UNIVERSIDAD MARIANO GALVEZ DE GUATEMALA

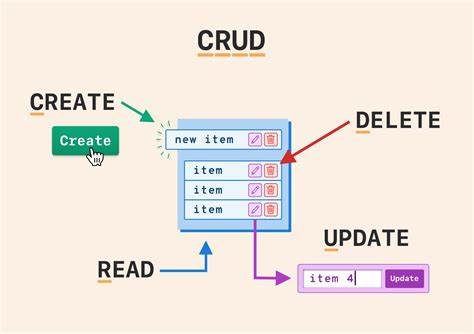
SEDE CHIMALTENANGO

ABNER JOSUE ESQUIT BARRENO

1990-17-15299

PROGRAMACION II

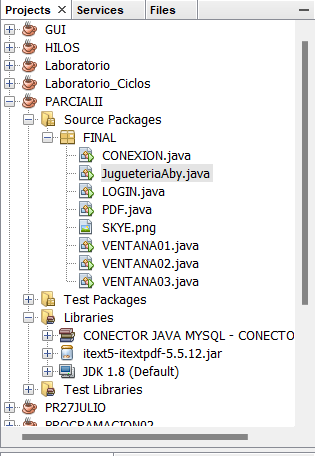
PROYECTO FINAL



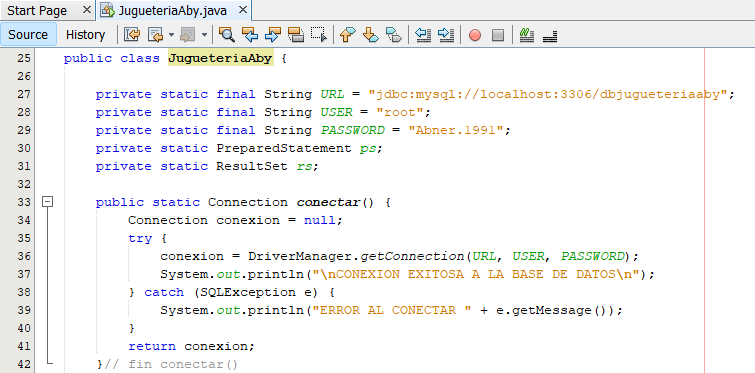
MANUAL TECNICO

PROYECTO JAVA (CONSOLA) DE CRUD CON CONEXIÓN A MYSQL WORKBENCH 8.0

Creamos una Clase en un paquete previamente utilizado en nuestro Segundo Parcial a la cual llamamos JugueteriaAby.java en la cual trabajaremos nuestro proyecto CRUD

