

Universidad Hispanoamericana

Escuela de informática

Anteproyecto de bachillerato

Título del proyecto

Potenciando la Presencia Digital del Super Alfolí: Implementación de Estrategias  
de Comercio Electrónico para Maximizar el Alcance y la Eficiencia

Sustentante:

Alejandro Salas Casanova

Marzo, 2024

## Tabla de Contenido

Índice de Tablas.....	6
Índice de Figuras .....	6
1. Capítulo 1: Planteamiento del Tema .....	12
Introducción .....	12
1.1 Antecedentes y Justificación del Anteproyecto .....	12
1.1.1. Marco de Referencia Empresarial y Contextual .....	12
1.1.2 Justificación del Anteproyecto .....	14
1.2 Definición del problema.....	15
1.2.1. Problemática .....	15
1.2.2. Problema General .....	17
1.2.3. Problemas Específicos .....	17
1.3. Objetivo General y Objetivos Específicos .....	18
1.3.1. Objetivo General.....	18
1.3.2 Objetivos específicos.....	19
1.4. Alcance y Limitaciones.....	21
1.4.1 Alcance del Proyecto .....	21
1.4.2 Limitaciones del Proyecto.....	22
1.5 Cronograma del Proyecto .....	22

Tabla 1 Cronograma del Proyecto .....	24
Tabla 2 Diagrama de Gantt.....	27
2. Capítulo 2: Objetivo 1 .....	29
2.1 Definición de conceptos .....	29
2.2 Diseño de la investigación .....	29
2.3 Resultados .....	32
2.3.1 Infraestructura Tecnológica .....	32
2.3.1.1 Hardware.....	32
2.3.1.2 Software .....	32
2.3.1.3 Redes .....	33
2.3.2 Personal y Gestión .....	33
2.3.3 Procesos .....	33
2.3.4 Experiencia del Cliente y Expectativas.....	33
2.4 Análisis y discusión .....	34
3. Capítulo 3: Objetivo 2 .....	35
3.1 Definición de conceptos .....	35
3.2 Desarrollo.....	37
3.2.1 Establecer los requerimientos .....	37
3.2.1.1 Requerimientos Funcionales .....	37
3.2.1.2 Requerimientos No Funcionales .....	38

3.2.2 Diagrama entidad relación.....	39
3.2.2.1 Entidades .....	39
3.2.2.2 Atributos .....	40
3.2.2.3 Relaciones.....	42
3.2.3 Diagrama de flujo .....	45
3.2.4 Diagrama de caso de uso.....	59
3.2.4.1 Actores .....	59
3.2.4.2 Casos de uso .....	59
3.2.4.3 Diagrama.....	60
3.2.5 Tecnologías y estilo .....	74
3.3 Resultados .....	81
3.4 Análisis y discusión .....	81
4. Capítulo 4: Objetivo 3 .....	81
4.1 Preparación y primeros pasos .....	81
4.2 Apartado del cliente .....	87
4.3 Apartado de administrador y empleado.....	109
4.4 Resultados .....	132
5. Capítulo 5: Objetivo 4 Implementación .....	133
5.1 Procedimiento .....	133
6. Capítulo 6: Conclusión .....	139

Recomendaciones .....	143
Realizar pruebas a pequeña escala .....	143
Fortalecimiento de la seguridad del sistema .....	143
Estrategias de difusión .....	144
Plan de mantenimiento y escalabilidad .....	144
Glosario .....	145
Lista de abreviaturas .....	151
Bibliografía.....	152
Anexos .....	153
Anexo A. Presentación de anteproyecto (A mano).....	153
Anexo B. Presentación de anteproyecto (Digital).....	159
Anexo C. Bitácora de reuniones .....	163
Anexo D. Carta de visto bueno del profesor tutor .....	173
Anexo E. Declaración jurada.....	174
Anexo F. Carta de aceptación de la empresa .....	175
Anexo G. Carta de lector .....	176
Anexo H. Autorización de publicación.....	177
Anexo I. Acta de aprobación de defensa .....	179

## Índice de Tablas

<b>Tabla 1</b> Cronograma del Proyecto.....	24
<b>Tabla 2</b> Diagrama de Gantt .....	27
<b>Tabla 3</b> <i>Diagrama caso de uso - Visualizar productos</i> .....	61
<b>Tabla 4</b> <i>Diagrama caso de uso - Registrarse</i> .....	62
<b>Tabla 5</b> <i>Diagrama caso de uso - Iniciar sesión</i> .....	63
<b>Tabla 6</b> <i>Diagrama caso de uso - Agregar productos al carrito</i> .....	64
<b>Tabla 7</b> <i>Diagrama caso de uso - Realizar pedido</i> .....	65
<b>Tabla 8</b> <i>Diagrama caso de uso - Cambiar datos de perfil</i> .....	66
<b>Tabla 9</b> <i>Diagrama caso de uso - Recuperar contraseña</i> .....	67
<b>Tabla 10</b> <i>Diagrama caso de uso - Gestionar productos</i> .....	68
<b>Tabla 11</b> <i>Gestionar usuarios</i> .....	69
<b>Tabla 12</b> <i>Diagrama caso de uso - Ver pedidos</i> .....	70
<b>Tabla 13</b> <i>Diagrama caso de uso - Cambiar el estado del pedido</i> .....	71

## Índice de Figuras

<b>Figura 1</b> <i>Diagrama Causa - Efecto</i> .....	16
<b>Figura 2</b> <i>Diagrama entidad relación clásico</i> .....	43
<b>Figura 3</b> <i>Diagrama modelo relacional</i> .....	44
<b>Figura 4</b> <i>Diagrama de flujo - Registro de usuario</i> .....	46
<b>Figura 5</b> <i>Diagrama de flujo - Inicio de sesión</i> .....	48
<b>Figura 6</b> <i>Diagrama de flujo - Recuperación de contraseña</i> .....	49
<b>Figura 7</b> <i>Diagrama de flujo - Visualización de productos</i> .....	50

<b>Figura 8</b> <i>Diagrama de flujo - Agregar productos al carrito</i> .....	50
<b>Figura 9</b> <i>Diagrama de flujo - Realizar pedidos</i> .....	51
<b>Figura 10</b> <i>Diagrama de flujo - Notificación de cambio de estado de pedido</i> .....	53
<b>Figura 11</b> <i>Diagrama de flujo - CRUD de tabla de productos</i> .....	54
<b>Figura 12</b> <i>Diagrama de flujo - CRUD de tabla usuarios</i> .....	55
<b>Figura 13</b> <i>Diagrama de flujo - Gestión de pedidos</i> .....	57
<b>Figura 14</b> <i>Diagrama de flujo - Actualizar información de perfil</i> .....	58
<b>Figura 15</b> <i>Diagrama de caso de uso</i> .....	60
<b>Figura 16</b> <i>Creación de base de datos</i> .....	82
<b>Figura 17</b> <i>Diagrama final de la base de datos</i> .....	83
<b>Figura 18</b> <i>Creación del proyecto</i> .....	84
<b>Figura 19</b> <i>Agregar paquete de EntityFramework</i> .....	85
<b>Figura 20</b> <i>Agregar paquete de EntityFramework 2</i> .....	86
<b>Figura 21</b> <i>Creación de la cadena de conexión en "appsettings.json"</i> .....	87
<b>Figura 22</b> <i>Creación del Builder Services en el "Program.cs"</i> .....	87
<b>Figura 23</b> <i>Generar modelos mediante la estructura de la base de datos</i> .....	87
<b>Figura 24</b> <i>Carpeta con los modelos creados</i> .....	87
<b>Figura 25</b> <i>Modificación del "Layout.cshtml" Incluir importaciones</i> .....	88
<b>Figura 26</b> <i>Modificación del "Layout.cshtml" Agregar botones de navegación</i> ....	88
<b>Figura 27</b> <i>Botones de navegación en "AdminLayout.cshtml"</i> .....	89
<b>Figura 28</b> <i>Configuración de los servicios de sesiones de ASP.NET Core</i> .....	90
<b>Figura 29</b> <i>Pantalla de inicio</i> .....	90
<b>Figura 30</b> <i>Página de productos</i> .....	91

<b>Figura 31</b> Configuración del DbContext en el "Program.cs" .....	92
<b>Figura 32</b> Ejemplo de declaración del DbContext en una clase controladora....	93
<b>Figura 33</b> Método para desplegar lista de productos verificando si existe algún filtro por categoría o nombre .....	93
<b>Figura 34</b> Método "GetImage" .....	94
<b>Figura 35</b> Método para mostrar los detalles de la cuenta .....	95
<b>Figura 36</b> Métodos para editar información de la cuenta .....	95
<b>Figura 37</b> Página para ver detalles de cuenta y cerrar sesión .....	96
<b>Figura 38</b> Página para editar información de la cuenta .....	96
<b>Figura 39</b> Método para agregar productos al carrito .....	97
<b>Figura 40</b> Método para desplegar lista de productos en carrito .....	98
<b>Figura 41</b> Método para eliminar objetos del carrito .....	99
<b>Figura 42</b> Página del carrito.....	100
<b>Figura 43</b> Pantalla de inicio de sesión .....	100
<b>Figura 44</b> Pantalla de registro.....	101
<b>Figura 45</b> Pantalla para recuperar contraseña.....	103
<b>Figura 46</b> Método "HashPassword" para encriptar contraseñas.....	103
<b>Figura 47</b> Método de inicio de sesión .....	104
<b>Figura 48</b> Método para registrar usuario.....	105
<b>Figura 49</b> Método para solicitar recuperación de contraseña .....	106
<b>Figura 50</b> Método para cambiar la contraseña .....	106
<b>Figura 51</b> Configuración para enviar correos en el "appsetting.json" .....	107
<b>Figura 52</b> Ejemplo de correo recibido para recuperar la contraseña .....	108

<b>Figura 53</b> Método para enviar correos en la clase "EmailService.cs" .....	108
<b>Figura 54</b> Método para realizar un pedido .....	109
<b>Figura 55</b> Estructura de clase "AdminAuthorizeAttribute.cs" .....	110
<b>Figura 56</b> Código Razor para desplegar lista de productos.....	110
<b>Figura 57</b> Página para los administradores para manejar los productos .....	111
<b>Figura 58</b> Método para crear un nuevo producto.....	111
<b>Figura 59</b> Página para agregar un nuevo producto .....	112
<b>Figura 60</b> Página para administrar las categorías .....	113
<b>Figura 61</b> Código para verificar la existencia del producto .....	113
<b>Figura 62</b> Código para eliminar producto después de confirmación .....	114
<b>Figura 63</b> Página de eliminación de producto.....	114
<b>Figura 64</b> Código para desplegar información actual del producto.....	115
<b>Figura 65</b> Código para guardar los nuevos datos ingresados.....	115
<b>Figura 66</b> Página para editar datos de los productos .....	116
<b>Figura 67</b> Código para generar lista de usuarios.....	117
<b>Figura 68</b> Pantalla con lista de usuarios .....	117
<b>Figura 69</b> Método para crear usuario.....	118
<b>Figura 70</b> Página para crear un nuevo usuario.....	118
<b>Figura 71</b> Métodos para eliminar usuarios (solicitud de confirmación y eliminación) .....	119
<b>Figura 72</b> Página para eliminar usuario .....	120
<b>Figura 73</b> Código para editar usuarios.....	121
<b>Figura 74</b> Página para editar usuarios.....	122

<b>Figura 75</b> Pantalla para seleccionar factura a imprimir .....	122
<b>Figura 76</b> <i>Ejemplo de factura impresa</i> .....	123
<b>Figura 77</b> <i>Código Razor para generar lista de pedidos y organizarlo según su estado actual</i> .....	124
<b>Figura 78</b> <i>Vista de lista de pedidos</i> .....	125
<b>Figura 79</b> <i>Código para ver detalles del pedido seleccionado</i> .....	126
<b>Figura 80</b> <i>Página para visualizar los detalles del pedido</i> .....	126
<b>Figura 81</b> <i>Método para cambiar la cantidad de un producto en el pedido</i> .....	127
<b>Figura 82</b> <i>Método para cambiar el estado del pedido a finalizado</i> .....	127
<b>Figura 83</b> <i>Método para cambiar el estado del pedido a cancelado</i> .....	128
<b>Figura 84</b> <i>Método para eliminar productos del pedido actual</i> .....	129
<b>Figura 85</b> <i>Método para recalcular el total del pedido después de modificar la cantidad de un producto o eliminarlo</i> .....	129
<b>Figura 86</b> <i>Pantalla para verificar las bitácoras</i> .....	130
<b>Figura 87</b> <i>Reporte listo para ser exportado en PDF</i> .....	132
<b>Figura 88</b> <i>Creación de la base de datos en Azure</i> .....	134
<b>Figura 89</b> <i>Instalación de la extensión Azure SQL Migration</i> .....	134
<b>Figura 90</b> <i>Conexión y apartado de extensiones</i> .....	135
<b>Figura 91</b> <i>Proceso para migrar la base de datos local a la nube</i> .....	136
<b>Figura 92</b> <i>Obtener la nueva cadena de conexión en Azure</i> .....	137
<b>Figura 93</b> <i>Implementación de la nueva cadena de conexión en el archivo "appsettings.json"</i> .....	138

<b>Figura 94</b> <i>Implementación de la nueva cadena de conexión en el archivo "SuperAlfoliDbContext.cs"</i> .....	138
<b>Figura 95</b> <i>Opción "Publicar SuperAlfoliApp"</i> .....	138
<b>Figura 96</b> <i>Apartado para publicar el proyecto</i> .....	139

## **1. Capítulo 1: Planteamiento del Tema**

### **Introducción**

En este capítulo se presentará un análisis detallado del tema a desarrollar, brindando una base sólida del ámbito que envuelve la propuesta. Se tratarán temas importantes como el contexto de la empresa, la definición del problema y los objetivos del proyecto.

El análisis contextual brindará información sobre los factores internos y externos de la empresa, lo que facilitará la comprensión del lector y ayudará en el análisis de la problemática y la resolución.

Se clarificará la importancia de llevar a cabo este proyecto, demostrando su impacto en la empresa.

Se identificará y detallará la problemática para fundamentar el desarrollo del proyecto, para ello, se aplicarán diversas técnicas de análisis para encontrar el problema principal.

Por último, se delinearán los objetivos generales los cuales formularán la meta final del proyecto y los objetivos específicos que muestran el paso a paso para llegar a ese fin.

### **1.1 Antecedentes y Justificación del Anteproyecto**

#### **1.1.1. Marco de Referencia Empresarial y Contextual**

El Super Alfolí es un abastecedor situado en Rio Segundo de Alajuela, Costa Rica, este fue fundado por su actual dueño Jose Luis Porras Solís en el año

1991; esta empresa cuenta con la cédula jurídica 502440240 y el número de teléfono 24426856; se dedica a proveer, de manera satisfactoria, una amplia variedad de productos a los clientes de la comunidad y de otras zonas.

El mercado objetivo de este negocio son las personas que viven en las cercanías del lugar, que buscan realizar sus compras diarias, igualmente, se intenta expandir a clientes de otras localidades.

Siempre se busca brindar los productos de mejor calidad, con un servicio al cliente amable y operando de manera ética y transparente, el servicio se diferencia de los competidores por su sensación de calidez y confianza al momento de realizar la compra.

En el área existen otros centros de ventas los cuales buscan competir contra el Super Alfolí, los más cercanos suelen ser pequeños negocios los cuales son mayormente concurridos por los consumidores de esas zonas, suelen tener inventarios reducidos y poca variedad; en las partes más alejadas se encuentran competidores de mayor índole, los cuales, tiene diversidad de productos y clientes fijos, además de que cuentan con sistemas digitales (como páginas web con información del local y los artículos en venta) que les permiten atraer clientes más allá de su ubicación física.

Como menciona (GuellCom., 2023), que un super cuenta con una página web tiene distintas ventajas tales como: mayor alcance, información de disponibilidad, facilitación en la realización de gestiones y una imagen más profesional, por eso, muchas tiendas buscan establecer su propio entorno digital

para competir con las otras empresas; El Super Alfolí es carente de un sistema electrónico lo cual puede afectarlo frente a la competencia y reducir el número de compradores a largo plazo.

### 1.1.2 Justificación del Anteproyecto

A pesar de que el Super Alfolí tenga una sólida reputación y clientes leales, este carece de presencia en línea, esto podría reducir el crecimiento del negocio. Con el aumento de los establecimientos que han aplicado por sistemas digitales es importante que esta empresa se adapte a estos cambios.

Según lo indicado por (Ecommerce Costa Rica, 2023) tener un sistema en línea ayuda en la competitividad del negocio mejorando muchas de sus características como:

- **Servicio:**

Los clientes pueden navegar a gusto cuando quieran y donde deseen en la página, sin tener que preocuparse por los horarios del local, permitiéndoles comprar a su propio ritmo y mejorando su experiencia.

- **Posicionamiento y alcance:**

Permite a usuarios de distintas zonas enterarse de sus precios y ofertas sin la necesidad de que acudan al local, esto es perfecto para los clientes que prefieren realizar compras en línea o que no acudan a diario al establecimiento; además, la presencia en línea mejora la imagen de la empresa.

- **Reducción de costos:**

Mejorar los sistemas de inventario optimizaría el proceso de gestión de este, lo que llevaría a una mejor eficiencia operativa y una reducción de costos en personal extra, también, se reduce el coste de entrega y logística del sistema de envío de pedidos convencional.

- **Incremento en ventas:**

Al ofrecer un sistema de pedidos en línea con recolección al local y contar con un inventario actualizado permite a los clientes acceder a estos servicios de manera más conveniente, fomentando las compras en el supermercado.

Además, el servicio de retiro en la tienda o “Pick up” tiene varios casos de éxito en el mundo, como menciona el estudio de (Huayana, 2022) un ejemplo destacado es el de Victoria’s Secret en Chile, el cual alcanzó un porcentaje elevado en ventas mediante su sitio web al implementar este servicio, este caso demuestra que aplicar este sistema puede ser una estrategia efectiva para aumentar los ingresos de la empresa.

## **1.2 Definición del problema**

### **1.2.1. Problemática**

En la era digital actual es necesario que los negocios tengan participación en línea, el Super Alfolí, una tienda bien establecida, carece de presencia en la web. Esta ausencia no solo limita su alcance y visibilidad, si no que afecta su capacidad de satisfacer las necesidades de los clientes.

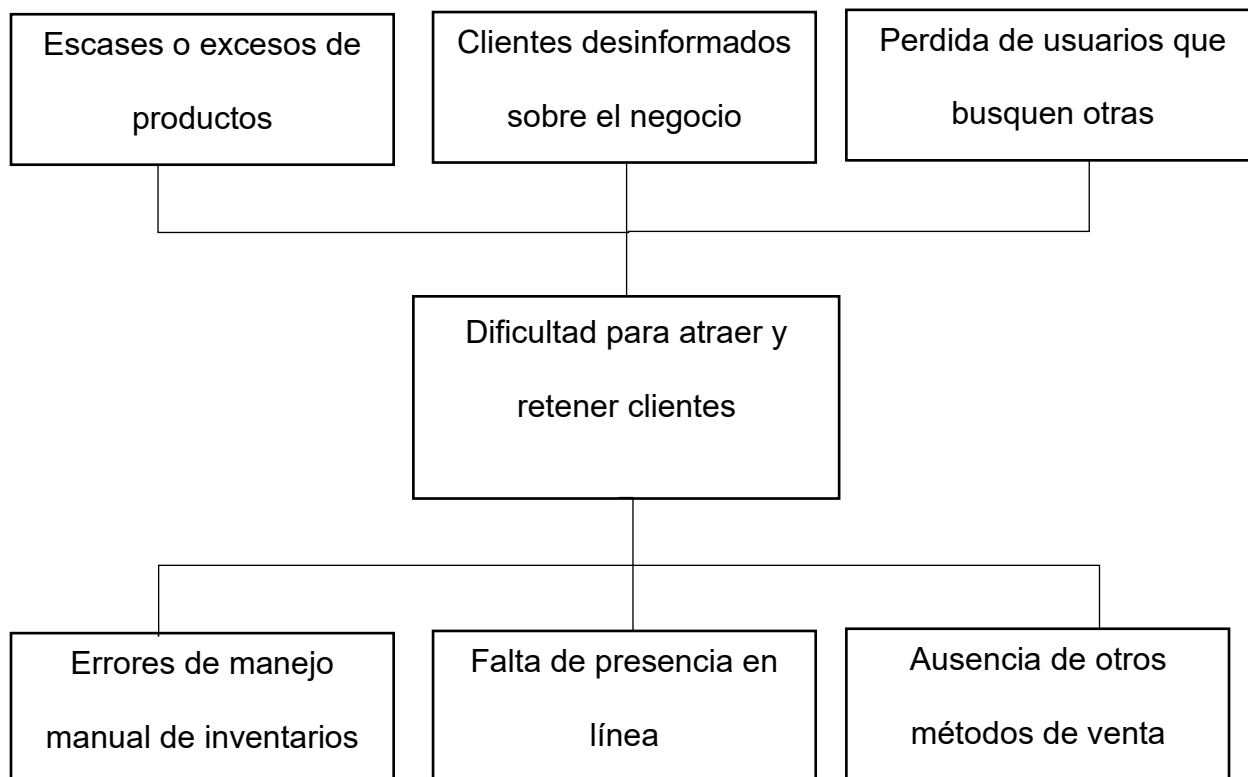
Esta dificultad para atraer y retener clientes puede deberse a la falta de presencia en línea y opciones de compra más convenientes, esto hace que los

compradores se sientan más desconectados del local y sientan una barrera a la hora de comprar debido a sus preferencias de compra, algunos prefieren evitar multitudes o tienen horarios ocupados, lo que los lleva a buscar otras alternativas que potencialmente lleven a la pérdida de clientes.

Esto también puede afectar la eficiencia interna; la gestión manual de inventarios puede llevar a cometer errores, generando una falta de existencia de productos o una sobrecompra de algunos otros, esto no solo afecta al usuario, sino que, puede resultar en pérdidas económicas para el super.

El siguiente diagrama causa – efecto mostrará el problema principal y se utilizará para visualizar las posibles causas de este:

**Figura 1** *Diagrama Causa - Efecto*



### **1.2.2. Problema General**

La falta de presencia en línea y estrategias de comercio electrónico del Super Alfolí plantea una serie de desafíos que afectan tanto su capacidad para satisfacer las necesidades de los clientes como su eficiencia interna. Con esta información, se plantea la siguiente pregunta; ¿Cómo puede el Super Alfolí mejorar su presencia en línea y estrategias de comercio electrónico para aumentar su alcance, visibilidad, satisfacción del cliente y eficiencia interna?

De esta manera se resume la problemática anteriormente mencionada y se establece un enfoque principal.

### **1.2.3. Problemas Específicos**

El problema general identificado se puede desglosar para obtener un enfoque más detallado, generando cuatro problemáticas de la siguiente manera:

¿Cuáles son las mejores estrategias para establecer una presencia en línea efectiva para el Super Alfolí?

Se debe buscar la manera de incluir a esta empresa en un entorno virtual, hay que realizar una investigación del estado actual de la empresa para identificar las fortalezas y desafíos, y de esta manera, saber cuál es la mejor manera de abordar este problema.

¿Qué herramientas y tecnologías pueden implementarse para optimizar la gestión de inventarios y procesos internos del Super Alfolí?

Esta pregunta nos plantea la necesidad de buscar las herramientas para solucionar el problema, se debe diseñar un sistema considerando los aspectos

técnicos, diseño, funcionalidades y tecnologías correctas para garantizar el funcionamiento óptimo, para ello, se comparará distintas opciones disponibles y se implementarán distintos diagramas para especificar el proceso de implementaciones de las tecnologías elegidas.

¿Cuál es el enfoque más adecuado para el desarrollo y la programación de la página web del Super Alfolí?

Este problema se centra en el desarrollo, de qué manera se aplicarán las tecnologías y diagramas generados para solventar la problemática, se buscará la manera más eficiente para completar los objetivos, este enfoque permitirá avanzar con la creación de una página web funcional y efectiva.

¿Cuál es la mejor manera de implementar el aplicativo desarrollado en el internet para permitir que los clientes la utilicen?

Se debe encontrar la manera de poder implementar el servicio creado teniendo en cuenta las limitaciones, el sistema debe ser capaz de soportar el flujo de clientes.

### **1.3. Objetivo General y Objetivos Específicos**

#### **1.3.1. Objetivo General**

Al analizar las problemáticas expuestas en el punto anterior, podemos llegar a la formulación del siguiente objetivo:

Desarrollar un sistema digital que permita gestionar inventarios y recibir pedidos de recolección al local mediante técnicas de desarrollo de aplicaciones

para aumentar la presencia en línea del Super Alfolí y así mejorar el alcance y la satisfacción al cliente en el segundo semestre del año 2024.

Mediante el cumplimiento de este objetivo se espera poder mejorar la situación actual de la empresa solucionando su problema principal; para abordar este propósito se dividirá en objetivos específicos los cuales detallarán las tareas a desarrollar para lograr la finalización de este.

### **1.3.2 Objetivos específicos**

Teniendo en mente el objetivo general podemos formular los objetivos específicos, estos detallarán las tareas a realizar a lo largo del proyecto; también tendremos en cuenta los problemas específicos como guía para la formulación de estos objetivos.

El primer objetivo debe estar relacionado con el estudio de la situación actual de la empresa, saber las fortalezas y las debilidades del negocio nos ayudará a decidir cuál es el mejor camino para solucionar este problema, lo que nos lleva a formular el siguiente objetivo:

Determinar la situación actual del Super Alfolí mediante herramientas de recaudación de información para entender las fortalezas y debilidades del negocio.

El segundo se enfocará en el diseño del sistema, se debe analizar las necesidades de la empresa y establecer las funciones que se implementarán, además, se debe identificar las herramientas y tecnologías que se van a utilizar en el proyecto, con todo eso aclarado, se formularán diagramas para facilitar el

desarrollo del aplicativo en el futuro, con esto en mente, podemos redactar el siguiente objetivo:

Diseñar la estructura del proyecto utilizando como base las necesidades de la empresa mediante diagramas y otras técnicas para facilitar el proceso de desarrollo.

El tercer objetivo se va enfocado en el tema de desarrollo, en este, se buscará desarrollar el sistema intentando que sea fácil de usar, con una estructura sólida y que cumpla las necesidades de la empresa, el planteamiento del objetivo sería:

Desarrollar el sistema digital utilizando la estructura y tecnologías planificadas para cumplir con los requisitos del aplicativo expuestos por la empresa.

El cuarto objetivo consiste en implementar el sistema a la web, mediante servicios de alojamiento en la nube, para que sea accesible para la clientela de la tienda y los miembros de la empresa:

Integrar el sistema informático al internet mediante herramientas de alojamiento en la nube con el fin de que esté disponible para los clientes.

## **1.4. Alcance y Limitaciones**

### **1.4.1 Alcance del Proyecto**

El primer entregable de proyecto es un informe de la situación actual, en él se incluye los resultados de las herramientas de recaudación de información, con estos datos se determinará la mejor manera de desarrollar el diseño del sistema.

El segundo entregable consiste en la elaboración de diagramas como entidad - relación, de flujo, de casos de uso, entre otros; además de otros planeamientos para definir la estructura del sistema, se busca diseñar un sistema el cual permita a los clientes visualizar los productos vendidos por la tienda y permitirles realizar pedido tipo recoger en el local o "Pick up" (es importante aclarar que el servicio de recolección al local no contará con ningún sistema de envíos ni con ningún tipo de pago electrónico, el cliente deberá acudir a la tienda para recoger sus productos y cancelar el monto); del lado del local debe permitir a los empleados identificarse para poder cambiar la disponibilidad de los productos y atender los encargos del servicio de recolección en el local.

El tercer entregable abarcaría la elaboración del sistema utilizando los planeamientos propuestos en el segundo entregable como base, se desarrollará la aplicación web de la manera más eficiente que cumpla con los requisitos necesarios por la empresa, se desarrollará la base de datos y cada módulo anteriormente descritos; cuando la página esté terminada será subida a la web utilizando las herramientas escogidas a la hora de realizar el diseño.

### 1.4.2 Limitaciones del Proyecto

En el proyecto existen ciertas limitaciones para tener en cuenta, que, aun que no puedan ser evitadas, no impiden la realización del proyecto.

Tercer entregable:

- No se espera comprar un nombre de dominio por lo cual habrá que utilizar un enlace brindado gratuitamente por la herramienta que se utilice.
- El dueño de la empresa menciona que de momento prefiere mantener los costos lo más bajo posible, por eso se debe buscar utilizar los servicios de alojamiento en la nube más económicos, igualmente, según el flujo de usuarios esperado, estos servicios deberían funcionar sin problemas.
- Al buscar un servicio gratuito o de bajo costo se espera que la velocidad de la página se pueda ver impactada según el flujo de usuarios en la página. En caso de que el flujo de usuarios supere el esperado se podría ver reflejado en la velocidad de carga del servicio.

### 1.5 Cronograma del Proyecto

Para completar este proyecto se estima un total de doce semanas, las cuales tres serán para el primer entregable, tres para el segundo y seis para el tercero, estos entregables se dividen en subetapas las cuales son:

Primer entregable:

- Crear una entrevista para realizar al dueño del Super Alfolí, buscando las preguntas necesarias, para recaudar información del estado actual del negocio.

- Aplicar la entrevista realizada para obtener los datos de la empresa necesarios para el diagnóstico.
- Utilizar la información obtenida para realizar un informe del estado de la empresa.

Segundo entregable:

- Escoger las tecnologías y herramientas que se utilizarán en el desarrollo del proyecto.
- Creación de diagrama de entidad relación y de flujo.
- Creación del diagrama de casos de uso y corrección de errores existentes.

Tercer entregable:

- Crear la base de datos del aplicativo.
- Confeccionar el módulo principal de la página.
- Realizar el sistema de pedidos de recolección en el local de la página.
- Realizar el sistema de modificación de inventarios del aplicativo.
- Realizar el sistema de visualización de pedidos.
- Aplicar los últimos detalles y realizar pruebas.

Se plantearán como objetivos semanales de la siguiente manera:

## Tabla

1

**Cronograma del Proyecto**

Semanas	Objetivo	Actividades
1	Crear una entrevista para realizar al dueño del Super Alfolí, buscando las preguntas necesarias, para recaudar información del estado actual del negocio.	Crear preguntas para realizarlas al dueño del establecimiento que ayuden a entender la situación de la empresa.
2	Aplicar la entrevista realizada para obtener los datos de la empresa necesarios para el diagnóstico.	Acudir al local para aplicar la entrevista al dueño del Super Alfolí apuntando las respuestas en una hoja de preguntas.
3	Utilizar la información obtenida para realizar un informe del estado de la empresa.	Desarrollar un plan estratégico aprovechando la información obtenida a través de los resultados de la encuesta.
4	Escoger las tecnologías y herramientas que se utilizarán en el desarrollo del proyecto.	Realizar una investigación, basándose en los resultados de la encuesta para escoger las

		tecnologías y herramientas a utilizar.
5	Creación de diagrama de entidad relación y de flujo.	Realizar el diagrama entidad relación para facilitar la creación de la base de datos y el diagrama de flujo para que sea más fácil la comprensión de los procesos
6	Creación del diagrama de casos de uso y corrección de errores existentes.	Realizar diagramas de caso de uso para los diversos procesos para entender la interacción de los actores con el sistema
7	Crear la base de datos del aplicativo.	Crear la base de datos del programa en la tecnología seleccionada.
8	Confeccionar los módulos de página principal y de página de información.	Crear la página principal en donde se pueden ver los productos y la página de información en donde se vean diversos datos del local.
9	Realizar la página de inicio de sesión y el sistema de pedidos de recolección en el local de la página.	Crear la página en donde los usuarios puedan iniciar sesión y el módulo donde los clientes con

		sesión iniciada puedan realizar los productos.
10	Realizar el sistema de modificación de inventarios del aplicativo.	Crear el módulo donde los usuarios empleados, que hayan iniciado sesión, puedan modificar la disponibilidad actual de los productos.
11	Realizar el sistema de visualización de pedidos, realizar pruebas y corregir los últimos detalles.	Crear el módulo donde los usuarios empleados, que hayan iniciado sesión, puedan visualizar los pedidos realizados recientemente.  Realizar pruebas al sistema y refinar los detalles pendientes.
12	Implementar el sistema en la web.	Subir el aplicativo a la web mediante las herramientas elegidas.

Se presentará los datos del cronograma como un diagrama de Gantt para facilitar su comprensión, la primera columna describe la actividad a realizar y las siguientes representan la semana de cada mes, las celdas sombreadas indican el tiempo estimado para la tarea.

