

**Fecha:**

06/02/2023

# Documento Plan de Pruebas para el Sistema

Composición de Relaciones

Versión 2.0

**Autor:**

## Tabla de contenido

SECCIÓN I: CONTROL	149
<b>1.1.- Revisiones</b>	<b>149</b>
<b>1.2.- Versiones</b>	<b>149</b>
SECCIÓN II: INFORMACIÓN ESENCIAL	149
<b>2.1.- Introducción</b>	<b>149</b>
<b>2.2.- Objetivos</b>	<b>150</b>
<b>2.2.1.- Objetivo general</b>	<b>150</b>
<b>2.2.2.- Objetivos específicos</b>	<b>150</b>
<b>2.3.- Alcance</b>	<b>151</b>
<b>2.4.- Fuera del alcance</b>	<b>153</b>
<b>2.5.- Modelo de pruebas</b>	<b>153</b>
SECCIÓN III: ADMINISTRACIÓN	154
<b>3.1.- Documentos base</b>	<b>154</b>
<b>3.2.- Escritos a entregar</b>	<b>154</b>
<b>3.3.- Participantes</b>	<b>155</b>
<b>3.4.- Cronograma de actividades</b>	<b>155</b>
SECCIÓN IV: DATOS TÉCNICOS	156
<b>4.1.- Elementos a probar</b>	<b>156</b>
<b>Resultados</b>	<b>156</b>
<b>4.2.- Control de riesgos</b>	<b>156</b>
<b>4.3.- Niveles en los defectos</b>	<b>156</b>
<b>4.4.- Ciclo de vida de los defectos</b>	<b>158</b>

<b>4.5.- Ejecución de Pruebas</b>	<b>158</b>
<b>4.6.- Suspensión de Pruebas</b>	<b>159</b>
<b>4.7.- Reanudación de pruebas</b>	<b>159</b>
<b>4.8- Validación y aceptación</b>	<b>160</b>
<b>4.9.-Revisión de documentos</b>	<b>160</b>
<b>4.10.- Comunicación</b>	<b>160</b>
<b>4.11.- Reporte de hechos</b>	<b>161</b>
<b>4.12.- Definiciones</b>	<b>161</b>
<b>SECCIÓN V: AUTÓGRAFOS</b>	<b>163</b>
<b>5.1.- Firmas</b>	<b>163</b>

## SECCIÓN I: CONTROL

### 1.1.- Revisiones

Nombre y Apellido	Cargo	Departamento u organización	Fecha
Maglioni Arana Caparachin	Docente Encargado del Proyecto	Universidad Continental	24/01/2023

### 1.2.- Versiones

Versión	Descripción	Autor	Fecha
1.0	Servicio de Software de ventas orientado a apoyar procesos de inventario, registro de usuarios y ventas diarias.	Capacoila Apolinario Jhon, Puchoc Laura Kenny, Torres Montes Alex	24/01/2023
2.0	Servicio de Software de ventas orientado a apoyar procesos de inventario, registro de usuarios y ventas diarias.	Capacoila Apolinario Jhon, Puchoc Laura Kenny, Torres Montes Alex	25/01/2023
3.0	Servicio de Software de ventas orientado a apoyar procesos de inventario, registro de usuarios y ventas diarias.	Capacoila Apolinario Jhon, Puchoc Laura Kenny, Torres Montes Alex	04/02/2023

## SECCIÓN II: INFORMACIÓN ESENCIAL

## **2.1.- Introducción**

El propósito de esta introducción es explicar los diversos beneficios que puede brindar el uso de un sistema de ventas (software) para un minimarket. Un sistema de ventas es una herramienta tecnológica diseñada para ayudar a los negocios a automatizar y optimizar sus operaciones de ventas. Al utilizar un sistema de ventas en un minimarket, se pueden obtener diversos beneficios, como:

- Mayor eficiencia en el registro de ventas y la gestión de inventario
- Mejora en la precisión de los informes financieros y de ventas
- Reducción de tiempo y esfuerzo en tareas administrativas repetitivas
- Aumento en la satisfacción del cliente a través de un proceso de checkout más rápido y fluido
- Mejor control sobre los precios y promociones de los productos

En resumen, el propósito de esta introducción es demostrar cómo un sistema de ventas puede mejorar significativamente la eficiencia y productividad de un minimarket, lo que se traduce en una mayor rentabilidad y éxito del negocio.

## **2.2.- Objetivos**

### **2.2.1.- Objetivo general**

El presente documento tiene como objetivo asegurar la calidad y la confiabilidad del sistema de ventas para el minimarket antes de su implementación y lanzamiento al mercado.

### **2.2.2.- Objetivos específicos**

- Identificar y corregir cualquier falla o error en el funcionamiento del software.
- Verificar la compatibilidad del sistema con los diferentes dispositivos y sistemas operativos.
- Evaluar la seguridad y privacidad de la información del usuario.
- Comprobar la eficiencia y velocidad del sistema en situaciones de alta demanda.
- Verificar la precisión y coherencia de los informes y registros financieros generados por el software.
- Evaluar la experiencia del usuario y la facilidad de uso del sistema.
- Probar si el software hace lo que no debe, es decir, si provoca efectos indeseados
- Establecer el alcance de las pruebas que se realizarán según su tipo y tiempo.
- Definir herramientas que se utilizarán en la ejecución de los diferentes tipos de pruebas.
- Mostrar hasta qué punto las funciones del software operan de acuerdo con las especificaciones y requisitos del cliente.

Estos objetivos específicos son solo algunos ejemplos y pueden variar dependiendo de las necesidades específicas del proyecto. Sin embargo, su cumplimiento garantiza una entrega de calidad y un funcionamiento confiable del sistema de ventas para el minimarket.

### **2.3.- Alcance**

El presente documento lista las distintas pruebas a ser ejecutadas durante el proceso de prueba del sistema que permitirán asegurar el cumplimiento de los objetivos y los requerimientos del software, la forma en la que se ejecutarán, los criterios de aceptación, así como la forma y medios para registrar los resultados de la ejecución de pruebas.

Dentro de los alcances de las pruebas al sistema se establecen los siguientes puntos

- Pruebas Funcionales
- Pruebas Integrales
- Pruebas de seguridad
- Pruebas de carga
- Pruebas de rendimiento

Así mismo cabe mencionar en base a lo anterior, dicho plan forma parte integral de la metodología basada en tres etapas:

- Diseño de pruebas.
- Construcción y ejecución de pruebas que se adapten a la metodología de desarrollo de software que la empresa u organización determine.
- Seguimiento y resultados.

Dentro de cada una de las tres principales etapas mencionadas se divide en subetapas que apoyaran al cumplimiento de los objetivos de la metodología propuesta. A continuación se describe la clasificación de cada una de ellas para posteriormente ser explicada detalladamente:

***Diseño de pruebas:***

- Gestión de plantillas para pruebas.
- Identificación de requerimientos y casos de usos del sistema.
- Generación de Plan de pruebas.

***Construcción y ejecución de pruebas:***

- Identificación, clasificación y eliminación de casos de prueba redundantes.
- Creación de casos y escenarios de prueba al sistema.
- Ejecución de casos y escenarios de prueba al sistema.

***Seguimiento y resultados:***

- Reporte y seguimiento de incidencias encontradas en la ejecución
- Cierre y resultados finales de incidencias encontradas
- Generación de reporte final y análisis.

Finalmente es importante señalar que cada una de dichas pruebas serán aplicadas a los módulos correspondientes o al sistema en general, esto teniendo un enfoque de pruebas de caja negra, donde se analizaron la entrada de los datos y las salida de la información para comparan con los objetivos preestablecidos.

## **2.4.- Fuera del alcance**

Los componentes y objetivos que no están descritos en la especificación de requerimientos del sistema ni en el Plan de Pruebas actual no están incluidos en el alcance.

## **2.5.- Modelo de pruebas**

La estrategia de pruebas se basa en un proceso específico descrito en la investigación llamada "Modelo para la generación y ejecución de pruebas como medio de verificación y validación de productos de software de calidad", que se aplica a empresas dedicadas al desarrollo de software. Este modelo define las directrices y procedimientos necesarios para asegurar el éxito de las pruebas y garantizar la calidad de los productos de software. Cuyas actividades principales de este proceso son la:

- Planificación de las pruebas.
- Seleccionar los componentes y productos a probar.
- Realizar los escenarios de prueba con su correspondiente información a implementar.
- Establecer el ambiente para las pruebas.
- Establecer la versión a probar.
- Ejecutar las pruebas de la solución tecnológica.
- Documentar los defectos, hallazgos y no conformidades detectados en la solución tecnológica.
- Analizar la resolución de defectos, hallazgos y no conformidades.

## SECCIÓN III: ADMINISTRACIÓN

### 3.1.- Documentos base

Nombre del documento	Versión
Servicios de software de ventas orientadas a apoyar procesos de inventario, registro de usuario y ventas diarias. (24/01/2023)	1.0
Servicios de software de ventas orientadas a apoyar procesos de inventario, registro de usuario y ventas diarias. (25/01/2023)	2.0
Servicios de software de ventas orientadas a apoyar procesos de inventario, registro de usuario y ventas diarias. (04/02/2023)	3.0

### 3.2.- Escritos a entregar

Nombre del documento	Fecha	Entregado
Presentación del plan de pruebas	23-01-2023	SI
Diseño del proyecto	25-01-2023	SI
Informe plan de pruebas funcionales respecto al software.	01-01-2023	SI
Testeo y control de calidad	06-02-2023	SI
Git commit cambios instantáneos del proyecto	08-02-2023	SI
Instalación y soporte técnico	13-02-2023	NO
Fin del proyecto	28-02-2023	NO

### **3.3.- Participantes**

Integrantes de pruebas:

Nombre / Rol	Área	Actividad
Puchoc Laura Kenny Willian	Prgramador	Diagnostica e identifica problemas.
Capacoila Apolinario Jhon Carlos	Programador desarrollo de testing	Realiza Testeos.
Torres Montes, Alex	Programador desarrollo de base de datos	Implementa soluciones

Integrantes de programadores:

Nombre / Rol	Nombre	Actividad
Gerente / Líder de proyecto	Puchoc Laura Kenny Willian	Desarrollar la aplicación.
Líder de pruebas de software	Capacoila Apolinario Jhon Carlos	Pruebas de software.
Lider de pruebas de Testing	Torres Montes, Alex	Revisiones y gestión de incidencias.

Los Recursos Materiales para la ejecución de las pruebas:

Cantidad	Recursos	Disponible
1	Laptop	Si
1	Servidor de Base de Datos	Si
1	Antivirus	Si
1	Apache NetBeans IDE 13	Si

### **3.4.- Cronograma de actividades**

A continuación se especifica el calendario de las actividades correspondiente a la creación y ejecución de las pruebas para el sistema.

## ACTIVIDADES:

**Prueba del servicio:** El objetivo de las pruebas de servicio es preparar a fondo el servicio verificando la integración de los módulos y comprobando el correcto funcionamiento de las interfaces.

**Pruebas Unitarias y de Integración:** Es necesario verificar que la persona que realiza la prueba es alguien que utilizará regularmente el sistema informático en el futuro.

**Pruebas de Funcionalidad:** El equipo de pruebas tendrá acceso al código fuente del sistema. Cuando se descubre un problema, el equipo trabajará para determinar el origen del problema y los componentes que deben limpiarse.

**Pruebas de Rendimiento:** El equipo de pruebas tendrá acceso al código fuente del servicio. Cuando se descubre un problema, el equipo trabajará para identificar el origen del problema y los componentes que deben limpiarse. El objetivo de las pruebas de rendimiento será determinar el estado del servicio cuando se somete a una gran cantidad de datos, lo que suele ser el caso después de un largo período de uso

**Prueba de seguridad y acceso:** Se verifica que los servicios de seguridad de aplicaciones brinden un nivel sólido de seguridad para la institución.

	CALENDARIO DE PRUEBAS																																																		
	ENERO															FEBRERO																																			
	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28
Pruebas del servicio																																																			
Pruebas unitarias y de integración																																																			
Prueba de funcionalidad																																																			
Prueba de rendimiento																																																			
Prueba de seguridad y acceso																																																			

## SECCIÓN IV: DATOS TÉCNICOS

### 4.1.- Elementos a probar

#### Resultados

Componentes a probar	Tipo de prueba a ejecutar	Satisfactorio
Verificar la integración y la eficiencia del software.	Prueba del servicio	Satisfactorio
Correcto funcionamiento de las interfaces	Prueba de unitarias y de integración	Satisfactorio
Actualización de software.	Prueba de unitarias y de integración	Satisfactorio
Identificar, solucionar y documentar fallos.	Prueba de Funcionalidad	Satisfactorio
Determinar el estado de servicio	Prueba del rendimiento	Satisfactorio
Verificar la seguridad de los servicios y aplicaciones.	Prueba de seguridad y acceso	Satisfactorio

#### 4.2.- Control de riesgos

Id	Fecha de descubrimiento	Severidad	Solución	Fecha de solución
01	23/02/2023	Alta	Corregir la inconsistencia de datos	25/02/2023
02	31/01/2023	Media	Hacer una revisión adecuada de la funcionalidad del servicio.	02/02/2023
03	04/02/2023	Media	Monitorear las normas de la empresa que proporcionó y basarse en ello.	05/02/2023
04	06/02/2023	Baja	Solucionar errores de diseño	07/02/2023

#### 4.3.- Niveles en los defectos

Se explicará a continuación la gravedad de los defectos para categorizarlos, con el objetivo de determinar su nivel y la acción correctiva correspondiente.

Los defectos incluyen lo siguiente:

*Crítico:*

- Existen enlaces rotos, inválidos o no implementados.
- El sistema no responde y no hay alternativas para ejecutar las pruebas.
- El sistema se cierra inesperadamente.

