

UNIVERSIDAD CONTINENTAL

FACULTAD DE INGENIERÍA

ESCUELA ACADÉMICO PROFESIONAL DE INGENIERÍA DE SISTEMAS E INFORMÁTICA



PROYECTO

**“Implementación de un Sistema de información para la Gestión de
Inventario de la ferretería “Crisol”**

PRESENTADO POR:

APELLIDOS Y NOMBRES	CÓDIGO
Flores Taipe Javier Paul	73209263
Reymundo Rodriguez Cristhian Jhon	75049277
Huali Cuyotupa Jhosep Steven	74975020

ASESOR:

DR. MAGLIONI ARANA CAPARACHIN

HUANCAYO – PERÚ

2025

RESUMEN

El proyecto de “Implementación de un Sistema de Información para la Gestión de Inventario de la Ferretería Crisol” tiene como objetivo principal desarrollar e implementar una solución tecnológica que optimice la gestión de inventarios en la ferretería "Crisol". Actualmente, la empresa enfrenta desafíos como la falta de control preciso sobre el stock de productos, la dificultad para realizar seguimientos en tiempo real y la ineficiencia en la gestión de pedidos y reposición de mercancías. El sistema propuesto busca automatizar los procesos de inventario, permitiendo un registro actualizado de entradas y salidas de productos, la generación de reportes en tiempo real, la gestión de proveedores y la alerta automática de niveles mínimos de stock. Para su desarrollo, se utilizó una metodología ágil que incluyó etapas de análisis de requisitos, diseño, implementación y pruebas. El sistema fue implementado utilizando tecnologías modernas como bases de datos relacionales, lenguajes de programación web y frameworks de desarrollo ágil. Los resultados obtenidos demostraron una mejora significativa en la eficiencia de la gestión de inventarios, reduciendo errores humanos, optimizando tiempos de respuesta y facilitando la toma de decisiones basada en datos precisos. Además, el sistema contribuyó a mejorar la satisfacción del cliente al garantizar la disponibilidad de productos y agilizar los procesos de venta.

Palabras Claves: Sistema de información, Gestión de inventarios, Ferretería Crisol, Automatización de procesos, Control de stock, Reportes en tiempo real, Tecnologías de desarrollo ágil, Optimización de recursos, Toma de decisiones, Satisfacción del cliente.

CONTENIDO

PORTADA	1
RESUMEN	2
CONTENIDO	6
CAPÍTULO I	6
PLANTEAMIENTO DEL ESTUDIO	6
1.1. Descripción del Problema	6
1.2. Delimitación del Problema	6
1.2.1. Delimitación económica	7
CAPÍTULO II	8
PLANEACIÓN DEL PROYECTO	8
2.1. Aspectos Generales de la Empresa	8
2.2. Diagnóstico del Problema	8
2.3. Procesos de la Empresa	8
2.3.1. Identificación de procesos	9
2.3.2. Descripción del proceso: “Recepción de Mercadería”	9
2.3.3. Descripción del proceso: “Almacenamiento y Clasificación”	9
2.3.4. Descripción del proceso: “Control de Inventario”	10
2.3.5 Descripción del proceso: “Venta de Facturación”	10
2.4. Detalles del Proyecto	11
2.5. Equipo de Desarrollo	11
2.5.1. Definición de roles	11
2.5.2. Actores del sistema	11
CAPÍTULO III	12
ANÁLISIS DEL PROYECTO	12
3.1. Análisis de Requerimientos	12
3.1.1. Metas del proyecto	12

3.1.2.Requerimientos del sistema	13
CAPÍTULO IV	15
ESPECIFICACIÓN DEL PROYECTO	15
4.1. Planificación de Pruebas	15
4.1.1.Alcance de pruebas	15
4.1.2.Criterios de aceptación y rechazo	17
4.1.3.Planificación y Organización	19
4.2. Estimación de costos	21
4.2.1.Métricas del proyecto	22
4.2.2.Medición del proyecto	23
4.2.3.Presupuesto del proyecto de desarrollo de software	24
CAPÍTULO V	25
DISEÑO DEL PROYECTO	25
5.1. Diseño del Negocio	25
5.1.1.Diagramas de caso de uso	25
5.1.2.Diagrama de clases	26
5.2. Diseño de Base de Datos	26
5.2.1.Diseño físico	27
5.2.2.Especificación de tablas de base de datos	29
5.3. Diseño de Interfaces	29
5.3.1.Interfaz de bienvenida	29
5.3.2.Interfaz de dashboard administrativo	30
5.3.3.Interfaz de módulo “...”	31
CAPÍTULO VI	32
IMPLEMENTACIÓN DEL PROYECTO	32
6.1. Pruebas de Programas	32
6.1.1.Principios básicos de las pruebas	32
6.1.2.Tipos de pruebas	32

6.1.3. Fases de proceso de prueba	32
6.1.4. Técnicas de prueba del software	38
6.2. Documentación de programas	40
6.3. Herramientas Software para Realizar las Pruebas	41
CONCLUSIONES	42
RECOMENDACIONES	43
REFERENCIAS BIBLIOGRÁFICAS	44
ANEXOS	45
ANEXO 01. Evidencia de Desarrollo	46

CAPÍTULO I

PLANTEAMIENTO DEL ESTUDIO

1.1. Descripción del Problema

La ferretería “Crisol” enfrenta dificultades en la gestión de su inventario debido a la dependencia de procesos manuales que generan inconsistencias en los registros, demoras en la consulta de datos y errores en el control de stock. Estas limitaciones afectan la eficiencia operativa y dificultan la toma de decisiones basadas en datos precisos, lo que pone en riesgo la rentabilidad del negocio.

1.2. Delimitación del Problema

El proyecto se enfocará en el desarrollo e implementación de un sistema de información que automatice los procesos de gestión de inventarios. Las funcionalidades principales incluirán:

- Registro de productos.
- Control y actualización del stock.
- Gestión de ventas y generación de reportes.
- Seguridad basada en roles de usuario (Administrador y Empleado).

Alcance Temporal: El proyecto tendrá una duración de **6 meses**.

Alcance Técnico: El sistema se implementará como una aplicación web, accesible desde dispositivos conectados a la red local de la ferretería.

1.2.1. Delimitación económica

El costo total estimado para el desarrollo del proyecto es de **USD 31,050**, considerando:

- **Costo del Personal:** USD 22,050.
- **Infraestructura y Licencias:** USD 5,000.
- **Capacitación y Gastos Administrativos:** USD 4,000.

Este presupuesto cubrirá todos los aspectos necesarios para garantizar la implementación exitosa del sistema.

1.2.2. Delimitación económica

La ferretería “Crisol” es un negocio familiar ubicado en AV. Ferrocarril N°1548 - El TAMBO - HUANCAYO, dedicado a la venta de materiales de construcción y herramientas. El sistema de información será utilizado principalmente por el **área de Ventas y Almacén**, quienes gestionarán el control de inventarios, las transacciones comerciales y la generación de reportes.

CAPÍTULO II

PLANEACIÓN DEL PROYECTO

2.1. Aspectos Generales de la Empresa

Nombre de la Empresa: Ferretería "Crisol"

Ubicación: AV. Ferrocarril N°1548 - El Tambo - Huancayo

Tipo de Empresa: Comercio minorista especializado en materiales de construcción, herramientas y productos ferreteros.

Misión: Brindar productos de calidad para satisfacer las necesidades del sector de la construcción y el bricolaje, garantizando un servicio ágil y eficiente.

Visión: Consolidarse como la ferretería líder en la región, reconocida por su innovación y gestión eficiente de inventarios.

2.2. Diagnóstico del Problema

La Ferretería "Crisol" enfrenta dificultades en la gestión de su inventario, lo que resulta en:

- Exceso de ciertos productos, generando costos innecesarios por almacenamiento.
- Faltante de productos clave, afectando la satisfacción del cliente.
- Procesos manuales que aumentan el riesgo de errores y retrasos en la actualización de inventarios.
- Falta de información en tiempo real para tomar decisiones rápidas y acertadas.

2.3. Procesos de la Empresa

2.3.1. Identificación de procesos

- Recepción de Mercadería
- Almacenamiento y Clasificación
- Control de Inventario
- Venta y Facturación

2.3.2.Descripción del proceso: “Recepción de Mercadería”

2.3.2.1.Objetivo y alcance del proceso.

Objetivo: Asegurar la entrada de productos en condiciones óptimas y su registro preciso en el sistema de inventario.

Alcance: Desde la llegada del proveedor hasta el registro en el sistema.

2.3.2.2.Cuadro de actividades.

Actividad	Responsable	Herramientas	Duración Aproximada
Verificación de guía de despacho	Encargado de almacén	Documentos del proveedor	15 minutos
Inspección de productos	Encargado de almacén	Lista de control	20 minutos
Registro en el sistema	Administrativo	Sistema de gestión	10 minutos
Almacenamiento inicial	Personal de almacén	Carretillas y estanterías	15 minutos

2.3.3.Descripción del proceso: “Almacenamiento y Clasificación”

2.3.3.1.Objetivo y alcance del proceso.

Objetivo: Monitorear y actualizar el inventario para evitar faltantes o excesos de productos.

Alcance: Desde la verificación periódica del inventario hasta la generación de reportes.

2.3.3.2.Cuadro de actividades.