Tabla 2 Diagrama de Gantt

Actividades	Tiempo de duración															
Meses	Junio				Julio				Agosto				Septiembre			
Semanas	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
Crear preguntas para realizarlas al dueño del establecimiento																
Acudir al local para aplicar la entrevista y apuntar las respuestas																
Desarrollar un plan estratégico basado en los resultados obtenidos																
Realizar investigación para seleccionar tecnologías y herramientas																
Diseñar diagramas para facilitar la comprensión de procesos y la base de datos																

Diseñar casos de uso para entender la interacción entre los actores y el sistema																		
Construir la base de datos en la tecnología seleccionada																		
Crear páginas de productos y de información del local																		
Crear el inicio de sesión y módulo de pedidos																		
Crear módulo de modificación de disponibilidad de productos																		
Crear módulo de pedidos, realizar pruebas y corregir detalles																		

Las actividades de este cronograma pueden estar sujetas a cambios en caso de que surjan nuevos requerimientos o situaciones imprevistas.

## **2. Capítulo 2: Objetivo 1**

### **2.1 Definición de conceptos**

Es importante para el proyecto conocer el estado actual de la empresa para saber sus fortalezas y debilidades, además, es necesario saber algunos datos extra para tenerlos en cuenta a la hora de realizar el diseño.

Como menciona (Lopez, 2021) realizar una entrevista es un método sencillo para recopilar datos a la hora de realizar un proyecto, aprovechando el fácil acceso que se tiene con el dueño del local, se utilizará este método para obtener información valiosa.

Igualmente hay que tener distintas cosas en mente como: realizar preguntas directas, y preguntar por hechos y no por opiniones.

Será importante abordar temas como: infraestructura tecnológica (hardware y software), redes y conectividad, personal y sus capacidades, procesos operativos y expectativas de la empresa; estas cuestiones las tendremos en cuenta a la hora de estructurar la entrevista.

Una vez completada la lista de preguntas, se deberá acudir al super para obtener las respuestas, las cuales serán analizadas y utilizadas para facilitar el diseño del sistema y evitar errores futuros.

### **2.2 Diseño de la investigación**

Esta entrevista buscará recaudar datos con un enfoque mixto para utilizar esta información en el diseño del sistema, se necesita saber la cantidad y calidad de los equipos del local, la cantidad de empleados, la calidad de las conexiones a la red,

información de cómo trabajan de momentos algunos temas como los inventarios y sus expectativas de la aplicación, para ello, se realizará las entrevistas al dueño del Super Alfolí el cual conoce perfectamente esta información.

Se acudirá al local con las preguntas y se escribirán las respuestas que el brinde para posteriormente analizarlas.

A continuación, se escribirán las preguntas planteadas y el objetivo o información esperada de ellas:

- 1- ¿Cuántas computadoras y otros equipos tecnológicos tiene la empresa actualmente?

Es importante saber cuántas computadoras o equipos tiene la empresa para saber si el equipo es suficiente para la aplicación.

- 2- ¿Cuál es la capacidad y estado de las computadoras (procesador, memoria RAM, almacenamiento)?

Es necesario ver si los equipos necesitaran alguna actualización en sus componentes.

- 3- ¿Qué software utilizan actualmente para la gestión de inventarios, ventas y administración?

Esta pregunta nos ayuda a identificar integraciones y a evitar redundancias de software.

- 4- ¿Cuál es la infraestructura de red actual (tipo de conexión a internet, velocidad, cobertura Wi-Fi)?

La velocidad y estabilidad de la conexión a internet son cruciales para el rendimiento de una aplicación web.

5- ¿Tienen problemas frecuentes de conectividad o interrupciones del servicio?

En caso de haber un problema frecuente de red se debe solucionar ya que la aplicación será un servicio web.

6- ¿Cuántos empleados trabajan en la empresa y cuál es la estructura del personal?

Entender la estructura de la empresa nos ayuda a crear una aplicación que se adapte a todas las necesidades de los usuarios.

7- ¿Cuántos empleados tienen acceso y utilizan regularmente las computadoras y cuentan con un nivel de competencia tecnológica?

Permite dimensionar el uso mínimo de la aplicación, a planificar el diseño de una aplicación sencilla de usar y nos ayuda a saber si los usuarios necesitaran una capacitación

8- ¿Cómo gestionan actualmente los catálogos de productos?

Nos ayuda a entender cuales distintas áreas podemos mejorar y que funciones nuevas debemos integrar a nuestra aplicación.

9- ¿Qué tipo de comentarios han recibido sobre la experiencia de compra y el catálogo de productos por parte de los clientes?

Nos ayuda a identificar las necesidades de los compradores para satisfacer sus necesidades.

10-¿Qué características y funcionalidades les gustaría tener en una nueva aplicación web?

Saber que se espera de la aplicación nos ayudará a priorizar las funcionalidades solicitadas.

11-¿Tiene alguna sugerencia específica sobre cómo debería ser el proceso de implementación de la nueva aplicación web?

Nos permite ver la perspectiva del dueño y como se espera que sea la implementación, en caso de que haya algún inconveniente o tema no tratado en la entrevista, se espera que sea mencionado en esta pregunta.

## **2.3 Resultados**

Con la lista terminada se acude al Super Alfolí y se le pregunta al actual dueño del local Jose Luis Porras Solís, se realizaron las preguntas y se apuntaron las respuestas recibidas transcribiendo la información importante.

Los resultados se han dividido en subgrupos para facilitar su análisis:

### **2.3.1 Infraestructura Tecnológica**

#### **2.3.1.1 Hardware**

Mencionan que en el local cuentan con dos computadoras, una se suele utilizar para un software de cobro y cajas y el otro equipo se utiliza para gestiones del banco como pagos electrónicos, depósitos y retiros de dinero.

Mencionan que, aunque no cuenta con la información exacta de los componentes del equipo creen que son gama media y que funcionan de manera óptima.

#### **2.3.1.2 Software**

Tienen un programa que habían comprado hace unos años el cual registra los productos por códigos de barra y permite realizar facturas de manera más sencilla, se utiliza en el día a día para realizar las ventas.

### **2.3.1.3 Redes**

Cuentan con un internet de fibra óptica con una velocidad de 50Mbs, en el local se conectan mediante wifi y una infraestructura cableada para las dos computadoras, afirman que el servicio suele ser estable y que pocas veces tienen problemas de conectividad.

### **2.3.2 Personal y Gestión**

Actualmente cuentan con seis empleados los cuales son, un encargado de caja, una persona de limpieza, un panadero, dos abarroteros y él, que, aunque es el jefe, realiza las funciones que se necesiten en el día, menciona que tres de sus empleados y él saben utilizar una computadora y serían los encargados de gestionar el inventario y recibir los pedidos.

### **2.3.3 Procesos**

Al momento de realizar la entrevista gestionan el inventario de productos en un cuaderno el en cual apuntan que suelen comprar y en donde, además, cuales productos están empezando a escasear para hacer un nuevo pedido.

### **2.3.4 Experiencia del Cliente y Expectativas**

Muchos clientes han mencionado que les gustaría poder revisar los productos disponibles mediante una página web y al mencionarles la idea del servicio de recolección al local, se reciben comentarios positivos sobre lo útil que podría llegar a ser.

El señor Jose nos comenta que le gustaría que él pudiera ver que productos están agotados para poder realizar un nuevo pedido, además, de que la aplicación se sencilla

de utilizar, menciona que espera poder implementarla en el último cuatrimestre del año 2024.

## **2.4 Análisis y discusión**

Las respuestas obtenidas nos dejan claro ciertos puntos, el local cuenta con la tecnología suficiente para poder implementar la aplicación web, los dos equipos son capaces de ejecutar sin problemas un navegador web con un sistema como el que se espera diseñar, además, cuenta con un sistema de redes suficientemente estable para evitar problemas de pérdida de conexión.

Los empleados son capaces de utilizar aplicativos en las computadoras con una capacitación básica sobre el funcionamiento.

Se debe implementar una mejor manera de poder administrar el inventario de productos, el cual es una de las funciones esperadas en la aplicación, ya que el sistema actual es lento, poco fiable y confuso.

Los clientes esperan la página web ya que facilita algunos de los procesos a la hora de realizar una compra, según el cronograma propuesto la aplicación estaría lista para el tercer cuatrimestre del año 2024, lo que encaja con el deseo del dueño de implementarlo al final de este año.

### 3. Capítulo 3: Objetivo 2

#### 3.1 Definición de conceptos

Antes de iniciar con la programación es importante definir la estructura del programa, esto nos ayuda a identificar las funciones necesarias y como interactúan entre ellas.

Para el trabajo se busca crear un diagrama de entidad relación, un diagrama de flujo y un diagrama de casos de uso, cada uno con su función e importancia, además, brindar una especificación clara de las tecnologías que se planean utilizar a lo largo del proyecto.

Como menciona (Navarro, 2019) el modelo entidad relación nos ayuda a identificar como las entidades se relacionan entre ellos, esto se hace mediante componentes los cuales son los elementos básicos del modelo.

- Las entidades se representan como un rectángulo y son los objetivos de los cuales se recoge la información.
- Las relaciones muestran la conexión entre las entidades, suelen ser verbos dentro de rombos los cuales asocian a las entidades según su interacción
- Los atributos son las propiedades de una entidad, se escriben dentro de un círculo, siempre debe existir uno que funcione como identificador único.
- La cardinalidad indica la cantidad máxima o mínima de ocurrencias en cada relación.

El diagrama de flujo nos ayuda a diseñar la base de datos para la aplicación, cada entidad representa una tabla, los atributos son sus columnas y la relaciones y la cardinalidad representa como se relacionan las tablas entre ellas.

Identificar los pasos de una tarea para poder completarla, de esta manera podemos diseñar un proceso, corroborar que se están cumpliendo las tareas y mejorar los pasos para que sea más eficiente.

Según escribió (Etecé, 2024) el diagrama está formado por símbolos y líneas, los rectángulos con esquinas redondeadas marcan el inicio y el final de la tarea, los rectángulos con esquinas normales son los procesos, los rombos son decisiones las cuales dividen el curso en dos caminos posibles y el romboide representa cuando el proceso requiere entradas de datos o genera salidas de datos, las líneas sirven para definir el flujo del proceso.

Y por último el diagrama de casos de uso, es un tipo de diagrama UML que nos permite ver cuáles son los roles de un sistema y como interactúan con la aplicación, este está conformado por:

- El actor, el cual es la entidad que interactúa con el sistema, se dibuja como un esqueleto.
- El caso de uso es una acción dentro del sistema se dibuja con un óvalo con la función escrita dentro de él.
- El sistema es todo el aplicativo, se representa con un gran rectángulo.

Todas estas herramientas nos ayudarán a crear un diseño del sistema que se tiene en mente y facilitará la creación de la aplicación web cuando comience la programación.

## **3.2 Desarrollo**

### **3.2.1 Establecer los requerimientos**

El primer paso a la hora de diseñar el proyecto será definir los requerimientos, estos nos ayudan a que el proyecto siga un camino claro y se comprenda que se espera del proyecto; los requerimientos se dividen en dos tipos, funcionales y no funcionales, juntos representan qué debe realizar la aplicación para que sea satisfactoria.

#### **3.2.1.1 Requerimientos Funcionales**

Los requerimientos funcionales describen qué debe hacer el sistema:

1. El sistema debe permitir al usuario registrarse mediante un correo electrónico, una contraseña, su nombre y apellidos y su teléfono.
2. El usuario debe poder iniciar sesión con sus credenciales.
3. Los usuarios pueden actualizar la información de su perfil como el nombre, el teléfono o la contraseña.
4. El sistema tiene que mostrar la dirección, números de teléfono y horario de atención.
5. El usuario debe poder visualizar una lista de productos desplegada y su cantidad actual.
6. El usuario debe poder buscar un producto utilizando una parte del nombre.
7. El usuario podrá añadir productos al carrito de compra.
8. El usuario debe poder realizar pedidos de recolección al local.
9. El sistema debe notificar mediante correo electrónico cuando el pedido esté listo.

10. El sistema debe notificar mediante correo electrónico en caso de que pedido sea cancelado.
11. Permitir a los administradores crear, leer, actualizar y eliminar los datos en la tabla de usuarios.
12. Permitir a los administradores crear, leer, actualizar y eliminar los datos en la tabla de productos.
13. El sistema debe mostrar el contenido de los carritos.
14. Permitir a los administradores ver los pedidos.
15. Permitir a los administradores cambiar el estado del pedido.
16. Permitir a los administradores modificar la cantidad de productos solicitados en un pedido.
17. Permitir a los administradores eliminar un pedido en caso de que no haya disponibilidad.

### **3.2.1.2 Requerimientos No Funcionales**

Describen como debe comportarse el sistema; incluye aspectos como rendimiento, usabilidad, seguridad, mantenimiento y más.

1. El sistema debe responder a cualquier solicitud en menos de 5 segundos.
2. El sistema debe soportar hasta 150 usuarios sin de grado de rendimiento.
3. El sistema debe estar disponible al menos un 90% del horario de atención.
4. Los datos de inicio de sesión deben estar encriptados.
5. El sistema debe ser compatible con los navegadores más utilizados.
6. El sistema debe ser sencillo de utilizar.
7. El sistema debe tener un diseño responsivo.

Con esta lista creada se pueden empezar a diseñar los diagramas asegurando que todos cumplan y sigan los requerimientos citados.

### **3.2.2 Diagrama entidad relación**

Para realizar el diagrama de entidad relación se deben definir ciertos elementos:

#### **3.2.2.1 Entidades**

Las entidades representan los elementos de los cuales se debe recaudar información, para este trabajo se identificaron las siguientes entidades:

- Usuarios: almacena la información del usuario de la aplicación, guarda registros como el nombre y los datos de inicio de sesión y de contacto.
- Productos: contiene la información sobre el inventario de la tienda, cuales y cuantos productos tiene.
- Pedidos: guarda la información de los pedidos realizados por los usuarios como el monto total, la fecha en que se realiza y el estado del pedido.
- Detalle de Pedidos: tiene el detalle de cada producto comprado en un pedido, guarda su cantidad de productos comprados para poder calcular un total.
- Carritos: almacena los datos de los carritos existentes, el usuario al que pertenecen y su fecha de creación.
- Objetos de carrito: contiene todos los productos que estén en el carrito de un usuario y su cantidad.
- Sesión: posee datos necesarios para manejar la sesión del usuario como la fecha de expiración o el token de sesión.

### 3.2.2.2 Atributos

#### **Usuarios:**

- IDUsuario (PK): sirve como identificador único para cada usuario.
- Nombre: nombre del usuario.
- Apellidos: apellidos del usuario.
- Correo: correo electrónico del usuario, se utiliza para iniciar sesión y para contactarlo en caso de ser necesario para informar sobre el estado de los pedidos que se realicen.
- Teléfono: se utiliza para contactar al usuario en caso de ser necesario hacer un cambio en el pedido o informar un inconveniente.
- Clave: contraseña encriptada del usuario.
- Rol: nos ayuda a identificar si la cuenta del usuario es de cliente o de administrador.
- Estado: muestra si la cuenta está actualmente activa o desactivada.

#### **Productos:**

- IDProducto (PK): sirve como identificador único para cada producto.
- Nombre: nombre del producto.
- Descripción: texto que muestra los detalles de un producto como su tamaño, peso, cantidad, envasado entre otras cosas.
- Cantidad: cantidad del producto disponible estimada actual.
- Precio: precio del producto.

- Categoría: categoría del producto, sirve para poder organizar los productos por categorías.
- Imagen: imagen representativa del producto que se utilizará en la página de visualización de productos.

**Pedidos:**

- IDPedido (PK): sirve como identificador único para cada pedido.
- IDUsuario (FK): sirve para identificar a cuál usuario pertenece el pedido.
- Fecha: fecha en la que se ha realizado el pedido.
- Estado: estado actual del pedido, este puede ser Pendiente - Completado - Cancelado.
- Total: cantidad total a pagar por el cliente.

**Detalles de pedidos:**

- IDDetalle (PK): sirve como identificador único para cada detalle de pedido.
- IDPedido (FK): sirve para identificar a cuál pedido pertenece el detalle.
- IDProducto (FK): sirve para saber que producto fue comprado.
- Cantidad: cantidad del producto comprado, se utiliza para calcular el total multiplicando cantidad por precio.
- Precio: precio del producto a la hora de la compra, se utiliza para calcular el total multiplicando cantidad por precio.
- Total: resultado de la cantidad comprada por el precio por unidad.

**Carritos:**

- IDCarrito (PK): sirve como identificador único para cada carrito.
- IDUsuario (FK): sirve para identificar a cuál usuario pertenece el carrito.
- Fecha: fecha en la que se creó el carrito, sirve para saber cuándo eliminar el carrito por falta de uso.

#### **Objetos de carrito:**

- IDObjetos (PK): sirve como identificador único para cada objeto en carrito.
- IDCarrito (FK): sirve para identificar a cuál carrito pertenece el objeto.
- IDProducto (FK): sirve para identificar cuál producto fue agregado al carrito.
- Cantidad: cantidad del producto agregado al carrito.
- Precio: precio del producto a la hora de ser agregado al carrito.
- FechaAgregado: fecha en la que se agregó al carrito.

#### **Sesión:**

- IDUsuario (FK): sirve como identificar a cuál usuario pertenece la sesión.
- IDSesion (PK): sirve como identificador único para cada sesión.
- FechaExpiracion: sirve para saber cuándo expira la sesión.
- Token: clave que sirve para verificar la sesión.

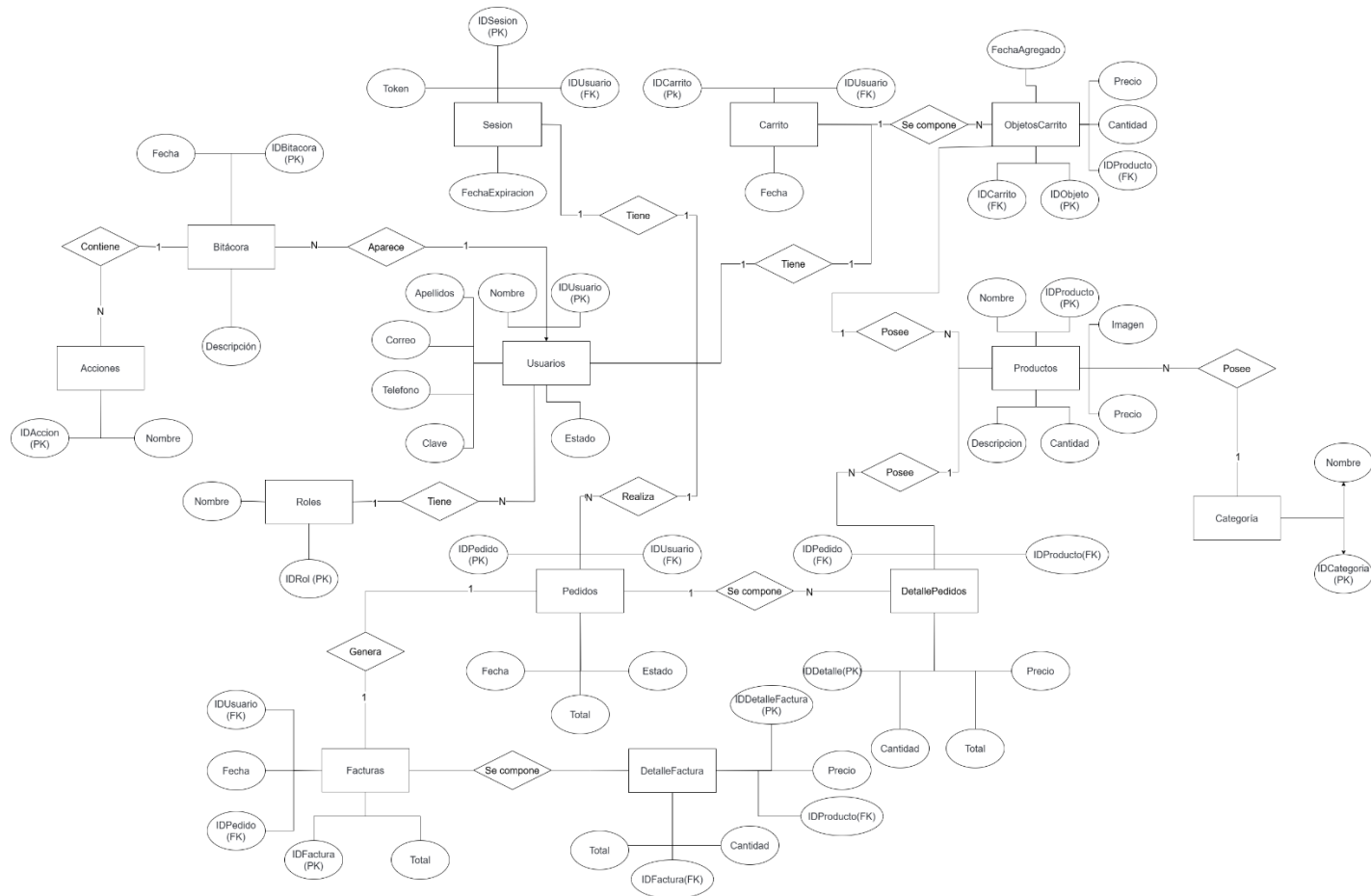
#### **3.2.2.3 Relaciones**

- Usuarios a Pedidos: un usuario puede hacer varios pedidos, pero un pedido solo pertenece a un usuario. N:1
- Usuarios a Carritos: un usuario puede tener solo un carrito y un carrito solo puede pertenecer a un usuario. 1:1

- Usuarios a Sesión: un usuario solo puede tener una sesión y una sesión solo puede pertenecer a un usuario. 1:1
- Pedidos a DetallePedidos: un pedido puede tener muchos detalles, pero un detalle solo un pedido. N:1
- Productos a DetallePedidos: un producto puede ser parte de distintos detalles, pero cada detalle tiene solo un producto. N:1
- Carritos a ObjetosCarritos: un carrito puede tener varios objetos, pero un objeto solo pertenece a un carrito. N:1
- Productos a ObjetosCarritos: un producto puede aparecer en muchos objetos, pero cada objeto solo tiene un producto N:1

Una vez definidas estas partes se puede crear el siguiente diagrama para resumir lo mencionado:

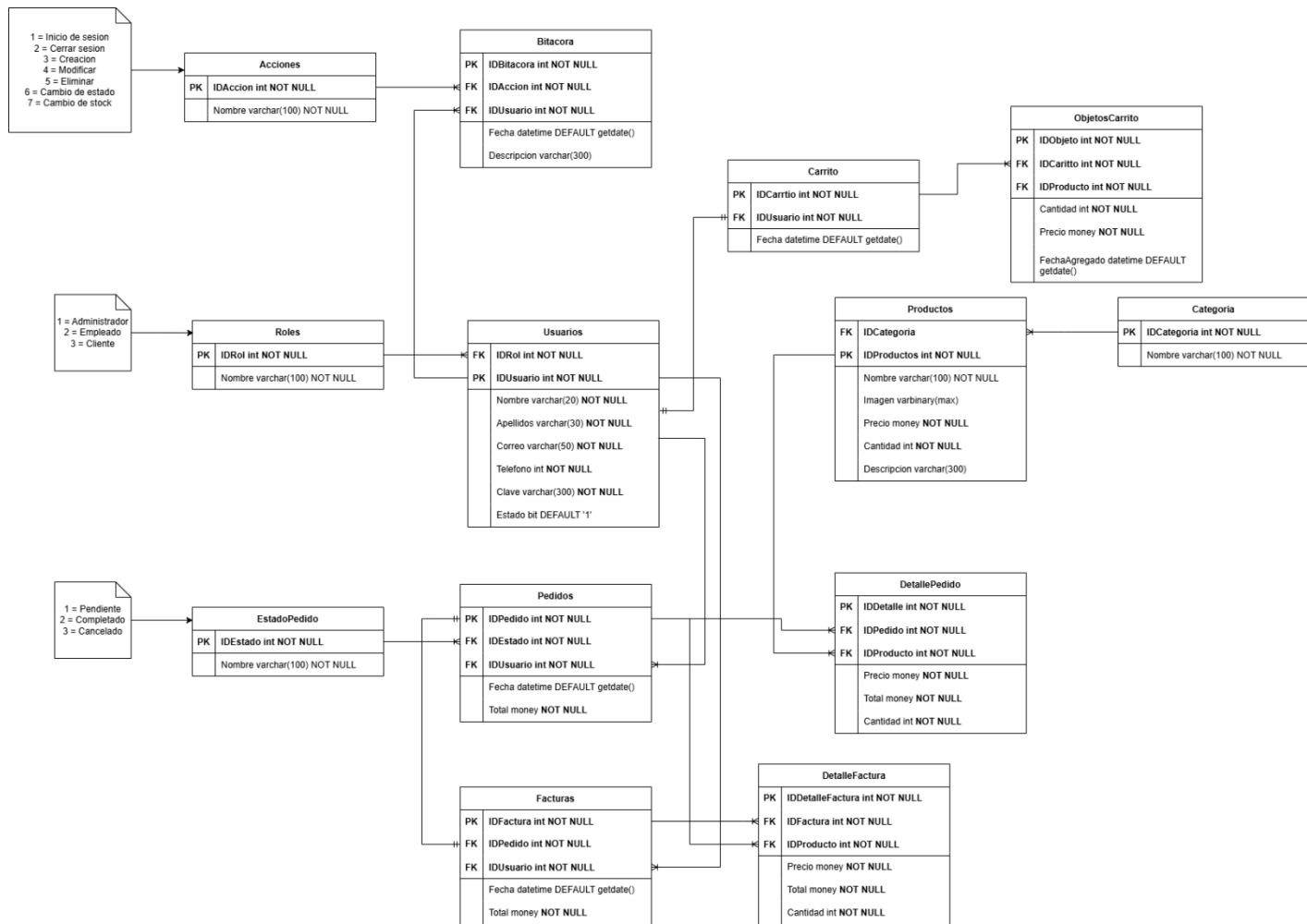
**Figura 2** *Diagrama entidad relación clásico*



El diagrama de entidad relación clásico es útil para resumir la información obtenida en el paso anterior mediante una visión general y abstracta, es perfecta para utilizarla con usuario con poco conocimiento técnico.

Si se busca una implementación más técnica, se opta por el diagrama del modelo relacional, esta transforma a las entidades en tablas, a los atributos en columnas y las relaciones en uniones de llaves foráneas y muestra el tipo de dato que se espera almacenar, es perfecta para facilitar el trabajo a la hora de crear bases de datos.

**Figura 3** Diagrama modelo relacional



Con el diagrama realizado será más fácil la creación de la base de datos a la hora de realizar el desarrollo.

### 3.2.3 Diagrama de flujo

El diagrama de flujo es una herramienta que nos permite ver cómo se desarrolla un proceso, esto nos permite entender cómo avanza un algoritmo sin la necesidad de leer largos textos y ayuda a los programadores a definir la estructura del código antes de iniciar a programar.

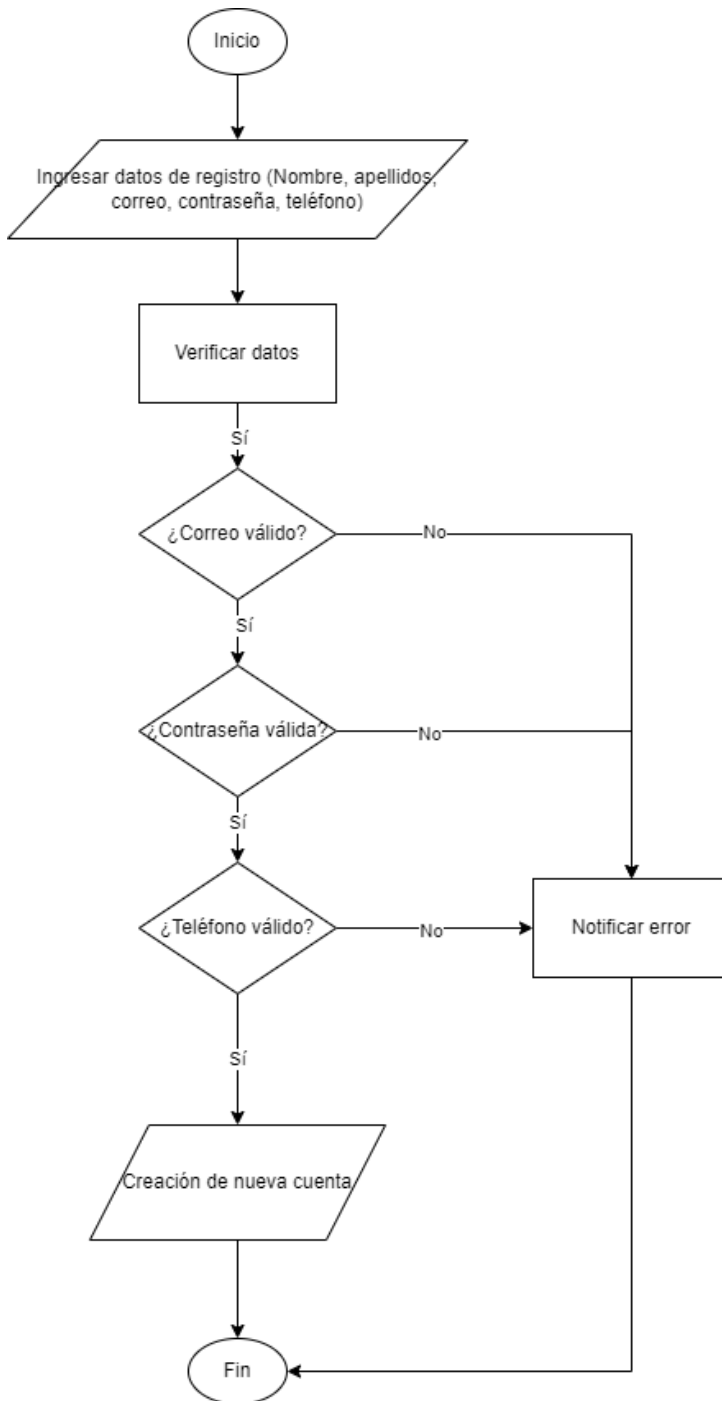
El primer paso para crear los diagramas es definir los procesos principales los cuales se quieren representar, para ayudarnos con esto podemos ver los requerimientos anteriormente redactados para tener una idea de que algoritmos debe realizar la aplicación.

Después se procede a diagramar. Para ello, como se explicó anteriormente, se utilizan figuras para representar los procesos, las decisiones, las salidas e ingresos de datos, el comienzo y el fin, además, se brinda una breve explicación de la representación para esclarecer cualquier duda.

En este proyecto destacan los siguientes procesos:

1. **Proceso de registro de usuario:**

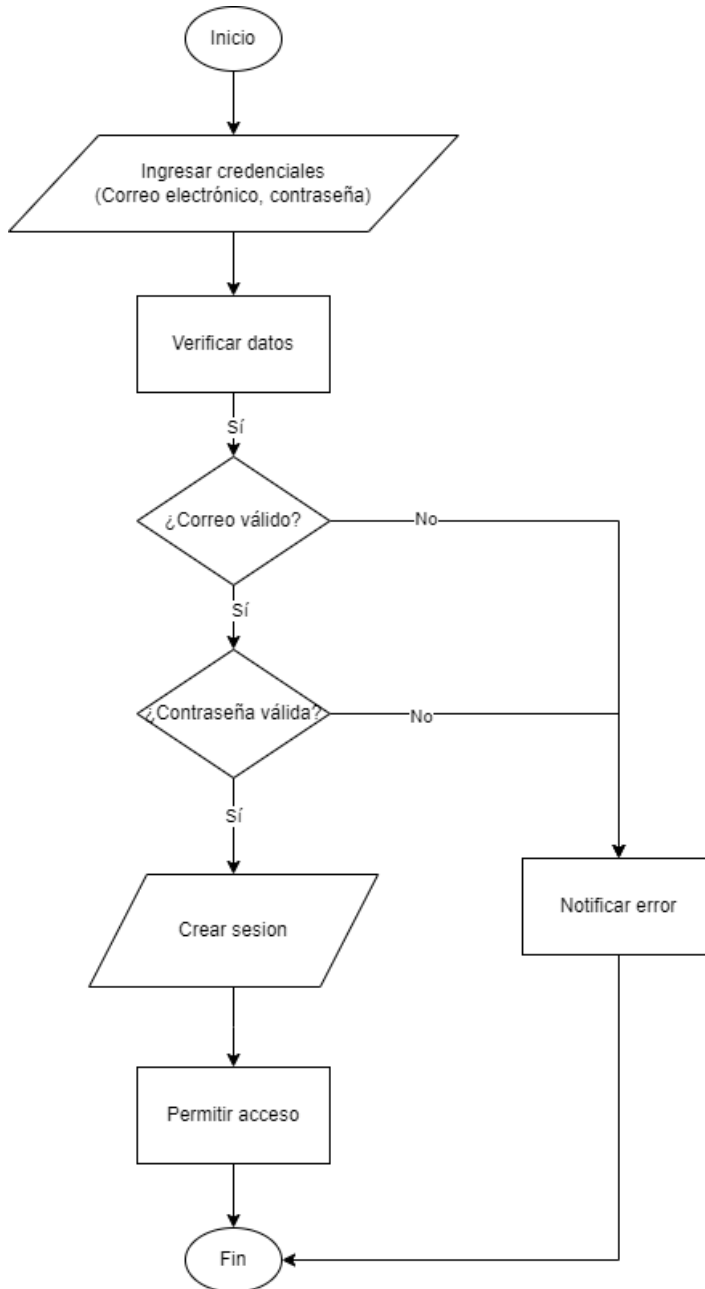
**Figura 4** *Diagrama de flujo - Registro de usuario*



Este proceso comienza desde la pantalla de registro de usuario en donde él ingresa sus datos personales, luego el sistema confirma que los datos sean válidos (en caso de no serlos se notifica para que el usuario los modifique) y después procede a guardar la información en la base de datos.