*Alto:*

- No cumple con los requerimientos.
- Datos inconsistentes.
- Secciones de un requerimiento faltantes.

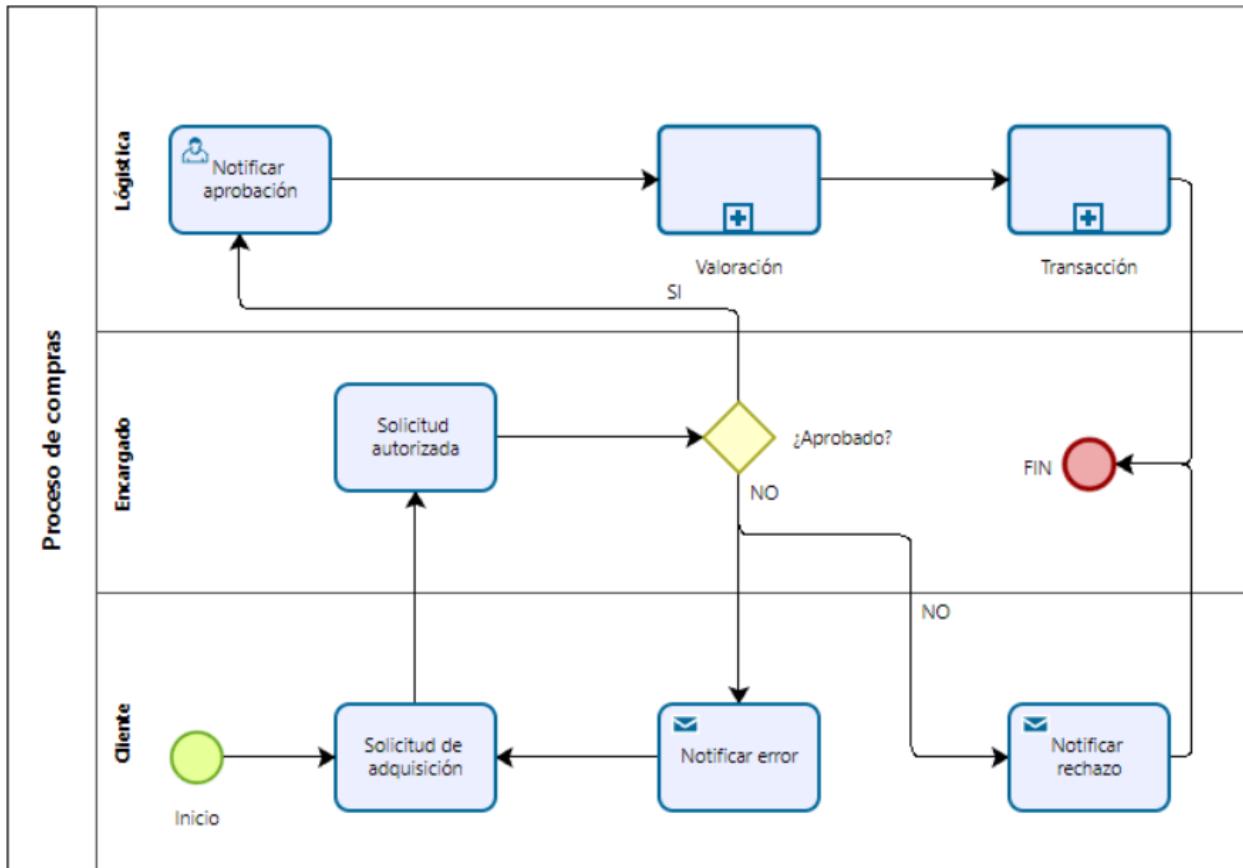
*Media:*

- Campos o pantallas faltantes.
- Errores que permitan continuar con el proceso.
- El sistema no envía mensajes al usuario.

*Baja:*

- Errores ortográficos o de diseño.
- El sistema no indica que son campos obligatorios con (\*) o notación resaltada.

#### 4.4.- Ciclo de vida de los defectos



#### 4.5.- Ejecución de Pruebas

Seguidamente se enumeran los requisitos mínimos que deben cumplirse antes de comenzar las pruebas.

Módulos	Prueba	Descripción
Registro de usuarios	Unitarias	El usuario podrá registrar sus datos para luego iniciar sesión.
Login	Unitarias	El usuario podrá iniciar sesión en la interfaz con sus credenciales registradas.
Ingreso con el rol del usuario	Unitarias	Que el usuario pueda visualizar la ventana que le corresponde según el rol que se le ha definido.

Administrar el ingreso de los productos	Unitarias	Todos los productos deben ser registrados de manera que no se pierda ningún dato de este.
Ingreso de nuevo cliente	Funcional	El usuario es responsable de ingresar los datos del nuevo cliente en el sistema y de asegurarse de que estos datos sean válidos y cumplan con los requisitos necesarios.
Administrar Retiro de Producto	Funcional	Que el stock correspondiente al producto sea actualizado, una vez se haya hecho el retiro de este.
Compra de Productos	Funcional	Se realizará la compra, con respecto si hay suficiente stock, por stock negativo y cuando se tenga de stock 0.
Generar orden de Compra	Funcional	Que a la compra de uno o varios de los productos, el servicio nos genere la orden de esta.

### Pruebas Caja negra

Caso prueba 01	Descripción

## -Registro nuevo cliente

The screenshot shows the WISA POS software interface. At the top, there is a logo for 'Universidad Continental' and the WISA logo with the tagline 'A cualquier hora del día...'. To the right of the logo is a graphic for 'TIENDAS WISA FRANQUICIAS' featuring a shopping bag icon.

The main window is titled 'REGISTRO DE CLIENTE' (Client Registration). It contains fields for 'Dni/Ruc', 'Nombre', 'Teléfono', and 'Dirección'. The 'Nombre' field is populated with 'Prueba01', 'Teléfono' with '999888777', and 'Dirección' with 'Av.San Luis'. Below these fields are four small icons: a blue square with a white circle, a grey square with a white circle, a grey square with a blue circle, and a blue square with a white circle.

To the left of the registration window is a vertical sidebar with the following menu items:

- Nueva Venta
- Cuentas
- Proveedor
- Productos
- Ventas
- Config
- Usuarios

At the bottom right of the registration window, a message box appears with the text 'Cliente Registrado' and an 'OK' button.

Below the registration window, there is a table showing the current client list:

ID	DNI/RUC	NOMBRE	TELÉFONO	DIRECCIÓN
1	74361715	Ponnygon	999888777	av.san carlos
2	12345678	PruebaClien	999666555	Calle Real

### Resultados

El caso de prueba se ha ejecutado perfectamente, registro exitoso

### Observaciones

Ninguno

### Caso prueba 02

### Descripción

### -Registro nuevo proveedor

**NUEVO PROVEEDOR**

Ruc:	741852963
Nombre:	Backus
Teléfono:	985274112
Dirección:	Calle Real #1992

ID	RUC	NOMBRE	TELÉFONO	DIRECCIÓN
1	998787	Open Services	798978879	Lima - Perú
2	515228	Plaza Norte	789456123	av.San Martin

Message X

**Proveedor Registrado**

ID	RUC	NOMBRE	TELÉFONO	DIRECCIÓN
1	998787	Open Services	798978879	Lima - Perú
2	515228	Plaza Norte	789456123	av.San Martin
3	741852963	Backus	985274112	Calle Real #1992

### Resultados

El caso de prueba se ha ejecutado perfectamente, registro exitoso

### Observaciones

Ninguno.

### Caso Prueba 03

#### Descripción

### -Registro nueva venta

Código 741	Descripción Leche Gloria 1L	Cant 30	Precio 9.0	<input type="button" value="GIF"/>	Seleccionar: 8 feb. 2023 <input type="button" value="Caja"/>
Stock Disponible 37					<input type="button" value="Imprimir"/>
ID 1	DESCRIPCIÓN Leche Gloria 1L	CANTIDAD 30	PRECIO U. 9.0	PRECIO TOTAL 270.0	

Dni/Ruc  
74361715      Nombre:  
Ponnygon      Imprimir      Total a Pagar: 270,00

### Resultados

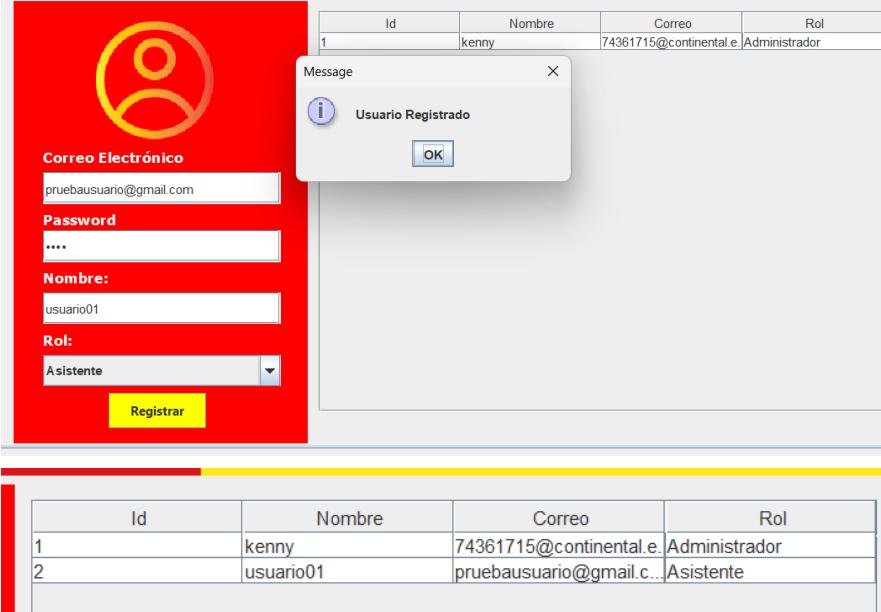
El caso de prueba se ha ejecutado perfectamente, registro exitoso

### Observaciones

Ninguno

Caso Prueba 04	Descripción																						
-Prueba de impresión de la boleta	<p>DATOS DEL CLIENTE</p> <table border="1"> <thead> <tr> <th>Nombre</th> <th>Teléfono</th> <th>Dirección</th> </tr> </thead> <tbody> <tr> <td>Ponnygon</td> <td>999888777</td> <td>av.san carlos</td> </tr> </tbody> </table> <table border="1"> <thead> <tr> <th>Cant.</th> <th>Descripción.</th> <th>P. unit.</th> <th>P. Total</th> </tr> </thead> <tbody> <tr> <td>20</td> <td>Galletas</td> <td>3.50</td> <td>70.0</td> </tr> <tr> <td>6</td> <td>Cerveza</td> <td>3.00</td> <td>18.0</td> </tr> <tr> <td>6</td> <td>Galleta Casino</td> <td>1.00</td> <td>6.0</td> </tr> </tbody> </table> <p>Total S/: 94.0</p> <p>Cancelacion</p> <p>Firma WISA</p>	Nombre	Teléfono	Dirección	Ponnygon	999888777	av.san carlos	Cant.	Descripción.	P. unit.	P. Total	20	Galletas	3.50	70.0	6	Cerveza	3.00	18.0	6	Galleta Casino	1.00	6.0
Nombre	Teléfono	Dirección																					
Ponnygon	999888777	av.san carlos																					
Cant.	Descripción.	P. unit.	P. Total																				
20	Galletas	3.50	70.0																				
6	Cerveza	3.00	18.0																				
6	Galleta Casino	1.00	6.0																				
Resultados	El caso de prueba se ha ejecutado perfectamente, registro exitoso																						
Observaciones	Ninguno																						

Caso Prueba 05	Descripción								
-Historial de ventas	<p>HISTORIAL DE VENTAS</p> <table border="1"> <thead> <tr> <th>ID</th> <th>CLIENTE</th> <th>VENDEDOR</th> <th>TOTAL</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Ponnygon</td> <td>kenny</td> <td>207.0</td> </tr> </tbody> </table>	ID	CLIENTE	VENDEDOR	TOTAL	1	Ponnygon	kenny	207.0
ID	CLIENTE	VENDEDOR	TOTAL						
1	Ponnygon	kenny	207.0						
Resultados	El caso de prueba se ha guardado cada compra que se realiza								
Observaciones	Ninguno								

Caso Prueba 06	Descripción																				
-Creen nuevo usuario, administrador o asistente	 <p>The screenshot shows a Java Swing application window. On the left, there is a registration form with fields for Correo Electrónico (pruebausuario@gmail.com), Password (****), Nombre (usuario01), and Rol (Asistente). A yellow 'Registrar' button is at the bottom. On the right, a message dialog box titled 'Message' with an information icon says 'Usuario Registrado'. Below the dialog is a table with two rows of data:</p> <table border="1"> <thead> <tr> <th>Id</th> <th>Nombre</th> <th>Correo</th> <th>Rol</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>kenny</td> <td>74361715@continental.e...</td> <td>Administrador</td> </tr> </tbody> </table> <p>Below this table is another table showing the full database state:</p> <table border="1"> <thead> <tr> <th>Id</th> <th>Nombre</th> <th>Correo</th> <th>Rol</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>kenny</td> <td>74361715@continental.e...</td> <td>Administrador</td> </tr> <tr> <td>2</td> <td>usuario01</td> <td>pruebausuario@gmail.c...</td> <td>Asistente</td> </tr> </tbody> </table>	Id	Nombre	Correo	Rol	1	kenny	74361715@continental.e...	Administrador	Id	Nombre	Correo	Rol	1	kenny	74361715@continental.e...	Administrador	2	usuario01	pruebausuario@gmail.c...	Asistente
Id	Nombre	Correo	Rol																		
1	kenny	74361715@continental.e...	Administrador																		
Id	Nombre	Correo	Rol																		
1	kenny	74361715@continental.e...	Administrador																		
2	usuario01	pruebausuario@gmail.c...	Asistente																		
Resultados	El caso de prueba se cumplido con éxito																				
Observaciones	Ninguno																				

