



# INGENIERÍA EN TECNOLOGÍAS DE LA INFORMACIÓN Y LA COMUNICACIÓN

## INGENIERÍA DE SOFTWARE

**GRUPO 1**

### INTEGRANTES

**Ligia Maricela Maldonado Paredes**

**Rommel Nicolás Zambrano Gaona**

**Silvia Liliana Yunga Quichimbo**

**Robert Denilson Sanguña Lanchimba**

**Saltos Cárdenas Brandon Xavier**

**Rubén Darío González Gudiño**

**Tema: CASO Práctico**

**Mgtr. Efraín Rodrigo Fonseca Carrera**

**Sexto Semestre**

**Año Lectivo 2022-2023**

**Plantee la problemática a ser abordada dando a conocer las características de su contexto y los requerimientos generales del sistema, considerando una posible evolución de estos.**

Aplicativo Sistema ventas **“Minimarket Technology”**

## **1. Introducción**

### **1.1. Propósito**

Hoy en día los negocios se reinventan para ofrecer un mejor servicio al cliente mediante el uso de la tecnología, ya que un sistema de ventas óptimo ayudaría mucho a llevar un control de precio de los productos, así como del stock y saber que producto se ha vendido más mediante un historial de ventas. En el caso de la empresa Minimarket Technology que busca agilizar los procesos de obtención de datos de los clientes y productos.

## **2. Planteamiento del Trabajo**

### **2.1. Formulación del Problema**

La problemática que este caso de estudio pretende abordar principalmente la forma en que el Minimarket Technology registra sus ventas ya que no cuenta con un control formal de las ventas, el vendedor las realiza anotando los datos del pedido en notas, donde cada hoja no posee un folio. En consecuencia, este desorden genera dificultades en la atención al cliente, ya que resulta complicado comprobar cada una de las ventas, al no existir un registro de ventas, ni del comportamiento de compra del cliente, lo cual dificulta la toma de decisiones al momento de surtir las mercaderías.

### **2.2. Justificación**

Con la implementación del sistema de ventas Minimarket Technology se podrá realizar un completo control de las ventas realizadas, el registro de los productos mediante el código, descripción, cantidad, precio y proveedor para poder llevar un control de precio como de stock y saber qué producto se ha vendido más mediante un historial de ventas de los vendedores. Resultando ser un gran avance y contrarrestando al método tradicional que lo único que hace es registrar en un cuaderno las diferentes ventas realizadas, la ganancia del día, llegando en algunos casos a generar pérdidas por información errónea.

## **Sistema de Objetivos**

### **2.3. Objetivo general**

Desarrollar e implementar un Sistema de Ventas que permita optimizar el proceso de ventas en un minimarket, facilitando el registro de productos, agilizando las consultas de la información de manera precisa y ordenada al momento en que el propietario requiera tomar decisiones.

### **2.4. Objetivos específicos**

- Conocer la arquitectura de software orientada a un prototipo de un software de un Sistema de Ventas específico para un minimarket.
- Aplicar los modelos de actualizaciones, reutilización y depuración a la estructura de software propuesta para conocer la mejora del mismo a un nivel práctico.

### 3. Alcance

El proyecto pretende cubrir el desarrollo y la implementación de un Sistema de Ventas Minimarket Technology el cual facilite el control de las ventas, los datos de los clientes, proveedores y productos para lo cual se necesitará obtener la siguiente información:

El sistema desarrollado contempla el siguiente ámbito de acción:

- Se encargará de registrar de manera digital toda la información sobre las ventas realizadas.
- Tendrá dos roles de usuario: un vendedor y un administrador.
- Permitirá al usuario utilizar sus funcionalidades directamente en la aplicación.

El usuario podrá realizar las siguientes actividades en la aplicación de sistema de Ventas Minimarket Technology

- Visualizar Clientes
- Agregar/Modificar/Eliminar Clientes
- Visualizar Proveedores
- Agregar/Modificar/Eliminar Proveedores
- Visualizar Productos
- Agregar/Modificar/Eliminar Productos
- Visualizar Ventas
- Ingresar Ventas y sus respectivos Detalles
- Visualizar y Registrar los pagos asociados a la Venta
- Configurar/editar/actualizar los datos de la empresa

### 4. Requerimientos generales del sistema

#### 4.1. Requisitos Funcionales

El Sistema ventas “Minimarket Technology” permitirá la gestión con el apoyo de los módulos del sistema informático el que se sustenta el proyecto de desarrollo de software.

