



**Creación de sistema móvil para la publicación de ventas y compras de repuestos  
de motocicleta**

**John Edison Gamboa Guzmán**

**Carlos Andrés Garzón**

**Andrés Camilo Santacruz Borda**

**Universidad Antonio Nariño**

Programa Especialización en ingeniería de software

Facultad de Ingeniería de Sistemas

Bogotá, Colombia

2023

**Creación de sistema móvil para la publicación de ventas y compras de repuestos  
de motocicleta**

**John Edison Gamboa Guzman**

**Carlos Andres Garzón**

**Andres Camilo Santacruz Borda**

Proyecto de grado presentado como requisito parcial para optar al título de:  
**Especialista en ingeniería del software**

Director (a):

Neme Prada Dianalin

**Universidad Antonio Nariño**

Especialización en ingeniería del software

Facultad de Ingeniería de Sistemas

Bogotá, Colombia

2023

## NOTA DE ACEPTACIÓN

El trabajo de grado titulado

\_\_\_\_\_.

Cumple con los requisitos para optar

Al título de \_\_\_\_\_.

Firma del Tutor

Firma Jurado

Firma Jurado

# TABLA DE CONTENIDOS

Tabla de contenido

<b>RESUMEN</b> .....	<b>13</b>
<b>ABSTRACT</b> .....	<b>15</b>
<b>INTRODUCCIÓN</b> .....	<b>17</b>
<b>1. FORMULACIÓN Y DESCRIPCIÓN DEL PROBLEMA</b> .....	<b>19</b>
1.1. FORMULACIÓN DEL PROBLEMA .....	19
1.2. DESCRIPCIÓN DEL PROBLEMA .....	19
<b>2. OBJETIVOS</b> .....	<b>23</b>
2.1. GENERAL .....	23
2.2. ESPECÍFICOS.....	23
<b>3. MARCO DE REFERENCIA</b> .....	<b>24</b>
3.1. ESTADO DEL ARTE .....	26
3.2. IMPACTO .....	28
3.3. COMPONENTE DE INNOVACIÓN .....	29
<b>3.4. MARCO TEÓRICO</b> .....	<b>30</b>
<b>4. METODOLOGÍA</b> .....	<b>47</b>
<b>5. PROCESO DE SOFTWARE</b> .....	<b>51</b>
5.1. REQUERIMIENTOS FUNCIONALES .....	51
5.2. HISTORIAS DE USUARIO (REQUERIMIENTOS NO FUNCIONALES).....	77
5.3. DIAGRAMA DE SECUENCIA .....	87
<b>7. PRUEBAS</b> .....	<b>106</b>
<b>8. CONFIGURACIÓN E INSTALACIÓN</b> .....	<b>116</b>

## LISTA DE FIGURAS

Ilustración 1	Componente de innovación basado en matriz RICE .....	29
Ilustración 2	Tablero KANBAN implementado en el proyecto .....	48
Ilustración 3	Repositorio Confluence usado para persistir el conocimiento relacionado al proyecto.....	50
Ilustración 4	Diagrama de contexto .....	78
Ilustración 5	Diagrama de contenedores.....	79
Ilustración 6	Diagrama de componentes.....	80
Ilustración 7	Diagrama de despliegue.....	84
Ilustración 8	Diagrama de caso de uso arquitecturalmente relevante.....	87
Ilustración 9	Diagrama de secuencia comprador .....	88
Ilustración 10	Diagrama de secuencia del vendedor.....	89
Ilustración 11	Diagrama de Clases .....	90
Ilustración 12	Instancia de la base de datos en Google Cloud.....	92
Ilustración 13	Ramas establecidas para el proceso de versionamiento de código.....	93
Ilustración 14	Proceso de fusión entre ramas y promoción al ambiente productivo .....	94
Ilustración 15	Convenciones.....	95
Ilustración 16	Capas implementadas a nivel de código .....	96
Ilustración 17	Configuración del proyecto en Firebase .....	98
Ilustración 18	Front-End compilado en web.....	99
Ilustración 19	Guards implementados .....	100
Ilustración 20	Xcode en proceso de compilación para iOS .....	101
Ilustración 21	Pipeline en GitHub Actions para análisis con SonarCloud .....	103
Ilustración 22	Pipeline en GitHub Actions para despliegue en Google Cloud.....	104
Ilustración 23	Microservicios desplegados en CloudRun.....	104
Ilustración 24	Pruebas unitarias en Back-End .....	106
Ilustración 25	Pruebas unitarias compiladas en local .....	106
Ilustración 26	Resultado de analisis de SonarCloud.....	107

## LISTA DE TABLAS

Tabla 1 Relación de estado del arte .....	26
Tabla 2 Diferentes tipos de arquitecturas de software .....	40
Tabla 3 Historia de usuario TR-00.....	53
Tabla 4 Historia de usuario TR-002.....	54
Tabla 5 Historia de usuario TR-003.....	56
Tabla 6 Historia de usuario TR-004.....	58
Tabla 7 Historia de usuario TR-005.....	60
Tabla 8 Historia de usuario TR-006.....	62
Tabla 9 Historia de usuario TR-007.....	64
Tabla 10 Historia de usuario TR-008.....	66
Tabla 11 Historia de usuario TR-009.....	67
Tabla 12 Historia de usuario TR-010.....	68
Tabla 13 Historia de usuario TR-011.....	69
Tabla 14 Historia de usuario TR-012.....	70
Tabla 15 Historia de usuario TR-013.....	71
Tabla 16 Historia de usuario TR-014.....	73
Tabla 17 Historia de usuario TR-015.....	75
Tabla 18 Requerimientos no funcionales.....	77
Tabla 19 Prueba funcional 1 .....	109
Tabla 20 Prueba funcional 2 .....	111

*(Dedicatoria)*

*Su uso es opcional y cada autor podrá determinar la distribución del texto en la página, se sugiere esta presentación. En ella el autor del trabajo dedica su trabajo en forma especial a personas y/o entidades.*

*Por ejemplo:*

*A mis padres*

*o*

*La preocupación por el hombre y su destino siempre debe ser el interés primordial de todo esfuerzo técnico. Nunca olvides esto entre tus diagramas y ecuaciones.*

## RESUMEN

En la ciudad de Bogotá D.C de Colombia, el mercado de motocicletas crece día a día por diferentes causas, entre ellas el crecimiento desordenado de la ciudad; lo que produce ubicaciones laborales lejanas a los sitios de residencia de los trabajadores y por ende, dificultades para acceder al sistema de transporte público, tráfico dispendioso entre otros.

Es así, que es posible identificar diferentes dificultades que surgen a los usuarios de motocicletas en la ciudad, como por ejemplo, el mantenimiento de estos automotores. Dicho esto, también se identifica la oportunidad de dar una solución a corto plazo a través de una propuesta de un sistema móvil para la publicación de ventas y compras de repuestos de motocicleta, basada en modelos existentes similares que permiten hacer publicaciones con una mejor capacidad de conexión entre los usuarios del sistema móvil; quienes serán compradores y vendedores conservando la identidad de un Mercado tradicional y generando un símil a un bazar de remates de manera Online.

De acuerdo a lo anterior, surge el siguiente proyecto “**Creación de sistema móvil para la publicación de ventas y compras de repuestos de motocicleta**”, que tiene como propósito conectar demandantes y ofertantes, haciendo una conexión en tiempo real de dos usuarios que se desconocen, pero que los vincula el entorno de las motocicletas en todos sus aspectos, usuario, coleccionista, reparador o restaurador, entre otros.

Detrás de este proyecto se involucra un equipo de desarrolladores de software que busca, a través del desarrollo de un aplicativo soportado en tecnologías recientes y enfoques modernos de arquitectura y buenas prácticas de desarrollo, satisfacer las necesidades del

mercado objetivo de los usuarios ya mencionados que requieren diferentes artículos para sus vehículos.

El alcance de dicho proyecto es preestablecer un asistente de búsqueda para conectar demandantes y ofertantes de artículos, determinando un filtro por ubicación y/o precio y dejar por el momento en libertad el método transaccional de pago para que este sea efectuado de manera directa sin ayuda del sistema móvil entre los involucrados. Adicionalmente la validación de identidades no será ejecutada para esta primera versión del software teniendo en cuenta las limitantes de tiempo .

Palabras claves: Repuestos de motocicletas, accesorios de motocicletas, modelos B2B y B2C, Microservicios, Integración y despliegue continuo, Firebase, Google Cloud, Angular, Ionic, NodeJS, Express, GitHub Actions, oferta, demanda, rango de búsqueda.

## ABSTRACT

In the city of Bogotá D.C., Colombia, the motorcycle market is growing day by day for different reasons, among them the disorderly growth of the city, which produces job locations far from the workers' places of residence and therefore, difficulties to access the public transportation system, wasteful traffic, among others.

Thus, it is possible to identify different difficulties that arise for motorcycle users in the city, such as, for example, the maintenance of these vehicles. Having said this, it is also identified the opportunity to provide a short-term solution through a proposal for a mobile system for the publication of sales and purchases of motorcycle parts, based on similar existing models that allow publications with a better connection capacity between users of the mobile system; who will be buyers and sellers while retaining the identity of a traditional market and generating a simile to a bazaar of auctions in an online way. According to the above, the following project "Creation of a mobile system for the publication of sales and purchases of motorcycle parts", which aims to connect buyers and sellers, making a real-time connection of two users who do not know each other, but that links the environment of motorcycles in all its aspects, user, collector, repairer or restorer among others.

Behind this project is involved a team of software developers that seeks, through the development of an application supported by recent technologies and modern approaches to architecture and good development practices, to meet the needs of the target market of users already mentioned that require different items for their vehicles. The scope of this project is to pre-establish a search wizard to connect buyers and sellers of items, determining a filter by location and/or price and leaving free for the moment the

transactional method of payment to be made directly without the help of the mobile system between those involved. Additionally, identity validation will not be performed for this first version of the software due to time constraints.

Keywords: motorcycle parts, motorcycle accessories, B2B and B2C models, Microservices, Continuous integration and deployment, Firebase, Google Cloud, Angular, Ionic, NodeJS, Express, GitHub Actions, supply, demand, search rank.

## INTRODUCCIÓN

En la actualidad el desarrollo de software se ha convertido en una disciplina que presenta un ciclo recurrente de mejora continua, día a día son más los consumidores de software debido a las necesidades de un mundo que presenta una revolución conocida como “Cuarta Revolución Industrial”. Todo esto trae como consecuencia retos de los cuales no hay antecedentes y problemas que necesitan un trabajo colaborativo de varias disciplinas en relación con el campo de la tecnología, cómo lo son la Arquitectura de Software, Arquitectura de Infraestructura, Especialista en Ciberseguridad, Especialista en Analítica de datos entre otros.

El presente documento expone el proceso de creación de un software que tiene cómo objetivo conectar ofertas con demandas en el mercado objetivo de usuarios de motocicletas, añadiendo funcionalidades destacadas de software similares y populares en el mercado como UBER, DiDi e inDrive. Nombrando algunas funcionalidades, se destaca el registro de usuario validando plataformas cómo google, búsquedas de usuarios por rango de posición geográfica y un asistente de búsqueda para orientar al usuario.

Para lograr esto se utilizará una metodología de trabajo dentro del marco de la agilidad, abordando todas las etapas del ciclo de vida del software; desde ingeniería de requerimientos, pasando por el diseño, implementación, pruebas y mantenimiento.

El alcance del proyecto descrito en este documento tiene aplicación desde la identificación de la necesidad de un insumo o accesorio de los usuarios de motocicleta hasta conectar el Usuario que demanda (Comprador) con el usuario que oferta (Vendedor) excluyendo la posible compra de los actores involucrados.

Este proyecto busca tener impacto comercial en el sector de tránsito y transporte ya que conectará diferentes actores como lo son pequeños, medianos y grandes comerciantes con pequeños, medianos y grandes consumidores de manera digital y con una eficiencia y eficacia mayor a la convencional del voz a voz o las búsquedas manuales realizadas por internet.

# 1. FORMULACIÓN Y DESCRIPCIÓN DEL PROBLEMA

## 1.1. FORMULACIÓN DEL PROBLEMA

Partiendo del hecho de que en Colombia no existe actualmente una herramienta digital que permita establecer un canal de comunicación directo e inmediato entre compradores y pequeños distribuidores de repuestos y accesorios de motocicleta, surge la siguiente interrogante:

¿Es posible desarrollar un software de oferta y demanda, al estilo de compañías de alquiler de vehículos y que funcione bajo los modelos Bussines to Bussines (B2B) que traduce al castellano negocio a negocio y Bussines to consumer (B2C) que traduce al castellano negocio a consumidor, que establezca un canal de comunicación directo e inmediato (que envíe notificaciones instantáneamente) entre compradores y vendedores de repuestos y accesorios de motocicleta?

## 1.2. DESCRIPCIÓN DEL PROBLEMA

En los últimos años, las ventas de motocicletas nuevas y usadas ha aumentado considerablemente; por ejemplo, según el periodico (País, 2022) solo en abril de ese año, en Colombia se matricularon cerca de 68.603 motocicletas nuevas y que las proyecciones de la Asociación Nacional de Movilidad Sostenible ADEMOS para dicho año, prevén que el mercado de motos superará las 900.000 unidades.

Del mismo modo, en una entrevista otorgada por el director comercial de la compañía fabricante y ensambladora de motocicletas AKT al periódico La República (Urrego, 2022),

él concluye que actualmente se evidencia un comportamiento positivo en la demanda y oferta de repuestos y accesorios de motocicleta incluso llegando a ser proporcional a las cifras de venta de estos vehículos.

Para la compra de estos repuestos y accesorios, los usuarios tienen diferentes alternativas que van desde acudir a las tiendas de los fabricantes, movilizarse a los sectores de ventas de repuestos como la avenida 1 de mayo en Bogotá o realizar búsquedas en internet en donde algunos vendedores tienen páginas web o publican a través de plataformas como Mercado Libre y OLX.

Sin embargo, de estas modalidades de compra disponibles, se desprenden diferentes situaciones que pueden ser abordadas en una misma solución de manera integral y eficaz. Por ejemplo, en el caso de los repuestos y accesorios que son vendidos directamente en locales de comercio, sin importar su ubicación, es necesario que los compradores en la mayoría de casos se desplacen hasta allí a buscar uno por uno, realizar llamadas o mensajes a través de plataformas como WhatsApp y en caso de no encontrarlos con facilidad, deben recurrir al voz a voz. Para el caso de plataformas como Mercado Libre, OLX y aplicativos web propios de proveedores, uno de los principales obstáculos es el contacto directo, ya que se está limitado en muchos casos a mensajes de texto que no son leídos ni resueltos de manera inmediata, impactando en muchos casos la necesidad de acceso a los repuestos de manera oportuna e incluso urgente.

Ahora, desde la perspectiva de los pequeños distribuidores, la captación y seducción de posibles clientes para la formalización de la compra, está limitado a que sean visitados en sus tiendas; si es que las tienen, o les contacten a través de las plataformas ya mencionadas.

Es por ello, que se desarrollará un software que permita la interacción directa e instantánea de compradores y vendedores a fin de que los compradores puedan acceder a diferentes ofertas en el momento de su necesidad y puedan elegir la que más le convenga en un criterio de distancia y precio y los vendedores puedan tener acceso a potenciales clientes y sus solicitudes de manera oportuna impulsando así la competencia del mercado y aportando indirectamente a la creación de una red de ayuda para los usuarios motociclistas junto con la disminución de las situaciones anteriormente mencionadas (Desplazamientos innecesarios, necesidad del voz a voz o tiempos de espera entre solicitudes y respuestas por medio de plataformas digitales actuales).

El software permitirá realizar el registro e ingreso a través de Google y Facebook y un switch entre roles, ya que en cualquier momento un vendedor puede también comprar y viceversa . Específicamente, permitirá a los posibles compradores buscar por medio de un asistente la pieza que están buscando y su solicitud será notificada a los vendedores que estén disponibles en el perímetro previamente configurado por el comprador. Esta notificación llegará inmediatamente a los vendedores y todos aquellos que consideren que pueden realizar una oferta podrán indicar que si tienen el repuesto o accesorio solicitado y además podrán agregar una breve descripción o comentario y el valor que ofertan. Los

compradores, a medida que los vendedores realicen su oferta, podrán visualizar una lista con todas las opciones disponibles y realizar ordenamiento por distancia o precio de manera ascendente y descendente. En cuanto a seguridad, el software permitirá calificar y visualizar en cualquier momento dicha calificación de manera bidireccional; es decir, compradores podrán calificar a vendedores y del mismo modo visualizar sus calificaciones y comentarios y viceversa.

## **2. OBJETIVOS**

### **2.1. GENERAL**

Desarrollar un aplicativo móvil (App) con enfoque B2B y B2C por medio del cual se establecerá un canal de comunicación entre vendedores de repuestos y accesorios para motos y los potenciales compradores de estos; contribuyendo con este a la centralización y digitalización de la oferta y demanda de repuestos para motocicletas en Bogotá.

### **2.2. ESPECÍFICOS**

- Permitir por una parte, a los compradores construir la compra con los repuestos y accesorios de su necesidad y, por otra parte, a los vendedores, construir la oferta a modo de respuesta a las solicitudes hechas por compradores con los productos que tengan a su disposición para la venta.
- Establecer un canal de comunicación entre vendedores y compradores, contribuyendo así en la construcción de una experiencia diferente en la búsqueda y venta de repuestos y accesorios para motocicletas en Bogotá.
- Lograr por medio de sistema de calificación bidireccional (compradores y vendedores) una forma confiable de hacer compras y ventas por medios digitales.
- Permitir elegir a los compradores la oferta que más le convenga según criterios de precio o distancia al vendedor.

### **3. MARCO DE REFERENCIA**

Desde el año 1969 momento en el cual fue lanzado el internet al mercado como medio de comunicación se ha evidenciado un crecimiento en las herramientas digitales que se han desarrollado. Por la línea del tiempo y la innovación de internet se ha visto pasar y quedarse los servicios de mensajería instantánea, conferencias en video, publicación de contenido, educación, ocio, entre otros. Esta gran red conocida como WWW world wide web se ha encargado de acortar las distancias. Hoy en día se cuenta con servicio en línea como lo es pagos de servicios públicos u otros, compra de artículos, compra de entradas a espectáculos, mapa satelital, educación, entre otros, todo esto a la distancia de un clic. No obstante, nuevas herramientas tangibles cómo el celular y otros están permitiendo que la famosa WWW siga creciendo y mostrándose más asequible a nuevos públicos, permitiendo la comunicación y educación de manera horizontal.

Resulta, entonces, que el desarrollo de tecnologías y herramientas digitales se convierte en una necesidad para la solución de problemas del mundo moderno (Pareja, 2021). Desde la invención de la rueda por parte del humano, los desarrollos e innovaciones no han dejado de surgir, la revolución industrial, la medicina, los medios de transporte, entre otros, requieren puesta en práctica de nuevas tecnologías como por ejemplo, plataformas digitales (software) que permite gestionar diferentes propósitos y alcances en tareas rutinarias.

Es una constante la reinención en la evolución de las herramientas usadas, siempre se está en la búsqueda de facilitar tareas. Pasaron muchos años para que el convencional

método de comunicación que era la carta física y tangible se convirtiera en lo que hoy conocemos cómo correo electrónico y que es capaz de viajar alrededor del mundo en cuestión de segundos.

De acuerdo con los indicadores publicados por el Ministerio de Tecnologías de la Información y las Comunicaciones, el Departamento Administrativo Nacional de Estadística (DANE) ha informado sobre un logro histórico en la expansión de la conectividad a internet de los ciudadanos colombianos durante el año 2020, habiéndose alcanzado un progreso del 56,7% respecto a la meta establecida. Se mencionan estadísticas que evidencian el aumento en el uso de internet por parte de los colombianos, no solo por las nuevas generaciones, y que la pandemia del Covid-19 ha obligado a muchas empresas a implementar el teletrabajo como medida para mitigar y controlar el problema de salud pública.

### 3.1. ESTADO DEL ARTE

Luego de realizar una investigación relacionada con desarrollos previos a este proyecto, se presenta el estado del arte, en el cual se mencionan productos y/o servicios que tienen similitud con la solución propuesta para la problemática mencionada anteriormente. Los resultados de dicha investigación son citados textualmente en una tabla que se presenta a continuación.

Tabla 1 Relación de estado del arte

<b>Descripción general</b>	<b>Resumen</b>	<b>Evaluación</b>
Mercado libre.com.co 	Plataforma digital con el propósito de permitir comerciar artículos de diferentes categorías para la posterior adquisición de usuarios compradores.	Herramienta tecnológica con 21 años de trayectoria, con presencia en 18 países que ha integrado comercio a diferentes niveles y categorías con modalidad de pago en línea.
Amazon.com.co 	Plataforma digital con el propósito de permitir comerciar artículos de diferentes categorías para la posterior adquisición de usuarios compradores.	Herramienta tecnológica con 25 años de trayectoria, con presencia mundial que ha integrado comercio a diferentes niveles y categorías, con modalidad de pago en línea.

<p>Motorepuestos.com.co</p> 	<p>Plataforma digital con el propósito de vender repuestos de motocicletas de un único proveedor.</p>	<p>Herramienta tecnológica con presencia y operaciones en Colombia con modalidad de pago en línea.</p>
<p>Mundimotos.com</p> 	<p>Plataforma digital con el propósito de vender repuestos de motocicletas de un único proveedor.</p>	<p>Herramienta tecnológica con presencia y operaciones en Colombia con modalidad de pago en línea.</p>
<p>Motomax.com.co</p> 	<p>Plataforma digital con el propósito de vender repuestos de motocicletas de un único proveedor.</p>	<p>Herramienta tecnológica con presencia y operaciones en Colombia con modalidad de pago en línea.</p>

Nota: Elaboración propia (2023)

Las herramientas analizadas, si bien son funcionales, no logran ofrecer una experiencia comparable a la proporcionada por el proyecto desarrollado por los siguientes motivos:

1. En el caso de Mercado Libre, aunque el vendedor puede publicar su stock, no existe un canal de comunicación directa que permita al comprador expresar completamente sus necesidades y que diferentes vendedores puedan responder a esas necesidades de manera integral.
2. Amazon, por su parte, opera a nivel internacional, lo que implica que los tiempos de entrega pueden ser más prolongados en comparación con una plataforma local.
3. Motorepuestos.com.co, Mundimotos.com y Motomax.com.co, muestran un portafolio de productos que puede no abarcar todos los elementos que constituyen la

necesidad del comprador. Esto implica que el comprador tendría que visitar varias páginas y realizar varias compras para satisfacer completamente su necesidad, lo que resulta en un proceso repetitivo y poco eficiente.