## CASOS DE PRUEBAS UNITARIAS:

Caso Prueba 01: (Registro de cliente)	Código
	<pre> private void btnGuardarClienteActionPerformed(java.awt.event.ActionEvent evt) {     // TODO add your handling code here:     if (!"".equals(txtDniCliente.getText())    !"".equals(txtNombreCliente.getText())    !"".equals(txtTelefonoCliente.getText())    !"".equals(txtDireccionCliente.getText())) {         cl.setDni(txtDniCliente.getText());         cl.setNombre(txtNombreCliente.getText());         cl.setTelefono(txtTelefonoCliente.getText());         cl.setDireccion(txtDireccionCliente.getText());         client.RegistrarCliente(cl);         JOptionPane.showMessageDialog(null, "Cliente Registrado");         LimpiarTable();         LimpiarCliente();         ListarCliente();         btnEditarCliente.setEnabled(false);         btnEliminarCliente.setEnabled(false);         btnGuardarCliente.setEnabled(true);     } else {         JOptionPane.showMessageDialog(null, "Los campos estan vacios");     } } </pre>
Descripción: BtnGuardar	Caso de prueba: todos los campos están vacíos Entrada: txtDniCliente = "", txtNombreCliente = "", txtTelefonoCliente = "", txtDireccionCliente = ""

	<p>Resultado esperado: Se mostrará un mensaje "Los campos están vacíos" y no se registrará ningún cliente.</p> <p>Caso de prueba: todos los campos están llenos  Entrada: txtDniCliente = "123456789", txtNombreCliente = "Juan",  txtTelefonoCliente = "123456789", txtDireccionCliente = "Calle 123"  Resultado esperado: Se registrará un cliente con los datos proporcionados y se mostrará un mensaje "Cliente Registrado".</p> <p>Caso de prueba: solo el campo de nombre está vacío  Entrada: txtDniCliente = "123456789", txtNombreCliente = "", txtTelefonoCliente = "123456789", txtDireccionCliente = "Calle 123"  Resultado esperado: Se mostrará un mensaje "Los campos están vacíos" y no se registrará ningún cliente.</p>
<i>Observaciones</i>	Ninguno

<b>Caso Prueba 02: (Registro de cliente)</b>	<b>Código</b>
	<pre>private void btnEditarClienteActionPerformed(java.awt.event.ActionEvent evt) {     // TODO add your handling code here:     if ("").equals(txtIdCliente.getText())) {         JOptionPane.showMessageDialog(null, "seleccione una fila");     } else {          if (!"".equals(txtDniCliente.getText())    !"".equals(txtNombreCliente.getText())    !""             cl.setDni(txtDniCliente.getText());             cl.setNombre(txtNombreCliente.getText());             cl.setTelefono(txtTelefonoCliente.getText());             cl.setDireccion(txtDireccionCliente.getText());             cl.setId(Integer.parseInt(txtIdCliente.getText()));             client.ModificarCliente(cl);             JOptionPane.showMessageDialog(null, "Cliente Modificado");             LimpiarTable();             LimpiarCliente();             ListarCliente();         } else {             JOptionPane.showMessageDialog(null, "Los campos estan vacios");         }     } }</pre>
<i>Descripción: BtnEditar</i>	<p>Caso de prueba:</p> <p>Campos llenos Entrada: txtDniCliente.getText() = "12345678", txtNombreCliente.getText() = "John Doe", txtTelefonoCliente.getText() = "555-555-5555", txtIdCliente.getText() = "1"  Salida esperada: JOptionPane con mensaje "Cliente Modificado" Condición: Si txtDniCliente, txtNombreCliente y txtTelefonoCliente no están vacíos, se llama al método ModificarCliente y se muestra un mensaje de éxito.</p> <p>Caso de prueba:</p> <p>Campos vacíos Entrada: txtDniCliente.getText() = "", txtNombreCliente.getText() = "", txtTelefonoCliente.getText() = "" Salida esperada: JOptionPane con mensaje "Los campos estan vacios" Condición: Si txtDniCliente, txtNombreCliente o txtTelefonoCliente están</p>

	<p>vacios, se muestra un mensaje de error.</p> <p>Caso de prueba:</p> <p>Selección de fila Entrada: txtIdCliente.getText() = "" Salida esperada: JOptionPane con mensaje "seleccione una fila" Condición: Si txtIdCliente está vacío, se muestra un mensaje de error indicando que no se ha seleccionado una fila.</p>
<i>Observaciones</i>	Ninguno

<b>Caso Prueba 03: (Registro de cliente)</b>	<i>Código</i>
	<pre>private void btnEliminarClienteActionPerformed(java.awt.event.ActionEvent evt) {     // TODO add your handling code here:     if (!"".equals(txtIdCliente.getText())) {         int pregunta = JOptionPane.showConfirmDialog(null, "Esta seguro de eliminar");         if (pregunta == 0) {             int id = Integer.parseInt(txtIdCliente.getText());             client.EliminarCliente(id);             LimpiarTable();             LimpiarCliente();             ListarCliente();         }     } }</pre>
<i>Descripción: BtnEliminar</i>	<p>Eliminar cliente exitosamente:</p> <p>Entrada: txtIdCliente.getText() = "1" Salida esperada: Se muestra un cuadro de diálogo para confirmar la eliminación, si el usuario elige sí, el cliente con ID 1 se elimina de la base de datos y se muestra un mensaje "Cliente Eliminado". Condición: Si txtIdCliente no está vacío, se elimina el cliente de la base de datos.</p> <p>No se puede eliminar cliente sin seleccionar una fila:</p> <p>Entrada: txtIdCliente.getText() = "" Salida esperada: Se muestra un mensaje "Seleccione una fila". Condición: Si txtIdCliente está vacío, no se puede eliminar el cliente y se muestra un mensaje de error.</p>

<i>Observaciones</i>	Ninguno
----------------------	---------

<b>Caso Prueba 04:</b> <i>(Nuevo Proveedor)</i>	<b>Código</b>
	<pre> private void btnguardarProveedorActionPerformed(java.awt.event.ActionEvent evt) {     // TODO add your handling code here:     if (!"".equals(txtRucProveedor.getText())    !"".equals(txtNombreproveedor.getText())    !""         pr.setRuc(txtRucProveedor.getText());         pr.setNombre(txtNombreproveedor.getText());         pr.setTelefono(txtTelefonoProveedor.getText());         pr.setDireccion(txtDireccionProveedor.getText());         PrDao.RegistrarProveedor(pr);         JOptionPane.showMessageDialog(null, "Proveedor Registrado");         LimpiarTable();         ListarProveedor();         LimpiarProveedor();         btnEditarProveedor.setEnabled(false);         btnEliminarProveedor.setEnabled(false);         btnguardarProveedor.setEnabled(true);     } else {         JOptionPane.showMessageDialog(null, "Los campos esta vacios");     } } </pre>
<i>Descripción: BtnGuardar</i>	<p>Escenario 1: Todos los campos están completos:</p> <p>Entrada: Ruc: 123456789, Nombre: Proveedor A, Teléfono: 1234567, Dirección: Calle 123  Condición: Los valores de los campos "txtRucProveedor", "txtNombreproveedor", "txtTelefonoProveedor", "txtDireccionProveedor" no están vacíos.  Salida esperada: Proveedor Registrado.</p> <p>Escenario 2: Al menos un campo está vacío:  Entrada: Ruc: 123456789, Nombre: Proveedor A, Teléfono: , Dirección: Calle 123  Condición: Al menos uno de los campos "txtRucProveedor", "txtNombreproveedor", "txtTelefonoProveedor", "txtDireccionProveedor" está vacío.  Salida esperada: Los campos están vacíos.</p> <p>Escenario 3: Todos los campos están vacíos:  Entrada: Ruc: , Nombre: , Teléfono: , Dirección:  Condición: Todos los campos "txtRucProveedor", "txtNombreproveedor", "txtTelefonoProveedor", "txtDireccionProveedor" están vacíos.  Salida esperada: Los campos están vacíos.</p>
<i>Observaciones</i>	Ninguno

<b>Caso Prueba 05:</b> <i>(Nuevo Proveedor)</i>	<b>Código</b>
--	---------------

	<pre> private void btnEditarProveedorActionPerformed(java.awt.event.ActionEvent evt) {     // TODO add your handling code here:     if ("").equals(txtIdProveedor.getText()) {         JOptionPane.showMessageDialog(null, "Seleccione una fila");     } else {         if (!"".equals(txtRucProveedor.getText())    !"".equals(txtNombreProveedor.getText())    !""             pr.setRuc(txtRucProveedor.getText());         pr.setNombre(txtNombreProveedor.getText());         pr.setTelefono(txtTelefonoProveedor.getText());         pr.setDireccion(txtDireccionProveedor.getText());         pr.setId(Integer.parseInt(txtIdProveedor.getText()));         PrDao.ModificarProveedor(pr);         JOptionPane.showMessageDialog(null, "Proveedor Modificado");         LimpiarTable();         ListarProveedor();         LimpiarProveedor();         btnEditarProveedor.setEnabled(false);         btnEliminarProveedor.setEnabled(false);         btnguardarProveedor.setEnabled(true);     } } </pre>
<i>Descripción: BtnEditar</i>	<p><b>Primer condicional:</b>  if ("").equals(txtIdProveedor.getText())  Este condicional verifica si el campo txtIdProveedor está vacío. Si está vacío, se muestra un mensaje de error "Seleccione una fila".</p> <p><b>Segundo condicional:</b>  if (!"".equals(txtRucProveedor.getText())    !"".equals(txtNombreProveedor.getText())     !"".equals(txtTelefonoProveedor.getText())    !"".equals(txtDireccionProveedor.getText()))</p> <p>Este condicional verifica si al menos uno de los campos txtRucProveedor, txtNombreProveedor, txtTelefonoProveedor, o txtDireccionProveedor no están vacíos. Si al menos uno de ellos no está vacío, se ejecutan las siguientes acciones:</p> <ul style="list-style-type: none"> <li>• Se establecen los valores de los campos en el objeto pr</li> <li>• Se llama al método PrDao.ModificarProveedor(pr)</li> <li>• Se muestra un mensaje "Proveedor Modificado"</li> <li>• Se llama a los métodos LimpiarTable(), ListarProveedor(), y LimpiarProveedor()</li> <li>• Se desactivan los botones btnEditarProveedor y btnEliminarProveedor</li> <li>• Se activa el botón btnguardarProveedor</li> </ul> <p>Prueba con todos los campos vacíos: se espera que se muestre un mensaje de error "Seleccione una fila" y no se realice ninguna acción.</p> <p>Prueba con al menos un campo lleno: se espera que se ejecuten las acciones descritas en el segundo condicional y se muestre un mensaje "Proveedor Modificado".</p> <p>Prueba con todos los campos llenos: se espera que se ejecuten las acciones descritas en el segundo condicional y se muestre un mensaje "Proveedor Modificado"</p>
<i>Observaciones</i>	Ninguno

<b>Caso Prueba 06: (Nuevo Proveedor)</b>	<b>Código</b>
--	---------------

	<pre> private void btnEliminarProveedorActionPerformed(java.awt.event.ActionEvent evt) {     // TODO add your handling code here:     if (!"".equals(txtIdProveedor.getText())) {         int pregunta = JOptionPane.showConfirmDialog(null, "Esta seguro de eliminar");         if (pregunta == 0) {             int id = Integer.parseInt(txtIdProveedor.getText());             PrDao.EliminarProveedor(id);             LimpiarTable();             ListarProveedor();             LimpiarProveedor();         }     } else {         JOptionPane.showMessageDialog(null, "Seleccione una fila");     } } </pre>
<i>Descripción: BtnEliminar</i>	<p>Verificación de que el campo "txtIdProveedor" no esté vacío:</p> <p>Si está vacío, se muestra un mensaje de error al usuario "Seleccione una fila".</p> <p>Pregunta al usuario si está seguro de eliminar el proveedor:</p> <p>Si la respuesta es sí (pregunta = 0), se elimina el proveedor llamando al método "EliminarProveedor" de PrDao con el ID del proveedor a eliminar.</p> <p>Después de la eliminación, se actualiza la tabla de proveedores y se limpian los campos de formulario de proveedor.</p> <p>Caso de prueba: Eliminación exitosa de un proveedor</p> <p>Entrada: Se selecciona un proveedor en la tabla y se presiona el botón "Eliminar Proveedor". Se responde "Sí" a la pregunta sobre la eliminación del proveedor.</p> <p>Salida esperada: Se elimina el proveedor y se actualiza la tabla de proveedores y se limpian los campos de formulario de proveedor.</p> <p>Caso de prueba: No se selecciona un proveedor para eliminar</p> <p>Entrada: Se presiona el botón "Eliminar Proveedor" sin seleccionar un proveedor en la tabla.</p> <p>Salida esperada: Se muestra un mensaje de error "Seleccione una fila".</p> <p>Caso de prueba: No se confirma la eliminación de un proveedor</p> <p>Entrada: Se selecciona un proveedor en la tabla y se presiona el botón "Eliminar Proveedor". Se responde "No" a la pregunta sobre la eliminación del proveedor.</p> <p>Salida esperada: No se elimina el proveedor y se mantiene la tabla de proveedores y los campos de formulario de proveedor intactos.</p>
<i>Observaciones</i>	Ninguno

