crecimiento de usuarios que se pueden encontrar. Por lo que se utiliza una arquitectura de microservicios basada en la réplica de éstos para poder cumplir con los parámetros mencionados, así como también se automatizo diversos procesos mediante la implementación de la aplicación web y móvil. Se demostró la utilidad de esta arquitectura en una microempresa para mejorar las ventas y se logró cumplir con los objetivos planteados inicialmente de generar una aplicación que se adapte a cualquier empresa generalizando los requerimientos de forma global, siendo que la aplicación se pueda utilizar en cualquier ámbito laboral o comercial. Al incluir este servicio de software mejora la calidad de atención y la experiencia con una nueva forma de Business Intelligence que manejan las franquicias de farmacias a nivel nacional.

Palabras clave: arquitectura de microservicios, motor de búsqueda, aplicación web progresiva, análisis de datos, análisis predictivo.

Abstract

The development of a software architecture focuses mainly on the continuous improvement of the components that allow to adapt to the needs of the franchise that may arise at certain points of growth. The system will allow to show real data of the products, sales, suppliers and profits of the company managing the company at the market level for it was necessary to comply with a high availability and scalability in the long term due to the growing number of users that can be found. Therefore, a microservices architecture based on the replication of these is used to meet the above parameters, as well as automating various processes through the implementation of web and mobile application. The usefulness of this architecture was demonstrated in a microenterprise to improve sales and it was possible to meet the initial objectives of generating an application that adapts to any company generalizing the requirements globally, being that the application can be used in any work or business environment. The inclusion of this software service improves the quality of service and the experience with a new form of Business Intelligence that pharmacy franchises manage nationwide.

Key words: microservices architecture, search engine, progressive web application, data analysis, predictive analytics.

Capítulo I

Introducción

Antecedentes

Actualmente las empresas públicas y privadas, cuentan con un software informático que las ayuda con la administración de su información. (*Comunidad_Emagister_8601_laudon.pdf*, s. f.) concuerda que las aplicaciones web tienen un gran alcance e incidencia en las actividades cotidianas de estas empresas, la necesidad que tienen las farmacias ya sean una cadena de farmacias o farmacias individuales es una forma más efectiva de la administración de su información por lo tanto el uso de las aplicaciones web serán una herramienta muy importante para cumplir esta disposición ya que podremos mejorar el uso de la información para que la farmacia pueda implementar diferentes servicios que pueda proveer a sus clientes.

La esencia de cualquier negocio es proveer servicios a los usuarios por ello el desarrollar una aplicación web que tiene muchas ventajas y usos los cuales han evolucionado en diferentes ámbitos ya sea para mejorar el rendimiento de la empresa gestionando de mejor forma la información, también como medio de publicidad cubriendo una necesidad de los clientes.

Por ello la empresa Farmacias Marthita desea implementar una aplicación web que permita gestionar de forma ágil la información de sus productos, clientes y órdenes de compra, aprovechando al máximo esta herramienta la cual desarrollaremos dos partes el aplicativo web y para mejorar y facilitar el uso de esta realizaremos un aplicativo móvil multiplataforma que ayudará a los clientes a relacionarse de manera más directa y sencilla desde un móvil dando un servicio único con el cual la empresa tendrá mayor impacto en los clientes (*Aplicaciones Web y ASPs – Mercadeo.Com*, s. f.).

Planteamiento del Problema

La empresa Farmacias Marthita ubicada en la ciudad de Quito dedicada a la compra y venta de sustancias químicas que son utilizadas para el tratamiento, prevención, curación o diagnóstico de una enfermedad, también denominada droga o agente farmacológico, Se ve en la necesidad de gestionar de mejor manera la información de sus clientes, proveedores por lo que también se realizará una mejora a nivel tecnológico ya que el problema de la empresa es que cuenta con tres sucursales sin ningún control o registro de la información para agilizar los procesos de la empresa: administrativo y operativo.

El impacto que genera la empresa es muy amplio por lo que es importante llevar un control específico en sus respectivas áreas por medio del aplicativo web que le servirá para gestionar de mejor manera los datos que genera está, ayudando al administrador a poder revisar procesos de manera más eficaz llevando un control de las 3 sucursales mediante informes detallados análisis de ventas que se generan,

La aplicación web y móvil brindará a la empresa Farmacias Marthita una mejora en el manejo de información, agilidad en procesos administrativo y operativos, servicios a clientes los cuales ayudará a generar mayor impacto en la población. Todo esto en tiempo real con ayuda de los aplicativos web y móvil.

Justificación

En la actualidad («La historia de las aplicaciones móviles», 2019) las aplicaciones web y móviles han ido evolucionando, cubriendo una amplia gama de problemas que se presentan hoy en día. Como podemos observar la tecnología va a seguir evolucionando por lo que las empresas deben adaptarse a este cambio. Por ello las empresas desean tener sistemas informáticos a su alcance para agilizar los procesos en tiempo real. Con esto la empresa podrá ofrecer soluciones a sus clientes y usuarios consiguiendo reducir costos, aumentar ventas y mejorar la imagen de la empresa.

Con el uso de aplicativos web y móvil podremos mejorar el rendimiento de la empresa de una manera eficaz y confiable por lo que ayudará a la manipulación de información en beneficio de la empresa para ellos es necesario la creación de estos aplicativos ya que así podrán obtener un beneficio de controlar en tiempo real y desde cualquier lugar donde se encuentren por medio de: celulares inteligentes tabletas o computadoras las cuales podrán acceder a estos aplicativos y así dar seguimiento a los procesos de la empresa.

Gracias a la implementación de estos aplicativos los beneficiarios serán todo el personal que trabaja en la empresa ya que podrán efectuar de manera más rápida la venta de productos y facturación de los mismos, también el personal administrativo que podrá gestionar de mejor manera la información para detallar los informes de venta, y gestión de productos.

Objetivos

Objetivo General:

Construcción de una aplicación web y aplicación móvil que administre los datos públicos de las Farmacias Marthita de la ciudad de Quito

Objetivos Específicos:

- Establecer información sobre aplicaciones web y móvil para conocer su funcionalidad en la farmacia
- Analizar la situación actual de la administración de datos en la farmacia
- Desarrollar la aplicación web y móvil en apoyo de las herramientas a utilizar para el proyecto

Alcance

El presente proyecto cuenta con el diseño y construcción de una aplicación web y móvil para las sucursales de la farmacia Marthita, con la que se beneficiarán los dueños, personal de la empresa y clientes. Para realizar el proyecto propuesto se hará uso de uso de la metodología de desarrollo proyectos ágiles "SCRUM" así como también se emplea como marco de desarrollo el lenguaje de JavaScript para el desarrollo de las aplicaciones web y en caso de la

aplicación móvil la librería React que utiliza el mismo código base JavaScript, pero multiplataforma con el sistema operativo para Android.

Al finalizar el proyecto se busca facilitar el uso de las aplicaciones para verificar los productos disponibles, costos de los medicamentos, auditoría e inventario de la farmacia y servicios de consultas generales, verificación de recetas médicas, medicamentos según síntomas que ingrese el usuario y devoluciones de productos a proveedores que están por caducar siendo de gran ayuda para evitar el contacto por la pandemia del covid-19 y adaptar las farmacias a nuevas propuestas tecnológicas para facilitar la administración de datos públicos de aquella.

Capítulo II

Fundamentación Teórica y Referencial

Software de Aplicación

Aplicaciones Web

Definición de Aplicaciones Web

Se define como aplicación web al conjunto de herramientas que permite interactuar con el usuario por medio de un servidor web a través del internet o intranet, en sí es un software que codifica el lenguaje de programación para prestar un servicio a los usuarios («Definición de aplicación web», 2015)

Estructura de Aplicaciones Web

Las aplicaciones web están constituidas por 3 capas:

Capa 1: Cliente de aplicación en este caso el navegador Web

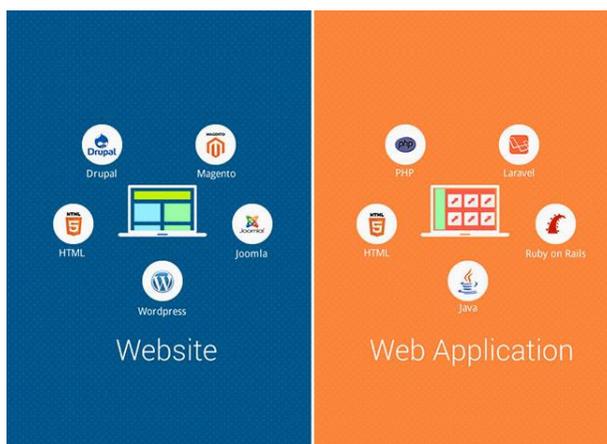
Capa 2: Uso de tecnología web dinámica, por ejemplo: JAVA, ASP.NET, PHP, Python.

Capa 3: Servidor de Datos: base de datos, servidor SMTP, HTTP.

Tipos de Aplicaciones Web

Figura 1

Aplicaciones web



Nota. Debemos tener en cuenta las grandes diferencias entre aplicaciones web. Tomado de (Diferencias entre un sitio web y una aplicación web, 2016).

Encontramos 5 tipos generales de aplicaciones web que se describen a continuación.

Aplicaciones Web Estáticas

La aplicación web estática se basa en una programación simple ya que su codificación utiliza objetos en movimiento como GIF animados, videos, imágenes, etc. El lenguaje se desarrolla en HTML y CSS que por lo general son muy sencillas («¿Qué son las Aplicaciones Web?», 2016)

Aplicaciones Web Dinámicas

La aplicación web dinámica es una herramienta que trabaja junto a la base de datos, es decir, que es una aplicación más completa con constantes cambios, la cual el usuario interactúa y permite que la información va cargando los contenidos a la web y los va actualizando (¿Qué son las Aplicaciones Web?, 2016)

Portales para Aplicaciones Web

Como su nombre lo dice es un página o sitio web diseñada exclusivamente para dispositivos móviles las cuales se adaptan a pantallas más pequeñas, que se ajustan a los requerimientos de la aplicación según sus contenidos («¿Qué son las Aplicaciones Web?», 2016)

Aplicación Web con “Gestor de Contenidos”

Es un sistema de gestión de contenidos (CMS) que permite la interacción del contenido tanto como administrador, editores y usuarios. Se encuentra en constante actualización es por ello que es una de las aplicaciones más utilizadas y populares para las empresas por ejemplo como diarios digitales o blogs (¿Qué tipos de aplicaciones web existen?, 2017)

Aplicaciones Web Animadas

Es una aplicación que usa contenido de efectos animados es por ellos que se encuentra relacionada con la tecnología FLASH, pero ha ido evolucionado utilizando CSS y SVG. Este tipo de aplicación es muy utilizada por diseñadores o desarrolladores de contenido animado por

lo práctica que es para empresas **publicitarias** («¿Qué tipos de aplicaciones web existen?», 2018)

Uso de Aplicaciones Web

Ejemplos de Aplicaciones Web Estáticas

- Currículos digitales
- Empresas para presentar datos de contacto entre otros.

Ejemplos de Aplicaciones Web Dinámicas

- Blogs personales y profesionales.
- Revistas digitales
- Blogs corporativos.
- Páginas de noticias.

Ejemplos de Portales para Aplicaciones Web

- Correo electrónico
- Foros

Ejemplos de Aplicaciones Web con “Gestor de Contenidos”.

- Drupal.
- WordPress.
- Joomla.

Ejemplos de Aplicaciones Web Animadas

- Android

Aplicaciones móviles

Definición de Aplicaciones Móviles

Es un software exclusivamente para dispositivos móviles que automáticamente se adapta a las necesidades del usuario la cual proporciona varias funcionalidades y se adapta al

sistema operativo para el cual ha sido asignado o puede ser **multiplataforma** (Florido-Benítez, 2016)

Estructura de Aplicaciones Móviles

La parte front-end, llamada parte cliente para visualizar

La parte web services, encontramos la lógica del negocio de las aplicaciones

Tipos de Aplicaciones Móviles

Aplicaciones Nativas

Como su nombre lo dice es nativo de acuerdo al sistema operativo para el que ha sido diseñado el cual se adapta a las características y restricciones que necesite el usuario («¿Cuáles son los tipos de aplicaciones móviles?», 2017)

Aplicaciones Web o Web App

Son llamadas multiplataformas y son las más utilizadas ya que se puede adaptar a cualquier sistema operativo que se esté utilizando, la desventaja es que va depender del navegador que se use. No requiere mayor complicidad ya que solo se necesita tener el navegador en el dispositivo móvil (*Tipos de aplicaciones móviles o apps - adaptative*, s. f.)

Aplicaciones Híbridas

Es la combinación de dos apps, diseñadas para ser utilizada en distintos sistemas operativos sin importar el navegador, utiliza un lenguaje de programación que trabaja junto con un framework (*Principales tipos de apps: ventajas e inconvenientes | App&Web*, s. f.)

Uso de Aplicaciones Móviles

Ejemplo Aplicaciones Nativas

- Hyperlapse para Instagram

Ejemplo Aplicaciones web o web app

- El paquete office de google

Ejemplo Aplicaciones Híbridas

- Rede sociales como Facebook, WhatsApp, Instagram, Twitter, etc.

Aplicación Web Progresiva (PWA)

Las aplicaciones web progresivas es una nueva posibilidad de adaptarlas en aplicaciones móviles y poder facilitar el uso de las APIs un ejemplo muy claro es los Service Worker el cual se ejecuta en segundo plano en un script. Es una tecnología que permite al usuario simular estar en una aplicación nativa, superando cualquier limitación (Aguirre et al., 2019).

Aplicaciones Web Móviles vs. PWA

Vamos a identificar cual es la clara diferencia entre las aplicaciones por eso es importante analizar el enfoque hacia el desarrollo de multiplataforma y verificar cual se adapta a mejor manera a nuestra aplicación (Aguirre et al., 2019).

Tabla 1

Comparación entre aplicaciones web móviles y aplicaciones web progresivas

Tecnología / Característica	Aplicaciones Web	PWA
Entorno de desarrollo integrado (IDE)	Múltiples opciones. No hay un IDE oficial.	Múltiples opciones. No hay un IDE oficial.
Lenguaje de programación	HTML, CSS, JavaScript más otro lenguaje del lado del servidor	HTML, CSS, JavaScript más otro lenguaje del lado del servidor
Open source / licencia y costo	Libre	Libre
Diseño de interfaz de usuario	HTML, CSS, JavaScript.	HTML, CSS, JavaScript.
Curva de aprendizaje	Muy bajo	Bajo
Acceso a características específicas del dispositivo.	Muy bajo	Medio

Nota. Podemos distinguir la diferencia entre una aplicación móvil y la aplicación web progresiva.

Tecnologías Web

Estas tecnologías ayudan con el desarrollo de aplicaciones web de ámbito empresarial, así como también existen diferentes tecnologías web que se emplean para compartir archivos en una intranet, así como también en la creación de interfaces de usuario dentro de un software. Estas tecnologías ayudan al desarrollo de aplicaciones web existen tecnologías del cliente que cumple con estándares que ayudan a ser compatible con cualquier dispositivo otras tecnologías que se enfocan en el servidor no es necesario el uso de estándares ya que cada empresa desarrolla su servidor con la tecnología que más le favorezca (Daniel Perovich, 2019)

Interfaz de aplicaciones para transferir datos (API REST)

Se puede definir como un conjunto de protocolos y herramientas que se utilizan para el desarrollo de aplicaciones, su función esencial es la conexión entre el usuario, el software y el dispositivo tecnológico usado. (Arquitectura de una API REST · Desarrollo de aplicaciones web, s. f.)

Interfaz de Programación de Aplicaciones (API)

Este método nos da la facilidad de determinar una programación que permita abstraer la información de otro software mediante comandos que el usuario no visualiza, por lo que un API cumple con proveer un servicio específico con ayuda de comandos funciones y protocolos, que permiten al usuario poder manipular de manera efectiva y eficaz los diferentes softwares que están diseñados con este método ya que el usuario solo manipula y visualiza la interfaz de esta mas no el proceso que desempeña internamente, una de las ventajas más favorables de las API es que ayudan tanto al programador como al usuario ya que el programador puede disponer de otros softwares que ayudan a esta (¿Qué es una API y para qué sirve?, 2015).

Existen 3 clases de interfaces de programación de aplicaciones cada una de ellas con su función específica

APIS locales: Son aquellas que necesitan los propios recursos de ya sea de un software propio incluso de un dispositivo

API web: ésta utiliza recursos y protocolos conocidos para tener una mejor comunicación entre ella un protocolo muy usado por ellas es HTTP

API Programas: utiliza una tecnología RPC llamada a procedimiento remoto que se basa en realizar actividades desde otro dispositivo de forma que parezca que se los realiza desde el dispositivo local

Para realizar una API es necesario tener en cuenta realizar un buen diseño de esta ya que se debe tener en cuenta que las conexiones de esta serán de vital importancia porque la función más importante de api es realizar conexiones de manera eficaz para que el usuario pueda satisfacer la necesidad que tenga (¿Qué es Interfaz de programación de aplicaciones (API) ?, s. f.).

Transferencia de Estado Representacional (REST)

Es un servicio que ayuda a realizar el envío y manipulación de datos a través de servicios web. Este servicio ayuda mucho a que el servidor no colapse por falta de memoria RAM ya que utiliza los recursos independientes por lo que debe tener protocolos sin estado y la función más esencial de este es evitar utilizar los recursos del servidor ya que esta no retiene información («¿Qué diferencias hay entre REST y SOAP?», 2016)

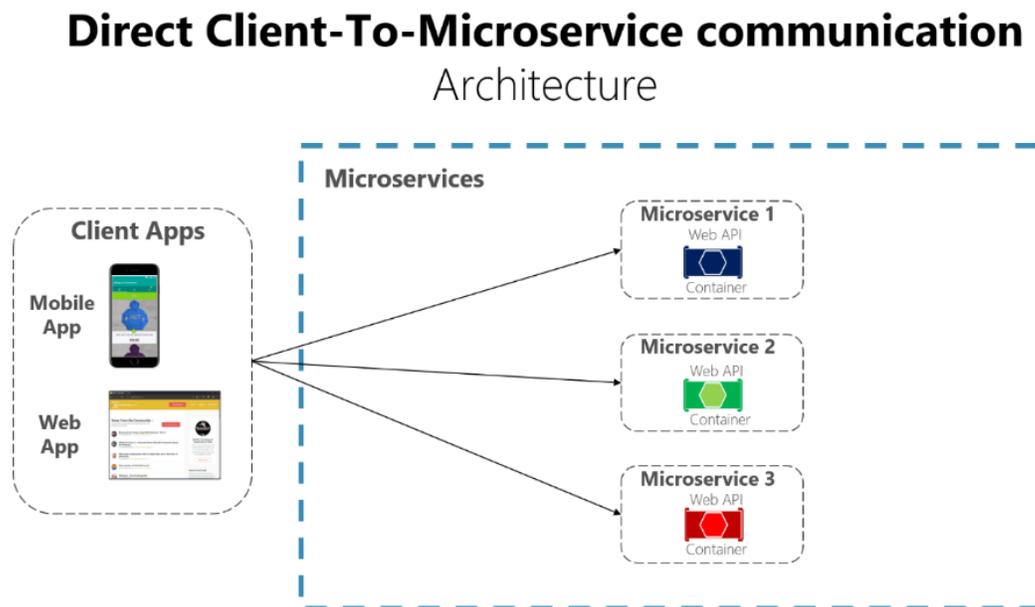
REST nos permite la creación de servicios más ligeros y simples, fácilmente visibles y utilizables a través de cualquier navegador web ya que los protocolos y estándares que usa son http por lo que permite realizar diferentes conexiones de recursos disponibles de las diferentes URI ya que estos son identificables (Naeem, 2020).

Entre las principales ventajas que tiene este servicio es la visibilidad que nos ofrece ya que todo debe estar etiquetado de manera descriptiva para poder identificarlo, otra de las ventajas es su gran escalabilidad ya que si se desea extender más la capacidad de los servidores no tendremos problemas al sincronizar con los usuarios ya que los recursos no se usan del servidor por lo que muchas personas pueden conectarse simultáneamente (Naeem, 2020).

API Gateway

Figura 2

Conexión de la aplicación web con microservicios



Nota. Reconocemos la gran relación entre los microservicios y nuestro api Gateway. Tomado de (Nishanil, s. f.)

Es un sistema que administra la API permitiéndoles crear y protegerla, nos proporciona hosting, registro o supervisión al momento de integrar, la controla gestionando el tráfico de todos los datos. Podemos actualizar el backend sin modificar la API trasladándose de un lugar a otro sin mayor cambio (Google Cloud, 2007).

Kong

Es una puerta de enlace de la API que se trabaja en entornos híbridos y multi nube también posee una entrada nativa de kubernetes, las características principales son: desarrollada para microservicios, Posee un núcleo liviano que permite un rendimiento rápido, es compatible con varios sistemas operativos como el contenedor de Docker (Kuhn, s. f.).

Konga

Nos proporciona una interfaz de usuario gráfica muy dinámica de trabajar al momento de desarrollar, su licencia es gratuita es una gran ventaja que nos brinda este software. Se la administra desde Kong Gateway siendo un sistema multiplataforma. También se puede ejecutar desde Git o Dockerhub (Kong, 2017).