Actividad	Responsable	Herramientas	Duración Aproximada
Conteo físico de productos	Personal de almacén	Escáner de códigos	2 horas
Comparación con registros	Administrativo	Sistema de gestión	30 minutos
Ajustes en el sistema	Administrativo	Sistema de gestión	20 minutos

Generación de reportes	Supervisor	Sistema de reportes	15 minutos
------------------------	------------	---------------------	------------

2.3.4.Descripción del proceso: “Control de Inventario”

2.3.4.1.Objetivo y alcance del proceso.

Objetivo: Monitorear y actualizar el inventario para evitar faltantes o excesos de productos.

Alcance: Desde la verificación periódica del inventario hasta la generación de reportes.

2.3.4.2.Cuadro de actividades.

Actividad	Responsable	Herramientas	Duración Aproximada
Conteo físico de productos	Personal de almacén	Escáner de códigos	2 horas
Comparación con registros	Administrativo	Sistema de gestión	30 minutos
Ajustes en el sistema	Administrativo	Sistema de gestión	20 minutos
Generación de reportes	Supervisor	Sistema de reportes	15 minutos

2.3.5.Descripción del proceso: “Venta y Facturación”

2.3.5.1.Objetivo y alcance del proceso.

Objetivo: Garantizar un proceso de venta eficiente y preciso, asegurando la satisfacción del cliente y el registro correcto de las transacciones.

Alcance: Desde la selección de productos hasta la entrega del comprobante de venta.

2.3.5.2.Cuadro de actividades.

Actividad	Responsable	Herramientas	Duración Aproximada
-----------	-------------	--------------	---------------------

Selección de productos	Cliente	Catálogo o mostrador	Variable
Escaneo de productos	Cajero	Escáner de códigos	5 minutos por cliente
Generación de factura	Cajero	Sistema de facturación	2 minutos por cliente
Cobro	Cajero	Caja registradora o POS	3 minutos por cliente
Entrega de productos	Personal de almacén	Ninguna	5 minutos

2.4. Detalles del Proyecto

2.5. Equipo de Desarrollo

2.5.1. Definición de roles

- **Líder del Proyecto:** Planifica y dirige el desarrollo del sistema.
- **Analista de Sistemas:** Identifica problemas y propone soluciones tecnológicas.
- **Desarrollador:** Implementa el código y las funcionalidades del sistema.
- **Tester:** Detecta errores y asegura que el sistema cumpla con los requisitos.

2.5.2. Actores del sistema

- **Administrador:** Gestiona el inventario y genera reportes.
- **Encargado de Almacén:** Registra la recepción y salida de productos.
- **Cliente:** Consulta disponibilidad de productos (en caso de un módulo de acceso externo).
- **Proveedor:** Consulta órdenes de compra y entregas pendientes (opcional).

CAPÍTULO III

ANÁLISIS DEL PROYECTO

3.1. Análisis de Requerimientos

En este apartado se describe la recopilación de las necesidades y expectativas tanto de los usuarios como de los stakeholders del sistema. Para implementar el sistema de información en la ferretería Crisol, se deben identificar los objetivos que el sistema debe cumplir, la forma en que los usuarios interactúan con el sistema y las especificaciones que deben satisfacerse.

- **Identificación de los usuarios del sistema:** El personal encargado de la gestión de inventarios, los gerentes, y otros usuarios involucrados, como personal de ventas y almacén.
- **Proceso de gestión de inventario actual:** Evaluación de los métodos manuales o semiautomáticos que se utilizan actualmente para gestionar el inventario.
- **Problemas a resolver:** Inconvenientes con la precisión de los inventarios, pérdidas de productos, desactualización de la información, y falta de visibilidad en tiempo real.
- **Expectativas y objetivos del sistema:** Mejorar la precisión del inventario, reducir los tiempos de gestión y mejorar la toma de decisiones con base en información precisa y actualizada.

3.1.1. Metas del proyecto

- **Automatización de la Gestión de Inventarios:**
 - Reducir la intervención manual en el proceso de registro y control de productos en el inventario.
 - Automatizar el proceso de actualización del inventario en tiempo real para facilitar el seguimiento y control.
- **Mejora de la Precisión de los Inventarios:**

Implementar un sistema que permita registrar las entradas y salidas de productos de forma precisa, con lo que se reducirá el margen de error.
- **Accesibilidad de la Información:**

Desarrollar un sistema accesible desde diferentes dispositivos para los diferentes usuarios del sistema, facilitando la consulta de inventarios y la toma de decisiones desde cualquier lugar de la empresa.

- **Generación de Informes y Reportes:**

Proveer de herramientas para generar informes detallados sobre el estado del inventario, productos más vendidos, productos con bajo stock, entre otros.

- **Optimización de la Toma de Decisiones:**

Facilitar la toma de decisiones estratégicas al proporcionar datos precisos sobre las existencias y el flujo de productos.

3.1.2.Requerimientos del sistema

3.1.2.1.Requisitos funcionales.

- Registrar y actualizar información sobre los productos: nombre, descripción, código de barras, cantidad disponible, precio de compra y venta, entre otros.
- Permitir la búsqueda de productos por diferentes criterios: nombre, categoría, proveedor, etc.
- Gestionar las categorías de los productos para facilitar su organización.
- Registrar las entradas de productos (compras, devoluciones de clientes, etc.).
- Registrar las salidas de productos (ventas, transferencias, pérdidas, etc.).
- Actualizar automáticamente el inventario en tiempo real con cada entrada o salida registrada.
- Generar alertas cuando el stock de un producto llegue a un nivel mínimo predeterminado.
- Enviar notificaciones por correo electrónico o dentro del sistema cuando se realicen movimientos importantes de inventario.
- Mantener un registro de los proveedores y las compras realizadas.
- Generar órdenes de compra a los proveedores cuando el inventario esté bajo.
- Crear reportes de inventarios, ventas, entradas y salidas, y de productos con bajo stock.
- Permitir la exportación de reportes en formatos como PDF y Excel.
- Permitir autenticación y autorización para que solo usuarios autorizados puedan acceder a funciones específicas del sistema.

- Permitir definir roles de usuarios con permisos específicos, como administrador, encargado de inventario, gerente, etc.

3.1.2.2.Requisitos no funcionales.

- El sistema debe ser capaz de manejar un alto volumen de transacciones y consultas de inventario sin afectar su desempeño.
- El tiempo de respuesta para consultas y registros debe ser menor a 2 segundos.
- El sistema debe garantizar la seguridad de los datos mediante el uso de cifrado y autenticación de usuarios.
- Se deben implementar copias de seguridad automáticas para evitar la pérdida de datos.
- El sistema debe ser fácil de usar, con interfaces intuitivas que permitan a los usuarios realizar tareas sin necesidad de una formación extensa.
- La interfaz debe ser accesible desde diferentes dispositivos (computadoras, tabletas, móviles).
- El sistema debe ser escalable para soportar el crecimiento de la ferretería, tanto en términos de volumen de productos como de número de usuarios.
- El sistema debe ser compatible con los principales navegadores web y sistemas operativos, para garantizar su acceso desde cualquier dispositivo.
- El sistema debe permitir actualizaciones periódicas y mantenimiento sin afectar su operatividad.

CAPÍTULO IV

ESPECIFICACIÓN DEL PROYECTO

4.1. Planificación de Pruebas

En este apartado se detalla el alcance de las pruebas, identificando los elementos que serán evaluados, las funcionalidades específicas a probar y las que no serán consideradas dentro del alcance de este proyecto.

4.1.1. Alcance de pruebas

El alcance de las pruebas comprende todos los módulos y funcionalidades críticas del sistema de información para la gestión de inventarios de la ferretería "Crisol". Las pruebas incluyen verificaciones de funcionalidad, usabilidad, rendimiento y seguridad.

4.1.1.1. Elementos de pruebas.

Interfaz de Usuario (UI):

- Validación de la usabilidad, accesibilidad y diseño.
- Comprobación de navegabilidad y fluidez en la interacción.

Base de Datos (BD):

- Integridad y consistencia de los datos almacenados.
- Validación de relaciones entre tablas y correcto funcionamiento de consultas.

Lógica del Negocio:

- Verificación de reglas específicas implementadas (gestión de inventario, cálculos de ventas, etc.).
- Correcta ejecución de las transacciones y procesos automatizados.

Seguridad:

- Autenticación y autorización de usuarios según roles (Administrador y Empleado).
- Protección contra vulnerabilidades como inyecciones SQL.

4.1.1.2. Funcionalidad a probar.

Gestión de Productos:

- Crear, editar, eliminar y consultar productos.
- Validación de los límites de stock mínimo y alerta visual en la interfaz.

Gestión de Inventario:

- Registro de movimientos de entrada y salida de productos.
- Actualización automática del stock al procesar transacciones.

Gestión de Ventas:

- Registro de ventas, cálculo de subtotales y totales.
- Generación de comprobantes de venta en formato PDF.

Gestión de Usuarios:

- Validación de inicio de sesión con credenciales correctas e incorrectas.
- Gestión de roles (Administrador y Empleado) y restricción de funcionalidades.

Generación de Reportes:

- Creación de reportes de inventario y ventas por períodos específicos.
- Exportación de reportes a formatos como Excel o PDF.

4.1.1.3.Funcionalidades a no probar.

Integración con sistemas externos:

- No se probará la conectividad o integración con otros sistemas, ya que este sistema es autónomo.