<b>Caso Prueba 07: (Nuevo Producto)</b>	<b>Código</b>
---	---------------

	<pre> private void btnEliminarProveedorActionPerformed(java.awt.event.ActionEvent evt) {     // TODO add your handling code here:     if (!"".equals(txtIdProveedor.getText())) {         int pregunta = JOptionPane.showConfirmDialog(null, "Esta seguro de eliminar");         if (pregunta == 0) {             int id = Integer.parseInt(txtIdProveedor.getText());             PrDao.EliminarProveedor(id);             LimpiarTable();             ListarProveedor();             LimpiarProveedor();         }     } else {         JOptionPane.showMessageDialog(null, "Seleccione una fila");     } } </pre>
<i>Descripción: BtnGuardar</i>	<p>Caso de prueba: Todos los campos están vacíos. Entrada: txtCodigoPro = "", txtDesPro = "", cbxProveedorPro = "", txtCantPro = "", txtPrecioPro = "" Salida esperada: JOptionPane con mensaje "Los campos estan vacios".</p> <p>Caso de prueba: Todos los campos están llenos y válidos. Entrada: txtCodigoPro = "123", txtDesPro = "producto", cbxProveedorPro = "Proveedor 1", txtCantPro = "10", txtPrecioPro = "50.0" Salida esperada: JOptionPane con mensaje "Productos Registrado".</p> <p>Caso de prueba: Uno o más campos están vacíos. Entrada: txtCodigoPro = "", txtDesPro = "producto", cbxProveedorPro = "Proveedor 1", txtCantPro = "10", txtPrecioPro = "50.0" Salida esperada: JOptionPane con mensaje "Los campos estan vacios".</p> <p>Caso de prueba: Se intenta registrar un producto con un código que ya existe Esperado: Se muestra un mensaje "El código del producto ya existe".</p>
<i>Observaciones</i>	Ninguno

<b>Caso Prueba 08:</b> <i>(Nuevo Producto)</i>	<i>Código</i>
	<pre> private void btnEditarproActionPerformed(java.awt.event.ActionEvent evt) {     // TODO add your handling code here:     if ("".equals(txtIdproducto.getText())) {         JOptionPane.showMessageDialog(null, "Seleccione una fila");     } else {         if (!"".equals(txtCodigoPro.getText())    !"".equals(txtDesPro.getText())    !"".equals(txtCantPro.getText()))             pro.setCodigo(txtCodigoPro.getText());         pro.setNombre(txtDesPro.getText());         pro.setProveedor((Combo cbxProveedorPro.getSelectedItem());         pro.setProveedor(itemP.getId());         pro.setStock(Integer.parseInt(txtCantPro.getText()));         pro.setPrecio(Double.parseDouble(txtPrecioPro.getText()));         pro.setId(Integer.parseInt(txtIdproducto.getText()));         PrDao.ModificarProductos(pro);         JOptionPane.showMessageDialog(null, "Producto Modificado");         LimpiarTable();         ListarProductos();         LimpiarProductos();         cbxProveedorPro.removeAllItems();         llenarProveedor();         btnEditarpro.setEnabled(false);         btnEliminarPro.setEnabled(false);         btnGuardarpromo.setEnabled(true);     } } </pre>

<i>Descripción: BtnEditar</i>	<p>Todos los campos están llenos: Se espera que se edite el producto correctamente y se muestre un mensaje "Producto Modificado".</p> <p>Algún campo está vacío: Se espera que se muestre un mensaje "Los campos están vacíos".</p> <p>No se ha seleccionado ninguna fila en la tabla: Se espera que se muestre un mensaje "Seleccione una fila".</p> <p><b>Prueba de caja negra con condicionales:</b>            Prueba de caso correcto: llenar todos los campos de texto y seleccionar un proveedor en la lista desplegable, asegurándose de que el producto se modifique correctamente.</p> <p>Prueba de caso inválido: no llenar ningún campo de texto, asegurarse de que se muestre un mensaje de error indicando que los campos están vacíos.</p> <p>Prueba de caso inválido: no seleccionar una fila de la tabla, asegurarse de que se muestre un mensaje de error indicando que debe seleccionarse una fila.</p> <p>Prueba de caso inválido: llenar solo algunos campos de texto, asegurarse de que se muestre un mensaje de error indicando que los campos están vacíos.</p>
<i>Observaciones</i>	Ninguno

<b>Caso Prueba 09: (Nuevo Producto)</b>	<i>Código</i>
---	---------------

	<pre> private void btnEliminarProActionPerformed(java.awt.event.ActionEvent evt) {     // TODO add your handling code here:     if (!"".equals(txtIdproducto.getText())) {         int pregunta = JOptionPane.showConfirmDialog(null, "Esta seguro de eliminar");         if (pregunta == 0) {             int id = Integer.parseInt(txtIdproducto.getText());             proDao.EliminarProductos(id);             LimpiarTable();             LimpiarProductos();             ListarProductos();             btnEditarpro.setEnabled(false);             btnEliminarPro.setEnabled(false);             btnGuardarpro.setEnabled(true);         }     } else {         JOptionPane.showMessageDialog(null, "Selecciona una fila");     } } </pre>
<i>Descripción: BtnEliminar</i>	<p>Prueba de eliminación exitosa:</p> <p>Ingrese un ID de producto válido en el campo "txtIdproducto".</p> <p>Seleccione "Sí" en la ventana de confirmación.</p> <p>Verifique que el producto correspondiente al ID se haya eliminado correctamente.</p> <p>Prueba sin seleccionar una fila:</p> <p>No ingrese un ID de producto en el campo "txtIdproducto".</p> <p>Verifique que se muestre un mensaje de error "Selecciona una fila".</p> <p>Prueba de cancelación de eliminación:</p> <p>Ingrese un ID de producto válido en el campo "txtIdproducto".</p> <p>Seleccione "No" en la ventana de confirmación.</p> <p>Verifique que el producto no se haya eliminado y que el ID de producto y otros campos de texto permanezcan intactos.</p>
<i>Observaciones</i>	Ninguno

<b>Caso Prueba 10: (Inicio de Login)</b>	<i>Código</i>
--	---------------

	<pre> package Vista;  import Modelo.LoginDAO; import Modelo.login; import java.awt.Desktop; import java.io.IOException; import java.net.URI; import java.net.URISyntaxException; import javax.swing.JOptionPane; </pre> <pre> public class Login extends javax.swing.JFrame {     login lg = new login();     LoginDAO login = new LoginDAO();     public Login() {         initComponents();         this.setLocationRelativeTo(null);         txtCorreo.setText("74361715@continental.edu.pe");         txtPass.setText("322");     }     public void validar(){         String correo = txtCorreo.getText();         String pass = String.valueOf(txtPass.getPassword());         if (!"".equals(correo)    !"".equals(pass)) {             lg = login.log(correo, pass);             if (lg.getCorreo() != null &amp;&amp; lg.getPass() != null) {                 Sistema sis = new Sistema(lg);                 sis.setVisible(true);                 dispose();             }else{                 JOptionPane.showMessageDialog(null, "Correo o la Contraseña incorrecta");             }         }     }     /* */ } </pre>
<i>Descripción:</i>	<p>Caso de prueba: Inicio de sesión exitoso Entrada: Correo = 74361715@continental.edu.pe, Contraseña = 322 Resultado esperado: Debe mostrar la ventana del sistema y cerrar la ventana de inicio de sesión.</p> <p>Caso de prueba: Inicio de sesión con correo o contraseña incorrecta Entrada: Correo = 74361715@continental.edu.pe, Contraseña = 123 Resultado esperado: Debe mostrar un mensaje de error "Correo o la Contraseña incorrecta".</p> <p>Caso de prueba: Inicio de sesión con campos vacíos Entrada: Correo = "", Contraseña = "" Resultado esperado: Debe mostrar un mensaje de error "Los campos están vacíos".</p>
<i>Observaciones</i>	Ninguno

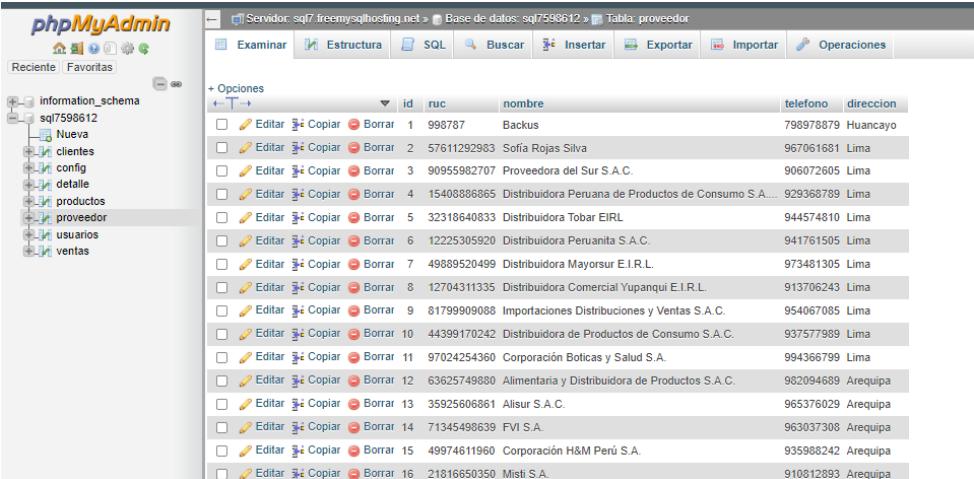
<b>Caso Prueba 11: (Nuevo Usuario)</b>	<b>Código</b>
--	---------------

	<pre> private void btnIniciarActionPerformed(java.awt.event.ActionEvent evt) {     if(txtNombre.getText().equals("")    txtCorreo.getText().equals("")    txtPass.getPassword().equals("")) {         JOptionPane.showMessageDialog(null, "Todos los campos son requeridos");     }else{         String correo = txtCorreo.getText();         String pass = String.valueOf(txtPass.getPassword());         String nom = txtNombre.getText();         String rol = cbxRol.getSelectedItem().toString();         lg.setNombre(nom);         lg.setCorreo(correo);         lg.setPass(pass);         lg.setRol(rol);         login.Registrar(lg);         JOptionPane.showMessageDialog(null, "Usuario Registrado");         LimpiearTable();         ListarUsuarios();         nuevoUsuario();     } } </pre>
<i>Descripción:</i> <b>BtnRegistrar</b>	<p>Campos vacíos:  Entrada: txtNombre.getText() = "", txtCorreo.getText() = "", txtPass.getPassword() = ""  Salida esperada: JOptionPane con mensaje "Todos los campos son requeridos"  Condición: Si txtNombre, txtCorreo o txtPass están vacíos, se muestra un mensaje de error</p> <p>Campos llenos:  Entrada: txtNombre.getText() = "John Doe", txtCorreo.getText() = "johndoe@email.com", txtPass.getPassword() = "123456"  Salida esperada: JOptionPane con mensaje "Usuario registrado" y los datos ingresados se agregaron a la lista de usuarios  Condición: Si txtNombre, txtCorreo y txtPass no están vacíos, se llama al método Registrar y se muestra un mensaje de éxito.</p>
<i>Observaciones</i>	<p>Se asume que el método LimpiearTable, ListarUsuarios y nuevoUsuario funcionan correctamente y se encargan de limpiar la tabla, listar los usuarios y preparar un nuevo registro de usuario respectivamente</p>

<b>Caso Prueba 12:</b> <i>(Actualización de Empresa)</i>	<i>Código</i>
	<pre> private void btnActualizarConfigActionPerformed(java.awt.event.ActionEvent evt) {     // TODO add your handling code here:     if (!"".equals(txtRucConfig.getText())    !"".equals(txtNombreConfig.getText())    !"".equals(         conf.setRuc(txtRucConfig.getText());         conf.setNombre(txtNombreConfig.getText());         conf.setTelefono(txtTelefonoConfig.getText());         conf.setDireccion(txtDireccionConfig.getText());         conf.setMensaje(txtMensaje.getText());         conf.setId(Integer.parseInt(txtIdConfig.getText()));         proDao.ModificarDatos(conf);         JOptionPane.showMessageDialog(null, "Datos de la empresa modificado");         ListarConfig();     } else {         JOptionPane.showMessageDialog(null, "Los campos estan vacios");     } } </pre>
<i>Descripción:</i> <b>BtnActualizar</b>	<p>Campos vacíos:  Entrada: txtRucConfig.getText() = "", txtNombreConfig.getText() = "",  txtTelefonoConfig.getText() = "", txtDireccionConfig.getText() = "" Salida esperada:  JOptionPane con mensaje "Los campos están vacíos" Condición: Si txtRucConfig,</p>

	<p>txtNombreConfig, txtTelefonoConfig o txtDireccionConfig están vacíos, se muestra un mensaje de error</p> <p>Campos llenos:</p> <p>Entrada: txtRucConfig.getText() = "123456789", txtNombreConfig.getText() = "Empresa X", txtTelefonoConfig.getText() = "987654321", txtDireccionConfig.getText() = "Calle 1" Salida esperada: JOptionPane con mensaje "Datos de la empresa modificados" y los datos ingresados se actualizan en la lista de configuraciones Condición: Si txtRucConfig, txtNombreConfig, txtTelefonoConfig y txtDireccionConfig no están vacíos, se llama al método proDao.ModificarDatos y se muestra un mensaje de éxito.</p>
<i>Observaciones</i>	Ninguno