## 2. Inicio de sesión

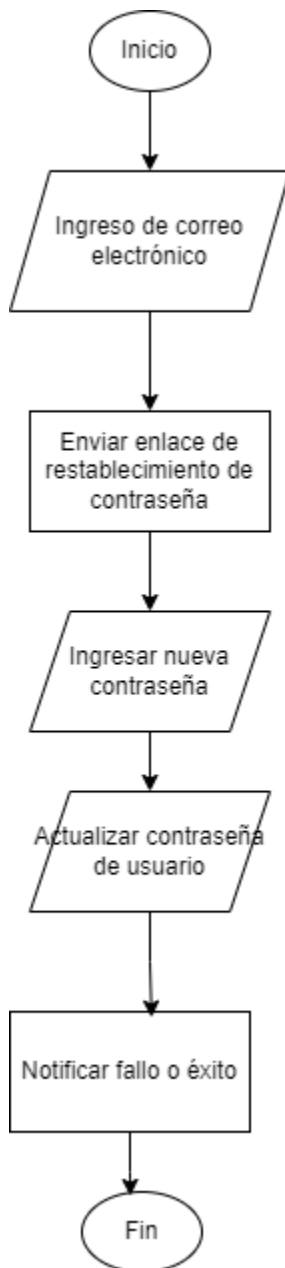
Figura 5 Diagrama de flujo - Inicio de sesión



Empezando desde la pantalla de inicio de sesión el usuario ingresa sus credenciales (correo electrónico y contraseña), el sistema valida que los datos sean correctos y en caso de serlos crea una sesión y permite el acceso.

### 3. Recuperación de contraseña

Figura 6 Diagrama de flujo - Recuperación de contraseña

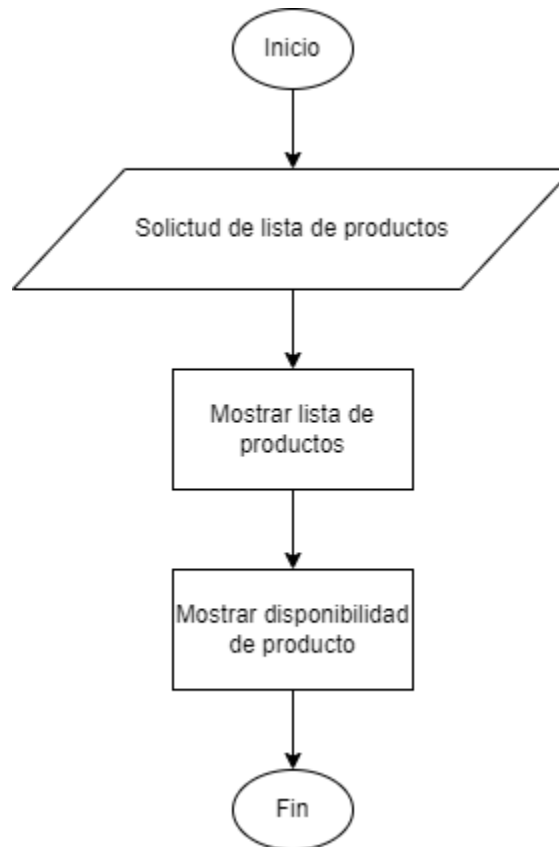


En la opción de recuperación de contraseña el navegante ingresa el correo electrónico con el cual está registrado, a este correo se envía un enlace de

restablecimiento de credenciales, en donde se debe ingresar la nueva clave y se actualiza en la base de datos.

#### 4. Visualización de productos

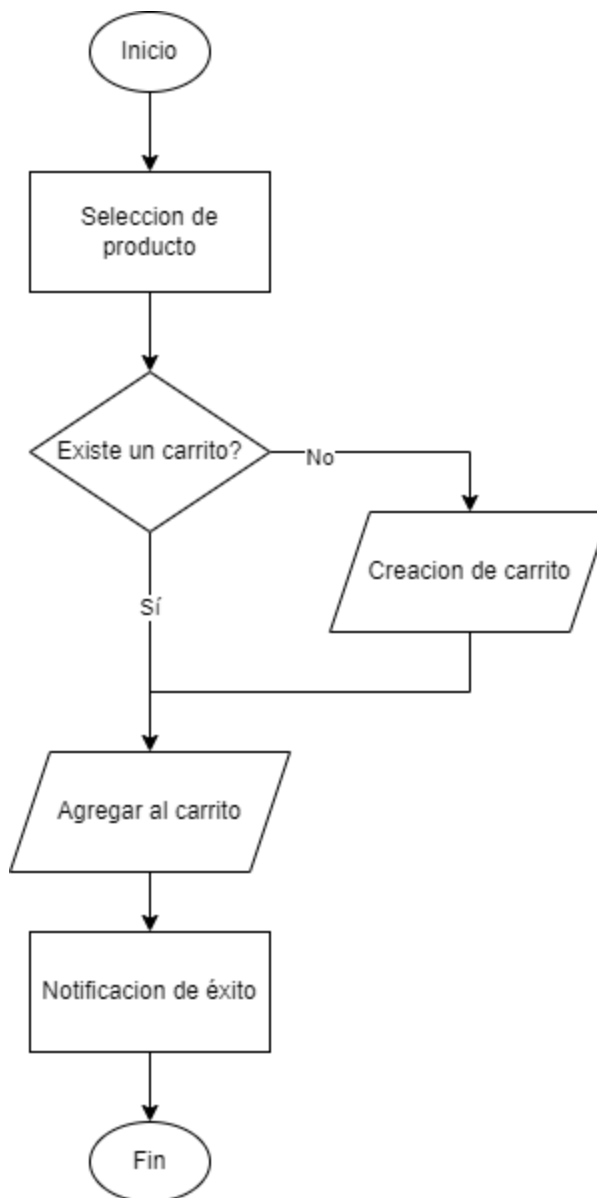
**Figura 7** Diagrama de flujo - Visualización de productos



Se solicita la información de los productos a la base de datos, después, estos se despliegan en una lista en donde se puede ver sus atributos y si el producto está actualmente disponible.

#### 5. Agregar productos al carrito

**Figura 8** Diagrama de flujo - Agregar productos al carrito



Al seleccionar “Agregar al carrito” sobre algún producto el sistema consulta si existe algún carrito (si no existe uno se crea) y se agrega el identificador del producto al carrito.

## 6. Realización de pedidos

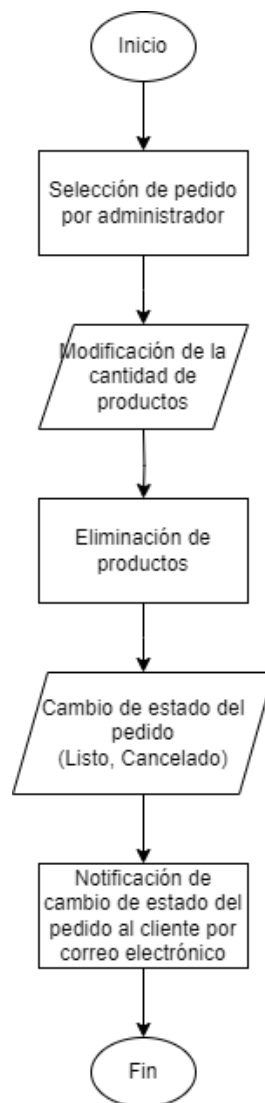
**Figura 9** Diagrama de flujo - Realizar pedidos



Al intentar realizar un pedido el sistema verifica la cantidad agregada al carrito con la existente en inventario, en caso de ser distinta, se elimina el excedente de productos y se notifica el detalle, después, se solicita una confirmación para realizar el pedido, en caso de ser aceptada se procede a crear el registro del pedido en la base de datos y se notifica al usuario mediante un correo electrónico.

## 7. Notificación de cambio de estado del pedido

**Figura 10** Diagrama de flujo - Notificación de cambio de estado de pedido



Un administrador puede modificar la cantidad de productos en el pedido en caso de que hubiera una confusión con la cantidad en bodega, después de eso puede actualizar el estado del pedido a listo cuando este pueda ser recogido o a cancelado en caso de que algo salga mal, independiente de la nueva condición el cliente recibe una notificación de cambio de estado por correo electrónico.

## 8. Proceso de CRUD de la tabla productos

Figura 11 Diagrama de flujo - CRUD de tabla de productos



El administrador selecciona que operación del “CRUD” desea realizar (crear, leer, actualizar o eliminar), en el caso de crear un nuevo producto debe escribir la información de este, después de una verificación se agrega los datos en la base y se notifica al usuario.

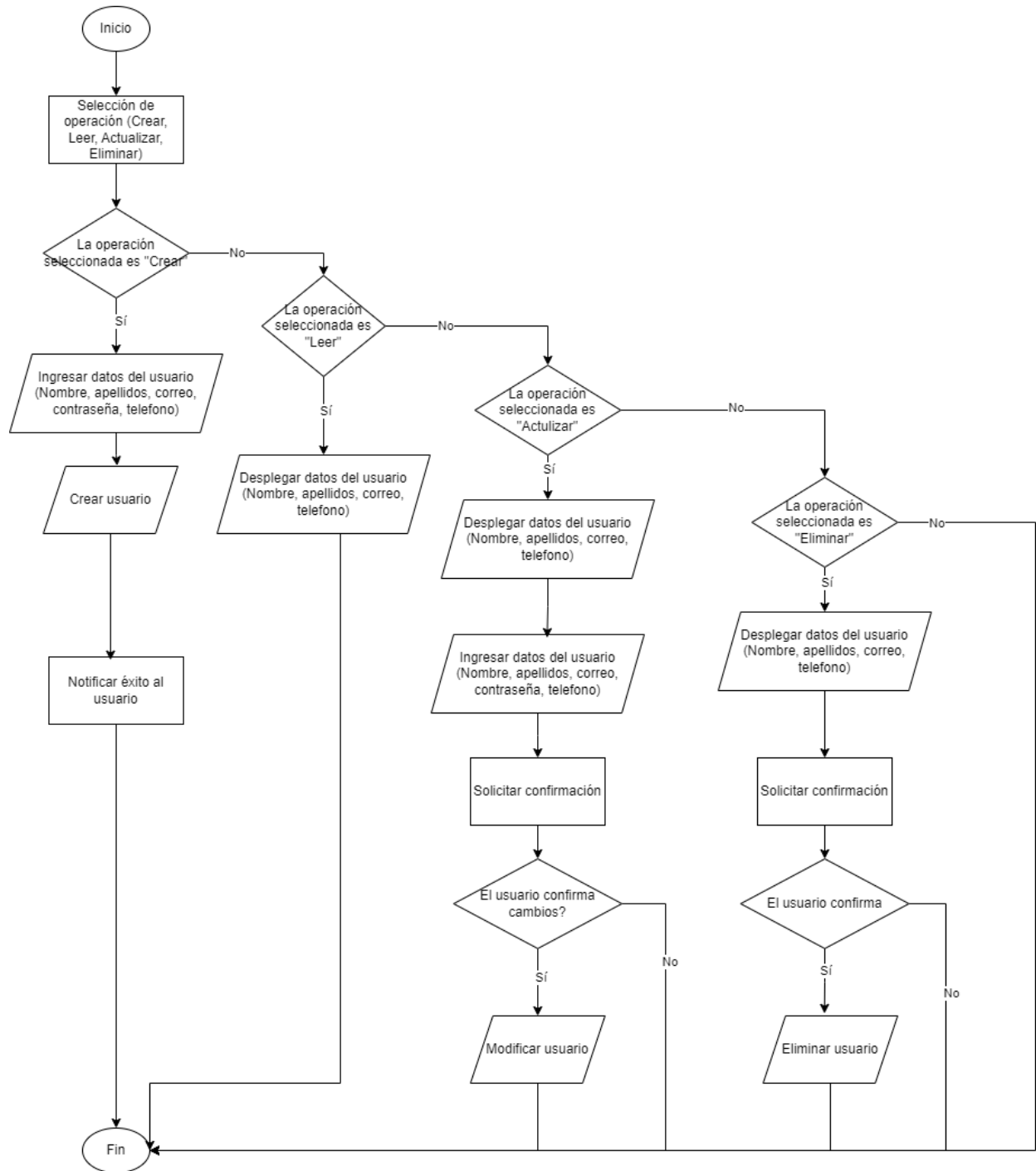
En caso de querer leer se desplegará los detalles de producto y el usuario podrá visualizarlo.

Para la operación de “Actualizar” igualmente se muestra el registro de la mercancía y el usuario puede cambiar los datos necesarios y, una vez confirmado, estos se actualizan en la base de datos.

Y, por último, la opción de eliminar permite, después de una aprobación, borrar del todo el registro del objeto seleccionado.

## **9. Proceso de CRUD de la tabla de usuarios**

**Figura 12** *Diagrama de flujo - CRUD de tabla usuarios*



El administrador selecciona que operación del “CRUD” desea realizar (crear, leer, actualizar o eliminar), en el caso de crear un nuevo usuario debe escribir la información de este, después de una verificación se agrega los datos en la base y se notifica al usuario.

En caso de querer leer se desplegará los detalles de usuario y estos se podrán visualizar.

Para la operación de “Actualizar” igualmente se muestra el registro del miembro y el usuario puede cambiar los datos necesarios y, una vez confirmado, estos se actualizan en la base de datos.

Y, por último, la opción de eliminar permite, después de una aprobación, borrar del todo el registro del objeto seleccionado.

## 10. Gestión de pedidos

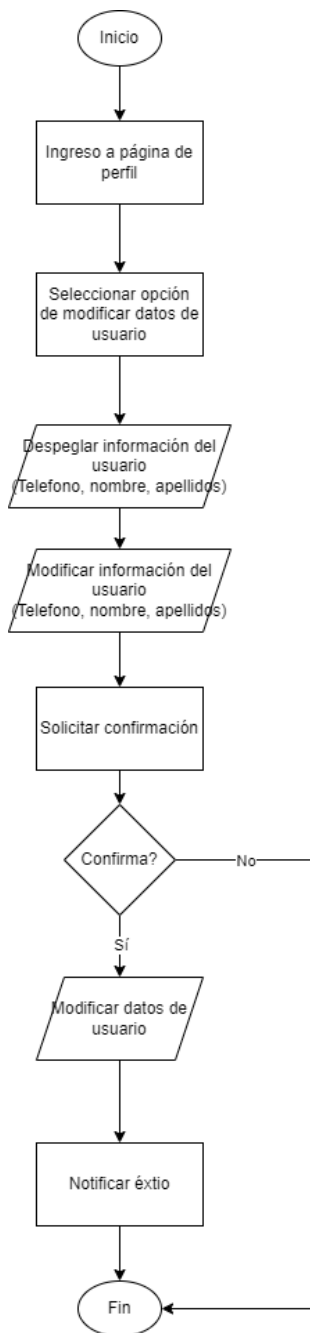
**Figura 13** Diagrama de flujo - Gestión de pedidos



Se solicita los pedidos registrados y estos se despliegan como una lista con los encargos con estado pendiente de primero.

## 11. Actualización de información de perfil

Figura 14 Diagrama de flujo - Actualizar información de perfil



Se muestra la información actual del usuario para que él decida que desea cambiar, al terminar de modificar y después de una confirmación, el registro se actualiza en la base de datos.

### 3.2.4 Diagrama de caso de uso

El diagrama de caso de uso nos ayuda a tener una referencia clara de cómo funciona el sistema y cómo interactúa con los usuarios.

Para crear un diagrama se deben definir los actores y los casos de uso, después mostrar de manera gráfica como interactúan estos mediante figuras para que sea sencillo de entender, además, se debe explicar cada caso de uso. Esta exposición debe mostrar datos como: el nombre del proceso, el nombre del autor, la fecha de creación, los actores, las precondiciones, el flujo normal y alternativo y las postcondiciones.

#### 3.2.4.1 Actores

Se pueden detectar dos actores principales.

1. **Cliente:** un cliente es cualquier usuario que visita la página y no tiene privilegios de administrador, ya sea que haya iniciado sesión o no. Crear una cuenta e ingresar les permite acceder a más funciones en la plataforma.
2. **Administrador:** son los usuarios que hayan iniciado sesión con una cuenta tipo administrador. Ellos pueden gestionar el inventario, visualizar datos de los usuarios y manejar los pedidos recibidos.

#### 3.2.4.2 Casos de uso

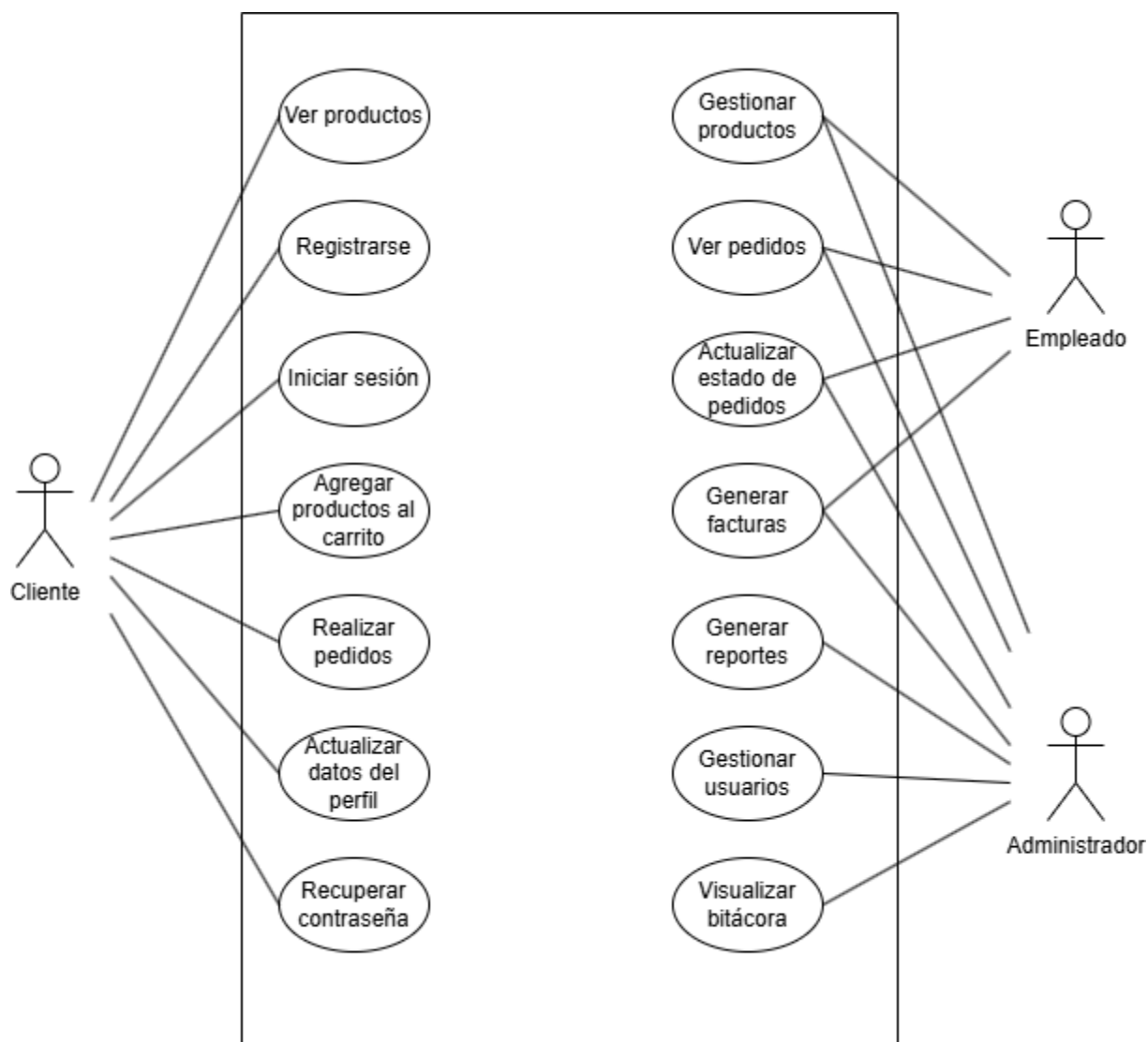
Un caso de uso se puede interpretar como la manera en la que un usuario interactúa con el sistema, cada actor suele tener sus propios casos.

1. **Ciente:** Ver productos, registrarse, iniciar sesión, agregar producto al carrito, realizar pedido, actualizar datos de perfil, recuperar contraseña.
2. **Administrador:** gestionar productos, gestionar usuarios, ver pedidos, actualizar estado de pedidos, notificar al cliente.

### 3.2.4.3 Diagrama

Al unir a los actores con los casos de uso podemos formar el siguiente diagrama

**Figura 15** Diagrama de caso de uso



Ahora se procede a explicar cada caso de uso:

**Tabla 3** Diagrama caso de uso - Visualizar productos

Nombre:	Visualizar productos
Autor:	Alejandro Salas C
Fecha:	15/07/2024
Descripción: permite a los clientes ver una lista de productos en venta y su disponibilidad.	
Actores: cliente	
Precondiciones: el cliente debe estar en la pestaña de productos.	
Flujo normal: 1. El cliente entra a la pestaña de productos. 2. El sistema muestra una lista de productos y su disponibilidad.	
Flujo alternativo: no hay flujo alternativo.	
Postcondiciones: no hay postcondiciones.	

**Tabla 4** Diagrama caso de uso - Registrarse

Nombre:	Registrarse
Autor:	Alejandro Salas C
Fecha:	15/07/2024
Descripción:	Permite a los clientes poder registrar un usuario en la base de datos.
Actores:	Cliente
Precondiciones:	El cliente no debe tener una cuenta, el cliente debe estar en la pestaña de registro
Flujo normal:	<ol style="list-style-type: none"> <li>1. El cliente ingresa su nombre, apellidos, correo electrónico, contraseña y teléfono.</li> <li>2. El sistema verifica los datos y crea una nueva cuenta si son válidos.</li> <li>3. El sistema notifica al cliente que la cuenta ha sido creada</li> </ol>
Flujo alternativo:	<ol style="list-style-type: none"> <li>2. Los datos ingresados no válidos, el sistema muestra un mensaje de error.</li> </ol>
Postcondiciones:	Se agrega un nuevo usuario a la base de datos.

**Tabla 5** Diagrama caso de uso - Iniciar sesión

Nombre:	Iniciar sesión
Autor:	Alejandro Salas C
Fecha:	15/07/2024
Descripción:	Permite a los clientes iniciar sesión.
Actores:	Cliente.
Precondiciones:	El cliente debe tener una cuenta y saber sus credenciales.
Flujo normal:	<ol style="list-style-type: none"> <li>1. El cliente accede a la página de inicio de sesión.</li> <li>2. El cliente ingresa su correo electrónico y su contraseña.</li> <li>3. El sistema verifica las credenciales y permite el acceso si son correctas.</li> </ol>
Flujo alternativo:	<ol style="list-style-type: none"> <li>3. Las credenciales son incorrectas, el sistema muestra un mensaje de error.</li> </ol>
Postcondiciones:	Se crea una nueva sesión en la base de datos.

**Tabla 6** Diagrama caso de uso - Agregar productos al carrito

Nombre:	Agregar productos al carrito
Autor:	Alejandro Salas C
Fecha:	15/07/2024
Descripción:	Permite a los clientes agregar productos al carrito de compras.
Actores:	Cliente.
Precondiciones:	El cliente debe haber iniciado sesión.
Flujo normal:	<ol style="list-style-type: none"> <li>1. El cliente selecciona un producto.</li> <li>2. El cliente selecciona agregar a el carrito.</li> <li>3. El sistema actualiza el carrito del cliente.</li> </ol>
Flujo alternativo:	No hay flujo alternativo.
Postcondiciones:	Se agrega un nuevo registro de carrito a la base de datos.

**Tabla 7** Diagrama caso de uso - Realizar pedido

Nombre:	Realizar pedido
Autor:	Alejandro Salas C
Fecha:	15/07/2024
Descripción:	Permite al cliente realizar un pedido para recoger en la tienda.
Actores:	Cliente.
Precondiciones:	El cliente debe haber iniciado sesión. El cliente debe tener al menos un artículo en el carrito.
Flujo normal:	<ol style="list-style-type: none"> <li>1. El cliente revisa su carrito.</li> <li>2. El sistema actualiza los productos en carrito según la disponibilidad actual.</li> <li>3. El cliente confirma el pedido.</li> <li>4. El sistema envía un correo de confirmación con el resumen del pedido al cliente.</li> <li>5. El sistema notifica al administrador sobre un nuevo pedido.</li> </ol>
Flujo alternativo:	<ol style="list-style-type: none"> <li>2. No hay ningún artículo disponible y el sistema no puede realizar el pedido.</li> </ol>
Postcondiciones:	Se crea un nuevo pedido en la base de datos.

**Tabla 8** Diagrama caso de uso - Cambiar datos de perfil

Nombre:	Cambiar datos de perfil
Autor:	Alejandro Salas C
Fecha:	15/07/2024
Descripción:	Permite a los clientes actualizar su información personal en su perfil.
Actores:	Cliente.
Precondiciones:	El cliente debe haber iniciado sesión.
Flujo normal:	<ol style="list-style-type: none"> <li>1. El cliente accede a la página de perfil.</li> <li>2. El cliente selecciona editar perfil.</li> <li>3. El cliente actualiza su información (nombre, apellidos, teléfono, contraseña).</li> <li>4. El cliente guarda los cambios.</li> <li>5. El sistema verifica y actualiza la información del perfil en la base de datos.</li> <li>6. El sistema notifica al cliente que los cambios se han realizado exitosamente.</li> </ol>
Flujo alternativo:	<ol style="list-style-type: none"> <li>3. Si los datos ingresados no son válidos, el sistema muestra un mensaje de error y no guarda los cambios.</li> </ol>
Postcondiciones:	La información del usuario se actualiza en la base de datos.

**Tabla 9** Diagrama caso de uso - Recuperar contraseña

Nombre:	Recuperar contraseña
Autor:	Alejandro Salas C
Fecha:	15/07/2024
<p>Descripción: Permite a los clientes recuperar su contraseña en caso de olvido.</p>	
<p>Actores: Cliente.</p>	
<p>Precondiciones: El cliente debe tener una cuenta registrada en el sistema.</p>	
<p>Flujo normal:</p> <ol style="list-style-type: none"> <li>1. El cliente accede a la página de inicio de sesión y selecciona la opción de “Olvidé mi contraseña”.</li> <li>2. El cliente ingresa a su correo electrónico registrado.</li> <li>3. El sistema verifica la existencia de una cuenta con ese correo electrónico.</li> <li>4. El sistema enviar un correo electrónico con un enlace para reestablecer la contraseña.</li> <li>5. El cliente abre el enlace.</li> <li>6. El cliente es redirigido a una página para ingresar una nueva contraseña.</li> <li>7. El cliente ingresa y confirma la contraseña.</li> <li>8. El sistema actualiza la contraseña.</li> <li>9. El sistema notifica al cliente que la contraseña se ha actualizado.</li> </ol>	
<p>Flujo alternativo:</p> <ol style="list-style-type: none"> <li>3. Si el correo digitado no presenta ninguna cuenta existente el sistema muestra un mensaje de error.</li> <li>7. Si la nueva contraseña no es válida el sistema muestra un mensaje de error.</li> </ol>	
<p>Postcondiciones: La contraseña del usuario es actualizada en la base de datos</p>	

**Tabla 10** Diagrama caso de uso - Gestionar productos

Nombre:	Gestionar productos
Autor:	Alejandro Salas C
Fecha:	15/07/2024
Descripción:	Permite al administrador realizar un CRUD en la tabla de productos.
Actores:	Empleado. Administrador.
Precondiciones:	El actor debe haber iniciado sesión.
Flujo normal:	<ol style="list-style-type: none"> <li>1. El actor accede a la página de gestión de productos.</li> <li>2. El actor puede crear un nuevo producto.</li> <li>3. El actor selecciona el producto con el que desea trabajar.</li> <li>4. El actor puede editar o eliminar el producto.</li> <li>5. El sistema actualiza la lista de productos en consecuencia.</li> </ol>
Flujo alternativo:	<ol style="list-style-type: none"> <li>2. Los datos ingresados no son válidos, el sistema muestra un mensaje de error.</li> </ol>
Postcondiciones:	La base de datos de productos se ve actualizada.

**Tabla 11** *Gestionar usuarios*

Nombre:	Gestionar usuarios
Autor:	Alejandro Salas C
Fecha:	15/07/2024
Descripción:	Permite al administrador realizar un CRUD en la tabla de usuarios.
Actores:	Administrador.
Precondiciones:	El administrador debe haber iniciado sesión.
Flujo normal:	<ol style="list-style-type: none"> <li>1. El administrador accede a la página de gestión de usuarios.</li> <li>2. El administrador puede crear un nuevo usuario.</li> <li>3. El administrador selecciona el usuario con el que desea trabajar.</li> <li>4. El administrador puede editar o eliminar el usuario.</li> <li>5. El sistema actualiza la lista de usuarios en consecuencia.</li> </ol>
Flujo alternativo:	<ol style="list-style-type: none"> <li>2. Los datos ingresados no son válidos, el sistema muestra un mensaje de error.</li> </ol>
Postcondiciones:	La base de datos de usuarios se ve actualizada.

**Tabla 12** Diagrama caso de uso - Ver pedidos

Nombre:	Ver pedidos
Autor:	Alejandro Salas C
Fecha:	15/07/2024
Descripción:	Permite al actor ver una lista de pedidos realizados.
Actores:	Empleado. Administrador.
Precondiciones:	El actor debe haber iniciado sesión. Debe haber al menos un pedido.
Flujo normal:	1. El actor accede a la página de pedidos. 2. El sistema muestra la lista de pedidos con sus estados teniendo como prioridad los pedidos en curso.
Flujo alternativo:	No hay flujo alternativo.
Postcondiciones:	No hay postcondiciones.

**Tabla 13** Diagrama caso de uso - Cambiar el estado del pedido

Nombre:	Cambiar el estado del pedido
Autor:	Alejandro Salas C
Fecha:	15/07/2024
Descripción:	Permite al actor actualizar el estado del pedido.
Actores:	Empleado. Administrador.
Precondiciones:	El actor debe haber iniciado sesión. Debe haber al menos un pedido.
Flujo normal:	<ol style="list-style-type: none"> <li>1. El actor accede a la página de pedidos.</li> <li>2. El sistema muestra la lista de pedidos con sus estados teniendo como prioridad los pedidos en curso.</li> <li>3. El actor selecciona un pedido.</li> <li>4. El actor actualiza el estado del pedido (pendiente, listo, cancelado).</li> <li>5. El sistema notifica al cliente sobre el cambio de estado.</li> </ol>
Flujo alternativo:	<ol style="list-style-type: none"> <li>4. Si el pedido no puede ser actualizado, el sistema muestra un mensaje de error.</li> </ol>
Postcondiciones:	Se actualiza el registro del pedido en la base de datos.