Pruebas en entornos de alta concurrencia:

- El sistema no está diseñado para grandes volúmenes de usuarios concurrentes (más de 50).

Módulos adicionales o ampliaciones futuras:

- Cualquier funcionalidad que no esté especificada en los requisitos iniciales.

Compatibilidad multiplataforma:

Las pruebas solo considerarán navegadores modernos y sistemas operativos estándar (Windows y navegadores como Chrome o Firefox).

4.1.2. Criterios de aceptación y rechazo

Este apartado detalla las condiciones bajo las cuales las pruebas serán consideradas exitosas (aceptación), fallidas (rechazo), y los criterios específicos para suspender o reanudar las pruebas en función de incidentes críticos.

4.1.2.1. Criterios de aceptación.

Cumplimiento de Requisitos Funcionales:

- Las funcionalidades principales (gestión de inventario, productos, ventas, usuarios, reportes) operan conforme a las especificaciones iniciales.
- Los cálculos (totales de ventas, actualización de inventario) son precisos y consistentes.

Interfaz de Usuario (UI):

- La interfaz es intuitiva, navegable y funcional para los roles definidos (Administrador y Empleado).
- No se presentan errores de diseño que afecten la experiencia del usuario.

Rendimiento y Estabilidad:

- El sistema responde en un tiempo máximo de 2 segundos para funciones críticas bajo condiciones normales de uso.
- Soporte para al menos 50 usuarios concurrentes sin pérdida de datos ni caídas del sistema.

Integridad de Datos:

- Los datos se almacenan correctamente en la base de datos.
- No se presentan inconsistencias en las operaciones de consulta, actualización o eliminación de registros.

Seguridad:

- Se valida correctamente la autenticación de usuarios y los permisos según roles.

- El sistema protege los datos sensibles frente a intentos de acceso no autorizado.

4.1.2.2.Criterios de rechazo.

Errores Críticos:

- Fallas en las funcionalidades principales que impiden su uso (por ejemplo, incapacidad para registrar ventas o actualizar el inventario).
- Errores en cálculos financieros o en los reportes generados.

Fallas de Seguridad:

- Vulnerabilidades críticas detectadas (por ejemplo, accesos no autorizados, inyección de SQL).

Problemas de Rendimiento:

- Tiempo de respuesta superior a 2 segundos en operaciones críticas.
- Caídas frecuentes o pérdida de datos bajo condiciones normales de uso.

Deficiencias en la Interfaz:

Interfaz poco intuitiva o errores que dificulten el uso del sistema.

4.1.2.3.Criterios de suspensión.

Incidentes Mayores:

Errores críticos en funciones esenciales que bloquean el avance de las pruebas.

Caída repetida del sistema durante las pruebas.

Falta de Recursos:

No disponibilidad de datos, herramientas o infraestructura necesarios para continuar con las pruebas.

Falta de Documentación:

La ausencia de especificaciones claras que impidan validar correctamente los resultados de las pruebas.

4.1.2.4. Criterios de reanudación.

Corrección de Errores Críticos:

- Los errores o incidentes que causaron la suspensión han sido resueltos y validados por el equipo técnico.

Disponibilidad de Recursos:

- Se restablecen los datos, herramientas o infraestructura necesaria para la ejecución de las pruebas.

Revisión de Documentación:

- Las especificaciones han sido aclaradas y aprobadas para continuar con el proceso de pruebas.

4.1.3. Planificación y Organización

Los procedimientos para las pruebas describen el conjunto de pasos organizados que se deben seguir durante la ejecución de las pruebas para garantizar que el sistema de información para la gestión de inventarios de la ferretería "Crisol" cumple con los requisitos establecidos.

A) Identificación de Tipos de Pruebas

Pruebas Unitarias:

- Validación de cada módulo individualmente (gestión de productos, ventas, inventario, etc.).
- Objetivo: Garantizar que cada componente funciona de manera independiente.

Pruebas de Integración:

- Evaluación del flujo de datos entre módulos relacionados (por ejemplo, actualización de inventario tras realizar una venta).
- Objetivo: Confirmar que los módulos trabajan correctamente en conjunto.

Pruebas del Sistema:

- Simulación del sistema completo en un entorno que represente la operación real.
- Objetivo: Garantizar el cumplimiento de los requisitos funcionales y no funcionales.

Pruebas de Aceptación del Usuario (UAT):

- Ejecución de pruebas con usuarios finales para validar que el sistema satisface sus necesidades.
- Objetivo: Lograr la aprobación para la implementación.

4.1.3.1.Procedimientos para las pruebas.

B) Procedimientos para la Ejecución de Pruebas

Paso 1: Preparación

- 1. Definir los casos de prueba:**
 - Crear un conjunto de escenarios de prueba basados en los requisitos funcionales y no funcionales.
- 2. Configurar el entorno de pruebas:**
 - Implementar el sistema en un entorno de pruebas independiente que simule las condiciones reales.
- 3. Cargar datos iniciales:**
 - Población de la base de datos con información necesaria para ejecutar las pruebas (productos, usuarios, etc.).

Paso 2: Ejecución de Pruebas

- 1. Ejecutar casos de prueba unitarios:**
 - Probar cada módulo de forma aislada siguiendo los escenarios definidos.

2. Realizar pruebas de integración:

- Ejecutar casos que involucren el flujo de datos entre diferentes módulos.

3. Evaluar el sistema completo:

- Realizar pruebas funcionales, de rendimiento, de seguridad y de usabilidad.

4. Recopilar resultados:

- Registrar los resultados de cada prueba, especificando si se cumplen los criterios de aceptación.

Paso 3: Análisis de Resultados

1. Documentar errores detectados:

- Describir cada incidencia con detalles como: descripción del error, pasos para reproducirlo, y módulo afectado.

2. Revisión y corrección:

- Remitir los errores al equipo de desarrollo para su solución.

3. Repetir pruebas:

- Volver a ejecutar los casos de prueba afectados por los errores corregidos.

Paso 4: Validación Final

1. Pruebas de Aceptación del Usuario (UAT):

- Permitir que los usuarios finales realicen pruebas en condiciones reales.
- Recopilar comentarios y realizar ajustes necesarios.

2. Informe de pruebas:

- Generar un informe detallado que incluya:
 - Resultados obtenidos.
 - Incidencias detectadas y su resolución.
 - Recomendaciones finales.

4.2. Estimación de costos

En esta sección, se realiza un análisis detallado de los recursos, tiempo, esfuerzo y presupuesto necesario para desarrollar el sistema de información para la gestión de inventarios de la ferretería "Crisol".

4.2.1. Métricas del proyecto

4.2.2. Medición del proyecto

4.2.2.1. Cálculo de puntos de función.

El método de puntos de función (Function Point Analysis - FPA) se utiliza para estimar el tamaño del sistema basándose en funcionalidades a desarrollar.

Identificación de Componentes:

- **Entradas Externas (EI):** Formularios como registro de productos, ventas, usuarios.
- **Salidas Externas (EO):** Reportes de ventas e inventarios.
- **Consultas Externas (EQ):** Consultas rápidas de productos o ventas.
- **Archivos Lógicos Internos (ILF):** Base de datos con tablas de productos, usuarios, inventarios.
- **Interfaz con Archivos Externos (EIF):** Ninguna, sistema autónomo.

Asignación de Complejidad y Puntos:

- **Entradas Externas (10 EI):** Complejidad media (4 puntos/EI) = 40 puntos.
- **Salidas Externas (5 EO):** Complejidad media (5 puntos/EO) = 25 puntos.
- **Consultas Externas (8 EQ):** Complejidad baja (3 puntos/EQ) = 24 puntos.
- **Archivos Lógicos Internos (6 ILF):** Complejidad alta (10 puntos/ILF) = 60 puntos.
- **Total de Puntos de Función (FP): 149 puntos.**

4.2.2.2. Productividad del equipo de trabajo.

Tasa de Productividad:

Basado en métricas estándar de la industria:

- 10 Puntos de Función por desarrollador por mes.

Esfuerzo Total Estimado:

149 FP / 10 FP por mes/desarrollador = 14.9 meses-hombre.

4.2.2.3. Estimación de esfuerzo y personal necesario.

Esfuerzo Total en Meses-Hombre:

- 14.9 meses-hombre (redondeado a 15).

Distribución del Personal:

- **Desarrolladores:** 2 (dedicación del 80%).
- **Tester/QA:** 1 (dedicación del 50%).
- **Analista de Requisitos:** 1 (dedicación del 25%).
- **Líder de Proyecto:** 1 (dedicación del 20%).

Duración Estimada del Proyecto:

- **6 meses** con el equipo propuesto.

4.2.2.4. Costo del personal del proyecto.

Costo por Rol (promedio mensual):

- Desarrollador: 1,200/mes.
- Tester/QA: 1,000/mes.
- Analista: 1,500/mes.
- Líder de Proyecto: 2,000/mes.

Cálculo Total del Costo del Personal:

- Desarrolladores: $2 \times \text{USD } 1,200 \times 6 \text{ meses} = \mathbf{14,400}$.
- Tester/QA: $0.5 \times \text{USD } 1,000 \times 6 \text{ meses} = \mathbf{3,000}$.
- Analista: $0.25 \times \text{USD } 1,500 \times 6 \text{ meses} = \mathbf{2,250}$.
- Líder de Proyecto: $0.2 \times \text{USD } 2,000 \times 6 \text{ meses} = \mathbf{2,400}$.
- **Total Costo del Personal: 22,050.**

4.2.2.5. Costo de unidades de medida del proyecto.

Infraestructura y Licencias:

- Computadoras y hardware: 3,000.
- Licencias de software (IDE, herramientas de pruebas, etc.): 2,000.

