

Área de Conocimiento de Tecnología de la Información y
Comunicación

DESARROLLO DE UN SISTEMA WEB PARA EL CONTROL DE INVENTARIO Y FACTURACIÓN DE LA PULPERÍA JEHOVÁ JIREH

TRABAJO MONOGRÁFICO PARA OPTAR AL TÍTULO DE
INGENIERO EN COMPUTACIÓN

Elaborado por:

Br. Cristhiam Omar
Suárez Núñez
Carnet: 2019 – 1244U

Tutor:

MSc. Luis Eduardo
Chávez Mairena

Br. Hosmand Rafael
Belli Céspedes
Carnet: 2019 – 0152U



Secretaría Académica
DACTIC

SECRETARIA DE ÁREA ACADÉMICA

F-8: CARTA DE FINALIZADO PLAN DE ASIGNATURA

El Suscrito Secretario del **ÁREA DE CONOCIMIENTO DE TECNOLOGÍA DE LA INFORMACIÓN Y COMUNICACIÓN** hace constar que:

BELLI CESPEDES HOSMAND RAFAEL

Carné: **2019-0152U** Turno: **Diurno** Plan de Asignatura: **2015** de conformidad con el Reglamento Académico vigente en la Universidad, ha aprobado todas las asignaturas correspondientes a la carrera de **INGENIERÍA EN COMPUTACIÓN**, en el año 2023 y solo tiene pendiente la realización de una de las formas de culminación de estudio.

Se extiende la presente **CARTA DE FINALIZADO PLAN DE ASIGNATURA**, a solicitud del interesado en la ciudad de Managua, a los siete días del mes de octubre del año dos mil veinte y cinco.

Atentamente,


Luisa Mercado
MSc. Luisa Massiel Mercado Gutiérrez
SECRETARIO DE ÁREA ACADÉMICA



Móvil: (505) 83803517



Recinto Universitario Simón Bolívar
Avenida Universitaria.
Managua, Nicaragua.
Apdo: 5595

IMPRESO POR SISTEMA DE REGISTRO ACADEMICO EL 07-oct-2023 hazzely.orozco@dactic.uni.edu.ni
www.uni.edu.ni



Secretaría Académica
DACTIC

SECRETARIA DE ÁREA ACADÉMICA

F-8: CARTA DE FINALIZADO PLAN DE ASIGNATURA

El Suscrito Secretario del ÁREA DE CONOCIMIENTO DE TECNOLOGÍA DE LA INFORMACIÓN Y COMUNICACIÓN hace constar que:

SUÁREZ NUÑEZ CRISTHIAM OMAR

Carné: 2019-1244U Turno: Diurno Plan de Asignatura: 2015 de conformidad con el Reglamento Académico vigente en la Universidad, ha aprobado todas las asignaturas correspondientes a la carrera de INGENIERÍA EN COMPUTACIÓN, en el año 2023 y solo tiene pendiente la realización de una de las formas de culminación de estudio.

Se extiende la presente **CARTA DE FINALIZADO PLAN DE ASIGNATURA**, a solicitud del interesado en la ciudad de Managua, a los siete días del mes de octubre del año dos mil veinte y cinco.

Atentamente,



Luisa Mercado

MSc. Luisa Massiel Mercado Gutiérrez
SECRETARIO DE ÁREA ACADÉMICA



Móvil: (505) 83803517



Recinto Universitario Simón Bolívar
Avenida Universitaria.
Managua, Nicaragua.
Apdo: 5595

IMPRESO POR SISTEMA DE REGISTRO ACADÉMICO EL 07-oct. 2023
12hazzely.orozco@dactic.uniedu.ni
www.uniedu.ni

Managua 31 de octubre 2025

MSc. Claudia Benavides
Directora Área de Conocimiento de Tecnología de la Información y Comunicación,
DACTIC

Sus manos:

Estimada directora Benavides.

Mediante la presente me dirijo a usted con el objetivo de certificar que he revisado y evaluado detenidamente el trabajo monográfico titulado **"Desarrollo de un Sistema Web para el Control de Inventario y Facturación de la Pulperia Jehová Jireh"**.

Realizado por los bachilleres **Cristhiam Omar Suárez Núñez con carnet 2019-1244U** y **Hosmand Rafael Belli Céspedes con carnet 2019-0152U**, ambos estudiantes del programa académico de Ingeniería en Computación.

El trabajo monográfico cumple con todas las normativas y requisitos establecidos por la universidad para ser presentado y defendido ante el tribunal evaluador en la fecha que usted programe.

Sin más que decir, me despido cordialmente.



MSc. Luis Eduardo Chávez Mairena
Docente Titular programa Ingeniería en Computación
Tutor

Cc. Parte interesada



Área de Conocimiento de Tecnología de la Información y Comunicación

Managua, 07 de enero 2025

Bachilleres
Br. Cristhiam Omar Suárez Núñez 2019-1244U
Br. Hosmand Rafael Belli Céstedes 2019-0152U
Egresados Programa académico Ingeniería en Computación
Sus manos.

Estimados Bachilleres:

Reciban cordiales saludos de mi parte y deseándole el mejor de los éxitos en sus actividades diarias.

Por medio de la presente, les comunico la aprobación e inscripción del Protocolo de trabajo monográfico, titulado: **“Desarrollo de un sistema web para el control de inventario y facturación de la pulperia “Jehová Jireh”**, el cual cumple con los requisitos establecidos en el capítulo II de la normativa para los trabajos Monográficos de la Universidad Nacional de Ingeniería (UNI) como forma de culminación de estudios.

No omito manifestar, que el Maestro **MSc. Luis Eduardo Chávez Mairena**, es el docente encargado de acompañarlos con responsabilidad y compromiso como tutor en el proceso de desarrollo de la monografía.

Así mismo, en correspondencia con la Normativa para los Trabajos Monográficos, a partir de la fecha de aprobación e inscripción tendrán un máximo de 12 meses para la ejecución y entrega de los ejemplares para la organización de la pre-defensa y defensa del mismo.

Sin más a que hace: referencia, les deseo el mejor de los éxitos en la culminación de esta etapa, les saludo.

Atentamente,


MSc. Claudia Benavides Rugama
Directora Área de Conocimiento de
Tecnología de la Información y Comunicación



CC: MSc. Luis Eduardo Chávez Mairena – Tutor
Secretaría Académica.
Archivo DACTIC 2025

 Móvil: (505) 8588 8333

 Recinto Universitario Simón Bolívar
Avenida Universitaria.
Managua. Nicaragua.
Apdo: 5595

 www.uni.edu.ni

Resumen del tema

El presente trabajo monográfico se centra en la pulperia Jehová Jireh, un negocio ubicado en Granada, Diria, que enfrenta desafíos significativos en la gestión de inventario y facturación debido a la dependencia de procesos manuales. Esta situación ha generado ineficiencias como errores de cálculo, imprecisión en los niveles de existencias y dificultad para rastrear transacciones.

Para dar solución a esta problemática, se desarrolló un sistema web de inventario y facturación, diseñado para optimizar y facilitar los procesos operativos del negocio. El sistema permite controlar el inventario, emitir facturas, generar reportes detallados y gestionar la seguridad de los datos.

El desarrollo se realizó bajo el entorno tecnológico de Microsoft, utilizando ASP.NET con el patrón de arquitectura MVC, y como base de datos SQL Server, lo que garantiza escalabilidad, seguridad y rendimiento. Además, se aplicó la metodología ágil Scrum para guiar el diseño, desarrollo e implementación del sistema, permitiendo una construcción iterativa y centrada en las necesidades reales del cliente. Como resultado, se obtuvo una herramienta funcional que mejora el control operativo, reduce errores humanos y contribuye al crecimiento sostenible del negocio.

Índice de contenido

I.	<i>Introducción</i>	1
II.	<i>Antecedentes</i>	2
III.	<i>Justificación</i>	3
IV.	<i>Objetivos</i>	4
4.1.	Objetivo General	4
4.2.	Objetivos Específicos	4
V.	<i>Marco teórico</i>	5
5.1.	Terminología aplicada al negocio	5
5.1.1.	Clientes	5
5.1.2.	Proveedores	5
5.1.3.	Empresa	5
5.1.4.	Productos	5
5.1.5.	Categorías	6
5.1.6.	Inventario	6
5.1.7.	Facturas	6
5.1.8.	Mermas	6
5.1.9.	Regalías	6
5.2.	Metodología Scrum	7
5.2.1.	Scrum Team	9
5.2.2.	Ceremonias de Scrum	11
5.2.3.	Artefactos Scrum	14
5.3.	Tecnologías y Herramientas	16
5.3.1.	Visual Studio	16
5.3.2.	.NET	16
5.3.3.	ASP.NET	16
5.3.4.	C#	16
5.3.5.	SQL Server	17
5.3.6.	Modelo Vista Controlador (MVC)	17

5.3.7.	Gitlab	17
5.3.8.	HTML.....	17
5.3.9.	CSS	18
5.3.10.	JavaScript.....	18
5.3.11.	Bootstrap	18
5.3.12.	Power BI Desktop.....	18
5.3.13.	IIS web service	19
5.3.14.	SmarterASP.NET.....	19
5.3.15.	ADO.NET.....	19
VI. Análisis y presentación de resultados. Metodología y desarrollo del tema 20		
6.1.	Creación del Product Backlog	20
6.1.1.	Elaboración y priorización del Product Backlog	20
6.1.2.	Criterios de aceptación	26
6.2.	Estructura de ejecución de la Metodología Scrum.....	27
6.2.1.	Scrum Team	27
6.2.2.	Sprints	27
6.3.	Resultados	32
6.3.1.	Sprint 1	32
6.3.2.	Sprint 2	56
6.3.3.	Diseño de Base de Datos Completo	84
6.3.4.	Pruebas unitarias.....	85
VII. Conclusiones y recomendaciones		93
7.1.	Conclusiones	93
7.2.	Recomendaciones.....	94
7.3.	Limitaciones	95
VIII. Bibliografía		96
IX.	Anexos	96

Índice de Figuras

<i>Figura 1: Procesos del scrum [6]</i>	8
<i>Figura 2: Comunicación del scrum team [6]</i>	9
<i>Figura 3: Resumen de Sprints</i>	28
<i>Figura 4: Plantilla del Sprint Backlog</i>	28
<i>Figura 5: Columnas de control</i>	29
<i>Figura 6: Ceremonias Scrum</i>	30
<i>Figura 7: Conexión con gitlab</i>	31
<i>Figura 8: Creación del proyecto en ASP.NET</i>	33
<i>Figura 9: Conexión con la base de datos</i>	33
<i>Figura 10: Configuración inicial del proyecto</i>	33
<i>Figura 11: Procedimiento almacenado Login</i>	36
<i>Figura 12: Lógica de Login</i>	36
<i>Figura 13: Controlador Login</i>	37
<i>Figura 14: Vista Login</i>	37
<i>Figura 15: Tabla de Usuario</i>	38
<i>Figura 16: Procedimiento almacenado Usuario</i>	38
<i>Figura 17: Clase Usuario</i>	39
<i>Figura 18: Lógica de Usuario</i>	39
<i>Figura 19: Controlador de Usuario</i>	40
<i>Figura 20: Vista de Usuario</i>	40
<i>Figura 21: Crear Usuario</i>	41
<i>Figura 22: Tablas de Roles</i>	41
<i>Figura 23: Procedimiento almacenado Rol</i>	42
<i>Figura 24: Clase Rol</i>	42
<i>Figura 25: Lógica Rol</i>	43
<i>Figura 26: Controlador Rol</i>	43
<i>Figura 27: Vista Rol</i>	44
<i>Figura 28: Crear Rol</i>	44
<i>Figura 29: Tabla Cliente</i>	46
<i>Figura 30: Tabla Proveedor</i>	46
<i>Figura 31: Procedimiento almacenado Cliente</i>	47
<i>Figura 32: Procedimiento almacenado Proveedor</i>	47
<i>Figura 33: Clase Cliente</i>	48
<i>Figura 34: Controlador Cliente</i>	48
<i>Figura 35: Clase Proveedor</i>	49

<i>Figura 36: Controlador Proveedor</i>	49
<i>Figura 37: Lógica Cliente</i>	50
<i>Figura 38: Lógica Proveedor</i>	50
<i>Figura 39: Vista Cliente</i>	51
<i>Figura 40: Vista Proveedor</i>	51
<i>Figura 41: Crear cliente</i>	52
<i>Figura 42: Crear proveedores</i>	52
<i>Figura 43: Clase Empresa</i>	53
<i>Figura 44: Lógica Empresa</i>	53
<i>Figura 45: Controlador Empresa</i>	53
<i>Figura 46: Vista Empresa</i>	54
<i>Figura 47: Crear Empresa</i>	54
<i>Figura 48: Catalogo de Productos</i>	56
<i>Figura 49: Tablas de gestión de inventario</i>	58
<i>Figura 50: Procedimiento almacenado Producto</i>	58
<i>Figura 51: Procedimiento almacenado Categoría</i>	59
<i>Figura 52: Procedimiento almacenado Unidad de Medida</i>	59
<i>Figura 53: Clase Categoría y Unidad de Medida</i>	60
<i>Figura 54: Clase Producto</i>	60
<i>Figura 55: Lógica Producto</i>	61
<i>Figura 56: Lógica Categoría</i>	62
<i>Figura 57: Lógica Unidad de Medida</i>	63
<i>Figura 58: Controlador Producto</i>	64
<i>Figura 59: Controlador Categoría</i>	64
<i>Figura 60: Controlador Unidad de Medida</i>	64
<i>Figura 61: Vista Producto</i>	65
<i>Figura 62: Vista Categoría</i>	65
<i>Figura 63: Vista Unidad de Medida</i>	66
<i>Figura 64: Crear Producto</i>	66
<i>Figura 65: Crear Unidad de Medida</i>	67
<i>Figura 66: Crear Categoría</i>	67
<i>Figura 67: Tablas de proceso de Ventas</i>	69
<i>Figura 68: Procedimiento almacenado de Venta</i>	69
<i>Figura 69: Clase Venta</i>	70
<i>Figura 70: Controlador Venta</i>	70
<i>Figura 71: Lógica Venta</i>	71
<i>Figura 72: Vista Detalle de Venta</i>	72
<i>Figura 73: Tablas Proceso de Compra</i>	72

<i>Figura 74: Procedimiento almacenado Compra</i>	73
<i>Figura 75: Clase Compra</i>	73
<i>Figura 76: Lógica Compra</i>	74
<i>Figura 77: Controlador Compra</i>	75
<i>Figura 78: Vista Detalle de Compra</i>	75
<i>Figura 79: Tabla de Merma</i>	77
<i>Figura 80: Tabla de Regalía</i>	77
<i>Figura 81: Clase Regalía</i>	77
<i>Figura 82: Procedimiento almacenado Merma</i>	78
<i>Figura 83: Procedimiento almacenado Regalía</i>	78
<i>Figura 84: Clase Merma</i>	79
<i>Figura 85: Lógica Regalía</i>	79
<i>Figura 86: Lógica Mermas</i>	80
<i>Figura 87: Controlador Regalía</i>	80
<i>Figura 88: Controlador Merma</i>	80
<i>Figura 89: Vista Regalía</i>	81
<i>Figura 90: Vista mermas</i>	81
<i>Figura 91: Crear mermas</i>	82
<i>Figura 92: Crear regalías</i>	82
<i>Figura 93: Diseño de Base de Datos Jehova Jireh</i>	84
<i>Figura 94: Carta de aceptación</i>	101
<i>Figura 95: Carta de agradecimiento</i>	102
<i>Figura 96: Acta de reunión exploración y recopilación de datos</i>	104
<i>Figura 97: Acta de reunión requerimiento inicial</i>	106
<i>Figura 98: Acta de reunión Sprint 1</i>	107
<i>Figura 99: Acta de reunión Sprint 2</i>	108
<i>Figura 100: Acta de reunión Sprint 3</i>	110
<i>Figura 101: Acta de reunión Sprint 4</i>	111
<i>Figura 102: Acta de reunión Sprint 5</i>	112
<i>Figura 103: Cuestionario</i>	114
<i>Figura 104: Requerimiento Oficial con Historias de Usuario</i>	129
<i>Figura 105: Requerimiento sprint 3 con Historias de Usuario</i>	143
<i>Figura 106: Reducción de intentos de Login</i>	154
<i>Figura 107: Validación de Sesión</i>	155
<i>Figura 108: Total de ventas Dashboard</i>	156
<i>Figura 109: Productos por vencerse Dashboard</i>	157
<i>Figura 110: Productos vencidos Dashboard</i>	158
<i>Figura 111: Stock actual del producto a mermar</i>	159

<i>Figura 112: Funcionalidad código de barras</i>	160
<i>Figura 113: Eliminación de filtros</i>	161
<i>Figura 114: Permitir venta por fracción</i>	162
<i>Figura 115: Filtro de fechas Venta</i>	163
<i>Figura 116: Funcionalidad de estados según el stock</i>	164
<i>Figura 117: Diagrama de tablas Registro de Actividades</i>	166
<i>Figura 118: Tabla Registro Actividades</i>	167
<i>Figura 119: Procedimiento almacenado Registro Actividades</i>	168
<i>Figura 120: Clase Registro Actividades</i>	169
<i>Figura 121: Controlador Registro Actividades</i>	169
<i>Figura 122: Lógica Registro Actividades</i>	170
<i>Figura 123: Vista Registro Actividades</i>	171
<i>Figura 124: Diagrama de tablas Registro Sesión</i>	173
<i>Figura 125: Tabla Registro Sesión</i>	173
<i>Figura 126: Procedimiento almacenado Registro Sesión</i>	174
<i>Figura 127: Clase Registro Sesión</i>	175
<i>Figura 128: Controlador Registro Sesión</i>	175
<i>Figura 129: Lógica Registro Sesión</i>	176
<i>Figura 130: Vista Registro Sesión</i>	177
<i>Figura 131: Validación de usuario desconocido</i>	178
<i>Figura 132: Diagrama de tablas Registro de Errores</i>	180
<i>Figura 133: Tabla Registro de Errores</i>	181
<i>Figura 134: Clase Registro de Errores</i>	182
<i>Figura 135: Lógica Registro de Errores</i>	183
<i>Figura 136: Controlador Registro de Errores</i>	184
<i>Figura 137: Vista Registro de Errores</i>	185
<i>Figura 138: Resumen del Sprint Review Sprint 3</i>	186
<i>Figura 139: Sprint Backlog del Sprint 4</i>	190
<i>Figura 140: Reporte general</i>	192
<i>Figura 141: Reporte de compras por número de orden</i>	193
<i>Figura 142: Reporte de compras por rango de fechas</i>	194
<i>Figura 143: Reporte de ventas por número de orden</i>	195
<i>Figura 144: Reporte de ventas por rango de fechas</i>	196
<i>Figura 145: Total de ventas realizadas</i>	197
<i>Figura 146: Total de compras realizadas</i>	198
<i>Figura 147: Reporte de productos</i>	199
<i>Figura 148: Resumen del Sprint Review Sprint 4</i>	200
<i>Figura 149: Sprint Backlog del Sprint 5</i>	203

Índice de tablas

Tabla 1: Tabla de prioridad _____	20
Tabla 2: Puntos de historia _____	20
Tabla 3: Product Backlog _____	21
Tabla 4: Requerimientos No funcionales _____	26
Tabla 5: Scrum Team _____	27
Tabla 6: Sprint Backlog de configuración inicial _____	32
Tabla 7: Sprint Backlog seguidad y acceso _____	34
Tabla 8: Sprint Backlog de módulos funcionales _____	46
Tabla 9: Resumen del Sprint Review Sprint 1 _____	55
Tabla 10: Sprint Backlog para la gestión de inventario _____	57
Tabla 11: Sprint Backlog para la gestión de Ventas y Compras _____	68
Tabla 12: Sprint Backlog para el control de Mermas y Regalías _____	76
Tabla 13: Resumen del Sprint Review Sprint 2 _____	83
Tabla 14: Matriz de pruebas Login _____	85
Tabla 15: Matriz de pruebas Usuario y Rol _____	86
Tabla 16: Matriz de pruebas Cliente _____	87
Tabla 17: Matriz de pruebas Proveedor _____	88
Tabla 18: Matriz de pruebas Empresa _____	89
Tabla 19: Matriz de pruebas Categoría _____	90
Tabla 20: Matriz de pruebas Producto, Venta y Compra _____	91
Tabla 21: Matriz de pruebas Merma _____	92
Tabla 22: Estudios aplicados _____	119
Tabla 23: Procesos operativos _____	120
Tabla 24: Hardware antiguo _____	121
Tabla 25: Software antiguo _____	121
Tabla 26: Conexión antigua _____	121
Tabla 27: Hardware Necesario _____	122
Tabla 28: Software necesario _____	122
Tabla 29: Conexión necesaria _____	122
Tabla 30: Talento humano _____	123
Tabla 31: Costos de hardware [23] _____	124
Tabla 32: Costos de software [24] _____	124
Tabla 33: Costos de conexiones de redes [22] _____	124

<i>Tabla 34: Costos de talento humano del proyecto</i>	125
<i>Tabla 35: Costos de talento humano en el mercado</i>	126
<i>Tabla 36: Product Backlog del Sprint 3</i>	151
<i>Tabla 37: Sprint Backlog mejora de funcionalidades</i>	153
<i>Tabla 38: Sprint Backlog Registro Actividades</i>	165
<i>Tabla 39: Sprint Backlog Registro Sesión</i>	172
<i>Tabla 39: Sprint Backlog Registro de Errores</i>	179
<i>Tabla 38: Reportes del sistema</i>	189

I. Introducción

El presente trabajo monográfico se centra en la pulperia Jehová Jireh, un negocio que, a pesar de su relevancia comunitaria, enfrenta ineficiencias significativas en su gestión de inventario y facturación debido a la dependencia de procesos manuales. Esta falta de control ha ocasionado problemas de imprecisión en los niveles de existencia, errores de cálculo y una incapacidad para realizar un seguimiento efectivo de las transacciones. Para resolver estas deficiencias, se desarrolló un sistema web para el control de inventario y facturación.

La concepción de esta solución surgió a partir de una evaluación de las necesidades de la propietaria, la Sra. Mariela Núñez, quien confirmó que las opciones de sistemas comerciales eran inviables. Esto se debió principalmente a su alto costo y a la falta de adaptabilidad a los requerimientos específicos de la pulperia, haciendo necesaria una alternativa a la medida.

Por lo tanto, se optó por un enfoque de desarrollo ágil, utilizando la metodología Scrum. Las tecnologías como .NET, C# y SQL Server, así como tecnologías web para la interfaz de usuario fueron seleccionadas para construir un sistema a la medida, capaz de adaptarse precisamente a los requerimientos del negocio y ofrecer una solución robusta y escalable.

El presente documento está estructurado para detallar cada fase del proyecto, desde su concepción hasta su implementación. Se comenzará con los antecedentes y la justificación, explicando el porqué del proyecto, seguido del marco teórico que explica las terminologías del negocio, la metodología y las tecnologías utilizadas. La sección de análisis y resultados describe el proceso de desarrollo a través del uso de la metodología Scrum, y finalmente, se presentan las conclusiones y recomendaciones derivadas del trabajo realizado.

II. Antecedentes

La gestión de inventario y los procesos de facturación digital representan un pilar fundamental en la sostenibilidad y crecimiento de cualquier negocio minorista. Históricamente, la optimización de los procesos operativos en el sector de las micro y pequeñas empresas se ha enfrentado a una disyuntiva tecnológica que define el marco de este proyecto.

A nivel de mercado, la oferta de software de gestión se centra en sistemas de planificación de recursos empresariales (ERP) y soluciones de punto de venta (POS) con altos niveles de funcionalidad. Estos sistemas, aunque robustos, han demostrado ser inaccesibles para el segmento de pequeños comercios como las pulperías. Tal como se identificó en el contexto del proyecto, la propietaria de la pulpería Jehová Jireh, Dña. Mariela Núñez, confirmó que estos sistemas enlatados eran inaccesibles por el precio y lo poco adaptables que son para las necesidades específicas de su negocio.

A su vez, en investigaciones realizadas a nivel nacional, se evaluó el estudio realizado por García Poveda y Porras Herrera, acerca del desarrollo de un sistema web para la gestión de los procesos de facturación y control de inventario para la tienda “Winnie’s Menorquinas”. Los procesos en los que se adentraron fueron sistematizados bajo las reglas del negocio como lo son el registro de clientes, gestión de proveedores, control de entrada y salida del inventario y el registro de facturas como resultado del proceso de compraventa [1].

Finalmente, como investigación realizada a nivel internacional, se evaluó estudio realizado por Chamorro López, y Ibujés Revelo, el cual fue implementado en la distribuidora “Colombina” ubicada en la ciudad de Tulcán, Ecuador. Este estudio detalla las problemáticas identificadas en el control de inventarios debido al registro manual de las entradas y salidas de mercadería [2].

El alcance de estos 2 antecedentes no contempla funcionalidades específicas, ni módulos funcionales para la gestión de mermas y el control de regalías.

III. Justificación

Jehová Jireh, representa un punto de abastecimiento fundamental para el sector aledaño. A lo largo de sus 7 años de servicio, la pulperia ha carecido de un sistema de información que agilice sus operaciones y procedimientos en términos de control de inventario y ventas.

En primera instancia, Jehová Jireh enfrenta diversos desafíos en cuanto a la gestión de sus inventarios, tales como el control impreciso de los niveles de existencia, el exceso o la escasez de productos, la demora en la emisión de facturas, errores de cálculos de precios, registro ineficiente de las mermas y regalías, además de la dificultad de realizar seguimientos efectivos en las transacciones. Todo esto ocurre debido a que los procesos operativos se hacen de forma manual.

Según lo planteando en los antecedentes, la propietaria contempló la posibilidad de comprar un sistema enlatado, pero no se materializó ya que el costo era inaccesible y su difícil adaptabilidad para cubrir los procesos antes mencionados.

Para dar solución a los problemas antes planteados, se implementó un sistema web de inventario y facturación que administre los datos de proveedores, clientes, productos, inventario y reportes funcionales. A través de la implementación de este sistema, se espera que Jehová Jireh logre mejorar su control de inventarios, agilizar el proceso de facturación y optimizar su gestión global, generando beneficios tangibles para el crecimiento y el éxito del negocio.

IV. Objetivos

4.1. Objetivo General

Desarrollar el Sistema web para el control de inventario y facturación de la pulpería Jehová Jireh.

4.2. Objetivos Específicos

- Definir los requerimientos funcionales y no funcionales del sistema considerando los procesos operativos de la pulpería.
- Diseñar la base de datos estableciendo las tablas, relaciones y atributos mediante el uso de la herramienta SQL Server.
- Codificar los distintos módulos funcionales del sistema web empleando la arquitectura ASP.NET Modelo Vista Controlador (MVC) y el lenguaje de programación C#.
- Evaluar los resultados del funcionamiento del sistema mediante la realización de pruebas de aceptación del usuario (UAT).

V. Marco teórico

En esta sección, se abordan las definiciones clave para respaldar el desarrollo de la aplicación web para el control del inventario y la facturación. Comenzaremos explorando la terminología específica aplicada al negocio de la pulpería Jehová Jireh. Se explicará la metodología seleccionada para el diseño e implementación del proyecto, y finalmente, se proporcionará una visión general de las herramientas de desarrollo que se utilizarán en este proceso.

5.1. Terminología aplicada al negocio

Los términos o definiciones abordados en esta sección corresponden a los utilizados por la pulpería Jehová Jireh. Cabe mencionar que, los conceptos recopilados fueron proporcionados en una reunión con la propietaria (Véase Figura 22 del Anexo II).

5.1.1. Clientes

Un cliente es quien adquiere los productos o servicios de la pulpería. Por lo tanto, es un consumidor, desde la perspectiva de Jehová Jireh puede tratarse de un cliente recurrente cuando este realiza compras regulares, o un cliente de única vez.

5.1.2. Proveedores

Un proveedor es aquel representante de una empresa con el objetivo de abastecer el inventario de Jehová Jireh, por lo general, al proveedor se le compra diversos productos con un margen de ganancias significativo para la pulpería.

5.1.3. Empresa

Una empresa es la que surte los productos para Jehová Jireh, una empresa puede tener varios productos asociados lo que genera en el cliente confianza y sugestiona la compra del producto asociado.

5.1.4. Productos

Los productos son aquellos elementos que conforman la oferta de la pulpería Jehová Jireh, mismos que van desde la variedad de alimentos y artículos de consumo diario, hasta servicios adicionales detallados en las secciones posteriores del marco teórico.

5.1.5. Categorías

Son grupos o clases en los que Jehová Jireh clasifica sus productos. Las categorías se utilizan para organizar los productos de una manera lógica, lo cual hace más fácil la tarea de identificarlos cuando el mismo está asociado a una categoría.

5.1.6. Inventario

El inventario es el conjunto de bienes y mercancías que pertenecen a una empresa. Su principal objetivo es llevar un control preciso de estas existencias para fines contables, logísticos o de gestión. Jehová Jireh tiende al modelo de inventario de productos periódico mensualmente, asegurando de que siempre haya suficiente stock disponible para satisfacer la demanda de los clientes [3].

5.1.7. Facturas

Una factura es un documento mercantil que acredita la entrega de un bien junto con el detalle de las condiciones de la operación. Es emitida por el vendedor y entregada al cliente. En algunos casos, las facturas permiten deducir el impuesto de valor agregado (IVA) además de ser indispensable para la declaración fiscal y normas tributarias. Las facturas suelen incluir la fecha de la transacción, detalle de productos y el precio total de la venta. Jehová Jireh utiliza recibos de venta a los que llama facturas, debido a que es pulpería de cuota fija, estos registran una transacción de venta entre la pulpería y el cliente [4].

5.1.8. Mermas

Son las pérdidas o desperdicios que se producen durante el almacenamiento de un producto. Según Jehová Jireh las mermas pueden ser causadas por una variedad de factores, como la obsolescencia, la caducidad o el daño físico.

5.1.9. Regalías

Jehová Jireh entiende por regalías a los regalos por compra que ofrecen los proveedores por la adquisición de ciertas ofertas o promociones de productos, estos regalos por lo general son productos que entran al inventario y con los cuales obtienen ganancias extras.

5.2. Metodología Scrum

Scrum es un marco de trabajo ágil y liviano que ayuda a personas, equipos y organizaciones a generar valor de manera eficiente y efectiva mediante la implementación de soluciones adaptativas para abordar problemas complejos. Scrum se basa en la inteligencia colectiva de las personas que lo utilizan, en lugar de proporcionar a las personas instrucciones detalladas, las reglas de scrum guían sus relaciones e interacciones. Esta metodología envuelve las prácticas existentes o las hace innecesarias, haciendo visible la eficacia relativa de las técnicas actuales de gestión, entorno y trabajo, de modo que se puedan realizar mejoras [5].

Scrum se basa en la teoría del empirismo. El empirismo afirma que el conocimiento proviene de la experiencia y de la toma de decisiones con base en la observado. Según sus principios de transparencia, inspección y adaptación scrum proporciona un enfoque que apoya el concepto de trabajar de forma iterativa. En última instancia, se trata de adoptar una mentalidad experimental; aprendiendo y adaptándose a través de cada ciclo de trabajo [6].

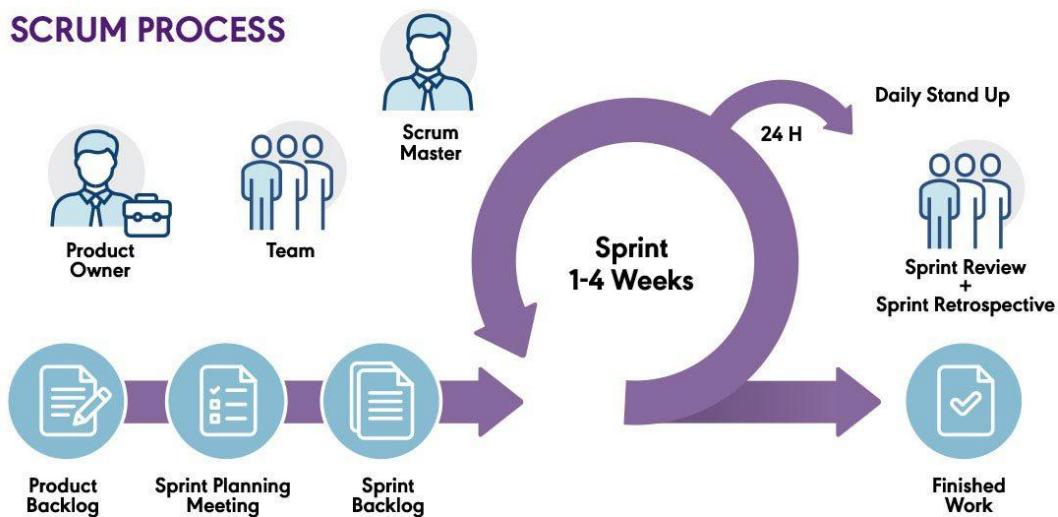
Entre los valores del scrum, tenemos el compromiso, foco, franqueza, respeto y coraje, la guía de scrum indica que el uso exitoso de scrum dependerá de que las personas se vuelvan más competentes en vivir con estos cinco valores. Estos valores dan dirección al equipo con respecto a su trabajo, acciones y comportamiento. Las decisiones que se tomen, los pasos que se den y la forma en que se use scrum debe reforzar estos valores, no disminuirlos ni socavarlos. Los miembros del scrum team aprenden y exploran los valores mientras trabajan con las ceremonias y artefactos scrum [5].

El objetivo principal de la metodología scrum es entregar valor al cliente de manera temprana y continua a lo largo del ciclo de vida del proyecto. Con este fin, se ordenan aquellas características e incrementos de producto que proporcionan el mayor valor para el cliente para que se desarrollen y entreguen primero [6].

Para facilitar este proceso, el marco de trabajo de scrum se basa en el scrum team, ceremonias y artefactos. Por lo general, se utiliza una representación visual del flujo de trabajo para mantener a todos los miembros del equipo informados sobre el trabajo en curso y lo que está por venir. Estos elementos permiten colectivamente la ejecución de proyectos con agilidad y efectividad [6].

En esencia, las herramientas y técnicas de scrum permiten que las tareas grandes se dividan en partes manejables que se pueden completar en ciclos cortos de un mes o menos [6].

Figura 1: Procesos del scrum [6]

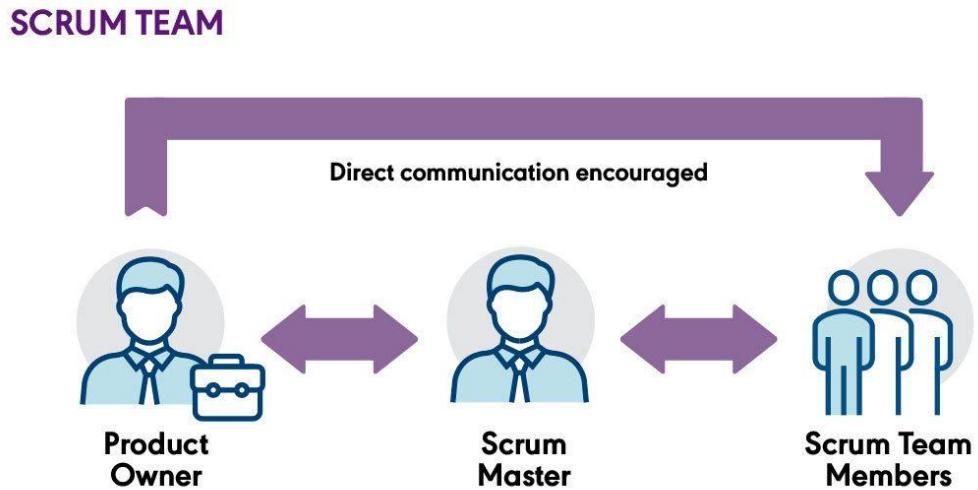


5.2.1. Scrum Team

La unidad fundamental de scrum es un pequeño equipo de personas el Scrum Team, el cual consta de un scrum Master, un product owner y developers. Dentro de un Scrum Team, no hay jerarquías. Es una unidad cohesionada de profesionales enfocados en un objetivo a la vez, el objetivo del producto [5].

De acuerdo con la Guía de Scrum, los equipos de scrum suelen estar formados por 10 personas o menos, son multifuncionales y autoorganizados. Esto significa que los miembros tienen colectivamente todas las habilidades necesarias para hacer el trabajo y deciden entre ellos quién hace qué y cuándo. A su vez, el scrum team es responsable de todas las actividades relacionadas con el producto, desde la colaboración de los interesados, la verificación, el mantenimiento, la operación, la experimentación, la investigación y el desarrollo, y cualquier otra cosa que pueda ser necesaria [5].

Figura 2: Comunicación del scrum team [6]



5.2.1.1. Product Owner

El product owner es responsable de maximizar el valor del producto resultante del trabajo del scrum team. La forma en que esto se hace puede variar ampliamente entre organizaciones, scrum team e individuos [5].

Son responsables de recopilar y priorizar los requisitos, gestionar el backlog del producto y garantizar la alineación entre los objetivos empresariales y los esfuerzos de desarrollo. En última instancia, el papel del producto owner consiste en maximizar el valor empresarial del trabajo del scrum team. Con este fin, colaboran estrechamente con el scrum master para eliminar obstáculos y brindar una orientación clara al equipo. En pocas palabras, el trabajo del product owner consiste en priorizar la plusvalía del proyecto para que los elementos de alto valor se desarrollen primero y decidan cuándo el trabajo está listo para su lanzamiento [6].

5.2.1.2. Scrum Master

El scrum master es responsable de establecer el scrum, lo hace ayudando a todos a comprender la teoría y la práctica de scrum, tanto dentro del scrum team como de la organización [5].

Son expertos en el marco y se aseguran de que todos entiendan y se adhieran a los valores, prácticas y ceremonias de scrum. Ayudan a crear un ambiente de trabajo óptimo, entrenando a los equipos, product owners y a la organización en los procesos scrum, promoviendo la transparencia y animando al equipo a reflexionar y mejorar su práctica [6].

Las tareas diarias de un scrum master incluyen facilitar eventos de scrum y ayudar al equipo a superar los desafíos, resolver conflictos y mantener el enfoque en la entrega de valor. Como líder de servicio, el scrum master habilita el aspecto de autoorganización del equipo y lo capacita para tomar decisiones y alcanzar sus objetivos [6].

5.2.1.3. Developers

El equipo de desarrollo es un grupo multifuncional autoorganizado responsable de traducir el backlog del producto owner en resultados tangibles. Esto significa que crean un plan para el sprint y deciden "cómo" llevarán a cabo su trabajo según los estándares acordados. Se reúnen diariamente para mantener la transparencia e inspeccionar y adaptar el trabajo que están haciendo [6].

El equipo de desarrollo se relaciona estrechamente con el product owner para garantizar que sus entregables cumplan con las expectativas, a su vez definen los criterios de "hecho" y ofrecen resultados que se adhieran a estos criterios. Las habilidades específicas que necesitan los developers suelen ser amplias y variarán según el ámbito de trabajo, puede ser una combinación de roles técnicos, incluidos diseñadores, ingenieros de software, analistas de calidad y especialistas en UX [6].

5.2.2. Ceremonias de Scrum

Las ceremonias son los eventos o reuniones que garantizan que todos estén en la misma página con respecto a los objetivos, el progreso y las mejoras [6].

Estos eventos están diseñados específicamente para permitir la transparencia necesaria. Si no se realizan los eventos según lo prescrito, se pierden oportunidades para inspeccionar y adaptarse. Los eventos se utilizan en scrum para crear regularidad y minimizar la necesidad de reuniones no definidas en scrum [5].

5.2.2.1. Sprint

Los sprints son el núcleo de scrum, donde las ideas se transforman en un valor tangible. Estos permiten la previsibilidad al garantizar la inspección y adaptación del progreso hacia un objetivo del producto al menos cada mes calendario. Cuando el horizonte de un sprint es demasiado largo, el objetivo del sprint puede volverse inválido, la complejidad puede crecer y el riesgo puede aumentar, para ello se pueden emplear sprints más cortos para generar más ciclos de aprendizaje y limitar el riesgo de costo y esfuerzo a un período de tiempo menor [5].

Durante el transcurso del sprint se aplican las siguientes reglas, estas se mencionan en la guía de scrum [5].

- No se introducen cambios que puedan poner en peligro el logro del objetivo del sprint.
- La calidad no disminuye.
- El product backlog se ajusta según sea necesario.
- El alcance del trabajo puede aclararse y renegociarse con el product owner a medida que se obtiene más información.

También es importante mencionar que un sprint puede cancelarse si el objetivo ya no es relevante. No obstante, solo el product owner tiene la autoridad para tomar esta decisión [5].

5.2.2.2. Sprint Planning

También conocidas como reunión de planificación del sprint, da inicio al sprint al definir las tareas que se llevarán a cabo durante todo ese período. El scrum team trabaja en conjunto para elaborar este plan [5].

El product owner se asegura de que los asistentes estén preparados para discutir los elementos más importantes del product backlog y cómo se relacionan con el objetivo del producto. En algunos casos, el equipo de scrum puede invitar a personas adicionales para proporcionar orientación durante el sprint planning [6].

Durante esta reunión se sigue un orden de temas a abordar según lo reflejado en la guía de scrum [5]:

1. Discusión de la importancia del trabajo planificado para este sprint.
2. Se determinan qué tareas específicas se pueden abordar durante este sprint.
3. Se define cómo se llevará a cabo el trabajo seleccionado.

El sprint planning tiene un límite de tiempo de máximo ocho horas para un sprint de un mes. Para sprints más cortos, el evento suele ser de menor duración [5].

5.2.2.3. Daily Scrum

La daily scrum es un evento de 15 minutos para los developers del scrum team. Para reducir la complejidad, se lleva a cabo a la misma hora y en el mismo lugar todos los días hábiles del sprint. En caso de que el product owner o el scrum master estén activamente involucrados en las tareas del sprint backlog, también participan como miembros del equipo de desarrollo. El propósito de la daily scrum es inspeccionar el progreso hacia el objetivo del sprint y adaptar el sprint backlog según sea necesario, ajustando el trabajo planificado entrante [5].

Los developers pueden seleccionar la estructura y las técnicas que deseen, siempre que su daily scrum se centre en el progreso hacia el objetivo del sprint y produzca un plan viable para el siguiente día de trabajo. Esto crea enfoque y mejora la autogestión. Esta reunión diaria promueve la comunicación, ayuda a identificar obstáculos, facilita la toma de decisiones rápidas y, como resultado, reduce la necesidad de llevar a cabo reuniones adicionales [5].

Es importante destacar que el daily scrum no es el único momento en el que los developers pueden ajustar su plan. Con frecuencia, durante el día, se reúnen para discutir con más detalle cómo adaptar o reorganizar el trabajo restante del sprint [5].

5.2.2.4. Sprint Review

El sprint review es el penúltimo evento del sprint y tiene un límite de tiempo de máximo cuatro horas para un sprint de un mes. El propósito del sprint review es inspeccionar el resultado del sprint y determinar futuras adaptaciones. El scrum team presenta los resultados de su trabajo a los interesados clave y se discute el progreso hacia el objetivo del producto [5].

Durante el evento, el scrum team y los interesados revisan lo que se logró en el sprint y lo que ha cambiado en su entorno. Con base en esta información, los asistentes colaboran para determinar los pasos a seguir. Asimismo, se pueden realizar ajustes en el Product Backlog para aprovechar nuevas oportunidades [5].

5.2.2.5. Sprint Retrospective

Tiene por finalidad diseñar estrategias para incrementar la calidad y la eficiencia del trabajo. Durante este evento, el equipo de scrum evalúa cómo se desarrolló el último sprint en cuanto a personas, interacciones, procesos, herramientas y su propia definición de “terminado”. Los aspectos puntuales examinados pueden variar según el ámbito de trabajo. Se buscan los puestos que llevaron a desviaciones en el camino y se investigan sus orígenes. Además, se analiza lo que funcionó correctamente durante el sprint, se identifican los problemas encontrados y se reflexiona sobre cómo se resolvieron o no dichos problemas [5].

El equipo scrum detecta las modificaciones más beneficiosas para mejorar su eficacia. Las mejoras más significativas se implementan lo antes posible, e incluso pueden agregarse al sprint backlog para el próximo sprint. El sprint retrospective marca la conclusión del sprint y tiene una duración limitada de un máximo de 3 horas para un sprint de un mes [5].

5.2.3. Artefactos Scrum

Los artefactos de scrum representan trabajo o valor. Están diseñados para maximizar la transparencia de la información clave. Por lo tanto, todas las personas que los inspeccionan tienen la misma base de adaptación [5].

5.2.3.1. Product Backlog

Representa una lista dinámica y organizada para mejorar el producto y es la única fuente de trabajo para el equipo scrum. Los elementos del product backlog que el equipo scrum considera como completados dentro de un sprint, se vuelven aptos para su elección en un evento de sprint planning. Esto, por lo general, ocurre después de las actividades de refinamiento, que implican dividir y definir con mayor precisión los elementos del product backlog en partes más pequeñas y detalladas. El refinamiento del product backlog es un proceso continuo que agrega detalles, como descripciones, prioridades y tamaños, y estos detalles pueden variar según el contexto del trabajo [5].

Los developers encargados de ejecutar el trabajo son responsables de este proceso de dimensionamiento. El Product Owner puede influir en los desarrolladores al ayudarlos a comprender y seleccionar las mejores alternativas disponibles [5].

5.2.3.2. Sprint Backlog

El sprint backlog, es un plan desarrollado por y para los desarrolladores. Actúa como una representación en tiempo real y altamente visible del trabajo planificado por los desarrolladores para alcanzar el objetivo del sprint. Por lo tanto, es una herramienta dinámica que se actualiza a lo largo del sprint a medida que se obtiene más información y se adquiere más conocimiento. Debe contener suficientes detalles para permitir que los desarrolladores inspecciones su progreso durante la daily scrum [5].

El Sprint Backlog está formado por tres componentes fundamentales, a como se define en la guía de scrum [5]:

- El objetivo del Sprint, que establece la razón del sprint.
- El conjunto de elementos del Product Backlog seleccionados para el sprint.
- Estrategia o plan que describe cómo se entregará el incremento.

5.2.3.3. Increment

Un incremento representa un avance tangible hacia el logro del objetivo del producto. Cada incremento se suma a todos los increments anteriores y se verifica minuciosamente, lo que garantiza que todos funcionen juntos [5].

Dentro de un sprint, es posible crear varios incrementos. La acumulación de estos incrementos se presenta durante el sprint review, lo que respalda el enfoque del empirismo. Asimismo, es factible entregar un incremento a los interesados antes de finalizar un sprint. No se debe considerar el sprint review como la única oportunidad para liberar valor [5].

El trabajo no puede considerarse parte de un incremento a menos que cumpla con la definición de terminado [5].

5.3. Tecnologías y herramientas

En el siguiente apartado se definen las tecnologías a utilizar para el desarrollo de la aplicación web y las herramientas necesarias para la elaboración del proyecto.

5.3.1. Visual Studio

Visual Studio es una plataforma de lanzamiento creativa que puede utilizar para editar, depurar y compilar código de una aplicación. Además, Visual Studio incluye herramientas de completado de código, diseñadores gráficos y muchas más funciones para mejorar el proceso de desarrollo de software. El uso de Visual Studio fue muy importante por sus características y funcionalidades que facilitan la creación de un sistema robusto y eficiente. [7].