**Tabla 14** Diagrama de caso de uso - Generar facturas

Nombre:	Generar facturas
Autor:	Alejandro Salas C
Fecha:	15/07/2024
<p>Descripción: Permite crear una visualización de una factura existente y exportarla como un PDF.</p>	
<p>Actores: Empleado. Administrador.</p>	
<p>Precondiciones: El actor debe haber iniciado sesión. Deben existir registros de facturas.</p>	
<p>Flujo normal:</p> <ol style="list-style-type: none"> <li>1. El actor accede a la página de generar factura.</li> <li>2. El actor selecciona la factura la cual desea exportar.</li> <li>3. El actor verifica los datos en la factura.</li> <li>4. El actor selecciona una carpeta para guardar el archivo PDF.</li> </ol>	
<p>Flujo alternativo:</p>	
<p>Postcondiciones: Se crea un archivo PDF.</p>	

**Tabla 15** Diagrama de caso de uso - Generar reportes

Nombre:	Generar reportes
Autor:	Alejandro Salas C
Fecha:	15/07/2024
Descripción:	Permite al administrador visualizar distintos reportes y expórtalos en PDF. Estos reportes son: diarios, mensuales, anuales, históricos.
Actores:	Administrador.
Precondiciones:	El administrador debe haber iniciado sesión.
Flujo normal:	<ol style="list-style-type: none"> <li>1. El administrador accede a la página de generación de reportes.</li> <li>2. El administrador selecciona el tipo de reporte a generar.</li> <li>3. El administrador selecciona la fecha a consultar.</li> <li>4. El sistema muestra un reporte de la fecha selecciona y una ventana para imprimir el reporte en PDF.</li> <li>5. El administrador selecciona una carpeta para guardar el archivo PDF.</li> </ol>
Flujo alternativo:	<ol style="list-style-type: none"> <li>4. Los datos ingresados no son válidos, el sistema muestra un mensaje de error.</li> <li>4. No existen pedidos en el rango de fechas seleccionadas, el sistema muestra un mensaje de error.</li> </ol>
Postcondiciones:	Se genera un archivo PDF.

**Tabla 16** Diagrama de caso de uso - Visualizar bitácora

Nombre:	Visualizar bitácora
Autor:	Alejandro Salas C
Fecha:	15/07/2024
Descripción:	Permite al administrador visualizar una tabla con la información de los registros de la bitácora.
Actores:	Administrador.
Precondiciones:	El administrador debe haber iniciado sesión.
Flujo normal:	<ol style="list-style-type: none"> <li>1. El administrador accede a la página de visualizar bitácora.</li> <li>2. El sistema genera una tabla con la información de los registros en la tabla de bitácora.</li> <li>3. El administrador observa los registros en una tabla organizada.</li> </ol>
Flujo alternativo:	
Postcondiciones:	

### 3.2.5 Tecnologías y estilo

Es importante plantear las tecnologías que se utilizarán para desarrollar la aplicación, hay que buscar que estas cumplan con los requerimientos del sistema y que faciliten la programación.

Con esto en mente se escogieron las siguientes tecnologías:

**ASP.NET Core Web App MVC .NET 8.0:** se utiliza para construir aplicaciones web empleando el patrón de diseño MVC (Model -View - Controller). Es completamente modular, lo que significa que se pueden incluir solo los componentes necesarios, lo que resulta en aplicaciones más ligeras, escalables y eficientes.

NET 8.0 representa la versión del marco de trabajo siendo una de las últimas versiones lanzadas, este vino con mayor rendimiento, posibilidad de utilizar Blazor, consultas de Lazy Loading, entre otras cosas.

ASP.NET Core es la plataforma de desarrollo, tiene muchas ventajas como ser modular y más ligero, además cuenta con integraciones a ciertas tecnologías, se puede utilizar código tipo Razor y es una de las plataformas más rápidas de desarrollo web.

El patrón de diseño MVC es un diseño arquitectónico ampliamente utilizado en el desarrollo de software, divide la aplicación en tres partes las cuales cumplen distintas funciones, esto facilita su mantenimiento de código y la escalabilidad del proyecto:

**Model:** es la parte lógica de los datos de la aplicación, suelen estar vinculados con la base de datos donde interactúan con las entidades y manejan la persistencia de datos, esto se logra mediante clases de C# las cuales representan las tablas.

**View:** gestiona la presentación y las interfaces del usuario, suelen utilizar Razor el cual es un motor de vista que permite combinar código C# con HTML de manera sencilla, normalmente no contienen lógica del negocio, solo se encarga de mostrar información al usuario.

**Controller:** procesa las solicitudes del usuario e interactúa con las vistas y el modelo, gestiona la lógica de la página, al recibir una petición interactúa con el modelo para obtener los datos necesarios y luego selecciona la vista para presentarla al usuario.

Se eligió esta tecnología debido a que cumple con las necesidades para la aplicación, es una tecnología con mucha documentación y una comunidad activa lo cual garantiza un acceso a recursos para resolver problemas y, también, es el lenguaje que más domino en la actualidad se traduce en un desarrollo más rápido y eficiente, si bien tecnologías como NodeJS ofrecen mayor velocidad, esta opción ofrece más rendimiento en aplicaciones de esta escala.

#### **Ventajas:**

1. El diseño dividido de MVC mantiene ordenado el código facilitando su creación y mantenimiento.
2. Ayuda a tener URLs más ordenadas.
3. Se pueden utilizar tecnologías como Entity Framework Core, Blazor y Razor.
4. Hay abundante documentación en línea en caso de ser necesaria una investigación.

#### **Desventajas:**

1. Se necesitan muchas configuraciones para los inicios de los proyectos.
2. Algunos aspectos como la interactividad avanzada o lograr ciertas funcionalidades pueden requerir un mayor esfuerzo.

#### **Microsoft SQL Server 2022:**

Microsoft SQL Server es un sistema de gestión de base la cual puede almacenar y administrar grandes cantidades de datos de forma segura y eficiente, además posee sistemas de autenticación, cifrado y auditoría.

Es muy utilizadas en pequeñas y medianas empresas en sistemas de gestión y en sitios web, además de todas estas ventajas se eligió principalmente debido a su integración con tecnologías como.NET y Visual Studio. Su combinación de seguridad, rendimiento, escalabilidad y fácil integración con herramientas de desarrollo hace que sea una opción preferida.

Cuenta con la posibilidad de utilizar SQL Server Management Studio la cual es una herramienta de administración de SQL Server que proporciona una interfaz gráfica sencilla e intuitiva reduciendo la cantidad de comandos necesarios para realizar el banco de datos, permite analizar los tiempos de respuesta, generar copias de seguridad y cuenta con editor de consultas y autocompletado para facilitar el trabajo.

#### **Ventajas:**

1. La integración con ASP.NET y otros productos de Microsoft lo convierte en una opción interesante.
2. La interfaz gráfica de permite gestionar la base de datos de manera sencilla acelerando la creación de esta.
3. Al ser un sistema muy utilizado se cuenta con mucha documentación en caso de ser necesaria.

#### **Desventajas:**

1. Puede ser muy exigente en temas de recursos del servidor.

2. Se requiere amplio conocimiento de la tecnología para usarlo de manera correcta.

### **Bootstrap v5.1.0:**

Es un framework front-end el cual se utiliza en el desarrollo de sitios web, es modular, ligero, ayuda a crear diseños responsivos, lo cual es importante para este proyecto, posee clases utilitarias como “mt-{n}” las cuales permiten aplicar estilos directamente en HTML sin tener que crear un CCS adicional e incluye una amplia variedad de componentes como botones, formularios, modales entre otros.

Además, el sistema de Flexbox y CSS permiten crear sistemas que se adapten a las pantallas de distintos tamaños y como los componentes son estilizados por defecto se garantiza una apariencia coherente en todo el proyecto.

Bootstrap será utilizado en elementos simples como botones o en la barra de navegación, además se utilizarán las clases utilitarias, lo cual lo hace una excelente opción.

### **Ventajas:**

1. Acelera el desarrollo evitando tener que crear estilos desde cero para cada componente
2. Permite el uso de clases utilitarias las cuales ayudan a mantener el código limpio.
3. Garantiza el diseño responsivo de la página sin esfuerzo adicional.

### **Desventajas:**

1. Las interfases creadas con Bootstrap pueden tener un estilo similar dando sensación de producto genérico.
2. Se requiere conocimiento extra para poder utilizar Bootstrap de manera adecuada.

### **Google Fonts:**

Permite utilizar fuentes personalizadas las cuales no están de manera predeterminada en los sistemas, mejorando el diseño y la identidad de la marca.

Para el proyecto se espera utilizar la fuente “Work Sans” cargándola desde Google Fonts, se aplicará en toda la página brindando una mejor calidad grafica.

Este servicio es muy utilizado por su accesibilidad, facilidad de integración y por ser un recurso optimizado para la web. Las fuentes de Google Fonts son de código abierto, lo que permite su uso sin restricciones comerciales ni licencias costosas.

### **Ventajas:**

1. Se puede agregar fácilmente al proyecto con una línea de código.
2. Posee soporte para múltiples idiomas.
3. Funciona en la mayoría de los navegadores modernos.
4. Están optimizadas para tener el mínimo impacto en el rendimiento del sitio web.

### **Desventajas:**

1. Requiere acceso a internet, en caso de que un usuario tenga problemas de conectividad obliga al navegador a utilizar fuentes predeterminadas.

Con estas tecnologías se espera cumplir con todas las necesidades, ahorrar tiempo en el desarrollo y mejorar el diseño del sitio aumentando la sensación de calidad.

### **Servicio en la nube de Microsoft Azure:**

Para poder implementar el servicio para que todos los usuarios puedan acceder se utilizará la plataforma de servicios en la nube de Microsoft Azure ya que cuenta con una infraestructura estable e integraciones con las otras tecnologías seleccionadas.

Mediante esta herramienta podremos migrar la base de datos local creada para el desarrollo y utilizarla en la aplicación, además, se puede exportar la solución del proyecto para garantizar su publicación en la web.

### **Ventajas:**

1. Posee una amplia gama de servicios lo cual permite administrar toda la infraestructura con una misma aplicación.
2. Se integra fácilmente con otras herramientas de Microsoft como Visual Studio y SQL Server.
3. Se paga solamente por los recursos utilizados, lo cual permite reducir gastos en sitios web que no son muy concurridos.

### **Desventajas:**

1. Es un programa complejo de utilizar con diversas opciones por lo cual requiere un conocimiento avanzado para poder utilizarlo.
2. Aunque el modelo de pago sea ventajoso, igualmente es un servicio de pago.

3. Depende de los servidores de Microsoft, en caso de que estos presenten problemas afectarían directamente a la aplicación.

### **3.3 Resultados**

Con todos estos componentes listos podemos comenzar con la creación del sistema, todos los elementos guiarán el desarrollo de la página web, esto ayudará a visualizar la estructura del proyecto facilitando la comprensión de la interacción de los módulos y alinear las expectativas del proyecto para que todas las partes estén en la misma página.

### **3.4 Análisis y discusión**

Se ha creado una estructura sólida para empezar a trabajar, se espera respetar la mayoría del diseño creado para respetar la concordancia, pero el proyecto aún puede cambiar en el desarrollo.

Los diagramas obtenidos muestran que el proyecto es viable, siguiendo la guía se logrará que el desarrollo avance sin inconvenientes y en caso de que hubiera uno, con unos ligeros cambios, se podrán resolver.

## **4. Capítulo 4: Objetivo 3**

### **4.1 Preparación y primeros pasos**

Para comenzar a trabajar debemos descargar una aplicación que nos sirva como ambiente de desarrollo, se utilizará Visual Studio 2022, este cuenta con herramientas que ayudan con la creación, depuración y mantenimiento de aplicaciones, esto nos ayudará a mejorar la productividad.

El primer paso es crear la base de datos, la cual, almacena la información de nuestro programa, con SQL Server Management Studio 19 podemos realizar consultas directamente para facilitar la creación de esta.

Mediante el comando “create database” podemos crear el banco de datos, para este proyecto le asignaremos el nombre “SuperAlfoliDB”, después debemos ingresar para poder crear las tablas; basándonos en el diagrama de entidad relación, podemos utilizar el “create table” para estructurar cada una de las tablas necesarias, hay que incluir los campos requeridos y su tipo de dato a almacenar, es importante que los campos identificadores sean “identity (1,1)” (1,1 representa que el campo incrementa de 1 en 1) y que sea “primary key”, además, los campos de llaves foráneas deben utilizar el “foreign key references Tabla(Campo)” para especificar a que tabla hacen referencia. No se utilizará la tabla de sesión ya que el manejo de sesiones se manejará con una librería de ASP.NET.

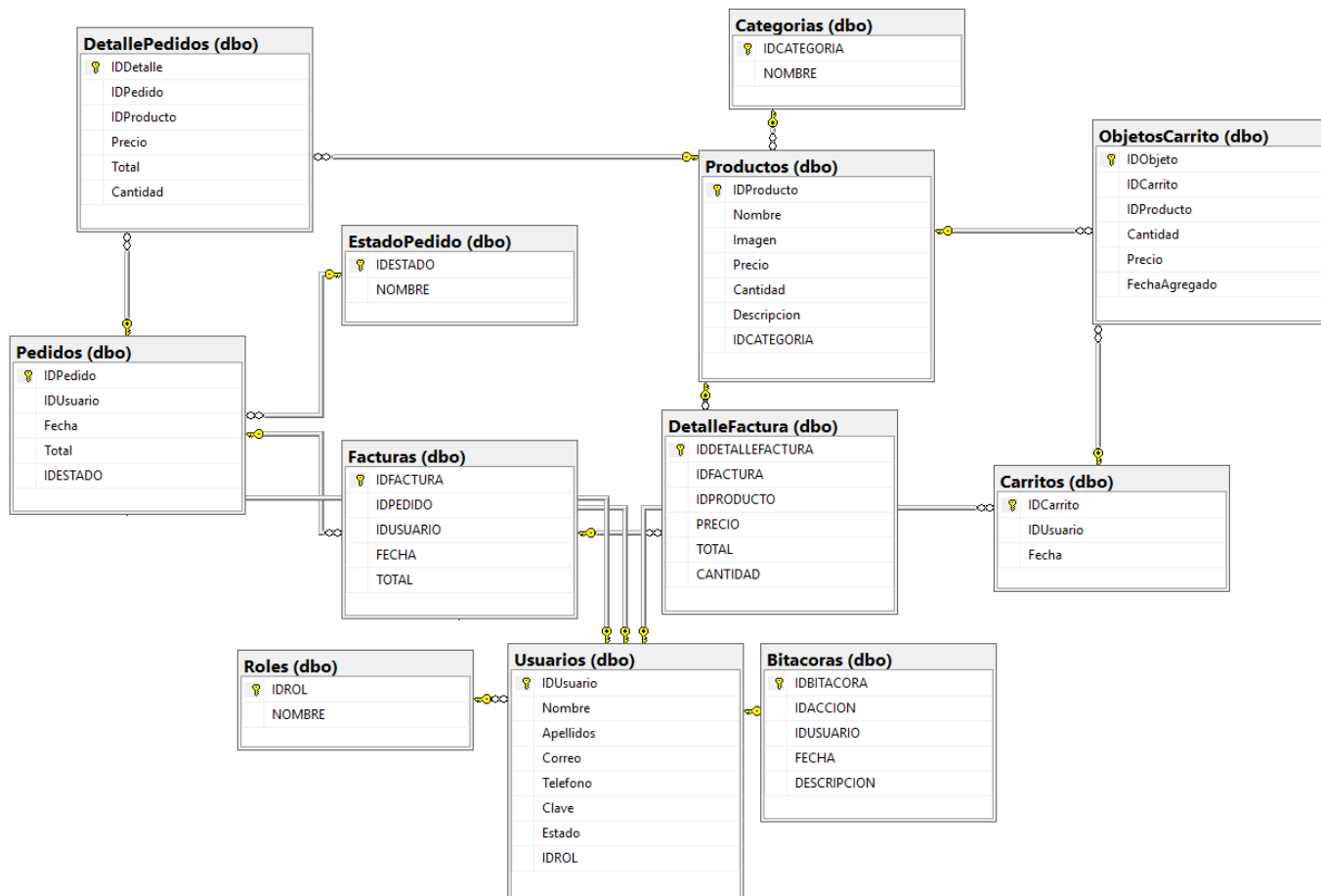
**Figura 16** *Creación de base de datos*

```

create table Usuarios (
  IDUsuario int identity(1,1) Primary key,
  Nombre varchar(30) not null,
  Apellidos varchar(30) not null,
  Correo varchar(30) not null,
  Telefono int not null,
  Clave varchar(300) not null,
  Rol varchar(20),
  Estado bit default '1'
);
create table Productos(
  IDProducto int identity(1,1) primary key,
  Nombre varchar(100) not null,
  Imagen varbinary(max),
  Categoria varchar(100),
  Precio money not null,
  Cantidad int not null,
  Descripcion varchar(300)
);
create table Pedidos(
  IDPedido int identity (1,1) primary key,
  IDUsuario int not null foreign key references Usuarios(IDUsuario),
  Estado varchar(30) not null,
  Fecha datetime default getdate(),
  Total money not null
);
create table DetallePedidos(
  IDDetalle int identity(1,1) primary key,
  IDPedido int not null foreign key references Pedidos(IDPedido),
  IDProducto int not null foreign key references Productos(IDProducto),
  Precio money not null,
  Total money not null,
  Cantidad int not null
)
create table Carritos(
  IDCarrito int identity (1,1) primary key,
  IDUsuario int not null foreign key references Usuarios(IDUsuario),
  Fecha datetime default getdate()
);
create table ObjetosCarrito(
  IDObjeto int identity(1,1) primary key,
  IDCarrito int not null foreign key references Carritos(IDCarrito),
  IDProducto int not null foreign key references Productos(IDProducto),
  Cantidad int,
  Precio money not null,
  FechaAgregado datetime Default getdate()
)

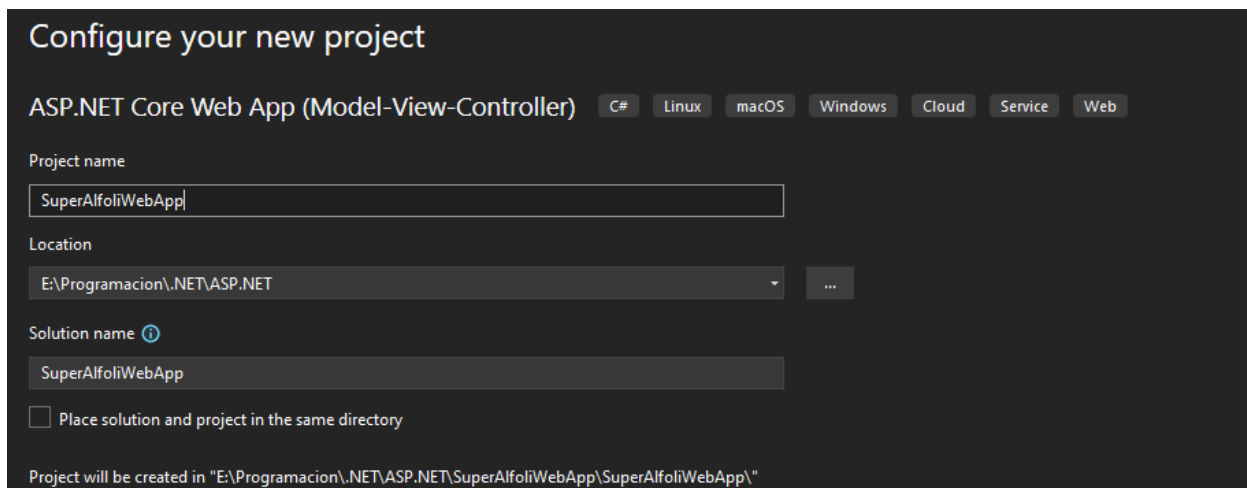
```

**Figura 17** Diagrama final de la base de datos



Ahora en Visual Studio debemos crear un proyecto, este debe ser con la plantilla de “ASP.NET Core Web App (Model – View - Controller)” y le ponemos como nombre “SuperAlfoliWebApp”.

**Figura 18** Creación del proyecto



En el proyecto debemos realizar la línea de conexión a la base de datos, esta se realiza en el archivo “appsettings.json” y debe estar compuesta del: nombre del servidor, el nombre de la base de datos y el tipo de conexión; es importante mencionar que hay que cambiar este texto cuando se suba a la nube.

Debemos agregar en el archivo `Program.cs` un “builder.Services.AddDbContext”, el cual nos ayuda a crear el contexto de la conexión a la base de datos, para ello debemos enviar como parámetro el objeto json creado anteriormente.

Ahora debemos instalar el paquete de EntityFramework con el cual podremos crear los modelos utilizando nuestra base de datos como apoyo, debemos escribir en la consola de administración de paquetes el siguiente comando “Scaffold-DbContext “Server=NombreServer Database=NombreBaseDeDatos Trusted\_Connection=True” Microsoft.EntityFrameworkCore.SqlServer -OutputDir Models” esto nos permite crear los modelos que hacen referencia a cada tabla en nuestra base de datos.

**Figura 19** *Agregar paquete de EntityFramework*

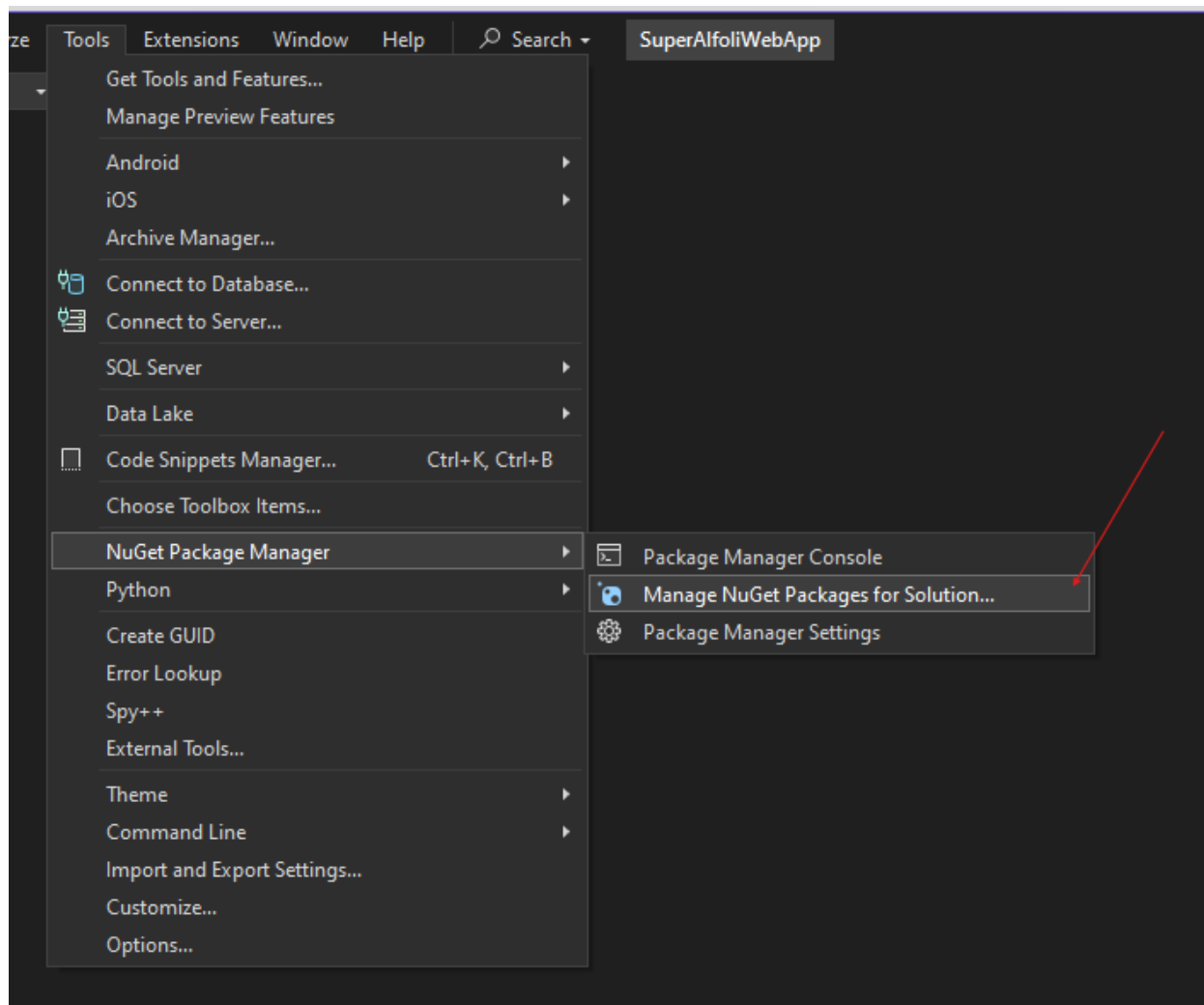
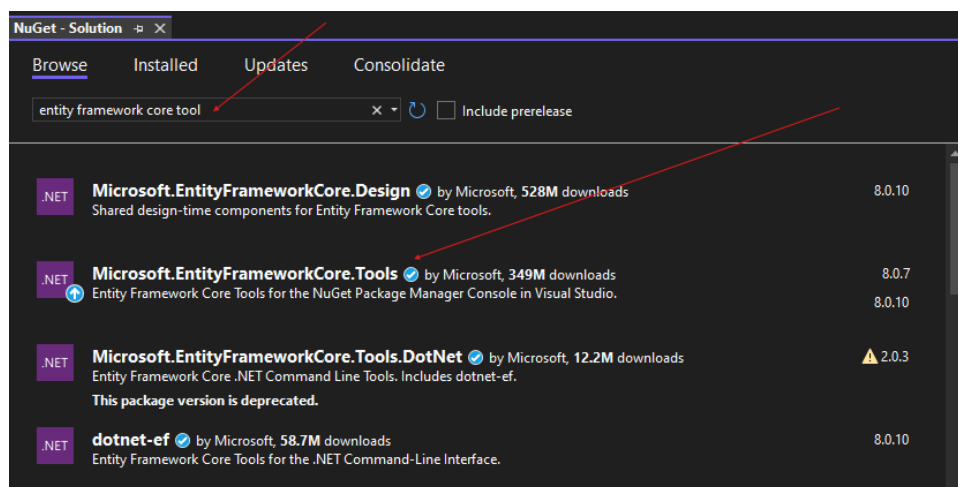


Figura 20 Agregar paquete de EntityFramework 2



**Figura 21** Creación de la cadena de conexión en "appsettings.json"

```

9  "ConnectionStrings": {
10     "DefaultConnection": "Server=ALETZ-PC\\SQLEXPRESS;Database=SuperAlfoliDB;Trusted_Connection=True;TrustServerCertificate=True;"
11  },
12

```

**Figura 22** Creación del Builder Services en el "Program.cs"

```

15  builder.Services.AddDbContext<SuperAlfoliDbContext>(options =>
16  {
17      options.UseSqlServer(builder.Configuration.GetConnectionString("DefaultConnection")); ;
18  });
19

```

**Figura 23** Generar modelos mediante la estructura de la base de datos

```

Package Manager Console
Package source: All Default project: SuperAlfoliWebApp

PM> "Scaffold-DbContext "Server=ALETZ-PC\\SQLEXPRESS Database=SuperAlfoliDB Trusted_Connection=True"
Microsoft.EntityFrameworkCore.SqlServer -OutputDir Models" |

```

Con esto listo tenemos el proyecto con una conexión a la base de datos mediante el DbContext y una carpeta con los modelos que hacen referencia a cada tabla.

**Figura 24** Carpeta con los modelos creados

```

Models
├── C# Carrito.cs
├── C# DetallePedido.cs
├── C# ErrorViewModel.cs
├── C# ObjetosCarrito.cs
├── C# Pedido.cs
├── C# Producto.cs
├── C# SuperAlfoliDbContext.cs
└── C# Usuario.cs

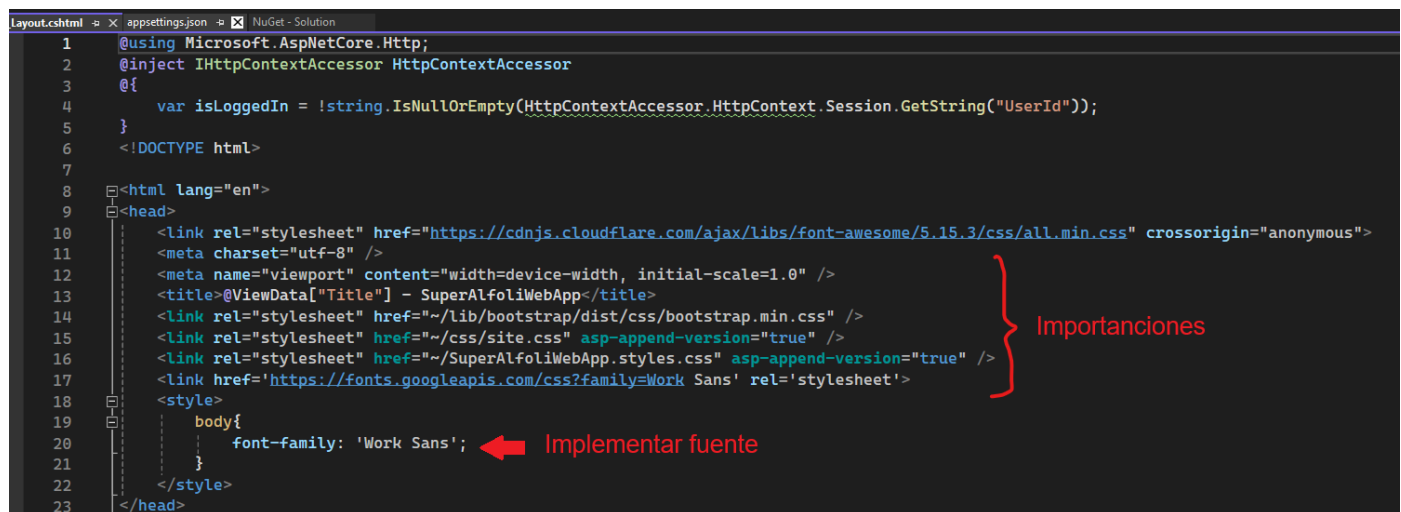
```

## 4.2 Apartado del cliente

Ahora modificamos el "Layout.cshtml" el cual contiene la barra de navegación, e importa los archivos requeridos para que funcione la página (hojas de estilo, fuentes de

Google Fonts, Scripts, entre otros), cambiamos las opciones predeterminadas y agregamos los hipervínculos necesarios (Inicio, Lista de productos, Cuenta, Carrito), además implementamos una función que verifica si el usuario ha iniciado sesión, si es el caso mostrará el botón para acceder al carrito

**Figura 25** Modificación del "Layout.cshtml" Incluir importaciones



```

1  @using Microsoft.AspNetCore.Http;
2  @inject IHttpContextAccessor HttpContextAccessor
3  @{
4      var isLoggedIn = !string.IsNullOrEmpty(HttpContextAccessor.HttpContext.Session.GetString("UserId"));
5  }
6  <!DOCTYPE html>
7
8  <html lang="en">
9  <head>
10     <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/font-awesome@5.15.3/css/all.min.css" crossorigin="anonymous">
11     <meta charset="utf-8" />
12     <meta name="viewport" content="width=device-width, initial-scale=1.0" />
13     <title>@ViewData["Title"] - SuperAlfoliWebApp</title>
14     <link rel="stylesheet" href="~/lib/bootstrap/dist/css/bootstrap.min.css" />
15     <link rel="stylesheet" href="~/css/site.css" asp-append-version="true" />
16     <link rel="stylesheet" href="~/SuperAlfoliWebApp.styles.css" asp-append-version="true" />
17     <link href='https://fonts.googleapis.com/css?family=Work Sans' rel='stylesheet'>
18     <style>
19         body{
20             font-family: 'Work Sans';
21         }
22     </style>
23 </head>

```

**Figura 26** Modificación del "Layout.cshtml" Agregar botones de navegación

```

<div class="navbar-collapse collapse d-sm-inline-flex justify-content-center height65px">
  <ul class="navbar-nav height65px">
    <li class="nav-item">
      <a class="nav-link navbar-link-custom" asp-area="" asp-controller="Home" asp-action="Index">INICIO</a>
    </li>
    <li class="nav-item">
      <a class="nav-link navbar-link-custom" asp-area="" asp-controller="Producto" asp-action="ListaProductos">PRODUCTOS</a>
    </li>
    <li class="nav-item">
      @if (isLoggedIn)
      {
        <a class="nav-link navbar-link-custom" asp-area="" asp-controller="Account" asp-action="Details">CUENTA</a>
      }
      else
      {
        <a class="nav-link navbar-link-custom" href="#" data-bs-toggle="modal" data-bs-target="#loginModal">CUENTA</a>
      }
    </li>
    <li class="nav-item">
      @if (isLoggedIn)
      {
        <a class="nav-link navbar-link-custom" asp-area="" asp-controller="Carrito" asp-action="VerCarrito">
          <svg xmlns="http://www.w3.org/2000/svg" width="16" height="16" fill="currentColor" class="bi bi-cart" viewBox="0 0 16 16">
            <path d="M0 1.5A.5.5 0 0 1 .5 1.5H2a.5.5 0 0 1 .485 3.79L2.89 3H14.5a.5.5 0 0 1 .491 5.92L1.5 8A.5.5 0 0 1 1.3 12H4a.5.5 0 0 1 1.5 0" />
          </svg> CARRITO
        </a>
      }
      else
      {
      }
    </li>
  </ul>
</div>

```

Para modificar ciertos estilos fijos o agregar algunos necesarios se crea el archivo “site.css” ubicado en la carpeta css en la raíz del proyecto, aquí se añadirán configuraciones para los elementos que deseamos cambiar.

