



**SERVICIO DE NOTIFICACIONES BASADO EN ARQUITECTURA DE MICROSERVICIOS
DIRIGIDA POR EVENTOS PARA LA AUTOMATIZACIÓN Y PERSONALIZACIÓN DE
MENSAJERÍA DIGITAL, EN EMPRESAS DEL SECTOR COMERCIAL**



ÍNDICE

SECCIÓN I: DESCRIPCIÓN GENERAL	3
1.1.- TÍTULO DEL PROYECTO	3
1.2.- DIRECCIÓN DE EJECUCIÓN DEL PROYECTO	3
1.3.- LAPSO DE EJECUCIÓN DEL PROYECTO	3
1.4.- INFORMACIÓN DE CONTACTO DE LOS ESTUDIANTES	3
1.5.- LÍNEA, SUBLÍNEA Y GRUPO DE INVESTIGACIÓN AL QUE SE SUSCRIBE EL PROYECTO	3
SECCIÓN II. DESCRIPCIÓN SITUACIONAL	4
2.1.- IDENTIFICACIÓN DEL PROBLEMA	4
2.2.- JUSTIFICACIÓN DE PROYECTO	5
2.3.- OBJETIVOS DEL PROYECTO	7
2.4 MARCO REFERENCIAL	8
2.5 MARCO METODOLÓGICO	17
SECCIÓN III. DESARROLLO CIENTÍFICO TECNOLÓGICO.	23
3.1 DESARROLLO DE LAS FASES DE LA METODOLOGÍA PROPUESTA.	23
3.2 ANÁLISIS DE RESULTADOS Y DISCUSIÓN.	42
3.3. CONCLUSIONES	44
3.4 RECOMENDACIONES	45
3.5 BIBLIOGRAFÍA	45

SECCIÓN I: DESCRIPCIÓN GENERAL

1.1.- TÍTULO DEL PROYECTO

Servicio de notificaciones basado en arquitectura de microservicios dirigida por eventos para la automatización y personalización de mensajería digital, en empresas del sector comercial.

1.2.- DIRECCIÓN DE EJECUCIÓN DEL PROYECTO

Transversal 29, Sabanas del Valle, Cesar. Teléfono +57 605 588 5592,
pqrs@unicesar.edu.co

1.3.- LAPSO DE EJECUCIÓN DEL PROYECTO

La ejecución del proyecto tendrá un lapso de ciento dieciséis días o tres meses y dieciséis días. 106 días o 3 meses y 16 días.

1.4.- INFORMACIÓN DE CONTACTO DE LOS ESTUDIANTES

Nombre	Apellido	Cédula	Teléfono	Correo
Valentina	Celedón	1003266012	3004595618	vceledonl@unicesar.edu.co
Deyner	Solano	1003380104	3024439080	deyneresolano@unicesar.edu.co

1.5.- LÍNEA, SUBLÍNEA Y GRUPO DE INVESTIGACIÓN AL QUE SE SUSCRIBE EL PROYECTO

Línea: Tecnologías de la Información y la comunicación

Sublínea: Ingeniería de software

Grupo de investigación: AITICE

La línea de Tecnologías de la información y la comunicación se enfoca en incorporar el uso de las tecnologías a nivel empresarial, familiar y también a nivel personal. Con el objetivo de crear, almacenar, intercambiar y dar uso a la información, las TICS acoplan las tecnologías como hardware, software y telecomunicaciones. Esta línea no puede lograr su objetivo por sí sola ya que necesita de las personas, los procesos, las metodologías y las mejores prácticas

que le podamos brindar para hacer una buena gestión de los recursos tecnológicos.

Con esto en mente, la sublínea de Ingeniería de software a la forma en cómo será construido un producto de Software, con el objetivo de gestionar, analizar y verificar la forma en que el software es construido a lo largo de su ciclo de vida logrando así la validación del producto de software construido considerando la calidad del producto como punto trascendental del ciclo del desarrollo. Esta sublínea cuenta con diferentes campos de aplicación donde cada campo se centra en el desarrollo de métodos y herramientas que posibiliten construir software de alta calidad; incluyendo así el proceso de desarrollo y arquitecturas de software.

SECCIÓN II. DESCRIPCIÓN SITUACIONAL

2.1.- IDENTIFICACIÓN DEL PROBLEMA

En el entorno empresarial actual, la comunicación digital (desea entender como todas las formas de interacción y transmisión de información a través de medios digitales) se ha convertido en un elemento crucial para la interacción efectiva con clientes [1]. Las empresas buscan implementar estrategias que les permitan mantener una relación constante con sus clientes a través de mensajes personalizados y oportunos. Sin embargo, la falta de sistemas ágiles y personalizados de mensajería digital puede limitar la capacidad de las empresas del sector comercial para mantener una comunicación efectiva y relevante con su base de clientes.

Actualmente, muchas empresas dependen de sistemas de notificaciones estáticos y poco flexibles que no se adaptan a las necesidades individuales de cada usuario, lo que resulta en una experiencia de usuario impersonal y poco satisfactoria. La encargada de Marketing y ventas Karelys Maestre de la sucursal de Colchones AO ubicada en el centro de Valledupar, comenta lo difícil que es mantener el contacto con todos sus clientes, ya que existen diferentes clientes dependiendo de factores internos de la empresa (desea entender como clientes Gold, clientes favoritos, longevidad del cliente, etc) y a los cuales se les envía diferentes mensajes al mes. En la indagación realizada menciona que “es difícil cada mes tener que redactar más de 4 mensajes diferentes para cada tipo de cliente [...] y luego tener que enviarlos por correo, otros por WhatsApp [...] Sí, se hace muy tedioso el trabajo”. Posteriormente se observa y analiza que dicha empresa no es la única que cuenta con el problema ya que diferentes empresas del sector se ven afectadas.

Luego del profundo impacto de la crisis sanitaria provocada por el Covid-19 y como esta

cambió la forma de comunicarnos, las empresas quieren tener un vínculo más cercano y mantener una presencia activa en la vida de sus clientes, no obstante, existen varias dificultades que obstaculizan este proceso. [2] Entre ellas la individualización de mensajes, enviar mensajes personalizados a cada cliente implica un esfuerzo considerable para adaptar el contenido a las preferencias y comportamientos específicos de cada individuo; por ejemplo, si los mensajes no están personalizados o no se adaptan a las necesidades del cliente, pueden ser ignorados o considerados como spam [5], lo que resulta en una carga de trabajo significativa para las empresas. Igualmente, la gestión manual de mensajes se vuelve cada vez más difícil y propensa a errores a medida que las empresas crecen, ocasionando deslices e información errata. Existe también el riesgo de mensajes genéricos, puesto que la automatización mal gestionada puede conducir a la entrega de mensajes genéricos y no relevantes, lo que puede afectar la percepción del cliente y disminuir la eficacia de las estrategias de marketing.

Existen diferentes plataformas de mensajería digital (incluye WhatsApp, Correo electrónico, Telegram, etc.) [12]; la integración de sistemas y plataformas para la gestión de mensajes programados puede resultar compleja, especialmente cuando la empresa utiliza una o más plataforma como parte de la comunicación, lo que puede llevar a fallos en la entrega y coordinación ineficiente; como ejemplo, si los clientes no pueden acceder a la misma información a través de diferentes canales, pueden sentirse frustrados y confundidos.[6]

En este contexto, surge la necesidad de desarrollar un Servicio de Notificaciones basado en arquitectura de microservicios dirigida por eventos, que permita la automatización y personalización de la mensajería digital en empresas del sector comercial. Esta solución deberá ser capaz de adaptarse dinámicamente a las preferencias y comportamientos de los usuarios, garantizando una comunicación relevante y oportuna en todos los puntos de contacto.

FORMULACIÓN DEL PROBLEMA:

¿Cómo mejorar de forma fácil y eficiente la comunicación empresas-clientes desarrollando un servicio de notificaciones, basado en arquitectura de microservicios dirigida por eventos para la automatización y personalización de mensajería digital, en las empresas del sector comercial?

2.2.- JUSTIFICACIÓN DE PROYECTO

La interacción con el cliente es esencial para el éxito de una empresa. Las empresas deben

adaptarse a las nuevas estrategias de comunicación para satisfacer, principalmente, a sus clientes y llevar a la práctica nuevas formas de comunicación. [1] Las empresas necesitan mantener una comunicación clara y eficaz con sus clientes, ya sea para fomentar el desarrollo, conseguir mayor alcance, promocionar y comunicar de forma pertinente; además de que esta comunicación se haga con el toque característico y enfoque personalizado de cada empresa permitiendo impactar mucho más en sus destinatarios.

El desarrollo de un Servicio de notificaciones basado en arquitectura de microservicios dirigida por eventos para la automatización y personalización de mensajería digital, en empresas del sector comercial se justifica por diversas razones. La capacidad de ofrecer una experiencia de usuario personalizada y oportuna a través de múltiples canales de comunicación es esencial para mantener la satisfacción del cliente y fomentar relaciones sólidas con la audiencia objetivo. Es una idea que puede aportar múltiples beneficios a diversos usuarios y organizaciones. Contando con el beneficio de la integración a otros sistemas o softwares externos y la opciones multicanales para realizar envíos de mensajes.

Desde un punto de vista teórico, el desarrollo de un Servicio de notificaciones basado en arquitectura de microservicios dirigida por eventos para la automatización y personalización de mensajería digital, en empresas del sector comercial se fundamenta en la evolución del marketing y la mensajería digital. La mensajería digital y la personalización se centra en la importancia de ofrecer mensajes relevantes y adaptados a las necesidades individuales de cada usuario a través de plataformas digitales [3], ofrece una oportunidad para mantener una presencia activa y relevante en la vida de los clientes [7]. La personalización en la comunicación [8] proponen la idea de que los mensajes individualizados tienen un impacto positivo en la percepción del cliente y fortalecen la relación marca-cliente.

Desde un punto de vista práctico, el sistema resolvería desafíos reales que enfrentan las empresas del sector comercial, ya que resuelve desafíos operativos en la gestión de la comunicación con los clientes. Automatizar el proceso de envío de mensajes permite a las empresas superar las limitaciones en cuanto a la individualización de mensajes y gestionar eficientemente la comunicación a gran escala [10]. Esto es especialmente crucial en entornos empresariales en constante crecimiento, donde la gestión manual de mensajes se vuelve impracticable. La programación de mensajes permite a los usuarios gestionar su tiempo de manera eficiente, recordando a los clientes eventos importantes y recibiendo esta información relevante cuando más la necesiten. Es importante destacar que la integración de plataformas a través del aplicativo aborda la necesidad de coordinación

eficiente entre los canales de comunicación, mejorando la coherencia y eficacia de las estrategias de marketing como lo plantean Paul Rusell y Jonathan Taylor en su libro Marketing Communications: An Integrated Approach [11]. En la práctica, la arquitectura dirigida por eventos facilita la escalabilidad y el manejo eficiente de eventos diversos, con eso garantiza que los mensajes se envíen de manera oportuna y sin interrupciones. Este patrón también proporciona flexibilidad para la incorporación de nuevas características y mejoras en el servicio de manera sencilla y sin afectar el rendimiento general [9]. Lo que se traduciría en una mayor satisfacción del cliente, un aumento en el compromiso del usuario y una mejora en los resultados comerciales.

Desde un punto de vista metodológico, la adopción de una arquitectura dirigida por eventos implica un enfoque modular y desacoplado tanto en el diseño como el desarrollo del servicio. La metodología se centrará en la identificación clara de eventos significativos (desea entender cómo cumpleaños, festividades, etc), la definición de interacciones entre componentes y la implementación de mecanismos eficientes para gestionar la distribución de mensajes (como los tiempos de envío de mensajes basados en un evento, desea entender como la frecuencia de los mensajes). La programación de mensajes se abordará mediante la implementación de algoritmos eficientes para gestionar la cola de eventos y garantizar entregas oportunas. Además, la metodología ágil podría ser aplicada para adaptarse a las necesidades cambiantes de los usuarios y recopilar retroalimentación constante sobre el comportamiento del usuario, preferencias y respuestas a la mensajería programada para iteraciones efectivas. Esto aseguraría que el proyecto se desarrolle de manera eficiente y efectiva, siguiendo mejores prácticas de desarrollo de software.

2.3.- OBJETIVOS DEL PROYECTO

Objetivo general: Implementar un servicio de notificaciones basado en arquitectura de microservicios dirigida por eventos para la automatización y personalización de mensajería digital, en empresas del sector comercial.

Objetivos Específicos:

- Determinar los requisitos funcionales y no funcionales del servicio de notificaciones de mensajes digitales, en las empresas del sector comercial.
- Diseñar la arquitectura de software del servicio de notificaciones de mensajería digital.

- Desarrollar los componentes propuestos en la arquitectura de software diseñada.
- Desplegar el servicio de notificaciones de mensajería digital en una empresa del sector comercial.

2.4 MARCO REFERENCIAL

2. 4. 1. Arquitectura de software

La estructura fundamental del software es lo que se conoce como Arquitectura de software, se encuentra compuesta por los elementos de software, las propiedades externamente visibles de esos elementos y las relaciones entre ellos. Ian Sommerville en su libro *Software Engineering* explica que "La arquitectura de software comprende las estructuras necesarias para un sistema de software, los componentes de software identificables dentro de esos sistemas y las relaciones entre ellos." [13]

Ahora, el principal objetivo de la arquitectura de software es ofrecer calidad al sistema de administración de datos, esto a partir de su desempeño, ahorro de tiempo, su disponibilidad y usabilidad, la capacidad de modificarse y adecuarse a las nuevas necesidades del sistema, entre otros atributos de calidad. [14] Para mantener la calidad y confianza la arquitectura de software requiere de un determinado ciclo de etapas que constituyen: requisitos, diseño, documentación y evaluación de su funcionamiento.