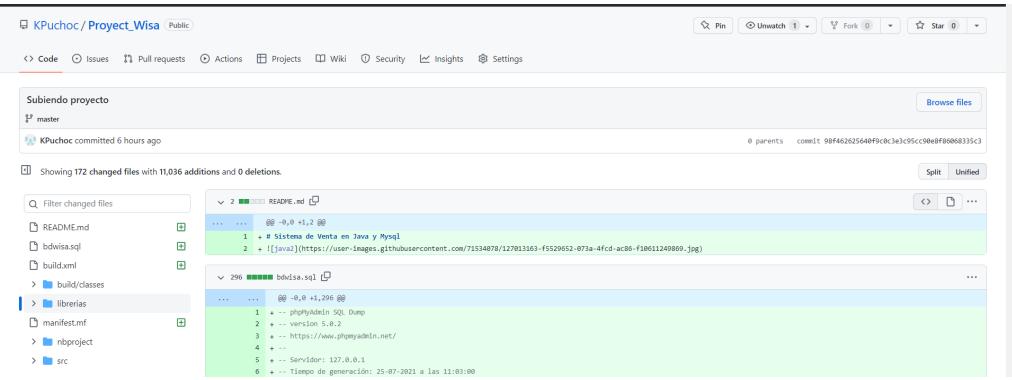
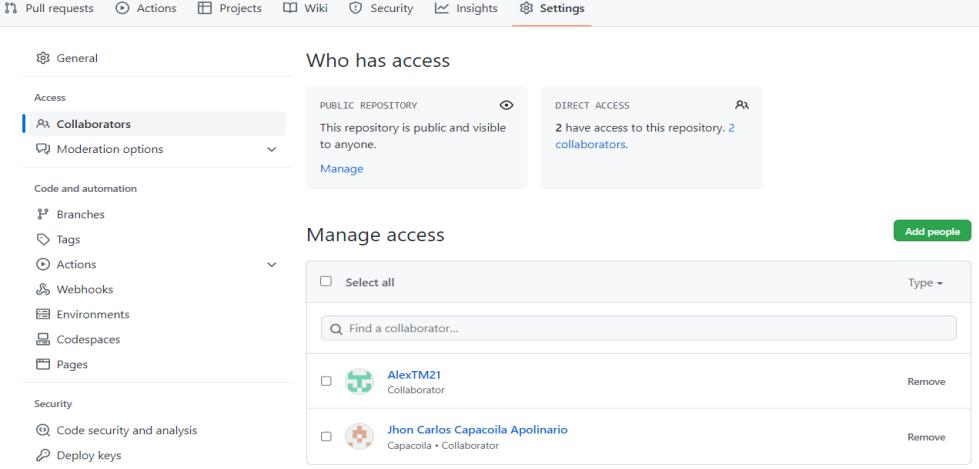
<b>Caso Prueba 13: (Inserción de Datos clientes)</b>	<b>Código</b>
	
<i>INSERT INTO</i>	<pre>INSERT INTO `clientes`(`id`, `dni`, `nombre`, `telefono`, `direccion`) VALUES (5,'66410628','Lucía Flores Pérez','998949513','Calle de las Pizzas'), (6,'95684385','Manuel Gómez Torres','943197140','Avenida Larco'), (7,'67147677','Ana Karenina García Torres','916042221','Avenida Grau'), (8,'27545650','Luis Antonio López González','987973210','Avenida Caminos del Inca'), (9,'80694914','Valeria Sánchez Romero','922213196','Avenida La Marina'), (10,'42529980','Santiago González Ríos','914267183','Avenida Salaverry'), (11,'42920416','Andrea Torres Mendoza','901646740','Avenida Angamos'), (12,'78664237','Gabriel González León','982030529','Avenida Brasil'), (13,'93995001','Martina Acosta Vega','949971752','Calle de las Brujas'), (14,'89721174','Pablo Romero Cruz','995635825','Jirón de la Unión'), ... </pre>
<i>Observaciones</i>	Ninguno

<b>Caso Prueba 14:</b> <i>(Inserción de Datos Proveedor)</i>	<b>Código</b>
	
<b>INSERT INTO</b>	<pre>INSERT INTO `proveedor`(`id`, `ruc`, `nombre`, `telefono`, `direccion`) VALUES (2,'57611292983','Sofía Rojas Silva','967061681','Lima'), (3,'90955982707','Proveedora del Sur S.A.C.','906072605','Lima'), (4,'15408886865','Distribuidora Peruana de Productos de Consumo S.A.C.','929368789','Lima'), (5,'32318640833','Distribuidora Tobar EIRL','944574810','Lima'), (6,'12225305920','Distribuidora Peruanita S.A.C.','941761505','Lima'), (7,'49889520499','Distribuidora Mayorsur E.I.R.L.','973481305','Lima'), (8,'12704311335','Distribuidora Comercial Yupanqui E.I.R.L.','913706243','Lima'), (9,'81799909088','Importaciones Distribuciones y Ventas S.A.C.','954067085','Lima'), (10,'44399170242','Distribuidora de Productos de Consumo S.A.C.','937577989','Lima'), (11,'97024254360','Corporación Boticas y Salud S.A.','994366799','Lima'), (12,'63625749880','Alimentaria y Distribuidora de Productos S.A.C.','982094689','Arequipa'), (13,'35925606861','Alisur S.A.C.','965376029','Arequipa'), (14,'71345498639','FVI S.A.','963037308','Arequipa'), (15,'49974611960','Corporación H&amp;M Perú S.A.','935988242','Arequipa'), (16,'21816650350','Misti S.A.','910812893','Arequipa'), (17,'42046388372','Americana de Desarrollo S.A.','975128907','Arequipa'), (18,'24089904355','Unión de Negocios S.A.C.','957427373','Arequipa'), (19,'16396207895','Distribuidora de Productos de Consumo y Comercio en General S.A.C.','927442932','Arequipa'), (20,'49077862206','Importadora y Distribuidora Comercial R&amp;J E.I.R.L.','908867490','Arequipa'), ...</pre>
<b>Observaciones</b>	Ninguno

<b>Caso Prueba 15:</b> <i>(Inserción de Datos Productos)</i>	<b>Código</b>																																																																																																																								
	 <table border="1"> <thead> <tr> <th>ID</th> <th>CÓDIGO</th> <th>DESCRIPCIÓN</th> <th>PROVEEDOR</th> <th>STOCK</th> <th>PRECIO</th> </tr> </thead> <tbody> <tr><td>43</td><td>P00043</td><td>Mayonesa</td><td>Distribuidora Co...</td><td>24</td><td>6.0</td></tr> <tr><td>42</td><td>P00042</td><td>Azucar morena</td><td>Distribuidora Co...</td><td>24</td><td>9.0</td></tr> <tr><td>41</td><td>P00041</td><td>Vinagre</td><td>Distribuidora Co...</td><td>24</td><td>3.0</td></tr> <tr><td>40</td><td>P00040</td><td>Salsa de tomate</td><td>Distribuidora May...</td><td>24</td><td>5.0</td></tr> <tr><td>39</td><td>P00039</td><td>Azucar blanca</td><td>Distribuidora May...</td><td>24</td><td>11.0</td></tr> <tr><td>38</td><td>P00038</td><td>Frijoles, lentejas</td><td>Distribuidora May...</td><td>12</td><td>6.0</td></tr> <tr><td>37</td><td>P00037</td><td>Arroz blanco</td><td>Distribuidora May...</td><td>12</td><td>15.0</td></tr> <tr><td>36</td><td>P00036</td><td>Harina y levadura para pa...</td><td>Distribuidora May...</td><td>12</td><td>6.0</td></tr> <tr><td>35</td><td>P00035</td><td>Carnes enlatadas, como...</td><td>Distribuidora Peru...</td><td>12</td><td>15.0</td></tr> <tr><td>34</td><td>P00034</td><td>Pollo fresco</td><td>Distribuidora Peru...</td><td>12</td><td>11.0</td></tr> <tr><td>33</td><td>P00033</td><td>Fiambres y quesos</td><td>Distribuidora Peru...</td><td>12</td><td>5.0</td></tr> <tr><td>32</td><td>P00032</td><td>Hamburguesas</td><td>Distribuidora Peru...</td><td>12</td><td>3.0</td></tr> <tr><td>31</td><td>P00031</td><td>Carne seca</td><td>Distribuidora Peru...</td><td>12</td><td>15.0</td></tr> <tr><td>30</td><td>P00030</td><td>Huevos frescos</td><td>Distribuidora Tob...</td><td>12</td><td>2.0</td></tr> <tr><td>29</td><td>P00029</td><td>Postres lácteos como flan</td><td>Distribuidora Tob...</td><td>12</td><td>5.0</td></tr> <tr><td>28</td><td>P00028</td><td>Crema de leche y nata liq.</td><td>Distribuidora Tob...</td><td>12</td><td>5.0</td></tr> <tr><td>27</td><td>P00027</td><td>Queso de varias variedad...</td><td>Distribuidora Tob...</td><td>50</td><td>9.0</td></tr> <tr><td>26</td><td>P00026</td><td>Mantequilla y margarina e...</td><td>Distribuidora Tob...</td><td>50</td><td>3.0</td></tr> <tr><td>25</td><td>P00025</td><td>Yogurt natural</td><td>Distribuidora Tob...</td><td>50</td><td>4.5</td></tr> </tbody> </table>	ID	CÓDIGO	DESCRIPCIÓN	PROVEEDOR	STOCK	PRECIO	43	P00043	Mayonesa	Distribuidora Co...	24	6.0	42	P00042	Azucar morena	Distribuidora Co...	24	9.0	41	P00041	Vinagre	Distribuidora Co...	24	3.0	40	P00040	Salsa de tomate	Distribuidora May...	24	5.0	39	P00039	Azucar blanca	Distribuidora May...	24	11.0	38	P00038	Frijoles, lentejas	Distribuidora May...	12	6.0	37	P00037	Arroz blanco	Distribuidora May...	12	15.0	36	P00036	Harina y levadura para pa...	Distribuidora May...	12	6.0	35	P00035	Carnes enlatadas, como...	Distribuidora Peru...	12	15.0	34	P00034	Pollo fresco	Distribuidora Peru...	12	11.0	33	P00033	Fiambres y quesos	Distribuidora Peru...	12	5.0	32	P00032	Hamburguesas	Distribuidora Peru...	12	3.0	31	P00031	Carne seca	Distribuidora Peru...	12	15.0	30	P00030	Huevos frescos	Distribuidora Tob...	12	2.0	29	P00029	Postres lácteos como flan	Distribuidora Tob...	12	5.0	28	P00028	Crema de leche y nata liq.	Distribuidora Tob...	12	5.0	27	P00027	Queso de varias variedad...	Distribuidora Tob...	50	9.0	26	P00026	Mantequilla y margarina e...	Distribuidora Tob...	50	3.0	25	P00025	Yogurt natural	Distribuidora Tob...	50	4.5
ID	CÓDIGO	DESCRIPCIÓN	PROVEEDOR	STOCK	PRECIO																																																																																																																				
43	P00043	Mayonesa	Distribuidora Co...	24	6.0																																																																																																																				
42	P00042	Azucar morena	Distribuidora Co...	24	9.0																																																																																																																				
41	P00041	Vinagre	Distribuidora Co...	24	3.0																																																																																																																				
40	P00040	Salsa de tomate	Distribuidora May...	24	5.0																																																																																																																				
39	P00039	Azucar blanca	Distribuidora May...	24	11.0																																																																																																																				
38	P00038	Frijoles, lentejas	Distribuidora May...	12	6.0																																																																																																																				
37	P00037	Arroz blanco	Distribuidora May...	12	15.0																																																																																																																				
36	P00036	Harina y levadura para pa...	Distribuidora May...	12	6.0																																																																																																																				
35	P00035	Carnes enlatadas, como...	Distribuidora Peru...	12	15.0																																																																																																																				
34	P00034	Pollo fresco	Distribuidora Peru...	12	11.0																																																																																																																				
33	P00033	Fiambres y quesos	Distribuidora Peru...	12	5.0																																																																																																																				
32	P00032	Hamburguesas	Distribuidora Peru...	12	3.0																																																																																																																				
31	P00031	Carne seca	Distribuidora Peru...	12	15.0																																																																																																																				
30	P00030	Huevos frescos	Distribuidora Tob...	12	2.0																																																																																																																				
29	P00029	Postres lácteos como flan	Distribuidora Tob...	12	5.0																																																																																																																				
28	P00028	Crema de leche y nata liq.	Distribuidora Tob...	12	5.0																																																																																																																				
27	P00027	Queso de varias variedad...	Distribuidora Tob...	50	9.0																																																																																																																				
26	P00026	Mantequilla y margarina e...	Distribuidora Tob...	50	3.0																																																																																																																				
25	P00025	Yogurt natural	Distribuidora Tob...	50	4.5																																																																																																																				
<b>INSERT INTO</b>	<pre>INSERT INTO `proveedor`(`id`, `ruc`, `nombre`, `telefono`, `direccion`) VALUES (2,'P00002','Cerveza Cristal','1','60','9.9'), (3,'P00003','Cerveza Pilsen','1','60','10.5'), (4,'P00004','Cerveza Cusqueña','1','60','11.5'), (5,'P00005','Cerveza Arequipeña','1','60','10'), (6,'P00006','Cerveza Trujillo','1','60','9'), (7,'P00007','Cerveza San Juan','1','60','8'), (8,'P00008','Cerveza Barena','1','60','9'), (9,'P00009','Cerveza Malta Cusqueña','1','60','8.5'), (10,'P00010','Cerveza Pilsen Trujillo','1','60','9'), (11,'P00011','Cerveza Pilsen Callao','1','60','10'), (12,'P00012','Chicles ','2','60','0.5'), (13,'P00013','Chocolate en barra y en paquete.','2','60','2.5'), (14,'P00014','Gomas de mascar','2','60','1'), (15,'P00015','Galletas de chocolate','2','60','1.5'), (16,'P00016','Barras de cereal','3','60','0.7'), (17,'P00017','Bombones / trufas.','3','50','1'), (18,'P00018','Paletas de dulce','3','50','1'),</pre>																																																																																																																								
<b>Observaciones</b>	Ninguno																																																																																																																								

