

22-9-2021

Manual Técnico

Proyecto 1



Abner Martín Noj Hernández
201801027

Índice

Descripción del Proyecto.....	2
Datos Técnicos.....	2
Flujo de trabajo	2
Descripción de clases	2

Descripción del Proyecto

Java es un lenguaje de programación, enfocado a la programación orientada a objetos, este tipo de programación permite al programador la reutilización de código. Su intención es permitir que los desarrolladores de aplicaciones escriban el programa una vez y lo ejecuten en cualquier dispositivo, lo que quiere decir que el código que es ejecutado en una plataforma no tiene que ser recompilado para correr en otra.

Se solicitó que se realizara una aplicación de punto de venta (POS) para una empresa del Sector Retail (Blue Mail), el cual permite administrar la información de la empresa, tales como: sucursales, productos, clientes, vendedores, facturas.

A continuación se describirá cada una de las funcionalidades del programa.

Datos Técnicos

Lenguaje utilizado: Java

IDE utilizado: IntelliJ IDEA

Sistema Operativo: Windows 8 (64 bits)

Flujo de trabajo

El flujo se describe de la siguiente manera:

1. Se abre el archivo .jar
2. Se ingresan los datos de acceso
3. Se realiza las acciones permitidas de acuerdo al usuario
4. Se puede generar reportes.

Descripción de clases

Clase Main:

inicia el programa

Clase Acciones:

En esta clase contiene métodos y funciones para realizar las acciones en el área de administración.

Clase Cliente:

esta clase se utiliza para hacer el objeto de cliente

Clase Compra:

esta clase se utiliza para hacer el objeto de compra

Clase Factura:

esta clase se utiliza para hacer el objeto de factura

Clase Producto:

esta clase se utiliza para hacer el objeto de Producto

Clase Sucursal:

esta clase se utiliza para hacer el objeto de sucursal

Clase Vendedor:

esta clase se utiliza para hacer el objeto de vendedor

Clase OperacionCompra:

Esta clase contiene los métodos y funciones para realizar las compras y facturas

Clase GenerarPdf:

esta clase contiene los métodos y funciones para realizar los reportes pdf

Clase GenerarGrafica:

esta clase contiene los métodos y funciones para realizar las graficas

Clase Inicio:

esta clase contiene todas las ventas y eventos que necesita el programa

inicio():

Este constructor inicializa todas las variables, además de recuperar la información serializada.

inicializar():

este método crea el tablero, el tablero de posición del jugador, además del tablero de penalizaciones, y mantiene el flujo del juego.

ventana_login():

Este método crea la ventana de login y contiene los eventos de esta ventana.

Admin()

Este método crea la ventana del administrador y todos los eventos

Vendedores()

Este método crea la ventana de vendedores

Seleccionar_compra()

Este método crea la pestaña para la realización de las compras

Ventas_facturas()

Este método crea la pestaña para la realización de la visualización de las facturas

opcionLista (String[] lista, int x, int y, int ancho, int alto, JLabel fondoLabel)

esta funcion crea un JComboBox y lo agrega a un JLabel

botonsalir(JButton boton, JFrame frame1)

este metodo crea el evento de cerrar sesión

defPestaña(JFrame frame, int x, int y, int ancho, int alto)

esta función crea las pestañas para almacenar los Jpaneles

inicializar(int x, int y, String titulo)

esta funcion crear un Frame para la ventana inicial

inicializar_extras(int x, int y, String titulo)

esta funcion es para crear los Frames de las ventanas de agregar y modificar

defPanel(JFrame frame)

esta funcion crea un panel y lo coloca en el frame

defPanel(JTabbedPane pestaña, String nombre)

esta funcion crea un panel para las pestañas y les asigna nombre a la pestaña

defEtiqueta(JPanel panel, int x, int y, int ancho, int alto, String texto, int tamaño)

este metodo es para colocar las etiquetas en el panel modificando la letra, ya sea negrita, cursiva, o subrayada

defEtiqueta(JPanel panel, int x, int y, int ancho, int alto, String texto)

este metodo es para colocar las etiquetas en el panel sin modificar la letra.

defEtiqueta(JLabel panel, int x, int y, int ancho, int alto, String texto)

este metodo crea una etiqueta para un JLabel

etiquetaFactura(JLabel panel, int x, int y, int ancho, int alto, String texto)

este metodo crea la etiqueta para las facturas

defEtiquetaEstilo(JLabel panel, int x, int y, int ancho, int alto, String texto, int n)

este metodo es crear las etiquetas con estilos en un JLabel

defEtiqueta(JFrame panel, int x, int y, int ancho, int alto, String texto)

metodo para crear una etiqueta en un JFrame

defTituloArea(JLabel panel, int x, int y, int ancho, int alto, String texto)

metodo para crear etiqueta con color

defFondo(JPanel panel, int x, int y, int ancho, int alto)

funcion para crear fondo de color y colocarlo en un panel

defAreaTexto(JPanel panel, int x, int y, int ancho, int alto)

funcion para crear un área de texto en un panel

defAreaTexto(JLabel panel, int x, int y, int ancho, int alto)

funcion para crear un área de texto en un JLabel

defBoton(JPanel panel, int x, int y, int ancho, int alto, String texto)

funcion para crear un botón en un panel

defBoton(JLabel panel, int x, int y, int ancho, int alto, String texto)

funcion para crear un botón en un JLabel

defBoton(JFrame frame, int x, int y, int ancho, int alto, String texto)

funcion para crear un botón en un frame

tablas(String[] var, JPanel panel, int x, int y, int ancho, int alto, JTable tabla)

funcion para crear un DefaultTableModel en un panel

tablasVendedores(String[] var, JPanel panel, int x, int y, int ancho, int alto, JTable tabla)

