



**UNIVERSIDAD DEL BÍO-BÍO**

FACULTAD DE CIENCIAS EMPRESARIALES

DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN Y TECNOLOGÍAS

DE LA INFORMACIÓN

# **Sistema de ventas de productos y prestación de servicios para Dekoi**

Rodrigo Ignacio Cifuentes Martínez

Memoria para optar al título de Ingeniero Civil en Informática.

Docente Guía : María Antonieta Soto Chico  
Docente Informante : Luis Gajardo Díaz

Marzo 2021

Chillán - Chile

## Resumen

Este proyecto se presenta para dar conformidad a los requisitos exigidos por la Universidad de Bío-Bío en el proceso de titulación para la carrera de Ingeniería civil en informática.

El proyecto titulado “Sistema de ventas de productos y prestación de servicios para Dekoi”, busca implementar un sistema de venta de productos y prestación de servicios mobiliarios. Para esto se establece desarrollar una aplicación web, en la cual el administrador de la empresa podrá gestionar tanto los productos destinados a la venta como los servicios que la empresa ofrece y gestionar las reuniones que sean agendadas. También permitirá a los clientes adquirir productos y solicitar reuniones para tratar los servicios ofrecidos por Dekoi.

En la elaboración de la aplicación web se utilizó la metodología iterativo incremental combinada con la metodología Kanban, logrando mayor cercanía con el cliente con el fin de obtener una retroalimentación constante para detectar errores de forma temprana y resolverlos a la brevedad.

Para el desarrollo del software, se seleccionó la arquitectura orientada a servicios para el back-end y MVP para el front-end. El back-end y el front-end interactúan mediante servicios REST. Se realizaron pruebas de sistema y de usabilidad para resguardar la calidad del producto.

Como resultado de este proyecto, se ha desarrollado una aplicación para la empresa Dekoi que cumple con todos los requisitos establecidos con el cliente, la que actualmente gestiona el inventario de productos y la compra de estos, un historial de compras y prestación de servicios personalizados.

## Abstract

This project is presented to accomplish the requirements demanded by the University of Bío-Bío in the process of qualification for the degree in Ingeniería civil en informática.

The project entitled "System of product sales and service provision for Dekoi", seeks to implement a system of product sales and service provision for furniture. To this end, it is established to develop a web application, in which the company's administrator will be able to manage both the products for sale and the services that the company offers, and manage the meetings that are scheduled. It will also allow customers to purchase products and request meetings to discuss the services offered by Dekoi.

In the development of the web application, the iterative incremental methodology was used, which was combined with the Kanban methodology, in order to have a greater proximity to the customer. In this way, a constant feedback is obtained, which helps to detect possible errors and, therefore, to solve them as soon as possible.

For the development of the software, the service oriented architecture was selected for the back-end and MVP for the front-end. The back-end and the front-end interact through REST services. System and usability tests were carried out to safeguard the quality of the product.

As a result of this project, an application has been developed for the company Dekoi that meets all the requirements established with the client, which currently manages the inventory of products and the purchase of these, a history of purchases and provision of personalized services.

# Índice General

<b>1</b>	<b>INTRODUCCIÓN.....</b>	<b>9</b>
<b>2</b>	<b>DEFINICIÓN DE LA EMPRESA .....</b>	<b>10</b>
2.1	DESCRIPCIÓN DE LA EMPRESA .....	10
2.2	DESCRIPCIÓN DE LA PROBLEMÁTICA .....	10
<b>3</b>	<b>DEFINICIÓN PROYECTO .....</b>	<b>11</b>
3.1	OBJETIVOS DEL PROYECTO .....	11
3.2	AMBIENTE DE INGENIERÍA DE SOFTWARE .....	11
3.2.1	METODOLOGÍA DE DESARROLLO .....	11
3.2.2	TÉCNICAS Y NOTACIONES.....	12
3.2.3	ESTÁNDARES DE DOCUMENTACIÓN, PRODUCTO O PROCESO: .....	12
3.2.4	HERRAMIENTAS DE APOYO AL DESARROLLO .....	13
<b>4</b>	<b>ESPECIFICACIÓN DE REQUERIMIENTOS DE SOFTWARE .....</b>	<b>14</b>
4.1	ALCANCES Y LIMITACIONES.....	14
4.2	OBJETIVO DEL SOFTWARE .....	14
4.3	DESCRIPCIÓN GLOBAL DEL PRODUCTO .....	14
4.3.1	INTERFAZ DE USUARIO.....	14
4.4	REQUERIMIENTOS FUNCIONALES DEL SISTEMA .....	15
4.4.1	REQUISITOS NO FUNCIONALES.....	17
4.4.2	INTERFACES EXTERNAS DE ENTRADA.....	17
4.4.3	INTERFACES EXTERNAS DE SALIDA .....	17
<b>5</b>	<b>FACTIBILIDAD.....</b>	<b>19</b>
5.1	FACTIBILIDAD TÉCNICA. ....	19
5.2	FACTIBILIDAD OPERATIVA. ....	19
5.3	FACTIBILIDAD ECONÓMICA.....	20
5.4	CONCLUSIÓN DE LA FACTIBILIDAD .....	21
<b>6</b>	<b>ANÁLISIS .....</b>	<b>22</b>
6.1	PROCESOS DE NEGOCIOS FUTUROS .....	22
6.1.1	COMPRA DE PRODUCTOS EN STOCK .....	22
6.1.2	ADQUISICIÓN DE UN SERVICIO .....	22
6.2	DIAGRAMA DE CASOS DE USO .....	25
6.2.1	ACTORES.....	25
6.2.2	CASOS DE USO Y DESCRIPCIÓN .....	25
6.2.3	ESPECIFICACIÓN DE LOS CASOS DE USO .....	26
6.3	MODELAMIENTO DE DATOS .....	28
<b>7</b>	<b>DISEÑO.....</b>	<b>30</b>
7.1	DISEÑO DE FÍSICO DE LA BASE DE DATOS .....	30
7.2	DISEÑO DE ARQUITECTURA .....	30
7.2.1	MODELO-VISTA-PRESENTADOR (MVP).....	30
7.2.2	ARQUITECTURA ORIENTADA A SERVICIOS (SOA).....	32
7.2.3	DIAGRAMA DE CLASES.....	34

<b>7.3</b>	<b>DISEÑO INTERFAZ Y NAVEGACIÓN .....</b>	<b>36</b>
7.3.1	JERARQUÍA DEL MENÚ DE LA APLICACIÓN.....	36
7.3.2	ESPECIFICACIÓN DE DISEÑO DE LA INTERFAZ .....	37
<b>8</b>	<b><u>SEGURIDAD Y PRUEBAS .....</u></b>	<b><u>42</u></b>
<b>8.1</b>	<b>SEGURIDAD .....</b>	<b>42</b>
<b>8.2</b>	<b>PRUEBAS .....</b>	<b>43</b>
8.2.1	ELEMENTOS DE PRUEBA .....	43
8.2.2	ESPECIFICACIÓN DE LAS PRUEBAS.....	44
8.2.3	DETALLE DE LAS PRUEBAS .....	45
8.2.4	PRUEBAS DE USABILIDAD .....	46
8.2.5	CONCLUSIONES DE PRUEBAS.....	47
<b>9</b>	<b><u>CONCLUSIONES .....</u></b>	<b><u>48</u></b>
9.1	TRABAJO FUTURO .....	49
<b><u>ANEXO A: DETALLE DE CASOS DE USO.....</u></b>		<b><u>52</u></b>
<b><u>ANEXO B: PRUEBAS DE UNIDAD.....</u></b>		<b><u>57</u></b>
<b><u>ANEXO C: ENCUESTA DE USABILIDAD .....</u></b>		<b><u>72</u></b>

## Índice Tablas

Tabla 1: Colores oficiales de Dekoi.....	15
Tabla 2: Requisitos funcionales - Primer Incremento .....	16
Tabla 3: Requisitos funcionales - Segundo Incremento.....	16
Tabla 4: Requisitos funcionales - Tercer Incremento .....	16
Tabla 5: Continuación, Requisitos funcionales - Tercer Incremento .....	17
Tabla 6: Interfaces externas de entrada.....	17
Tabla 7: Interfaces externas de salida.....	18
Tabla 8: Especificaciones del equipo de desarrollo .....	19
Tabla 9: Herramientas del ambiente de desarrollo del software .....	19
Tabla 10: Comparación de beneficios antes y después de la implementación del sistema.....	20
Tabla 11: Inversión inicial del proyecto.....	21
Tabla 12: Costos fijos sobre la mantención del proyecto.....	21
Tabla 13: flujo de caja del proyecto .....	21
Tabla 14: Actor Administrador y sus funciones dentro del sistema.....	25
Tabla 15: Actor Cliente y sus funciones dentro del sistema .....	25
Tabla 16: [CU] Adquirir Productos .....	26
Tabla 17: Continuación, [CU] adquirir productos.....	27
Tabla 18: [CU] Agregar producto al carrito de compra.....	27
Tabla 19: [CU] Eliminar producto del carrito de compra .....	27
Tabla 20: Continuación, [CU] Eliminar producto del carrito de compras.....	28
Tabla 21: [CU] Agregar Producto.....	28
Tabla 22: Ejemplo de petición POST de productos .....	33
Tabla 23: Ejemplo de petición DELETE de productos.....	33
Tabla 24: Continuación, Ejemplo de petición DELETE de productos.....	34
Tabla 25: Pruebas de sistema .....	44
Tabla 26: Pruebas de usabilidad .....	44
Tabla 27: Resumen de las pruebas realizadas.....	45
Tabla 28: Encuesta de Usabilidad.....	46
Tabla 29: [CU] Editar Producto.....	52
Tabla 30: [CU] Eliminar Producto .....	52
Tabla 31: [CU] Ver Productos .....	53
Tabla 32: [CU] Crear cuenta de usuario .....	53
Tabla 33: [CU] Editar cuenta de usuario .....	54
Tabla 34: [CU] Agregar Servicio .....	54
Tabla 35: [CU] Modificar Servicio.....	55
Tabla 36: [CU] Eliminar Servicio.....	55
Tabla 37: [CU] Ver Servicio .....	56
Tabla 38: [PS] Agregar Categoría.....	57
Tabla 39: [PS] Agregar Producto .....	58
Tabla 40: [PS] Editar Producto .....	59
Tabla 41: [PS] Eliminar Producto.....	60
Tabla 42: [PS] Agregar producto al carrito de compra. ....	61
Tabla 43: [PS] Listar carrito de compras .....	62
Tabla 44: [PS] Inicio de sesión.....	63
Tabla 45: [PS] Cierre de sesión .....	64
Tabla 46: [PS] Crear cuenta de usuario .....	64
Tabla 47: [PS] Modificar datos de usuario.....	65

Tabla 48: [PS] Generar compra.....	66
Tabla 49: [PS] Mostrar Productos .....	67
Tabla 50: [PS] Ver Compras.....	67
Tabla 51: [PS] Modificar Compra.....	68
Tabla 52: [PS] Agregar Servicio .....	68
Tabla 53: [PS] Listar Servicios.....	69
Tabla 54: [PS] Modificar Servicio.....	70
Tabla 55: [PS] Eliminar Servicio.....	71

## Índice Figuras

Figura 1: Logotipo de Dekoi.....	15
Figura 2: Icono del logo Dekoi .....	15
Figura 3: Modelo de proceso de negocio de venta de productos.....	23
Figura 4: Modelo de proceso de negocios para adquisición de un servicio .....	24
Figura 5: Diagrama de Casos de Uso.....	26
Figura 6: Modelo de entidad relación del proyecto .....	29
Figura 7: Diseño físico de la base de datos.....	30
Figura 8: Ejemplo parcial Modelo Vista Presentador.....	31
Figura 9: Diagrama de interacción SOA (Sommerville, 2020) .....	32
Figura 10: Diagrama de clases Back-end de Dekoi.....	35
Figura 11: Diagrama de clases Front-end y Back-end de Dekoi.....	36
Figura 12: Diagrama de navegación.....	37
Figura 13: Página de inicio .....	38
Figura 14: Continuación, Página de inicio.....	38
Figura 15: Pagina ver productos .....	39
Figura 16: Pagina del carrito .....	40
Figura 17: Página de Compra.....	41
Figura 18: Proceso de autenticación mediante tokens .....	42
Figura 19: Fórmula para calcular el resultado de las pruebas.....	46
Figura 20: Encuesta de usabilidad P1-P2 .....	72
Figura 21: Encuesta de usabilidad P3-P4.....	73
Figura 22: Encuesta de usabilidad P5-P6.....	74
Figura 23: Encuesta de usabilidad P7-P8.....	75
Figura 24: Encuesta de usabilidad P9-P10 .....	76



---

## 1 INTRODUCCIÓN

---

En un mundo cada vez más inmerso en la era digital, términos como “aplicación” y “página web” se han hecho cada vez más comunes en el conocimiento de las personas. En una época donde la gente comienza a preferir ver y comprar objetos, así también contratar servicios a través de internet, las empresas se adaptan para satisfacer las necesidades de las personas ofreciendo opciones tecnológicas para vender o promocionar sus productos. Debido a esta necesidad surge la propuesta de realizar este proyecto, que facilita la adquisición de diversos productos mediante una solución TI.

Este proyecto busca desarrollar e implementar una aplicación web para la empresa Dekoi, que tiene como fin, mostrar y vender productos mobiliarios a un público abierto, además de presentar servicios personalizados que realiza la empresa.

El documento que se presenta a continuación consta de nueve capítulos, siendo el primero de ellos esta introducción. En el capítulo dos “definición de la empresa”, se presenta una descripción completa de la empresa y se explica la problemática que esta posee. Posteriormente, en el capítulo tres se encuentra la definición del proyecto, donde se exponen los objetivos generales y específicos de este. Luego se detallan las metodologías de desarrollo Kanban e iterativa e incremental, utilizadas en el desarrollo del proyecto. En el capítulo cuatro se encuentra la especificación de requerimientos de software, donde se detallan los alcances y las limitaciones que presenta el proyecto y los requisitos funcionales desarrollados en cada incremento. A continuación, se encuentra el capítulo cinco, que detalla el estudio de factibilidad realizado desde tres perspectivas; factibilidad técnica, operativa y económica. En conjunto buscan medir si el proyecto de software es factible de realizar o no. En el apartado seis se encuentra el análisis, donde se exponen los procesos de negocios detectados, el diagrama de casos de uso y el modelamiento de datos, los cuales buscarán cómo resolver la problemática. En el capítulo siete, se encuentra todo lo relacionado al diseño del software, incluyendo el diseño físico de la base de datos, la arquitectura de desarrollo para el front-end y el back-end, el diseño de la interfaz y, por último, la navegación del sistema. Posteriormente, en el capítulo ocho, se especifican los tipos de pruebas a realizar y las pruebas realizadas. Finalmente, en el capítulo nueve se presenta las conclusiones y trabajos futuros.

---

## 2 DEFINICIÓN DE LA EMPRESA

---

### 2.1 Descripción de la empresa

Dekoi es una empresa chilena perteneciente a la región de Ñuble, localizada en la ciudad de Chillán, la cual se desempeña en el rubro de la arquitectura, diseño, construcción y prestación de servicios. Dentro de los diversos servicios que ofrece Dekoi se encuentran, asesorías decorativas, ampliaciones y remodelaciones, entre otros proyectos mobiliarios, ajustándose a las necesidades del cliente.

Antecedentes generales de la Empresa.

- Nombre: Dekoi
- Rubro: Arquitectura, diseño, construcción y prestación de servicios en el área mobiliaria.
- Productos y servicios que ofrece: Productos y servicios mobiliarios.

### 2.2 Descripción de la problemática

La empresa de decoración y desarrollo mobiliario Dekoi, en el pasado contaba con un sistema web el cual únicamente presentaba los diversos servicios que realizaban, tales como asesoría en decoración, ampliaciones y remodelaciones, proyectos personalizados para diferentes clientes, entre otros proyectos mobiliarios. El sistema anterior quedó en desuso, ya que no era cómodo para el administrador gestionar dicho sistema. En la actualidad Dekoi da a conocer y concreta sus prestaciones de servicios mediante contactos y por redes sociales. Sin embargo, la empresa desea contar con un sistema que le permita dar a conocer sus servicios e iniciar el contacto con los clientes. Por otra parte, Dekoi no cuenta con una plataforma para vender productos ya manufacturados (stock), solo vende productos a pedido, y desea incursionar en la venta de productos contra stock. A continuación, se desglosará y explicarán las posibles oportunidades para el negocio.

Primero, Prestación de servicios vía video conferencia. Como se mencionó anteriormente, la prestación de servicios en la actualidad se efectúa mediante el traspaso de contactos y las redes sociales. El cliente se contactará mediante una reunión física o telefónica con la empresa para solicitar algún servicio del área de decoración o servicios mobiliarios. Después de escuchar la necesidad del cliente, la empresa realiza un presupuesto, si el cliente se encuentra interesado en adquirir el servicio se lleva a cabo la visita a terreno, en caso contrario rechaza la propuesta de la empresa.

Segundo, Venta de productos en stock. Actualmente, la empresa no posee un sistema de ventas de productos que existan en un inventario, por lo tanto, las ventas solo se realizan a pedido. Se propone la creación de un catálogo que presente de forma clara y atractiva una lista de productos en stock, para que los clientes puedan realizar compras.

---

## 3 DEFINICIÓN PROYECTO

---

### 3.1 Objetivos del proyecto

General:

Desarrollar un sistema de venta de productos mobiliarios y prestación de diversos servicios de decoración y mobiliarios para la empresa Dekoi, utilizando tecnologías actuales.

Específicos:

- Ofrecer un catálogo de productos intuitivo para que los clientes adquieran productos.
- Presentar reportes de stock y de ventas de productos.
- Brindar una interfaz intuitiva para que el administrador gestione fácilmente el catálogo de productos y servicios.
- Ampliar el mercado potencial de la empresa.

### 3.2 Ambiente de Ingeniería de Software

#### 3.2.1 Metodología de desarrollo

Para el desarrollo de este proyecto se ha decidido trabajar bajo el paradigma de las metodologías ágiles, más concretamente bajo la metodología Kanban, ya que es necesario establecer metas asequibles, permitiendo una gestión de trabajo más fluida y manteniendo un trabajo constante en el proyecto. Kanban es una metodología ágil de desarrollo y gestión de proyectos, centrada en el uso de tableros con tareas con estados por realizar, en progreso y hechas (Mas, 2019). Además, se utilizará el desarrollo iterativo incremental, donde se realizan entregas funcionales del producto al cliente y también se realizarán reuniones semanales con el cliente.

**Beneficios de la metodología Kanban** (Mas, 2019):

1. **Mejorar la calidad del producto:** Con buena organización y con un buen control de los tiempos, se podrán detectar posibles problemas y, por consecuencia, encontrar la solución.
2. **Organización del flujo de trabajo:** El flujo de trabajo queda claro en cada fase del proyecto gracias a la organización de las tareas por estados y de la duración estimada que se indica para cada una de ellas. Por ello, es posible prever una cantidad de trabajo adecuada y asequible para el cumplimiento de las metas.
3. **Distribución de tareas:** Poder ver las tareas que se han hecho, las que se están haciendo y las que quedan por hacer ayuda a una correcta distribución: todo el equipo puede ver cómo va el transcurso del trabajo y saber cuál es el siguiente paso.
4. **Evitar la acumulación de trabajo:** Al previsualizar todas las tareas previstas y el tiempo estimado para su cumplimiento, el equipo podrá organizarse y asignar cada trabajo sin sobrecargar.
5. **Cumplir los tiempos de entrega:** Ligado al segundo punto, gracias a la organización del flujo de trabajo podremos medir y cumplir con los tiempos de entrega demandados por el cliente

### 3.2.2 Técnicas y notaciones

El uso de técnicas y notaciones en el desarrollo de proyectos de software se realiza para facilitar la comprensión del problema abordado, además de establecer protocolos relacionados al modelado de la aplicación, al producto de software y el análisis del problema.

- **Back-End:** Es un término empleado en el área de la ingeniería de software que se utiliza para referirse a la parte lógica de la aplicación que se ejecuta, principalmente del lado del servidor.
- **BDD:** Es una abreviatura utilizada en el área de la ingeniería de software que significa “Bases de datos”.
- **BPMN:** Se refiere al “Modelo y Notación de Procesos de Negocio”, es una notación gráfica estandarizada que permite el modelado de procesos de negocio, en un formato de flujo de trabajo (Object Management Group, 2005). En este proyecto se utilizará para modelar los procesos de negocios sobre la compra de un producto y la adquisición de un servicio por parte de un cliente de la empresa Dekoi.
- **Front-End:** Es un término empleado en el área de la ingeniería de software que se utiliza para referirse a la parte de la interfaz visual de la aplicación, que se ejecuta normalmente del lado del cliente.
- **MER:** Significa “Modelo Entidad Relación”, y consiste en una herramienta para el modelamiento de datos que establece las relaciones existentes entre las distintas entidades de un sistema, El diagrama resultante de esta técnica facilita el entendimiento del sistema por parte de los involucrados e interesados. En este proyecto se utilizará para modelar de manera lógica la solución a implementar. (Chen, 1976)
- **UML:** Significa “Lenguaje de Modelado Unificado” y establece un lenguaje visual de fácil comprensión para la arquitectura, el diseño y la implementación de tecnologías de información en proyectos de software. En este proyecto se utilizará para elaborar el diagrama de casos de uso. (Object Management Group, 1994)
- **Api-REST:** Se refiere a “Transferencia de estado representacional” (en inglés, representational state transfer o REST) es un conjunto de restricciones con las que se puede “crear” un estilo de arquitectura de software, la cual tiene como foco central las aplicaciones web respetando HTTP. (Fielding, 2000)

### 3.2.3 Estándares de documentación, producto o proceso:

**Estándar de gestión de requisitos ISO/IEC 25010:** En este modelo se determinan las características de calidad que se van a tener en cuenta a la hora de evaluar los requisitos del software por parte del cliente. (International Organization for Standardization, 2011)

**Estándar para la gestión de la calidad ISO/IEC 9001:** Determina un conjunto de normas relacionadas con la gestión de la calidad del producto de software con el fin de concentrarse en los procesos del producto y en la satisfacción del cliente. (International Organization for Standardization, 2015)

### 3.2.4 Herramientas de apoyo al desarrollo

Para el desarrollo del proyecto se utilizará software de libre acceso que proporcionen las funcionalidades necesarias para implementar la aplicación. A continuación, se presentan las tecnologías que serán utilizadas.