En arquitectura se le llama patrón, en un contexto dado, a cualquier solución general y reutilizable para problemas recurrentes en ingeniería del software [15] Una vez se tienen los requerimientos bien definidos y razonados el siguiente paso es escoger un patrón de arquitectura, que se puede elegir por cuestiones de familiaridad, comodidad o simple preferencia. Entre esos patrones se encuentran: cliente-servidor, capas, Master-slave, modelo-vista-controlador (MVC), etc.

2. 4. 2. Arquitectura de software basada en eventos

La arquitectura basada en eventos es un patrón de arquitectura diseñado a partir de servicios pequeños y desacoplados que publican, consumen o enrutan eventos. En su libro *Patterns of Enterprise Application Architecture* Martin Fowler expresa que "La arquitectura de software basada en eventos se enfoca en la producción, detección,

consumo y reacción a eventos en el sistema, lo que permite un acoplamiento más débil, una mayor escalabilidad y una mejor capacidad de respuesta." [16]

Este modelo de integración está creado en torno a la publicación, la captura, el procesamiento y el almacenamiento (o la persistencia) de eventos. [17] Un evento puede representar un cambio de estado o una actualización, estos registran los sucesos y los cambios importantes en el estado del hardware y el software del sistema.

Al contrario que la arquitectura de solicitud/respuesta (o modelo de integración) este modelo ofrece ventajas ya que la arquitectura basada en eventos permite un acoplamiento dinámico entre las aplicaciones y los servicios conectados, esto hace que sea más fácil escalar, actualizar y desplegar de manera independiente componentes individuales de un sistema. [18]

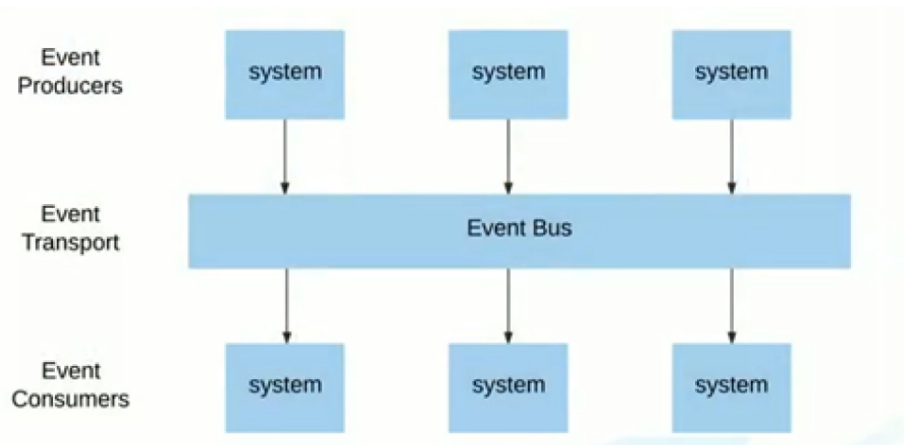


Imagen 1: Arquitectura de software basada en eventos. Fuente [21].

Para entender cómo funciona, en este modelo las aplicaciones actúan como generadores de eventos o consumidores de eventos. Un *generador de eventos* transmite un evento (en forma de un mensaje) a un intermediario donde se mantiene su orden cronológico en relación con otros eventos. Un *consumidor de eventos* ingiere un mensaje (en tiempo real, cuando ocurre, o en otro momento) y lo procesa para desencadenar otra acción, flujo de trabajo o evento propio. Los generadores de eventos desconocen, no se preocupan ni se sienten abrumados por ninguna actividad de los consumidores de los eventos que producen.

2. 4. 3. Microservicios

Los microservicios (o arquitectura de microservicios) son un enfoque arquitectónico nativo de la nube y organizativo para el desarrollo de software en el que el software está compuesto por pequeños servicios independientes que se comunican a través de API bien definidas. Estos servicios normalmente son equipos pequeños independientes, se comunican entre sí mediante API-REST, transmisión de eventos, etc. [19]

Los microservicios son una arquitectura de diseño de software que descompone una aplicación monolítica en un conjunto de servicios autónomos, pequeños y altamente cohesivos, cada uno de los cuales se centra en una función específica. Sam Newman, autor del libro *Building Microservices* [20] explica que "La arquitectura de microservicios estructura una aplicación como una colección de servicios pequeños, autónomos, que funcionan juntos. Cada servicio es autocontenido y debe implementarse, escalarse y actualizarse de forma independiente."

Con la arquitectura monolítica, en comparación con la arquitectura de microservicios donde una aplicación se crea con componentes independientes que ejecutan cada proceso de la aplicación como un servicio, todos los procesos están estrechamente asociados y se ejecutan como un solo servicio.

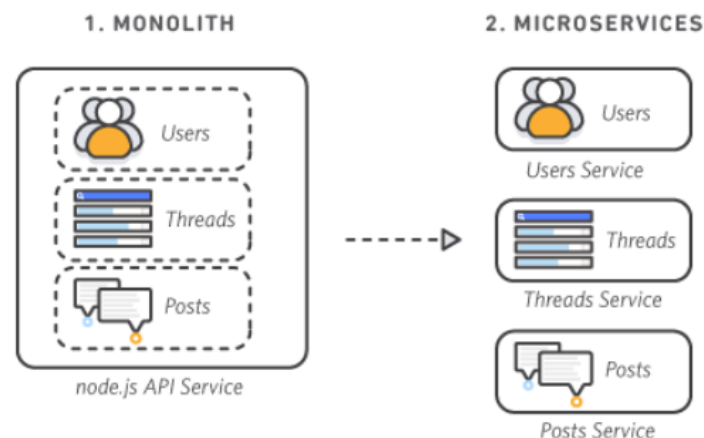


Imagen 2: Diferencia entre arquitectura monolito y microservicios. Fuente [22].

Los microservicios se caracterizan por ser autónomos, entendiendo que cada servicio se puede desarrollar, implementar, operar y escalar sin afectar el funcionamiento de otros servicios; y por ser especializados, donde cada servicio está diseñado para un

conjunto de capacidades y se enfoca en resolver un problema específico. Entre sus beneficios encontramos la agilidad, implementación sencilla, escalado flexible y muchos más. [20]

2. 4. 4. Arquitectura basada en eventos y microservicios

En los microservicios las aplicaciones se ensamblan a partir de servicios acoplados y de despliegue independiente, los microservicios pueden comunicarse entre sí usando API REST. No es raro que la arquitectura basada en eventos se considere ampliamente una buena práctica para las implementaciones de microservicios. [15] La combinación de la arquitectura basada en eventos y los microservicios son una poderosa forma de diseñar sistemas de software flexibles, escalables y resilientes. Los microservicios, en esta arquitectura, se comunican entre sí a través del intercambio de eventos, lo que permite una mayor descomposición de la aplicación en componentes independientes y la mejora del desacoplamiento entre ellos.

Chris Richardson, en su libro *Microservices Patterns* detalla que "La arquitectura basada en eventos y microservicios es un enfoque donde los servicios son diseñados para ser pequeños, autónomos y responsables de una función de negocio específica. Estos servicios se comunican entre sí mediante la emisión y suscripción de eventos, lo que permite una integración flexible y una mayor escalabilidad horizontal." [23]

2. 4. 5. Apache Kafka

Según su sitio web [24] Apache Kafka es una plataforma de transmisión de eventos distribuida de código abierto utilizada por miles de empresas para canalizaciones de datos de alto rendimiento, análisis de transmisión, integración de datos y aplicaciones de misión crítica. Neha Narkhede co-creadora de Apache Kafka lo define como "una plataforma de streaming distribuido que proporciona una arquitectura escalable y tolerante a fallos para el procesamiento de flujos de datos en tiempo real. Permite la publicación y suscripción de datos a través de topics, lo que facilita la construcción de aplicaciones de procesamiento de eventos a gran escala".

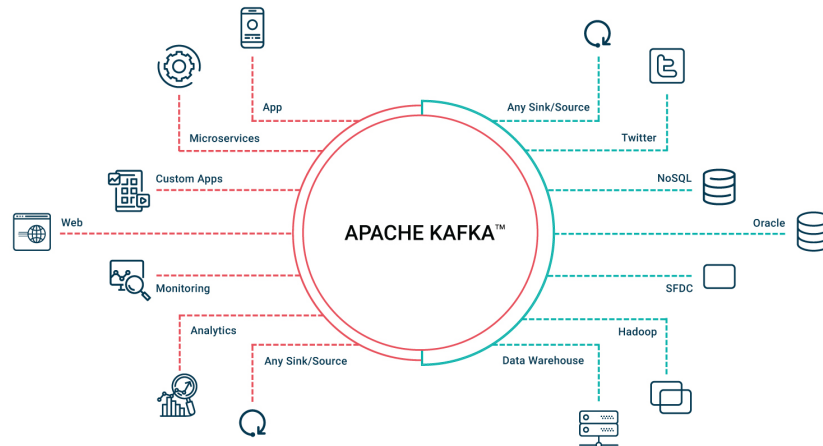


Imagen 3: What is Apache Kafka used for?. Fuente [25].

Kafka se utiliza para crear canalizaciones de datos de transmisión en vivo y aplicaciones de transmisión en vivo. Las canalizaciones de datos procesan datos de forma segura y los mueven de un sistema a otro, mientras que una aplicación de transmisión es una aplicación que consume flujos de datos. Kafka combina modelos de mensajería de cola y de publicación-suscripción, brindando a los consumidores los beneficios clave de cada modelo. Las colas permiten que el procesamiento de datos se distribuya entre muchas instancias de consumidores, lo que las hace altamente escalables. [26]

Con lo anterior se puede concluir que Apache Kafka se utiliza principalmente para crear canalizaciones de datos de *streaming* o transmisión en tiempo real y aplicaciones que se adapten a las secuencias de datos. Combina mensajería, almacenamiento y procesamiento de transmisiones para permitir el almacenamiento y el análisis de datos históricos y en tiempo real.

2. 4. 6. Interfaz de usuario

En términos cortos y más entendibles se define la interfaz de usuario como un medio visual que combina una serie de controles y elementos que permiten a un usuario comunicarse e interactuar con un dispositivo electrónico. La interfaz de usuario (UI) se refiere a la parte de un sistema informático, programa o dispositivo que permite la interacción entre el usuario y la máquina. [27] La interfaz gráfica de usuario (GUI) es el contenido gráfico mediante el cual se visualiza información del equipo en una pantalla.

The Humane Interface escrito por Jef Raskin explica que "La interfaz de usuario es todo lo que permite a los usuarios comunicarse con la máquina. Esto incluye la pantalla, el teclado, el ratón y, en el futuro, quizás, otros dispositivos de entrada como el lápiz, los guantes y los auriculares de realidad virtual." [28]

2. 4. 7. API REST

Según la IBM [29] "una API, o interfaz de programación de aplicaciones, es un conjunto de reglas que definen cómo las aplicaciones o los dispositivos pueden conectarse y comunicarse entre sí." Luego, una API REST es una API que cumple las restricciones de diseño del estilo de arquitectura REST.

API REST (en inglés, Representational State Transfer), es una interfaz de programación de aplicaciones que sigue los principios de la arquitectura REST, la historia de estos comienza desde el 2000, el Dr. Roy Fielding presentó su tesis doctoral en este año, donde definió los principios y conceptos detrás de este enfoque. Define API REST como "una interfaz que usa el protocolo HTTP para transferir representaciones de recursos entre un cliente y un servidor". Fielding también fue uno de los principales autores de la especificación HTTP (en inglés, Hypertext Transfer Protocol).

Hay que tener en cuenta que un servicio REST no es una arquitectura software, sino un conjunto de restricciones que tener en cuenta en la arquitectura software que usaremos para crear aplicaciones web respetando HTTP. [30] Ordoñez en su blog explica que para Fielding un sistema RestFul se define según las siguientes restricciones:

- **Cliente-servidor:** El servidor se encarga de controlar los datos mientras que el cliente se encarga de manejar las interacciones del usuario.
- **Sin estado:** cada petición al servidor debería ser independiente y contener todo lo necesario para ser procesada.
- **Cacheable:** debe admitir un sistema de almacenamiento en caché. Este almacenamiento evitará repetir varias conexiones entre el servidor y el cliente para recuperar un mismo recurso.
- **Interfaz uniforme:** los API REST siguen una interfaz uniforme que consiste en métodos HTTP estándar y convenciones para manipular recursos. Esto facilita la

comprensión y el uso de los API.

- **Sistema de capas:** el servidor puede disponer de varias capas para su implementación. Esto ayuda a mejorar la escalabilidad, el rendimiento y la seguridad.