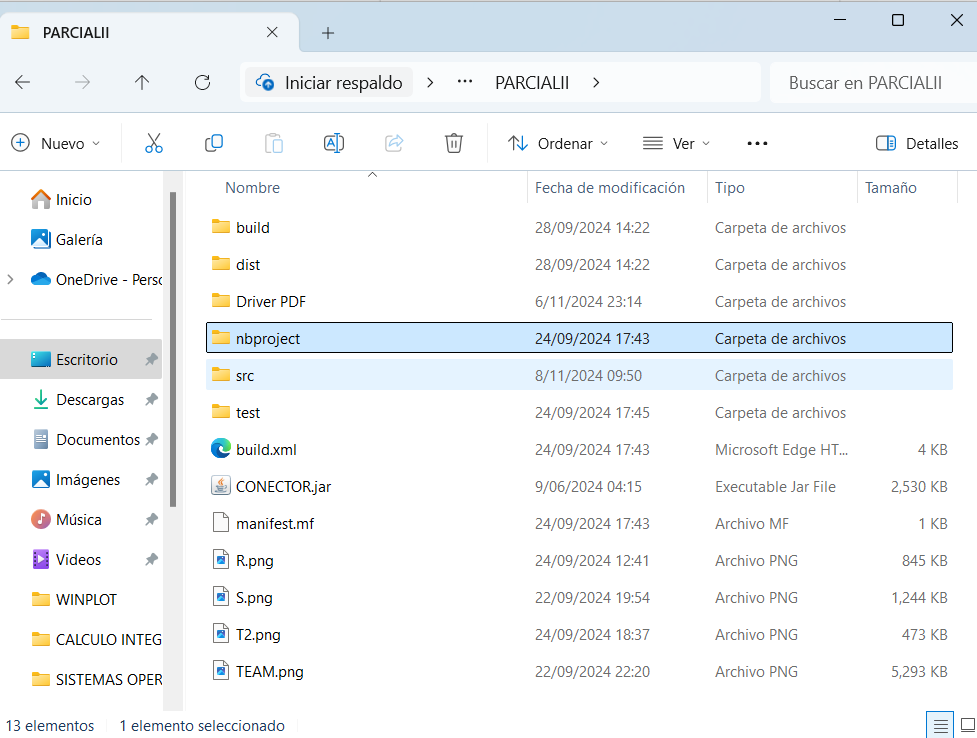


Se empieza creando nuestro método para realizar nuestra conexión a la base de datos MYSQL WORKBENCH 8.0 utilizando el siguiente código, cabe mencionar que si se desea utilizar el código fuente de nuestro proyecto debe cambiar los datos de URL USER Y PASSWORD por los que se vayan a utilizar

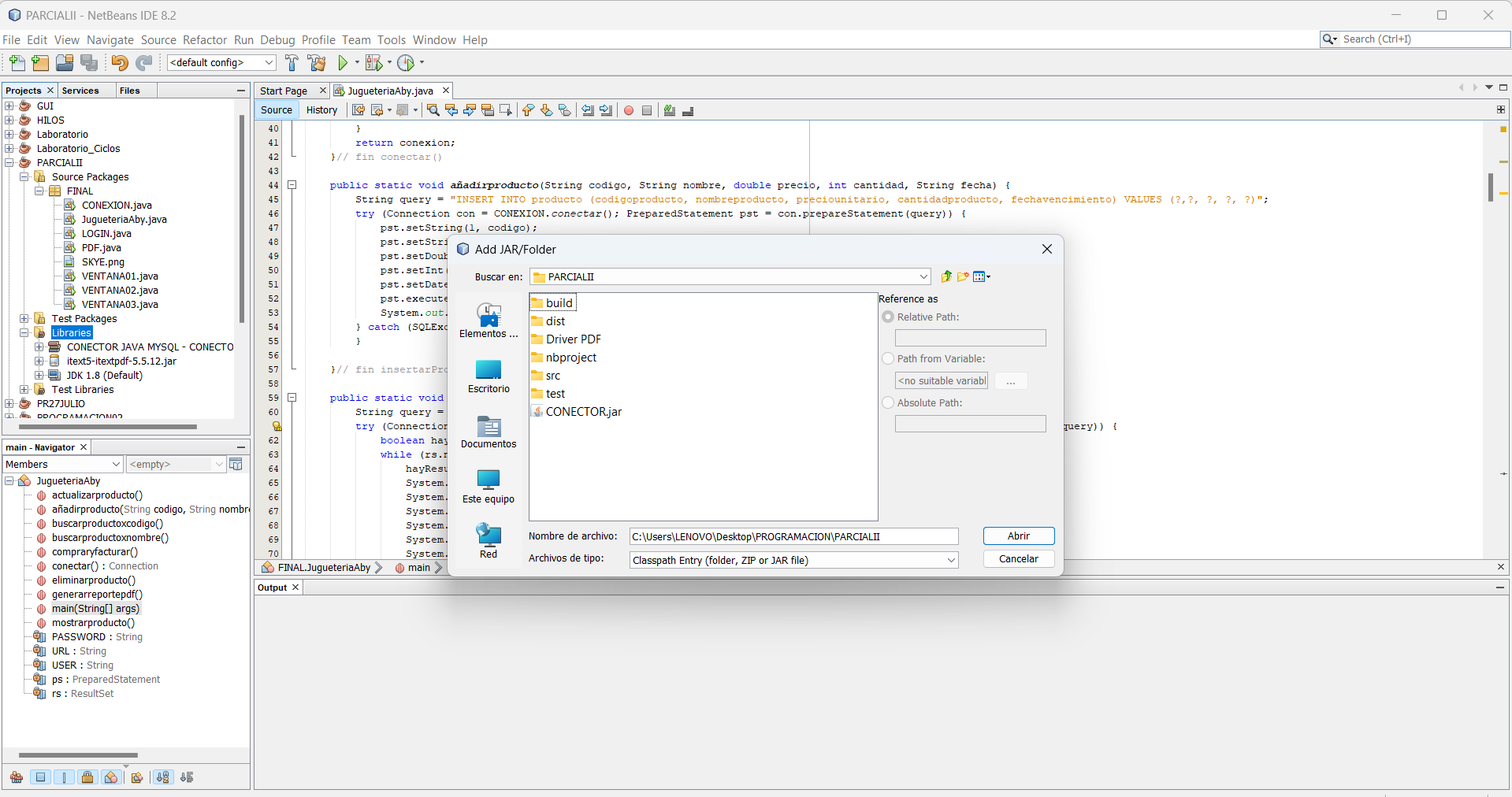


También se descarga de internet un plugin que nos ayudara a conectar nuestro proyecto a MYSQL desde la pagina oficial del fabricante como se muestra en la imagen siguiente