Lenguajes de programación

Es una herramienta que nos permite crear programas con un lenguaje que el computador pueda entender para poder procesar los datos, es la parte fundamental para construir que permite crear algoritmos o realizar cálculos completos (Vértice, 2009).

Lenguaje de Marca de Hipertexto (HTML)

Significa HyerText Markup o también llamado lenguaje de marcas de hipertextos, es un lenguaje básico que tiene hipertextos que posee referencias como enlaces, vínculos a otro texto. Es decir, sirve para estructurar documentos ofreciendo las herramientas necesarias de acuerdo al almacenamiento de servidor de la página web, la gran ventaja que nos ofrece es la compatibilidad y facilidad al momento de diseñar. El programador se encarga de editar y crear el texto, la extensión para guardar el documento es *.html. La sintaxis son pasos que se deben cumplir al momento de construir el programa como son las etiquetas, atributo, elementos etc (Vértice, 2009).

Hojas de Estilo en Cascada (CSS)

Se define como Las hojas de estilo en cascada (CSS, Cascading Style Sheets) nos permite mejorar el diseño de la interfaz en el diseño, posición y aspecto. Separa y organiza el contenido. Al poder administrar bien los datos la velocidad de la página web va a aumentar y economizar la línea de códigos. El científico Håkon Wium Liel en 1994 fue el impulsor de mejorar la página web separando y mejorando al hacerlo en cascada.CSS1:es la primera versión de css que podía configuras estilos como combinación de selectores el cual ya era aprobado en ese tiempo por W3C (Rubiales Gómez Mario, 2021).

JavaScript

El objetivo de este lenguaje de programación es crear páginas web dinámicas que se puedan interpretar y no es necesario compilar en los programas para ejecutar. Todo comenzó por los años 90 el internet iba desarrollando a una gran velocidad y es ahí donde se buscaron nuevas técnicas que permita mejor el desarrollo de las páginas web , agilizándolas y complementarlas en mejor manera, este lenguaje le permitiría ejecutar el programa sin ejecutar el navegador del usuario, Brendan Eich fue el que dio una solución al gran problema de la época, adoptó varias tecnologías como ScriptEase que iba a lanzar en 1995 Inicialmente denominó a su lenguaje LiveScript .Nescape firmó una alianza en mixrosystem y decidió cambiar el nombre a JavaScript(Eguíluz Pérez Javier, s. f.).

Python

Es un lenguaje de programación sencillo y corto de alto nivel, se caracteriza por ser muy elegante y fácil a comparación con otros lenguajes al momento de desarrollarlo. Su entorno de trabajo es muy interactivo e incluso detecta los errores sin necesidad de compilar para poder corregirlos. Su lenguaje está orientado a la programación de objetos. Fue creado por Guido van Rossum quien con el largo de los años fue mejorando la experiencia del usuario con la facilidad de uso (Marzal Varó & Gracia Luengo, 2009).

Python ha sido parte importante de Google desde el principio, y lo sigue a medida que el sistema crece y evoluciona. Hoy día, docenas de ingenieros de Google usan Python y seguimos buscando gente diestra en este lenguaje. PETER NORVIG, director de calidad de búsquedas de Google Inc

NodeJS

Ryan Dahl publicó la primera versión de node.js en 2009 quien evolucionó el concepto de Javascript, el servidor se ejecuta independientemente. Es un entorno para ejecutar la aplicación usando el lenguaje de programación de Javascript. La gran ventaja es que su código

es abierto donde cualquier persona puede acceder para ser instalado en nuestros computadores sin tener que pagar alguna licencia (Puciarelli, 2020).

ReactJS

Guarda el código abierto como una biblioteca de Javascript el cual permite crear interfaces de usuario facilitando el desarrollo de la aplicación. Las características de React.js es ser sencillo y fácil de utilizar. Debemos tener en cuenta que solo maneja el frontend de la aplicación usando el patrón de MVC que significa MODELO-VISTA-CONTROL. La migración de versiones es muy fácil de manejar (Anchundia Medrano, 2022).

Tabla 2

Comparación de frameworks

Parámetros	AngularJS	ReactJS
Tipo	Marco de trabajo completo	Biblioteca JavaScript
Tipo de DOM	DOM real	DOM virtual
Vinculación de datos	Vinculación de datos bidireccional	Vinculación de datos unidireccional
Escrito en	TypeScript	JavaScript
Plantillas	JSX + J% (ES5/ES6)	HTML + TypeScript
Abstracción	Medio	Fuerte
Inclusión de la biblioteca JavaScript en el código fuente	No es posible	Posible
Modelo	Modelo MVC	DOM virtual
Prueba y depuración conjunto adicional de herramientas	Solución completa en una sola herramienta	Necesita un conjunto adicional de herramientas
Libertad	Limitada	Permite elegir las bibliotecas, la arquitectura y las herramientas

Nota. El objetivo es distinguir de forma clara las diferencias entre estos frameworks

JSX

Es una extensión de javascript que utiliza React que nos permite agregar lenguajes como HTML para ejecutar componentes. El proceso que cumple es interpretar el código y

compilarlo en JSX y traducir a JS para enviar la petición al navegador. La gran ventaja es que reduce el código para el desarrollo de la aplicación (Jiménez Calderón, Mario, 2020).

Servidor web y servicios en la nube

Para comprender en qué consiste un servidor web se deben definir los siguientes conceptos que engloban su funcionamiento.

Servicio Web

Se trata de un amplio sistema de acceso a información multimedia que posibilita a los terminales que estén conectados a dicha red la petición de información. Este sistema fue inicialmente desarrollado por el CERN (Centre Européen de Recherche Nucléaire) en 1990 en donde se fue implementando para mejorar el flujo de información que requería dicha entidad en sus investigaciones. En 1992 comenzó a ser implementado al público con el primer navegador web conocido como "Mosaic" (Cobo, Gómez, Pérez, & Rocha, 2005).

Modelo Cliente Servidor

Este modelo define de forma clara cómo se comporta un servidor web al momento de ser utilizado, basándose en dos equipos conectados bajo la misma red en donde se reparte el rol de un cliente y un servidor (Cobo, Gómez, Pérez, & Rocha, 2005).

Servidor

Se trata de equipos informáticos de altos recursos que ofrecen servicios al resto de equipos que estén conectados bajo la misma red. Al tratarse de equipos que prestan servicios es necesario que su dirección IP sea estática para evitar cualquier problema de conexión futura. Estos equipos son en donde se alojan páginas web, aplicaciones móviles, sistemas gestores de bases de datos, entre otros servicios (Cobo, Gómez, Pérez, & Rocha, 2005).

Cliente

Son equipos informáticos utilizados por usuarios individuales que realizan peticiones a un servidor para obtener información siempre y cuando esté conectada bajo la misma red, es la

razón por la que no es necesario mantener una IP estática puesto que solo realiza peticiones mas no las resuelve (Cobo, Gómez, Pérez, & Rocha, 2005).

Protocolo TCP/IP

Es un conjunto de protocolos basado en la estructura del modelo OSI definido para establecer la comunicación entre diferentes dispositivos que estén conectados a diferentes redes. Por esto mantiene los protocolos más fundamentales del modelo OSI para poder obtener un tráfico de información mucho más rápido (Villada, 2014).

Protocolo de Transferencia de Hipertexto

¿Qué significa HTTP?

Es un protocolo de transferencia de hipertextos como su nombre lo dice, es el intercambio de información entre las peticiones que se hace desde el usuario y la respuesta que debe dar el servidor (Editorial Etecé, 2013).

¿Cómo funciona?

Se basa en el principio de cliente-servidor que anteriormente lo habíamos explicado, las peticiones se envían por medio de una entidad que es el usuario que en la mayoría puede ser un navegador web o programa –robot y esa petición se envía al servidor y lo analiza y genera la respuesta. Entre este proceso de intercambio hay intermediarios que los conocemos como proxies (Generalidades del protocolo HTTP - HTTP | MDN, s. f.).

Servidor Node.js

Se trata de un entorno de ejecución de JavaScript orientado a eventos asíncronos, con el objetivo de poder desarrollar una aplicación web escalable. En Node.js se realizan muchas conexiones simultáneas en el código que mantiene al sistema en movimiento hasta que deje de cumplir con devolución de llamadas y es cuando Node.js deja de ejecutarse.

Además, ofrece la posibilidad de establecer un servidor HTTP diseñado para una mejor transmisión de datos con streaming y baja latencia. Es por esto que puede ser utilizado con

bastante facilidad en una aplicación web que trabaja con frameworks web (OpenJS Foundation, 2021).

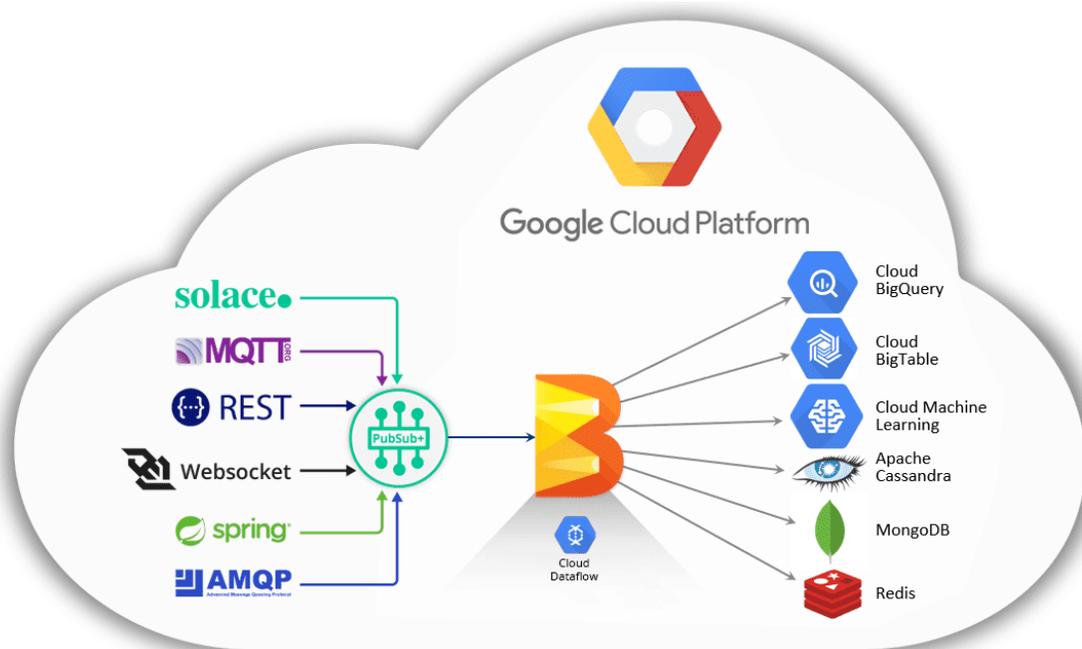
Amazon Web Services

Es un proveedor de nube que ofrece una gran variedad de tecnologías alojando todo el servicio en un servidor elástico adaptado a las necesidades de una empresa con productos desde tecnologías de infraestructura como cómputo, almacenamiento y bases de datos hasta tecnologías emergentes como aprendizaje automático e inteligencia artificial, lagos de datos y análisis e internet de las cosas. Esto hace que llevar las aplicaciones existentes a la nube sea más rápido, fácil y rentable y permite crear casi cualquier servicio o hasta producto que requiera de software específico. (Amazon Web Services, Inc., 2021)

Google Cloud

Figura 3

Google Cloud



Nota. Conocemos como trabaja google cloud plataforma. Tomado de (*Consuelo con Google Cloud Platform - Consuelo, s. f.*)

Qué es Google Cloud Platform

Es una infraestructura que trabaja en la nube de Google que maneja el tráfico de datos de los usuarios, normalmente la utilizan para la implementación de aplicaciones o sitios web. Los servicios que ofrece son: IaaS, SaaS y PaaS que almacena la base de datos en la nube y proporciona soluciones en la red (Google Cloud Platform, s. f.).

La infraestructura como servicio de Google (IaaS): Puede crear máquinas virtuales bajo demanda en la nube. Es excelente para crear un centro de datos en minutos y proteger su infraestructura con recuperación ante algún daño. La plataforma como servicio (PaaS): nos permite crear aplicaciones dándonos la prioridad de ser generando mejores resultados.

Kubernetes le permite organizar sus contenedores en la nube duplicándolos para no congestionarlos (Google Cloud Platform, s. f.).

Farmacias

Historia de las Farmacias

Antes de la existencia de la medicina, se pueden observar muchos factores que llevaron a conformar un establecimiento de venta farmacéutica. Esto se debe a que el ser humano presenta una tendencia natural a mejorar sus condiciones de vida, y, de hecho, esto está estrechamente relacionado con la necesidad de opulencia que el ser humano desea obtener. Por ello, los inicios de las farmacias vienen dados por emprendedores y gente con iniciativa propia que entiende el valor de mejorar la calidad de vida, esto va de acuerdo al pensamiento de Adam Smith.

El origen de las farmacias viene dado no sólo por dicha necesidad, sino también por los medicamentos y profesionales que han desarrollado a éstos para combatir la enfermedad. Y, de hecho, se puede considerar que gracias a la medicina el ser humano ha sido capaz de crear una civilización en la cual resulta fácil habitar. Por naturaleza el ser humano no está hecho para soportar el sistema moderno y civilizado en el que habita, pero gracias a la medicina este ambiente resulta habitable. (Sagrera, 2005)

Farmacia Arcaica

La base de una farmacia es su medicamento, por lo tanto, las primeras farmacias ofertaban medicamentos vegetales de acuerdo al entorno que les rodeaba, recolectaban aquellas plantas consideradas medicinales para su uso. Con el tiempo la necesidad de una farmacia por disponer de medicina que no se encontraba en su zona tuvo su desarrollo con el trueque y el transporte de medicina a diferentes zonas, esto permitió que la civilización pueda disponer de todos los medicamentos básicos. (Sagrera, 2005)

Tecnología y Farmacia

La tecnología abre nuevas posibilidades en la medicina, ya que en farmacias arcaicas solo se trataba de una mezcla de elementos medicinales provenientes de recursos vegetales, animales y minerales. Este sistema de elaboración de medicamentos era tan simple que sus únicas vías de administración eran oral y dérmica. Con el avance tecnológico se innova en métodos de extracción de los recursos de manera más precisa aumentando en la producción de nuevos medicamentos modernos. (Sagrera, 2005)

Comercio y Farmacia

Las farmacias crecen a medida de la economía del país, existen países capaces de soportar la inversión monetaria de investigación de medicamentos, mientras que existen países que sólo podrán consumirlos, e incluso, otros países pobres que ni siquiera podrán consumir éstos debido a no disponer del capital necesario para obtenerlos.

Es la razón por la cual la malaria tardó tanto tiempo en obtener un medicamento que pueda combatirlo, puesto que los países desarrollados no se preocupaban por una enfermedad que no afecta su región y mucho menos por un país que no podrá adquirir y recuperar la inversión de estudio que se necesita para la elaboración efectiva de un medicamento.

De esta manera se llega a la conclusión de que el comercio de las farmacéuticas varía mucho por la gran inversión que tiene la investigación de medicamentos, y para países subdesarrollados sólo se comercializan para su consumo. (Sagrera, 2005)

Farmacias del Ecuador

En la actualidad del Ecuador se puede encontrar una gran variedad de farmacias, mientras algunas son franquicias, otras trabajan a nivel parroquial por su disponibilidad de distribuidores. A continuación, se mencionan algunas de las franquicias farmacéuticas con mayor presencia en el Ecuador:

- Farmacias Cruz Azul
- Farmacias Económicas

- Farmacias FarmaRED'S
- Farmacias Fybeca
- Farmacias Medicity
- Farmacias Nueva Selecta
- Farmacias Sana
- Farmacias Sumedica
- Pharmacys

Metodología de Desarrollo de Aplicaciones

En el flujo de desarrollo de tecnologías es necesario la inclusión de estrategias y metodologías de trabajo que permitan mejorar tanto el proceso como el tiempo de entrega del proyecto de software. Es por lo que se trata de una parte fundamental para el desarrollo de cualquier producto o proyecto de software.

Metodología Scrum

Antecedentes

Anteriormente era muy utilizada la metodología en cascada para el desarrollo de aplicaciones de software, pero esta metodología de trabajo fue quedando obsoleta debido a la necesidad actual de un desarrollo mucho más acelerado. Desde 1990 aparece la metodología SCRUM que fue obteniendo popularidad con el paso del tiempo por su gran capacidad de poder entregar proyectos de software en un periodo de tiempo más reducido. (Dimes, 2015)

Definición

Más allá de una metodología, se trata de un marco de referencia que utiliza la metodología de desarrollo de software Agile. Cómo utilizar dicha metodología está en la capacidad de ofrecer un producto de calidad de manera acelerada, pero una de sus bases fundamentales es el uso de una “pila del producto”, que busca cumplir con aquellas funcionalidades que ofrecen un producto de calidad priorizando su desarrollo sobre todo lo demás que se ha definido previamente. (Dimes, 2015)

Desarrollo de Software Agile

Es una metodología con un enfoque humano en donde se reduce el uso de documentación innecesaria y se basa en el comportamiento y sugerencias del cliente final para obtener un producto más preciso y adaptado a necesidades reales. Este es el factor diferenciador puesto que las metodologías tradicionales solo buscan generar un estudio previo e ignorar opiniones en el ciclo de vida de un proyecto. (Dimes, 2015)

Perfiles de la Metodología Scrum

Como es una metodología de trabajo se debe definir el equipo que se encargará del desarrollo de la aplicación de software, para lo cual se definen cargos en el flujo de trabajo.

Product Owner

Es la persona encargada de definir los objetivos del proyecto en base a las opiniones de los clientes finales, y con estas bases es capaz de definir de manera clara y concisa los objetivos que debe cumplir el equipo de desarrollo.

Scrum Master

Es el líder del equipo de trabajo, esto se debe a que se encarga de mantener al equipo motivado y dispuesto para cumplir con los objetivos que se les ha planteado. Es la base motivacional del proyecto a la vez que guía en caso de presentar dificultades en el mismo.

Scrum Team

Es el equipo que se encarga del desarrollo y entrega del producto final, es una estructura horizontal que se auto-organiza con el propósito de poder conseguir definir puestos y tareas de acuerdo a las habilidades de los integrantes del equipo.

Stakeholder

Son personas externas al equipo y que pueden intervenir de manera indirecta al ser perfiles interesados en el producto como el mismo cliente que solicita el desarrollo del mismo o el gerente de una compañía.

Etapas de la Metodología Scrum

Las etapas de la metodología Scrum mantienen un sistema de inicio y un proceso cíclico que se genera una vez comienza la fase de desarrollo.

Etapas 1: Product Backlog

El proceso comienza con la elaboración del llamado Product Backlog. Se trata de un archivo genérico que recoge el conjunto de tareas, los requerimientos y las funcionalidades requeridas por el proyecto. Cualquier miembro del equipo puede modificar este documento, pero el único con autoridad para agregar prioridades es el Product Owner, responsable del documento.

El autor Mike Cohn, experto en compañías software, utilizó el acrónimo «DEEP» para referirse a las diferentes etapas o fases de un buen Product Backlog. Corresponden a las iniciales de:

Detailed Appropriately: en esta fase se definen los requisitos del producto con las características que afectan al desarrollo del producto.

Emergent: esta parte define el Product Backlog como algo que no para de evolucionar y cambiar, ya que siempre se adaptará a las demandas del cliente y por ende, a las decisiones del Product Owner.

Estimated: esta parte quiere decir «estimado», ya que se refiere al valor aproximado según el esfuerzo y el valor que se le dé al proyecto.

Prioritized: esta parte define el hecho de que, en el Product Backlog todos los elementos deben estar priorizados y repartidos en categorías.

Etapas 2: Sprint Backlog

Es un documento que recoge las tareas a realizar y quién las desempeña. Es interesante asignar las horas de trabajo que va a suponer realizar cada una de ellas y asignarlas a un coste. Si su volumen es muy grande, crear metas intermedias será un acierto.

El Sprint es el periodo en el que se realizan todas las acciones pactadas en el Sprint Backlog , que supone entregas parciales para ir testeando el producto final.

El ciclo anterior deberá repetirse hasta que todos los elementos del Blacklog hayan sido entregados. Entre los distintos Sprints no se deben dejar tiempos sin productividad.

Etapa 3: Sprint Review

Todas las acciones que realicemos han de tener un control. Es en el Burn Down donde marcamos el estado y la evolución del mismo indicando las tareas y requerimientos pendientes de ser retratados.

En esta fase final del Sprint, se revisa todo el trabajo, una buena oportunidad para tener feedback sobre el desarrollo del producto.

Puede ser una reunión informal, siempre y cuando se tenga el objetivo claro del Sprint Review: brindar transparencia tanto al equipo como al cliente.

Suele durar unas cuatro horas para Sprints de cuatro semanas. Es decir, una hora por cada semana. La persona encargada de realizar este trabajo es el Product Owner, mientras que el Scrum Master se asegura de que esto sea así y de que se cumplan los tiempos establecidos.