- **Angular 8:** Es un framework de desarrollo basado en TypeScript, es decir, es un marco de trabajo que ofrece herramientas para facilitar el desarrollo del Front-end, enfocado en aplicaciones web (Google, 2010). En este proyecto se utilizará para desarrollar el código del Front-End de la aplicación web.
- **Atom:** Es un IDE de código abierto el cual soporta múltiples lenguajes de desarrollo (GitHub, 2014). En este IDE se llevará a cabo el desarrollo del Front-End de la aplicación.
- **Bizagi modeler:** Es una herramienta de acceso libre para el modelamiento de procesos de negocio de manera gráfica, sigue la notación BPMN. En este proyecto se utilizará para realizar el modelo de proceso de negocio de las actividades relacionadas al préstamo de dispositivos y reserva de salas de laboratorio (Bizagi, 1989).
- **Bootstrap:** Es uno de los principales framework de desarrollo en entornos web para la creación de páginas responsivas. En este proyecto se utilizará para ofrecer una mejor estética a la aplicación, además de facilitar la responsividad, es decir, la adaptación de la interfaz a diferentes dimensiones de pantalla (Otto & Thornton, 2011).
- **CSS:** Es un lenguaje usado para definir y crear la presentación de un documento estructurado escrito en HTML o XML. Su uso es imprescindible para la creación de páginas web complejas. En este proyecto se utilizará para ofrecer una visual estética al Front-End (CSS Working Group, 1995).
- **Eclipse:** Es un IDE de desarrollo de software de código abierto enfocado a lo que es el Back-End de las aplicaciones ( Object Technology International, 2001).
- **Java:** Es un lenguaje de programación compilado e interpretado que en la actualidad se utiliza en una variada gama de aplicaciones (Oracle Corporation, 1996). En este proyecto se utilizará en el desarrollo del Back-End de la aplicación por el lado del servidor.
- **Javascript:** Es un lenguaje de programación interpretado que se utiliza principalmente para la creación de páginas web dinámicas (Netscape Communications, Mozilla Foundation, 1995). En este proyecto se utilizará para el desarrollo de la aplicación por el lado del cliente
- **MySQL:** Es un sistema de gestión de bases de datos relacionales, se utiliza para almacenar los datos que contendrá la aplicación (Oracle Corporation, 1995). En este proyecto se utilizará para la gestión de la información de los productos y servicios asociados al problema.
- **Spring:** Es un framework de desarrollo de aplicaciones y contenedor de versiones de código abierto para el lenguaje de programación Java (Pivotal Software, 2002). En el proyecto se utilizará para el desarrollo del Back-end de la aplicación.

---

## 4 ESPECIFICACIÓN DE REQUERIMIENTOS DE SOFTWARE

---

La especificación de los requerimientos del software está compuesta por cuatro puntos, en el primer punto tenemos los alcances, seguido por los objetivos del software, además de una descripción global del producto que contiene la interfaz del usuario. Por último, los requerimientos específicos desglosados en cuatro incrementos, además de los requisitos no funcionales.

### 4.1 Alcances y limitaciones.

El proyecto contempla la creación de un Sistema Web que permitirá gestionar los procesos de ventas de productos mobiliarios, ingreso de productos mobiliarios y presentación de servicios.

El sistema tiene como alcance gestionar la compra y el envío de los productos comprados, además de presentar un historial con los detalles de las compras realizadas. Otro alcance que posee el proyecto es administrar las reuniones para tratar las adquisiciones de servicios que ofrece la empresa.

El sistema tiene como limitante, la implementación de web pay, debido a que su adquisición e implementación requiere mayor cantidad de tiempo, los cuales sobrepasan los tiempos estimados del proyecto. Como segunda limitante se tiene que el sistema presente no registrará cotizaciones, ni contrataciones de los servicios, solo gestionará reuniones de estos.

### 4.2 Objetivo del software

El software tiene como objetivo presentar de manera ordenada un catálogo de productos que ofrece la empresa y también gestionar las ventas de dichos productos. Además, el software administrará las prestaciones de servicios que ofrece la empresa, ofreciéndolos al igual que los productos en forma de catálogo a los clientes, ellos podrán agendar reuniones vía online o presencial con la administración para conversar sobre dicho servicio y si es factible realizarlo.

### 4.3 Descripción Global del Producto

En el siguiente apartado se presentan los elementos necesarios para la elaboración de la interfaz de usuario, los requisitos funcionales y los requerimientos del sistema.

#### 4.3.1 Interfaz de usuario

Debido a que Dekoi rediseñó el logotipo de su marca, la gama de colores que esta presentaba en el sistema anterior fue descartada, por ende, se estableció una nueva gama de colores con relación a la nueva marca. En términos generales, para el desarrollo de la aplicación se utilizaron los colores naranjas, gris con una tonalidad un poco más oscura y el color negro. En la Tabla 1 se muestra el detalle de cada color con su código RGB, HSV y el código web.




Uso	Color 1	Color 2	Color 3
			
Web	#FF5100	#878787	#1D1D1B
RGB	R255-G81-B0	R135-G135-B135	R29-G29-B29
HSV	H20-S100-V100	H60-S0-V53	H60-S7-V11

Tabla 1: Colores oficiales de Dekoi

Como fue mencionado anteriormente, la gama de colores fue elaborada en base al nuevo logotipo de la empresa, este debe mostrarse en la barra superior de la aplicación y el icono de la empresa que se mostrará en las pantallas de inicio de sesión y el pago.



Figura 1: Logotipo de Dekoi



Figura 2: Icono del logo Dekoi

#### 4.4 Requerimientos Funcionales del sistema

Los requerimientos funcionales del sistema fueron separados en tres diferentes incrementos, el primero se expone en la Tabla 2, el segundo en la Tabla 3 y en el tercero en la Tabla 4 y Tabla

5. Cada tabla se encuentra compuesta de un ID que representa un código identificador para cada requisito, seguido del nombre del requisito y una descripción.

<b>PRIMER INCREMENTO</b>		
<b>ID</b>	<b>NOMBRE</b>	<b>DESCRIPCIÓN</b>
RF01	Ingresar producto	Permite registrar un nuevo producto al sistema.
RF02	Modificar producto	Permite actualizar los datos de un producto existente.
RF03	Eliminar Producto	Permite eliminar un producto específico del sistema.
RF04	Listar productos	Permite listar los productos existentes en el sistema.
RF05	Listar productos por categoría	Permite listar los productos por categoría existentes en el sistema.
RF06	Ingresar categoría	Permite ingresar los datos de una categoría al sistema.
RF07	Listar categorías	Permite listar las categorías existentes en el sistema.
RF08	Eliminar categoría	Permite borrar la categoría seleccionada con los productos que contenga.
RF09	Modificar categoría	Permite modificar una categoría existente del sistema.
RF10	Ingresar imágenes a un producto	Permite ingresar varias imágenes a un producto existente.
RF11	Mostrar imágenes de un producto	Permite mostrar las imágenes que posee un producto existente en el sistema.
RF12	Eliminar imágenes de un producto	Permite borrar una imagen seleccionada de un producto.

Tabla 2: Requisitos funcionales - Primer Incremento

<b>SEGUNDO INCREMENTO</b>		
<b>ID</b>	<b>NOMBRE</b>	<b>DESCRIPCIÓN</b>
RF13	Ingresar producto al carrito de compra	Permite agregar un nuevo producto al carrito de compra.
RF14	Modificar producto del carrito de compra	Permite modificar los productos de un carrito existente.
RF15	Listar productos del carrito de compra	Permite listar los productos existentes en el carrito de compra.
RF16	Eliminar Producto del carrito de compra	Permite eliminar productos del carrito de compra.
RF17	Generar Compra	Permite generar la compra de los productos existentes en el carrito añadiendo la dirección de envío.
RF18	Mostrar la compra realizada	Permite mostrar el detalle de la compra realizada, con los productos previamente seleccionados y el total de la compra.
RF19	Mostrar Historial de compra	Mostrará un listado con todas las compras realizadas.
RF20	Editar Compra del cliente	Permite modificar el estado de la compra.
RF21	Registro de Usuarios	Permite a un usuario registrarse en el sistema.
RF22	Modificar el Usuario	Permite modificar la propia cuenta de usuario.
RF23	Ingreso de Usuario	Permite a un usuario ingresar en el sistema.

Tabla 3: Requisitos funcionales - Segundo Incremento

<b>TERCER INCREMENTO</b>		
<b>ID</b>	<b>NOMBRE</b>	<b>DESCRIPCIÓN</b>
RF24	Ingresar Servicio	Permite agregar un nuevo servicio al sistema.
RF25	Modificar Servicio	Permite modificar un servicio existente en el sistema.
RF26	Eliminar Servicio	Permite eliminar un servicio del sistema.

Tabla 4: Requisitos funcionales - Tercer Incremento



TERCER INCREMENTO		
ID	NOMBRE	DESCRIPCIÓN
RF27	Listar Servicio	Permite listar los servicios que existen en el sistema.
RF28	Solicitar Reunión	Permite al cliente solicitar una reunión para tratar sobre un servicio.
RF29	Agendar Reunión	Permite al administrador agendar una reunión solicitada.

Tabla 5: Continuación, Requisitos funcionales - Tercer Incremento

#### 4.4.1 Requisitos no funcionales

A continuación, se presentan los requisitos no funcionales detectados:

1. El sistema debe ser de fácil interacción tanto para el cliente como para el administrador. Ellos deben realizar las funcionalidades sin ayuda de terceros y realizarlas en un plazo de tiempo no superior a 2 horas.
2. El sistema tendrá dos niveles de seguridad, el primero por separación de roles en el Front-End con un token único y el segundo en el Back-End, lo cual permitirá que los usuarios accedan solo a las funcionalidades permitidas.

#### 4.4.2 Interfaces externas de entrada

En la Tabla 6 se presenta las interfaces de externas de entrada, estas son ingresadas vía formularios del sistema, esta tabla está compuesta por un ID que representa un identificador, el nombre de cada entrada externa y un detalle de esta.

INTERFACES EXTERNAS DE ENTRADA		
ID	NOMBRE	DETALLE
DE01	Datos Categoría	Nombre de la categoría.
DE02	Datos de Producto	Nombre, Cantidad, Valor Unitario, Descripción del producto, Categoría a la que pertenece.
DE03	Imagen de producto	Nombre de la Imagen, Producto al que pertenece.
DE04	Registro de Usuario	Nombre, Apellido, Dirección, Fecha de nacimiento, Rut, Email, Contraseña, Ciudad.
DE05	Datos de usuario	Email, Nombre, Apellido, Número de teléfono.
DE06	Dirección de usuario	Ciudad, Dirección.
DE07	Datos de inicio de sesión	Email, Contraseña.
DE08	Dirección de envío	Ciudad, Calle, Numero de casa, Departamento.
DE09	Estado de la compra de cliente	Estado.
DE10	Datos servicio	Nombre, Descripción
DE11	Datos reunión	Nombre del servicio, Estado, Fecha Inicio, Fecha Fin, Link reunión.

Tabla 6: Interfaces externas de entrada

#### 4.4.3 Interfaces externas de Salida

El software permite generar múltiples salidas del sistema como la presentación de productos, esta se puede visualizar en el formato que se especifica en la Tabla 7 .

<b>INTERFACES EXTERNAS DE SALIDA</b>		
<b>ID</b>	<b>NOMBRE</b>	<b>MEDIO SALIDA</b>
DS01	Vista previa de los productos	Vista HTML
DS02	Presentación del carrito	Vista HTML
DS03	Detalle de la compra	Vista HTML
DS04	Datos de pago	Vista HTML
DS05	Historial de compras de un cliente	Vista HTML
DS06	Datos de la cuenta de usuario	Vista HTML
DS07	Detalle de un producto	Vista HTML
DS08	Presentación de las categorías	Vista HTML
DS09	Presentación de los servicios	Vista HTML
DS10	Presentación del calendario de reuniones	Vista HTML

*Tabla 7: Interfaces externas de salida*

## 5 FACTIBILIDAD

### 5.1 Factibilidad técnica.

En el siguiente apartado se detallan los elementos de hardware y software, tanto para el desarrollo como para la operación del sistema.

El hardware que será utilizado para el desarrollo del proyecto considerará solo la computadora donde trabajará el equipo de desarrollo. Debido a que la empresa no posee servidores físicos, será necesaria la adquisición de un Hosting web para la operación del sistema. El detalle del equipo de trabajo se presenta en la Tabla 8.

<b>Especificaciones del equipo de desarrollo</b>	
Procesador	Intel(R) Core™ i7-7700HQ CPU 2.8GHz
Memoria RAM	16 GB
Almacenamiento	1 TB
Sistema Operativo	Windows 10

*Tabla 8: Especificaciones del equipo de desarrollo*

Los equipos que serán necesarios para acceder a la aplicación no serán especificados, ya que, cualquier equipo con conexión a internet y con un navegador web puede acceder a la aplicación.

El software que será utilizado para el desarrollo del sistema web se desglosa en la Tabla 9.

<b>Especificaciones software de desarrollo</b>	
Sistema Operativo	Windows 10
Entorno de desarrollo	Atom, Eclipse
Framework	Angular 8, Spring 4
Navegador web	Google Chrome
Base de datos	MySql
Motor de base de datos	PhpMyAdmin
Control de versiones	GitHub

*Tabla 9: Herramientas del ambiente de desarrollo del software*

Dada la información que se presentó anteriormente sobre los aspectos técnicos que serán utilizados para el desarrollo del sistema, se llegó a la conclusión de que es factible técnicamente, debido a que el equipo de desarrollo ya posee las herramientas necesarias para llevar a cabo el proyecto y la empresa solo debe adquirir el host del servidor.

### 5.2 Factibilidad operativa.

El sistema que se implementará tendrá un impacto positivo en la empresa, ya que, al tener una tienda online, esta otorgará ingresos de manera activa para la empresa. Además, le otorgará la facilidad al administrador para gestionar el sistema de manera fácil y rápida.

En lo que respecta a la capacitación sobre el uso del sistema, el administrador recibirá una breve capacitación para familiarizarse con el sistema.

Se estima que dicha capacitación deberá tener una duración aproximada de una hora, tiempo en el cual se le presentará y explicará al administrador el funcionamiento principal del sistema de stock de productos, el flujo de compra de productos y adquisición de servicios.

### 5.3 Factibilidad económica.

El análisis de factibilidad económica se enfoca en cuantificar los beneficios generados producto de la implementación del software, versus los costos asociados y la inversión necesaria, como resultado se comprobará si realizar el proyecto es factible económicamente.

#### Beneficios tangibles

- **Incremento en ventas:** Este factor se logró medir, mediante una entrevista con la dueña de la empresa, donde se consultó sobre las ventas anuales de la empresa. Posteriormente, se estimó que con la implementación del nuevo sistema se tendría un alza de un 15% en ventas. En la Tabla 10 se presentan los beneficios anuales de la empresa.

Beneficios	
Beneficios anuales de la empresa	<b>\$10.000.000</b>
Beneficios anuales estimados con el sistema	<b>\$11.500.000</b>

*Tabla 10: Comparación de beneficios antes y después de la implementación del sistema*

#### Beneficios Intangibles

- Tener un registro sobre los artículos que se venden con mayor frecuencia, generando gráficas de ventas para que el administrador decida a qué artículos debe darle más énfasis en su manufacturación.
- Captar la atención de más clientes utilizando publicidad patrocinada (Google Ads, Facebook Ads, Etc.), de este modo se espera aumentar las visitas al sitio web.

#### Inversión Inicial

Considerando que Dekoi no posee servidores físicos, se optó por el arriendo de un hosting para el levantamiento del software, por lo cual será añadido en el apartado de costos. Además, se consideró el sueldo mensual de un desarrollador Full Stack<sup>1</sup> por tres meses de trabajo de desarrollo y la capacitación para el administrador de la página web. Se descartó el valor de licencias debido a que se utilizará software de libre acceso para el desarrollo de este proyecto.

<sup>1</sup>

<https://neuvoo.cl/remuneracion/?job=Desarrollador%20Full%20Stack#:~:text=El%20salario%20promedio%20de%20Desarrollador%20Full%20Stack%20en%20Chile%20es,llegar%20a%20ganar%20hasta%20%2426.520.>

<b>Inversión del proyecto</b>				
	<b>Cantidad</b>	<b>Valor</b>	<b>Meses</b>	<b>Total</b>
Desarrollador	1	\$1.300.000	4	\$5.200.000
Capacitación	1	\$20.000	1	\$20.000

Tabla 11: Inversión inicial del proyecto

### Costos

Los costos presentados a continuación se encuentran relacionados con el comienzo del proyecto los cuales constan de, entre otros, un servicio de anfitrión (host) y de dominio, servicios cuyo coste especulativo es de \$34.900, anuales. En adición, se incluyen diversos costos asociados a artículos de oficina, dichas herramientas se encuentran destinadas a actividades propias de la tienda y su correcto funcionamiento, es decir, emisión de boletas y/o facturas al momento de realizarse compras, entre otras. Cabe mencionar que, en la actualidad, las boletas son generadas en el Servicio de impuestos Internos (SII).

<b>Costos</b>				
	<b>Cantidad</b>	<b>Valor</b>	<b>Mensual</b>	<b>Anual</b>
Host y Dominio	1	\$34.900	-	\$34.900
Plan de telefonía	1	\$9.990	\$9.990	\$119.880
Útiles de oficina	1	\$25.000	\$25.000	\$300.000
Total				\$ 454.780

Tabla 12: Costos fijos sobre la mantención del proyecto

### Cálculo del VAN

El VAN es un indicador financiero para determinar si un proyecto es económicamente viable o no. En el siguiente apartado se presenta el flujo de caja del proyecto, tomando en consideración los beneficios, la inversión inicial y los costos calculados anteriormente.

<b>Flujo de caja</b>						
Año						
	0	1	2	3	4	5
Beneficios		\$11.500.000	\$11.500.000	\$11.500.000	\$11.500.000	\$11.500.000
Costos		\$ -454.780	\$ -454.780	\$ -454.780	\$ -454.780	\$ -454.780
Inversión	\$-5.220.000					
Beneficios-Costos	\$-5.220.000	\$11.045.220	\$11.045.220	\$11.045.220	\$11.045.220	\$11.045.220
VAN	\$36.650.074					

Tabla 13: flujo de caja del proyecto

Para el cálculo del valor actual neto (VAN), se utilizó una tasa de interés del 10%, con un horizonte de cinco años, obteniendo como resultado un VAN positivo de \$36.650.074, por lo cual se puede concluir que el proyecto es rentable y, por ende, factible económicamente.

## 5.4 Conclusión de la factibilidad

Dado el análisis de factibilidad operativa, técnica y económica, se logró comprobar que es factible desarrollar el Sistema de ventas de productos y prestación de servicios para Dekoi.

---

## **6 ANÁLISIS**

---

### **6.1 Procesos de Negocios futuros**

En este proyecto se identificaron dos procesos de negocios para la empresa, uno de ellos presenta el antiguo núcleo central del sistema de Dekoi, este constaba en la prestación de servicios personalizados para un cliente. Por otro lado, el segundo modelo que se desea integrar al nuevo sistema consiste en la venta de productos de stock, este proceso de negocios puede presentar una oportunidad de negocios para Dekoi.

A continuación, se presentan los procesos anteriormente mencionados.

#### **6.1.1 Compra de productos en stock**

En este proceso se detalla cómo se realiza la compra de uno o varios productos en stock, en el cual interactúan tanto los clientes como el administrador. Como primer objetivo el cliente busca o visualiza algún producto de interés, si este desea comprar el producto, selecciona la cantidad del producto deseado y se verifica que dicha cantidad pueda ser provista. Una vez verificado se realiza la compra y los productos son enviados al cliente.

En la Figura 3 se puede observar el detalle del proceso de negocio.

#### **6.1.2 Adquisición de un servicio**

En este proceso se detalla cómo un cliente puede adquirir un servicio mobiliario. Como primer paso se tiene al cliente que consulta por los servicios mobiliarios que la empresa ofrece, posteriormente la empresa ofrece un horario de disponibilidad, donde se puede solicitar una reunión presencial en caso de que se encuentre dentro de la región de Ñuble o vía video conferencia en caso contrario. Una vez realizada la reunión, se le entregará al cliente un presupuesto estimado del proyecto. Luego de realizar la cotización y con la aprobación del cliente, previo pago del servicio, se coordinarán los ajustes pertinentes para que la empresa desarrolle las prestaciones requeridas. El sistema no cubrirá los pagos de los servicios, puesto que cada uno de ellos tendrán presupuestos diferentes según la complejidad del servicio. El desglose del proceso de negocio se ve en la Figura 4.

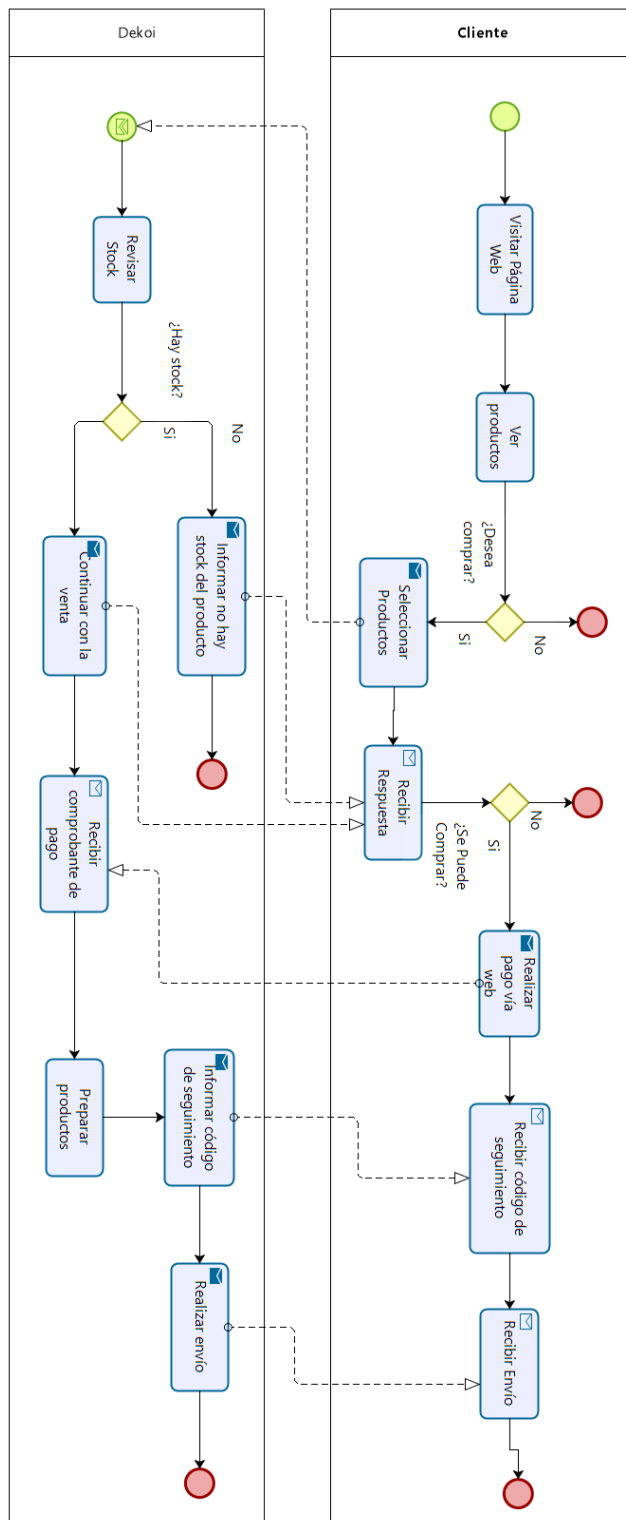


Figura 3: Modelo de proceso de negocio de venta de productos

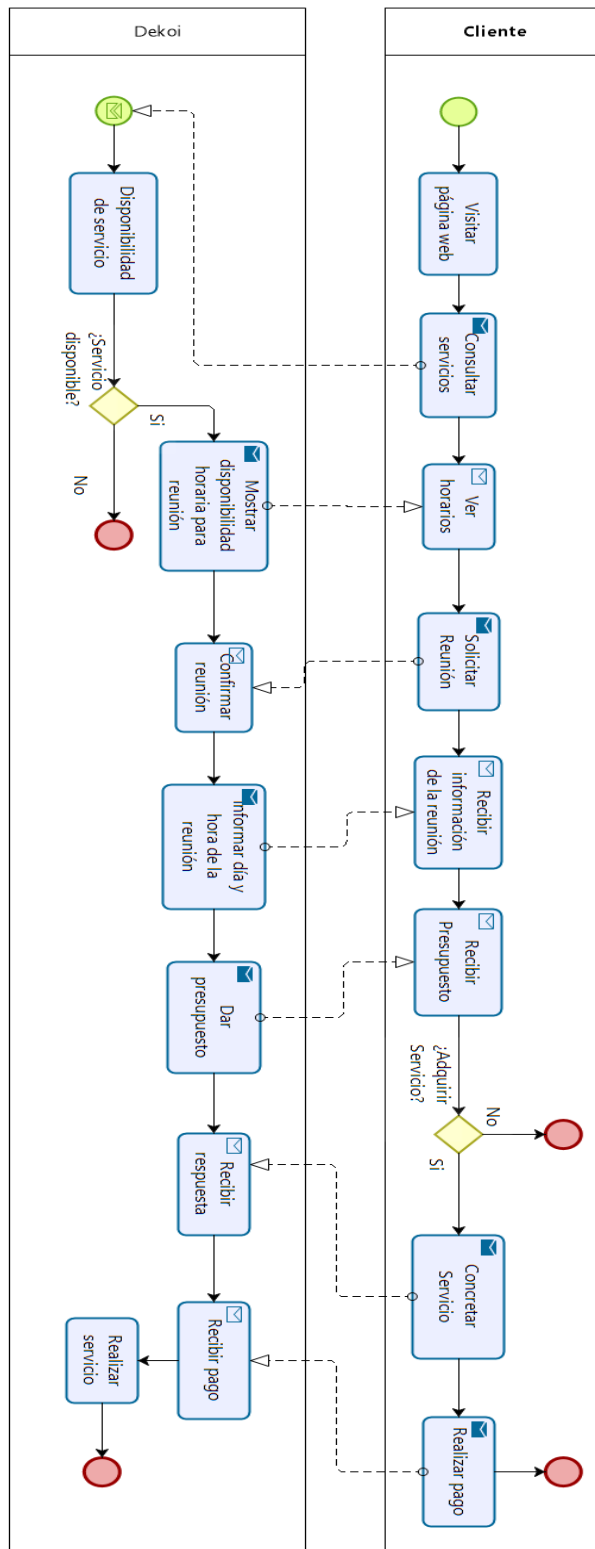


Figura 4: Modelo de proceso de negocios para adquisición de un servicio



## 6.2 Diagrama de casos de uso

El diagrama de casos de uso sirve para especificar la comunicación y el comportamiento de un sistema mediante su interacción con los usuarios y/u otros sistemas (EcuRed, s.f.). Estos usuarios detectados reciben el nombre de actores y los elementos o acciones del sistema se llaman casos de uso (Jacobson, 1986).