5.3.2. .NET

.NET es una plataforma de desarrollo de aplicaciones ampliamente utilizada y respaldada por Microsoft. Proporciona un conjunto de herramientas y marcos de trabajo que facilitan la creación de aplicaciones escalables y seguras, además es compatible con múltiples lenguajes y de una arquitectura sólida y escalable. [8].

5.3.3. ASP.NET

ASP.NET es un framework open source multiplataforma creado por Microsoft. Es un entorno de trabajo basado en código abierto que está pensado para el desarrollo y la ejecución de aplicaciones y servicios web modernos. Al ser multiplataforma es posible ejecutar aplicaciones ASP.NET tanto en Windows, Linux y macOS como en contenedores [9].

5.3.4. C#

C# es un lenguaje de programación desarrollado por Microsoft, orientado a objetos, que ha sido diseñado para compilar diversas aplicaciones que se ejecutan en .NET Framework. Se trata de un lenguaje simple, eficaz y con seguridad de tipos. Las numerosas innovaciones de C# permiten desarrollar aplicaciones rápidamente y mantener la expresividad y elegancia de los lenguajes de estilo de C [10].

5.3.5. SQL Server

SQL Server es un sistema de administración de bases de datos relacionales desarrollado por Microsoft. Se usa mucho en el entorno empresarial por su confiabilidad, escalabilidad y administración de datos. En este proyecto, se utilizó SQL Server para almacenar y administrar los datos relacionados con el inventario, las ventas y otros aspectos relevantes del negocio [11].

5.3.6. Modelo Vista Controlador (MVC)

El patrón de arquitectura MVC convierte el desarrollo de aplicaciones complejas en un proceso mucho más manejable. Permite a varios desarrolladores trabajar simultáneamente en la aplicación. En el proyecto, el uso de MVC fue esencial para dividir el sistema en componentes distintos y promover una mejor organización del código. En este caso, el modelo fue responsable de interactuar con la base de datos SQL Server, realizar operaciones de lectura y escritura, y mantener la integridad de los datos relacionados con el inventario, la vista sería la interfaz de usuario que Jehová Jireh visualiza. Por otra parte, el controlador es el cerebro de la aplicación que controla como se muestran los datos [12].

5.3.7. Gitlab

GitLab es una plataforma de desarrollo colaborativo basada en web que proporciona herramientas para la gestión del ciclo de vida del desarrollo de software, utilizando Git como sistema de control de versiones. básicamente se utilizó durante el proyecto para el versionado, despliegue y también como repositorio online [13].

5.3.8. HTML

HTML son las siglas en inglés de HyperText Markup Language, que significa Lenguaje de Marcado de Hipertexto. Se llama así al lenguaje de programación empleado en la elaboración de páginas Web. HTML se utilizó para el sistema web ya que es un estándar de referencia para la codificación y estructuración de la web [14].

5.3.9. CSS

CSS son las siglas en inglés para «hojas de estilo en cascada» (Cascading Style Sheets). Básicamente, es un lenguaje que maneja el diseño y presentación de las páginas web. CSS caso sirvió para estilizar los diferentes módulos desarrollados del sistema como el catálogo de productos, inventario, clientes [15].

5.3.10. JavaScript

JavaScript es un lenguaje de programación de alto nivel, interpretado y orientado a objetos, utilizado principalmente en el desarrollo web. Japara agregar potencial de interacción y dinamismo al sistema web de inventario y facturación como la creación de animaciones de pagos, validación de los distintos formularios como el de clientes, proveedores y productos [16].

5.3.11. Bootstrap

Bootstrap es una biblioteca de herramientas de código abierto optimizadas para el diseño de sitios y aplicaciones web. Esta plataforma se basa en lenguaje HTML y CSS, e incluye una amplia gama de elementos de diseño, como formularios, botones y menús que se adaptan a diferentes formatos de navegación [17].

5.3.12. Power BI Desktop

Power BI Desktop es una aplicación gratuita que se instala en el equipo local que le permite conectarse, transformar y visualizar los datos. Con Power BI Desktop, se puede conectar a varios orígenes de datos diferentes y combinarlos. Este modelo le permite crear objetos visuales y colecciones de objetos visuales que puede compartir como informes, con otras personas dentro de su organización [18]

5.3.13. IIS web service

Son las iniciales de Internet Information Services y si bien es más conocido como servidor web en realidad son un conjunto de servicios que transforman un sistema Microsoft Windows en un servidor capaz de ofrecer servicios Web, FTP y SMTP entre otros. Es apto tanto como servidor web en Internet y en Intranet, siendo en Intranet o en servidores privados donde tiene mayor cuota debido al mercado empresarial que lo utilizan para aplicaciones internas en empresas de mediana y gran escala ya que va de la mano de Visual Studio .NET [19].

5.3.14. SmarterASP.NET

SmarterASP.NET es una plataforma de alojamiento diseñada específicamente para aplicaciones que utilizan el framework ASP.NET. Este servicio se ha posicionado como una de las mejores opciones en el mercado gracias a su enfoque en la facilidad de uso, rendimiento y soporte técnico. A diferencia de muchas otras plataformas de alojamiento, SmarterASP.NET está completamente optimizada para aplicaciones ASP.NET. Esto significa que el servidor está configurado específicamente para ofrecer un rendimiento óptimo, lo cual es crucial para aplicaciones web dinámicas y complejas. [20].

5.3.15. ADO.NET

ADO.NET es un conjunto de clases que exponen servicios de acceso a datos para programadores de .NET Framework. ADO.NET proporciona un amplio conjunto de componentes para crear aplicaciones distribuidas y de uso compartido de datos. Es una parte integral de .NET Framework, que proporciona acceso a datos relacionales, XML y de aplicación, además, admite una variedad de funcionalidades de desarrollo, incluida la creación de clientes de base de datos front-end y objetos empresariales de nivel intermedio usados por aplicaciones, herramientas, lenguajes o exploradores de Internet [21].

VI. Análisis y presentación de resultados.

Metodología y desarrollo del tema

El siguiente acápite detalla la implementación del Sistema Web de Inventario y Facturación, cuyo proceso inicio con el levantamiento de requerimientos o historias de usuario las cuales, generan el Product Backlog. Posteriormente, se estableció el Marco de Trabajo Scrum, definiendo los roles, artefactos y una estructura de cinco Sprints de desarrollo incremental.

6.1. Creación del Product Backlog

6.1.1. Elaboración y priorización del Product Backlog

El levantamiento se realizó mediante reuniones clave con la propietaria. Se siguió el formato estándar de SCRUM que consiste en plantear un enunciado con la estructura "Como [Rol], necesito [Funcionalidad] con el fin de [Razón o Resultado]". Estas historias se priorizarán según su prioridad, impacto en el sistema y nivel de dificultad.

Este enfoque colaborativo permitió documentar los Requerimientos Funcionales No Funcionales que debía cumplir el sistema, alcanzando un acuerdo formal sobre la hoja de ruta del desarrollo.

Tabla 1: Tabla de prioridad

Prioridad	Color	Descripción
Alta	Rojo	Clasifica la tarea de forma muy necesaria
Media	Amarillo	Clasifica la tarea como grado medio
Baja	Celeste	Clasifica la tarea relativamente de baja prioridad

Tabla 2: Puntos de historia

Puntos	Descripción
1	Nivel de dificultad nula
2	Nivel de dificultad básica
3	Nivel de dificultad intermedio
4	Nivel de dificultad avanzado
5	Nivel de dificultad extremo

Se procedió al levantamiento detallado y colaborativo de las historias de usuario, lo que permitió profundizar en las expectativas de la propietaria sobre el sistema dando origen al Product Backlog del proyecto.

Tabla 3: Product Backlog

N.	Rol	Funcionalidad	Resultado	Prioridad	Puntos
1	Usuario	Iniciar sesión	Acceso al sistema	Alta	3
2	Usuario	Recuperar contraseña	Acceso al sistema	Media	4
3	Administrador	Asignar roles	Definir permisos	Alta	4
4	Vendedor	generar ventas y órdenes de compra	Transaccionar eficientemente	Alta	4
5	Administrador	Crear un usuario	nuevo empleado pueda acceder al sistema.	Alta	2
6	Administrador	Editar la información de un usuario	Mantener la información de los empleados actualizada.	Media	2
7	Administrador	Eliminar un usuario	Bloquear el acceso al sistema a empleados inactivos o retirados.	Alta	3
8	Administrador	Crear un nuevo cliente	Tener listas de clientes recurrentes.	Alta	2
9	Administrador	Editar un cliente	Mantener actualizados los datos de contacto.	Media	1
11	Administrador	Crear una nueva Empresa	Asociar productos a su marca o compañía de origen.	Media	2
12	Administrador	Editar una Empresa	Mantener los datos de la empresa correctos.	Baja	1
13	Administrador	Eliminar una Empresa	Retirar del catálogo las marcas que no surten productos a la pulperia.	Baja	1

N.	Rol	Funcionalidad	Resultado	Prioridad	Puntos
14	Administrador	Crear un nuevo proveedor	Contar con una base de datos de contactos para futuras órdenes de compra.	Alta	2
15	Administrador	Editar la información de un proveedor	Mantener los datos de contacto del representante actualizados para el abastecimiento.	Media	2
16	Administrador	Inactivar un proveedor	Gestionar y depurar la lista de contactos de proveedores inactivos.	Media	1
17	Administrador	Crear un nuevo producto	Registrar la oferta de la pulpería y obtener un control exacto de las existencias.	Alta	4
18	Administrador	Editar la información de un producto	Mantener la base de datos de inventario y precios de venta siempre actualizada.	Alta	3
19	Administrador	Inactivar un producto	Retirar del catálogo aquellos artículos que ya no se venden.	Alta	2
20	Administrador	Crear una nueva categoría	Mantener el inventario organizado lógicamente y facilitar la búsqueda de productos.	Media	2
21	Administrador	Editar una categoría	Corregir o actualizar el nombre de las categorías existentes.	Baja	1

N.	Rol	Funcionalidad	Resultado	Prioridad	Puntos
22	Administrador	Inactivar una categoría	Limpiar el catálogo de categorías que no se están utilizando.	Baja	1
23	Administrador	Crear una nueva unidad de medida	Registrar la magnitud de los productos de forma precisa.	Media	2
24	Administrador	Editar una unidad de medida	Corregir o actualizar la información de las unidades de medida.	Baja	1
25	Administrador	Inactivar una unidad de medida	Eliminar del catálogo aquellas que ya no se utilizan en nuevos productos.	Baja	1
26	Vendedor	Crear y gestionar una nueva venta	Procesar la compra de un cliente de forma eficiente, precisa y con el cálculo de cambio.	Alta	5
27	Vendedor	Finalizar una venta	Completar la transacción, generar la factura y descontar el stock del inventario.	Alta	4
28	Encargado	Consultar historial de ventas	Acceder a transacciones pasadas para auditoría o consulta.	Alta	3
29	Encargado	Reimprimir una factura	Proporcionar una copia de la factura al cliente que lo solicite.	Media	2

N.	Rol	Funcionalidad	Resultado	Prioridad	Puntos
30	Encargado	Crear y gestionar una nueva compra	Reabastecer el inventario de forma organizada	Alta	4
31	Encargado	Procesar una compra	Finalizar la transacción con el proveedor y actualizar el inventario automáticamente.	Alta	4
32	Encargado	Ver historial de compras	Controlar los gastos y las entradas de productos para fines contables y de stock.	Media	3
33	Encargado	Registrar una merma de producto	Mantener un control exacto y registrar las pérdidas	Alta	3
34	Encargado	Editar un registro de merma	Corregir errores en la justificación o cantidad registrada de la pérdida.	Media	2
35	Encargado	Inactivar un registro de merma	Anular el registro de la pérdida si el producto se recupera o se devuelve a proveedor.	Media	2
36	Encargado	Registrar una regalía de producto	Mantener un control exacto de los artículos recibidos sin costo, aumentando las ganancias extras.	Alta	3
37	Encargado	Editar un registro de regalía	Corregir errores en la cantidad o descripción del producto regalado.	Media	2
38	Encargado	Inactivar un registro de regalía	Eliminar el registro si se ingresó por error o ya no es válido.	Media	1

N.	Rol	Funcionalidad	Resultado	Prioridad	Puntos
39	Administrador	Generar Reporte de Ventas	Analizar los ingresos del negocio por identificador, cliente o productos vendidos para toma de decisiones.	Alta	5
40	Administrador	Generar Reporte de Compras	Analizar las adquisiciones y gastos	Alta	4
41	Administrador	Generar Reporte Detallado de Productos	Obtener una visión completa del stock por marca, categoría, nombre y fecha de caducidad.	Alta	5
42	Administrador	Generar Reporte de Mermas	Analizar las pérdidas de inventario por producto y justificación.	Alta	4
43	Administrador	Generar Reporte de Regalías	Tener un registro exacto de los productos recibidos sin costo.	Media	3
44	Administrador	Generar Reporte de Proveedores	Analizar el estado y base de contactos de los proveedores, con capacidad de búsqueda por nombre.	Media	2
45	Administrador	Generar Reporte de Clientes	Analizar la base de clientes activos e inactivos y buscarlos por nombre.	Media	2

6.1.2. Criterios de aceptación

Para garantizar la calidad y la entrega de incrementos funcionales y aceptables, el Developer Team y la propietaria acordaron criterios de aceptación. Estos criterios dictaban las condiciones mínimas que una funcionalidad debía cumplir al final de un Sprint para ser considerada "Terminada" y lista para su inspección en la Sprint Review.

Estos criterios se basaron fundamentalmente en los Requerimientos No Funcionales documentados (Véase Anexo V Requerimiento de Usuario).

Tabla 4: Requerimientos No funcionales

Requerimientos No Funcionales	Descripción
Usabilidad	Facilidad de uso para el personal de la pulperia, incluso con mínima experiencia tecnológica.
Rendimiento	Tiempos de respuesta reducidos, incluso en operaciones críticas.
Seguridad	Protección de la información sensible como las ventas, inventario y clientes.
Confidencialidad	La información dentro del sistema será exclusivamente administrada por los usuarios designados.
Integridad	La información dentro del sistema será consistente con la operatividad y procesos internos gestionados por los administradores del sistema.
Disponibilidad	Garantizar el acceso y la operatividad constante del sistema.

6.2. Estructura de ejecución de la Metodología Scrum

Una vez definido el Product Backlog, el proyecto se implementó bajo la metodología Scrum, organizada mediante un marco operativo que garantizó la transparencia, la inspección y la adaptación en cada ciclo.

6.2.1. Scrum Team

El desarrollo del proyecto se llevó a cabo por un equipo reducido de dos integrantes. Dada esta composición, las responsabilidades se fusionaron, actuando ambos como Equipo de Desarrollo y compartiendo las funciones de Scrum Master y Product Owner. La pulpería Jehová Jireh se mantuvo como cliente final, participando activamente en la definición del alcance y validando el Incremento al finalizar cada Sprint.

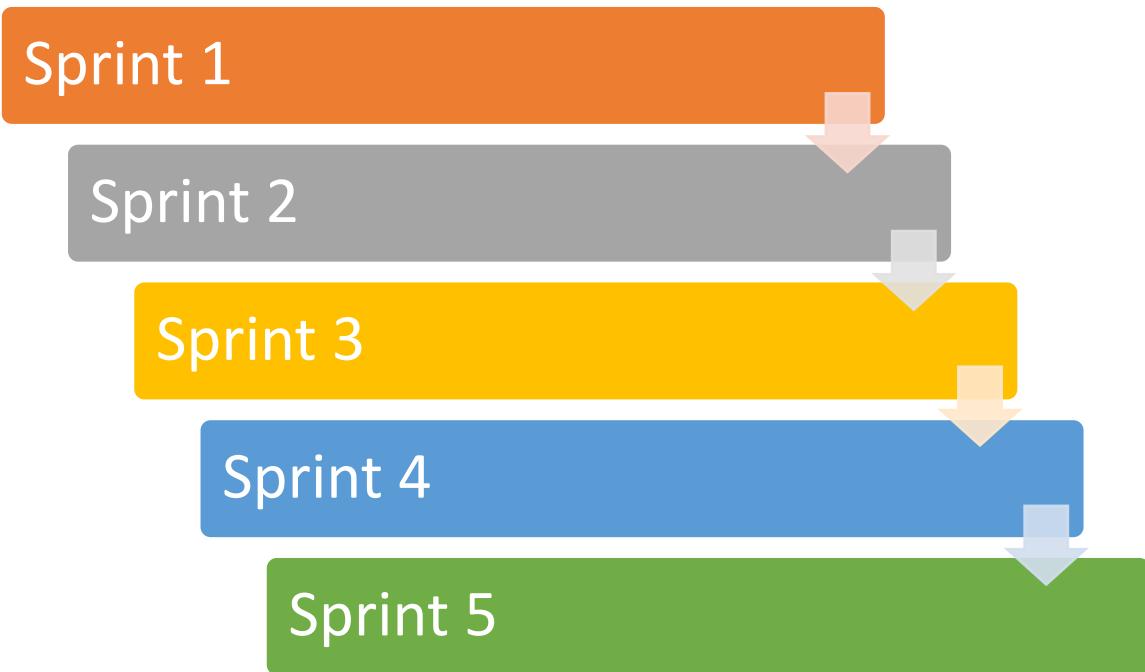
Tabla 5: Scrum Team

SCRUM TEAM	
Product Owner	Responsable de la gestión efectiva del Product Backlog (Cristhiam Súarez)
Scrum Master	Responsable del seguimiento y cumplimiento de las etapas del proyecto (Hosmand Belli)
Developer Team	Desarrolladores (Hosmand Belli, Cristhiam Súarez)

6.2.2. Sprints

El proyecto se llevó a cabo en un total de 5 Sprints, cada ciclo de trabajo se estableció con una duración fija de cuatro semanas. En cada Sprint se abordaron las historias de usuario seleccionadas del Sprint Backlog. Además, se garantizó un Incremento funcional al final de cada iteración, optimizando el ritmo de desarrollo y la entrega de un producto de valor.

Figura 3: Resumen de Sprints



6.2.2.1. Sprint Backlog

El Sprint Backlog fue determinado durante el Sprint Planning. Se evaluó según el criterio de estimación del Developer Team a cada historia de usuario definida en el Product Backlog. Se basó en la complejidad de la tarea y su prioridad.

Figura 4: Plantilla del Sprint Backlog

ORIGEN	ROL	ASIGNADO	ACCION	ESTADO	PRIORIDAD	HORAS	INICIO	FIN	DURACION (DIAS)

El registro formal del Sprint Backlog incluyó métricas esenciales de gestión como las detalladas a continuación:

Figura 5: Columnas de control

Columna de Control	Propósito
Origen	Define la naturaleza del ítem de trabajo extraída del Product Backlog
Asignado	Indica el miembro del equipo responsable de la ejecución de la tarea.
Acción	Descripción detallada de la tarea o historia de usuario a ejecutar.
Estado	Muestra la etapa actual del trabajo (ej: En Espera, Completado, En Ejecución, Testeo), crucial para la inspección en la Daily Scrum.
Prioridad	Clasifica la urgencia y el impacto del ítem dentro del Sprint Backlog (ej: Alta, Media, Baja).
Horas	Mide el esfuerzo real estimado en horas.
Inicio y Fin	Establecen la trazabilidad de la tarea enfocado en las fechas.

6.2.2.2. Resumen de eventos

Para controlar el alcance, el ritmo de trabajo y la aceptación del Incremento, se implementaron los siguientes eventos clave al inicio y al final de cada ciclo:

Figura 6: Ceremonias Scrum

Ceremonias Scrum	Descripción
Sprint planning	Se realizaron al previo al inicio de cada Sprint. En este evento, el equipo acordaba el objetivo del Sprint y seleccionaban los ítems del Product Backlog que se trasladarían al Sprint Backlog.
Daily Scrum	Estas reuniones de máximo quince minutos tenían como objetivo sincronizar las actividades del equipo, asegurar el ritmo de desarrollo diario y el identificar rápidamente algún impedimento que surgiera en el camino.
Sprint Review	Al finalizar un Sprint, está el Sprint Review, que son las reuniones donde se presentan los avances del proyecto. Estas reuniones se realizaron con la propietaria para la validación y aceptación formal del Incremento funcional, marcando el cierre oficial del Sprint.

6.2.2.3. Construcción del software

El desarrollo del software se llevó a cabo utilizando las tecnologías y herramientas reportadas en el 5.3. Tecnologías y Herramientas, específicamente las tecnologías como HTML, CSS, JavaScript y Bootstrap para el enfoque de desarrollo Frontend.

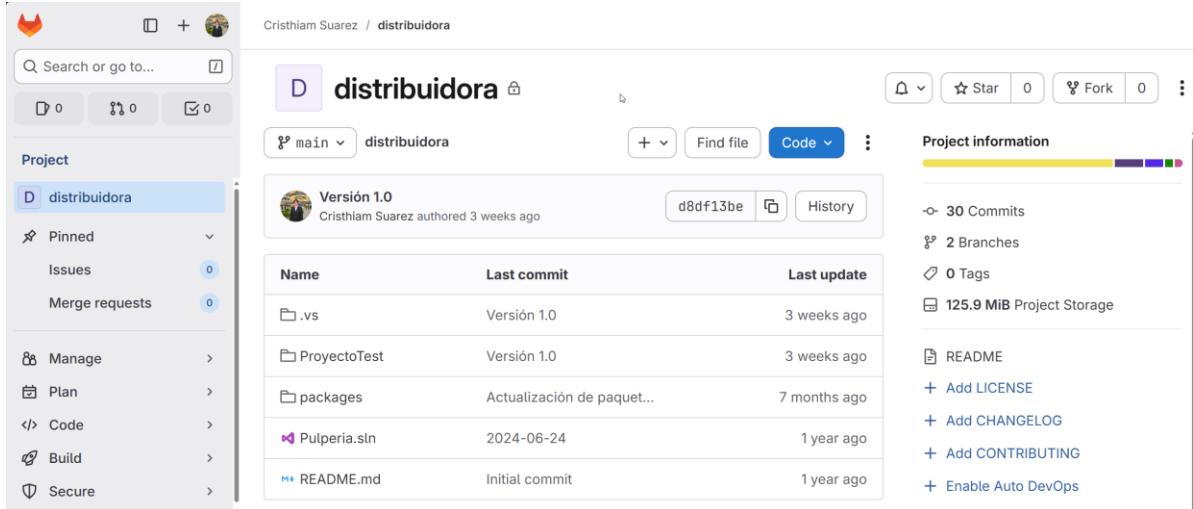
Por otro lado, el enfoque de desarrollo Backend las tecnologías como Visual Studio, C# ADO.NET, ASP.NET fueron las escogidas. El diseño, la gestión y diagramas relacionales de la base de datos se realizó utilizando las herramientas de SQL Server. Por otra parte, para la inteligencia de negocio, se utilizó Power BI enfocado en la creación de reportes y la toma de decisiones estratégicas.

Conforme al Developer Team ejecuto cada una de las tareas del Sprint Backlog, se realizaron pruebas unitarias, asegurando el correcto funcionamiento del software.

6.2.2.3.1. Control de versiones con GitLab

Para el control de versiones se utilizó Gitlab ya que es una plataforma en la nube que proporciona una variedad de herramientas para la gestión del ciclo de vida del desarrollo de software.

Figura 7: Conexión con gitlab



The screenshot shows the GitLab interface for the 'distribuidora' project. The left sidebar includes sections for 'Project' (distribuidora), 'Pinned', 'Issues', 'Merge requests', 'Manage', 'Plan', 'Code', 'Build', and 'Secure'. The main content area shows a commit history for 'Versión 1.0' by 'Cristhiam Suarez' 3 weeks ago. The commit list includes: '.vs' (Versión 1.0, 3 weeks ago), 'ProyectoTest' (Versión 1.0, 3 weeks ago), 'packages' (Actualización de paquet..., 7 months ago), 'Pulperia.sln' (2024-06-24, 1 year ago), and 'README.md' (Initial commit, 1 year ago). The right sidebar provides 'Project information' with metrics: 30 Commits, 2 Branches, 0 Tags, and 125.9 MiB Project Storage. It also includes links for README, LICENSE, CHANGELOG, CONTRIBUTING, and Auto DevOps.

6.2.2.3.2. Despliegue de incrementos

En cada Sprint Review se realizó el despliegue de una versión completa de los resultados obtenidos en cada uno de los Sprints en un entorno de pruebas, con el objetivo de la exploración e identificación de errores por parte del cliente. En esta fase, la pulperia tuvo la oportunidad de poner a prueba cada ítem, tarea, modulo o funcionalidad detallada en el Product Backlog.

6.3. Resultados

En esta sección del trabajo monográfico se abordan los resultados obtenidos durante las fases de desarrollo, diseño y pruebas realizadas al sistema web de inventario y facturación. Debido a limitaciones de espacio, se seleccionó un subconjunto de los Sprints para presentarse en esta sección, y los Sprints restantes se documentaron en Anexos.

6.3.1. Sprint 1

Comenzamos con el Sprint 1, el objetivo de este centró en establecer la infraestructura inicial del proyecto, implementando funciones completas de seguridad y acceso, además de entregar funcionalidades en módulos como los Usuarios, Clientes, Proveedores y Empresa.

6.3.1.1. Desarrollo del Sprint Backlog

6.3.1.1.1. Configuración inicial e infraestructura

Se completaron las tareas de infraestructura básicas, incluyendo la creación del proyecto ASP.NET MVC, su configuración inicial, la conexión con GitLab para el control de versiones y la conexión para la base de datos.

Tabla 6: Sprint Backlog de configuración inicial

ORIGEN	ROL	ASIGNADO	ACCION	ESTADO	PRIORIDAD	HORAS
INFRAESTRUCTURA	Developer	Cristhiam Omar Suarez Nuñez	Creación de Proyecto ASP.NET MVC	Completada	Media	4.00
INFRAESTRUCTURA	Developer	Cristhiam Omar Suarez Nuñez	Configuracion inicial del Proyecto	Completada	Baja	5.00
INFRAESTRUCTURA	Developer	Cristhiam Omar Suarez Nuñez	Conexión con GitLab	Completada	Baja	3.00
INFRAESTRUCTURA	DBA	Cristhiam Omar Suarez Nuñez	Conexiones de BD con Proyecto ASP.NET MVC	Completada	Alta	4.00
INFRAESTRUCTURA	DBA	Cristhiam Omar Suarez Nuñez	Definición e Implementación de roles	Completada	Media	8.00
INFRAESTRUCTURA	DBA	Cristhiam Omar Suarez Nuñez	Conexiones de BD con Proyecto ASP.NET MVC	Completada	Media	4.00

Figura 8: Creación del proyecto en ASP.NET

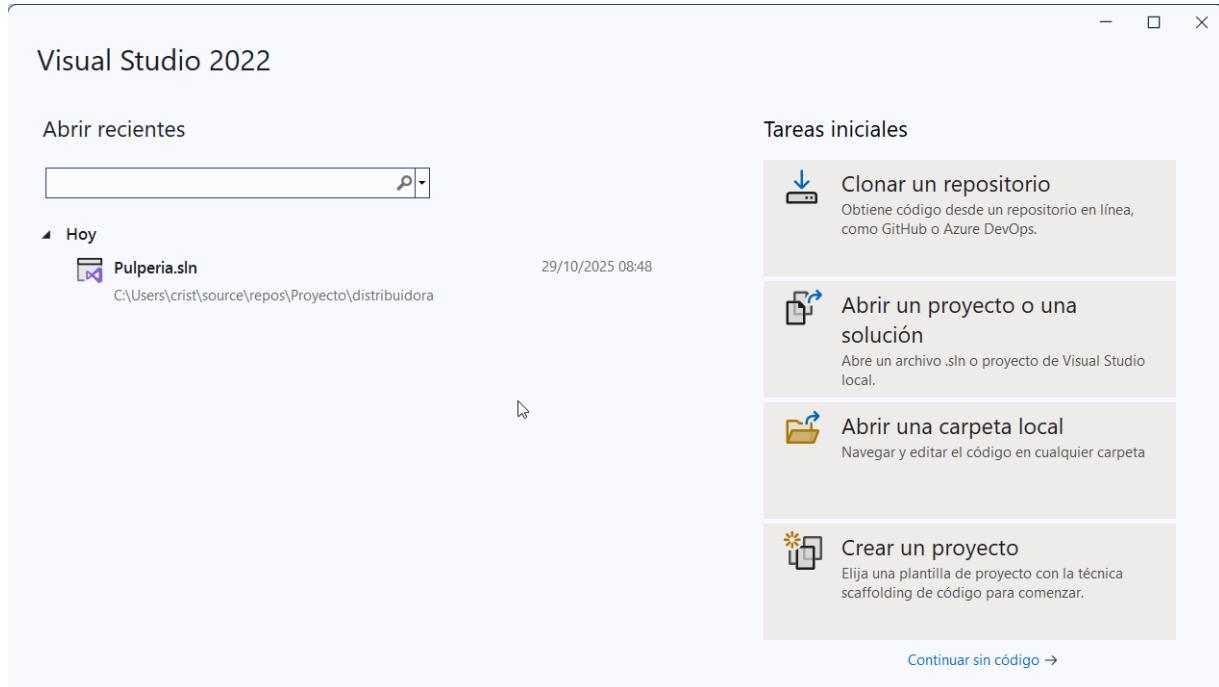


Figura 9: Conexión con la base de datos

```
namespace ProyectoTest.Logica
{
    public class Conexion
    {
        public static string CN = "Data Source=CRISTHIAM\\SQLEXPRESS;Initial Catalog=DBA_JehovaJireh;Integrated Security=True";
    }
}
```

Figura 10: Configuración inicial del proyecto

```
1 referencia
public class RouteConfig
{
    1 referencia
    public static void RegisterRoutes(RouteCollection routes)
    {
        routes.IgnoreRoute("{resource}.axd/{*pathInfo}");

        routes.MapRoute(
            name: "Default",
            url: "{controller}/{action}/{id}",
            defaults: new { controller = "Login", action = "Index", id = UrlParameter.Optional }
        );

        routes.MapRoute(
            name: "CatchAll",
            url: "{*url}",
            defaults: new { controller = "Error", action = "NotFound" }
        );
    }
}
```

6.3.1.1.2. Seguridad y acceso

Se implementó el módulo de Login con todas sus capas, paralelamente con el módulo de Usuarios. Finalmente se completó la estructura de Roles y Permisos, lo que incluye la creación de múltiples tablas relacionales, la lógica y vistas correspondientes.

Tabla 7: Sprint Backlog seguridad y acceso

ORIGEN	ROL	ASIGNADO	ACCION	ESTADO	PRIORIDAD	HORAS
LOGIN	Developer	Hosmand Rafael Belli Cespedes	Creación de [dbo].[Usuario]	Completada	Alta	3.00
LOGIN	Developer	Hosmand Rafael Belli Cespedes	Creación de SPS para Login	Completada	Alta	3.00
LOGIN	Developer	Hosmand Rafael Belli Cespedes	Creación de logica: Login	Completada	Media	4.00
LOGIN	Developer	Hosmand Rafael Belli Cespedes	Creación de controlador: Login	Completada	Alta	1.00
LOGIN	Developer	Hosmand Rafael Belli Cespedes	Creación de vista: Login	Completada	Alta	4.00
LOGIN	Qua Tester	Hosmand Rafael Belli Cespedes	Pruebas unitarias de la lógica de autenticación.	Completada	Alta	2.00
USUARIOS	Developer	Hosmand Rafael Belli Cespedes	Creación de SPS para Usuarios	Completada	Media	2.00
USUARIOS	Developer	Hosmand Rafael Belli Cespedes	Creación de clases: Usuarios	Completada	Media	1.00
USUARIOS	Developer	Hosmand Rafael Belli Cespedes	Creación de logica: Usuarios	Completada	Media	2.00
USUARIOS	Developer	Hosmand Rafael Belli Cespedes	Creación de controlador: Usuarios	Completada	Media	1.00
USUARIOS	Developer	Hosmand Rafael Belli Cespedes	Creación de vista: Usuarios	Completada	Media	2.00

USUARIOS	Qua Tester	Hosmand Rafael Belli Cespedes	Pruebas unitarias de Usuarios	Completada	Media	2.00
ROLES	Developer	Cristhiam Omar Suarez Nuñez	Creación de [dbo].[Rol]	Completada	Media	1.00
ROLES	Developer	Cristhiam Omar Suarez Nuñez	Creación de [dbo].[RolAccion]	Completada	Media	1.00
ROLES	Developer	Cristhiam Omar Suarez Nuñez	Creación de [dbo].[RolModulo]	Completada	Media	1.00
ROLES	Developer	Cristhiam Omar Suarez Nuñez	Creación de [dbo].[RolPermiso]	Completada	Media	1.00
ROLES	Developer	Cristhiam Omar Suarez Nuñez	Creación de SPS para Rol	Completada	Media	3.00
ROLES	Developer	Cristhiam Omar Suarez Nuñez	Creación de clases: Rol	Completada	Media	1.00
ROLES	Developer	Cristhiam Omar Suarez Nuñez	Creación de logica: Rol	Completada	Media	3.00
ROLES	Developer	Cristhiam Omar Suarez Nuñez	Creación de controlador: Rol	Completada	Media	1.00
ROLES	Developer	Cristhiam Omar Suarez Nuñez	Creación de vista: Rol	Completada	Media	2.00
ROLES	Qua Tester	Cristhiam Omar Suarez Nuñez	Pruebas unitarias de Rol	Completada	Media	2.00

Figura 11: Procedimiento almacenado Login

```
ALTER proc [dbo].[sp_obtenerUsuario](
    @Usuario varchar(100),
    @Contrasena varchar(100)
)
as
begin
    DECLARE @EncriptarContrasena VARBINARY(8000)
    SELECT @EncriptarContrasena = ENCRYPTBYPASSPHRASE('password', @Contrasena)

    DECLARE @DesencriptarContrasena VARCHAR(100)
    SELECT @DesencriptarContrasena = CONVERT(VARCHAR(MAX), DECRYPTBYPASSPHRASE('password', ContrasenaUsuario))

    IF EXISTS (SELECT * FROM usuario WHERE CorreoUsuario = @Usuario)
    begin
        SELECT
            u.IdUsuario,
            u.IdRol,
            r.NombreRol,
            r.EstadoRol AS [ActivoRol],
            u.NombreUsuario,
            u.ApellidoUsuario,
            u.CorreoUsuario,
            u.ContrasenaUsuario,
            u.TelefonoUsuario,
            u.DireccionUsuario,
            u.EstadoUsuario,
            u.FechaRegistroUsuario
        FROM Usuario u INNER JOIN Rol r ON u.IdRol = r.IdRol
        WHERE CorreoUsuario = @Usuario and @Contrasena = @DesencriptarContrasena
    end
end
```

Figura 12: Lógica de Login

```
1 referencia
public void EnviarCorreo(string para, string token)
{
    string url = Url.Action("ConfirmarToken", "Login", new { token = token }, protocol: Request.Url.Scheme);

    MailMessage mail = new MailMessage();
    mail.From = new MailAddress("jehova.jireh45u@gmail.com");
    mail.To.Add(para);
    mail.Subject = "Recuperación de contraseña";
    mail.Body = $"<p>Haz clic en el siguiente enlace para reestablecer tu contraseña:</p><p><a href='{url}'>F
    mail.IsBodyHtml = true;

    SmtpClient smtp = new SmtpClient("smtp.gmail.com", 587);
    smtp.Credentials = new NetworkCredential("jehova.jireh45u@gmail.com", "qzwc dges lstk okdc");
    smtp.EnableSsl = true;

    smtp.Send(mail);
}
```

Figura 13: Controlador Login

```
namespace ProyectoTest.Controllers
{
    public class LoginController : Controller
    {
        private static Usuario oUsuario;
        // GET: Login
        public ActionResult Index()
        {
            return View();
        }

        public ActionResult RecuperarContrasena()
        {
            return View();
        }

        public ActionResult ComenzarRecuperarContrasena()
        {
            return View();
        }
    }
}
```

Figura 14: Vista Login

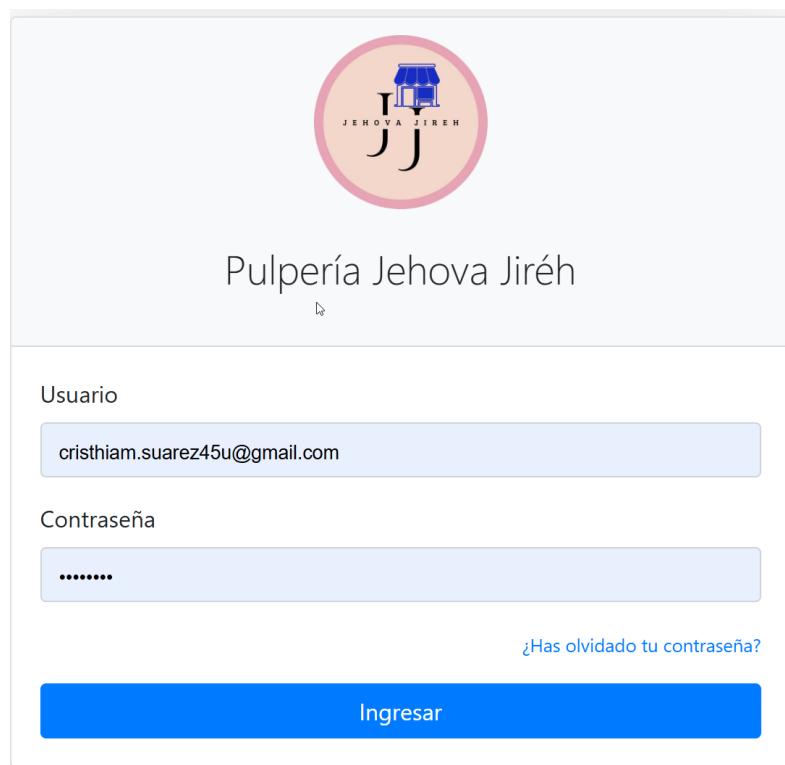


Figura 15: Tabla de Usuario

Usuario	
IdUsuario	
NombreUsuario	
ApellidoUsuario	
CorreoUsuario	
ContrasenaUsuario	
EstadoUsuario	
FechaRegistroUsuario	
TelefonoUsuario	
DireccionUsuario	
TokenRecoveryUsuario	
IdRol	

Figura 16: Procedimiento almacenado Usuario

```
begin
    set nocount on;
    set @Resultado = 1;

    -- Encriptar contraseña
    declare @EncriptarContrasena varbinary(8000);
    select @EncriptarContrasena = encryptbypassphrase('password', @Contrasena);

    -- Verificar si el usuario ya existe
    if not exists (select 1 from Usuario where CorreoUsuario = @Usuario)
    begin
        -- Insertar el nuevo usuario
        insert into Usuario (
            NombreUsuario, ApellidoUsuario, CorreoUsuario, ContrasenaUsuario,
            TelefonoUsuario, DireccionUsuario, IdRol, EstadoUsuario
        )
        values (
            @Nombres, @Apellidos, @Usuario, @EncriptarContrasena,
            @Telefono, @Direccion, @IdRol, @Activo
        );

        declare @IdModulo int = (select top 1 IdRolModulo from RolModulo where NombreRolModulo = 'Usuarios');
        declare @IdAccion int = (select top 1 IdRolAccion from RolAccion where NombreRolAccion = 'Crear');
        declare @NombreRol varchar(100) = (select top 1 NombreRol from Rol where IdRol = @IdRol);
    end
end;
```

Figura 17: Clase Usuario

```
6     < namespace ProyectoTest.Models
7     {
8         < 31 referencias
9         < public class Usuario
10        {
11            < 64 referencias
12            public int IdUsuario { get; set; }
13            < 7 referencias
14            public Rol oRol { get; set; }
15            < 6 referencias
16            public string Nombres { get; set; }
17            < 6 referencias
18            public string Apellidos { get; set; }
19            < 9 referencias
20            public string Correo { get; set; }
21            < 3 referencias
22            public string Contrasena { get; set; }
23            < 6 referencias
24            public int Telefono { get; set; }
25            < 6 referencias
26            public string Direccion { get; set; }
27            < 4 referencias
28            public string FechaRegistro { get; set; }
29            < 6 referencias
30            public bool Activo { get; set; }
31        }
32    }
```

Figura 18: Lógica de Usuario

```
public List<Usuario> Listar(int _idusuario)
{
    List<Usuario> rptListaUsuario = new List<Usuario>();
    using (SqlConnection oConexion = new SqlConnection(Conexion.CN))
    {
        SqlCommand cmd = new SqlCommand("sp_obtenerListaDeUsuario", oConexion);
        cmd.CommandType = CommandType.StoredProcedure;

        try
        {
            oConexion.Open();
            SqlDataReader dr = cmd.ExecuteReader();

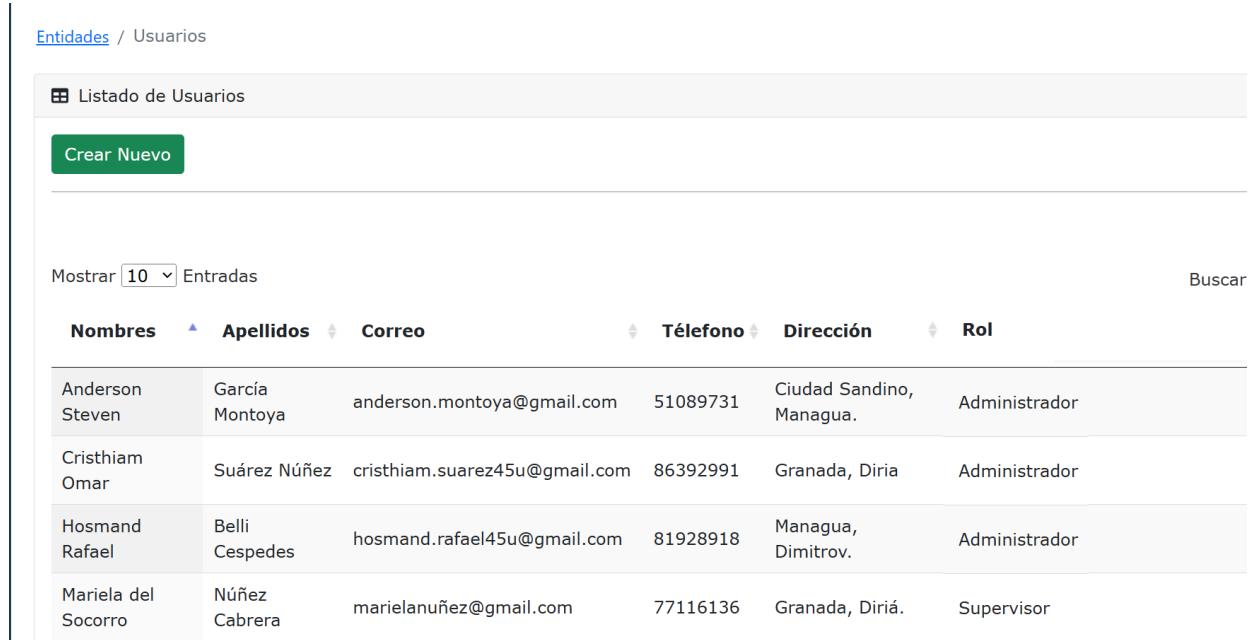
            while (dr.Read())
            {
                rptListaUsuario.Add(new Usuario()
                {
                    IdUsuario = Convert.ToInt32(dr["IdUsuario"].ToString()),
                    oRol = new Rol() { IdRol = Convert.ToInt32(dr["IdRol"].ToString()), NombreRol = dr["NombreRol"].ToString },
                    Nombres = dr["NombreUsuario"].ToString(),
                    Apellidos = dr["ApellidoUsuario"].ToString(),
                    Correo = dr["CorreoUsuario"].ToString(),
                    Activo = Convert.ToBoolean(dr["EstadoUsuario"].ToString()),
                    FechaRegistro = dr["FechaRegistroUsuario"].ToString(),
                    Telefono = Convert.ToInt32(dr["TelefonoUsuario"].ToString()),
                    Direccion = dr["DireccionUsuario"].ToString()
                });
            }
            dr.Close();
        }
    }
}
```

Figura 19: Controlador de Usuario

```
0 referencias
public ActionResult Usuario()
{
    if (Session["Usuario"] == null)
        return RedirectToAction("Index", "Login");

    return View();
}
```

Figura 20: Vista de Usuario



The screenshot shows a user management interface. At the top, there is a breadcrumb navigation: 'Entidades / Usuarios'. Below this is a header with a 'Listado de Usuarios' button and a 'Crear Nuevo' button. The main area is a table with the following data:

Nombres	Apellidos	Correo	Télefono	Dirección	Rol
Anderson Steven	García Montoya	anderson.montoya@gmail.com	51089731	Ciudad Sandino, Managua.	Administrador
Cristhiam Omar	Suárez Núñez	cristhiam.suarez45u@gmail.com	86392991	Granada, Diriá	Administrador
Hosmand Rafael	Belli Cespedes	hosmand.rafael45u@gmail.com	81928918	Managua, Dimitrov.	Administrador
Mariela del Socorro	Núñez Cabrera	marielanuñez@gmail.com	77116136	Granada, Diriá.	Supervisor

Figura 21: Crear Usuario

Usuarios

Nombres	Apellidos
<input type="text"/>	<input type="text"/>
Nombre de Usuario	
<input type="text"/>	
Contraseña	
<input type="text"/>	
Teléfono	Dirección
<input type="text"/>	<input type="text"/>
Es Administrador	Estado
<input type="button" value="Si"/>	<input type="button" value="Activo"/>

Figura 22: Tablas de Roles

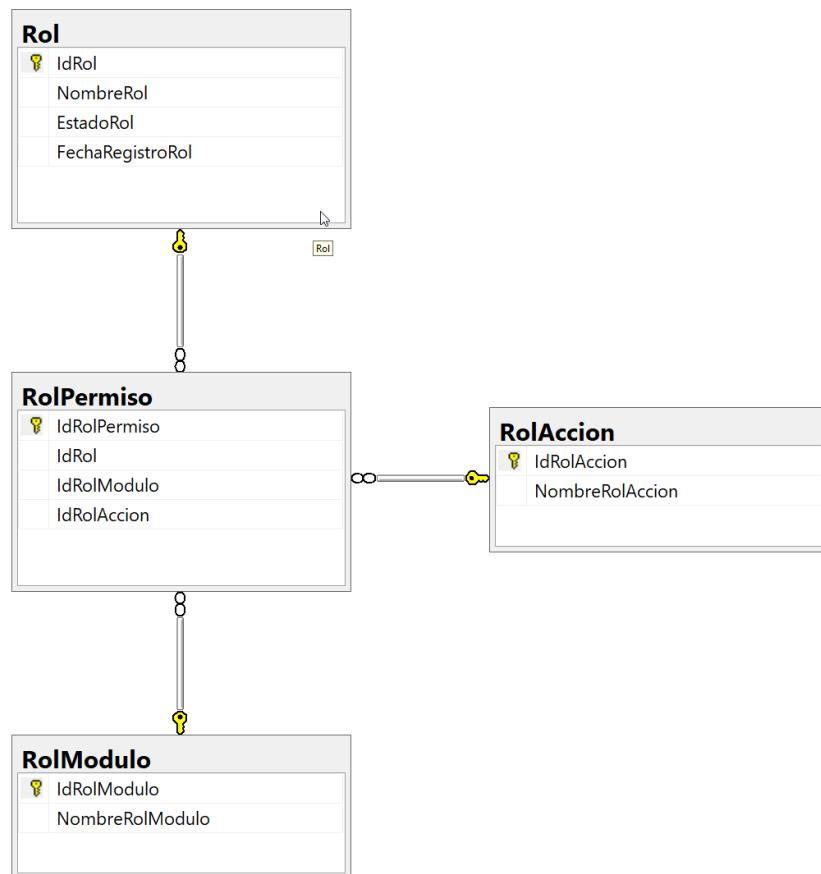


Figura 23: Procedimiento almacenado Rol

```

BEGIN
    SET NOCOUNT ON;
    SET @Resultado = 1;
    SET @IdRol = 0 parameter @Resultado bit OUTPUT

    BEGIN TRY
        IF EXISTS (
            SELECT 1 FROM Rol WHERE LOWER(RTRIM(NombreRol)) = LOWER(RTRIM(@NombreRol))
        )
        BEGIN
            SET @Resultado = 0;
            RETURN;
        END

        INSERT INTO Rol (NombreRol, EstadoRol)
        VALUES (@NombreRol, @Activo);

        SET @IdRol = SCOPE_IDENTITY();

        DECLARE @IdModulo INT = (SELECT IdRolModulo FROM RolModulo WHERE NombreRolModulo = 'Roles');
        DECLARE @IdAccion INT = (SELECT IdRolAccion FROM RolAccion WHERE NombreRolAccion = 'Crear');
        DECLARE @Estado NVARCHAR(20) = CASE WHEN @Activo = 1 THEN 'Activo' ELSE 'No Activo' END;

        IF @IdModulo IS NOT NULL AND @IdAccion IS NOT NULL
        BEGIN
            INSERT INTO RegActividad (IdUsuario, IdRolModulo, IdRolAccion, DetallesRegActividad)
            VALUES (
                @IdUsuario,
                @IdModulo,
                @IdAccion,
                'Se registró el rol: ' + @NombreRol + ' con estado: ' + @Estado
            );
        END
    END TRY
    BEGIN CATCH
        SET @Resultado = 0;
        SET @IdRol = 0;
    END CATCH
END

```

Figura 24: Clase Rol

```

8     13 referencias
9
10    12 referencias
11    7 referencias
12    7 referencias
13    4 referencias
14

```

public class Rol

```

{
    public int IdRol { get; set; }
    public string NombreRol { get; set; }
    public bool Activo { get; set; }
    public List<Permiso> Permisos { get; set; }
}

```

Figura 25: Lógica Rol



```
1 referencia
public List<Rol> Listar(int _idusuario)
{
    List<Rol> rptListaRol = new List<Rol>();
    using (SqlConnection oConexion = new SqlConnection(Conexion.CN))
    {
        SqlCommand cmd = new SqlCommand("sp_obtenerListarRoles", oConexion);
        cmd.CommandType = CommandType.StoredProcedure;

        try
        {
            oConexion.Open();
            SqlDataReader dr = cmd.ExecuteReader();

            while (dr.Read())
            {
                rptListaRol.Add(new Rol()
                {
                    IdRol = Convert.ToInt32(dr["IdRol"].ToString()),
                    NombreRol = dr["NombreRol"].ToString(),
                    Activo = Convert.ToBoolean(dr["EstadoRol"].ToString()),
                    FechaRegistroRol = dr["FechaRegistroRol"].ToString(),
                });
            }
            dr.Close();
        }
        return rptListaRol;
    }
}
```

Figura 26: Controlador Rol

0 referencias

```
public ActionResult Rol()
{
    if (Session["Usuario"] == null)
        return RedirectToAction("Index", "Login");

    return View();
}
```

Figura 27: Vista Rol

[Entidades](#) / Roles

Listado de Roles	
Crear Nuevo	
 Copiar	 Excel
 PDF	 CSV
 Imprimir	
Nombre del Rol	Estado
Administrador	Activo
Empleado	Activo
Supervisor	Activo

Figura 28: Crear Rol

Permisos del Rol	
Menú Principal	<input type="checkbox"/> Consultar
Categorías	<input type="checkbox"/> Consultar <input type="checkbox"/> Crear <input type="checkbox"/> Editar <input type="checkbox"/> Eliminar
Productos	<input type="checkbox"/> Consultar <input type="checkbox"/> Crear <input type="checkbox"/> Editar <input type="checkbox"/> Eliminar
Roles	<input type="checkbox"/> Consultar
Clientes	<input type="checkbox"/> Consultar <input type="checkbox"/> Crear <input type="checkbox"/> Editar <input type="checkbox"/> Eliminar
Empresas	<input type="checkbox"/> Consultar <input type="checkbox"/> Crear <input type="checkbox"/> Editar <input type="checkbox"/> Eliminar
Regalías	<input type="checkbox"/> Consultar <input type="checkbox"/> Crear <input type="checkbox"/> Eliminar
Usuarios	<input type="checkbox"/> Consultar
Proveedores	<input type="checkbox"/> Consultar <input type="checkbox"/> Crear <input type="checkbox"/> Editar <input type="checkbox"/> Eliminar
Mermas	<input type="checkbox"/> Consultar <input type="checkbox"/> Crear <input type="checkbox"/> Eliminar
Unidades de Medida	<input type="checkbox"/> Consultar <input type="checkbox"/> Crear <input type="checkbox"/> Editar <input type="checkbox"/> Eliminar
Compras	<input type="checkbox"/> Consultar

6.3.1.1.3. Módulos funcionales

Se implementaron los módulos de Clientes, Proveedores y Empresa.