Entonces ¿qué ocurre en el Sprint Review? Dentro de esta fase, en la que ya sabemos que el Product Owner es el encargado total de la revisión, se encarga de explicar los items del Product Backlog y asegurarse de que hayan sido finalizados.

A su vez, el equipo de desarrollo se encarga de hacer la demostración del incremento terminado durante el Sprint y responde a dudas relacionadas con ello.

A partir de ahí se hace una review del proyecto y se tratan los siguientes pasos a seguir. Si hay una respuesta por parte del cliente, el Product Owner reorganizaría el Product Backlog.

Por último, se hace una última revisión sobre tiempos, presupuesto y alcance del proyecto.

Etapa 4: Sprint Planning Meeting

¿Quién no ha perdido horas de trabajo inútiles en reuniones poco productivas porque estaban mal preparadas? Esto no tiene cabida en los métodos ágiles. Cada minuto cuesta dinero. Las reuniones han de estar también planificadas, como una parte más del proceso. En este «Sprint Planning Meeting» el Product Owner prioriza las tareas contenidas en el Product Backlog.

Con estas tareas en mente se determina el objetivo del nuevo sprint priorizando las tareas a realizar por el Scrum Team y asignando tiempo a cada una de ellas. El objetivo debe ser alcanzable y el equipo sólo abordará un conjunto de tareas asumible.

Diariamente se hace un seguimiento del proyecto en esta reunión en la que se controla el cumplimiento de las tareas asumidas. Quizás has oído hablar de la Daily Scrum, que es el nombre adoptado del inglés. En dicha cita se pactan los objetivos para el día siguiente y se analizan los posibles problemas que hayan limitado o impedido directamente el cumplimiento de los objetivos. (GALIANA, 2021)

Ciclo de vida de un proyecto de Software

Todos los productos llevan a cabo un proceso del cual se realiza un análisis para ver los resultados por lo que el desarrollo de un proyecto de software no se queda atrás se requiere un proceso por el cual debe pasar para que el producto pueda ser expuesto a los clientes para realizarlo existen diferentes modelos entre los cuales tenemos modelo en cascada, modelo en espiral, desarrollo por etapas, etc. Todos estos métodos tienen que satisfacer al cliente por lo tanto todos deben satisfacer los objetivos propuestos por los clientes finales (JAVIER MARTÍNEZ, 2013).

Recopilación de los Requerimientos

Para recopilar las necesidades del cliente se examinaron los problemas de la organización, los objetivos estratégicos del negocio y las principales características del sector

de la empresa y se estableció un diálogo para revisar los requerimientos de la empresa (JAVIER MARTÍNEZ, 2013).

Análisis

Recibir y comunicar las necesidades del cliente a la organización, recepción y clasificación de solicitudes, despacho de solicitudes. En esta parte se establecieron protocolos de comunicación entre los analistas y los usuarios finales (JAVIER MARTÍNEZ, 2013).

Limitaciones

Se comunica a los clientes que determinen un clímax de la aplicación para empezar a delimitar investigación y recursos de los que podamos disponer (JAVIER MARTÍNEZ, 2013).

Especificación

Representar la funcionalidad del sistema, así como también sus restricciones de desarrollo (JAVIER MARTÍNEZ, 2013).

Diseño y Arquitectura

Se adquiere los requisitos generales de la arquitectura de la aplicación para llevar a cabo construcción de un prototipo. El prototipo ayudará a que el cliente/usuario para definir los requisitos del proyecto (JAVIER MARTÍNEZ, 2013).

Programación

Producir componentes software que correspondan con la definición del producto y satisfagan los requerimientos. Se implementan las funcionalidades especificadas mediante el lenguaje de programación elegido. Se realizan las ayudas en línea y los manuales de usuario. Una vez el producto está diseñado se construye el sistema de interacción (JAVIER MARTÍNEZ, 2013).

Pruebas de software

Se prueban los componentes software basados en el plan de calidad, y se verifica que el software cumple con el requerimiento Se realiza el plan de pruebas, se especifican las

pruebas a desarrollar, se ejecutan y se evalúan. Se realiza el reporte de los resultados (JAVIER MARTÍNEZ, 2013).

Implementación

Se planea y ejecuta la implantación del software garantizando la puesta en marcha del sistema bajo el uso de una metodología que asegure el cumplimiento de los estándares de calidad de la empresa, las expectativas del cliente y la seguridad e integridad de la información. Se da asistencia a los usuarios (JAVIER MARTÍNEZ, 2013).

Mantenimiento

Se presenta el enfoque estructurado utilizado para proponer los mecanismos de control del plan de mejoramiento de los procesos de desarrollo de software en la empresa, que comprende el diseño de indicadores de eficiencia, eficacia y satisfacción de los clientes internos y externos de la empresa y se diseñan los mecanismos de control y seguimiento a los procesos (JAVIER MARTÍNEZ, 2013).

Diseño para arquitecturas para Aplicaciones

Microservicios

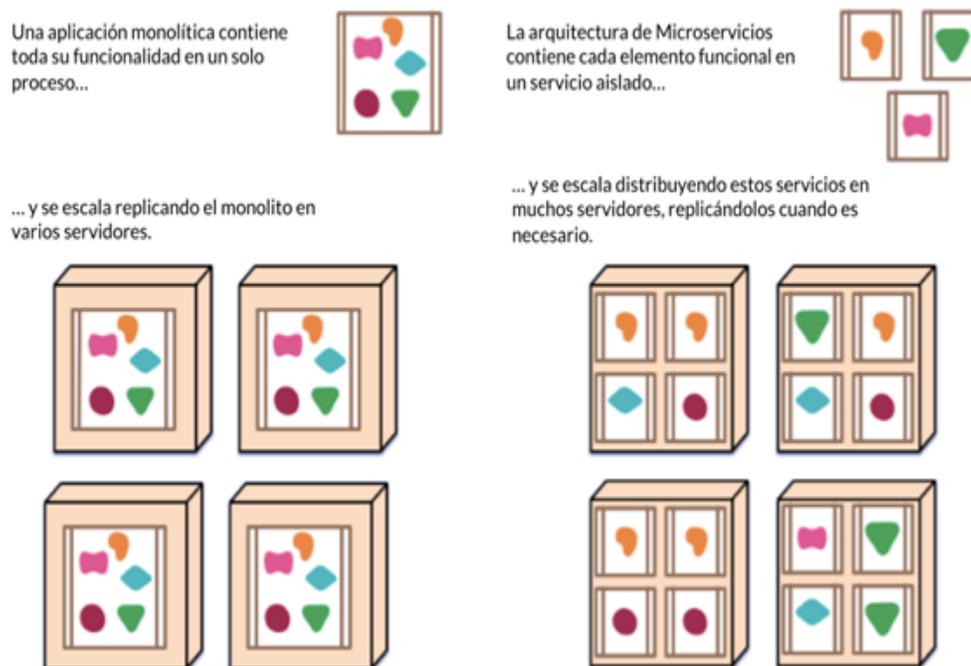
Es un nuevo método para desarrollar aplicaciones implementando servicios independientemente, se lo puede utilizar en diferentes lenguajes por lo tanto es fácil de interpretar. La gran ventaja es que si un servicio falla el otro sigue funcionando con normalidad, no es necesario modificar en toda la aplicación para editar un servicio. Trabajan con el protocolo HTTP (LEWIS JAMES, 2014).

Monolítico

Como su nombre lo dice es construido por una sola unidad integrada con la interfaz de usuario, la base de datos y la aplicación a lado del servidor (LEWIS JAMES, 2014).

Figura 4

Microservicios



Nota. Reconocer la diferencia que existe entre monolito y microservicio. Tomado es (LEWIS JAMES, 2014)

Devops

Qué es Devops

Es el desarrollo de las operaciones, permite que las funciones aisladas al momento de desarrollarse coordinen y mejoren. La ventaja es que mejora el rendimiento y mejora la calidad cumpliendo con los requerimientos del usuario. Automatiza los procesos con la mejor tecnología. Cumple con un ciclo de vida que se comprende de: Plan, Desarrollar, Entregar y funcionamiento que hace mejorar la calidad y rendimiento (azure, s. f.).

Docker

Es un contenedor que empaqueta el código para poder ejecutar la aplicación y es compatible con cualquier sistema operativo, es decir, es estándar. Se ejecuta en Docker engine siendo muy seguro con el manejo de nuestros datos y ligero (Docker, 2022).

Kubernetes

Es una plataforma open source que administra los contenedores, facilita la automatización, eliminando procesos manuales que congestionan nuestro contenedor principal. El impulsor de los kubernetes fue Google en el 2004 creciendo de una manera gradual. La característica de esta plataforma es que trabaja con contenedores, microservicios y la nube (kubernetes, 2020).

Git

Es un sistema que controla las versiones de los sistemas operativos, permitiendo trabajar de forma remota sencillamente. La característica es que es flexible al momento de usarla (mijacobs, 2022).

Gitlab

Es software libre en Gitlab Community Edition para empresas de control de versiones de código y desarrollo de Gitlab EE. Los creadores de este software son Dmitriy Zaporozhets y Valery Sizov en el 2011, Esta tecnología permite un control de todos los cambios realizados en el código base que realiza los cambios en un servidor centralizado proporcionando una interfaz para poder acceder (GitLab, s. f.).

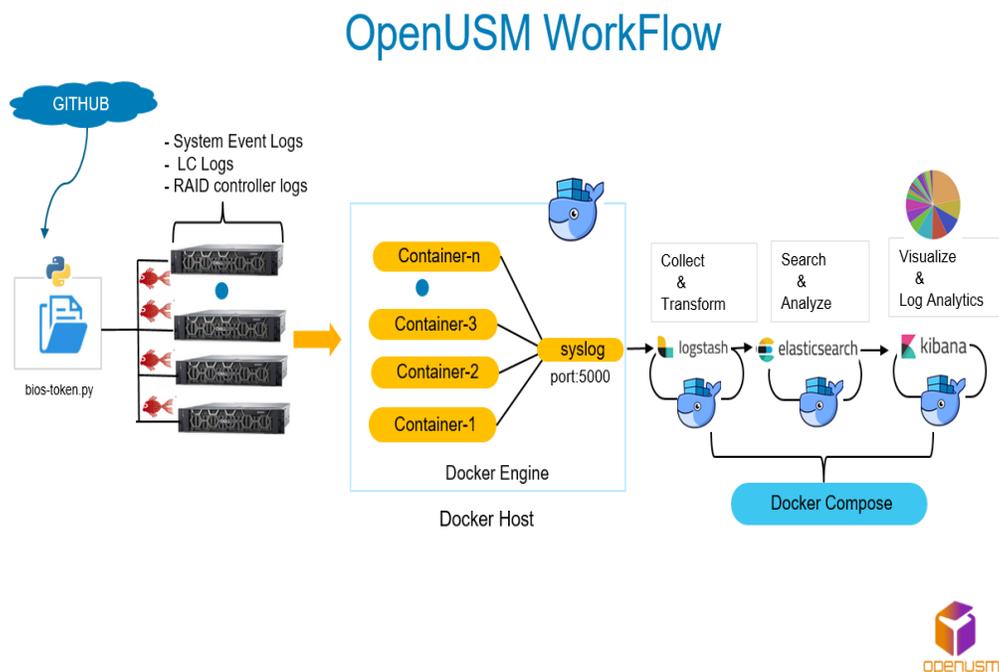
Motor de búsqueda

Es una herramienta para localizar información requerida por el usuario sobre cualquier tema, la información reflejada por el buscador es rastreada y mostrada identificada con una palabra que coincida. Un ejemplo claro son Google (Pastor, 2016).

Elasticsearch

Figura 5

Elasticsearch



Nota. El motor de búsqueda en este caso elasticsearch tiene un proceso con el que debe cumplir. Tomado de (*Getting Started with OpenUSM on Docker for Windows Platform – Collabnix, s. f.*)

Es un motor de búsqueda la gran ventaja es que es gratuito y abierto para el manejo de todos los datos. Se basa en Apache lucene quien fue presentado en el año 2010 Elasticsearch N.V. ahora conocido como Elastic. Contiene unas API REST simples con gran escalabilidad. Se usa por ejemplo en la búsqueda de aplicaciones, sitios web, monitoreo de contenedores, analítica de seguridad o negocios. El proceso que hace es ingesta los datos para poder parafrasear y comprenderlos. Elasticsearch almacena todos los datos como documentos JSON.El proceso de los datos es inmediato (elastic, s. f.-a).

Monstache

Es un Daemon en español conocido como demonio que analiza todas las peticiones de forma sincrónica. Monstache se puede usar con Elasticsearch para realizar búsquedas complejas y agregaciones de sus datos de MongoDB y crear fácilmente visualizaciones y tableros de Kibana en tiempo real(Ryan Wynn, 2016).

Kibana

Es una herramienta que nos permite visualizar y gestionar los datos para elasticsearch para reflejar una serie de datos estadísticos como mapas, gráficos en tiempo real. Se basa en la parte de frontend de las aplicaciones y la ventaja es que es gratuita y abierta. Fue creada por elasticsearch en el 2013. Un dashboard de Kibana es una recopilación de gráficos, búsquedas y mapas que se recopilaron en una hoja, un ejemplo claro es Kibana Lens o Canvas(elastic, s. f.-b).

Tiene las siguientes características:

- Kibana es la interfaz oficial de Elasticsearch
- Kibana aborda muchos casos de uso
- Kibana tiene un gran apoyo de la comunidad

Big Data

Para conocer cómo surgió este término debemos saber que la tecnología trajo grandes cambios a las empresas, ya que necesitaban un método para administrar sus datos es por ellos que a necesidad de poder gestionarla información nació el BD. Al principio eran pocos datos, pero después se necesitan manejar variedad de datos y en poco tiempo. El campo del Big Data cuenta con 3 partes importantes (Stratebi, 2014).

Fuente de datos que son conectores a BD NoSQL . Un ejemplo claro es de las empresas que están usando MongoDB para guardar datos no estructurados cual reto es obtener la información requerida por medio del consultor de BI, por ello, uno de los grandes retos para un consultor BI es poder obtener esa información (Stratebi, 2014).

Stage es reúna base de datos y pueda ser distribuida en el cual retorne los datos históricos y hagan agregaciones para difundirlos en otros nodos de manera distribuida (Stratebi, 2014).

DW final :El objetivo de las consultas de BI puedan analizar los datos el cual cree un ecosistema que nos permita crear un entorno que sigan creando nuevos proyectos para almacenar los datos y podamos generar un resultado como un Dashboards teniendo mayor velocidad el cual cree nodos que necesitemos para nuestro clúster (Stratebi, 2014).

BD es procesar una gran cantidad de datos utilizando herramientas o procesos tecnológicos que nos permita administrar los datos. Lo importante no es manejar los volúmenes de datos si no sacar la mejora potencial de ellos (GOYZUETA RIVERA, 2015).

Capítulo III

Desarrollo

Metodología SCRUM

Inicialmente se procedió a realizar una planificación para repartir las actividades a cada miembro, donde se organiza las fases a cumplir durante todo el proceso de implementación de la aplicación. Se realizan reuniones diarias para verificar los avances que se vayan dando. Todas las actividades tienen un tiempo estimado con una fecha de inicio y fin que se deben cumplir rigurosamente.

Tabla 3*Metodología Scrum*

Fase	Proceso	Responsable	Fecha inicio	Fecha fin	Duración	Status	Proceso
Planificación	Reunión para definir necesidades	Joel, Samira, Leo, gerente de la franquicia	10/05/2022	10/05/2022	1 día	100%	100%
	Identificar a los interesados	Leo	11/05/2022	13/05/2022	3 días	100%	100%
	Desarrollar el Acta de Constitución del proyecto	Samira	14/05/2022	17/05/2022	4 días	100%	100%
	Desarrollar el plan de Constitución del proyecto	Joel	18/05/2022	18/05/2022	1 día	100%	100%
	Planificar la Gestión del Alcance	Leo	19/05/2022	20/05/2022	2 días	100%	50%
	Estimar la duración de las actividades	Samira	21/05/2022	23/05/2022	3 días	100%	45%
	Planificación de Gestión	Leo	24/05/2022	29/05/2022	6 días	100%	42%

Fase	Proceso	Responsable	Fecha inicio	Fecha fin	Duracion	Status	Proceso
Ejecución	Crear	Samira	30/06/2022	01/06/2022	3 días	100%	30%
	Desarrollará cronograma	Joel	02/06/2022	05/06/2022	4 días	100%	29%
	Estimar costos	Joel	06/06/2022	07/06/2022	2 días	100%	25%
	Planificar gestión	Leo	08/06/2022	16/06/2022	9 días	100%	20%
	Dirigir proyecto	Samira	17/06/2022	18/06/2022	2 días	100%	13%
	Gestionar proyecto	Joel	19/06/2022	22/06/2022	4 días	100%	10%
	Adquirir equipo	Samira	23/06/2022	23/06/2022	1 día	100%	50%
	Desarrollar equipo	Joel	24/06/2022	25/06/2022	2 día	100%	10%
	Control de alcance	Leo	26/06/2022	30/06/2022	5 días	100%	0%
	Monitorear	Samira	01/07/2022	04/07/2022	4 días	100%	0%
Monitoreo	Control cronograma	Joel	05/07/2022	10/07/2022	6 día	100%	0%
	Control de costos	Leo	11/07/2022	12/07/2022	2 días	100%	0%
Cierre	Fin proyecto	Todos	13/07/2020	22/07/2022	10 días	100%	0%

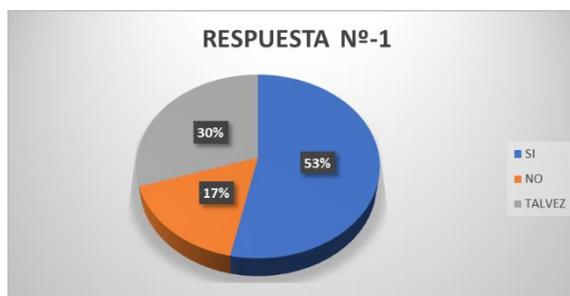
Análisis de Encuestas

Encuestas Personal de las Sucursales

Pregunta 1 ¿Conoce sobre aplicaciones web o móviles?

Figura 6

Encuesta pregunta 1

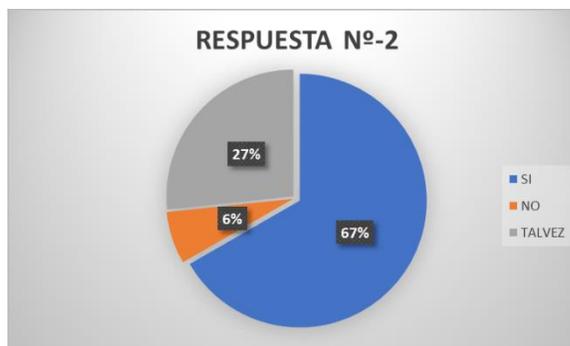


Nota. Mediante la encuesta realizada se obtuvo un 53% de la población tiene conocimiento sobre que es una aplicación web o móvil

Pregunta 2 ¿Considera usted que se deberían administrar los datos de las farmacias con aplicaciones web o móviles?

Figura 7

Encuesta pregunta 2



Nota. Mediante la encuesta realizada se obtuvo un 67% de la población considera que las aplicaciones web y móvil son una herramienta para administrar los datos de las farmacias

Pregunta 3. ¿Considera necesario que se implemente una aplicación para consultar los productos de la farmacia?

Figura 8

Encuesta pregunta 3

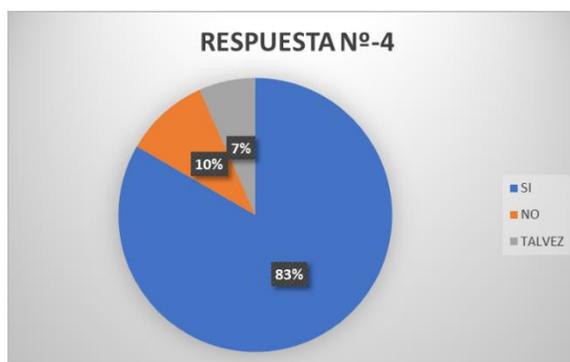


Nota. Mediante la encuesta realizada se obtuvo un 67% de la población considera que las aplicaciones web y móvil son una herramienta para administrar los datos de las farmacias.

Pregunta 4 ¿Considera usted que los clientes utilizarían la aplicación web o móvil de las farmacias Marthita?

Figura 9

Encuesta pregunta 4



Nota. Mediante la encuesta realizada se obtuvo un 83% de la población considera que las usarían las aplicaciones web y móvil

Pregunta 5 ¿Considera necesario que se le facilite un sistema de facturación para las farmacias?

Figura 10

Encuesta pregunta 5



Nota. Mediante la encuesta realizada se obtuvo que el 90% de la población considera que es necesario la implementación de un sistema de facturación

Pregunta 6 ¿Considera usted que la aplicación web y móvil facilita a los empleados realizar consultas sobre recetas médicas sin necesidad de visualizar el producto?