Capacitación del Equipo:

- Talleres y certificaciones: 1,000.

Gastos Administrativos:

Energía, internet, oficina: USD 500/mes × 6 meses = 3,000.

4.2.3. Presupuesto del proyecto de desarrollo de software

Elementos	Costos
Costo del personal	22 050
Infraestructura y Licencias	5 000
Capacitación	1 000
Gastos Administrativos	3 000
TOTAL ESTIMADO	31 050

CAPÍTULO V

DISEÑO DEL PROYECTO

5.1. Diseño del Negocio

El diseño del negocio se basa en el análisis de las necesidades de la ferretería "Crisol" para optimizar la gestión de su inventario. Este apartado incluye el diseño de diagramas que representan las interacciones del sistema y su estructura interna.

5.1.1. Diagramas de caso de uso

El diagrama de casos de uso ilustra las interacciones entre los actores y el sistema. Los actores principales son:

Administrador: Encargado de gestionar productos, proveedores y usuarios.

Empleado: Responsable del registro de ventas y actualización del inventario.

Sistema de Información: Automatiza las tareas relacionadas con el control de inventario.

Casos de Uso Identificados:

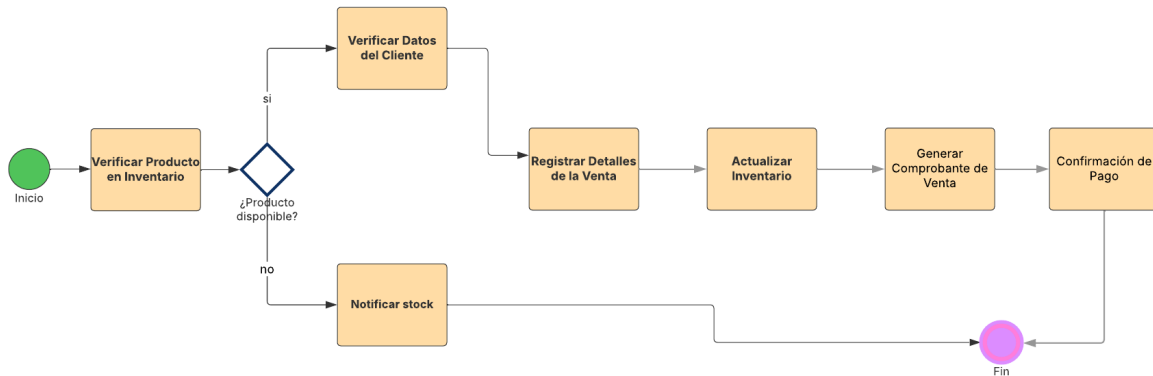
- 1. Gestión de Productos:**
 - Crear, actualizar, eliminar y consultar productos.
- 2. Gestión de Inventario:**
 - Registro de entradas y salidas.
 - Control de stock mínimo.
- 3. Gestión de Ventas:**
 - Registro de ventas realizadas.
 - Generación de comprobantes.
- 4. Gestión de Proveedores:**
 - Registrar y actualizar información de proveedores.
- 5. Generación de Reportes:**
 - Reportes de inventario y ventas.

Diagrama de Caso de Uso:

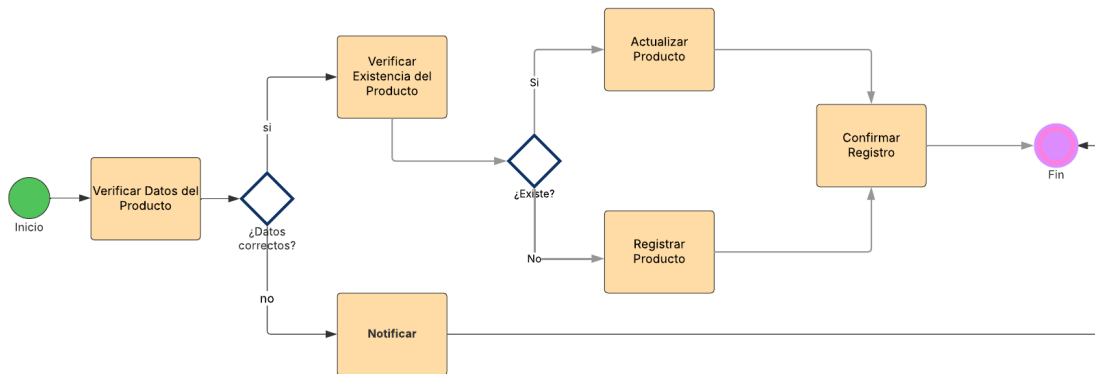
Incluye los actores, sus relaciones con los casos de uso, y las funcionalidades principales.

Diseño Lógico:

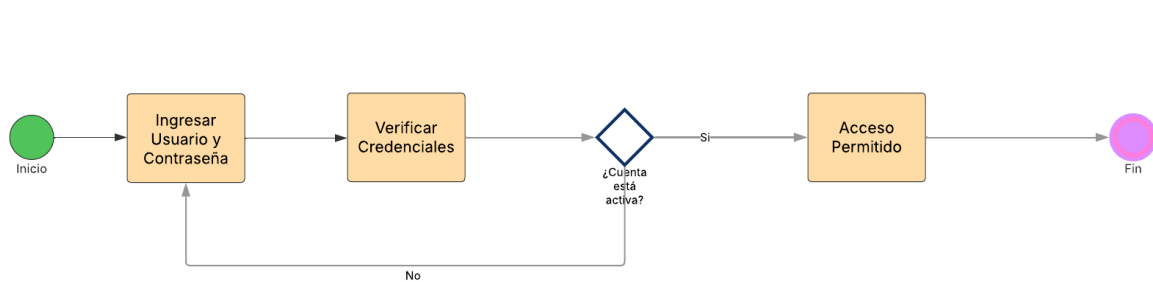
Registro de Ventas



Registro de Productos



Registro de Usuarios



5.1.2. Diagrama de clases

El diagrama de clases modela la estructura del sistema, representando las clases, sus atributos, métodos y relaciones.

a) Clases Identificadas:

1. **Producto:**

- **Atributos:** IDProducto, nombre, categoría, precio, stock, stockMinimo.
- **Métodos:** agregarProducto(), actualizarProducto(), eliminarProducto().

2. **Proveedor:**

- **Atributos:** IDProveedor, nombre, dirección, contacto.
- **Métodos:** registrarProveedor(), actualizarProveedor().

3. **Venta:**

- **Atributos:** IDVenta, fecha, total, detallesVenta.
- **Métodos:** registrarVenta(), generarComprobante().

4. **Usuario:**

- **Atributos:** IDUsuario, nombre, rol, contraseña.
- **Métodos:** autenticarUsuario(), gestionarUsuarios().

5. **Inventario:**

- **Atributos:** IDInventario, fecha, cantidad, tipoMovimiento.
- **Métodos:** registrarMovimiento(), consultarStock().

b) Relaciones:

- Un **Producto** puede estar asociado a varios registros en el **Inventario**.
- Una **Venta** puede contener múltiples **Productos**.
- Un **Proveedor** puede suministrar varios **Productos**.
- Un **Usuario** puede realizar acciones de gestión.

5.2. Diseño de Base de Datos

5.2.1. Diseño físico

El diseño físico define cómo los datos serán almacenados en el sistema de gestión de bases de datos (SGBD). Se sugiere el uso de MySQL debido a su robustez y escalabilidad.

Estructura General

El diseño incluye las siguientes tablas principales:

1. Productos

```
1  <?php
2
3  use Illuminate\Database\Migrations\Migration;
4  use Illuminate\Database\Schema\Blueprint;
5  use Illuminate\Support\Facades\Schema;
6
7  return new class extends Migration
8  {
9      /**
10       * Run the migrations.
11       */
12     public function up()
13     {
14         Schema::create('productos', function (Blueprint $table) {
15             $table->id();
16             $table->string('nombre');
17             $table->decimal('precio', 10, 2);
18             $table->integer('stock');
19             $table->unsignedBigInteger('proveedor_id');
20             $table->foreign('proveedor_id')->references('id')->on('proveedores')->onDelete('cascade');
21             $table->timestamps();
22         });
23     }
24
25     /**
26      * Reverse the migrations.
27      */
28     public function down(): void
29     {
30         Schema::dropIfExists('productos');
31     }
32 };
33
```

2. Proveedores

```

1  <?php
2
3  use Illuminate\Database\Migrations\Migration;
4  use Illuminate\Database\Schema\Blueprint;
5  use Illuminate\Support\Facades\Schema;
6
7  return new class extends Migration
8  {
9      /**
10       * Run the migrations.
11       */
12     public function up()
13     {
14         Schema::create('proveedores', function (Blueprint $table) {
15             $table->id();
16             $table->string('nombre');
17             $table->string('telefono');
18             $table->string('email')->unique();
19             $table->timestamps();
20         });
21     }
22
23     /**
24      * Reverse the migrations.
25      */
26     public function down(): void
27     {
28         Schema::dropIfExists('proveedores');
29     }
30 };
31