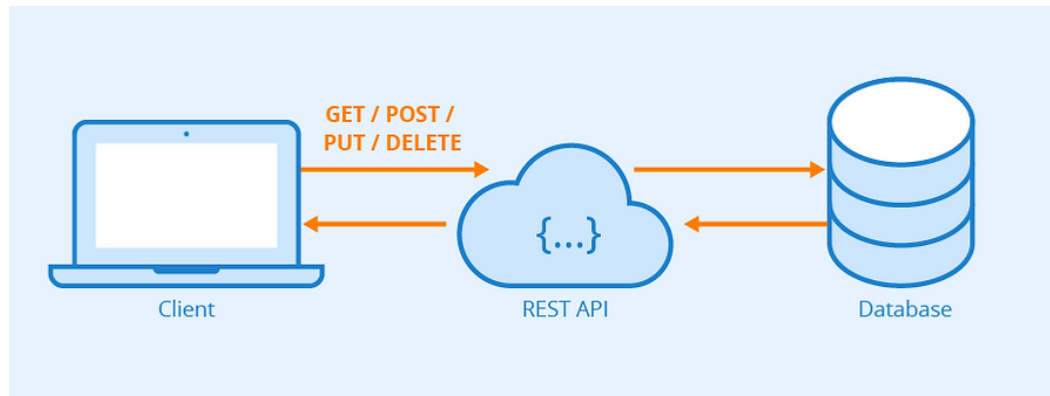


Imagen 5: Basics of REST API!. Fuente [31].

Las API REST se comunican a través de solicitudes HTTP para ejecutar funciones de base de datos estándar como; crear, leer, actualizar y eliminar registros dentro de un recurso. Por ser un estándar lógico y eficiente actualmente la mayoría de las empresas utilizan API REST para crear servicios web.

2. 4. 8. COMUNICACIÓN DIGITAL

El avance de la tecnología ha permitido crear nuevas formas de comunicación a través de redes sociales y sitios web. La comunicación digital es un término amplio que abarca todas las formas de interacción y transmisión de información a través de medios digitales. O como también lo explica el CEI (Centro de Estudios de Innovación) “aquella en la que se intercambian mensajes a través de un medio digital” [1]. En esta transferencia de información se puede incluir no solo la mensajería directa entre individuos, sino también la comunicación a través de redes sociales, blogs, foros, videoconferencias, transmisiones en vivo, entre otros. La comunicación digital puede ser tanto uno a uno como uno a muchos, y puede tener un alcance más amplio y diverso en términos de audiencia y propósito.

El escritor y filósofo Pierre Lévy, investigador en ciencias de la comunicación, opina que “La comunicación digital es un fenómeno complejo que implica la interacción entre humanos y máquinas, mediada por símbolos digitales, como texto, imágenes y

sonidos, que se transmiten y procesan a través de redes de comunicación electrónica” [32].

2. 4. 8. 1. Canales de comunicación digital

Conocer las herramientas que se pueden utilizar para intercambiar información o ideas es importante para saber cuál usar al momento de comunicarnos. Existen distintos canales para comunicarse y vienen revolucionando el mundo de las relaciones sociales y el mundo empresarial. [1]

- **Video:** forma de comunicación más efectiva a la hora de atraer tráfico a un sitio web y generar conversiones
- **Redes sociales:** este medio permite mantener mayor contacto tanto con profesionales como personas cercanas ya sean familiares o amigos
- **Blog:** usados principalmente para difundir información de temas particulares o compartir contenido informativo o de carácter personal.
- **Audio:** pueden ayudar a las campañas de publicidad de empresas más allá de las pantallas en canales como podcast, streaming de música y altavoces inteligentes.
- **Email marketing:** mensajes a través de correos electrónicos que trae diversos beneficios para aquella empresa que lo implemente, entre estos, generar una comunidad y aumentar las ventas

2. 4. 9. MENSAJERÍA DIGITAL

Se refiere al intercambio de mensajes específicos entre dos o más personas a través de plataformas digitales. En su trabajo *La mensajería digital y su importancia en la empresa del siglo XXI*, Rafael R. Díaz y Francisco J. Cazorla proponen que la mensajería digital es un conjunto de tecnologías y herramientas que permiten la comunicación instantánea entre individuos o grupos a través de dispositivos electrónicos conectados a internet. Destacan su utilidad en el ámbito empresarial para mejorar la eficiencia y la colaboración. [33]

La mensajería digital es un tipo de comunicación que se caracteriza por su naturaleza más directa y pueden incluir texto, imágenes, videos o archivos adjuntos. La mensajería digital se centra en la comunicación uno a uno o en grupos pequeños y

tiende a ser más informal y rápida. Este término se refiere al proceso de intercambio de mensajes digitales entre dos o más personas a través de medios electrónicos. Esto incluye tanto la acción de enviar como la de recibir mensajes digitales.

La mensajería digital ha evolucionado significativamente con el avance de la tecnología, y actualmente incluye servicios como correo electrónico, servicios de mensajería instantánea, redes sociales, aplicaciones de chat y sistemas de comunicación empresarial, entre otros. [34] La mensajería digital puede ser síncrona, cuando los mensajes se intercambian en tiempo real, como en una conversación de chat, o asíncrona, cuando los mensajes se envían y reciben en momentos diferentes, como en el correo electrónico.

2. 4. 10. Azure DevOps

Azure DevOps es un conjunto de servicios y herramientas proporcionado por Microsoft para optimizar el ciclo de desarrollo de software. Azure DevOps permite un conjunto de procesos y una cultura colaborativa que reúnen a desarrolladores, administradores de proyectos y colaboradores para desarrollar software. Permite a las organizaciones crear y mejorar productos a un ritmo más rápido de lo que es posible con los enfoques tradicionales de desarrollo de software [35]. Esta plataforma en la nube ofrece todo lo necesario para las empresas que se dedican a crear software y aplicaciones, permitiéndoles implementar la metodología DevOps [36].

Dentro de las distintas herramientas que componen Azure DevOps se destacan:

- **Azure Repos:** ofrece servicio de almacenamiento de código fuente en cualquier lenguaje de programación que soporta control de versiones como Git y Team Foundation Versión.
- **Azure Pipelines:** es ideal para el despliegue continuo y la integración continua y automatizar todas las etapas de desarrollo de software.
- **Azure Boards:** proporciona un conjunto de herramientas para implementar técnicas de gestión de tareas bajo preceptos ágiles, como tableros Kanban o sprints Scrum.
- **Azure Test Plans:** ofrece herramientas para poder realizar test y pruebas que garanticen el mayor nivel de calidad del software que se entrega al cliente.
- **Azure Artifacts:** permite a los equipos compartir paquete para almacenar

componentes de aplicaciones y poder reutilizarlas fácilmente en otros proyectos de desarrollo.

2.5 MARCO METODOLÓGICO

La metodología desarrollada en el presente proyecto toma como marco de trabajo ágil SCRUM para la gestión del proyecto y desarrollo iterativo e incremental.

2. 5. 1. SCRUM

La palabra “scrum” proviene originalmente del rugby. [37] En este deporte, “scrum” se refiere a una formación en la que los jugadores se agrupan de manera compacta para reiniciar el juego. El método Scrum fue inicialmente introducido alrededor de 1986 por Ikujiro Nonaka e Hirotaka Takeuchi, quienes lo crearon a partir de un estudio que realizaron en varias empresas que estaban adoptando un nuevo enfoque de trabajo. [38] Posteriormente expusieron su modelo en espiral de gestión del conocimiento en 1995, algunas veces conocido como el ‘modelo SECI’ el cual aborda las conexiones entre conocimiento tácito y explícito e ilustra cómo el conocimiento es expandido y mejorado a través del proceso. [39]

Jeff Sutherland, uno de los co-creadores de SCRUM, en su libro *SCRUM: The Art of Doing Twice the Work in Half the Time*, [40] define SCRUM como un marco de trabajo que se basa en la iteración y la colaboración para abordar problemas complejos; recalca “la importancia de los principios empíricos de transparencia, inspección y adaptación en SCRUM”.

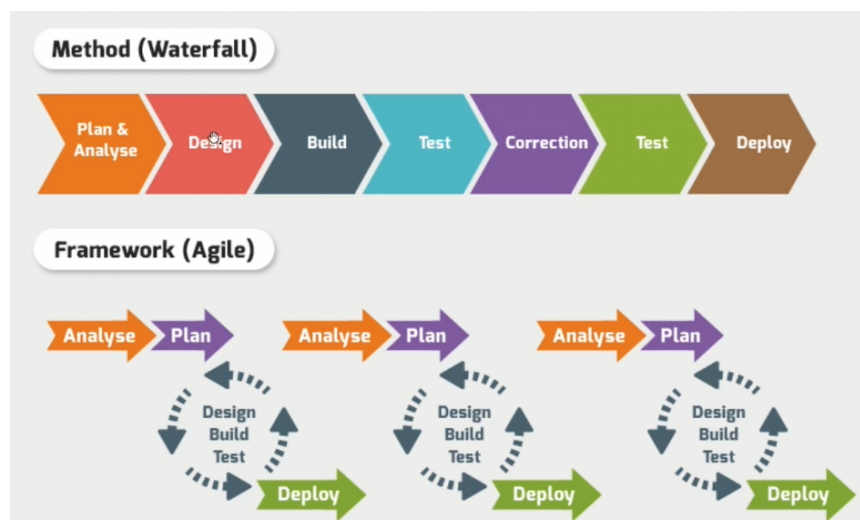


Imagen 6: Diferencias entre metodología tradicional y ágil. Fuente [38].

Scrum es un marco de gestión de proyectos de metodología ágil que ayuda a los

equipos a estructurar y gestionar el trabajo mediante un conjunto de valores, principios y prácticas; permite gestionar proyectos de forma colaborativa, adaptativa y eficiente, especialmente en entornos donde los requisitos y objetivos pueden cambiar con frecuencia. [41] El enfoque SCRUM se basa en la división del proyecto en ciclos de trabajo cortos, llamados Sprints, durante los cuales se desarrolla, prueba y entrega un incremento del producto.

2. 5. 1. 1. Eventos

Un sprint es un período breve de tiempo fijo en el que un equipo de scrum trabaja para completar una cantidad de trabajo establecida. [41] Megan Cook, gestora de productos del grupo de Jira Software en Atlassian menciona que "Con scrum, un producto se basa en una serie de iteraciones llamadas sprints que dividen proyectos grandes y complejos en porciones minúsculas".

La planificación de sprints es un evento que inicia el sprint; el objetivo de esta planificación es definir lo que se puede entregar en el sprint y cómo se conseguirá ese trabajo. Conocido como **sprint planning** es la reunión en la que el Product Owner y el Equipo de Desarrollo acuerdan qué elementos del Product Backlog serán entregados durante el próximo Sprint y cómo se llevará a cabo el trabajo.

La reunión diaria de Scrum o **Daily Scrum** es una breve reunión diaria en la que el Equipo de Desarrollo sincroniza sus actividades y actualiza el progreso hacia el logro del Sprint. Los miembros del equipo se reportan y planifican el día. Informan sobre el trabajo concluido y expresan cualquier desafío para alcanzar los objetivos del Sprint.

Al finalizar el Sprint, el equipo se reúne en el marco de una sesión informal para revisar el trabajo concluido y exponerlo ante las partes interesadas, a esto se le llama **Sprint Review**. Sesión al final de cada Sprint en la que el equipo presenta el Incremento del Producto y recibe retroalimentación del Product Owner y otros stakeholders.

El **Sprint Retrospective** es donde el equipo de trabajo se reúne para documentar, reflexionar y hablar sobre qué funcionó y qué no durante el Sprint para identificar qué salió bien, qué se puede mejorar y define acciones para incrementar la eficacia y la calidad en los siguientes Sprints.

2. 5. 1. 2. Roles

- **Propietario de producto o Product Owner:** Responsable de definir y priorizar los requisitos del producto, representando los intereses del cliente y asegurando

el valor del producto.

- **Scrum master:** es el encargado de garantizar que el equipo de desarrollo comprenda y siga los principios y prácticas de SCRUM. Facilita la colaboración y resuelve obstáculos que puedan afectar al equipo. Forman a los equipos, a los propietarios de los productos y al negocio para mejorar los procesos de Scrum y optimizar la entrega.
- **Equipo de desarrollo:** se encuentra integrado por personas que realizan pruebas, diseñadores, especialistas en la experiencia del usuario, ingenieros de operaciones y desarrolladores, ellos son responsables de llevar a cabo el trabajo necesario para entregar un incremento del producto al final de cada Sprint. [40]

2. 5. 1. 3. Artefactos

Los artefactos son herramientas que usan los equipos de trabajo en SCRUM, sirven para resolver problemas y administrar los proyectos. Estos artefactos proporcionan información crítica de planificación tanto a los miembros del equipo de trabajo como a las partes interesadas. [41]

SCRUM ARTIFACTS



Imagen 7: Artefactos de SCRUM. Fuente [41].

- **Product backlog:** se refiere a una lista dinámica de características, requisitos, mejoras y arreglos que se deben completar para que el proyecto tenga éxito. La lista prioriza todas las funcionalidades, mejoras y correcciones que se desean en el producto. Es responsabilidad del Product Owner mantener el Product Backlog

actualizado y asegurarse de que los elementos más importantes están en la parte superior de la lista.

- **Sprint backlog:** se refiere a una lista de los elementos que el equipo de desarrollo debe completar durante el ciclo Sprint en curso. Antes de cada Sprint, el equipo escoge en qué elementos trabajará a partir del Product Backlog. En otras palabras, es un conjunto de elementos del Product Backlog seleccionados para ser desarrollados durante un Sprint, así como el plan para llevar a cabo dicho desarrollo.
- **Incremento del producto:** o solo incremento, se refiere al resultado del trabajo completado durante un Sprint, que debe ser potencialmente entregable y capaz de ser liberado en producción. A pesar de la flexibilidad, el objetivo fundamental del Sprint, lo que el equipo desea alcanzar a partir del Sprint en curso, no puede verse comprometido.