Estos aspectos serán profundizados en el capítulo "Componente de innovación" del proyecto desarrollado.

### 3.2. IMPACTO

Con la puesta en marcha del software, se espera vincular oferta y demanda, conectando pequeños comercios físicos y compradores de repuestos y accesorios de motocicletas, estableciendo un canal de comunicación de respuesta inmediata que permita negociar diferentes repuestos y servicios. Adicionalmente, también se espera reducir costos debido a desplazamientos innecesarios por parte de los compradores.

Paralelamente, también se espera que el uso del software contribuya en la consolidación de una red local de colaboración entre los usuarios de motocicletas, brindando una alternativa cómoda e innovadora al proceso de venta y adquisición de repuestos y accesorios de motocicletas.

### 3.3. COMPONENTE DE INNOVACIÓN

Ilustración 1 Componente de innovación basado en matriz RICE



Nota: Elaboración propia (2023)

### 3.4. MARCO TEÓRICO

Dadas las problemáticas evidenciadas tangiblemente como ciudadanos de Bogotá - Colombia se aborda la oportunidad del desarrollo de una herramienta tecnológica. Basados en referencia del medio de comunicación Diario “La República” en su titular “**Venta de motocicletas al cierre de julio superó en 349.011 unidades a las de carros**” (Núñez, 2022) en el cual se mencionan indicadores que afirman que en promedio al día se venden en Colombia 2.340 motos y la tasa de crecimiento de este parque automotor es del 37% en relación con el año antecesor. Menciona el artículo que lo evidenciado es histórico, ya que relacionando y contrastando los datos con el fenómeno de la pandemia en la cual el aislamiento obligó a guardar los vehículos, este año ha sido el momento de adquirir nuevamente dichos bienes. Este artículo periodístico señala que la razón por la cual las motocicletas tienen más demanda que los autos en Colombia se debe a su accesibilidad como medio de transporte y ha sido impulsada por los problemas de tráfico en el país y la baja calidad del servicio de transporte público. Sin embargo, al adquirir estas motocicletas, se adquieren obligaciones como la compra del Seguro Obligatorio contra Accidentes de Tránsito (SOAT), la revisión técnica mecánica, el mantenimiento preventivo que incluye cambios de piezas consumibles, y el mantenimiento correctivo, como la reparación de piezas rotas.

Como modelo de negocio se resalta la propuesta B2B que por sus siglas en lenguaje anglo son Business to business, lo cual traduce al castellano negocio a negocio e indica que se generan operaciones comerciales directas entre empresas, en las que no intervienen los

consumidores finales. Por otro lado está el modelo B2C que por sus siglas en lenguaje anglo son business to consumer, lo cual traduce al castellano negocio a consumidor. En este modelo se incluye directamente al consumidor final (Juan, 2016).

**Kanban:** Como marco metodológico toma el concepto justo a tiempo (JIT) (Japan Management Association, 2018) y facilita la gestión de cambio de manera sencilla y visual en donde los integrantes del equipo deciden cuándo y cuánto trabajo realiza, minimizando así, las sobrecargas de actividades en los integrantes.

Algunas de las principales características de KANBAN se detallan a continuación:

1. Provee un mecanismo para la visualización del flujo de trabajo (workflow) ya que divide las diferentes actividades en tareas más pequeñas y específicas que a su vez son representadas por medio de tarjetas que se van moviendo a través de diferentes columnas que ilustran el estado o etapa del proceso en que se encuentran (Bustínduy, 2021).
2. Desde el principio, establece acuerdos de procesos y actividades como por ejemplo:
  - a. Qué integrante del equipo es el responsable de mover las actividades del tablero.
  - b. Periodicidad y horario de las reuniones de seguimiento.
  - c. Estructura y detalle de la información que se registra en cada tarjeta.
  - d. Clasificación de las actividades o tareas.

3. Con base en los acuerdos del equipo, limita la cantidad de actividades que pueden ser realizadas a la vez en cada columna del workflow.

KANBAN, no define roles en el equipo, sino que, por el contrario, brinda la libertad de que el equipo defina los roles que considere relevantes, aunque recomienda que se tengan en cuenta ciertas responsabilidades a fin de promover la correcta ejecución del proyecto (Patil, 2021). Para la gestión de dichas responsabilidades, se tienen en cuenta los siguientes roles:

- **Service Request Manager:** Encargado de gestionar los requerimientos y manejar las relaciones con los stakeholders asemejándose al rol de product owner en SCRUM.
- **Service Delivery Manager:** Sus responsabilidades están asociadas al facilitamiento de las diferentes reuniones que se acuerden junto con actividades de mejora continua asemejándose al rol de SCRUM MASTER.

**JIRA:** Como herramienta Web con una capa gratuita, ayuda a que los equipos trabajen juntos de manera altamente productiva, ya que permite el trabajo colaborativo. Es comúnmente conocido por el impacto generado en las áreas de Gestión de proyectos, gestión de tareas, entre otros, y es usado especialmente en el desarrollo de software porque ayuda a describir por pasos de trabajo el proyecto.

JIRA se ofrece como herramienta de ayuda visual que optimiza el flujo de trabajo ya que es usada por administradores y colaboradores .

Esta herramienta permite plasmar el tablero KANBAN para crear tickets que son asignados de manera individual a los integrantes del grupo de trabajo. Es importante mencionar que JIRA trabaja sobre la Web y funciona indistintamente del sistema operativo instalado en la computadora (ionos - Desarrollo web, 2022).

**Ventajas:**

1. Permite trabajo asincrónico.
2. Los progresos se muestran en tiempo real.
3. Respuestas sobre el ticket (TAREA).
4. Priorización de tareas.
5. Copia de seguridad de los datos.
6. Agregar fecha de vencimiento a las tareas.
7. Se puede utilizar en diferentes dispositivos (computadora, tablet, móvil, entre otros).

**CONFLUENCE:** Como herramienta Web gratuita ofrece un espacio de trabajo donde se consigna conocimiento de manera organizada en forma de repositorio. Agregando a esto es una plataforma colaborativa en tiempo real que permite a un equipo de trabajo realizar aportes para consolidar información.

Confluence permite organizar contenidos del proyecto a desarrollar, crear tantos como se necesiten en el equipo, se pueden personalizar con diferentes tipos de ilustraciones, entre otros. Existen varias plantillas en Confluence, pero también se puede partir desde ceros para una creación más personalizada (Torres, 2020).

### **Ventajas:**

1. Permite trabajo asincrónico.
2. Los progresos se muestran en tiempo real.
3. Copia de seguridad de los datos.
4. Generar restricciones personalizadas.
5. Destacar documentos y clasificarlos.
6. Los cambios se guardan y sincronizan automáticamente.
7. Agregar fecha de vencimiento a las tareas.

**Angular:** En pocas palabras, Angular es un conjunto de herramientas de código abierto que se utilizan para desarrollar aplicaciones web de manera rápida y eficiente, incluyendo aplicaciones de una sola página (SPA) y aplicaciones web progresivas (PWA). Este conjunto de herramientas fue creado por Google y está basado en el lenguaje de programación TypeScript y en el patrón de diseño Modelo-Vista-Controlador (MVC) (Google, 2023).

Entre las ventajas de usar Angular para la creación de aplicaciones híbridas se encuentran las siguientes:

1. **Eficacia en el rendimiento:** Angular utiliza un enfoque basado en componentes y un sistema de detección de cambios que optimiza el

rendimiento de la aplicación. Esto lo hace especialmente adecuado para aplicaciones web complejas y de alta carga.

2. **Alta escalabilidad:** Angular es un framework que se basa en la utilización de componentes y cuenta con un sistema de detección de cambios que ayuda a mejorar el rendimiento de las aplicaciones web. Gracias a estas características, es especialmente útil para el desarrollo de aplicaciones web complejas y que requieren un alto rendimiento.
3. **Integración con otras tecnologías:** Angular se puede integrar fácilmente con otras tecnologías, como TypeScript, RxJS y Firebase, lo que lo hace muy flexible y fácil de usar (Yakov Fain, 2018).
4. **Gran comunidad y soporte:** Angular cuenta con una amplia comunidad de desarrolladores y usuarios que aportan nuevas características, soluciones a problemas y mejoras constantemente. Además, Google ofrece un amplio soporte técnico y una documentación detallada para ayudar a los desarrolladores a utilizar y mejorar el framework de manera efectiva.

En comparación con otros frameworks para la creación de aplicaciones híbridas, como React Native, Angular tiene una sintaxis más clara y organizada, lo que facilita la legibilidad y el mantenimiento del código. Además, su arquitectura modular y escalable lo hace más adecuado para aplicaciones de mayor tamaño y complejidad.

**Ionic:** Se trata de un conjunto de herramientas de código abierto diseñado para el desarrollo de aplicaciones móviles híbridas, el cual se basa en tecnologías web como HTML, CSS y JavaScript. Con este framework, los desarrolladores pueden crear aplicaciones móviles de alta calidad que funcionan en varias plataformas a partir de una sola base de código, lo que aumenta la eficiencia y reduce los costos de desarrollo (Ionic, 2023).

Entre las ventajas de Ionic sobre otras herramientas de desarrollo móvil híbrido, como Cordova, se encuentran las siguientes:

1. **Facilidad de uso:** Ionic es fácil de aprender y usar para desarrolladores con experiencia en tecnología web. Además, su enfoque en la creación de aplicaciones móviles híbridas utilizando tecnologías web facilita la integración con otros marcos y herramientas.
2. **Diseño atractivo:** Ionic provee a los desarrolladores un conjunto de elementos de diseño ya establecidos y adaptables que son útiles para crear aplicaciones móviles con buena presentación sin dejar de lado la calidad.
3. **Gran comunidad:** La plataforma Ionic cuenta con una extensa comunidad de programadores y usuarios que aportan continuamente nuevas funcionalidades, soluciones a problemas y mejoras. Asimismo, existe una amplia gama de recursos y documentación disponible en línea para su uso.
4. **Compatibilidad con Angular:** Es posible emplear la plataforma Ionic en conjunto con Angular con el fin de producir aplicaciones móviles híbridas de gran escalabilidad y eficiencia. Con ello, los programadores pueden reutilizar

tanto el código como los componentes de Angular para crear apps móviles híbridas de calidad superior.

Cuando se utiliza la combinación de Ionic y Angular, los programadores tienen la posibilidad de disfrutar de las ventajas que brindan ambos frameworks, lo que resulta en la creación de aplicaciones móviles híbridas altamente escalables, eficientes y atractivas. En particular, Ionic facilita la producción de este tipo de aplicaciones al proveer una capa de abstracción adicional sobre Cordova. Además, al trabajar con Angular, los desarrolladores pueden beneficiarse de la habilidad de Angular para producir aplicaciones web robustas capaces de soportar alta carga (Ionic, 2023).

**Node js:** es un entorno de ejecución de JavaScript que posibilita la creación de aplicaciones web de alto rendimiento y escalables del lado del servidor. En contraste a otros lenguajes de programación como C# o Java, Node.js se apoya en un modelo de entrada y salida no bloqueante y enfocado en eventos, lo que lo convierte en una opción adecuada para desarrollar microservicios.

Entre las ventajas de Node.js para la implementación de microservicios se encuentran las siguientes:

1. **Escalabilidad:** Node.js es altamente escalable y puede manejar grandes volúmenes de solicitudes simultáneas. Además, su modelo de entrada/salida no bloqueante permite un mejor uso de los recursos del sistema.

2. **Velocidad de desarrollo:** Node.js se destaca por su rapidez en el proceso de producción de aplicaciones web y microservicios. Gracias a su sintaxis sencilla y su habilidad para reutilizar código ya existente, es una opción muy recomendable para proyectos de desarrollo que busquen implementar metodologías ágiles.
3. **Facilidad de implementación:** Node.js es fácil de implementar y ejecutar en múltiples plataformas. Además, al ser capaz de trabajar tanto en el lado del cliente como del servidor, permite una mayor flexibilidad en la arquitectura de la aplicación.
4. **Comunidad activa:** Node.js cuenta con una comunidad activa de desarrolladores y usuarios que contribuyen con nuevas características y mejoras, lo que lo hace una tecnología en constante evolución.

En comparación con Java, Node.js ofrece una sintaxis más sencilla y una mayor eficiencia en el uso de los recursos del sistema. Java, por otro lado, puede ser más adecuado para aplicaciones empresariales complejas y de alto rendimiento que requieren características de seguridad avanzadas y un alto grado de confiabilidad (SHAH, 2018).

**MySQL:** es un software de código abierto que se encarga de administrar bases de datos relacionales, siendo muy popular en el mundo empresarial y en la web. Su origen se remonta a MySQL AB, aunque actualmente es propiedad de Oracle Corporation. Para realizar operaciones de consulta, actualización y administración de datos, MySQL hace uso del lenguaje SQL (Structured Query Language), el cual es ampliamente utilizado en el

mundo de las bases de datos. Este sistema de gestión de bases de datos se ha convertido en una herramienta esencial para la organización y acceso a grandes cantidades de información (Silvia Botros, 2021).

Entre las ventajas de usar MySQL como motor de base de datos se encuentran las siguientes:

1. **Rendimiento:** MySQL es conocido por su rápido rendimiento, lo que lo hace una buena opción para aplicaciones que necesitan procesar grandes cantidades de datos.
2. **Escalabilidad:** MySQL es altamente escalable y puede manejar grandes cantidades de datos y usuarios simultáneos.
3. **Fiabilidad:** Durante muchos años, MySQL ha sido un motor de base de datos de gran fiabilidad y resistencia, ampliamente utilizado en entornos de producción debido a su capacidad para mantener la integridad de los datos y su durabilidad a largo plazo.
4. **Comunidad activa:** Como un motor de base de datos de código abierto, MySQL cuenta con una comunidad activa de desarrolladores y usuarios que contribuyen con nuevas características y mejoras.
5. **Costo:** MySQL es de código abierto y gratuito para su uso, lo que lo hace una opción atractiva para empresas y organizaciones que buscan reducir costos. (Schwartz et al., 2012)

En cuanto a la definición arquitectónica, hay varias posibilidades que son populares y ampliamente utilizadas en la industria hoy en día, incluyendo:

Tabla 2 Diferentes tipos de arquitecturas de software

Arquitectura	Ventajas	Desventajas
<b>Microservicios</b>	Escalabilidad, resiliencia, flexibilidad y mantenibilidad, el desarrollo y despliegue de servicios individuales se puede hacer de forma independiente (James Lewis, 2014).	Mayor complejidad operativa, comunicación y coordinación entre servicios, mayor latencia (James Lewis, 2014).
<b>Sin servidor</b>	Menores costes operativos, escalabilidad, facilidad de desarrollo y despliegue, aprovisionamiento automático de recursos (Roberts, martinowler, 2018).	Control limitado de la infraestructura, tiempo de arranque en frío, dependencia del proveedor, posibilidad de que aumenten los problemas de latencia y rendimiento (Roberts, martinowler, 2018).
<b>Arquitectura basada en eventos</b>	Escalabilidad, procesamiento en tiempo real, acoplamiento flexible entre componentes, mayor capacidad de respuesta del sistema (Fowler, martinowler, 2017).	Mayor complejidad, la depuración puede resultar complicada, posibilidad de pérdida de eventos (Fowler, martinowler, 2017).
<b>Diseño basado en dominios</b>	Mejor alineación entre el desarrollo de software y los objetivos empresariales, modelado de	Mayor complejidad, el diseño puede estar demasiado centrado en los

	sistemas complejos, encapsulación de la lógica empresarial (Vernon, Domain-Driven Design Distilled, 2016).	dominios, la implementación puede ser difícil (Vernon, Domain-Driven Design Distilled, 2016)..
<b>Nativo de la nube</b>	Escalabilidad, resiliencia, mayor agilidad y tiempo de comercialización, mayor disponibilidad (Naresh Kumar Sehgal, 2019)	Es dependiente de la infraestructura en la nube, tiene problemas de seguridad y mayor complejidad operativa (Naresh Kumar Sehgal, 2019).

Nota: Elaboración propia (2023)

En el proyecto desarrollado, se utilizan microservicios y en el capítulo de proceso de software, se profundizará en la justificación de diseño que respalda esta arquitectura, demostrando cómo contribuye a la eficiencia, modularidad y mantenibilidad del proyecto.

**SonarCloud:** Con relación a la calidad del software, se busca hacer uso de SonarCloud, que es una plataforma en la nube que realiza análisis de la calidad del código con el objetivo de ayudar a los desarrolladores a mejorar la calidad del mismo. Esta herramienta se encarga de examinar el código fuente de los proyectos de software para detectar posibles problemas, como errores, vulnerabilidades de seguridad y violaciones de las normas de codificación, brindando información detallada para que los desarrolladores puedan corregir estos problemas de manera efectiva.

SonarCloud ofrece un completo conjunto de herramientas de análisis estático para varios lenguajes de programación, como Java, C#, JavaScript y Python. Adicionalmente, permite la integración con herramientas populares como GitHub, Azure DevOps y Bitbucket para proporcionar retroalimentación continua y sugerencias de mejora. Algunas de las características clave de SonarCloud incluyen:

1. **Análisis de Calidad de Código:** proporciona un informe detallado de la calidad del código, incluyendo métricas como la cobertura del código, la deuda técnica y la duplicación.
2. **Detección de vulnerabilidades de seguridad:** analiza el código fuente en busca de posibles vulnerabilidades de seguridad y ofrece recomendaciones para remediarlas.
3. **Aplicación de normas de codificación:** soporta estándares de codificación y mejores prácticas, y puede hacer cumplir estos estándares para asegurar que el código es mantenible y legible.
4. **Reglas personalizables:** proporciona un amplio conjunto de reglas incorporadas, y también permite a los usuarios crear y modificar reglas para satisfacer sus necesidades específicas.
5. **Revisión colaborativa del código:** proporciona una plataforma para la colaboración, lo que permite a los desarrolladores discutir los problemas de calidad del código y colaborar para solucionarlos (Krief, 2022).

En síntesis, se puede decir que SonarCloud es una herramienta muy útil para analizar la calidad del software, lo que puede ayudar a las empresas a mejorar la calidad de su código y reducir el riesgo de errores y vulnerabilidades de seguridad. Gracias a su capacidad para detectar problemas y ofrecer información detallada, los desarrolladores pueden trabajar de manera más eficiente y tomar medidas para mejorar la calidad del software en general.

**CI/CD:** es una práctica de desarrollo de software que se enfoca en automatizar y mejorar el proceso de construcción, prueba y entrega de software. Esta práctica implica la integración y prueba automática de los cambios en el código fuente en un entorno aislado, lo que permite detectar rápidamente cualquier problema y reducir el tiempo necesario para entregar el software. En resumen, CI/CD busca hacer más eficiente y efectivo el proceso de desarrollo de software a través de la automatización y la detección temprana de problemas.

La integración continua se refiere a la práctica de integrar el código de los desarrolladores en un repositorio central varias veces al día. Esto permite detectar errores de forma temprana y resolver problemas de integración de forma más rápida (Belmont J. M., 2018).

El despliegue continuo, por otro lado, se refiere a la práctica de entregar el software al entorno de producción de forma automática después de que se ha realizado una integración exitosa y se han superado todas las pruebas necesarias. Esto permite a los desarrolladores entregar nuevas características y correcciones de errores a los usuarios de manera más rápida y eficiente (Belmont J. M., 2018).

Entre las ventajas de la integración y despliegue continuo se encuentran:

1. **Reducción de errores y tiempo de corrección:** Al integrar y probar el código de forma continua, se pueden detectar errores de forma temprana y reducir el tiempo necesario para corregirlos.
2. **Mejora de la calidad del software:** Al automatizar las pruebas, se puede garantizar una mayor calidad del software.
3. **Mayor velocidad de entrega:** Al entregar el software de forma continua, se puede reducir el tiempo que lleva pasar del desarrollo a la entrega de software.
4. **Mayor eficiencia:** La automatización de las pruebas y la entrega de software permite a los equipos de desarrollo centrarse en tareas más importantes y estratégicas.
5. **Mayor colaboración y transparencia:** La integración continua permite a los desarrolladores trabajar en el mismo código base y resolver problemas de integración de forma más rápida.

En resumen, la integración y despliegue continuo es una práctica de desarrollo de software que permite a los equipos de desarrollo automatizar el proceso de construcción, prueba y entrega de software, lo que permite una mayor eficiencia y una mejor calidad del software (Belmont J. M., 2018).

**GitHub:** Actions es un servicio de integración y despliegue continuo ofrecido por GitHub. Con GitHub Actions, los desarrolladores pueden automatizar el proceso de integración, prueba y despliegue de su código directamente desde su repositorio en GitHub (Chaminda Chandrasekara, 2021). Entre las ventajas de usar GitHub Actions para la integración y despliegue continuo se encuentran:

1. **Integración nativa con GitHub:** GitHub Actions está integrado directamente en GitHub, lo que facilita la configuración y el uso para los desarrolladores que ya están familiarizados con la plataforma.
2. **Amplia variedad de acciones predefinidas:** GitHub Actions proporciona una amplia variedad de acciones predefinidas que pueden ser utilizadas para automatizar diferentes tareas de integración y despliegue.
3. **Personalización:** Los desarrolladores pueden crear sus propias acciones personalizadas y compartirlas con la comunidad.
4. **Gratuita para proyectos de código abierto:** GitHub Actions es gratuito para proyectos de código abierto, lo que lo hace una opción atractiva para proyectos sin fines de lucro.
5. **Integración con otros servicios:** GitHub Actions se integra con una amplia variedad de servicios, incluyendo AWS, Google Cloud Platform, y Microsoft Azure (Chaminda Chandrasekara, 2021).