### Control de Versiones github

Descripción	Imagen
-------------	--------

<h3>Crear Repositorio público</h3>	<p><b>Create a new repository</b></p> <p>A repository contains all project files, including the revision history. Already have a project repository elsewhere? <a href="#">Import a repository.</a></p> <hr/> <p><b>Owner *</b> <b>Repository name *</b></p> <p>KPuchoc / Proyecto_Wisa ✓</p> <p>Great repository names are short and memorable. Need inspiration? How about <a href="#">curly-eureka?</a></p> <p><b>Description (optional)</b></p> <p>Implementación de un Software de Ventas para la Gestión y Control de Ventas del Minimarket WISA</p> <hr/> <p><input checked="" type="radio"/> <b>Public</b> Anyone on the internet can see this repository. You choose who can commit.</p> <p><input type="radio"/> <b>Private</b> You choose who can see and commit to this repository.</p>
<h3>Remotamente se añadió el proyecto</h3>	
<h3>Añadir a colaboradores</h3>	

<p><b>Pruebas Commit</b> <b>Fecha: 24/01/23</b></p>	<p>Commits on Jan 24, 2023</p> <table border="1"> <tbody> <tr><td>Prueba de commit 1</td><td>KPuchoc committed 5 hours ago</td><td></td><td>555cc9</td><td></td></tr> <tr><td>prueva9</td><td>KPuchoc committed 6 hours ago</td><td></td><td>d0ec869</td><td></td></tr> <tr><td>Merge branch 'master' of https://github.com/KPuchoc/Proyect_Wisa</td><td>KPuchoc committed 6 hours ago</td><td></td><td>47e3e61</td><td></td></tr> <tr><td>Subio nueva imagenes</td><td>KPuchoc committed 6 hours ago</td><td></td><td>2618983</td><td></td></tr> <tr><td>Update README.md</td><td>KPuchoc committed 6 hours ago</td><td></td><td>Verified 97daed</td><td></td></tr> <tr><td>Subiendo proyecto</td><td>KPuchoc committed 6 hours ago</td><td></td><td>98f4626</td><td></td></tr> </tbody> </table>	Prueba de commit 1	KPuchoc committed 5 hours ago		555cc9		prueva9	KPuchoc committed 6 hours ago		d0ec869		Merge branch 'master' of https://github.com/KPuchoc/Proyect_Wisa	KPuchoc committed 6 hours ago		47e3e61		Subio nueva imagenes	KPuchoc committed 6 hours ago		2618983		Update README.md	KPuchoc committed 6 hours ago		Verified 97daed		Subiendo proyecto	KPuchoc committed 6 hours ago		98f4626																										
Prueba de commit 1	KPuchoc committed 5 hours ago		555cc9																																																					
prueva9	KPuchoc committed 6 hours ago		d0ec869																																																					
Merge branch 'master' of https://github.com/KPuchoc/Proyect_Wisa	KPuchoc committed 6 hours ago		47e3e61																																																					
Subio nueva imagenes	KPuchoc committed 6 hours ago		2618983																																																					
Update README.md	KPuchoc committed 6 hours ago		Verified 97daed																																																					
Subiendo proyecto	KPuchoc committed 6 hours ago		98f4626																																																					
<p><b>Pruebas Commit</b> <b>Fecha: 25/01/23</b></p>	<p>Commits on Jan 25, 2023</p> <table border="1"> <tbody> <tr><td>Update README.md</td><td>KPuchoc committed 16 minutes ago</td><td></td><td>Verified 9d204d1</td><td></td></tr> <tr><td>Update README.md</td><td>KPuchoc committed 16 minutes ago</td><td></td><td>Verified c1b8754</td><td></td></tr> <tr><td>Update README.md</td><td>KPuchoc committed 17 minutes ago</td><td></td><td>Verified 5b6a48c</td><td></td></tr> <tr><td>Update README.md</td><td>KPuchoc committed 24 minutes ago</td><td></td><td>Verified d9cb368</td><td></td></tr> <tr><td>Prueba 4 - Actualizacion de iconos</td><td>KPuchoc committed 35 minutes ago</td><td></td><td>ca2b9ae</td><td></td></tr> <tr><td>Prueba 3 - LOGIN FINAL</td><td>KPuchoc committed 2 hours ago</td><td></td><td>4b20933</td><td></td></tr> <tr><td>Prueba 2 Login y Base de Datos</td><td>KPuchoc committed 3 hours ago</td><td></td><td>840979b</td><td></td></tr> </tbody> </table>	Update README.md	KPuchoc committed 16 minutes ago		Verified 9d204d1		Update README.md	KPuchoc committed 16 minutes ago		Verified c1b8754		Update README.md	KPuchoc committed 17 minutes ago		Verified 5b6a48c		Update README.md	KPuchoc committed 24 minutes ago		Verified d9cb368		Prueba 4 - Actualizacion de iconos	KPuchoc committed 35 minutes ago		ca2b9ae		Prueba 3 - LOGIN FINAL	KPuchoc committed 2 hours ago		4b20933		Prueba 2 Login y Base de Datos	KPuchoc committed 3 hours ago		840979b																					
Update README.md	KPuchoc committed 16 minutes ago		Verified 9d204d1																																																					
Update README.md	KPuchoc committed 16 minutes ago		Verified c1b8754																																																					
Update README.md	KPuchoc committed 17 minutes ago		Verified 5b6a48c																																																					
Update README.md	KPuchoc committed 24 minutes ago		Verified d9cb368																																																					
Prueba 4 - Actualizacion de iconos	KPuchoc committed 35 minutes ago		ca2b9ae																																																					
Prueba 3 - LOGIN FINAL	KPuchoc committed 2 hours ago		4b20933																																																					
Prueba 2 Login y Base de Datos	KPuchoc committed 3 hours ago		840979b																																																					
<p><b>Pruebas Commit</b> <b>Fecha: 25/01/23</b></p>	<p>Commits on Jan 25, 2023</p> <table border="1"> <tbody> <tr><td>Prueba 6 - Seguir editando</td><td>KPuchoc committed 2 weeks ago</td><td></td><td>7b50591</td><td></td></tr> <tr><td>Merge branch 'master' of https://github.com/KPuchoc/Proyect_Wisa</td><td>KPuchoc committed 2 weeks ago</td><td></td><td>18cc1ea</td><td></td></tr> <tr><td>Prueba 5 - Seguir editando</td><td>KPuchoc committed 2 weeks ago</td><td></td><td>453d71c</td><td></td></tr> <tr><td>Signed-off-by: KPuchoc &lt;74361715@continental.edu.pe&gt;</td><td>KPuchoc committed 2 weeks ago</td><td></td><td>7ed88d9</td><td></td></tr> <tr><td>Update README.md</td><td>KPuchoc committed 2 weeks ago</td><td></td><td>Verified 9d204d1</td><td></td></tr> <tr><td>Update README.md</td><td>KPuchoc committed 2 weeks ago</td><td></td><td>Verified c1b8754</td><td></td></tr> <tr><td>Update README.md</td><td>KPuchoc committed 2 weeks ago</td><td></td><td>Verified 5b6a48c</td><td></td></tr> <tr><td>Update README.md</td><td>KPuchoc committed 2 weeks ago</td><td></td><td>Verified d9cb368</td><td></td></tr> <tr><td>Prueba 4 - Actualizacion de iconos</td><td>KPuchoc committed 2 weeks ago</td><td></td><td>ca2b9ae</td><td></td></tr> <tr><td>Prueba 3 - LOGIN FINAL</td><td>KPuchoc committed 2 weeks ago</td><td></td><td>4b20933</td><td></td></tr> <tr><td>Prueba 2 Login y Base de Datos</td><td>KPuchoc committed 2 weeks ago</td><td></td><td>840979b</td><td></td></tr> </tbody> </table>	Prueba 6 - Seguir editando	KPuchoc committed 2 weeks ago		7b50591		Merge branch 'master' of https://github.com/KPuchoc/Proyect_Wisa	KPuchoc committed 2 weeks ago		18cc1ea		Prueba 5 - Seguir editando	KPuchoc committed 2 weeks ago		453d71c		Signed-off-by: KPuchoc <74361715@continental.edu.pe>	KPuchoc committed 2 weeks ago		7ed88d9		Update README.md	KPuchoc committed 2 weeks ago		Verified 9d204d1		Update README.md	KPuchoc committed 2 weeks ago		Verified c1b8754		Update README.md	KPuchoc committed 2 weeks ago		Verified 5b6a48c		Update README.md	KPuchoc committed 2 weeks ago		Verified d9cb368		Prueba 4 - Actualizacion de iconos	KPuchoc committed 2 weeks ago		ca2b9ae		Prueba 3 - LOGIN FINAL	KPuchoc committed 2 weeks ago		4b20933		Prueba 2 Login y Base de Datos	KPuchoc committed 2 weeks ago		840979b	
Prueba 6 - Seguir editando	KPuchoc committed 2 weeks ago		7b50591																																																					
Merge branch 'master' of https://github.com/KPuchoc/Proyect_Wisa	KPuchoc committed 2 weeks ago		18cc1ea																																																					
Prueba 5 - Seguir editando	KPuchoc committed 2 weeks ago		453d71c																																																					
Signed-off-by: KPuchoc <74361715@continental.edu.pe>	KPuchoc committed 2 weeks ago		7ed88d9																																																					
Update README.md	KPuchoc committed 2 weeks ago		Verified 9d204d1																																																					
Update README.md	KPuchoc committed 2 weeks ago		Verified c1b8754																																																					
Update README.md	KPuchoc committed 2 weeks ago		Verified 5b6a48c																																																					
Update README.md	KPuchoc committed 2 weeks ago		Verified d9cb368																																																					
Prueba 4 - Actualizacion de iconos	KPuchoc committed 2 weeks ago		ca2b9ae																																																					
Prueba 3 - LOGIN FINAL	KPuchoc committed 2 weeks ago		4b20933																																																					
Prueba 2 Login y Base de Datos	KPuchoc committed 2 weeks ago		840979b																																																					

Pruebas Commit  
Fecha: 04/02/23

Commits on Feb 4, 2023	<a href="#">6c81e6c</a>	<>
Prueba 10 - Eliminar copia de la base de datos	<a href="#">6c81e6c</a>	<>
KPuchoc committed 3 days ago		
Prueba 9 - Arreglo de errores de la bdwisa	<a href="#">8afdfc2</a>	<>
KPuchoc committed 3 days ago		
Prueba 8 - Agregar nueva base de datos	<a href="#">379c40a</a>	<>
KPuchoc committed 3 days ago		
Prueba 7 - Copia de Base de Datos	<a href="#">2945953</a>	<>
KPuchoc committed 4 days ago		

Showing 1 changed file with 9 additions and 37 deletions.

Line	Line	Changes
248	219	- MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=7;
	220	+ MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=2;
249	221	
250	222	--
251	223	-- AUTO_INCREMENT de la tabla `config`
	224	@@ -257,31 +229,31 @@ ALTER TABLE `config`
257	229	-- AUTO_INCREMENT de la tabla `detalle`
258	230	--
259	231	ALTER TABLE `detalle`
260	232	- MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=14;
	233	+ MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=1;
261	234	
262	235	--
263	236	-- AUTO_INCREMENT de la tabla `productos`
264	237	--
265	238	ALTER TABLE `productos`
266	239	- MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=6;
	240	+ MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=1;
267	241	
268	242	--
269	243	-- AUTO_INCREMENT de la tabla `proveedor`
270	244	--
271	245	ALTER TABLE `proveedor`
272	246	- MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=4;
	247	+ MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=2;
273	248	
274	249	--
275	250	-- AUTO_INCREMENT de la tabla `usuarios`
276	251	--
277	252	ALTER TABLE `usuarios`
278	253	- MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=6;
	254	+ MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=2;
279	255	
280	256	--
281	257	-- AUTO_INCREMENT de la tabla `ventas`
282	258	--
283	259	ALTER TABLE `ventas`
284	260	- MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=11;
	261	+ MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=1;
285	262	
286	263	--

<p>Pruebas Commit Fecha: 08/02/23</p>	
<p>Pruebas Commit Fecha: 20/02/23</p>	
<p>git bas</p>	<pre>willian@DESKTOP-SJLTNMT MINGW64 ~/Downloads/wisa3 (master) \$ git status On branch master Your branch is up to date with 'origin/master'.  Changes to be committed:   (use "git restore --staged &lt;file&gt;..." to unstage)     new file:   Ventas01/wisa3.jar     new file:   VentasNube/wisa3.jar     modified:  build.xml     modified:  build/built-jar.properties     deleted:   build/classes/.netbeans_automatic_build     deleted:   build/classes/.netbeans_update_resources     modified:  build/classes/Modelo/Cliente.class     modified:  build/classes/Modelo/ClienteDao.class     modified:  build/classes/Modelo/Combo.class     modified:  build/classes/Modelo/Conexion.class     modified:  build/classes/Modelo/Config.class     modified:  build/classes/Modelo/Detalle.class     modified:  build/classes/Modelo/Eventos.class     modified:  build/classes/Modelo/LoginDAO.class     modified:  build/classes/Modelo/Productos.class     modified:  build/classes/Modelo/ProductosDao.class     modified:  build/classes/Modelo/Proveedor.class     modified:  build/classes/Modelo/ProveedorDao.class     modified:  build/classes/Modelo/Venta.class     modified:  build/classes/Modelo/VentaDao.class     modified:  build/classes/Modelo/login.class     modified:  build/classes/Reportes/Grafico.class     modified:  build/classes/Vista/Login\$1.class     modified:  build/classes/Vista/Login\$2.class</pre>

```

willian@DESKTOP-SJLTNMT MINGW64 ~/Downloads/wisa3 (master)
$ git commit -m "Prueba 15 - Pruebas de integracion"
[master 6506502] Prueba 15 - Pruebas de integracion
 91 files changed, 12 insertions(+), 3026 deletions(-)
 create mode 100644 Ventas01/wisa3.jar
 create mode 100644 VentasNube/wisa3.jar
 delete mode 100644 build/classes/.netbeans_automatic_build
 delete mode 100644 build/classes/.netbeans_update_resources
 delete mode 100644 build/classes/Vista/Login.form
 delete mode 100644 build/classes/Vista/Sistema.form
 delete mode 100644 build/classes/prueba001.txt
 create mode 100644 insercion de datos.xlsx
 delete mode 100644 src/prueba001.txt

willian@DESKTOP-SJLTNMT MINGW64 ~/Downloads/wisa3 (master)
$ git push -u origin master
Enumerating objects: 189, done.
Counting objects: 100% (189/189), done.
Compressing objects: 100% (96/96), done.
Writing objects: 100% (100/100), 7.08 MiB | 793.00 KiB/s, done.
Total 100 (delta 84), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (84/84), completed with 58 local objects.
To https://github.com/KPuchoc/Proyect_Wisa.git
  5bc9df0..6506502 master -> master
branch 'master' set up to track 'origin/master'.

willian@DESKTOP-SJLTNMT MINGW64 ~/Downloads/wisa3 (master)
$ git push -u origin master

```

## GitHub del proyecto

[https://github.com/KPuchoc/Proyect\\_Wisa](https://github.com/KPuchoc/Proyect_Wisa)

## 4.6.- Suspensión de Pruebas

A continuación se describen los motivos por los que se puede interrumpir la ejecución de las pruebas.