### 6.2.1 Actores

Los actores identificados en el problema son:

<b>Administrador</b>	
Rol dentro de la empresa	Encargado de la administración interna
Nivel de conocimientos técnicos requeridos	Nivel de conocimientos intermedio
Nivel de privilegio del sistema	Alto, posee todos los privilegios
Funcionalidades del software a las cuales tiene acceso	Acceso total de las funcionalidades del sistema

*Tabla 14: Actor Administrador y sus funciones dentro del sistema*

<b>Cliente</b>	
Rol dentro de la empresa	Solo pose el rol de cliente, puede adquirir productos y solicitar reuniones por servicios.
Nivel de conocimientos técnicos requeridos	Nivel de conocimientos intermedio
Nivel de privilegio del sistema	Bajo
Funcionalidades del software a las cuales tiene acceso	Realizar reserva de servicios, visualizar servicios y productos, adquirir productos, administrar su cuenta, revisar compras y seguimiento de envíos.

*Tabla 15: Actor Cliente y sus funciones dentro del sistema*

### 6.2.2 Casos de Uso y descripción

A continuación, en la Figura 5 se presenta el diagrama de casos de uso del sistema de ventas y prestación de servicios Dekoi.

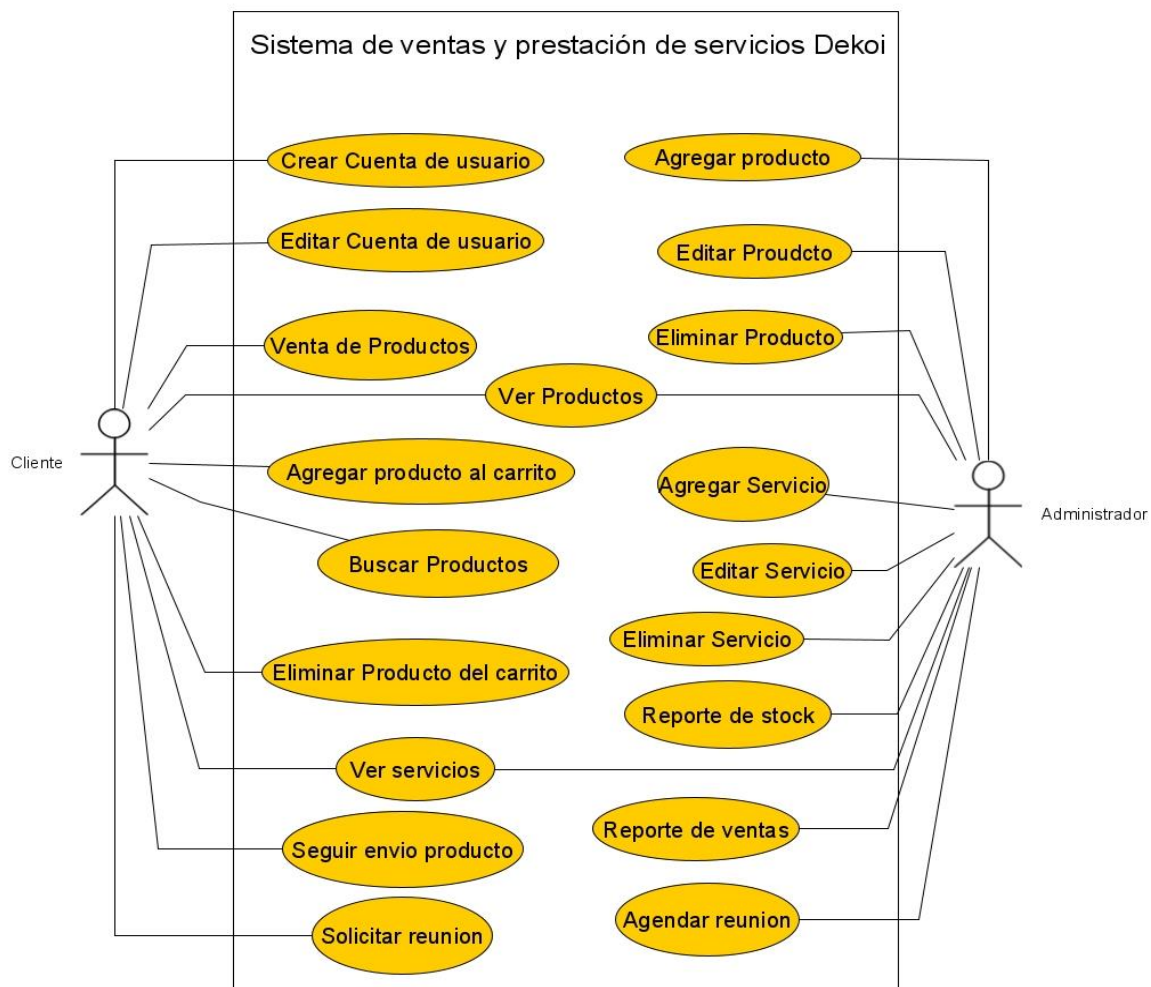


Figura 5: Diagrama de Casos de Uso

### 6.2.3 Especificación de los Casos de Uso

En el siguiente apartado se describe dos de los principales casos de uso del sistema, en la Tabla 16, Tabla 18, Tabla 20, Tabla 21 se desglosa el sistema de venta de un producto mobiliario y de gestión de productos. Entre paréntesis se indican las entradas y/o salidas externas empleadas (ver 4.4)

<b>Nombre Caso de Uso</b>	Venta de Productos		
<b>Identificador</b>	CU05	<b>Requisito Funcional</b>	RF17- RF18
<b>Descripción</b>	Permite a un cliente comprar productos mediante un carrito de compras.		
<b>Actores involucrados</b>	Cliente.		
<b>Pre-Condiciones</b>	Haber iniciado sesión en el sistema. Tener productos en el carrito de compras.		
<b>Flujo Básico</b>	<ol style="list-style-type: none"> <li>1. El cliente accede al carrito de compra.</li> <li>2. El sistema muestra el carrito de compra (DS02).</li> </ol>		

Tabla 16: [CU] Adquirir Productos

<b>Nombre Caso de Uso</b>	Venta de Productos
<b>Flujo Básico</b>	<ol style="list-style-type: none"> <li>3. El sistema despliega un formulario para añadir la dirección de envío de la compra (XX99).</li> <li>4. El cliente introduce los datos de envío (DE08).</li> <li>5. Si los campos del formulario están llenados correctamente.                         <ol style="list-style-type: none"> <li>5.1. El sistema permite generar la compra con los datos de envío.</li> </ol> </li> <li>6. El cliente genera la compra.</li> <li>7. El sistema lo redirige a la pantalla de pago (DS04).</li> <li>8. El sistema registra la compra.</li> </ol>
<b>Flujo Alternativo</b>	<ol style="list-style-type: none"> <li>5a. Si los datos de envío son vacíos                         <ol style="list-style-type: none"> <li>5a.1 El sistema deshabilita el botón para generar la compra</li> <li>5a.2 Debe volver al punto 5.</li> </ol> </li> </ol>
<b>Post-Condición</b>	<p>El stock de los productos comprados disminuyó. El carrito de compras se actualizó</p>

Tabla 17: Continuación, [CU] adquirir productos

<b>Nombre de caso de uso</b>	Agregar productos al carrito de compra		
<b>Identificador</b>	CU06	<b>Requisito Funcional</b>	RF13
<b>Descripción</b>	Permite al cliente agregar productos al carrito de compra.		
<b>Actores involucrados</b>	Cliente		
<b>Pre-Condiciones</b>	<p>Haber iniciado sesión. Debe existir al menos un producto.</p>		
<b>Flujo Básico</b>	<ol style="list-style-type: none"> <li>1. El cliente debe ingresar al listado de productos (DS01).</li> <li>2. El cliente selecciona el producto que desea agregar al carrito de compras.</li> <li>3. El cliente agrega el producto seleccionado al carrito de compras.</li> <li>4. Si el cliente desea agregar más de un producto del mismo tipo al carrito.                         <ol style="list-style-type: none"> <li>4.1. Debe entrar al detalle del producto, seleccionar la cantidad del producto y agregar al carrito.</li> <li>4.2. De otro modo, solo debe agregar el producto al carrito.</li> </ol> </li> <li>5. El sistema informa que el producto se ha agregado con éxito.</li> <li>6. El cliente solicita ver el carrito actual.</li> <li>7. Finalmente, el sistema muestra el carrito con todos los productos (DS02).</li> </ol>		
<b>Flujo Alternativo</b>	<ol style="list-style-type: none"> <li>4a. Si el producto no tiene stock.                         <ol style="list-style-type: none"> <li>4a.1. El sistema no permite agregar el producto al carrito.</li> <li>4a.2. Vuelve al punto 1.</li> </ol> </li> </ol>		
<b>Post-Condiciones</b>	El carrito de compras fue actualizado.		

Tabla 18: [CU] Agregar producto al carrito de compra

<b>Nombre de caso de uso</b>	Eliminar producto del carrito de compra		
<b>Identificador</b>	CU07	<b>Requisito Funcional</b>	RF16
<b>Descripción</b>	Permite eliminar un producto del carrito de compra		
<b>Actores involucrados</b>	Cliente		
<b>Pre-Condiciones</b>	<p>Haber iniciado sesión. Debe existir al menos un producto en el carrito de compras.</p>		

Tabla 19: [CU] Eliminar producto del carrito de compra

<b>Nombre de caso de uso</b>	Eliminar producto del carrito de compra
<b>Flujo Básico</b>	<ol style="list-style-type: none"> <li>1. El cliente debe ingresar al listado del carrito.</li> <li>2. El sistema despliega el carrito con todos los productos que contiene (DS02).</li> <li>3. El cliente presiona el botón para eliminar el producto del carrito.</li> <li>4. El sistema despliega un mensaje para confirmar la eliminación del producto. <ol style="list-style-type: none"> <li>4.1. Si confirma la eliminación, el producto es retirado del carrito.</li> </ol> </li> <li>5. El sistema muestra el carrito de compras actualizado (DS02).</li> </ol>
<b>Flujo Alternativo</b>	<ol style="list-style-type: none"> <li>5a. Si no confirma la eliminación del producto, el producto no es retirado <ol style="list-style-type: none"> <li>5a.1. El sistema muestra el carrito con los productos.</li> <li>5a.2. Vuelve al punto 3.</li> </ol> </li> </ol>
<b>Post-Condiciones</b>	El carrito de compra se actualizó

Tabla 20: Continuación, [CU] Eliminar producto del carrito de compras.

<b>Nombre de caso de uso</b>	Agregar Producto		
<b>Identificador</b>	CU01	<b>Requisito funcional</b>	RF1
<b>Descripción</b>	Permite al administrador agregar productos al sistema.		
<b>Actores involucrados</b>	Administrador		
<b>Pre-Condiciones</b>	Haber iniciado sesión. Debe existir al menos una categoría.		
<b>Flujo Básico</b>	<ol style="list-style-type: none"> <li>1. El administrador solicita agregar un producto al sistema.</li> <li>2. El sistema despliega un formulario con los campos necesarios para crear el producto (DE02).</li> <li>3. El administrador agrega la información del producto.</li> <li>4. Si los datos del producto son correctos. <ol style="list-style-type: none"> <li>4.1. El sistema permite agregar el nuevo producto en el sistema.</li> </ol> </li> <li>5. El sistema redirige al listado de productos (DS01).</li> </ol>		
<b>Flujo Alternativo</b>	<ol style="list-style-type: none"> <li>4a. Si los datos del producto no son correctos <ol style="list-style-type: none"> <li>4a.1. El sistema deshabilita el envío del formulario.</li> <li>4a.2. Vuelve al punto 3.</li> </ol> </li> <li>5a. Si desea agregar una imagen al producto. <ol style="list-style-type: none"> <li>5a.1. El administrador debe acceder al detalle del producto (DS01).</li> <li>5a.2. El sistema desplegará un campo para subir imágenes (DE03).</li> <li>5a.3. El administrador selecciona la imagen que desea subir y la envía.</li> </ol> </li> <li>6a. El sistema muestra un mensaje informando que la imagen se subió exitosamente.</li> </ol>		
<b>Post-Condiciones</b>	El sistema tiene un nuevo producto para ser presentado en el listado de productos.		

Tabla 21: [CU] Agregar Producto

Para analizar en profundidad el resto de los casos de uso del sistema leer el apartado 0 Anexo A: Detalle de casos de uso

### 6.3 Modelamiento de datos

Para la elaboración del modelo de datos, se utilizó la herramienta de modelado Yed Graph editor. En primera instancia se encuentra la entidad “producto” que tiene un nombre, stock (cantidad del producto), descripción, el valor unitario del producto y el id de la categoría a la que pertenece. Posteriormente se tiene la entidad “imagen” que tiene el nombre de la imagen y el id del producto, esto debido a que un producto puede tener más de una imagen.

En la entidad “carrito” se destaca el atributo estado, permite informar si el carrito se encuentra habilitado para agregar productos. Luego se tiene la entidad “carrito producto” que cumple la función de enlace entre producto y carrito, además, contiene el atributo cantidad, este permite saber la cantidad de productos que contiene el carrito.

La entidad “compra” tiene un atributo estado que señala si la compra ya fue pagada o se encuentra sin pagar, el id del cliente permite saber a qué cliente pertenece dicha compra, el id de la dirección, el cual permite identificar los datos de envío a los que pertenece y, por, último el id del carrito, este id sirve para saber qué productos contiene la compra realizada y para evitar consultas extensas a la base de datos.

Por último, se puede destacar la entidad “reunión”, donde se puede enfatizar la fecha inicio y fecha fin, donde también se incluye la hora de la reunión, el código de la reunión que representará el código para la videoconferencia a realizar en Google Meet, el estado de la reunión que representará si la reunión fue realizada o por realizar.

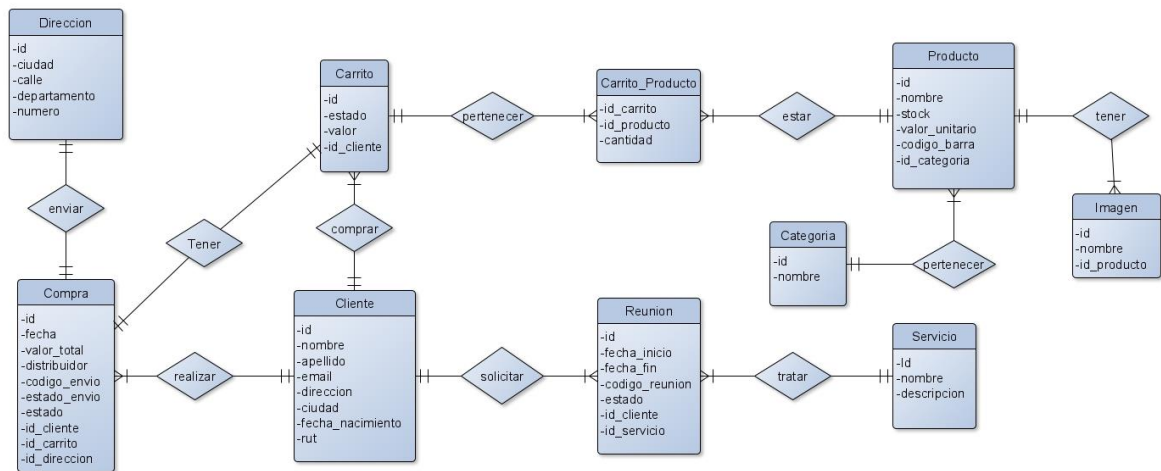


Figura 6: Modelo de entidad relación del proyecto

## 7 DISEÑO

El apartado de diseño del proyecto consta de tres ítems, donde se expone, en primera instancia, el diseño físico de la base de datos. Posteriormente, se presenta el diseño de la arquitectura funcional del sistema, donde se detallan la interacción entre los módulos desarrollados. Finalmente, se presenta el diseño de la interfaz y navegación de la aplicación.

### 7.1 Diseño de Físico de la Base de datos

El diseño físico de la base de datos busca optimizar el rendimiento y también asegura la integridad de los datos, evitando redundancias innecesarias (IBM, s.f.). Para la elaboración del diseño físico de la base de datos fue considerado el modelo de entidad relación de la Figura 6 mostrando la cardinalidad y las referencias entre las tablas.

A continuación, se presenta en la Figura 7 el modelo físico para el sistema de venta de productos y prestación de servicios para Dekoi.

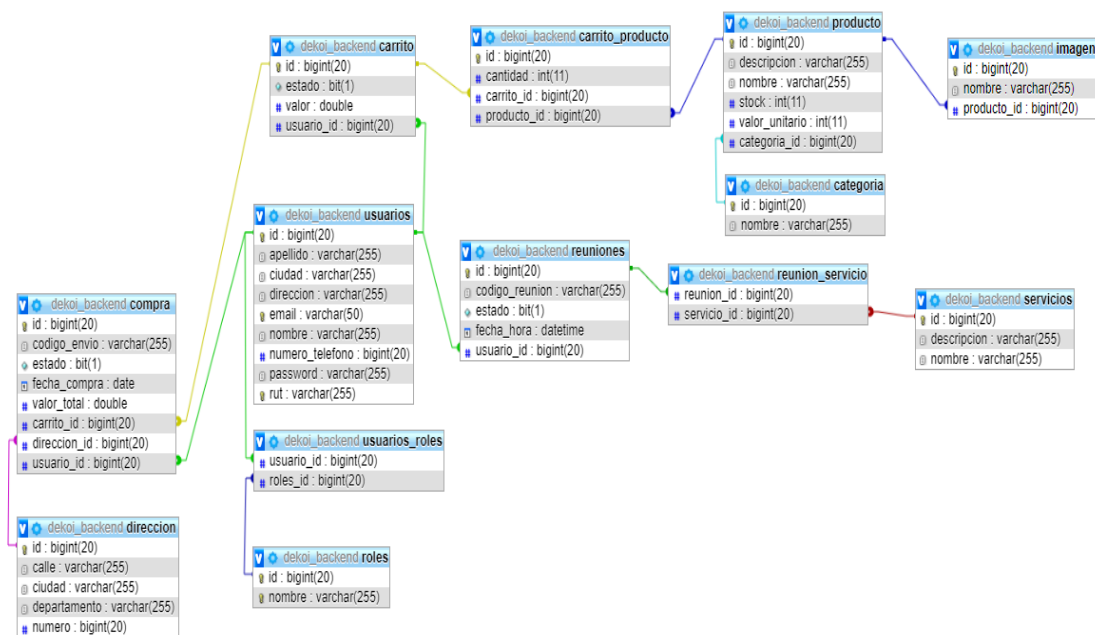


Figura 7: Diseño físico de la base de datos

### 7.2 Diseño de arquitectura

#### 7.2.1 Modelo-Vista-Presentador (MVP)

MVP es un patrón de diseño arquitectural para implementar interfaces de usuario de una aplicación. Se utiliza con el fin de minimizar la lógica compleja que pueda existir entre las clases, funciones y módulos los cuales presentan mayor dificultad al momento de testear. De este modo, evitaremos la complejidad en los componentes específicos de la interfaz de usuario (PuntoTech, s.f.).

El Modelo Vista Presentador consta de tres partes (PuntoTech, s.f.):

1. Modelo: Es una interfaz que define los datos a presentar en la vista del usuario.
2. Presentador: Funciona como un intermediario entre la vista y el modelo, recopilando los datos del modelo, para posteriormente enviarlos a la vista.
3. Vista: Exhibe los datos enviados desde el modelo y los eventos o funciones delegados por el presentador.

Para el desarrollo del front-end de la aplicación fue empleado el patrón MVP utilizando la herramienta Angular 8, donde los componentes de la aplicación conforman el presentador y la vista, y los servicios representan el modelo.

Para explicar el proceso que se realiza empleando MVP se utiliza como ejemplo el CU01 Agregar Producto.