Las prácticas empleadas por SCRUM sirven para mantener un control ágil en el proyecto que son: revisión de las iteraciones, desarrollo incremental, desarrollo evolutivo, autoorganización del equipo y colaboración.

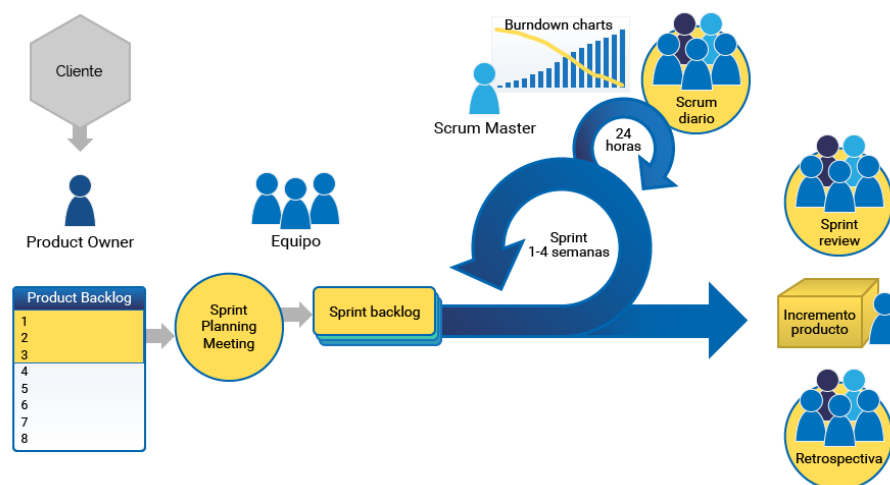


Imagen 8: Metodología SCRUM. Fuente [43].

2. 5. 2. Diseño de Investigación

2. 5. 2. 1. Tipo de investigación:

Este proyecto se basará en una investigación aplicada. La investigación aplicada se centra en la resolución de problemas prácticos mediante la aplicación de conocimientos teóricos y técnicos existentes. En este caso, el objetivo del proyecto es desarrollar un servicio de notificaciones basado en una arquitectura específica

(microservicios dirigidos por eventos) para mejorar la automatización y personalización de la mensajería digital en empresas del sector comercial.

Dentro de la investigación aplicada, este proyecto también tiene elementos de investigación tecnológica o de ingeniería, ya que implica el diseño, desarrollo e implementación de una solución tecnológica innovadora para abordar una necesidad específica en el ámbito empresarial.

2. 5. 2. 2. Población y muestra

La población objetivo incluye las empresas del sector comercial que podrán beneficiarse del servicio de notificaciones basado en la arquitectura de microservicios dirigida por eventos para la automatización y personalización de la mensajería digital. La población está compuesta por todas las empresas dentro de este sector que podrían potencialmente utilizar el servicio. Se seleccionará una muestra representativa de esta población para participar en las pruebas y evaluaciones del aplicativo, con el objetivo de obtener una perspectiva amplia y diversa sobre la viabilidad y la efectividad del servicio de notificaciones propuesto.

2. 5. 3. Metodología de desarrollo

2. 5. 3. 1. Etapas del proyecto

El desarrollo del proyecto se dividirá en las siguientes etapas:

- 1. Análisis:** se identifican las necesidades de los usuarios y los requisitos funcionales del aplicativo; se examina y evalúa la información y los datos relevantes para el tema o problema que se está abordando.
- 2. Diseño:** se elabora un plan detallado de cómo se llevará a cabo el desarrollo del proyecto, la arquitectura del servicio de notificaciones basado en microservicios dirigidos por eventos, la definición de las funcionalidades y características del servicio (incluyendo la automatización y personalización de los mensajes digitales) y el diseño de la interfaz de usuario.
- 3. Desarrollo:** se realiza la codificación para la automatización y personalización de los mensajes, se desarrollan las interfaces de usuario y se integran los diferentes componentes del proyecto, se implementan los microservicios y la infraestructura necesaria para soportar la arquitectura, se realiza la integración con sistemas existentes, entre tanto se van ejecutando

las pruebas de integración.

4. **Evaluación y pruebas:** se efectúa la evaluación del aplicativo, como las pruebas de integración para verificar la interoperabilidad entre los diferentes microservicios, con usuarios reales para obtener retroalimentación y realizar mejoras.

2. 5. 3. 2 Tecnologías y herramientas a usar

Se hará uso de las siguientes tecnologías y herramientas en el desarrollo del aplicativo:

- RabbitMQ para la implementación de sistemas de message queuing o encolado de mensajes.
- .NET 7 y .NET 8 para el desarrollo de los servicios de notificaciones.
- Visual studio y Visual studio code como editores de código fuente.
- Azure App services para el despliegue del sistema.
- Devops para la implementación del marco SCRUM y CI/CD.
- MongoDB como base de datos para el almacenamiento de notificaciones.
- Microsoft SQL server como motor de base de datos para el almacenamiento de plantillas, usuarios, requerimientos y datos de cuenta.
- Angular como framework para la construcción del frontend del sistema.
- Docker para la contenerización del frontend.
- Git para el control de código fuente.
- Figma como herramienta de diseño de interfaz de usuario.

2. 5. 3. 3. Actividades a desarrollar

En la siguiente tabla se especifican las actividades a desarrollar por cada objetivo específico del proyecto.

Objetivo específico	Actividad a desarrollar
- Determinar los requisitos funcionales y no funcionales del servicio de notificaciones de mensajes digitales, en las empresas del sector comercial.	Realizar un análisis detallado de las necesidades y expectativas de las empresas en cuanto a notificaciones de mensajes digitales, para luego clasificar los requisitos identificados según su importancia y urgencia para establecer una jerarquía y orientar el desarrollo del servicio de notificaciones.

- Diseñar la arquitectura de software del servicio de notificaciones de mensajería digital.	Escoger las tecnologías/herramientas que sean adecuadas para la implementación de la arquitectura, para posterior realizar un diseño detallado que especifique con mayor precisión la implementación de cada componente, incluyendo API, bases de datos, algoritmos y patrones de comunicación.
- Desarrollar los componentes propuestos en la arquitectura de software diseñada.	Codificar los diferentes componentes del software, siguiendo las especificaciones y los diseños establecidos para luego integrarlos en un sistema completo, asegurándose de que trabajen de manera conjunta y se comuniquen correctamente entre sí.
- Desplegar el servicio de notificaciones de mensajería digital en una empresa del sector comercial.	Configurar el servidor para alojar el servicio de notificaciones, preparar los componentes para su implementación en el entorno de producción, realizando las configuraciones necesarias, para la final integración del servicio de notificaciones con los sistemas y aplicaciones que se usen en la empresa.

Tabla 1: Actividades a desarrollar. Elaboración propia.

SECCIÓN III. DESARROLLO CIENTÍFICO TECNOLÓGICO.

3.1 DESARROLLO DE LAS FASES DE LA METODOLOGÍA PROPUESTA.

Los procesos que se llevaron a cabo en cada una de las fases en las que se diseñó, desarrolló e implementó el servicio de notificaciones fueron los siguientes:

3.1.1. Fase 1. Análisis:

3. 1. 1. 1. Ingeniería de requerimientos

La ingeniería de requerimientos (IR) es el procedimiento donde se conoce, analiza, documenta y se verifican las descripciones de lo que el sistema debe hacer: el servicio que ofrece y las restricciones en su operación. [36]

3. 1. 1. 1. 1. Requerimientos funcionales

Los requisitos funcionales son premisas sobre los servicios que el sistema debe proporcionar, son el cómo debe responder el sistema a ciertas entradas y cómo debe comportarse en situaciones específicas.

RF#	Requerimiento	Descripción
RF01	Registro de usuarios	El sistema debe permitir que los usuarios de la empresa se registren en el servicio de notificaciones para la gestión de información de contacto y preferencias de notificación de cada usuario.
RF02	Configuración de notificación	El sistema debe permitir la parametrización de la notificación, especificando el tiempo de envío y la recurrencia (cada hora, diaria, mensual, anual).
RF03	Automatización de notificaciones	El sistema debe automatizar el envío de notificaciones según la configuración previa.
RF04	Gestión de plantillas	El sistema debe proporcionar una biblioteca de plantillas de mensajes predefinidas para facilitar la creación y el diseño de mensajes, así como también permitir agregar nuevas y la personalización de estas para adaptarse a las necesidades de cualquier usuario.
RF05	Integración con sistemas externos	El sistema debe permitir la integración con los sistemas externos, exponiendo un servicio rest que permita el envío de notificaciones en eventos específicos.
RF06	Gestión de requerimientos	El sistema debe tener un módulo donde se permita levantar requerimientos, ejemplo: el diseño de una plantilla a partir de mockups, tiempos de envío que no soporte el sistema, etc.
RF07	Inicio de sesión	El sistema debe permitir una verificación de

		usuario a partir de un usuario y contraseña.
RF08	Canales de comunicación	El sistema debe permitir seleccionar el canal de comunicación (Email, SMS y WhatsApp) por el cual se va a enviar la notificación.

Tabla 2: Tabla de requerimientos. Elaboración propia.

3. 1. 1. 1. 2. Historias de usuario

HU01	Registro de usuario
Como	Usuario
Quiero	poder registrarme en el servicio de notificaciones ingresando un correo y contraseña
Para	poder iniciar sesión en el sistema
Condiciones	

Tabla 3: HU01. Elaboración propia.

HU02	Iniciar sesión
Como	Usuario
Quiero	iniciar sesión a partir de un usuario y contraseña en el sistema
Para	hacer uso de los servicios que brinda el sistema
Condiciones	HU01 - Registro de usuario

Tabla 4: HU02. Elaboración propia.

HU03	Registro de plantillas
Como	Usuario
Quiero	que se permita el registro de plantilla de mensaje; agregando el Nombre de la plantilla, tipo (Email, SMS y WhatsApp), <i>from</i> o desde donde se envía la plantilla, asunto y el archivo html del diseño de la plantilla

Para	adicionar plantillas que se adecuen a mis necesidades o a alguna función que requiera
Condiciones	HU02 - Iniciar sesión

Tabla 5: HU03. Elaboración propia.

HU04	Editar plantillas
Como	Usuario
Quiero	que se permita la edición de plantilla de mensaje
Para	editar desde donde envió la plantilla, asunto y el archivo html del diseño de la plantilla en caso de ser necesario
Condiciones	HU02 - Iniciar sesión, HU03 - Registro de plantillas

Tabla 6: HU04. Elaboración propia.

HU05	Biblioteca de plantillas
Como	Usuario
Quiero	que el sistema tenga una biblioteca de plantillas
Para	poder visualizar las plantillas creadas o por defecto y aquellas que yo haya agregado
Condiciones	HU03 - Registro de plantillas

Tabla 7: HU05. Elaboración propia.

HU06	Eliminar plantillas
Como	Usuario
Quiero	que se permita eliminar la plantilla de mensaje que desee
Para	poder retirarla de mi biblioteca de plantillas
Condiciones	HU03 - Registro de plantillas, HU05 - Biblioteca de plantillas

Tabla 8: HU06. Elaboración propia.

HU07	Configuración de notificación
Como	Usuario
Quiero	poder parametrizar la notificación que deseo enviar especificando el tiempo de envío y la recurrencia (cada hora, diaria, mensual, anual)
Para	adaptar la notificación a mis necesidades y preferencias
Condiciones	HU03 - Registro de plantillas

Tabla 9: HU07. Elaboración propia.

HU08	Automatización de notificaciones
Como	Usuario
Quiero	que los mensajes sean automatizados según la configuración previa
Para	la eficiencia y el ahorro de mi tiempo, sin necesidad de tener que estar pendientes de su respectivo envío
Condiciones	HU07 - Configuración de notificación

Tabla 10: HU08. Elaboración propia.

HU09	Integración con sistemas externos
Como	Usuario
Quiero	se me permita la integración con los sistemas externos, exponiendo un servicio rest
Para	enviar notificaciones que se desencadenan a partir de un evento específico, ejemplo: confirmación de compra.
Condiciones	HU03 - Registro de plantillas

Tabla 11: HU09. Elaboración propia.

HU10	Registrar requerimiento
Como	Usuario

Quiero	que el sistema me permita levantar requerimientos de plantilla o notificación; agregando el Nombre del requerimiento, descripción, prioridad y anexo
Para	dar soporte a las necesidades que el sistema no supe en el momento y solicitar la creación de plantillas a partir de un diseño
Condiciones	HU02 - Iniciar sesión

Tabla 12: HU10. Elaboración propia.

HU11	Editar requerimiento
Como	Usuario
Quiero	que el sistema me permita modificar requerimientos
Para	actualizar el Nombre del requerimiento, descripción, prioridad y anexos en caso de ser necesario
Condiciones	HU10 - Registrar requerimiento

Tabla 13: HU11. Elaboración propia.

HU12	Eliminar requerimiento
Como	Usuario
Quiero	que el sistema me permita eliminar requerimientos
Para	descartar y que ya no se realice el requerimiento
Condiciones	HU10 - Registrar requerimiento

Tabla 14: HU12. Elaboración propia.