```

3. **Usuarios**

4. **Ventas**

```

1  <?php
2
3  use Illuminate\Database\Migrations\Migration;
4  use Illuminate\Database\Schema\Blueprint;
5  use Illuminate\Support\Facades\Schema;
6
7  return new class extends Migration {
8      public function up() {
9          Schema::create('ventas', function (Blueprint $table) {
10             $table->id();
11             $table->string('cliente');
12             $table->decimal('total', 10, 2)->default(0);
13             $table->timestamps();
14         });
15
16         Schema::create('producto_venta', function (Blueprint $table) {
17             $table->id();
18             $table->unsignedBigInteger('venta_id');
19             $table->unsignedBigInteger('producto_id');
20             $table->integer('cantidad');
21             $table->decimal('subtotal', 10, 2);
22
23             $table->foreign('venta_id')->references('id')->on('ventas')->onDelete('cascade');
24             $table->foreign('producto_id')->references('id')->on('productos')->onDelete('cascade');
25         });
26     }
27
28     public function down() {
29         Schema::dropIfExists('producto_venta');
30         Schema::dropIfExists('ventas');
31     }
32 };
33

```

5. Detalles de Ventas

6. Inventario

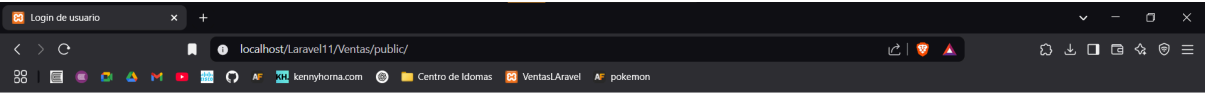
Características Técnicas

- Motor de almacenamiento: InnoDB (para manejo de transacciones y claves foráneas).
- Clave primaria (PK): Identificadores únicos para cada tabla.
- Claves foráneas (FK): Relaciones entre tablas para mantener la integridad referencial.