A continuación, en la Figura 8 se presenta un modelo de clases parcial, correspondiente al front-end. Las implementaciones realizadas en cada capa se exponen a continuación:

- En la capa Vista, se presentan la interfaz de usuario con la función de agregar producto, la cual enviará el objeto producto al presentador.
- Posteriormente, el presentador recibe el objeto enviado por la vista y lo prepara para ser transportado a la capa modelo y consumir el servicio.
- Finalmente, en la capa modelo se gatilla la función “createProducto”, recibiendo por parámetro el objeto de tipo producto. Dentro de la función se genera la petición Http de tipo post y el objeto producto es transformado en formato Json y enviado al servicio correspondiente en el back-end.

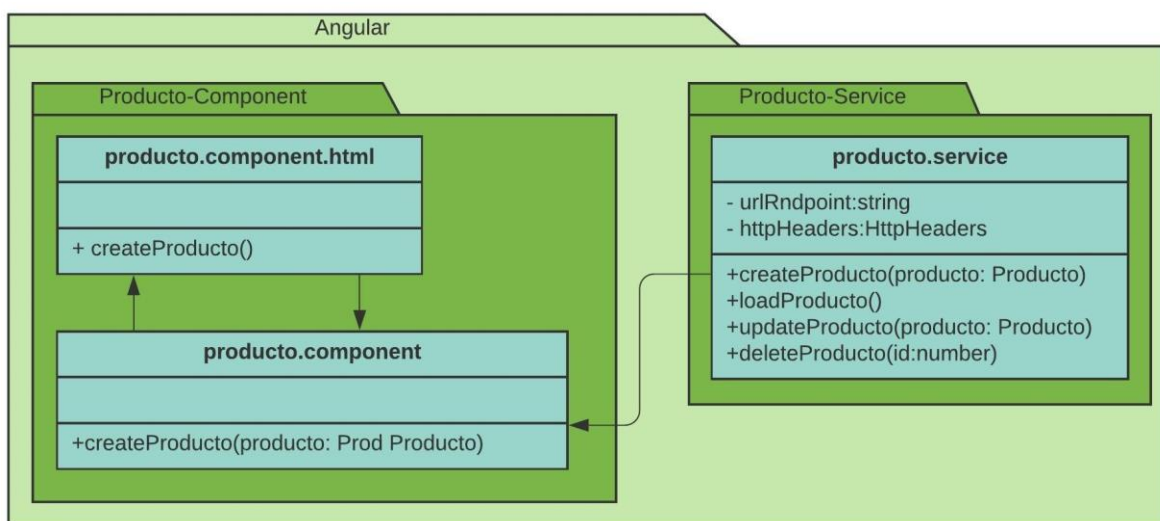


Figura 8: Ejemplo parcial Modelo Vista Presentador

En conclusión, el desarrollo del front-end cumple con las características del patrón MVP, puesto que la vista envía el objeto requerido al presentador. El presentador cumple la función

de intermediario entre la vista y el modelo y, por último, el modelo recibe los datos que envía el presentador y consume el servicio del back-end.

### 7.2.2 Arquitectura Orientada a Servicios (SOA)

Para el desarrollo del proyecto se ha utilizado la Arquitectura Orientada a Servicios (SOA), esta arquitectura se basa en la utilización de servicios que cumplen una función en específico sobre la lógica de negocios (SEI, s.f.). En el desarrollo del software se encuentra el front-end (Angular) que cumple la función de web server y el back-end (SpringBoot) cumple la función de service gateway. En la Figura 9 se representa el flujo donde existen clientes que generan peticiones en el web server, posteriormente este envía la petición al service gateway. La petición enviada desde el web server es mediante una URL, mediante esa URL el service gateway sabe qué servicio utilizar. Finalmente, el back-end envía la respuesta de vuelta al front-end y este la envía al cliente.

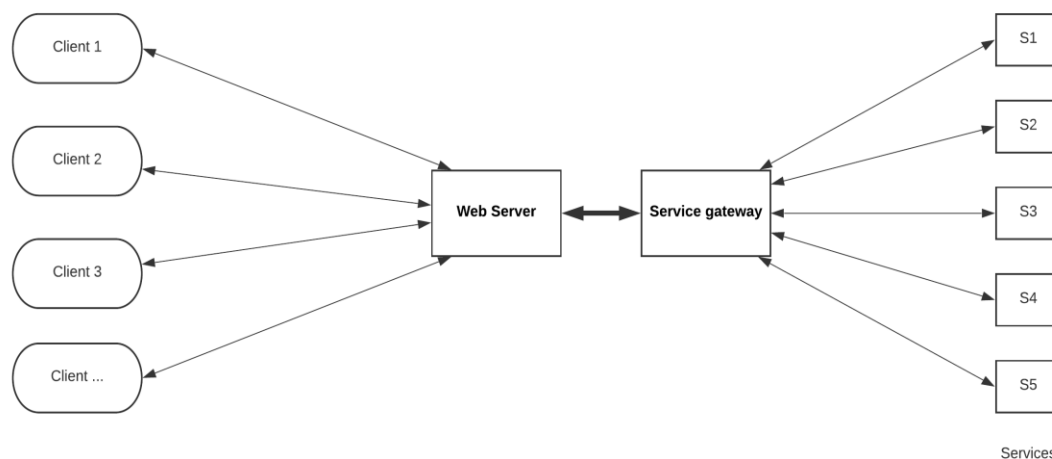


Figura 9: Diagrama de interacción SOA (Sommerville, 2020)

El back-end se ha construido utilizando la opción tecnológica REST, empleando peticiones o servicios de tipo HTTP para, posteriormente, enviar respuestas en formato json al cliente que consume dichos servicios.

El código del back-end se encuentra organizado por capas. La capa de entidades contiene clases que representan un elemento del modelo y que corresponden directamente a una tabla de la base de datos. La capa de repositorios (DAO) está compuesta por interfaces que tienen como fin generar la persistencia en la base de datos. En la capa de servicios se utilizan interfaces y clases de implementación, donde se definen operaciones a nivel de negocio. Finalmente, se encuentra la capa de los controladores REST los cuales utilizan las funciones que se encuentran en la capa de servicios. Además, en esta capa se definen las respuestas enviadas al front-end.

En la Tabla 22 y Tabla 24, se presentan ejemplos de las peticiones realizadas a las funciones de agregar producto y eliminar producto, correspondientes a los CU01 y CU02, estas peticiones son de tipo POST y DELETE, respectivamente.



ID	RF01
URL	http://localhost:8080/api/producto/
Método	POST
Parámetros de URL	Ninguno
Parámetros de datos	Bearer Token: {token}
Respuesta exitosa	<p>Ejemplo: Token: {token} Contenido: {   "nombre": "Mesa de roble ancestral",   "stock": 687,   "descripcion": "Mesa de roble tallada a mano",   "valorUnitario": 100000,   "categoria": {     "id": 1,     "nombre": "Mesas"   } }</p> <p>Estado: 201, Creado. Mensaje: El producto ha sido creado con éxito</p>
Respuesta fallida.	<p>Ejemplo: Token: {token} Contenido: {} Estado: 400, Petición errónea. Mensaje: Los campos del producto no deben ser nulos</p> <p>Ejemplo: Token: null Contenido: {   "nombre": "Mesa de roble ancestral",   "stock": 687,   "descripcion": "Mesa de roble tallada a mano",   "valorUnitario": 100000,   "categoria": {     "id": 1,     "nombre": "Mesas"   } }</p> <p>Estado: 201, No autorizado. Mensaje: La autenticación es requerida.</p>

Tabla 22: Ejemplo de petición POST de productos

ID	RF03
URL	http://localhost:8080/api/producto/
Método	DELETE
Parámetros de URL	{id}
Parámetros de datos	Bearer Token: {token}
Respuesta exitosa	<p>Ejemplo: Token: {token} Contenido: id: 1 Estado: 200, Ok Mensaje: Producto eliminado con éxito.</p>

Tabla 23: Ejemplo de petición DELETE de productos

Respuesta fallida.	Ejemplo: Token: {token} Contenido: id: 10000 Estado: 500 Mensaje: Producto null.  Ejemplo: Token: null Contenido: id: 2 Estado: 401, No autorizado Mensaje: La autenticación es requerida.
--------------------	--

Tabla 24: Continuación, Ejemplo de petición DELETE de productos.

### 7.2.3 Diagrama de clases

En el siguiente apartado se presenta el diagrama de clases correspondientes al CU01, CU02, CU03 y CU04.

Debido a la magnitud del proyecto no será posible representar todo el sistema a través de un solo diagrama, puesto que el proyecto cuenta con más de cincuenta clases considerando todos los requisitos del sistema. Por otro lado, el funcionamiento del resto de las clases es similar.

En la Figura 10 se muestra el diagrama de clases correspondiente a la gestión del producto. En el diagrama se puede apreciar la capa de entidades (Producto), la capa de repositorios (ProductoDao), la capa de servicios (ProductoServiceImpl y la interfaz IProductoService) y, por último, la capa del controlador (ProductoRestController), la cual cumple la función de enviar las respuestas y recibir las peticiones por parte del front-end.

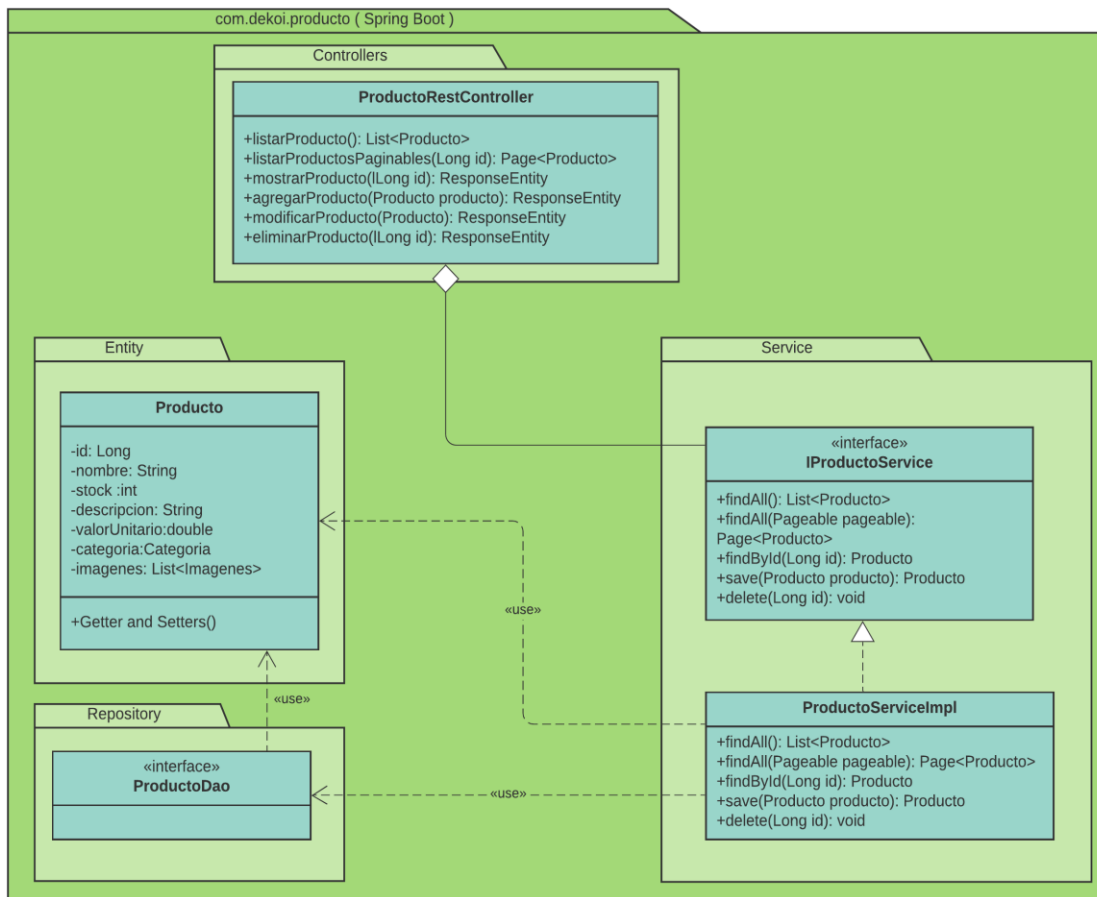


Figura 10: Diagrama de clases Back-end de Dekoi

En la Figura 11, se expone cómo son consumidos los servicios correspondientes a los productos, enfocándose en el servicio “Agregar Producto” (CU01), empezando desde la petición generada en la vista (Angular) y finalizando con la respuesta que entrega el back-end (Spring Boot) a Angular.

En el módulo “Angular”, se puede observar el servicio “Producto-Service” que consume los servicios del back-end y el componente “Producto-Component”. Este a su vez contiene la vista “producto.component.html” que es presentada al usuario. La interacción entre estos componentes fue explicada en el apartado 7.2.1.

La vista correspondiente a “Producto-Component” se encuentra vinculada al “producto.component.ts”, este contiene las llamadas al servicio y la lógica de programación. Para que la función “createProducto()” se pueda ejecutar, el administrador debe iniciar sesión y llenar el formulario del producto. Una vez enviado el formulario, “Angular” invoca la función “createProducto()” en el componente, e invoca el mismo método en el servicio “producto.service” (presentado en la Figura 11), que finalmente consume el servicio del back-end mediante el atributo “urlEndpoint” (1).

Por el lado del back-end, se encuentra el controlador “ProductoController”, el cual ocupa diferentes métodos dependiendo de la URL que sea solicitada (en este caso “crearProducto()”), el controlador tiene el atributo “IProductoService” el cual inyecta dicha interfaz con el método de creación de productos (2). Esta es provista por la clase

“ProductoServiceImpl” (utilizando la función “saveProducto()”) donde se ejecuta la lógica de negocio e implementa la interfaz “IProductoDao” (3), la cual utiliza el modelo “Producto” (4) para almacenar el producto en la base de datos.

Posterior a que la capa DAO registre el nuevo producto en la base de datos, devuelve el producto creado a la capa de servicios (5), donde es retornado al controlador (6), para luego ser añadido a la respuesta la cual se envía al “Producto-Service”. Finalmente se notifica en la vista que el producto fue ingresado con éxito o, por el contrario, no fue posible agregar el producto.

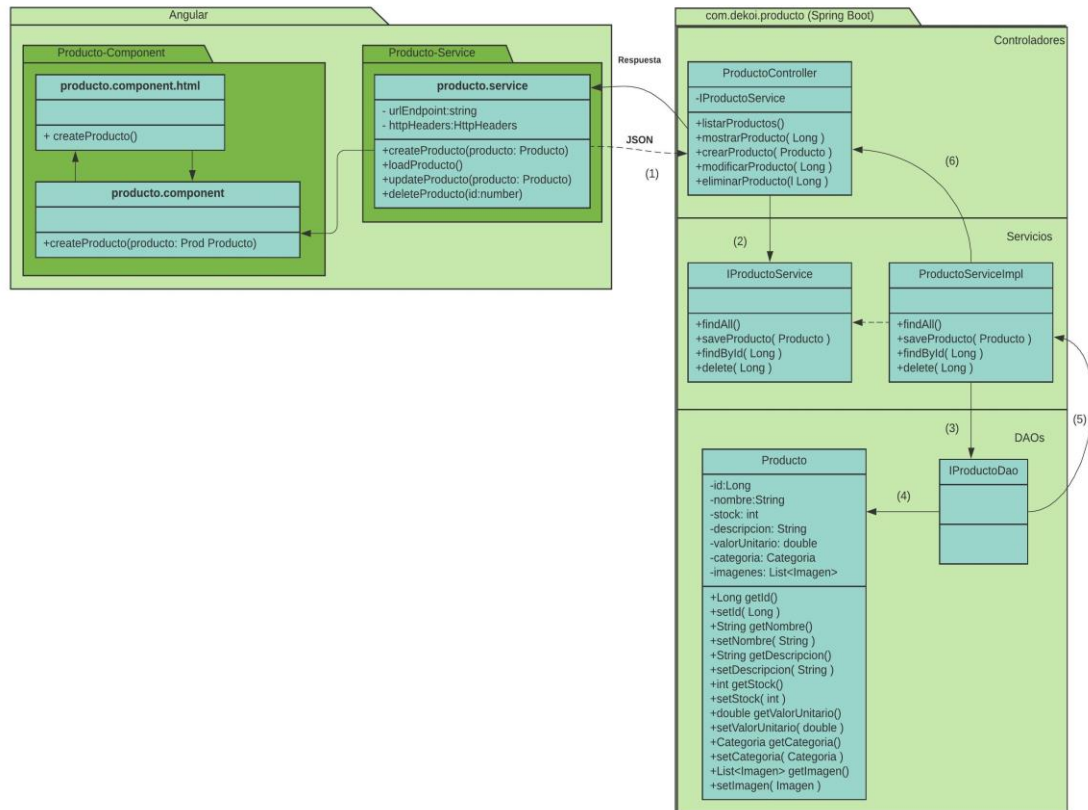


Figura 11: Diagrama de clases Front-end y Back-end de Dekoi

### 7.3 Diseño interfaz y navegación

En el siguiente apartado se presenta el diseño de la interfaz y las jerarquías de menú de la aplicación, incluyendo imágenes de las pantallas del software.

#### 7.3.1 Jerarquía del menú de la aplicación

La Figura 12 representa la forma de navegar entre las diferentes vistas, tomando como punto de partida el inicio.

El sistema de navegación se encuentra compuesto por los rectángulos que representan las vistas del sistema de Dekoi. Las flechas bidireccionales muestran desde y hasta que pantalla se puede dirigir. Por último, las líneas punteadas representan las vistas alternativas dependientes del perfil que tenga el usuario.

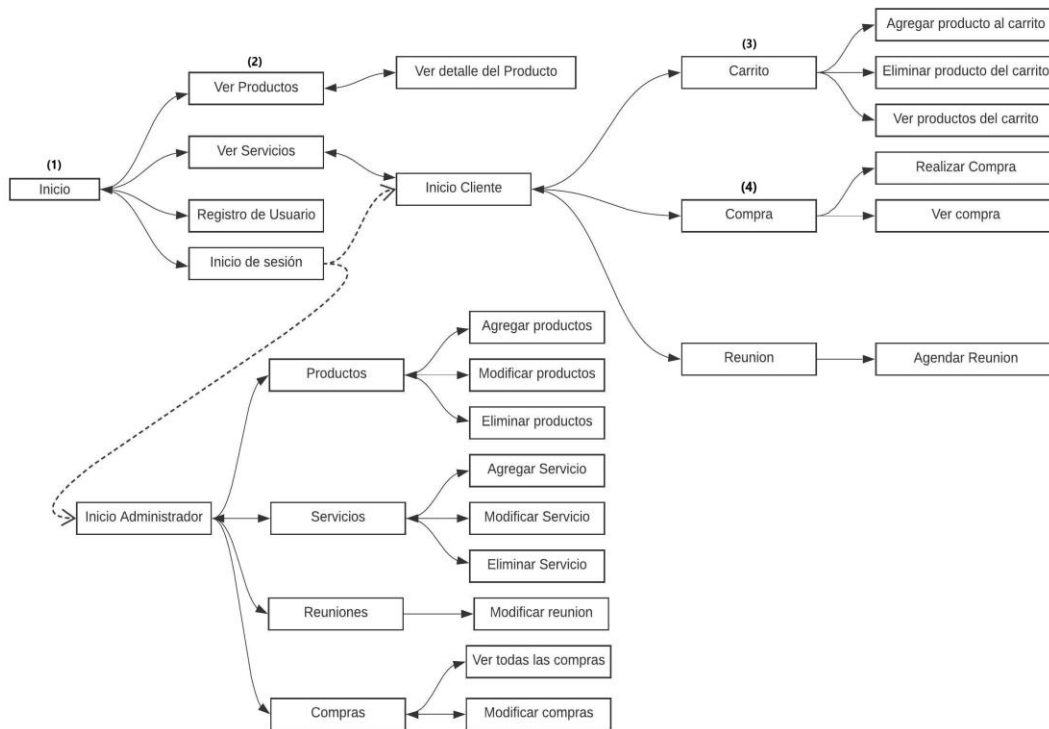


Figura 12: Diagrama de navegación

### 7.3.2 Especificación de diseño de la interfaz

En el apartado de especificación de diseño de la interfaz, se presentan cuatro vistas, enumeradas previamente en la Figura 12, con el fin de evidenciar la organización de los elementos de cada pantalla.

En la Figura 13 y Figura 14 se presenta la página de inicio de Dekoi. Esta página es visualizada por cualquier usuario, registrado o no. La página se compone por cuatro partes, las cuales se detallan a continuación:

1. Barra de navegación: Está compuesta por el logo de la empresa, el cual redirige a la página de inicio. También contiene los botones de navegación: Inicio, Productos, Servicios, Sobre Nosotros, Contacto, el carrito y el Inicio de sesión
2. Contenido: Muestra una serie de imágenes pertenecientes a los trabajos realizados por la empresa e información sobre la misma.
3. Productos destacados: Muestra una serie de productos destacados del sistema, con el nombre, precio y un botón para agregar al carrito.
4. Servicios: Expone los servicios que realiza la empresa.

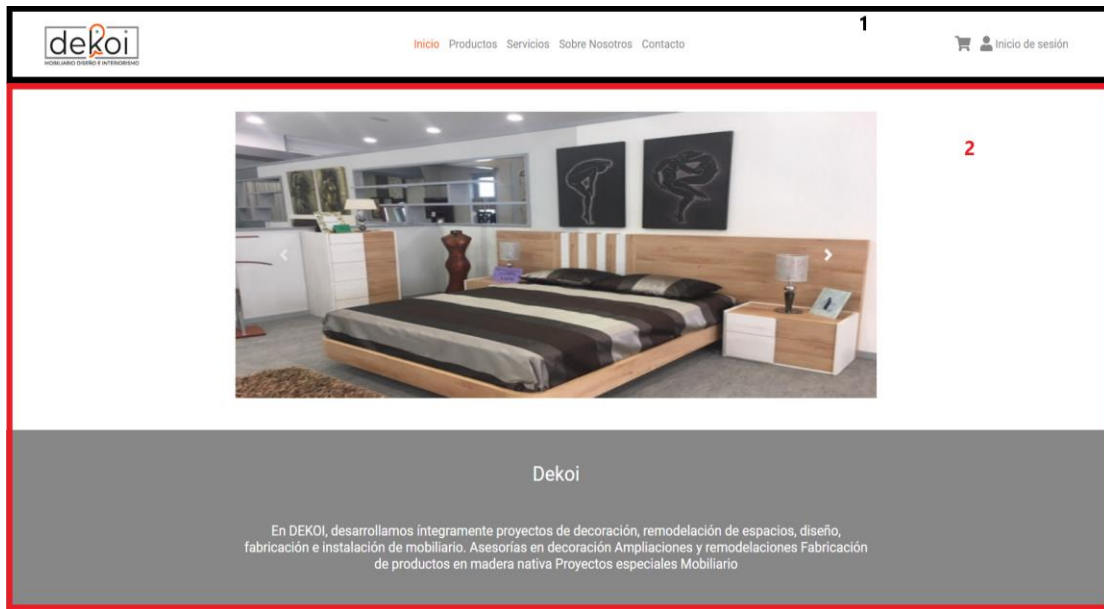


Figura 13: Página de inicio

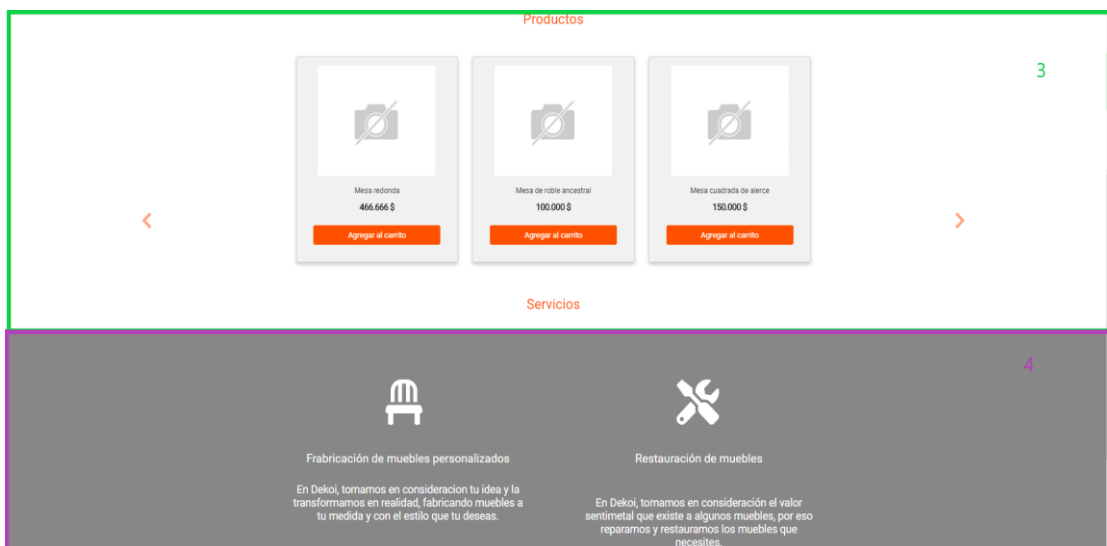


Figura 14: Continuación, Página de inicio

En la Figura 15 se presenta la pantalla “Ver productos”, a esta página puede acceder cualquier usuario, registrado o no en el sistema. Para acceder a esta pantalla, se debe seleccionar el botón “Productos” de la barra de navegación.