En relación a la protección de datos exigidas por el gobierno nacional se identifica de obligatorio cumplimiento la **ley 1581 de 2012 “por la cual se dictan disposiciones**

**generales para la protección de datos”** la cual se fundamenta en el artículo 15 de la constitución política de Colombia que cita **“Todas las personas tienen derecho a su intimidad personal y familiar y a su buen nombre, y el Estado debe respetarlos y hacerlos respetar. De igual modo, tienen derecho a conocer, actualizar y rectificar las informaciones que se hayan recogido sobre ellas en bancos de datos y en archivos de entidades públicas y privadas. En la recolección, tratamiento y circulación de datos se respetarán la libertad y demás garantías consagradas en la Constitución. La correspondencia y demás formas de comunicación privada son inviolables. Sólo pueden ser interceptadas o registradas mediante orden judicial, en los casos y con las formalidades que establezca la ley (Ley 1581 De 2012 - Gestor Normativo, n.d.)**

## 4. METODOLOGÍA

Para el desarrollo de la solución expuesta en la formulación del problema, se requiere una metodología que, por su naturaleza, facilite los mecanismos necesarios para identificar y priorizar entregables, tomando como criterio el beneficio aportado. Adicionalmente, dicha metodología también debe permitir un aprovechamiento satisfactorio de los recursos humanos, técnicos y tecnológicos; esto, teniendo en cuenta las limitantes de tiempo con que se cuenta.

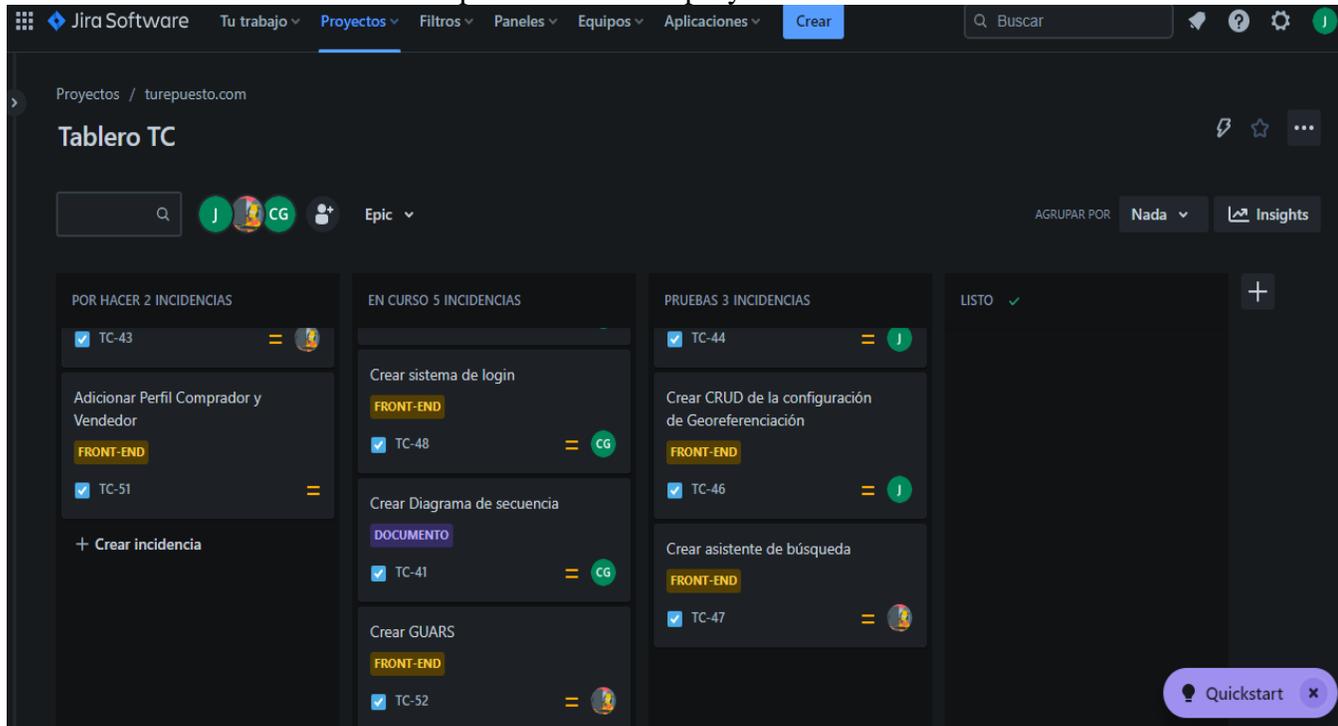
Por tanto, se elige la aplicación de la metodología Ágil de trabajo KANBAN para la ejecución del presente proyecto y se ha definido de la siguiente forma el equipo de trabajo:

- **Distribución del equipo:**
  - **Service Request Manager:** John Edisson Gamboa Guzman.
  - **Service Delivery Manager:** Andres Camilo Santacruz Borda.
  - **Equipo de desarrollo:** Carlos Andres Garzon, Andres Camilo Santacruz Borda y John Edisson Gamboa Guzman.
- Se ha definido una reunión diaria de 15 minutos para evaluar el estado de las actividades en ejecución y la definición de las siguientes actividades a ejecutar.
- Los requerimientos funcionales serán definidos mediante *HISTORIAS DE USUARIO*.
- Se implementará un tablero de tipo taskboard (Albert Nogués, 2017) en la herramienta JIRA para llevar el control de las actividades y los estados en los que se encuentran. Específicamente, se documentará el tablero tradicional característico de

la metodología KANBAN el cual propone mostrar gráficamente el flujo de trabajo por columnas, manifestando:

1. TAREAS POR HACER
2. TAREAS EN CURSO
3. TAREAS EN PRUEBA
4. TAREAS LISTAS.

Ilustración 2 Tablero KANBAN implementado en el proyecto



Nota: Elaboración propia (2023)

Relacionado con este tablero de registro de Responsabilidades y Responsables, se usará la herramienta Confluence, a fin de persistir de manera colaborativa el conocimiento desarrollado en el transcurso del proyecto. Específicamente, en Confluence se encontrarán documentos del tipo:

- 1. Diagramas:** Referentes a orientaciones visuales de la ejecución de una actividad o los exigidos por el presente documento.
- 2. Spikes:** investigaciones referentes al levantamiento de información para la toma de decisiones en la etapa de diseño.
- 3. Estándares de trabajo:** Acuerdos de equipo construidos para estandarizar la forma de trabajo, entre otros.

## Ilustración 3 Repositorio Confluence usado para persistir el conocimiento relacionado al proyecto

The screenshot shows a Confluence page in a workspace named "los repuesteros". The page title is "TC-8 (Spike) Capítulo 10.3 - Diseño y arquitectura". The page was created by CARLOS GARZON and last updated by Andres Santacruz on December 9, 2022. The page content includes a metadata table, a "Resumen" section, and a "Documentos analizados" section. A table at the bottom lists repository items with columns for ID, Url, Nombre, and Pagina Modulo.

**los repuesteros** CAMBIOS NO PUBLICADOS Compartir

### TC-8 (Spike) Capítulo 10.3 - Diseño y arquitectura

CC Creado por CARLOS GARZON, con una plantilla  
Última actualización: dic 09, 2022 por Andres Santacruz

<b>Fecha</b>	8 dic 2022
<b>Responsable</b>	@CARLOS GARZON
<b>En esta página</b>	<ul style="list-style-type: none"> <li>Resumen</li> <li>Documentos analizados</li> <li>Notas</li> <li>Contenido del modulo</li> </ul>

#### Resumen

El siguiente documento se realiza con el fin de dar a conocer al equipo (TuRepuesto.com) la forma de implementar el modulo 10.3 del proyecto de grado para optar al titulo de la especialización en ingeniera de software.

#### Documentos analizados

En la siguiente investigación se analizaron diferentes proyectos alojados en el repositorio de la universidad Antonio Nariño [Universidad Antonio Nariño: Página de inicio](#) donde se evidencia el modulo de **Diseño y arquitectura**.

ID Repositorio UAN	Url	Nombre	Pagina Modulo
5754	<a href="http://repositorio.uan.edu.co/bits/stream/123456789/5754/1/2021">http://repositorio.uan.edu.co/bits/stream/123456789/5754/1/2021</a>	Aplicación web que permite la búsqueda	Inicio rápido

Nota: Elaboración propia (2023)

## 5. PROCESO DE SOFTWARE

En el presente capítulo, se encontrará la explicación del desarrollo junto con descripciones de las tareas necesarias para crear el software en cuestión.

Específicamente, tendrá en cuenta las siguientes pautas:

1. **Análisis:** Levantamiento de requerimientos convertidos en historias de usuario según las metodologías ágiles indican.
2. **Diseño:** Opciones de implementación para la construcción, definición general de la estructura del software y su operatividad.
3. **Implementación:** Elección de herramientas, entorno, lenguajes de programación, entre otros.
4. **Pruebas:** Validación de funcionalidades y de calidad.
5. **Instalación o despliegue:** Poner software en funcionamiento.
6. **Uso y mantenimiento:** Eliminar defectos, adaptar a nuevas necesidades, evolución.

### 5.1. REQUERIMIENTOS FUNCIONALES

Los requerimientos funcionales (RF) tipificados como *HISTORIAS DE USUARIO (HU)* según las metodologías ágiles, se entienden como actividades o movimientos que son requeridas para dar identidad a este desarrollo de software, es decir, por ejemplo: “Quiero que la herramienta me permita cambiar el tamaño de la letra”, esto da a entender que si la herramienta desarrollada no permite esta funcionalidad, no está cumpliendo los requisitos previamente establecidos.

A continuación se cita previamente los historias de usuario a manera introductoria, entendiéndose que el desarrollo de software expuesto en este documento debe cumplirlas en su totalidad y posteriormente se encuentra el detalle técnico de las mismas.

- a. Crear home
- b. Asistente de búsqueda de accesorios
- c. Asistente de búsqueda de repuesto
- d. Revisar compra
- e. Publicar compra
- f. Recibir publicaciones de compras
- g. Publicar oferta
- h. Recibir notificación de ofertas
- i. Organizar ofertas
- j. Ver oferta
- k. Aceptar oferta
- l. Mostrar datos de contacto
- m. Configurar radio de búsqueda de vendedores
- n. Sistema de calificaciones
- o. Login y registro

Tabla 3 Historia de usuario TR-00

<b>Tipo</b>	Historia de usuario	<b>ID</b>	TR-001	<b>Solicitado por</b>	Vendedor y comprador
<b>Nombre</b>	Crear home				
<b>Prioridad (1 alta – 5 baja)</b>	1	<b>Estimación (s-m-l-xl-xxl)</b>	L	<b>Iteración #</b>	
<b>DESCRIPCIÓN</b>					
<b>Yo como:</b>	Comprador y vendedor				
<b>Necesito:</b>	ver la pantalla principal del aplicativo				
<b>Para:</b>	que posteriormente se puedan construir sobre ellas las demás funcionalidades				
<b>Entradas (Insumos / Archivos)</b>				<b>Resultado esperado</b>	
<ul style="list-style-type: none"> <li>N/A</li> </ul>				Ventana que se ajusta a los criterios de aceptación	
<b>Atributos de calidad</b>			Usabilidad		
<b>CRITERIOS DE ACEPTACIÓN</b>					
<b>#</b>	<b>Escenario</b>	<b>Contexto</b>	<b>Evento</b>	<b>Reglas</b>	
1		El vendedor o comprador ingresa a el app	Cuando el vendedor o comprador ingrese al sistema, entonces podrá visualizar una pantalla principal acorde a los diseños UX.		
2.		El vendedor o comprador hace click en el menú	Cuando el vendedor o comprador, haga click en el botón de ver menú, se deberá desplegar una barra lateral con las siguientes opciones: <ul style="list-style-type: none"> <li>Configuraciones</li> </ul>		

Nota: Elaboración propia (2023)

Tabla 4 Historia de usuario TR-002

<b>Tipo</b>	Historia de usuario	<b>ID</b>	TR-002	<b>Solicitado por</b>	Comprador
<b>Nombre</b>	Asistente de búsqueda de repuesto				
<b>Prioridad (1 alta – 5 baja)</b>	4	<b>Estimación (s-m-l-xl-xxl)</b>	XXL	<b>Iteración #</b>	
<b>DESCRIPCIÓN</b>					
<b>Yo como:</b>	Comprador				
<b>Necesito:</b>	seleccionar la opción buscar repuesto en el menú				
<b>Para:</b>	poder utilizar el asistente de búsqueda de repuesto				
<b>Entradas (Insumos / Archivos)</b>			<b>Resultado esperado</b>		
<ul style="list-style-type: none"> <li>N/A</li> </ul>			El comprador podrá seleccionar la opción buscar accesorios y hacer uso del asistente para seleccionar el accesorio que está buscando.		
<b>Atributos de calidad</b>		Usabilidad			
<b>CRITERIOS DE ACEPTACIÓN</b>					
<b>#</b>	<b>Escenario</b>	<b>Contexto</b>	<b>Evento</b>	<b>Reglas</b>	
1		El comprador selecciona la opción buscar accesorio	Cuando el comprador seleccione la opción buscar accesorios entonces verá una pantalla con dos botones: Repuestos y accesorios.		
2.		El comprador selecciona la opción Repuesto	Cuando el comprador seleccione el botón Repuesto, entonces verá el asistente para seleccionar repuestos		
3.		El comprador visualiza el asistente de	Cuando el comprador visualiza el asistente de búsqueda de repuestos, entonces podrá seguir		

		busqueda de repuestos	los pasos indicados para seleccionar el repuesto de su interés	
4.		El comprador identifica el repuesto de su interés	Cuando el comprador selecciona el repuesto de su interés, entonces ese repuesto deberá ser guardado en una lista de repuestos seleccionados y aparecerán los botones de buscar otro repuesto o terminar	
5.		El comprador selecciona la opción terminar	Cuando el comprador seleccione el botón terminar, entonces será redirigido a la pantalla principal.	
6.		El comprador selecciona la opción buscar otro repuesto	Cuando el comprador seleccione la opción buscar otro repuesto, entonces será redirigido de nuevo al primer paso del asistente y podrá continuar el flujo	
7.		El asistente de búsqueda debe ser parametrizable por base de datos.	Cada vez que se requiera agregar, quitar o modificar un producto nuevo, turepuesto.com lo hará por base de datos.	

Nota: Elaboración propia (2023)

Tabla 5 Historia de usuario TR-003

<b>Tipo</b>	Historia de usuario	<b>ID</b>	TR-003	<b>Solicitado por</b>	Comprador
<b>Nombre</b>	Asistente de búsqueda de accesorio				
<b>Prioridad (1 alta – 5 baja)</b>	1	<b>Estimación (s-m-l-xl-xxl)</b>	XL	<b>Iteración #</b>	
<b>DESCRIPCIÓN</b>					
<b>Yo como:</b>	Comprador				
<b>Necesito:</b>	seleccionar la opción buscar accesorio en el menú				
<b>Para:</b>	poder utilizar el asistente de búsqueda de accesorio				
<b>Entradas (Insumos / Archivos)</b>				<b>Resultado esperado</b>	
<ul style="list-style-type: none"> <li>N/A</li> </ul>				El comprador podrá seleccionar la opción buscar accesorios y hacer uso del asistente para seleccionar el accesorio que está buscando.	
<b>Atributos de calidad</b>			Usabilidad		
<b>CRITERIOS DE ACEPTACIÓN</b>					
<b>#</b>	<b>Escenario</b>	<b>Contexto</b>	<b>Evento</b>	<b>Reglas</b>	
1		El comprador selecciona la opción buscar accesorio	Cuando el comprador seleccione la opción buscar accesorios entonces verá una pantalla con dos botones: Repuestos y accesorios.		
2.		El comprador selecciona la opción accesorio	Cuando el comprador seleccione el botón accesorios, entonces verá el asistente para seleccionar accesorio		
3.		El comprador visualiza el asistente de búsqueda de	Cuando el comprador visualiza el asistente de búsqueda de repuestos, entonces podrá seguir los pasos indicados para		

		accesorios	seleccionar el accesorio de su interés	
4.		El comprador identifica el accesorio de su interés	Cuando el comprador selecciona el accesorio de su interés, entonces ese repuesto deberá ser guardado en una lista de accesorio y aparecerán los botones de buscar otro accesorio o terminar	
5.		El comprador selecciona la opción terminar	Cuando el comprador seleccione el botón terminar, entonces será redirigido a la pantalla principal.	
6.		El comprador selecciona la opción buscar otro accesorio	Cuando el comprador seleccione la opción buscar otro accesorio, entonces será redirigido de nuevo al primer paso del asistente y podrá continuar el flujo	
7.		El asistente de búsqueda debe ser parametrizable por base de datos.	Cada vez que se requiera agregar, quitar o modificar un producto nuevo, turepuesto.com lo hará por base de datos.	

Nota. Elaboración propia (2023)

Tabla 6 Historia de usuario TR-004

<b>Tipo</b>	Historia de usuario	<b>ID</b>	TR-004	<b>Solicitado por</b>	Comprador
<b>Nombre</b>	Revisar compra				
<b>Prioridad (1 alta – 5 baja)</b>	1	<b>Estimación (s-m-l-xl-xxl)</b>	XL	<b>Iteración #</b>	
<b>DESCRIPCIÓN</b>					
<b>Yo como:</b>	Comprador				
<b>Necesito:</b>	revisar los repuestos y accesorios que seleccione en mi compra				
<b>Para:</b>	para eliminar o agregar más productos si lo necesito y posteriormente publicar				
<b>Entradas (Insumos / Archivos)</b>			<b>Resultado esperado</b>		
<ul style="list-style-type: none"> <li>N/A</li> </ul>			El comprador podrá revisar los repuestos y accesorios que seleccionó para comprar y eliminar de la lista antes de publicar.		
<b>Atributos de calidad</b>		Usabilidad			
<b>CRITERIOS DE ACEPTACIÓN</b>					
<b>#</b>	<b>Escenario</b>	<b>Contexto</b>	<b>Evento</b>	<b>Reglas</b>	
1		El comprador selecciona la opción revisar y publicar compra	Cuando el comprador entre a la sección de revisar su compra, entonces verá un listado de los repuestos y accesorios que seleccionó		
2.		El comprador elimina un repuesto y/o accesorio de la lista	Cuando el comprador elimina un repuesto o accesorio de la lista, entonces dicho artículo se eliminará de la lista que será		

			publicada y desaparecerá de la pantalla.	
3.		El comprador podrá ver el botón publicar compra	Cuando el comprador esté en la opción revisar compra entonces visualiza el botón publicar compra pero este aún no tendrá funcionalidad.	

Nota. Elaboración propia (2023)

Tabla 7 Historia de usuario TR-005

<b>Tipo</b>	Historia de usuario	<b>ID</b>	TR-005	<b>Solicitado por</b>	Comprador
<b>Nombre</b>	Publicar compra				
<b>Prioridad (1 alta – 5 baja)</b>	1	<b>Estimación (s-m-l-xl-xxl)</b>	XL	<b>Iteración #</b>	
<b>DESCRIPCIÓN</b>					
<b>Yo como:</b>	Comprador				
<b>Necesito:</b>	publicar los repuestos y accesorios que seleccione en mi compra				
<b>Para:</b>	notificar a los vendedores que se encuentren en los criterios de búsqueda				
<b>Entradas (Insumos / Archivos)</b>			<b>Resultado esperado</b>		
<ul style="list-style-type: none"> <li>N/A</li> </ul>			El comprador podrá publicar los repuestos y accesorios que seleccionó para comprar y eliminar de la lista antes de publicar.		
<b>Atributos de calidad</b>		Usabilidad			
<b>CRITERIOS DE ACEPTACIÓN</b>					
<b>#</b>	<b>Escenario</b>	<b>Contexto</b>	<b>Evento</b>	<b>Reglas</b>	
1		El comprador hace click sobre el botón publicar compra	Cuando el comprador hace click sobre el botón publicar compra, entonces la compra será notificada a todos los vendedores que se encuentran en su radio de búsqueda.		
2.		La compra no pudo ser publicada	Cuando el comprador publica su compra y hay error, entonces se desplegará un tooltip como se ve en los diseños indicando el error		
3.		Se debe usar la formula haversine para	Para notificar a vendedores se debe usar la formula mencionada ue solicita como	La latitud y la longitud del vendedor estarán en	

		determinar a qué vendedores notificar	parámetros latitud, y longitud.	base de datos y se habrán guardado en su proceso de registro en el aplicativo.
--	--	---------------------------------------	---------------------------------	--------------------------------------------------------------------------------

Nota. Elaboración propia (2023)

Tabla 8 Historia de usuario TR-006

<b>Tipo</b>	Historia de usuario	<b>ID</b>	TR-006	<b>Solicitado por</b>	Vendedor
<b>Nombre</b>	Recibir publicaciones de compras				
<b>Prioridad (1 alta – 5 baja)</b>	1	<b>Estimación (s-m-l-xl-xxl)</b>	XL	<b>Iteración #</b>	
<b>DESCRIPCIÓN</b>					
<b>Yo como:</b>	Vendedor				
<b>Necesito:</b>	recibir la notificación de las solicitudes de productos que realizan posibles compradores				
<b>Para:</b>	posteriormente poder evaluar y hacer una oferta				
<b>Entradas (Insumos / Archivos)</b>				<b>Resultado esperado</b>	
<ul style="list-style-type: none"> <li>N/A</li> </ul>				El vendedor recibirá una notificación de las solicitudes de artículos realizadas por compradores	
<b>Atributos de calidad</b>			Usabilidad		
<b>CRITERIOS DE ACEPTACIÓN</b>					
<b>#</b>	<b>Escenario</b>	<b>Contexto</b>	<b>Evento</b>	<b>Reglas</b>	
1		El vendedor recibirá una notificación cuando aplique para poder responder a una solicitud	Cuando un comprador realice la publicación de una compra, entonces los vendedores que se encuentran en su radio de búsqueda recibirán una notificación que indique que puede realizar una oferta		
2		El vendedor podrá ver en su pantalla principal la nueva solicitud de compra para poder responder a ella	Cuando ha llegado la notificación de una solicitud de compra, el vendedor podrá verla en el listado de su pantalla principal		

3.		Las solicitudes de compras que llegan al vendedor deben visualizarse en el home y estar ordenadas bajo el criterio LIFO (last in first out)	Cuando llega una nueva notificación de solicitud de compra a un vendedor, entonces esta debe listarse en el home de primeras bajo el criterio de cola (LIFO)	
----	--	---------------------------------------------------------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------	--

Nota. Elaboración propia (2023)

Tabla 9 Historia de usuario TR-007

<b>Tipo</b>	Historia de usuario	<b>ID</b>	TR-007	<b>Solicitado por</b>	Vendedor
<b>Nombre</b>	Publicar oferta				
<b>Prioridad (1 alta – 5 baja)</b>	1	<b>Estimación (s-m-l-xl-xxl)</b>	XL	<b>Iteración #</b>	
<b>DESCRIPCIÓN</b>					
<b>Yo como:</b>	Vendedor				
<b>Necesito:</b>	poder abrir las solicitudes de compras listadas en mi página principal				
<b>Para:</b>	poder realizar una oferta				
<b>Entradas (Insumos / Archivos)</b>				<b>Resultado esperado</b>	
<ul style="list-style-type: none"> <li>N/A</li> </ul>				El vendedor podrá abrir una solicitud de compra listada en su página principal y realizar una oferta para cada uno de los productos allí listados	
<b>Atributos de calidad</b>			Usabilidad		
<b>CRITERIOS DE ACEPTACIÓN</b>					
<b>#</b>	<b>Escenario</b>	<b>Contexto</b>	<b>Evento</b>	<b>Reglas</b>	
1		El vendedor podrá hacer clic sobre cualquiera de las solicitudes listadas en su home	Cuando el vendedor hace click sobre alguna de las solicitudes listadas en su página principal, entonces debe poder ver el listado de los artículos solicitados por el posible comprador.		
2.		El vendedor podrá seleccionar los artículos que tenga en stock y hacer una oferta en valor a cada uno de ellos y además agregar una breve descripción	Cuando el vendedor seleccione alguno de los artículos que está en el listado de la solicitud de compra, entonces podrá hacer una oferta en valor y además agregar una breve descripción de 100 caracteres.		

		de 100 caracteres.		
3.		El vendedor podrá responder a la compra con los artículos que posee, los valores y la descripción	Cuando el vendedor ha seleccionado los artículos que posee, ha diligenciado el precio de venta y las descripciones, entonces podrá hacer click en el botón publicar oferta.	