*Tabla 1. Ingresar al sistema*

REQ001	Ingresar al Sistema
Descripción	Antes de poder ingresar al sistema se necesitará que se hayan registrado previamente, sí ya se cuenta con un usuario y contraseña entonces se podrá ingresar al sistema.
Actor	Usuarios (Administrador – Vendedor)
Prioridad	Alta
Comentarios	

*Tabla 2. Registrar Usuarios*

REQ002	Registrar Usuarios
Descripción	Los usuarios deberán registrar los datos como (correo electrónico, contraseña, nombre y rol) para poder acceder al sistema.
Actor	Usuarios
Prioridad	Alta
Comentarios	

*Tabla 3. Ingresar venta*

REQ003	Ingresar venta
Descripción	Se ingresan los datos de la venta (cliente, productos, precios, cantidades)
Actor	Usuarios
Prioridad	Alta
Comentarios	

*Tabla 4. Agregar cliente*

REQ004	Agregar cliente
Descripción	Permite agregar un cliente, ingresando sus datos (ruc, nombre cliente, teléfono, dirección) también se puede eliminar o modificar los datos del cliente existente en el listado de clientes.
Actor	Usuarios
Prioridad	Alta
Comentarios	

*Tabla 5. Registrar Proveedores*

REQ005	Registrar Proveedores
Descripción	El administrador tiene la opción de registrar proveedores donde estos se registrarán en la BD y se ingresarán los datos del proveedor (ruc, nombre cliente, teléfono, dirección) también se puede eliminar o modificar los datos del proveedor existente en el listado de los proveedores.
Actor	Administrador
Prioridad	Alta
Comentarios	

*Tabla 6. Registrar Productos*

REQ006	Registrar Productos
Descripción	Los usuarios podrán registrar los productos mediante el Código, descripción, cantidad, precio y proveedor para poder llevar un control de precio como de stock.
Actor	Usuarios
Prioridad	Alta
Comentarios	

*Tabla 7 . Generar historial de ventas*

REQ007	Generar Historial de Ventas
Descripción	Muestra un historial de las ventas realizadas por los usuarios durante el día, semanal, mensual, o anual en un archivo PDF.
Actor	Administrador
Prioridad	Alta

*Tabla 8. Configurar datos de la empresa*

REQ008	Configurar Datos de la empresa
Descripción	Las configuraciones ayudarán a modificar o actualizar los datos de la empresa que este haciendo uso del sistema de ventas “Minimarket Technology”.
Actor	Administrador
Prioridad	Alta
Comentarios	

*Tabla 9. Generar listado de usuarios*

REQ009	Generar listado de usuarios
Descripción	Se despliega una lista con todos los usuarios registrados en el sistema mediante su ID.
Actor	Administrador
Prioridad	Alta
Comentarios	

### Requisitos No Funcionales

Para el Sistema ventas “Minimarket Technology” se ha identificado los siguientes requerimientos no funcionales en relación con la tecnología de la información.

- **USABILIDAD-ENTENDIMIENTO.** Que los tiempos de adaptación para el uso del sistema por usuario no superen las 4 horas de entrenamiento.
- **EFICIENCIA-TIEMPO DE RESPUESTA.** El sistema debe de garantizar que su tiempo de respuestas de las consultas a la base de datos nunca supere los 30 segundos.
- **MANTENIBILIDAD-ESTABILIDAD.** Ante futuras modificaciones del sistema, el producto no pierda funcionalidades antiguas.
- **FUNCIONALIDAD-SEGURIDAD.** El sistema debe asegurar que cada usuario solo tenga acceso a sus funciones dadas, no a opciones que no le corresponden. Además de ello debe mantener un control a través de email y contraseña.
- **PORTABILIDAD-ADAPTIBILIDAD.** El sistema debe funcionar bajo distintos navegadores, sistemas operativos y dispositivos portátiles.

### 2. Diseñe el software con las interfaces requeridas basados en el paradigma de reutilización.

Para esta actividad se aplicó la reutilización con actividades hechas por compañeros en semestres anteriores en donde se implementó un sistema parecido al que se seleccionó actualmente para el ingreso de pedidos de un restaurante, algunas secciones de código fueron reutilizadas y se diseñaron otras nuevas, según los requerimientos solicitados.