Es importante mencionar que se necesita dos layout en el proyecto, una para usuarios clientes y otra para administradores, los botones del administrador son distintos y dan acceso a opciones como el inicio de administrador, gestión de la tabla productos, gestión de la tabla de usuarios y un apartado para ver los pedidos realizados.

**Figura 27** Botones de navegación en “AdminLayout.cshtml”

```

<div class="navbar-collapse collapse d-sm-inline-flex justify-content-center height65px">
  <ul class="navbar-nav height65px">
    <li class="nav-item">
      <a class="nav-link navbar-link-custom" asp-area="" asp-controller="Admin" asp-action="Index">INICIO</a>
    </li>
    <li class="nav-item">
      <a class="nav-link navbar-link-custom" asp-area="" asp-controller="Producto" asp-action="Index">PRODUCTOS</a>
    </li>
    <li class="nav-item">
      <a class="nav-link navbar-link-custom" asp-area="" asp-controller="Usuario" asp-action="Index">USUARIOS</a>
    </li>
    <li class="nav-item">
      <a class="nav-link navbar-link-custom" asp-area="" asp-controller="Pedidos" asp-action="VerPedidos">PEDIDOS</a>
    </li>
  </ul>
</div>

```

La página se divide en dos partes, una en donde los clientes podrán ver una lista de productos e información del local y al iniciar sesión poder crear pedidos de tipo recolección al local, y otra que, si al ingresar al sistema, la cuenta es de tipo administrador, el usuario podrá realizar funciones típicas de gestión de la página como las mencionadas anteriormente, para verificar el tipo de cuenta actual se lleva registro de la sesión mediante el bloque de configuración de ASP.NET Core el cual permite almacenar y mantener datos temporales, este se coloca en el archivo "Program.cs", funciona mediante la creación de una cookie de sesión y le asocia valores como, el tiempo de inactividad, si es necesario las cookies y los datos a guardar, en cada solicitud la cookie se envía al servidor permitiendo que se acceda a los datos almacenados.

**Figura 28** Configuración de los servicios de sesiones de ASP.NET Core

```
builder.Services.AddSession(options =>
{
    options.IdleTimeout = TimeSpan.FromMinutes(30);
    options.Cookie.HttpOnly = true;
    options.Cookie.IsEssential = true;
});
```

En la pantalla de inicio se escribió un texto descriptivo para que el navegante entienda el contexto de la empresa, hay un marco donde se puede ver la ubicación del local en el mapa y un apartado donde se muestra los horarios de atención y el teléfono.

**Figura 29** Pantalla de inicio


[INICIO](#)
[PRODUCTOS](#)
[CUENTA](#)


### Quiénes somos

Somos super mercado que busca ofrecer productos de calidad, un servicio excepcional y un ambiente cálido para ser el preferido de nuestra comunidad. Brindamos a nuestros clientes una experiencia de compra conveniente y confiable ofreciendo una amplia variedad de productos frescos y de alta calidad a precios justos con un enfoque en el servicio cercano y amigable.

### Información

Estamos ubicados en Río Segundo de Alajuela.

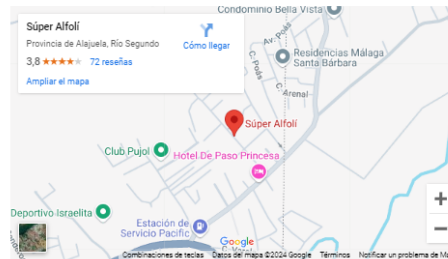
### Horarios

L - S 6:00am a 12:30pm / 1:30pm a 7:00pm

D 6:00am a 12:00pm

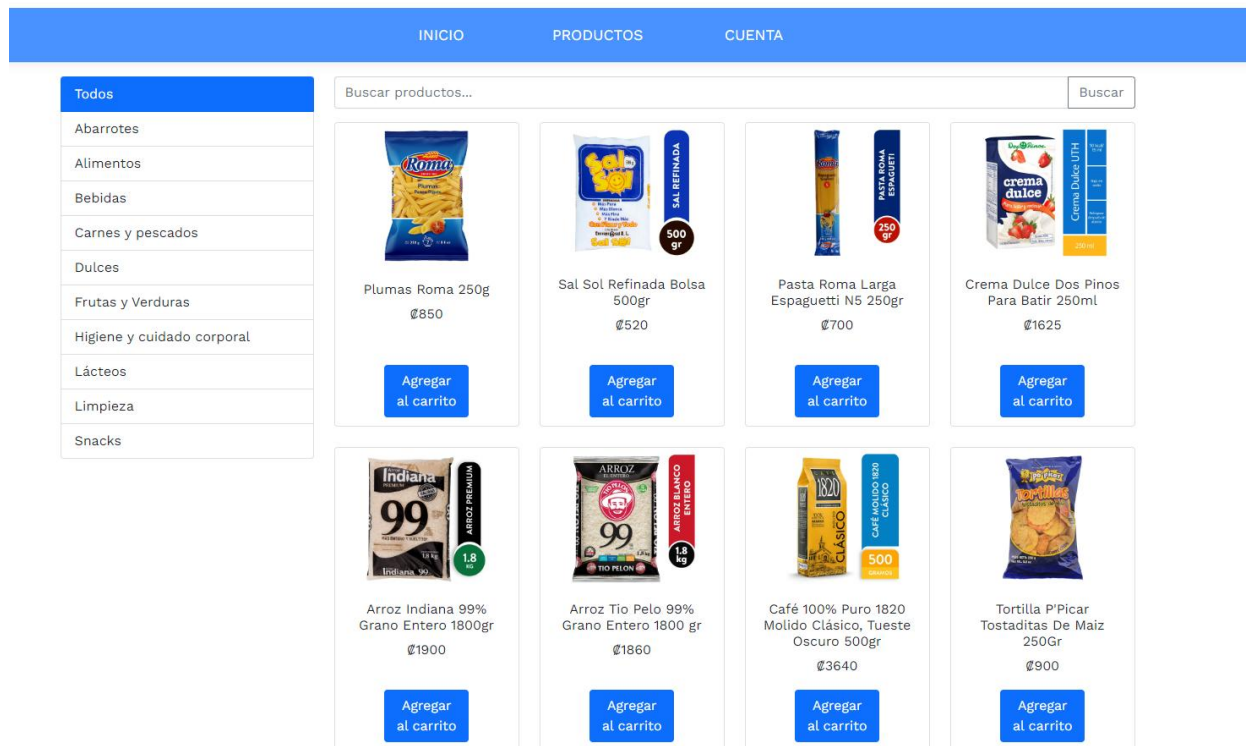
### Teléfono

2442 6856



En el apartado de “PRODUCTOS” se ve una lista de todos los productos disponibles con un botón de “Agregar al carrito”, se puede filtrar los resultados por categorías, o buscar por nombre, al intentar agregar al carrito solicita al cliente iniciar sesión para poder continuar; los clientes autenticados podrán agregar artículos a su carrito digitando la cantidad deseada.

**Figura 30** *Página de productos*



Para poder desplegar la lista de los productos usaremos el “\_context” la cual es una instancia del contexto de la base de datos, esta se utiliza para interactuar con ella directamente, para esto es que creamos el “DbContext” y los modelos, mediante la cadena de conexión este se conecta a la base de datos, cuando se llama a los métodos correspondientes, el Entity Framework Core construye los SQL correspondientes a la consulta generada y los ejecuta, esto nos facilita las tareas de CRUD (Crear, Leer, Actualizar y Eliminar) y nos ayuda a manejar la gestión de cambios; esta clase será muy usada a lo largo del proyecto cada vez que necesitemos realizar una consulta a la base de datos.

**Figura 31** Configuración del DbContext en el "Program.cs"

```
builder.Services.AddDbContext<SuperAlfolidbContext>(options =>
{
    options.UseSqlServer(builder.Configuration.GetConnectionString("DefaultConnection")); ;
});
```

**Figura 32** Ejemplo de declaración del DbContext en una clase controladora

```

namespace SuperAlfoLiWebApp.Controllers
{
    1 reference
    public class ProductoController : Controller
    {
        private readonly SuperAlfoLiDbContext _context;

        0 references
        public ProductoController(SuperAlfoLiDbContext context)
        {
            _context = context;
        }
    }
}

```

Declaración del DbContext en el controlador

Con lo anterior en mente solo debemos solicitar todos los registros de los productos y enviarlos al “view” para darles un formato; pero, debido a que en la página se puede filtrar por categoría o por búsqueda, hay que agregar otros procesos en caso de que se haya optado por una de estas opciones. Se crea una variable que almacena la consulta que se vaya a realizar y se pasa por un grupo de consultas, si existe una categoría seleccionada el valor será la solicitud de todos los artículos correspondientes, si no, verifica si se ha escrito un término de búsqueda, si es así, busca productos donde el nombre contenga la cadena escrita; se crea una función llamada “GetImage” que nos ayuda a transformar la imagen de un bitmap a una foto completa.

**Figura 33** Método para desplegar lista de productos verificando si existe algún filtro por categoría o nombre

```

public async Task<IActionResult> ListaDeProductos(string categoria, string terminoBusqueda, int pagina = 1)
{
    int tamanoPagina = 12;
    var consulta = _context.Productos.AsQueryable();

    if (!string.IsNullOrEmpty(categoria))
    {
        consulta = consulta.Where(p => p.Categoria == categoria);
    }

    if (!string.IsNullOrEmpty(terminoBusqueda))
    {
        consulta = consulta.Where(p => p.Nombre.ToLower().Contains(terminoBusqueda.ToLower()));
    }

    var productos = await consulta.Skip((pagina - 1) * tamanoPagina).Take(tamanoPagina).ToListAsync();
    var categorias = await _context.Productos.Select(p => p.Categoria).Distinct().ToListAsync();

    ViewData["Categorias"] = categorias;

    return View(new ListaDeProductosViewModel
    {
        Productos = productos,
        CategoriaActual = categoria,
        PaginaActual = pagina,
        TotalProductos = await consulta.CountAsync(),
        TerminoBusqueda = terminoBusqueda
    });
}

```

Figura 34 Método "GetImage"

```

0 references
public async Task<IActionResult> GetImage(int id)
{
    var product = await _context.Productos.FindAsync(id);
    if (product == null || product.Imagen == null)
    {
        return NotFound();
    }

    return File(product.Imagen, "image/jpeg");
}

```

En la parte de "CUENTA" permite iniciar sesión utilizando las credenciales de la cuenta, en caso de que no recuerde la contraseña podrá recuperarla accediendo al vínculo "Recuperar contraseña" en la cual al ingresar el correo se envía un enlace de recuperación para poder cambiar la clave. O en caso de que no se posea una cuenta pueda crear una con unos simples pasos. Al ingresar al sistema se puede acceder al

apartado de “CUENTA” en donde vienen nuestros datos registrados, podemos cambiar ciertos datos en caso de que sean erróneos o desactualizados o se puede cerrar sesión si es necesario; Además, nos abre el apartado de “CARRITO” en donde podemos ver los artículos que hemos añadido y realizar un pedido.

**Figura 35** Método para mostrar los detalles de la cuenta

```
[HttpGet]
0 references
public IActionResult Details()
{
    var userId = HttpContext.Session.GetString("UserId");
    if (int.TryParse(userId, out int parsedUserId))
    {
        var user = _context.Usuarios.SingleOrDefault(u => u.Idusuario == parsedUserId);
        if (user == null)
        {
            return NotFound();
        }
        return View(user);
    }
    else
    {
        return RedirectToAction("Login");
    }
}
```

**Figura 36** Métodos para editar información de la cuenta

```

0 references
public IActionResult Edit()
{
    var userId = HttpContext.Session.GetString("UserId");
    if (string.IsNullOrEmpty(userId))
    {
        return RedirectToAction("Login", "Account");
    }
    var user = _context.Usuarios.SingleOrDefault(u => u.Idusuario.ToString() == userId);
    if (user == null)
    {
        return NotFound();
    }
    return View(user);
}

[HttpPost]
[ValidateAntiForgeryToken]
0 references
public IActionResult Edit(Usuario user)
{
    var currentUser = _context.Usuarios.SingleOrDefault(u => u.Idusuario.ToString() == HttpContext.Session.GetString("UserId"));
    if (currentUser == null)
    {
        return NotFound();
    }
    currentUser.Nombre = user.Nombre;
    currentUser.Apellidos = user.Apellidos;
    currentUser.Telefono = user.Telefono;

    if (!string.IsNullOrEmpty(user.Clave))
    {
        currentUser.Clave = HashPassword(user.Clave);
    }
    _context.SaveChanges();
    return RedirectToAction("Details");
}

```

Figura 37 Página para ver detalles de cuenta y cerrar sesión

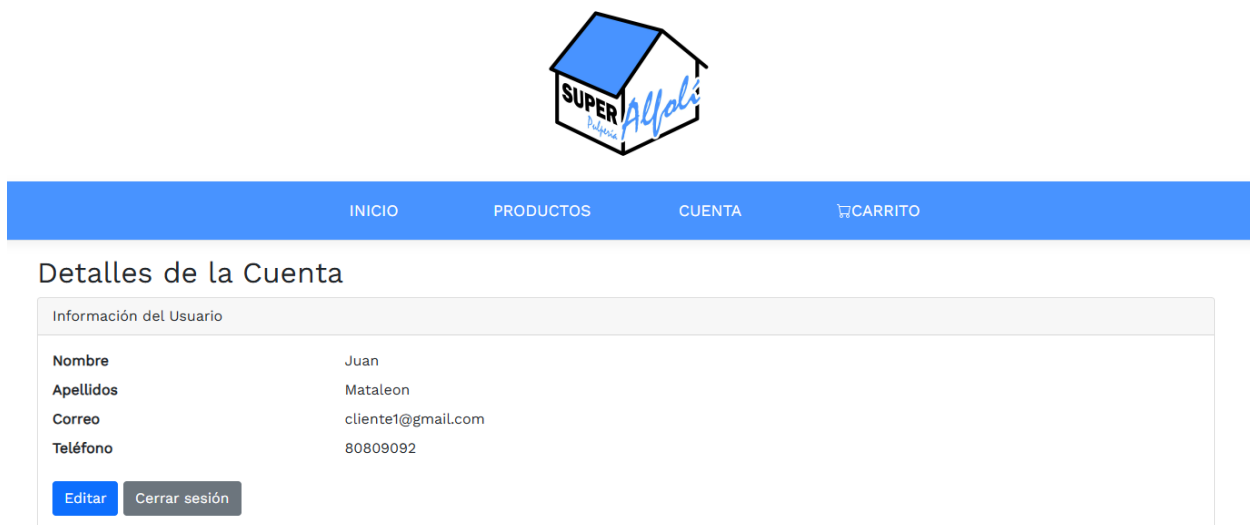


Figura 38 Página para editar información de la cuenta



INICIO PRODUCTOS CUENTA CARRITO

### Editar Cuenta

Nombre  
Juan

Apellidos  
Mataleon

Correo  
cliente1@gmail.com

Telefono  
80809092

Clave

Guardar Cancelar

**Figura 39** Método para agregar productos al carrito

```
[HttpPost]
0 references
public async Task<IActionResult> AgregarAlCarrito(int productId, int quantity)
{
    var userId = HttpContext.Session.GetString("UserId");
    if (userId == null)
    {
        return Unauthorized();
    }

    var carrito = await _context.Carritos
        .Include(c => c.ObjetosCarritos)
        .FirstOrDefaultAsync(c => c.Idusuario.ToString() == userId);

    if (carrito == null)
    {
        carrito = new Carrito
        {
            Idusuario = int.Parse(userId),
            Fecha = DateTime.Now
        };
        _context.Carritos.Add(carrito);
        await _context.SaveChangesAsync();
    }

    var objetoCarrito = carrito.ObjetosCarritos.FirstOrDefault(oc => oc.Idproducto == productId);
    if (objetoCarrito == null)
    {
        objetoCarrito = new ObjetosCarrito
        {
            Idcarrito = carrito.Idcarrito,
            Idproducto = productId,
            Cantidad = quantity,
            Precio = await _context.Productos.Where(p => p.Idproducto == productId).Select(p => p.Precio).FirstOrDefaultAsync(),
            FechaAgregado = DateTime.Now
        };
        _context.ObjetosCarritos.Add(objetoCarrito);
    }
    else
    {
        objetoCarrito.Cantidad += quantity;
    }

    await _context.SaveChangesAsync();

    return Ok();
}
```

Figura 40 Método para desplegar lista de productos en carrito

```

0 references
public async Task<IActionResult> VerCarrito()
{
    var userId = int.Parse(HttpContext.Session.GetString("UserId"));
    var carrito = await _context.Carritos
        .Include(c => c.ObjetosCarritos)
        .ThenInclude(oc => oc.IdproductoNavigation)
        .FirstOrDefaultAsync(c => c.Idusuario == userId);

    if (carrito == null)
    {
        return RedirectToAction("ListaProductos", "Producto");
    }

    List<string> productosModificados = new List<string>();

    foreach (var item in carrito.ObjetosCarritos)
    {
        var producto = item.IdproductoNavigation;

        if (item.Cantidad > producto.Cantidad)
        {
            if (producto.Cantidad > 0)
            {
                productosModificados.Add($"{producto.Nombre}: cantidad ajustada a {producto.Cantidad}");
                item.Cantidad = producto.Cantidad;
            }
            else
            {
                productosModificados.Add($"{producto.Nombre}: eliminado del carrito");
                _context.ObjetosCarritos.Remove(item);
            }
        }
    }

    await _context.SaveChangesAsync();
    if (productosModificados.Count > 0)
    {
        TempData["ProductosModificados"] = productosModificados;
    }

    return View(carrito);
}

```

Figura 41 Método para eliminar objetos del carrito


```

[HttpPost]
0 references
public async Task<IActionResult> EliminarDelCarrito(int idObjetoCarrito)
{
    var objetoCarrito = await _context.ObjetosCarritos.FindAsync(idObjetoCarrito);

    if (objetoCarrito != null)
    {
        _context.ObjetosCarritos.Remove(objetoCarrito);
        await _context.SaveChangesAsync();
    }

    return RedirectToAction("VerCarrito");
}

```

**Figura 42** *Página del carrito*


INICIO PRODUCTOS CUENTA CARRITO

### Carrito de Compras

Producto	Cantidad	Precio Unitario	Total	Acciones
Sal Sol Refinada Bolsa 500gr	1	€520	€520	Eliminar
Pasta Roma Larga Espaguetti N5 250gr	1	€700	€700	Eliminar
Chocolate Twix 50.7gr	1	€900	€900	Eliminar
Refresco Tropical Té Melocotón -3000ml	1	€1450	€1450	Eliminar
<b>Total General</b>			<b>€3570</b>	

Realizar Pedido

El inicio de sesión y el registro utiliza una encriptación tipo SHA256 el cual genera una salida de 32 bytes a partir de una entrada de cualquier tamaño, es un algoritmo muy seguro en general, funciona de manera rápida y precisa y viene integrada en las bibliotecas de C#. Para su uso creamos un método llamado "HashPassword" la cual encripta la clave a la hora de guardarla o de realizar una consulta.

**Figura 43** *Pantalla de inicio de sesión*

[INICIO](#)[PRODUCTOS](#)[CUENTA](#)

## Iniciar sesión

Correo

Contraseña

[Iniciar Sesión](#)

[Registrarse](#)

[Recuperar contraseña](#)

**Figura 44** *Pantalla de registro*

[INICIO](#)[PRODUCTOS](#)[CUENTA](#)

## Registrar

Nombre

Apellidos

Correo

Teléfono

Contraseña

[Registrarse](#)

**Figura 45** Pantalla para recuperar contraseña



INICIO PRODUCTOS CUENTA

## Recuperar Contraseña

Correo electrónico

Enviar enlace de recuperación

**Figura 46** Método "HashPassword" para encriptar contraseñas

```
private string HashPassword(string password)
{
    using (SHA256 sha256 = SHA256.Create())
    {
        byte[] bytes = sha256.ComputeHash(Encoding.UTF8.GetBytes(password));
        StringBuilder builder = new StringBuilder();
        for (int i = 0; i < bytes.Length; i++)
        {
            builder.Append(bytes[i].ToString("x2"));
        }
        return builder.ToString();
    }
}
```

Para iniciar sesión busca un usuario con el cual concuerde el correo y después verifica la contraseña, en caso de ser correcto, redirecciona a su correspondiente página inicial (depende si es administrador, empleado o cliente) y se crea la sesión, si no encuentra coincidencias muestra un error.

**Figura 47** Método de inicio de sesión

```
[HttpPost]
[ValidateAntiForgeryToken]
0 references
public IActionResult Login(string correo, string clave)
{
    if (string.IsNullOrEmpty(correo) || string.IsNullOrEmpty(clave))
    {
        ViewBag.Error = "Debe ingresar un correo y clave válidos.";
        return View();
    }

    string hashedPassword = HashPassword(clave);
    var usuario = _context.Usuarios.FirstOrDefault(u => u.Correo == correo && u.Clave == hashedPassword);

    if (usuario != null)
    {
        HttpContext.Session.SetString("UserId", usuario.Idusuario.ToString());
        HttpContext.Session.SetString("UsuarioNombre", usuario.Nombre);
        HttpContext.Session.SetString("Rol", usuario.Rol ?? "Cliente");

        if (usuario.Rol == "Admin")
        {
            return RedirectToAction("Index", "Admin");
        }
        else
        {
            return RedirectToAction("Index", "Home");
        }
    }

    ModelState.AddModelError(string.Empty, "Correo o contraseña incorrectos.");
    return View();
}
```

Para registrar muestra una nueva vista formulario donde se deben agregar datos como, nombre, apellidos, correo, teléfono y contraseña, en caso de que la información brindada sea válida se guarda el usuario nuevo en la base de datos.

**Figura 48** Método para registrar usuario

```
[HttpPost]
[ValidateAntiForgeryToken]
0 references
public IActionResult Register(Usuario usuario)
{
    if (ModelState.IsValid)
    {
        usuario.Clave = HashPassword(usuario.Clave);
        usuario.Rol = "Cliente";
        usuario.Estado = true;

        _context.Usuarios.Add(usuario);
        _context.SaveChanges();

        return RedirectToAction("Login");
    }

    return View(usuario);
}
```

Para recuperar la contraseña se solicita al usuario el correo con el cual se registró, luego se envía un enlace de recuperación el cual se crea mediante una ficha aleatoria; si el usuario entra en el enlace del correo su clave será cambiada a una nueva generada por un método el cual selecciona aleatoriamente seis caracteres entre letras (mayúsculas y minúsculas) y números (del 0 al 9) para que la persona pueda ingresar con este código a su cuenta y cambiarlo por una nueva.

**Figura 49** Método para solicitar recuperación de contraseña

```
[HttpPost]
0 references
public async Task<IActionResult> EnviarEnlaceRecuperacion(string correo)
{
    var usuario = await _context.Usuarios.FirstOrDefaultAsync(u => u.Correo == correo);
    if (usuario == null)
    {
        ViewBag.Error = "Correo no encontrado";
        return View("RecuperarContraseña");
    }

    string token = Guid.NewGuid().ToString();

    string enlaceRecuperacion = Url.Action("CambiarContraseña", "Account", new { token = token, id = usuario.Idusuario },
        protocol: HttpContext.Request.Scheme);

    usuario.Clave = HashPassword(token);
    await _context.SaveChangesAsync();

    string asunto = "Recuperación de contraseña";
    string mensaje = $"<p>Para recuperar su contraseña, haga clic en el siguiente enlace:</p><a href='{enlaceRecuperacion}'>Recuperar contraseña</a>";
    await _emailService.EnviarCorreoAsync(correo, asunto, mensaje);

    ViewBag.Message = "Se ha enviado un enlace de recuperación a su correo.";
    return View("RecuperarContraseña");
}
```

**Figura 50** Método para cambiar la contraseña

```
0 references
public async Task<IActionResult> CambiarContraseña(string token, int id)
{
    var usuario = await _context.Usuarios.FindAsync(id);
    if (usuario == null || usuario.Clave != HashPassword(token))
    {
        return NotFound();
    }

    string nuevaContraseña = GenerarContraseñaAleatoria();

    usuario.Clave = HashPassword(nuevaContraseña);
    await _context.SaveChangesAsync();

    ViewBag.NuevaContraseña = nuevaContraseña;
    return View();
}
1 reference
private string GenerarContraseñaAleatoria(int longitud = 6)
{
    const string caracteres = "abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789";
    var random = new Random();
    return new string(Enumerable.Repeat(caracteres, longitud)
        .Select(s => s[random.Next(s.Length)]).ToArray());
}
```

Al realizar el pedido se envía un correo de confirmación de pedido al correo. Para lograr esto se utiliza la clase "SmtpClient" de la biblioteca "System.Net.Mail" la cual permite enviar estos mensajes mediante el protocolo SMTP (Simple Mail Transfer

Protocol), para ello debemos brindar las credenciales y configurar ciertas opciones para lograr el objetivo.

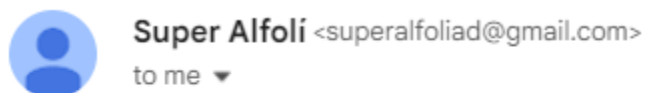
En el "appsetting.json" debemos establecer las configuraciones que se van a utilizar como: el host, el puerto, el nombre, el email y la contraseña, para este sistema creamos el correo [superalfoliad@gmail.com](mailto:superalfoliad@gmail.com) el cual es el encargado de notificar a los usuarios el estado de su pedido. Es importante activar ciertas opciones en la cuenta para que permita ser usada en aplicaciones.

**Figura 51** Configuración para enviar correos en el "appsetting.json"

```
"EmailSettings": {  
  "SmtpHost": "smtp.gmail.com",  
  "SmtpPort": 587,  
  "FromName": "Super Alfolí",  
  "FromEmail": "superalfoliad@gmail.com",  
  "SmtpUser": "superalfoliad@gmail.com",  
  "SmtpPass": "cwwh mprd fdnn fiwl",  
  "EnableSsl": true  
},
```

Con esto listo solo debemos crear un método que envíe los correos los cuales estarán en la clase "EmailService.cs" en donde se debe verificar las credenciales, después crear el correo, agregar los destinatarios e intentar enviar el correo de manera asincrónica, esta clase será usada cada vez que se necesite enviar un correo a un usuario (notificando el estado del pedido, para cambiar de contraseña, cuando se registra un cliente).

**Figura 52** Ejemplo de correo recibido para recuperar la contraseña



Para recuperar su contraseña, haga clic en el siguiente enlace:

[Recuperar contraseña](#)

**Figura 53** Método para enviar correos en la clase "EmailService.cs"

```
public async Task EnviarCorreoAsync(string destinatario, string asunto, string mensajeHtml)
{
    var smtpHost = _configuration["EmailSettings:SmtpHost"];
    var smtpPort = int.Parse(_configuration["EmailSettings:SmtpPort"]);
    var smtpUser = _configuration["EmailSettings:SmtpUser"];
    var smtpPass = _configuration["EmailSettings:SmtpPass"];
    var fromEmail = _configuration["EmailSettings:FromEmail"];
    var fromName = _configuration["EmailSettings:FromName"];
    var enableSsl = bool.Parse(_configuration["EmailSettings:EnableSsl"]);

    using (var client = new SmtpClient(smtpHost, smtpPort))
    {
        client.Credentials = new NetworkCredential(smtpUser, smtpPass);
        client.EnableSsl = enableSsl;

        var mailMessage = new MailMessage
        {
            From = new MailAddress(fromEmail, fromName),
            Subject = asunto,
            Body = mensajeHtml,
            IsBodyHtml = true
        };

        mailMessage.To.Add(destinatario);

        try
        {
            await client.SendMailAsync(mailMessage);
        }
        catch (SmtpException ex)
        {
            throw new Exception($"Error al enviar el correo: {ex.Message}", ex);
        }
    }
}
```

Figura 54 Método para realizar un pedido

```

public async Task<IActionResult> RealizarPedido()
{
    var userId = HttpContext.Session.GetString("UserId");
    var usuario = await _context.Usuarios.FindAsync(int.Parse(userId));
    var carrito = await _context.Carritos
        .Include(c => c.ObjetosCarritos)
        .ThenInclude(oc => oc.IdproductoNavigation)
        .FirstOrDefaultAsync(c => c.Idusuario.ToString() == userId);

    if (carrito == null || !carrito.ObjetosCarritos.Any())
    {
        return RedirectToAction("VerCarrito", "Carrito");
    }

    var pedido = new Pedido
    {
        Idusuario = int.Parse(userId),
        Fecha = DateTime.Now,
        Estado = "Pendiente",
        Total = carrito.ObjetosCarritos.Sum(i => i.Precio * i.Cantidad ?? 0)
    };

    foreach (var item in carrito.ObjetosCarritos)
    {
        var detallePedido = new DetallePedido
        {
            Idproducto = item.Idproducto,
            Precio = item.Precio,
            Cantidad = item.Cantidad ?? 0,
            Total = item.Precio * (item.Cantidad ?? 0)
        };
        pedido.DetallePedidos.Add(detallePedido);
    }

    _context.Pedidos.Add(pedido);
    _context.ObjetosCarritos.RemoveRange(carrito.ObjetosCarritos);
    await _context.SaveChangesAsync();

    var detallesPedido = pedido.DetallePedidos.ToList();
    string mensaje = _emailService.CrearMensajePedido(usuario.Nombre, detallesPedido, pedido.Total);

    await _emailService.EnviarCorreoAsync(usuario.Correo, "Pedido Realizado - Super Alfolí", mensaje);

    TempData["PedidoExitoso"] = "Tu pedido se ha realizado con éxito";
    return RedirectToAction("VerCarrito", "Carrito");
}

```

### 4.3 Apartado de administrador y empleado

Si la cuenta es de tipo administrador será llevado al apartado de gestión en donde tiene otras opciones; esto se verifica mediante la clase "AdminAuthorizeAttribute.cs" y "EmployeeAndAdminAuthorizeAttribute.cs" las cuales son un atributo personalizado que se utiliza para restringir el acceso a las acciones que solo debería realizar los super

usuarios, esta verifica el rol del usuario y en caso de no tener rol (no ha iniciado sesión) o ser cliente será redirigido a la página de acceso restringido, para aplicar el atributo debemos ir a las acciones en el controlador y añadir la siguiente línea de código “[AdminAuthorize]”.

**Figura 55** Estructura de clase "AdminAuthorizeAttribute.cs"

```

using Microsoft.AspNetCore.Mvc;
using Microsoft.AspNetCore.Mvc.Filters;

24 references
public class AdminAuthorizeAttribute : ActionFilterAttribute
{
    0 references
    public override void OnActionExecuting(ActionExecutingContext context)
    {
        var userRole = context.HttpContext.Session.GetString("Rol");

        if (string.IsNullOrEmpty(userRole) || userRole != "Admin")
        {
            context.Result = new RedirectToActionResult("AccesoRestringido", "Account", null);
        }

        base.OnActionExecuting(context);
    }
}

```

Una vez verificado el usuario este podrá acceder a distintas funciones. En el apartado de “PRODUCTOS” se podrá agregar productos nuevos y eliminar, modificar o ver detalles de los ya existentes, se utiliza de nuevo el “DbContext” para interactuar con la base de datos, primero se hace una solicitud para desplegar todos los productos en una lista.