ORIGEN	ROL	ASIGNADO	ACCION	ESTADO	PRIORIDAD	HORAS
CLIENTES	Developer	Cristhiam Omar Suarez Nuñez	Creación de [dbo].[Clientes]	Completada	Alta	4.00
CLIENTES	Developer	Cristhiam Omar Suarez Nuñez	Creación de SPS para Clientes	Completada	Alta	4.00
CLIENTES	Developer	Cristhiam Omar Suarez Nuñez	Creación de clases: Clientes	Completada	Media	1.00
CLIENTES	Developer	Cristhiam Omar Suarez Nuñez	Creación de logica: Clientes	Completada	Media	3.00
CLIENTES	Developer	Cristhiam Omar Suarez Nuñez	Creación de controlador:Clientes	Completada	Media	1.00
CLIENTES	Qua Tester	Cristhiam Omar Suarez Nuñez	Pruebas unitarias de Clientes	Completada	Media	2.00
PROVEEDORES	Developer	Hosmand Rafael Belli Cespedes	Creación de [dbo].[Proveedor]	Completada	Alta	4.00
PROVEEDORES	Developer	Hosmand Rafael Belli Cespedes	Creación de SPS para Proveedor	Completada	Alta	4.00
PROVEEDORES	Developer	Hosmand Rafael Belli Cespedes	Creación de clases:Proveedor	Completada	Media	1.00
PROVEEDORES	Developer	Hosmand Rafael Belli Cespedes	Creación de logica: Proveedor	Completada	Media	2.00
PROVEEDORES	Developer	Hosmand Rafael Belli Cespedes	Creación de controlador: Proveedor	Completada	Media	1.00
PROVEEDORES	Qua Tester	Hosmand Rafael Belli Cespedes	Pruebas unitarias de Proveedor	Completada	Media	2.00
EMPRESA	Developer	Cristhiam Omar Suarez Nuñez	Creación de [dbo].[Empresa]	Completada	Media	4.00
EMPRESA	Developer	Cristhiam Omar Suarez Nuñez	Creación de SPS para Empresa	Completada	Media	4.00
EMPRESA	Developer	Hosmand Rafael Belli Cespedes	Creación de clases:Empresa	Completada	Media	4.00
EMPRESA	Developer	Cristhiam Omar Suarez Nuñez	Creación de logica: Empresa	Completada	Media	1.00
EMPRESA	Developer	Cristhiam Omar Suarez Nuñez	Creación de controlador: Empresa	Completada	Media	1.00
EMPRESA	Qua Tester	Cristhiam Omar Suarez Nuñez	Pruebas unitarias de Empresa	Completada	Media	4.00

Tabla 8: Sprint Backlog de módulos funcionales

Para estas dos entidades, se ejecutó el ciclo completo de desarrollo por capas. Esto incluyó la creación de las tablas correspondientes en la Base de Datos, el diseño y la implementación de sus respectivos procedimientos almacenados.

Figura 29: Tabla Cliente

Cliente	
	IdCliente
	Cliente
	NombreCliente
	ApellidoCliente
	CedulaCliente
	DireccionCliente
	TelefonoCliente
	EstadoCliente
	FechaRegistroCliente

Figura 30: Tabla Proveedor

Proveedor	
	IdProveedor
	Proveedor
	NombreProveedor
	TelefonoProveedor
	EstadoProveedor
	FechaRegistroProveedor
	ApellidoProveedor

Figura 31: Procedimiento almacenado Cliente

```

|BEGIN
|  SET NOCOUNT ON;
|  SET @Resultado = 1;

|  IF NOT EXISTS (SELECT 1 FROM Cliente WHERE CedulaCliente = @Cedula)
|  BEGIN
|    INSERT INTO Cliente (NombreCliente, ApellidoCliente, CedulaCliente, DireccionCliente,
|    TelefonoCliente, EstadoCliente)
|    VALUES (@NombreCliente, @ApellidoCliente, @Cedula, @Direccion, @Telefono, @Activo);

|    -- Obtener IDs de módulo y acción
|    DECLARE @IdModulo INT = (SELECT IdRolModulo FROM RolModulo WHERE NombreRolModulo = 'Clientes');
|    DECLARE @IdAccion INT = (SELECT IdRolAccion FROM RolAccion WHERE NombreRolAccion = 'Crear');

|    -- Insertar en UsuarioActividad
|    IF @IdModulo IS NOT NULL AND @IdAccion IS NOT NULL
|    BEGIN
|      INSERT INTO RegActividad (IdUsuario, IdRolModulo, IdRolAccion, DetallesRegActividad)
|      VALUES (
|        @IdUsuario,
|        @IdModulo,
|        @IdAccion,
|        'Se registró un nuevo cliente: ' + @NombreCliente + ' ' + @ApellidoCliente + ', cédula: ' +
|        @Cedula + ', con estado: ' + CASE WHEN @Activo = 1 THEN 'Activo' ELSE 'No Activo' END
|      );
|    END
|  END
|END

```

Figura 32: Procedimiento almacenado Proveedor

```

|BEGIN
|  SET NOCOUNT ON;
|  SET @Resultado = 1;

|  IF EXISTS (SELECT 1 FROM Proveedor WHERE TelefonoProveedor = @Telefono)
|  BEGIN
|    SET @Resultado = 0;
|    RETURN;
|  END

|  INSERT INTO Proveedor (NombreProveedor, ApellidoProveedor, TelefonoProveedor, EstadoProveedor)
|  VALUES (@NombreProveedor, @ApellidoProveedor, @Telefono, @Activo);

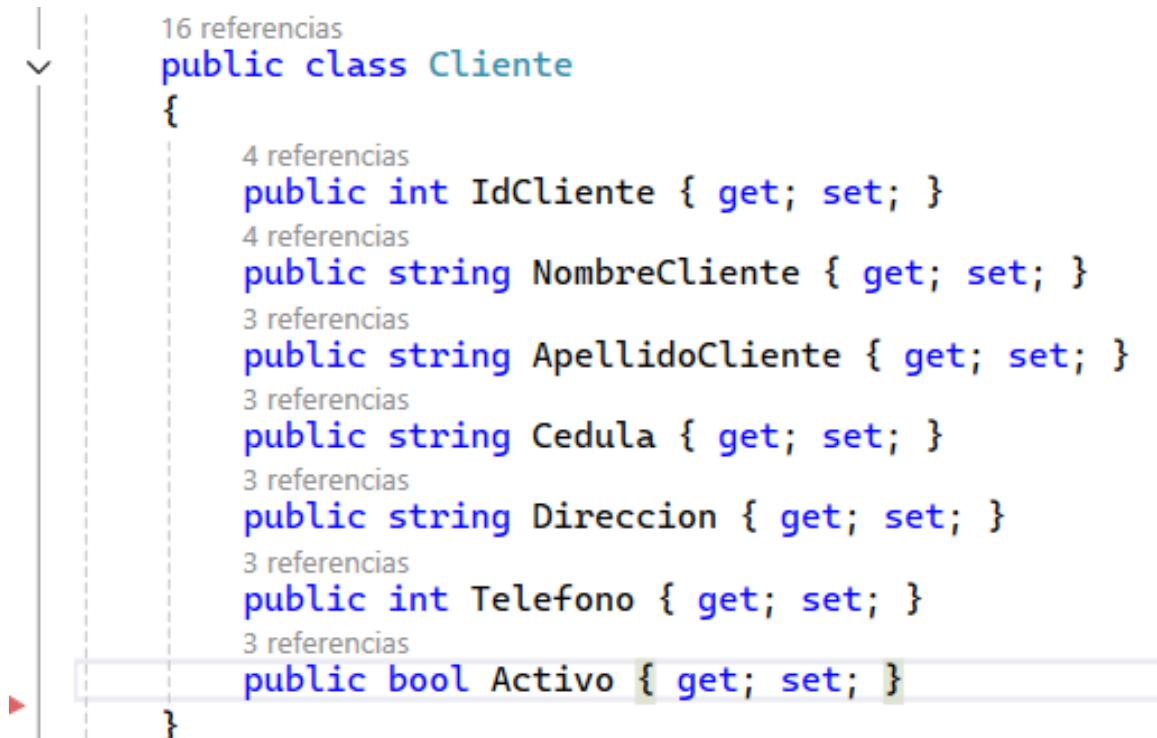
|  DECLARE @IdModulo INT = (SELECT TOP 1 IdRolModulo FROM RolModulo WHERE NombreRolModulo = 'Proveedores');
|  DECLARE @IdAccion INT = (SELECT TOP 1 IdRolAccion FROM RolAccion WHERE NombreRolModulo = 'Crear');

|  IF @IdModulo IS NOT NULL AND @IdAccion IS NOT NULL
|  BEGIN
|    -- Solo registrar log si hubo una fila afectada
|    IF @@ROWCOUNT > 0
|    BEGIN
|      INSERT INTO RegActividad(IdUsuario, IdRolModulo, IdRolAccion, DetallesRegActividad)
|      VALUES (
|        @IdUsuario,
|        @IdModulo,
|        @IdAccion,
|        'Se registró un proveedor: ' + @NombreProveedor +
|        ', Teléfono: ' + CAST(@Telefono AS NVARCHAR) +
|        ', con estado: ' + CASE WHEN @Activo = 1 THEN 'Activo' ELSE 'No Activo' END
|      );
|    END
|  END
|END

```

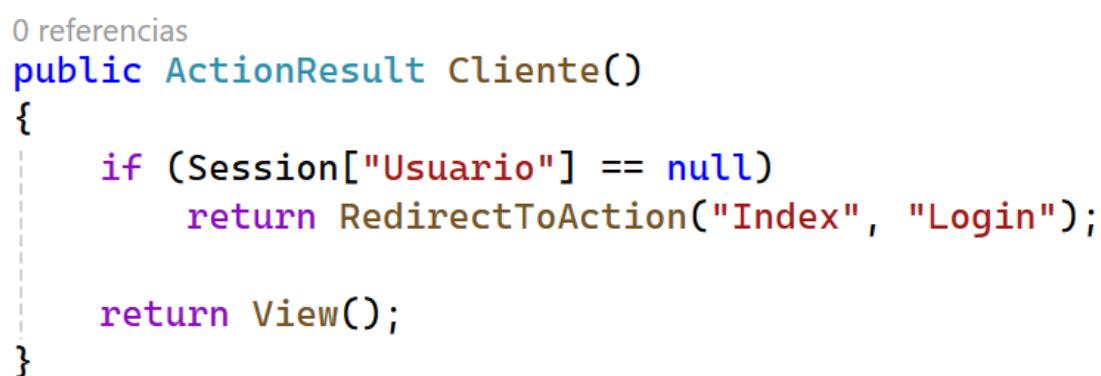
A nivel de codificación, se implementó la lógica de negocio en el backend, y el desarrollo de las Vistas de registro y listado. Este logro habilitó al sistema a registrar y mantener los contactos comerciales, sentando el soporte estructural para los futuros módulos.

Figura 33: Clase Cliente



```
16 referencias
public class Cliente
{
    4 referencias
    public int IdCliente { get; set; }
    4 referencias
    public string NombreCliente { get; set; }
    3 referencias
    public string ApellidoCliente { get; set; }
    3 referencias
    public string Cedula { get; set; }
    3 referencias
    public string Direccion { get; set; }
    3 referencias
    public int Telefono { get; set; }
    3 referencias
    public bool Activo { get; set; }
}
```

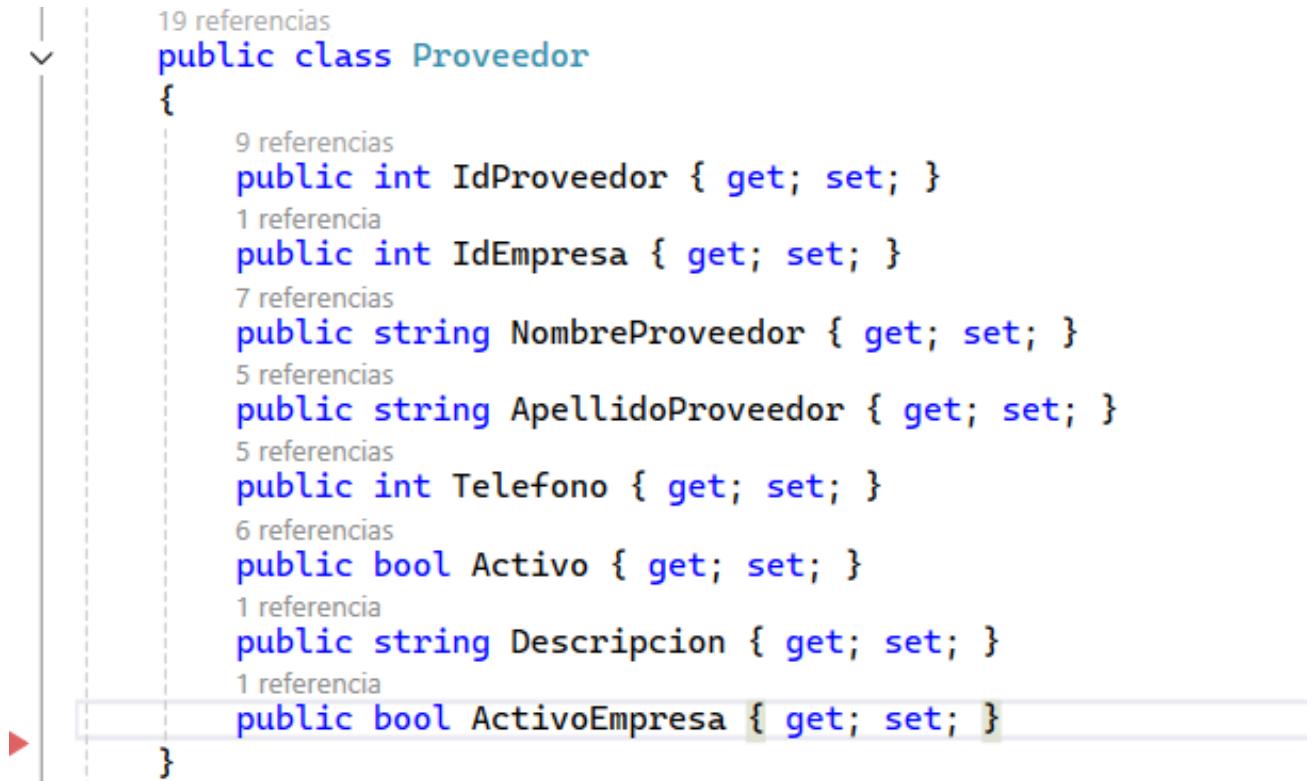
Figura 34: Controlador Cliente



```
0 referencias
public ActionResult Cliente()
{
    if (Session["Usuario"] == null)
        return RedirectToAction("Index", "Login");

    return View();
}
```

Figura 35: Clase Proveedor



```
19 referencias
public class Proveedor
{
    9 referencias
    public int IdProveedor { get; set; }
    1 referencia
    public int IdEmpresa { get; set; }
    7 referencias
    public string NombreProveedor { get; set; }
    5 referencias
    public string ApellidoProveedor { get; set; }
    5 referencias
    public int Telefono { get; set; }
    6 referencias
    public bool Activo { get; set; }
    1 referencia
    public string Descripcion { get; set; }
    1 referencia
    public bool ActivоЕmpresa { get; set; }
}
```

Figura 36: Controlador Proveedor

```
0 referencias
public ActionResult Proveedor()
{
    if (Session["Usuario"] == null)
        return RedirectToAction("Index", "Login");

    return View();
}
```

Figura 37: Lógica Cliente

```
List<Cliente> rptListaCliente = new List<Cliente>();
using (SqlConnection oConexion = new SqlConnection(Conexion.CN))
{
    SqlCommand cmd = new SqlCommand("sp_obtenerCliente", oConexion);
    cmd.CommandType = CommandType.StoredProcedure;

    try
    {
        oConexion.Open();
        SqlDataReader dr = cmd.ExecuteReader();

        while (dr.Read())
        {
            rptListaCliente.Add(new Cliente())
            {
                IdCliente = Convert.ToInt32(dr["IdCliente"].ToString()),
                NombreCliente = dr["NombreCliente"].ToString(),
                ApellidoCliente = dr["ApellidoCliente"].ToString(),
                Cedula = dr["CedulaCliente"].ToString(),
                Direccion = dr["DireccionCliente"].ToString(),
                Telefono = Convert.ToInt32(dr["TelefonoCliente"].ToString()),
                Activo = Convert.ToBoolean(dr["EstadoCliente"].ToString()),
                FechaRegistroCliente = dr["FechaRegistroCliente"].ToString()
            });
        }
        dr.Close();
    }
}
```

Figura 38: Lógica Proveedor

```
List<Proveedor> rptListaProveedor = new List<Proveedor>();
using (SqlConnection oConexion = new SqlConnection(Conexion.CN))
{
    SqlCommand cmd = new SqlCommand("sp_obtenerProveedor", oConexion);
    cmd.CommandType = CommandType.StoredProcedure;

    try
    {
        oConexion.Open();
        SqlDataReader dr = cmd.ExecuteReader();

        while (dr.Read())
        {
            rptListaProveedor.Add(new Proveedor())
            {
                IdProveedor = Convert.ToInt32(dr["IdProveedor"].ToString()),
                NombreProveedor = dr["NombreProveedor"].ToString(),
                ApellidoProveedor = dr["ApellidoProveedor"].ToString(),
                Telefono = Convert.ToInt32(dr["TelefonoProveedor"].ToString()),
                Activo = Convert.ToBoolean(dr["EstadoProveedor"].ToString()),
                FechaRegistroProveedor = dr["FechaRegistroProveedor"].ToString()
            });
        }
        dr.Close();
    }
}
```

Figura 39: Vista Cliente

Entidades / Cliente

>Listado de Clientes

Crear Nuevo

Copiar Excel PDF CSV Imprimir Buscar:

Cedula	Nombre del Cliente	Dirección	Telefono	Estado	Actions
101-280902-0001B	Bryan Miguel Rosales Galeano	Ciudad sandino	84748499	No Activo	
201-040500-0001F	Freddy Manuel Ruiz Rios	Parque Central	87251122	Activo	
201-070901-0001P	Pepito Rafael Belli Cespedes	Ciudad sandino	82773377	Activo	
201-200900-2009M	Maria Jose	El aguacate, diriomo.	86298921	Activo	
601-240165-0004M	Francisco Ramon Hunter Duran	Contiguo a Escuela Salvador Mendieta	84322243	Activo	
No posee	Cliente no Habitual	No posee	00000000	Activo	

Mostrando registros del 1 al 6 de un total de 6 registros

Anterior 1 Siguiente

Copyright © Jehova Jireh 2025

Figura 40: Vista Proveedor

Entidades / Proveedor

Listado de Proveedores

Crear Nuevo

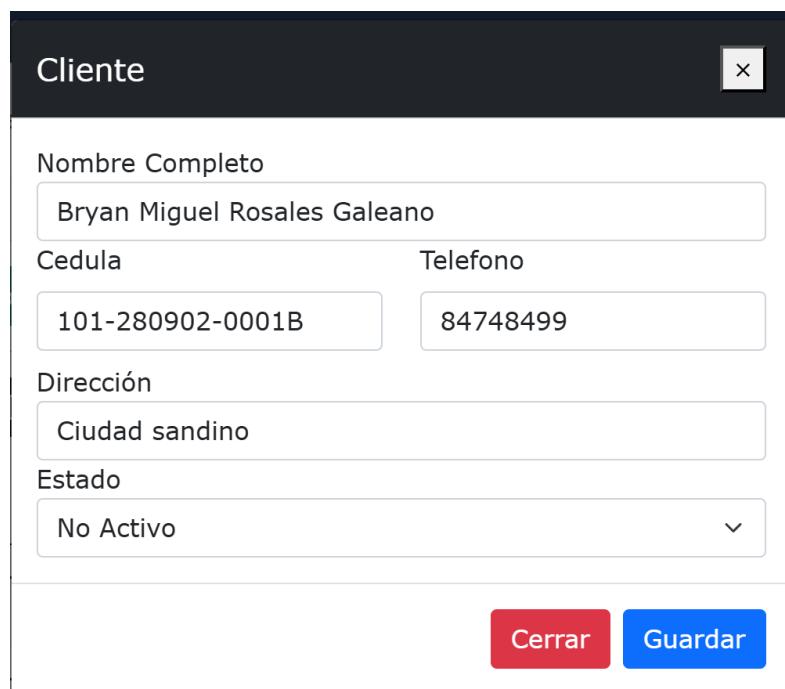
Copiar Excel PDF CSV Imprimir Buscar:

Empresa	Nombre del Proveedor	Telefono	Estado	Actions
Agricorp	Pablo Ernesto Mejía Oviedo	85300358	Activo	
Carne San Martin	Manuel Antonio Gómez Pérez	81928722	Activo	
Coca-Cola	Carlos Fabian Ruiz Hernández	87664213	Activo	

Mostrando registros del 1 al 3 de un total de 3 registros

Anterior 1 Siguiente

Figura 41: Crear cliente



Formulario para crear un cliente. Los campos son:

- Nombre Completo: Bryan Miguel Rosales Galeano
- Cedula: 101-280902-0001B
- Telefono: 84748499
- Direccion: Ciudad sandino
- Estado: No Activo

Botones: Cerrar (rojo) y Guardar (azul).

Figura 42: Crear proveedores



Formulario para crear un proveedor. Los campos son:

- Nombre Completo: Pablo Ernesto Mejía Oviedo
- Empresa: Agricorp
- Telefono: 85300358
- Estado: Activo

Botones: Cerrar (rojo) y Guardar (azul).

Figura 43: Clase Empresa

```
8     ↴ 19 referencias
9
10    ↴ 7 referencias
11    ↴ 3 referencias
12    ↴ 18 referencias
13    ↴ 3 referencias
14    ↴ 1 referencia
15    ↴ }
```

```
public class Empresa
{
    public int IdMarca { get; set; }
    public Proveedor oProveedor { get; set; }
    public string Descripcion { get; set; }
    public bool Activo { get; set; }
    public string FechaRegistroEmpresa { get; set; }
}
```

Figura 44: Lógica Empresa

```
public List<Empresa> Listar(int _idusuario)
{
    List<Empresa> rptListaMarca = new List<Empresa>();
    using (SqlConnection oConexion = new SqlConnection(Conexion.CN))
    {
        SqlCommand cmd = new SqlCommand("sp_obtenerEmpresa", oConexion);
        cmd.CommandType = CommandType.StoredProcedure;

        try
        {
            oConexion.Open();
            SqlDataReader dr = cmd.ExecuteReader();

            while (dr.Read())
            {
                rptListaMarca.Add(new Empresa())
                {
                    IdMarca = Convert.ToInt32(dr["IdEmpresa"].ToString()),
                    oProveedor = new Proveedor() { IdProveedor = Convert.ToInt32(dr["IdProveedor"].ToString()) },
                    Descripcion = dr["DescripcionEmpresa"].ToString(),
                    Activo = Convert.ToBoolean(dr["EstadoEmpresa"].ToString()),
                    FechaRegistroEmpresa = dr["FechaRegistroEmpresa"].ToString()
                });
            }
            dr.Close();
        }
    }
}
```

Figura 45: Controlador Empresa

```
0 referencias
public ActionResult Empresa()
{
    if (Session["Usuario"] == null)
        return RedirectToAction("Index", "Login");

    return View();
}
```

Figura 46: Vista Empresa

Descripción	Estado	
Agricorp	Activo	
Avicasa	Activo	
Big Cola	Activo	
Carne San Martin	Activo	
Coca-Cola	Activo	
Compañía Licorera de Nicaragua S.A	Activo	
Danone	Activo	
Dicegca	Activo	
Dos Robles	Activo	
Florida Ice & Farm Co	Activo	

Mostrando registros del 1 al 10 de un total de 22 registros

Anterior 1 2 3 Siguiente

Figura 47: Crear Empresa

Empresa X

Descripcion

Estado

▼

Cerrar Guardar

6.3.1.2. Validación del Incremento

Al finalizar la iteración, el Incremento fue presentado al cliente para la Sprint Review. Para ver más detalles, véase Figura 98: Acta de reunión sprint 1 del Anexo II.

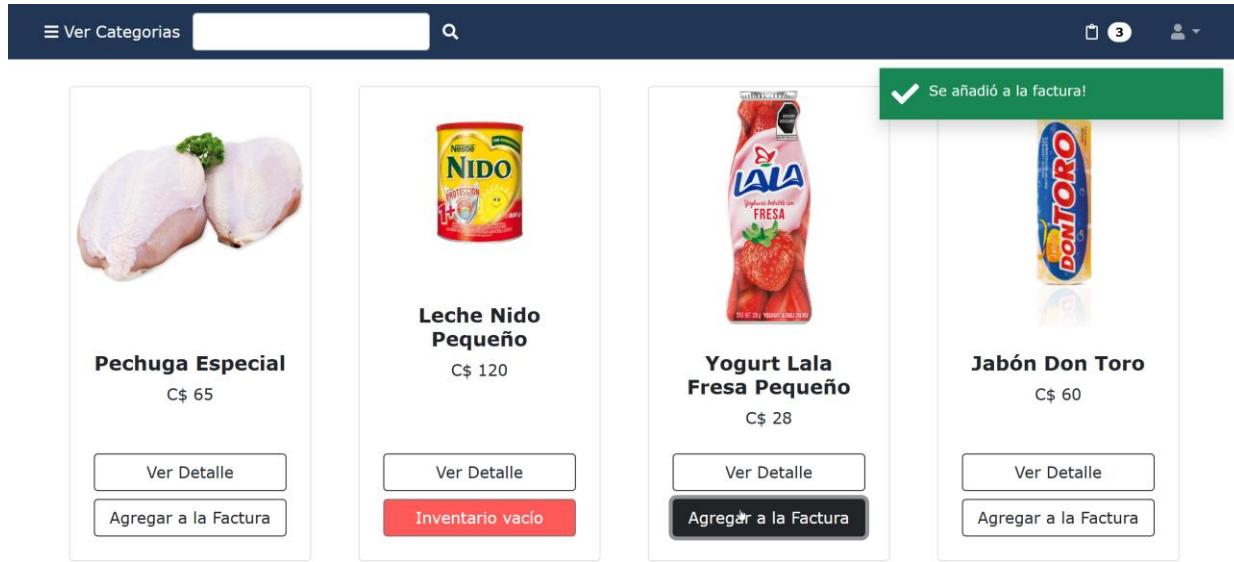
Tabla 9: Resumen del Sprint Review Sprint 1

Sprint Review	Valor entregado
Pruebas de Usuario	Se realizaron pruebas de aceptación por parte de la pulpería, las cuales confirmaron la funcionalidad de Login y el flujo básico de navegación de la aplicación. Esta fase de pruebas validó que el sistema cumple con las expectativas iniciales de acceso seguro, y que las interfaces de los módulos clave están listas para la operación.
Resultado Tangible	Se entregó una aplicación totalmente navegable. Este resultado incluye la totalidad de las pantallas del sistema con su interfaz final, valor funcional de la seguridad y los datos maestros.
Preparación	El Sprint concluyó con la cimentación técnica del proyecto, que incluye el establecimiento de la base de control de versiones y la configuración del entorno de desarrollo. Esta preparación crítica aseguró que el equipo estaba listo para la implementación eficiente de las funcionalidades de Venta y Producto en el siguiente ciclo.

6.3.2. Sprint 2

El Sprint 2 se enfocó en dar vida al esqueleto funcional creado en el ciclo anterior. El objetivo principal fue habilitar la lógica completa y la persistencia de datos para la totalidad de los módulos, transformando las interfaces de navegación en un sistema de gestión de datos eficaz (Véase Anexo IX).

Figura 48: Catalogo de Productos



6.3.2.1. Detalle del Sprint Backlog

El trabajo se concentró en la implementación de la lógica CRUD para las demás entidades del negocio.

6.3.2.1.1. Gestión del inventario

En el área de gestión del inventario, se implementó el ciclo de vida completo de los datos para los módulos de Producto, Categoría y Unidad de Medida. Estas tareas iban desde la creación de la tabla principal hasta las pruebas para garantizar la correcta funcionalidad.

Tabla 10: Sprint Backlog para la gestión de inventario

ORIGEN	ROL	ASIGNADO	ACCION	ESTADO	PRIORIDAD	HORAS
PRODUCTO	Developer	Cristhiam Omar Suarez Nuñez	Creación de [dbo].[Producto]	Completada	Alta	5.00
PRODUCTO	Developer	Cristhiam Omar Suarez Nuñez	Creación de SPS para Producto	Completada	Media	2.00
PRODUCTO	Developer	Cristhiam Omar Suarez Nuñez	Creación de clases: Producto	Completada	Media	1.00
PRODUCTO	Developer	Cristhiam Omar Suarez Nuñez	Creación de lógica: Producto	Completada	Media	4.00
PRODUCTO	Developer	Cristhiam Omar Suarez Nuñez	Creación de controlador: Producto	Completada	Media	2.00
PRODUCTO	Developer	Cristhiam Omar Suarez Nuñez	Creación de vista: Producto	Completada	Media	3.00
PRODUCTO	Qua Tester	Cristhiam Omar Suarez Nuñez	Pruebas unitarias de Producto	Completada	Media	2.00
CATEGORIA	Developer	Hosmand Rafael Belli Cespedes	Creación de [dbo].[Categoria]	Completada	Alta	3.00
CATEGORIA	Developer	Hosmand Rafael Belli Cespedes	Creación de SPS para Categoria	Completada	Alta	3.00
CATEGORIA	Developer	Hosmand Rafael Belli Cespedes	Creación de clases: Categoria	Completada	Media	1.00
CATEGORIA	Developer	Hosmand Rafael Belli Cespedes	Creación de lógica: Categoria	Completada	Media	2.00
CATEGORIA	Developer	Hosmand Rafael Belli Cespedes	Creación de controlador: Categoria	Completada	Media	2.00
CATEGORIA	Developer	Hosmand Rafael Belli Cespedes	Creación de vista: Categoria	Completada	Media	2.00
CATEGORIA	Qua Tester	Hosmand Rafael Belli Cespedes	Pruebas unitarias de Categoria	Completada	Media	2.00
UDM	Developer	Hosmand Rafael Belli Cespedes	Creación de [dbo].[UnidadMedida]	Completada	Media	3.00
UDM	Developer	Hosmand Rafael Belli Cespedes	Creación de SPS para Unidad de Medida	Completada	Alta	3.00
UDM	Developer	Hosmand Rafael Belli Cespedes	Creación de clases: Unidad de Medida	Completada	Alta	3.00
UDM	Developer	Hosmand Rafael Belli Cespedes	Creación de lógica: Unidad de Medida	Completada	Media	1.00
UDM	Developer	Hosmand Rafael Belli Cespedes	Creación de controlador: Unidad de Medida	Completada	Media	2.00
UDM	Developer	Hosmand Rafael Belli Cespedes	Creación de vista: Unidad de Medida	Completada	Media	2.00
UDM	Qua Tester	Hosmand Rafael Belli Cespedes	Pruebas unitarias de Unidad de Medida	Completada	Media	2.00

Figura 49: Tablas de gestión de inventario

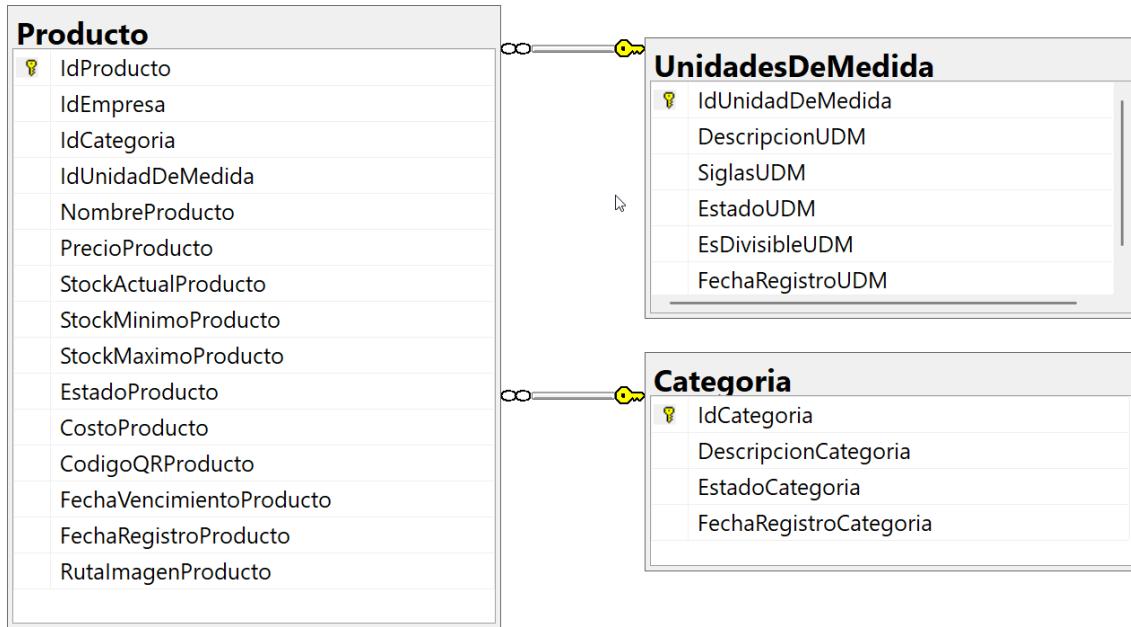


Figura 50: Procedimiento almacenado Producto

```
-- Registrar detalles con toda la información relevante
IF @IdModulo IS NOT NULL AND @IdAccion IS NOT NULL
BEGIN
    DECLARE @Estado VARCHAR(20) = CASE WHEN @Activo = 1 THEN 'Activo' ELSE 'No Activo' END;
    DECLARE @Detalle NVARCHAR(MAX) =
        'Producto registrado: ' +
        'Nombre = ' + @Nombre + ', ' +
        'Nombre de Empresa = ' + @NombreEmpresa + ', ' +
        'Nombre de Categoria = ' + @NombreCategoria + ', ' +
        'FechaVencimiento = ' + CONVERT(VARCHAR, @FechaVencimiento, 120) + ', ' +
        'Costo = ' + CAST(@Costo AS VARCHAR) + ', ' +
        'Precio = ' + CAST(@Precio AS VARCHAR) + ', ' +
        'Stock = ' + CAST(@Stock AS VARCHAR) + ', ' +
        'StockMinimo = ' + CAST(@StockMinimo AS VARCHAR) + ', ' +
        'StockMaximo = ' + CAST(@StockMaximo AS VARCHAR) + ', ' +
        'CodigoQR = ' + @CodigoQR + ', ' +
        'UnidadMedida = ' + CAST(@IdUnidadDeMedida AS VARCHAR) + ', ' +
        'RutaImagen = ' + @Ruta +
        'Estado = ' + @Estado;

    INSERT INTO RegActividad (IdUsuario, IdRolModulo, IdRolAccion, DetallesRegActividad)
    VALUES (@IdUsuario, @IdModulo, @IdAccion, @Detalle);
END
END
ND
```

Figura 51: Procedimiento almacenado Categoría

```

| BEGIN
|   SET NOCOUNT ON;
|   SET @Resultado = 0;

| IF NOT EXISTS (SELECT 1 FROM Categoria WHERE DescripcionCategoria = @Descripcion)
| BEGIN
|   -- Insertar categoría
|   INSERT INTO Categoria (DescripcionCategoria, EstadoCategoria)
|   VALUES (@Descripcion, @Activo);

|   -- Obtener IdModulo e IdAccion correspondientes
|   DECLARE @IdModulo INT = (SELECT TOP 1 IdRolModulo FROM RolModulo
|   WHERE NombreRolModulo = 'Categorias');
|   DECLARE @IdAccion INT = (SELECT TOP 1 IdRolAccion FROM RolAccion
|   WHERE NombreRolAccion = 'Crear');

|   -- Insertar en UsuarioActividad
|   INSERT INTO RegActividad (IdUsuario, IdRolModulo, IdRolAccion, DetallesRegActividad)
|   VALUES (
|     @IdUsuario,
|     @IdModulo,
|     @IdAccion,
|     'Se ha registrado una nueva categoría con la Descripción: ' + @Descripcion +
|     ', con estado: ' + CASE WHEN @Activo = 1 THEN 'Activo' ELSE 'No Activo' END
|   );

```

Figura 52: Procedimiento almacenado Unidad de Medida

```

| BEGIN
|   SET @Resultado = 1

| IF NOT EXISTS (SELECT 1 FROM UnidadesDeMedida WHERE DescripcionUDM = @Descripcion)
| BEGIN
|   INSERT INTO UnidadesDeMedida(DescripcionUDM, SiglasUDM, EstadoUDM, EsDivisibleUDM)
|   VALUES (@Descripcion, @Siglas, @Activo, @Divisible)

|   -- Auditoría
|   DECLARE @IdModulo INT = (SELECT IdRolModulo FROM RolModulo
|   WHERE NombreRolModulo = 'Unidades de Medida')
|   DECLARE @IdAccion INT = (SELECT IdRolAccion FROM RolAccion
|   WHERE NombreRolAccion = 'Crear')
|   DECLARE @Estado NVARCHAR(20) = CASE WHEN @Activo = 1
|   THEN 'Activo' ELSE 'No Activo' END
|   DECLARE @DivisibleTxt NVARCHAR(20) = CASE WHEN @Divisible = 1 THEN 'Sí' ELSE 'No' END

|   INSERT INTO RegActividad (IdUsuario, IdRolModulo, IdRolAccion, DetallesRegActividad)
|   VALUES (
|     @IdUsuario, @IdModulo, @IdAccion,
|     'Se registró una unidad de medida: "' + @Descripcion +
|     '" con estado: ' + @Estado + ', divisible: ' + @DivisibleTxt
|   )
| END

```

Figura 53: Clase Categoría y Unidad de Medida

```
    <namespace> ProjetoTest.Models
    {
        <class> Categoria
        {
            9 referencias
            public int IdCategoria { get; set; }

            5 referencias
            public string Descripcion { get; set; }

            3 referencias
            public bool Activo { get; set; }
        }
    }

    <namespace> ProjetoTest.Models
    {
        <class> UnidadDeMedida
        {
            6 referencias
            public int IdUnidadDeMedida { get; set; }

            10 referencias
            public string Descripcion { get; set; }

            3 referencias
            public string Siglas { get; set; }

            3 referencias
            public bool Activo { get; set; }

            3 referencias
            public bool Divisible { get; set; }
        }
    }
```

Figura 54: Clase Producto

```
    8
    9
    10
    11
    12
    13
    14
    15
    16
    17
    18
    19
    20
    21
    22
    23
    24
    25
    26
    27
    28

    <class> Producto
    {
        59 referencias
        public int IdProducto { get; set; }

        35 referencias
        public string Nombre { get; set; }

        0 referencias
        public string Descripcion { get; set; }

        0 referencias
        public Empresa oEmpresa { get; set; }

        17 referencias
        public Categoria oCategoria { get; set; }

        19 referencias
        public UnidadDeMedida oUnidadDeMedida { get; set; }

        21 referencias
        public string FechaVencimiento { get; set; }

        21 referencias
        public float Precio { get; set; }

        8 referencias
        public float Costo { get; set; }

        21 referencias
        public float Stock { get; set; }

        56 referencias
        public string RutaImagen { get; set; }

        14 referencias
        public bool Activo { get; set; }

        6 referencias
        public bool Divisible { get; set; }

        11 referencias
        public string base64 { get; set; }

        11 referencias
        public string extension { get; set; }

        16 referencias
        public string CodigoQR { get; set; }

        16 referencias
        public float StockMaximo { get; set; }

        16 referencias
        public float StockMinimo { get; set; }
    }
```