## Entorno de desarrollo:

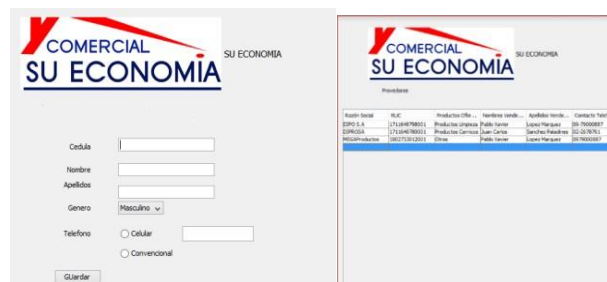
- Java 15.0
- Apache NetBeans IDE 11.3
- PHPMyAdmin – MySQL

A continuación, se presentan algunas capturas de las interfaces que se mejoraron o se implementaron siendo completamente nuevas:

Se diseñó una interfaz de Login o **INICIAR SESIÓN** para la conexión con una base de datos con usuarios autorizados correspondiente al sistema.



En el apartado de registro de **CLIENTE** se reutilizó parte del código para la interfaz.



A esta interfaz se realizaron mejoras de diseño agregando íconos y botones de acciones, específicamente un botón para vaciar los campos de texto en caso de que se haya cometido un error y vaciar todos los campos para añadir un nuevo cliente.



En el apartado **VENTAS** se visualiza las ventas creadas a partir de los clientes y vendedores respectivos asignados en la creación de una nueva venta, aquí se reutilizó parte de código e interfaz vista anteriormente. Así la visualización del historial de ventas se ve primero de cierta forma y luego con algunas mejoras, correcciones y adaptaciones a nuestro respectivo proyecto se ve así:

	nombre	cantidad_venta	precio_unitario
	Shampoo Pant...	10.00	24.00
	Shampoo Pant...	10.00	24.00



ID	CLIENTE	VENDEDOR	TOTAL
4	Carla Iana	Angel Sitientes	19000.0
5	Iana	Robert Sangulla	25.0
6	Iana	Robert Sangulla	12.5
7	Iana	Robert Sangulla	241.5

Se agregaron características de diseño de interfaz, para visualizar más datos necesarios, el botón para actualizar los datos enviados y se guardan en la base de datos. Y todo esto se implementa en el programa del sistema de ventas para que funcione correctamente.

**DATOS DE LA EMPRESA**

RUC: 1728575482485 | NOMBRE: Megatrónica

DIRECCIÓN: Ecuador | TELÉFONO: 975614855

MENSAJE: Todo en componentes

**ACTUALIZAR**

Se añadieron botones de fecha y su campo de visualización que muestra la fecha actual, también el reporte de ventas, las cuales nos muestra las ventas realizadas en ese día, el botón para borrar una venta que se haya realizado y finalmente iconos para interactuar con el usuario y el apartado del total que debe pagar por la compra realizada.

**registro de ventas**

Código: | Descripción: | Cant: | Precio: | Seleccionar: 2 jul 2022

Stock Disponible: | fecha y calendario: 2022

ID	DESCRIPCIÓN	CANTIDAD	PRECIO U.	PRECIO TOTAL
040	impresora	5	80.5	402.5
002	teclados	7	4.5	31.5
001	audífonos	4	2.5	10.0

