## • Contenido Típico:

- Introducción (propósito, alcance, audiencia).

Los sistemas de cobro son herramientas fundamentales para mejorar la gestión y control dentro de empresas, tiendas y comercios. Su propósito principal de este sistema es facilitar el manejo diario de la empresa, ayudando con la organización interna, teniendo una experiencia más ágil al momento de realizar pagos.
El objetivo de este documento es describir las funciones principales del sistema y cómo se va evaluar su funcionamiento para asegurar que cumpla con las necesidades del negocio.

## • Elementos a probar/no probar.

- Pantalla de acceso al sistema (Login)
- Registro y administración de empleados
- Gestión de productos disponibles
- Control de entradas de mercancía (Compras)
- Proceso de venta al cliente
- Sección para proveedores y abastecimiento

## • Características a probar

**1. Pantalla de inicio de sesión**

- El sistema debe permitir el ingreso únicamente si los datos coinciden con los registrados. Si hay un error, se debe informar al usuario de forma clara que los datos no son correctos.

**2. Administración de empleados**

- El nombre del empleado debe contener solo letras; si se ingresan números o símbolos, el sistema debe evitarlo.

- Al escribir el correo, debe ser obligatorio incluir el símbolo “@”, y de no hacerlo, debe mostrarse un mensaje de error.

- Las acciones como añadir, editarlo o eliminar empleados, deben ser respuestas claras del sistema que indica si se completaron con éxito no.

**3. Gestión de productos disponibles**

- El debe permitir registrar nuevos productos indicando nombre, categoría, precio y cantidad en inventario.

    - Se debe validar que al editar un producto, los cambios se reflejen correctamente en el listado.

    - Debe haber función de búsqueda o filtro por nombre o categoría para facilitar la gestión.

    - Si un producto está cerca de agotarse, el sistema debe generar una alerta visual o mensaje de aviso.

**4. Control de entrada de compras**

    - Al registrar una compra, el sistema debe aumentar automáticamente, el inventario de los productos involucrados.

    - Se deben verificar que se puedan registrar detalles como proveedor, fecha, productos adquiridos y costo total.

    - El sistema debe evitar el ingreso de compras con campos vacíos o cantidades inválidas (como números negativos).

- Debe mostrar un historial de compras anteriores para consulta rápida.

5. **Proceso de venta al cliente**

- Cada vez que se registre una venta, el inventario debe actualizarse de forma automática, restando las cantidades vendidas.

- Debe validarse que los precios, descuentos y totales se calculen correctamente.

- El sistema debe generar un comprobante de venta (ticket o factura) que pueda imprimirse o enviarse por correo.

- Debe mostrar un mensaje claro cuando se intente vender un producto sin stock disponible.

6. **Sección de proveedores y abastecimiento**

- El sistema debe permitir agregar nuevos proveedores con datos como nombre, contacto y productos que suministra.

- Las funciones de edición o eliminación debe estar activas solo para usuarios autorizados.

- Al seleccionar un proveedor, debe mostrarse su historial de entregas y productos vinculados.

- Es importante validar que la relación entre proveedores y productos este bien registrada, para facilitar futuros pedidos.

- ## Estrategia de prueba

- **Tipos de Prueba**

Se aplicarán pruebas funcionales para validar que cada módulo del sistema de cobro, desarrollado en Java, funcione como se espera.

- ## Enfoque

El enfoque será por componentes: se probará cada sección del sistema de forma individual (como el login, los empleados, productos, ventas, compras y proveedores) y luego se harán pruebas integradas. Se evaluarán tanto el comportamiento de lógica implementada en Java como la interacción con la interfaz gráfica (Java Swing) y la base de datos a través de JDBC.

- ## Herramientas utilizadas

- Interfaz gráfica: (Swing)
- Base de Datos: MySQL
- Entorno de desarrollo: Netbeans

- ### Criterios de entrada y salida

**Entrada:**

El sistema de cobro debe estar compilado y ejecutándose correctamente en el entorno de Java. La conexión a la base de datos (mediante JBDC) debe estar activa y configurada. Debe existir datos de prueba (productos, usuarios, proveedores, etc.).

Las interfaces deben estar accesibles desde el menú principal (Swing).

Cada módulo debe haber sido probado al menos una vez. Los errores críticos deben haber sido solucionados.

Se deben haber documentado los resultados de cada paso de prueba.

Debe existir un informe al final con los hallazgos y recomendaciones para el sistema.

- ### Criterios de suspensión y reanudación

**Suspensión:**

Fallas graves en la interfaz (por ejemplo, que el sistema no abra por errores al conectarse a JDBC).

Ausencia de datos clave en la base.

Inestabilidad del entorno Java que impida continuar.

**Reanudación:**

Corrección del error en código Java o en la configuración de la base.

Restauración de los datos de prueba.

Verificación de que el sistema vuelve a estar operativo en el entorno.

- **Entregables de prueba**
- Documento de plan de pruebas.
- Casos de prueba (formato tabla, uno por modulo).
- Código fuente con comentarios y anotaciones de pruebas unitarias.
- Evidencias visuales de pruebas.
- Informe final con resumen de resultados, errores encontrados y sugerencias.

- **Roles y Responsabilidades**

**Diseñador:**

- Apoya con restauración o creación de datos necesarios para las pruebas.

**Desarrollador:**

- Corrige errores en el código fuente (controladores, modelos, validaciones).

**Pruebas**

- Ejecuta pruebas manuales sobre la interfaz y verifica el comportamiento del sistema.
- Lleva seguimiento del proceso y consolida los resultados.
- Administrador de base de datos.

- **Recursos (personal, hardware y software).**
- Personal: diseñador, 2 desarrolladores y un Tester (Pruebas).
- Hardware: JDK instalado en computadoras.
- Software: JDK 8 o superior, IDE para Java y gestor de base de datos (MySQL Workbench.)

❖ **Calendario de Pruebas.**

- **Riesgos y contingencias**

- Mal funcionamiento de eventos (botones, formularios).
- Errores lógicos en la base de datos (consultas mal hechas, datos incorrectos).
- Dificultad para simular ciertos flujos si no se tienen todos los datos.
- **Planes de contingencia**
- Validar la conexión con JBDC antes de iniciar las pruebas
- Usar try-catch para capturar y reportar errores en tiempo real.
- Crear copias de seguridad de la base de datos para restaurar en caso necesario.
- Contar con scripts SQL de carga de datos de prueba.