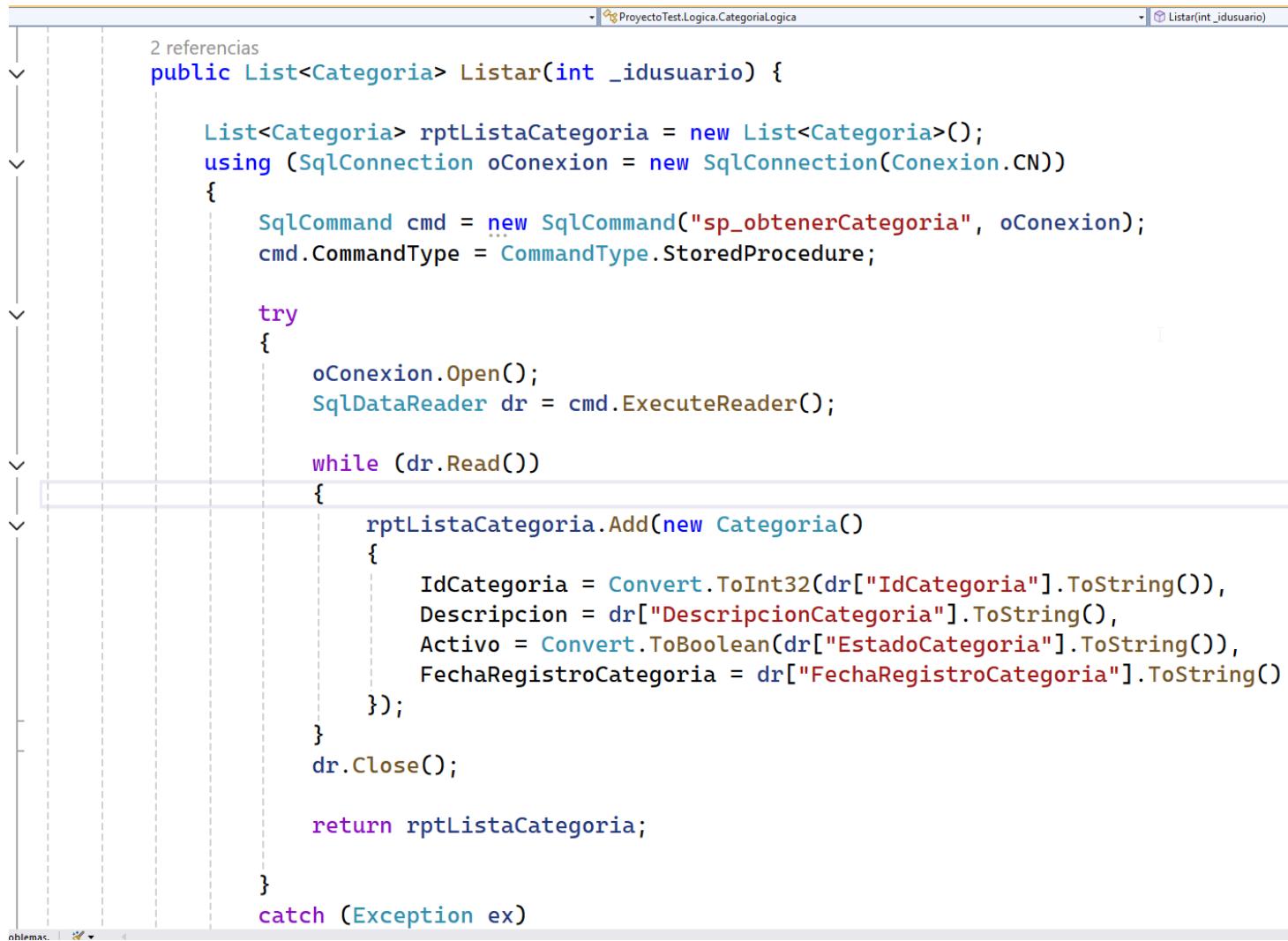
Figura 55: Lógica Producto

```
public List<Producto> Listar(int _idusuario)
{
    List<Producto> rptListaProducto = new List<Producto>();
    using (SqlConnection oConexion = new SqlConnection(Conexion.CN))
    {
        SqlCommand cmd = new SqlCommand("sp_obtenerProducto", oConexion);
        cmd.CommandType = CommandType.StoredProcedure;

        try
        {
            oConexion.Open();
            SqlDataReader dr = cmd.ExecuteReader();

            while (dr.Read())
            {
                rptListaProducto.Add(new Producto())
                {
                    IdProducto = Convert.ToInt32(dr["IdProducto"].ToString()),
                    Nombre = dr["NombreProducto"].ToString(),
                    oMarca = new Empresa() { IdMarca = Convert.ToInt32(dr["IdEmpresa"].ToString()), Descripcion = dr["DescripcionEmpresa"].ToString() },
                    oCategoria = new Categoria() { IdCategoria = Convert.ToInt32(dr["IdCategoria"].ToString()), Descripcion = dr["DescripcionCategoria"].ToString() },
                    oUnidadDeMedida = new UnidadDeMedida() { IdUnidadDeMedida = Convert.ToInt32(dr["IdUnidadDeMedida"].ToString()), Descripcion = dr["DescripcionUnidadDeMedida"].ToString() },
                    FechaVencimiento = dr["FechaVencimientoProducto"].ToString(),
                    Precio = float.Parse(dr["PrecioProducto"].ToString(), new CultureInfo("es-NI")),
                    Costo = float.Parse(dr["CostoProducto"].ToString(), new CultureInfo("es-NI")),
                    Stock = float.Parse(dr["StockActualProducto"].ToString()),
                    RutaImagen = dr["RutaImagenProducto"].ToString(),
                    StockMinimo = float.Parse(dr["StockMinimoProducto"].ToString()),
                    StockMaximo = float.Parse(dr["StockMaximoProducto"].ToString()),
                    Activo = Convert.ToBoolean(dr["EstadoProducto"].ToString()),
                    CodigoQR = dr["CodigoQRProducto"].ToString(),
                    FechaRegistroProducto = dr["FechaRegistroProducto"].ToString()
                });
            }
        }
    }
}
```

Figura 56: Lógica Categoría



```
2 referencias
public List<Categoria> Listar(int _idusuario) {
    List<Categoria> rptListaCategoria = new List<Categoria>();
    using (SqlConnection oConexion = new SqlConnection(Conexion.CN))
    {
        SqlCommand cmd = new SqlCommand("sp_obtenerCategoria", oConexion);
        cmd.CommandType = CommandType.StoredProcedure;

        try
        {
            oConexion.Open();
            SqlDataReader dr = cmd.ExecuteReader();

            while (dr.Read())
            {
                rptListaCategoria.Add(new Categoria()
                {
                    IdCategoria = Convert.ToInt32(dr["IdCategoria"].ToString()),
                    Descripcion = dr["DescripcionCategoria"].ToString(),
                    Activo = Convert.ToBoolean(dr["EstadoCategoria"].ToString()),
                    FechaRegistroCategoria = dr["FechaRegistroCategoria"].ToString()
                });
            }
            dr.Close();
        }
        return rptListaCategoria;
    }
    catch (Exception ex)
}
```

Figura 57: Lógica Unidad de Medida

```
1 referencia
- ProyectoTest.Logica.UnidadDeMedidaLogica
- instancia

public List<UnidadDeMedida> Listar(int _idusuario)
{
    List<UnidadDeMedida> rptListaUnidadDeMedida = new List<UnidadDeMedida>();
    using (SqlConnection oConexion = new SqlConnection(Cexion.CN))
    {
        SqlCommand cmd = new SqlCommand("sp_obtenerUnidadDeMedida", oConexion);
        cmd.CommandType = CommandType.StoredProcedure;

        try
        {
            oConexion.Open();
            SqlDataReader dr = cmd.ExecuteReader();

            while (dr.Read())
            {
                rptListaUnidadDeMedida.Add(new UnidadDeMedida()
                {
                    IdUnidadDeMedida = Convert.ToInt32(dr["IdUnidadDeMedida"].ToString()),
                    Descripcion = dr["DescripcionUDM"].ToString(),
                    Siglas = dr["SiglasUDM"].ToString(),
                    Activo = Convert.ToBoolean(dr["EstadoUDM"].ToString()),
                    Divisible = Convert.ToBoolean(dr["EsDivisibleUDM"].ToString()),
                    FechaRegistroUDM = dr["FechaRegistroUDM"].ToString()
                });
            }
            dr.Close();
        }
        return rptListaUnidadDeMedida;
    }
}
```

Figura 58: Controlador Producto

```
0 referencias
public ActionResult Producto()
{
    if (Session["Usuario"] == null)
        return RedirectToAction("Index", "Login");

    return View();
}
```

Figura 59: Controlador Categoría

```
0 referencias
public ActionResult Categoria()
{
    if (Session["Usuario"] == null)
        return RedirectToAction("Index", "Login");

    return View();
}
```

Figura 60: Controlador Unidad de Medida

```
public ActionResult UnidadDeMedida()
{
    if (Session["Usuario"] == null)
        return RedirectToAction("Index", "Login");

    return View();
}
```

Figura 61: Vista Producto

[Artículos](#) / Productos

Listado de Productos

Crear Nuevo

[Copiar](#) [Excel](#) [PDF](#) [CSV](#) [Imprimir](#) Buscar:

Nombre	Descripción	Empresa	Categoría	Fecha de Caducidad	Costo	Precio	Cantidad	Estado
Agua Cielo 500ml	Agua embotellada	Big Cola	Agua embotellada	19/01/2024 00:00:00	15	20	15	No Activo
Arroz Faisan 400g	Arroz en bolsa	Agricorp	Granos y cereales	03/03/2024 00:00:00	17	20	29	Activo
Carne Molida Super	Libra	Carne San Martin	Carnes y aves	02/06/2023 00:00:00	85	105	19	Activo
Chicharrones Pequeños 100 gr	Unidad	Yummies	Bocadillos y alimentos procesados	23/09/2023 00:00:00	10	13	15	Activo
Cloro Magia Blanca	Bolsa	PRONISA	Productos de limpieza y utensilios de cocina	30/09/2023 00:00:00	13	18	23	Activo

Figura 62: Vista Categoría

[Artículos](#) / Categorías

Listado de Categorías

Crear Nuevo

[Copiar](#) [Excel](#) [PDF](#) [CSV](#) [Imprimir](#) Buscar:

Descripción	Estado
Aceites y condimentos	Activo
Agua embotellada	Activo
Alimento para mascotas	Activo
Alimentos congelados	Activo
Alimentos para niños y bebés	Activo
Bebidas	Activo
Bebidas Alcohólicas	Activo
Bebidas energéticas y deportivas	Activo
Bocadillos y alimentos procesados	Activo

Figura 63: Vista Unidad de Medida

Listado de Unidades De Medida			
Crear Nuevo			
Copiar Excel PDF CSV Imprimir			
Descripción	Siglas	Divisible	Estado
Bolsa	Bls	Si	Activo
Botella	Btl	No	Activo
Caja	Cj	No	Activo

Showing 1 to 3 of 3 entries

Figura 64: Crear Producto

Producto x



Nombre

Descripcion

Elegir archivo

No se eligió ...gún archivo

Marca

Categoría

Fecha de Vencimiento

 🕒

Costo

Precio

Cantidad

Estado

Figura 65:Crear Unidad de Medida



Unidades De Medida

Descripcion
Bolsa

Divisible
No

Estado
Activo

Cerrar Guardar

Este formulario es para crear una nueva unidad de medida. Los campos incluyen: Descripción (Bolsa), Divisible (No) y Estado (Activo). Los botones para acción son Cerrar (en rojo) y Guardar (en azul).

Figura 66: Crear Categoría



Categoria

Descripcion
Aceites y condimentos

Estado
Activo

Cerrar Guardar

Este formulario es para crear una nueva categoría. Los campos incluyen: Descripción (Aceites y condimentos) y Estado (Activo). Los botones para acción son Cerrar (en rojo) y Guardar (en azul).

6.3.2.1.2. Módulos transaccionales

En segundo lugar, se desarrolló el núcleo transaccional al construir el módulo de Compras para gestionar la entrada de mercancía, y el módulo de Ventas para generar facturas y registrar la salida del inventario, creando un sistema de inventario operativo de entrada y salida.

Tabla 11: Sprint Backlog para la gestión de Ventas y Compras

ORIGEN	ROL	ASIGNADO	ACCION	ESTADO	PRIORIDAD	HORAS
VENTAS	Developer	Hosmand Rafael Belli Cespedes	Creación de [dbo].[Venta]	Completada	Alta	4.00
VENTAS	Developer	Hosmand Rafael Belli Cespedes	Creación de [dbo].[VentaDetalle]	Completada	Alta	5.00
VENTAS	Developer	Hosmand Rafael Belli Cespedes	Creación de SPS para Venta	Completada	Alta	6.00
VENTAS	Developer	Cristhiam Omar Suarez Nuñez	Creación de Clases: Venta	Completada	Media	1.00
VENTAS	Developer	Cristhiam Omar Suarez Nuñez	Creación de Lógica: Registrar Venta	Completada	Alta	8.00
VENTAS	Developer	Cristhiam Omar Suarez Nuñez	Creación de Controlador: Venta	Completada	Media	2.00
VENTAS	Developer	Cristhiam Omar Suarez Nuñez	Creación de Vista: Venta	Completada	Media	5.00
VENTAS	Qua Tester	Cristhiam Omar Suarez Nuñez	Pruebas unitarias de Venta	Completada	Media	4.00
COMPRAS	Developer	Hosmand Rafael Belli Cespedes	Creación de [dbo].[Compra]	Completada	Alta	4.00
COMPRAS	Developer	Hosmand Rafael Belli Cespedes	Creación de [dbo].[CompraDetalle]	Completada	Alta	5.00
COMPRAS	Developer	Cristhiam Omar Suarez Nuñez	Creación de SPS para Compra	Completada	Alta	6.00
COMPRAS	Developer	Cristhiam Omar Suarez Nuñez	Creación de Clases: Compra	Completada	Media	1.00
COMPRAS	Developer	Cristhiam Omar Suarez Nuñez	Creación de Lógica: Registrar Compra	Completada	Alta	8.00
COMPRAS	Developer	Cristhiam Omar Suarez Nuñez	Creación de Controlador: Compra	Completada	Media	2.00
COMPRAS	Developer	Cristhiam Omar Suarez Nuñez	Creación de Vista: Compra	Completada	Media	5.00
COMPRAS	Qua Tester	Hosmand Rafael Belli Cespedes	Pruebas unitarias de Compra	Completada	Media	4.00

Figura 67: Tablas de proceso de Ventas

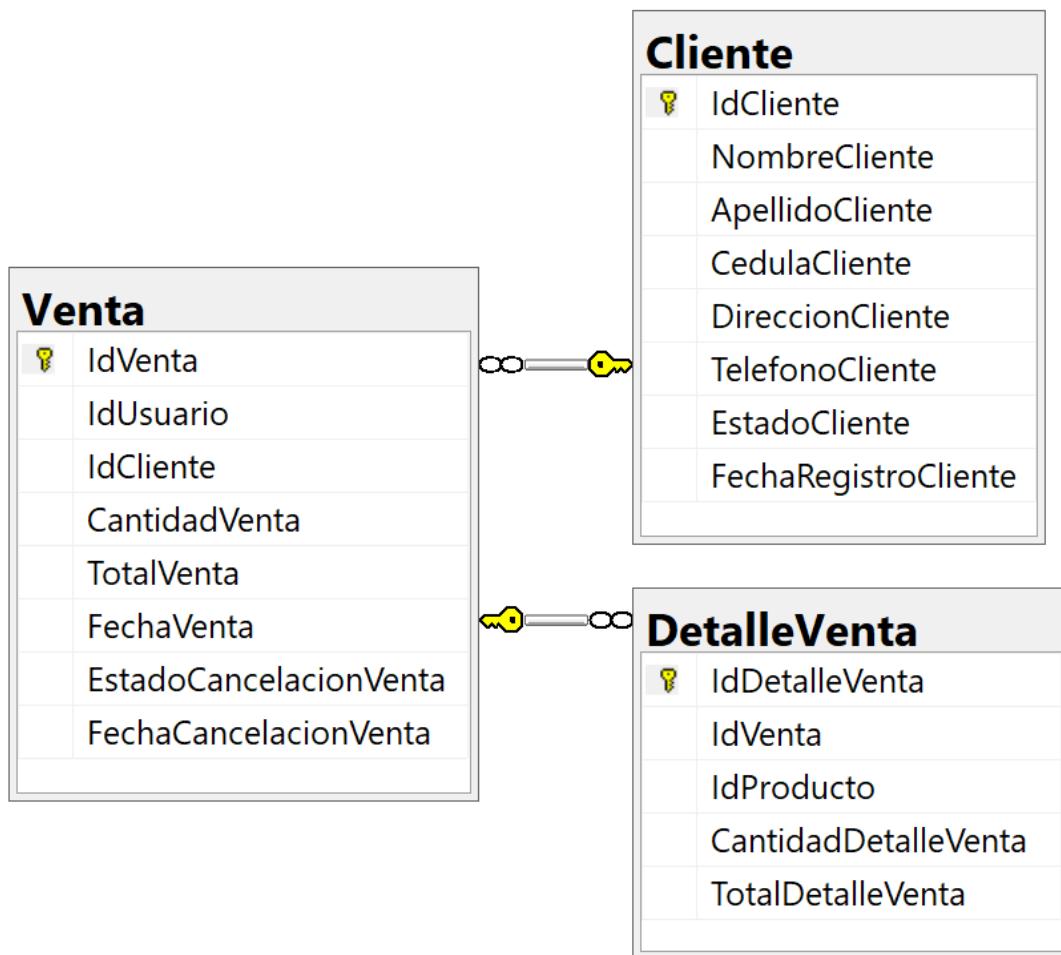


Figura 68:Procedimiento almacenado de Venta

```

BEGIN
    BEGIN TRY
        SET @Resultado = 0;
        BEGIN TRANSACTION;

        DECLARE @IdCompra INT = 0;
        INSERT INTO COMPRA (IdUsuario, CantidadCompra, TotalCompra, IdProveedor)
        VALUES (@IdUsuario, @TotalProducto, @Total, @IdProveedor);

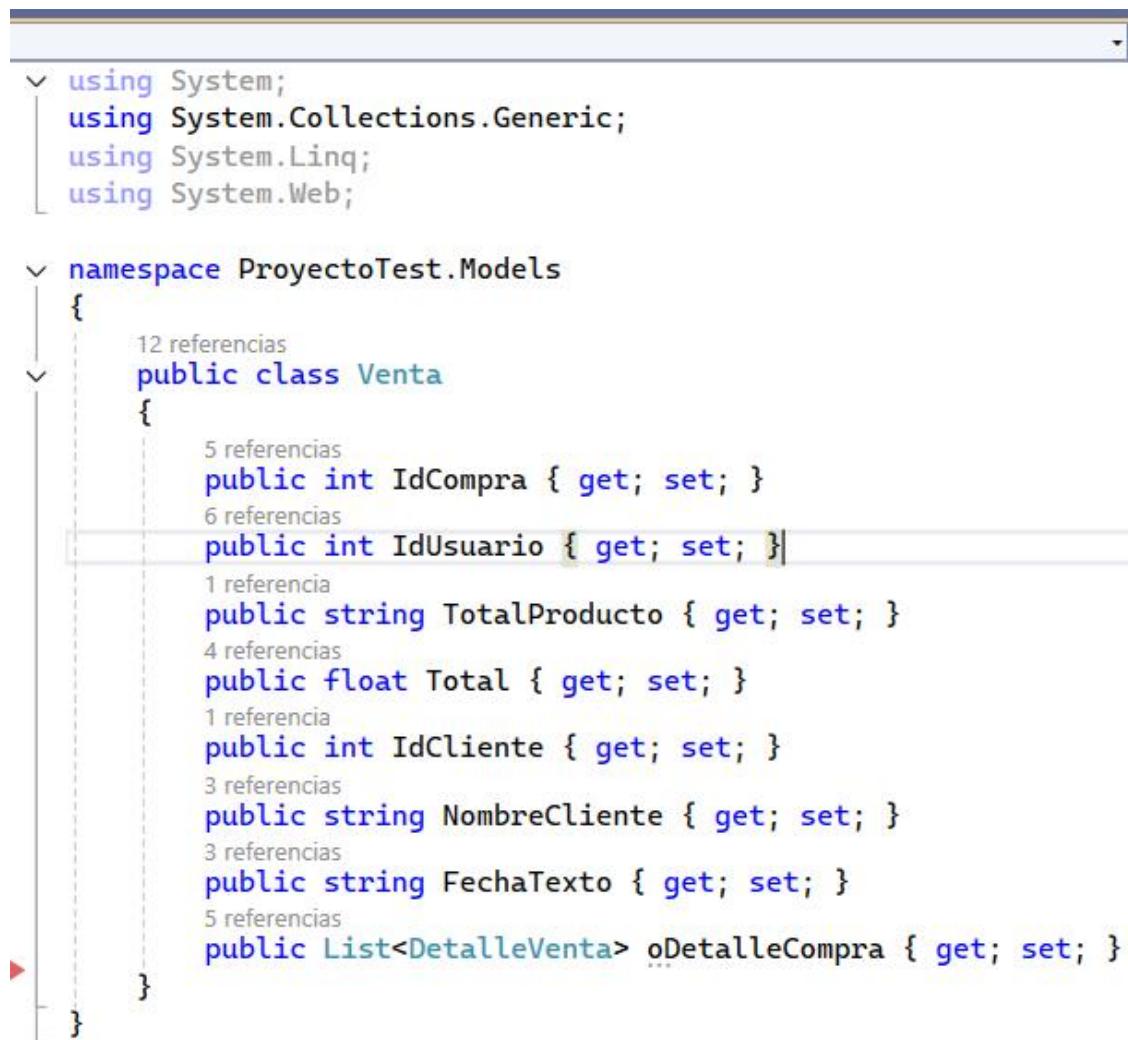
        SET @IdCompra = SCOPE_IDENTITY();

        SET @QueryDetalleCompraProveedor = REPLACE(@QueryDetalleCompraProveedor, '!idcompra!', @IdCompra)
        EXECUTE sp_executesql @QueryDetalleCompraProveedor;

        DELETE FROM CarritoCompra WHERE IdUsuario = @IdUsuario;

        -- Registrar en UsuarioActividad
        DECLARE @IdModulo INT = (SELECT IdRolModulo FROM RolModulo WHERE NombreRolModulo = 'Compras');
        DECLARE @IdAccion INT = (SELECT IdRolAccion FROM RolAccion WHERE NombreRolAccion = 'Consultar');
        DECLARE @NombreProveedor NVARCHAR(100) = (SELECT (NombreProveedor + ' ' + ApellidoProveedor)
        AS NombreProveedor FROM Proveedor WHERE IdProveedor = @IdProveedor);
    END TRY
    BEGIN CATCH
        SET @Resultado = 1;
    END CATCH
END
    
```

Figura 69: Clase Venta



```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;

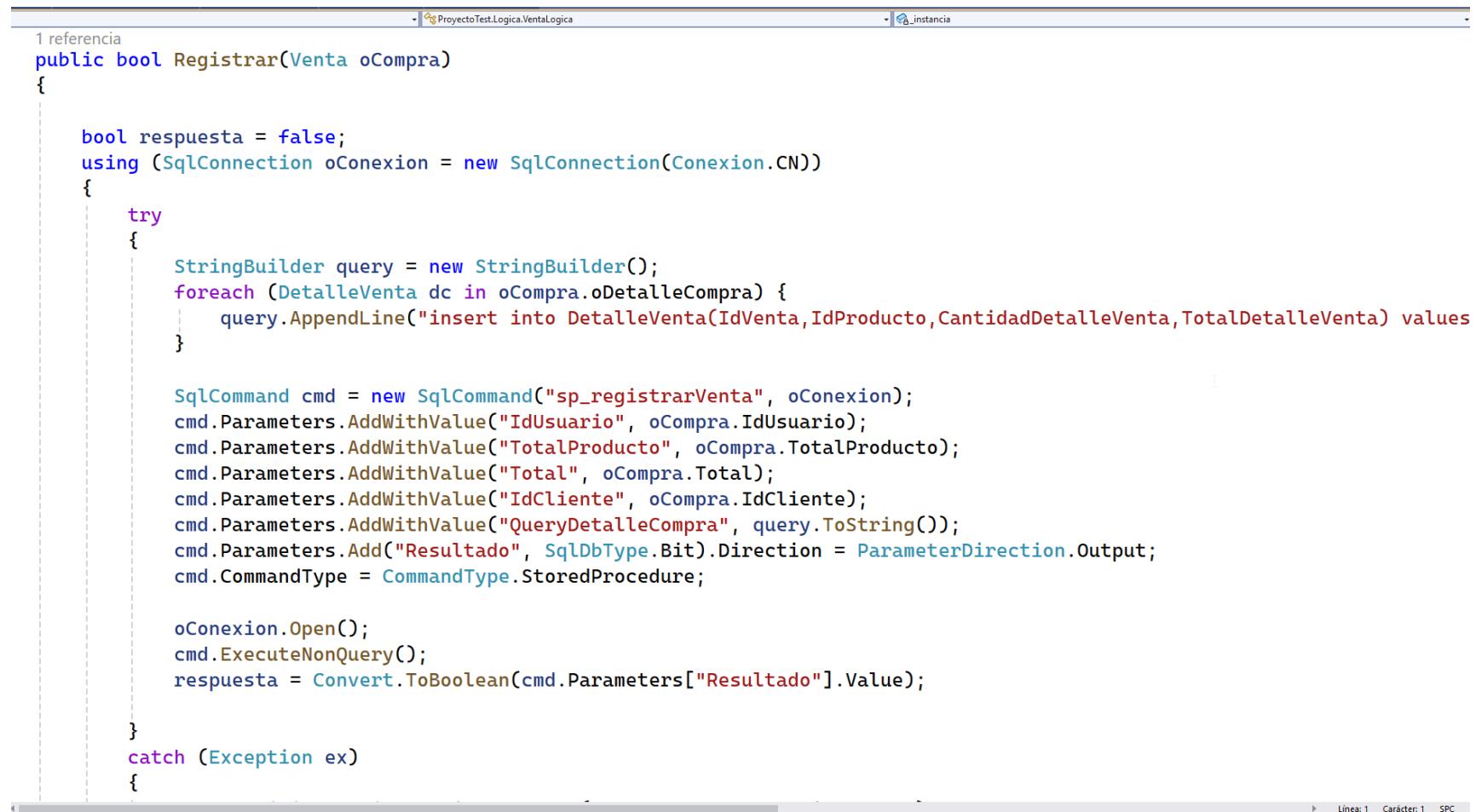
namespace ProyectoTest.Models
{
    public class Venta
    {
        public int IdCompra { get; set; }
        public int IdUsuario { get; set; }
        public string TotalProducto { get; set; }
        public float Total { get; set; }
        public int IdCliente { get; set; }
        public string NombreCliente { get; set; }
        public string FechaTexto { get; set; }
        public List<DetalleVenta> oDetalleCompra { get; set; }
    }
}
```

Figura 70: Controlador Venta

```
public ActionResult Venta()
{
    if (Session["Usuario"] == null)
        return RedirectToAction("Index", "Login");
    else
        oUsuario = (Usuario)Session["Usuario"];

    return View();
}
```

Figura 71: Lógica Venta



```
1 referencia
public bool Registrar(Venta oCompra)
{
    bool respuesta = false;
    using (SqlConnection oConexion = new SqlConnection(Conexion.CN))
    {
        try
        {
            StringBuilder query = new StringBuilder();
            foreach (DetalleVenta dc in oCompra.oDetalleCompra)
            {
                query.AppendLine("insert into DetalleVenta(IdVenta,IdProducto,CantidadDetalleVenta,TotalDetalleVenta) values");
            }

            SqlCommand cmd = new SqlCommand("sp_RegistrarVenta", oConexion);
            cmd.Parameters.AddWithValue("IdUsuario", oCompra.IdUsuario);
            cmd.Parameters.AddWithValue("TotalProducto", oCompra.TotalProducto);
            cmd.Parameters.AddWithValue("Total", oCompra.Total);
            cmd.Parameters.AddWithValue("IdCliente", oCompra.IdCliente);
            cmd.Parameters.AddWithValue("QueryDetalleCompra", query.ToString());
            cmd.Parameters.Add("Resultado", SqlDbType.Bit).Direction = ParameterDirection.Output;
            cmd.CommandType = CommandType.StoredProcedure;

            oConexion.Open();
            cmd.ExecuteNonQuery();
            respuesta = Convert.ToBoolean(cmd.Parameters["Resultado"].Value);
        }
        catch (Exception ex)
        {
    }
```

Figura 72: Vista Detalle de Venta

Detalle de la Venta

Tienes 2 producto(s) en la factura

Agricorp
Arroz Faisan 400g
Cantidad disponible : 29
Precio : C\$20

Lala
Leche Eskimo Fresa 400ml
Cantidad disponible : 17
Precio : C\$25

Volver

Detalle de Pago

Dólares Córdobas

0 0

Efectivo C\$ 0

Total a Pagar C\$ 45

Cambio C\$ 45

Procesar Pago

El efectivo que se ha ingresado no es suficiente para cancelar la factura

Figura 73: Tablas Proceso de Compra

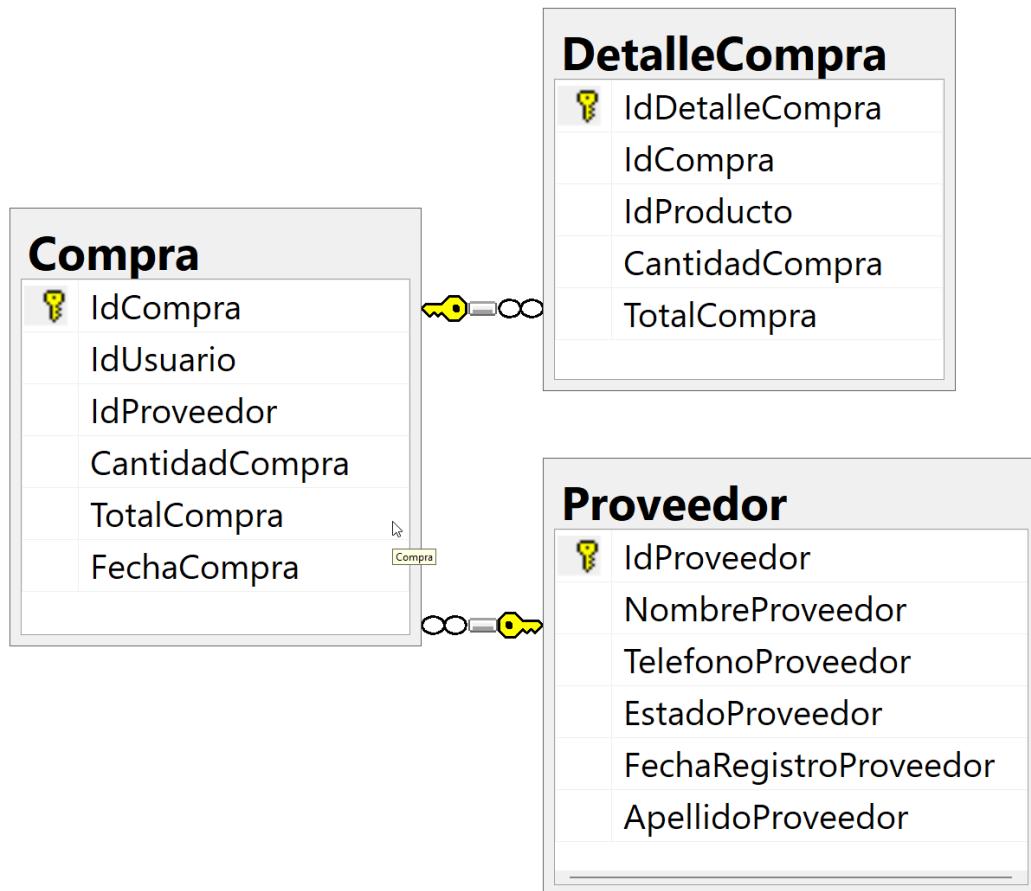


Figura 74: Procedimiento almacenado Compra

```
BEGIN
    BEGIN TRY
        SET @Resultado = 0;
        BEGIN TRANSACTION;

        DECLARE @IdCompra INT = 0;
        INSERT INTO COMPRA (IdUsuario, CantidadCompra, TotalCompra, IdProveedor)
        VALUES (@IdUsuario, @TotalProducto, @Total, @IdProveedor);

        SET @IdCompra = SCOPE_IDENTITY();

        SET @QueryDetalleCompraProveedor = REPLACE(@QueryDetalleCompraProveedor, 'idcompra!', @IdCompra)
        EXECUTE sp_executesql @QueryDetalleCompraProveedor;

        DELETE FROM CarritoCompra WHERE IdUsuario = @IdUsuario;

        -- Registrar en UsuarioActividad
        DECLARE @IdModulo INT = (SELECT IdRolModulo FROM RolModulo WHERE NombreRolModulo = 'Compras');
        DECLARE @IdAccion INT = (SELECT IdRolAccion FROM RolAccion WHERE NombreRolAccion = 'Consultar');
        DECLARE @NombreProveedor NVARCHAR(100) = (SELECT (NombreProveedor + ' ' + ApellidoProveedor)
        AS NombreProveedor FROM Proveedor WHERE IdProveedor = @IdProveedor);
```

Figura 75: Clase Compra

```
10 referencias
public class Compra
{
    3 referencias
    public string NombreEmpresa { get; set; }
    0 referencias
    public int IdCompraProveedor { get; set; }
    3 referencias
    public int IdUsuario { get; set; }
    1 referencia
    public string TotalProducto { get; set; }
    4 referencias
    public float Total { get; set; }
    1 referencia
    public int IdProveedor { get; set; }
    3 referencias
    public string FechaTexto { get; set; }
    4 referencias
    public List<DetalleCompra> oDetalleCompraProveedor { get; set; }

}
```

Figura 76: Lógica Compra

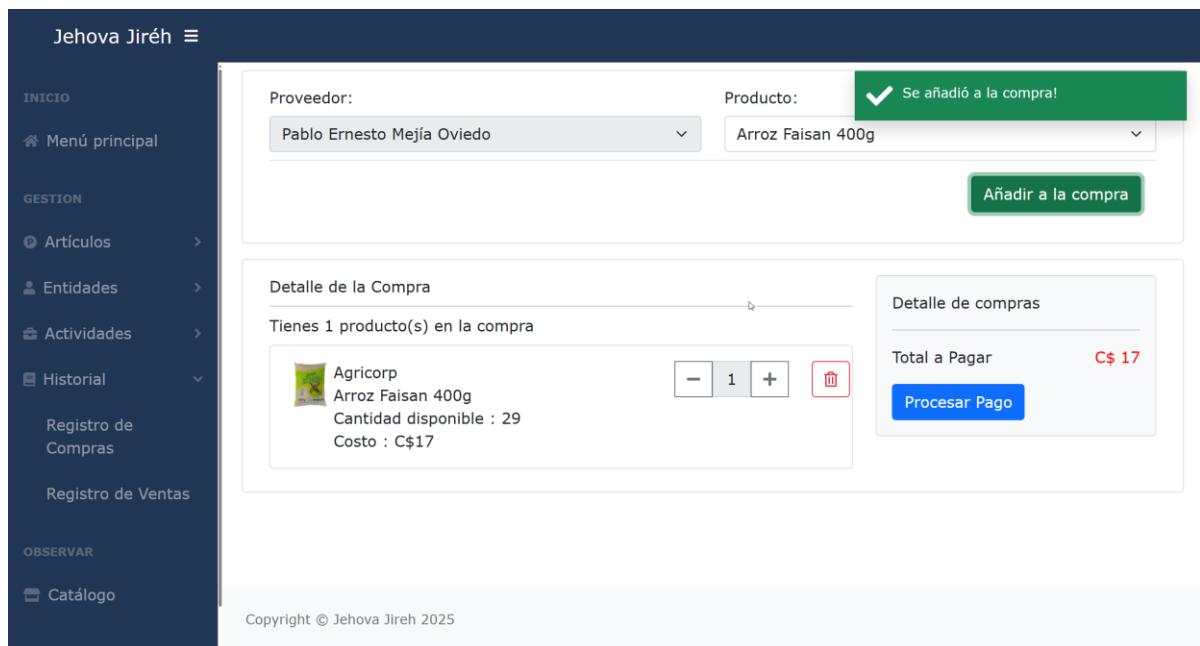
```
public bool Registrar(Compra oCompraProveedor)
{
    bool respuesta = false;
    using (SqlConnection oConexion = new SqlConnection(Conexion.CN))
    {
        try
        {
            StringBuilder query = new StringBuilder();
            foreach (DetalleCompra dc in oCompraProveedor.oDetalleCompraProveedor)
            {
                query.AppendLine("INSERT INTO DetalleCompra(IdCompra,IdProducto,CantidadCompra,TotalCompra) VALUES (" + dc.IdCompra + "," + dc.IdProducto + "," + dc.CantidadCompra + "," + dc.TotalCompra + ")");
                query.AppendLine("UPDATE Producto SET " +
                    "CostoProducto = " + dc.Costo.ToString().Replace(",",".") + "," +
                    "PrecioProducto = " + dc.Precio.ToString().Replace(",",".") + "," +
                    "FechaVencimientoProducto = '" + dc.FechaVencimiento + "' " +
                    "WHERE IdProducto = " + dc.IdProducto + ";");
            }
            SqlCommand cmd = new SqlCommand("sp_RegistrarCompra", oConexion);
            cmd.Parameters.AddWithValue("IdUsuario", oCompraProveedor.IdUsuario);
            cmd.Parameters.AddWithValue("IdProveedor", oCompraProveedor.IdProveedor);
            cmd.Parameters.AddWithValue("TotalProducto", oCompraProveedor.TotalProducto);
            cmd.Parameters.AddWithValue("Total", oCompraProveedor.Total);
            cmd.Parameters.AddWithValue("QueryDetalleCompraProveedor", query.ToString());
            cmd.Parameters.Add("Resultado", SqlDbType.Bit).Direction = ParameterDirection.Output;
            cmd.CommandType = CommandType.StoredProcedure;
            oConexion.Open();
            respuesta = cmd.ExecuteNonQuery() > 0;
        }
    }
}
```

Figura 77: Controlador Compra

```
0 referencias
public ActionResult Compra()
{
    if (Session["Usuario"] == null)
        return RedirectToAction("Index", "Login");
    else
        oUsuario = (Usuario)Session["Usuario"];

    return View();
}
```

Figura 78: Vista Detalle de Compra



6.3.2.1.3. Control operacional

Finalmente, en el área de control operacional, se completó la implementación de los módulos de control de inventario no comercial: Regalía y Merma, que son críticos para llevar un control de pérdidas.

Tabla 12: Sprint Backlog para el control de Mermas y Regalías

ORIGEN	ROL	ASIGNADO	ACCION	ESTADO	PRIORIDAD	HORAS
MERMAS	Developer	Hosmand Rafael Belli Cespedes	Creación de [dbo].[Merma]	Completada	Baja	3.00
MERMAS	Developer	Hosmand Rafael Belli Cespedes	Creación de SPS para Merma	Completada	Media	3.00
MERMAS	Developer	Hosmand Rafael Belli Cespedes	Creación de Clases: Merma	Completada	Baja	2.00
MERMAS	Developer	Hosmand Rafael Belli Cespedes	Creación de Lógica: Registrar Merma	Completada	Media	3.00
MERMAS	Developer	Hosmand Rafael Belli Cespedes	Creación de Controlador: Merma	Completada	Baja	2.00
MERMAS	Developer	Hosmand Rafael Belli Cespedes	Creación de Vista: Merma	Completada	Media	4.00
MERMAS	Qua Tester	Hosmand Rafael Belli Cespedes	Pruebas unitarias de Merma	Completada	Media	3.00
REGALIAS	Developer	Cristhiam Omar Suarez Nuñez	Creación de [dbo].[Regalia]	Completada	Baja	3.00
REGALIAS	Developer	Cristhiam Omar Suarez Nuñez	Creación de SPS para Regalia	Completada	Baja	3.00
REGALIAS	Developer	Cristhiam Omar Suarez Nuñez	Creación de Clases: Regalia	Completada	Media	2.00
REGALIAS	Developer	Cristhiam Omar Suarez Nuñez	Creación de Lógica: Registrar Regalia	Completada	Baja	3.00
REGALIAS	Developer	Cristhiam Omar Suarez Nuñez	Creación de Controlador: Regalia	Completada	Media	2.00
REGALIAS	Developer	Cristhiam Omar Suarez Nuñez	Creación de Vista: Regalia	Completada	Media	4.00
REGALIAS	Qua Tester	Hosmand Rafael Belli Cespedes	Pruebas unitarias de Regalias	Completada	Media	3.00

Figura 79: Tabla de Merma

Merma	
IdMerma	
IdProducto	
DescripcionMerma	
CantidadMerma	
FechaRegistroMerma	

Figura 80: Tabla de Regalía

Regalia	
IdRegalia	
DescripcionRegalia	
CantidadRegalia	
IdProducto	
FechaRegistroRegalia	

Figura 81: Clase Regalía

```
namespace ProyectoTest.Models
{
    public class Regalia
    {
        public int IdRegalia { get; set; }
        public Producto oProducto { get; set; }
        public string Descripcion { get; set; }
        public string FechaRegalia { get; set; }
        public float Cantidad { get; set; }
    }
}
```

Figura 82: Procedimiento almacenado Merma

```
    BEGIN
        SET NOCOUNT ON;
        SET @Resultado = 0;

        BEGIN TRY
            INSERT INTO Merma (DescripcionMerma, CantidadMerma, IdProducto)
            VALUES (@Descripcion, @Cantidad, @IdProducto);

            SET @Resultado = SCOPE_IDENTITY();

            -- Obtener IdModulo e IdAccion
            DECLARE @IdModulo INT = (SELECT IdRolModulo FROM RolModulo
            WHERE NombreRolModulo = 'Mermas');
            DECLARE @IdAccion INT = (SELECT IdRolAccion FROM RolAccion
            WHERE NombreRolAccion = 'Crear');
            DECLARE @NombreProducto VARCHAR(200) = (SELECT NombreProducto
            FROM Producto WHERE IdProducto = @IdProducto);
```

Figura 83: Procedimiento almacenado Regalía

```
    BEGIN
        SET @Resultado = 0;

        BEGIN
            INSERT INTO Regalia (DescripcionRegalia, CantidadRegalia, IdProducto)
            VALUES (@Descripcion, @Cantidad, @IdProducto);

            DECLARE @IdRegalia INT = SCOPE_IDENTITY();

            -- Obtener IdModulo y IdAccion
            DECLARE @IdModulo INT = (SELECT IdRolModulo FROM RolModulo
            WHERE NombreRolModulo = 'Regalías');
            DECLARE @IdAccion INT = (SELECT IdRolAccion FROM RolAccion
            WHERE NombreRolAccion = 'Crear');
            DECLARE @NombreProducto VARCHAR(200) = (SELECT NombreProducto
            FROM Producto WHERE IdProducto = @IdProducto);
```

Figura 84: Clase Merma

```
namespace ProyectoTest.Models
{
    12 referencias
    public class Merma
    {
        7 referencias
        public int IdMerma { get; set; }
        5 referencias
        public Producto oProducto { get; set; }
        5 referencias
        public string Descripcion { get; set; }
        3 referencias
        public string FechaMerma { get; set; }
        7 referencias
        public float Cantidad { get; set; }
    }
}
```

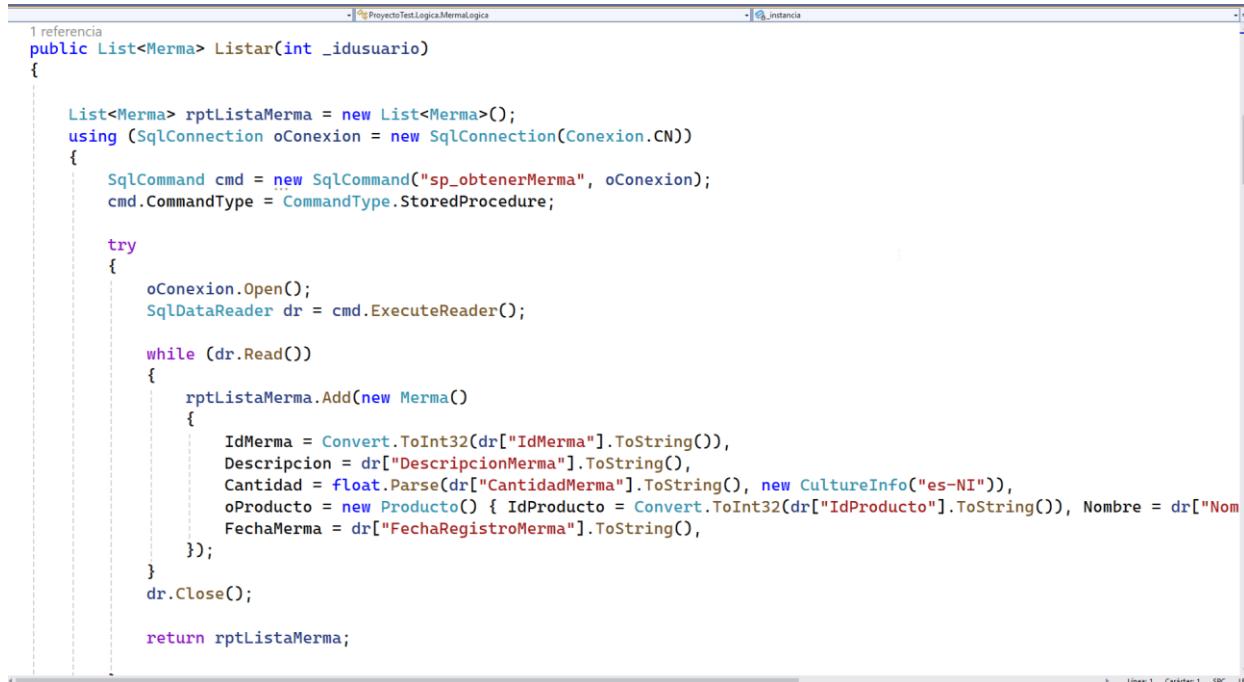
Figura 85: Lógica Regalía

```
1 referencia
public List<Regalia> Listar(int _idusuario)
{
    List<Regalia> rptListaRegalia = new List<Regalia>();
    using (SqlConnection oConexion = new SqlConnection(Conexion.CN))
    {
        SqlCommand cmd = new SqlCommand("sp_obtenerRegalia", oConexion);
        cmd.CommandType = CommandType.StoredProcedure;

        try
        {
            oConexion.Open();
            SqlDataReader dr = cmd.ExecuteReader();

            while (dr.Read())
            {
                rptListaRegalia.Add(new Regalia()
                {
                    IdRegalia = Convert.ToInt32(dr["IdRegalia"].ToString()),
                    Descripcion = dr["DescripcionRegalia"].ToString(),
                    Cantidad = float.Parse(dr["CantidadRegalia"].ToString()),
                    oProducto = new Producto() { IdProducto = Convert.ToInt32(dr["IdProducto"].ToString()), Nombre = dr["Nombre"].ToString() },
                    FechaRegalia = dr["FechaRegistroRegalia"].ToString(),
                });
            }
            dr.Close();
        }
        return rptListaRegalia;
    }
}
```

Figura 86: Lógica Mermas



```
1 referencia
public List<Merma> Listar(int _idusuario)
{
    List<Merma> rptListaMerma = new List<Merma>();
    using (SqlConnection oConexion = new SqlConnection(Conexion.CN))
    {
        SqlCommand cmd = new SqlCommand("sp_obtenerMerma", oConexion);
        cmd.CommandType = CommandType.StoredProcedure;

        try
        {
            oConexion.Open();
            SqlDataReader dr = cmd.ExecuteReader();

            while (dr.Read())
            {
                rptListaMerma.Add(new Merma()
                {
                    IdMerma = Convert.ToInt32(dr["IdMerma"].ToString()),
                    Descripcion = dr["DescripcionMerma"].ToString(),
                    Cantidad = float.Parse(dr["CantidadMerma"].ToString(), new CultureInfo("es-NI")),
                    oProducto = new Producto() { IdProducto = Convert.ToInt32(dr["IdProducto"].ToString()), Nombre = dr["Nombre"].ToString() },
                    FechaMerma = dr["FechaRegistroMerma"].ToString(),
                });
            }
            dr.Close();
        }
        return rptListaMerma;
    }
}
```

Figura 87: Controlador Regalía

```
0 referencias
public ActionResult Regalia()
{
    if (Session["Usuario"] == null)
        return RedirectToAction("Index", "Login");

    return View();
}
```

Figura 88: Controlador Merma

```
0 referencias
public ActionResult Merma()
{
    if (Session["Usuario"] == null)
        return RedirectToAction("Index", "Login");

    return View();
}
```

Figura 89: Vista Regalía

Actividades / Regalías

>Listado de Regalías

Crear Nuevo

Buscar:

Descripción	Cantidad	Nombre del Producto	Fecha de la Regalía
Regalía de Agua por Fuente Pura	1	Agua Cielo 500ml	16/11/2024 10:35:02
Regalía de Agua por Fuente Pura	5	Agua Cielo 500ml	16/11/2024 15:11:59
Bono de Leche Nido	9	Leche Nido Pequeño	16/11/2024 14:48:19
Regalía de Queso Lala	11	Queso Amarillo	15/11/2024 15:07:27

Mostrando registros del 1 al 10 de un total de 10 registros

AI

Figura 90: Vista mermas

Actividades / Merma

Listado de Mermas

Crear Nuevo

Search: []

Descripción	Cantidad	Nombre del Producto	Fecha de la Merma
Agua	5	Agua Cielo 500ml	22/11/2024 18:23:40
Agua en mal estado	1	Agua Cielo 500ml	22/11/2024 10:35:02
Agua en mal estado	5	Agua Cielo 500ml	21/11/2024 15:20:39
Agua ponchadas	5	Agua Cielo 500ml	19/11/2024 15:11:59
Bolsas dañadas	6	Ranchitas Naturales	19/11/2024 14:56:46
Envoltura de queso dañado	1	Queso Amarillo	19/11/2024 16:26:47

Figura 91: Crear mermas



Formulario para crear mermas:

- Título: Merma
- Botón de cierre (X) en la parte superior derecha.
- Campos de texto:
 - Descripción
 - Producto:
 - Agua Cielo 500ml
 - Cantidad
- Botones de acción en la parte inferior:
 - Cerrar (rojo)
 - Guardar (azul)

Figura 92: Crear regalías



Formulario para crear regalías:

- Título: Regalía
- Botón de cierre (X) en la parte superior derecha.
- Campos de texto:
 - Descripción
 - Producto:
 - Agua Cielo 500ml
 - Cantidad
- Botones de acción en la parte inferior:
 - Cerrar (rojo)
 - Guardar (azul)

6.3.2.2. Validación del Incremento

Al finalizar la iteración, el Incremento fue presentado al cliente para la Sprint Review. Para ver más detalles, véase Figura 99: Acta de reunión sprint 2 del Anexo II.