Nota. Elaboración propia (2023)

Tabla 10 Historia de usuario TR-008

<b>Tipo</b>	Historia de usuario	<b>ID</b>	TR-008	<b>Solicitado por</b>	Comprador
<b>Nombre</b>	Recibir notificación de ofertas				
<b>Prioridad (1 alta – 5 baja)</b>	1	<b>Estimación (s-m-l-xl-xxl)</b>	XL	<b>Iteración #</b>	
<b>DESCRIPCIÓN</b>					
<b>Yo como:</b>	Comprador				
<b>Necesito:</b>	poder recibir notificaciones de las ofertas hechas por los vendedores				
<b>Para:</b>	poder verlas en la pantalla principal.				
<b>Entradas (Insumos / Archivos)</b>			<b>Resultado esperado</b>		
<ul style="list-style-type: none"> <li>N/A</li> </ul>			El comprador recibirá notificaciones de la ofertas realizadas por vendedores y las verá listadas en su pantalla principal		
<b>Atributos de calidad</b>		Usabilidad			
<b>CRITERIOS DE ACEPTACIÓN</b>					
	<b>Escenario</b>	<b>Contexto</b>	<b>Evento</b>	<b>Reglas</b>	
1.		El comprador recibirá notificaciones	Cuando un vendedor realiza una oferta, entonces el comprador debe recibir una notificación con el mensaje descrito en los diseños UX		
2.		El comprador podrá ver el listado de ofertas que han hecho los diferentes vendedores.	Cuando un vendedor realiza una oferta y el comprador ha recibido la notificación, entonces debe visualizar la nueva oferta junto con las otras en un listado en su pantalla principal		

Nota. Elaboración propia (2023)

Tabla 11 Historia de usuario TR-009

<b>Tipo</b>	Historia de usuario	<b>ID</b>	TR-009	<b>Solicitado por</b>	Comprador
<b>Nombre</b>	Organizar ofertas				
<b>Prioridad (1 alta – 5 baja)</b>	1	<b>Estimación (s-m-l-xl-xxl)</b>	XL	<b>Iteración #</b>	
<b>DESCRIPCIÓN</b>					
<b>Yo como:</b>	Comprador				
<b>Necesito:</b>	poder organizar las diferentes ofertas que me hacen vendedores				
<b>Para:</b>	seleccionar la que más me convenga según criterios de distancia o precio				
<b>Entradas (Insumos / Archivos)</b>			<b>Resultado esperado</b>		
<ul style="list-style-type: none"> <li>N/A</li> </ul>			<p>El comprador podrá organizar las ofertas que hacen los diferentes vendedores en criterio de precio o distancia y de manera ascendente o descendente.</p>		
<b>Atributos de calidad</b>		Usabilidad			
<b>CRITERIOS DE ACEPTACIÓN</b>					
	<b>Escenario</b>	<b>Contexto</b>	<b>Evento</b>	<b>Reglas</b>	
1.		El comprador podrá organizar la lista de ofertas de vendedores por precio	Cuando el comprador haga click sobre el botón ordenar por precio, entonces podrá ordenar las ofertas de los vendedores por precio de manera ascendente y descendente tal como muestra el diseño UX		
2.		El comprador podrá organizar la lista de ofertas de vendedores por distancia respecto al vendedor	Cuando el comprador haga click sobre el botón ordenar por distancia, entonces podrá ordenar las ofertas por distancia de manera ascendente y descendente tal como muestra el diseño UX		

Nota. Elaboración propia (2023)

Tabla 12 Historia de usuario TR-010

<b>Tipo</b>	Historia de usuario	<b>ID</b>	TR-010	<b>Solicitado por</b>	Comprador
<b>Nombre</b>	Ver oferta				
<b>Prioridad (1 alta – 5 baja)</b>	1	<b>Estimación (s-m-l-xl-xxl)</b>	XL	<b>Iteración #</b>	
<b>DESCRIPCIÓN</b>					
<b>Yo como:</b>	Comprador				
<b>Necesito:</b>	ver el detalle de una oferta				
<b>Para:</b>	evaluarla y decidir si quiero comprarla				
<b>Entradas (Insumos / Archivos)</b>			<b>Resultado esperado</b>		
<ul style="list-style-type: none"> <li>N/A</li> </ul>			El comprador podrá ingresar a cualquiera de las ofertas que están listadas en su pantalla principal y ver los productos que le oferta el vendedor		
<b>Atributos de calidad</b>		Usabilidad			
<b>CRITERIOS DE ACEPTACIÓN</b>					
	<b>Escenario</b>	<b>Contexto</b>	<b>Evento</b>	<b>Reglas</b>	
1.		El comprador ingresa a una oferta de las publicadas por los vendedores	Cuando el comprador ingresa al detalle de una oferta, entonces verá un listado con todos los artículos que él pidió en la compra		
2.		El comprador ve los productos que oferto el vendedor	Cuando el comprador está viendo el listado de productos dentro de la oferta, vera en gris los que no oferto el vendedor (respecto a los que él pidió) y los que si oferto el vendedor los verá en color y adicionalmente visualizará el precio ofertado y la descripción		

Nota. Elaboración propia (2023)

Tabla 13 Historia de usuario TR-011

<b>Tipo</b>	Historia de usuario	<b>ID</b>	TR-011	<b>Solicitado por</b>	Comprador
<b>Nombre</b>	Aceptar oferta				
<b>Prioridad (1 alta – 5 baja)</b>	1	<b>Estimación (s-m-l-xl-xxl)</b>	XL	<b>Iteración #</b>	
<b>DESCRIPCIÓN</b>					
<b>Yo como:</b>	Comprador				
<b>Necesito:</b>	aceptar una oferta de las realizadas por los vendedores				
<b>Para:</b>	notificar al vendedor que me interesa comprar sus productos				
<b>Entradas (Insumos / Archivos)</b>				<b>Resultado esperado</b>	
<ul style="list-style-type: none"> <li>N/A</li> </ul>				El comprador podrá indicar que le interesó una oferta y notificar al vendedor que desea realizar la compra	
<b>Atributos de calidad</b>			Usabilidad		
<b>CRITERIOS DE ACEPTACIÓN</b>					
	<b>Escenario</b>	<b>Contexto</b>	<b>Evento</b>	<b>Reglas</b>	
1.		El comprador haga click sobre una el botón comprar	Cuando el comprador haga click sobre el botón comprar, entonces se notificará al vendedor correspondiente que el comprador desea adquirir sus productos.		

Nota. Elaboración propia (2023)

Tabla 14 Historia de usuario TR-012

<b>Tipo</b>	Historia de usuario	<b>ID</b>	TR-012	<b>Solicitado por</b>	Comprador
<b>Nombre</b>	Mostrar datos de contacto				
<b>Prioridad (1 alta – 5 baja)</b>	1	<b>Estimación (s-m-l-xl-xxl)</b>	XL	<b>Iteración #</b>	
<b>DESCRIPCIÓN</b>					
<b>Yo como:</b>	Comprador y vendedor				
<b>Necesito:</b>	que se muestran nuestros datos de contacto				
<b>Para:</b>	poder formalizar la venta				
<b>Entradas (Insumos / Archivos)</b>			<b>Resultado esperado</b>		
<ul style="list-style-type: none"> <li>N/A</li> </ul>			El comprador podrá realizar la comunicación vía whatsapp a fin de que puedan establecer comunicación y formalizar la compra.		
<b>Atributos de calidad</b>		Usabilidad			
<b>CRITERIOS DE ACEPTACIÓN</b>					
	<b>Escenario</b>	<b>Contexto</b>	<b>Evento</b>	<b>Reglas</b>	
1.		Cuando se ha notificado al vendedor que el comprador está interesado, se enviará un mensaje vía whatsapp con la lista de los productos de interés.	Cuando el vendedor ha recibido la notificación de que el comprador está interesado en su oferta, se realizará la comunicación a través del whatsapp del comprador para que formalicen la venta.		

Nota. Elaboración propia (2023)

Tabla 15 Historia de usuario TR-013

<b>Tipo</b>	Historia de usuario	<b>ID</b>	TR-013	<b>Solicitado por</b>	Comprador
<b>Nombre</b>	Configurar radio de búsqueda de vendedores				
<b>Prioridad (1 alta – 5 baja)</b>	1	<b>Estimación (s-m-l-xl-xxl)</b>	XL	<b>Iteración #</b>	
<b>DESCRIPCIÓN</b>					
<b>Yo como:</b>	Comprador				
<b>Necesito:</b>	poder configurar el radio de búsqueda				
<b>Para:</b>	que las notificaciones de compra que haga, lleguen solo a los vendedores en ese radio				
<b>Entradas (Insumos / Archivos)</b>			<b>Resultado esperado</b>		
<ul style="list-style-type: none"> <li>N/A</li> </ul>			El comprador podrá configurar cada vez que quiera el radio de búsqueda en que quiere que se busque a vendedores		
<b>Atributos de calidad</b>		Usabilidad			
<b>CRITERIOS DE ACEPTACIÓN</b>					
	<b>Escenario</b>	<b>Contexto</b>	<b>Evento</b>	<b>Reglas</b>	
1.		El comprador hace click en el botón configurar radio de búsqueda	Cuando el comprador hace click sobre el botón configurar radio de búsqueda, entonces verá un modal con un campo numérico que le permitirá ingresar los kilómetros de manera numérica		
2.		El comprador hace clic en aceptar.	Cuando el comprador hace clic en aceptar, entonces la información será almacenada en base de datos y verá un mensaje de éxito		

3.		El comprador hace clic en cancelar.	Cuando el comprador hace clic en cancelar, entonces se cerrará el modal.	
4.		Hay un error al guardar la información	Cuando se presenta un error al intentar guardar la configuración en bd, entonces deberá aparecer un mensaje como el estipulado en los diseños UX.	

Nota. Elaboración propia (2023)

Tabla 16 Historia de usuario TR-014

<b>Tipo</b>	Historia de usuario	<b>ID</b>	TR-014	<b>Solicitado por</b>	Comprador y vendedor
<b>Nombre</b>	Sistema de calificaciones				
<b>Prioridad (1 alta – 5 baja)</b>	1	<b>Estimación (s-m-l-xl-xxl)</b>	XL	<b>Iteración #</b>	
<b>DESCRIPCIÓN</b>					
<b>Yo como:</b>	Comprador o vendedor				
<b>Necesito:</b>	poder calificar vendedores o compradores según sea el caso				
<b>Para:</b>	brindar información que contribuya a la seguridad				
<b>Entradas (Insumos / Archivos)</b>			<b>Resultado esperado</b>		
<ul style="list-style-type: none"> <li>N/A</li> </ul>			Los vendedores y compradores podrán calificar según sea el caso a las personas o compañías con los que hayan concretado compras.		
<b>Atributos de calidad</b>		Usabilidad			
<b>CRITERIOS DE ACEPTACIÓN</b>					
	<b>Escenario</b>	<b>Contexto</b>	<b>Evento</b>	<b>Reglas</b>	
1.		El comprador puede calificar de 1 a 5 a un vendedor con el que haya concretado una compra	Cuando el comprador da clic en el botón calificar al vendedor, entonces verá una pantalla en la que podrá realizar la calificación y agregar de manera opcional una descripción		
2.		El vendedor puede calificar de 1 a 5 a un comprador con el que haya concretado una compra	Cuando el vendedor da clic en el botón calificar al comprador, entonces verá una pantalla en la que podrá realizar la calificación y agregar de manera opcional una descripción		

3.		El usuario podrá calificar cuando regresa al app después de haber enviado o recibido la compra por Whatsapp	Después de que se ha enviado el mensaje de compra, el usuario en el aplicativo podrá hacer uso de un pop-up para calificar.	
----	--	-------------------------------------------------------------------------------------------------------------	-----------------------------------------------------------------------------------------------------------------------------	--

Nota. Elaboración propia (2023)

Tabla 17 Historia de usuario TR-015

<b>Tipo</b>	Historia de usuario	<b>ID</b>	TR-015	<b>Solicitado por</b>	Comprador y vendedor
<b>Nombre</b>	Login y registro				
<b>Prioridad (1 alta – 5 baja)</b>	1	<b>Estimación (s-m-l-xl-xxl)</b>	XL	<b>Iteración #</b>	
<b>DESCRIPCIÓN</b>					
<b>Yo como:</b>	Comprador o vendedor				
<b>Necesito:</b>	poderme registrar y logear en el aplicativo				
<b>Para:</b>	poder hacer uso de las funcionalidades				
<b>Entradas (Insumos / Archivos)</b>				<b>Resultado esperado</b>	
<ul style="list-style-type: none"> <li>N/A</li> </ul>				Los vendedores y compradores podrán registrarse y loguearse a través de correo con contraseña o google	
<b>Atributos de calidad</b>			Usabilidad		
<b>CRITERIOS DE ACEPTACIÓN</b>					
	<b>Escenario</b>	<b>Contexto</b>	<b>Evento</b>	<b>Reglas</b>	
1.		El usuario podrá loguearse haciendo uso de google	Cuando el usuario en la pantalla de login del sistema, indique que quiere iniciar por google, se iniciara la pantalla de inicio de sesión de google		
2.		El usuario podrá loguearse haciendo uso de correo y contraseña	Cuando el usuario en la pantalla login del sistema, Digite correo y contraseña válidos y presione el botón iniciar sesión.		
3.		El usuario podrá registrarse haciendo uso de correo y contraseña	Cuando el usuario haga clic en el botón crea una cuenta, entonces podrá digitar correo y contraseña válidos y presionar el botón regístrate ahora.		

4.		El usuario podrá loguearse haciendo uso de google	Cuando el usuario en la pantalla de login del sistema, indique que quiere iniciar por google, se iniciara la pantalla de inicio de sesión de google	
----	--	---------------------------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------	--

Nota. Elaboración propia (2023)

## 5.2. HISTORIAS DE USUARIO (REQUERIMIENTOS NO FUNCIONALES)

Tabla 18 Requerimientos no funcionales

Clasificación	Atributo de Calidad	Descripción
Producto	Usabilidad	El sistema debe correr en celulares móviles iOS y Android.
		El sistema debe contar con mensajes informativos y de error que orienten al usuario final.
	Ayudas para el aprendizaje	El sistema debe contar con un tutorial introductorio al funcionamiento de la app.
	Eficiencia	El sistema debe ser capaz de procesar más de 10 peticiones simultáneas en línea.
		El sistema debe ser capaz de operar adecuadamente con sesiones concurrentes.
	Seguridad	Compradores podrán ver información básica (Nombre completo, promedio de calificación) de los vendedores que le hagan ofertas a sus compras.
		Vendedores podrán ver información básica (Nombre completo, promedio de calificación) de los compradores de los cuales han recibido solicitudes de compra
		El sistema debe delegar la seguridad a google cuando se intenta logear por medio de esta.
	Organizacionales	Entorno
El sistema correrá al menos en las versiones 13.0 de iOS y 8.0 (Oreo) de Android.		
Los compilados del sistema no deben ser mayores a 180 MB por restricción de tiendas.		
Desarrollo	El sistema se desarrollará en frameworks para el desarrollo móvil híbrido.	
Externos	Legislativos	El sistema debe solicitar al usuario final la autorización para el tratamiento de sus datos personales de acuerdo a la Ley 2157 de 2021.
		Las obligaciones civiles y/o comerciales derivadas de las relaciones que se entablen al interior del sistema serán responsabilidad únicamente de los contratantes.

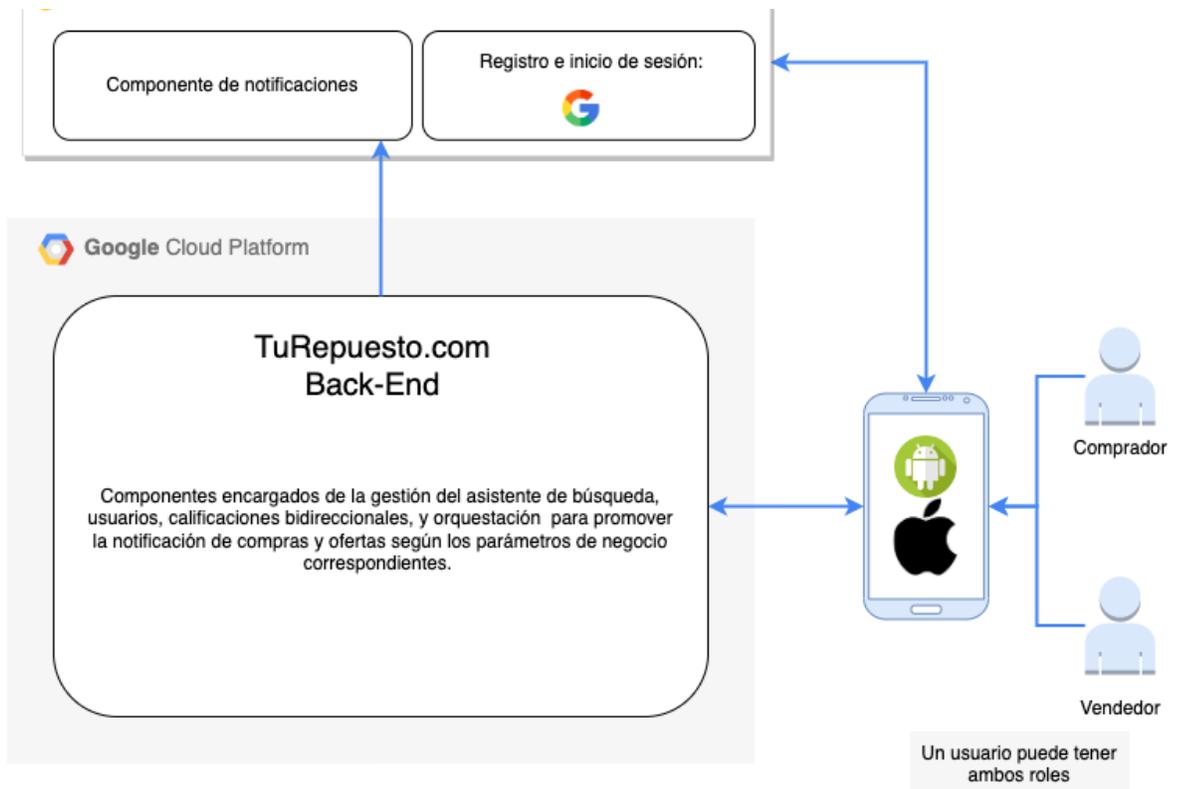
Nota. Elaboración propia (2023)

### 5.3. DISEÑO Y ARQUITECTURA

En esta sección, se profundiza en las definiciones arquitectónicas y de diseño construidas para la implementación del software propuesto, y para ello, se hará uso de los tres primeros niveles del modelo C4:

- **Diagrama de contexto:**

Ilustración 4 Diagrama de contexto



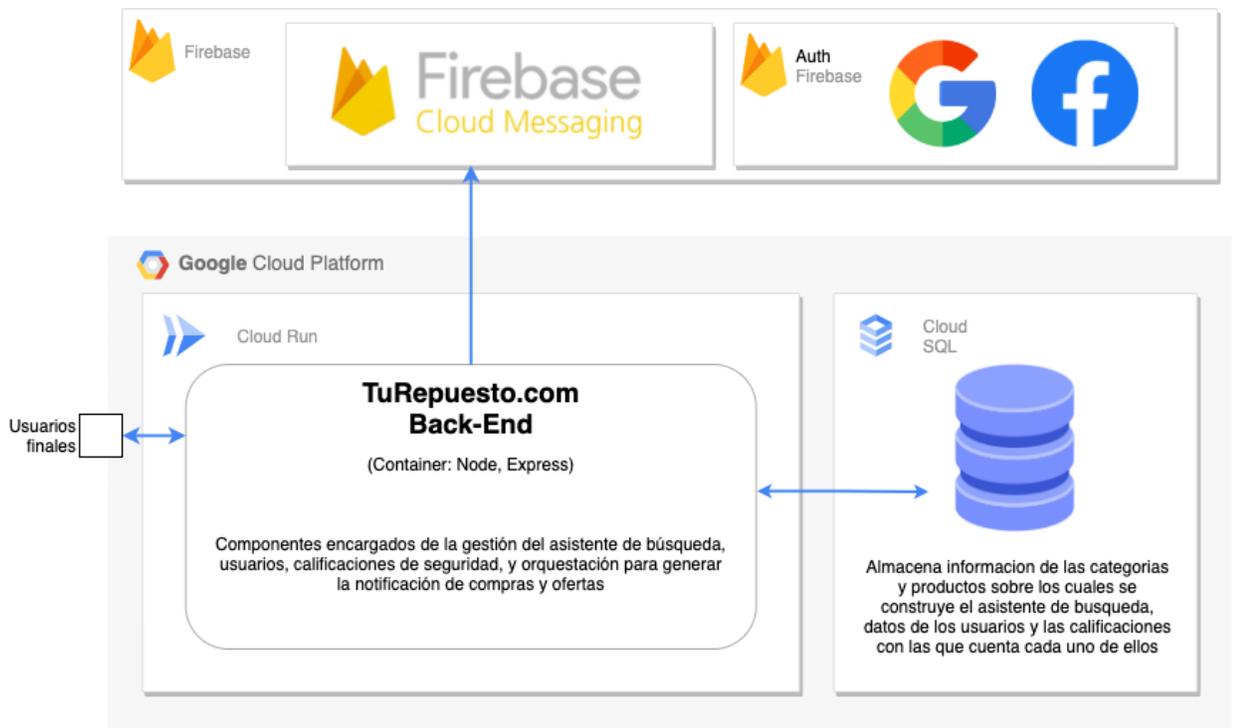
Nota: Elaboración propia (2023)

En este diagrama, se puede observar que, existen dos roles que pueden interactuar con el sistema a través de un aplicativo móvil disponible en Android e iOS y el proceso de registro e inicio de sesión se realiza por medio de Google

aprovechando herramientas dispuestas por Firebase. Existe un componente de Back-End que está desplegado en la nube de Google y tendrá diferentes responsabilidades asociadas al proceso de negocio apoyándose en Firebase para el envío de notificaciones a vendedores y compradores.