5.2.2. Especificación de tablas de base de datos

5.3 Diseño de Interfaces

5.3.1. Interfaz de bienvenida



Crisol 

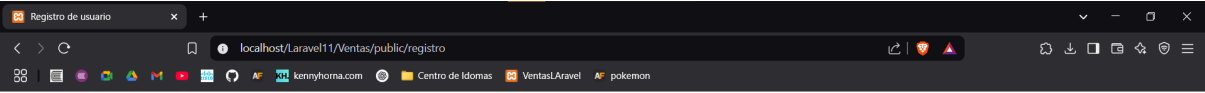
Inicia sesión para continuar


Correo

Contraseña

Entrar

Regístrate aquí



Crisol 

Regístrate para comenzar

Nombre

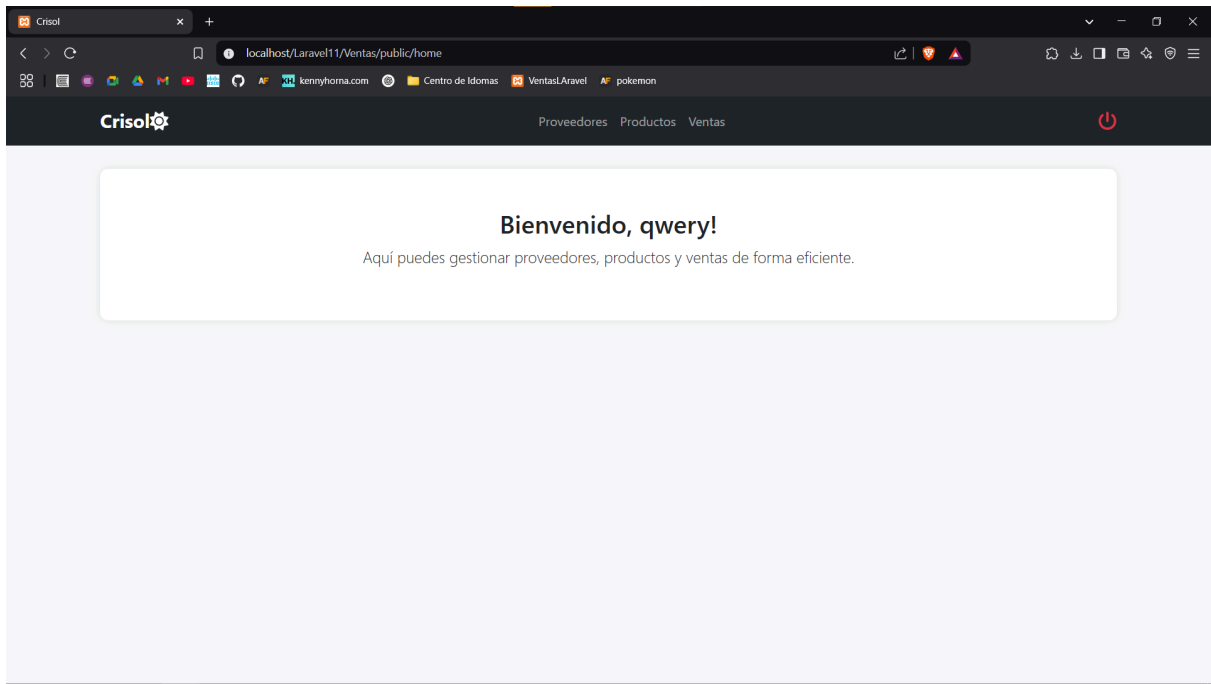
Correo

Contraseña

Confirmar Contraseña

Registrarse

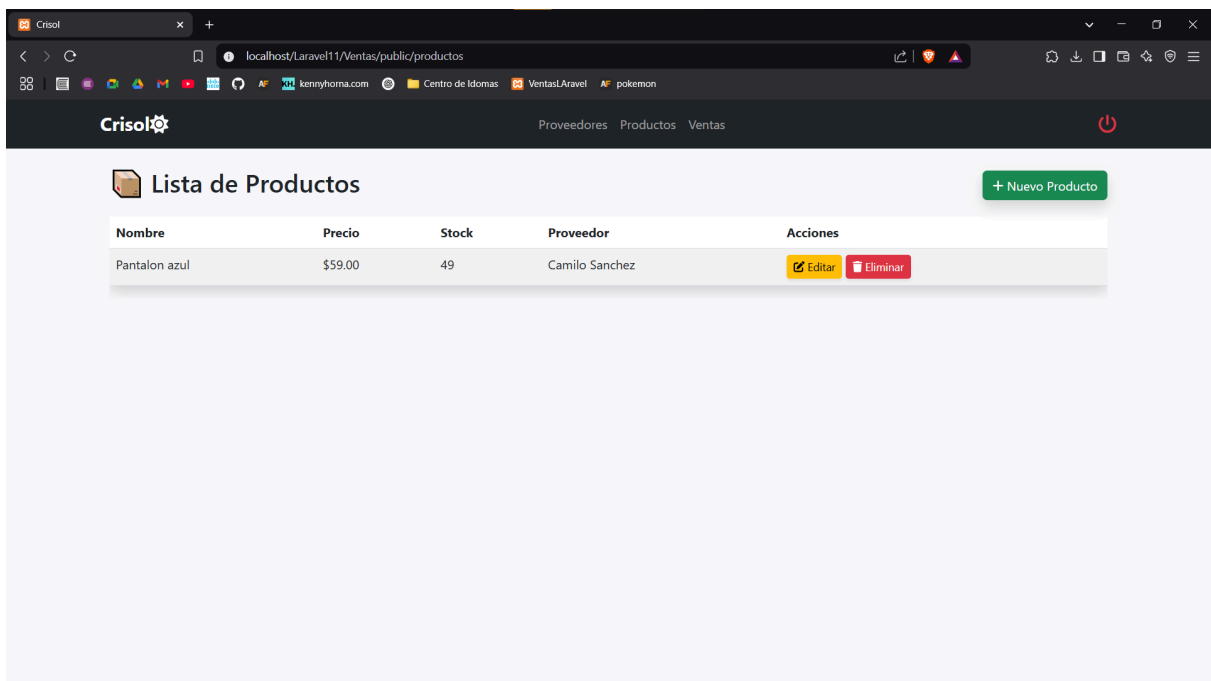
Iniciar sesión



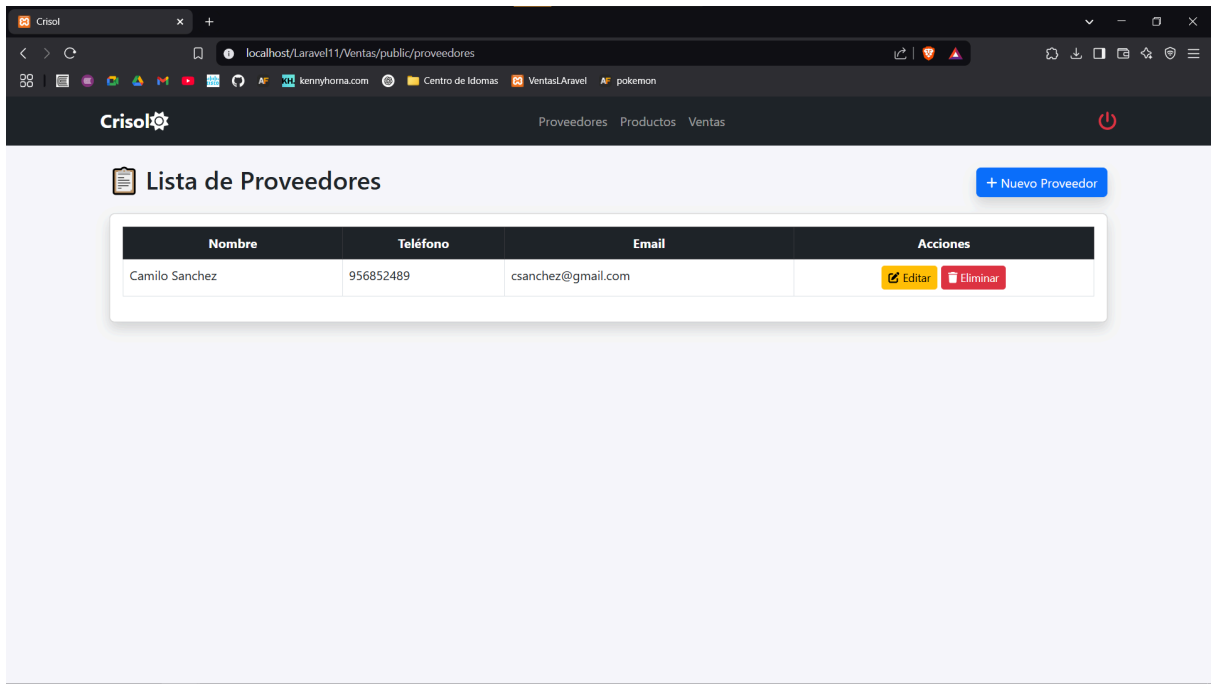
5.3.2. Interfaz de dashboard administrativo

5.3.3. Interfaz de módulos

5.3.3.1. Modulo “Producto”

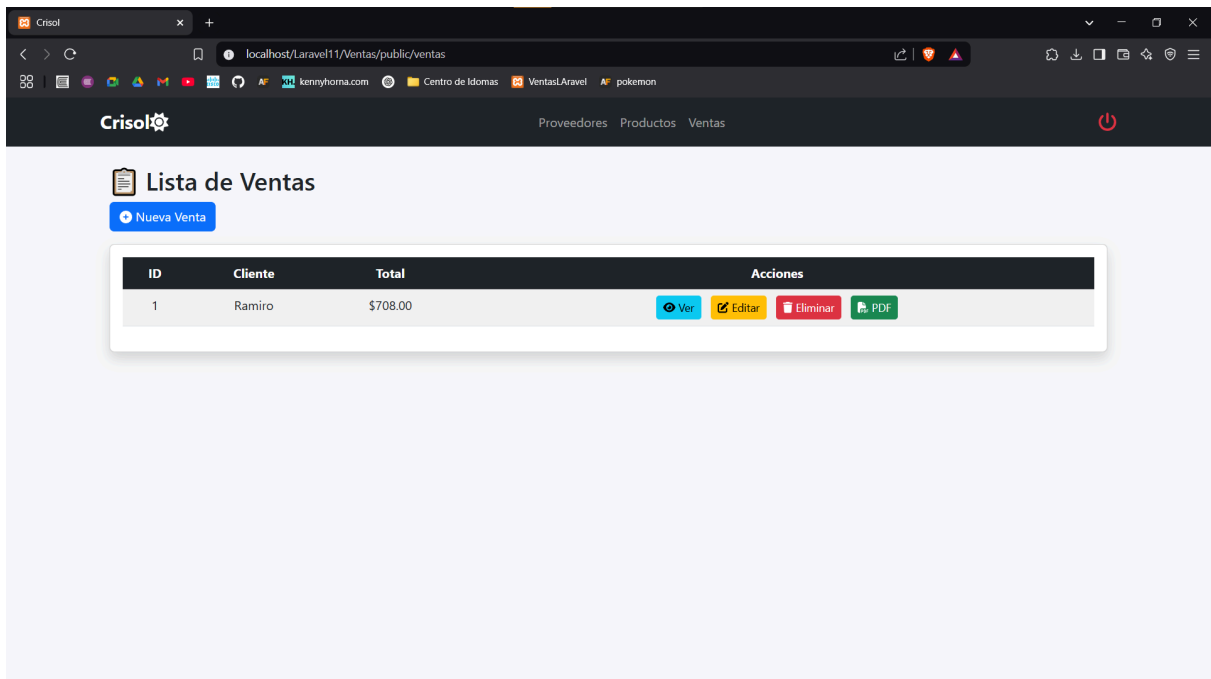


5.3.3.2. Módulo “Proveedor”



5.3.3.3. Modulo “Usuarios”

5.3.3.4. Modulo “Ventas”x



5.3.3.5 Modulo “Detalle de Ventas”

5.3.3.6 Modulo “Inventario”

CAPÍTULO VI

IMPLEMENTACIÓN DEL PROYECTO

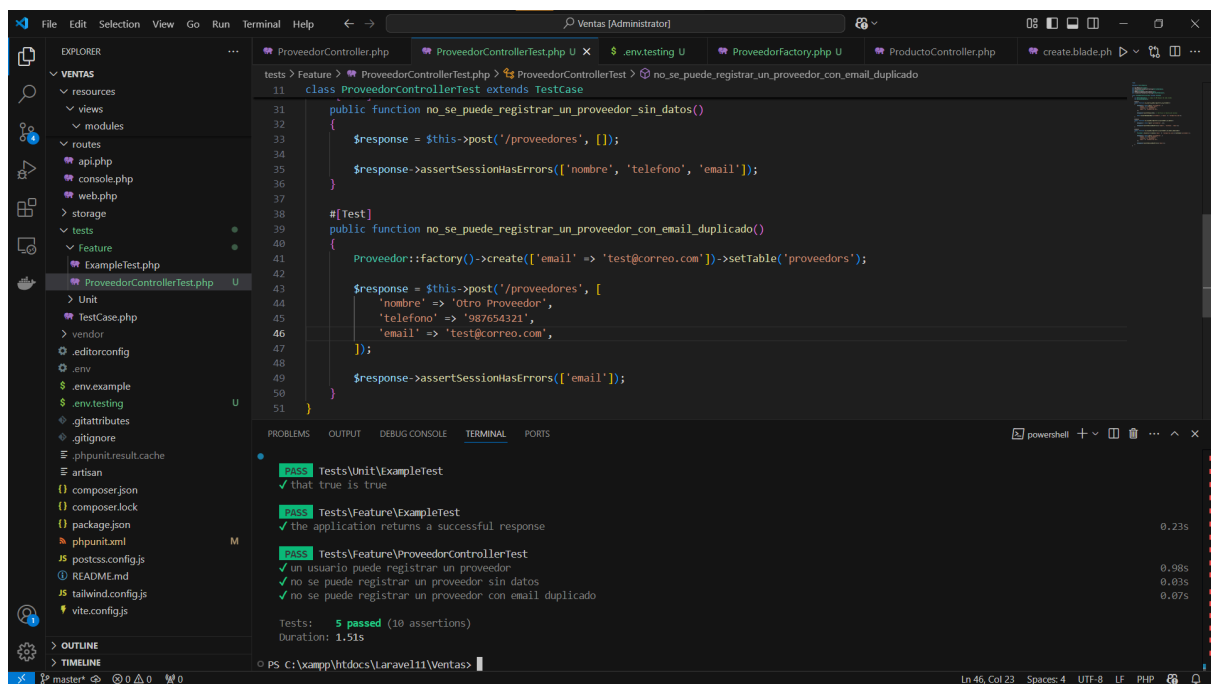
6.1. Pruebas de Programas

6.1.1.Principios básicos de las pruebas

6.1.2.Tipos de pruebas

6.1.3.Fases de proceso de prueba

6.1.3.1.Pruebas unitarias.



The screenshot shows a code editor with a file explorer on the left and a terminal at the bottom. The file explorer shows a project named 'VENTAS' with various files and folders. The code editor displays a PHP file named 'ProveedorControllerTest.php' with the following code:

```
11 class ProveedorControllerTest extends TestCase
31     public function no_se_puede_registrar_un_proveedor_sin_datos()
32     {
33         $response = $this->post('/proveedores', []);
34         $response->assertSessionHasErrors(['nombre', 'telefono', 'email']);
35     }
36
37
38     #[Test]
39     public function no_se_puede_registrar_un_proveedor_con_email_duplicado()
40     {
41         Proveedor::factory()->create(['email' => 'test@correo.com']->setTable('proveedores'));
42
43         $response = $this->post('/proveedores', [
44             'nombre' => 'Otro Proveedor',
45             'telefono' => '987654321',
46             'email' => 'test@correo.com',
47         ]);
48
49         $response->assertSessionHasErrors(['email']);
50     }
51 }
```

The terminal output shows the results of the tests:

```
Tests\Unit\ExampleTest
✓ that true is true

Tests\Feature\ExampleTest
✓ the application returns a successful response 0.23s

Tests\Feature\ProveedorControllerTest
✓ un usuario puede registrar un proveedor 0.98s
✓ no se puede registrar un proveedor sin datos 0.03s
✓ no se puede registrar un proveedor con email duplicado 0.07s

Tests: 5 passed (10 assertions)
Duration: 1.51s
```

6.1.3.2.Pruebas de integración.

```
1  <?php
2
3  namespace App\Models;
4
5  // use Illuminate\Contracts\Auth\MustVerifyEmail;
6  use Illuminate\Database\Eloquent\Factories\HasFactory;
7  use Illuminate\Foundation\Auth\User as Authenticatable;
8  use Illuminate\Notifications\Notifiable;
9
10 class User extends Authenticatable
11 {
12     /** @use HasFactory<\Database\Factories\UserFactory> */
13     use HasFactory, Notifiable;
14
15     /**
16      * The attributes that are mass assignable.
17      *
18      * @var list<string>
19      */
20     protected $fillable = [
21         'name',
22         'email',
23         'password',
24     ];
25
26     /**
27      * The attributes that should be hidden for serialization.
28      *
29      * @var list<string>
30      */
31     protected $hidden = [
32         'password',
33         'remember_token',
34     ];
35 }
```

PS C:\xampp\htdocs\Laravel11\Ventas> php artisan migrate

INFO Preparing database.