Figura 11

Encuesta pregunta 6

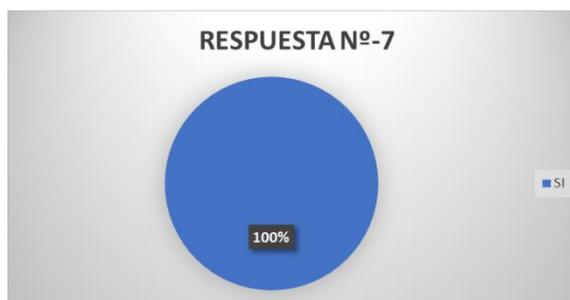


Nota. Mediante la encuesta realizada se obtuvo que el 40% de la población considera que si facilitaría la consulta de productos

Pregunta 7 ¿Considera usted que este proyecto de implementación de una aplicación web y móvil es una herramienta útil ?

Figura 12

Encuesta pregunta 7



Nota. Mediante la encuesta realizada se obtuvo que el 100% de la población considera que esta implementación es una herramienta útil

Pregunta 8 ¿Considera que con esta implementación aumentará las ventas de sus farmacias?

Figura 13

Encuesta pregunta 8



Nota. Mediante la encuesta realizada se obtuvo que el 97% de la población considera que si aumentarían las ventas de las farmacias

Pregunta 9 ¿Cree usted que las farmacias deben implementar tecnología para brindar mejores servicios de atención a los clientes?

Figura 14

Encuesta pregunta 9



Nota. Mediante la encuesta realizada se obtuvo que el 93% de la población considera que si se debería implementar tecnología

Requerimiento

Requerimientos de negocio

Inicialmente para poder definir los requerimientos de la empresa se programó una reunión donde estuvieron presentes los administradores de las sucursales , en este caso la administradora de la franquicia (Farmacias Marthita) quien expreso el propósito general que desea al implementar estas nuevas tecnologías a la empresa, con el antecedente que toda la administración de la empresa se lo hacía manual sin reflejar datos reales , nos comentó un poco de la historia de cómo empezó su microempresa y lo importante que era este nuevo paso en su negocio . En la reunión detallamos las necesidades generales de la franquicia, así como también las limitaciones que se tiene, también le comentamos las ideas que tenemos y como vamos hacer que su negocio crezca con la implementación de la aplicación.

Análisis del mercado

Se realizó un estudio comparando aplicaciones existentes que cumplen con las funciones requeridas, para así tener una perspectiva de cómo está el mercado de aplicaciones en base a su funcionalidad, con esto pudimos corroborar los servicios requeridos para poder

dar solución a los requerimientos que solicita el cliente, así como también el estudio de diversas herramientas digitales que proporcionaran la ayuda necesaria, entre las aplicaciones de mercado pudimos encontrar: Odo, Treinta.

Revisado el mercado se propuso desarrollar una arquitectura de software que ayude a satisfacer las necesidades del cliente por lo cual nos enfocaremos en una arquitectura de microservicios.

Funcionalidades

El requerimiento principal es el desarrollo general de una aplicación que les facilite el manejo de información que se genera en la empresa para así poder realizar análisis estadísticos, así como también llevar un registro de ingresos y egresos que se tiene. Por otro lado, también deberá realizar un análisis de datos que ayudará en la toma de decisiones con forme a los datos recolectados, para esto se realizará la implementación de microservicios que ayude cada uno de estos aspectos

Por otra parte, se necesita que esta aplicación tenga alta disponibilidad que se refiere a que no exista ningún fallo, por ellos se deberá trabajar con un sistema de replicas que nos garantizara la disponibilidad de los diversos servicios que se requieren.

Los requerimientos faltantes se muestran en el Anexo D.

Análisis

Análisis de procesos de negocios

En este apartado se realizará un análisis de todos los procesos que se van a automatizar dentro de la empresa para lo cual se llevó a cabo una recopilación de dichos procesos entre los cuales están:

Se desea poder realizar un mejor manejo de ingresos y egresos dentro de la empresa ya que esta, no cuenta con un registro de datos efectivo para poder llevar un seguimiento de estos datos, por lo cual esta debe contar con un sistema automatizado que nos recopile toda la

información, así como también proteger los datos para que solo el personal encargado pueda ejercer cambio en esta información

El siguiente proceso que se desea automatizar es la devolución de productos próximos a expirar, en este proceso el producto a caducar debe ser puesto en cuarentena para poder realizar su respectiva devolución a los proveedores, para así no tener perdidas en la empresa, ya que este proceso se realizaba de manera manual revisando todos los productos, con el desarrollo de esta aplicación se automatizara este proceso dando a conocer a cliente los productos que ya están por caducar, por lo cual se realizara de manera automatizada.

El problema que conlleva tener varias sucursales es el no poder tener un control en estas ya que se debe ir de manera presencial para poder controlar el buen funcionamiento de las sucursales. Este proceso se lleva a cabo semanalmente por lo que no se tienen la certeza de cómo están sus demás franquicias. Se va a realizar un modelo de aplicación que va a permitir al propietario a llevar un control diario exacto de sus demás franquicias sin necesidad de ir presencialmente a los establecimientos

Desarrollar casos de uso

Tabla 4

Caso de estudio para usuario administrador

Servicios	Descripción
- Lectura de datos	Poder ver todo el contenido de la aplicación
- Escritura de datos	Puede agregar datos
- Actualización de datos	Puede Modificar datos
- Crear usuarios para supervisores y vendedores	Da el nivel que pueden tener sus empleados

Nota. Es un usuario creado al iniciar el sistema.

Tabla 5*Caso de estudio para usuario supervisor*

Servicios	Descripción
Registro de ingreso y egreso de productos	Ingresa Productos nuevos y también factura
Actualización de tabla de productos	Modifica productos
Registro de clientes	Puede registrar clientes nuevos
Actualización de tabla de clientes	Puede modificar la tabla de clientes
Recibir los códigos de factura y buscar en las facturas del día con detalle de productos.	Puede visualizar el campo de compra de productos

Nota. Es un usuario en cargo de realizar un control a nivel sucursal.

Tabla 6*Caso de estudio de usuario vendedor*

Servicios	Descripción
Registro de ventas (notificar venta al administrador)	Puede recibir un pedido, pero siempre reportando al administrador o supervisor
Registro de clientes	Puede registrar clientes nuevos
Actualización de tabla de clientes	Puede modificar la tabla de clientes
Facturación	Puede realizar facturaras a los clientes

Nota. Este usuario es que tiene menos rango y menos servicios a elegir

Diseño

Investigación de la Oferta

En base a los datos obtenidos con anterioridad, se procede a realizar una investigación que permita determinar cuáles son aquellas tecnologías que se ajustan a las necesidades de la franquicia y a las necesidades técnicas con las que se debe cumplir. Por lo que, a continuación, se enlistan las ofertas más relevantes para cada uno de los servicios requeridos.

Servidores Web

Se requiere de un servidor web en donde se aloje el frontend de la aplicación, el cual estará directamente conectado a los servicios del backend. Para cumplir con ello se presentan las siguientes ofertas en la.

Tabla 7

Comparativa de ofertas en servidores web

Oferta	Ventajas técnicas	Desventajas técnicas
Apache	<ul style="list-style-type: none"> • Software estable y confiable. • Código abierto y gratuito, incluso para uso comercial. • Parches de seguridad actualizados regularmente. 	<ul style="list-style-type: none"> • Problemas de rendimiento en alto tráfico. • Puede agendar vulnerabilidades de seguridad.
Django	<ul style="list-style-type: none"> • Ofrece un gran rendimiento y flexibilidad, pudiendo escalar proyectos de forma sencilla. • Incorpora una amplia variedad de paquetes de librerías (más de 4000). 	<ul style="list-style-type: none"> • Documentación puede ser confusa y dispersa. • Compleja para el desarrollo de una API REST o con sockets.
NodeJS	<ul style="list-style-type: none"> • Ideal para aplicaciones en tiempo real. • Cuenta con un administrador de paquetes de más de 50.000 recursos. • Permite un alto nivel de escalabilidad. 	<ul style="list-style-type: none"> • Falta de librerías estándar, no cuenta con una base inicial. • Pierde potencia de cálculo debido a sus eventos asíncronos.

Bases de Datos

Para la base de datos se debe considerar aquellas que sean capaces de almacenar grandes cantidades información debido a la expansión de la franquicia. Por lo que se decide considerar las ofertas de la Tabla 8.

Tabla 8

Comparativa de ofertas en bases de datos

Oferta	Ventajas técnicas	Desventajas técnicas
SQL Server	<ul style="list-style-type: none"> • Permite un escalado entres varias bases de datos SQL Server. • Administra de forma sencilla los datos para su consulta. 	<ul style="list-style-type: none"> • Excede en consumo de memoria RAM para instalaciones y utilización de software. • Aunque sea escalable, puede fallar si las consultas aumentan debido a su esquema relacional.
PostgreSQL	<ul style="list-style-type: none"> • Software estable y confiable. Código abierto y gratuito, incluso para uso comercial. 	<ul style="list-style-type: none"> • Al ser una base de datos relacional puede perder escalabilidad en proyectos de gran información • No cuenta con soporte oficial.
MongoDB	<ul style="list-style-type: none"> • Es un complemento ideal para trabajar en conjunto con Javascript. • Al ser no relacional, permite una gran cantidad información de escritura y lectura. 	<ul style="list-style-type: none"> • No se recomienda en aplicaciones de transacciones complejas. • Puede tener problemas de relación debido a su base de documentos.
Elasticsearch	<ul style="list-style-type: none"> • Contiene un motor de búsqueda integrado que auto genera índices para consultas de datos más rápidas. • Aunque no es una base de datos, trabaja como un lago de datos optimizado para lectura de información y una gran cantidad de información. 	<ul style="list-style-type: none"> • Si se utiliza individualmente puede causar problemas de rendimiento en un gran número de peticiones. • Puede ser complejo de manipular.

Frameworks o Librerías para Maquetado Frontend

La aplicación web debe cumplir con su apartado de interfaz gráfica que le permita al usuario la interacción con la misma. Se consideran las ofertas más relevantes del mercado en base a los servidores web definidos anteriormente.

Tabla 9

Comparativa de ofertas en librerías para maquetado frontend

Oferta	Ventajas técnicas	Desventajas técnicas
ExpressJS	<ul style="list-style-type: none"> • Ofrece ventajas en sus peticiones, agregando análisis y cabeceras automáticas. • Admite código de transformación de información. 	<ul style="list-style-type: none"> • Al ser un framework relativamente joven, presenta problemas en su desarrollo que va en crecimiento. • Documentación y comunidad poco extendida.
NestJS	<ul style="list-style-type: none"> • Sus módulos y controladores permiten una simple gestión de microservicios. • Resulta útil en el desarrollo de backend por su estructura. 	<ul style="list-style-type: none"> • Al estar más enfocada en backend, puede presentar problemas o complicaciones en su infraestructura. • Su sintáxis puede volverse muy extensa y confusa en un desarrollo frontend, lo cual dificultaría su implementación.
ReactJS	<ul style="list-style-type: none"> • Sintaxis simple que permite una simple lectura si se conoce HTML y Javascript. • Contiene un DOM virtual capaz de moldearse a diferentes tipos de lenguajes de representación visual. 	<ul style="list-style-type: none"> • Al no contar con un estándar puede volverse confuso en la cantidad de soluciones de las que dispone. • Requiere tiempo para dominar de forma eficaz las manipulaciones del DOM.
Angular	<ul style="list-style-type: none"> • Documentación detallada que facilita la información de su uso. • Permite compartir datos en una misma sección. 	<ul style="list-style-type: none"> • Está basado en Typescript, por lo que su sintaxis es más compleja. • Tiene problemas en migración de anteriores versiones.

Aplicaciones Móviles Multiplataforma

El proyecto debe cumplir con el soporte multiplataforma, y esto se debe considerar particularmente para la versión móvil. Las ofertas tecnológicas existentes se consideran de acuerdo a la complejidad de la aplicación.

Tabla 10

Comparativa de ofertas en aplicaciones multiplataforma

Oferta	Ventajas técnicas	Desventajas técnicas
Aplicaciones híbridas	<ul style="list-style-type: none"> • Permite maquetar una vez para implementarla en varias plataformas. • En mantenimiento solo se debe realizarlo a una fuente de código. 	<ul style="list-style-type: none"> • Para ofrecer una buena experiencia de usuario puede volverse complejo por sus limitaciones no nativas. • Son más lentas que una aplicación nativa, lo cual limita el tipo de aplicación que se puede montar.
Aplicaciones web progresivas	<ul style="list-style-type: none"> • A pesar de tener como motor el navegador, son capaces de mantener ejecuciones en segundo plano y notificaciones. • Puede ser multiplataforma, incluso más allá que solo celulares. 	<ul style="list-style-type: none"> • Es una tecnología en desarrollo que puede tener errores. • No puede tener acceso a las funciones del celular como contactos o cámara.

Servicios en la Nube

En el caso del alojamiento de la aplicación debe ser capaz de cumplir con la futura demanda que solicita la franquicia. Para poder cumplir con ello se define el uso de un servicio en la nube con las ofertas.

Tabla 11

Comparativa de ofertas en servicios en la nube

Oferta	Ventajas técnicas	Desventajas técnicas
Google Cloud	<ul style="list-style-type: none"> • Precios más competitivos en comparación con la competencia. • Se enfoca en expansión y escalabilidad. 	<ul style="list-style-type: none"> • No cuenta con funciones para la automatización de CI/CD u otras necesidades. • Fuera del nivel gratuito, todo su servicio es pago.
Amazon Web Services	<ul style="list-style-type: none"> • Es muy útil para un equipo de DevOps, implementando maneras sencillas para trabajar con CI/CD. • Dispone de una amplia gama de funcionalidades en configuración, escalabilidad, instalación, etc. 	<ul style="list-style-type: none"> • Su proceso de implementación es muy extenso. • No tiene un gestor que facilite la implementación de una aplicación.

Servidor Web para Implementación de APIs

Las interfaces de programación de aplicaciones requieren de un servidor web en donde poder ejecutar su escucha activa. Las ofertas disponibles varían y se consideran aquellos más recomendados por la comunidad de desarrolladores de API.

Tabla 12

Comparativa de ofertas en de servidores web

Oferta	Ventajas técnicas	Desventajas técnicas
Flask	<ul style="list-style-type: none"> • Es un servidor web muy utilizado como API por su fácil implementación. • Al ser un servidor web basado en Python, permite extenderlo con librerías de seguridad, conexión, etc. 	<ul style="list-style-type: none"> • No cuenta con soporte asíncrono nativo, por lo que puede presentar problemas en consultas de larga duración. • Falta de validación de datos por defecto.
ASP.NET	<ul style="list-style-type: none"> • Tiene comunicación estable con SQL Server. • Cuenta con soporte continuo y es un sistema estable. 	<ul style="list-style-type: none"> • Limita las conexiones a otras bases de datos. • No es multiplataforma. • Mayor tiempo de desarrollo.

Sistema de Integración Continua y Despliegue Continuo

La implementación y depuración de errores que se pueden presentar a largo plazo, son requeridos contar con un sistema que permita desplegar el servicio en caso de correcciones o mejoras. De esa manera, la aplicación se adapta a la franquicia.

Tabla 13

Comparativa de ofertas en sistema de integracion continua y despliege continuo

Oferta	Ventajas técnicas	Desventajas técnicas
GitLab CI/CD	<ul style="list-style-type: none"> • Permite almacenar contenedores privados de manera gratuita. • Permite enlazar un pipeline a una rama para recargar la imagen en Kubernetes. 	<ul style="list-style-type: none"> • Contiene runners que dependiendo de dónde se ejecuten, presente problemas de conexión.

Oferta	Ventajas técnicas	Desventajas técnicas
Jenkins	Tiene fácil implementación. Se puede instalar en un servidor propio. Cuenta con más de 1000 complementos para facilitar el trabajo.	Su interfaz está desactualizada y tiende a ser más compleja de configurar. Difícil para realizar mantenimiento.

Representación Gráfica de Datos para su Análisis

La necesidad de poder presentar los datos de manera gráfica es uno de los requerimientos a cumplir. Utilizar la tecnología adecuada con menos tiempo de renderización sería lo ideal para esta sección.

Tabla 14

Comparativa de ofertas en representación gráfica de datos para su análisis

Oferta	Ventajas técnicas	Desventajas técnicas
Kibana	<ul style="list-style-type: none"> • Tiene su propia sección para aprovechar gráficos prefabricados. • Soporta la manipulación de Vega y Vega Lite. • Contiene un motor de machine learning. 	<ul style="list-style-type: none"> • Sólo se puede utilizar en conjunto con Elasticsearch, lo cual, limita si se requiere de otra fuente de datos. • Para realizar gráficos personalizados se debe tener conocimiento en Vega Lite.
PowerBI	<ul style="list-style-type: none"> • Sistema fácil de manipular. • Al ser un servicio de pago, tiene soporte inmediato de Microsoft. • Soporte con Vega Lite para representaciones gráficas. 	<ul style="list-style-type: none"> • Es un servicio de pago con altos costes. • No se puede compartir informes con otros usuarios.

Autenticación de Usuarios para API

Para proteger los datos privados y públicos, se requiere de un medio de autenticación para verificar que el cliente de la API es válido para recibir una respuesta.

Tabla 15

Comparativa de ofertas en Autenticación de usuarios

Oferta	Ventajas técnicas	Desventajas técnicas
JSON Web Token	<ul style="list-style-type: none"> • Modelo de acceso granular. • Permite una autenticación rápida para el cliente. • Es escalable porque no necesita validar en cliente y servidor. 	<ul style="list-style-type: none"> • Un token puede ser problemático si tiene muchos datos en la consulta. • Los tokens son extensos, lo que consume más transferencia de datos.
Cookies	<ul style="list-style-type: none"> • Se puede utilizar una misma sesión en todos los subdominios. • Reducen el uso de Javascript en el lado del cliente. • Requieren de poco almacenamiento. 	<ul style="list-style-type: none"> • Es propenso a ataques de falsificación de solicitudes. • Impide el escalado cuando se requiere de más consultas simultáneas.

Evaluación de la oferta

A continuación, se presentan todas las tecnologías seleccionadas de acuerdo a los datos más relevantes recopilados en el paso anterior.

NodeJS: Servidor Web

Su amplio repertorio de herramientas en el gestor de paquetes NPM, ofrecerá la posibilidad de implementar librerías que pueden ser de gran utilidad en el proceso de maquetado de la aplicación web. Además, que al estar basado en Javascript permite que se pueda manipular datos JSON de una forma más sencilla.

Su rendimiento es superior y demostrado por la cantidad de desarrolladores que lo utilizan para el desarrollo de aplicaciones web. Disminuye significativamente el margen de errores técnicos que puede presentar un servidor web.

MongoDB Atlas: Base de Datos

Se solicita que la base de datos sea capaz de almacenar una gran cantidad de datos, y para lograrlo, precisamente se aprovecha el uso de MongoDB, que trabaja como una base de

datos no relacional que contiene la información en documentos JSON. Lo cual resulta muy compatible con el servidor NodeJS trabajando a nivel de Javascript.

Para cumplir con la disponibilidad de la base de datos se aprovecha el uso de MongoDB Atlas, el cual es un servicio de MongoDB que ofrece la posibilidad de hospedar un cluster de MongoDB con soporte técnico por sus propios desarrolladores. Esto permite mantener la información siempre disponible sin problemas de caída.

Elasticsearch: Motor de Búsqueda y Base de Datos

Aunque se pueda considerar una base de datos, no se recomienda su uso para el almacenamiento masivo de información. Pero es el complemento ideal para MongoDB, puesto que mientras MongoDB registra información, Elasticsearch se encargará de indexar la información y tener un sistema de consulta más acelerado del que se obtendría solo con MongoDB.

Además de añadirle un potente motor de búsqueda, el cual no se queda atrás y permitirá realizar consultas inmediatas de un producto con sinónimos, entra otras consideraciones para una búsqueda eficiente.

ReactJS: Librería para Maquetado Frontend

Una de las razones principales para utilizar ReactJS es que se puede enrutar en diferentes endpoints sin perder componentes previamente renderizados, lo cual agiliza el proceso de carga en las aplicaciones y genera una sensación de que es una aplicación instalada en el dispositivo donde se encuentra.

La potencia de ReactJS se encuentra en el uso de su diseño modular, el cual permite aprovechar cada elemento creado y poderle adaptar a su versión móvil. Lo cual disminuye el código de maquetado.

Aplicaciones Web Progresivas: Aplicación web multiplataforma

Aunque su tecnología es relativamente nueva, resulta perfecta para el proyecto, puesto que la carga de trabajo no está en el frontend sino en el backend. Lo cual ayuda a generar

interfaces con el mismo código generado en ReactJS con la ventaja de poder crear su versión ejecutable con un motor de navegador de fondo en cualquier dispositivo capaz de ejecutar un navegador (Android, iOS, Windows, Linux y MacOS).

Google Cloud: Servicio en la Nube

Principalmente se utiliza este servicio por su precio competitivo de acuerdo al presupuesto disponible para el proyecto.

Aunque también se debe destacar que genera protocolos de seguridad capaces de proteger el clúster de Kubernetes, además de poder escalar el clúster según las necesidades que puede llegar a presentar la aplicación en el futuro.