Esta pantalla está compuesta por cuatro áreas:

1. Barra de navegación: Está compuesta por el logo de la empresa, el cual redirige a la página de inicio. También contiene los botones de navegación: Inicio, Productos, Servicios, Sobre Nosotros, Contacto, el carrito y el Inicio de sesión.
2. Migajas de pan: Muestra el lugar de la página donde está ubicado.
3. Categorías: Muestra un listado de las categorías que existen en el sistema.
4. Productos: Exhibe el listado de los productos, en la figura muestra dos productos con sus respectivas imágenes.

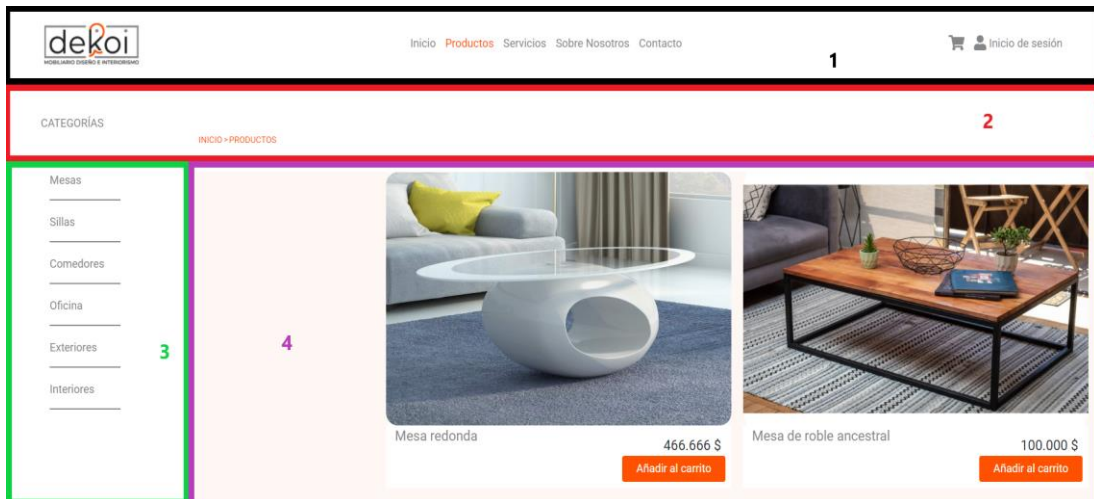


Figura 15: Pagina ver productos

En la Figura 16 se presenta la página de “Carrito”, a esta página pueden acceder solo usuarios registrados. Para acceder a ella se debe presionar el icono del carrito en la parte derecha de la barra de navegación

Esta pantalla está compuesta por cuatro áreas:

1. Barra de navegación: Está compuesta por el logo de la empresa, el cual redirige a la página de inicio. También contiene los botones de navegación: Inicio, Productos, Servicios, Sobre Nosotros, Contacto, el carrito y el Inicio de sesión.
2. Migajas de pan: Muestra el lugar de la página donde está ubicado.
3. Listado de productos: Muestra un listado de los productos que se encuentran en el carrito de compras, además de una opción eliminar productos del carrito y un subtotal de la compra.
4. Dirección de envío: Muestra un formulario con la dirección de envío para la compra.

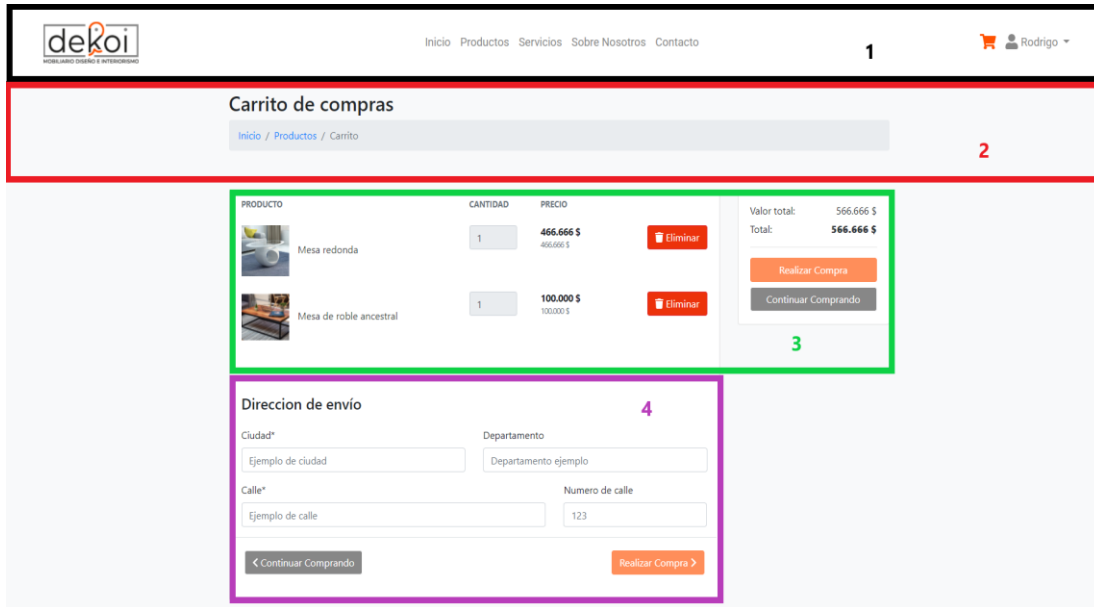


Figura 16: Pagina del carrito

En la Figura 17, se muestra la página del perfil de usuario, donde este contenido el detalle de las compras realizadas por un usuario, a esta vista solo pueden acceder los usuarios registrados. Para acceder a esta página se debe presionar el icono de usuario en la barra de navegación y luego presionar “mi cuenta”.

Esta pantalla está compuesta por cuatro secciones:

1. Migajas de pan: Muestra el lugar de la página donde está ubicado.
2. Menú lateral: Muestra una opción para la configuración de la cuenta y para ver todas las compras realizadas.
3. Datos de usuario: Muestra los datos del usuario con posibilidad de hacer cambios en ellos.
4. Última compra: Muestra los datos de la última compra realizada.



The screenshot shows a user account page with the following elements:

- 1**: Top navigation bar containing "Inicio / Mi cuenta" and a user profile icon.
- 2**: Left sidebar menu with options: "Configuracion de cuenta", "Mis Compras", and "Cerrar sesion".
- 3**: User profile section for "Rodrigo Cifuentes" (ro080895@gmail.com) with a "Modificar datos" link. Below it, the address "Chillán, Monte Tibidabo 1339" is shown with a "Modificar direccion" link.
- 4**: Order summary section for "ID de la compra: #2" dated "20 de noviembre de 2020" with status "Por Pagar". It includes:
  - Total: 666.666 \$
  - Contacto: Rodrigo Cifuentes, 942750783, ro080895@gmail.com
  - Direccion de envío: 1339, Monte Tibidabo, Chillán
  - Order items:
    - Mesa redonda: 466.666 \$ X 1
    - Mesa de roble ancestral: 100.000 \$ X 1
    - Mesa de roble ancestral: 100.000 \$ X 1
  - A "Pagar" button at the bottom.

Figura 17: Página de Compra

## 8 SEGURIDAD Y PRUEBAS

### 8.1 Seguridad

En el siguiente apartado se detalla la implementación de seguridad en relación del back-end con el front-end, desde el inicio de sesión de un usuario hasta las diversas peticiones que este pueda generar.

Para la arquitectura presentada, en el apartado 7.2.2, se tomó la decisión de implementar una securitización basada en token de acceso utilizando JWT. "JSON Web Token (JWT) es un estándar abierto (RFC 7519) que define un modo compacto y autónomo para transmitir de forma segura la información entre las partes como un objeto JSON. Esta información puede ser verificada, además de ser confiable por poseer una firma digital. Los JWT se pueden firmar usando un secreto (con el algoritmo HMAC) o utilizando un par de claves públicas y privadas usando RSA o ECDSA" (jwt.io, 2015). El JWT está compuesto por tres partes, la primera contiene el hash para encriptar el token (header), una serie de atributos propios del usuario (payload) y la concatenación entre el header y el payload ya encriptados (firma).

El proceso de la autenticación por token da comienzo cuando se genera una petición de tipo POST en el método de inicio de sesión del cliente (Angular), esta petición lleva las credenciales del usuario hacia el servidor (Spring), el servidor valida dichas credenciales y le devuelve al cliente el token de acceso (JWT) y el código HTTP 200 en caso de ser exitoso el inicio de sesión. En caso contrario este devuelve un código HTTP 401 y le niega el acceso. Una vez el usuario recibe el JWT este se encuentra habilitado para generar peticiones de tipo HTTP. Cuando genera alguna de estas peticiones, envía al servidor el tipo de esta, el JWT y algún objeto o parámetro que sea necesario. El servidor valida si este token corresponde al usuario que tiene la sesión iniciada o si el token no ha expirado. Si este es el caso, el servidor realiza las operaciones en el back-end y en la base de datos para posteriormente entregar la respuesta y el código HTTP 200 si la operación es exitosa como se presenta en la Figura 18.

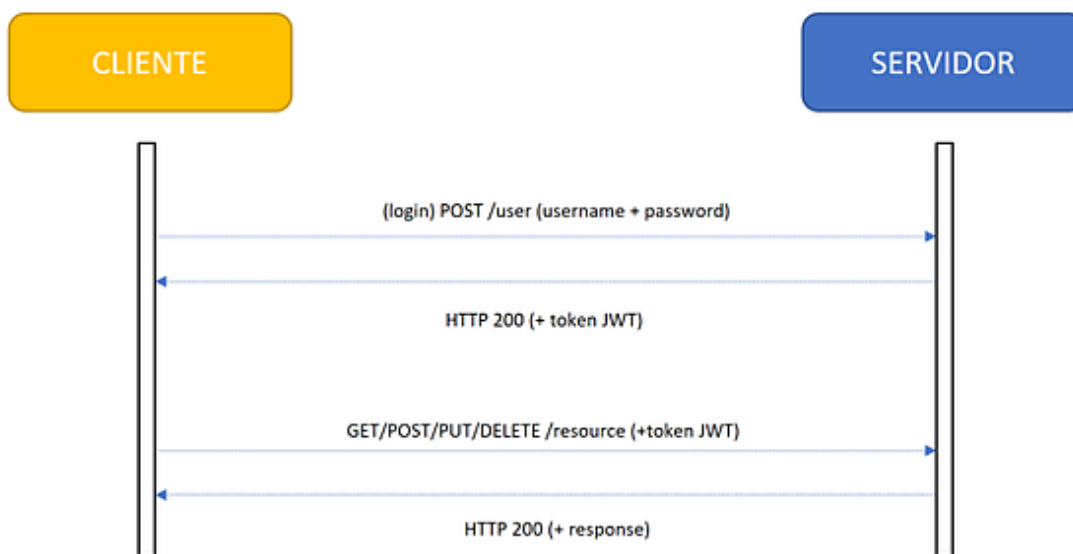


Figura 18: Proceso de autenticación mediante tokens

Con la finalidad de diferenciar y restringir las diversas funcionalidades del sistema se crearon dos roles, ofreciendo así, una mayor confidencialidad a los usuarios respecto a los datos sensibles. Los roles mencionados se dividen en rol de administrador, quien tienen acceso total, permitiéndole realizar funciones exclusivas para la administración y también las funciones de rol cliente; y rol de cliente cuando se registra en el sistema, teniendo acceso restringido a las funciones del mismo, esto se ve reflejado en el apartado 6.2.

Respecto a lo anteriormente señalado, en relación a la autenticación mediante tokens, la diferenciación se realiza por el lado del servidor. El proceso inicia con la petición de inicio de sesión enviada al servidor, el cual recibe las credenciales del usuario y realiza una búsqueda por el nombre del usuario. Si los datos corresponden, se comparan la clave enviada con la clave encriptada almacenada en la base de datos, todo esto manejado de manera interna por las dependencias de seguridad de Spring. Una vez autenticado el usuario en el servidor se concatena el rol del usuario al payload del token de acceso. Identificando el rol que tiene el usuario por el lado de Angular y Spring, restringiendo así las funciones que no correspondan a dicho rol.

## 8.2 Pruebas

El apartado de pruebas está compuesto por cuatro secciones, donde se detallan las pruebas realizadas a los diversos elementos del software. El primer apartado presenta los elementos a analizar, donde se introducen los componentes o módulos del software que serán sujetos de estas pruebas. Posteriormente, se encuentran las especificaciones de las pruebas a realizar, donde se detallan los aspectos que evaluará cada tipo de prueba. En el tercer apartado se detallan, con más especificidad, las pruebas a los módulos mencionados en el punto 8.2.1, precisando el objetivo, el flujo principal, resultados de éxito y de fallo.

Finalmente, se presenta una breve conclusión referente a los resultados de las pruebas realizadas.

### 8.2.1 Elementos de prueba

Para poder corroborar el correcto funcionamiento del software se realizaron pruebas en los módulos más importantes del sistema de ventas y prestación de servicios de Dekoi.

#### Gestión de Productos

- Agregar Producto
- Editar Producto
- Eliminar Producto
- Ver Productos

#### Gestión de carrito de compras

- Agregar Producto al carrito de compras
- Eliminar Producto del carrito de compras
- Mostrar Carrito de compras.

#### Gestión de cuentas de usuario

- Inicio de sesión
- Registro de usuarios
- Modificar datos de usuario

#### Gestión de Compra

- Generar Compra
- Modificar Compra
- Ver Compras

Gestión de servicios

- Agregar Servicio
- Modificar Servicio
- Eliminar Servicio
- Ver Servicio

### 8.2.2 Especificación de las pruebas

Los aspectos que se prueban son de funcionalidad, interfaz y navegación. A continuación, se presentan las especificaciones de las pruebas realizadas.

<b>Pruebas de Sistema</b>	
<b>Características a probar</b>	Funcionalidad
<b>Nivel de prueba</b>	Sistema
<b>Objetivo de la prueba</b>	Asegurar el correcto funcionamiento del sistema
<b>Enfoque para la definición de casos de prueba</b>	Caja negra
<b>Técnicas para la definición de pruebas</b>	Para las funcionalidades a probar se utilizan datos válidos como inválidos para verificar: <ul style="list-style-type: none"> <li>• Los resultados esperados cuando se utilizan datos válidos.</li> <li>• Mensajes de error por datos inválidos o vacíos.</li> </ul>
<b>Actividades de prueba</b>	<ul style="list-style-type: none"> <li>• Definir los casos de pruebas considerando datos inválidos.</li> <li>• Aplicar pruebas a funcionalidades específicas.</li> </ul>
<b>Criterios de aceptación</b>	Los casos de pruebas se ejecutan según lo esperado.

*Tabla 25: Pruebas de sistema*

<b>Pruebas de Usabilidad</b>	
<b>Características a probar</b>	Interfaz y Navegación
<b>Nivel de prueba</b>	Aceptación
<b>Objetivo de la prueba</b>	Asegurar la adecuada interacción del usuario con el sistema
<b>Enfoque para la definición de casos de prueba</b>	Caja negra
<b>Técnicas para la definición de pruebas</b>	El sistema no debe presentar los siguientes problemas de usabilidad <ul style="list-style-type: none"> <li>• El sistema es demasiado complejo y difícil de usar.</li> <li>• Mensajes de error no representativos para las acciones realizadas.</li> </ul>
<b>Actividades de prueba</b>	<ul style="list-style-type: none"> <li>• Crear casos de prueba para que funcione de manera adecuada.</li> <li>• Analizar los datos enviados y recibidos por los usuarios.</li> </ul>
<b>Criterios de aceptación</b>	Para el administrador y el cliente les resulta sencillo utilizar el sistema. El administrador puede gestionar sin problemas la página.

*Tabla 26: Pruebas de usabilidad*

### 8.2.3 Detalle de las pruebas

Las pruebas realizadas en el proyecto fueron efectuadas en el front-end y en el back-end, de este modo se asegura el correcto funcionamiento de ambas partes.

En la Tabla 27 se presentan las pruebas efectuadas en el sistema, los resultados obtenidos, un indicador de fallo o éxito y por último una observación. El detalle de las pruebas se encuentra en el Anexo B: Pruebas de Unidad

ID	Tipo de Prueba	Resultado Esperado	Resultado Obtenido	Indicador	Observación
P01	Prueba de sistema	El sistema agrega la categoría.	El sistema agrega la categoría.	Éxito	Ninguna.
P02	Prueba de sistema	El sistema agrega el producto.	El sistema agrega el producto.	Éxito	Ninguna.
P03	Prueba de sistema	El sistema modifica el producto.	El sistema agrega el producto.	Éxito	Ninguna.
P04	Prueba de sistema	El sistema elimina el producto.	El sistema no puede eliminar el producto	Fallo	Debido a que existe un carrito o una compra con un producto, el sistema no puede eliminar el producto.
P05	Prueba de sistema	El sistema agrega un producto al carrito	El sistema agrega un producto al carrito	Éxito	Si el producto no tiene stock no es agregado al carrito.
P06	Prueba de sistema	El sistema muestra los productos del carrito.	El sistema muestra los productos del carrito	Éxito	Ninguna.
P07	Prueba de seguridad	El sistema permite iniciar sesión a un usuario.	El sistema permite iniciar sesión a un usuario.	Éxito	Ninguna.
P08	Prueba de seguridad	El sistema cierra la sesión del usuario.	El sistema cierra la sesión del usuario.	Éxito	Ninguna.
P09	Prueba de sistema	El sistema permite crear una cuenta de usuario.	El sistema permite crear una cuenta de usuario.	Éxito	Ninguna.
P10	Prueba de sistema	El sistema modifica los datos del usuario.	El sistema modifica los datos del usuario.	Éxito	Ninguna.
P11	Prueba de sistema	El sistema genera la compra exitosamente.	El sistema genera la compra exitosamente.	Éxito	El pago de la compra se realiza después de enviar el formulario.
P12	Prueba de sistema	El sistema despliega la lista de los productos.	El sistema despliega la lista de los productos.	Éxito	Ninguna.
P13	Prueba de sistema	El sistema despliega las compras realizadas.	El sistema despliega las compras realizadas.	Éxito	Ninguna.
P14	Prueba de sistema	El sistema modifica el estado de la compra generada.	El sistema modifica el estado de la compra generada.	Éxito	Ninguna.
P15	Prueba de sistema	El sistema permite ingresar el servicio.	El sistema permite ingresar el servicio.	Éxito	Ninguna.
P16	Prueba de sistema	El sistema lista todos los servicios	El sistema lista todos los servicios	Éxito	Ninguna.
P17	Prueba de sistema	El sistema permite modificar el servicio	El sistema permite modificar el servicio	Éxito	Ninguna.
P18	Prueba de sistema	El sistema permite eliminar el servicio.	El sistema permite eliminar el servicio.	Éxito	Ninguna.

Tabla 27: Resumen de las pruebas realizadas

### 8.2.4 Pruebas de usabilidad

Las pruebas de usabilidad realizadas en el proyecto fueron efectuadas en el front-end, con la finalidad de medir la interacción entre los usuarios y el software.

Para la realización de las pruebas de usabilidad, se utilizó la “Escala de Usabilidad de un Sistema” (EUS). La EUS es una herramienta metodológica que tiene como fin medir la usabilidad de un objeto, dispositivo o aplicación y es uno de los métodos de medición más utilizados en la “Experiencia de Usuario” (UX) (Devin, 2017).

La escala consta de 10 preguntas, cada una de las preguntas esta puntuada de uno a cinco, donde uno significa Totalmente desacuerdo y cinco significa Totalmente acuerdo (Devin, 2017).

En la Tabla 28 se presenta la encuesta realizada a diez usuarios. Cada pregunta muestra la calificación promedio entre todos los participantes.

Preguntas	Calificación
1. Creo que usaría esta aplicación frecuentemente	3,8
2. Encuentro esta aplicación innecesariamente compleja.	1,4
3. Creo que la aplicación fue fácil de usar.	4,6
4. Creo que necesitaría ayuda de una persona con conocimientos técnicos para usar esta aplicación.	1,2
5. Las funciones de esta aplicación están bien integradas.	4,4
6. Creo que la aplicación es muy inconsistente.	1,4
7. Imagino que la mayoría de la gente aprendería a usar esta aplicación en forma muy rápida.	4,6
8. Encuentro que la aplicación es muy difícil de usar.	2,4
9. Me siento confiado al usar esta aplicación.	4,8
10. Necesité aprender muchas cosas antes de ser capaz de usar esta aplicación.	1,8

Tabla 28: Encuesta de Usabilidad

Para calcular el resultado de la encuesta de usabilidad, se utilizó el algoritmo de SUS. El algoritmo consiste en tomar los valores asignados por los usuarios, restarles un valor y, finalmente, multiplicarlos por 2,5.

Para las preguntas impares se les restará 1 al valor asignado por los usuarios y a las preguntas pares será de 5 restado por el número asignado por los entrevistados.

En la Figura 19 se muestra la fórmula para calcular el resultado de la encuesta.

$$X = ((c1 - 1) + (5 - c2) + (c3 - 1) + (5 - c4) + (c5 - 1) + (5 - c6) + (c7 - 1) + (5 - c8) + (c9 - 1) + (5 - c10)) * 2,5$$

Figura 19: Fórmula para calcular el resultado de las pruebas

Una vez aplicada la formula con las calificaciones obtenidas, el resultado de la encuesta SUS es de 85 puntos de un total de 100, lo que quiere decir que el software es altamente aceptable.