Creating migration table

INFO Running migrations.

0001_01_01_000000_create_users_table
0001_01_01_000001_create_cache_table
0001_01_01_000002_create_jobs_table

```
1  <?php
2
3  use Illuminate\Database\Migrations\Migration;
4  use Illuminate\Database\Schema\Blueprint;
5  use Illuminate\Support\Facades\Schema;
6
7  return new class extends Migration
8  {
9      /**
10       * Run the migrations.
11       */
12     public function up()
13     {
14         Schema::create('proveedors', function (Blueprint $table) {
15             $table->id();
16             $table->string('nombre');
17             $table->string('telefono');
18             $table->string('email')->unique();
19             $table->timestamps();
20         });
21     }
22
23     /**
24      * Reverse the migrations.
25      */
26     public function down(): void
27     {
28         Schema::dropIfExists('proveedors');
29     }
30 };
31
```

```

1  <?php
2
3  use Illuminate\Database\Migrations\Migration;
4  use Illuminate\Database\Schema\Blueprint;
5  use Illuminate\Support\Facades\Schema;
6
7  return new class extends Migration
8  {
9      /**
10       * Run the migrations.
11       */
12      public function up()
13      {
14          Schema::create('productos', function (Blueprint $table) {
15              $table->id();
16              $table->string('nombre');
17              $table->decimal('precio', 10, 2);
18              $table->integer('stock');
19              $table->unsignedBigInteger('proveedor_id');
20              $table->foreign('proveedor_id')->references('id')->on('proveedores')->onDelete('cascade');
21              $table->timestamps();
22          });
23      }
24
25      /**
26       * Reverse the migrations.
27       */
28      public function down(): void
29      {
30          Schema::dropIfExists('productos');
31      }
32  };
33

```

```

INFO Running migrations.

2025_02_24_154437_create_proveedores_table ..... 408.40ms DONE
2025_02_24_154505_create_productos_table ..... 12.91ms DONE

INFO Running migrations.

2025_02_24_154437_create_proveedores_table ..... 408.40ms DONE
2025_02_24_154505_create_productos_table ..... 12.91ms DONE

2025_02_24_154437_create_proveedores_table ..... 408.40ms DONE
2025_02_24_154505_create_productos_table ..... 12.91ms DONE

```

6.1.3.3.Pruebas del sistema.

```

1 APP_NAME=Laravel
2 APP_ENV=local
3 APP_KEY=base64:Y9eCYx21UEXE7iodpEG0Gpzz2m1ihC6vsmeqXkvjRpc=
4 APP_DEBUG=true
5 APP_TIMEZONE=UTC
6 APP_URL=http://localhost
7
8 APP_LOCALE=en
9 APP_FALLBACK_LOCALE=en
10 APP_FAKER_LOCALE=en_US
11
12 APP_MAINTENANCE_DRIVER=file
13 # APP_MAINTENANCE_STORE=database
14
15 PHP_CLI_SERVER_WORKERS=4
16
17 BCRYPT_ROUNDS=12
18
19 LOG_CHANNEL=stack
20 LOG_STACK=single
21 LOG_DEPRECATIONS_CHANNEL=null
22 LOG_LEVEL=debug
23
24 DB_CONNECTION=mysql
25 DB_HOST=127.0.0.1
26 DB_PORT=3306
27 DB_DATABASE=login
28 DB_USERNAME=root
29 DB_PASSWORD=
30
31 SESSION_DRIVER=database
32 SESSION_LIFETIME=120
33 SESSION_ENCRYPT=false
34 SESSION_PATH=/
35 SESSION_DOMAIN=null
36
37 BROADCAST_CONNECTION=log
38 FILESYSTEM_DISK=local
39 QUEUE_CONNECTION=database
```

```

40
41  CACHE_STORE=database
42  CACHE_PREFIX=
43
44  MEMCACHED_HOST=127.0.0.1
45
46  REDIS_CLIENT=phpredis
47  REDIS_HOST=127.0.0.1
48  REDIS_PASSWORD=null
49  REDIS_PORT=6379
50
51  MAIL_MAILER=log
52  MAIL_SCHEME=null
53  MAIL_HOST=127.0.0.1
54  MAIL_PORT=2525
55  MAIL_USERNAME=null
56  MAIL_PASSWORD=null
57  MAIL_FROM_ADDRESS="hello@example.com"
58  MAIL_FROM_NAME="${APP_NAME}"
59
60  AWS_ACCESS_KEY_ID=
61  AWS_SECRET_ACCESS_KEY=
62  AWS_DEFAULT_REGION=us-east-1
63  AWS_BUCKET=
64  AWS_USE_PATH_STYLE_ENDPOINT=false
65
66  VITE_APP_NAME="${APP_NAME}"
67

```

6.1.3.4.Pruebas de aceptación.

The screenshot shows an IDE with a file explorer on the left, a code editor in the center, and a terminal at the bottom. The file explorer shows a project structure with folders like 'resources', 'views', 'modules', 'proveedores', 'routes', 'storage', 'tests', and 'vendor'. The code editor displays a PHP file named 'ProveedorControllerTest.php' with the following content:

```

tests > Feature > ProveedorControllerTest.php > ProveedorControllerTest > un_usuario_puede_registrar_un_proveedor
11 class ProveedorControllerTest extends TestCase
12 {
13     use RefreshDatabase; // Limpia la BD después de cada prueba
14     use WithoutMiddleware;
15
16     #[Test]
17     public function un_usuario_puede_registrar_un_proveedor()
18     {
19         $response = $this->post('/proveedores', [
20             'nombre' => 'Proveedor Test',
21             'telefono' => '123456789',
22             'email' => 'test@correo.com',
23         ]);
24
25         $response->assertStatus(302); // Verifica la redirección exitosa
26
27         $this->assertDatabaseHas('proveedores', ['email' => 'test@correo.com']);
28     }
29
30     #[Test]
31     public function no_se_puede_registrar_un_proveedor_sin_datos()
32     {
33         $response = $this->post('/proveedores', []);
34     }
35 }

```

The terminal window shows the following output:

```

PASS Tests\Feature\ProveedorControllerTest
✓ un usuario puede registrar un proveedor 0.98s
✓ no se puede registrar un proveedor sin datos 0.03s
✓ no se puede registrar un proveedor con email duplicado 0.07s

Tests: 5 passed (10 assertions)
Duration: 1.51s

```

Below the terminal output, there are two PowerShell commands and their outputs:

```

PS C:\xampp\htdocs\Laravel11\Ventas>
History restored

PS C:\xampp\htdocs\Laravel11\Ventas>
History restored

```

A notification banner at the bottom right of the terminal area states: "No access to GitHub Copilot found. You are currently logged in as Cristian12. Source: GitHub Copilot".

6.1.3.5.Pruebas de regresión.

Tabla	Acción	Filas	Tipo	Cotejamiento	Tamaño	Residuo a depurar
<input type="checkbox"/> cache	★ Examinar Estructura Buscar Insertar Vaciar Eliminar	0	InnoDB	utf8mb4_unicode_ci	16.0 KB	-
<input type="checkbox"/> cache_locks	★ Examinar Estructura Buscar Insertar Vaciar Eliminar	0	InnoDB	utf8mb4_unicode_ci	16.0 KB	-
<input type="checkbox"/> failed_jobs	★ Examinar Estructura Buscar Insertar Vaciar Eliminar	0	InnoDB	utf8mb4_unicode_ci	32.0 KB	-
<input type="checkbox"/> jobs	★ Examinar Estructura Buscar Insertar Vaciar Eliminar	0	InnoDB	utf8mb4_unicode_ci	32.0 KB	-
<input type="checkbox"/> job_batches	★ Examinar Estructura Buscar Insertar Vaciar Eliminar	0	InnoDB	utf8mb4_unicode_ci	16.0 KB	-
<input type="checkbox"/> migrations	★ Examinar Estructura Buscar Insertar Vaciar Eliminar	6	InnoDB	utf8mb4_unicode_ci	16.0 KB	-
<input type="checkbox"/> password_reset_tokens	★ Examinar Estructura Buscar Insertar Vaciar Eliminar	0	InnoDB	utf8mb4_unicode_ci	16.0 KB	-
<input type="checkbox"/> productos	★ Examinar Estructura Buscar Insertar Vaciar Eliminar	1	InnoDB	utf8mb4_unicode_ci	32.0 KB	-
<input type="checkbox"/> producto_venta	★ Examinar Estructura Buscar Insertar Vaciar Eliminar	2	InnoDB	utf8mb4_unicode_ci	48.0 KB	-
<input type="checkbox"/> proveedores	★ Examinar Estructura Buscar Insertar Vaciar Eliminar	1	InnoDB	utf8mb4_unicode_ci	32.0 KB	-
<input type="checkbox"/> sessions	★ Examinar Estructura Buscar Insertar Vaciar Eliminar	1	InnoDB	utf8mb4_unicode_ci	48.0 KB	-
<input type="checkbox"/> users	★ Examinar Estructura Buscar Insertar Vaciar Eliminar	6	InnoDB	utf8mb4_unicode_ci	32.0 KB	-
<input type="checkbox"/> ventas	★ Examinar Estructura Buscar Insertar Vaciar Eliminar	1	InnoDB	utf8mb4_unicode_ci	16.0 KB	-
13 tablas	Número de filas	18	InnoDB	utf8mb4_general_ci	352.0 KB	0 8

6.1.4. Técnicas de prueba del software

6.1.4.1.Pruebas de caja negra.