Flask: Servidor Web para la Implementación de APIs

Se trata de una librería para Python que permite implementar un servidor web de manera simple, pero es muy utilizado como un servidor web para APIs debido a su sintaxis sencilla de manejar. El único inconveniente es que no tiene limitadores de información por defecto, por lo que se debe crear con código propio para poder almacenar los datos en una misma estructura.

A la par con la librería Pymongo, que se encarga de la conexión con la base de datos MongoDB y se puede gestionar toda la información y modificar los datos internamente. Lo que potencia enormemente su uso y permite desarrollar APIs vitaminadas que no sólo almacenan registran información, sino que también permite obtener datos adicionales que no son necesarios almacenar, pero sí presentar en GUI.

GitLab CI/CD: Sistema de Implementación Continua y Despliegue Continuo

Permite generar pipelines que son capaces de conectarse a un clúster de Kubernetes y realizar cambios en el mismo. Todo esto ofrece la posibilidad de que cuando se realice un “git push” en la rama principal se generen los cambios directamente en el clúster.

Además de que tiene un apartado para almacenar contenedores, lo cual facilita el trabajo y carga automáticamente la última versión de un contenedor sin tener que construirlo en una máquina local.

Kibana: Representación Gráfica de Datos para Análisis

Al tener una conexión estable y desarrollado específicamente para su uso con Elasticsearch, es perfecto para obtener datos estadísticos. Además de que se puede trabajar con Vega Lite, una de las librerías JavaScript más potentes para representación de gráficos. Existe Vega que es a bajo nivel (lo cual permite un renderizado de gráficos muy veloz) y Vega Lite que es una versión más ligera de desarrollar, pero se pierde parte de esa velocidad.

Uno de los puntos fuertes para seleccionar esta tecnología es su motor de machine learning, capaz de predecir datos de acuerdo a la información almacenada en Elasticsearch.

JSON Web Token (JWT): Autenticación de Usuarios para API

Este complemento ofrece la posibilidad de generar tokens, los cuales tienen un tiempo de caducidad y permiten la autenticación con la API. La ventaja principal es que no es necesario tener varios servidores para su autenticación, debido a que trabaja con una clave que se validará en el servidor y automáticamente devolverá los datos requeridos.

Arquitectura de servicios

Al trabajar con un clúster de Kubernetes, se puede escalar cada uno de los servicios de manera independiente, para lograr este objetivo se plantea la siguiente arquitectura.

Arquitectura Básica de la Aplicación

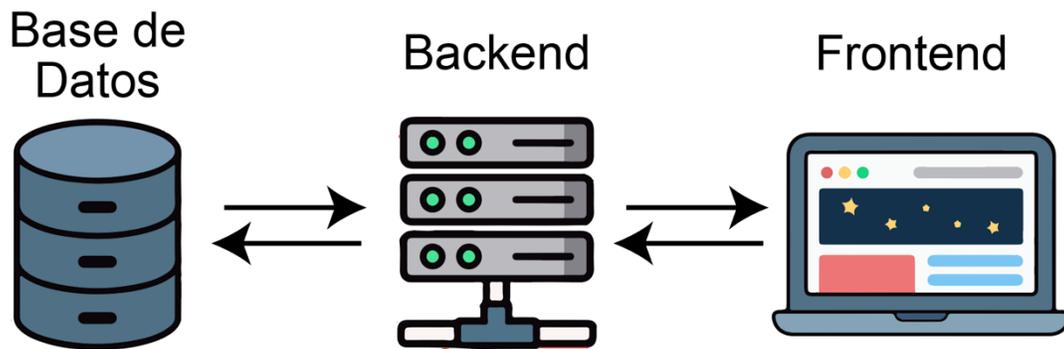
Se defina la arquitectura base para la comunicación entre cada uno de los servicios principales.

- Frontend: se encarga de renderizar la interfaz gráfica para interacción directa con el usuario.

- Backend: que está compuesto por los microservicios que se encargan de la lógica de toda la aplicación. De esa manera se evita sobrecargar al frontend con procesos que pueden costarle rendimiento.
- Base de datos: almacena toda la información para su posterior uso.

Figura 15

Arquitectura básica de una aplicación web



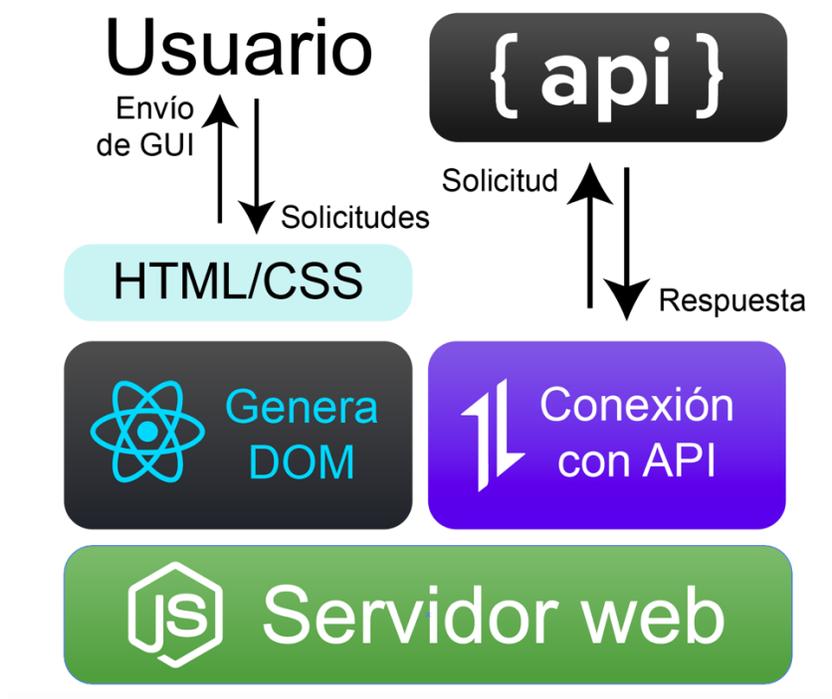
Arquitectura del Frontend

Al ser la sección encargada de interactuar directamente con el usuario, consta de su interfaz gráfica para conseguir este propósito. Y se compone por:

- NodeJS: servidor web
- ReactJS: lenguaje de maquetado
- Axios: cliente API REST

Figura 16

Comunicación de la aplicación web con el backend y el usuario



Arquitectura del Backend

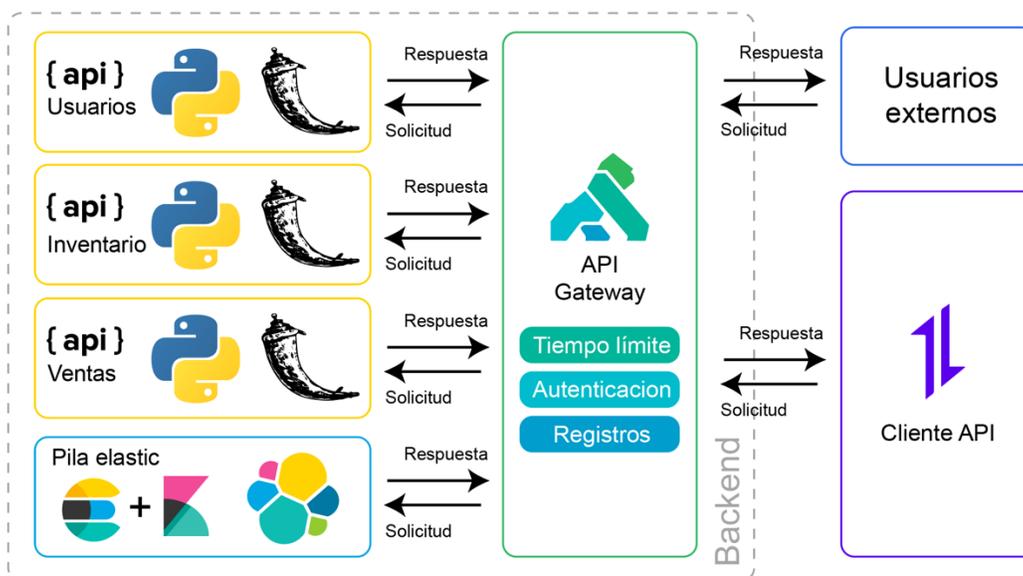
Los microservicios que se alojan en el clúster de Kubernetes se encargan de la lógica de programación de todo el sistema. Permiten registrar datos y obtener datos relevantes que no son necesarios almacenar. Consideran los siguientes elementos en esta arquitectura:

- Microservicios: desarrollados en Python con Flask.
- Elastic: pila compuesta por Elasticsearch, Monstache y Kibana.
- Kong: la API que trabajará como proxy para conectar a todas las APIs y servicios en una sola entrada principal de solicitudes.

Estos servicios al estar en Kubernetes, se puede realizar réplicas si el número de usuarios llegase a incrementar. Por lo que esto permite escalar la aplicación y mantenerla siempre disponible y sin perder rendimiento.

Figura 17

Comunicación de la API gateway con la aplicación web

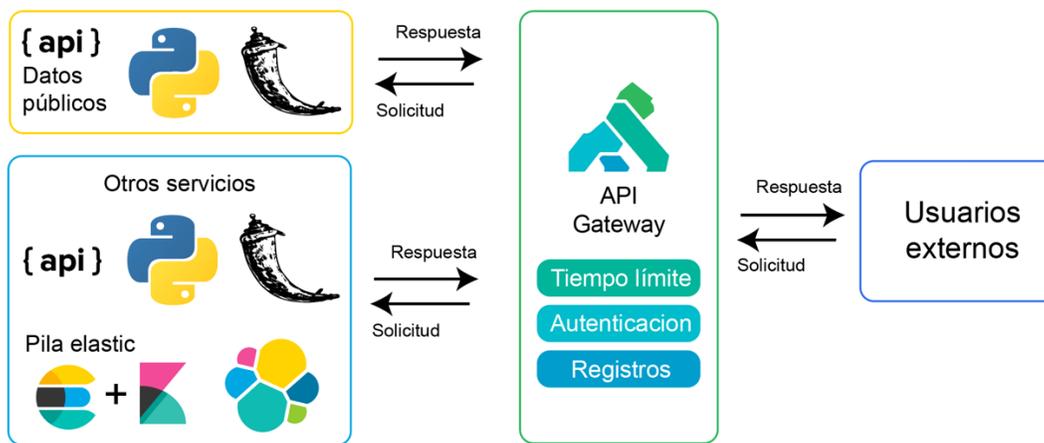


Arquitectura para Datos Públicos

Para el caso de los datos públicos se opta por desarrollar una API específica para ello, de esa manera se mantiene la seguridad que ofrece Kong a los demás servicios sin afectar a la arquitectura definida.

Figura 18

Comunicación de la API de los datos públicos con usuarios externos



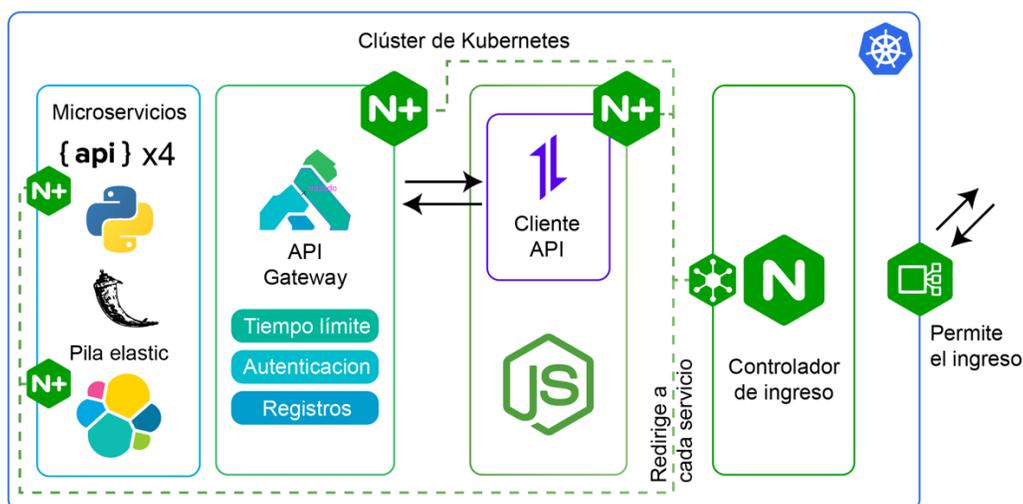
Los servicios privados tienen su propia configuración de seguridad dentro de Kong para que no se pueda comprometer la información sensible.

Arquitectura para Conexión de Clientes a Kubernetes

Finalmente se debe mencionar que el servicio debe tener un método de ingreso para los usuarios, para ello se aprovecha el uso de NGINX Ingress Controller que permite conectar al clúster con una única dirección IP, sin tener que crear varias direcciones IP, lo que aumentaría significativamente los costes en Google Cloud.

Figura 19

Esquema conceptual del funcionamiento NGINX Ingress Controller



Una de las mayores ventajas de usar este medio de conexión a parte de la dirección única es que es capaz de redirigir el tráfico de acuerdo al subdominio con el que se ingresa.

Ingeniería de procesos de negocio

Para poder almacenar la información en las franquicias “Farmacias Marthita”, se requiere tener una arquitectura de software basada en microservicios que ayude con el proceso, un servicio de base de datos es el encargado de recolectar información para ser utilizada.

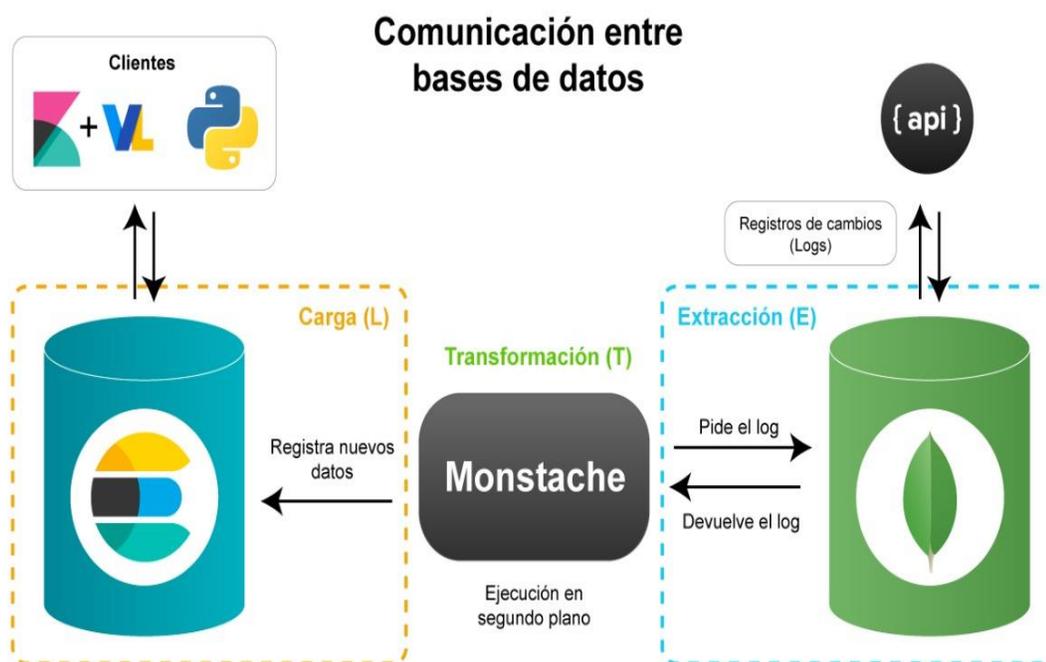
Se requiere utilizar un servicio que nos ayude con la visualización de datos, por ello se primero utiliza Elasticsearch encargado de guarda los datos indexados para tener lectura de

datos más eficiente y poder generar un motor de búsqueda optimizado para la consulta de información

Con los datos ya consultados se los representa de forma gráfica utilizando Kibana (interfaz gráfica de Elastic Search), así como también la representación estadística de información para lo cual se maneja Vega-Lite (librería de JavaScript) en cargada de mostrar graficas de datos.

Figura 20

Comunicación entre bases de datos



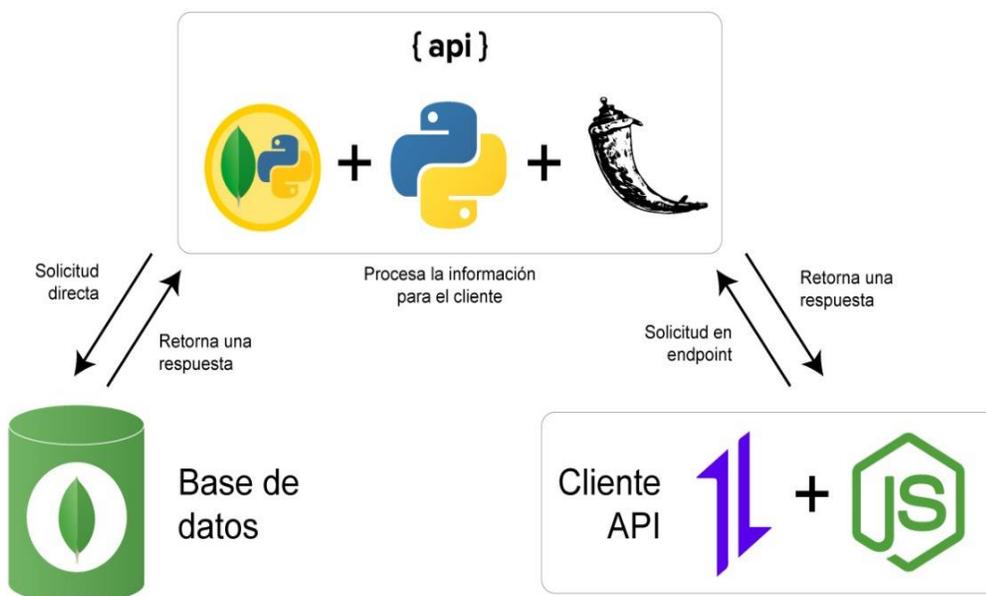
Nota. Diagrama ETL de Elasticsearch

Para solventar el proceso de devolución de productos, así como también llevar un seguimiento de los ingresos y egresos de las Farmacias, se requirió la construcción de API REST para cada proceso a solventar. Para la creación de las API usamos un servidor web basado en Python llamado Flask el cual permitió crear rutas de acceso para generar los

endpoints del API. Para tener una conexión entre la API y la base de datos se requirió de una librería que realice ese trabajo en este caso PyMongo (Librería de Python)

Figura 21

Esquema de comunicación de API REST



Nota. Requerimientos para la conexión de una API REST

Documentación tradicional

Se realiza la documentación de las APIs REST para facilitar su manejo en el cliente API que contiene la aplicación web y poder así manipular los datos de una manera rápida y eficaz almacenados en el clúster de MongoDB Atlas.

Figura 22

Documentación API

```

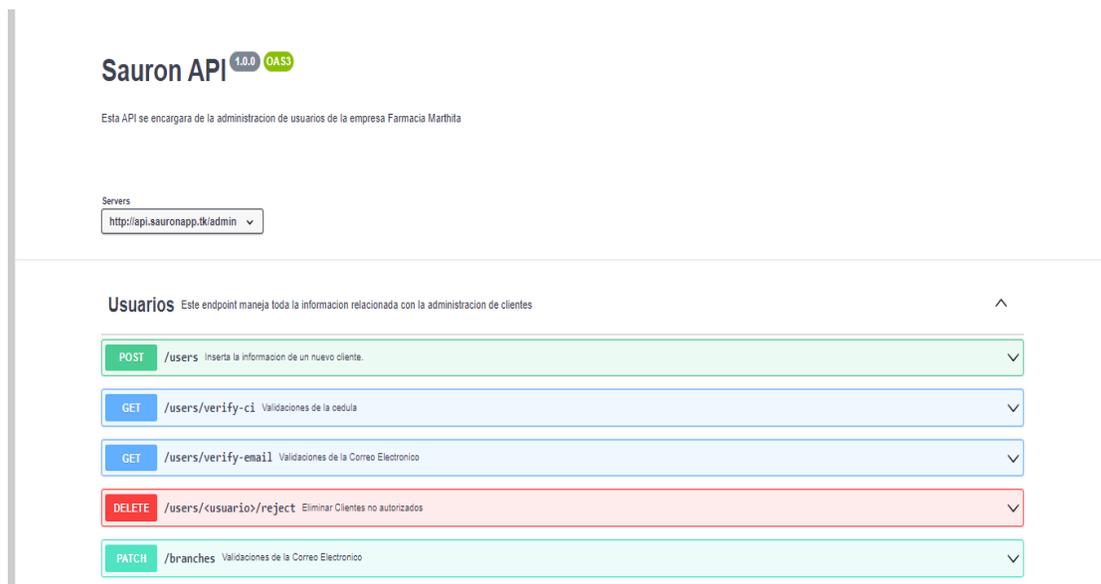
1 openapi: 3.0.1
2 info:
3   title: Sauron API
4   description: Esta API se encargara de la administracion de usuarios de la empresa Farmacia Marthita
5   version: 1.0.0
6 servers:
7   - url: http://api.sauronapp.tk/admin
8 tags:
9   - name: Usuarios
10  description: Este endpoint maneja toda la informacion relacionada con la administracion de clientes
11 paths:
12   /users:
13     post:
14       tags:
15         - Usuarios
16       summary: Inserta la informacion de un nuevo cliente.
17       parameters:
18         - nombre:
19           in: body
20           name: Nombre
21           description: Nombre del cliente
22           required: true
23           schema:
24             type: string
25       requestBody:
26         content:
27           application/json:
28             schema:
29               $ref: '#/component/schemas/BodyClientePost'
30       responses:
31         200:
32           description: (OK) la informacion del cliente se guardo
33           content:
34             application/json:
35               schema:
36                 $ref: '#/component/schemas/clienteExitoso'
37         400:
38           $ref: '#/component/response/Unauthorized'
39         401:
40           $ref: '#/component/response/NotFound'
41         404:
42           $ref: '#/component/response/Unauthorized'
43         500:
44           $ref: '#/component/response/ServerError'
45   /users/verify-ci:
46     get:
47       tags:

```

Nota. Documentación API mediante código de configuración YML

Figura 23

Interfaz de documentación de API



Nota. Para la documentación de las APIs se utilizó el programa Swagger Editor.