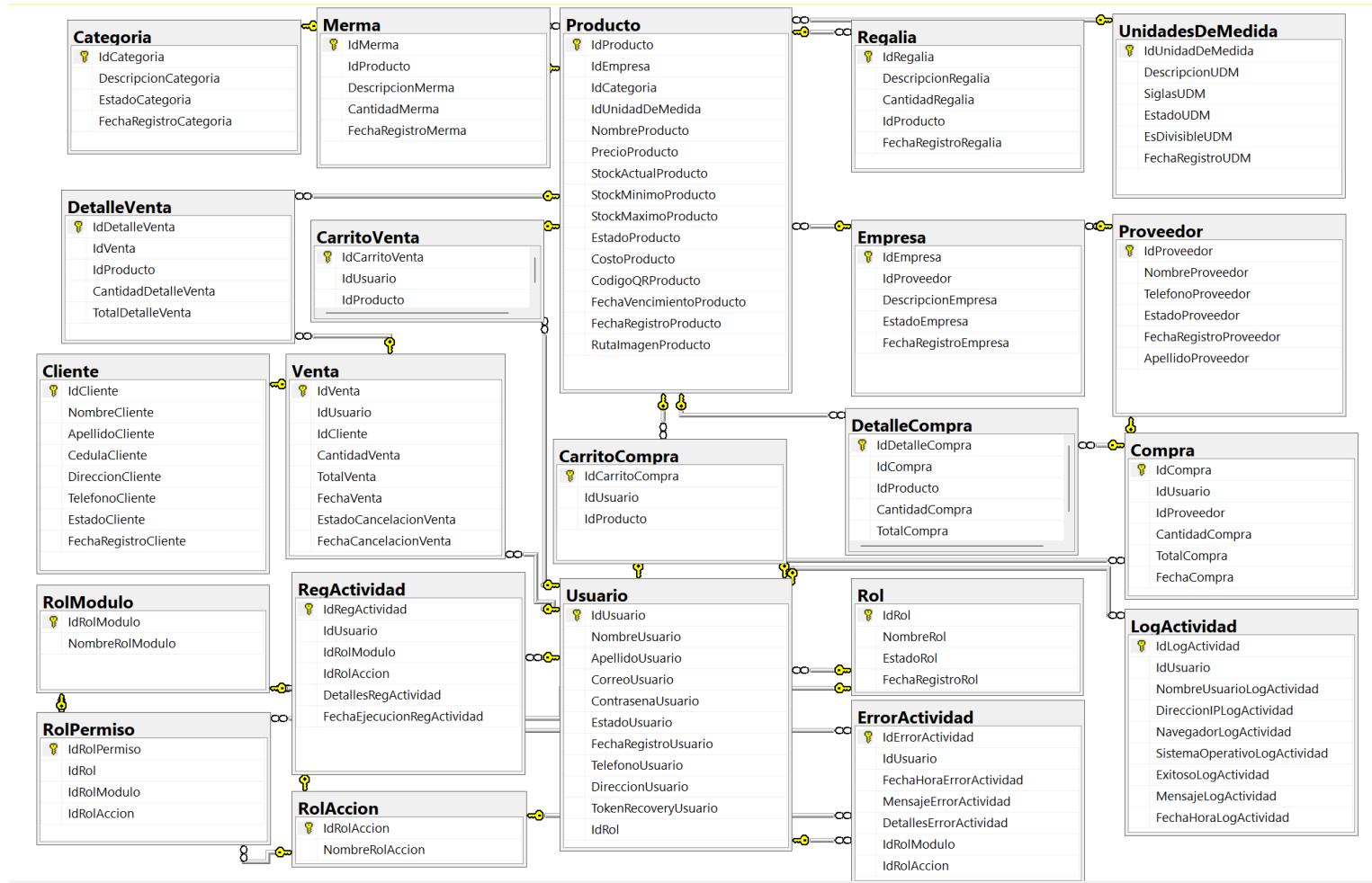
Tabla 13: Resumen del Sprint Review Sprint 2

Sprint Review	Valor entregado
Pruebas de Usuario	Se realizaron pruebas de aceptación por parte de la Pulperia Jehová Jireh. Se validó la operatividad y funcionalidad de todos los módulos implementados, confirmando que el flujo transaccional y la gestión de inventario cumplían con la lógica de negocio básica.
Resultado Tangible	Se realizó la entrega de las nuevas funcionalidades para las operaciones transaccionales y de inventario. Este resultado incluyó la interfaz, lógica y base de datos de los módulos de Producto, Categoría, Unidad de Medida, Ventas, Compras, Mermas y Regalías.
Cambios y Solicitudes del Cliente	El sprint concluyó con una reorientación estratégica. A pesar de la alta funcionalidad lograda, el cliente solicitó modificaciones en funcionalidades existentes. Esto, dio origen a nuevas historias de usuario y se procedió al levantamiento del Product Backlog. Esta adaptabilidad, respaldada por Scrum, comprometió al equipo a priorizar estas correcciones y refinamientos en el siguiente ciclo de desarrollo, asegurando la plena satisfacción del cliente sobre la expansión de nuevos módulos.

6.3.3. Diseño de Base de Datos Completo

Cada avance en el diseño de la base de datos que se generaba en cada sprint resultó en este diagrama.

Figura 93: Diseño de Base de Datos Jehova Jireh



6.3.4. Pruebas unitarias

El equipo priorizó la calidad, dedicando tiempo específico de los sprint para verificar la precisión y el correcto funcionamiento de la lógica de negocio en los módulos clave.

Tabla 14: Matriz de pruebas Login

Módulo	Prueba	Escenario	Pasos	Criterios	Estado
Login	Iniciar sesión	Usuario válido inicia sesión	1. Ir a login 2. Ingresar usuario/claves 3. Presionar "Ingresar"	El sistema permite el acceso y muestra el menú principal	Exitoso
Login	Recuperación	Usuario solicita recuperación	1. Ir a Inicio de Sesión 2. Click en "Olvidé mi contraseña" 3. Ingresar correo válido	El sistema envía un correo con instrucciones para restablecer contraseña	Exitoso
Login	Inicio de Sesión	Restablecer contraseña	1. Abrir correo 2. Click en enlace recibido 3. Ingresar nueva contraseña 4. Reestablecer Contraseña	El sistema actualiza la contraseña y permite el acceso con la nueva clave	Exitoso

Tabla 15: Matriz de pruebas Usuario y Rol

Módulo	Prueba	Escenario	Pasos	Criterios	Estado
Usuario	Iniciar sesión	Usuario válido inicia sesión	1. Ir a login 2. Ingresar usuario/claves 3. Presionar "Ingresar"	El sistema permite el acceso y muestra el menú principal	Exitoso
Usuario	Recuperación	Usuario solicita recuperación	1. Ir a Inicio de Sesión 2. Click en "Olvidé mi contraseña" 3. Ingresar correo válido	El sistema envía un correo con instrucciones para restablecer contraseña	Exitoso
Rol	Inicio de Sesión	Restablecer contraseña	1. Abrir correo 2. Click en enlace recibido 3. Ingresar nueva contraseña 4. Confirmar Reestablecer Contraseña	El sistema actualiza la contraseña y permite el acceso con la nueva clave	Exitoso
Usuario	Crear usuarios	Registrar un nuevo usuario	1. Ir a Usuarios 2. Click en "Crear" 3. Completar campos 4. Guardar	El usuario se guarda y aparece en la tabla	Exitoso
Usuarios	Editar usuario	Modificar usuarios	1. Ir a Usuarios 2. Seleccionar registro 3. Editar campos 4. Guardar	Los cambios se reflejan en la tabla	Exitoso
Usuarios	Reestablecer contraseña	Reestablecer contraseña de usuario	1. Ir a Usuarios 2. Seleccionar registro 3. Reestablecer contraseña	La contraseña del usuario se reestablece por una generada aleatoriamente.	Exitoso

Tabla 16: Matriz de pruebas Cliente

Módulo	Prueba	Escenario	Pasos	Criterios	Estado
Cliente	Crear cliente	Registrar un nuevo cliente	1. Ir a Clientes 2. Click en “Crear” 3. Completar formulario 4. Guardar	El cliente se guarda y aparece en la tabla	Exitoso
Cliente	Editar cliente	Modificar datos de un cliente	1. Ir a Clientes 2. Seleccionar un registro 3. Editar campos 4. Guardar	Los cambios se reflejan en la tabla	Exitoso
Cliente	Eliminar cliente	Eliminar un cliente registrado	1. Ir a Clientes 2. Seleccionar registro 3. Eliminar	El registro desaparece de la tabla y no debe mostrarse en búsquedas	Exitoso
Cliente	Validar campos obligatorios	Validación de formulario al crear cliente	1. Ir a “Crear Cliente” 2. Dejar campos vacíos 3. Guardar	El sistema no debe permitir guardar y debe mostrar mensajes de error.	Exitoso

Tabla 17: Matriz de pruebas Proveedor

Módulo	Prueba	Escenario	Pasos	Criterios	Estado
Proveedor	Crear proveedor	Registrar un nuevo proveedor	1. Ir a proveedor 2. Click en "Crear" 3. Completar formulario 4. Guardar"	El cliente se guarda y aparece en la tabla	Exitoso
Proveedor	Editar proveedor	Modificar datos de un proveedor	1. Ir a proveedor 2. Seleccionar un registro 3. Editar campos 4. Guardar	Los cambios se reflejan en la tabla	Exitoso
Proveedor	Eliminar proveedor	Eliminar un cliente proveedor	1. Ir a proveedor 2. Seleccionar registro 3. Eliminar	El registro desaparece de la tabla y no debe mostrarse en búsquedas	Exitoso
Proveedor	Validar campos obligatorios	Validación de formulario	1. Ir a "Crear Proveedor" 2. Dejar campo vacío	El sistema no debe permitir guardar y debe mostrar mensajes de error.	Exitoso

Tabla 18: Matriz de pruebas Empresa

Módulo	Prueba	Escenario	Pasos	Criterios	Estado
Empresa	Consultar módulo	Usuario accede a la vista de empresas	1. Ir a menú 2. Desplegar opciones de Administración Interna 3. Seleccionar “Empresa”	El sistema muestra tabla de empresas con sus proveedores asociados	Exito
Empresa	Crear empresa	Registrar una nueva empresa	1. Ir a Empresa 2. Click en “Crear” 3. Completar campos 4. Guardar	La empresa se guarda y aparece en la tabla	Exito
Empresa	Editar empresa	Modificar datos de una empresa	1. Ir a Empresa 2. Seleccionar registro 3. Editar campos 4. Guardar	Los cambios se reflejan en la tabla	Exito
Empresa	Eliminar empresa	Eliminar una empresa registrada	1. Ir a Empresa 2. Seleccionar registro 3. Eliminar	El registro desaparece de la tabla	Exito

Tabla 19: Matriz de pruebas Categoría

Módulo	Prueba	Escenario	Pasos	Criterios	Estado
Categorías	Consultar módulo	Usuario accede a la vista de categorías	1. Ir a menú 2. Desplegar opciones de Administración Interna 3. Seleccionar “Categorías”	El sistema muestra tabla de categorías	Exitoso
Categorías	Crear categoría	Registrar una nueva categoría	1. Ir a Categorías 2. Click en “Crear” 3. Completar campos 4. Guardar	La categoría se guarda y aparece en la tabla	Exitoso
Categorías	Editar categoría	Modificar datos de una categoría	1. Ir a Categorías 2. Seleccionar categoría 3. Editar campos 4. Guardar	Los cambios se reflejan en la tabla	Exitoso
Categorías	Eliminar categoría	Eliminar una categoría registrada	1. Ir a Categorías 2. Seleccionar categoría 3. Eliminar	El registro desaparece de la tabla	Exitoso
Categorías	Validar campo obligatorio	Validación de formulario	1. Intentar guardar categoría con campos vacíos 2. Guardar	El sistema no debe permitir guardar y muestra mensajes de error	Exitoso

Tabla 20: Matriz de pruebas Producto, Venta y Compra

Módulo	Prueba	Escenario	Pasos	Criterios	Estado
Productos	Ver catálogo	Mostrar productos disponibles	1. Ir a Ventas	Se muestran productos con stock y precios	Exitoso
Productos	Ver detalle de producto	Consultar información completa del producto	1. Seleccionar producto 2. Click en "Detalle"	Se muestran todas las características y stock del producto	Exitoso
Productos	Editar fecha de vencimiento	Ajustar fecha de caducidad del producto	1. Seleccionar producto 2. Editar fecha 3. Guardar	Fecha de caducidad se refleja correctamente	Exitoso
Productos	Editar costo/precio	Modificar datos de producto antes de comprar	1. Seleccionar producto 2. Editar costo o precio 3. Guardar	El costo o precio se actualiza correctamente	Exitoso
Compras	Crear nueva orden de compra	Registrar una compra con productos y proveedor	1. Click "Nueva orden" 2. Seleccionar proveedor 3. Seleccionar productos 4. Añadir a la compra	La compra queda registrada, los productos se añaden al listado	Exitoso
Ventas	Añadir a la factura	Añadir productos a una venta	1. Seleccionar productos 2. Click "Añadir a factura"	Productos aparecen en la factura	Exitoso
Ventas	Crear nueva orden de venta	Registrar venta de productos	1. Añadir productos 2. Seleccionar cliente 3. Confirmar Venta	La venta queda registrada y productos se descuentan del inventario	Exitoso
Ventas	Calcular venta total	Validar total de la factura	1. Seleccionar productos 2. Revisar total 3. Confirmar Venta	El total coincide con la suma de productos	Exitoso

Tabla 21: Matriz de pruebas Merma

Módulo	Prueba	Escenario	Pasos	Criterios	Estado
Mermas	Consultar módulo	Usuario accede a la vista de mermas	1. Ir a menú 2. Desplegar opciones de Administración Interna 3. Seleccionar "Mermas"	El sistema muestra la tabla de mermas	Exitoso
Mermas	Crear merma	Registrar una nueva merma	1. Ir a Mermas 2. Click en "Crear" 3. Completar campos 4. Guardar	La merma se guarda y aparece en la tabla	Exitoso
Mermas	Eliminar merma	Eliminar un registro de merma	1. Ir a Mermas 2. Seleccionar registro 3. Eliminar	El registro desaparece de la tabla	Exitoso
Mermas	Listar productos	Consultar productos relacionados con mermas	1. Ir a Mermas 2. Seleccionar "Listar productos"	El sistema muestra los productos relacionados	Exitoso
Mermas	Quitar del inventario mermas registradas	Validar impacto en inventario	1. Crear merma 2. Guardar 3. Revisar inventario	El sistema descuenta del inventario la cantidad mermada	Exitoso
Mermas	Devolver inventario al eliminar merma	Validar reversión al eliminar	1. Seleccionar Merma 2. Eliminar 3. Revisar inventario	El sistema devuelve al inventario la cantidad eliminada	Exitoso
Mermas	Filtrar contenido de la tabla	Usuario aplica filtros	1. Ir a Mermas 2. Ingresar criterio en filtros	La tabla muestra solo coincidencias	Exitoso

VII. Conclusiones y recomendaciones

7.1. Conclusiones

Tras la finalización de este proyecto, se han obtenido los siguientes resultados:

Se logró exitosamente el desarrollo del Sistema web para el control de inventario y facturación de la pulperia Jehová Jireh, cumpliendo con cada uno de los objetivos específicos.

Se logró definir los requerimientos funcionales y no funcionales a través del levantamiento de historias de usuario resultando en un Product Backlog, el cual guio toda la ejecución en cada uno de los Sprints.

Posteriormente, la meta de diseñar la base de datos y codificar los distintos módulos del sistema web fueron ejecutados a medida que el proyecto fue avanzando en cada sprint realizado, empleando el gestor de base de datos SQL Server y la arquitectura ASP.NET Modelo Vista Controlador y con C# como lenguaje principal.

Finalmente, el objetivo de evaluar los resultados del funcionamiento del sistema se verificó a través de las pruebas de aceptación del usuario realizadas con el cliente durante cada Sprint Review, confirmando la funcionalidad de cada incremento.

Por tanto, el sistema web desarrollado optimiza los procesos de inventario y facturación de la pulperia Jehová Jireh.

7.2. Recomendaciones

Para garantizar el buen desempeño y maximizar el valor del sistema desarrollado, se sugiere lo siguiente:

Uso del Manual de Usuario: Para asegurar la operación eficiente y autónoma del sistema, se sugiere que el personal utilice el manual de usuario como la principal fuente de consulta.

Plan de Mantenimiento y Respaldo de Datos: El mantenimiento periódico es fundamental para la salud del sistema. Se aconseja establecer un calendario para tareas esenciales, tales como:

- **Mantenimiento del Servidor Local:** Asegurar que los equipos de hardware donde se aloja el sistema estén en óptimas condiciones.
- **Copia de Seguridad (Backup):** Implementar una rutina diaria de copias de seguridad de la base de datos.

Roles y Permisos: Utilizar las funcionalidades de gestión de roles del sistema para asignar los permisos mínimos necesarios a cada usuario.

Seguridad de Credenciales: Capacitar a los usuarios sobre la importancia de la seguridad de sus contraseñas y la responsabilidad que conlleva el acceso al sistema.

Transición a un hosting web: Para una experiencia completa y el crecimiento futuro de la pulpería, se sugiere la migración a un entorno web considerando un proveedor especializado en tecnología Microsoft como SmarterASP.net.

7.3. Limitaciones

El término factura se utiliza comúnmente para referirse a los documentos fiscales autorizados, como los recibos o comprobantes de venta, que los contribuyentes deben emitir para cada operación comercial realizada. Sin embargo, en el contexto de pequeños comercios, como las pulperías, el uso de la palabra factura suele ser más amplio y no siempre corresponde con la definición fiscal exacta. Esto se debe a que muchos de estos comerciantes están inscritos en regímenes de cuota fija. [22]

Para el contexto de este proyecto, el término de factura no se aplica rigurosamente con lo desarrollado, ya que la Pulperia Jehova Jireh es de cuota fija y no tiene ninguna obligación tributaria para tramitar facturas que generen crédito fiscal, es decir, que puedan ser utilizadas para la deducción de impuestos como el IVA. En su lugar, se presentan los comprobantes de venta o recibos que cumplen la función de documentar la transacción y no permiten las deducciones fiscales asociadas a las facturas tradicionales.

VIII. Bibliografía

- [1] R. J. García Poveda y M. N. Porras Herrera, «Desarrollo de sistema web para la gestión de los procesos de compraventa, facturación y control de inventario para la tienda Winnie's Menorquinas.,» Universidad Nacional de Ingeniería, Nicaragua, 2021.
- [2] A. J. Chamorro López y C. A. Ibujés Revelo, «Sistema de información para la gestión del inventario de la Distribuidora Colombina - Tulcán,» Universidad Politécnica Estatal del Carchi, Ecuador, 2023.
- [3] G. Westreicher, «Inventario: Qué es, origen e importancia,» Mayo 2020. [En línea]. Available: <https://economipedia.com/definiciones/inventario.html>.
- [4] contabilidadfinanzas, «Factura: Qué es, Para qué Sirve, Tipos,» contabilidadfinanzas, [En línea]. Available: <https://contabilidadfinanzas.com/blog/que-es-una-factura>.
- [5] K. Schwaber y J. Sutherland, «La Guía Definitiva de Scrum: Las Reglas del Juego,» scrumguides.org, Boston, 2020.
- [6] pm-partners, «What is Scrum? An overview of Scrum and The Agile Journey,» Agile and Scaled Agile, Learning and Development, Australia, 2024.
- [7] Microsoft Inc, «Visual Studio: IDE and Code Editor for Software Developers,» Microsoft, Redmond, 2024.
- [8] Microsoft Inc, «¿Qué es .NET?,» Microsoft Inc, Redmond, 2024.
- [9] J. Carmona, «Qué es ASP.NET y cuáles son sus puntos fuertes,» 2022.
- [10] D. Ortego, «Qué es C#: Introducción,» OpenWebinars, Andalucía, 2017.
- [11] R. West y M. Ray, «¿Qué es SQL Server?,» Microsoft, Redmond, 2024.

- [12] R. D. Hernandez, «El patrón modelo-vista-controlador: Arquitectura y frameworks explicados,» FreeCodeCamp, San Francisco, 2021.
- [13] D. Walsh, «GitHub vs. GitLab: diferencias, similitudes y usos,» HubSpot Inc, Cambridge, 2023.
- [14] M. Coppola, «¿Qué es HTML y cómo utilizarlo?,» HubSpot Inc, Cambridge, 2023.
- [15] J. Juviler, «The Ultimate Guide to CSS,» HubSpot Inc, Cambridge, 2024.
- [16] M. Coppola, «¿Qué es JavaScript, para qué sirve y cómo funciona?,» HubSpot Inc, Cambridge, 2023.
- [17] P. Londoño, «¿Qué es Bootstrap?,» HubSpot Inc, Cambridge, 2023.
- [18] Microsoft Inc, «¿Qué es Power BI Desktop?,» 2025.
- [19] A. De León, «Servidor IIS: ¿Qué es? Características, Ventajas, Requisitos e Instalación,» 2019.
- [20] wolfdisenoweb.com, «SmarterASP.NET: La Mejor Plataforma de Alojamiento para Aplicaciones ASP.NET,» 2025.
- [21] E. Nunez, «¿Qué es ADO.NET y por qué deberías aprenderlo?,» 20 Mayo 2025. [En línea]. Available: <https://blog.edticsacademy.com/2025/05/que-es-adonet-y-por-que-deberias.html>.
- [22] La Gaceta, «REGISTRO DE IMPRENTAS E IMPORTADORES DE FACTURAS,» La Gaceta, Managua, 2007.
- [23] Sevasa, «Sevasa: Tecnología a precios accesibles,» [En línea]. Available: www.sevasaonline.com.
- [24] X. S.A., «Licencias Office, Licencias Windows, Licencias Antivirus,» [En línea]. Available: <https://activaya.com.ni/index.php/es/>.

[25] SmarterASP.NET, «SmarterASP.net - Unlimited ASP.NET Web Hosting,» [En línea]. Available: <https://www.smarterasp.net/>.

IX. Anexos

Anexo I. Cartas de aceptación	100
Anexo II. Actas de reunión	103
Anexo III. Cuestionario.....	113
Anexo IV. Estudio de Factibilidad.....	118
Anexo V. Requerimiento inicial.....	128
Anexo VI. Requerimiento sprint 3.....	142
Anexo VII. Sprint 3.....	149
Anexo VIII. Sprint 4.....	187
Anexo IX. Sprint 5.....	201
Anexo X. Acta cierre de proyecto	204

Anexo I. Cartas de aceptación

Figura 94: Carta de aceptación

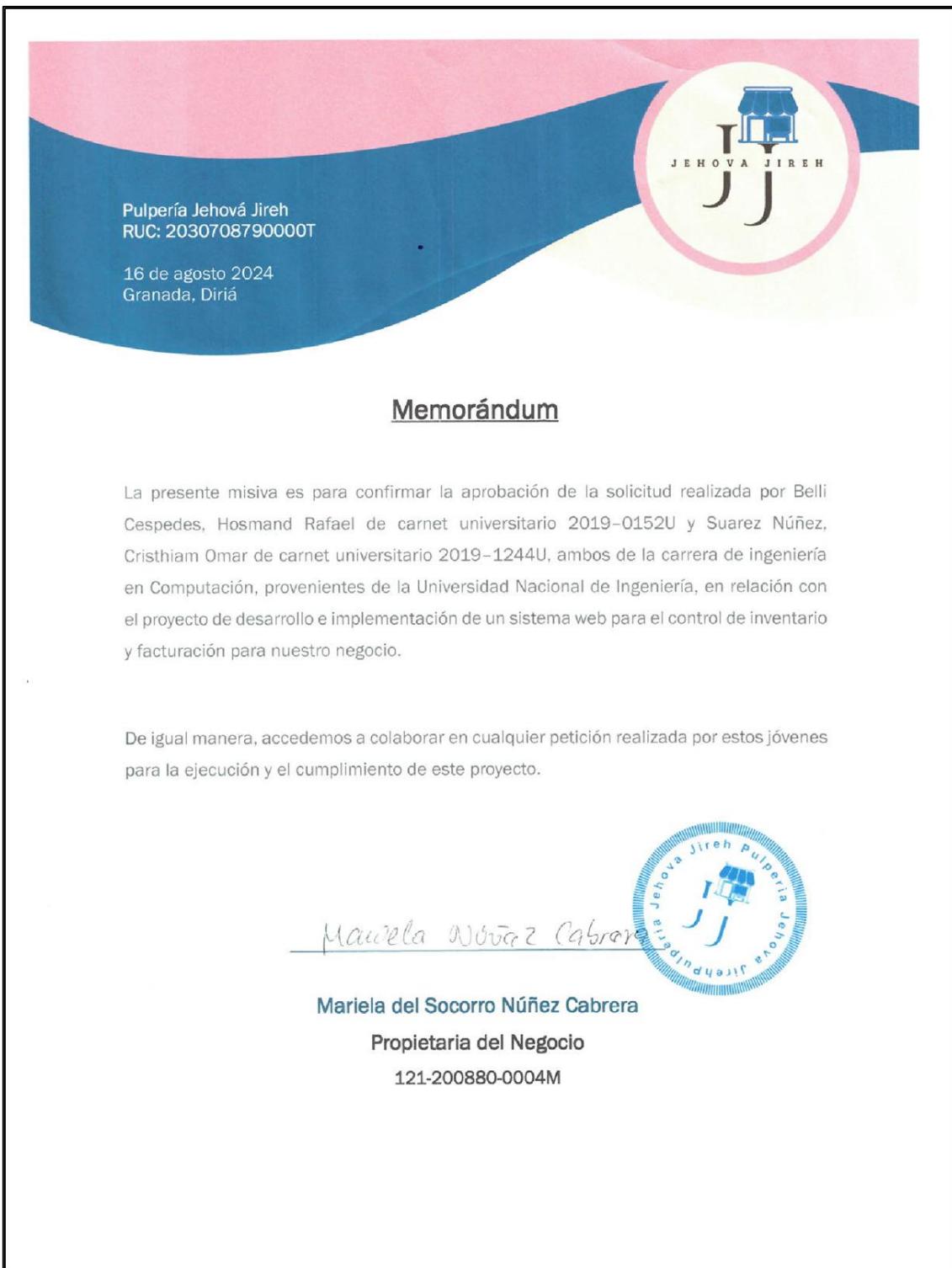
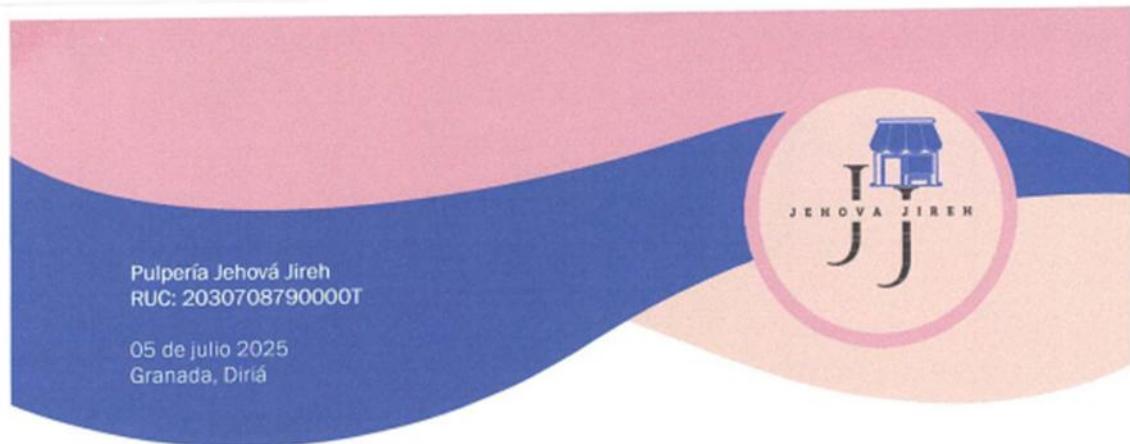


Figura 95: Carta de agradecimiento



Agradecimiento

La presente misiva es para agradecer a los estudiantes provenientes de la Universidad Nacional de Ingeniería, ambos de la carrera de ingeniería en Computación, Br. Belli Cespedes, Hosmand Rafael de carnet universitario 2019-0152U y Br. Suarez Núñez, Cristhiam Omar de carnet universitario 2019-1244U, por realizar de manera impecable el proyecto de desarrollo e implementación de un sistema web para el control de inventario y facturación cumpliendo con cada uno de los requerimientos propuestos para nuestro negocio.

La implementación de este sistema marca un antes y un después para la pulperia Jehová Jireh. Representa un gran paso hacia la modernización de nuestra infraestructura, optimizando nuestra gestión interna, potenciando significativamente el crecimiento del negocio y permitiendo brindar una mejor experiencia a nuestros clientes.

Mariela Núñez Cabrera

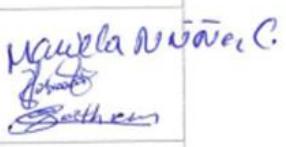
Mariela del Socorro Núñez Cabrera

Propietaria del Negocio

121-200880-0004M

Anexo II. Actas de reunión

Figura 96: Acta de reunión exploración y recopilación de datos

 UNI - Ingeniería en Computación			Sistema Web de Inventario y Facturación Jehová Jireh	
Acta de Reunión				
Fecha	Granada, 24 de agosto del 2024			
Lugar	Pulpería Jehová Jireh			
Participantes	Mariela Núñez Cabrera Hosmand Rafael Belli Cespedes Cristhiam Omar Suarez Núñez	Propietaria del Negocio Scrum Máster/Developer Team Product Owner/Developer Team		
Puntos del orden del día	01 - Exploración del entorno 02 - Recopilación de datos 03 - Operatividad e infraestructura 04 - Realización de cuestionario			
Detalles				
Punto 1: Exploración del entorno Jehová Jireh nos instruye acerca de su negocio, sus inicios y su estado actual.				
Resultados/Acuerdos	¿Quién Solicita?	¿Cuándo?		
Jehova Jireh expone su perspectiva de los inicios del negocio y el crecimiento exponencial que ha tenido hasta estos momentos.	Jehova Jireh	24/08/24		
Punto 2: Recopilación de datos Jehová Jireh expone su visión y algunas funcionalidades deseadas para el sistema.				
Resultados/ Acuerdos	¿Quién Solicita?	¿Cuándo?		
Jehová Jireh expone su visión acerca del sistema definiendo algunas entidades como clientes, empresa, proveedores, productos y otras que debemos tomar en consideración como las mermas y las regalías. Jehová Jireh expone algunas funcionalidades deseadas como el control total y preciso del inventario de productos, la ejecución eficiente de ventas a clientes, seguimiento de mermas y regalías y algunos reportes de control.	Jehova Jireh	24/08/24		

Punto 3: Operatividad e Infraestructura

Jehová Jireh nos explica el funcionamiento del negocio y su infraestructura.

Resultados/ Acuerdos	¿Quién Solicita?	¿Cuándo?
<p>Jehova Jireh muestra las distintas formas de operar en la pulperia según sea el caso.</p> <ul style="list-style-type: none"> • Atención al cliente • Atención al proveedor • Gestión de inventario • Control de mermas <p>Jehová Jireh muestra los componentes electrónicos de software y hardware que utiliza para la gestión de su negocio.</p>	Jehova Jireh	24/08/24

Punto 4: Cuestionario de exploración

Jehová Jireh realiza un cuestionario exploratorio para entender las necesidades que el sistema deberá solventar.

Resultados/ Acuerdos	¿Quién Solicita?	¿Cuándo?
<p>Jehová Jireh realiza un cuestionario dividido en categorías como negocio, inventario, facturación y ventas, reportes, etc, para aterrizar la visión general del sistema y sentar bases para un requerimiento inicial.</p>	Jehova Jireh	24/08/24

Manuela Vazquez
Garcia

Figura 97: Acta de reunión requerimiento inicial

 UNI - Ingeniería en Computación	Sistema Web de Inventario y Facturación Jehová Jireh	
Acta de Reunión		
Fecha	Granada, 14 de septiembre del 2024	
Lugar	Pulperia Jehová Jireh	
Participantes	Mariela Núñez Cabrera	Propietaria del Negocio
	Hosmand Rafael Belli Cespedes	Scrum Máster/Developer Team
	Cristhiam Omar Suarez Núñez	Product Owner/Developer Team
Puntos del orden del día	01 – Presentación de estudio de factibilidad 02 – Levantamiento de Requerimiento Inicial	
Detalles		
Punto 1: Presentación de estudio de factibilidad Jehová Jireh se le presenta el estudio de factibilidad realizado.		
Resultados/Acuerdos ¿Quién Solicita? ¿Cuándo?		
Jehová Jireh aceptó los hallazgos del estudio de factibilidad y está de acuerdo con los beneficios que se obtendrán con la implementación del sistema. Jehová jireh se compromete a garantizar algunos de los componentes necesarios para la implementación del sistema.	Jehova Jireh	14/09/24
Punto 2: Levantamiento de requerimiento Jehová Jireh realiza el levantamiento del requerimiento del sistema.		
Resultados/ Acuerdos ¿Quién Solicita? ¿Cuándo?		
Jehova jireh expone los puntos de manera detallada acerca de las funcionalidades deseadas para el sistema web de inventario y facturación.	Jehova Jireh	14/09/24

Figura 98: Acta de reunión Sprint 1

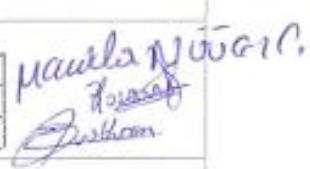
 UNI - Ingeniería en Computación	Sistema Web de Inventario y Facturación Jehová Jireh						
Acta de Reunión							
Fecha	Granada, 23 de noviembre del 2024						
Lugar	Pulperia Jehová Jireh						
Participantes	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%;">Mariela Núñez Cabrera</td> <td style="width: 50%;">Propietaria del Negocio</td> </tr> <tr> <td>Hosmand Rafael Belli Cespedes</td> <td>Scrum Master/Developer Team</td> </tr> <tr> <td>Cristhiam Omar Suarez Núñez</td> <td>Product Owner/Developer Team</td> </tr> </table> <div style="text-align: right; margin-top: -10px;">  </div>	Mariela Núñez Cabrera	Propietaria del Negocio	Hosmand Rafael Belli Cespedes	Scrum Master/Developer Team	Cristhiam Omar Suarez Núñez	Product Owner/Developer Team
Mariela Núñez Cabrera	Propietaria del Negocio						
Hosmand Rafael Belli Cespedes	Scrum Master/Developer Team						
Cristhiam Omar Suarez Núñez	Product Owner/Developer Team						
Puntos del orden del día	01 – Sesión de Pruebas UAT 02 – Revisión de Entregables						
Detalles							
Punto 1: Sesión de Pruebas UAT							
<p>Se llevó a cabo la sesión de pruebas del Sprint #1 en la que Jehová Jireh interactuó directamente con el sistema para validar su estructura, funcionalidades básicas y navegación.</p>							
Resultados/Acuerdos	¿Quién Solicita?	¿Cuándo?					
Jehová Jireh valida el dashboard inicial y las funcionalidades de los módulos entregados como Usuario, Rol, Cliente, Proveedor confirmando que la estructura del sistema, el diseño del dashboard y sus funcionalidades cumplen con sus expectativas.	Jehova Jireh	23/11/24					
No se reportaron incidencias. El cliente confirmó que el diseño del dashboard y las funcionalidades entregadas están correctas.							
Punto 2: Revisión de Entregables							
<p>Se realiza formal entrega del esqueleto funcional del sistema y la primera versión del dashboard.</p>							
Resultados/ Acuerdos	¿Quién Solicita?	¿Cuándo?					
Jehová Jireh Valida los resultados del Sprint 1 y se le realiza formal entrega de la primera versión del sistema.	Jehova Jireh	23/11/24					
Se dio la aprobación formal para cerrar el Sprint 1 y dar inicio al Sprint 2.							

Figura 99: Acta de reunión Sprint 2

 UNI - Ingeniería en Computación	Sistema Web de inventario y Facturación Jehová Jireh						
Acta de Reunión							
Fecha	Granada, 11 de enero del 2025						
Lugar	Pulperia Jehová Jireh						
Participantes	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 33%;">Mariela Núñez Cabrera</td> <td style="width: 33%;">Propietaria del Negocio</td> </tr> <tr> <td>Hosmand Rafael Belli Cespedes</td> <td>Scrum Master/Developer Team</td> </tr> <tr> <td>Cristhiam Omar Suárez Núñez</td> <td>Product Owner/Developer Team</td> </tr> </table> <div style="text-align: right; margin-top: -10px;">  </div>	Mariela Núñez Cabrera	Propietaria del Negocio	Hosmand Rafael Belli Cespedes	Scrum Master/Developer Team	Cristhiam Omar Suárez Núñez	Product Owner/Developer Team
Mariela Núñez Cabrera	Propietaria del Negocio						
Hosmand Rafael Belli Cespedes	Scrum Master/Developer Team						
Cristhiam Omar Suárez Núñez	Product Owner/Developer Team						
Puntos del orden del día	01 – Sesión de Pruebas UAT 02 – Revisión de Entregables 03 – Nuevo Requerimiento						
Detalles							
Punto 1: Sesión de Pruebas UAT <p>Se llevó a cabo la sesión de pruebas del Sprint #2 en la que Jehová Jireh interactuó directamente con el sistema para validar nuevas funcionalidades implementadas.</p>							
Resultados/Acuerdos	¿Quién Solicita?	¿Cuándo?					
Jehová Jireh valida las funcionalidades de los módulos restantes como Ventas, Compras, Productos, Mermas y Regalías, interactuando directamente con el sistema evaluando la operatividad y el flujo de trabajo.	Jehova Jireh	11/01/25					
Punto 2: Revisión de Entregables <p>Se demostraron las funcionalidades completas del Sprint 2, incluyendo los módulos de usuarios, inventario, compras y ventas.</p>							
Resultados/ Acuerdos	¿Quién Solicita?	¿Cuándo?					
Jehová Jireh Valida los resultados del Sprint 2 y se le realiza formal entrega de las nuevas funcionalidades del sistema.	Jehova Jireh	11/01/25					
Se dio la aprobación formal para cerrar el Sprint 2 y dar inicio al Sprint 3 para abordar los nuevos requerimientos y las modificaciones solicitadas.	Jehova Jireh	11/01/25					

Punto 3: Nuevo Requerimiento

Se realiza el levantamiento en conjunto de nuevo requerimiento añadiendo funcionalidades y mejorando las ya existentes.

Resultados/ Acuerdos	¿Quién Solicita?	¿Cuándo?
Se documentaron nuevos requerimientos solicitados por Jehová Jireh, los cuales se agregarán al backlog del proyecto.		
Jehova Jireh solicita cambios en funcionalidades operativas para alinearlas con sus preferencias.	Jehova Jireh	11/01/25
Se acordó realizar los cambios en funcionalidades ya operativas para alinearlas con las preferencias del cliente.		

Ruthian *Hevian* *recibido*

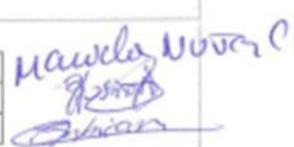
Figura 100: Acta de reunión Sprint 3

 UNI - Ingeniería en Computación	Sistema Web de Inventario y Facturación Jehová Jireh							
Acta de Reunión								
Fecha	Granada, 22 de febrero del 2025							
Lugar	Pulpería Jehová Jireh							
Participantes	Mariela Núñez Cabrera	Propietaria del Negocio						
	Hosmand Rafael Belli Cespedes	Scrum Máster/Developer Team						
	Cristhiam Omar Suárez Núñez	Product Owner/Developer Team						
Puntos del orden del día	01 – Sesión de Pruebas UAT 02 – Revisión de Entregables							
Detalles								
Punto 1: Sesión de Pruebas UAT Se llevó a cabo la sesión de pruebas del Sprint 3 en la que Jehová Jireh interactuó directamente con el sistema para validar las mejoras realizadas y las nuevas funcionalidades implementadas.								
<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left; padding: 5px;">Resultados/Acuerdos</th> <th style="text-align: left; padding: 5px;">¿Quién Solicita?</th> <th style="text-align: left; padding: 5px;">¿Cuándo?</th> </tr> </thead> <tbody> <tr> <td style="padding: 5px;"> Jehová Jireh valida y da por aceptadas las funcionalidades de los módulos entregados, tras la exitosa ejecución de las pruebas UAT. No se reportaron incidencias. El cliente confirmó que las nuevas funcionalidades son de su total agrado. </td> <td style="text-align: left; padding: 5px; vertical-align: top;"> Jehova Jireh </td> <td style="text-align: left; padding: 5px; vertical-align: top;"> 22/02/25 </td> </tr> </tbody> </table>			Resultados/Acuerdos	¿Quién Solicita?	¿Cuándo?	Jehová Jireh valida y da por aceptadas las funcionalidades de los módulos entregados, tras la exitosa ejecución de las pruebas UAT. No se reportaron incidencias. El cliente confirmó que las nuevas funcionalidades son de su total agrado.	Jehova Jireh	22/02/25
Resultados/Acuerdos	¿Quién Solicita?	¿Cuándo?						
Jehová Jireh valida y da por aceptadas las funcionalidades de los módulos entregados, tras la exitosa ejecución de las pruebas UAT. No se reportaron incidencias. El cliente confirmó que las nuevas funcionalidades son de su total agrado.	Jehova Jireh	22/02/25						
Punto 2: Revisión de Entregables Se realiza formal entrega de las funcionalidades completas del Sprint 3 incluyendo las correcciones a los módulos existentes y las nuevas funcionalidades de auditoría.								
<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left; padding: 5px;">Resultados/ Acuerdos</th> <th style="text-align: left; padding: 5px;">¿Quién Solicita?</th> <th style="text-align: left; padding: 5px;">¿Cuándo?</th> </tr> </thead> <tbody> <tr> <td style="padding: 5px;"> Jehová Jireh Valida los resultados del Sprint 3 y se le realiza formal entrega de las nuevas funcionalidades del sistema. Se dio la aprobación formal para cerrar el Sprint 3 y dar inicio al Sprint 4. </td> <td style="text-align: left; padding: 5px; vertical-align: top;"> Jehova Jireh </td> <td style="text-align: left; padding: 5px; vertical-align: top;"> 22/02/25 </td> </tr> </tbody> </table>			Resultados/ Acuerdos	¿Quién Solicita?	¿Cuándo?	Jehová Jireh Valida los resultados del Sprint 3 y se le realiza formal entrega de las nuevas funcionalidades del sistema. Se dio la aprobación formal para cerrar el Sprint 3 y dar inicio al Sprint 4.	Jehova Jireh	22/02/25
Resultados/ Acuerdos	¿Quién Solicita?	¿Cuándo?						
Jehová Jireh Valida los resultados del Sprint 3 y se le realiza formal entrega de las nuevas funcionalidades del sistema. Se dio la aprobación formal para cerrar el Sprint 3 y dar inicio al Sprint 4.	Jehova Jireh	22/02/25						

Figura 101: Acta de reunión Sprint 4

 UNI - Ingeniería en Computación	Sistema Web de Inventario y Facturación Jehová Jireh							
Acta de Reunión								
Fecha	Granada, 12 de abril del 2025							
Lugar	Pulperia Jehová Jireh							
Participantes	Mariela Núñez Cabrera	Propietaria del Negocio						
	Hosmand Rafael Belli Cespedes	Scrum Máster/Developer Team						
	Cristhiam Omar Suarez Núñez	Product Owner/Developer Team						
Puntos del orden del día	01 – Sesión de Pruebas UAT 02 – Revisión de Entregables							
Detalles								
Punto 1: Sesión de Pruebas UAT <p>Se llevó a cabo la sesión de pruebas del sprint 4 en la que Jehová Jireh interactuó directamente con el sistema para validar los reportes implementados en el sistema.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center; background-color: #0070C0; color: white; padding: 5px;">Resultados/Acuerdos</th> <th style="text-align: center; background-color: #0070C0; color: white; padding: 5px;">¿Quién Solicita?</th> <th style="text-align: center; background-color: #0070C0; color: white; padding: 5px;">¿Cuándo?</th> </tr> </thead> <tbody> <tr> <td style="padding: 10px;"> Jehová Jireh valida y da por aceptadas las funcionalidades de los reportes entregados, tras la exitosa ejecución de las pruebas UAT. No se reportaron incidencias. Jehova Jireh expresa su total agrado hacia los reportes implementados en el sistema. </td> <td style="text-align: center; padding: 10px;"> Jehova Jireh </td> <td style="text-align: center; padding: 10px;"> 12/04/25 </td> </tr> </tbody> </table>			Resultados/Acuerdos	¿Quién Solicita?	¿Cuándo?	Jehová Jireh valida y da por aceptadas las funcionalidades de los reportes entregados, tras la exitosa ejecución de las pruebas UAT. No se reportaron incidencias. Jehova Jireh expresa su total agrado hacia los reportes implementados en el sistema.	Jehova Jireh	12/04/25
Resultados/Acuerdos	¿Quién Solicita?	¿Cuándo?						
Jehová Jireh valida y da por aceptadas las funcionalidades de los reportes entregados, tras la exitosa ejecución de las pruebas UAT. No se reportaron incidencias. Jehova Jireh expresa su total agrado hacia los reportes implementados en el sistema.	Jehova Jireh	12/04/25						
Punto 2: Revisión de Entregables <p>Se realiza formal entrega de las funcionalidades completas del Sprint 4 referente al apartado de reportes del sistema.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center; background-color: #0070C0; color: white; padding: 5px;">Resultados/ Acuerdos</th> <th style="text-align: center; background-color: #0070C0; color: white; padding: 5px;">¿Quién Solicita?</th> <th style="text-align: center; background-color: #0070C0; color: white; padding: 5px;">¿Cuándo?</th> </tr> </thead> <tbody> <tr> <td style="padding: 10px;"> Jehová Jireh Valida los resultados del Sprint 4 y se le realiza formal entrega de las nuevas funcionalidades del sistema. Se dio la aprobación formal para cerrar el Sprint 4 y dar inicio al sprint 5. </td> <td style="text-align: center; padding: 10px;"> Jehova Jireh </td> <td style="text-align: center; padding: 10px;"> 12/04/25 </td> </tr> </tbody> </table>			Resultados/ Acuerdos	¿Quién Solicita?	¿Cuándo?	Jehová Jireh Valida los resultados del Sprint 4 y se le realiza formal entrega de las nuevas funcionalidades del sistema. Se dio la aprobación formal para cerrar el Sprint 4 y dar inicio al sprint 5.	Jehova Jireh	12/04/25
Resultados/ Acuerdos	¿Quién Solicita?	¿Cuándo?						
Jehová Jireh Valida los resultados del Sprint 4 y se le realiza formal entrega de las nuevas funcionalidades del sistema. Se dio la aprobación formal para cerrar el Sprint 4 y dar inicio al sprint 5.	Jehova Jireh	12/04/25						

Figura 102: Acta de reunión Sprint 5

 UNI - Ingeniería en Computación		Sistema Web de Inventario y Facturación Jehová Jireh												
Acta de Reunión														
Fecha	Granada, 31 de mayo del 2025													
Lugar	Pulpería Jehová Jireh													
Participantes	<table border="1" style="width: 100%;"> <tr> <td>Mariela Núñez Cabrera</td> <td>Propietaria del Negocio</td> </tr> <tr> <td>Hosmand Rafael Belli Cespedes</td> <td>Scrum Máster/Developer Team</td> </tr> <tr> <td>Cristhiam Omar Suarez Núñez</td> <td>Product Owner/Developer Team</td> </tr> </table> <div style="text-align: right; margin-top: -20px;">  </div>		Mariela Núñez Cabrera	Propietaria del Negocio	Hosmand Rafael Belli Cespedes	Scrum Máster/Developer Team	Cristhiam Omar Suarez Núñez	Product Owner/Developer Team						
Mariela Núñez Cabrera	Propietaria del Negocio													
Hosmand Rafael Belli Cespedes	Scrum Máster/Developer Team													
Cristhiam Omar Suarez Núñez	Product Owner/Developer Team													
Puntos del orden del día	01 – Lanzamiento del Sistema 02 – Entrega de Documentación Técnica													
Detalles														
Punto 1: Lanzamiento del Sistema <p>Se llevaron a cabo los procesos de implantación del sistema Jehová Jireh de manera local.</p> <table border="1" style="width: 100%;"> <thead> <tr> <th>Resultados/Acuerdos</th> <th>¿Quién Solicita?</th> <th>¿Cuándo?</th> </tr> </thead> <tbody> <tr> <td>Jehova Jireh valida la implantación del sistema, el proceso de estabilización y la migración de sus datos al sistema.</td> <td></td> <td></td> </tr> <tr> <td>Jehová Jireh da por aceptada la implantación del sistema en su infraestructura actual de acuerdo con el plan estipulado por el Developer Team.</td> <td>Jehova Jireh</td> <td>31/05/25</td> </tr> <tr> <td>Jehová Jireh solicita que se dé seguimiento del sistema durante el mes de junio del vigente año por cualquier situación que pueda ocurrir.</td> <td></td> <td></td> </tr> </tbody> </table>			Resultados/Acuerdos	¿Quién Solicita?	¿Cuándo?	Jehova Jireh valida la implantación del sistema, el proceso de estabilización y la migración de sus datos al sistema.			Jehová Jireh da por aceptada la implantación del sistema en su infraestructura actual de acuerdo con el plan estipulado por el Developer Team.	Jehova Jireh	31/05/25	Jehová Jireh solicita que se dé seguimiento del sistema durante el mes de junio del vigente año por cualquier situación que pueda ocurrir.		
Resultados/Acuerdos	¿Quién Solicita?	¿Cuándo?												
Jehova Jireh valida la implantación del sistema, el proceso de estabilización y la migración de sus datos al sistema.														
Jehová Jireh da por aceptada la implantación del sistema en su infraestructura actual de acuerdo con el plan estipulado por el Developer Team.	Jehova Jireh	31/05/25												
Jehová Jireh solicita que se dé seguimiento del sistema durante el mes de junio del vigente año por cualquier situación que pueda ocurrir.														
Punto 2: Entrega de Documentación Técnica <p>Se realiza formal entrega de la documentación técnica para el entendimiento y el correcto uso del sistema.</p> <table border="1" style="width: 100%;"> <thead> <tr> <th>Resultados/ Acuerdos</th> <th>¿Quién Solicita?</th> <th>¿Cuándo?</th> </tr> </thead> <tbody> <tr> <td>Jehová Jireh recibe el manual de usuario y el diccionario de datos del sistema.</td> <td></td> <td></td> </tr> <tr> <td>Se dio la aprobación formal para cerrar el Sprint 5, finalizando con todas las fases del proyecto.</td> <td>Jehova Jireh</td> <td>31/05/25</td> </tr> </tbody> </table>			Resultados/ Acuerdos	¿Quién Solicita?	¿Cuándo?	Jehová Jireh recibe el manual de usuario y el diccionario de datos del sistema.			Se dio la aprobación formal para cerrar el Sprint 5, finalizando con todas las fases del proyecto.	Jehova Jireh	31/05/25			
Resultados/ Acuerdos	¿Quién Solicita?	¿Cuándo?												
Jehová Jireh recibe el manual de usuario y el diccionario de datos del sistema.														
Se dio la aprobación formal para cerrar el Sprint 5, finalizando con todas las fases del proyecto.	Jehova Jireh	31/05/25												

Anexo III. Cuestionario

En esta sección, se detalla el cuestionario realizado en las primeras reuniones con la propietaria. El objetivo principal fue la recopilación y documentación de requisitos de manera sistemática y estructurada. El cuestionario funcionó como un mecanismo esencial para el mapeo y optimización de procesos. Esto ayudó a identificar los cuellos de botella y debilidades en los procesos operativos de la pulperia, también capturó sus expectativas de optimización. Finalmente, la información recolectada se tradujo directamente en Historias de Usuario sirviendo como la base sólida y documentada para realizar una estimación precisa del esfuerzo necesario para el desarrollo.