**Figura 56** Código Razor para desplegar lista de productos

```

<tbody >
  @foreach (var item in Model)
  {
    <tr>
      <td>@item.Nombre</td>
      <td>@item.Categoria</td>
      <td>@item.Precio.ToString("0")</td>
      <td>@item.Cantidad</td>
      <td>
        <a asp-action="Edit" asp-route-id="@item.Idproducto" class="btn btn-primary mt-3 mb-3">Editar</a>
        <a asp-action="Details" asp-route-id="@item.Idproducto" class="btn btn-primary mt-3 mb-3">Detalles</a>
        <a asp-action="Delete" asp-route-id="@item.Idproducto" class="btn btn-primary mt-3 mb-3">Eliminar</a>
      </td>
    </tr>
  }
</tbody>

```

**Figura 57** Página para los administradores para manejar los productos



INICIO PRODUCTOS USUARIOS PEDIDOS BITÁCORA

Lista de productos

[Añadir producto](#)
[Ver lista de productos](#)
[Productos pronto a acabarse](#)

Mostrar  registros por página Buscar:

Nombre	Categoría	Precio	Cantidad	Acciones
Arroz Indiana 99% Grano Entero 1800gr	Abarrotes	₡1850	21	<a href="#">Editar</a> <a href="#">Detalles</a> <a href="#">Eliminar</a>
Arroz Tio Pelo 99% Grano Entero 1800 gr	Abarrotes	₡1850	20	<a href="#">Editar</a> <a href="#">Detalles</a> <a href="#">Eliminar</a>
Bistec De Res Especial por Kilo	Carnes	₡5700	39	<a href="#">Editar</a> <a href="#">Detalles</a> <a href="#">Eliminar</a>
Café 100% Puro 1820 Molido Clásico, Tueste Oscuro 500gr	Abarrotes	₡4050	54	<a href="#">Editar</a> <a href="#">Detalles</a> <a href="#">Eliminar</a>
Chocolate Twix 50.7 g	Snacks dulces	₡880	19	<a href="#">Editar</a> <a href="#">Detalles</a> <a href="#">Eliminar</a>

Para crear un nuevo registro, se solicita la información del artículo y una imagen la cual debe ser transformada a un stream de memoria y después a un arreglo, de esta manera podemos guardarla sin problema en la base de datos, con el método “SaveChangesAsync” se crea el producto.

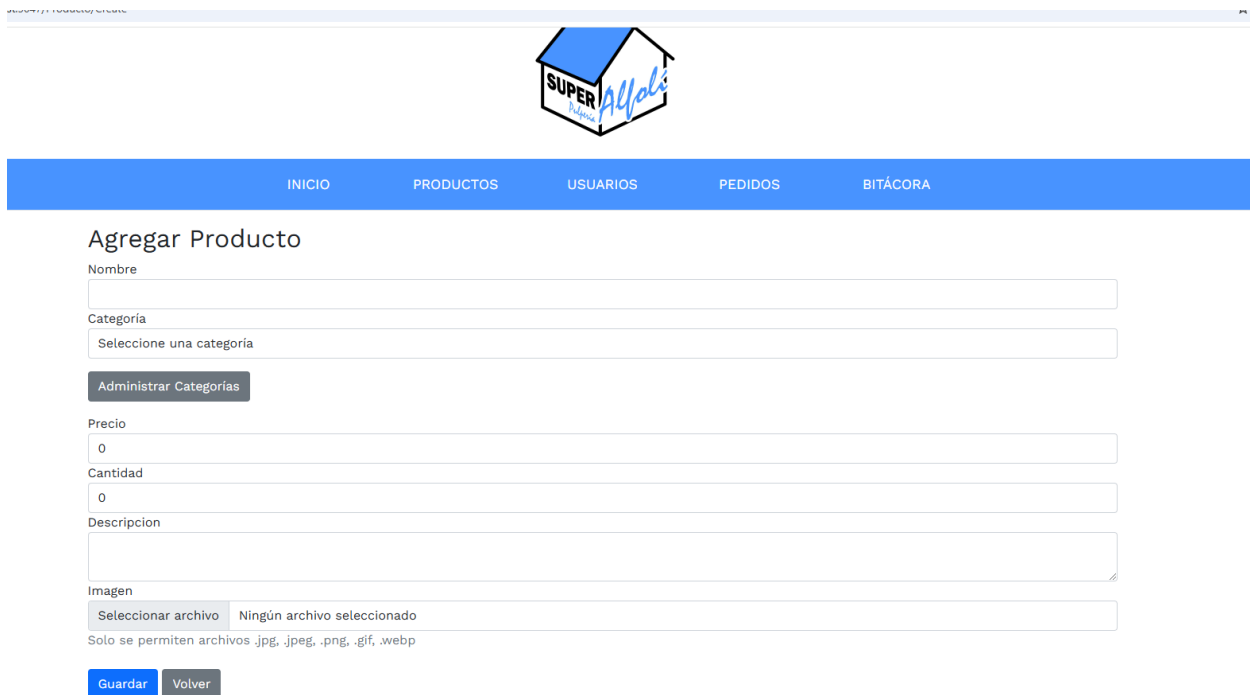
**Figura 58** Método para crear un nuevo producto


```

[HttpPost]
[ValidateAntiForgeryToken]
[AdminAuthorize]
0 references
public async Task<IActionResult> Create(Producto producto, IFormFile imageFile)
{
    if (ModelState.IsValid)
    {
        if (imageFile != null && imageFile.Length > 0)
        {
            using (var memoryStream = new MemoryStream())
            {
                await imageFile.CopyToAsync(memoryStream);
                producto.Imagen = memoryStream.ToArray();
            }
        }
        else
        {
            producto.Imagen = null;
        }
        _context.Add(producto);
        await _context.SaveChangesAsync();
        return RedirectToAction(nameof(Index));
    }
    return View(producto);
}

```

Figura 59 Página para agregar un nuevo producto





INICIO    PRODUCTOS    USUARIOS    PEDIDOS    BITÁCORA

### Agregar Producto

Nombre

Categoría

[Administrar Categorías](#)

Precio

Cantidad

Descripción

Imagen  Ningún archivo seleccionado

Solo se permiten archivos .jpg, .jpeg, .png, .gif, .webp

[Guardar](#) [Volver](#)

Se puede agregar una nueva categoría accediendo al botón de “Administrar Categorías”, este nos redirige a una vista en donde podemos escribir una nueva y en caso de que no exista, agregarla, además, podremos eliminar las categorías ya existentes (siempre que no estén vinculadas a algún producto).

**Figura 60** *Página para administrar las categorías*

Nombre	Acciones
Abarrotes	X
Bebidas	X
Verduras	X
Limpieza	X
Higiene personal	X
Lácteos	X
Snacks dulces	X
Snacks salados	X

Para eliminar un artículo se envía el número identificador por método POST, después, se busca la existencia de este registro en la base de datos, cuando el usuario confirma la eliminación se utiliza el “.Remove(producto)” para borrar los datos.

**Figura 61** *Código para verificar la existencia del producto*

```

[HttpGet]
[AdminAuthorize]
0 references
public async Task<IActionResult> Delete(int? id)
{
    if (id == null)
    {
        return NotFound();
    }

    var producto = await _context.Productos
        .FirstOrDefaultAsync(m => m.Idproducto == id);
    if (producto == null)
    {
        return NotFound();
    }

    return View(producto);
}

```

Figura 62 Código para eliminar producto después de confirmación

```

[HttpPost, ActionName("Delete")]
[ValidateAntiForgeryToken]
[AdminAuthorize]
0 references
public async Task<IActionResult> DeleteConfirmed(int? id) {
    var producto = await _context.Productos.FindAsync(id);
    _context.Productos.Remove(producto);
    await _context.SaveChangesAsync();
    return RedirectToAction(nameof(Index));
}

```

Figura 63 Página de eliminación de producto



INICIO    PRODUCTOS    USUARIOS    PEDIDOS    BITÁCORA

### Eliminar Producto

⚠ ¿Está seguro que desea eliminar este usuario?

---

Nombre	Arroz Indiana 99% Grano Entero 1800gr
Categoría	Abarrotes
Precio	1850.00
Cantidad	21
Descripción	Arroz Indiana 99% Grano Entero -1800gr

Y por último para editar se solicitan los datos actuales para desplegarlos, cuando el administrador termina de modificar la información presiona el botón guardar en donde con el comando “.SetValues(producto)” se actualiza la información.

**Figura 64** Código para desplegar información actual del producto

```

[HttpGet]
[AdminAuthorize]
0 references
public async Task<IActionResult> Edit(int id)
{
    var producto = await _context.Productos.FindAsync(id);
    if (producto == null)
    {
        return NotFound();
    }
    return View(producto);
}
```

**Figura 65** Código para guardar los nuevos datos ingresados

```

[HttpPost]
[ValidateAntiForgeryToken]
[AdminAuthorize]
0 references
public async Task<IActionResult> Edit(int id, [Bind("Idproducto, Nombre, Categoria, Precio, Cantidad, Descripcion, Imagen")] Producto producto, IFormFile imageFile)
{
    if (id != producto.Idproducto)
    {
        return NotFound();
    }
    var actualInfo = await _context.Productos.FindAsync(id);
    if (actualInfo == null)
    {
        return NotFound();
    }
    try
    {
        if (imageFile != null && imageFile.Length > 0)
        {
            using (var memoryStream = new MemoryStream())
            {
                await imageFile.CopyToAsync(memoryStream);
                producto.Imagen = memoryStream.ToArray();
            }
        }
        if (producto.Imagen == null) {
            producto.Imagen = actualInfo.Imagen;
        }
        _context.Entry(actualInfo).CurrentValues.SetValues(producto);
        await _context.SaveChangesAsync();
        return RedirectToAction(nameof(Index));
    }
    catch (DbUpdateConcurrencyException)
    {
        if (!_context.Productos.Any(e => e.Idproducto == id))
        {
            return NotFound();
        }
        else
        {
            return View(producto);
            throw;
        }
    }
}

```

Figura 66 Página para editar datos de los productos

INICIO	PRODUCTOS	USUARIOS	PEDIDOS	BITÁCORA
--------	-----------	----------	---------	----------

## Editar Producto

Nombre


Categoría

Precio

Cantidad

Descripcion

Imagen  
 Ningún archivo seleccionado

Imagen Actual  


En la parte de “USUARIOS” permite, a las cuentas con rol de administrador, realizar las mismas acciones de CRUD (Create, Read, Update, Delete) con las cuentas de las personas registradas, claramente no se podrá cambiar datos como el correo o la contraseña de un usuario, pero si su rol, nombre, apellidos, teléfono y estado de la cuenta (En caso de que desee suspenderla por algún motivo especial), los usuarios con el rol de empleado sola mente podrán ver los detalles de los usuarios, pero, por seguridad, no podrá realizar ningún cambio. El proceso es el mismo que con la tabla de productos con algunas variaciones como, al crear un usuario se envía al correo una contraseña temporal.

**Figura 67** *Código para generar lista de usuarios*

```
[AdminAuthorize]
3 references
public async Task<IActionResult> Index()
{
    return View(await _context.Usuarios.ToListAsync());
}
```

**Figura 68** *Pantalla con lista de usuarios*



INICIO PRODUCTOS USUARIOS PEDIDOS BITÁCORA

## Lista de Usuarios

[Crear nuevo usuario](#)

Mostrar  registros por página Buscar:

Nombre	Apellidos	Correo	Teléfono	Rol	Estado	Acciones
Julian	Gomez	juliangomez2684@gmail.com	88321312	Empleado	Activo	<a href="#">Editar</a> <a href="#">Detalles</a> <a href="#">Eliminar</a>
Luis	Soto	luissoto2233411@gmail.com	88008800	Administrador	Activo	<a href="#">Editar</a> <a href="#">Detalles</a> <a href="#">Eliminar</a>
Maria	Rodriguez	mariarodriguez292900@gmail.com	881188111	Cliente	Activo	<a href="#">Editar</a> <a href="#">Detalles</a> <a href="#">Eliminar</a>

Mostrando 1 a 3 de 3 usuarios Anterior  Siguiente

Figura 69 Método para crear usuario

```
[HttpPost]
[ValidateAntiForgeryToken]
[AdminAuthorize]
0 references
public async Task<IActionResult> Create(Usuario usuario)
{
    string contraseña = GenerarContraseñaAleatoria();
    usuario.Clave = HashPassword(contraseña);
    _context.Add(usuario);
    await _context.SaveChangesAsync();

    await EnviarCorreo(usuario.Correo, contraseña);

    return RedirectToAction(nameof(Index));
}
```

Figura 70 Página para crear un nuevo usuario



INICIO PRODUCTOS USUARIOS PEDIDOS BITÁCORA

### Crear Usuario

Nombre

Apellidos

Correo

Telefono

Rol

Estado

**Figura 71** Métodos para eliminar usuarios (solicitud de confirmación y eliminación)

```
[HttpGet]
[AdminAuthorize]
0 references
public async Task<IActionResult> Delete(int? id)
{
    if (id == null)
    {
        return NotFound();
    }

    var usuario = await _context.Usuarios
        .FirstOrDefaultAsync(m => m.Idusuario == id);

    if (usuario == null)
    {
        return NotFound();
    }

    return View(usuario);
}

[HttpPost]
[AdminAuthorize]
0 references
public async Task<IActionResult> DeleteConfirmed(int id)
{
    var usuario = await _context.Usuarios.FindAsync(id);
    if (usuario == null)
    {
        return NotFound();
    }

    _context.Usuarios.Remove(usuario);
    await _context.SaveChangesAsync();
    return RedirectToAction(nameof(Index));
}
```

Figura 72 *Página para eliminar usuario*



[INICIO](#)    [PRODUCTOS](#)    [USUARIOS](#)    [PEDIDOS](#)

## Eliminar Usuario

¿Está seguro que desea eliminar este usuario?

---

Nombre	Alejandro
Apellidos	Salas Casanova
Correo	prueba@gmail.com
Teléfono	87093678
Rol	Admin
Estado	True


**Figura 73** Código para editar usuarios

```
[HttpPost]
[ValidateAntiForgeryToken]
[AdminAuthorize]
0 references
public async Task<IActionResult> Edit(int id, [Bind("Idusuario,Nombre,Apellidos,Telefono,Rol,Estado")] Usuario usuario)
{
    if (id != usuario.Idusuario)
    {
        return NotFound();
    }

    var usuarioActual = await _context.Usuarios.FindAsync(id);
    if (usuarioActual == null)
    {
        return NotFound();
    }

    usuarioActual.Nombre = usuario.Nombre;
    usuarioActual.Apellidos = usuario.Apellidos;
    usuarioActual.Telefono = usuario.Telefono;
    usuarioActual.Rol = usuario.Rol;
    usuarioActual.Estado = usuario.Estado;

    try
    {
        await _context.SaveChangesAsync();
        return RedirectToAction(nameof(Index));
    }
    catch (DbUpdateConcurrencyException)
    {
        if (!_context.Usuarios.Any(e => e.Idusuario == id))
        {
            return NotFound();
        }
        else
        {
            throw;
        }
    }
}
```

**Figura 74** *Página para editar usuarios*

INICIO PRODUCTOS USUARIOS PEDIDOS BITÁCORA

### Editar Usuario

Nombre  
Julian

Apellidos  
Gomez

Telefono  
88321312

Rol  
Empleado

Estado  
Activo

En la pestaña de “PEDIDOS” se puede ver una lista de pedidos, esta está organizada en colores según su estado actual (verde: pendiente, negro: finalizado, rojo: cancelado), además, cuenta con un botón para poder imprimir las facturas.

Para hacerlo se debe buscar la factura que desea imprimir (se recomienda utilizar la barra de buscar en la parte superior de la tabla) y se presiona sobre el botón “imprimir”, después, se selecciona la carpeta en la que se desea guardar el archivo.

**Figura 75** *Pantalla para seleccionar factura a imprimir*



INICIO

PRODUCTOS

USUARIOS

PEDIDOS

BITÁCORA

## Lista de Facturas

Mostrar  facturasBuscar: 

ID Factura	Cliente	Correo	Productos Comprados	Total	Fecha	Acciones
3	Maria Rodriguez	maria-rodriguez292900@gmail.com	<ul style="list-style-type: none"> <li>Arroz Tio Pelo 99% Grano Entero 1800 gr x 1 - ¢1850</li> <li>Chocolate Twix 50.7 g x 1 - ¢880</li> </ul>	¢2730	12/02/2025 17:20	<a href="#">Imprimir</a>
2	Maria Rodriguez	maria-rodriguez292900@gmail.com		¢0	12/02/2025 17:06	<a href="#">Imprimir</a>
1	Maria Rodriguez	maria-rodriguez292900@gmail.com	<ul style="list-style-type: none"> <li>Culantro En Rollo Grande x 1 - ¢500</li> <li>Bistec De Res Especial por Kilo x 3 - ¢17100</li> <li>Sal Sol Refinada Bolsa 500gr x 2 - ¢490</li> </ul>	¢18090	11/02/2025 16:04	<a href="#">Imprimir</a>

Mostrando 1 a 3 de 3 facturas

Anterior

Siguiente

**Figura 76** Ejemplo de factura impresa



### Factura #3

Fecha: 12/02/2025 17:20

**Cliente:** Maria Rodriguez

**Correo:** mariarodriguez292900@gmail.com

#### Productos

Producto	Cantidad	Precio	Total
Arroz Tio Pelo 99% Grano Entero 1800 gr	1	₡1850	₡1850
Chocolate Twix 50.7 g	1	₡880	₡880

**Total: ₡2730.00**

Imprimir

**Figura 77** Código Razor para generar lista de pedidos y organizarlo según su estado actual.

```

<div class="container mt-3">
  <h2>Lista de Pedidos</h2>
  <h3 class="mt-4">Pedidos Pendientes</h3>
  <div class="list-group">
    @foreach (var pedido in Model.Where(p => p.Estado == "Pendiente"))
    {
      <a href="@Url.Action("DetallesPedido", "Pedidos", new { id = pedido.Idpedido })" class="list-group-item list-group-item-action bg-success text-white">
        <div class="d-flex w-100 justify-content-between">
          <h5 class="mb-1">Pedido #@pedido.Idpedido</h5>
          <small>@pedido.Fecha.Value.ToString("dd/MM/yyyy HH:mm")</small>
        </div>
        <p class="mb-1">Estado: @pedido.Estado</p>
        <small>Total: @pedido.Total.ToString("0")</small>
      </a>
    }
  </div>
  <h3 class="mt-4">Pedidos Finalizados</h3>
  <div class="list-group">
    @foreach (var pedido in Model.Where(p => p.Estado == "Finalizado"))
    {
      <a href="@Url.Action("DetallesPedido", "Pedidos", new { id = pedido.Idpedido })" class="list-group-item list-group-item-action bg-dark text-white">
        <div class="d-flex w-100 justify-content-between">
          <h5 class="mb-1">Pedido #@pedido.Idpedido</h5>
          <small>@pedido.Fecha.Value.ToString("dd/MM/yyyy HH:mm")</small>
        </div>
        <p class="mb-1">Estado: @pedido.Estado</p>
        <small>Total: @pedido.Total.ToString("0")</small>
      </a>
    }
  </div>
  <h3 class="mt-4">Pedidos Cancelados</h3>
  <div class="list-group">
    @foreach (var pedido in Model.Where(p => p.Estado == "Cancelado"))
    {
      <a href="@Url.Action("DetallesPedido", "Pedidos", new { id = pedido.Idpedido })" class="list-group-item list-group-item-action bg-danger text-white">
        <div class="d-flex w-100 justify-content-between">
          <h5 class="mb-1">Pedido #@pedido.Idpedido</h5>
          <small>@pedido.Fecha.Value.ToString("dd/MM/yyyy HH:mm")</small>
        </div>
        <p class="mb-1">Estado: @pedido.Estado</p>
        <small>Total: @pedido.Total.ToString("0")</small>
      </a>
    }
  </div>
</div>

```

Figura 78 Vista de lista de pedidos



INICIO
PRODUCTOS
USUARIOS
PEDIDOS
BITÁCORA

## Lista de Pedidos

Ver facturas

### Pedidos Pendientes

Pedido #5	12/02/2025 17:17
Estado: Pendiente	
Total: ₡2840	
Pedido #1	10/02/2025 13:24
Estado: Pendiente	
Total: ₡0	

### Pedidos Finalizados

12/02/2025 - Total generado en el día: ₡2730

Pedido #6	12/02/2025 17:19
Estado: Finalizado	
Total: ₡2730	

10/02/2025 - Total generado en el día: ₡18090

Pedido #4	10/02/2025 17:07
-----------	------------------

Al acceder a un pedido se pueden ver datos de interés como la fecha y hora en el que se realizó, el estado, el total, la información del cliente y los productos solicitados; en caso de que no haya disponibilidad del producto se puede cambiar la cantidad o directamente eliminarlo, cuando se termine de armar el pedido se presiona el botón “Finalizar Pedido” el cual cambia el estado actual de la solicitud y envía un mensaje al cliente indicando que ya puede acudir al local a recoger el pedido, además incluye todos los cambios realizados al pedido y el nuevo total. También, según la necesidad lo requiera (el pedido está mal, lo hicieron dos veces, no hay ningún artículo, no se pudo contactar al cliente, entre otros) se puede presionar la opción de “Cancelar pedido” la cual lo asigna como cancelado y se envía un mensaje al usuario avisando que su pedido ha sido denegado.

**Figura 79** Código para ver detalles del pedido seleccionado

```
[AdminAuthorize]
0 references
public async Task<IActionResult> DetallesPedido(int id)
{
    var pedido = await _context.Pedidos
        .Include(p => p.DetallePedidos)
        .ThenInclude(dp => dp.IdproductoNavigation)
        .Include(p => p.IdusuarioNavigation)
        .FirstOrDefaultAsync(p => p.Idpedido == id);

    if (pedido == null)
    {
        return NotFound();
    }

    return View(pedido);
}
```

**Figura 80** Página para visualizar los detalles del pedido

INICIO
PRODUCTOS
USUARIOS
PEDIDOS

### Detalles del Pedido #1011

**Fecha:** 24/09/2024 15:56

**Estado:** Pendiente

**Total:** ₡520

#### Información del Cliente

**Nombre:** Maria Ramirez

**Correo:** mariarodriguez292900@gmail.com

**Teléfono:** 84271052

#### Productos en el Pedido

Sal Sol Refinada Bolsa 500gr

Cantidad actual: 1

Precio Unitario: ₡520

Total: ₡520

Modificar Cantidad

Modificar Cantidad
Eliminar Producto

Finalizar Pedido
Cancelar Pedido

**Figura 81** Método para cambiar la cantidad de un producto en el pedido

```
[HttpPost]
[AdminAuthorize]
0 references
public async Task<IActionResult> ModificarCantidad(int id, int idDetallePedido, int nuevaCantidad)
{
    var detalle = await _context.DetallePedidos
        .Include(dp => dp.IdproductoNavigation)
        .FirstOrDefaultAsync(dp => dp.Iddetalle == idDetallePedido);

    if (detalle == null)
    {
        return NotFound();
    }

    detalle.Cantidad = nuevaCantidad;
    detalle.Total = detalle.Precio * nuevaCantidad;

    await _context.SaveChangesAsync();

    await RecalcularTotalPedido(detalle.Idpedido);

    return RedirectToAction("DetallesPedido", new { id = detalle.Idpedido });
}
```

**Figura 82** Método para cambiar el estado del pedido a finalizado

```

[AdminAuthorize]
0 references
public async Task<IActionResult> FinalizarPedido(int id)
{
    var pedido = await _context.Pedidos
        .Include(p => p.DetallePedidos)
        .ThenInclude(dp => dp.IdproductoNavigation)
        .Include(p => p.IdusuarioNavigation)
        .FirstOrDefaultAsync(p => p.Idpedido == id);

    if (pedido == null)
    {
        return NotFound();
    }

    var nombreUsuario = pedido.IdusuarioNavigation.Nombre;

    foreach (var detalle in pedido.DetallePedidos)
    {
        var producto = detalle.IdproductoNavigation;
        if (producto.Cantidad >= detalle.Cantidad)
        {
            producto.Cantidad -= detalle.Cantidad;
        }
        else
        {
            TempData["Error"] = $"No hay suficiente stock para el producto {producto.Nombre}.";
            return RedirectToAction("DetallesPedido", new { id = pedido.Idpedido });
        }
    }

    pedido.Estado = "Finalizado";
    await _context.SaveChangesAsync();

    var detallesPedido = pedido.DetallePedidos.ToList();
    string mensaje = _emailService.CrearMensajePedidoListo(nombreUsuario, detallesPedido, pedido.Total);
    await _emailService.EnviarCorreoAsync(pedido.IdusuarioNavigation.Correo, "Tu Pedido Está Listo", mensaje);

    return RedirectToAction("VerPedidos");
}

```

Figura 83 Método para cambiar el estado del pedido a cancelado

```

[HttpPost]
[AdminAuthorize]
0 references
public async Task<IActionResult> CancelarPedido(int id)
{
    var pedido = await _context.Pedidos
        .Include(p => p.IdusuarioNavigation)
        .FirstOrDefaultAsync(p => p.Idpedido == id);

    if (pedido == null)
    {
        return NotFound();
    }

    var nombreUsuario = pedido.IdusuarioNavigation.Nombre;

    pedido.Estado = "Cancelado";
    await _context.SaveChangesAsync();

    string mensaje = _emailService.CrearMensajePedidoCancelado(nombreUsuario);
    await _emailService.EnviarCorreoAsync(pedido.IdusuarioNavigation.Correo, "Cancelación de Pedido", mensaje);

    return RedirectToAction("VerPedidos");
}

```

**Figura 84** Método para eliminar productos del pedido actual

```
[HttpPost]
[AdminAuthorize]
0 references
public async Task<IActionResult> EliminarProducto(int idDetallePedido)
{
    var detallePedido = await _context.DetallePedidos
        .Include(dp => dp.IdproductoNavigation)
        .FirstOrDefaultAsync(dp => dp.Iddetalle == idDetallePedido);

    if (detallePedido == null)
    {
        return NotFound();
    }

    _context.DetallePedidos.Remove(detallePedido);
    await _context.SaveChangesAsync();

    await RecalcularTotalPedido(detallePedido.Idpedido);

    return RedirectToAction("DetallesPedido", new { id = detallePedido.Idpedido });
}
```

**Figura 85** Método para recalcular el total del pedido después de modificar la cantidad de un producto o eliminarlo

```
[AdminAuthorize]
2 references
private async Task RecalcularTotalPedido(int idPedido)
{
    var pedido = await _context.Pedidos
        .Include(p => p.DetallePedidos)
        .FirstOrDefaultAsync(p => p.Idpedido == idPedido);

    if (pedido == null)
    {
        return;
    }

    pedido.Total = pedido.DetallePedidos.Sum(dp => dp.Total);

    await _context.SaveChangesAsync();
}
```

Los administradores cuentan con una opción extra en su barra de navegación la cual es “Bitácora” en donde al ingresar se visualiza una tabla que muestra los datos guardados en la base de datos como bitácora, estos pueden ser de distintos tipos: iniciar sesión, cerrar sesión, crear, modificar, eliminar, cambiar estado de pedido, modificar stock y otros. Estos registros ayudaran al administrador a estar informado de todas las acciones importantes que realicen los empleados y los otros administradores y poder detectar problemas en caso de alguna actividad extraña.

**Figura 86** Pantalla para verificar las bitácoras



INICIO PRODUCTOS USUARIOS PEDIDOS BITÁCORA				
Bitácora				
<a href="#">Generar reportes</a>				
Mostrar <input type="text" value="10"/> registros por página				Buscar: <input type="text"/>
Usuario	Correo	Acción	Fecha	Descripción
Luis Soto	luissoto2233411@gmail.com	Modificar	2025-02-18 12:56:14	El usuario modificó al usuario
Luis Soto	luissoto2233411@gmail.com	Modificar	2025-02-18 12:56:24	El usuario modificó al usuario
Luis Soto	luissoto2233411@gmail.com	Modificar	2025-02-18 12:57:48	El usuario modificó al usuario
Luis Soto	luissoto2233411@gmail.com	Modificar	2025-02-18 14:02:54	El usuario modificó el producto Arroz Indiana 99% Grano Entero 1800gr
Luis Soto	luissoto2233411@gmail.com	Modificar	2025-02-18 15:52:49	El usuario modificó al usuario
Luis Soto	luissoto2233411@gmail.com	Modificar	2025-02-18 15:52:53	El usuario modificó al usuario

En la misma pantalla de bitácoras se encuentra el acceso a la ventana para generar los reportes, hay cuatro tipos de reporte y cada uno brinda información distinta:

Reporte diario:

- Ventas del día seleccionado.

- Producto más vendido del día.
- Lista de productos vendidos.
- Cantidad de pedidos.
- Cantidad de pedidos cancelados.
- Comparación con las ganancias del día anterior.

#### Reporte mensual:

- Total de ventas del mes.
- Productos más vendidos del mes.
- Clientes más frecuentes.
- Ganancia mensual.
- Comparación con la ganancia del mes anterior

#### Reporte anual:

- Ventas totales anuales.
- Mejor mes del año en ventas.
- Mejores productos del año.
- Mejores clientes del año.
- Comparación de ganancias con el año anterior.

#### Reporte histórico:

- Ventas totales desde que se implementó el sistema.
- Mejores productos.
- Mejores meses.

- Mejores clientes.
- Mejor año en ventas.
- Ganancias totales

Para acceder a estos reportes solo se debe seleccionar la fecha que se desea consultar (si es necesario) y el sistema generará el reporte listo para imprimirse en PDF.

**Figura 87** Reporte listo para ser exportado en PDF

18/2/25, 20:48 localhost:5047/Factura/ReporteDiario?fecha=2025-02-12

Reporte Diario - 18/02/2025

Ventas del día  
**Producto más vendido:** Arroz Tio Pelo 99% Grano Entero 1800 gr

Lista de productos vendidos sumados

Producto	Cantidad Vendida
Arroz Tio Pelo 99% Grano Entero 1800 gr	1
Chocolate Twix 50.7 g	1

Resumen de pedidos  
 Cantidad de pedidos: 2  
 Cantidad de pedidos cancelados: 0

Comparación con el día anterior  
 Ventas del día anterior: ₡18090  
 Ventas del día actual: ₡2730  
 Diferencia de ventas: ₡-15360

Detalle de Facturas

Factura ID	Cliente	Fecha	Total
3	María Rodriguez	12/02/2025	₡2730
2	María Rodriguez	12/02/2025	₡0

Generar Reporte PDF

localhost:5047/Factura/ReporteDiario?fecha=2025-02-12 1/1

Imprimir 1 hoja de papel

Destino

Páginas

Diseño

Color

Más ajustes

## 4.4 Resultados

Con esto finalizado la aplicación está lista para empezar a funcionar, se ha cumplido con todos los requerimientos establecidos en el planteamiento del proyecto, se cuenta con una página web la cual permite a los clientes visualizar los productos

disponibles en la tienda y realizar pedidos de recolección al local, estos pueden iniciar sesión, registrarse, cambiar su información de cuenta y administrar su carrito; al realizar un pedido reciben un correo electrónico que les avisa cuando su pedido esté en curso, listo para recoger o haya sido cancelado.

Además, permite a los administradores poder gestionar el inventario y a los usuarios registrados; visualizar y administrar los pedidos recibidos, permitiendo modificar la estructura del pedido en caso de ser necesario y notificando al cliente de estos cambios y cuando puede venir a retirar su pedido.

Se espera que el sitio web cumpla con todas las necesidades de manera efectiva y con un uso intuitivo.

## **5. Capítulo 5: Objetivo 4 Implementación**

### **5.1 Procedimiento**

Con la página web lista solo falta hacerla accesible para el público, para ello utilizaremos la herramienta Microsoft Azure la cual nos permite alojar los distintos componentes de la aplicación en la nube y publicar la solución para que los usuarios pueden interactuar con el sistema.

El primer paso es crearnos una cuenta en Azure, para ello utilizaremos el correo electrónico [superalfoli9@gmail.com](mailto:superalfoli9@gmail.com) el cual se entregará junto a sus credenciales de ingreso al dueño del super al finalizar. Debemos migrar la estructura de la base de datos que utilizamos en el ambiente de prueba a una nueva base creada en la nube, para ello debemos crear el banco de datos en Azure, para lograrlo nos dirigimos al apartado de “Bases de datos SQL” y seleccionamos crear; después llenamos el formulario con la

información necesaria, entre ellos se encuentra, el nombre de la base, el servidor donde se alojará, la suscripción y las opciones de seguridad como la creación del superadministrador el cual se nombrará “sys\_superuser” y su contraseña también se compartirá con el dueño del local.