Login:

Login exitoso

DATOS DE ENTRADA	RESULTADO
LOGIN - EMAIL & PASSWORD	ACCESO AL SISTEMA
Tipo de flujo de datos	
Archivo Pantalla Informe(x) Formulario Interno	
La estructura de datos que viaja con el flujo	
EMAIL & PASSWORD DEL USUARIO	
COMENTARIOS:	
Al ingresar credenciales correctas, el sistema valida y permite el acceso.	

Contraseña incorrecta

DATOS DE ENTRADA	RESULTADO
LOGIN - EMAIL & PASSWORD INCORRECTO	ERROR: CONTRASEÑA INCORRECTA
Tipo de flujo de datos	
Archivo Pantalla Informe(x) Formulario Interno	
La estructura de datos que viaja con el flujo	
EMAIL & PASSWORD DEL USUARIO	
COMENTARIOS:	
El sistema muestra un mensaje de error cuando la contraseña es incorrecta.	

Usuario no registrado

DATOS DE ENTRADA	RESULTADO
LOGIN - EMAIL NO REGISTRADO & PASSWORD	ERROR: USUARIO NO REGISTRADO
Tipo de flujo de datos	
Archivo Pantalla Informe(x) Formulario Interno	
La estructura de datos que viaja con el flujo	
EMAIL & PASSWORD DEL USUARIO	
COMENTARIOS:	
El sistema indica que el usuario no está registrado en la base de datos.	

Inyeccion SQL

DATOS DE ENTRADA	RESULTADO
LOGIN - ' OR '1'='1 -- & CUALQUIER CONTRASEÑA	ERROR: FORMATO INVÁLIDO
Tipo de flujo de datos	
Archivo Pantalla Informe(x) Formulario Interno	
La estructura de datos que viaja con el flujo	
EMAIL & PASSWORD DEL USUARIO	
COMENTARIOS:	
El sistema detecta y bloquea intentos de inyección SQL, asegurando la integridad del acceso.	

6.1.4.2.Pruebas de caja blanca.

Login:

Ingreso exitoso con credenciales correctas

DATOS DE ENTRADA	RESULTADO
CÓDIGO FUENTE	ACCESO PERMITIDO
Tipo de flujo de datos	
Archivo Pantalla Informe Formulario (X) Interno	
La estructura de datos que viaja con el flujo	
Usuario, Contraseña	
COMENTARIOS:	
El código sigue el flujo correcto cuando las credenciales son válidas, permitiendo el acceso al sistema.	

Fallo por contraseña incorrecta

DATOS DE ENTRADA	RESULTADO
CÓDIGO FUENTE	ERROR: CONTRASEÑA INCORRECTA
Tipo de flujo de datos	
Archivo Pantalla Informe Formulario (X) Interno	
La estructura de datos que viaja con el flujo	
Usuario, Contraseña Incorrecta	
COMENTARIOS:	
El código muestra el mensaje de error correcto, pero no implementa bloqueo tras múltiples intentos fallidos.	

Usuario no registrado

DATOS DE ENTRADA	RESULTADO
CÓDIGO FUENTE	ERROR: USUARIO NO REGISTRADO
Tipo de flujo de datos	
Archivo Pantalla Informe Formulario (X) Interno	
La estructura de datos que viaja con el flujo	
Usuario No Existente, Contraseña	
COMENTARIOS:	
El sistema identifica correctamente usuarios no registrados, pero no ofrece opción para registro inmediato.	

Campo vacío en usuario o contraseña

DATOS DE ENTRADA	RESULTADO
CÓDIGO FUENTE	ERROR: CAMPOS VACÍOS
Tipo de flujo de datos	
Archivo Pantalla Informe Formulario (X) Interno	
La estructura de datos que viaja con el flujo	
Usuario vacío / Contraseña vacía	
COMENTARIOS:	
El código previene intentos de login con campos vacíos, pero no sugiere corrección específica.	

6.2. Documentación de programas

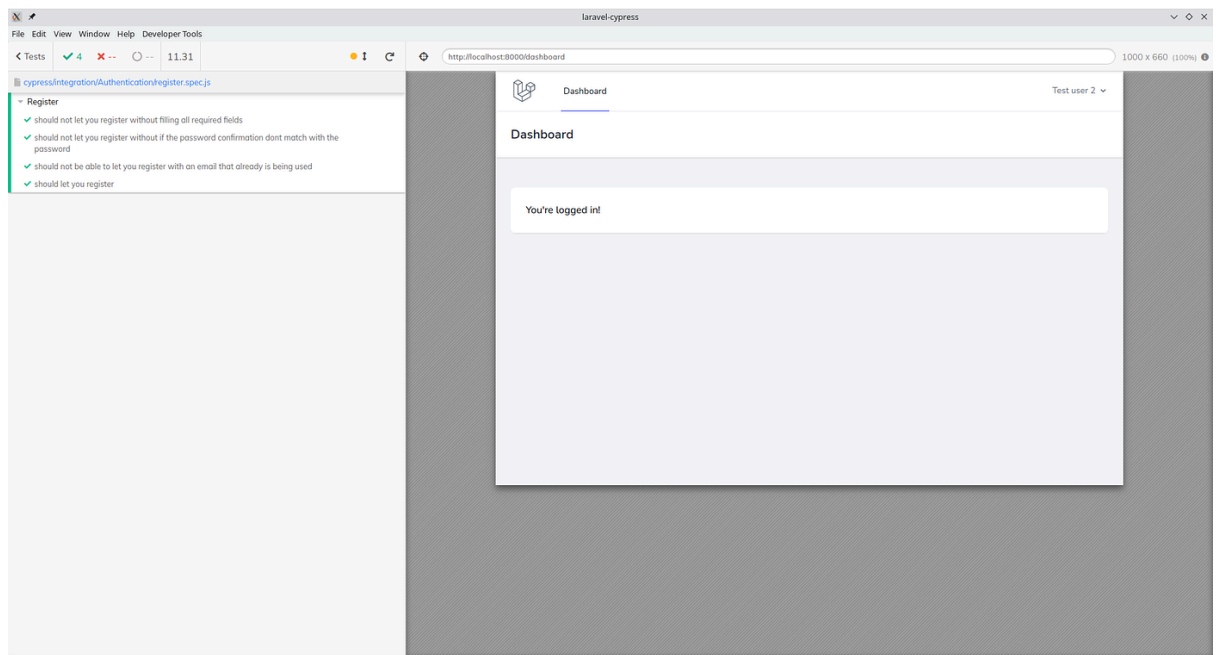
```
PS C:\xampp\htdocs\Laravel11\Ventas> php artisan test

PASS Tests\Unit\ExampleTest
✓ that true is true

PASS Tests\Feature\ExampleTest
✓ the application returns a successful response

Tests: 2 passed (2 assertions)
Duration: 0.31s
```

6.3. Herramientas Software para Realizar las Pruebas



CONCLUSIONES

RECOMENDACIONES

REFERENCIAS BIBLIOGRÁFICAS

Booch, G., Rumbaugh, J., & Jacobson, I. (2005). *The Unified Modeling Language User Guide*. Addison-Wesley.

Sommerville, I. (2015). *Software Engineering* (10th Edition). Pearson.

Pressman, R. S. (2020). *Ingeniería de Software: Un Enfoque Práctico*. McGraw-Hill Education.

IEEE Computer Society. (2014). *Guide to the Software Engineering Body of Knowledge (SWEBOK)*.

Dennis, A., Wixom, B. H., & Tegarden, D. (2020). *Systems Analysis and Design: An Object-Oriented Approach with UML*. Wiley.

ISTQB. (2018). *Foundations of Software Testing: ISTQB Certification*. Cengage Learning.

IEEE Standard 29119-3 (2013). *Software and Systems Engineering - Software Testing*.

Myers, G. J., Sandler, C., & Badgett, T. (2012). *The Art of Software Testing*. Wiley

Albrecht, A. J., & Gaffney, J. E. (1983). *Software Function, Source Lines of Code, and Development Effort Prediction: A Software Science Validation*.

Boehm, B. (1981). *Software Engineering Economics*. Prentice Hall.

Jones, C. (2017). *Estimating Software Costs: Bringing Realism to Estimating*.

ANEXOS

ANEXO 01. Evidencia de Desarrollo