- **Diagrama de contenedores:**

Ilustración 5 Diagrama de contenedores



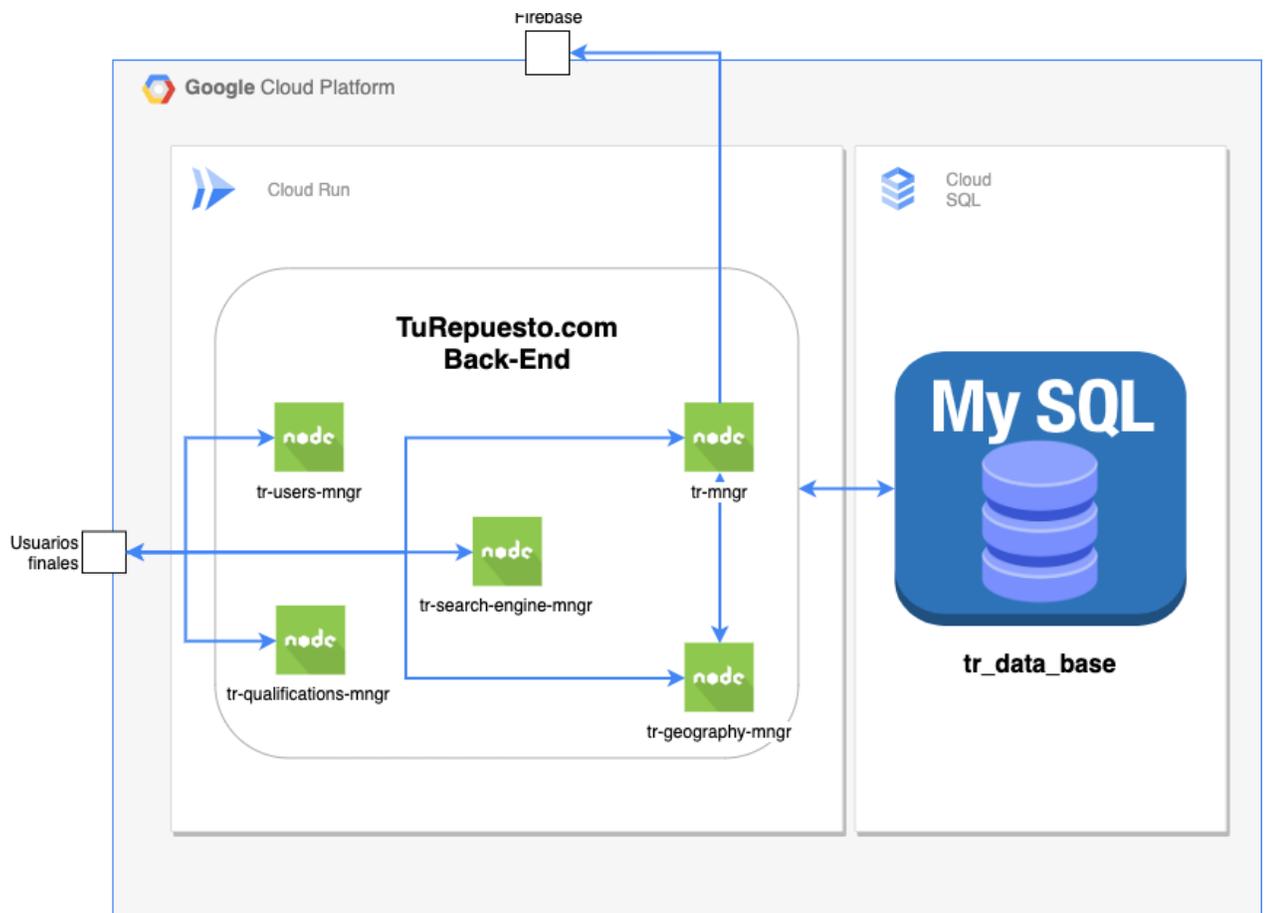
Nota: Elaboración propia (2023)

Profundizando más en la definición y diseño del sistema, el diagrama de contenedores muestra que el Back-End está compuesto por componentes hechos en node y express y desplegados con **Cloud Run** y que conviven e interactúan con una base de datos MySQL alojada en **Cloud SQL**.

Adicionalmente, se muestra que se aprovecharan dos tecnologías de Firebase, **Firestore** y **Cloud Messaging**, para gestionar el inicio de sesión a través de Google y el envío de notificaciones a compradores y vendedores respectivamente.

- **Diagrama de componentes:**

Ilustración 6 Diagrama de componentes



Nota. Elaboración propia (2023)

En cuanto al diagrama de componentes, se muestra a más detalle que en total son cinco microservicios los construidos para la gestión de las diferentes necesidades del software. La separación y asignación de responsabilidades de dichos microservicios se realiza con un enfoque de “Análisis de Dominio” por lo que cada uno se encarga de una parte específica del dominio y es responsable de las operaciones relacionadas:

- **tr-users-mngr:** Este microservicio es el encargado de gestionar todos los procesos de registro y lectura de información de los usuarios en base de datos.
- **tr-qualification-mngr:** Este Microservicio es el encargado de gestionar la calificación de vendedores y compradores y el retorno del promedio de dichas calificaciones para cada usuario.
- **tr-search-engine-mngr:** Este microservicio, es el encargado de proveer los datos necesarios para construir en el app el asistente de búsqueda, ya que es configurable debido a que el aplicativo tiene potencial para expandirse a un nicho de mercado más amplio que el de las motocicletas.
- **tr-mngr:** Este microservicio hace de orquestador al momento de publicar una compra o una oferta, por lo que se encarga de llamar a los microservicios requeridos para lograr satisfactoriamente la interacción entre computadores y vendedores.
- **tr-geography-mngr:** Este microservicio es el encargado de gestionar los procesos relacionados con la ubicación; en específico,

configuración de radios de búsqueda de compradores y definición de vendedores a quienes se debe notificar con base en los criterios de ubicación configurados por los diferentes compradores.

En conclusión, entendiendo que las arquitecturas no son mutuamente excluyentes; por el contrario, a menudo se solapan y la elección de la arquitectura depende de los requisitos y objetivos específicos de cada proyecto, se decide utilizar **Microservicios** para la administración de la lógica de negocio y **Cliente-Servidor** soportado a través de Firebase para el envío de notificaciones.

En cuanto a la elección de microservicios, la justificación radica en que se ha decidido separar la lógica bajo un enfoque de responsabilidad de negocio; es decir, cada microservicio responde a un proceso del negocio específico. Dicho enfoque, permite aportar de manera positiva a los siguientes apartados:

**Escalabilidad:** Al separarse la lógica del negocio en servicios independientes, es posible escalar cada uno de ellos de manera independiente en función de las diferentes situaciones o decisiones que se puedan presentar, aportando un mayor control sobre los recursos y una mayor capacidad de adaptación.

**Mantenibilidad:** Al separar la lógica de negocio en servicios independientes, cada uno de ellos se puede mantener y evolucionar de manera independiente, facilitando de este modo la identificación y corrección de errores, así como la implementación de nuevas funcionalidades.

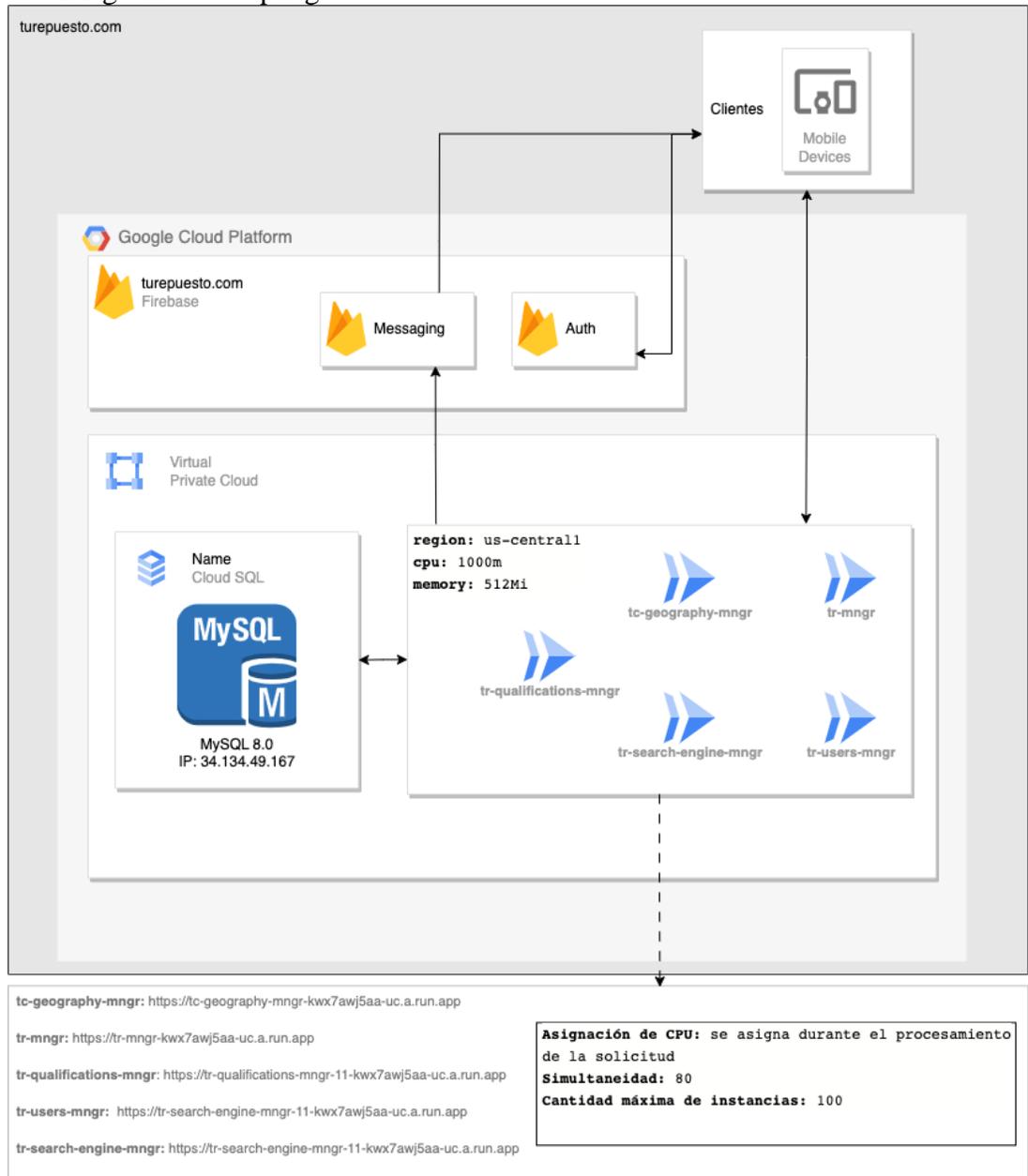
**Resiliencia:** Al tener servicios independientes, el fallo de uno de ellos no afecta al funcionamiento del resto del sistema, lo que aumenta la resiliencia del sistema en su conjunto.

Respecto a la base de datos, teniendo en cuenta las limitantes de tiempo y recursos, se ha decidido que en la primera versión del sistema, esta sea compartida. Una de las principales ventajas es la simplificación de la gestión de datos y la reducción de la complejidad del sistema en general. Dado que es necesario acceder a los mismos datos desde diferentes microservicios, esta decisión, en primera instancia, reduce tener que duplicar datos o integrar sistemas de bases de datos diferentes; permitiendo, adicionalmente, una mayor coherencia y consistencia de datos entre los diferentes microservicios.

## 5.4. DIAGRAMA DE DESPLIEGUE

A continuación, se expone el diagrama de despliegue del sistema diseñado y construido.

Ilustración 7 Diagrama de despliegue



Nota: Elaboración propia (2023)

El diagrama de despliegue incluye los componentes y servicios que se utilizan en la arquitectura del sistema. En concreto, se pueden identificar los siguientes elementos:

- **Google Cloud:** Es la plataforma de nube donde se alojan los microservicios y la base de datos.
- **Microservicios:** Diferentes microservicios del sistema desplegados en **Cloud Run**. Como nota, se exponen las URL de cada uno y la configuración de CPU, instancias y simultaneidad.
- **Cloud SQL:** Es la base de datos alojada en Google Cloud SQL que se utiliza para almacenar y recuperar información. Está protegida por una conexión privada mediante VPC, lo que limita el acceso solo a los microservicios desplegados en la misma VPC.
- **Firebase:** Servicio externo utilizado para la gestión de autenticación y el envío de notificaciones push a la aplicación móvil.
- **Aplicación móvil:** Es la aplicación desarrollada con Angular + Ionic que se comunica con los microservicios para recibir y enviar información. La aplicación también utiliza Firebase para la gestión de autenticación.

En resumen, el diagrama de despliegue muestra los diferentes componentes y servicios que componen la arquitectura del sistema, así como las conexiones entre ellos y la forma en que se comunican para proporcionar una experiencia coherente y segura tanto a los usuarios como a los desarrolladores.

## 5.5. CASO DE USO ARQUITECTURALMENTE RELEVANTE

Con el objetivo de exponer el comportamiento del sistema desde la perspectiva de los usuarios finales, se presenta el siguiente diagrama de caso de uso arquitecturalmente relevante, el cual está basado en casos de uso individuales por actor con el propósito de representar la usabilidad del desarrollo en un aspecto general.

En este diagrama se manifiestan usos más no vistas y se hace claridad de la interacción del modelo de negocio que conecta oferta con demanda y se evidencian los usos compartidos y exclusivos de los actores que hacen uso de turepuesto.com:

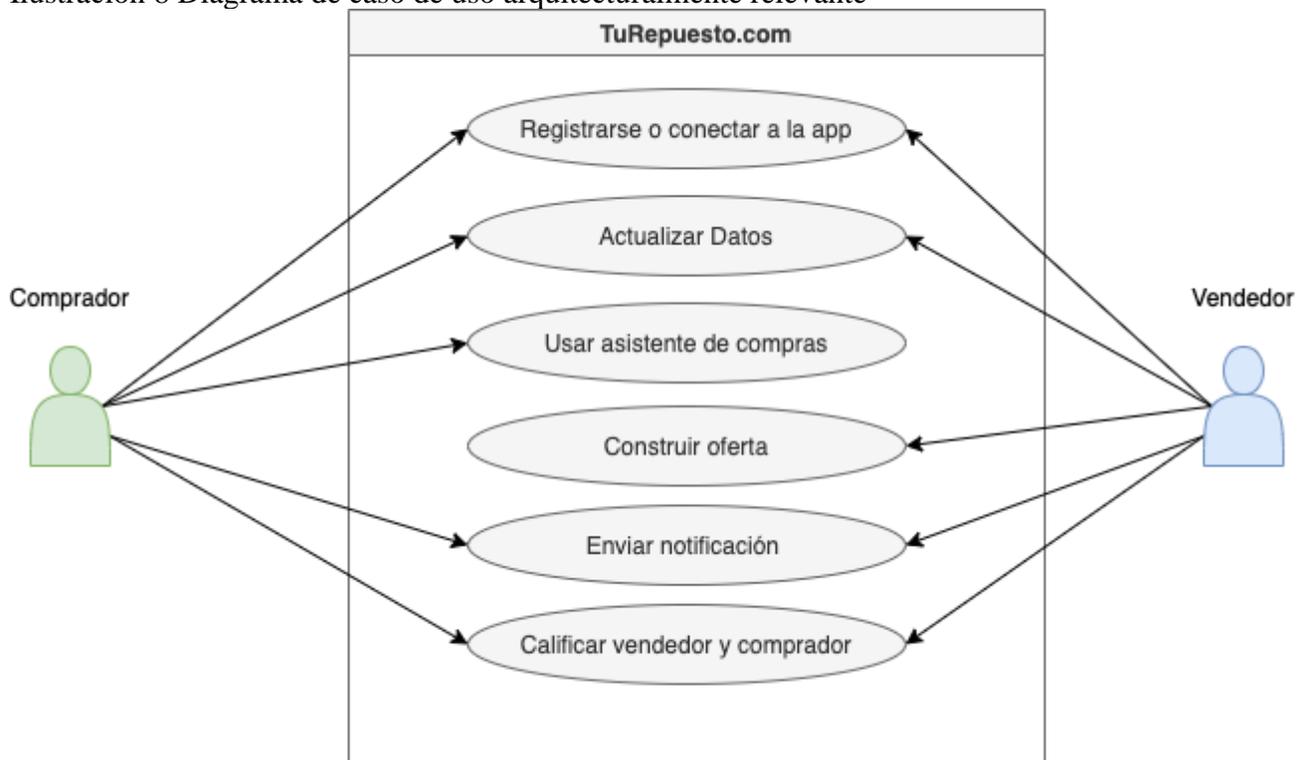
El Rol “Comprador” podrá hacer uso de:

- Registrarse y/o conectarse con la app.
- Actualizar datos.
- Usar asistente de compras.
- Enviar notificación.
- Calificar vendedor y comprador.

El Rol “Vendedor” podrá hacer uso de:

- Registrarse y/o conectarse a la app.
- Actualizar datos.
- Construir oferta
- Enviar notificación.
- Calificar vendedor y comprador.

Ilustración 8 Diagrama de caso de uso arquitecturalmente relevante



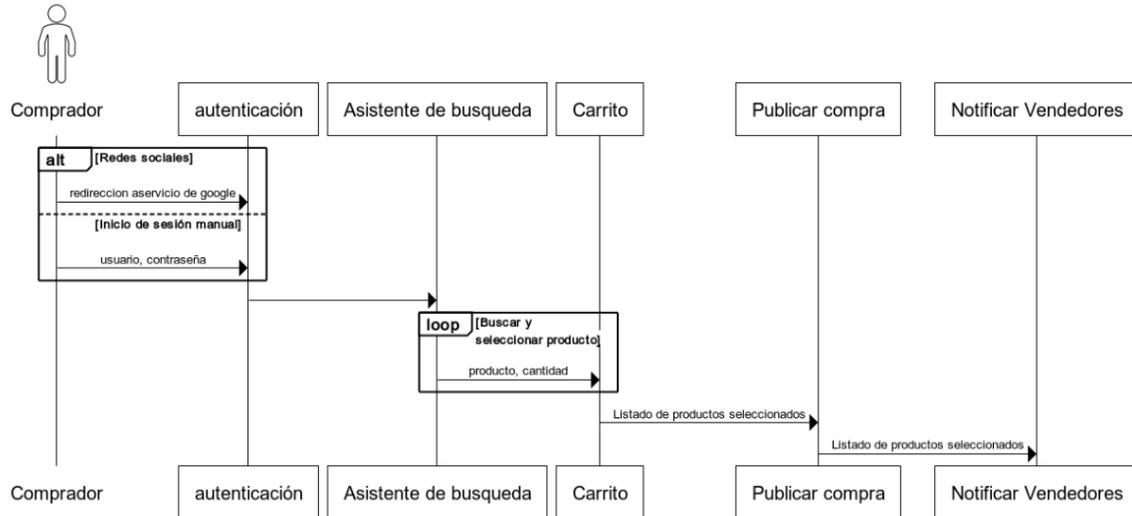
Nota: Elaboración propia (2023)

### 5.3. DIAGRAMA DE SECUENCIA

A continuación se presentan una serie de diagramas de secuencia que se utilizan para representar cómo interactúan los objetos de un sistema durante la ejecución de los procesos de negocio o los casos de uso. Estos diagramas son herramientas útiles para analizar un sistema y capturar la lógica del negocio de manera clara y precisa.