### **8.2.5 Conclusiones de Pruebas**

Las pruebas de sistema realizadas tanto en el front-end como en el back-end, donde se pusieron a prueba los casos de usos, lograron alcanzar y satisfacer los criterios de aceptación de cada caso de uso. Sin embargo, con mayor tiempo y conocimientos se puede abordar completamente todas las funcionalidades.

El único fallo que se presentó en las pruebas realizadas fue ocasionado por un problema en la eliminación del producto, debido a que este debe presentar una eliminación lógica en el sistema. Esto fue corregido y permitió mejorar el funcionamiento del sistema.

Las pruebas de usabilidad fueron realizadas con el fin de evaluar la interacción del usuario con el front-end del software, para esto se utilizó la encuesta SUS descrita en el apartado 8.2.4. Gracias a la encuesta de usabilidad se pudo evaluar el nivel de satisfacción de los clientes al usar la aplicación, además de identificar fallos visuales de la presentación de productos.

De lo anteriormente expuesto, es posible concluir que la realización de pruebas de software permite identificar errores y fallos que puedan ocurrir en el transcurso del proceso de desarrollo y que se hayan pasado por alto previamente. De este modo, cualquier error existente puede ser corregido para lograr un mejor desempeño del software, por lo que la inclusión o realización de pruebas dentro de cualquier proyecto es imprescindible.

---

## 9 CONCLUSIONES

---

Dado que los objetivos generales y los objetivos específicos del proyecto fueron abordados en su mayoría, se puede concluir que el proyecto ha finalizado con éxito. Pese a esto, existe un objetivo específico que no fue posible desarrollar, debido a que, para ampliar el mercado potencial de la empresa, el software debe estar en operativo en un servidor anexo al de la universidad. Esto no pudo ser posible debido a que aún no se adquiere el host y el dominio.

En cuanto a la metodología de desarrollo, la metodología iterativo incremental en conjunto con Kanban resultó de suma importancia debido a que ayudó a la detección de errores, estos fueron detectados a través de las reuniones realizadas con el cliente al final de cada incremento, para posteriormente ser corregidos. Además, gracias a la metodología Kanban se logró mantener una estructura de trabajo, estableciendo metas y tareas por semana las cuales fueron cumplidas.

Los modelos BPMN cumplieron un rol fundamental al momento de identificar dos posibles oportunidades de negocios. Los modelos sirvieron para entender la lógica de negocios sobre cómo la empresa vende sus productos y presta sus servicios. En cuanto al diagrama de casos de uso, permitió identificar cuántos tipos de usuarios iban a interactuar con el sistema y qué funciones deberían realizar.

En cuanto a la arquitectura orientada a servicios y la opción tecnológica Api Rest, se comprueba la creación de servicios independientes y altamente mantenibles, puesto que cuando era necesario realizar modificaciones en algún servicio, esta se realizaba sin problemas y no afectaba al resto del código. En cuanto a la mantenibilidad que ofrece Api Rest, esta se vio reflejada en la separación del código de Angular, del código de SpringBoot, debido a esto la incorporación de nuevos servicios y/o módulos no presentó dificultades mayores.

El software desarrollado logró satisfacer las necesidades del cliente desarrollando la mayor parte de los requisitos solicitados. Por otra parte, hubo requisitos no abarcados, tales como: la generación de reportes, el sistema de seguimiento de envíos y la implementación de Google Calendar. Cabe destacar que se realizaron encuestas al cliente y diferentes usuarios sobre el aspecto visual del sistema, obteniendo resultados positivos por parte de ellos.

El proyecto tenía una planificación de tres meses, donde fueron abarcados la mayoría de los requisitos, a pesar de ello por motivos de la contingencia mundial por el "Covid-19", las reuniones, tanto con el diseñador como con el cliente, se dificultaron. Por el motivo anterior, realizar pruebas, consultas y obtener retroalimentación fue dificultoso.

Académicamente fue una experiencia llena de aprendizajes, donde se puso en práctica los aprendizajes obtenidos a lo largo de la carrera. Además, se adquirieron nuevos conocimientos utilizando el framework de desarrollo Angular, esto representó un desafío durante todo el proyecto, puesto que no se tenía conocimiento previo de la herramienta, sin embargo, se logró un aprendizaje fluido y enriquecedor. En cuanto a Spring Boot, se tenía conocimiento previo, lo que permitió realizar investigaciones sobre conceptos más específicos lo que permitió alcanzar mejores resultados.

En cuanto a lo personal, el desarrollo del proyecto fue fructífero, puesto que se adquirió la estructura de trabajo bajo las metodologías ágiles, utilizando tableros para mantener una



buena organización. también, se adquirió una mentalidad autodidacta y perseverante para resolver problemas de programación, buscando información en libros y en diferentes foros de programación.

Para finalizar, el proyecto será utilizado por la administradora de “Dekoi”, y posteriormente, abierta al público.

## 9.1 Trabajo futuro

Debido a la contingencia mundial se dificultaron las reuniones con el cliente, debido a esto algunas de las funcionalidades no se abordaron. Por otro lado, algunas de estas funcionalidades no se lograron realizar por falta de conocimiento del desarrollador. Por lo anterior se decidió dejar como trabajo futuro las siguientes funcionalidades:

- El sistema debe realizar pagos automáticos vía WebPay.
- El sistema puede generar informes de ventas y stock, con el fin de registrar cuales productos son los más vendidos y cuales tienen baja cantidad en inventario.
- El aspecto visual funcione correctamente en dispositivos móviles en un futuro.
- El sistema debe generar descuentos sobre los productos más vendidos.
- Integrar un calendario de reuniones mediante la api de Google Calendar para automatizar la gestión de reuniones.
- El sistema debe presentar un seguimiento de envío sobre la compra realizada.
- El sistema debe tener un módulo de postventa para que los clientes realicen comentarios con respecto a las compras realizadas.

## Bibliografía

- Object Technology International. (7 de Noviembre de 2001). *www.eclipse.org/*. Obtenido de [https://es.wikipedia.org/wiki/Eclipse\\_\(software\)](https://es.wikipedia.org/wiki/Eclipse_(software))
- Bizagi. (1989). *www.Bizagi.com*. Obtenido de <https://es.wikipedia.org/wiki/Bizagi>
- Chen, P. (1976). Obtenido de [https://es.wikipedia.org/wiki/Modelo\\_entidad-relaci3n](https://es.wikipedia.org/wiki/Modelo_entidad-relaci3n)
- CSS Working Group. (17 de Diciembre de 1995). Obtenido de <https://www.hostinger.es/tutoriales/que-es-css/>
- Devin, F. (25 de Febrero de 2017). *uxpanol*. Obtenido de <https://uxpanol.com/teoria/sistema-de-escalas-de-usabilidad-que-es-y-para-que-sirve/>
- ecityclíc. (s.f.). *ecityclíc*. Obtenido de <https://www.ecityclíc.com/es/noticias/que-es-soa-o-arquitectura-orientada-a-servicios#:~:text=La%20Arquitectura%20Orientada%20a%20Servicios%20de%20cliente%2C%20conocida%20tambi3n%20como,a%20los%20requisitos%20del%20negocio.>
- EcuRed. (s.f.). *EcuRed*. Obtenido de [https://www.ecured.cu/Caso\\_de\\_uso](https://www.ecured.cu/Caso_de_uso)
- EcuRed. (s.f.). *EcuRed.cu*. Obtenido de [https://www.ecured.cu/Caso\\_de\\_uso](https://www.ecured.cu/Caso_de_uso)
- Fielding, R. (2000). *Geeky Theory*. Obtenido de <https://geekytheory.com/que-es-una-api-rest-y-para-que-se-utiliza>
- GitHub. (26 de Febrero de 2014). Obtenido de [https://es.wikipedia.org/wiki/Atom\\_\(software\)](https://es.wikipedia.org/wiki/Atom_(software))
- Google. (20 de Octubre de 2010). Obtenido de <https://es.wikipedia.org/wiki/AngularJS>
- Google. (2010). *Angular.io*. Obtenido de [https://es.wikipedia.org/wiki/Angular\\_\(framework\)](https://es.wikipedia.org/wiki/Angular_(framework))
- IBM. (s.f.). *IBM*. Obtenido de [https://www.ibm.com/support/knowledgecenter/es/SSEPEK\\_10.0.0/intro/src/tpc/db2z\\_physicaldatabasesdesign.html](https://www.ibm.com/support/knowledgecenter/es/SSEPEK_10.0.0/intro/src/tpc/db2z_physicaldatabasesdesign.html)
- International Organization for Standardization. (03 de 2011). *ISO*. Obtenido de <https://www.iso.org/standard/35733.html>
- International Organization for Standardization. (09 de 2015). *ISO*.
- Jacobson, I. (1986). Obtenido de [https://es.wikipedia.org/wiki/Caso\\_de\\_uso#:~:text=Los%20diagramas%20de%20casos%20de,de%20uso%20en%20un%20sistema.](https://es.wikipedia.org/wiki/Caso_de_uso#:~:text=Los%20diagramas%20de%20casos%20de,de%20uso%20en%20un%20sistema.)
- jwt.io. (8 de Mayo de 2015). *jwt.io*. Obtenido de <https://jwt.io/introduction>
- Mas, D. (27 de Marzo de 2019). *fhios*. Obtenido de <https://www.fhios.es/metodologia-kanban-pros-y-contras/>
- Netscape Communications, Mozilla Foundation. (1995). Obtenido de <https://es.wikipedia.org/wiki/JavaScript>
- Object Management Group. (1994). Obtenido de [https://es.wikipedia.org/wiki/Lenguaje\\_unificado\\_de\\_modelado](https://es.wikipedia.org/wiki/Lenguaje_unificado_de_modelado)
- Object Management Group. (2005). Obtenido de [https://es.wikipedia.org/wiki/Business\\_Process\\_Model\\_and\\_Notation](https://es.wikipedia.org/wiki/Business_Process_Model_and_Notation)
- Oracle Corporation. (23 de Mayo de 1995). Obtenido de <https://es.wikipedia.org/wiki/MySQL>
- Oracle Corporation. (1996). <https://www.java.com/es/>. Obtenido de [https://es.wikipedia.org/wiki/Java\\_\(lenguaje\\_de\\_programaci3n\)](https://es.wikipedia.org/wiki/Java_(lenguaje_de_programaci3n))
- Otto, M., & Thornton, J. (19 de Agosto de 2011). Obtenido de <https://rockcontent.com/es/blog/bootstrap/>
- Pivotal Software. (1 de Octubre de 2002). *spring.io*. Obtenido de <https://spring.io/why-spring>
- PuntoTech. (s.f.). *Medium*. Obtenido de <https://medium.com/puntotech/modelo-vista-presentador-con-angular-f30c4b4f99f0>

- Ribas, E. (29 de Mayo de 2018). <https://juanda.gitbooks.io>. Obtenido de IEBS: <https://www.iebschool.com/blog/que-es-api-rest-integrar-negocio-business-tech/>
- Robards, T. (9 de Noviembre de 2019). Obtenido de <http://developingspanish.com/2019/11/09/fundamentos-javascript-introduccion-a-las-api-rest/>
- SEI. (s.f.). Obtenido de <https://resources.sei.cmu.edu/library/asset-view.cfm?assetid=8443>
- SEI. (Diciembre de 2017). <https://www.sei.cmu.edu>. Obtenido de [https://www.sei.cmu.edu/research-capabilities/all-work/display.cfm?customel\\_datapageid\\_4050=21328](https://www.sei.cmu.edu/research-capabilities/all-work/display.cfm?customel_datapageid_4050=21328)
- Sommerville, I. (2020). *Engineering Software Products*.
- Wikipedia. (s.f.). *Wikipedia*. Obtenido de <https://es.wikipedia.org/wiki/Modelo-vista-presentador>

## Anexo A: Detalle de casos de uso

<b>Nombre de caso de uso</b>	Editar Producto		
<b>Identificador</b>	CU02	<b>Requisito Funcional</b>	RF02
<b>Descripción</b>	Permite al administrador editar productos al sistema.		
<b>Actores involucrados</b>	Administrador		
<b>Pre-Condiciones</b>	Haber iniciado sesión. Debe existir al menos una categoría. Debe existir el producto que se desea modificar.		
<b>Flujo Básico</b>	<ol style="list-style-type: none"> <li>1. El cliente debe ingresar al listado de productos.</li> <li>2. Posteriormente debe presionar en el botón "Ver más"</li> <li>3. El sistema redirecciona al detalle del producto y debe seleccionar el botón "Editar Producto".</li> <li>4. El sistema despliega un formulario con los datos del producto cargados.</li> <li>5. El cliente modifica los datos del producto                         <ol style="list-style-type: none"> <li>5.1. Si los datos del producto son correctos, el sistema permite modificar el producto.</li> </ol> </li> <li>6. El sistema muestra el producto con los datos editados.</li> </ol>		
<b>Flujo Alternativo</b>	<ol style="list-style-type: none"> <li>5a. Si los datos del producto no son correctos, el sistema no permite modificar el producto.                         <ol style="list-style-type: none"> <li>5a.1. Vuelve al punto 5</li> </ol> </li> </ol>		
<b>Post-Condiciones</b>	El sistema muestra el producto modificado.		

Tabla 29: [CU] Editar Producto

<b>Nombre de caso de uso</b>	Eliminar Producto		
<b>Identificador</b>	CU03	<b>Requisito Funcional</b>	RF03
<b>Descripción</b>	Permite al administrador eliminar productos al sistema.		
<b>Actores involucrados</b>	Administrador		
<b>Pre-Condiciones</b>	Haber iniciado sesión. Debe existir al menos una categoría. Debe existir el producto que se desea eliminar.		
<b>Flujo Básico</b>	<ol style="list-style-type: none"> <li>1. El cliente debe ingresar al listado de productos.</li> <li>2. Posteriormente debe presionar en el botón "Ver más"</li> <li>3. El sistema redirecciona al detalle del producto y debe seleccionar el botón "Eliminar Producto"</li> <li>4. Luego el sistema despliega una alerta verificando si realmente desea eliminar el producto.                         <ol style="list-style-type: none"> <li>4.1. Si el cliente acepta la eliminación del producto el producto es eliminado.</li> </ol> </li> <li>5. El sistema redirecciona al listado de productos.</li> </ol>		
<b>Flujo Alternativo</b>	<ol style="list-style-type: none"> <li>4a. Si el cliente cancela la eliminación del producto, el producto no es eliminado                         <ol style="list-style-type: none"> <li>4a.1. Vuelve al punto 3.</li> </ol> </li> </ol>		
<b>Post-Condiciones</b>	El producto es retirado del sistema. El listado de productos es actualizado.		

Tabla 30: [CU] Eliminar Producto

<b>Nombre de caso de uso</b>	Ver Productos		
<b>Identificador</b>	CU04	<b>Requisito Funcional</b>	RF04
<b>Descripción</b>	Permite al administrador y al cliente ver los productos al sistema.		
<b>Actores involucrados</b>	Administrador Cliente		
<b>Pre-Condiciones</b>	Deben existir categorías y productos		
<b>Flujo Básico</b>	<ol style="list-style-type: none"> <li>1. Para hacer uso de esta función se debe ingresar al listado de "productos"</li> <li>2. El sistema muestra todos los productos y las categorías correspondientes.</li> </ol>		
<b>Flujo Alternativo</b>	No posee.		
<b>Post-Condiciones</b>	No requiere.		

Tabla 31: [CU] Ver Productos

<b>Nombre de caso de uso</b>	Crear Cuenta de Usuario		
<b>Identificador</b>	CU08	<b>Requisito Funcional</b>	RF21
<b>Descripción</b>	Permite a un cliente registrar una cuenta en el sistema		
<b>Actores involucrados</b>	Cliente		
<b>Pre-Condiciones</b>	No haber iniciado sesión. No tener una cuenta registrada.		
<b>Flujo Básico</b>	<ol style="list-style-type: none"> <li>1. El cliente solicita acceso al sistema.</li> <li>2. El sistema redirige al registro de usuario.</li> <li>3. El sistema despliega un formulario para el registro.</li> <li>4. El cliente llena el formulario con los datos de acceso al sistema.</li> <li>5. El sistema valida los datos de acceso.                         <ol style="list-style-type: none"> <li>5.1. Si los datos son válidos, el sistema permite crear la cuenta.</li> </ol> </li> <li>6. El sistema despliega un mensaje notificando que se registró efectivamente.</li> </ol>		
<b>Flujo Alternativo</b>	<ol style="list-style-type: none"> <li>5a. Si los datos del usuario no son válidos, el sistema deshabilita el botón de registro.                         <ol style="list-style-type: none"> <li>5a.1. Vuelve al punto 4.</li> </ol> </li> </ol>		
<b>Post-Condiciones</b>	El sistema habilita las credenciales del nuevo usuario en el sistema.		

Tabla 32: [CU] Crear cuenta de usuario

<b>Nombre de caso de uso</b>	Editar Cuenta de Usuario		
<b>Identificador</b>	CU09	<b>Requisito Funcional</b>	RF22
<b>Descripción</b>	Permite a un usuario registrar una cuenta en el sistema		
<b>Actores involucrados</b>	Cliente		
<b>Pre-Condiciones</b>	Haber iniciado sesión. Tener una cuenta registrada.		
<b>Flujo Básico</b>	<ol style="list-style-type: none"> <li>1. El cliente accede al su perfil de usuario.</li> <li>2. El cliente solicita modificar los datos de su cuenta.</li> <li>3. El sistema despliega un formulario con los datos del cliente ya cargados.</li> <li>4. El cliente modifica los datos de la cuenta.</li> <li>5. El sistema verifica los datos.</li> <li>6. Si los datos son correctos             <ol style="list-style-type: none"> <li>6.1. Permite modificar la cuenta.</li> </ol> </li> <li>7. El cliente envía el formulario</li> <li>8. El sistema despliega un mensaje informando que los datos fueron modificados con éxito.</li> </ol>		
<b>Flujo Alternativo</b>	<ol style="list-style-type: none"> <li>6a. Si los datos no son correctos, el sistema no permite modificar los datos.             <ol style="list-style-type: none"> <li>6a.1. Vuelve al punto 3</li> </ol> </li> </ol>		
<b>Post-Condiciones</b>	El sistema actualiza los datos del usuario en el sistema.		

Tabla 33: [CU] Editar cuenta de usuario

<b>Nombre de caso de uso</b>	Agregar Servicio		
<b>Identificador</b>	CU11	<b>Requisito Funcional</b>	RF24
<b>Descripción</b>	Permite al administrador agregar un servicio.		
<b>Actores involucrados</b>	Administrador		
<b>Pre-Condiciones</b>	Haber iniciado sesión.		
<b>Flujo Básico</b>	<ol style="list-style-type: none"> <li>1. El Administrador accede al listado de los servicios.</li> <li>2. El sistema despliega el listado de los servicios.</li> <li>3. El Administrador solicita agregar un nuevo servicio.</li> <li>4. El sistema despliega un formulario para agregar un servicio.</li> <li>5. El Administrador llena el formulario.</li> <li>6. El sistema verifica los datos.</li> <li>7. Si los datos son correctos             <ol style="list-style-type: none"> <li>7.1. Permite agregar el servicio.</li> </ol> </li> <li>8. El Administrador envía el formulario</li> <li>9. El sistema despliega un mensaje informando que el servicio fue agregado con éxito.</li> </ol>		
<b>Flujo Alternativo</b>	<ol style="list-style-type: none"> <li>7a. Si los datos no son correctos, el sistema no permite agregar el servicio.             <ol style="list-style-type: none"> <li>7a.1. Vuelve al punto 5</li> </ol> </li> </ol>		
<b>Post-Condiciones</b>	El sistema tiene un nuevo servicio que presentar.		

Tabla 34: [CU] Agregar Servicio

<b>Nombre de caso de uso</b>	Modificar Servicio		
<b>Identificador</b>	CU12	<b>Requisito Funcional</b>	RF25
<b>Descripción</b>	Permite al administrador modificar un servicio.		
<b>Actores involucrados</b>	Administrador		
<b>Pre-Condiciones</b>	Haber iniciado sesión. Existir al menos un servicio.		
<b>Flujo Básico</b>	<ol style="list-style-type: none"> <li>1. El Administrador accede al listado de los servicios.</li> <li>2. El sistema despliega el listado de los servicios.</li> <li>3. El Administrador solicita modificar un servicio.</li> <li>4. El sistema despliega un formulario con los datos del servicio ya cargados.</li> <li>5. El Administrador modifica los datos del servicio.</li> <li>6. El sistema verifica los datos.</li> <li>7. Si los datos son correctos                         <ol style="list-style-type: none"> <li>7.1. Permite modificar el servicio.</li> </ol> </li> <li>8. El Administrador envía el formulario</li> <li>9. El sistema despliega un mensaje informando que el servicio fue modificado con éxito.</li> </ol>		
<b>Flujo Alternativo</b>	<ol style="list-style-type: none"> <li>7a. Si los datos no son correctos, el sistema no permite modificar el servicio.                         <ol style="list-style-type: none"> <li>7a.1. Vuelve al punto 5</li> </ol> </li> </ol>		
<b>Post-Condiciones</b>	El sistema presenta el servicio modificado.		

*Tabla 35: [CU] Modificar Servicio*

<b>Nombre de caso de uso</b>	Eliminar Servicio		
<b>Identificador</b>	CU13	<b>Requisito Funcional</b>	RF26
<b>Descripción</b>	Permite al administrador eliminar un servicio.		
<b>Actores involucrados</b>	Administrador		
<b>Pre-Condiciones</b>	Haber iniciado sesión.		
<b>Flujo Básico</b>	<ol style="list-style-type: none"> <li>1. El Administrador accede al listado de los servicios.</li> <li>2. El sistema despliega el listado de los servicios.</li> <li>3. El Administrador solicita eliminar un servicio.</li> <li>4. El sistema despliega un mensaje de confirmación para la eliminación.</li> <li>5. Si el administrador confirma la eliminación.                         <ol style="list-style-type: none"> <li>5.1. El sistema despliega un mensaje confirmando la eliminación el servicio.</li> </ol> </li> <li>6. El sistema redirige al listado de servicios.</li> </ol>		
<b>Flujo Alternativo</b>	<ol style="list-style-type: none"> <li>5a. Si el administrador cancela la eliminación, el sistema no permite eliminar el servicio.                         <ol style="list-style-type: none"> <li>5a.1. Vuelve al punto 3</li> </ol> </li> </ol>		
<b>Post-Condiciones</b>	El sistema elimina el servicio del sistema.		

*Tabla 36: [CU] Eliminar Servicio*

<b>Nombre de caso de uso</b>	Ver Servicio		
<b>Identificador</b>	CU14	<b>Requisito Funcional</b>	RF27
<b>Descripción</b>	Permite a cualquier usuario mostrar los servicios.		
<b>Actores involucrados</b>	Cliente y Administrador		
<b>Pre-Condiciones</b>	Debe existir un servicio.		
<b>Flujo Básico</b>	<ol style="list-style-type: none"> <li>1. El usuario accede al listado de los servicios.</li> <li>2. El sistema despliega el listado de los servicios.</li> <li>3. El administrador puede visualizar los servicios.</li> </ol>		
<b>Flujo Alternativo</b>	No posee.		
<b>Post-Condiciones</b>	No posee.		