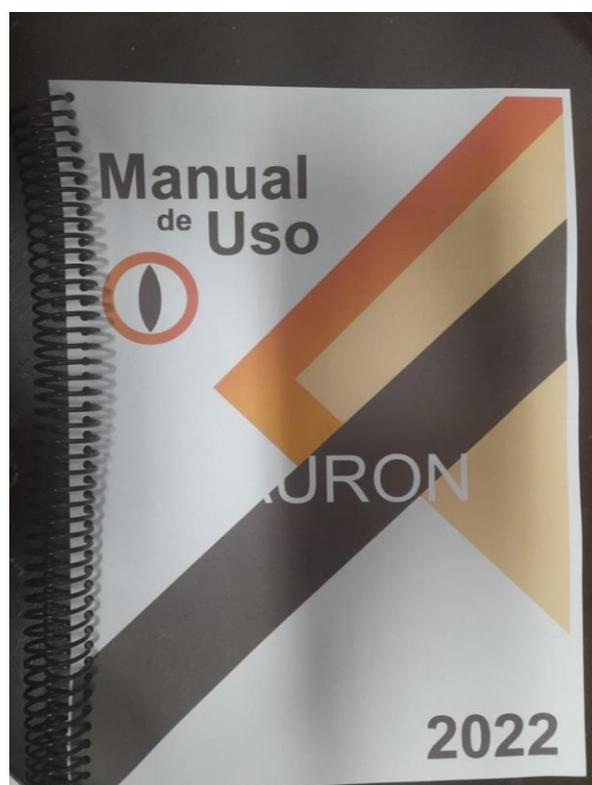
Para el caso de la API de datos públicos se puede revisar la documentación en el siguiente enlace: <https://api.sauron.app/docs/public>

El siguiente manual es un documento técnico que permitirá a los usuarios entender las funcionalidades de la aplicación móvil y web de Sauron, en el cual está detallada cada interfaz de manera interactiva.

El objetivo primordial de este manual fue crear una guía clara que permita inducir al usuario de forma rápida sin leer grandes textos y guiarse por las imágenes detalladas paso a paso en el ya mencionado manual.

Figura 24

Manual de Usuario



Nota. Se habla de la documentación de Apis su funcionamiento.

Diseñar casos de prueba

Tabla 16

Casos de prueba

ID Caso de prueba	Descripción de la prueba	Supuestos y condiciones previas	Datos de prueba	Pasos a ejecutar	Resultado Esperado
CP01	Validar la aplicación para ver su funcionamiento en Google Chrome	Instalar el navegador en nuestros dispositivos que vayamos a realizar las pruebas en primera instancia, implementar el servicio en la nube para poder realizar las pruebas correctivas teniendo conexión con internet.	Se debe ingresar al siguiente enlace para realizar la prueba https://sauron.app	Ingresar al URL Visualizar su renderización Verificar contenido	El tiempo de respuesta debe ser menor a 3s Presentar los componentes establecidos sin errores
CP02	Validar la aplicación para ver su funcionamiento en Edge	Instalar el navegador en nuestros dispositivos que vayamos a realizar las pruebas en primera instancia, implementar el servicio en la nube para poder realizar las pruebas correctivas teniendo conexión con internet.	Se debe ingresar al siguiente enlace para realizar la prueba https://sauron.app	Ingresar al URL Visualizar su renderización Verificar contenido	Tiempo de respuesta esperado 3s Componentes renderizados sin novedad
CP03	Validar la aplicación para ver su funcionamiento en Firefox	Instalar el navegador en nuestros dispositivos que vayamos a realizar las pruebas en primera instancia, implementar el servicio en la nube para poder realizar las pruebas correctivas teniendo conexión con internet.	Se debe ingresar al siguiente enlace para realizar la prueba https://sauron.app	Ingresar al URL Visualizar su renderización Verificar contenido	El tiempo de respuesta debe ser menor a 3s Presentar los componentes establecidos sin errores

ID Caso de prueba	Descripción de la prueba	Supuestos y condiciones previas	Datos de prueba	Pasos a ejecutar	Resultado Esperado
CP04	Validar página cómo funciona en el navegador de Dispositivos móviles	Verificar que el dispositivo que vayamos a utilizar cuente con el navegador instalado.	Se debe ingresar al siguiente enlace para realizar la prueba https://sauron.app	- Ingresar al URL - Visualizar su renderización - Verificar contenido	El tiempo de respuesta debe ser menor a 3s Presentar los componentes establecidos sin errores
CP05	Validar cómo funciona Tiempo de carga	Analizar la cantidad de usuarios a simular en segundo plano definir el tipo de pruebas en este caso de carga y rendimiento, Saber manejar la plataforma de Google Page Speed Insights para poder medir el tiempo de carga	Se debe ingresar al siguiente enlace para realizar la prueba https://pagespeed.web.dev/	Ingresar al URL Visualizar su renderización Verificar contenido	El tiempo de respuesta debe ser menor a 3s Presentar los componentes establecidos sin errores
CP06	Validación de la aplicación en un entorno móvil	Analizar la cantidad de usuarios a simular en segundo plano. Poder visualizar la adaptación en un entorno móvil	Se debe ingresar al siguiente enlace para realizar la prueba https://pagespeed.web.dev/	Ingresar al URL Visualizar su renderización Verificar contenido	El tiempo de respuesta debe ser menor a 3s Presentar los componentes establecidos sin errores

Nota. Se refleja casos en diferentes campos para el buen funcionamiento de la aplicación

Diseño de Interfaz de Usuario

En el diseño gráfico se definen dos prototipos base que permiten el maquetado de la aplicación.

Diseño gráfico conceptual y experiencia de usuario

Se elabora un bosquejo de la estructura básica y organización visual de cada uno de los elementos que contiene el proyecto, los cuales se pueden ver reflejados en la Figura 16.

Para este bosquejo se aprovecha el manejo de experiencia de usuario basado en la psicología humana, buscando una interfaz intuitiva que no le genere dificultad en su manipulación diaria al usuario objetivo.

Diseño gráfico digital

Estos diseños se llevaron a cabo con el uso de Adobe Illustrator que permite trabajar con elementos independientes y aporta cierta libertad en los aspectos técnicos del diseño.

Figura 25

Desarrollo de diseño de interfaz facturación con Adobe Illustrator.



Es fundamental establecer la interfaz de usuario digitalizada para tener una aproximación más acertada de cuál sería el maquetado final. Además de que ofrece la

posibilidad de poder utilizar medidas exactas como base para el posterior desarrollo del maquetado.

Figura 26

Interfaz de inicio de sesión



Los colores tomados para la aplicación se definen en base al logotipo que se diseña para la misma. Esto permite mantener un mismo ambiente en todo el concepto, aportando de un ambiente propio o una manera de identificar que el usuario se encuentra en el entorno de la aplicación.

Figura 27

Diseño de interfaz de ordenes de compras

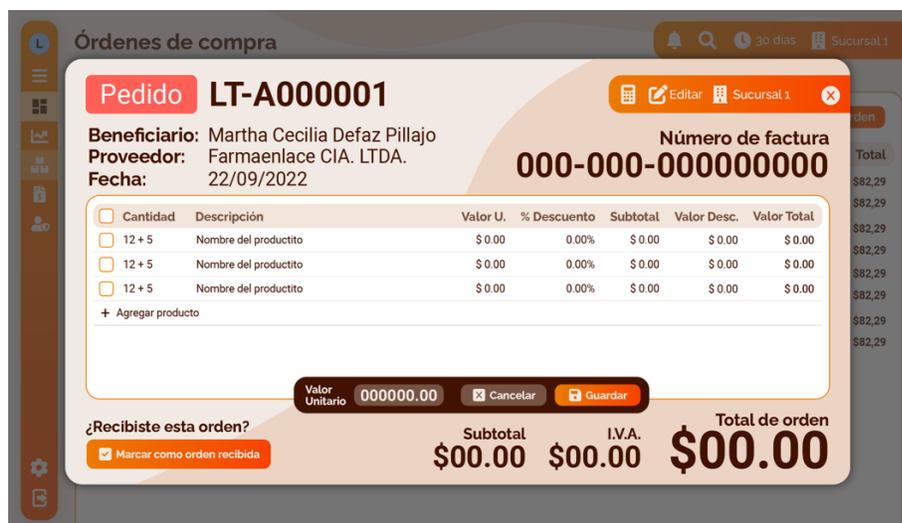


Figura 28

Diseño de interfaz móvil

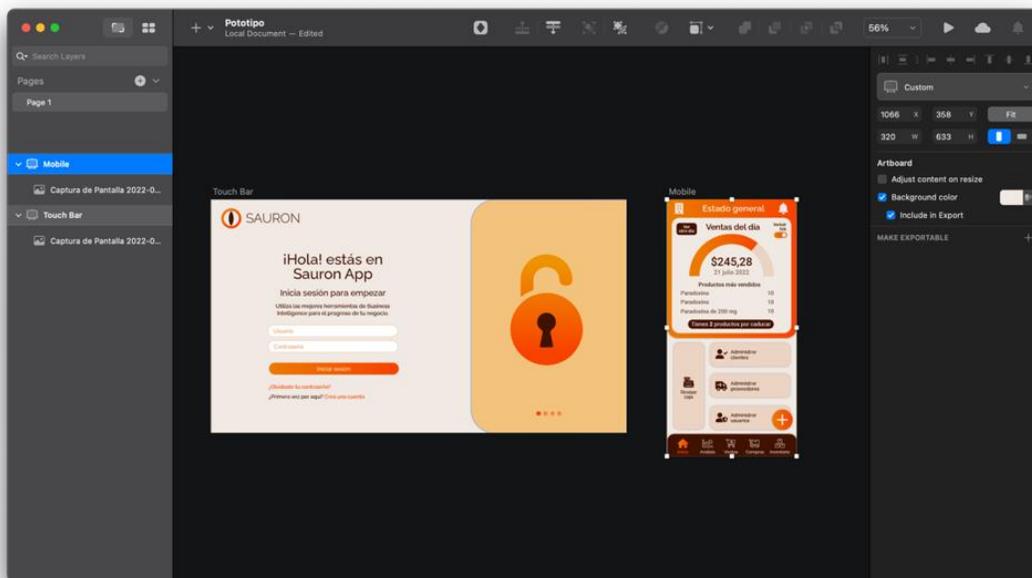


Maqueta

Para el desarrollo de una versión prototipada, se aprovecha el uso de Sketch como herramienta que permite "combinar" de cierto modo cada imagen para enlazarlas en algún punto específico de la aplicación y poder realizar pruebas de experiencia de usuario.

Figura 29

Desarrollo de diseños de interfaz con Sketch



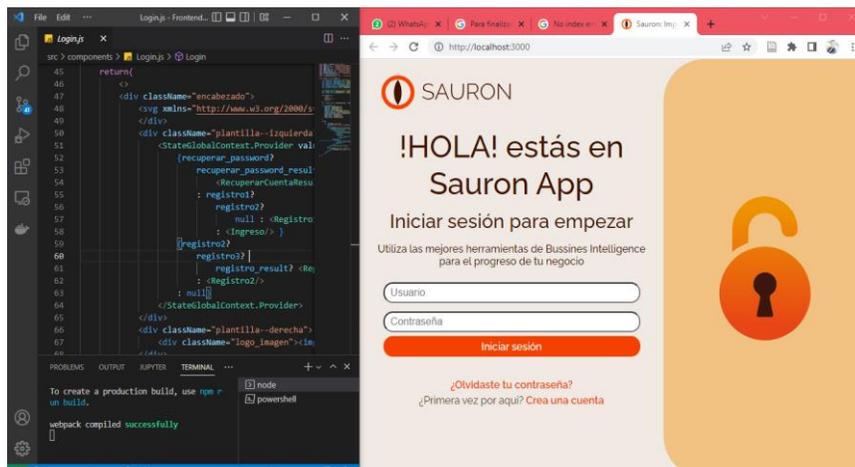
Implementación

Desarrollo de servicio de negocio

Se desarrolla la codificación de interfaces mediante lenguaje JavaScript y se utiliza un servidor de cliente NodeJS con su librería de ReactJS para la maquetación.

Figura 30

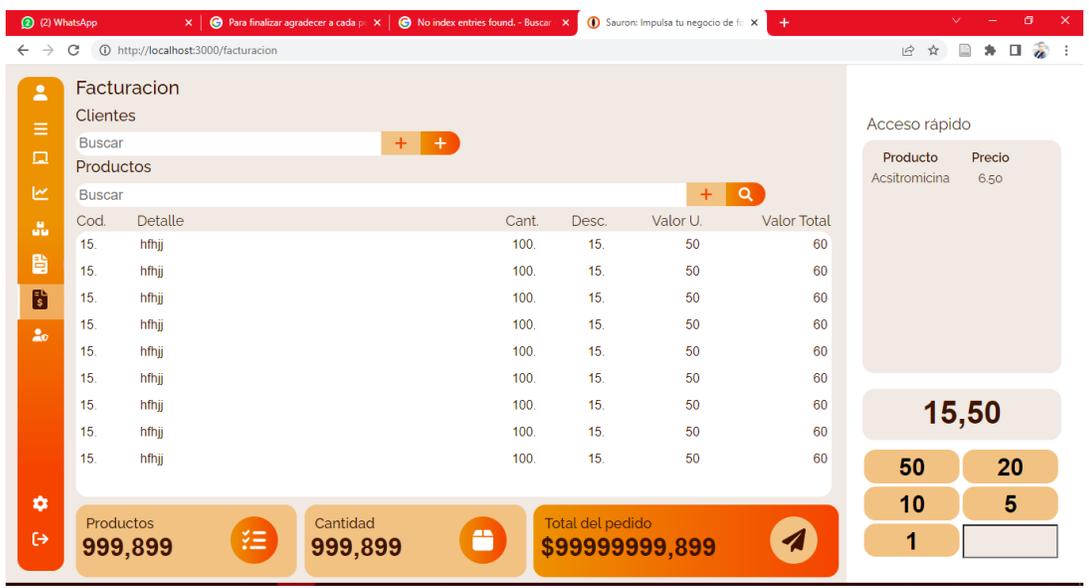
Maquetado de diseños Inicio de sesión



Nota. Codificación y representación mediante lenguaje de programación diseños

Figura 31

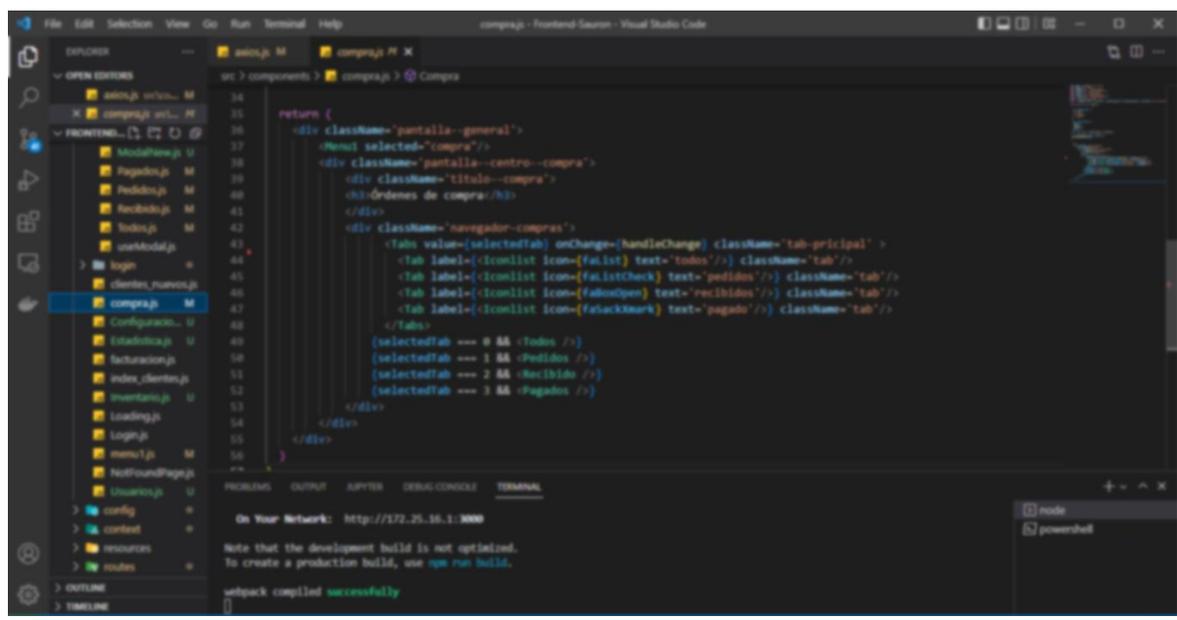
Maquetación de interfaz de Facturación



Nota. Codificación y representación mediante lenguaje de programación diseños

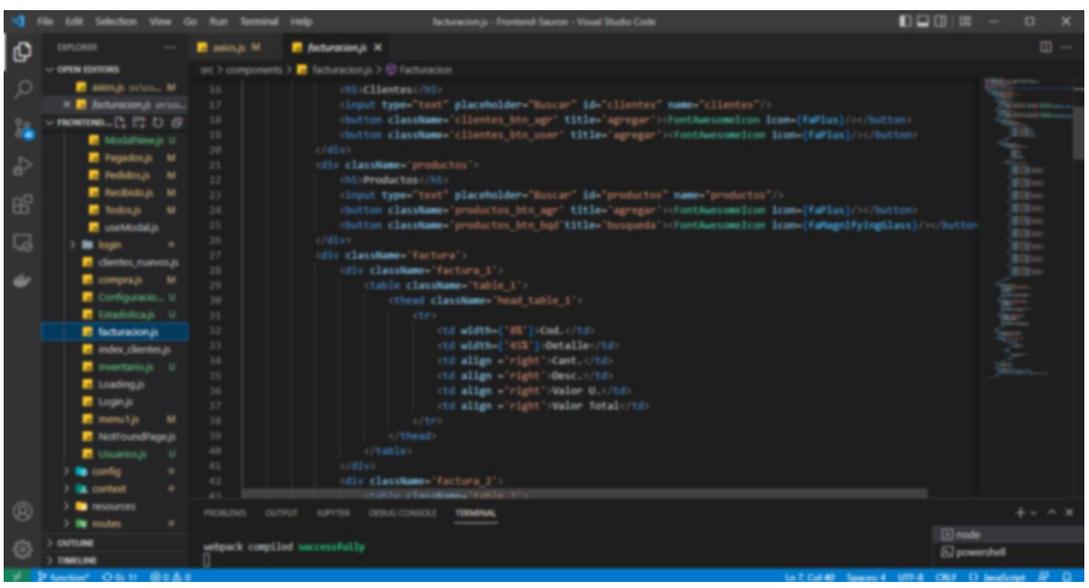
Figura 32

Codificación en JavaScript



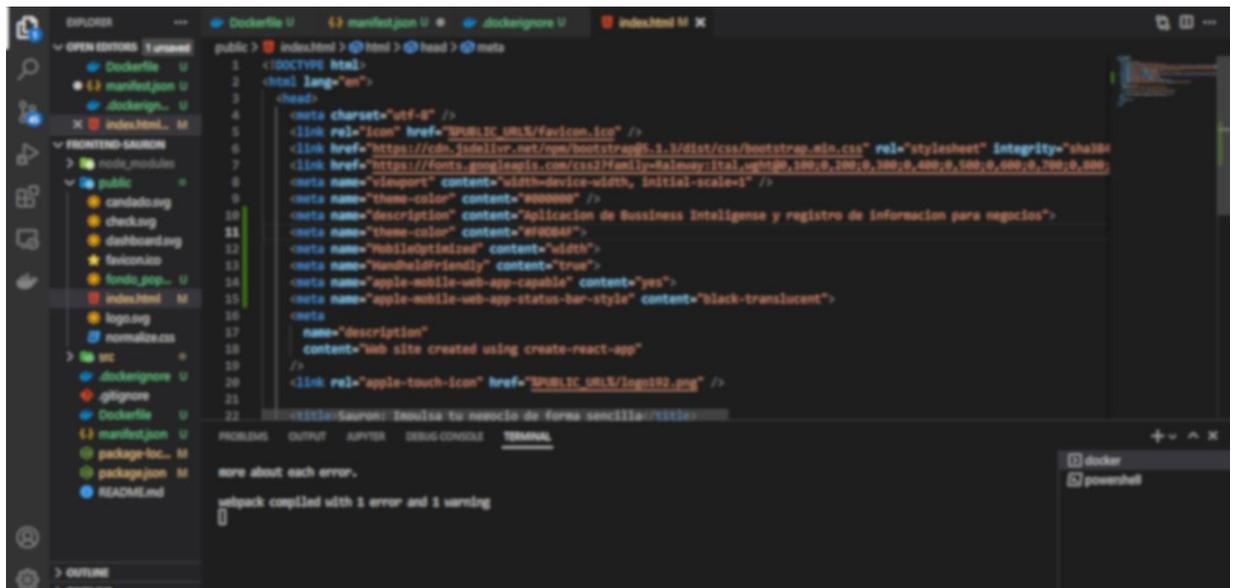
Nota. Codificación y representación mediante lenguaje de programación diseños