3. 1. 1. 2. Requerimientos no funcionales

Los requerimientos no funcionales no se relacionan directamente con los servicios específicos que el sistema entrega a sus usuarios. Pueden relacionarse con propiedades emergentes del sistema, como fiabilidad, tiempo de respuesta y uso de almacenamiento.

RNF#	Requerimiento no funcional	Descripción
RNF01	Usabilidad	El servicio debe proporcionar mensajes de error que sean informativos cuando ocurra una excepción orientados al usuario.
		El servicio debe proporcionar mensajes de éxito que sean informativos cuando el procedimiento realizado sea correcto orientados al usuario.
		El servicio debe poseer interfaces gráficas bien formadas.
		El servicio debe contar con mensajes informativos, de ayuda, cuando el usuario esté realizando cualquier proceso.
RNF02	Eficiencia	El servicio tiene al menos 5 segundos para mostrar cualquier información modificada y/o actualizada en la base de datos al usuario que accede.
		El servicio debe responder a una petición en un tiempo promedio de 3 Segundos.
RNF03	Portabilidad	El servicio debe ejecutarse bajo los navegadores Chrome, Edge.
RNF04	Interoperabilidad	El servicio debe tener la capacidad de comunicarse entre sistemas diferentes, compartiendo e intercambiando información.
RNF05	Fiabilidad	Si el servicio falla, el tiempo para reiniciarlo no podrá ser mayor a 5 minutos.

Tabla 15: Requerimientos no funcionales. Elaboración propia.

3. 1. 1. 3. Diagrama de secuencia

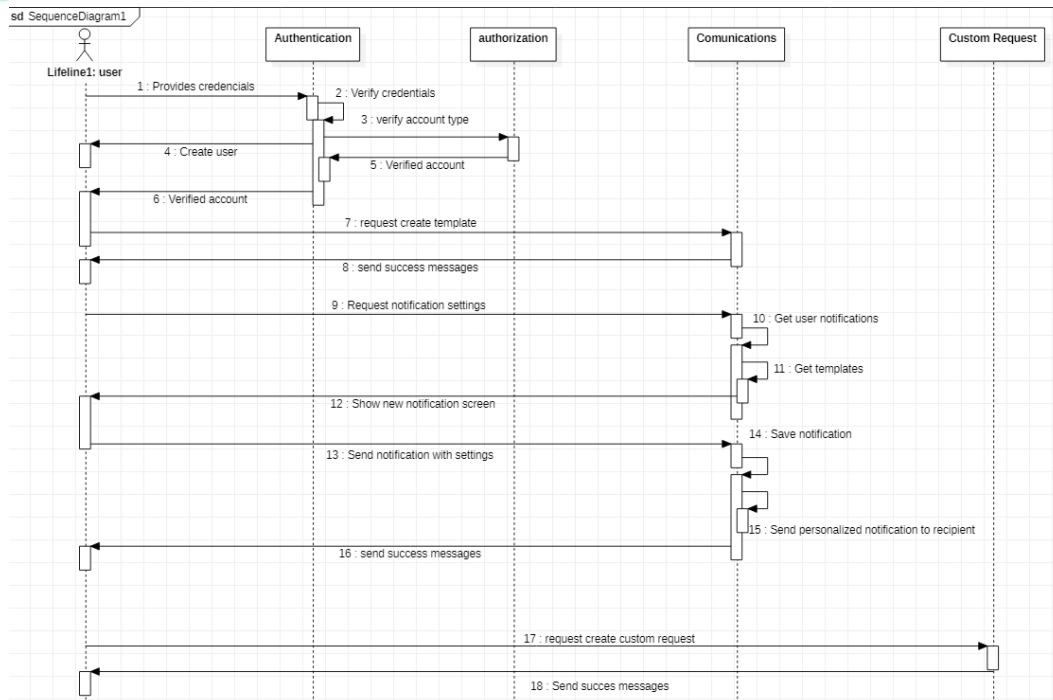


Imagen 9: Diagrama de secuencia del proyecto. Elaboración propia.

3. 1. 1. 3. Diagrama de componentes

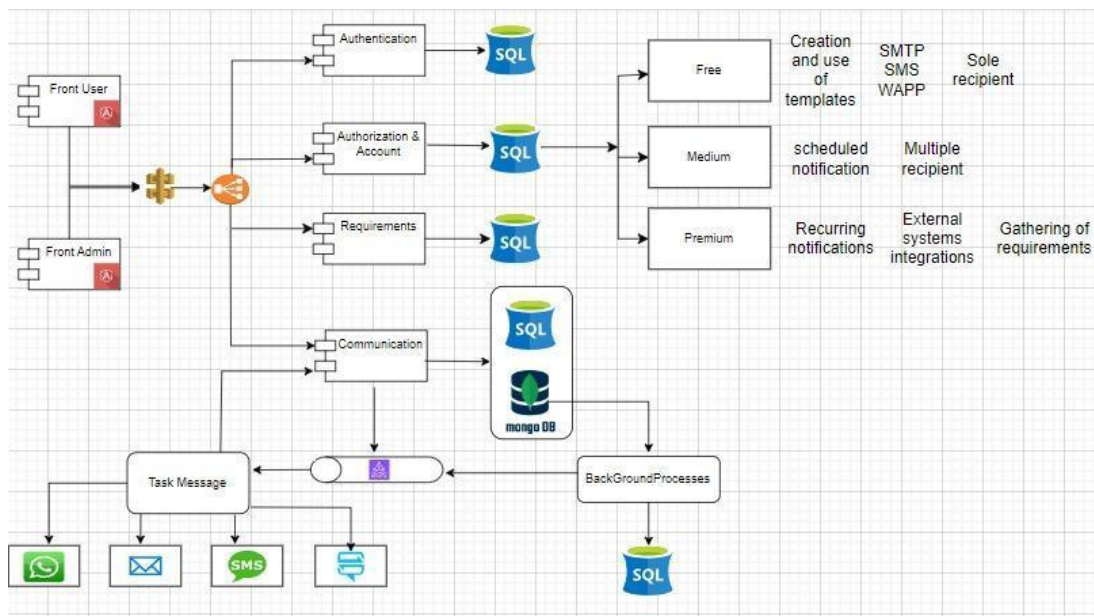


Imagen 10: Diagrama de componentes del proyecto. Elaboración propia.

3. 1. 2. Fase 2. Diseño:

En la siguiente etapa se realiza el diseño de la arquitectura del servicio de notificaciones y el diseño de la interfaz de usuario.

3. 1. 2. 1. Diseño de la experiencia de usuario

Para [37] el mockup es un prototipo de un producto que muestra cómo funciona un objeto en el mundo real. Se emplea para mostrar una idea. Para el desarrollo del servicio de notificaciones se tuvieron presentes los siguientes diseños para el UX/UI, diseñados con la herramienta de diseño gráfico y prototipado *Figma*:

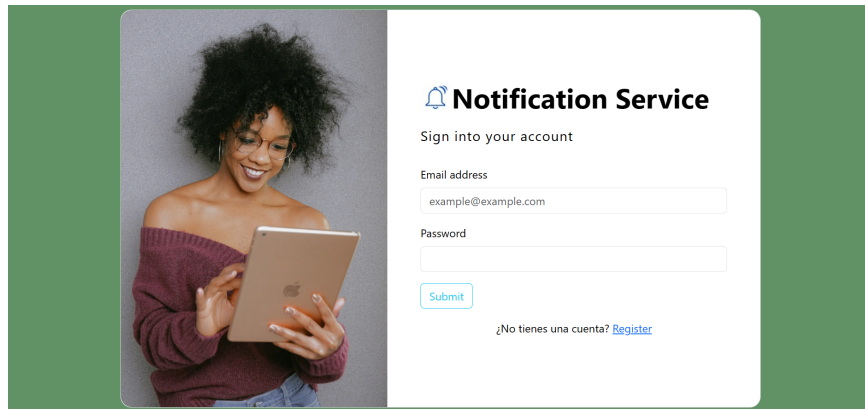


Imagen 11: Log In. Elaboración propia.

En la imagen se puede observar el inicio de sesión del servicio de notificaciones solicitando correo y contraseña, así como la opción de registrarse si no se tiene una cuenta.

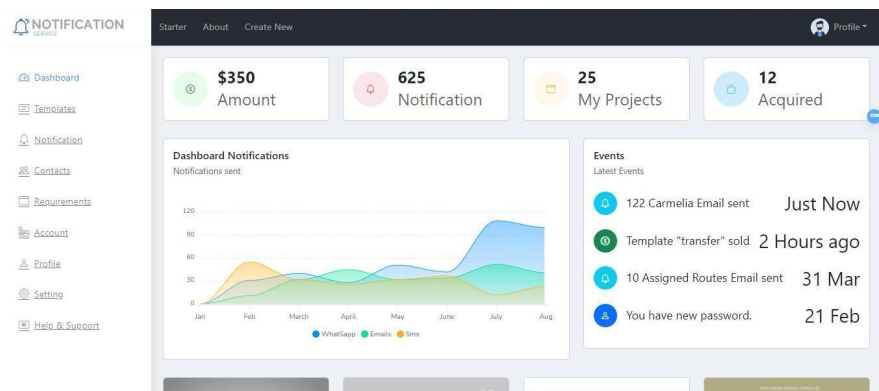


Imagen 12: Dashboard. Elaboración propia.

Se presenta la pantalla inicial del servicio, siendo esta un dashboard que muestra las tareas realizadas a lo largo del uso. Presentando una estadística de cuantas notificaciones se han guardado, cuantos proyectos (en este caso, plantillas) existen, entres otros.

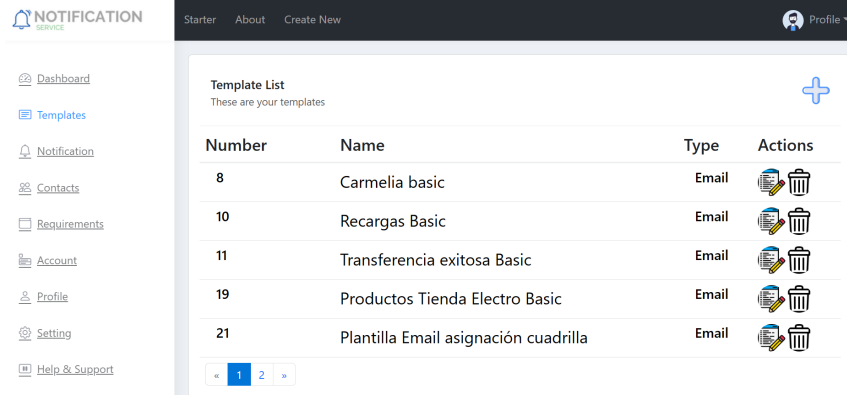


Imagen 13: Templates. Elaboración propia.

En la anterior se observa el dashboard inicial del servicio, así como la opción de *templates* donde se podrá hacer la gestión de las plantillas.

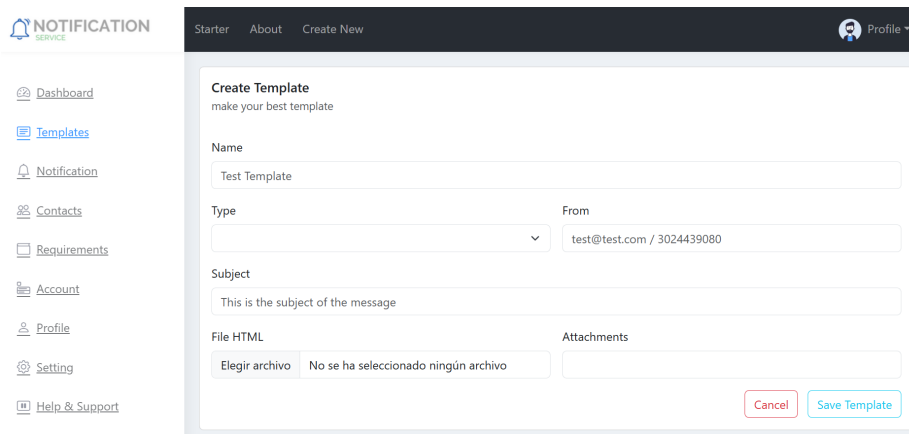


Imagen 14: Create Templates. Elaboración propia.

En el anterior se aprecia la creación de plantillas, solicitando campos como el nombre de la plantilla, el tipo (SMS, Email, WhatsApp), desde donde se enviará, el asunto, el archivo html de la plantilla, así como la opción de guardar o cancelar el proceso.

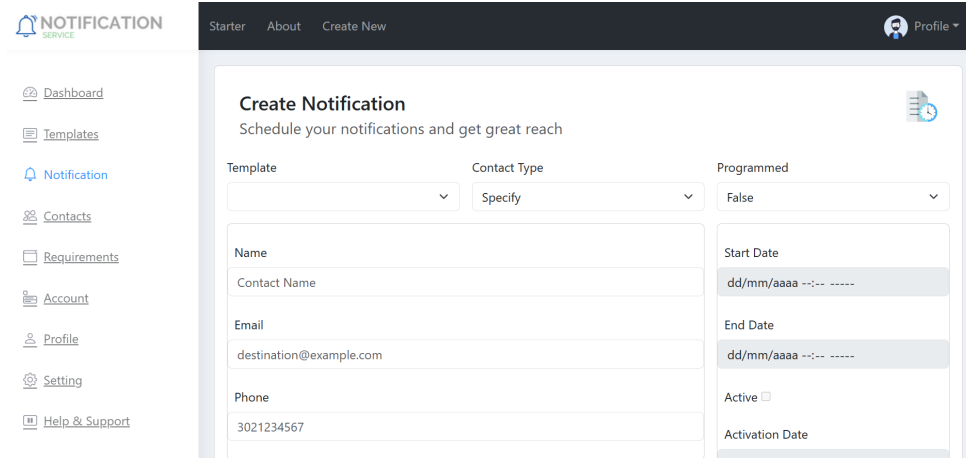


Imagen 15: Notification. Elaboración propia.