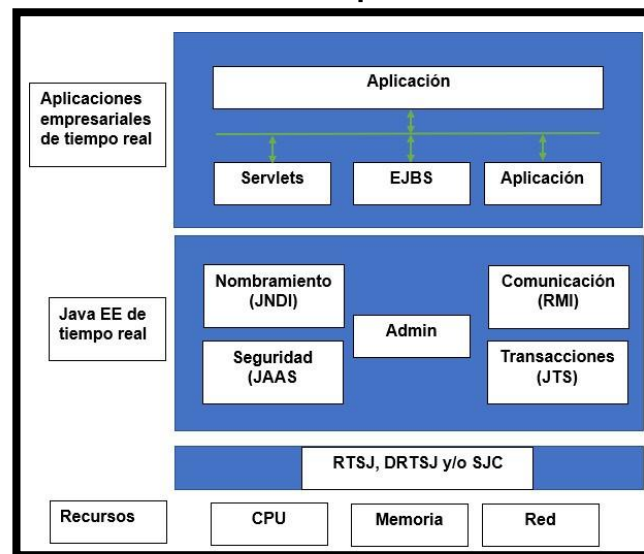
**borrar venta**

Ruc: 1728575474 | Nombre: Ana Reyes

**Total a Pagar: 444.00**

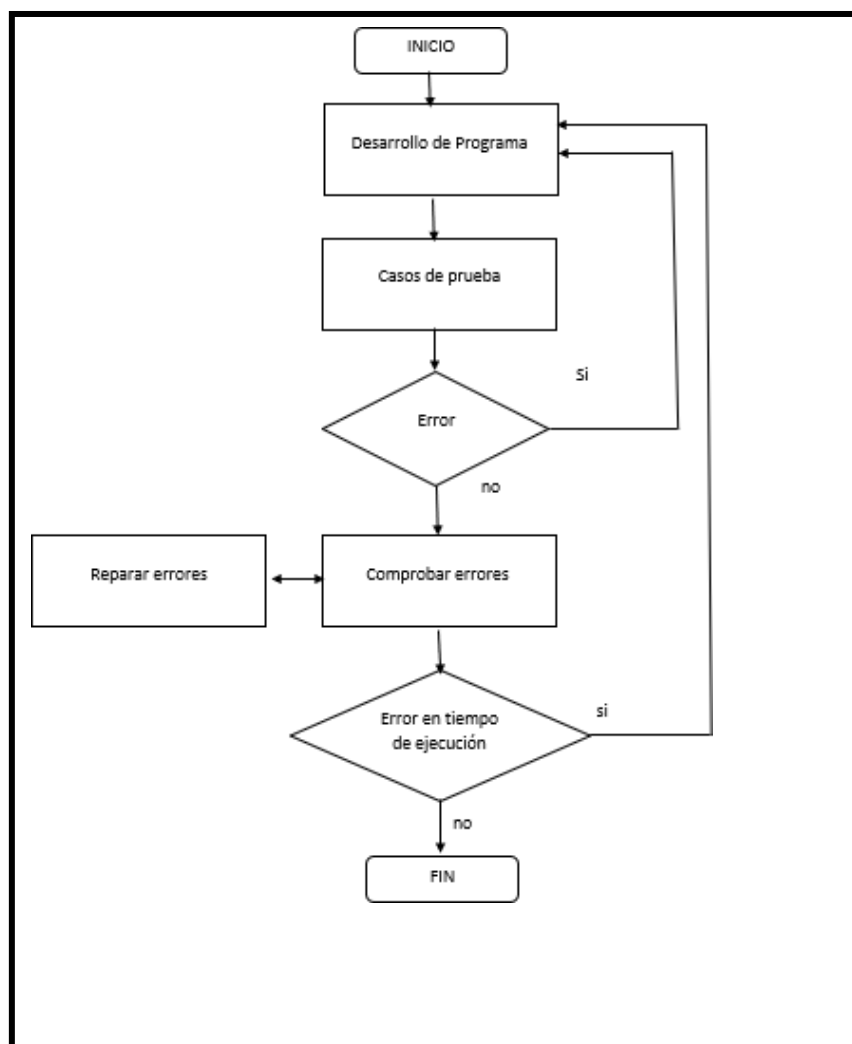
### 3. Depure la arquitectura del sistema propuesta.

#### Arquitectura



Nota: Arquitectura del sistema de ventas Minimarket Technology.

#### Proceso de depuración del sistema



Nota: Proceso de depuración del sistema



#### 4. Proponga actualizaciones en el software y realice el mantenimiento inherente a las mismas.

Nuestro proyecto está orientado a un sistema de ventas para un Minimarket Technology lo cual conlleva que los datos tanto de los clientes, proveedores, administradores y vendedores sean protegidos cuidadosamente. Todos esos datos se almacenarán en el sistema gestor de base de datos MySQL. Una vez que el software entró en funcionamiento y es revisado por algunos días su correcto funcionamiento, nos planteamos una serie de actualizaciones y mantenimiento con el fin de corregir alguna falla o error que se presenten en el software.

Actualizaciones del software son:

ACTUALIZACIONES	MEJORAS	STATUS
Optimizar el funcionamiento del registro de usuario y rol.	Permitir al sistema saber si el usuario que ingresa es el administrador o un asistente para brindarle ciertos privilegios.	Terminado
Complementar el módulo de producto de apartado stock para saber su disponibilidad.	Actualizar el stock de productos dependiendo de la cantidad de compras que se realiza.	Terminado
Diseñar un reporte de venta de ingresos económicos.	Crear un reporte de ventas del día para saber los productos más y menos vendidos y tener un control de estos mismos	Terminado
Implementar restricciones de seguridad para salvaguardar la información de los clientes, proveedores y usuarios	Crear seguridades y restricciones para que los usuarios que no estén autorizados para ingresar al sistema no puedan acceder a la información.	Terminado
Notificar el ingreso incorrecto de la información en el sistema.	Mediante ventanas de alerta indicar al usuario que ha ingresado mal la información en los campos requeridos.	Terminado
Implementar un método de pago electrónico.	Tarjeta de crédito	En proceso