- Si hay fallas en exportar el historial de ventas, no tomar acción puede resultar errores coincidencias con los datos guardados .
- Si no se han definido los casos de prueba.
- Tras realizar una actualización en la base de datos, se han identificado uno o más fallos graves que obstaculizaron podría afectar el avance de las pruebas.
- No se dispone de la versión final del software
- Si el software destinado a ser probado no está disponible en los plazos establecidos del cronograma del proyecto.

#### **4.7.- Reanudación de pruebas**

A continuación se indican los criterios por los cuales se podrá reanudar la ejecución de las pruebas:

- Se reanudarán cuando se solucionen los problemas que las originaron. Si la razón de la interrupción fue un fallo grave, se debe comprobar la reparación antes de reiniciar oficialmente las actividades de prueba.
- Contar con la versión final del sistema.
- Disponibilidad de herramientas para la ejecución del servicio.
- Actualización de datos de la empresa.
- Reorganización de la base de datos.

#### **4.8- Validación y aceptación**

A continuación se indican los criterios por los cuales se podrá dar por satisfactoria la validación de Calidad:

- Precisión y cálculo de las fórmulas mal implementadas. Por otro lado la

sobrecarga el sistema funciona bien con 100 usuarios, pero no con 110.

- Capacidad o límites, el sistema funciona bien con importes menores a 1000
- Relativos al hardware o software del sistema, las incompatibilidades entre componentes.

Solo en estos casos se considerará que una prueba y el sistema en general son válidos y se autorizará su liberación por parte del departamento de TESTEO.

#### **4.9.-Revisión de documentos**

En esta sección la gestión de estos documentos es asignada al director del proyecto o al responsable de la almacenamiento y gestión de información para la validación del proyecto. Se pueden almacenar los resultados a verificar en un servidor o en la nube mediante herramientas o servicios. La elección de la herramienta dependerá del tipo de proyecto, y en algunos casos se recomienda el uso de herramientas especializadas.

#### **4.10.- Comunicación**

<b>Actor 1 / Rol</b>	<b>Medio de comunicación</b>	<b>Evidencia</b>	<b>Fecha</b>
Director del proyecto	Reunión meet/Presencial	Firma/Firma digital	01-02-2023
Equipo de trabajo	Reunión meet/Presencial	Firma/Firma digital	01-02-2023
Líder pruebas de software	Reunión meet/Presencial	Firma/Firma digital	01-02-2023
Líder pruebas de Testing	Reunión meet/Presencial	Firma/Firma digital	04-02-2023
Patrocinador Principal	Reunión meet/Presencial	Firma/Firma digital	04-02-2023
Cliente	Reunión meet/Presencial	Firma/Firma digital	06-02-2023

#### **4.11.- Reporte de hechos**

<b>Id</b>	<b>Incidencia</b>	<b>Fecha</b>	<b>Responsable / Rol</b>
R001	Inicio del Proyecto	16-01-2023	Director del proyecto
R002	Presentación del plan de pruebas	23-01-2023	Equipo de trabajo
R003	Diseño del proyecto	25-01-2023	Equipo de trabajo
R004	Informe plan de pruebas funcionales respecto al software.	01-01-2023	Equipo de trabajo
R005	Testeo y control de calidad	06-02-2023	Equipo de trabajo
R006	Instalación y soporte técnico	13-02-2023	Equipo de trabajo
R007	Fin del proyecto	28-02-2023	Equipo de trabajo

#### **4.12.- Definiciones**

*Pruebas unitarias:* Las pruebas unitarias verifican si cada uno de los componentes de forma independiente funciona correctamente a través de su interfaz, cubren la funcionalidad establecida, y se ajustan a los requisitos especificados en las verificaciones correspondientes.

*Pruebas funcionales:* Las pruebas de integración verifican si los componentes o subsistemas en su conjunto interactúan correctamente a través de su interfaz, cubren la funcionalidad establecida, y se ajustan a los requisitos especificados en las verificaciones correspondientes.

*Pruebas integrales:* Se prueban las funcionalidades del sistema con las relaciones a los módulos definidos.

*Pruebas de rendimientos:* Las Pruebas de Rendimiento se ejecutan tanto para determinar cómo responde un sistema ante una cierta carga, como para validar otros atributos relacionados con la calidad, como pueden ser la escalabilidad, la fiabilidad o

el uso de recursos entre otros.

*Pruebas de seguridad:* Las Pruebas de Seguridad pretenden medir la Confidencialidad, Integridad y Disponibilidad de los datos tratando de identificar amenazas y riesgos desde el uso o interface de usuario final.

*Pruebas de caja negra:* Se prueban las funcionalidades de cada módulo con las respectivas entradas de datos y salidas de información.

*El plan de prueba:* Describe todos los métodos que se utilizarán para verificar que el software satisface la especificación del producto y las necesidades del cliente. Incluye los objetivos de calidad, necesidades de recursos, cronograma, asignaciones, métodos, etc.

*Casos de prueba:* Lista los ítems específicos que serán probados y describe los pasos detallados que serán seguidos para verificar el software.

*Reporte de pruebas:* Describen los problemas encontrados al ejecutar los casos de prueba.

*Herramientas de pruebas y automatización:* Documentación de las herramientas empleadas en el proceso de pruebas.

*Métricas, estadísticas y resúmenes:* Indican como ha sido el progreso del proceso de prueba.

*Estrategia:* Serie de acciones/actividades a realizar para establecer un marco de trabajo.

*Verificación:* Actividad para revisar si un entregable o producto es correcto porque cumple con la especificación de su diseño.

*Producto:* Documento que es requerido en cada una de las fases y/o documentación con la que debe cumplir un rol determinado.

**Incidencia:** Documento que es requerido en cada una de las fases y/o documentación con la que debe cumplir un rol determinado.

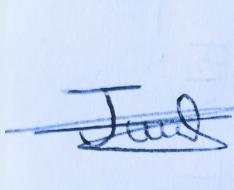
## SECCIÓN V: AUTÓGRAFOS

### 5.1.- Firmas

Por parte del área de Pruebas y desarrollo:

<b>Lider Proyecto</b>

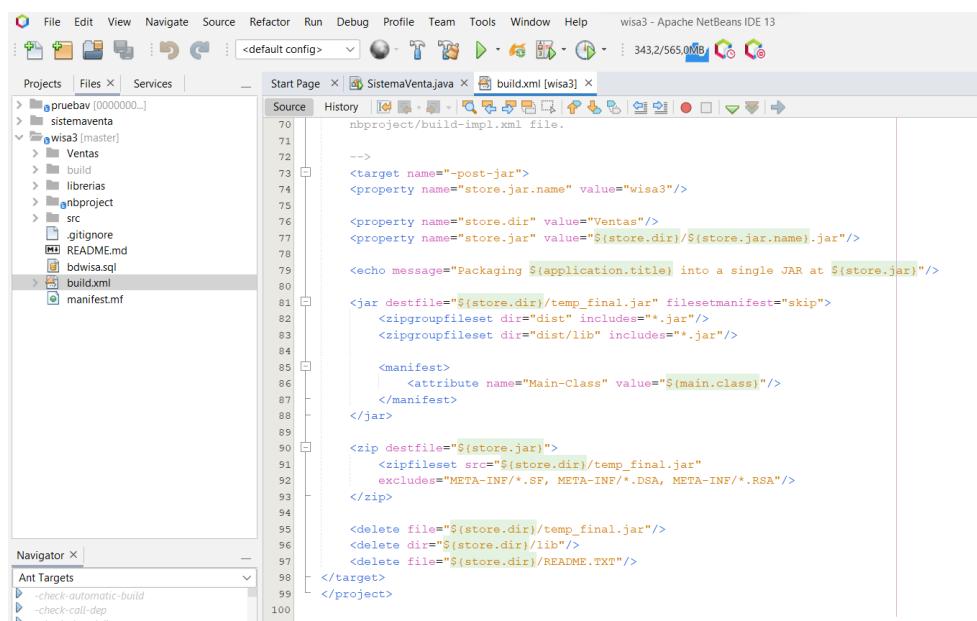
Puchoc Laura Kenny William
<b>Pruebas de software</b>

Capacoila Apolinario Jhon
<b>Pruebas de testing</b>

Torres Montes Alex Junior

## SECCIÓN VI: ANEXOS

### 6.1.- Instalador del proyecto de java

Creación del punto jar para la instalación.



```
<target name="-post-jar">
<property name="store.jar.name" value="wisa3"/>

<property name="store.dir" value="Ventas"/>
<property name="store.jar" value="${store.dir}/${store.jar.name}.jar"/>

<echo message="Packaging ${application.title} into a single JAR at ${store.jar}"/>

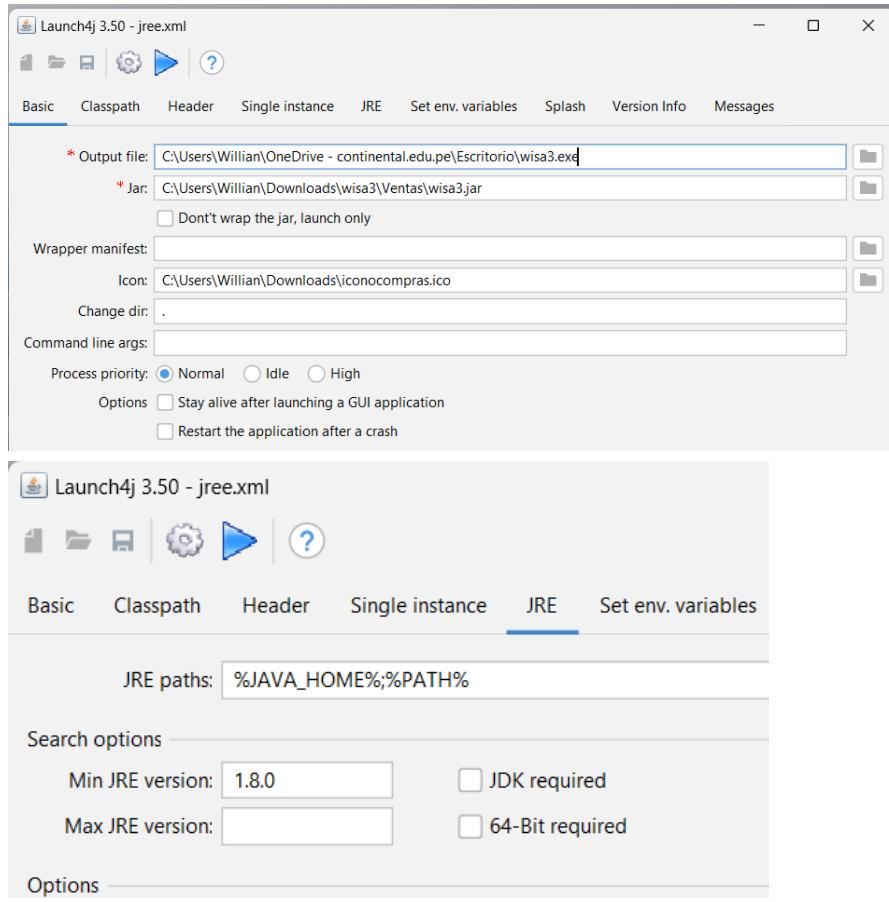
<jar destfile="${store.dir}/temp_final.jar" filesetmanifest="skip">
<zipgroupfileset dir="dist" includes="*.jar"/>
<zipgroupfileset dir="dist/lib" includes="*.jar"/>

<manifest>
<attribute name="Main-Class" value="${main.class}"/>
</manifest>
</jar>

<zip destfile="${store.jar}">
<zipfileset src="${store.dir}/temp_final.jar"
excludes="META-INF/*.SF, META-INF/*.DSA, META-INF/*.RSA"/>
</zip>

<delete file="${store.dir}/temp_final.jar"/>
<delete dir="${store.dir}/lib"/>
<delete file="${store.dir}/README.TXT"/>
</target>
</project>
```