Figura 103: Cuestionario

The screenshot shows a survey form with the following sections and questions:

Información general del negocio

1. ¿Cuántos productos maneja actualmente en su pulperia?
 - Menos de 50
 - Entre 50 y 200
 - Más de 200
2. ¿Cuántas personas utilizarían el sistema en el negocio?
 - Solo yo
 - 2 a 3 personas
 - Más de 3 personas
3. ¿Qué tipo de productos vende principalmente?
 - Alimentos y bebidas
 - Productos de limpieza
 - Artículos de uso diario
 - Todas las anteriores

Inventario

4. ¿Cómo desea clasificar sus productos en el sistema?
 - Por categoría (ej. alimentos, limpieza)
 - Por proveedor
 - Por fecha de vencimiento
 - No necesita clasificación
5. ¿Desea recibir alertas cuando un producto esté por agotarse?
 - Sí
 - No
 - No estoy seguro
6. ¿Con qué frecuencia actualiza su inventario?
 - Diario
 - Semanal
 - Mensual
 - Solo cuando hay reposición

Mauricio Novoa



Facturación

7. ¿Qué tipo de comprobantes necesita emitir?

- Factura con nombre del cliente
- Ticket simple sin datos del cliente
- Ambos

8. ¿Qué formas de pago acepta?

- Solo efectivo
- Efectivo y tarjeta
- Efectivo, tarjeta y crédito a clientes

9. ¿Desea aplicar descuentos o promociones desde el sistema?

- Sí
- No
- Tal vez en el futuro

Reportes y control

10. ¿Qué reportes le interesan más?

- Ventas diarias
- Productos más vendidos
- Ganancias por período
- Todos los anteriores

11. ¿Desea acceder al sistema desde otros dispositivos (celular, tablet)?

- Sí
- No
- Tal vez en el futuro

12. ¿Necesita que el sistema tenga usuarios con diferentes niveles de acceso?

- Sí (administrador, vendedor, etc.)
- No, todos con el mismo acceso
- No estoy seguro

Mauricio Vargas C.

 **Seguridad y acceso**

13. ¿Desea que el sistema tenga inicio de sesión con usuario y contraseña?

- Sí, para todos los usuarios
- Solo para el administrador
- No es necesario

14. ¿Qué nivel de seguridad espera del sistema?

- Básico (contraseña simple)
- Medio (usuarios con roles y permisos)
- Alto (auditoría, bloqueo de intentos, respaldo automático)

15. ¿Desea que el sistema registre quién hizo cada acción (ventas, cambios, etc.)?

- Sí
- No
- Tal vez en el futuro

 **Interfaz y experiencia visual**

16. ¿Qué tipo de diseño visual prefiere?

- Sencillo y funcional
- Colorido y moderno
- Similar a aplicaciones móviles
- No tengo preferencia

17. ¿Qué tan importante es que el sistema sea fácil de usar?

- Muy importante
- Moderadamente importante
- No es prioridad

18. ¿Desea que el sistema tenga vistas adaptadas para celular o tablet?

- Sí
- No
- No es necesario

Hawilio Navío C

 **Mermas****19. ¿Desea registrar las mermas en el sistema?**

- Sí, cada vez que ocurren
- Solo al final del mes
- No es necesario

20. ¿Qué tipo de mermas desea controlar?

- Productos vencidos
- Productos dañados
- Pérdidas por robo o error
- Todas las anteriores

21. ¿Desea que el sistema genere reportes de mermas?

- Sí, con detalle por producto
- No necesito reportes de mermas

Regalías**22. ¿Recibe productos de regalo por parte de sus proveedores?**

- Sí, frecuentemente
- A veces
- No

23. ¿Desea registrar las regalías en el sistema?

- Sí, como entrada de inventario sin costo
- Sí, pero separadas del inventario regular
- No es necesario

24. ¿Qué desea hacer con las regalías?

- Venderlas como cualquier producto
- Usarlas para promociones o regalos a clientes
- No las uso comercialmente

Marcela Núñez
C.

Anexo IV. Estudio de Factibilidad

Durante reuniones con la propietaria, se llevó a cabo un estudio de factibilidad con la información recopilada. Este estudio fue vital para validar la viabilidad del proyecto desde múltiples perspectivas:

Tabla 22: Estudios aplicados

Estudio Aplicado	Descripción
Factibilidad Operativa	Se nos mostró el día a día de la pulperia, permitiéndonos observar cómo se realizan actualmente sus procesos internos, desde la atención al cliente hasta la gestión de inventario. Se analizó la manualidad de sus registros de ventas, compras y mermas, identificando ineficiencias y puntos de mejora.
Factibilidad Técnica	Se observaron los distintos componentes tecnológicos presentes en la infraestructura actual del negocio, evaluando su compatibilidad y la necesidad de nuevas herramientas para soportar el sistema propuesto.
Factibilidad Económica	La fase concluyó con una evaluación preliminar de los costos estimados para la implementación del proyecto, considerando tanto el desarrollo del sistema como la posible adquisición de hardware o software adicional.

Factibilidad Operativa:

Este segmento comenzó con una inmersión profunda en las actividades diarias de la pulperia. Esta etapa fue vital para comprender los procedimientos existentes. A través de estas observaciones y comentarios, pudimos mapear los procesos operativos esenciales.

Tabla 23: Procesos operativos

Procesos operativos	Descripción
Gestión de Inventario y Adquisiciones	Observamos y documentamos el proceso de atender a proveedores y la emisión de órdenes de compra. Esto incluye desde la recepción de mercancía hasta su almacenamiento, identificando el flujo de entrada de productos.
Proceso de Ventas y Atención al Cliente	Analizamos cómo se atienden a los clientes y se generan las ventas, desde la interacción inicial hasta la transacción final. Esto abarca la rapidez del servicio, los métodos de pago, y la experiencia del cliente.
Control y Conteo de Productos	Se nos mostró la complejidad del conteo de los muchos productos según sus características (tamaño, tipo, fecha de caducidad). Este proceso es manual y consume tiempo, además de ser propenso a errores.
Identificación de productos mermados	Uno de los hallazgos más críticos fue la identificación de los productos mermados. La propietaria destacó la recurrente problemática de encontrar productos dañados o vencidos. Este hallazgo no solo representa un costo directo, sino que también pueden indicar ineficiencias en la gestión de inventario, el almacenamiento o la rotación de productos.

En conjunto, esta fase permitió evaluar la capacidad operativa actual de la pulpería, identificar los cuellos de botella y comprender las necesidades no satisfechas por los procesos manuales o la falta de herramientas tecnológicas.

Factibilidad Técnica:

En esta sección se definió la infraestructura disponible, evaluando si estas cumplían con lo adecuado para soportar la implementación del sistema.

Inicialmente, se detalló la infraestructura tecnológica que la pulpería poseía. Esto incluyó hardware, software, conexiones de redes y cualquier otro equipo de relevancia.

Tabla 24: Hardware antiguo

Cantidad	Equipo	Características
1	Computadora Dell 3080 SFF	<ul style="list-style-type: none"> • Intel i3 Segunda Generación • 4 GB de RAM DDR3 • Disco duro de 500 GB
1	Monitor Hacer G185HV	<ul style="list-style-type: none"> • Panta 1366 X 768 px • 60Hz
1	Teclado Royal Kludge RKM75	<ul style="list-style-type: none"> • Tamaño 80% • Mecánico
1	Mouse Logitech G203	<ul style="list-style-type: none"> • DPI 800 • Sensor Lighsync

Tabla 25: Software antiguo

Cantidad	Equipo	Versión	Estado
1	Sistema operativo	<ul style="list-style-type: none"> • Windows 7 	Desactualizado
1	Herramientas ofimáticas	<ul style="list-style-type: none"> • Office 2013 	Desactualizado
1	Antivirus Avast Free	<ul style="list-style-type: none"> • 7.0.1474 	Desactualizado
1	Navegador Internet Explorer	<ul style="list-style-type: none"> • Default 	Desactualizado

Tabla 26: Conexión antigua

Cantidad	Servicios	Características
1	Internet de Fibra Óptica	35 MB TELECABLE

Luego, se detallaron los requisitos de hardware, software y conexiones de redes específicas que serían indispensables para el correcto funcionamiento del sistema web de inventario y facturación.

Tabla 27: Hardware Necesario

Cantidad	Equipo	Características (Mínimas)
1	Impresora térmica	<ul style="list-style-type: none"> • 80mm de impresión.
1	Lector de código de barras	<ul style="list-style-type: none"> • Unidireccional.
1	Computador	<ul style="list-style-type: none"> • Ram 8GB DDR4. • Intel-I5 6ta Generación. • Disco SSD 256GB.
1	Servidor	<ul style="list-style-type: none"> • Ram 8GB DDR4. • Intel-I5 6ta Generación. • Disco SSD 256GB.

Tabla 28: Software necesario

Cantidad	Equipo	Versión (Mínima)
1	Sistema operativo	Windows 10 Pro
1	Navegador Google Chrome	Ultima versión
1	Office365	Office profesional 2019
1	Motor Sql Server	2019
1	McAfee Total	2024

Tabla 29: Conexión necesaria

Cantidad	Servicio	características
1	Hosting ASP.NET	Windows server 2022

Finalizando, con la sección que identificó a los roles necesarios para la implementación del sistema.

Tabla 30: Talento humano

Cantidad	Rol	Función
1	DBA (Administrador De bases de Datos)	<ul style="list-style-type: none"> • Diseñar y estructurar la base de datos, optimizar el almacenamiento. • Analizar los requisitos específicos de la pulpería y las ocupaciones del negocio para adaptar la base de datos a sus necesidades.
1	Desarrollador de Software	<ul style="list-style-type: none"> • Diseñar interfaces de usuario intuitivas y atractivas utilizando tecnologías como HTML, CSS, JavaScript. • Implementar la lógica de negocio y funcionalidad del sistema utilizando lenguajes de programación como C#, JavaScript, entre otros.
1	Analista QA/Tester	<ul style="list-style-type: none"> • Definir y ejecutar casos de prueba y escenarios funcionales que simulen el uso real del software por parte de los usuarios. • Realizar pruebas manuales para identificar defectos y problemas en el software.
1	Scrum Master	<ul style="list-style-type: none"> • Organizar y facilitar todas las ceremonias Scrum, incluyendo las reuniones diarias planificaciones de sprint y revisiones. • Identificar y eliminar impedimentos que puedan afectar el progreso del equipo. • Monitorear el progreso del sprint y ajustar los planes según sea necesario para mantener el ritmo y la dirección.
1	Product Owner	<ul style="list-style-type: none"> • Definir y priorizar el backlog del producto, asegurando que las historias de usuario y los requisitos estén claramente especificados. • Colaborar con los interesados para entender sus necesidades y traducirlas en funcionalidades del producto.

Factibilidad Económica:

La factibilidad económica de este proyecto evaluó si la implementación del sistema web era una inversión justificada para la pulpería, considerando los recursos tangibles e intangibles que se destinarán. El análisis se enfocó en los costos operativos y la inversión en equipamiento y en los beneficios que el sistema aportaría al negocio. Todos los costos detallados en este apartado están expresados en moneda nacional.

Este estudio comenzó detallando la estimación de costos del proyecto.

Tabla 31: Costos de hardware [23]

Cantidad	Equipo	Costo
1	Impresora térmica	C\$ 3,660.00
1	Lector de código de barras	C\$ 1,200.00
1	Computador	C\$ 13,800.00
Subtotal		C\$ 18,660.00

Tabla 32: Costos de software [24]

Cantidad	Equipo	Costo
1	Windows 11 Pro, 5 dispositivos	C\$ 3,900.00
1	Navegador Google Chrome	C\$ 0.00
1	Office 365, 5 dispositivos	C\$ 815.00
1	SQL Server	C\$0.00
1	McAfee Total, 5 dispositivos	C\$ 1,715.00
1	PowerBI Desktop	C\$ 590.00
Subtotal		C\$ 7,020.00

Tabla 33: Costos de conexiones de redes [25]

Cantidad	Equipo	Costo
1	Hosting ASP.NET	C\$150.00
Subtotal		C\$150.00

Los costos plasmados en este apartado fueron cuidadosamente investigados y validados a partir de fuentes con sede nacional. Se utilizaron datos de sitios web de empresas reconocidas y confiables, lo que asegura la precisión y pertinencia de los precios para el mercado local.

Los costos asociados al talento humano son considerados una inversión de tiempo y conocimiento. Esto no representa un desembolso monetario directo para la pulperia.

Tabla 34: Costos de talento humano del proyecto

Cantidad	Rol	Tiempo	Costo
1	DBA (Administrador de Bases de Datos)	Según Product Backlog	C\$0.00
1	Desarrollador de Software	Según Product Backlog	C\$0.00
1	Oficial QA/Tester	Según Product Backlog	C\$0.00
1	Scrum Master	Según Product Backlog	C\$0.00
1	Product Owner	Según Product Backlog	C\$0.00
Subtotal			\$0.00

Si bien los costos directos son de C\$ 0.00, es importante destacar el valor de mercado de este equipo de trabajo. Basado en salarios promedio, el costo estimado de adquirir estos perfiles podría ascender a lo siguiente:

Tabla 35: Costos de talento humano en el mercado

Cantidad	Rol	Tiempo	Costo
1	DBA (Administrador de Bases de Datos)	Según Product Backlog	C\$24,000.00
1	Desarrollador de Software	Según Product Backlog	C\$26,000.00
1	Oficial QA/Tester	Según Product Backlog	C\$23,000.00
1	Scrum Master	Según Product Backlog	C\$24,000.00
1	Product Owner	Según Product Backlog	C\$24,000.00
Subtotal			C\$121,000.00

Como muestran las tablas, el talento humano asignado al proyecto tiene un costo de C\$ 0.00, una cifra incomparable a los costos promedio del mercado para roles similares. Este ahorro sustancial se traduce en un beneficio invaluable para Jehová Jireh. En lugar de destinar recursos a la mano de obra, se mostró la posibilidad de reinvertir ese capital en su propio crecimiento y expansión.

A pesar de la mínima inversión monetaria directa en desarrollo, el sistema aporta beneficios económicos y operativos significativos a la pulperia.

➤ Reducción de pérdidas y errores

- Gestión eficaz de mermas: El sistema permite un control preciso sobre los productos dañados o vencidos, esto se identificó en la fase inicial por lo que, esto resulta en una reducción directa de pérdidas para el negocio.
- Precisión de inventario: La automatización del inventario eliminó errores manuales, evitando la pérdida de productos y optimizando el stock, lo que se traduce en un mejor aprovechamiento del capital.

➤ Aumento de la eficiencia operativa

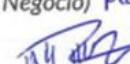
- Agilización de procesos: La optimización de ventas y la generación de órdenes de compra redujeron el tiempo dedicado a tareas administrativas, permitiendo a la propietaria y al personal enfocarse más en la atención al cliente y otras labores productivas.
- Optimización del tiempo: El sistema disminuyó el tiempo que se invierte en el conteo manual de productos.

➤ Transformación digital y continuidad del negocio

- Información oportuna: El acceso a reportes detallados y en tiempo real permite a la propietaria tomar decisiones más estratégicas y rápidas sobre precios, costos y reposición de inventario.
- Profesionalización del negocio: Un sistema automatizado proyecta una imagen más moderna y eficiente de la pulperia, lo que puede atraer y retener a más clientes.

Anexo V. Requerimiento inicial

Figura 104: Requerimiento Oficial con Historias de Usuario

REQUERIMIENTO DE USUARIO	
I. DATOS DEL REQUERIMIENTO	
Nombre del requerimiento:	Sistema Web Jehová Jireh
Fecha de solicitud:	Septiembre 2024
Descripción del requerimiento:	Desarrollar un Sistema web para el control de inventario y facturación de la pulperia Jehová Jireh, adaptándose a la estructura y operaciones actuales.
Usuarios:	Mariela Núñez Cabrera (Propietaria del Negocio)  Ricardo Rivera Rivas (Administrador) 
Responsables:	Br. Cristhiam Omar Suárez Núñez  Br. Hosmand Rafael Belli Cespedes 

II. DESCRIPCIÓN DEL REQUERIMIENTO

La pulperia "Jehová Jireh" es un negocio ubicado en Diriá, departamento de Granada, dedicado a la venta de diversos productos de consumo diario, como granos básicos, bebidas, lácteos, carnes, entre otros.

Jehová Jireh, representa un punto de abastecimiento fundamental para el sector aledaño. A lo largo de los años, la pulperia ha carecido de un sistema de información que agilice las operaciones y procedimientos en términos de control de inventario y ventas.

En consecuencia, se requiere implementar un sistema de inventario y facturación para gestionar correctamente el inventario del negocio, evitando el control impreciso de los productos en existencia, la demora en la emisión de facturas, malos manejos de los productos y los errores en los cálculos de precios y descuentos. La implementación de este sistema brindara beneficios significativos en la pulperia, disminuirán los tiempos de venta y mejorará la atención de cara al cliente.



III. ALCANCE DEL REQUERIMIENTO

Se necesita implementar un sistema web que proporcione intervalos de tiempo rápidos en los procesos internos de la pulpería, como la búsqueda de los productos existentes, facturación de productos a clientes, generación de órdenes de compras a proveedores, etc. A su vez, el sistema deberá tener un diseño simple, de rápida comprensión e intuitivo, que cualquier persona con conocimientos básicos de informática sea capaz de usarlo. Además de ser seguro para garantizar la integridad de los datos dentro del sistema.

Autenticación y Roles:

- Acceso al Sistema
 - Como usuario, quiero iniciar sesión en el sistema con un usuario y contraseña para acceder a las funcionalidades de acuerdo con mi rol.
 - Como usuario, quiero poder recuperar mi contraseña si la olvido para no perder el acceso al sistema.
- Gestión de Roles y Permisos
 - Como administrador, quiero asignar roles a los usuarios para definir sus permisos y prohibiciones.
 - Como administrador, quiero tener acceso total a todas las funcionalidades del sistema para poder gestionar usuarios, roles y cualquier configuración necesaria.
- Funcionalidades de Vendedor
 - Como vendedor, quiero poder generar ventas y órdenes de compra para realizar mis tareas diarias de manera eficiente.
 - Como vendedor, quiero acceder únicamente a mis propios registros de ventas y órdenes de compra para visualizar mi rendimiento personal.

Módulos de trabajo del sistema:

Usuario: Se considerará usuario al acceso que tenga un trabajador de la pulperia al sistema.

Dentro de las funcionalidades se necesita que el sistema permita:

- Crear un nuevo usuario
 - Como administrador, quiero crear una nueva cuenta de usuario para que un nuevo empleado pueda acceder al sistema.
 - Detalles: El sistema debe permitir capturar el nombre de usuario, correo, teléfono, dirección y una contraseña inicial.
- Editar la información de un usuario
 - Como administrador, quiero poder editar los datos de un usuario existente para mantener su información actualizada.
 - Detalles: Los campos editables deben ser correo, teléfono, dirección y contraseña.
- Inactivar un usuario
 - Como administrador, quiero poder inactivar una cuenta de usuario para bloquear su acceso al sistema por el motivo que sea.
 - Detalles: Inactivar un usuario debe impedir que inicie sesión, pero sus datos deben conservarse con fines históricos.

Cliente: Se considera cliente a quien adquiere los productos o servicios de la pulperia.

Puede tratarse de un cliente recurrente cuando este realiza compras regulares o un cliente no habitual. Dentro de las funcionalidades se necesita que el sistema permita:

- Crear un nuevo cliente
 - Como administrador de la pulperia, quiero crear un registro para un nuevo cliente para tener una base de datos de clientes recurrentes y no habituales.
 - Detalles: El sistema debe permitir capturar el nombre, cédula, teléfono y dirección del cliente.

- Editar la información de un cliente
 - Como administrador de la pulperia, quiero poder editar la información de un cliente existente para mantener sus datos de contacto actualizados.
 - Detalles: Se deben poder modificar el nombre, cédula, teléfono y dirección del cliente.
- Inactivar a un cliente
 - Como administrador de la pulperia, quiero poder inactivar a un cliente para gestionar adecuadamente la lista de mis clientes.
 - Detalles: Inactivar a un cliente no lo borra del sistema, pero lo marca como inactivo por el motivo que sea.

Empresa: Una empresa es la que surte los productos para Jehová Jireh, una empresa puede tener varios productos asociados lo que genera en el cliente confianza y sugestiona la compra del producto asociado. Dentro de las funcionalidades se necesita que el sistema permita:

- Crear una nueva Empresa
 - Como administrador de la pulperia, quiero crear un registrar una nueva empresa y poder asociar los productos con su marca o compañía de origen.
 - Detalles: El sistema debe permitir registrar el nombre de la empresa, su estado, fecha de registro y el identificador del proveedor.
- Editar una Empresa
 - Como administrador de la pulperia, quiero poder editar la información de una Empresa existente para mantener sus datos actualizados.
- Inactivar una Empresa
 - Como administrador de la pulperia, quiero poder inactivar una Empresa para eliminar aquellas que ya no surten productos a la pulperia.

Producto: Los productos son aquellos elementos que conforman la oferta de la pulpería como la variedad de alimentos y artículos de consumo diario. Dentro de las funcionalidades se necesita que el sistema permita:

- Crear un nuevo producto
 - Como administrador de la pulpería, quiero registrar un nuevo producto en el sistema para tener un control exacto de las existencias disponibles.
 - Detalles: El sistema debe permitir capturar los siguientes datos: nombre, imagen, categoría, unidad de medida, marca, fecha de vencimiento, costo, cantidad, código y precio de venta.
- Editar la información de un producto
 - Como administrador de la pulpería, quiero poder editar los datos de un producto existente para mantener la base de datos del inventario siempre actualizada.
 - Detalles: Se deben poder modificar todos los campos de un producto: nombre, imagen, categoría, unidad de medida, marca, fecha de vencimiento, costo, cantidad, código y precio de venta.
- Inactivar un producto
 - Como administrador de la pulpería, quiero poder inactivar un producto para retirar del catálogo aquellos artículos que ya no se venden o están vencidos.
 - Detalles: Inactivar un producto lo marca como no disponible para la venta, pero su registro se mantiene en el sistema.

Proveedor: Un proveedor es aquel representante de una empresa con el objetivo de abastecer el inventario de Jehová Jireh, por lo general, al proveedor se le compra diversos productos con un margen de ganancias significativo para la pulperia. Dentro de las funcionalidades se necesita que el sistema permita:

- Crear un nuevo proveedor
 - Como administrador de la pulperia, quiero crear un registro para un nuevo proveedor para tener una base de datos de mis contactos para futuras compras.
 - Detalles: El sistema debe permitir capturar el ID del proveedor, nombre, apellido, teléfono, estado y fecha de registro.
- Editar la información de un proveedor
 - Como administrador de la pulperia, quiero poder editar los datos de un proveedor existente para mantener su información de contacto actualizada.
 - Detalles: Se deben poder modificar el nombre, apellido, teléfono y estado del proveedor.
- Inactivar un proveedor
 - Como administrador de la pulperia, quiero poder inactivar a un proveedor para gestionar adecuadamente mi lista de contactos.

Categoría: Son grupos o clases en los que se clasifican nuestros productos. Las categorías se utilizan para organizar los productos de una manera lógica, lo cual hace más fácil la tarea de identificarlos cuando el mismo está asociado a una categoría. Dentro de las funcionalidades se necesita que el sistema permita:

- Crear una nueva categoría
 - Como administrador de la pulperia, quiero crear nuevas categorías de productos para mantener el inventario organizado y facilitar su búsqueda.
 - Detalles: El sistema debe permitir definir el nombre de la categoría.
- Editar una categoría
 - Como administrador de la pulperia, quiero poder editar el nombre de una categoría existente para corregir errores o actualizar su descripción.
 - Detalles: Se debe poder modificar el nombre de la categoría.
- Inactivar una categoría
 - Como administrador de la pulperia, quiero poder inactivar cualquier categoría que no se esté utilizando.
 - Detalles: Inactivar una categoría la marca como no disponible para la clasificación de productos.

Unidades de medida: Son las magnitudes en las cuales los productos de una empresa están divididos. Dentro de las funcionalidades se necesita que el sistema permita:

- Crear una nueva unidad de medida
 - Como administrador de la pulperia, quiero crear nuevas unidades de medida para poder registrar la magnitud de los productos del inventario de forma precisa.
 - Detalles: El sistema debe permitir definir el nombre de la unidad (ej. Gramo, Kilogramo) y su magnitud.
- Editar una unidad de medida
 - Como administrador de la pulperia, quiero poder editar una unidad de medida existente para corregir errores o actualizar su información.
 - Detalles: Se deben poder modificar el nombre de la unidad y su magnitud.

- Inactivar una unidad de medida
 - Como administrador de la pulperia, quiero poder inactivar una unidad de medida para eliminar del catálogo aquellas que ya no se utilizan.
 - Detalles: Inactivar una unidad de medida la marca como no disponible para su uso en nuevos productos

Ventas: se define un módulo de facturación para registrar una transacción de venta entre la pulperia y el cliente, dentro de las funcionalidades se necesita que el sistema permita:

- Crear y gestionar una nueva venta
 - Como vendedor de la pulperia, quiero registrar una nueva venta y agregar productos para procesar la compra de un cliente de forma eficiente y precisa.
 - Detalles:
 - La venta debe incluir el número de venta, la fecha, el cliente y la lista de productos.
 - Antes de finalizar, debo poder agregar, quitar y ajustar la cantidad de los productos en la factura.
 - El sistema debe calcular automáticamente el precio total y el cambio a dar.
- Finalizar una venta
 - Como encargado de la pulperia, quiero poder procesar una venta para completar la transacción y generar la factura.
- Consultar historial de ventas
 - Como encargado de la pulperia, quiero acceder a un historial de ventas realizadas para consultar transacciones pasadas en caso de ser necesario.
- Reimprimir una factura
 - Como encargado de la pulperia, quiero poder volver a imprimir una factura para darle una copia al cliente que lo solicite.

Compra: Es el documento que se utiliza para registrar una transacción de compra entre la pulperia y el proveedor, dentro de las funcionalidades se necesita que el sistema permita:

- Crear y gestionar una nueva compra
 - Como encargado de la pulperia, quiero crear una nueva orden de compra para un proveedor para reabastecer mi inventario de forma organizada.
 - Detalles:
 - La orden debe incluir el número de compra, la referencia, la fecha, el proveedor, los productos, sus costos unitarios y el costo total.
 - Debo poder agregar, quitar y ajustar la cantidad de productos antes de procesar la compra.
- Procesar una compra
 - Como encargado de la pulperia, quiero poder procesar una compra para finalizar la transacción con el proveedor y actualizar mi inventario automáticamente.
- Ver historial de compras
 - Como encargado de la pulperia, quiero ver un historial de las compras realizadas para llevar un control de mis gastos y las entradas de productos.

Merma: Son las pérdidas o desperdicios que se producen durante el almacenamiento de un producto. Las mermas pueden ser causadas por una variedad de factores, como la obsolescencia, la caducidad o el daño físico, dentro de las funcionalidades se necesita que el sistema permita:

- Registrar una merma de producto
 - Como encargado de la pulperia, quiero registrar una merma de un producto para mantener un control exacto y documentar las pérdidas.
 - Detalles: El registro debe incluir una descripción o justificación, una referencia, el producto afectado, la fecha y la cantidad.

- Editar un registro de merma
 - Como encargado de la pulperia, quiero poder editar un registro de merma existente para corregir cualquier error en la información registrada.
- Inactivar un registro de merma
 - Como encargado de la pulperia, quiero poder inactivar un registro de merma para anularlo en caso de que el producto se devuelva a un proveedor.

Regalía: Son los obsequios por compra que ofrecen los proveedores por la adquisición de ciertas ofertas o promociones de productos, estos regalos por lo general son productos que entran al inventario y con los cuales obtienen ganancias extras, dentro de las funcionalidades se necesita que el sistema permita:

- Registrar una regalía de producto
 - Como encargado de la pulperia, quiero registrar una regalía de producto para mantener un control exacto de los artículos que recibo sin costo.
 - Detalles: El registro debe incluir una descripción o justificación, una referencia, el producto, la fecha y la cantidad.
- Editar un registro de regalía
 - Como encargado de la pulperia, quiero poder editar un registro de regalía existente para corregir cualquier error en la información registrada.
- Inactivar un registro de regalía
 - Como encargado de la pulperia, quiero poder inactivar un registro de regalía para eliminarlo si se ingresó por error o ya no es válido.

Reportería

Se requieren reportes generales y específicos de cada uno de los módulos del sistema, los cuales servirán para que se lleve un control estadístico del negocio.

Reportes categorías:

- Como administrador de la pulpería, quiero generar un reporte de categorías para ver cuáles están activas e inactivas y auditar su estado.

Reportes empresas:

- Como administrador de la pulpería, quiero generar un reporte de marcas para ver cuáles están activas e inactivas en mi catálogo de empresas.

Reporte productos:

- Como administrador de la pulpería, quiero generar reportes detallados de productos para obtener una visión completa de mi inventario por diferentes criterios como marca, categoría, nombre o fecha de caducidad.

Reporte de mermas:

- Como administrador de la pulpería, quiero generar un reporte de mermas para analizar las pérdidas de inventario por producto y la justificación de cada una.

Reporte de regalías:

- Como administrador de la pulpería, quiero generar un reporte de regalías para tener un registro de los productos que se reciben sin costo y por producto.

Reporte de unidades de medida:

- Como administrador de la pulpería, quiero generar un reporte de unidades de medida para ver su estado (activas/inactivas) y buscarlas por su descripción.

Reporte de clientes:

- Como administrador de la pulperia, quiero generar un reporte de clientes para analizar mi base de clientes activos e inactivos y buscarlos por nombre.

Reporte de proveedor:

- Como administrador de la pulperia, quiero generar un reporte de proveedores para analizar el estado de mis proveedores y buscarlos por nombre.

Reporte de usuarios:

- Como administrador de la pulperia, quiero generar un reporte de usuarios para ver quiénes están activos e inactivos y filtrar por nombre o por su rol asignado.

Reporte de compras:

- Como administrador de la pulperia, quiero generar reportes de compras para analizar mis adquisiciones por identificador, proveedor o por los productos comprados.

Reporte de ventas:

- Como administrador de la pulperia, quiero generar reportes de ventas para analizar mis ingresos por identificador, cliente o por los productos vendidos.

Expectativas de Calidad del Sistema:

Se requiere que el sistema de inventario y facturación que se desarrolle cumpla con una serie de características de calidad esenciales. Es fundamental que el personal de la pulperia encuentre el sistema altamente usable, lo que implica que sea fácil de aprender y operar, incluso para aquellos con mínima experiencia tecnológica.

Además, se espera un rendimiento óptimo, con tiempos de respuesta rápidos para las operaciones críticas, como el registro de una venta, para asegurar que el flujo de trabajo no se vea afectado.

Además, el sistema deberá contar con medidas robustas para proteger toda la información sensible de ventas, inventario y clientes. Finalmente, es crucial que el sistema sea confiable, garantizando su acceso y operatividad de manera constante para no interrumpir las operaciones diarias.

IV. CRITERIOS DE ACEPTACIÓN Y APROBACIÓN

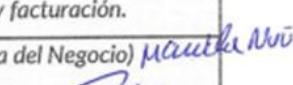
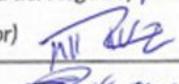
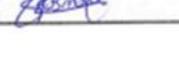
N.º	Descripción	Criterios de Aceptación
1	Módulo del sistema	El sistema deberá garantizar el correcto funcionamiento de cada uno de los módulos descritos.
2	Reportería por módulos	Cada reporte mencionado deberá funcionar correctamente según el propósito descrito.
3	Criterios de calidad del sistema	El sistema deberá garantizar cada una de las expectativas de calidad descritas en el requerimiento.

V. CONTROL DE CAMBIO

Fecha	Funcionario que lo solicita	Descripción

Anexo VI. Requerimiento sprint 3

Figura 105: Requerimiento sprint 3 con Historias de Usuario

REQUERIMIENTO DE USUARIO	
I. DATOS DEL REQUERIMIENTO	
Nombre del requerimiento:	Mejoras y Nuevas Funcionalidades para el Sistema de Jehová Jireh
Fecha de solicitud:	Enero 2025
Descripción del requerimiento:	Mejoras e implementación de nuevas funcionalidades para el sistema web de inventario y facturación.
Usuarios:	Mariela Núñez Cabrera (Propietaria del Negocio)  Ricardo Rivera Rivas (Administrador) 
Responsables:	Br. Cristhiam Omar Suárez Núñez  Br. Hosmand Rafael Belli Cespedes 
II. DESCRIPCIÓN DEL REQUERIMIENTO	
La pulperia "Jehová Jireh" solicita la corrección e implementación de nuevas funcionalidades para el sistema web de inventario y facturación.	
En lineamiento con los estándares máximos de la calidad de software, se hace imperativo realizar mejoras en los módulos ya existentes en el sistema con el fin de adaptar al máximo nuestras necesidades, así como la incorporación de funcionalidades clave que fortalecerán la seguridad y optimizarán la gestión de inventario y ventas.	
Con el propósito de mejorar la usabilidad, se requiere la optimización de los flujos de trabajo en los módulos de login, productos y ventas. Asimismo, se ha priorizado la creación de un robusto módulo de registro y auditoría que permitirá garantizar la transparencia total en las operaciones, facilitando la trazabilidad de cada acción, inicio de sesión y error del sistema, lo cual es fundamental para el control y la toma de decisiones estratégicas.	



III. ALCANCE DEL REQUERIMIENTO

Mejoras y Optimización de Módulos Existentes:

Módulo de Login:

Como parte de la funcionalidad del login, quiero que se reduzca la cantidad de intentos de inicio de sesión a tres, para mejorar la seguridad del sistema.

Dashboard:

Como administrador, quiero ver los totales de ventas del día, semana y mes en el Dashboard, para tener una visión inmediata del rendimiento del negocio.

Como administrador, quiero ver un listado de productos a punto de vencer y ya vencidos en el Dashboard, para tomar acciones correctivas de manera proactiva y evitar pérdidas.

Como usuario, quiero ver mi nombre en el Dashboard, para confirmar que mi sesión está activa y tener una experiencia más personalizada.

Modulo Mermas:

Como usuario, quiero ver la cantidad total de un producto en stock al registrar una merma, para asegurarme de que no estoy registrando una cantidad mayor a la disponible en el inventario.

Modulo Productos:

Como usuario, quiero poder escanear el código de barras de un producto, para agilizar y evitar errores en el ingreso de nuevos productos al inventario.

Como usuario, quiero ingresar productos al inventario en fracciones, para manejar de forma precisa productos a granel como granos o perecederos.

Como administrador, quiero ver el estado del stock de los productos con colores (Vacio, Escaso, Balanceado, Sobrecargado), para identificar rápidamente la necesidad de resurtido y optimizar el inventario.



Módulo de Ventas:

Como usuario, quiero que el filtro de productos vencidos se elimine, para que la información se gestione desde el Dashboard y no sea redundante.

Como usuario, quiero poder buscar productos por código de barras o por empresa, para encontrar rápidamente el artículo deseado en el catálogo.

Como usuario, quiero que el sistema restrinja la venta en fracciones si el producto no lo permite, para evitar errores y garantizar la integridad de las ventas.

Como usuario, quiero poder buscar facturas por rango de fechas, para encontrar y consultar ventas pasadas de manera eficiente.

Módulo de Compras:

Como usuario, quiero seleccionar el nombre de la empresa al registrar una compra, en lugar del nombre del proveedor, para estandarizar el proceso y facilitar la búsqueda.

Como usuario, quiero poder buscar compras por rango de fechas, para revisar y auditar el historial de compras de manera sencilla.

Nuevas Funcionalidades para implementar en el sistema:

Módulo de Registro y Auditoría

Como administrador, quiero que el sistema registre cada actividad de los usuarios, para tener un historial completo y garantizar la transparencia y el control de las operaciones.

La información debe registrarse de la siguiente manera:

- Correo del Usuario
- Nombre del Usuario
- Módulo afectado
- Acción realizada
- Detalle del movimiento
- Fecha de acción



Como administrador, quiero que el sistema registre los inicios de sesión de los usuarios, para monitorear la seguridad y detectar actividades sospechosas.

La data debe registrarse de la siguiente manera:

- Correo del usuario
- Nombre del usuario
- Criterio de Éxito (Exitoso/Fallido)
- Detalle del movimiento
- Dirección IP
- Navegador
- Fecha de Acción

Como administrador, quiero que el sistema me alerte cuando un "Usuario Desconocido" intente iniciar sesión, para responder rápidamente a posibles amenazas de seguridad.

Acción: Se debe enviar un correo de advertencia a los administradores del sistema.

Como administrador, quiero que el sistema registre los errores, para tener un registro detallado de fallas que me ayude a diagnosticar y resolver problemas.

La data debe registrarse de la siguiente manera:

- Nombre de usuario
- Mensaje de error
- Detalle del mensaje
- Módulo afectado
- Acción realizada
- Fecha de acción

IV. CRITERIOS DE ACEPTACIÓN Y APROBACIÓN

N.º	Descripción	Criterios de Aceptación
1	funcionalidad de Login	El sistema debe restringir el acceso del usuario después de 3 intentos fallidos consecutivos en un periodo de 5 minutos, mostrando el mensaje de error correspondiente.
2	Funcionalidad del Dashboard	La pantalla de inicio debe mostrar el total de ventas del día, semana y mes, y esta información debe actualizarse en tiempo real.
3	Módulo de Mermas	El campo de cantidad de producto a mermar no debe permitir ingresar un valor superior al stock actual disponible para ese producto.
4	Módulo de Productos	<p>La funcionalidad de escaneo de código de barras debe considerarse una característica más en los productos.</p> <p>El sistema debe permitir el registro de productos con unidades fraccionadas (ej. 2.5 kg, 0.75 L) para aquellos productos marcados como 'granel'.</p> <p>El listado de productos debe mostrar un indicador visual (color y nombre) que refleje su estado de stock (Vacio, Escaso, Balanceado, Sobrecargado) basado en los umbrales predefinidos.</p>
5	Módulo de Ventas	<p>La barra de búsqueda de productos debe aceptar entradas de texto para buscar por código de barras y nombre de la empresa asociada.</p> <p>El sistema debe restringir la venta de cantidades fraccionadas para productos cuya unidad de medida no lo permita.</p> <p>La funcionalidad de búsqueda de facturas debe permitir al usuario seleccionar un rango de fechas para filtrar los resultados de forma precisa.</p>

6	Módulo de Compras	<p>El campo de selección del proveedor debe mostrar el nombre de la empresa a la que representa, y no el nombre del proveedor.</p> <p>La búsqueda de compras realizadas debe incluir un filtro que permita al usuario seleccionar un rango de fechas para ver el historial.</p>
7	Módulo de auditoría	<p>El sistema debe registrar de forma automática todas las acciones (crear, editar, eliminar) de cualquier usuario, capturando el correo, el módulo afectado, el detalle del movimiento y la fecha.</p> <p>Si un usuario no registrado intenta iniciar sesión, el sistema debe enviar una notificación por correo electrónico a los administradores, identificando el intento como de un "Usuario Desconocido".</p> <p>Cualquier error del sistema debe ser capturado y guardado en un log con el nombre de usuario, el módulo afectado, el mensaje de error y la fecha de ocurrencia.</p>

V. CONTROL DE CAMBIO

Fecha	Funcionario que lo solicita	Descripción

Anexo VII. Sprint 3

El sprint 3 representó un ciclo de trabajo crucial de cinco semanas, un período extendido para asegurar que se pudieran atender plenamente los requerimientos levantados durante la revisión del sprint anterior. El objetivo principal fue optimizar la experiencia del usuario y fortalecer la seguridad del sistema con la implementación de un sistema de auditoría integral.

Las tareas se centraron en dos áreas clave del desarrollo. Primero, se optimizó la usabilidad y la eficiencia de los módulos existentes. Se mejoró la visualización de datos en el dashboard para ofrecer métricas en tiempo real, se integró la lectura de códigos de barra en el módulo de Productos y se habilitó la venta y el ingreso de productos a granel en fracciones. Además, se implementaron filtros de búsqueda por fechas en las facturas de ventas y compras para facilitar la auditoría.

Asimismo, se dio prioridad al desarrollo de un sólido registro de auditoría, abordando un punto clave de los requerimientos de seguridad y transparencia del sistema. Se construyeron tres módulos fundamentales:

- Registro de Actividades: Una herramienta para garantizar la trazabilidad de cada acción dentro del sistema.
- Registro de Sesiones: Una funcionalidad crítica para auditar los accesos de los usuarios.
- Registro de Errores: Un mecanismo proactivo para la detección, diagnóstico y resolución de fallas.