#### 5. Proponga casos de prueba de caja blanca y de caja negra y realice las pruebas (automáticas o manuales).

##### REQ003 Ingresar Venta

##### 1. Código fuente

```

public class VentaDao {
    Connection con;
    Conexion cn = new Conexion();
    PreparedStatement ps;
    ResultSet rs;
    int x;

    public int IdVenta() {
        int id = 0;
        String sql = "SELECT MAX(id) FROM ventas";
        try {
            con = cn.getConnection();
            ps = con.prepareStatement(sql);
            rs = ps.executeQuery();
            if (rs.next()) {
                id = rs.getInt(1);
            }
        } catch (SQLException e) {
            System.out.println(e.toString());
        }
        return id;
    }
}

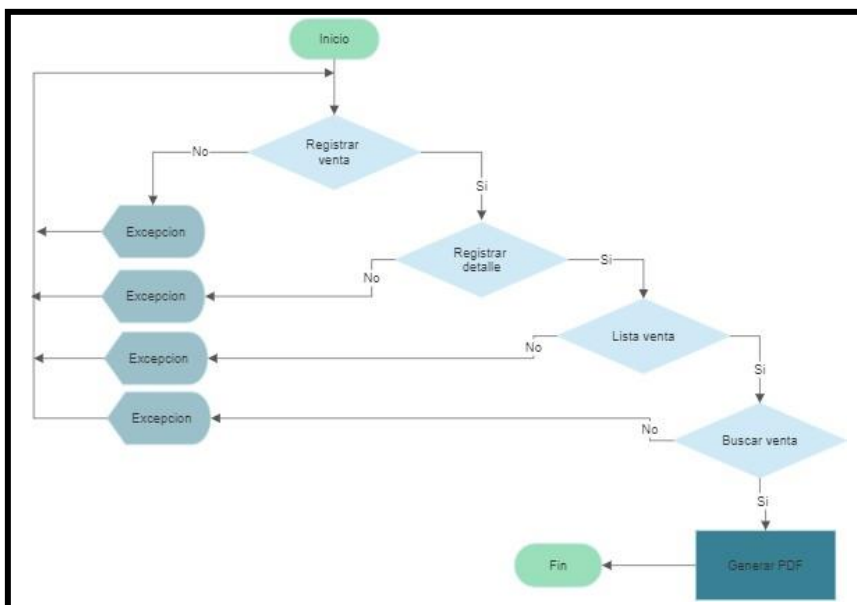
```

```

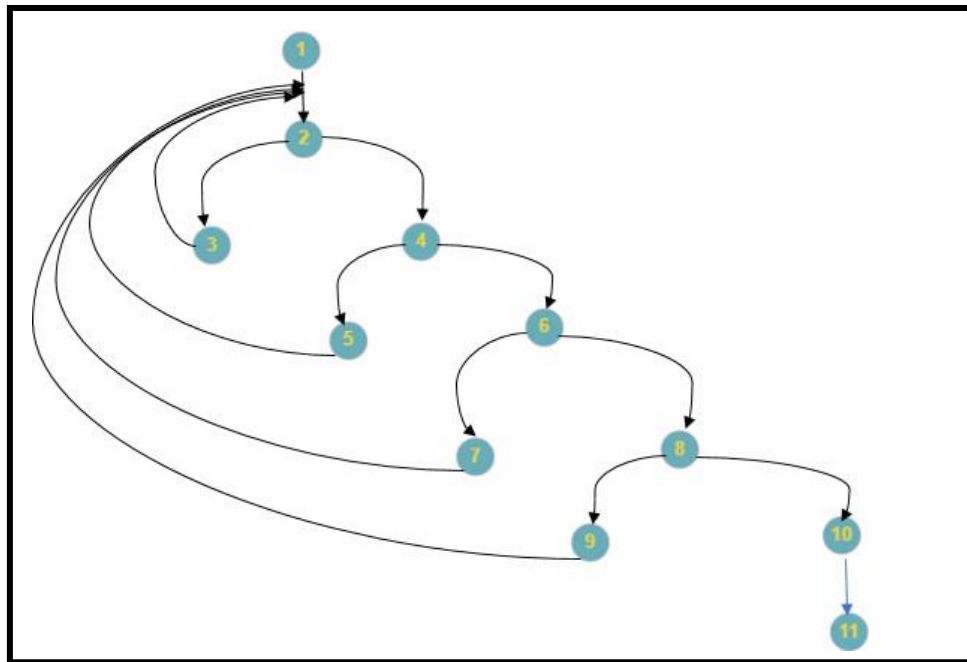
public int RegistrarVenta(Venta v) {
    String sql = "INSERT INTO ventas (cliente, vendedor, total, fecha) VALUES (?, ?, ?, ?)";
    try {
        con = cn.getConnection();
        ps = con.prepareStatement(sql);
        ps.setInt(1, v.getCliente());
        ps.setString(2, v.getVendedor());
        ps.setDouble(3, v.getTotal());
        ps.setString(4, v.getFecha());
        ps.execute();
    } catch (SQLException e) {
        System.out.println(e.toString());
    } finally {
        try {
            con.close();
        } catch (SQLException e) {
            System.out.println(e.toString());
        }
    }
    return x;
}