Launch4j, creación para el archivo exe



## Creación de setup

Inno Setup Script Wizard

**Application Information**  
Please specify some basic information about your application.

**Application name:**  
Wisa

**Application version:**  
1.0

**Application publisher:**  
Estudiante Universiad Continental JKA

**Application website:**  
<https://www.example.com/>

**bold = required** Back Next Cancel

Inno Setup Script Wizard

**Application Folder**  
Please specify folder information about your application.

**Application destination base folder:**  
Program Files folder

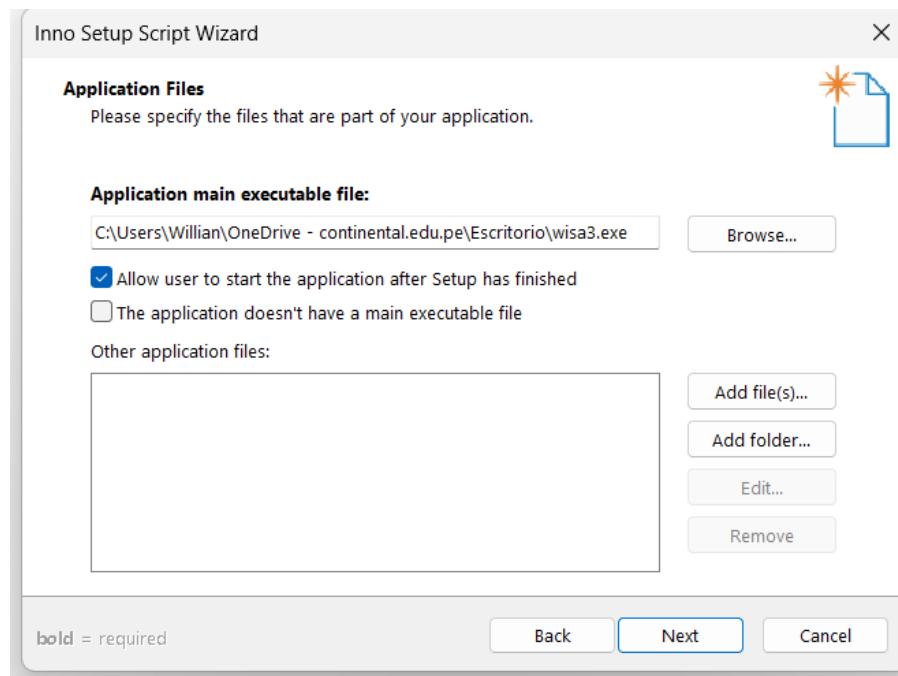
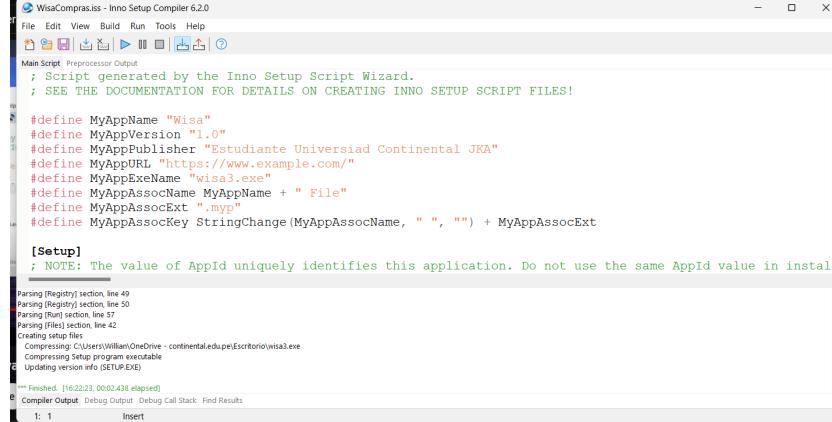
**Application folder name:**  
Wisa

Allow user to change the application folder  
 Other:  
 The application doesn't need a folder

**bold = required** Back Next Cancel

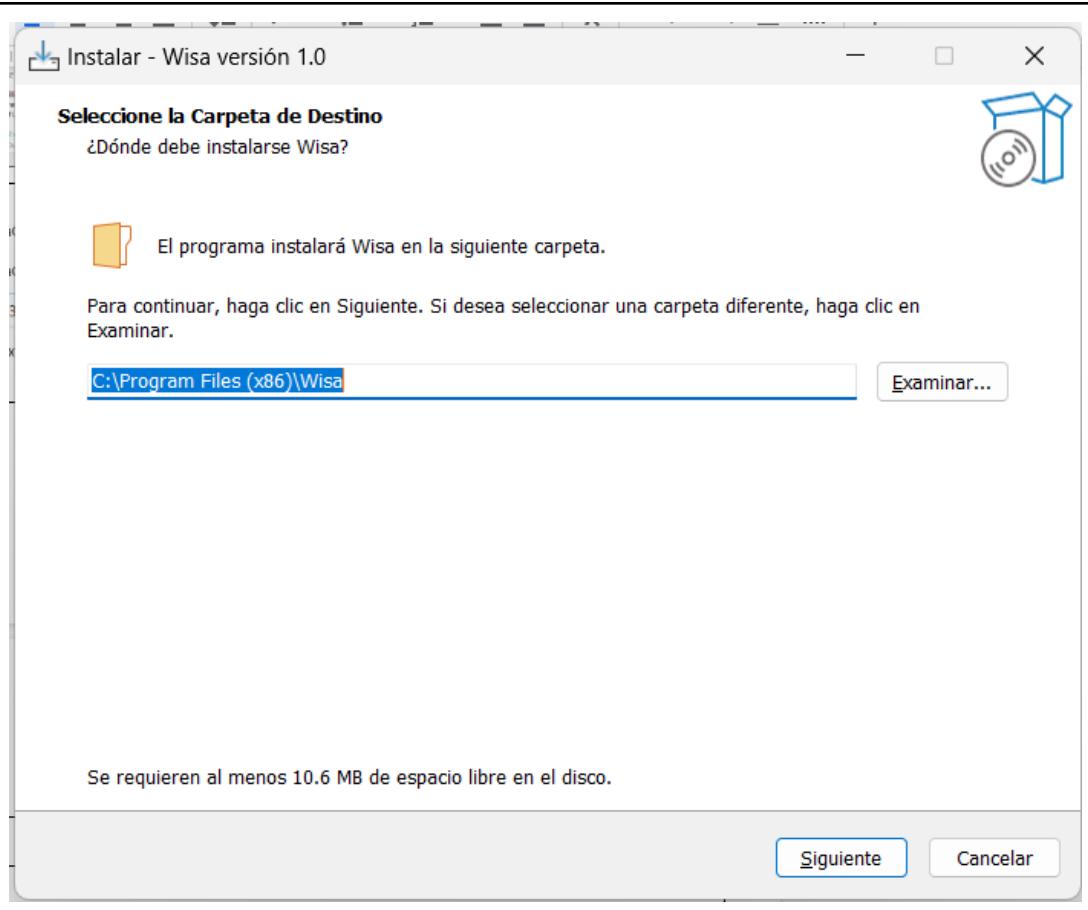
**Archivos del instalador**

Nombre	Fecha de modificación	Tipo
322.iss	20/02/2023 16:08	Inno Setup Script
Wisa	08/02/2023 16:27	Acceso directo
wisa_nube.exe	20/02/2023 16:04	Aplicación
Wisa3.0	13/02/2023 5:11	Acceso directo
wisaNube	20/02/2023 16:08	Acceso directo
wisaNube.xml	20/02/2023 16:04	Archivo XML
wisanubesetup.exe	20/02/2023 16:08	Aplicación

## Instalación del Software

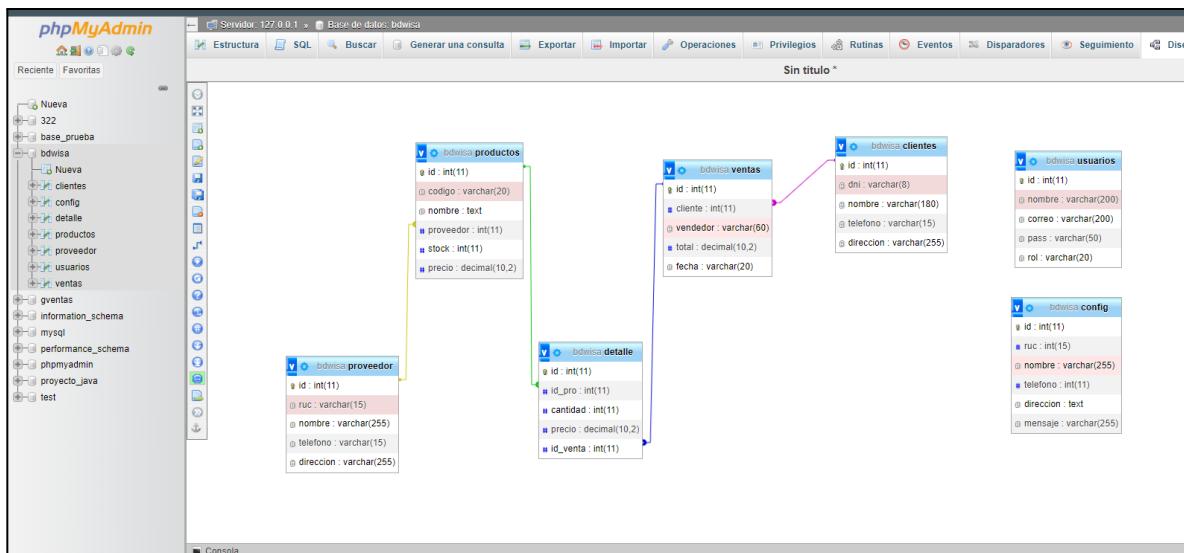
## Instalar software



## Acceso Directo del Software



## Base de datos



phpMyAdmin

Servidor: 127.0.0.1 > Base de datos: bdwisa

Estructura SQL Buscar Generar una consulta Exportar Importar Operaciones Privilegios Rutas Eventos

Filtros

Que contengan la palabra:

Tabla	Acción	Filas	Tipo	Cotejamiento	Tamaño	Residuo a depurar
clientes	Examinar Estructura Buscar Insertar Vaciar Eliminar	1	InnoDB	utf8_spanish_ci	16.0 KB	-
config	Examinar Estructura Buscar Insertar Vaciar Eliminar	1	InnoDB	utf8_spanish_ci	16.0 KB	-
detalle	Examinar Estructura Buscar Insertar Vaciar Eliminar	4	InnoDB	utf8_spanish_ci	48.0 KB	-
productos	Examinar Estructura Buscar Insertar Vaciar Eliminar	2	InnoDB	utf8_spanish_ci	32.0 KB	-
proveedor	Examinar Estructura Buscar Insertar Vaciar Eliminar	2	InnoDB	utf8_spanish_ci	16.0 KB	-
usuarios	Examinar Estructura Buscar Insertar Vaciar Eliminar	1	InnoDB	utf8_spanish_ci	16.0 KB	-
ventas	Examinar Estructura Buscar Insertar Vaciar Eliminar	3	InnoDB	utf8_spanish_ci	32.0 KB	-
7 tablas Número de filas		14	InnoDB	utf8mb4_general_ci	176.0 KB	0 B

## Interfaces del software

Descripción	Imagen
Inicio Sesión	 

## Nueva Venta

**wisa**  
A cualquier hora del día...

**TIENDAS wisa FRANQUICIAS**

Código	Descripción	Cant	Precio	Seleccionar:
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	25 ene 2023 <input type="button" value=""/>

**Stock Disponible**

ID	DESCRIPCIÓN	CANTIDAD	PRECIO U.	PRECIO TOTAL

Dni/Ruc  Nombre:  Imprimir  Total a Pagar: ----

## Registro de Nuevo Cliente

**wisa**  
A cualquier hora del día...

**TIENDAS wisa FRANQUICIAS**

**REGISTRO DE CLIENTE**

ID	DNI/RUC	NOMBRE	TELÉFONO	DIRECCIÓN
1	74361715	Ponnygon	999888777	av.san carlos

Dni/Ruc:   
 Nombre:   
 Teléfono:   
 Dirección:

## Nuevo Proveedor

**wisa**  
A cualquier hora del día...

**TIENDAS wisa FRANQUICIAS**

**NUEVO PROVEEDOR**

ID	RUC	NOMBRE	TELÉFONO	DIRECCIÓN
1	998787	Open Services	798978879	Lima - Perú
2	515228	Plaza Norte	789456123	av.San Martín

Ruc:   
 Nombre:   
 Teléfono:   
 Dirección:

## Nuevo Producto

The screenshot shows the WISA POS interface. On the left, there's a vertical menu bar with icons for Nueva Venta, Clientes, Proveedor, Productos, Ventas, Config, and Usuarios. The main area has the WISA logo at the top with the slogan "A cualquier hora del día...". Below it is a yellow panel titled "NUEVO PRODUCTO" containing fields for Código, Descripción, Cantidad, Precio, and Proveedor (with a dropdown menu showing "Open Services"). At the bottom of this panel are several small buttons. To the right is a table showing product inventory:

ID	CODIGO	DESCRIPCIÓN	PROVEEDOR	STOCK	PRECIO
4	0001	Galleta Casino	Plaza Norte	30	1.0
1	322	Cerveza	Plaza Norte	14	3.0

## Historial de Ventas

The screenshot shows the WISA POS interface. The left menu is identical to the previous screen. The main area features the WISA logo and slogan. Below is a yellow panel titled "HISTORIAL DE VENTAS" which contains a table of sales data. A "PDF" button is located in the top right corner of this panel.

ID	CLIENTE	VENDEDOR	TOTAL
1	Ponnygon	kenny	30.0
2	Ponnygon	kenny	60.0
9	Ponnygon	kenny	48.0

## Crear nuevo usuario

The screenshot shows the WISA POS interface. The left menu is identical. The main area features the WISA logo and slogan. It includes a red panel for creating a new user with fields for Correo Electrónico, Password, Nombre, and Rol (with a dropdown menu showing "Administrador"). Below this is a table for managing users:

Id	Nombre	Correo	Rol

## Actualización de datos de la Empresa

DATOS DE LA EMPRESA

RUC	NOMBRE
11111111	WISA
DIRECCIÓN	TELÉFONO
Huancayo - Perú	987654321
MENSAJE	
WISA	

ACTUALIZAR

PROMOS... Y MÁS PROMOS