Diagrama de secuencia comprador:

Ilustración 9 Diagrama de secuencia comprador

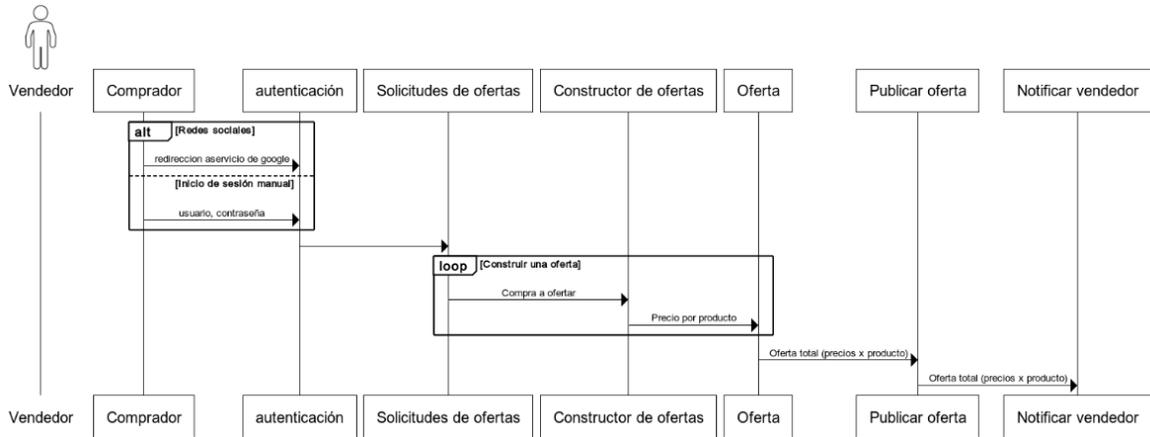


Nota: Elaboración propia (2023)

La secuencia del usuario comprador está determinada para iniciar realizando una autenticación de usuario y registrarse a la app a través de google y correo con contraseña, después de esto el usuario está listo para usar el asistente de búsqueda el cual está configurado partiendo de dos necesidades: Repuestos y accesorios, esto le permitir filtrar y organizar compra; posterior a esto lanzar compra para que sea notificado a vendedores.

## Diagrama de secuencia vendedor:

Ilustración 10 Diagrama de secuencia del vendedor



Nota: Elaboración propia (2023)

La secuencia del usuario vendedor está determinada para iniciar realizando una autenticación de usuario y registrarse a la app a través de google y correo con contraseña, después de esto podrá ver solicitudes que recibe en primer y segundo plano, esto para poder buscar, filtrar y seleccionar solicitudes y posteriormente publicar oferta a la solicitud la cuál es lanzada por medio de notificación al usuario comprador.



## 6. CONSTRUCCIÓN

En el presente capítulo, se describe el proceso de desarrollo de la aplicación desde la perspectiva técnica, incluyendo la construcción de la base de datos, el backend compuesto por microservicios y la integración con Firebase, el frontend basado en Angular e Ionic, el uso del flujo de trabajo Git Flow y el enfoque de CI/CD utilizado para garantizar la calidad del software y su entrega continua. Este capítulo es fundamental para comprender cómo se construyó la aplicación y cómo se aseguró su calidad en cada paso del proceso de desarrollo.

### 6.1. Base de datos

Para la gestión de datos en una aplicación en la nube, una opción conveniente es utilizar una base de datos alojada en la nube. En este caso, se ha utilizado Google Cloud SQL para crear una base de datos MySQL.

Una vez creada la instancia de la base de datos, se configuraron los permisos correspondientes para que los microservicios desplegados en Google Cloud Run puedan acceder a ella. Para esto, se creó un usuario en la base de datos con los permisos necesarios para realizar operaciones de lectura y escritura. Luego, se creó una conexión desde los microservicios a la instancia de la base de datos, utilizando las credenciales de este usuario.

Además, para tener un control más preciso sobre la evolución de la base de datos y contar con backups en caso de necesidad, se creó un repositorio en GitHub. En este repositorio se respaldaron todos los scripts y diferentes versiones de la base de datos utilizados en el proceso de creación y mantenimiento de la misma. Esto permite tener un registro completo de todas las modificaciones realizadas, y poder revertir a versiones anteriores en caso de ser necesario.

Ilustración 12 Instancia de la base de datos en Google Cloud



<input type="checkbox"/>	ID de instancia <span>?</span> <span>↑</span>	Tipo	Dirección IP pública	Acciones
<input type="checkbox"/>	<input checked="" type="checkbox"/> turepuestobd	MySQL 8.0	34.134.59.167	<span>⋮</span>

Nota: Elaboración propia (2023)

## 6.2. Git-Flow

Para el control de versiones del código desarrollado por los diferentes integrantes del equipo se adoptó un modelo de ramificación basado en Git Flow, enfocado en separar las ramas para las diferentes fases de desarrollo del proyecto.

En concreto, para cada uno de los repositorios se creó la rama **dev**, de la cual se crean ramas separadas para diferentes características o funciones llamadas

**feature branches.** Una vez que las características se han completado, se fusionan de nuevo en la rama de desarrollo y de allí se fusionan con la rama **staging** sobre la cual se certifican las nuevas características para finalmente, fusionarse con la rama **master** creando así, una nueva versión del software que será desplegada a través de CI/CD a la infraestructura correspondiente. La Ilustración 13, expone de manera más concreta las ramas establecidas para el control de versiones de código para cada uno de los repositorios del proyecto.

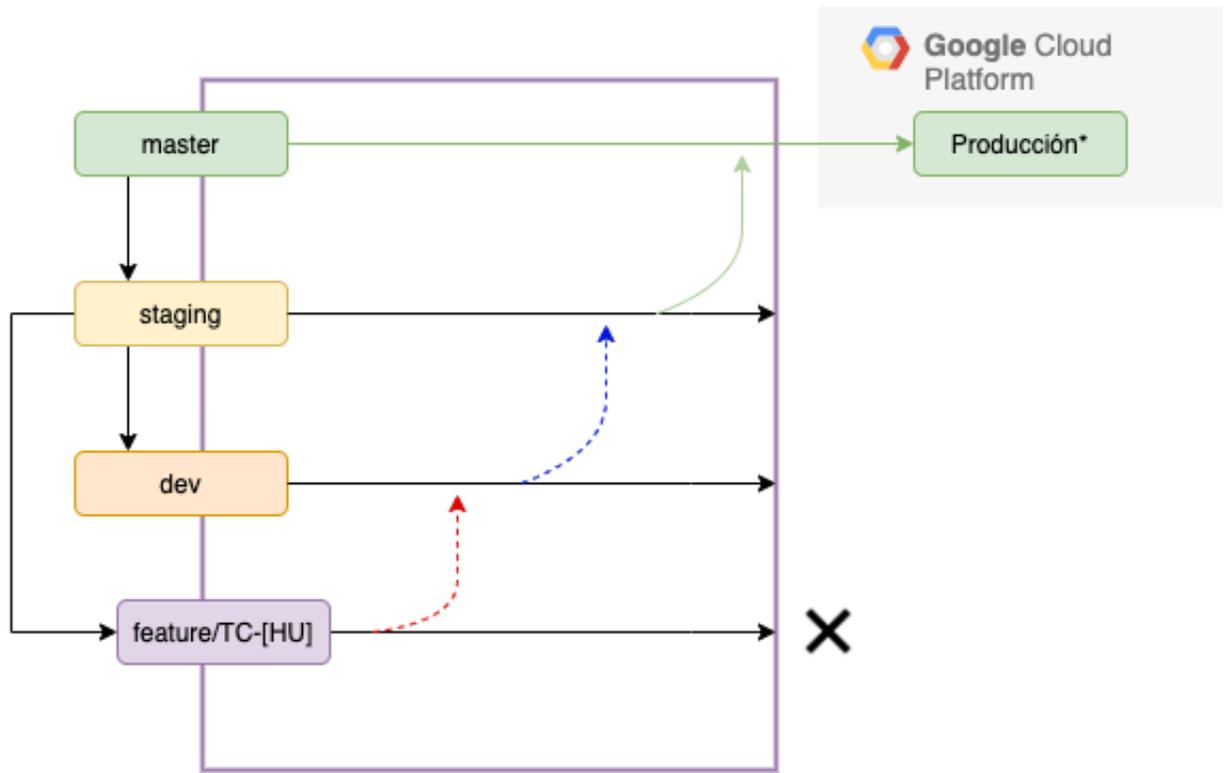
Ilustración 13 Ramas establecidas para el proceso de versionamiento de código

feature/TC-[HU]	Rama para el desarrollo de nuevas funcionalidades del producto
dev.	Rama estable en donde se fusionan inicialmente las nuevas características
staging	Rama en donde se certifican las nuevas funcionalidades
master	Rama principal en donde se fusionan las nuevas características ya certificadas y listas para ser desplegadas en la infraestructura

Nota: Elaboración propia (2023)

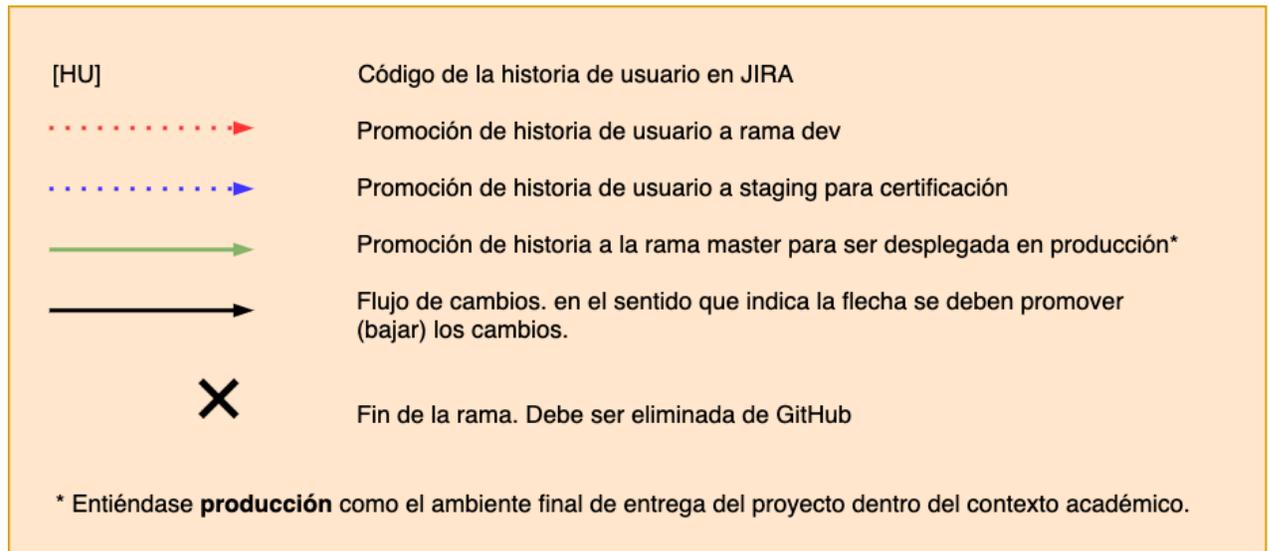
En cuanto al proceso de fusión entre ramas y promoción al ambiente productivo\*, la Ilustración 16 expone el proceso de manera gráfica y la Ilustración 15 detalla las convenciones pertinentes.

Ilustración 14 Proceso de fusión entre ramas y promoción al ambiente productivo



Nota: Elaboración propia (2023)

## Ilustración 15 Convenciones



Nota: Elaboración propia (2023)

### 6.3. Back-End

#### 6.3.1. Microservicios:

La planificación fue una etapa crucial en la construcción de los microservicios, ya que se buscaba una arquitectura adecuada que permitiera la escalabilidad, el mantenimiento y la modularidad de los servicios. Se utilizó el concepto de "Portafolio de servicios" de ITIL para definirlos por dominio y posteriormente, se establecieron las responsabilidades concretas de cada uno en relación con los objetivos del proyecto.

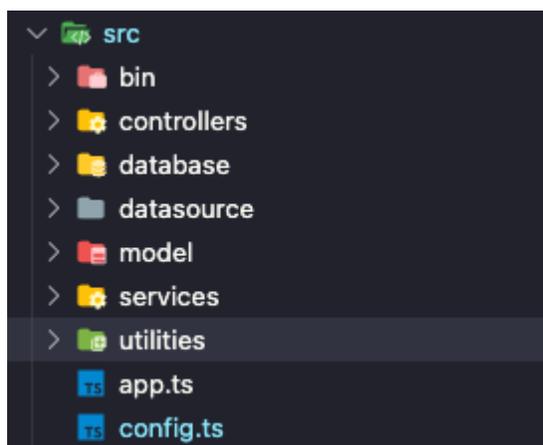
Los microservicios fueron desarrollados utilizando Node.js 18 y Express 4. Como se describió en el marco teórico, Node.js es una plataforma basada en JavaScript que permite el desarrollo de aplicaciones de alta

escalabilidad y Express es un framework para Node.js que facilita la creación de aplicaciones web. Se utilizó Visual Studio Code como editor de desarrollo.

Con respecto a código, los microservicios se estructuraron en cuatro capas principales (La Ilustración 18 muestra las capas mencionadas):

- **Controller:** Esta capa maneja las solicitudes HTTP y las respuestas correspondientes.
- **Service:** Esta capa maneja la lógica de negocio y se comunica con la capa de datos.
- **Datasource:** Esta capa se encarga de la interacción con la base de datos y la ejecución de consultas.
- **Fetch:** Esta capa maneja la comunicación con otros microservicios.

Ilustración 16 Capas implementadas a nivel de código



Nota: Elaboración propia (2023)

También es importante mencionar que, se realizaron pruebas unitarias. Las pruebas unitarias se centraron en la verificación de la funcionalidad individual de cada uno de los componentes.

En conclusión, la construcción de los microservicios fue un proceso complejo que requirió una planificación adecuada y una estructura bien definida. Se utilizaron las tecnologías y herramientas pertinentes para cada una de las capas de los microservicios, se usó un flujo de trabajo basado en GitFlow y se implementaron pruebas unitarias.

#### 6.3.2. Firebase (Notificaciones)

Para la integración con Firebase se siguieron algunos pasos importantes. En primer lugar, se creó un proyecto en Firebase Console y se habilitaron los servicios necesarios para el envío de notificaciones push.

Posteriormente, se trabajó en la implementación del envío de notificaciones push mediante Cloud Messaging. Para ello, se implementó un microservicio específico encargado de la gestión de las notificaciones y se utilizó la librería de **Firestore Admin SDK** para Node.js.

Una vez configurado y probado, se realizó la integración con el microservicio correspondiente para que pudiera enviar notificaciones a los



## 6.4. Front-End

Para la construcción del Front-End, se decidió utilizar Angular 12 e Ionic 4, debido a su alta compatibilidad y facilidad de uso. El proyecto se organizó en una estructura basada en componentes, donde cada componente es responsable de una vista específica dentro de la aplicación.

Ilustración 18 Front-End compilado en web

The image shows a web browser window displaying a login page. The page has a dark blue header with the text "¡BIENVENIDO!". Below the header, there is a form with a text input field containing "ansantacruz93@gmail.com" and a password input field with masked characters "\*\*\*\*\*". A message below the form states: "La contraseña debe tener mínimo 6 caracteres y contener al menos una letra mayúscula y un número." There is a link that says "¿No recuerda su contraseña? Haga Clic Aquí". A yellow button labeled "INICIAR SESIÓN" is positioned below the form. Below the button, there is a horizontal line with "Or" in the center, followed by two circular icons for Facebook and Google. At the bottom, there is a link "¿No es un miembro?" and a button labeled "Crea Una Cuenta".

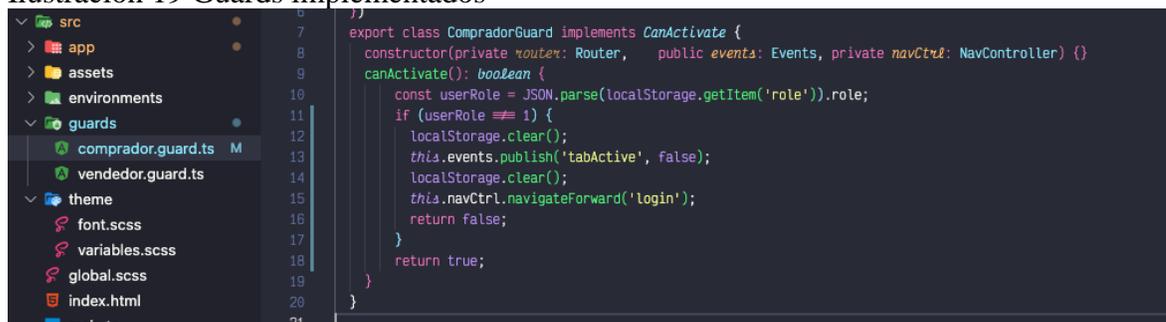
On the right side of the browser window, the developer tools are open, showing the "Elements" panel. The selected element is an Ionic component structure, including a shadow root, an icon, and an input field. The "Console" panel is also visible, showing a list of messages: "No user messages", "No errors", "No warnings", "No info", and "No verbose". The "Filter" section is expanded, showing options like "Hide network", "Preserve log", "Selected context only", "Group similar messages in console", and "Show CORS errors in console".

Nota: Elaboración propia (2023)

El equipo de desarrollo se dividió en dos partes: una responsable del rol Comprador y otra del rol Vendedor. Se comenzó por la construcción de las ventanas del rol Comprador y su conexión con el back-end, utilizando datos de usuarios mockeados. Posteriormente, se trabajó en las ventanas del rol Vendedor y su conexión con el back-end.

El control de roles en la aplicación se realizó mediante guards de Angular, los cuales son servicios que se encargan de proteger las rutas y componentes de la aplicación. Se definieron diferentes guards para los roles de comprador y vendedor, y estos se encargaron de permitir o denegar el acceso a las rutas y componentes correspondientes a cada rol. De esta manera, se logró garantizar la seguridad y la privacidad de los datos de la aplicación, siguiendo prácticas de seguridad recomendadas en el desarrollo de aplicaciones web. (La Ilustración 19 muestra los GUARDS implementados)