Por otra parte en la sección *Notification* se muestra la interfaz de la creación de notificaciones, así como todas aquellas preferencias o configuraciones que el usuario desee escoger para su mensaje.

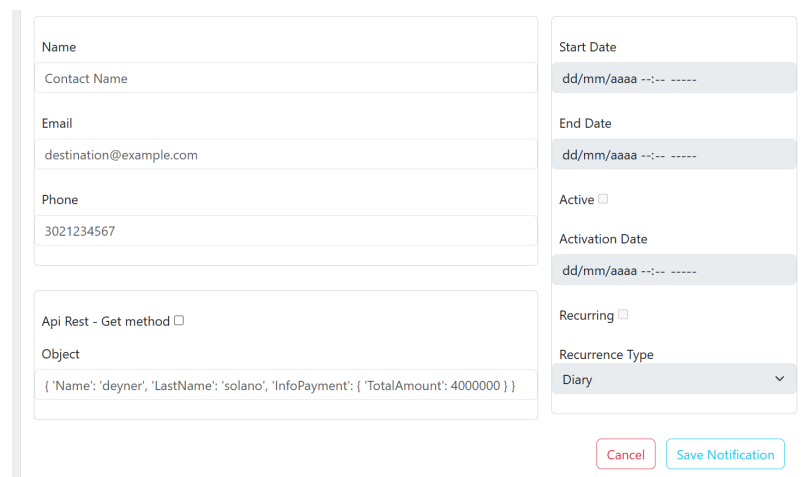


Imagen 16: Notification 2. Elaboración propia.

Resto de campos de la interfaz *Notification* para la configuración de la notificación.

3. 1. 2. 2. Arquitectura del proyecto

La siguiente imagen muestra cómo será el diseño de la arquitectura general del presente proyecto, se puede observar cuáles son sus conexiones y componentes.

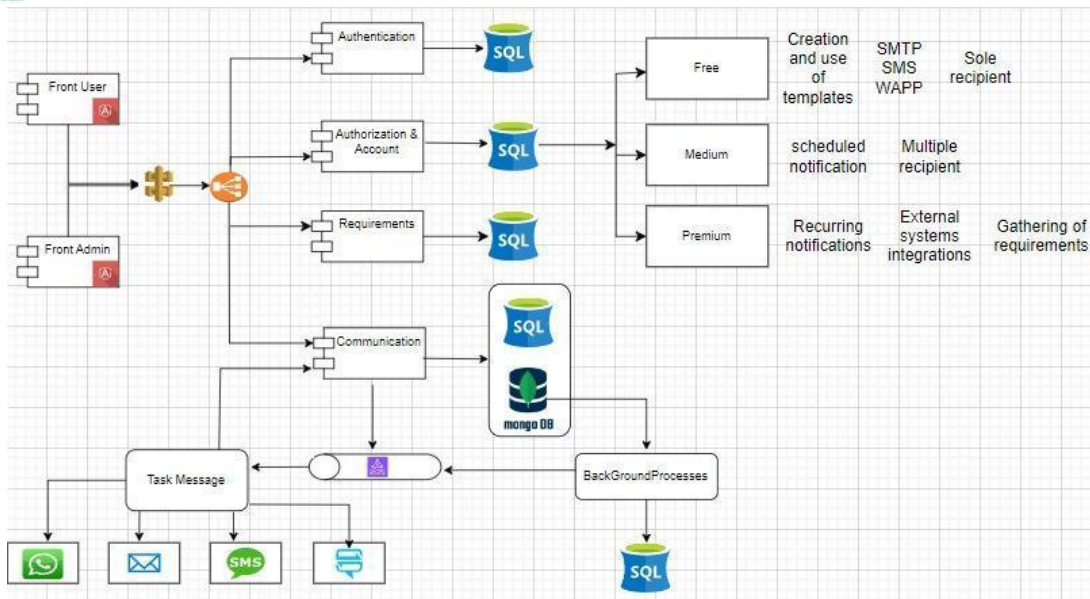


Imagen 17: Arquitectura del proyecto. Elaboración propia.

3. 1. 3. Fase 3. Desarrollo:

Se implementa SCRUM como marco metodológico de trabajo para el desarrollo del proyecto, recordando que este se basa en la iteración y la colaboración para abordar problemas complejos. Teniendo en cuenta lo anterior se hace uso de la herramienta Jira para la gestión del proyecto en las actividades iniciales.

Para el cumplimiento del primer objetivo se realiza un análisis detallado de las necesidades y expectativas de las empresas en cuanto a notificaciones de mensajes digitales. Con el propósito de cumplir con la actividad se dialoga con distintas personas del sector comercial, siendo el primer caso la encargada de Marketing y Ventas de la empresa Colchones AO, quien comenta los casos de frustración que ha generado el envío de mensajes personalizados para sus diferentes clientes. Luego de observar y analizar que dicha empresa no es la única con estas dificultades, se clasifican los requisitos identificados según su importancia y urgencia para establecer una jerarquía y orientar el desarrollo del servicio de notificaciones, como se observa en los requerimientos funcionales.

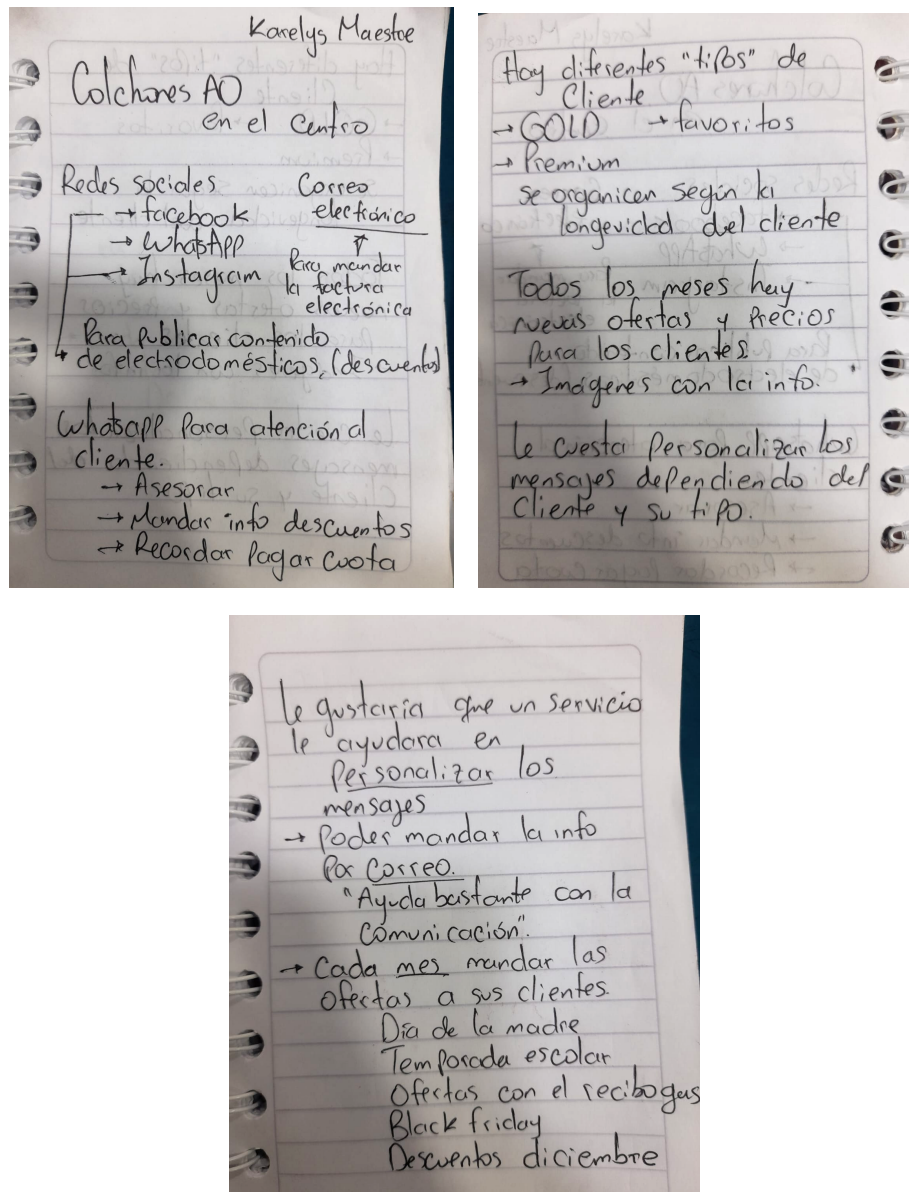


Imagen 18: Requisitos que se tomaron en la investigación. Elaboración propia.

Para la ejecución del segundo objetivo se hace un estudio de las tecnologías y herramientas actuales en el mercado y posteriormente se escogen las adecuadas para la implementación de la arquitectura de microservicios dirigida por eventos, *imagen 15*. A fin de realizar un diseño detallado en donde se especifique la implementación de cada componente, se usa la herramienta draw.io para diseñar la arquitectura de software del servicio de notificaciones de mensajería digital.

Seguidamente, con el fin de realizar el tercer objetivo se pasa a la fase de codificación de los diferentes componentes del software, siguiendo las especificaciones y los diseños establecidos en el diagrama de secuencia y de componentes e igualmente en la arquitectura

del proyecto. Siguiendo la implementación del marco SCRUM con Jira, se muestran las actividades realizadas en el desarrollo de funcionalidades de notificación.

SCRUM-16	Consultar lista de templates	Historia	FINALIZADA
SCRUM-17	Editar Template	Historia	FINALIZADA
SCRUM-18	Render Template	Historia	FINALIZADA
SCRUM-19	Programar o enviar notificación front	Historia	FINALIZADA
SCRUM-15	Realizar Cronograma de actividades	Historia	FINALIZADA
SCRUM-20	Redactar marco metodológico	Historia	FINALIZADA

Imagen 19: Sprint 2. Elaboración propia.

Como principio se desarrolla el CRUD de plantillas, en el proyecto llamadas *template*, seguido se prepara la funcionalidad “crear notificaciones”, “notificaciones programadas”, “notificaciones recurrentes”. El desarrollo de esta funcionalidad se realiza en la API de *communication* y en Angular la interfaz de usuario.

Posteriormente, se muestran las actividades realizadas en el desarrollo de interfaz de usuario.

SCRUM-21	Crear API Authentication	Historia	FINALIZADA
SCRUM-22	Implementar Login	Historia	FINALIZADA
SCRUM-23	Crear Api Account	Historia	FINALIZADA
SCRUM-26	Crear Usuarios y asignar permisos	Historia	FINALIZADA
SCRUM-29	Validar según permisos los accesos Front	Historia	FINALIZADA
SCRUM-30	Redactar sección I de anteproyecto	Historia	FINALIZADA
SCRUM-31	Investigar línea, sublínea y grupo de investigación...	Historia	FINALIZADA
SCRUM-32	Mejorar estructura del documento	Historia	FINALIZADA
SCRUM-35	Asignar punto 1 y 2 a las sección II del anteproyecto...	Historia	FINALIZADA

Imagen 20: Sprint 3. Elaboración propia.

Luego de implementar el CRUD *template* y las funcionalidades se desarrolla inicio de sesión, así como la implementación de usuarios y roles, y la validación de permisos según el usuario. Se muestran las actividades realizadas en el desarrollo de funcionalidades de autenticación backend, como el registro de usuarios.

Del sprint 4 en adelante se hace uso de Azure DevOps como marco de trabajo para la implementación de SCRUM. Se da inicio al despliegue del sistema, empezando por el despliegue de la API de autenticación y las bases de datos.

Order	ID	Title	Assigned To	State
1	3	Deploy Authentication Api	deyner solano...	Done
	28	create app service azure	deyner solano...	Done
	29	implementacion CI	deyner solano...	Done
	30	Implementattion CD	deyner solano...	Done
2	8	Desplegar bases de datos SQL	deyner solano...	Done
	13	montar BD en azure	deyner solano...	Done
	14	ajustar connection string in projects	deyner solano...	Done
	15	setup app configuration azure	deyner solano...	Done
3	12	desplegar base de datos mongodb	deyner solano...	Done
4	11	Investigar y redactar marco referencial	VALENTINA	Done
	19	Investigar y redactar Arquitectura de software	VALENTINA	Done
	20	Investigar y redactar Arquitectura de software basada en ...	VALENTINA	Done
	21	Investigar y redactar Microservicios	VALENTINA	Done

Imagen 21: Sprint 4. Elaboración propia

Se da comienzo al siguiente sprint siguiendo las actividades de despliegue de sistema, siendo estas, el despliegue de la API de comunicación, de cuenta, la Task Message, Task de background process.