```

## 2. Diagrama de flujo



## 3. Grafo



#### 4. Rutas

**R1:** 1, 2, 4, 6, 8, 10, 11

**R2:** 1, 2, 3, 4, 6, 8, 10, 11

**R3:** 1, 2, 4, 5, 2, 4, 6, 8, 10, 11

**R4:** 1, 2, 4, 6, 7, 2, 4, 6, 8, 10, 11

**R5:** 1, 2, 4, 6, 8, 9, 2, 4, 6, 8, 10, 11

#### 5. Complejidad ciclomática

Se puede calcular de las siguientes formas:

- $V(G) = \text{número de nodos predicados(decisiones)} + 1$   
 $V(G) = 4 + 1 = 5$
- $V(G) = A - N + 2$   
 $V(G) = 14 - 11 + 2$   
 $= 5$

DONDE:

**P:** Número de nodos predicado = 5

**A:** Número de aristas = 14

**N:** Número de nodos = 11

#### REQ005 Registrar Proveedor

##### 1. Código fuente

```

package Modelo;

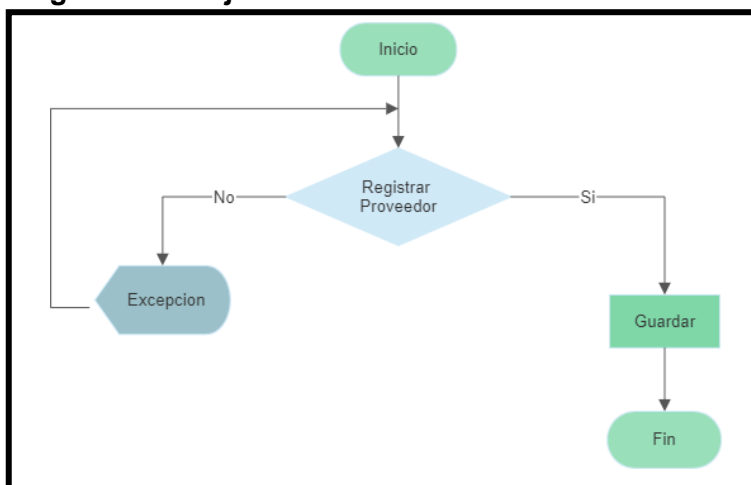
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.ArrayList;
import java.util.List;

public class ProveedorDao {
    Connection con;
    Conexion cn = new Conexion();
    PreparedStatement ps;
    ResultSet rs;

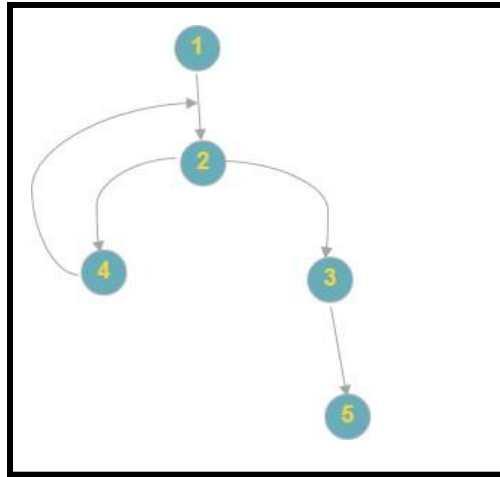
    public boolean RegistrarProveedor(Proveedor pr){
        String sql = "INSERT INTO proveedor(ruc, nombre, telefono, direccion) VALUES (?, ?, ?, ?)";
        try {
            con = cn.getConnection();
            ps = con.prepareStatement(sql);
            ps.setString(1, pr.getRuc());
            ps.setString(2, pr.getNombre());
            ps.setString(3, pr.getTelefono());
            ps.setString(4, pr.getDireccion());
            ps.execute();
            return true;
        } catch (SQLException e) {
            System.out.println(e.toString());
            return false;
        } finally{
            try {
                con.close();
            } catch (SQLException e) {
                System.out.println(e.toString());
            }
        }
    }
}