*Tabla 37: [CU] Ver Servicio*



## Anexo B: Pruebas de Unidad

Pruebas del Sistema Gestión de Productos.				
<b>ID Prueba</b>	P01	<b>Fecha</b>	18-10-2020	
<b>Requisito Funcional</b>	Agregar Categoría			
<b>Actor</b>	Administrador			
<b>Objetivo</b>	Verificar que se pueda crear una categoría			
<b>Tipo de Prueba</b>	Caja Negra			
<b>Pre-Condición</b>	<ul style="list-style-type: none"> <li>• El actor debe haber iniciado sesión</li> </ul>			
<b>Flujo principal</b>	<ol style="list-style-type: none"> <li>1. El actor presiona el botón de “productos”.</li> <li>2. El sistema muestra el listado de productos y las categorías.</li> <li>3. El actor presiona el botón “Agregar Categoría”</li> <li>4. El sistema despliega un formulario para agregar la nueva categoría.</li> <li>5. El actor completa el formulario y presiona “Agregar categoría”                             <ol style="list-style-type: none"> <li>5.1. El sistema valida que los datos ingresados son correctos.                                     <ol style="list-style-type: none"> <li>5.1.1. Si los datos son correctos, el sistema muestra el mensaje “Categoría creada con éxito”</li> </ol> </li> </ol> </li> <li>6. El sistema actualiza el listado de las categorías y muestra la nueva categoría ingresada previamente en el listado.</li> </ol>			
<b>Flujo alternativo</b>	<ol style="list-style-type: none"> <li>1. Si los datos contienen campos vacíos o son inválidos, el sistema deshabilitará el botón “Agregar categoría” y vuelve al punto 4 del flujo principal.</li> <li>2. El actor presiona el botón “Cancelar” y vuelve al listado de productos.</li> </ol>			
<b>Prueba</b>	Datos de Entrada	Resultado Esperado	Resultado Obtenido	Evaluación
<b>Datos válidos</b>	Agrega el nombre de una categoría con un largo mayor a 4 caracteres.	El sistema verifica el formulario y despliega un mensaje “Categoría creada con éxito”.	“Categoría creada con éxito”	Éxito
<b>Datos no válidos</b>	Agregar un nombre de la categoría con el campo en blanco.	El sistema verifica el formulario y deshabilita el botón de para agregar categoría.	“El botón se deshabilita y bloquea el envío del formulario”	Éxito

Tabla 38: [PS] Agregar Categoría

<b>Pruebas del Sistema Gestión de Productos</b>				
<b>ID Prueba</b>	P02	<b>Fecha</b>	18-10-2020	
<b>Requisito Funcional</b>	Agregar Producto			
<b>Actor</b>	Administrador			
<b>Objetivo</b>	Verificar que se pueda crear un producto			
<b>Tipo de Prueba</b>	Caja Negra			
<b>Pre-Condición</b>	<ul style="list-style-type: none"> <li>• El actor debe iniciar sesión</li> <li>• Debe haber al menos una categoría creada en el sistema</li> </ul>			
<b>Flujo principal</b>	<ol style="list-style-type: none"> <li>1. El actor presiona el botón “productos”.</li> <li>2. El sistema muestra el listado de productos y las categorías.</li> <li>3. El actor presiona el botón “Agregar producto”.</li> <li>4. El sistema despliega un formulario para agregar el nuevo producto.</li> <li>5. El actor completa el formulario y presiona “Agregar producto”.                             <ol style="list-style-type: none"> <li>5.1. El sistema valida que los datos ingresados son correctos.                                     <ol style="list-style-type: none"> <li>5.1.1. Si los datos son correctos, el sistema muestra el mensaje “Producto creado con éxito”.</li> </ol> </li> </ol> </li> <li>6. Vuelve al listado de productos.</li> </ol>			
<b>Flujo alternativo</b>	<ol style="list-style-type: none"> <li>1. Si los datos contienen campos vacíos o son inválidos, el sistema deshabilitará el botón “Agregar producto” y vuelve al punto 4 del flujo principal.</li> <li>2. El actor presiona el botón “Cancelar” y vuelve al listado de productos.</li> </ol>			
<b>Prueba</b>	Datos de Entrada	Resultado Esperado	Resultado Obtenido	Evaluación
<b>Datos válidos</b>	Se agrega un producto con su nombre, cantidad, valor unitario, descripción y la categoría a la que pertenece.	El sistema verifica que los datos ingresados son correctos y despliega el mensaje “Producto creado con éxito”.	“Se ha ingresado el producto exitosamente”	Éxito
<b>Datos no válidos</b>	Se agrega un producto con el nombre vacío, cantidad de productos negativo, valor unitario y sin la categoría a la que pertenece	El sistema verifica el formulario y bloquea el botón de envío de “Agregar producto”	“El botón de Agregar producto es bloqueado y no permite enviar el formulario”	Éxito

Tabla 39: [PS] Agregar Producto

<b>Pruebas del Sistema Gestión de Productos.</b>				
<b>ID Prueba</b>	P03	<b>Fecha</b>	18-10-2020	
<b>Requisito Funcional</b>	Editar Producto			
<b>Actor</b>	Administrador			
<b>Objetivo</b>	Verificar que se pueda editar un producto			
<b>Tipo de Prueba</b>	Caja Negra			
<b>Pre-Condición</b>	<ul style="list-style-type: none"> <li>• El actor debe iniciar sesión</li> <li>• Debe haber al menos una categoría creada en el sistema</li> </ul>			
<b>Flujo principal</b>	<ol style="list-style-type: none"> <li>1. El actor presiona el botón de "productos".</li> <li>2. El sistema muestra el listado de productos y las categorías.</li> <li>3. El actor presiona el botón "ver más".</li> <li>4. El sistema despliega el detalle del producto con sus imágenes.</li> <li>5. El actor presiona el botón "Editar producto".</li> <li>6. El sistema despliega el formulario del producto con los datos ya cargados.</li> <li>7. El actor modifica los datos el formulario y presiona "Editar producto".                             <ol style="list-style-type: none"> <li>7.1. El sistema valida que los datos ingresados son correctos.                                     <ol style="list-style-type: none"> <li>7.1.1. Si los datos son correctos, el sistema muestra el mensaje "Se ha modificado el producto exitosamente".</li> </ol> </li> </ol> </li> <li>8. Vuelve al listado de productos mostrando el producto modificado.</li> </ol>			
<b>Flujo alternativo</b>	<ol style="list-style-type: none"> <li>1. Si los datos contienen campos vacíos o son inválidos, el sistema deshabilitará el botón "Editar producto" y vuelve al punto 4 del flujo principal.</li> <li>2. El actor presiona el botón "Cancelar" y vuelve al listado de productos.</li> </ol>			
<b>Prueba</b>	Datos de Entrada	Resultado Esperado	Resultado Obtenido	Evaluación
<b>Datos válidos</b>	Se modifica un producto con su nombre nuevo, cantidad de stock nuevos, valor unitario, descripción y la categoría a la que pertenece.	El sistema verifica que los datos ingresados son correctos y despliega el mensaje "Se ha modificado el producto exitosamente"	"Se ha modificado el producto exitosamente"	Éxito
<b>Datos no válidos</b>	Se modifica un producto con el nombre vacío, cantidad de productos negativo, valor unitario y sin la categoría a la que pertenece	El sistema verifica el formulario y muestra un mensaje de error y deshabilita el botón.	"El botón de editar producto es bloqueado y no permite enviar el formulario"	Éxito

Tabla 40: [PS] Editar Producto

<b>Pruebas del Sistema Gestión de Productos.</b>				
<b>ID Prueba</b>	P04	<b>Fecha</b>	18-10-2020	
<b>Requisito Funcional</b>	Eliminar Producto			
<b>Actor</b>	Administrador			
<b>Objetivo</b>	Verificar que se pueda eliminar un producto			
<b>Tipo de Prueba</b>	Caja Negra			
<b>Pre-Condición</b>	El actor debe iniciar sesión Debe existir el producto a eliminar.			
<b>Flujo principal</b>	<ol style="list-style-type: none"> <li>1. El actor presiona el botón de "productos".</li> <li>2. El sistema muestra el listado de productos y las categorías.</li> <li>3. El actor presiona el botón "ver más".</li> <li>4. El sistema despliega el detalle del producto con sus imágenes.</li> <li>5. El actor presiona el botón "Eliminar producto", el cual envía el ID del producto.</li> <li>6. El sistema despliega un mensaje consultando si desea eliminar el producto.</li> <li>6.1. Si el actor presiona aceptar se elimina el producto.</li> <li>7. Vuelve al listado de productos.</li> </ol>			
<b>Flujo alternativo</b>	1. El actor presiona el botón "Cancelar" y vuelve al detalle del producto.			
<b>Prueba</b>	Datos de Entrada	Resultado Esperado	Resultado Obtenido	Evaluación
<b>Datos válidos</b>	Se envía el ID:1 del producto que se desea eliminar	Verifica si existe el ID del producto y lo elimina, mostrando un mensaje "Se ha eliminado exitosamente el producto"	"No se pudo eliminar el producto"	Fallo
<b>Datos no válidos</b>	Se envía el ID: 100000 el cual no tiene producto asignado	El sistema busca el ID enviado y despliega un mensaje de error, producto no encontrado	"Producto no encontrado"	Fallo

Tabla 41: [PS] Eliminar Producto

<b>Pruebas del Sistema Gestión de Carrito de Compras.</b>				
<b>ID Prueba</b>	P05	<b>Fecha</b>	03-11-2020	
<b>Requisito Funcional</b>	Agregar producto al carrito			
<b>Actor</b>	Cliente			
<b>Objetivo</b>	Verificar que se pueda agregar un producto al carrito de compras			
<b>Tipo de Prueba</b>	Caja Negra			
<b>Pre-Condición</b>	<ul style="list-style-type: none"> <li>El actor debe haber iniciado sesión</li> </ul>			
<b>Flujo principal</b>	<ol style="list-style-type: none"> <li>El actor presiona el botón de “productos”.</li> <li>El sistema muestra el listado de productos y las categorías.</li> <li>El actor presiona el botón “Agregar al carrito” en un producto.</li> <li>El sistema despliega el mensaje “Producto agregado con éxito al carrito”.</li> <li>El sistema deja al actor en el listado de productos para que pueda agregar más productos al carrito.</li> </ol>			
<b>Flujo alternativo</b>	<ol style="list-style-type: none"> <li>El actor accede en el apartado de “ver más” del producto.</li> <li>El sistema lo redirige al detalle del producto.</li> <li>El actor puede seleccionar la cantidad de productos que desea.</li> <li>El actor presiona agregar producto.</li> <li>El sistema despliega el mensaje “Producto agregado con éxito al carrito”.</li> <li>El sistema deja al actor en el listado de productos para que pueda agregar más productos al carrito.</li> </ol>			
<b>Prueba</b>	Datos de Entrada	Resultado Esperado	Resultado Obtenido	Evaluación
<b>Datos válidos</b>	Seleccionar un producto que se desea agregar al carrito.	El sistema verifica que exista el producto y la cantidad en stock del producto. El sistema despliega el mensaje “Producto agregado con éxito al carrito”.	“Producto agregado con éxito al carrito”.	Éxito
<b>Datos no válidos</b>	Agregar un producto que no tiene stock en el sistema.	El sistema verifica la cantidad en stock del producto y le envía el mensaje “No se puede agregar el producto al carrito”	“No se puede agregar el producto al carrito”	Éxito

Tabla 42: [PS] Agregar producto al carrito de compra.

<b>Pruebas del Sistema Gestión de Carrito de Compras.</b>				
<b>ID Prueba</b>	P06	<b>Fecha</b>	03-11-2020	
<b>Requisito Funcional</b>	Listar carrito de compras			
<b>Actor</b>	Cliente			
<b>Objetivo</b>	Mostrar el carrito con los productos agregados			
<b>Tipo de Prueba</b>	Caja Negra			
<b>Pre-Condición</b>	<ul style="list-style-type: none"> <li>• El actor debe haber iniciado sesión</li> <li>• El actor debe haber agregado un producto al carrito</li> </ul>			
<b>Flujo principal</b>	<ol style="list-style-type: none"> <li>1. El actor presiona el botón de "Carrito".</li> <li>2. El sistema muestra el listado de productos que se encuentran en el carrito de compra.</li> <li>3. El actor puede visualizar todos los productos que agrego al carrito de compra, con el valor unitario, la cantidad agregada, el valor total del carrito.</li> </ol>			
<b>Flujo alternativo</b>	1. No posee.			
<b>Prueba</b>	Datos de Entrada	Resultado Esperado	Resultado Obtenido	Evaluación
<b>Datos válidos</b>	Selecciona la opción carrito.	El sistema verifica que haya productos en el carrito y despliega el detalle del carrito.	"El carrito con todos los productos agregados"	Éxito
<b>Datos no válidos</b>	Seleccionar el carrito y este se encuentre vacío.	El sistema verificará que el carrito se encuentre vacío y mostrará un carrito vacío.	"Muestra el carrito vacío"	Éxito

Tabla 43: [PS] Listar carrito de compras

<b>Pruebas del Sistema Gestión de Cuentas de Usuario.</b>				
<b>ID Prueba</b>	P07	<b>Fecha</b>	03-11-2020	
<b>Requisito Funcional</b>	Inicio de sesión.			
<b>Actor</b>	Cliente y Administrador			
<b>Objetivo</b>	Iniciar sesión en el sistema.			
<b>Tipo de Prueba</b>	Caja Negra			
<b>Pre-Condición</b>	<ul style="list-style-type: none"> <li>El actor debe tener una cuenta registrada en el sistema.</li> </ul>			
<b>Flujo principal</b>	<ol style="list-style-type: none"> <li>El actor presiona el apartado de "Iniciar Sesión".</li> <li>El sistema despliega el formulario de inicio de sesión.</li> <li>El actor proporciona las credenciales de autenticación del sistema.</li> <li>El sistema corrobora las credenciales del usuario.                             <ol style="list-style-type: none"> <li>Si las credenciales son correctas.                                     <ol style="list-style-type: none"> <li>El sistema despliega el mensaje "Ha ingresado exitosamente al sistema"</li> </ol> </li> </ol> </li> <li>El sistema redirecciona a la página principal y muestra el nombre del usuario que inicio sesión en la esquina superior derecha.</li> </ol>			
<b>Flujo alternativo</b>	1. En caso de que las credenciales son incorrectas, el sistema despliega el mensaje "Usuario o contraseña equivocados" y vuelve al punto 3.			
<b>Prueba</b>	Datos de Entrada	Resultado Esperado	Resultado Obtenido	Evaluación
<b>Datos válidos</b>	Llenar el formulario de inicio de sesión, con las credenciales correctas	El sistema verifica los datos del formulario de Inicio de sesión y despliega el mensaje "Ha ingresado exitosamente al sistema".	"Ha ingresado exitosamente al sistema"	Éxito
<b>Datos no válidos</b>	Llena el formulario de inicio de sesión con credenciales erróneas.	El sistema verifica los datos del formulario de Inicio de sesión y despliega el mensaje "Usuario o contraseña equivocados".	"Usuario o contraseña equivocados"	Éxito

Tabla 44: [PS] Inicio de sesión

<b>Pruebas del Sistema Gestión de Cuentas de Usuario.</b>				
<b>ID Prueba</b>	P08	<b>Fecha</b>	03-11-2020	
<b>Requisito Funcional</b>	Cerrar de sesión.			
<b>Actor</b>	Cliente y Administrador			
<b>Objetivo</b>	Cerrar la sesión del usuario			
<b>Tipo de Prueba</b>	Caja Negra			
<b>Pre-Condición</b>	<ul style="list-style-type: none"> <li>El actor debe haber iniciado sesión.</li> </ul>			
<b>Flujo principal</b>	<ol style="list-style-type: none"> <li>El actor presiona el apartado en la esquina superior derecha que contiene el nombre del usuario.</li> <li>El sistema despliega un menú pequeño con la opción "Perfil" y "Cerrar Sesión".</li> <li>El actor presiona la opción "Cerrar Sesión".</li> <li>El sistema cierra la sesión del usuario y despliega el mensaje "Ha cerrado sesión con éxito"</li> </ol>			
<b>Flujo alternativo</b>	1. No posee.			
<b>Prueba</b>	Datos de Entrada	Resultado Esperado	Resultado Obtenido	Evaluación
<b>Datos válidos</b>	El actor presiona el botón "cerrar sesión"	El sistema cierra la sesión del usuario y despliega el mensaje "Ha cerrado sesión con éxito"	"Ha cerrado sesión con éxito"	Éxito

Tabla 45: [PS] Cierre de sesión

<b>Pruebas del Sistema Gestión de Cuentas de Usuario.</b>				
<b>ID Prueba</b>	P09	<b>Fecha</b>	03-11-2020	
<b>Requisito Funcional</b>	Crear cuenta de usuario.			
<b>Actor</b>	Cliente			
<b>Objetivo</b>	Crear cuenta de usuario			
<b>Tipo de Prueba</b>	Caja Negra			
<b>Pre-Condición</b>	Ninguna.			
<b>Flujo principal</b>	<ol style="list-style-type: none"> <li>El actor presiona el apartado de "Iniciar Sesión".</li> <li>El sistema despliega el formulario de inicio de sesión y bajo el formulario se encuentra la opción "Regístrate aquí".</li> <li>El actor hace clic en "Regístrate aquí".</li> <li>El sistema despliega el formulario de registro de usuario.</li> <li>El actor llena el formulario de registro de usuario.                             <ol style="list-style-type: none"> <li>El sistema verifica los datos del formulario.                                     <ol style="list-style-type: none"> <li>Si no hay errores en el formulario, el sistema despliega el mensaje "Usuario creado con éxito"</li> </ol> </li> </ol> </li> <li>El sistema redirecciona al formulario de inicio de sesión.</li> </ol>			
<b>Flujo alternativo</b>	1. En caso de que haya errores en el formulario, el sistema deshabilitará el botón "Registrar".			
<b>Prueba</b>	Datos de Entrada	Resultado Esperado	Resultado Obtenido	Evaluación
<b>Datos válidos</b>	Llenar el formulario de registro de usuario, con un usuario que no se encuentre registrado.	El sistema verifica los datos del formulario de registro de usuario y despliega el mensaje "Usuario creado con éxito".	"Usuario creado con éxito"	Éxito
<b>Datos no válidos</b>	Llena el formulario de inicio de registro de usuario con un email sin formato y un rut erróneo.	El sistema verifica los datos del formulario de registro de usuario y deshabilita el botón del registro	"El sistema deshabilita el botón del registro"	Éxito

Tabla 46: [PS] Crear cuenta de usuario



<b>Pruebas del Sistema Gestión de Cuentas de Usuario.</b>				
<b>ID Prueba</b>	P10	<b>Fecha</b>	28-11-2020	
<b>Requisito Funcional</b>	Modificar datos de la cuenta de usuario.			
<b>Actor</b>	Cliente y Administrador			
<b>Objetivo</b>	Modificar los datos de la cuenta de usuario.			
<b>Tipo de Prueba</b>	Caja Negra			
<b>Pre-Condición</b>	<ul style="list-style-type: none"> <li>El actor debe tener una cuenta registrada en el sistema.</li> </ul>			
<b>Flujo principal</b>	<ol style="list-style-type: none"> <li>El actor presiona su nombre en el lado superior derecho.</li> <li>El sistema despliega un menú.</li> <li>El actor presiona la opción "Perfil"</li> <li>El sistema redirige al detalle de los datos de la cuenta.</li> <li>El actor presiona el link "Modificar datos"</li> <li>El sistema despliega el formulario con los datos ya cargados del usuario.</li> <li>El actor llena el formulario de modificación de datos del usuario.                             <ol style="list-style-type: none"> <li>El sistema verifica los datos del formulario.                                     <ol style="list-style-type: none"> <li>Si no hay errores en el formulario, el sistema despliega el mensaje "Cuenta actualizada con éxito"</li> </ol> </li> </ol> </li> <li>El sistema redirecciona al detalle de la cuenta de usuario.</li> </ol>			
<b>Flujo alternativo</b>	<ol style="list-style-type: none"> <li>En caso de que haya errores en el formulario, el sistema deshabilitará el botón "Registrar".</li> </ol>			
<b>Prueba</b>	Datos de Entrada	Resultado Esperado	Resultado Obtenido	Evaluación
<b>Datos válidos</b>	Modificar un campo del formulario con datos correctos.	El sistema verifica los datos del formulario de registro de usuario y despliega el mensaje "Cuenta modificada con éxito".	"Cuenta modificada con éxito"	Éxito
<b>Datos no válidos</b>	Borra los datos de un campo del formulario.	El sistema verifica los datos del formulario de registro de usuario y deshabilita el botón de la modificación de datos	"El sistema deshabilita el botón de la modificación de datos"	Éxito

Tabla 47: [PS] Modificar datos de usuario

<b>Pruebas del Sistema Gestión de Compras.</b>				
<b>ID Prueba</b>	P11	<b>Fecha</b>	28-11-2020	
<b>Requisito Funcional</b>	Generar compra			
<b>Actor</b>	Cliente y Administrador			
<b>Objetivo</b>	Generar la compra			
<b>Tipo de Prueba</b>	Caja Negra			
<b>Pre-Condición</b>	<ul style="list-style-type: none"> <li>• El actor debe tener una cuenta registrada en el sistema.</li> <li>• El actor debe tener productos en el carrito de compra.</li> </ul>			
<b>Flujo principal</b>	<ol style="list-style-type: none"> <li>1. El actor presiona el icono del carrito en el lado superior derecho.</li> <li>2. El sistema despliega una lista con todos los productos agregados.</li> <li>3. El sistema despliega un formulario de envío en la parte inferior del carrito.</li> <li>4. El actor llena el formulario de envío de la compra.                             <ol style="list-style-type: none"> <li>4.1. El sistema verifica los datos del formulario.                                     <ol style="list-style-type: none"> <li>4.1.1. Si no hay errores en el formulario, el sistema permite enviar el formulario</li> <li>4.1.2. El actor envía el formulario.</li> </ol> </li> </ol> </li> <li>5. El sistema redirecciona a la página de pago.</li> </ol>			
<b>Flujo alternativo</b>	<ol style="list-style-type: none"> <li>1. En caso de que haya errores en el formulario, el sistema deshabilitará el botón "Realizar Compra".</li> </ol>			
<b>Prueba</b>	Datos de Entrada	Resultado Esperado	Resultado Obtenido	Evaluación
<b>Datos válidos</b>	Llenar el formulario con los datos de entrada correctamente	El sistema verifica los datos del formulario de envío y despliega el mensaje "Se generó la compra exitosamente".	"Se generó la compra exitosamente"	Éxito
<b>Datos no válidos</b>	El formulario tiene el campo de "Ciudad" vacío.	El sistema verifica los datos del formulario de envío y deshabilita el botón de compra.	"El sistema deshabilita el botón de compra."	Éxito