funcion para crear DefaultTableModel para el área de vendedores

tablasVenta(String[] var, JLabel panel, int x, int y, int ancho, int alto, JTable tabla)

funcion para crear DefaultTableModel para el área de Ventas

tablas(String[] var, JLabel panel, int x, int y, int ancho, int alto, JTable tabla)

funcion para crear DefaultTableModel para colocarlo en un JLabel

llenado_tablaProd(Producto[] productos, DefaultTableModel modelo_productos, JTable tabla_productos)

metodo para llenar la tablas de productos

llenado_tablaClien(Cliente[] clientes, DefaultTableModel modelo_clientes, JTable tabla_clientes)

metodo para llenar la tablas de clientes

llenado_tablaSucur(Sucursal[] sucursales, DefaultTableModel modelo_sucursales, JTable tabla_sucursales)

metodo para llenar la tabla de sucursales

llenado_tablaVendedor(Vendedor[] vendedores, DefaultTableModel modelo_vendedores, JTable tabla_vendedores)

metodo para llenar la tabla de vendedores

llenado_tablaCompra(Compra[] compras, DefaultTableModel modelo_compra, JTable tabla_compra)

metodo para llenar la tabla de compras

llenado_tablaFacturas(Factura[] facturas, DefaultTableModel modelo_factura, JTable tabla_factura)

metodo para llenar la tabla de facturas

borrrartablaCompra(JTable tabla)

metodo para borrar la tabla

crear_Producto()

ventana para crear un nuevo producto

eliminar_Producto()

ventana para eliminar un producto

actualizar_Producto()

ventana para actualizar un producto

crear_Cliente()

ventana para crear un nuevo cliente

eliminar_Cliente()

ventana para eliminar un cliente

actualizar_Cliente()

ventana para actualizar un cliente

crear_Sucursal()

ventana para crear una sucursal

eliminar_Sucursal()

ventana para eliminar una sucursal

actualizar_Sucursal()

ventana para actualizar una sucursal

crear_Vendedor()

ventana para crear un vendedor

eliminar_Vendedor()

ventana para eliminar un vendedor

actualizar_Vendedor()

ventana para actualizar un vendedor

crear_ClienteVendedor()

ventana para crear un vendedor desde el área de ventas

buscarUsuario(Vendedor user, Vendedor[] vendedores)

funcion para buscar usuarios

validacionContra(Vendedor user, Vendedor[] vendedores)

funcion para validar la contraseña del usuario

cMasiva_productos(Producto[] productos)

metodo para hacer la carga masiva de productos

cdata_productos(JsonArray arreglo, Producto[] productos)

metodo para asignar el json al arreglo

buscar_producto(int codigo, Producto[] var)

funcion para buscar un producto

agregar_producto(int codigo, String nombre, String descripcion, int cantidad, double precio, Producto[] productos)

funcion para agregar producto

eliminar_producto(int codigo, Producto[] productos)

funcion para eliminar producto

actualizar_producto(int codigo, String nombre, String descripcion, int cantidad, double precio, Producto[] productos)

funcion para actualizar un producto

cMasiva_clientes(Cliente[] clientes)

metodo para hacer la carga masiva de clientes

cdata_clientes(JsonArray arreglo, Cliente[] clientes)

metodo para asignar el json al arreglo

buscar_cliente(int codigo, Cliente[] var)

funcion para buscar un cliente

agregar_cliente(int codigo, String nombre, int nit, String correo, char genero, Cliente[] clientes)

metodo para agregar un cliente

eliminar_cliente(int codigo, Cliente[] clientes)

metodo para eliminar un cliente

actualizar_cliente(int codigo, String nombre, int nit, String correo, char genero, Cliente[] clientes)

metodo para actualizar un cliente

cMasiva_sucursales(Sucursal[] sucursales)

metodo para hacer la carga masiva de sucursales

cdata_sucursales(JsonArray arreglo, Sucursal[] sucursales)

metodo para asignar el json al arreglo

buscar_sucursal(int codigo, Sucursal[] var)

funcion para buscar una sucursal

agregar_sucursal(int codigo, String nombre, String direccion, String correo, int telefono, Sucursal[] sucursales)

metodo para agregar una nueva sucursal

eliminar_sucursal(int codigo, Sucursal[] sucursales)

metodo para eliminar una sucursal

actualizar_sucursal(int codigo, String nombre, String direccion, String correo, int telefono, Sucursal[] sucursales)

metodo para actualizar una sucursal

cMasiva_vendedores(Vendedor[] vendedor)

metodo para hacer la carga masiva de vendedores

cdata_vendedores(JsonArray arreglo, Vendedor[] vendedores)

metodo para asignar el json al arreglo

buscar_vendedor(int codigo, Vendedor[] var)

funcion para buscar a un vendedor

agregar_vendedor(int codigo, String nombre, String correo, int caja, int ventas, char genero, String psw, Vendedor[] vendedores)

metodo para crear un vendedor

eliminar_vendedor(int codigo, Vendedor[] vendedores)

metodo para eliminar un vendedor

actualizar_vendedor(int codigo, String nombre, String correo, int caja, int ventas, char genero, String psw, Vendedor[] vendedores)

metodo para actualizar un vendedor

buscar_vendedorCorreo(String correo, Vendedor[] var)

funcion para buscar un vendedor

serializar(Object[] arreglo, String ruta)

metodo para serializar los arreglos

leerArregloSerializado(String ruta)

funcion para leer los archivos serializados

Ordenamiento(Object[] var)

metodo para ordenar un objeto, dejando los null al final

leerArchivoJson()

funcion para leer un archivo json

contador_Actual(Object[] var)

funcion para actualizar el contador al iniciar

borrarTodo (Object [] arreglo, String ruta, int inicio)

metodo para hacer un reset de la informacion