**Figura 88** Creación de la base de datos en Azure

Home > SQL databases >

## Create SQL Database

Microsoft

Basics Networking Security Additional settings Tags Review + create

Create a SQL Database with your preferred configurations. Complete the Basics tab then go to Review + Create to provision with smart defaults, or visit each tab to customize. [Learn more](#)

Want to try Azure SQL Database for free? Create a free serverless database with the first 100,000 vCore seconds, 32GB of data, and 32GB of backup storage free per month for the lifetime of the subscription. [Learn more](#)

Apply offer (Preview)

SQL Database Hyperscale: Low price, high scalability, and best feature set. [Learn more](#)

### Project details

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription \* Azure for Students

Resource group \* superalfoli  
[Create new](#)

### Database details

Enter required settings for this database, including picking a logical server and configuring the compute and storage resources

Database name \* SuperAlfoliDB  
A database with the same name already exists on this server.

Server \* super-alfoli (East US 2)  
[Create new](#)

Review + create Next: Networking >

**SQL**

Cost summary

**General Purpose (GP\_S\_Gen5\_1)**

Cost per GB (in USD) 0.12

Max storage selected (in GB) x 41.6

**ESTIMATED STORAGE COST / MONTH 4.78 USD**

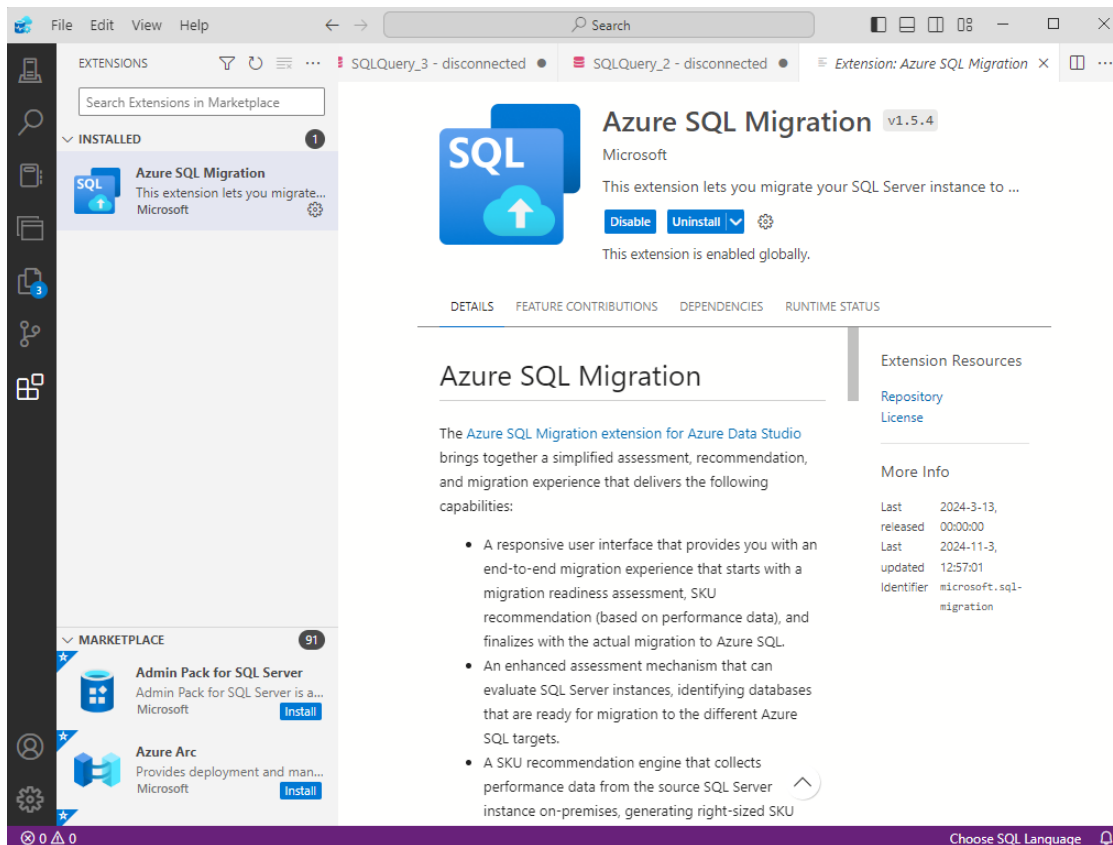
**COMPUTE COST / VCORE SECOND<sup>1</sup> 0.000145 USD**

**NOTES**

<sup>1</sup> Serverless databases are billed in vCore seconds based on a combination of CPU and memory utilization. [Learn more about serverless billing](#)

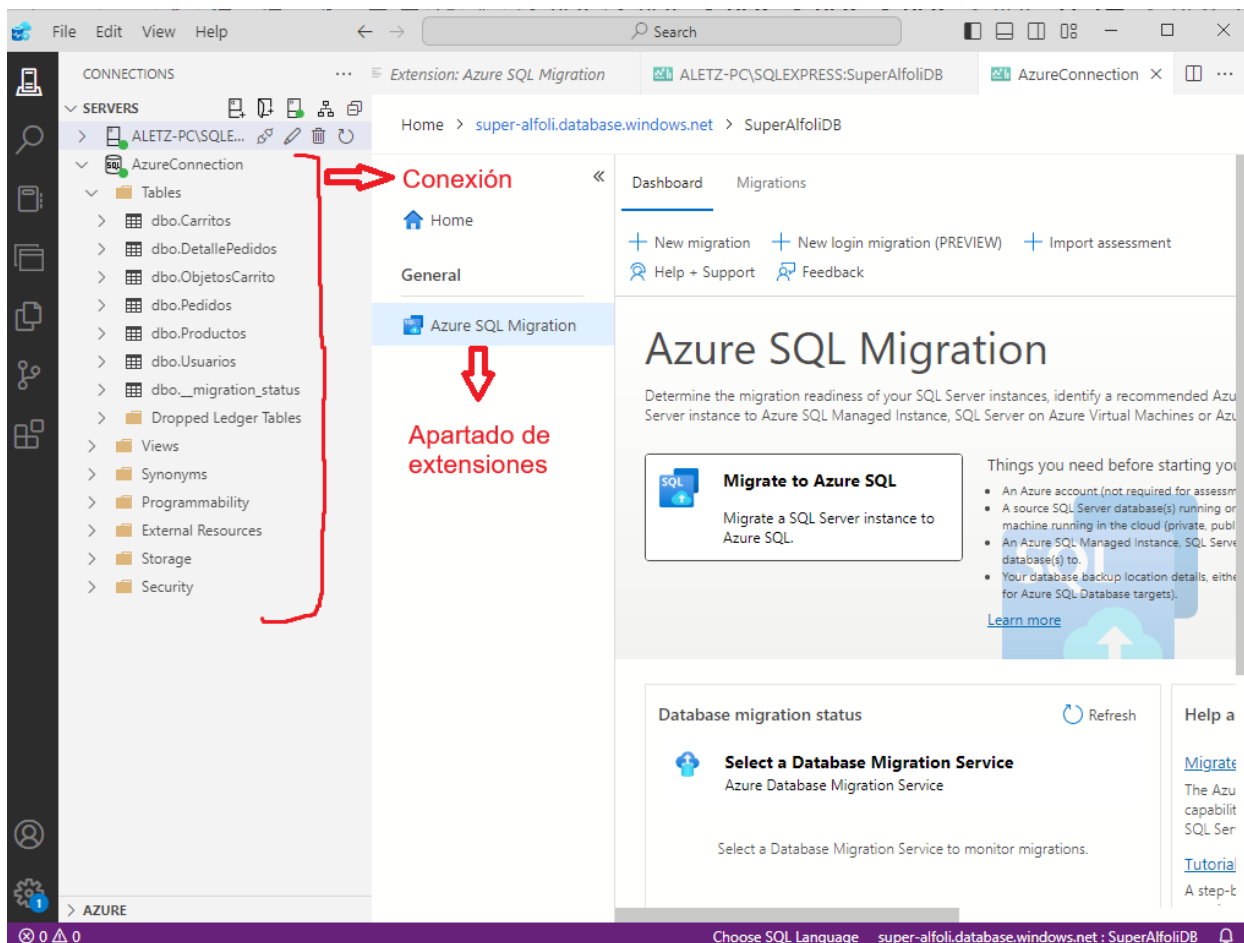
Ahora debemos instalar la aplicación de “Azure Data Studio” y junto a ella la extensión “Azure SQL Migration” la cual nos facilita la migración de la base de datos con un formulario.

**Figura 89** Instalación de la extensión Azure SQL Migration



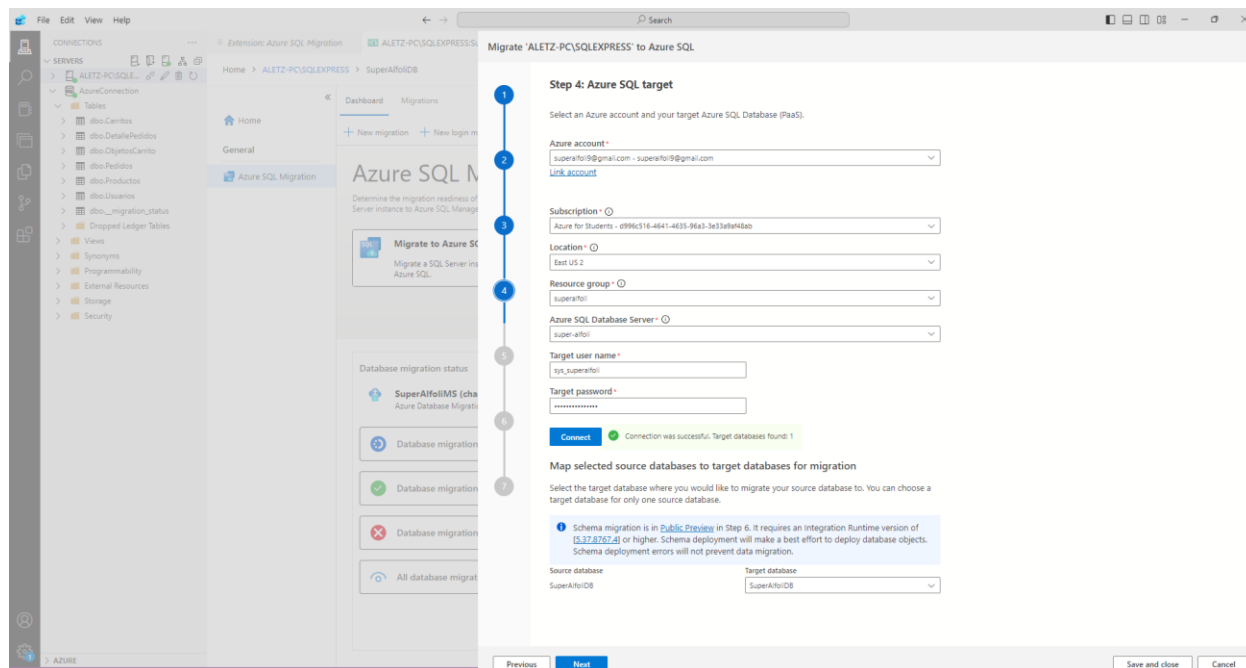
Ahora debemos conectarnos a la base de datos que creamos mediante el superadministrador y ahí seleccionaremos la opción de “Nueva migración” en el apartado de extensiones en el administrador de la conexión.

**Figura 90** Conexión y apartado de extensiones



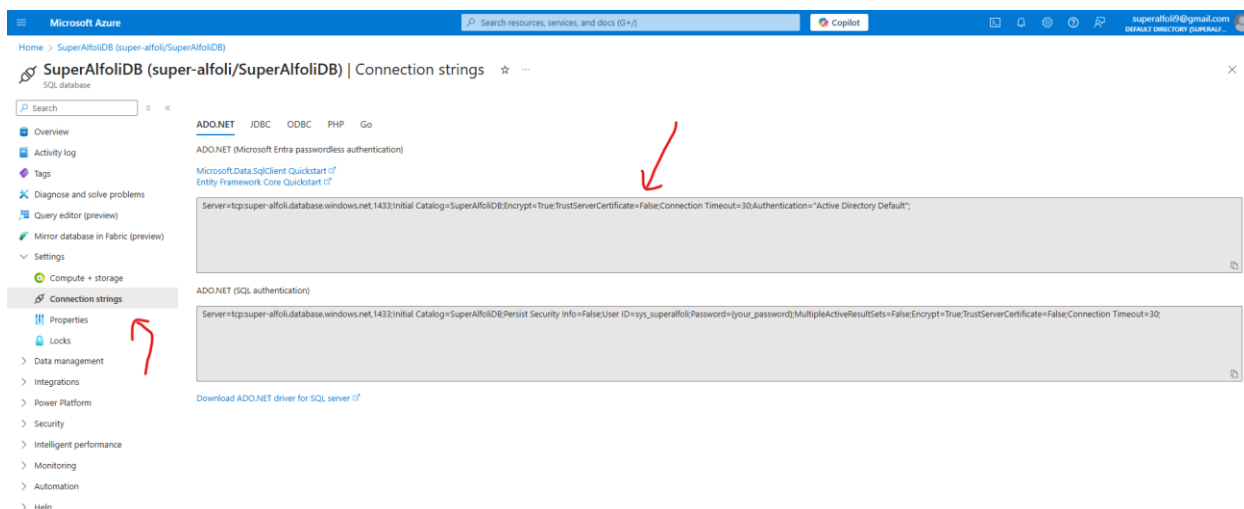
Llenando los campos con la información que se solicita como la base de datos a migrar, la cuenta de Azure destino, el nombre de la base en la nube, las credenciales del superadministrador y las tablas a migrar el proceso se vuelve casi automático, logrando así, tener la réplica de la base en la nube.

**Figura 91** *Proceso para migrar la base de datos local a la nube*



El siguiente paso es subir la aplicación a Azure, pero primero debemos realizar algunos cambios al código de la página. Debemos obtener la nueva cadena de conexión que se utilizará para interactuar con la base de datos, Azure nos brinda una cadena ya hecha en el apartado de "Connection Strings"; esta debe reemplazar a la utilizada en el ambiente de prueba, de esta manera el "ContextDb" interactuará con la nueva base creada.

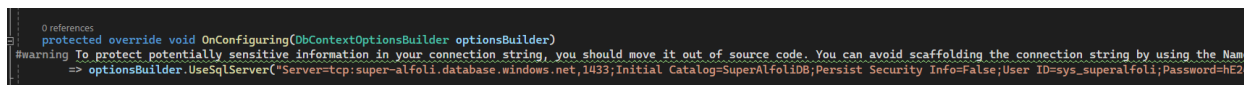
**Figura 92** Obtener la nueva cadena de conexión en Azure



**Figura 93** Implementación de la nueva cadena de conexión en el archivo "appsettings.json"



**Figura 94** Implementación de la nueva cadena de conexión en el archivo "SuperAlfoliDbContext.cs"



Con estos cambios solo debemos exportar el proyecto a la nube, primero debemos revisar que todo ande correctamente y después nos seleccionamos la opción "Publicar SuperAlfoliApp", al iniciar sesión podemos publicar directamente el proyecto en Azure; además nos genera el enlace a la página que debemos brindar a los clientes para que empiecen a utilizar la herramienta (enlace de la página: <https://super-alfoli-azd0b9gahebza3bb.eastus2-01.azurewebsites.net>).

**Figura 95** Opción "Publicar SuperAlfoliApp"

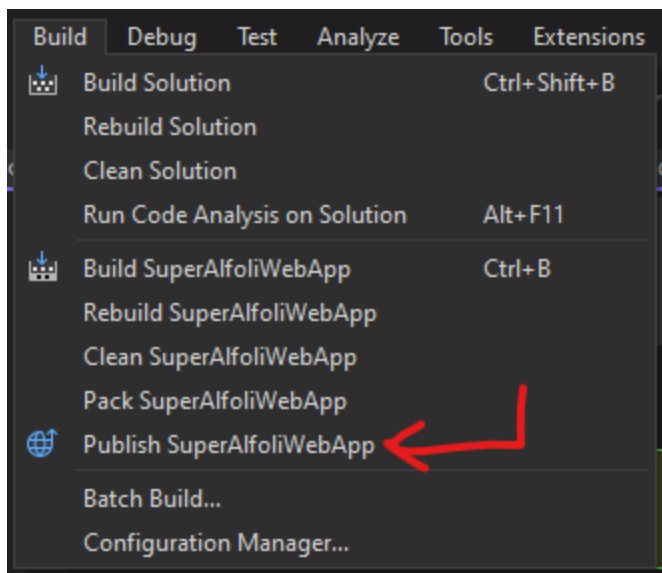
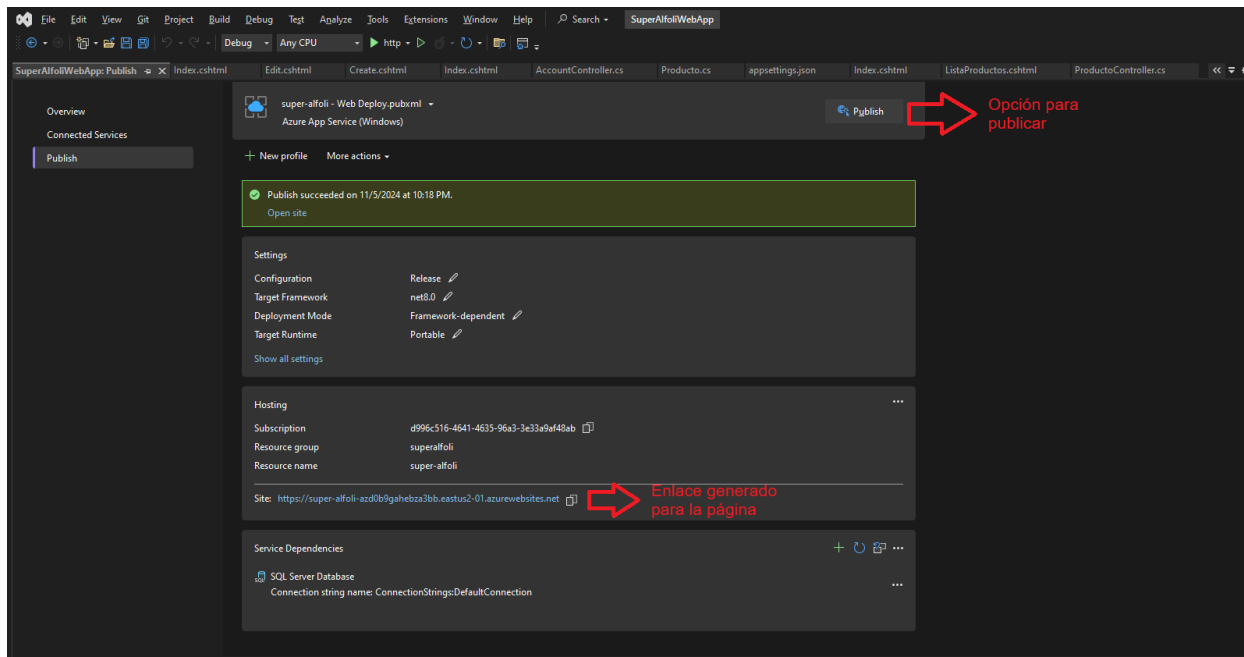


Figura 96 Apartado para publicar el proyecto



## 6. Capítulo 6: Conclusión

El presente trabajo logró desarrollar un sistema que permite la gestión de inventarios y realizar pedidos de recolección al local de manera satisfactoria cumpliendo

con los objetivos establecidos en la planificación del proyecto. Proporcionando herramientas para aumentar la presencia en línea del Super Alfolí. El sistema cuenta con un apartado de visualización de artículos para los clientes, ya que, el abastecedor cuenta con una gran variedad de artículos, los interesados podrán consultar los productos en venta en página web para poder acudir al local, atrayendo nuevos clientes.

El sistema de encriptación de claves mediante SHA-256 protege a las credenciales de los usuarios contra filtraciones, ataques de fuerza bruta e incluso contra miembros internos maliciosos que quieran robar esta información, haciendo que la aplicación sea tan útil como segura. Como menciona (Blanco, 2024) la implementación de esta tecnología aumenta tanto el tiempo de descifrado de contraseñas que hace casi imposible obtenerla por métodos de fuerza bruta. Además, la implementación del sistema para recuperar contraseñas permite al registrado recuperar su cuenta en caso de haber olvidado su código de acceso.

El cliente podrá administrar fácilmente sus compras gracias al módulo de carrito generado, el cual permite agregar una cantidad de productos y actualizarse automáticamente basándose en el inventario actual, ahorrando tiempo y evitando errores.

La gestión de productos y usuarios permite visualizar mediante tablas con filtros y búsquedas los datos de los productos, en caso de que se requiera una nueva categoría puede ser agregada de manera sencilla y permite llevar un inventario de los productos actuales, además, muestra una ventana la cual alerta de los productos próximos a acabarse para evitar quedarse sin mercancía.

Los pedidos pueden ser gestionados de manera intuitiva mediante la vista de los pedidos la cual muestra una lista con los pedidos pendientes recibidos y dos tablas, una para los pedidos finalizados y otra para los cancelados, ambas tablas con sus respectivos filtros y búsquedas para facilitar su uso.

Se puede imprimir las facturas de los pedidos realizados permitiendo llevar un control en caso de que sea necesario.

Con la tabla de bitácoras el administrador podrá llevar a cabo un control de las acciones realizadas por los empleados y otros administradores, esto ayuda a poder reconocer a tiempo un uso malintencionado de la aplicación antes de que sea demasiado tarde.

Además, con la vista de generación de reportes se pueden crear reportes, diarios, mensuales, anuales e históricos para poder realizar una auditoría y tener información de cómo funciona el negocio.

La implementación de todas estas funciones cumple y supera con todo lo expuesto en los alcances del proyecto y aborda las necesidades del negocio y mejora la experiencia del cliente, lo cual es fundamental para lograr los objetivos estratégicos de expansión y eficiencia.

Durante el desarrollo uno de los principales desafíos fue la constante investigación para agregar ciertas funciones críticas como el bloqueo a usuarios no autorizados para evitar que accedan a vistas sensibles, los métodos para enviar correos que permite informar a los usuarios de los estados de sus pedidos, sistema de encriptación de contraseñas con SHA-256, despliegue de listas en formatos específicos y aprendizaje

sobre manejo de aplicaciones en la nube, todo este esfuerzo no solo mejoró la calidad de la página, sino que, también fortaleció el conocimiento en estas áreas de ASP.NET MVC.

Se espera que, con la implementación del sistema, se logren beneficios tangibles, tales como:

- Optimización en la gestión del inventario, reduciendo pérdidas por desorganización o errores humanos.
- Expansión del mercado a clientes de zonas geográficas más alejadas, aumentando el alcance del negocio.
- Incremento en las ventas gracias al sistema de pedidos en línea, facilitando la adquisición anticipada de productos.
- Reducción del tiempo de espera en tienda, agilizando el proceso de recolección de productos.

Si bien el sistema cumple de manera satisfactoria todos los requerimientos, aun se necesita algunas gestiones para poder ser utilizada, se brindará una capacitación a los empleados del Super Alfolí para puedan familiarizarse con la plataforma de la aplicación asegurando una correcta utilización de la misma, de esta manera los miembros podrán incluir a los usuarios empleados o administradores y los productos que tengan a la venta, para que los usuarios lo puedan visualizar sin ningún problema, además, será fundamental la difusión del sistema entre los clientes para maximizar su adopción y garantizar el aprovechamiento de sus funcionalidades.

En conclusión, este proyecto ha cumplido con los objetivos propuestos, logrando, implementar una aplicación que satisfaga las necesidades planteadas por la información obtenida mediante una entrevista, diseñando un sistema que cumple con los requisitos mencionados para facilitar el desarrollo de este mediante el uso de las tecnologías pensadas completando así una página web que realiza todas las acciones necesarias para aumentar la presencia en línea del Super Alfolí. La implementación de este sistema representa un avance significativo en términos de eficiencia y modernización, beneficiando tanto a la empresa como a sus clientes al ofrecer un servicio más accesible y organizado.

### **Recomendaciones**

A partir de los resultados obtenidos de la investigación, se plantean las siguientes recomendaciones:

#### **Realizar pruebas a pequeña escala**

Se recomienda implementar una fase de prueba con un grupo reducido de empleados y clientes para poder evaluar el desempeño del sistema. Esto permitirá identificar mejoras posibles en el sistema.

#### **Fortalecimiento de la seguridad del sistema**

Aunque el sistema cuenta con ciertos sistemas de seguridad como rigurosidad en la contraseña y encriptación, se sugiere implementar ciertas medidas adicionales, como la autenticación de dos pasos.

### **Estrategias de difusión**

Para garantizar que los clientes conozcan el sistema, se sugiere implementar una estrategia de difusión de la página mediante campañas de promoción, para que los usuarios puedan enterarse de su existencia y sacar provecho de la herramienta.

### **Plan de mantenimiento y escalabilidad**

Se recomienda establecer un cronograma de mantenimiento preventivo para corregir fallas del sistema y, además, evaluar el funcionamiento del sistema para diagnosticar la necesidad de alguna actualización en su funcionamiento.

## **Glosario**

**Atributo:** características de una entidad, que definen su comportamiento o propiedad.

**Aplicación web:** software que se utiliza a través de un navegador web y no requiere instalación en dispositivos locales.

**Anteproyecto:** propuesta preliminar de un proyecto que define los objetivos, justificación, problemas y estrategias a seguir antes de su ejecución.

**ASP.NET Core Web App MVC .NET 8.0:** es una plataforma moderna para crear aplicaciones web usando el patrón de diseño MVC (Modelo-Vista-Controlador). Es modular, lo que permite incluir solo los componentes necesarios para crear aplicaciones más eficientes y escalables.

**Back-end:** parte del desarrollo web encargada de la lógica del servidor y la base de datos que no es visible para los usuarios finales.

**Base de datos:** sistema organizado para almacenar, gestionar y recuperar datos electrónicos de manera eficiente.

**Bootstrap v5.1.0:** es un framework front-end que facilita la creación de diseños web responsivos. Incluye herramientas para crear interfaces de usuario con clases utilitarias y componentes predefinidos como botones, formularios y modales.

**Cadena de conexión (Connection String):** es un conjunto de parámetros utilizados por una aplicación para establecer una conexión con una base de datos u otro recurso de datos.

**Carrito de Compras:** es el conjunto de productos que un usuario ha agregado en preparación para hacer un pedido.

**Causa-efecto:** relación entre un evento y su consecuencia o resultado.

**Cliente:** persona que utiliza los servicios o adquiere los productos ofrecidos por una empresa.

**Comercio electrónico (E-commerce):** modelo de negocio que permite la compra y venta de bienes o servicios a través de internet.

**Controlador:** maneja la interacción del usuario, interactúa con el modelo y selecciona la vista adecuada.

**Detalle de Pedido:** información detallada sobre los productos incluidos en un pedido, como la cantidad comprada, precio y total.

**Diagrama:** representación gráfica que organiza información para visualizar relaciones, procesos o estructuras.

**Diagrama de casos de uso:** herramienta visual para representar las interacciones entre los usuarios y un sistema.

**Diagrama de Entidad-Relación:** diagrama utilizado para mostrar las relaciones entre entidades en un sistema de base de datos. Se usa para visualizar las entidades, atributos y relaciones entre ellas.

**Diagrama de flujo:** representación gráfica que muestra los pasos o decisiones en un proceso.

**Diagrama de Gantt:** es una herramienta de gestión de proyectos que permite visualizar las tareas, actividades o hitos de un proyecto en una línea de tiempo, mostrando las relaciones entre las tareas, su duración y su progreso.

**Diagrama Relacional:** representación detallada de un diagrama ER donde las entidades se convierten en tablas y las relaciones en llaves foráneas. Es útil para la implementación técnica de la base de datos.

**Dominio web:** dirección única que identifica un sitio web en Internet.

**Entidad:** representa un objeto o concepto dentro de la base de datos del sistema que contiene atributos relacionados entre sí.

**Entidad-relación:** modelo utilizado en diseño de bases de datos para representar objetos y sus relaciones.

**Foreign Key:** es un campo en una tabla que hace referencia a la clave primaria de otra tabla, estableciendo una relación entre ambas.

**Front-end:** parte visible y con la que interactúan los usuarios en una aplicación web.

**Gestión de inventarios:** proceso de administración de los productos disponibles en una empresa, para garantizar un suministro eficiente y evitar faltantes o excesos.

**Google Fonts:** es un servicio que permite utilizar fuentes tipográficas personalizadas que no están instaladas de manera predeterminada en los sistemas de los usuarios.

**Herramientas digitales:** aplicaciones, tecnologías o sistemas informáticos utilizados para optimizar procesos, como la gestión de inventarios y la presencia en línea.

Hosting: servicio que permite almacenar un sitio web o aplicación en un servidor para hacerlo accesible a través de Internet.

Identity (1,1): es una propiedad de los campos de una tabla en SQL que indica que el campo es auto incrementable, comenzando desde 1 y aumentando de uno en uno cada vez que se inserta un nuevo registro.

Infraestructura tecnológica: conjunto de hardware, software, redes y otros recursos necesarios para operar un sistema.

Interfaz de usuario: parte visual y de interacción de un sistema que el usuario puede ver y utilizar.

Inventario: registro detallado de los bienes disponibles en una empresa.

Microsoft SQL Server 2022: es un sistema de gestión de bases de datos relacional utilizado para almacenar y administrar grandes cantidades de datos. Es ampliamente utilizado por su integración con tecnologías como .NET y Visual Studio.

Modelo: representa los datos y la lógica del negocio. Interactúa con la base de datos.

Notificación por Correo Electrónico: el sistema debe enviar correos electrónicos al usuario para notificarle sobre eventos importantes como la confirmación de un pedido, su estado o el cambio de información.

Pedido: registro de productos seleccionados por un usuario para su compra, que incluye detalles como cantidad, fecha y estado de cada pedido.

Pick-up: servicio de recolección en tienda que permite a los clientes hacer pedidos en línea y recogerlos en el establecimiento físico.

Presencia en línea: estrategia que incluye la creación de un entorno digital (como una página web) para aumentar el alcance, visibilidad y competitividad de una empresa.

Primary Key: es un campo de una tabla en una base de datos que identifica de manera única cada registro. No puede tener valores duplicados.

Producto: elemento que se ofrece en el sistema para la compra, con atributos como nombre, precio, cantidad, categoría, y otros.

Program.cs: es un archivo en un proyecto de ASP.NET Core donde se configuran los servicios y el middleware de la aplicación.

Pruebas de software: evaluación de un sistema para detectar errores y asegurarse de que cumple con los requisitos.

Razor: es un motor de plantillas que permite combinar código C# con HTML para generar dinámicamente las vistas en ASP.NET Core.

Requerimientos Funcionales: son las características y funcionalidades específicas que el sistema debe cumplir, describiendo qué debe hacer el sistema para satisfacer las necesidades del usuario.

Requerimientos No Funcionales: son aspectos del sistema relacionados con el rendimiento, la seguridad, la usabilidad, la disponibilidad y otros factores que no están directamente relacionados con las funcionalidades del sistema pero que son necesarios para su correcto funcionamiento.

Servidor: computadora o sistema que proporciona servicios, datos o recursos a otros dispositivos en una red.

Sistema digital: conjunto de herramientas, tecnologías y métodos desarrollados para automatizar procesos y mejorar la eficiencia operativa de una organización.

Software de gestión: programa diseñado para facilitar tareas específicas como inventarios, ventas o administración.

SQL Server Management Studio (SSMS): es una herramienta de administración para SQL Server que facilita la gestión de bases de datos mediante una interfaz gráfica.

Super Alfolí: abastecedor ubicado en Río Segundo de Alajuela, Costa Rica, fundado en 1991, dedicado a ofrecer productos de calidad a clientes locales.

Tecnología de bajo costo: soluciones tecnológicas económicas que priorizan reducir costos sobre el rendimiento.

Tecnologías en la nube: servicios de almacenamiento y procesamiento de datos a través de internet, permitiendo a las empresas alojar sistemas accesibles desde cualquier lugar.

Token: un código o clave única utilizada para verificar la autenticación o autorización de una sesión, generalmente se asocia con la duración de la sesión.

Usuario final: persona que interactúa directamente con una aplicación o sistema después de su implementación.

Vista: es la interfaz de usuario que muestra los datos al usuario.

## **Lista de abreviaturas**

.cs: tipo de archivo utilizado para código en lenguaje C#.

C#: lenguaje de programación desarrollado por Microsoft.

CRUD: siglas en inglés de Crear, Leer, Actualizar y Eliminar, operaciones básicas en la gestión de datos.

CSS: Cascading Style Sheets (Hojas de Estilo en Cascada).

DB: database (Base de Datos).

ER: Entidad-Relación.

FK: foreign Key (Clave Foránea).

HTTP: HyperText Transfer Protocol (Protocolo de Transferencia de Hipertexto).

MVC: Model-View-Controller (Modelo-Vista-Controlador).

PK: primary Key (Clave Primaria).