```

## 2. Diagrama de flujo



## 3. Grafo



#### 4. Rutas

**R1:** 1, 2, 3, 5

**R2:** 1, 2, 4, 2, 3, 5

#### 5. Complejidad ciclomática

Se puede calcular de las siguientes formas:

$V(G) = \text{número de nodos predichados(decisiones)} + 1$   
 $V(G) = 1 + 1 = 2$

$V(G) = A - N + 2$   
 $V(G) = 5 - 5 + 2 = 2$

DONDE:

**P:** Número de nodos predichado = 2

**A:** Número de aristas = 5

**N:** Número de nodos = 5

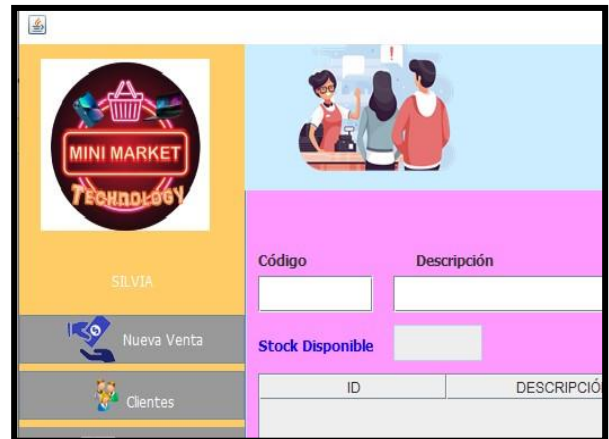
#### Pruebas de Caja negra

**REQ001 :Ingresar al sistema**

1. VARIABLE	CLASE DE EQUIVALENCIA	ESTADO	REPRESENTANTE
<b>Usuario</b>	EC1: this.correro == string	Válido	0104999016001
	EC2: this.correro!= string	No Válido	ERROR, ingrese solo números
<b>Contraseña</b>	EC1: this.pass ==String	Válido	Silvia Yunga
	EC2: this.pass!= String;	No Válido	ERROR, ingrese solo caracteres
<b>VALIDACION</b>	this.ruc =  correo;  this.nombre =  pass;	VALIDO	<a href="mailto:slyunga@gmail.co">slyunga@gmail.co</a>  <a href="#">m</a> &&admin

<b>VALIDACION</b>	this.ruc! = correo;	<b>NO VALIDO</b>	<a href="mailto:slyunga@gmail.com">slyunga@gmail.com</a> &&
	this.nombre = pass;		1234

### Ingreso correcto de datos



### Ingreso incorrecto de datos



### REQ004 Agregar cliente

#### Partición de clases equivalentes

VARIABLE	CLASE DE EQUIVALENCIA	ESTADO	REPRESENTANTE
RUC	EC1: this.ruc = ruc;	Válido	Ingreso de RUC
	EC2: this.ruc != ruc;	No Válido	ERROR, ingrese solo números
Nombre	EC1 this.nombre = nombre	Válido	Ingreso de Nombre
	EC2: this.nombre! = nombre	No Válido	ERROR, ingrese la cedula del usuario
Direccion	EC1: this.direccion = direccion;	Válido	Nombre del Usuario

	EC2: this.direccion! = direccion;	No Válido	Ingrese una dirección correcta
<b>Telefono</b>	EC1: this.telefono = telefono;	Válido	Registro de contacto
	EC2: this.telefono! = telefono	No válido	ERROR, ingrese un numero valido
<b>VALIDACION</b>	this.ruc = ruc;	VALIDO	1754895632001 &&
	this.nombre = nombre;		Rommel &&
	this.telefono = telefono;		0998027880 &&
	this.direccion = direccion;		Monjas
<b>VALIDACION</b>	this.ruc = ruc;	NO VALIDO	Hola &&
	this.nombre = nombre;		3456789 &&
	this.telefono = telefono;		Avenue12
	this.direccion = direccion;		