Ilustración 19 Guards implementados



```
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60  
61  
62  
63  
64  
65  
66  
67  
68  
69  
70  
71  
72  
73  
74  
75  
76  
77  
78  
79  
80  
81  
82  
83  
84  
85  
86  
87  
88  
89  
90  
91  
92  
93  
94  
95  
96  
97  
98  
99  
100  
101  
102  
103  
104  
105  
106  
107  
108  
109  
110  
111  
112  
113  
114  
115  
116  
117  
118  
119  
120  
121  
122  
123  
124  
125  
126  
127  
128  
129  
130  
131  
132  
133  
134  
135  
136  
137  
138  
139  
140  
141  
142  
143  
144  
145  
146  
147  
148  
149  
150  
151  
152  
153  
154  
155  
156  
157  
158  
159  
160  
161  
162  
163  
164  
165  
166  
167  
168  
169  
170  
171  
172  
173  
174  
175  
176  
177  
178  
179  
180  
181  
182  
183  
184  
185  
186  
187  
188  
189  
190  
191  
192  
193  
194  
195  
196  
197  
198  
199  
200  
201  
202  
203  
204  
205  
206  
207  
208  
209  
210  
211  
212  
213  
214  
215  
216  
217  
218  
219  
220  
221  
222  
223  
224  
225  
226  
227  
228  
229  
230  
231  
232  
233  
234  
235  
236  
237  
238  
239  
240  
241  
242  
243  
244  
245  
246  
247  
248  
249  
250  
251  
252  
253  
254  
255  
256  
257  
258  
259  
260  
261  
262  
263  
264  
265  
266  
267  
268  
269  
270  
271  
272  
273  
274  
275  
276  
277  
278  
279  
280  
281  
282  
283  
284  
285  
286  
287  
288  
289  
290  
291  
292  
293  
294  
295  
296  
297  
298  
299  
300  
301  
302  
303  
304  
305  
306  
307  
308  
309  
310  
311  
312  
313  
314  
315  
316  
317  
318  
319  
320  
321  
322  
323  
324  
325  
326  
327  
328  
329  
330  
331  
332  
333  
334  
335  
336  
337  
338  
339  
340  
341  
342  
343  
344  
345  
346  
347  
348  
349  
350  
351  
352  
353  
354  
355  
356  
357  
358  
359  
360  
361  
362  
363  
364  
365  
366  
367  
368  
369  
370  
371  
372  
373  
374  
375  
376  
377  
378  
379  
380  
381  
382  
383  
384  
385  
386  
387  
388  
389  
390  
391  
392  
393  
394  
395  
396  
397  
398  
399  
400  
401  
402  
403  
404  
405  
406  
407  
408  
409  
410  
411  
412  
413  
414  
415  
416  
417  
418  
419  
420  
421  
422  
423  
424  
425  
426  
427  
428  
429  
430  
431  
432  
433  
434  
435  
436  
437  
438  
439  
440  
441  
442  
443  
444  
445  
446  
447  
448  
449  
450  
451  
452  
453  
454  
455  
456  
457  
458  
459  
460  
461  
462  
463  
464  
465  
466  
467  
468  
469  
470  
471  
472  
473  
474  
475  
476  
477  
478  
479  
480  
481  
482  
483  
484  
485  
486  
487  
488  
489  
490  
491  
492  
493  
494  
495  
496  
497  
498  
499  
500  
501  
502  
503  
504  
505  
506  
507  
508  
509  
510  
511  
512  
513  
514  
515  
516  
517  
518  
519  
520  
521  
522  
523  
524  
525  
526  
527  
528  
529  
530  
531  
532  
533  
534  
535  
536  
537  
538  
539  
540  
541  
542  
543  
544  
545  
546  
547  
548  
549  
550  
551  
552  
553  
554  
555  
556  
557  
558  
559  
560  
561  
562  
563  
564  
565  
566  
567  
568  
569  
570  
571  
572  
573  
574  
575  
576  
577  
578  
579  
580  
581  
582  
583  
584  
585  
586  
587  
588  
589  
590  
591  
592  
593  
594  
595  
596  
597  
598  
599  
600  
601  
602  
603  
604  
605  
606  
607  
608  
609  
610  
611  
612  
613  
614  
615  
616  
617  
618  
619  
620  
621  
622  
623  
624  
625  
626  
627  
628  
629  
630  
631  
632  
633  
634  
635  
636  
637  
638  
639  
640  
641  
642  
643  
644  
645  
646  
647  
648  
649  
650  
651  
652  
653  
654  
655  
656  
657  
658  
659  
660  
661  
662  
663  
664  
665  
666  
667  
668  
669  
670  
671  
672  
673  
674  
675  
676  
677  
678  
679  
680  
681  
682  
683  
684  
685  
686  
687  
688  
689  
690  
691  
692  
693  
694  
695  
696  
697  
698  
699  
700  
701  
702  
703  
704  
705  
706  
707  
708  
709  
710  
711  
712  
713  
714  
715  
716  
717  
718  
719  
720  
721  
722  
723  
724  
725  
726  
727  
728  
729  
730  
731  
732  
733  
734  
735  
736  
737  
738  
739  
740  
741  
742  
743  
744  
745  
746  
747  
748  
749  
750  
751  
752  
753  
754  
755  
756  
757  
758  
759  
760  
761  
762  
763  
764  
765  
766  
767  
768  
769  
770  
771  
772  
773  
774  
775  
776  
777  
778  
779  
780  
781  
782  
783  
784  
785  
786  
787  
788  
789  
790  
791  
792  
793  
794  
795  
796  
797  
798  
799  
800  
801  
802  
803  
804  
805  
806  
807  
808  
809  
810  
811  
812  
813  
814  
815  
816  
817  
818  
819  
820  
821  
822  
823  
824  
825  
826  
827  
828  
829  
830  
831  
832  
833  
834  
835  
836  
837  
838  
839  
840  
841  
842  
843  
844  
845  
846  
847  
848  
849  
850  
851  
852  
853  
854  
855  
856  
857  
858  
859  
860  
861  
862  
863  
864  
865  
866  
867  
868  
869  
870  
871  
872  
873  
874  
875  
876  
877  
878  
879  
880  
881  
882  
883  
884  
885  
886  
887  
888  
889  
890  
891  
892  
893  
894  
895  
896  
897  
898  
899  
900  
901  
902  
903  
904  
905  
906  
907  
908  
909  
910  
911  
912  
913  
914  
915  
916  
917  
918  
919  
920  
921  
922  
923  
924  
925  
926  
927  
928  
929  
930  
931  
932  
933  
934  
935  
936  
937  
938  
939  
940  
941  
942  
943  
944  
945  
946  
947  
948  
949  
950  
951  
952  
953  
954  
955  
956  
957  
958  
959  
960  
961  
962  
963  
964  
965  
966  
967  
968  
969  
970  
971  
972  
973  
974  
975  
976  
977  
978  
979  
980  
981  
982  
983  
984  
985  
986  
987  
988  
989  
990  
991  
992  
993  
994  
995  
996  
997  
998  
999  
1000
```

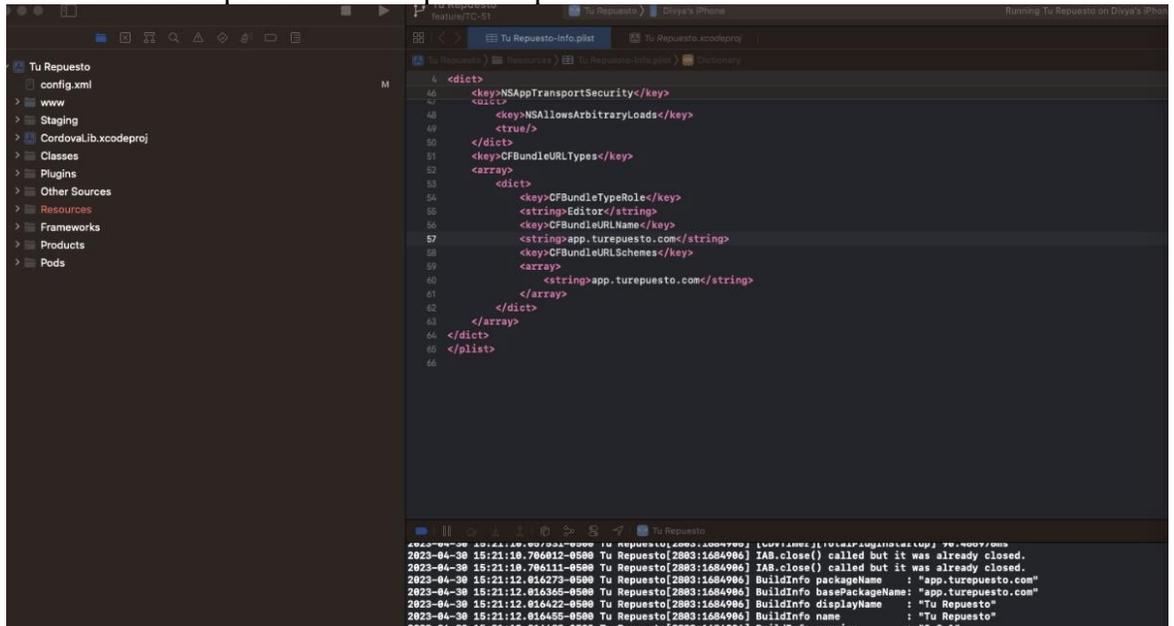
Nota: Elaboración propia (2023)

Se implementó el patrón de nomenclatura Block Element Modifier (BEM) para los estilos del front-end, siguiendo las prácticas de clean code. Esto permitió

tener un código CSS estructurado, escalable y fácil de mantener, además de evitar conflictos y problemas de especificidad en los estilos. La implementación de BEM también ayudó a separar la lógica de presentación del HTML y a mejorar la reutilización de componentes en diferentes secciones de la aplicación.

Una vez construido el Front-end, se procedió a la configuración e implementación de Firebase para la gestión de la autenticación mediante Google. Asimismo, se trabajó en el proceso de compilación para iOS y Android, con el objetivo de lanzar la aplicación en ambas plataformas.

#### Ilustración 20 Xcode en proceso de compilación para iOS



Nota: Elaboración propia (2023)

Como resultado, se logró construir una aplicación Front-end robusta y escalable, que se integra perfectamente con el back-end y Firebase. La aplicación se lanzó con éxito para los sistemas operativos de iOS y Android.

## 6.5. CI/CD

La implementación de CI/CD (Integración continua/Despliegue Continuo) se consideró un elemento fundamental en el desarrollo del proyecto, ya que permitió una integración temprana y frecuente de los cambios en el código y una entrega constante y confiable al ambiente de producción.

Para garantizar la calidad del código fuente y mantener la coherencia en la estructura del proyecto, se integró una herramienta de análisis estático de código en el proceso de CI/CD (se abordará este apartado en el siguiente capítulo).

## Ilustración 21 Pipeline en GitHub Actions para análisis con SonarCloud

```
13 jobs:
14   sonarcloud:
15     name: SonarCloud
16     runs-on: ubuntu-latest
17     steps:
18       - uses: actions/checkout@v2
19         with:
20           fetch-depth: 0
21       - name: Use Node.js
22         uses: actions/setup-node@v2
23         with:
24           node-version: 16.x
25       - name: Install Dependencies
26         run: |
27           npm install --force
28       - name: Run Coverage
29         shell: bash
30         run: |
31           npm run test && npm run coverage && npm run coverage:report
32       - name: SonarCloud Scan
33         uses: SonarSource/sonarcloud-github-action@master
34         env:
35           GITHUB_TOKEN: ${ secrets.GITHUB_TOKEN }
36           SONAR_TOKEN: ${ secrets.SONAR_TOKEN }
37
```

Nota: Elaboración propia (2023)

En cuanto al proceso de despliegue continuo, también se hizo uso de pipelines de GitHub Actions. Estos pipelines se ejecutaron en cada integración a la rama master (tal como se mencionó anteriormente) permitiendo la compilación y posterior despliegue en la nube de Google.

## Ilustración 22 Pipeline en GitHub Actions para despliegue en Google Cloud

```

1 name: Artifact Registry CI/CD
2 on:
3   push:
4     branches:
5       - master
6 jobs:
7   build-and-deploy:
8     runs-on: ubuntu-latest
9     steps:
10    - name: Checkout code
11      uses: actions/checkout@v2
12    - name: Set up Node.js
13      uses: actions/setup-node@v2
14      with:
15        node-version: 14.x
16    - name: Install dependencies
17      run: npm ci
18    - name: Build Docker image
19      run: |
20        docker build -t gcr.io/${{ secrets.GCLOUD_PROJECT_ID }}/my-node-app:${{ github.sha }} -t gcr.io/${{ secrets.GCLOUD_PROJECT_ID }}/my-node-app:latest .
21    - name: Push Docker image to Artifact Registry
22      run: |
23        docker push gcr.io/${{ secrets.GCLOUD_PROJECT_ID }}/my-node-app:${{ github.sha }}
24        docker push gcr.io/${{ secrets.GCLOUD_PROJECT_ID }}/my-node-app:latest
25    - name: Configure gcloud SDK
26      uses: google-github-actions/setup-gcloud@v0.3.0
27      with:
28        project_id: ${{ secrets.GCLOUD_PROJECT_ID }}
29        service_account_key: ${{ secrets.GCLOUD_AUTH_KEY }}
30    - name: Deploy to Cloud Run
31      run: |
32        gcloud run deploy my-node-app \
33          --image gcr.io/${{ secrets.GCLOUD_PROJECT_ID }}/my-node-app:${{ github.sha }} \
34          --platform managed \
35          --region us-central1 \
36          --allow-unauthenticated
  
```

Nota: Elaboración propia (2023)

## Ilustración 23 Microservicios desplegados en CloudRun

**Filtro** Filtrar servicios

<input type="checkbox"/>	<input checked="" type="checkbox"/>	Nombre <span>↑</span>	Solicitudes/seg <span>?</span>	Región
<input type="checkbox"/>	<input checked="" type="checkbox"/>	<a href="#">tc-geography-mngr</a>	0	us-central1
<input type="checkbox"/>	<input checked="" type="checkbox"/>	<a href="#">tr-mngr</a>	0	us-central1
<input type="checkbox"/>	<input checked="" type="checkbox"/>	<a href="#">tr-qualifications-mngr</a>	0	us-central1
<input type="checkbox"/>	<input checked="" type="checkbox"/>	<a href="#">tr-search-engine-mngr</a>	0	us-central1
<input type="checkbox"/>	<input checked="" type="checkbox"/>	<a href="#">tr-users-mngr</a>	0	us-central1

Nota: Elaboración propia (2023)

La implementación de un proceso de CI/CD permitió una entrega temprana y frecuente de los cambios en el código, mantuvo la calidad del software, redujo los tiempos de respuesta en caso de errores y permitió una mayor colaboración y

comunicación entre el equipo de desarrollo. En definitiva, fue una práctica clave para el éxito del proyecto.

## 7. PRUEBAS

El proceso de pruebas es una etapa crítica en el desarrollo de software, ya que permite detectar y corregir errores y asegurar que el producto final cumpla con las especificaciones y necesidades del usuario. En el presente proyecto, se llevaron a cabo diferentes tipos de pruebas para garantizar la calidad del producto.

### 7.1. Pruebas Unitarias

En primer lugar, se realizaron pruebas unitarias en todos los repositorios del proyecto. Para el backend, que fue construido en Node.js, se utilizó la herramienta de pruebas Mocha, mientras que para el frontend se utilizó Karma. Estas pruebas permitieron verificar que cada componente funcione de manera aislada y que las funciones y métodos cumplan con su propósito específico.

Ilustración 24 Pruebas unitarias en Back-End



```
105
106
107
108
109
110
111
112
113
114
115
116
117
it('SparePartsService getSapareParts reject', () => {
  sinon.replace(SparePartsDataSource, 'getSapareParts', (): Promise<any> => {
    return Promise.reject(REJECT_RESPONSE);
  });

  SparePartsService.getSapareParts(1,1)
    .catch((err) => {
      expect(err.Status.StatusCode).equal(404);
    });
});
```

Nota: Elaboración propia (2023)

Ilustración 25 Pruebas unitarias compiladas en local

```
✓ SparePartsService getSapareParts reject

DebugUtilities
✓ should return reason as error instance
✓ should return reason no error instance
✓ should return defined error
✓ should return general error

RequestLogger
✓ should return X-RqUID header valid

53 passing (106ms)

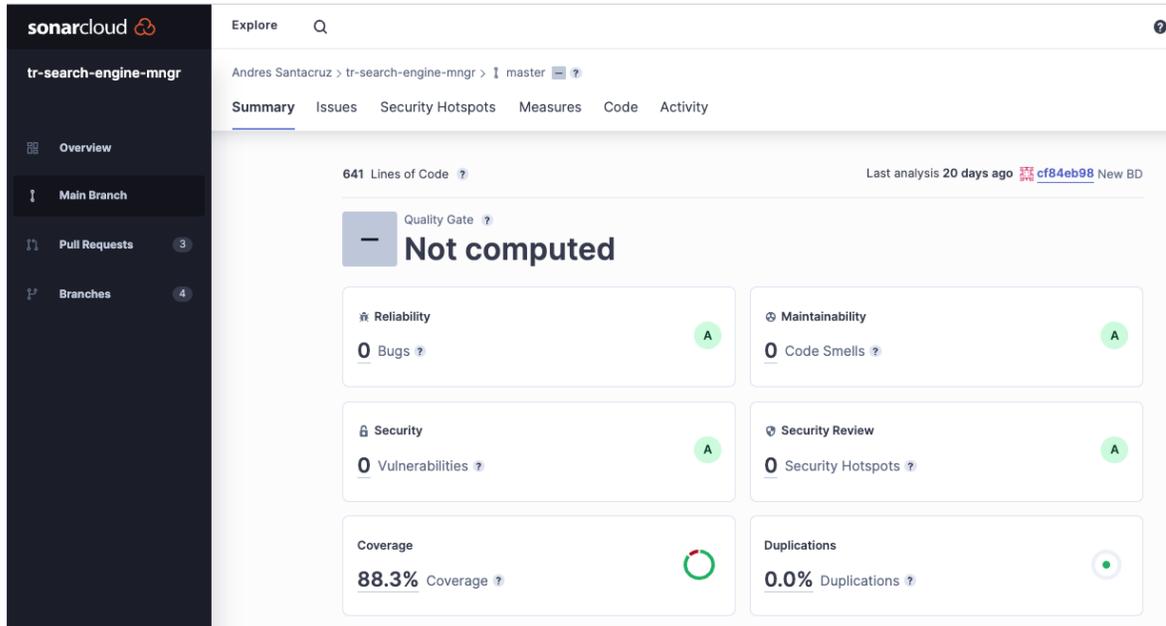
BDMC9081992:tr-search-engine-mngr andrescamilosantacruzborde$
```

Nota: Elaboración propia (2023)

## 7.2. Análisis Automático en SonarCloud

Para garantizar la calidad del código y evitar posibles errores en la implementación, se implementó el análisis automático en SonarCloud mediante GitHub Actions. Estos análisis se ejecutaron después de cada commit y evaluaron que todos los ítems estuvieran en A y que la cobertura de pruebas fuera superior al 80%. Esto permitió identificar y corregir errores de manera oportuna.

Ilustración 26 Resultado de análisis de SonarCloud



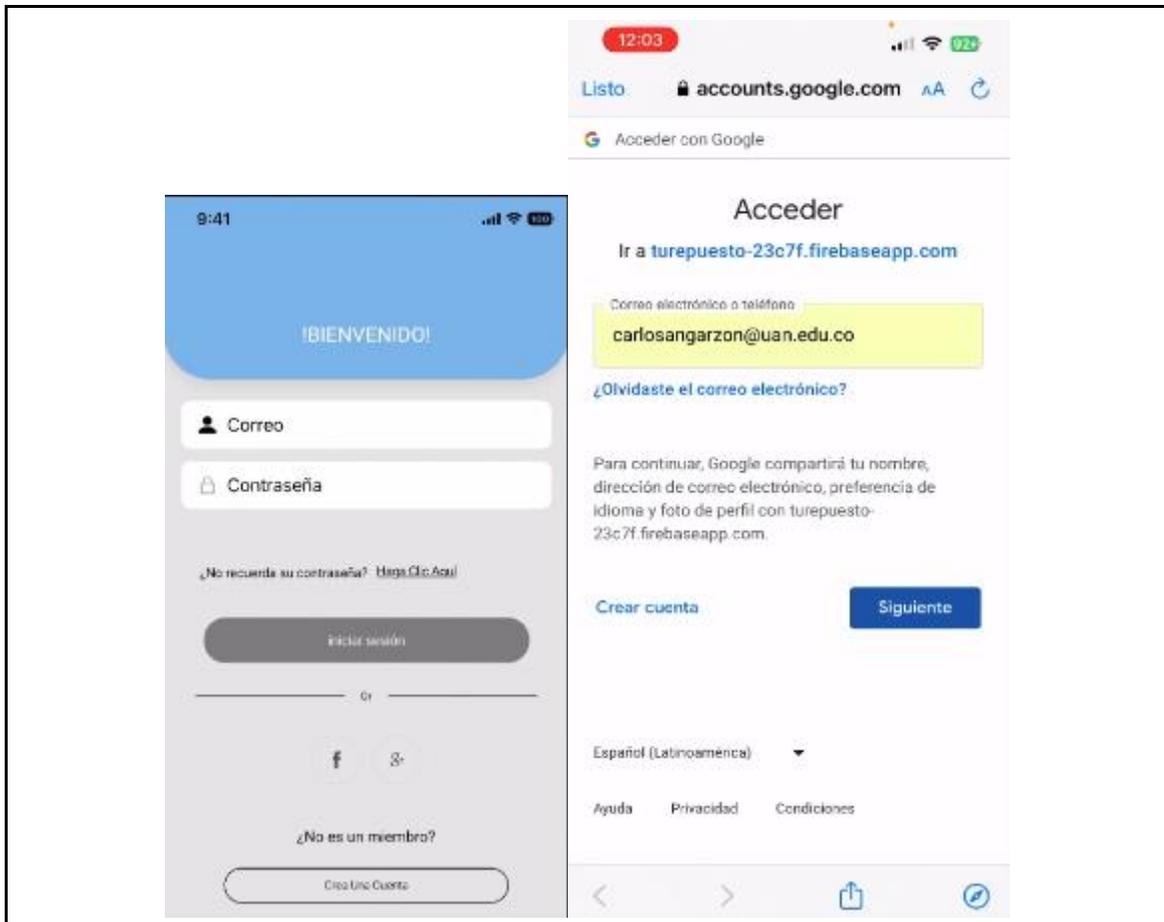
Nota: Elaboración propia (2023)

### 7.3. Pruebas de Funcionalidad

Se realizaron pruebas de funcionalidad a dos historias de usuario, el inicio de sesión y el asistente de búsqueda. Estas pruebas se llevaron a cabo manualmente para comprobar que la funcionalidad de las aplicaciones funcionara correctamente y que la interacción con el usuario fuera intuitiva y satisfactoria. Se utilizaron datos mock para realizar las pruebas.

Tabla 19 Prueba funcional 1

<b>Descripción de la Funcionalidad</b>
Permitir el registro y la conexión a la aplicación móvil “Turepuestocom” usando dispositivo móvil que opere con sistema operativo IOS o ANDROID.
<b>Condiciones de la prueba</b>
<ol style="list-style-type: none"> <li>1. El usuario debe poseer una cuenta de correo electrónico.</li> <li>2. El usuario debe conocer las credenciales de dicho correo electrónico.</li> <li>3. El usuario debe disponer de la apertura de su correo electrónico en los equipos de cómputo proporcionados.</li> <li>4. El usuario debe disponer de su cuenta de correo electrónico para realizar ingreso a la aplicación móvil.</li> </ol>
<b>Objetivos de la prueba</b>
Validar que llegue el correo de confirmación de registro a una cuenta de correo electrónico elegida por el usuario participante de la prueba.
<b>Elementos requeridos para la prueba</b>
<ol style="list-style-type: none"> <li>1. Sala de cómputo con ordenadores de escritorio o portátiles.</li> <li>2. Equipos móviles (celular o tablet) con los sistemas operativos IOS y Android.</li> <li>3. Conexión continua al servicio de internet de manera alámbrica e inalámbrica</li> </ol>
<b>Lista de ítems que deberán ser probados</b>
<ol style="list-style-type: none"> <li>1. El usuario debe dar click en el botón iniciar sesión.</li> <li>2. El usuario debe iniciar sesión con Google.</li> <li>3. El usuario debe registrar un correo electrónico en la pantalla.</li> <li>4. El usuario debe validar el correo de confirmación de creación de cuenta.</li> <li>5. El usuario deberá ingresar el correo inicial y proporcionar una clave para ingresar.</li> </ol>



### Lista de items que no deben ser probados

1. No se debe probar la función crear cuenta.
2. No se debe probar la función “¿no recuerda su contraseña?”.
3. No se debe probar las funcionalidades de la siguiente pantalla.

### Entregables en la documentación de pruebas

1. Captura de pantalla de la llegada del correo de validación de Registro a la aplicación móvil.
2. Datos personales de los participantes de la prueba, (nombre, número de teléfono y dirección de correo electrónico).