Figura 33
Maquetación mediante JSX



Nota. Codificación y representación mediante lenguaje de programación diseños

Figura 34
Aplicación Web Progresiva PWA



Nota. se desarrolla los comandos para convertir nuestra aplicación en una PWA

Figura 35

Archivo “manifest”

```

1  {
2    "name": "Sauron App",
3    "short_name": "Sauron",
4    "description": "Aplicación de Business Inteligencia y registro de información para negocios",
5    "background_color": "#000000",
6    "theme_color": "#000000",
7    "orientation": "portrait",
8    "display": "standalone",
9    "start_url": "../vite_source-web_app_manifest",
10   "scope": ".",
11   "lang": "es-ES",
12   "icons": [
13     {
14       "src": "../img/icon_192.png",
15       "sizes": "192x192",
16       "type": "image/png"
17     },
18     {
19       "src": "../img/icon_512.png",
20       "sizes": "512x512",
21       "type": "image/png"
22     }
23   ]
24 }

```

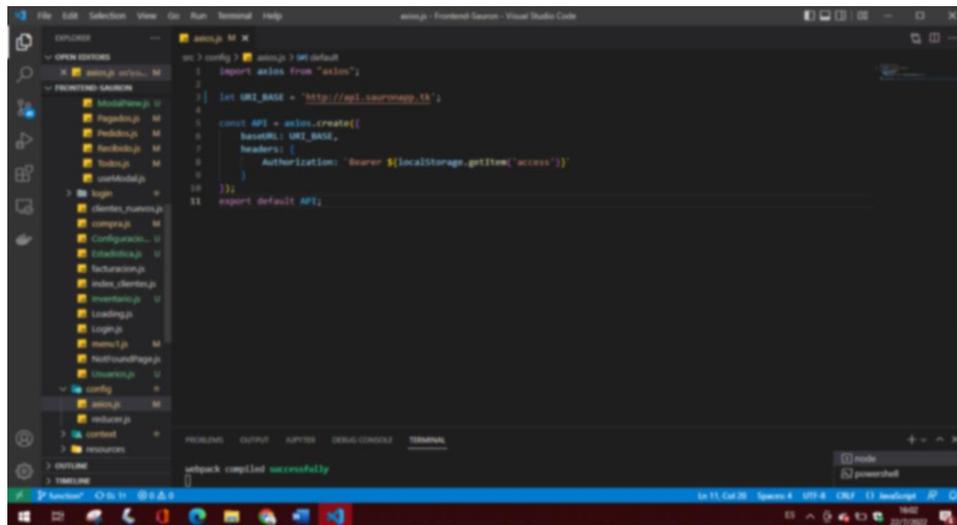
Nota. Codificación tipo JSON para poder convertir nuestra aplicación en una aplicación web progresiva

Integración con la plataforma

Se desarrolla las conexiones con los microservicios mediante la librería de NodeJS establecida para realizar peticiones a las APIs así poder tener la comunicación con el cliente, esta librería es Axios que es un cliente API REST.

Figura 36

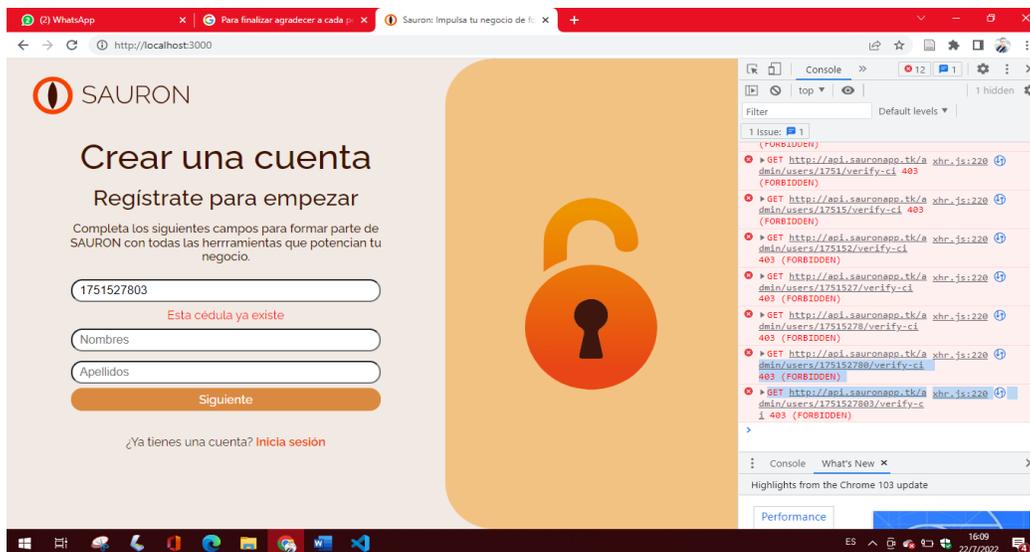
Conexiones a las APIs mediante Axios



Nota. Codificación de las conexiones a las APIs mediante Axios

Figura 37

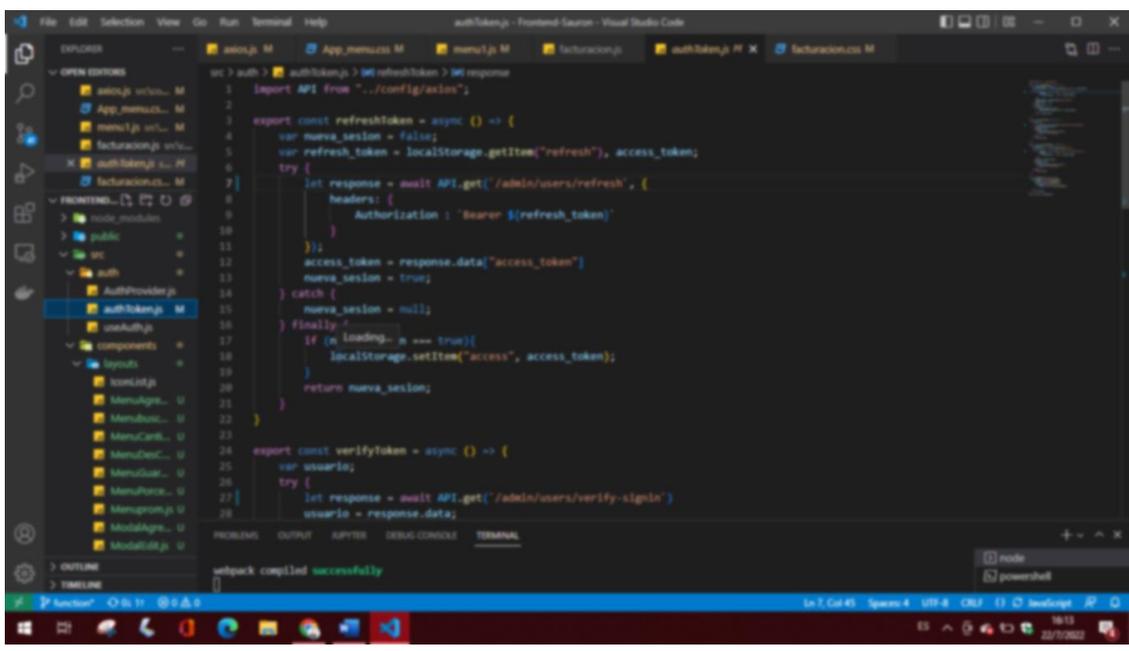
Validación de API



Nota. Codificación de las conexiones a las APIs mediante Axios

Figura 38

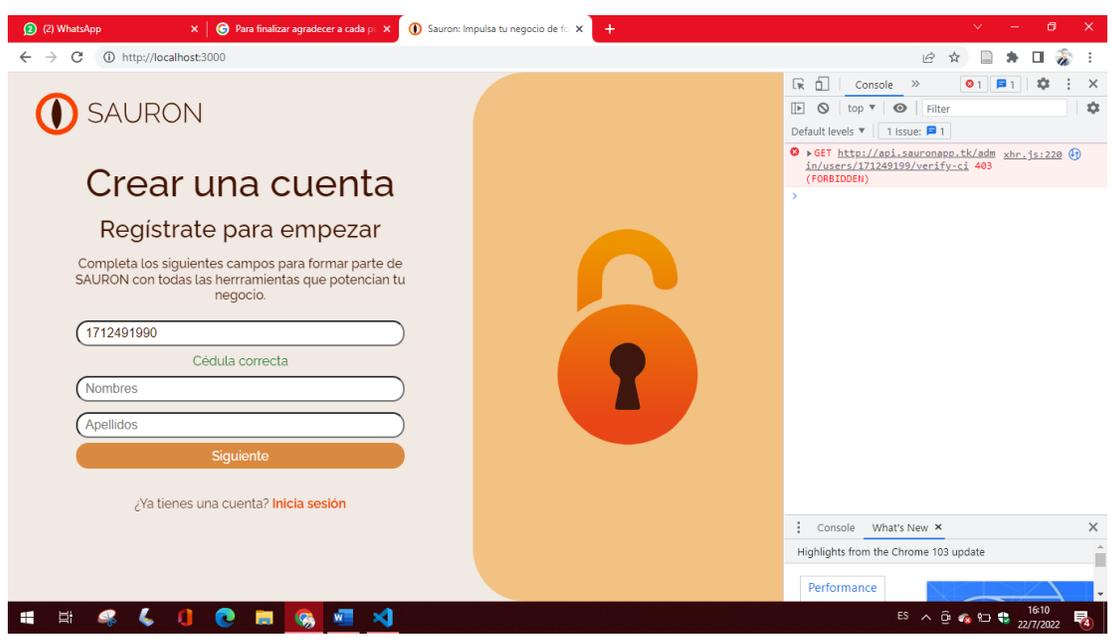
Codificaciones de las conexiones



Nota. Codificación de las conexiones a las APIs mediante Axios

Figura 39

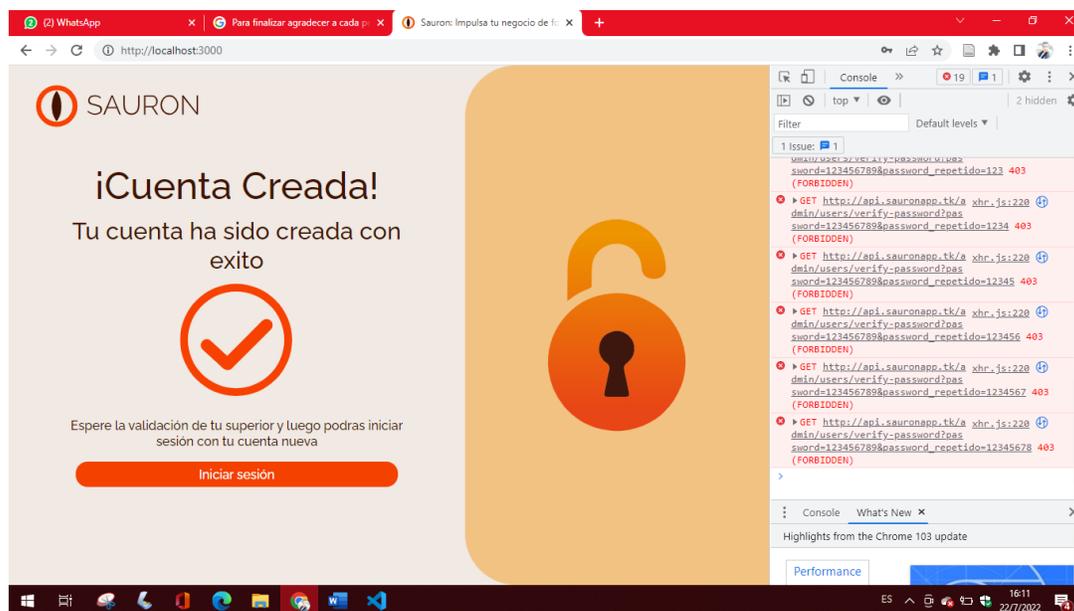
Validación mediante API



Nota. Codificación de las conexiones a las APIs mediante Axios

Figura 40

Envío de nuevo usuario



Nota. Codificación de las conexiones a las APIs mediante Axios

Implementación

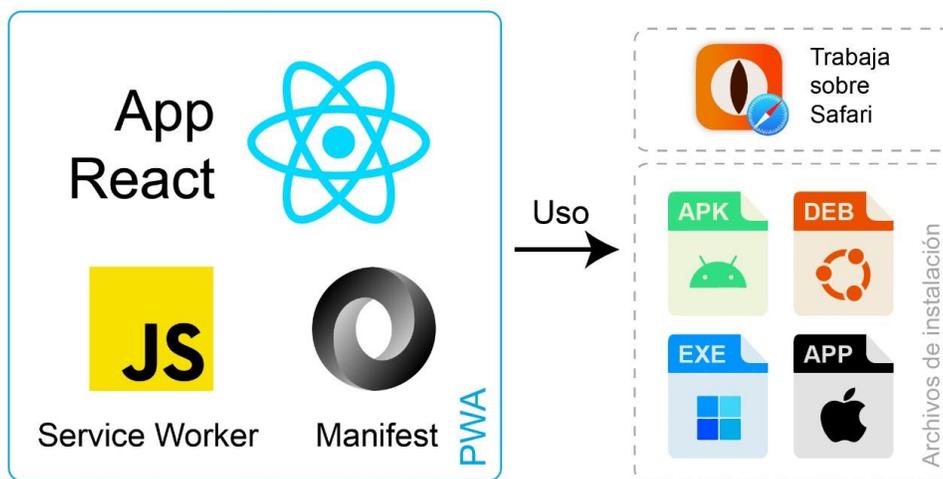
Aplicación Móvil

Se requiere usar un service worker que será el encargado de realizar la conexión entre el host y el navegador web ayudando a utilizar servicios en segundo plano así se tiene una aplicación con características nativas este funcionamiento se realiza en conjunto con la configuración del manifest, ubicado en el archivo raíz de toda la aplicación que será el encargado de dar configuraciones a nuestra aplicación para convertirla en una aplicación web progresiva, en este archivo se ubica el nombre de la aplicación así como también el service worker que estamos trabajando

Para la estructuración de esta aplicación web progresiva se utilizará los mismos componentes que tenemos en nuestra aplicación web lo cual nos dará una escalabilidad en esta creando así una aplicación nativa con las mismas funcionalidades de la aplicación

Figura 41

Estructura Aplicación Web Progresiva



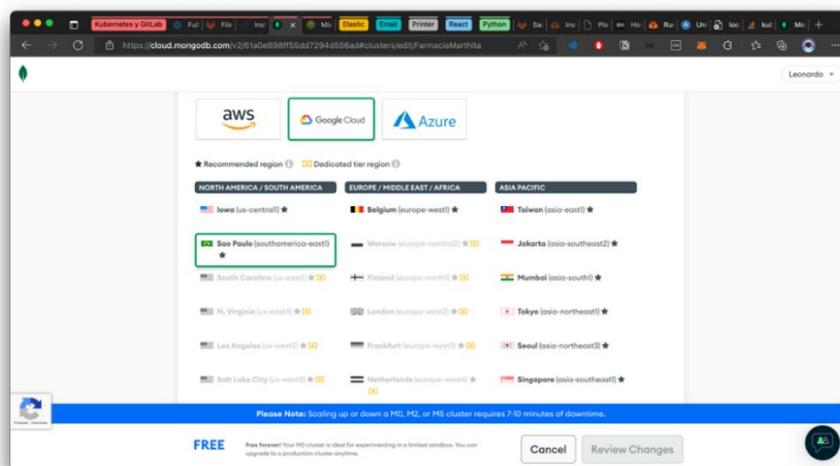
Base de Datos

Finalmente se despliega todos los servicios previamente desarrollados contratando los dos clústers que maneja la aplicación: MongoDB Atlas y Kubernetes.

Se procede a contratar y desplegar un clúster en su versión gratuita, la cual puede presentar limitaciones en cargas de trabajo, pero es ideal para la implementación inicial.

Figura 42

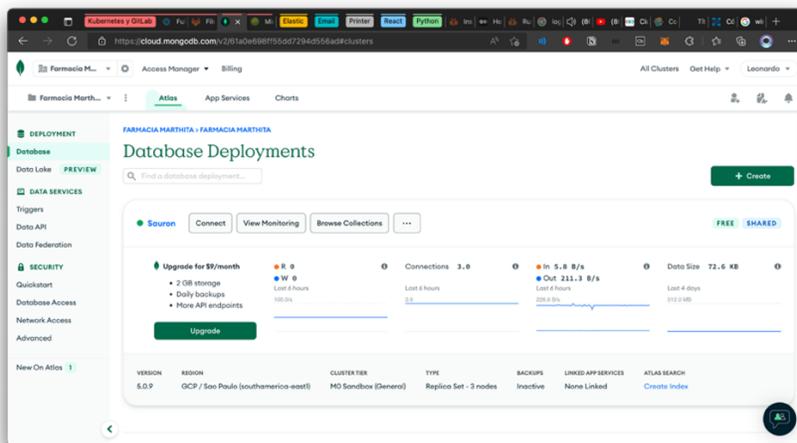
Selección del proveedor de Cloud y ubicación del clúster para MongoDB Atlas



Se selecciona la ubicación que esté en conjunto con el clúster de Kubernetes para obtener mayor velocidad al momento de realizar consultas entre microservicios y la base de datos.

Figura 43

Interfaz para revisión del estado del clúster

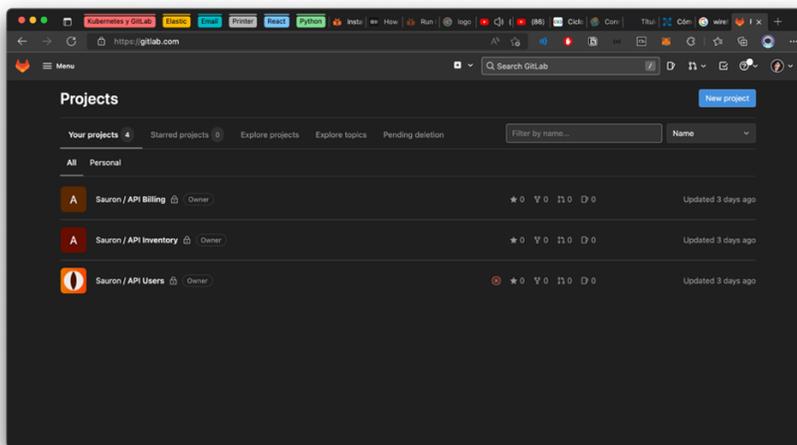


GitLab CI/CD

Para alojar los repositorios y contenedores se debe crear una cuenta GitLab que se conecte con el servicio de GitLab CI.

Figura 44

Microservicios alojados en GitLab



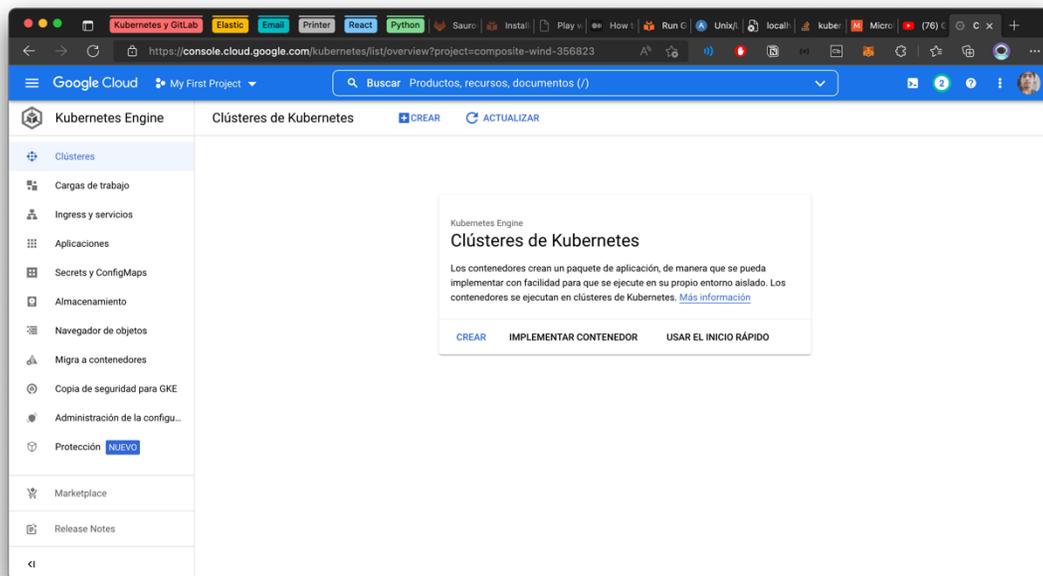
En la sección de repositorio de contenedores permite realizar el llamado a los mismos una vez que se realiza el montaje de la aplicación en el clúster de Kubernetes. Esto permite ir actualizando automáticamente los servicios.