Order	ID	Title	Assigned To	State
1	61	Preparar presentación evento emprendimiento	deyner solano...	Done
2	5	Deploy Account Api	deyner solano...	Done
3	6	Deploy Communication Api	deyner solano...	Done
4	9	Desplegar server RabbitMq	deyner solano...	Done
5	32	Add Hanfire Task Message	deyner solano...	Done
6	33	Add Hanfire Background Processes	deyner solano...	Done
7	43	Sección I: Descripción general	VALENTINA	Done
8	52	Sección II: Descripción situacional	VALENTINA	Done
9	58	Marco metodologico	VALENTINA	Done

Imagen 22: Sprint 5. Elaboración propia

Se configura el servidor para alojar el servicio de notificaciones, preparar los componentes para su implementación en el entorno de producción, realizando las configuraciones necesarias, para la final integración del servicio de notificaciones con los sistemas y aplicaciones que se usen en la empresa.

3. 1. 4. Fase 4: Evaluación y pruebas

Se realizan las pruebas de integración en el proyecto, a continuación se presentan las pruebas incrementales y sus casos de prueba.

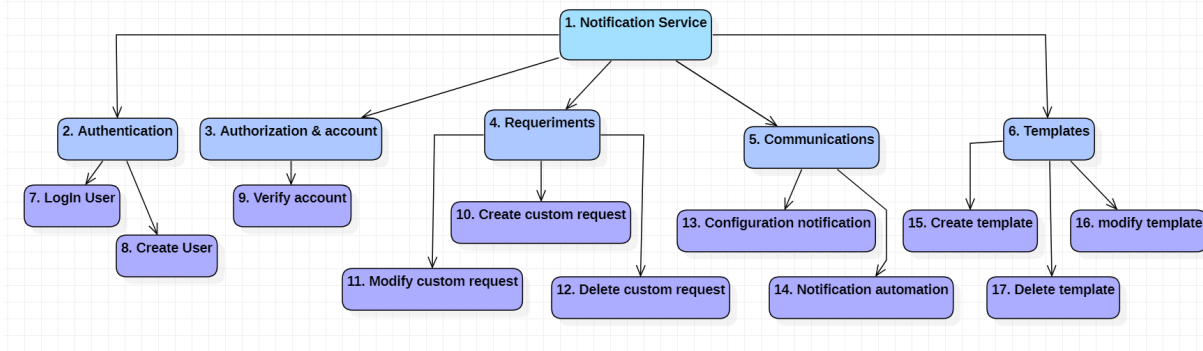


Imagen 23: Diagrama pruebas de integración. Elaboración propia

3. 1. 4. PRUEBAS INCREMENTALES

• INCREMENTAL ASCENDENTE

1. Unitarias: 7, 8, 9, 10, 11, 12, 13, 14, 15, 16 y 17.
2. Integración de (2 con 7), (2 con 8), (3 con 9), (4 con 10), (4 con 11), (4 con 12), (5 con 13), (5 con 14), (6 con 15), (6 con 16), (6 con 17).
3. Integración de (1 con 2), (1 con 3), (1 con 4), (1 con 5), (1 con 6).

• INCREMENTAL DESCENDENTE

- Profundidad: 1, 2, 7, 8, 3, 9, 4, 10, 11, 12, 5, 13, 14, 6, 15, 16, 17.
- Anchura: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17.

• DISEÑO DE CASOS DE PRUEBAS

#Caso de prueba	Componente	Descripción	Prerrequisitos
CP01	2 con 7	Se probará el caso de uso Iniciar sesión para verificar el funcionamiento de los métodos.	El usuario debe estar registrado para poder ingresar al servicio.
CP02	2 con 8	Se probará el caso de uso Registrar usuario para verificar el funcionamiento de los métodos.	
CP03	3 con 9	Se probará el caso de uso Verificar cuenta para verificar el funcionamiento de los métodos.	El usuario debe haber digitado correo y contraseña con los que se registró.

CP04	4 con 10	Se probará el caso de uso Crear solicitud personalizada para verificar el funcionamiento de los métodos.	
CP05	4 con 11	Se probará el caso de uso Modificar solicitud personalizada para verificar el funcionamiento de los métodos.	La solicitud debe estar registrada.
CP06	4 con 12	Se probará el caso de uso Eliminar solicitud personalizada para verificar el funcionamiento de los métodos.	La solicitud debe estar registrada.
CP07	5 con 13	Se probará el caso de uso Notificación de configuración para verificar el funcionamiento de los métodos.	Deben haber plantillas registradas para configurar la notificación.
CP08	5 con 14	Se probará el caso de uso Automatización de notificaciones para verificar el funcionamiento de los métodos.	Debe funcionar correctamente el caso de uso Notificación de configuración.
CP09	6 con 15	Se probará el caso de uso Crear plantilla para verificar el funcionamiento de los métodos.	
CP10	6 con 16	Se probará el caso de uso Modificar plantilla para verificar el funcionamiento de los métodos.	La plantilla debe estar registrada.
CP11	6 con 17	Se probará el caso de uso Eliminar plantilla para verificar el funcionamiento de los métodos.	La plantilla debe estar registrada.

Tabla 16: Diseño de casos de prueba. Elaboración propia

CP01					
Paso	Descripción	Datos entrada	Salida esperada	¿OK?	Observaciones
1	Se probará el funcionamiento del método Iniciar sesión.	1. Email address = "deiner@gmail.com" 2. Password = "Deiner.123"	Se ha ingresado correctamente.	OK	
2	Se probará el funcionamiento del método Registrar usuario.	1. Email address = "valen@gmail.com" 2. Password = "valen1234" 3. Account type = "Premium"	Se ha registrado correctamente.	OK	

Tabla 17: CP01. Elaboración propia

CP02					
Paso	Descripción	Datos entrada	Salida esperada	¿OK?	Observaciones
1	Se probará el funcionamiento del método Registrar solicitud personalizada.	1. Nombre = "Requisito 1" 2. Prioridad = "High" 3. Descripción = "Prueba de requisito uno" 4. Adjuntos = archivo.word	Se ha creado correctamente.	OK	
2	Se probará el funcionamiento del método Modificar solicitud personalizada.	1. Nombre = "Requisito Uno" 2. Prioridad = "Low" 3. Descripción = "Prueba de requisito uno." 4. Adjuntos = archivo.word 5. Estado = "Activo"	Se ha modificado correctamente.	OK	
3	Se probará el		Se ha eliminado	OK	

	funcionamiento del método Eliminar solicitud personalizada.		correctamente.		
--	---	--	----------------	--	--

Tabla 18: CP02. Elaboración propia

CP03					
Paso	Descripción	Datos entrada	Salida esperada	¿OK?	Observaciones
1	Se probará el funcionamiento del método Crear plantilla.	1. Nombre = "Prueba Plantilla Email asignación cuadrilla" 2. Tipo = "Email" 3. From = "bajaimenes@unicesar.edu.co" 4. Asunto = "PRUEBA ASIGNACIÓN CUADRILLA" 5. Archivo HTML = archivo.html 6. Archivos adjuntos = ninguno.	Se ha creado correctamente.	OK	
2	Se probará el funcionamiento del método Modificar plantilla.	1. Nombre = "Plantilla Email asignación cuadrilla" 2. Tipo = "Email" 3. From = "bajaimenes@unicesar.edu.co" 4. Asunto = "ASIGNACIÓN CUADRILLA" 5. Archivo HTML = archivo.html 6. Archivos adjuntos = ninguno.	Se ha modificado correctamente	OK	
3	Se probará el funcionamiento del método Eliminar		Se ha eliminado correctamente	OK	

	plantilla.				
--	------------	--	--	--	--

Tabla 19: CP03. Elaboración propia

CP04					
Paso	Descripción	Datos entrada	Salida esperada	¿OK?	Observaciones
1	Se probará el funcionamiento del método Notificación de configuración.	1. Template= "Plantilla Email asignación cuadrilla" 2. Tipo de contacto = "Específico" 3. Programado = "false" 4. Name= "Valentina Celedón" 5. Email = "valentinaceledon1@gmail.com" 6. Teléfono = "3004595618".	Se ha creado correctamente.	OK	
2	Se probará el funcionamiento del método Automatización de notificaciones		Se ha enviado correctamente la notificación.	OK	

Tabla 20: CP04. Elaboración propia

3.2 ANÁLISIS DE RESULTADOS Y DISCUSIÓN.

Se presentan y se examinan de manera crítica los resultados obtenidos durante el desarrollo e implementación del proyecto. Se determina el cumplimiento de los requerimientos funcionales del proyecto, teniendo como resultado el logro total y funcional de los requerimientos, desde la personalización de plantillas de las notificaciones para adaptarse a las necesidades y preferencias individuales de las empresas del sector comercial, dando como resultado la personalización efectiva de plantillas gracias a que proporcionan mensajes relevantes y específicos para cada usuario, hasta la configuración de las notificaciones como los tiempos de envío, canales, etc.

Se analiza el rendimiento del servicio en términos de velocidad de entrega, confiabilidad y escalabilidad teniendo en cuenta los requisitos no funcionales especificados en la documentación, obteniendo como resultado que el servicio cumple con los requisitos de tiempo real y la capacidad de manejar un buen volumen de notificaciones.

07:03:02 07:03:11

Imagen 24: Lapso de entrega de notificación. Diferencia de 8 msg. Elaboración propia



Imagen 25: Comprobación de notificación recibida. Elaboración propia

Cumpliendo con la usabilidad, el sistema proporciona mensajes de éxito y error, según sea el caso, para indicar a los usuarios la satisfacción del procedimiento realizado. Obedeciendo con el requisito de la eficiencia, el servicio responde en menos de 5 segundos para mostrar cualquier información modificada y/o actualizada en la base de datos al usuario y en menos de 3 segundos para responder a una petición.

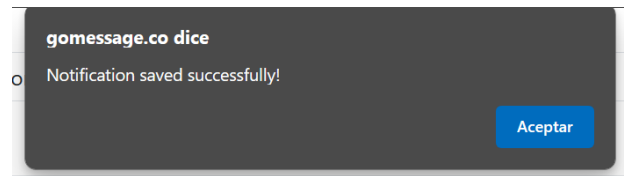


Imagen 26: Mensajes informativos. Elaboración propia

Cumpliendo con la portabilidad, el servicio obedece a la ejecución en diferentes plataformas web, como Microsoft Edge, y en distintos ambientes, como el celular.

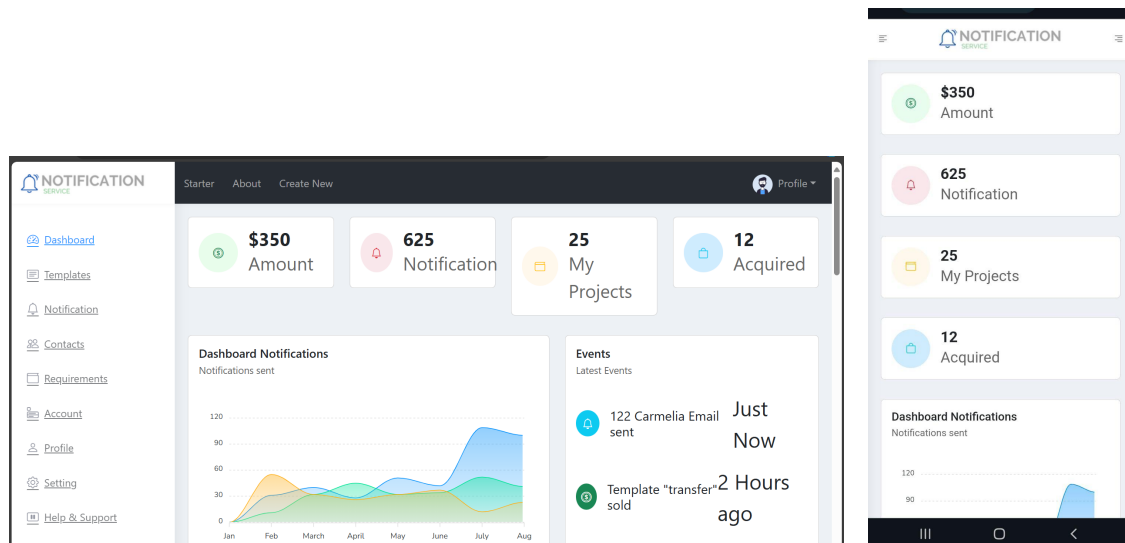


Imagen 27: Portabilidad. Derecha: Microsoft Edge. Izquierda: Chrome celular. Elaboración propia

Luego del desarrollo total del proyecto se reconocen los beneficios que brinda el uso de microservicios ya que esta arquitectura permite escalar cada componente de manera independiente según la demanda, así como también la flexibilidad y adaptabilidad que ofrece implementar la arquitectura de microservicios dirigida a eventos puesto que permite

responder rápidamente a cambios y eventos en tiempo real, ejemplo de esto sería la implementación del módulo de requerimientos que en un principio no se tenía en cuenta para el desarrollo del proyecto, sin embargo, cuando surgió la necesidad de agregarlo no se vieron afectados los módulos desarrollados.

Implementar el servicio de notificaciones en la empresa AO, luego de que la encargada de Marketing y ventas comentara las diferentes dificultades que atraviesan al momento de enviar mensajes digitales o tener una comunicación cercana con el cliente, ofreció numerosas ventajas a la empresa, entre las cuales podemos mencionar la flexibilidad y agilidad en la automatización de mensajes; la creación de notificaciones programadas permite a los usuarios la automatización, lo que reduce la carga de trabajo manual y minimiza el riesgo de errores humanos, asegurando que los clientes reciban la información adecuada en el momento oportuno. Igualmente, la combinación de personalización, automatización y respuesta en tiempo real de los mensajes digitales mejoran significativamente la experiencia del cliente.