### Ingreso correcto de datos

The screenshot displays the 'PUNTO DE VENTA' application window. On the left is a sidebar menu with options: Nueva Venta, Clientes, Proveedor, Productos, Ventas, Config, and Usuarios. The main area is divided into two sections. The left section, titled 'Registro Cliente', contains input fields for 'Dni/ Ruc:', 'Nombre:', 'Telefono:', and 'Dirección:', each with a corresponding value entered. The right section is a table with columns: ID, DNI/RUC, NOMBRE, TELÉFONO, and DIRECCIÓN. The table contains six rows of data, with the last row highlighted in blue.

ID	DNI/RUC	NOMBRE	TELÉFONO	DIRECCIÓN
1	1234598155004	Carla luna	1785239248	Quito- Ecuador
3	1728575474	ana	0986423574	los postes
4	0104999016	Silvia Yunga	0979086011	Cuenca
5		123456789		
6	1754895632001	Rommel	0998027880	Monjas

### Ingreso incorrecto de datos, los campos están vacíos



## Anexo

### Link del video:

[https://drive.google.com/file/d/1ZfUCHqYMaZwTu2Zryq3rZRncKnL\\_a08Z/view?usp=sharing](https://drive.google.com/file/d/1ZfUCHqYMaZwTu2Zryq3rZRncKnL_a08Z/view?usp=sharing)

## Conclusiones

- Para realizar el desarrollo e implementación del sistema fue necesario realizar un análisis de cada una de las funcionalidades del software para determinar los posibles errores y establecer los cambios necesarios de cada módulo.
- Se rediseñó la interfaz del software aplicando el método de reutilización de código donde se implementaron mejoras en las funcionalidades del sistema a fin de hacerlo más intuitivo para los usuarios, se agregaron íconos y botones de acción permitiendo el ahorro de tiempo en la búsqueda de los procesos.
- Se realizó pruebas de funcionalidad a fin de verificar el correcto funcionamiento del sistema y evaluar si está acorde a los requerimientos planteados. Durante la ejecución de las pruebas se midió los tiempos de respuestas en los procesos principales.
- Nos planteamos una serie de actualizaciones y mantenimiento con el objetivo de corregir fallas o errores, modificamos y eliminamos ciertos módulos dentro de la aplicación que no estaban bien fundamentados entre ellos : Optimización del funcionamiento del registro de usuario y rol, complementamos el módulo de producto el apartado stock para saber su disponibilidad, rediseño de un reporte de ventas de ingresos económicos, implementación de restricciones de seguridad para salvaguardar la información de los clientes, proveedores y usuarios.



## Bibliografía

- Capítulo tercero Planteamiento del problema. (s/f-a). Biblus.us.es. Recuperado el 2 de julio de 2022, de <https://biblus.us.es/bibing/proyectos/abreproy/11369/fichero/I+Memoria%252FCap%C3%ADtulo+tercero+--+Planteamiento+del+problema.pdf+>
- Arquitectura java aplicaciones. (s/f). Buap.mx. Recuperado el 1 de julio de 2022, de [https://www.cs.buap.mx/~andrex/descargas/Material\\_Curso\\_Algoritmos/Caracteristicas\\_y\\_LengJAVA/java\\_Alg\\_Est\\_Dat.pdf](https://www.cs.buap.mx/~andrex/descargas/Material_Curso_Algoritmos/Caracteristicas_y_LengJAVA/java_Alg_Est_Dat.pdf)
- Booch, G. (2009). "Handbook of software architecture". Recuperado el 2 de julio de 2022, de <http://www.handbookofsoftwarearchitecture.com/>.
- Sommerville, I. (2005). Ingeniería del software. Pearson educación. Recuperado el 3 de julio de 2022, de [https://www.cgn.gub.uy/innovaportal/file/83018/1/material\\_concurso\\_r14\\_cgn\\_2017.pdf](https://www.cgn.gub.uy/innovaportal/file/83018/1/material_concurso_r14_cgn_2017.pdf)
- Hernández, W. G. (2017). La depuración como competencia en la formación del profesional informático. Revista Ingeniería, Matemáticas y Ciencias de la Información, 4(7). Recuperado el 3 de julio de 2022, de <https://urepublicana.edu.co/ojs/index.php/ingenieria/article/view/317/310>