Tabla 36: Product Backlog del Sprint 3

N.	Rol	Funcionalidad	Resultado	Prioridad	Puntos
1	Usuario	Reducir cantidad de intentos de sesión	Mejorar la seguridad del sistema	Baja	2
2	Administrador	Total ventas en Dashboard	Visión inmediata del rendimiento del negocio	Media	3
3	Administrador	Listado de productos a punto de vencer y ya vencidos en Dashboard	Acciones correctivas de manera proactiva y evitar pérdidas	Media	3
4	Usuario	Validar si estoy en sesión	Sesión está activa y tener una experiencia más personalizada.	Baja	2
5	Usuario	Cantidad total de un producto en stock al registrar una merma	Asegurar el correcto ingreso según el stock en el inventario.	Baja	3
6	Usuario	Escanear el código de barras	Agilizar y evitar errores en el ingreso de nuevos productos al inventario	Alta	4
7	Usuario	ingresar productos al inventario en fracciones	Bloquear el acceso al sistema a empleados inactivos o retirados.	Alta	4
8	Administrador	Estado del stock de los productos con colores	Identificar rápidamente la necesidad de resurtido y optimizar el inventario.	Alta	3
9	Administrador	Restringir la venta en fracciones si el producto no lo permite	Evitar errores y garantizar la integridad de las ventas.	Alta	5
11	Administrador	Buscar ventas y compras por rango de fechas	Consultar ventas y compras pasadas de manera eficiente	Media	4

N.	Rol	Funcionalidad	Resultado	Prioridad	Puntos
12	Administrador	Registrar actividad de usuarios	Tener un historial completo y garantizar la transparencia de las operaciones.	Alta	4
13	Administrador	Registrar inicios de sesión	Monitorear la seguridad y detectar actividades sospechosas	Alta	4
14	Administrador	Alertar de un "Usuario Desconocido" intentando iniciar sesión	Responder rápidamente a posibles amenazas de seguridad	Alta	5
15	Administrador	Registrar Errores	Registro detallado de fallas que me ayude a diagnosticar y resolver problemas	Alta	4

Tabla 37: Sprint Backlog mejora de funcionalidades

ORIGEN	ROL	ASIGNADO	ACCION	ESTADO	PRIORIDAD	HORAS
LOGIN	Developer	Cristhiam Omar Suarez Nuñez	Reducir cantidad de intentos en Login	Completada	Baja	2.00
DASHBOARD	Developer	Cristhiam Omar Suarez Nuñez	Agregar Nombre de usuario en sesión	Completada	Baja	2.00
DASHBOARD	Developer	Cristhiam Omar Suarez Nuñez	Mejoras al Dashboard inicial	Completada	Alta	15.00
MERMA	Developer	Hosmand Rafael Belli Cespedes	Agregar el Stock actual del producto a mermar	Completada	Media	3.00
PRODUCTO	Developer	Hosmand Rafael Belli Cespedes	Agregar Funcionalidad de Codigo de Barras	Completada	Alta	15.00
PRODUCTO	Developer	Hosmand Rafael Belli Cespedes	Permitir/Restringir la funcionalidad de venta por fracción	Completada	Alta	8.00
VENTA	Developer	Hosmand Rafael Belli Cespedes	Agregar filtro de fechas en la búsqueda de facturas realizadas	Completada	Media	4.00
COMPRA	Developer	Hosmand Rafael Belli Cespedes	Agregar filtro de fechas en la búsqueda de Compras realizadas	Completada	Media	4.00
PRODUCTO	Developer	Cristhiam Omar Suarez Nuñez	Ingresar productos al inventario en fracciones	Completada	Alta	8.00
PRODUCTO	Developer	Cristhiam Omar Suarez Nuñez	Agregar funcionalidad de estados según el stock	Completada	Media	4.00
PROVEEDOR	Developer	Cristhiam Omar Suarez Nuñez	Reemplazar la selección del nombre del proveedor	Completada	Baja	4.00

Figura 106: Reducción de intentos de Login

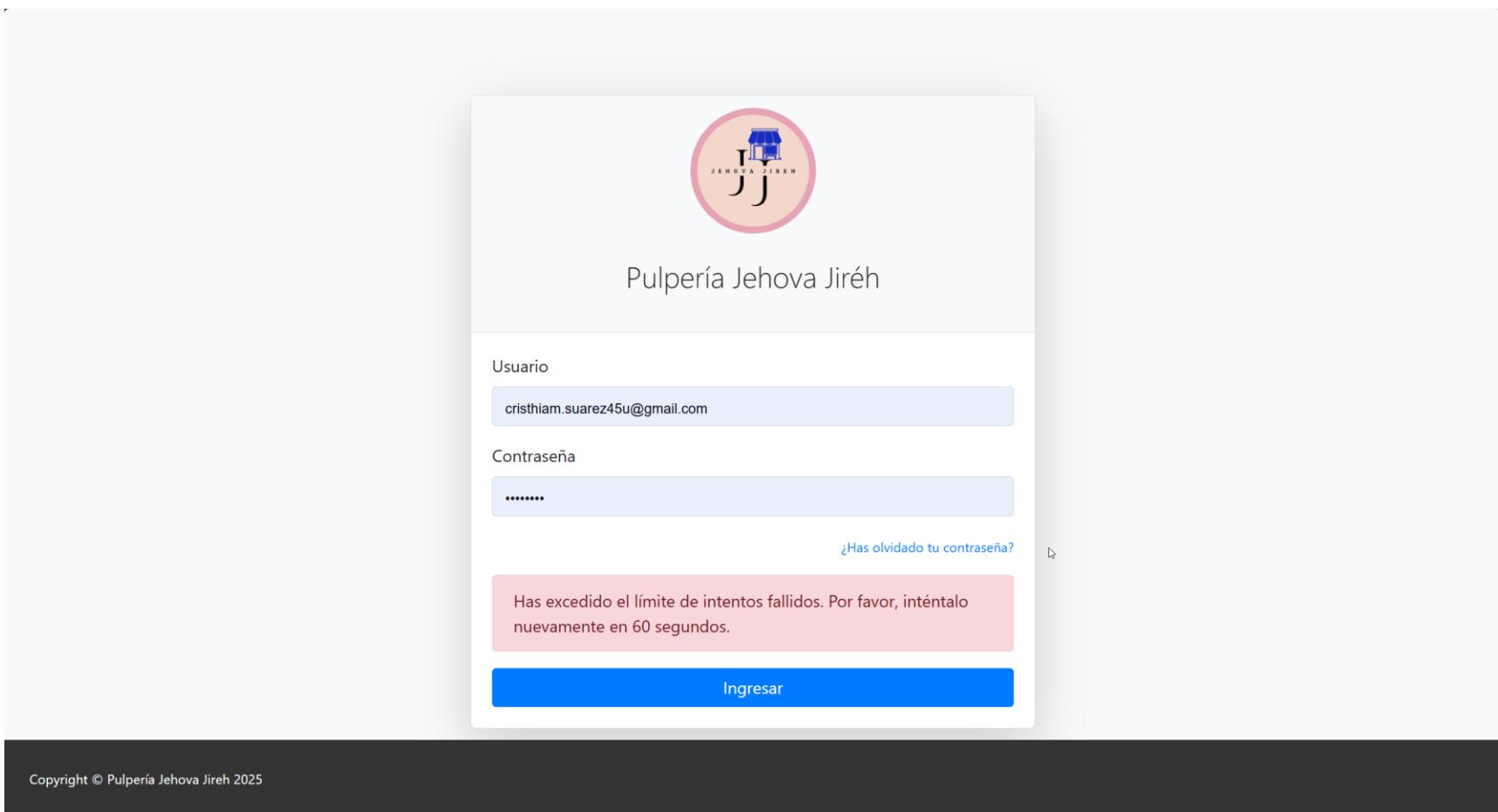


Figura 107: Validación de Sesión

Jehova Jiréh ≡

Cristhiam Omar Suárez Núñez ▾

INICIO

Menú principal

ADMINISTRACIÓN

Externa >

Interna >

Seguridad >

OPERACIONES

Actividades >

REGISTROS

Historial >

Menú principal

Total de ventas

Hoy C\$0 Semana C\$ 0 Mes C\$ 25

Productos por vencerse



MAGGI SAZONADOR
COSTILLA CRIOLLA RES 10
G.
Fecha de Vencimiento:
31/03/2025



NESTLE LECHE CONDESADA
LA LECHERA 375 G.
Fecha de Vencimiento:
31/03/2025



CAFÉ MOLIDO SELECTO 25
G
Fecha de Vencimiento:
01/04/2025



GATORADE PONCHE DE
FRUTAS 500 ML.
Fecha de Vencimiento:
03/04/2025

Productos vencidos

155

Figura 108: Total de ventas Dashboard

Jehova Jiréh ≡

Configuración y más (Alt+F)

Cristhiam Omar Suárez Núñez ▾

INICIO

Menú principal

ADMINISTRACIÓN

Externa Interna Seguridad

OPERACIONES

Actividades

REGISTROS

Historial

Menú principal

Total de ventas

Hoy C\$0 Semana C\$ 0 Mes C\$ 25

Productos por vencerse



MAGGI SAZONADOR COSTILLA CRIOLLA RES 10 G.
Fecha de Vencimiento: **31/03/2025**



NESTLE LECHE CONDESADA LA LECHERA 375 G.
Fecha de Vencimiento: **31/03/2025**



CAFÉ MOLIDO SELECTO 25 G
Fecha de Vencimiento: **01/04/2025**



GATORADE PONCHE DE FRUTAS 500 ML.
Fecha de Vencimiento: **03/04/2025**

Productos vencidos

156

Figura 109: Productos por vencerse Dashboard

Jehova Jiréh ≡

Cristhiam Omar Suárez Núñez ▾

INICIO

Menú principal

ADMINISTRACIÓN

Externa Interna Seguridad

OPERACIONES

Actividades

REGISTROS

Historial

Total de ventas

Hoy C\$0 Semana C\$ 0 Mes C\$ 25

Productos por vencerse

 MAGGI SAZONADOR COSTILLA CRIOLLA RES 10 G. Fecha de Vencimiento: 31/03/2025	 NESTLE LECHE CONDESADA LA LECHERA 375 G. Fecha de Vencimiento: 31/03/2025	 CAFÉ MOLIDO SELECTO 25 G Fecha de Vencimiento: 01/04/2025	 GATORADE PONCHE DE FRUTAS 500 ML. Fecha de Vencimiento: 03/04/2025
---	---	--	---

Productos vencidos

			
---	--	---	---

Figura 110: Productos vencidos Dashboard

Jehova Jiréh ≡

Cristhiam Omar Suárez Núñez ▾

INICIO

Menú principal

ADMINISTRACIÓN

Externa >

Interna >

Seguridad >

OPERACIONES

Actividades >

REGISTROS

Historial >

Productos vencidos

			
MAGGI CREMA DE MARISCOS 80 G. Fecha de Vencimiento: 31/01/2025	MAGGI SABOR Y COLOR POLLO Y CÚRCUMA 10 G. Fecha de Vencimiento: 31/01/2025	YUMMIX TROPICAL 31 GR Fecha de Vencimiento: 04/02/2025	COCA COLA NEGRA 355ML. Fecha de Vencimiento: 26/02/2025
			
DFRUTTA SANDIA 3 LITROS. Fecha de Vencimiento: 06/03/2025	MIRINDA NARANJA 3 LITROS. Fecha de Vencimiento: 06/03/2025	AGUA CIELO 1 LITRO. Fecha de Vencimiento: 12/03/2025	VOLT 300 ML. Fecha de Vencimiento: 15/03/2025

Figura 111: Stock actual del producto a mermar

The screenshot shows a web-based application for managing inventory. On the left, a sidebar menu is visible with categories like INICIO, ADMINISTRACIÓN, and various sub-options. The main content area displays a 'Merma' (Write-off) form and a list of previous write-offs.

Merma Form:

- Descripción:** (Description) input field.
- Producto:** (Product) dropdown menu showing 'Coca Cola 3 Litros'.
- Cantidad en Bodega:** (Warehouse Quantity) input field showing '20'.
- Cantidad:** (Quantity) input field.
- Cerrar:** (Close) button.
- Guardar:** (Save) button.

List of Write-offs:

Descripción	Cantidad	Producto	Fecha de la Merma	Actions
Leche vencida	5	Leche Eskimo Sabor a fresa 450ml	07/02/2024 15:09:56	edit trash
Pechuga vencida.	39	Pechuga de pollo Avicasa	16/05/2024 16:16:28	edit trash
Pesaje de hielo.	1.75	Pechuga de pollo Avicasa	29/02/2024 12:43:06	edit trash
Pollo caducado	1.5	Pechuga de pollo Avicasa	27/02/2024 16:20:21	edit trash
Producto vencido	1.5	Pechuga de pollo Avicasa	27/02/2024 16:35:02	edit trash
Producto vencido por mala conservación	1	Desodorante Rexona	07/02/2024 14:01:16	edit trash

At the bottom of the list, it says 'Mostrando 1 a 6 de 6 Entradas' (Showing 1 to 6 of 6 entries). Navigation buttons 'Anterior' (Previous) and 'Siguiente' (Next) are also present.

Page Footer:

Copyright © Pulperia Jehova Jireh 2025

Figura 112: Funcionalidad código de barras

The screenshot shows a software interface with a dark header bar. On the left is a 'Ver Categorías' button, a search bar containing the number '789363809337', and a magnifying glass icon. On the right are icons for a cart (0 items), user profile ('Cristhiam Omar Suárez Núñez'), and a dropdown menu. Below the header are two toggle buttons: 'Mostrar productos inactivos' (disabled) and 'Mostrar productos sin inventario' (enabled). The main content area displays a product card for 'LECHE ESKIMO 450 ML.' It features a blue and white bag of milk with the 'ESKIMO' logo and text: 'Producto lácteo ultrapasteurizado adornado con Vitaminas A, D, B1, B2 y B6, Ácido fólico y Hierro Larga Vida', 'Proceso UAT (UHT)', and '450 ml'. Below the image, the product name 'LECHE ESKIMO 450 ML.' is displayed in bold, followed by the price 'C\$ 22'. At the bottom are two buttons: 'Ver Detalle' and 'Agregar a la Factura'.

Figura 113: Eliminación de filtros

Ver Categorías Buscar productos... 

Mostrar productos Inactivos
 Mostrar productos sin inventario

 <p>Agua Cielo 500ml C\$ 15</p> <p>Ver Detalle Agregar a la Factura</p>	 <p>Arroz El Faisán 400gr C\$ 23</p> <p>Ver Detalle Agregar a la Factura</p>	 <p>Chicarrones Yummies 40gr C\$ 12</p> <p>Ver Detalle Agregar a la Factura</p>	 <p>Coca Cola 3 Litros C\$ 78</p> <p>Ver Detalle Agregar a la Factura</p>
			

Figura 114: Permitir venta por fracción

The screenshot shows a POS system interface with the following details:

Detalle de la Venta

Tienes 2 producto(s) en la factura

Product 1: Agricorp
Lomo de Cerdo
Unidad de Medida : Libra
Cantidad disponible : 48.5
Precio : C\$110

Product 2: Avicasa
Pechuga de pollo Avicasa
Unidad de Medida : Libra
Cantidad disponible : 99
Precio : C\$65

Detalle de Pago

Dólares Córdobas

0 0

Efectivo C\$ 0

Total a Pagar C\$ 1658.75

Cambio C\$ 1658.75

Buttons:

- Volver (Return)
- Buttons for adjusting quantities (minus, plus, delete) for each product.
- Procesar Pago (Process Payment) button.

Message:

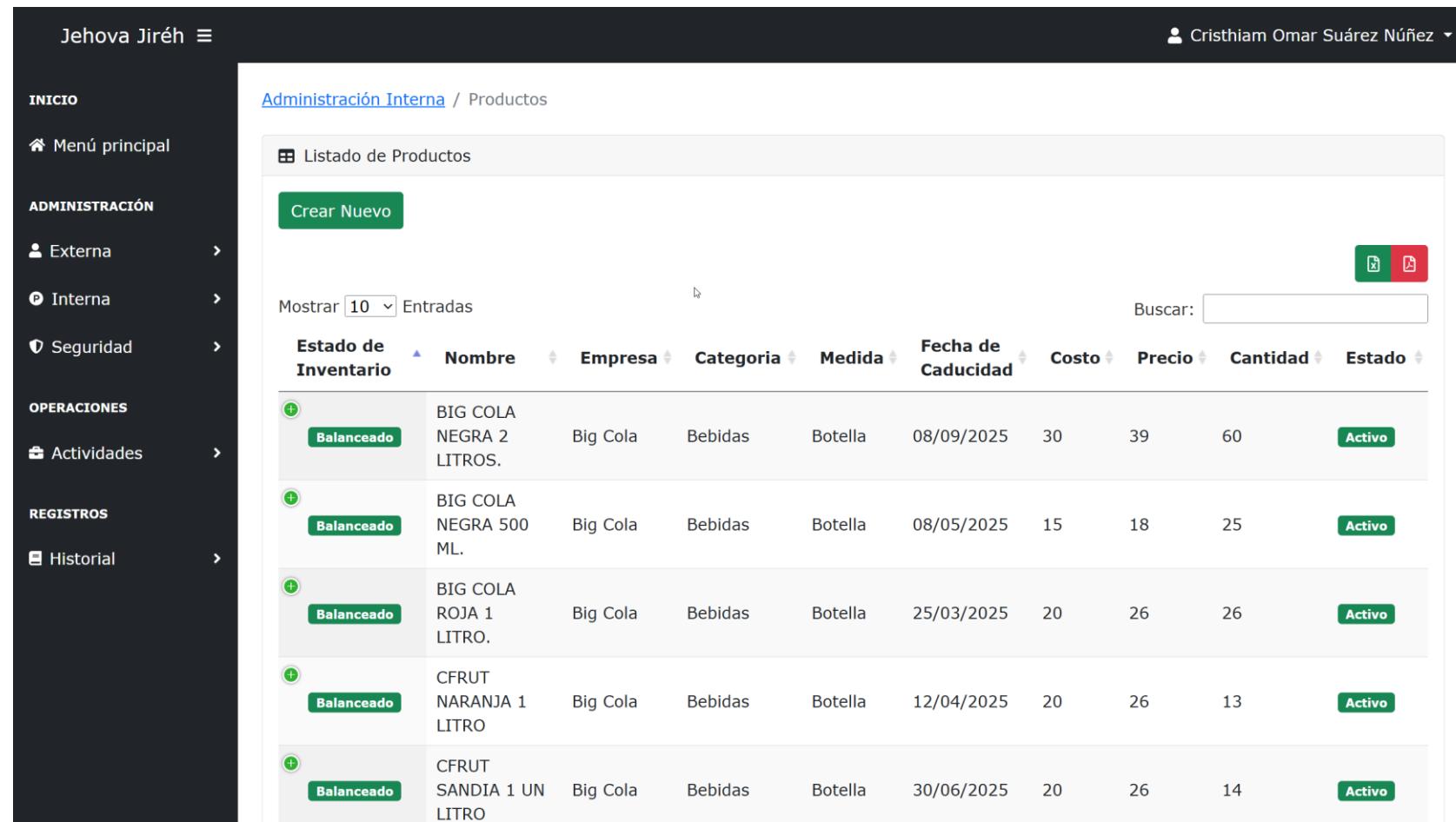
El efectivo que se ha ingresado no es suficiente para cancelar la factura

Figura 115: Filtro de fechas Venta

The screenshot shows a user interface for filtering sales data. At the top, there is a navigation bar with 'Ver Categorías', a search bar 'Buscar productos...', and a user profile 'Cristhiam Omar Suárez Núñez'. Below the navigation bar, the title 'Ventas Realizadas' is displayed. The main area is titled 'Buscar factura...' and includes a green 'Exportar a Excel' button. There are two date input fields labeled 'dd/mm/aaaa' and a blue 'Filtrar' button. The total sales amount 'Venta total: C\$ 1414.00' is shown on the right. The data is presented in a table with the following entries:

Nº	Cliente	Fecha Compra	Fecha de Cancelación
000006	Cliente No Habitual	31/03/2025	31/03/2025
000005	Samuel de Luque Lopez Santana	23/03/2025	-
000004	Samuel de Luque Lopez Santana	22/03/2025	-
000003	Cliente No Habitual	28/02/2025	-
000002	Cliente No Habitual	28/02/2025	28/02/2025
000001	Cliente No Habitual	23/02/2025	23/02/2025

Figura 116: Funcionalidad de estados según el stock



The screenshot shows a web-based application interface for managing products. The left sidebar contains a navigation menu with sections like INICIO, ADMINISTRACIÓN, OPERACIONES, and REGISTROS. The main content area is titled 'Administración Interna / Productos' and shows a table titled 'Listado de Productos'. The table includes columns for Estado de Inventario (Inventory Status), Nombre (Name), Empresa (Company), Categoría (Category), Medida (Unit), Fecha de Caducidad (Expiration Date), Costo (Cost), Precio (Price), Cantidad (Quantity), and Estado (Status). The table displays five product entries, all of which are currently 'Balanceado' (Balanced) in inventory status. The products listed are: BIG COLA NEGRA 2 LITROS, BIG COLA NEGRA 500 ML, BIG COLA ROJA 1 LITRO, CFRUT NARANJA 1 LITRO, and CFRUT SANDIA 1 UN LITRO. Each product entry includes its name, company (Big Cola), category (Bebidas), unit (Botella), expiration date, cost, price, quantity, and an 'Activ' status indicator.

Estado de Inventario	Nombre	Empresa	Categoría	Medida	Fecha de Caducidad	Costo	Precio	Cantidad	Estado
Balanceado	BIG COLA NEGRA 2 LITROS.	Big Cola	Bebidas	Botella	08/09/2025	30	39	60	Activ
Balanceado	BIG COLA NEGRA 500 ML.	Big Cola	Bebidas	Botella	08/05/2025	15	18	25	Activ
Balanceado	BIG COLA ROJA 1 LITRO.	Big Cola	Bebidas	Botella	25/03/2025	20	26	26	Activ
Balanceado	CFRUT NARANJA 1 LITRO	Big Cola	Bebidas	Botella	12/04/2025	20	26	13	Activ
Balanceado	CFRUT SANDIA 1 UN LITRO	Big Cola	Bebidas	Botella	30/06/2025	20	26	14	Activ

Tabla 38: Sprint Backlog Registro Actividades

ORIGEN	ROL	ASIGNADO	ACCION	ESTADO	PRIORIDAD	HORAS
REG ACTIVIDAD	Developer	Hosmand Rafael Belli Cespedes	Diagrama de base de datos para Registro de Actividades	Completada	Media	4.00
REG ACTIVIDAD	Developer	Hosmand Rafael Belli Cespedes	Creación de tablas de Registro de Actividades	Completada	Media	2.00
REG ACTIVIDAD	Developer	Hosmand Rafael Belli Cespedes	Diseño de procedimientos almacenados	Completada	Media	4.00
REG ACTIVIDAD	Developer	Cristhiam Omar Suarez Nuñez	Creacion de clase de Registro de Actividades	Completada	Media	2.00
REG ACTIVIDAD	Developer	Cristhiam Omar Suarez Nuñez	Creacion de Logica de Registro de Actividades	Completada	Media	2.00
REG ACTIVIDAD	Developer	Cristhiam Omar Suarez Nuñez	Creación de controlador de Registro de Actividades	Completada	Media	4.00
REG ACTIVIDAD	Developer	Hosmand Rafael Belli Cespedes	Creación de Vista de Registro de Actividades	Completada	Media	4.00

Figura 117: Diagrama de tablas Registro de Actividades

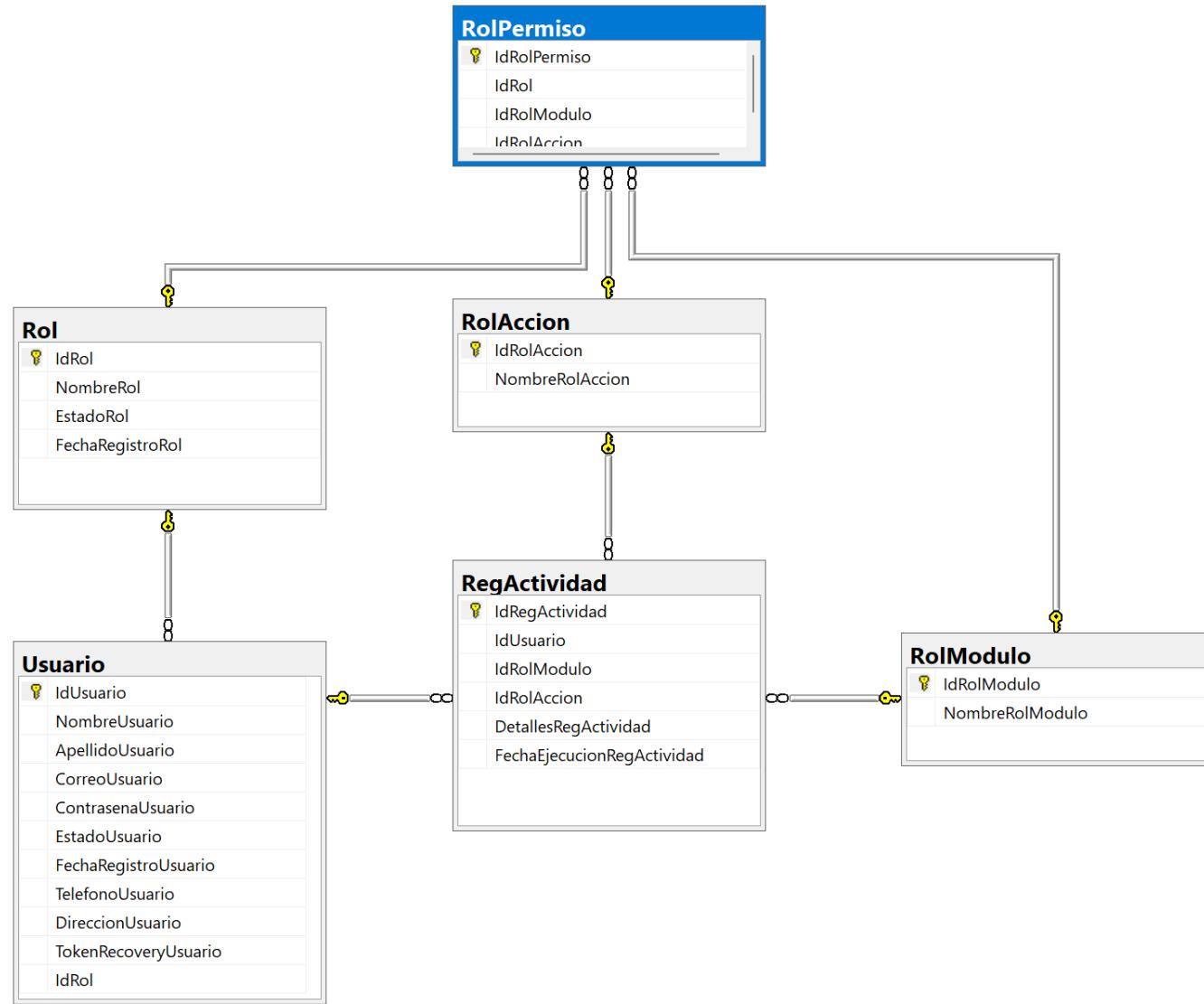


Figura 118: Tabla Registro Actividades

RegActividad	
	IdRegActividad
	IdUsuario
	IdRolModulo
	IdRolAccion
	DetallesRegActividad
	FechaEjecucionRegActividad

Figura 119: Procedimiento almacenado Registro Actividades

```
USE [DBA_JehovaJireh]
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO

ALTER proc [dbo].[sp_obtenerRegActividad]
as
begin
    SELECT
        u.CorreoUsuario,
        u.NombreUsuario + ' ' + u.ApellidoUsuario AS NombreUsuario,
        m.NombreRolModulo,
        a.NombreRolAccion,
        ua.DetallesRegActividad,
        ua.FechaEjecucionRegActividad
    FROM RegActividad ua
    INNER JOIN Usuario u ON ua.IdUsuario = u.IdUsuario
    INNER JOIN RolModulo m ON ua.IdRolModulo = m.IdRolModulo
    INNER JOIN RolAccion a ON ua.IdRolAccion = a.IdRolAccion
    ORDER BY FechaEjecucionRegActividad DESC;
end
```

Figura 120: Clase Registro Actividades

```
6     namespace ProyectoTest.Models
7     {
8         public class RegActividad
9         {
10             public string Usuario { get; set; }
11             public string NombreCompleto { get; set; }
12             public string NombreModulo { get; set; }
13             public string NombreAccion { get; set; }
14             public string Detalle { get; set; }
15             public string FechaEjecucion { get; set; }
16         }
17     }
```

Figura 121: Controlador Registro Actividades

```
public ActionResult RegActividad()
{
    if (Session["Usuario"] == null)
        return RedirectToAction("Index", "Login");

    return View();
}
```

Figura 122: Lógica Registro Actividades

```
1 referencia
public List<RegActividad> Listar(int _idusuario)
{
    List<RegActividad> rptListaUsuarioActividad = new List<RegActividad>();
    using (SqlConnection oConexion = new SqlConnection(Conexion.CN))
    {
        SqlCommand cmd = new SqlCommand("sp_obtenerRegActividad", oConexion);
        cmd.CommandType = CommandType.StoredProcedure;

        try
        {
            oConexion.Open();
            SqlDataReader dr = cmd.ExecuteReader();

            while (dr.Read())
            {
                rptListaUsuarioActividad.Add(new RegActividad())
                {
                    Usuario = dr["CorreoUsuario"].ToString(),
                    NombreCompleto = dr["NombreUsuario"].ToString(),
                    NombreModulo = dr["NombreRolModulo"].ToString(),
                    NombreAccion = dr["NombreRolAccion"].ToString(),
                    Detalle = dr["DetallesRegActividad"].ToString(),
                    FechaEjecucion = dr["FechaEjecucionRegActividad"].ToString()
                });
            }
            dr.Close();
        }
        return rptListaUsuarioActividad;
    }
}
```

Figura 123: Vista Registro Actividades

[Historial](#) / Registro de Actividades de Usuarios

Listado de Actividades de Usuario					
Correo de Usuario	Nombre del Usuario	Módulo	Acción	Detalle	Fecha de Registro
cristhiam.suarez45u@gmail.com	Cristhiam Omar Suárez Núñez	Productos	Crear	Producto registrado: Nombre = LECHE NUTRI LETY 1 LITRO., Nombre de Empresa = Fuente Pura, Nombre de Categoría = Productos Lácteos, FechaVencimiento = 2025-06-04 00:00:00, Costo = 40, Precio = 45, Stock = 4, StockMinimo = 6, StockMaximo = 18, CódigoQR = 7893630210054, UnidadMedida = 5, RutaImagen = ~/Imagenes/Productos/118.webp, Estado = Activo	09/02/2025 18:55:06
cristhiam.suarez45u@gmail.com	Cristhiam Omar Suárez Núñez	Productos	Crear	Producto registrado: Nombre = GATORADE PONCHE DE FRUTAS 500 ML., Nombre de Empresa = PepsiCO, Nombre de Categoría = Bebidas energéticas y deportivas, FechaVencimiento = 2025-11-03 00:00:00, Costo = 33, Precio = 40, Stock = 12, StockMinimo = 24, StockMaximo = 36, CódigoQR = 36731039477, UnidadMedida = 8, RutaImagen = ~/Imagenes/Productos/119.jfif, Estado = Activo	09/02/2025 18:55:06

Tabla 39: Sprint Backlog Registro Sesión

ORIGEN	ROL	ASIGNADO	ACCION	ESTADO	PRIORIDAD	HORAS
REG SESIONES	Developer	Hosmand Rafael Belli Cespedes	Diagrama de base de datos para Registro de Sesiones	Completada	Media	4.00
REG SESIONES	Developer	Hosmand Rafael Belli Cespedes	Creación de tablas de Registro de Sesiones	Completada	Media	2.00
REG SESIONES	Developer	Hosmand Rafael Belli Cespedes	Diseño de procedimientos almacenados	Completada	Media	4.00
REG SESIONES	Developer	Cristhiam Omar Suarez Nuñez	Creacion de clase de Registro de Sesiones	Completada	Media	2.00
REG SESIONES	Developer	Cristhiam Omar Suarez Nuñez	Creacion de Logica de Registro de Sesiones	Completada	Media	2.00
REG SESIONES	Developer	Cristhiam Omar Suarez Nuñez	Creación de controlador de Registro de Sesiones	Completada	Media	4.00
REG SESIONES	Developer	Hosmand Rafael Belli Cespedes	Creación de Vista de Registro de Sesiones	Completada	Media	4.00
REG SESIONES	Developer	Hosmand Rafael Belli Cespedes	Validación de Correo para envio de usuarios no registrados	Completada	Media	4.00

Figura 124: Diagrama de tablas Registro Sesión

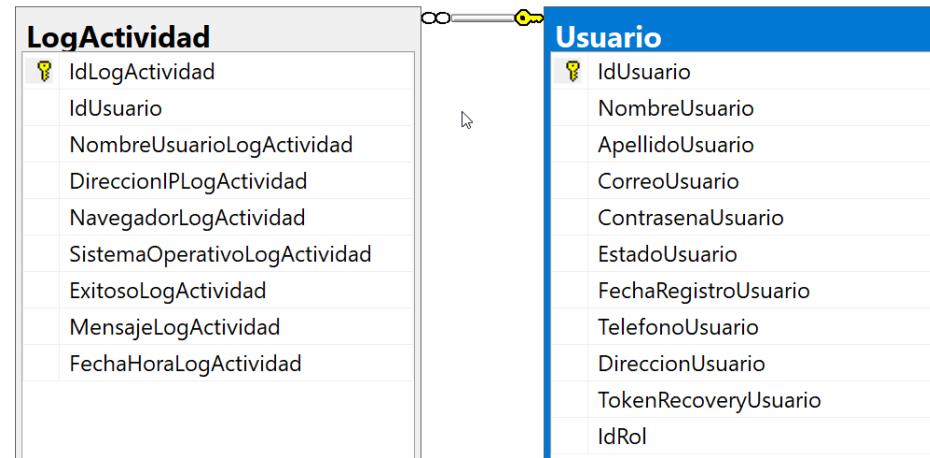


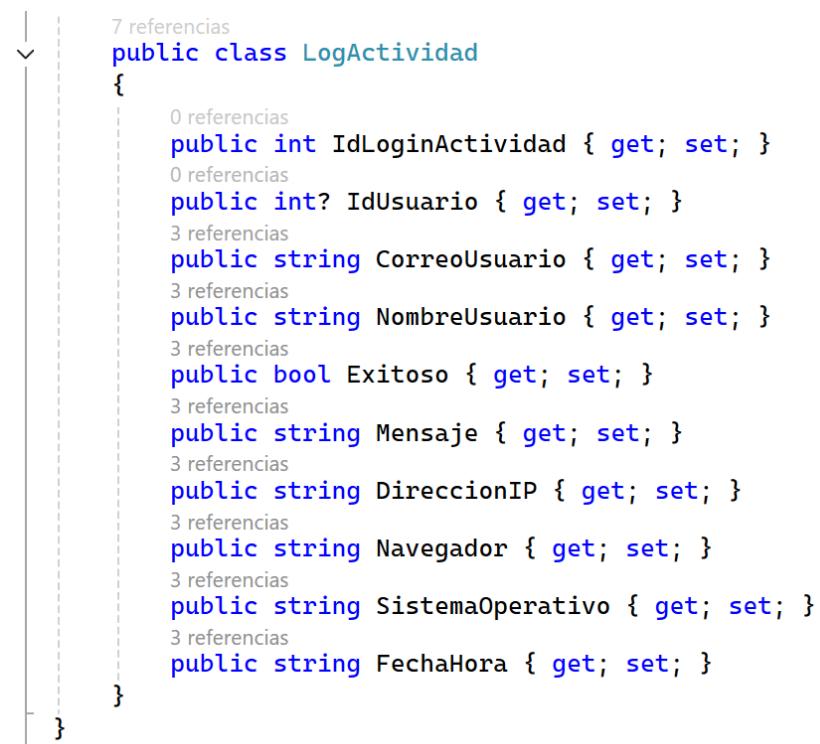
Figura 125: Tabla Registro Sesión

LogActividad	
IdLogActividad	llave primaria
IdUsuario	
NombreUsuarioLogActividad	
DireccionIPLogActividad	
NavegadorLogActividad	
SistemaOperativoLogActividad	
ExitosoLogActividad	
MensajeLogActividad	
FechaHoraLogActividad	

Figura 126: Procedimiento almacenado Registro Sesión

```
USE [DBA_JehovaJireh]
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER PROCEDURE [dbo].[sp_obtenerLogActividad]
AS
BEGIN
    SELECT
        ISNULL(L.NombreUsuarioLogActividad, 'Usuario Desconocido') AS [Correo de Usuario],
        ISNULL(U.NombreUsuario + ' ' + U.ApellidoUsuario, 'Usuario Desconocido') AS [Nombre del Usuario],
        L.ExitosoLogActividad,
        L.MensajeLogActividad,
        L.DireccionIPLogActividad,
        L.NavegadorLogActividad,
        L.SistemaOperativoLogActividad,
        L.FechaHoraLogActividad [Fecha de Registro]
    FROM dbo.LogActividad L
    LEFT JOIN dbo.Usuario U ON L.IdUsuario = U.IdUsuario
    ORDER BY L.FechaHoraLogActividad DESC
END
```

Figura 127: Clase Registro Sesión



```
7 referencias
public class LogActividad
{
    0 referencias
    public int IdLoginActividad { get; set; }
    0 referencias
    public int? IdUsuario { get; set; }
    3 referencias
    public string CorreoUsuario { get; set; }
    3 referencias
    public string NombreUsuario { get; set; }
    3 referencias
    public bool Exitoso { get; set; }
    3 referencias
    public string Mensaje { get; set; }
    3 referencias
    public string DireccionIP { get; set; }
    3 referencias
    public string Navegador { get; set; }
    3 referencias
    public string SistemaOperativo { get; set; }
    3 referencias
    public string FechaHora { get; set; }
}
```

Figura 128: Controlador Registro Sesión

```
0 referencias
public ActionResult LogActividad()
{
    if (Session["Usuario"] == null)
        return RedirectToAction("Index", "Login");

    return View();
}
```

Figura 129: Lógica Registro Sesión

```
1 referencia
public List<LogActividad> Listar(int _idusuario)
{
    List<LogActividad> rptListaUsuarioActividad = new List<LogActividad>();
    using (SqlConnection oConexion = new SqlConnection(Conexion.CN))
    {
        SqlCommand cmd = new SqlCommand("sp_obtenerLogActividad", oConexion);
        cmd.CommandType = CommandType.StoredProcedure;

        try
        {
            oConexion.Open();
            SqlDataReader dr = cmd.ExecuteReader();

            while (dr.Read())
            {
                rptListaUsuarioActividad.Add(new LogActividad())
                {
                    CorreoUsuario = dr["Correo de Usuario"].ToString(),
                    NombreUsuario = dr["Nombre del Usuario"].ToString(),
                    Exitoso = Convert.ToBoolean(dr["ExitosoLogActividad"].ToString()),
                    Mensaje = dr["MensajeLogActividad"].ToString(),
                    DireccionIP = dr["DireccionIPLogActividad"].ToString(),
                    Navegador = dr["NavegadorLogActividad"].ToString(),
                    SistemaOperativo = dr["SistemaOperativoLogActividad"].ToString(),
                    FechaHora = dr["Fecha de Registro"].ToString(),
                });
            }
            dr.Close();
        }
        return rptListaUsuarioActividad;
    }
}
```

Figura 130: Vista Registro Sesión

[Historial](#) / Registro de Inicio de Sesión

>Listado de Inicio de Sesión

Mostrar 10 Entradas Buscar:

Correo de Usuario	Nombre del Usuario	Exitoso	Mensaje	Dirección IP	Navegador	Sistema Operativo
+ cristhiam.suarez45u@gmail.com	Cristhiam Omar Suárez Núñez	Exitoso	Inicio de Sesión exitoso	::1	Chrome	WinNT
+ cristhiam.suarez45u@gmail.c1om	Usuario Desconocido	Fallido	Usuario bloqueado temporalmente por intentos fallidos	::1	Chrome	WinNT
+ cristhiam.suarez45u@gmail.c1om	Usuario Desconocido	Fallido	Bloqueado por múltiples intentos fallidos	::1	Chrome	WinNT
+ cristhiam.suarez45u@gmail.c1om	Usuario Desconocido	Fallido	Intento de login con correo inexistente	::1	Chrome	WinNT
+ cristhiam.suarez45u@gmail.com1	Usuario Desconocido	Fallido	Intento de login con correo inexistente	::1	Chrome	WinNT
+ 1	Usuario Desconocido	Fallido	Intento de login con correo inexistente	::1	Chrome	WinNT
+ 1	Usuario	Fallido	Intento de login con	...	Chrome	WinNT

Figura 131: Validación de usuario desconocido

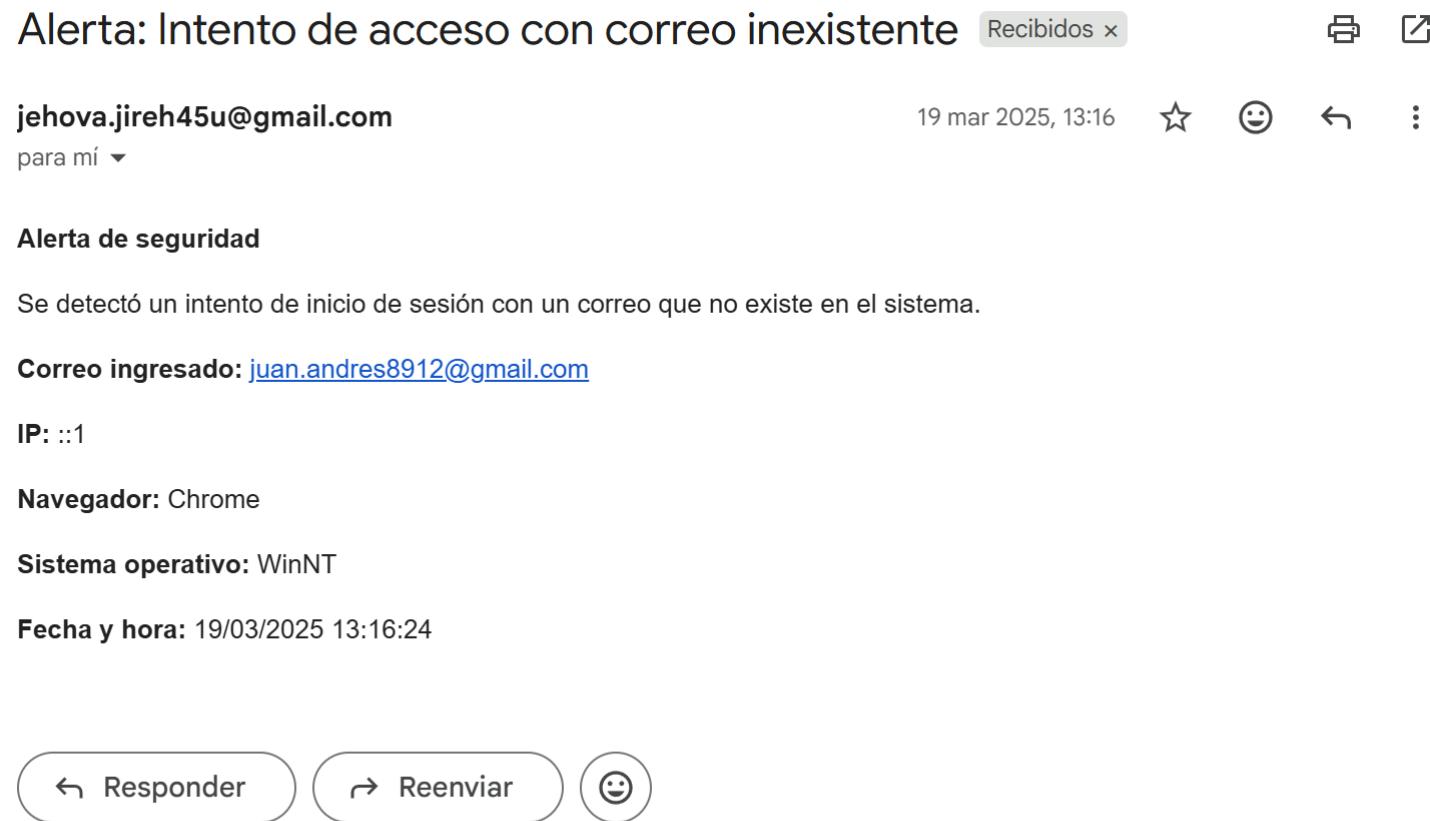


Tabla 40: Sprint Backlog Registro de Errores

ORIGEN	ROL	ASIGNADO	ACCION	ESTADO	PRIORIDAD	HORAS
REG ERRORES	Developer	Hosmand Rafael Belli Cespedes	Diagrama de base de datos para Registro de Errores	Completada	Media	4.00
REG ERRORES	Developer	Hosmand Rafael Belli Cespedes	Creación de tablas de Registro de Errores	Completada	Media	2.00
REG ERRORES	Developer	Hosmand Rafael Belli Cespedes	Diseño de procedimientos almacenados	Completada	Media	4.00
REG ERRORES	Developer	Cristhiam Omar Suarez Nuñez	Creacion de clase de Registro de Errores	Completada	Media	2.00
REG ERRORES	Developer	Cristhiam Omar Suarez Nuñez	Creacion de Logica de Registro de Errores	Completada	Media	2.00
REG ERRORES	Developer	Cristhiam Omar Suarez Nuñez	Creación de controlador de Registro de Errores	Completada	Media	4.00
REG ERRORES	Developer	Hosmand Rafael Belli Cespedes	Creación de Vista de Registro de Errores	Completada	Media	4.00

Figura 132: Diagrama de tablas Registro de Errores

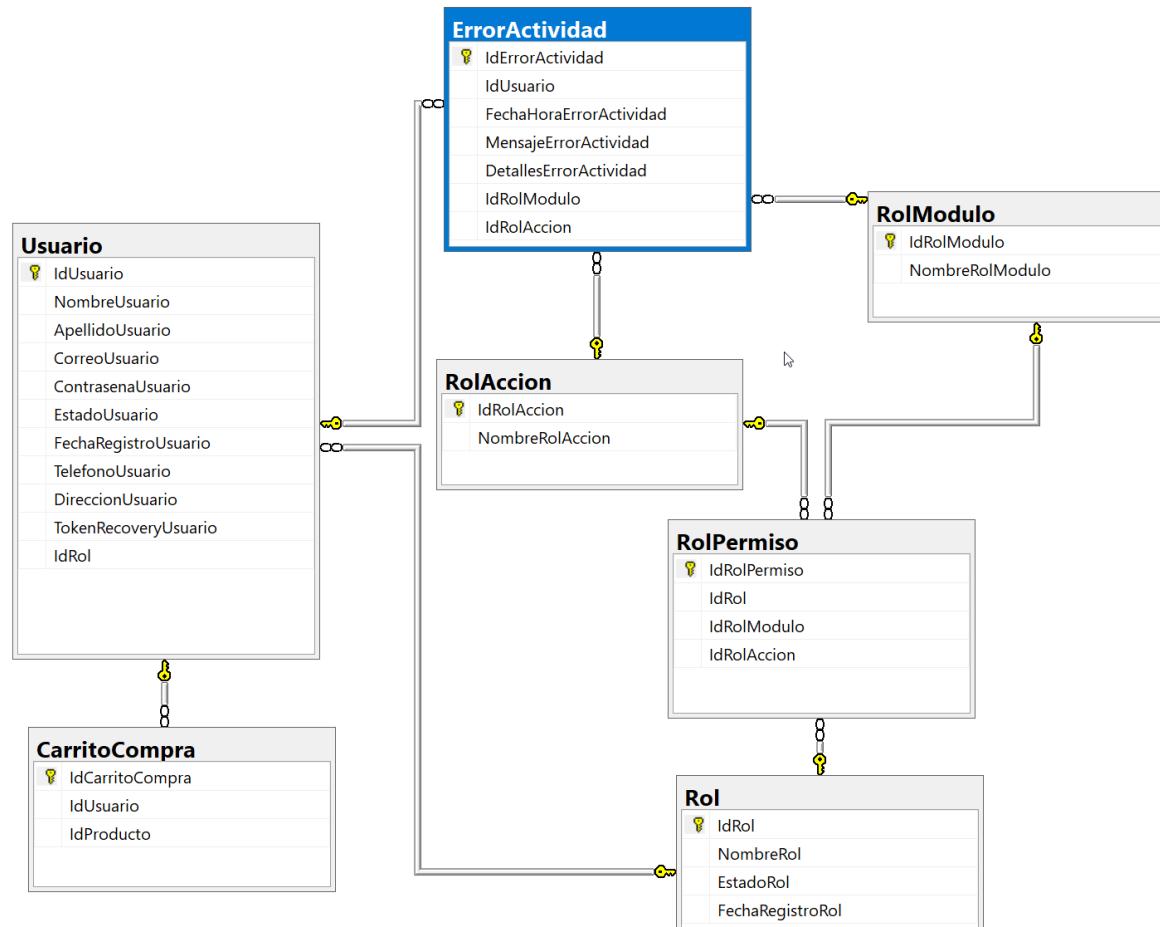


Figura 133: Tabla Registro de Errores

ErrorActividad	
	IdErrorActividad
	
	IdUsuario
	FechaHoraErrorActividad
	MensajeErrorActividad
	DetallesErrorActividad
	IdRolModulo
	IdRolAccion