SMTP: Simple Mail Transfer Protocol (Protocolo Simple de Transferencia de Correo).

SQL: Structured Query Language (Lenguaje de Consulta Estructurada).

SSMS: SQL Server Management Studio.

UI: User Interface (Interfaz de Usuario).

## Bibliografía

Blanco, M. (2024, noviembre 12). *Specops*. Retrieved from Specops:  
<https://specopssoft.com/blog/sha256-hashing-password-cracking/>

Ecommerce Costa Rica. (28 de enero de 2023). *Ecommerce Costa Rica*. Obtenido de Ventajas de tener un sistema ecommerce en Costa Rica:  
<https://ecommercecostarica.com/index.php/2023/01/28/ventajas-de-tener-un-sistema-ecommerce-en-costa-rica/>

Etecé, E. (9 de Abril de 2024). *concepto*. Obtenido de concepto:  
<https://concepto.de/diagrama-de-flujo/>

GuellCom. (15 de Noviembre de 2023). *¿Por qué tu supermercado debe tener una página web?* Obtenido de GuellCom: <https://guellcom.com/por-que-tu-supermercado-debe-tener-una-pagina-web>

Huayana, Y. (15 de agosto de 2022). *Linkedin.com*. Obtenido de Linkedin.com:  
<https://www.linkedin.com/pulse/click-collect-una-oportunidad-para-generar-m%C3%A1s-ventas-yuriko-huayana/?originalSubdomain=es>

Lopez, R. (19 de 12 de 2021). *PMCOLLEGE*. Obtenido de PMCOLLEGE:  
<https://pmcollege.edu.ni/la-entrevista-tecnica-eficaz-para-recopilar-requerimientos/>

Navarro, M. (12 de Noviembre de 2019). *Ilerna*. Obtenido de Ilerna:  
<https://www.ilerna.es/blog/modelo-entidad-relacion-base-datos>

## Anexos

### Anexo A. Presentación de anteproyecto (A mano)



#### I. INTRODUCCIÓN

---

El presente documento resume los elementos a tomar en consideración para presentar el Anteproyecto para tesis y tesina a desarrollar en el proyecto de graduación para la Escuela de Informática de la Universidad Hispanoamericana, teniendo como objetivo servir de guía en la elaboración de este por parte del estudiante, sin ser contradictorio a lo señalado en el reglamento de la Universidad.

#### II. ANTECEDENTES: FORMULACIÓN DEL ANTEPROYECTO

---

El objetivo de la formulación del anteproyecto es detallar de forma clara y concisa en que va a consistir el posible proyecto de graduación y establecer los elementos básicos necesarios que se requieren para su aprobación, entre ellos considerar el grado de innovación, la profundidad del tema y considerando las líneas de investigación de la escuela.

#### III. LINEAMIENTOS PARA LA PRESENTACIÓN DEL ANTEPROYECTO DE GRADUACIÓN

---

Los siguientes lineamientos son esenciales para poder presentar el Anteproyecto de tesina o tesis para el proyecto de graduación. Es necesario que los cumpla antes de presentar el documento de propuesta. Los mismos deben ir firmados al inicio del Anteproyecto para garantizar que los conoce y los cumple a cabalidad.



Estos se detallan a continuación:

- a. Debe tener una empresa u organización que lo haya aceptado formalmente mediante una carta que debe presentar adjunta a esta solicitud. Recuerde que no puede trabajar en el área o departamento donde va a desarrollar su anteproyecto.
- b. Usted es el encargado de su Proyecto de Graduación y es el principal responsable, el tutor que se le asignará hasta después de la matrícula, es solo un guía del proceso, esto significa que el desarrollo de todo el trabajo depende de usted.
- c. Debe estar pendiente de todo el proceso desde que matricula el anteproyecto hasta su aprobación o rechazo por el comité técnico de la escuela, los resultados se enviarán al correo de la UH que tiene registrado, estar atento.
- d. La escuela facilitará un taller para indicar algunos aspectos generales a contemplar para el desarrollo del anteproyecto, es totalmente gratuito y es una iniciativa para que usted reciba un acompañamiento durante todo este proceso. Su participación es voluntaria y deberá estar atento a las fechas que se publican por los canales oficiales de la universidad.
- e. Debe seguir la **GUIA PARA PRESENTAR ANTEPROYECTOS DE TESIS O TESINA PARA PROYECTOS DE GRADUACIÓN Ed. 01-23**, indicada en este documento para poder cumplir con todos los elementos solicitados para su presentación.
- f. Recuerde que no debería presentar su anteproyecto si no tiene una empresa u organización comprometida con usted, que le permita ir frecuentemente a sus instalaciones y que le facilite datos e información relevante como toma de requerimientos, observación, revisión de documentos entre otros. En caso de que tenga problemas es usted el



responsable de tomar las medidas correspondientes para continuar normalmente el proyecto o en su defecto tendrá que llevarlo de nuevo con la correspondiente erogación de dinero.

Comprendo lo anterior y lo cumpliré según corresponda:

Nombre y firma del estudiante responsable:

Alejandro Salas Casanova 



#### IV. HOJA DE PRESENTACIÓN DEL ANTEPROYECTO

Esta hoja debe venir acompañada de su propuesta, y bajo ninguna situación se recibirá una propuesta sin ella. Gracias.

##### Información general del estudiante

Nombre completo: Alejandro Salas Casanova

Cédula: 118150869

Correo electrónico: alejx200@gmail.com

Teléfono 1: 870143678 Teléfono 2: \_\_\_\_\_

Grado: BACHILLERATO:  LICENCIATURA:

##### Información general del proyecto

TIPOS DE REQUISITO DE GRADUACIÓN A CURSAR:

Proyecto tesina  Proyecto tesis

EMPRESA U ORGANIZACIÓN DONDE REALIZARÁ SU PROYECTO: Indicar a qué se dedica la empresa, si abarca toda la empresa o un área específica, lugar donde se ubica.

El trabajo se realizará en el super Alfali, en el área general  
Esta está ubicada en Rio Segundo de Alajuela

PROBLEMA Y SU CONTEXTO: Describa el problema y todos los detalles involucrados, detalle el contexto de este y como a través de un proceso informático esto podría resolverse.

El Super Alfali carece de presencia en la web lo cual limita su alcance y visibilidad, además, afecta el trato con los clientes, una página web permitiría a los clientes informarse y acceder a los servicios de manera más conveniente  
OPORTUNIDAD DE NEGOCIO O MEJORA A LA SOCIEDAD: Explique cómo impactaría a la empresa u organización la realización de su proyecto o tesis.

Mejoraría la satisfacción del cliente al permitirles visualizar los servicios disponibles, los horarios y la ubicación lo que también ayudaría a obtener nuevos clientes



¿Cuál es el objetivo general de su proyecto? Recuerde redactarlo en función de verbos en infinitivo considerando el qué se va a hacer, ¿cómo y para qué?

Desarrollar un sistema digital que permita gestionar inventarios y recibir pedidos de recolección al local mediante técnicas de desarrollo de aplicaciones para aumentar la presencia en línea del Super Alfóli y así mejorar el alcance y la satisfacción del cliente

¿Cuál sería el título de su **TRABAJO DE INVESTIGACIÓN**? El título parte de la problemática definida así donde se quiere llegar, alcance. Redactarlo en función de **VERBO CON SUFIJO** (Que se va a hacer) + **DONDE** (Platea la contextualización del estudio, la ubicación geográfica o situacional) + **CUANDO** (Remite a la temporalidad)

Potenciando la presencia digital del Super Alfóli: Implementación de estrategias de comercio electrónico para maximizar el alcance y la eficiencia

¿Cuál sería la **LÍNEA DE INVESTIGACIÓN** de la Escuela de Ingeniería Informática que responde a su proyecto? ([Ver Apéndice 3](#)).

- Gobernabilidad de las Tecnologías
- Ingeniería de Software
- Arquitectura e Infraestructura Tecnológica
- Tecnologías Emergentes
- Gestión de los Aprendizajes Digitales



Incluya la carta de aceptación de la empresa u organización firmada por el representante o jefe del área o departamento en donde se desarrollará el proyecto.

Para uso exclusivo del Comité de investigación de la Escuela de Informática de la UH

La Resolución de aprobación, del Comité de Evaluación de Trabajos Finales de Graduación de la UH indica: La presente propuesta ha sido revisada por: \_\_\_\_\_, siendo la misma:

**APROBADA:**       **DENEGADA:**       **REQUIERE MODIFICACIONES:**

**RECOMENDACIONES DE MEJORA:**

---

---

---

---

---

Se envió la notificación por correo el día y la hora señalada, indicar el correo al que se envió:

---

---

## Anexo B. Presentación de anteproyecto (Digital)



responsable de tomar las medidas correspondientes para continuar normalmente el proyecto o en su defecto tendrá que llevarlo de nuevo con la correspondiente erogación de dinero.

Comprendo lo anterior y lo cumpliré según corresponda:

Nombre y firma del estudiante responsable:

Alejandro Salas Casanova 



Incluya la carta de aceptación de la empresa u organización firmada por el representante o jefe del área o departamento en donde se desarrollará el proyecto.

Para uso exclusivo del Comité de investigación de la Escuela de Informática de la UH

La Resolución de aprobación, del Comité de Evaluación de Trabajos Finales de Graduación de la UH indica: La presente propuesta ha sido revisada por: \_\_\_\_\_, siendo la misma:

**APROBADA:**       **DENEGADA:**       **REQUIERE MODIFICACIONES:**

**RECOMENDACIONES DE MEJORA:**

---

---

---

---

---

Se envió la notificación por correo el día y la hora señalada, indicar el correo al que se envió:

---

---



¿Cuál es el objetivo general de su proyecto? Recuerde redactarlo en función de verbos en infinitivo considerando el qué se va a hacer, ¿cómo y para qué?

Desarrollar un sistema digital que permita gestionar inventarios y recibir pedidos de recolección al local mediante técnicas de desarrollo de aplicaciones para aumentar la presencia en línea del Super Alfolí y así mejorar el alcance y la satisfacción del cliente.

¿Cuál sería el título de su **TRABAJO DE INVESTIGACIÓN**? El título parte de la problemática definida así donde se quiere llegar, alcance. Redactarlo en función de **VERBO CON SUFIJO** (Que se va a hacer) + **DONDE** (Platea la contextualización del estudio, la ubicación geográfica o situacional) + **CUANDO** (Remite a la temporalidad)

Potenciando la presencia digital del Super Alfolí: Implementación de estrategias de comercio electrónico para maximizar el alcance y la eficiencia.

¿Cuál sería la **LÍNEA DE INVESTIGACIÓN** de la Escuela de Ingeniería Informática que responde a su proyecto? ([Ver Apéndice 3](#)).

- Gobernabilidad de las Tecnologías
- Ingeniería de Software
- Arquitectura e Infraestructura Tecnológica
- Tecnologías Emergentes
- Gestión de los Aprendizajes Digitales



#### IV. HOJA DE PRESENTACIÓN DEL ANTEPROYECTO

---

Esta hoja debe venir acompañada de su propuesta, y bajo ninguna situación se recibirá una propuesta sin ella. Gracias.

##### Información general del estudiante

Nombre completo: Alejandro Salas Casanova

Cédula: 118150869

Correo electrónico: alejx200@gmail.com

Teléfono 1: 87093678 Teléfono 2: \_\_\_\_\_

Grado: BACHILLERATO:  LICENCIATURA:

##### Información general del proyecto

TIPOS DE REQUISITO DE GRADUACIÓN A CURSAR:

Proyecto tesina  Proyecto tesis

EMPRESA U ORGANIZACIÓN DONDE REALIZARÁ SU PROYECTO: Indicar a qué se dedica la empresa, si abarca toda la empresa o un área específica, lugar donde se ubica.

El trabajo se realizará en el Super Alfolí, en el área general, esta está ubicada en Rio Segundo de Alajuela.

---

PROBLEMA Y SU CONTEXTO: Describa el problema y todos los detalles involucrados, detalle el contexto de este y como a través de un proceso informático esto podría resolverse.

El Super Alfolí carece de presencia en la web, lo que limita su alcance y visibilidad. Además, afecta la experiencia del cliente. Una página web permitiría a los clientes informarse y acceder a los servicios de una manera más conveniente.

---

OPORTUNIDAD DE NEGOCIO O MEJORA A LA SOCIEDAD: Explique cómo impactaría a la empresa u organización la realización de su proyecto o tesis.

Mejoraría la satisfacción del cliente al permitirles visualizar los servicios disponibles, los horarios y la ubicación, lo que también ayudaría a obtener nuevos clientes

---

## Anexo C. Bitácora de reuniones

### Universidad Hispanoamericana

SEDE	Microsoft teams
FECHA	28-05-24
LUGAR	Microsoft teams

#### REGISTRO DE TUTORIAS PARA TESINA O TESIS

SESION	1	2	3	4	5	6	7	8	9	10
	X									

HORA DE INICIO	HORA DE CIERRE	PUNTUALIDAD
6:00pm	6:30pm	

#### TEMAS TRATADOS :

Definir la metodología de trabajo y hablar de las correcciones del anteproyecto

#### ACUERDOS:

Entregar correcciones del anteproyectos.  
Entregar avance de capítulo 2.

#### AVANCES

#### LIMITACIONES

PROXIMA SESIÓN : FECHA  HORA  LUGAR

Firma Estudiante:

Firma Tutor:

*[Firma manuscrita]*  
*Marcelo Sacasa J.*

## Universidad Hispanoamericana

SEDE	Microsoft Teams
FECHA	11/06/24
LUGAR	Microsoft Teams

### REGISTRO DE TUTORIAS PARA TESINA O TESIS

SESION	1	2	3	4	5	6	7	8	9	10
		X								

HORA DE INICIO
6:00pm

HORA DE CIERRE
6:30pm

PUNTUALIDAD
✓

#### TEMAS TRATADOS :

Se revisó el adelanto del objetivo 1 entregado.  
Se planteó la próxima entrega.

#### ACUERDOS:

Corregir avance de objetivo 1.  
Realizar la entrevista y entregar los resultados.

#### AVANCES

Objetivo 1.

#### LIMITACIONES

PROXIMA SESIÓN : FECHA 19/06/24. HORA 6:00pm LUGAR Microsoft Teams

Firma Estudiante:

Firma Tutor:

*Alfonso S.*  
*Marco Sacasa S.*

## Universidad Hispanoamericana

SEDE	Microsoft Teams
FECHA	19/06/24.
LUGAR	Microsoft Teams

### REGISTRO DE TUTORIAS PARA TESINA O TESIS

SESIÓN	1	2	3	4	5	6	7	8	9	10
			X							

HORA DE INICIO
6:00pm

HORA DE CIERRE
6:30pm

PUNTUALIDAD
✓

#### TEMAS TRATADOS :

Se revisó el objetivo 1.  
Se planteó como se debería desarrollar el objetivo 2.

#### ACUERDOS:

Corregir puntos del objetivo 1.  
Comenzar con el objetivo 2.

#### AVANCES

Objetivo 1.

#### LIMITACIONES

**PROXIMA SESIÓN :** FECHA Pendiente. HORA LUGAR Microsoft Teams

Firma Estudiante:  
Firma Tutor:

*[Firma]*  
*Marco Sacasa S.*

## Universidad Hispanoamericana

SEDE	Microsoft Teams
FECHA	28/06/24
LUGAR	Microsoft Teams

### REGISTRO DE TUTORIAS PARA TESINA O TESIS

SESION	1	2	3	4	5	6	7	8	9	10
				X						

HORA DE INICIO
6:00pm

HORA DE CIERRE
6:30pm

PUNTUALIDAD
✓

#### TEMAS TRATADOS :

Se revisa el adelanto del objetivo 2 (Estructura de la base de datos y el diagrama entidad relación).

#### ACUERDOS:

Lectura del profesor para recomendaciones.  
Seguir avanzando con el objetivo 2.

#### AVANCES

Objetivo 2 (Diagrama entidad-relación, Diagrama modelo relacional).

#### LIMITACIONES

PROXIMA SESIÓN : FECHA  HORA  LUGAR

Firma Estudiante:

Firma Tutor:

*Dyanada S.*  
*Marco Sacasa S.*

## Universidad Hispanoamericana

SEDE	Microsoft Teams
FECHA	11/07/24
LUGAR	Microsoft Teams

### REGISTRO DE TUTORIAS PARA TESINA O TESIS

SESION	1	2	3	4	5	6	7	8	9	10
					X					

HORA DE INICIO
6:00pm

HORA DE CIERRE
6:30pm

PUNTUALIDAD
✓

#### TEMAS TRATADOS :

Revisión del avance del objetivo 2.  
Revisión del avance anterior del objetivo 2.

#### ACUERDOS:

Lectura por parte del profesor para recomendaciones.  
Seguir avanzando con el objetivo 2.

#### AVANCES

Objetivo 2 (Análisis de requerimientos, Diagrama de flujo).

#### LIMITACIONES

PROXIMA SESIÓN : FECHA  HORA  LUGAR

Firma Estudiante:  
Firma Tutor:

*Alfonso S.*  
*Marco Sacasa S.*

## Universidad Hispanoamericana

<b>SEDE</b>	Microsoft Teams
<b>FECHA</b>	05/08/24
<b>LUGAR</b>	Microsoft Teams

### REGISTRO DE TUTORIAS PARA TESINA O TESIS

SESION	1	2	3	4	5	6	7	8	9	10
						X				

<b>HORA DE INICIO</b>
6:00pm

<b>HORA DE CIERRE</b>
6:30pm

<b>PUNTUALIDAD</b>
✓

#### TEMAS TRATADOS :

Entrega del objetivo 2 completo.  
Entrega del documento que une al objetivo 1 y 2.

#### ACUERDOS:

Empezar a trabajar en el objetivo 3.  
Lectura del profesor para recomendaciones.

#### AVANCES

Objetivo 2 (Diagramas de caso de uso).

#### LIMITACIONES

**PROXIMA SESIÓN :** FECHA Pendiente HORA LUGAR Microsoft Teams

Firma Estudiante:  
Firma Tutor:

*[Firma]*  
*Marco Sacasa S.*

## Universidad Hispanoamericana

SEDE	Microsoft Teams
FECHA	
LUGAR	Microsoft Teams

### REGISTRO DE TUTORIAS PARA TESINA O TESIS

SESIÓN	1	2	3	4	5	6	7	8	9	10
							X			

HORA DE INICIO	HORA DE CIERRE	PUNTUALIDAD
6:00pm	6:30pm	✓

#### TEMAS TRATADOS :

Revisión del objetivo 1 y 2.  
Revisión del objetivo 3.

#### ACUERDOS:

Seguir trabajando en el objetivo 3.

#### AVANCES

Objetivo 3 (Módulo de inicio, despliegue de productos y CRUD de productos).

#### LIMITACIONES

PROXIMA SESIÓN : FECHA | Pendiente | HORA | | LUGAR | Microsoft Teams

Firma Estudiante:

Firma Tutor:

*Alfonso*  
*Marco Sacasa S.*

## Universidad Hispanoamericana

SEDE	Microsoft Teams
FECHA	15/09/24
LUGAR	Microsoft Teams

### REGISTRO DE TUTORIAS PARA TESIS O TESIS

SESION	1	2	3	4	5	6	7	8	9	10
								X		

HORA DE INICIO
6:00 pm

HORA DE CIERRE
6:30 pm

PUNTUALIDAD
✓

#### TEMAS TRATADOS :

Revisión final del objetivo 1 y 2.  
Revisión de los avances del objetivo 3 (Módulo de inicio, despliegue de productos y CRUD de productos).

#### ACUERDOS:

Corregir aspectos del objetivo 3.  
Continuar con el objetivo 3.

#### AVANCES

Objetivo 3 (Módulo de inicio de sesion, manejo de carrito).

#### LIMITACIONES

**PROXIMA SESIÓN :** FECHA  HORA  LUGAR

Firma Estudiante:

Firma Tutor:

*Diana S.*  
*Marco Sacasa S.*

## Universidad Hispanoamericana

SEDE	Microsoft Teams
FECHA	30/09/24
LUGAR	Microsoft Teams

### REGISTRO DE TUTORIAS PARA TESINA O TESIS

SESION	1	2	3	4	5	6	7	8	9	10
									X	

HORA DE INICIO
6:00 pm

HORA DE CIERRE
6:30 pm

PUNTUALIDAD
✓

#### TEMAS TRATADOS :

Revisión del objetivo 3 (Módulo de inicio de sesion, manejo de carrito y documentación escrita).

#### ACUERDOS:

Corregir detalles del objetivo 3.  
Completar el objetivo 3 (Finalizar documentación escrita y aplicación).

#### AVANCES

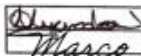
Módulo de inicio de sesion, manejo de carrito y documentación escrita.

#### LIMITACIONES

**PROXIMA SESIÓN :** FECHA Por definir. HORA Por definir. LUGAR Microsoft Teams

Firma Estudiante:

Firma Tutor:

  
Marco Sacasa J.

## Universidad Hispanoamericana

SEDE	Microsoft Teams
FECHA	22/10/24
LUGAR	Microsoft Teams

### REGISTRO DE TUTORIAS PARA TESINA O TESIS

SESIÓN	1	2	3	4	5	6	7	8	9	10
										X

HORA DE INICIO	HORA DE CIERRE	PUNTUALIDAD
6:00pm	6:30pm	✓

#### TEMAS TRATADOS :

Revisión de la aplicación y el trabajo escrito.

#### ACUERDOS:

Corregir los detalles expuestos de la aplicación y el trabajo escrito.

#### AVANCES

Aplicación y el trabajo escrito en revisión.

#### LIMITACIONES

**PROXIMA SESIÓN :** FECHA  HORA  LUGAR

Firma Estudiante:  
Firma Tutor:



## Anexo D. Carta de visto bueno del profesor tutor

### CARTA DEL TUTOR

Puntarenas, 7 de noviembre de 2024

**Esteban José González Vargas**  
**Sub Director Ingeniería Informática**  
**Universidad Hispanoamericana**  
**Sede Llorente**

Estimado

El estudiante **Alejandro Salas Casanova**, cédula de identidad número **118150869** me ha presentado, para efectos de revisión y aprobación, el trabajo de investigación denominado **“Potenciando la Presencia Digital del Super Alfolí: Implementación de Estrategias de Comercio Electrónico para Maximizar el Alcance y la Eficiencia”**, el cual ha elaborado para optar por el grado académico de Bachiller en Ingeniería Informática.

En mi calidad de tutor, he verificado que se han hecho las correcciones indicadas durante el proceso de tutoría y he evaluado los aspectos relativos a la elaboración del problema, objetivos, justificación; antecedentes, marco teórico, marco metodológico, tabulación, análisis de datos; conclusiones y recomendaciones.

De los resultados obtenidos por el postulante, se obtiene la siguiente calificación:

a) Original del tema	10%	8%
b) Cumplimiento de entrega de avances	20%	20%
c) Coherencia entre los objetivos, los instrumentos aplicados y los resultados de la investigación	30%	30%
d) Relevancia de las conclusiones y recomendaciones	20%	18%
e) Calidad, detalle del marco teórico	20%	18%
<b>TOTAL</b>		<b>94%</b>

En virtud de la calificación obtenida, se avala el traslado al proceso de lectura.

Atentamente,

MARCO VINICIO  
 SACASA SOTO  
 (FIRMA)

Firmado digitalmente  
 por MARCO VINICIO  
 SACASA SOTO (FIRMA)  
 Fecha: 2024.11.07  
 18:51:57 -06'00'

**Marco Sacasa Soto**

**Cédula 602820120**



## Anexo F. Carta de aceptación de la empresa



**Super Alfóli**  
**Río Segundo, Alajuela**  
**Teléfono: (+506) 2442-6856**

1 de abril de 2024  
Oficio N° 001

**Señor**  
**Julián Córdoba Sanabria**  
**Coordinador de Investigación**  
**Universidad Hispanoamericana**

Estimado Señor:

Reciba un cordial saludo. Por este medio, me permito manifestar el interés del **Super Alfóli** en el proyecto denominado **"Potenciando la Presencia Digital del Super Alfóli: Implementación de Estrategias de Comercio Electrónico para Maximizar el Alcance y la Eficiencia"** presentado por el estudiante **Alejandro Salas Casanova**, cédula de identidad **118150869** para ser realizado en el **área general**, este proyecto es de suma importancia para nuestra **empresa** ya que con él **se aumenta la presencia en línea del Super Alfóli aumentando el alcance de nuestra empresa y mejorando la calidad del servicio, problema con el que hemos luchado recientemente**, adicionalmente el **gerente Jose Luis Porras Solís** participará en la defensa del estudiante.

Quedo atento a cualquier consulta o detalle adicional.

Atentamente,

**Jose Luis Porras Solís**

**Gerente**

**Super Alfóli**

## Anexo G. Carta de lector

### CARTA DE LECTOR

San José,

Universidad Hispanoamericana  
Sede Llorente  
Carrera de Informática

Estimado señor

El estudiante Alejandro Salas Casanova, cédula de identidad 118150869, me ha presentado para efectos de revisión y aprobación, el trabajo de investigación denominado "Potenciando la Presencia Digital del Super Alfóli: Implementación de Estrategias de Comercio Electrónico para Maximizar el Alcance y la Eficiencia".

He revisado y he hecho las observaciones relativas al contenido analizado, particularmente lo relativo a la coherencia entre el marco teórico y análisis de datos, la consistencia de los datos recopilados y la coherencia entre éstos y las conclusiones; asimismo, la aplicabilidad y originalidad de las recomendaciones, en términos de aporte de la investigación.

Por consiguiente, este trabajo cuenta con mi aval para ser presentado en la defensa pública.

Atte.

Randall  
Vargas  
Villalobos

Firmado  
digitalmente por  
Randall Vargas  
Villalobos  
Fecha: 2024.12.20  
16:12:51 -06'00'

Firma  
Randall Vargas Villalobos  
Cédula: 1-1140-0113

## Anexo H. Autorización de publicación



**UNIVERSIDAD HISPANOAMERICANA  
CENTRO DE INFORMACION TECNOLOGICO (CENIT)  
CARTA DE AUTORIZACIÓN DE LOS AUTORES PARA LA CONSULTA, LA  
REPRODUCCION PARCIAL O TOTAL Y PUBLICACIÓN ELECTRÓNICA  
DE LOS TRABAJOS FINALES DE GRADUACION**

San José, martes, 11 de marzo de 2025.

Señores:  
Universidad Hispanoamericana  
Centro de Información Tecnológico (CENIT)

Estimados Señores:

El suscrito (a) Alejandro Salas Casanova, con número de identificación 118150869, autor (a) del trabajo de graduación titulado Potenciando la Presencia Digital del Super Alfolí: Implementación de Estrategias de Comercio Electrónico para Maximizar el Alcance y la Eficiencia, presentado y aprobado en el año 2025 como requisito para optar por el título de **Bachillerato**,  SÍ /  NO autorizo al Centro de Información Tecnológico (CENIT) para que con fines académicos, muestre a la comunidad universitaria la producción intelectual contenida en este documento.

De conformidad con lo establecido en la Ley sobre Derechos de Autor y Derechos Conexos N° 6683, Asamblea Legislativa de la República de Costa Rica.

Cordialmente,

  
\_\_\_\_\_  
**Alejandro Salas Casanova**  
118150869



**ANEXO 1 (Versión en línea dentro del Repositorio)  
LICENCIA Y AUTORIZACIÓN DE LOS AUTORES PARA PUBLICAR Y  
PERMITIR LA CONSULTA Y USO**

**Parte 1. Términos de la licencia general para publicación de obras en el repositorio institucional**

Como titular del derecho de autor, confiero al Centro de Información Tecnológico (CENIT) una licencia no exclusiva, limitada y gratuita sobre la obra que se integrará en el Repositorio Institucional, que se ajusta a las siguientes características:

- a) Estará vigente a partir de la fecha de inclusión en el repositorio, el autor podrá dar por terminada la licencia solicitándolo a la Universidad por escrito.
- b) Autoriza al Centro de Información Tecnológico (CENIT) a publicar la obra en digital, los usuarios puedan consultar el contenido de su Trabajo Final de Graduación en la página Web de la Biblioteca Digital de la Universidad Hispanoamericana
- c) Los autores aceptan que la autorización se hace a título gratuito, por lo tanto, renuncian a recibir beneficio alguno por la publicación, distribución, comunicación pública y cualquier otro uso que se haga en los términos de la presente licencia y de la licencia de uso con que se publica.
- d) Los autores manifiestan que se trata de una obra original sobre la que tienen los derechos que autorizan y que son ellos quienes asumen total responsabilidad por el contenido de su obra ante el Centro de Información Tecnológico (CENIT) y ante terceros. En todo caso el Centro de Información Tecnológico (CENIT) se compromete a indicar siempre la autoría incluyendo el nombre del autor y la fecha de publicación.
- e) Autorizo al Centro de Información Tecnológica (CENIT) para incluir la obra en los índices y buscadores que estimen necesarios para promover su difusión.
- f) Acepto que el Centro de Información Tecnológico (CENIT) pueda convertir el documento a cualquier medio o formato para propósitos de preservación digital.
- g) Autorizo que la obra sea puesta a disposición de la comunidad universitaria en los términos autorizados en los literales anteriores bajo los límites definidos por la universidad en las "Condiciones de uso de estricto cumplimiento" de los recursos publicados en Repositorio Institucional.

SI EL DOCUMENTO SE BASA EN UN TRABAJO QUE HA SIDO PATROCINADO O APOYADO POR UNA AGENCIA O UNA ORGANIZACIÓN, CON EXCEPCIÓN DEL CENTRO DE INFORMACIÓN TECNOLÓGICO (CENIT), EL AUTOR GARANTIZA QUE SE HA CUMPLIDO CON LOS DERECHOS Y OBLIGACIONES REQUERIDOS POR EL RESPECTIVO CONTRATO O ACUERDO.

## Anexo I. Acta de aprobación de defensa



## ACTA DE GRADUACION

De conformidad con la Reglamentación de la Universidad se presenta el estudiante **SALAS CASANOVA ALEJANDRO** Cédula 1-1815-0869 para defender su Trabajo de Graduación titulado "Potenciando la Presencia Digital del Super Alfóli: Implementación de Estrategias de Comercio Electrónico para Maximizar el Alcance y la Eficiencia" para optar al Grado de Bachillerato en la carrera de Ingeniería Informática y bajo la modalidad de TESINA.

Estando presente los siguientes miembros del Tribunal Calificador:

GONZALEZ VARGAS ESTEBAN JOSE

SACASA SOTO MARCO VINICIO

VARGAS VILLALOBOS RANDALL ANDRES

Una vez concluida la exposición, el periodo de preguntas y efectuada la deliberación, se acordó otorgarle el siguiente resultado:

Calificación:	80	Aprobado:	X	Reprobado:	
NOTA: La nota mínima de aprobación es de 80% según el reglamento de evaluación.					
<b>Firmas del tribunal Calificador:</b>	MARCO VINICIO SACASA SOTO (FIRMA)	Firmado digitalmente por MARCO VINICIO SACASA SOTO (BIMAI) Fecha: 2025.03.24 11:02:08 -06:00	Randall Vargas Villalobos	Firmado digitalmente por ESTEBAN JOSE GONZALEZ VARGAS (FIRMA)	Fecha: 1/24/25

ORIGINAL EXPEDIENTE - COPIA ESTUDIANTE



## ACTA DE GRADUACION

De conformidad con la Reglamentación de la Universidad se presenta el estudiante **SALAS CASANOVA ALEJANDRO** Cédula 1-1815-0869 para defender su Trabajo de Graduación titulado "Potenciando la Presencia Digital del Super Alfóli: Implementación de Estrategias de Comercio Electrónico para Maximizar el Alcance y la Eficiencia" para optar al Grado de Bachillerato en la carrera de Ingeniería Informática y bajo la modalidad de TESINA.

Estando presente los siguientes miembros del Tribunal Calificador:

GONZALEZ VARGAS ESTEBAN JOSE

SACASA SOTO MARCO VINICIO

VARGAS VILLALOBOS RANDALL ANDRES

Una vez concluida la exposición, el periodo de preguntas y efectuada la deliberación, se acordó otorgarle el siguiente resultado:

Calificación:	80	Aprobado:	X	Reprobado:	
NOTA: La nota mínima de aprobación es de 80% según el reglamento de evaluación.					
<b>Firmas del tribunal Calificador:</b>			Randall Vargas Villalobos	Firmado digitalmente por ESTEBAN JOSE GONZALEZ VARGAS (FIRMA)	Fecha: 1/24/25

ORIGINAL EXPEDIENTE - COPIA ESTUDIANTE