Kubernetes: Google Cloud

Para el alojamiento de todo el servicio se contrata el servicio de Google Kubernetes Engine, que inicialmente ofrece la ventaja de prueba gratuita para poder verificar que todo trabaje sin complicaciones.

Figura 45

Interfaz de estado para el clúster de Kubernetes



Al igual que el clúster de la base de datos, se debe contratar el clúster de Kubernetes en la misma ubicación.

Pruebas Unitarias

Aplicación web

Las pruebas presentan que la aplicación web cumple con los aspectos técnicos requeridos en base a las pruebas realizadas.

Tabla 17

Resultados obtenidos para la carga de la aplicación en navegadores y carga de componentes

ID Caso de prueba	Descripción de la prueba	Datos de prueba	Pasos a ejecutar	Resultado Esperado	Resultado real	Contraseña errónea
CP01	Validar la aplicación para ver su funcionamiento en Google Chrome	Se debe ingresar al siguiente enlace para realizar la prueba: https://sauron.app	<ul style="list-style-type: none"> • Ingresar al URL • Visualizar su renderización • Verificar contenido 	Test React debe cumplir con el renderizado exitoso de cada componente	Test React renderizó sin problemas, aquellos componentes con error fueron corregidos.	Pasado
CP02	Validar la aplicación para ver su funcionamiento en Edge	Se debe ingresar al siguiente enlace para realizar la prueba: https://sauron.app	<ul style="list-style-type: none"> • Ingresar al URL • Visualizar su renderización • Verificar contenido 	Presentar los componentes establecidos sin errores	Cada elemento de la interfaz fue renderizado por el navegador sin dificultad al igual que los estilos definidos	Pasado
CP03	Validar la aplicación para ver su funcionamiento en Firefox	Se debe ingresar al siguiente enlace para realizar la prueba: https://sauron.app	<ul style="list-style-type: none"> • Ingresar al URL • Visualizar su renderización • Verificar contenido 	Presentar los componentes establecidos sin errores	Cada elemento de la interfaz fue renderizado por el navegador sin dificultad al igual que los estilos definidos	Pasado

ID Caso de prueba	Descripción de la prueba	Datos de prueba	Pasos a ejecutar	Resultado Esperado	Resultado real	Contraseña errónea
CP04	Renderizado de componentes	Componente a renderizar	<ul style="list-style-type: none"> • Ejecutar el comando de renderizado • Verificar los datos obtenidos 	Presentar los componentes establecidos sin errores	Cada elemento de la interfaz fue renderizado por el navegador sin dificultad al igual que los estilos definidos	Pasado

Nota. Se puede apreciar que los datos obtenidos se aproximan al resultado esperado, sin mayor dificultad se obtuvo la ejecución adecuada de la interfaz en todos los navegadores. El renderizado de componentes pudo presentar errores que fueron corregidos acertadamente.

Microservicios

Para poder obtener una respuesta inmediata en la API se debe reducir el número de consultas a la base de datos, por lo que se verifica el tiempo que tardaría en realizar la petición y la manera de mejorar esta comunicación. Este dato se representa en la Tabla 18.

Tabla 18

Resultados obtenidos en los microservicios para su tiempo de consulta optimizado

ID Caso de prueba	Descripción de la prueba	Datos de prueba	Pasos a ejecutar	Resultado Esperado	Resultado real	Contraseña errónea
CP06	Tiempo de consulta en cada endpoint	Tiempo que genera Postman	<ul style="list-style-type: none"> • Ejecutar una solicitud por endpoint en Postman • Verificar el tiempo resultante en la aplicación Postman 	Presentar tiempos inferiores a 1 segundo en pruebas locales.	Se presentaron consultas que superaban el resultado esperado y fueron corregidas oportunamente.	Pasado

Nota. Para este caso se realizan los cambios respectivos para mantener consultas óptimas y poder devolver consultas más rápidas

Pruebas de Integración

Una vez realizadas las conexiones entre los microservicios y la aplicación, se consideran dos estados de prueba: local y en remoto.

Tabla 19

Resultados obtenidos en pruebas para la integración entre microservicios y aplicación web

ID Caso de prueba	Descripción de la prueba	Datos de prueba	Pasos a ejecutar	Resultado Esperado	Resultado real	Contraseña errónea
CP07	Verificar el tiempo de respuesta de los microservicios	Tiempo de recepción y muestreo de datos	<ul style="list-style-type: none"> • Cargar sintaxis de debug en código de React • Verificar resultados en navegador 	Obtener respuesta en menos de 1,5 segundos y presentarla en menos de 1 segundo.	Se obtuvo un tiempo de respuesta promedio de 2 segundos.	Pasado
CP08	Validar tiempo de respuesta al utilizar API Gateway	Tiempo de recepción y muestreo de datos	<ul style="list-style-type: none"> • Cargar sintaxis de debug en código de React • Verificar resultados en navegador 	El tiempo total de respuesta no debe superar los 3 segundos.	Se obtuvo un valor máximo de consulta con el resultado de 2,83 segundos.	Pasado
CP09	Obtener el tiempo de carga en remoto	Tiempo de recepción y muestreo de datos	<ul style="list-style-type: none"> • Ingresar a la URL de Sauron • Verificar los registros de los microservicios y aplicación web 	El tiempo de respuesta de microservicios debe ser similar que en local	El tiempo de respuesta para microservicios es de 0,6 segundos y para la aplicación web de 1 segundo	Pasado

Nota. Las verificaciones en local, semi local y en remoto permiten obtener procesos de retroalimentación para que en el despliegue final se obtenga un resultado superior debido al nivel de procesamiento de un servicio en nube.

Pruebas de Rendimiento

El rendimiento se obtiene en la optimización del código y pruebas que validen estos resultados, además de una aplicación que verifique esto con simulaciones de varios clientes simultáneos como GTmetrix.

Tabla 20

Resultados obtenidos en pruebas de rendimiento

ID Caso de prueba	Descripción de la prueba	Datos de prueba	Pasos a ejecutar	Resultado Esperado	Resultado real	Contraseña errónea
CP10	Verificar CSS y Javascript no utilizado	Muestreo de CSS y Javascript no utilizado	<ul style="list-style-type: none"> • Ingresar a GTmetrix • Cargar el enlace de Sauron • Verificar datos resultantes 	La herramienta debe notificar que no encuentra CSS y Javascript sin utilizar	La aplicación encontró CSS y Javascript no utilizado el cual fue eliminado Se presentaron secciones que realizan más de 3 peticiones simultáneas y se corrige oportunamente	Pasado
CP11	Verificar los procesos que realiza	Cantidad de solicitudes simultáneas que se realizan a los microservicios	<ul style="list-style-type: none"> • Ingresar a GTmetrix • Cargar el enlace de Sauron • Verificar datos resultantes 	Manipular el menor número de consultas simultáneas a los microservicios		Pasado

Nota. Considerando los consejos de la aplicación GTmetrix permite obtener mejorar y corregir aquellos aspectos técnicos que pueden ralentizar la aplicación.

Presupuesto

Tabla 21

Presupuesto primario

Material	Cantidad	P/unit	Valor Total
VPS	1	\$ 280,00	\$ 280,00
Certificado SSL	1	\$ 13,95	\$ 13,95
Dominio	1	\$ 15,00	\$ 15,00
Alojamiento en tiendas de aplicaciones	1	\$ 25,00	\$ 25,00
Subtotal			\$ 333,95

Nota. Se realizo un presupuesto general de nuestros gastos del proyecto

Tabla 22

Presupuesto secundario

Material	Cantidad	P/unit	Valor Total
Resmas de papel Bond	1	\$ 3,75	\$ 3,75
Impresiones B/N	450	\$ 0,10	\$ 45,00
Impresión Color	30	\$0,25	\$ 7,50
Curso de JavaScript y React	1	\$150,00	\$ 150,00
Curso de JSON	1	\$ 63,00	\$ 63,00
Subtotal			\$ 269,25

Nota. Se detalla cada gasto secundario

Tabla 23*Costo Total*

Detalle	Valor Total
Costos primarios	\$ 333,95
Costos Secundarios	\$ 269,25
Total	\$ 603,20

Nota. Costo total del proyecto son **\$ 603,20** dólares americanos

Capítulo IV

Conclusiones

Se desarrolló la aplicación Sauron para facilitar la automatización de la información de la farmacia, recolectando todos los requerimientos necesarios y cumplir con todas las necesidades solicitadas. Se crearon interfaces sencillas e intuitivas para que el usuario puede manejar la aplicación rápidamente, nos va mostrar datos reales en las cuales se observan estadísticas que mejoraran las ventas y compras de la empresa a nivel del mercado creando herramientas que faciliten el trabajo.

Por medio del desarrollo de los kubernetes y contenedores (Docker) se generó una alta escalabilidad de la aplicación utilizando el servicio de gitlab generando actualizaciones inmediatas en las API's que se crearon con los microservicios. Se implementarán cambios y actualizaciones anuales de acuerdo al crecimiento de la farmacia.

En conclusión, implementamos una estrategia y herramienta que permita a la empresa optimizar los datos masivos, el objetivo fue trabajar con el recurso de business intelligence para poder generar estadísticas de las compras y ventas sacando un reporte focalizado en las funcionalidades frecuentes de la empresa, como identificar los ingresos y egresos por los cuales debe invertir tiempo y dinero. Lo más importante fue identificar las ganancias o pérdidas de la empresa. Se facilito el manejo de la administración en la Farmacia Marthita.

Recomendaciones

Se recomienda a la farmacia realizar un mantenimiento anual a la aplicación para garantizar el buen funcionamiento de aquella permitiendo mejorar el rendimiento y crear nuevas funcionalidades de acuerdo al crecimiento progresivo de la empresa.

Es importante realizar una auditoría de ser posible diaria para verificar cómo están trabajando los microservicios de tal manera depurar cualquier inconveniente que se presente en la aplicación.

Trabajar junto a la contadora de la empresa que permita gestionar el sistema de facturación de acuerdo con los requerimientos y declaraciones legales de una microempresa.

Generar anotaciones de futuros requerimientos que se presenten con el tiempo y poder implementarlos a futuro.

Glosario de Términos

Aplicación móvil: Una aplicación móvil, también llamada app móvil, es un tipo de aplicación diseñada para ejecutarse en un dispositivo móvil, que puede ser un teléfono inteligente o una tableta. Incluso si las aplicaciones suelen ser pequeñas unidades de software con funciones limitadas, se las arreglan para proporcionar a los usuarios servicios y experiencias de calidad (¿Qué es una aplicación móvil? | Anincubator - Blog, s. f.)

Framework: Un framework es un marco o esquema de trabajo generalmente utilizado por programadores para realizar el desarrollo de software. Utilizar un framework permite agilizar los procesos de desarrollo ya que evita tener que escribir código de forma repetitiva, asegura unas buenas prácticas y la consistencia del código (Qué es Framework - Definición, significado y ejemplos, s. f.).

SCRUM: Scrum es un proceso en el que se aplican de manera regular un conjunto de buenas prácticas para trabajar colaborativamente, en equipo, y obtener el mejor resultado posible de un proyecto. Estas prácticas se apoyan unas a otras y su selección tiene origen en un estudio de la manera de trabajar de equipos altamente productivos («Qué es SCRUM», 2008).

Aplicaciones Web: Una aplicación web o "aplicación web" es un programa. programa que se ejecuta en un servidor web. A diferencia de las aplicaciones de escritorio tradicionales, que son lanzadas por su sistema operativo, se debe acceder a las aplicaciones web a través de un navegador web (Definición de aplicación web, s. f.) .

Android: Android es un sistema operativo para móviles diseñado por la compañía estadounidense Google. Basado en el sistema operativo Linux, su objetivo inicial fue fomentar

el uso de un sistema de tipo abierto, gratuito, multiplataforma y muy seguro, adaptado a los dispositivos móviles como Smartphone y tabletas. Desde su creación, el sistema ha realizado una fuerte apuesta para atraer a desarrolladores, por ello cuenta con una variación de Java denominada Dalvik que permite desarrollar aplicaciones que exploten las utilidades de los dispositivos de manera muy sencilla (Qué es Android - Definición, significado y ejemplos, 2020)

iOS: es un sistema operativo lanzado y utilizado por Apple. Su nombre proviene de iPhone OS. Es decir, iPhone Operative System o Sistema Operativo de iPhone (Qué es iOS, s. f.)

React: (también llamada React.js o ReactJS) es una biblioteca JavaScript de código abierto para crear interfaces de usuario con el objetivo de animar al desarrollo de aplicaciones en una sola página ((3) DesarrolladorSoft - Publicaciones | Facebook, s. f.)

JavaScript: es una especie de lenguaje de programación ligera, interpretado por la mayoría de los navegadores y que les proporciona a las páginas web, efectos y funciones complementarias a las consideradas como estándar HTML (¿Qué es JavaScript?, s. f.)

Bibliografía

- Definición de aplicación web.* (s. f.). Recuperado 9 de marzo de 2021, de https://techlib.net/definicion/web_application.html
- Qué es Android—Definición, significado y ejemplos.* (2020, enero 28). <https://www.arimetrics.com/glosario-digital/android>
- Qué es Framework—Definición, significado y ejemplos.* (s. f.). Recuperado 8 de marzo de 2021, de <https://www.arimetrics.com/glosario-digital/framework>
- Qué es SCRUM. (2008, agosto 4). *Proyectos Ágiles.* <https://proyectosagiles.org/que-es-scrum/>
- ¿Qué es una aplicación móvil? | Anincubator—Blog.* (s. f.). Recuperado 9 de marzo de 2021, de <https://anincubator.com/que-es-una-aplicacion-movil/>
- (3) DesarrolladorSoft—Publicaciones | Facebook.* (s. f.). Recuperado 9 de marzo de 2021, de <https://www.facebook.com/1426068187689871/posts/1743287429301277/>
- Qué es iOS: Características y versiones del sistema operativo de Apple.* (s. f.). ADSLZone. Recuperado 9 de marzo de 2021, de <https://www.adslzone.net/reportajes/software/que-es-ios/>
- ¿Qué es JavaScript?» Su Definición y Significado [2021].* (s. f.). Concepto de - Definición de. Recuperado 9 de marzo de 2021, de [//conceptodefinicion.de/javascript/](http://conceptodefinicion.de/javascript/)
- Aplicaciones Web y ASPs – mercadeo.com.* (s. f.). Recuperado 9 de marzo de 2021, de <http://www.mercadeo.com/blog/2010/01/aplicaciones-web-y-asps/>
- Comunidad_Emagister_8601_laudon.pdf.* (s. f.). Recuperado 9 de marzo de 2021, de https://www.emagister.com/uploads_user_home/Comunidad_Emagister_8601_laudon.pdf
- La historia de las aplicaciones móviles. (2019, noviembre 14). *App&Web.* <https://www.appandweb.es/blog/historia-aplicaciones-moviles/>
- Aguirre, V., Ortu, A., Delía, L., Thomas, P., Corbalán, L., Cáseres, G., & Pesado, P. (2019). PWA para unificar el desarrollo Desktop, Web y Mobile. 9.

Anchundia Medrano, L. A. (2022). Análisis comparativo de tecnologías Front End Angular Js Vs React Js, en el modelo de procesos para el desarrollo de aplicaciones web. <http://dspace.utb.edu.ec/handle/49000/11386>

azure. (s. f.). ¿Qué es DevOps? Explicación de DevOps | Microsoft Azure. Recuperado 17 de julio de 2022, de <https://azure.microsoft.com/es-es/resources/cloud-computing-dictionary/what-is-devops/>

Docker. (2022). Page not found—Docker. [https://www.docker.com/resources/what-container/%20D.%20Inc.%20\(2019\)%20Enterprise%20container%20platform%20for%20high-velocity%20innovation.%20](https://www.docker.com/resources/what-container/%20D.%20Inc.%20(2019)%20Enterprise%20container%20platform%20for%20high-velocity%20innovation.%20) [Online]. %20Available: %20<https://www.docker.com/>

Editorial Etecé. (2013). HTTP - Concepto, para qué sirve y cómo funciona. Concepto. <https://concepto.de/http/>

Eguíluz Pérez Javier. (s. f.). Introducción a JavaScript. 140.

elastic. (s. f.-a). ¿Qué es Elasticsearch? | Elastic. Recuperado 17 de julio de 2022, de <https://www.elastic.co/es/what-is/elasticsearch>

elastic. (s. f.-b). ¿Qué es Kibana? | Elastic. Recuperado 17 de julio de 2022, de <https://www.elastic.co/es/what-is/kibana>

Generalidades del protocolo HTTP - HTTP | MDN. (s. f.). Developer. Recuperado 11 de julio de 2022, de <https://developer.mozilla.org/es/docs/Web/HTTP/Overview>

Gitlab. (s. f.). Executors | GitLab. Recuperado 17 de julio de 2022, de <https://docs.gitlab.com/runner/executors/>

Google Cloud. (2007). Acerca de API Gateway | Documentación de API Gateway | Google Cloud. <https://cloud.google.com/api-gateway/docs/about-api-gateway?hl=es-419>

Google Cloud Platform. (s. f.). uCloudStore. Recuperado 11 de julio de 2022, de <https://www.ucloudstore.com/google-cloud-platform/>

GOYZUETA RIVERA, S. I. (2015). Big Data marketing: Una aproximación (147-158; Revista Perspectivas).

Jiménez Calderón, Mario. (2020). idUS - Ingeniería de pruebas: Aplicación, alcance y rendimiento en un banco de test para una aplicación cliente-servidor. Ingeniería de pruebas: aplicación, alcance y rendimiento en un banco de test para una aplicación cliente-servidor.

<https://idus.us.es/handle/11441/104616>

konga. (2017). Konga—More than just another GUI to Kong Admin API.

<https://pantse1.github.io/konga/>

kubernetes. (2020, abril 29). ¿Qué es Kubernetes? | Kubernetes.

<https://kubernetes.io/es/docs/concepts/overview/what-is-kubernetes/>

Kuhn, A. (s. f.). Kong Open-Source API Management Gateway for Microservices. Kong Inc. Recuperado 17 de julio de 2022, de <https://konghq.com/products/api-gateway-platform>

LEWIS JAMES. (2014). Microservices. martinowler.com.

<https://martinowler.com/articles/microservices.html>

Marzal Varó, A., & Gracia Luengo, I. (2009). Introducción a la programación con Python. Universitat Jaume I. <http://repositori.uji.es/xmlui/handle/10234/24305>

mijacobs. (2022, julio 12). ¿Qué es Git? - Azure DevOps. <https://docs.microsoft.com/es-es/devops/develop/git/what-is-git>

Pastor, J. J. C. (2016). Motores de búsqueda y derechos de autor: Infracción y responsabilidad. Aranzadi Thomson Reuters.

Puciarelli, L. (2020). Node JS - Vol. 1: Instalación - Arquitectura - node y npm. RedUsers.

Rubiales Gómez Mario. (2021). CURSO DE DESARROLLO WEB: HTML, CSS Y JAVASCRIPT. En Manual Imprescindible (2021.a ed., p. 345). ANAYA. https://anayamultimedia.es/primer_capitulo/curso-de-desarrollo-web-html-css-y-javascript-edicion-2021.pdf

Ryan Wynn. (2016). Monstache. <https://rwygn.github.io/monstache-site/>

Vértice, E. (2009). Diseño básico de páginas web en HTML. Editorial Vértice.

Consuelo con Google Cloud Platform—Consuelo. (s. f.). Recuperado 17 de julio de 2022, de <https://solace.com/gcp/>

Diferencias entre un sitio web y una aplicación web. (2016, septiembre 26). *Blog IDA Chile | Estrategia para el éxito de tu negocio.* <https://blog.ida.cl/estrategia-digital/diferencias-aplicacion-web-sitio-web/>

Getting Started with OpenUSM on Docker for Windows Platform – Collabnix. (s. f.). Recuperado 17 de julio de 2022, de <https://collabnix.com/getting-started-with-openusm-on-docker-for-windows-platform/>

LEWIS JAMES. (2014). *Microservices.* martinowler.com.
<https://martinowler.com/articles/microservices.html>

nishanil. (s. f.). *Diferencias entre el patrón de puerta de enlace de API y la comunicación directa de cliente a microservicio.* Recuperado 17 de julio de 2022, de <https://docs.microsoft.com/es-es/dotnet/architecture/microservices/architect-microservice-container-applications/direct-client-to-microservice-communication-versus-the-api-gateway-pattern>

Stratebi. (2014). *Libro Verde Big Data* (Madrid). www.stratebi.com

Anexos

ANEXOS