Tabla 48: [PS] Generar compra

<b>Pruebas del Sistema Gestión de Productos.</b>				
<b>ID Prueba</b>	P12	<b>Fecha</b>	28-11-2020	
<b>Requisito Funcional</b>	Listar Producto			
<b>Actor</b>	Cliente			
<b>Objetivo</b>	Mostrar el carrito con los productos agregados			
<b>Tipo de Prueba</b>	Caja Negra			
<b>Pre-Condición</b>	Ninguna.			
<b>Flujo principal</b>	<ol style="list-style-type: none"> <li>1. El actor presiona el botón de "Producto".</li> <li>2. El sistema muestra el listado de productos.</li> </ol>			
<b>Flujo alternativo</b>	<ol style="list-style-type: none"> <li>1. No posee.</li> </ol>			
<b>Prueba</b>	Datos de Entrada	Resultado Esperado	Resultado Obtenido	Evaluación
<b>Datos válidos</b>	Selecciona la opción "Productos".	El sistema verifica que haya productos en el sistema y muestra todos los productos.	"El listado de los productos"	Éxito
<b>Datos no válidos</b>	Selecciona la opción "Productos".	El sistema verifica que haya productos en el sistema y si no se encuentra ningún producto, muestra el listado vacío.	"Muestra el listado de productos vacío"	Éxito

Tabla 49: [PS] Mostrar Productos

<b>Pruebas del Sistema Gestión de Compras.</b>				
<b>ID Prueba</b>	P13	<b>Fecha</b>	28-11-2020	
<b>Requisito Funcional</b>	Ver Compras			
<b>Actor</b>	Cliente y Administrador			
<b>Objetivo</b>	Mostrar el historial de las compras			
<b>Tipo de Prueba</b>	Caja Negra			
<b>Pre-Condición</b>	<ul style="list-style-type: none"> <li>• Debe haber al menos una compra en el sistema.</li> <li>• Para el cliente debe tener una compra registrada.</li> <li>• Haber iniciado sesión.</li> <li>• Estar en el detalle de la cuenta de usuario.</li> </ul>			
<b>Flujo principal</b>	<ol style="list-style-type: none"> <li>1. El sistema despliega la última compra realizada.</li> <li>2. El actor puede ver los detalles de la compra, incluyendo los productos realizados.</li> <li>3. El actor debe presionar el botón "Mis Compras" al costado izquierdo.</li> <li>4. El Sistema despliega todas las compras correspondientes a ese actor.</li> </ol>			
<b>Flujo alternativo</b>	<ol style="list-style-type: none"> <li>1. El Administrador presiona el botón "Compras Clientes"</li> <li>2. El sistema despliega todas las compras realizadas en el sistema.</li> </ol>			
<b>Prueba</b>	Datos de Entrada	Resultado Esperado	Resultado Obtenido	Evaluación
<b>Datos válidos</b>	Un Cliente selecciona la opción "Mis Compras".	El sistema verifica que haya productos en el sistema y muestra todas las compras realizadas por él.	"El listado de las compras realizadas"	Éxito

Tabla 50: [PS] Ver Compras

<b>Pruebas del Sistema Gestión de Compras.</b>				
<b>ID Prueba</b>	P14	<b>Fecha</b>	03-11-2020	
<b>Requisito Funcional</b>	Modificar compras			
<b>Actor</b>	Administrador			
<b>Objetivo</b>	Modificar el estado de la compra			
<b>Tipo de Prueba</b>	Caja Negra			
<b>Pre-Condición</b>	<ul style="list-style-type: none"> <li>• Debe haber al menos una compra en el sistema.</li> <li>• Haber iniciado sesión.</li> <li>• Estar en el detalle de la cuenta de usuario.</li> </ul>			
<b>Flujo principal</b>	<ol style="list-style-type: none"> <li>1. El actor debe presionar el botón "Compras Clientes" al costado izquierdo.</li> <li>2. El sistema despliega el historial de las compras.</li> <li>3. El actor presiona el botón "Detalle de la compra"</li> <li>4. El sistema despliega el detalle de la compra y un campo para cambiar el estado de la compra.</li> <li>5. El actor selecciona el estado en que se encuentre la compra y presiona el botón "Editar Compra".</li> <li>6. El sistema despliega el mensaje "Compra actualizada con éxito".</li> </ol>			
<b>Flujo alternativo</b>	1. No posee			
<b>Prueba</b>	Datos de Entrada	Resultado Esperado	Resultado Obtenido	Evaluación
<b>Datos válidos</b>	El administrador selecciona la opción "pagado"	El sistema despliega el mensaje "Compra actualizada con éxito"	"Compra actualizada con éxito"	Éxito

Tabla 51: [PS] Modificar Compra

<b>Pruebas del Sistema Gestión de Servicios.</b>				
<b>ID Prueba</b>	P15	<b>Fecha</b>	28-11-2020	
<b>Requisito Funcional</b>	Agregar Servicio.			
<b>Actor</b>	Administrador			
<b>Objetivo</b>	Agregar un Servicio al sistema.			
<b>Tipo de Prueba</b>	Caja Negra			
<b>Pre-Condición</b>	<ul style="list-style-type: none"> <li>• Haber iniciado sesión.</li> </ul>			
<b>Flujo principal</b>	<ol style="list-style-type: none"> <li>1. El actor presiona el botón "Servicios".</li> <li>2. El sistema muestra el listado de servicios.</li> <li>3. El actor presiona el botón "Agregar Servicio".</li> <li>4. El sistema despliega un formulario para agregar el nuevo Servicio.</li> <li>5. El actor completa el formulario y presiona "Agregar Servicio".               <ol style="list-style-type: none"> <li>5.1. El sistema valida que los datos ingresados son correctos.                   <ol style="list-style-type: none"> <li>5.1.1. Si los datos son correctos, el sistema muestra el mensaje "Servicio creado con éxito".</li> </ol> </li> </ol> </li> <li>6. Vuelve al listado de Servicios.</li> </ol>			
<b>Flujo alternativo</b>	2. No posee			
<b>Prueba</b>	Datos de Entrada	Resultado Esperado	Resultado Obtenido	Evaluación
<b>Datos válidos</b>	Se agrega un servicio con su nombre, descripción.	El sistema despliega el mensaje "Servicio agregado con éxito"	"Servicio agregado con éxito"	Éxito
<b>Datos no válidos</b>	Se agrega un servicio con el campo del nombre vacío.	El sistema deshabilita el botón de "Agregar Servicio"	"El sistema deshabilita el botón de Agregar Servicio"	Éxito

Tabla 52: [PS] Agregar Servicio

<b>Pruebas del Sistema Gestión de Servicios.</b>				
<b>ID Prueba</b>	P16	<b>Fecha</b>	28-11-2020	
<b>Requisito Funcional</b>	Listar Servicios			
<b>Actor</b>	Cliente			
<b>Objetivo</b>	Mostrar los servicios que provee la empresa.			
<b>Tipo de Prueba</b>	Caja Negra			
<b>Pre-Condición</b>	Ninguna.			
<b>Flujo principal</b>	1. El actor presiona el botón de "Servicios". 2. El sistema muestra el listado de Servicios.			
<b>Flujo alternativo</b>	2. No posee.			
<b>Prueba</b>	Datos de Entrada	Resultado Esperado	Resultado Obtenido	Evaluación
<b>Datos válidos</b>	Selecciona la opción "Servicios".	El sistema verifica que haya Servicios en el sistema y muestra todos los productos.	"El listado de los Servicios"	Éxito
<b>Datos no válidos</b>	Selecciona la opción "Servicios".	El sistema verifica que haya Servicios en el sistema y si no se encuentra ningún Servicio, muestra el listado vacío.	"Muestra el listado de servicios vacío"	Éxito

*Tabla 53: [PS] Listar Servicios*

<b>Pruebas del Sistema Gestión de Servicios.</b>				
<b>ID Prueba</b>	P17	<b>Fecha</b>	28-11-2020	
<b>Requisito Funcional</b>	Editar Servicio			
<b>Actor</b>	Administrador			
<b>Objetivo</b>	Verificar que se pueda modificar un Servicio			
<b>Tipo de Prueba</b>	Caja Negra			
<b>Pre-Condición</b>	<ul style="list-style-type: none"> <li>• El actor debe iniciar sesión</li> <li>• Debe haber al menos un servicio en el sistema</li> </ul>			
<b>Flujo principal</b>	<ol style="list-style-type: none"> <li>1. El actor presiona el botón de "Servicios".</li> <li>2. El sistema muestra el listado de servicios.</li> <li>3. El actor presiona el botón "Modificar Servicio".</li> <li>4. El sistema despliega el formulario del servicio con los datos ya cargados.</li> <li>5. El actor modifica los datos el formulario y presiona "Editar servicio".                             <ol style="list-style-type: none"> <li>5.1. El sistema valida que los datos ingresados son correctos.                                     <ol style="list-style-type: none"> <li>5.1.1. Si los datos son correctos, el sistema muestra el mensaje "Se ha modificado el servicio exitosamente".</li> </ol> </li> </ol> </li> <li>6. Vuelve al listado de servicios mostrando el servicio modificado.</li> </ol>			
<b>Flujo alternativo</b>	<ol style="list-style-type: none"> <li>4. Si los datos contienen campos vacíos o son inválidos, el sistema deshabilitará el botón "Editar servicios" y vuelve al punto 4 del flujo principal.</li> <li>5. El actor presiona el botón "Cancelar" y vuelve al listado de productos.</li> </ol>			
<b>Prueba</b>	Datos de Entrada	Resultado Esperado	Resultado Obtenido	Evaluación
<b>Datos válidos</b>	Se modifica un servicio con su nombre nuevo, descripción.	El sistema verifica que los datos ingresados son correctos y despliega el mensaje "Se ha modificado el servicio exitosamente"	"Se ha modificado el servicio exitosamente"	Éxito
<b>Datos no válidos</b>	Se modifica un producto con el nombre vacío.	El sistema verifica el formulario y deshabilita el botón.	"El sistema verifica el formulario y deshabilita el botón"	Éxito

Tabla 54: [PS] Modificar Servicio

<b>Pruebas del Sistema Gestión de Servicios.</b>				
<b>ID Prueba</b>	P18	<b>Fecha</b>	28-11-2020	
<b>Requisito Funcional</b>	Eliminar Servicio			
<b>Actor</b>	Administrador			
<b>Objetivo</b>	Verificar que se pueda eliminar un servicio			
<b>Tipo de Prueba</b>	Caja Negra			
<b>Pre-Condición</b>	El actor debe iniciar sesión Debe existir el producto a eliminar.			
<b>Flujo principal</b>	<ol style="list-style-type: none"> <li>1. El actor presiona el botón de "Servicios".</li> <li>2. El sistema muestra el listado de Servicios.</li> <li>3. El actor presiona el botón "Eliminar Servicio", el cual envía el ID del servicio.</li> <li>4. El sistema despliega un mensaje consultando si desea eliminar el servicio.                             <ol style="list-style-type: none"> <li>4.1. Si el actor presiona aceptar se elimina el servicio.</li> </ol> </li> <li>5. Vuelve al listado de servicios.</li> </ol>			
<b>Flujo alternativo</b>	2. El actor presiona el botón "Cancelar" y vuelve al detalle del producto.			
<b>Prueba</b>	Datos de Entrada	Resultado Esperado	Resultado Obtenido	Evaluación
<b>Datos válidos</b>	Se envía el ID:1 del servicio que se desea eliminar	Verifica si existe el ID del servicio y lo elimina, mostrando un mensaje "Se ha eliminado exitosamente el servicio"	"Se ha eliminado exitosamente el servicio"	Fallo
<b>Datos no válidos</b>	Se envía el ID: 100000 el cual no tiene servicio asignado	El sistema busca el ID enviado y despliega un mensaje de error, producto no encontrado	"Servicio no encontrado"	Fallo

Tabla 55: [PS] Eliminar Servicio

## Anexo C: Encuesta de usabilidad

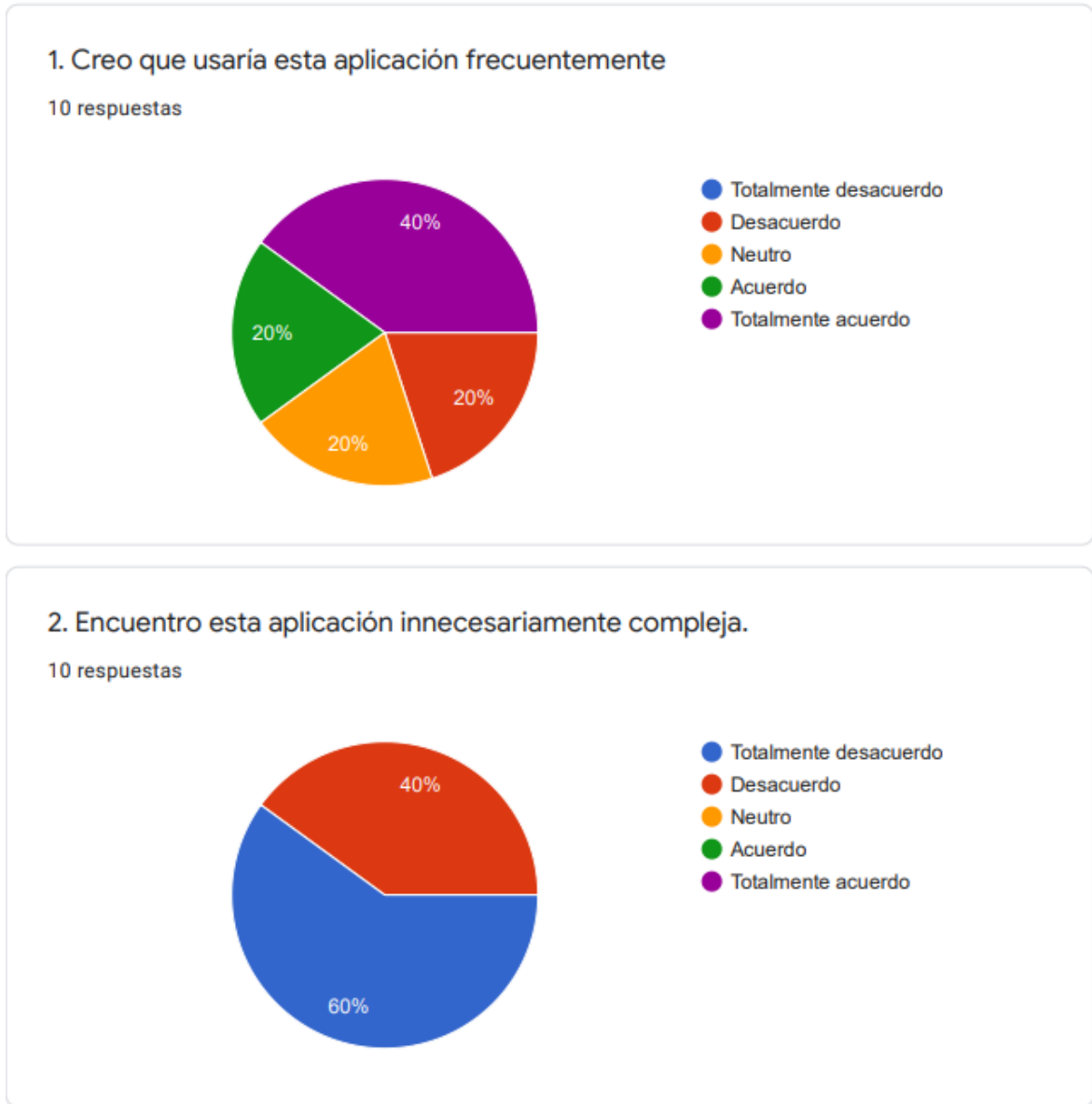
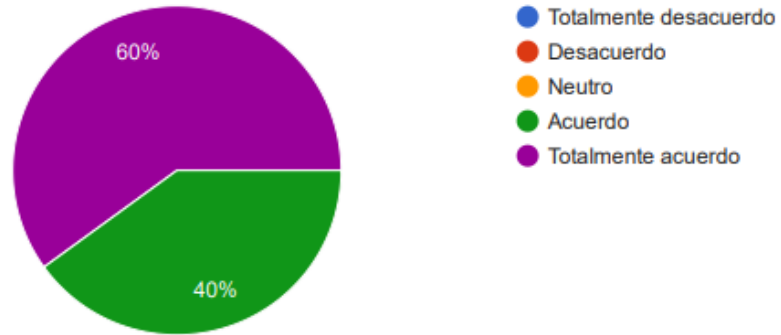


Figura 20: Encuesta de usabilidad P1-P2



### 3. Creo que la aplicación fue fácil de usar.

10 respuestas



### 4. Creo que necesitaría ayuda de una persona con conocimientos técnicos para usar esta aplicación.

10 respuestas

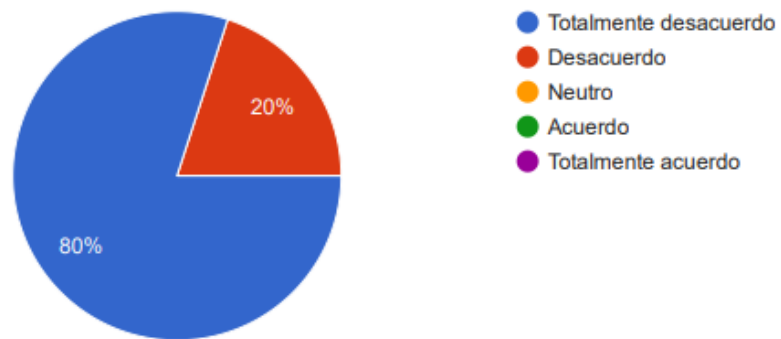
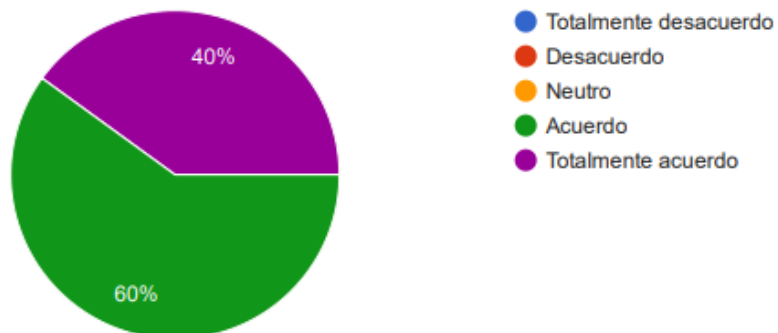


Figura 21: Encuesta de usabilidad P3-P4

5. Las funciones de esta aplicación están bien integradas.

10 respuestas



6. Creo que la aplicación es muy inconsistente.

10 respuestas

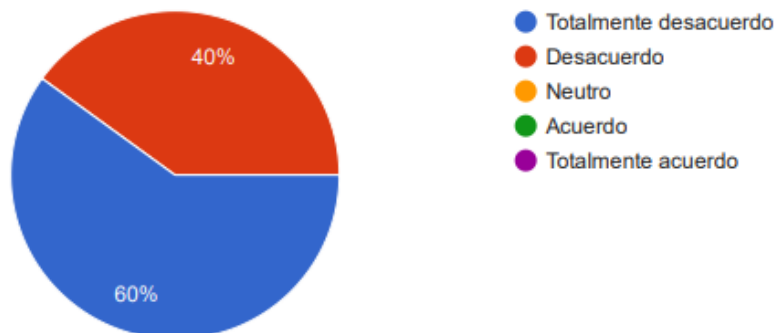
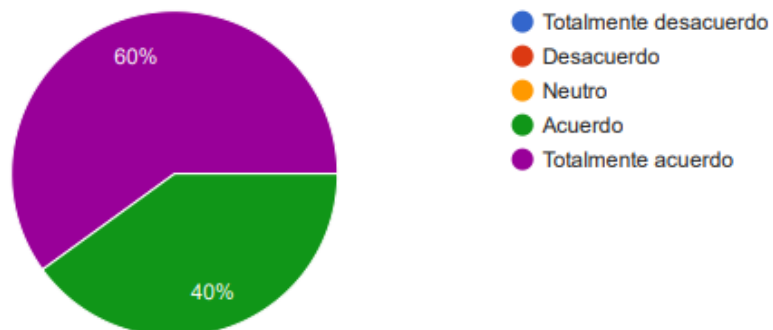


Figura 22: Encuesta de usabilidad P5-P6

7. Imagino que la mayoría de la gente aprendería a usar esta aplicación en forma muy rápida.

10 respuestas



8. Encuentro que la aplicación es muy difícil de usar.

10 respuestas

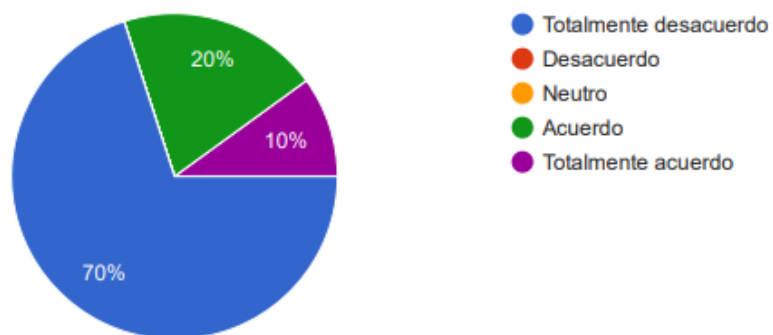
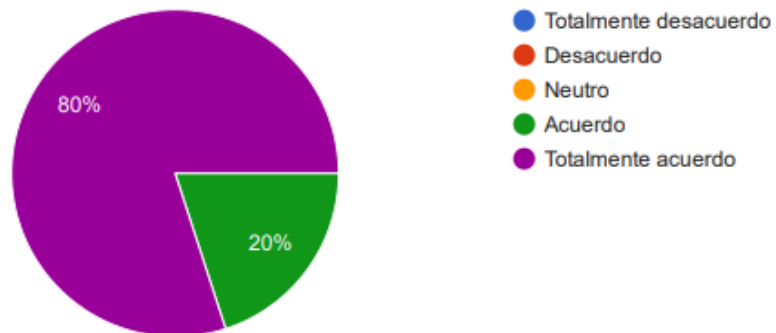


Figura 23: Encuesta de usabilidad P7-P8

9. Me siento confiado al usar esta aplicación.

10 respuestas



10. Necesité aprender muchas cosas antes de ser capaz de usar esta aplicación.

10 respuestas

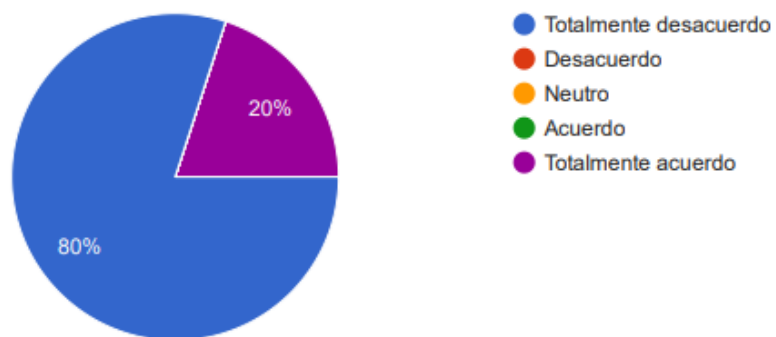


Figura 24: Encuesta de usabilidad P9-P10