3.3. CONCLUSIONES

Los resultados obtenidos demuestran que se ha logrado implementar un sistema eficiente que cumple con las expectativas iniciales y el objetivo principal del proyecto “Implementar un servicio de notificaciones basado en arquitectura de microservicios dirigida por eventos para la automatización y personalización de mensajería digital, en empresas del sector comercial”.

Aplicar el marco de trabajo SCRUM nos facilitó una buena planeación de actividades, dividiendo estas en sprints de 15 días; lo que nos permitió entregar resultados y darle valor al sistema en diferentes fases del proyecto. SCRUM es un marco grande que trabaja con distintos conceptos, en el presente proyecto no se trabajan con todos estos, en ese sentido, no se vió la necesidad de realizar dailys o reuniones diarias (o también el sprint retrospective), ya que el desarrollo del proyecto lo llevamos a cabo dos personas y es un proyecto considerablemente pequeño, nos bastaba con la reunión de sprint planning donde se planificaba qué elementos del Product Backlog serian entregados durante el próximo Sprint.

La arquitectura implementada contribuyó al cumplimiento de los requerimientos no funcionales, como la fiabilidad, que entre sus características se encuentra la tolerancia a fallos. En el proyecto nos benefició de forma que si un servicio receptor no está disponible

en el momento, las notificaciones que envía el emisor quedan en cola en el bus de eventos esperando a ser procesadas cuando el servicio vuelva a estar disponible. De esta manera, los servicios no necesitan saber o depender de otros servicios.

En conclusión, este proyecto ha demostrado que una arquitectura de microservicios dirigida por eventos es una solución viable y efectiva para la automatización y personalización de la mensajería digital en el sector comercial. La adopción de esta tecnología puede ofrecer ventajas competitivas significativas, mejorando tanto la eficiencia operativa como la satisfacción del cliente.

3.4 RECOMENDACIONES

En el desarrollo de un proyecto, hacer una buena planeación es esencial ya que de esta depende el cumplimiento de los tiempos de entrega. Para conseguir una buena planeación se debe definir los objetivos de cada sprint y que cada uno de ellos entregue algo de valor para alcanzar el objetivo general del proyecto. Se debe evaluar la complejidad del proyecto y el tamaño del equipo para definir las herramientas a utilizar, ofrecidas por el marco de trabajo, es decir, es conveniente usar sólo aquellas herramientas que brindan un beneficio significativo al proyecto.

La arquitectura de microservicios dirigida por eventos es una metodología que descompone una aplicación en servicios pequeños, autónomos y especializados que se comunican entre sí mediante eventos. Aplicar esta arquitectura beneficia principalmente si en el proyecto se desea escalabilidad, flexibilidad y capacidad de adaptación rápida a los cambios. Cada microservicio puede escalarse de manera independiente según las necesidades de carga, además, existe la ventaja de que los sistemas que implementen esta arquitectura serán resilientes y tendrán mayor tolerancia a fallos, es decir, si un microservicio falla, los demás pueden seguir funcionando porque cada uno maneja sus errores de manera independiente, evitando que un fallo en un servicio afecte a toda la aplicación.

3.5 BIBLIOGRAFÍA

- [1] German. (2022, 21 diciembre). ¿Qué es la comunicación digital? CEI: Escuela de Diseño y Marketing. <https://cei.es/que-es-la-comunicacion-digital/>
- [2] Empresas de VALLEDUPAR: listado y beneficios para tu negocio. (s. f.). Directorio de Empresas - Directorio de empresas en Colombia. https://www.informacolombia.com/directorio-empresas/localidad_valledupar
- [3] Nuseir, M. T., Refae, G. A. E., Aljumah, A. I., Alshurideh, M., Urabi, S., & Kurdi, B. A.

- (2023). Digital Marketing Strategies and the Impact on Customer Experience: A Systematic Review. En Studies in computational intelligence (pp. 21-44). https://doi.org/10.1007/978-3-031-12382-5_2
- [4] De Las Casas, J. (2020, 17 diciembre). Cómo ha cambiado la comunicación entre empresas y consumidores. EXPANSION. <https://www.expansion.com/economia/2020/12/17/5fda76ed468aeb4f638b45ef.html>
 - [5] Hammond, M. (2023, 28 marzo). Qué es la comunicación con el cliente: definición, canales y estrategias. HubSpot. <https://blog.hubspot.es/service/guia-comunicacion-cliente>
 - [6] Da Silva, D. (2020, 26 noviembre). Omnicanal o multicanal: ¿Cuál es la mejor estrategia? Zendesk MX. <https://www.zendesk.com.mx/blog/estrategia-multicanal-omnicanal/>
 - [7] Kotler, P., & Armstrong, G. M. (2019). Principles of marketing.
 - [8] Peppers, D., & Rogers, M. (2016). Managing Customer Experience and Relationships: A Strategic Framework. John Wiley & Sons.
 - [9] Arquitectura dirigida por eventos. (s. f.). Lucidchart. <https://www.lucidchart.com/blog/es/arquitectura-dirigida-por-eventos>
 - [10] Panel, E. (2022, 23 marzo). 15 tips to help companies communicate with customers via SMS text. Forbes. <https://www.forbes.com/sites/forbescommunicationscouncil/2022/03/23/15-tips-to-help-companies-communicate-with-customers-via-sms-text/?sh=7cc01fe07950>
 - [11] Smith, P. R., & Taylor, J. (2004). Marketing Communications: An Integrated Approach. Kogan Page Publishers.
 - [12] Coppola, M. (2023, 21 enero). Qué es una plataforma digital, qué tipos existen y ejemplos. HubSpot. <https://blog.hubspot.es/website/que-es-plataforma-digital>.
 - [13] Sommerville, I. (2005). Ingeniería del software. Pearson Educación.
 - [14] KeepCoding, R. (2023, 12 diciembre). ¿Qué es una Arquitectura de Software? KeepCoding Bootcamps. <https://keepcoding.io/blog/arquitectura-de-software/>
 - [15] Huet, P. (2023, 13 abril). Arquitectura de software: Qué es y qué tipos existen. OpenWebinars.net. <https://openwebinars.net/blog/arquitectura-de-software-que-es-y-que-tipos-existen/>
 - [16] Fowler, M. (2017). Patterns of Enterprise Application Architecture. Createspace Independent Publishing Platform.

- [17] Arquitectura basada en eventos | IBM. (s. f.).
<https://www.ibm.com/es-es/topics/event-driven-architecture>
- [18] ¿Qué es la EDA? - Explicación sobre la arquitectura basada en eventos - AWS. (s. f.). Amazon Web Services, Inc. <https://aws.amazon.com/es/what-is/eda/>
- [19] ¿Qué son los microservicios? | IBM. (s. f.).
<https://www.ibm.com/mx-es/topics/microservices>
- [20] Newman, S. (2015). Building microservices. Oreilly & Associates Incorporated.
- [21] Satpathy, R. (2022, 29 septiembre). Salesforce Platform Events: Features, Types & Uses - CEPTEs. CEPTEs. <https://ceptes.com/platform-events-in-salesforce/>
- [22] ¿Qué son los microservicios? | AWS. (s. f.). Amazon Web Services, Inc. <https://aws.amazon.com/es/microservices/>
- [23] Richardson, C. (2018). Microservices patterns: With examples in Java. Manning Publications.
- [24] Apache Kafka. (s. f.). Apache Kafka. <https://kafka.apache.org/>
- [25] Robertus, J. (2022, 22 marzo). What is Apache Kafka used for? Axual. <https://axual.com/what-is-apache-kafka/>
- [26] ¿Qué es Kafka? - Explicación sobre Apache Kafka - AWS. (s. f.). Amazon Web Services, Inc. <https://aws.amazon.com/es/what-is/apache-kafka/>
- [27] Nico, & Nico. (2021, 8 enero). Interfaz de usuario: Qué es, Tipos de interfaces y más. . . | Workana. Glosario - Workana | El Glosario Workana Explica Terminología del Mundo Freelance, Conceptos Fundamentales del Marketing y los Negocios. <https://i.workana.com/glosario/interfaz-de-usuario/>
- [28] Raskin, J. (2000). The humane interface: New Directions for Designing Interactive Systems. Addison-Wesley Professional.
- [29] ¿Qué es una API REST? | IBM. (s. f.). Available:
<https://www.ibm.com/es-es/topics/rest-apis>
- [30] Ordóñez, J. (2021, 9 marzo). ▷ ¿Qué es una API REST? | Guía 【2023】 - Identó. Identó. Available:
https://www.idento.es/blog/desarrollo-web/que-es-una-api-rest/?_adin=01833301559
- [31] Priyak. (2021, 12 mayo). Basics of REST API! Numpy Ninja. <https://www.numpyninja.com/post/basics-of-rest-api>
- [32] Comunicación, R. (2022, 24 febrero). Pierre Lévy: “Aunque muchos no lo crean, ya éramos muy malos antes de que existiera internet”. Revista Comunicación.

- <https://comunicacion.gumilla.org/2022/02/24/pierre-levy-aunque-muchos-no-lo-crean-y-eramos-muy-malos-antes-de-que-existiera-internet/>
- [33] Diaz, R. R., & Cazorla, F. J. (s. f.). La mensajería digital y su importancia en la empresa del siglo XXI.
 - [34] Jervis, T. M. (2022, 19 agosto). Mensajería instantánea. Lifeder. <https://www.lifeder.com/mensajeria-instantanea/>
 - [35] Chcomley. (2024, 25 marzo). ¿Qué es Azure DevOps? - Azure DevOps. Microsoft Learn. <https://learn.microsoft.com/es-es/azure/devops/user-guide/what-is-azure-devops?view=azure-devops>
 - [36] Chaves, S. (2023, 6 septiembre). Azure DevOps, ¿qué es y para qué sirve? Formadores IT. <https://formadoresit.es/azure-devops-que-es-y-para-que-sirve/>
 - [37] Cruzito. (2020, 17 octubre). ¿Qué es Scrum? - Definición y terminología | Estudyando. <https://estudyando.com/que-es-scrum-definicion-y-terminologia/>
 - [38] Mesa, A. R. (2023, 12 mayo). Qué es Scrum: Origen y características. OpenWebinars.net. <https://openwebinars.net/blog/que-es-scrum/>
 - [39] Nonaka y Takeuchi - Praxis Framework. (s. f.). <https://www.praxisframework.org/es/library/nonaka-and-takeuchi>
 - [40] Sutherland, J., & Sutherland, J. (2014). Scrum: The Art of Doing Twice the Work in Half the Time. Crown Currency.
 - [41] ¿En qué consiste Scrum? - Explicación sobre la metodología Scrum - AWS. (s. f.). Amazon Web Services, Inc. <https://aws.amazon.com/es/what-is/scrum/>
 - [42] Harris, D. C. (s. f.). Obtén más información sobre los artefactos del scrum ágil. Atlassian. <https://www.atlassian.com/es/agile/scrum/artifacts>
 - [43] Scrum – DESIRE. (s. f.). <https://desire.webs.uvigo.es/contenidos/scrum/>

.MODELOS DE LOS ANEXOS DE LA PROPUESTA

Anexo A. Modelo Carta del director del proyecto

Valledupar, 28/05/2024.

Señores:

COMITÉ DE INVESTIGACIÓN

Facultad de Ingenierías y Tecnológicas

Programa de Ingeniería de Sistemas

Universidad Popular Del Cesar

Cordial saludo

Yo **Salomón Cadena**, identificado con la cédula de ciudadanía No. **7571320**, certifico que he revisado el documento correspondiente al proyecto que lleva por título **“Servicio de notificaciones basado en arquitectura de microservicios dirigida por eventos para la automatización y personalización de mensajería digital, en empresas del sector comercial.”**, presentada por los estudiantes **Valentina Celedón Lozano** y **Deyner Eduardo Solano Tafur**, y, después de haberle realizado las respectivas correcciones, cuenta con mi aprobación para ser presentada ante el comité. Sugiero la aprobación por parte de ustedes.

Línea de investigación: Tecnologías de la Información y la comunicación

Sublínea: Ingeniería de software

Agradezco la atención prestada

Atentamente,



Salomón Cadena

CC 7571320 de Valledupar

Director de Proyecto

Anexo B. Modelo Carta de los estudiantes

Valledupar, 28/05/2024.

Señores:

COMITÉ DE INVESTIGACIÓN

Facultad de Ingenierías y Tecnológicas

Programa de Ingeniería de Sistemas

Universidad Popular Del Cesar

Cordial saludo

Nosotros los abajo firmantes, estudiantes del programa de Ingeniería de sistemas, presentamos a ustedes el documento correspondiente al proyecto de grado denominado **“Servicio de notificaciones basado en arquitectura de microservicios dirigida por eventos para la automatización y personalización de mensajería digital, en empresas del sector comercial”**.

Quedamos a la espera del concepto emitido por el comité respecto de la viabilidad y aceptación de dicha propuesta.

Agradecemos la atención prestada

Atentamente,

Valentina Celedón L.

Valentina Celedón

CC. 1003266012 de La Jagua de Ibirico

Deyner Solano

Deyner Solano

CC. 1003380104 de Valledupar