### Roles y responsabilidades

1. Coordinador de prueba, Responsable de dirigir a la persona objeto de la prueba.
2. Analista de calidad, Responsable de documentar el flujo de la prueba.

### Plan de trabajo General

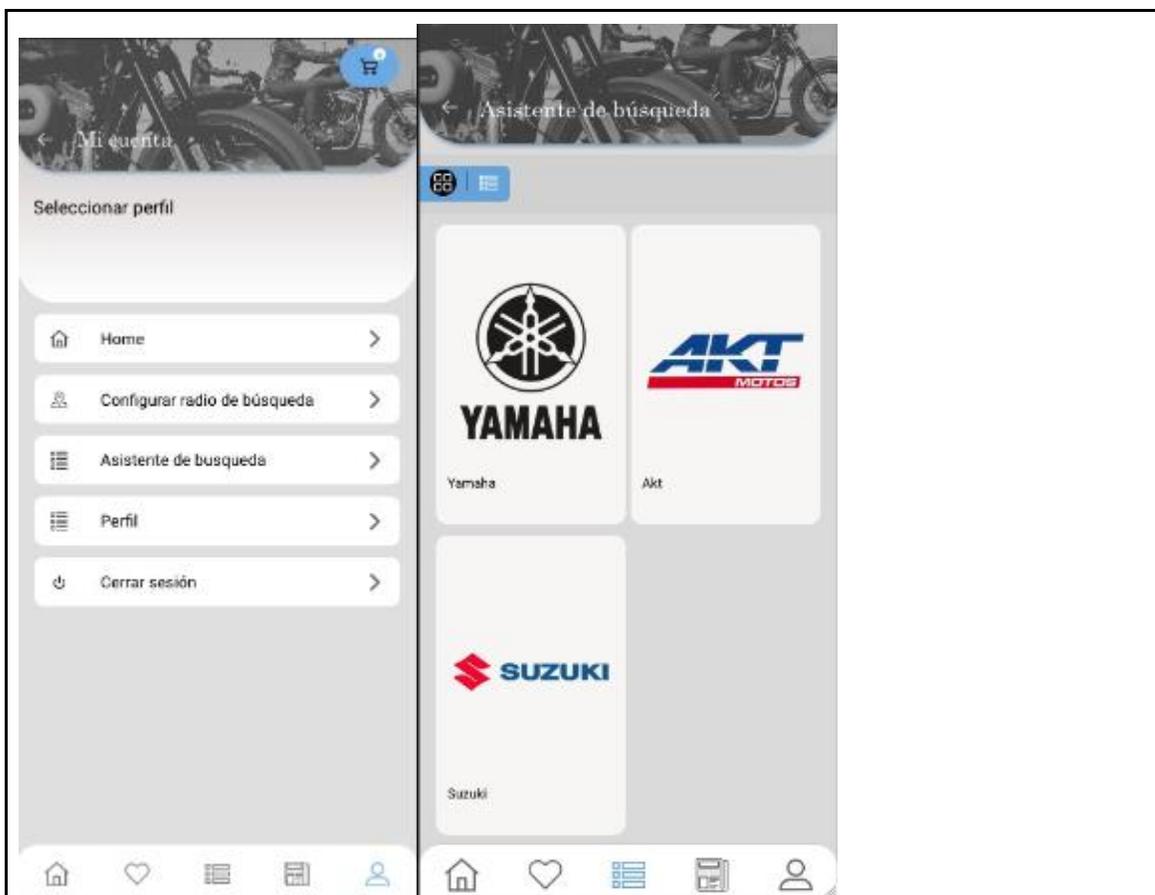
1. Presentación de la aplicación (breve resumen de 10 minutos).
2. Explicación de la prueba a realizar (5 minutos).

3. Ejecución de la prueba (5 minutos).
<b>Riesgos identificados a la hora de realizar las pruebas</b>
<ol style="list-style-type: none"> <li>1. Ausencia de servicio eléctrico.</li> <li>2. Los equipos de cómputo presentan fallas.</li> <li>3. El servicio de Internet falla.</li> <li>4. Los dispositivos móviles presentan fallas.</li> <li>5. La persona objeto de la prueba no tiene claras las credenciales de su correo electrónico.</li> <li>6. La persona objeto de la prueba ve con desconfianza proporcionar el correo electrónico.</li> </ol>

Nota: Elaboración propia (2023)

Tabla 20 Prueba funcional 2

<b>Descripción de la Funcionalidad</b>
Navegar el asistente de búsqueda de la aplicación móvil “Turepuesto.com” por un usuario usando dispositivo móvil que opere con sistema operativo IOS o ANDROID.
<b>Condiciones de la prueba</b>
<ol style="list-style-type: none"> <li>1. El usuario debe dar click en el submenú asistente de búsqueda.</li> <li>2. Al dar click en dicho submenú debe redirigir al usuario a una nueva pantalla.</li> </ol>
<b>Objetivos de la prueba</b>
<ol style="list-style-type: none"> <li>1. Validar que el usuario navegue con confianza por el submenu del asistente de búsqueda.</li> <li>2. Validar que el asistente de búsqueda oriente al usuario valiéndose del multimedia.</li> </ol>



### Elementos requeridos para la prueba

1. Sala de cómputo con ordenadores de escritorio o portátiles.
2. Equipos móviles (celular o tablet) con los sistemas operativos IOS y Android.
3. Conexión continua al servicio de internet de manera alámbrica e inalámbrica

### Lista de ítems que deberán ser probados

1. El usuario debe dar click en el botón asistente de búsqueda.
2. El usuario debe navegar libremente por los diferentes submenús del asistente de búsqueda.

### Lista de ítems que no deben ser probados

1. Otros submenús de la aplicación están fuera del asistente de búsqueda en la pantalla principal.

### Entregables en la documentación de pruebas

1. Encuesta de satisfacción de prueba de uso
  - a. ¿Considera que el nombre asistente de búsqueda hace referencia a la funcionalidad expuesta? a la respuesta preguntar ¿Por qué?

<ol style="list-style-type: none"> <li>2. ¿Fue cómodo para usted la navegación con el asistente de búsqueda? a la respuesta preguntar ¿Por qué?</li> <li>3. ¿Encontró lo que buscaba?</li> </ol>
<b>Roles y responsabilidades</b>
<ol style="list-style-type: none"> <li>1. Coordinador de prueba, Responsable de dirigir a la persona objeto de la prueba.</li> <li>2. Analista de calidad, Responsable de documentar el flujo de la prueba.</li> </ol>
<b>Plan de trabajo General</b>
<ol style="list-style-type: none"> <li>1. Presentación de la aplicación (breve resumen de 10 minutos).</li> <li>2. Explicación de la prueba a realizar (5 minutos).</li> <li>3. Ejecución de la prueba (5 minutos).</li> </ol>
<b>Riesgos identificados a la hora de realizar las pruebas</b>
<ol style="list-style-type: none"> <li>1. Ausencia de servicio eléctrico.</li> <li>2. Los equipos de cómputo presentan fallas.</li> <li>3. El servicio de Internet falla.</li> </ol>

Nota: Elaboración propia (2023) 1

#### 7.4. Modelado de Vulnerabilidades

Para analizar la seguridad del software, se realizó el modelado de vulnerabilidades recomendado por DevSecOps. Esto permitió identificar posibles vulnerabilidades en el software a fin de que a futuro se corrijan algunas de ellas (las que no fueron corregidas para este alcance).

Vulnerabilidades comunes en aplicaciones web que se deben considerar:

1. **Inyección SQL en el backend:** Dado que el backend utiliza una base de datos relacional, es posible que se produzcan vulnerabilidades de inyección SQL si no se manejan correctamente las consultas a la base de datos. Se debe validar que todas las consultas de la base de datos estén correctamente parametrizadas y que no permitan la ejecución de código malicioso.

2. **XSS en el frontend:** El frontend está construido en Angular e Ionic, por lo que es importante validar que todas las entradas de usuario sean correctamente sanitizadas antes de ser mostradas en la interfaz. Si no se hace correctamente, se pueden producir vulnerabilidades de XSS que permitan la ejecución de código malicioso en el navegador del usuario.
3. **Exposición de datos sensibles:** El proyecto maneja datos personales de los usuarios, como números de contacto, correos electrónicos y coordenadas geográficas. Es importante asegurarse de que estos datos se manejen de forma segura, evitando su exposición en la red y protegiéndolos con técnicas de cifrado en caso de ser almacenados.
4. **Falta de validación de entradas de usuario:** Es importante validar todas las entradas de usuario para evitar posibles vulnerabilidades, como la inyección de código malicioso o la realización de acciones no autorizadas. Se debe validar tanto en el frontend como en el backend, y utilizar técnicas como la validación de formatos y la limitación de caracteres para asegurar que los datos introducidos son seguros.
5. **Falta de autenticación en el backend:** Dado que se manejan datos sensibles, es importante asegurarse de que solo los usuarios autorizados puedan acceder a ellos. Es necesario implementar autenticación en el backend, utilizando técnicas como tokens JWT para validar la identidad del usuario y evitar accesos no autorizados.

6. **Falta de control de acceso:** Además de autenticar al usuario, es necesario controlar qué acciones están autorizadas para cada usuario en particular. Se debe implementar un control de acceso que permita restringir el acceso a funciones específicas en función de los roles y permisos del usuario.
7. **Autenticación inadecuada:** esto ocurre cuando las credenciales de un usuario son vulnerables a ataques de fuerza bruta o ataques de diccionario. Para prevenir esto, se deben implementar medidas de seguridad adecuadas, como el uso de contraseñas seguras y la autenticación de dos factores.
8. **Divulgación de información sensible:** esto ocurre cuando la aplicación revela información sensible, como contraseñas o números de tarjetas de crédito, a través de mensajes de error u otras vías. Para prevenir esto, es importante implementar medidas de seguridad adecuadas, como la encriptación de datos y la validación de acceso a los datos.

## 8. CONFIGURACIÓN E INSTALACIÓN

Este manual tiene como objetivo proporcionar información detallada y guiar al equipo técnico a través del proceso de configuración del proyecto. Asegúrate de seguir cada uno de los pasos descritos en este manual para garantizar que la configuración sea correcta.

### Requisitos previos

Antes de comenzar la instalación, asegúrate de que tu equipo cumpla con los siguientes requisitos:

- Sistema operativo: se recomienda utilizar Windows 10, macOS o una distribución de Linux moderna.
- Acceso a internet: el software requiere acceso a internet para funcionar correctamente.
- Espacio en disco: se recomienda tener al menos 2 GB de espacio libre en disco para instalar el software.

### Configuración de Google Cloud

1. Crear una cuenta de Google Cloud y un proyecto.
2. Configurar el proyecto:
  - Accede a la Consola de Google Cloud.
  - Selecciona el proyecto que deseas usar.

- Abre la sección de Configuración en el menú de la izquierda.
  - Configura los ajustes de facturación, ajustes de identidad y acceso, ajustes de la API y otros ajustes necesarios para el proyecto.
3. Crear una cuenta de servicio de Google Cloud:
- Accede a la Consola de Google Cloud.
  - Abre la sección IAM y Administración en el menú de la izquierda.
  - Selecciona "Cuentas de servicio".
  - Haz clic en "Crear cuenta de servicio".
  - Configura la cuenta de servicio con los permisos y roles necesarios para tu proyecto.
4. Crear credenciales de autenticación:
- Accede a la Consola de Google Cloud.
  - Abre la sección IAM y Administración en el menú de la izquierda.
  - Selecciona "Cuentas de servicio".
  - Haz clic en la cuenta de servicio que acabas de crear.
  - Haz clic en "Crear clave de cuenta".
  - Selecciona el tipo de clave que deseas crear (JSON o P12).
  - Descarga la clave de cuenta y guárdala en un lugar seguro.

## Configuración de Firebase

1. Crear una cuenta de Firebase y un proyecto.

2. Configurar las opciones de autenticación:

- Accede a la Consola de Firebase.
- Abre la sección "Authentication" en el menú de la izquierda.
- Configura las opciones de autenticación, como el proveedor de autenticación de Google y el inicio de sesión anónimo.

3. Configurar las opciones de la base de datos:

- Accede a la Consola de Firebase.
- Abre la sección "Database" en el menú de la izquierda.
- Configura las opciones de la base de datos, como el tipo de base de datos que deseas usar y la región en la que se almacenarán tus datos.

4. Configurar las opciones de notificaciones:

- Accede a la Consola de Firebase.
- Abre la sección "Cloud Messaging" en el menú de la izquierda.
- Configura las opciones de notificaciones, como el tipo de notificación que deseas enviar y cómo deseas que se vean las notificaciones en los dispositivos del usuario.

5. Configurar las opciones de almacenamiento:

- Accede a la Consola de Firebase.
- Abre la sección "Storage" en el menú de la izquierda.

- Configura las opciones de almacenamiento, como el tipo de almacenamiento que deseas usar y la región en la que se almacenarán tus datos.

## Configuración del Back-End

### 1. Clonar el repositorio de GitHub:

- Clona el repositorio de GitHub en tu máquina local.
- Asegúrate de tener los permisos necesarios para acceder al repositorio.

### 2. Configurar las variables de entorno:

- Abre el archivo `.env.example` en el directorio raíz del proyecto.
- Crea un archivo `.env` y copia las variables de entorno necesarias en él.
- Reemplaza los valores de las variables de entorno con los valores correspondientes para tu proyecto.

### 3. Instalar las dependencias:

- Abre la terminal y navega hasta el directorio raíz del proyecto.
- Ejecuta el comando `"npm install"` para instalar las dependencias necesarias.

### 4. Ejecutar los microservicios en Cloud Run:

- Configura el archivo de configuración de Cloud Run para cada microservicio en el repositorio.
- Despliega los microservicios en Cloud Run usando el pipeline de GitHub Actions.

5. Configurar el servicio de Firebase Cloud Messaging:
  - Abre el archivo de configuración del servicio de Firebase Cloud Messaging.
  - Configura las opciones de notificación según sea necesario.
  
6. Probar el back-end:
  - Abre un navegador web o una herramienta de prueba de API, como Postman.
  - Realiza solicitudes a los microservicios para verificar que funcionen correctamente.

## Configuración del Front-End

1. Clonar el repositorio de GitHub:
  - Clona el repositorio de GitHub en tu máquina local.
  - Asegúrate de tener los permisos necesarios para acceder al repositorio.
  
2. Instalar las dependencias:
  - Abre la terminal y navega hasta el directorio raíz del proyecto.
  - Ejecuta el comando "npm install" para instalar las dependencias necesarias.
  
3. Configurar las variables de entorno:
  - Abre el archivo environment.ts en el directorio src del proyecto.
  - Configura las variables de entorno necesarias, como la URL de los microservicios y la clave de autenticación de Firebase.

4. Compilar la aplicación:

- Ejecuta el comando "ionic build" para compilar la aplicación.
- Si deseas compilar la aplicación para Android o iOS, ejecuta los comandos correspondientes.

5. Probar el front-end:

- Abre la aplicación en un navegador web o en un dispositivo móvil.
- Realiza pruebas para verificar que la aplicación funcione correctamente.

## REFERENCIAS

- Lewis, J., & Fowler, M. (2014). *Microservices*. Recuperado el 22 de February de 2023, de Martin Fowler: <https://martinfowler.com/articles/microservices.html>
- Roberts, M. (2018). *Serverless Architectures*. Recuperado el 22 de February de 2023, de Martin Fowler: <https://martinfowler.com/articles/serverless.html>
- Fowler, M. (7 de February de 2017). *What do you mean by "Event-Driven"?* Recuperado el 22 de February de 2023, de Martin Fowler: <https://martinfowler.com/articles/201701-event-driven.html>
- Vernon, V. (2016). *Domain-Driven Design Distilled, 1/e [Book]*. Recuperado el 22 de February de 2023, de Domain-Driven Design Distilled, 1/e [Book]: <https://www.oreilly.com/library/view/domain-driven-design-distilled/9780134434964/>
- MELL, P., & Grance, T. (2011). *NIST SP 800-145, The NIST Definition of Cloud Computing*. Recuperado el 22 de February de 2023, de NIST Technical Series Publications: <https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-145.pdf>
- Appigatla, K. (2018). *MySQL 8 Cookbook*. Recuperado el 22 de February de 2023, de Packt: <https://www.packtpub.com/product/mysql-8-cookbook/9781788395809>
- Schwartz, B., Zaitsev, P., & Tkachenko, V. (2012). *High Performance MySQL, 3rd Edition [Book]*. Recuperado el 22 de February de 2023, de High Performance MySQL, 3rd Edition [Book]: <https://www.oreilly.com/library/view/high-performance-mysql/9781449332471/>
- Gonzalez, D. (2016). *Developing Microservices with Node.js*. Recuperado el 22 de February de 2023, de Packt: <https://www.packtpub.com/product/developing-microservices-with-nodejs/9781785887406>
- Fain, Y., & Moiseev, A. (2018). *Angular Development with TypeScript*. Manning.
- Google. (2023). *Introduction to the Angular*. Recuperado el 22 de February de 2023, de Angular: <https://angular.io/docs>
- Ionic. (2023). *Introduction to Ionic*. Recuperado el 22 de February de 2023, de Ionic Framework: <https://ionicframework.com/docs>
- Ionic. (2023). *Build Your First Ionic Mobile App: Angular Development Tutorial*. Recuperado el 22 de February de 2023, de Ionic Framework: <https://ionicframework.com/docs/angular/your-first-app>
- Farley, D., & Humble, J. (2010). *Continuous Delivery: Reliable Software Releases Through Build, Test, and Deployment Automation*. Addison-Wesley Professional.
- Smart, J. (2011). *Jenkins: The Definitive Guide*. O'Reilly Media.
- Chandrasekara, C., & Herath, P. (2021). *Hands-on GitHub Actions: Implement CI/CD with GitHub Action Workflows for Your Applications*. Apress.

- País, E. (29 de 05 de 2022). *El País*. Obtenido de El País:  
<https://www.elpais.com.co/economia/venta-de-motos-en-colombia-aumento-a-toda-velocidad-en-el-primer-cuatrimestre-del-ano.html>
- Urrego, A. (7 de 03 de 2022). *La Republica*. Obtenido de La Republica:  
<https://www.larepublica.co/empresas/el-mercado-de-motos-ha-crecido-37-en-2022-y-preve-850-000-unidades-vendidas-3316182>
- Pareja, D. J. (20 de 05 de 2021). *Academia Pragma*. Obtenido de Pragma:  
<https://www.pragma.com.co/blog/transformacion-digital-por-que-es-tan-importante-para-las-empresas>
- Núñez, A. G. (1 de 9 de 2022). *La Republica*. Obtenido de La Republica:  
<https://www.larepublica.co/empresas/venta-de-motocicletas-al-cierre-de-julio-supero-en-349-011-unidades-a-las-de-carros-3437919>
- Juan, C. (20 de 10 de 2016). *Escuela de Negocios de la innovación y los Emprendedores*. Obtenido de ieb school: <https://www.iebschool.com/blog/diferencias-entre-b2b-y-b2c-marketing-estrategico/>
- Japan Management Association. (2018). *Kanban Just-in Time at Toyota*. JapanManagementAssociation.
- Bustínduy, I. (2021). *La gestión AGILE del tiempo: Método ATM para incrementar la productividad personal y de equipo en tiempos líquidos*. Editorial UOC.
- Patil, R. (2021). *Mastering Advanced Scrum: Advanced Scrum Techniques for Scrum Teams, Roles, Artifacts, Events, Metrics, Working Agreements, Advanced Engineering Practices, and Technical Agility*. BPB Publications.
- ionos - Desarrollo web. (11 de 4 de 2022). *ionos*. Obtenido de ionos:  
<https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/que-es-jira/>
- Torres, A. (27 de 10 de 2020). *excentia*. Obtenido de excentia:  
<https://www.excentia.es/que-es-confluence-y-a-que-aporta-al-trabajo-remoto>
- Google. (2023). *angular*. Obtenido de angular: <https://angular.io/docs>
- Yakov Fain, A. M. (2018). *Angular Development with TypeScript*. Manning Publications.
- Ionic. (2023). *ionicframework*. Obtenido de ionicframework: <https://ionicframework.com/>
- SHAH, D. N. (2018). *Node .js*. BPB Publications.
- Silvia Botros, J. T. (2021). *High Performance MySQL*. O'Reilly Media.
- James Lewis, M. F. (2014). *martinfowler*. Obtenido de martinfowler: <https://martinfowler.com/articles/microservices.html>
- Roberts, M. (22 de 05 de 2018). *martinfowler*. Obtenido de martinfowler:  
<https://martinfowler.com/articles/serverless.html>
- Fowler, M. (07 de 02 de 2017). *martinfowler*. Obtenido de martinfowler:  
<https://martinfowler.com/articles/201701-event-driven.html>
- Vernon, V. (2016). *Domain-Driven Design Distilled*. Addison-Wesley Professional.
- Naresh Kumar Sehgal, P. C. (2019). *Cloud Computing with Security - Concepts and Practices*. Springer International Publishing. Obtenido de <https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-145.pdf>
- Krief, M. (2022). *Learning DevOps: A Comprehensive Guide to Accelerating DevOps Culture Adoption with Terraform, Azure DevOps, Kubernetes, and Jenkins*. Packt Publishing.
- Belmont, J.-M. (2018). *Hands-On Continuous Integration and Delivery: Build and Release Quality Software at Scale with Jenkins, Travis CI, and CircleCI*.

Belmont, J. M. (2018). *Hands-On Continuous Integration and Delivery*. Packt Publishing.

Chaminda Chandrasekara, P. H. (2021). *Imagen de cubierta para Hands-on GitHub actions : implement CI/CD with GitHub action workflows for your applications*  
*Hands-on GitHub actions : implement CI/CD with GitHub action workflows for your applications* . Apress.

Albert Nogués, J. V. (2017). *Business Intelligence Tools for Small Companies : a Guide to Free and Low-Cost Solutions*. Berkeley.