Figura 134: Clase Registro de Errores

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace ProyectoTest.Models
{
    7 referencias
    public class ErrorActividad
    {
        0 referencias
        public int IdErrorActividad { get; set; }
        0 referencias
        public int? IdUsuario { get; set; }
        3 referencias
        public string NombreUsuario { get; set; }
        3 referencias
        public string MensajeErrorActividad { get; set; }
        3 referencias
        public string DetallesErrorActividad { get; set; }
        3 referencias
        public string FechaHoraErrorActividad { get; set; }
        0 referencias
        public int IdRolModulo { get; set; }
        0 referencias
        public int IdRolAccion { get; set; }
        3 referencias
        public string NombreRolModulo { get; set; }
        3 referencias
        public string NombreRolAccion { get; set; }
    }
}
```

Figura 135: Lógica Registro de Errores

```
public static void RegistrarError(Exception ex, int? idUsuario, int? idRolModulo, int? idRolAccion)
{
    try
    {
        using (SqlConnection oConexion = new SqlConnection(Conexion.CN))
        {
            SqlCommand cmd = new SqlCommand("sp_RegistrarErrorActividad", oConexion);
            cmd.CommandType = CommandType.StoredProcedure;

            string mensajeNormalizado = ProyectoTest.Helpers.ErrorTraductor.NormalizarMensaje(ex);

            cmd.Parameters.AddWithValue("@IdUsuario", idUsuario ?? (object)DBNull.Value);
            cmd.Parameters.AddWithValue("@MensajeError", mensajeNormalizado);
            cmd.Parameters.AddWithValue("@DetallesError", $"{ex.Message}\n{ex.StackTrace}");
            cmd.Parameters.AddWithValue("@IdRolModulo", idRolModulo ?? (object)DBNull.Value);
            cmd.Parameters.AddWithValue("@IdRolAccion", idRolAccion ?? (object)DBNull.Value);

            oConexion.Open();
            cmd.ExecuteNonQuery();
        }
    }
    catch
    {
        ErrorActividadLogica.RegistrarError(ex, idUsuario, 18, 2);
    }
}
```

Figura 136: Controlador Registro de Errores

```
0 referencias
public ActionResult ErrorActividad()
{
    if (Session["Usuario"] == null)
        return RedirectToAction("Index", "Login");

    return View();
}
```

Figura 137: Vista Registro de Errores

[Historial](#) / Registro de Errores

>Listado de Errores

Mostrar 10 Entradas

Buscar:

Nombre del Usuario	Mensaje de Error	Detalle Técnico del Error	Módulo	Acción	Fecha de Registro
Cristhiam Omar Suárez Núñez	No se puede eliminar este registro porque tiene información relacionada en el sistema.	The DELETE statement conflicted with the REFERENCE constraint "FK_Usuario_R...". Ver más	Roles	Eliminar	01/03/2025 11:35:00
Cristhiam Omar Suárez Núñez	Ha ocurrido un error inesperado. Contacte con el equipo de soporte.	Invalid column name 'IdMerma'. en System.Data.SqlClient.SqlConnection.On... Ver más	Regalías	Eliminar	28/02/2025 15:50:49
Cristhiam Omar Suárez Núñez	Ha ocurrido un error inesperado. Contacte con el equipo de soporte.	Invalid column name 'IdMerma'. en System.Data.SqlClient.SqlConnection.On... Ver más	Regalías	Eliminar	28/02/2025 15:50:49
Cristhiam Omar Suárez Núñez	Ha ocurrido un error inesperado. Contacte con el equipo de soporte.	Invalid column name 'IdMerma'. en System.Data.SqlClient.SqlConnection.On... Ver más	Regalías	Eliminar	28/02/2025 15:50:42
Mostrando 1 a 6 de 6 Entradas			Anterior	1	Siguiente

Al finalizar la iteración, el Incremento fue presentado al cliente para la Sprint Review. Para ver más detalles, véase Figura 100: Acta de reunión sprint 3 del Anexo II.

El valor entregado se centró en la Optimización de la Usabilidad y la Seguridad del Sistema. Se implementó el manejo de Códigos de Barra y la lógica de Venta/Ingreso en Fracciones para la gestión de inventario a granel. El principal activo estratégico es la entrega de un Sistema de Auditoría Integral que incluye los Registros de Actividades, Sesiones y Errores.

Figura 138: Resumen del Sprint Review Sprint 3

Sprint Review	Valor entregado
Pruebas de Usuario	Se realizó una demostración y Pruebas de aceptación por parte de la pulpería. Se confirmó la operatividad del manejo de fracciones y la efectividad de los filtros de búsqueda por fechas en Ventas y Compras. Jehová Jireh se mostró satisfecho con el cumplimiento de todos los requerimientos acordados.
Resultado Tangible	Se entregó un sistema que pasó de ser funcional a ser robusto y auditável. El resultado incluye módulos optimizados y tres nuevos módulos de registro que garantizan la trazabilidad y la transparencia de las operaciones, asegurando que el producto final cumplió plenamente con las expectativas de la Pulpería Jehová Jireh.
Preparación	El Sprint concluyó con la aprobación formal del cliente, lo que valida la calidad y el cumplimiento del trabajo realizado. El equipo demostró estar listo para el siguiente ciclo de desarrollo, el Sprint 4, que se centró en la consolidación de reportes.

Anexo VIII. Sprint 4

El Sprint 4 fue un ciclo de cuatro semanas enfocado en desarrollar la reportaría del sistema. En línea con los principios de Scrum, el objetivo principal fue transformar los datos brutos almacenados en información estratégica y visualmente accesible, permitiendo a la Pulperia Jehová Jireh tomar decisiones informadas y optimizar sus operaciones. Para lograr este hito, el equipo se dedicó por completo a la creación de una suite de reportes integral y la integración de Power BI al sistema para una visualización de datos avanzada.

El proceso comenzó en la creación de informes específicos. Se inició con el desarrollo de reportes para el análisis de productos, empresas y categorías. Luego, se crearon informes para la gestión de relaciones comerciales, analizando el comportamiento de clientes y proveedores. Posteriormente, se implementaron reportes para la auditoría de operaciones internas, evaluando el desempeño de los usuarios y el movimiento de las compras y ventas. Finalmente, se cerró el sprint con la creación de reportes para el control de Regalías y Mermas, lo que brindó una visión clara de las pérdidas de inventario.

Tabla 41: Reportes del sistema

Reportes	Descripción
Gestión de Inventario	Se desarrollaron informes para el análisis de productos, empresas y categorías, incluyendo reportes esenciales como el listado de productos por caducar y los productos más vendidos.
Relación Comercial	Se crearon informes para analizar el comportamiento de clientes y proveedores, permitiendo la identificación del cliente más frecuente y el proveedor con más compras.
Operaciones Internas y Auditoría	Se implementaron reportes analíticos para evaluar el desempeño del personal, incluyendo el usuario con más ventas y el movimiento de inventario al identificar los productos más comprados y vendidos.
Control de Inventario Especial	Se crearon reportes estratégicos para los módulos de Regalías y Mermas, brindando una visión clara de las pérdidas de inventario, asegurando un control completo del flujo de mercancías por vías no transaccionales.

Figura 139: Sprint Backlog del Sprint 4

FECHA DE INICIO	DIAS	PROGRESO
03/03/2025	27	100%

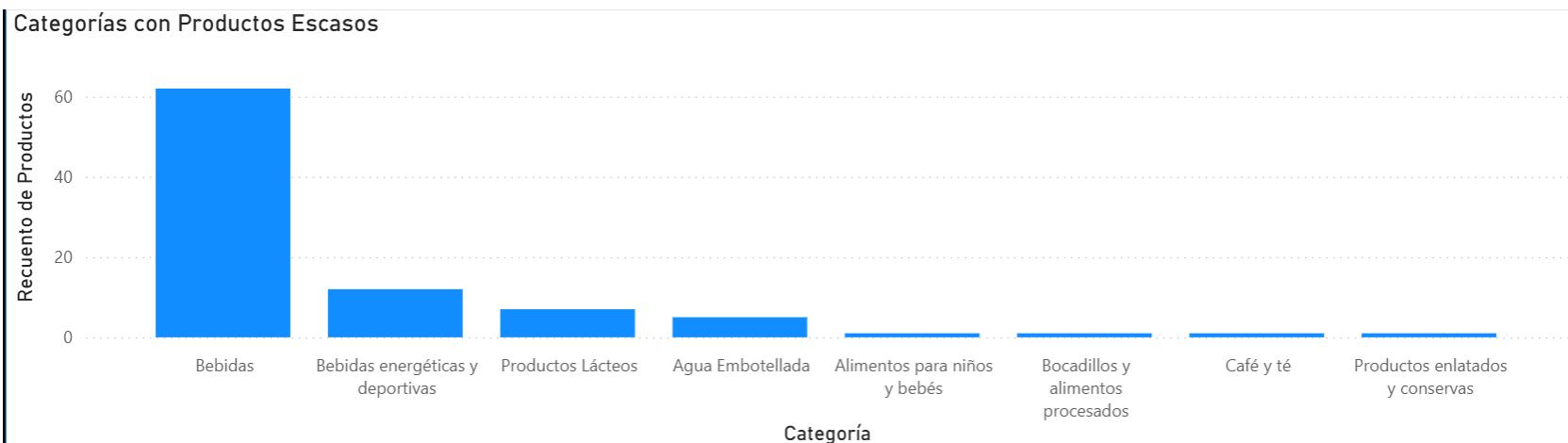
BACKLOG SPRINT 4

CATEGORIA	ROL	ASIGNADO	ACCION	ESTADO	PRIORIDAD	HORAS
Funcionalidad	Developer Team	Hosmand Rafael Belli Cespedes	Reporte de productos General	Completada	Media	4.00
Funcionalidad	Developer Team	Hosmand Rafael Belli Cespedes	Reporte: productos mas vendidos	Completada	Media	4.00
Funcionalidad	Developer Team	Hosmand Rafael Belli Cespedes	Reporte: productos por caducarse	Completada	Media	4.00
Funcionalidad	Developer Team	Hosmand Rafael Belli Cespedes	Reporte: productos caducados	Completada	Media	4.00
Funcionalidad	Developer Team	Cristhiam Omar Suarez Nuñez	Reporte de empresas General	Completada	Media	4.00
Funcionalidad	Developer Team	Cristhiam Omar Suarez Nuñez	Reporte: Empresas con mas ventas	Completada	Media	4.00
Funcionalidad	Developer Team	Cristhiam Omar Suarez Nuñez	Reporte de Categorias General	Completada	Media	4.00
Funcionalidad	Developer Team	Cristhiam Omar Suarez Nuñez	Reporte: Categoria con mas movimientos	Completada	Media	4.00
Funcionalidad	Developer Team	Cristhiam Omar Suarez Nuñez	Reporte: categoria con mas ventas	Completada	Media	4.00
Funcionalidad	Developer Team	Hosmand Rafael Belli Cespedes	Reporte de clientes General	Completada	Media	4.00

Funcionalidad	Developer Team	Cristhiam Omar Suarez Nuñez	Reporte: Cliente mas frecuente	Completada	Media	4.00
Funcionalidad	Developer Team	Cristhiam Omar Suarez Nuñez	Reporte: Cliente con mas compras	Completada	Media	4.00
Funcionalidad	Developer Team	Cristhiam Omar Suarez Nuñez	Reporte de proveedores General	Completada	Media	4.00
Funcionalidad	Developer Team	Cristhiam Omar Suarez Nuñez	Reporte: proveedor con mas productos	Completada	Media	4.00
Funcionalidad	Developer Team	Cristhiam Omar Suarez Nuñez	Reporte: Proveedor con mas compras	Completada	Media	4.00
Funcionalidad	Developer Team	Cristhiam Omar Suarez Nuñez	Reporte de usuario General	Completada	Media	4.00
Funcionalidad	Developer Team	Cristhiam Omar Suarez Nuñez	Reporte: Usuario con mas ventas	Completada	Media	4.00
Funcionalidad	Developer Team	Cristhiam Omar Suarez Nuñez	Reporte de compras General	Completada	Media	4.00
Funcionalidad	Developer Team	Cristhiam Omar Suarez Nuñez	Reporte: productos mas comprados	Completada	Media	4.00
Funcionalidad	Developer Team	Cristhiam Omar Suarez Nuñez	Reporte: productos menos comprados	Completada	Media	4.00
Funcionalidad	Developer Team	Hosmand Rafael Belli Cespedes	Reporte de ventas General	Completada	Media	4.00
Funcionalidad	Developer Team	Hosmand Rafael Belli Cespedes	Reporte: Productos mas vendidos	Completada	Media	4.00
Funcionalidad	Developer Team	Hosmand Rafael Belli Cespedes	Reporte: ventas mas altas	Completada	Media	4.00
Funcionalidad	Developer Team	Cristhiam Omar Suarez Nuñez	Reporte de regalias General	Completada	Media	4.00
Funcionalidad	Developer Team	Cristhiam Omar Suarez Nuñez	Reporte: cantidad de productos por regalia	Completada	Media	4.00
Funcionalidad	Developer Team	Cristhiam Omar Suarez Nuñez	Reporte: concepto de las regalias	Completada	Media	4.00
Funcionalidad	Developer Team	Cristhiam Omar Suarez Nuñez	Reporte: Empresa con mas regalias	Completada	Media	4.00
Funcionalidad	Developer Team	Hosmand Rafael Belli Cespedes	Reporte de mermas General	Completada	Media	4.00
Funcionalidad	Developer Team	Hosmand Rafael Belli Cespedes	Reporte: Producto con mas mermas	Completada	Media	4.00

Figura 140: Reporte general

Categorías con Productos Escasos



84

Total de Productos con escaso Stock.

6

Total de Productos con escaso Stock.

Productos con Stock Escaso

Producto	Stock	Stock Mínimo
AGUA ALPINA 1 LITRO	9,00	24
AGUA CIELO 1 LITRO.	5,00	24
AGUA FUENTE PURA 1.5 LITROS.	8,00	12
AGUA MAYA 600 ML.	2,00	12
BIG COLA NEGRA 1 LITRO	39,00	48

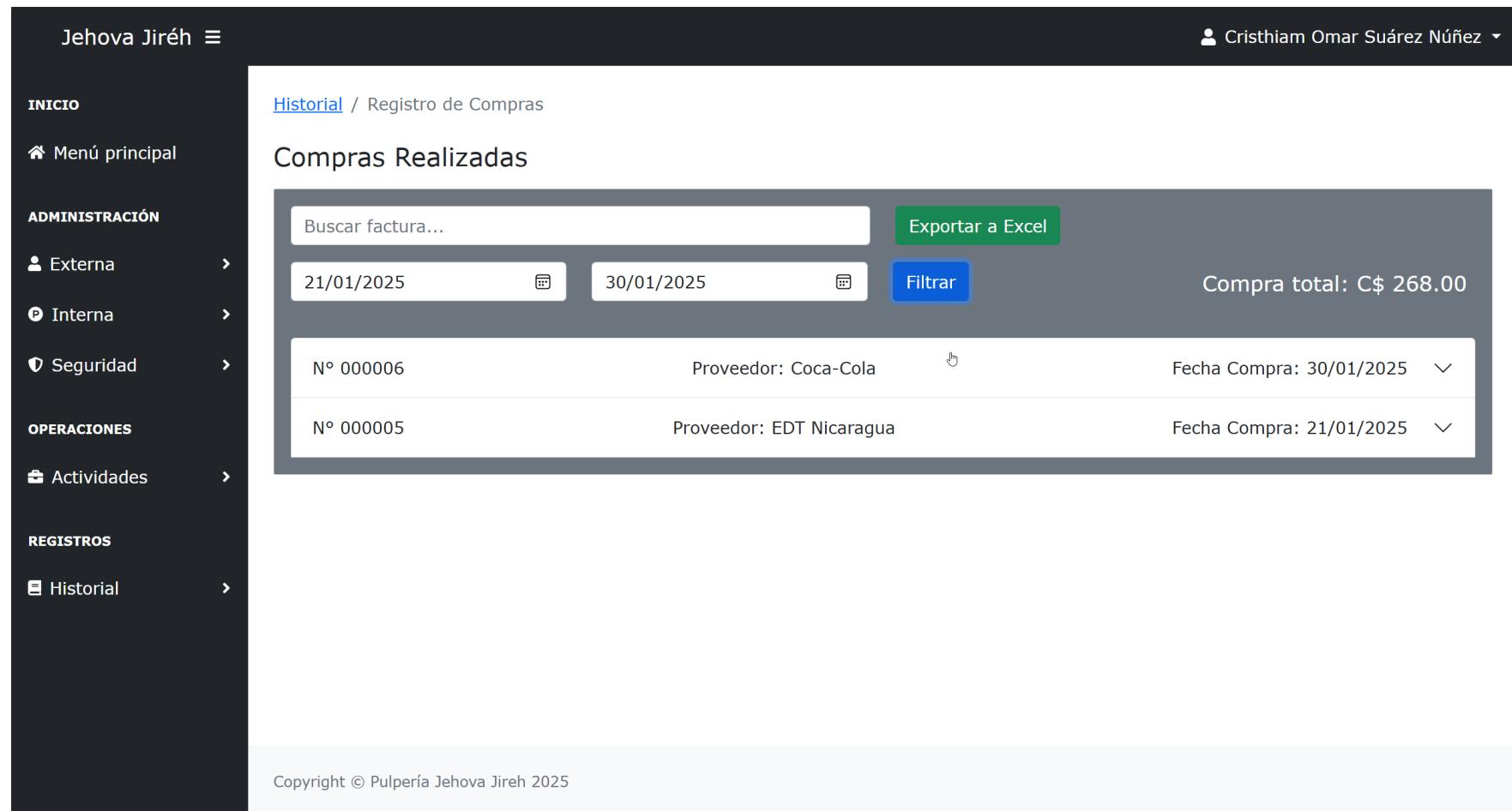
Productos sin Stock

Producto	Stock	Stock Mínimo
CANADA DRY GINGER ALE 354 ML.	0,00	12
COCA COLA 354 ML LATA.	0,00	12
Jugo De La Granja Naranja 270 ML	0,00	12
Jugo De La Granja Naranja 500 ML.	0,00	12
LECHE ESKIMO SABOR FRESCA 450 ML.	0,00	6

Figura 141: Reporte de compras por número de orden

The screenshot shows the Jehova Jiréh application interface. The top navigation bar includes the logo 'Jehova Jiréh' and a user profile for 'Cristhiam Omar Suárez Núñez'. The left sidebar contains a navigation menu with sections: INICIO, ADMINISTRACIÓN, OPERACIONES, and REGISTROS. The 'INICIO' section has a link to 'Historial / Registro de Compras'. The main content area is titled 'Compras Realizadas' and displays a purchase history. The search bar shows '000005'. To the right are buttons for 'Exportar a Excel' (Export to Excel) and 'Filtrar' (Filter). The total purchase amount is 'Compra total: C\$ 256.00'. Below this, a table shows a single item: 'Nº 000005' (Purchase Number), 'Proveedor: EDT Nicaragua' (Supplier: EDT Nicaragua), and 'Fecha Compra: 21/01/2025'. The item details are: 'Jugo De La Granja Naranja 500 ML.' (Orange juice from La Granja 500 ML), 'Cantidad: 8 Total: C\$ 256' (Quantity: 8 Total: C\$ 256). The bottom of the page includes a copyright notice: 'Copyright © Pulperia Jehova Jireh 2025'.

Figura 142: Reporte de compras por rango de fechas



Jehova Jiréh ≡

INICIO

Historial / Registro de Compras

Compras Realizadas

ADMINISTRACIÓN

- Externa >
- Interna >
- Seguridad >

OPERACIONES

- Actividades >

REGISTROS

- Historial >

Buscar factura... Exportar a Excel

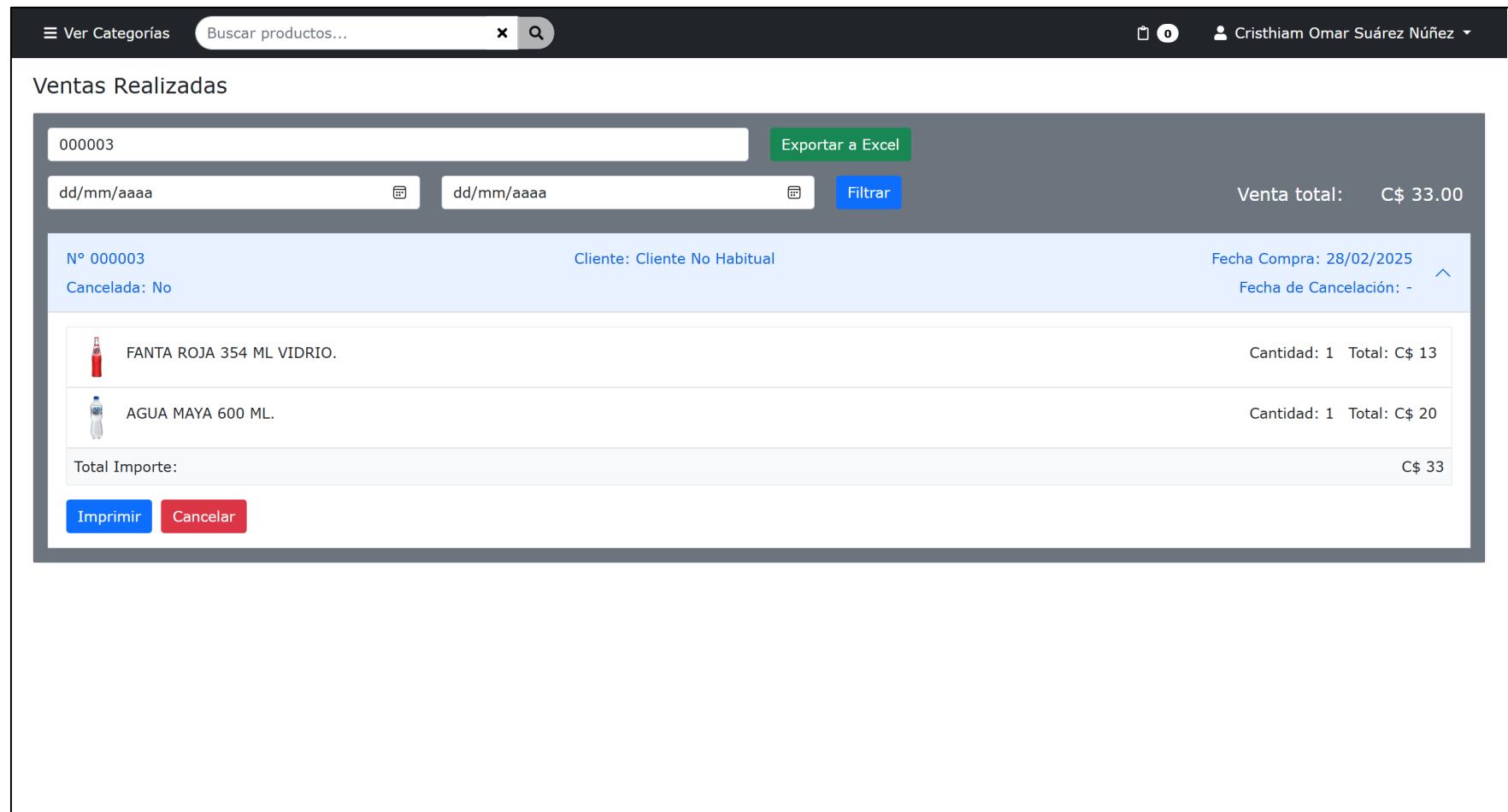
21/01/2025 30/01/2025 Filtrar

Compra total: C\$ 268.00

Nº	Proveedor	Fecha Compra
000006	Coca-Cola	30/01/2025
000005	EDT Nicaragua	21/01/2025

Copyright © Pulperia Jehova Jireh 2025

Figura 143: Reporte de ventas por número de orden



The screenshot shows a sales report interface with the following details:

Header: Ver Categorías, Buscar productos..., Exportar a Excel, Cristhiam Omar Suárez Núñez

Section: Ventas Realizadas

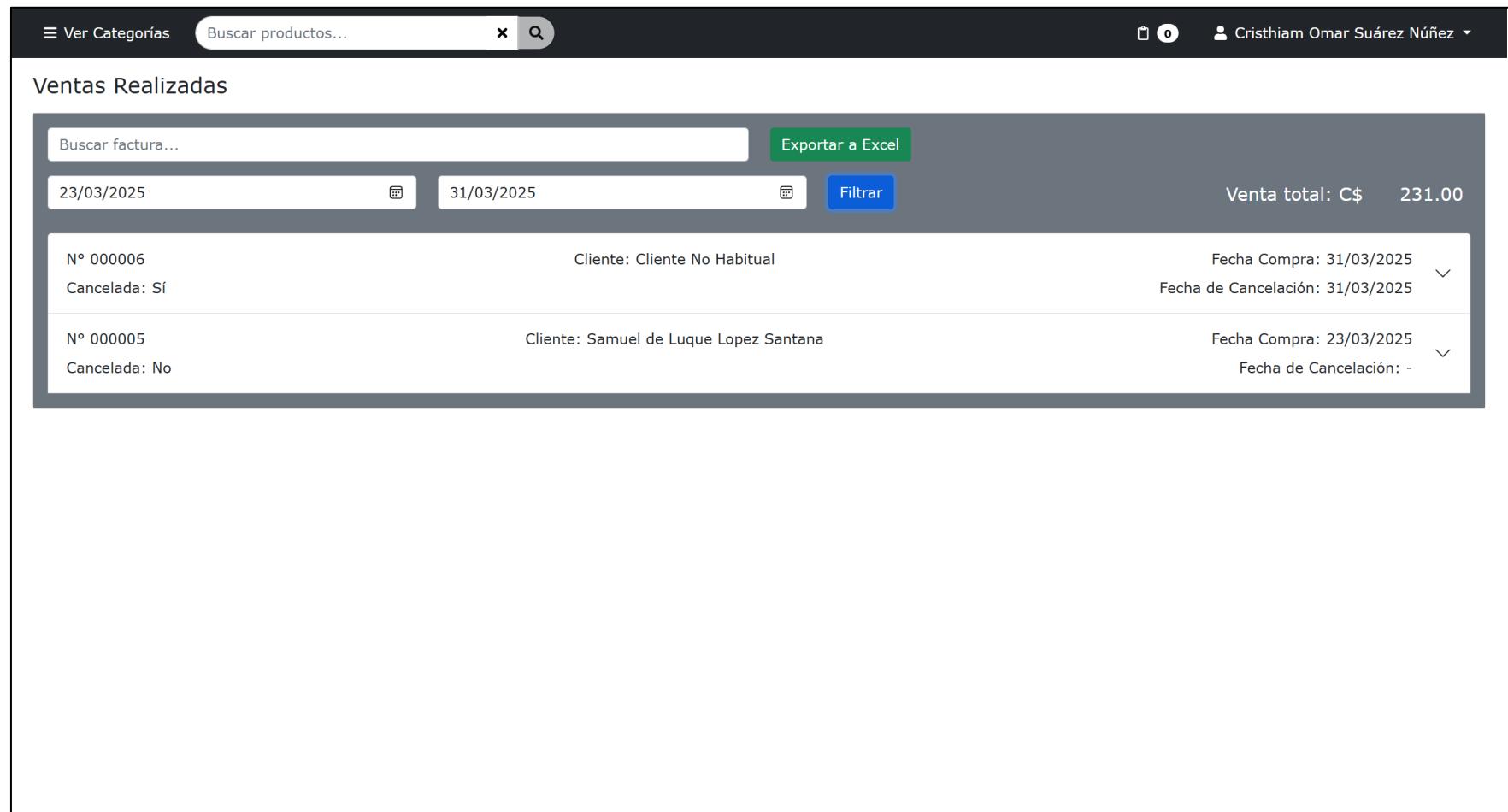
Order Information: N° 000003, Cliente: Cliente No Habitual, Fecha Compra: 28/02/2025, Fecha de Cancelación: -, Cancelada: No

Items:

Item	Cantidad	Total
FANTA ROJA 354 ML VIDRIO.	1	C\$ 13
AGUA MAYA 600 ML.	1	C\$ 20
Total Importe:		C\$ 33

Buttons: Imprimir, Cancelar

Figura 144: Reporte de ventas por rango de fechas



The screenshot shows a sales report interface with the following details:

Header: Ver Categorías, Buscar productos..., cristhiam Omar Suárez Núñez

Section: Ventas Realizadas

Search and Export: Buscar factura..., Exportar a Excel

Filter: 23/03/2025 - 31/03/2025, Filtrar

Total Sales: Venta total: C\$ 231.00

Transactions:

Nº	Cliente	Fecha Compra	Fecha de Cancelación
000006	Cliente No Habitual	31/03/2025	31/03/2025
000005	Samuel de Luque Lopez Santana	23/03/2025	-

Details for Transaction 000006:
Nº 000006
Cliente: Cliente No Habitual
Fecha Compra: 31/03/2025
Fecha de Cancelación: 31/03/2025
Cancelada: Sí

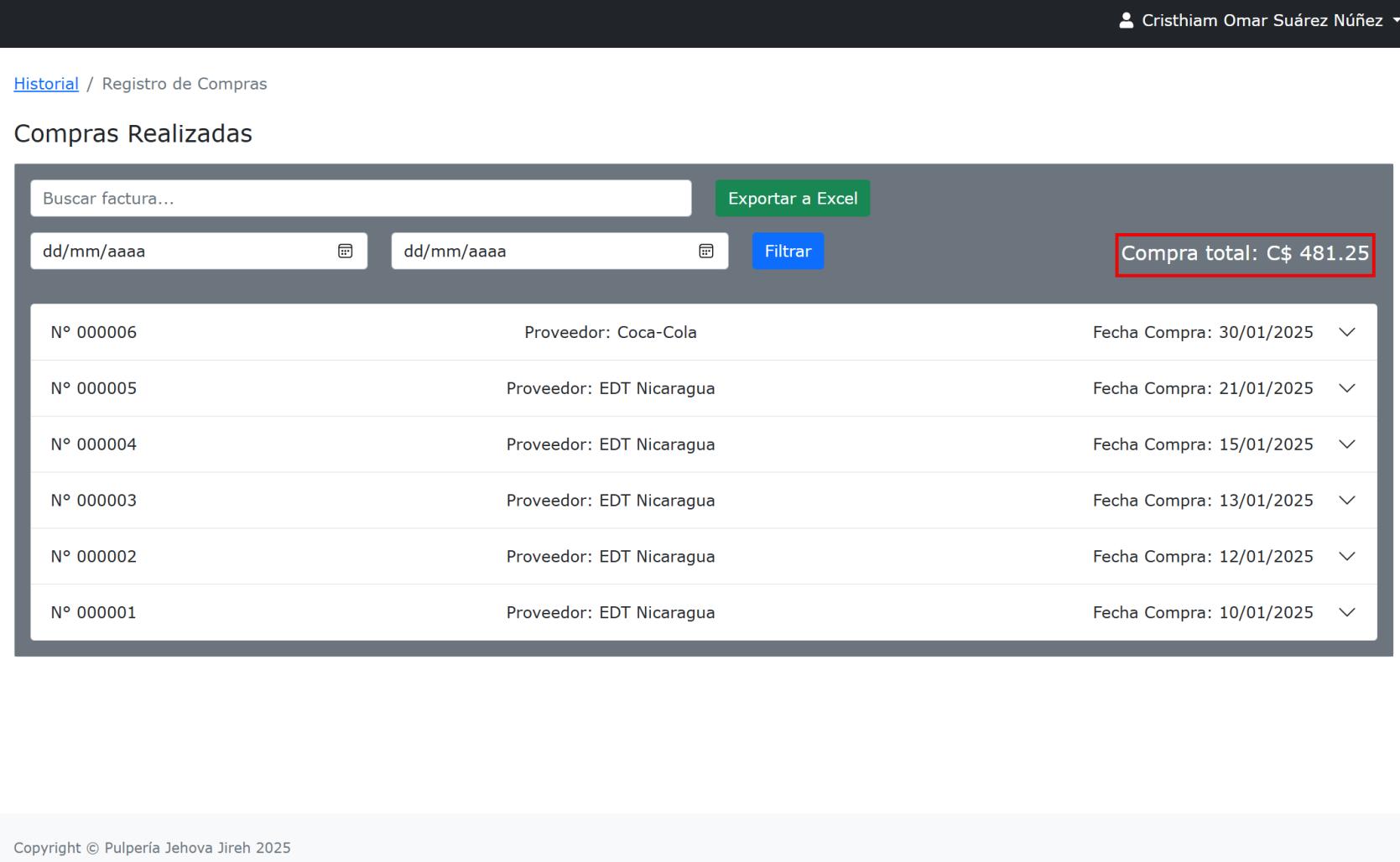
Details for Transaction 000005:
Nº 000005
Cliente: Samuel de Luque Lopez Santana
Fecha Compra: 23/03/2025
Fecha de Cancelación: -
Cancelada: No

Figura 145: Total de ventas realizadas

The screenshot shows a web-based application for managing sales. At the top, there is a navigation bar with links for 'Ver Categorías' and 'Buscar productos...', a search bar, and a user profile for 'Cristhiam Omar Suárez Núñez'. Below the navigation is a section titled 'Ventas Realizadas' (Sales Made). This section includes a search bar for 'Buscar factura...', a green 'Exportar a Excel' button, and two date input fields for filtering. A blue 'Filtrar' button is located to the right of the date fields. In the top right corner of this section, the text 'Venta total: C\$ 1364.00' is displayed, which is highlighted with a red box. The main content area lists seven sales transactions in a table format. Each row contains the transaction number (Nº), the client name, and the purchase and cancellation dates. The last transaction in the list is highlighted with a red box.

Nº	Cliente	Fecha Compra	Fecha de Cancelación
000006	Cliente: Cliente No Habitual	31/03/2025	31/03/2025
000005	Cliente: Samuel de Luque Lopez Santana	23/03/2025	-
000004	Cliente: Samuel de Luque Lopez Santana	22/03/2025	-
000003	Cliente: Cliente No Habitual	28/02/2025	-
000002	Cliente: Cliente No Habitual	28/02/2025	28/02/2025
000001	Cliente: Cliente No Habitual	23/02/2025	23/02/2025

Figura 146: Total de compras realizadas



The screenshot shows a web-based application interface for managing purchases. At the top, a dark header bar displays a user profile with the name 'Cristhiam Omar Suárez Núñez'. Below this, a navigation bar includes a link to 'Historial / Registro de Compras' and the title 'Compras Realizadas'. The main content area is a table with the following data:

Compra	Proveedor	Fecha Compra
Nº 000006	Proveedor: Coca-Cola	Fecha Compra: 30/01/2025
Nº 000005	Proveedor: EDT Nicaragua	Fecha Compra: 21/01/2025
Nº 000004	Proveedor: EDT Nicaragua	Fecha Compra: 15/01/2025
Nº 000003	Proveedor: EDT Nicaragua	Fecha Compra: 13/01/2025
Nº 000002	Proveedor: EDT Nicaragua	Fecha Compra: 12/01/2025
Nº 000001	Proveedor: EDT Nicaragua	Fecha Compra: 10/01/2025

A red box highlights the total amount 'Compra total: C\$ 481.25' located in the top right corner of the table header area.

At the bottom of the page, a copyright notice reads 'Copyright © Pulperia Jehova Jireh 2025'.

Figura 147: Reporte de productos

Jehova Jiréh ≡

Cristhiam Omar Suárez Núñez ▾

INICIO

Menú principal

Total de ventas

ADMINISTRACIÓN

Externa

Interna

Seguridad

OPERACIONES

Actividades

REGISTROS

Historial

Menú principal

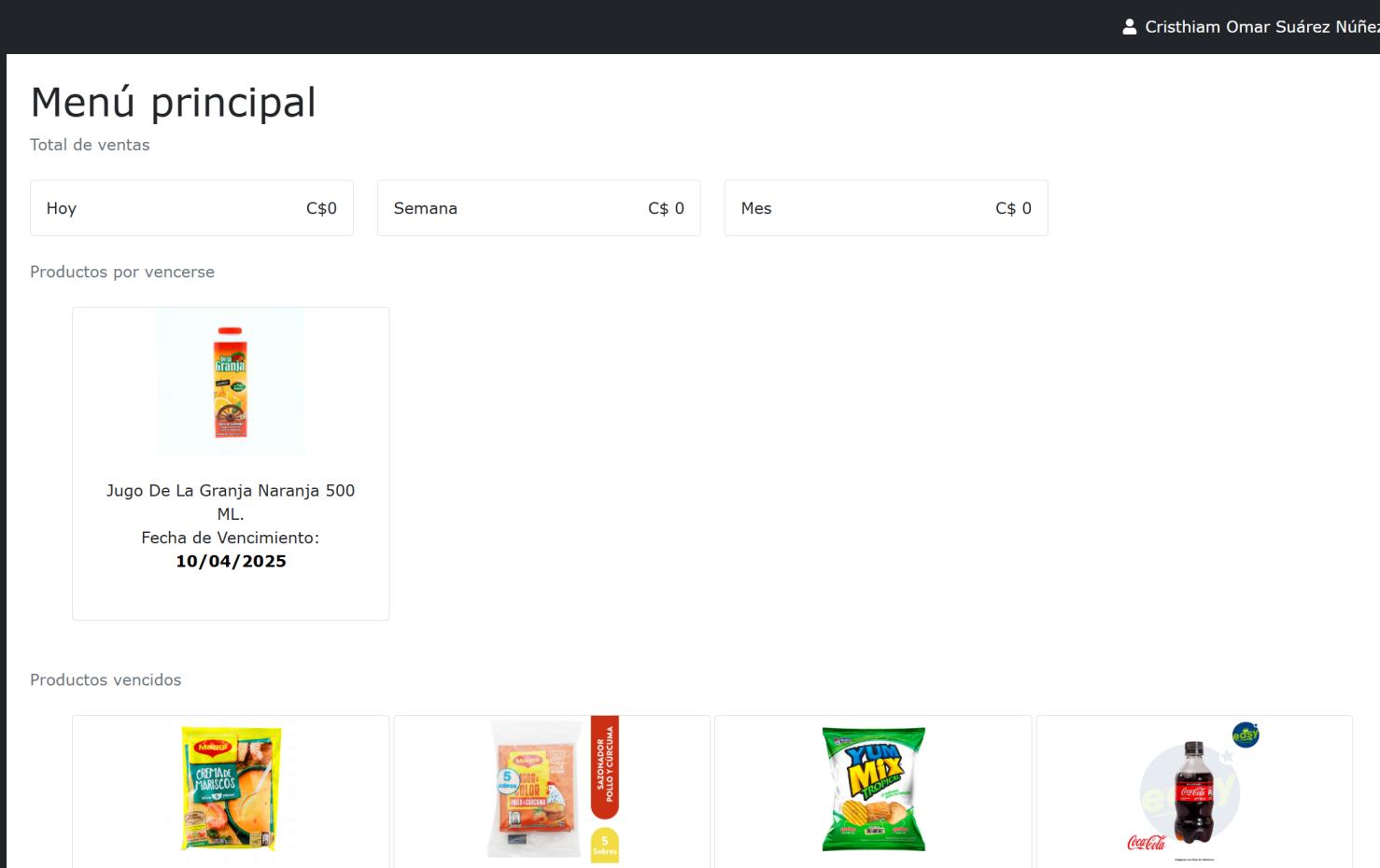
Total de ventas

Hoy C\$ 0 Semana C\$ 0 Mes C\$ 0

Productos por vencerse

Jugo De La Granja Naranja 500 ML.
Fecha de Vencimiento:
10/04/2025

Productos vencidos



Al finalizar la iteración, el Incremento fue presentado al cliente para la Sprint Review. Para ver más detalles, véase Figura 101: Acta de reunión sprint 4 del Anexo II.

Se entregó una suite de reportes y consultas esenciales para la gestión, lo que permite a la Pulpería Jehová Jireh pasar de la operación transaccional al análisis estratégico para la toma de decisiones.

Figura 148: Resumen del Sprint Review Sprint 4

Sprint Review	Valor entregado
Pruebas de Usuario	Se realizaron Pruebas de Aceptación de Usuario centradas en la validación de los reportes. El cliente confirmó la precisión de los datos mostrados en los informes y la utilidad estratégica de los mismos. La aprobación del cliente valida que la información es confiable y pertinente para monitorear el rendimiento del negocio.
Resultado Tangible	El resultado tangible fue la entrega de una versión del sistema con la Inteligencia de Negocio completamente operativa junto con la integración de Power BI. Este incremento proporcionó a la pulpería herramientas de visualización de datos para monitorear el rendimiento y obtener una ventaja competitiva.
Preparación	Con la entrega formal y la aprobación del cliente de la suite de reportes, el proyecto alcanza un punto de alta madurez funcional. La finalización exitosa del Sprint 4 marca la conclusión de las funcionalidades principales, dejando al sistema en un estado ideal para su despliegue y uso final.

Anexo IX. Sprint 5

El Sprint 5 fue el ciclo de trabajo final, un período de cuatro semanas enfocado en el cierre, la implantación y la estabilización del sistema. El objetivo principal fue preparar la plataforma para su entrega formal y puesta en producción, garantizando su operación óptima en el entorno de la empresa. Las tareas no se centraron en el desarrollo de nuevas funcionalidades, sino en la preparación técnica y la creación de documentación esencial para una transición fluida.

El trabajo se concentró en los hitos finales necesarios para la transferencia de la solución directamente en la pulpería.

Se priorizó el desarrollo del Manual de Usuario y el Diccionario de Datos. Estos entregables son de vital importancia para garantizar la correcta adopción de la plataforma por parte de los usuarios de la Pulpería Jehová Jireh, asegurando su autonomía y minimizando la necesidad de soporte técnico constante. A su vez, se preparó el entorno de producción para la entrega final. Esto incluyó la configuración del servidor local IIS y el despliegue del Sistema Web, asegurando que todos los componentes y bases de datos funcionaran de forma correcta.

Siguiendo este enfoque, se llevó a cabo un proceso de estabilización del sistema en el entorno real para corregir errores de última hora y optimizar el rendimiento. Por último, se realizó la migración de los datos históricos de la empresa al nuevo sistema, permitiendo a la plataforma estar lista para su uso inmediato desde el primer día de operación.

Al finalizar el Sprint 5, se realizó la entrega formal de todas las funcionalidades, documentos y el entorno operativo. El enfoque en la implementación, estabilización y migración de datos garantizó una transición fluida y exitosa para Jehová Jireh. La entrega del proyecto culminó con la aprobación total del cliente, marcando el cierre de la fase de ejecución del desarrollo.

Para ver más detalles, véase Figura 102: Acta de reunión sprint 5 del Anexo II.

Figura 149: Sprint Backlog del Sprint 5

FECHA DE INICIO	DIAS	PROGRESO
05/05/2025	27	100%

BACKLOG SIPRINT 5

CATEGORIA	ROL	ASIGNADO	ACCION	ESTADO	PRIORIDAD	HORAS
Tarea	Developer Team	Cristhiam Omar Suarez Nuñez	Desarrollo de Documentación Tecnica	Completada	Alta	12.00
Tarea	Developer Team	Hosmand Rafael Belli Cespedes	Desarrollo de Documentación Tecnica	Completada	Alta	12.00
Tarea	Developer Team	Cristhiam Omar Suarez Nuñez	Configuración de servidor local IIS	Completada	Alta	8.00
Tarea	Developer Team	Hosmand Rafael Belli Cespedes	Despliegue de Sistema Web de Jehova Jireh	Completada	Alta	8.00
Tarea	Developer Team	Cristhiam Omar Suarez Nuñez	Proceso de estabilización	Completada	Media	20.00
Tarea	Developer Team	Hosmand Rafael Belli Cespedes	Proceso de estabilización	Completada	Media	20.00
Tarea	Developer Team	Cristhiam Omar Suarez Nuñez	Migración de Datos	Completada	Media	2.00
Tarea	Developer Team	Hosmand Rafael Belli Cespedes	Migración de Datos	Completada	Media	2.00

Anexo X. Acta cierre de proyecto

Figura 150: Acta de cierre de proyecto

 UNI - Ingeniería en Computación	Sistema Web de Inventario y Facturación Jehová Jireh	
Cierre de Proyecto		
Titulo	Sistema Web de Inventario y Facturación para la pulperia Jehová Jireh	
Fecha	Granada, 05 de julio del 2025	
Lugar	Pulperia Jehová Jireh	
Participantes	Mariela Núñez Cabrera	Propietaria del Negocio
	Hosmand Rafael Belli Cespedes	Scrum Máster/Developer Team
	Cristhiam Omar Suarez Núñez	Product Owner/Developer Team
Motivo de Cierre		
Se completaron con éxito todas las fases del proyecto cumpliendo con los objetivos propuestos		
Aceptación de Entregables		
Titulo	Fecha de Entrega	Aceptación (Sí/No)
Sprint #1	23/11/2024	Sí
Sprint #2	11/01/2025	Sí
Sprint #3	22/02/2025	Sí
Sprint #4	12/04/2025	Sí
Sprint #5	31/05/2025	Sí
Diccionario de Datos	31/05/2025	Sí
Manual de Usuario	31/05/2025	Sí
Aclaraciones		
Los abajo firmantes establecen su conformidad con la culminación y entrega de este proyecto el cual ha sido revisado y validado por las personas autorizadas dando fe que reúnen los objetivos y requerimientos establecidos previamente definidos.		
El aplicativo entra en periodo de garantía durante 6 meses a partir del día de la fecha por lo cual el código fuente del producto u objetos de base de datos no debe ser modificado del original entregado.		
 Hosmand Rafael Belli Cespedes Scrum Máster/Developer Team		
 Cristhiam Omar Suarez Núñez Product Owner/Developer Team		
 Mariela del Socorro Núñez Cabrera Propietaria del Negocio		