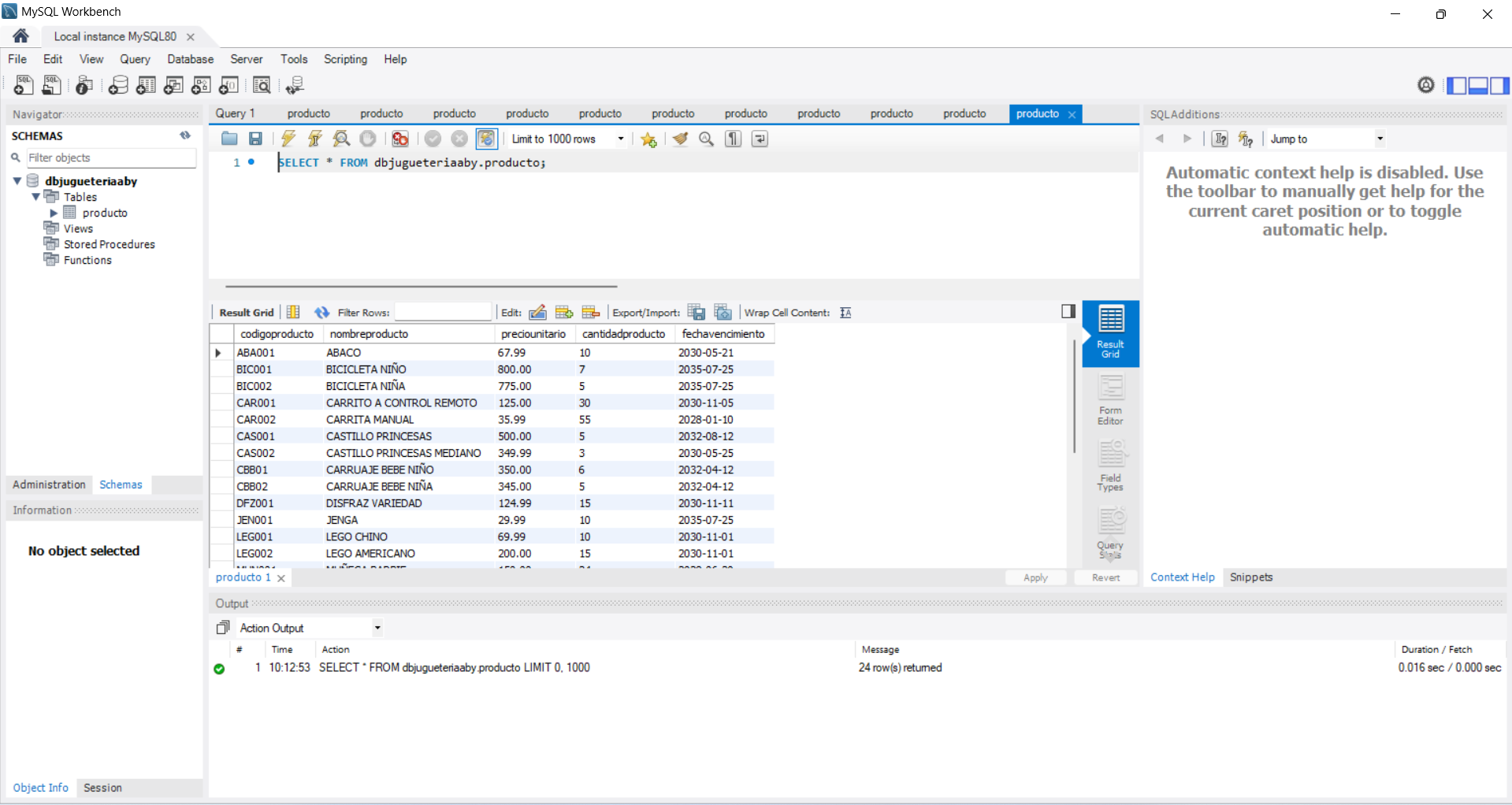


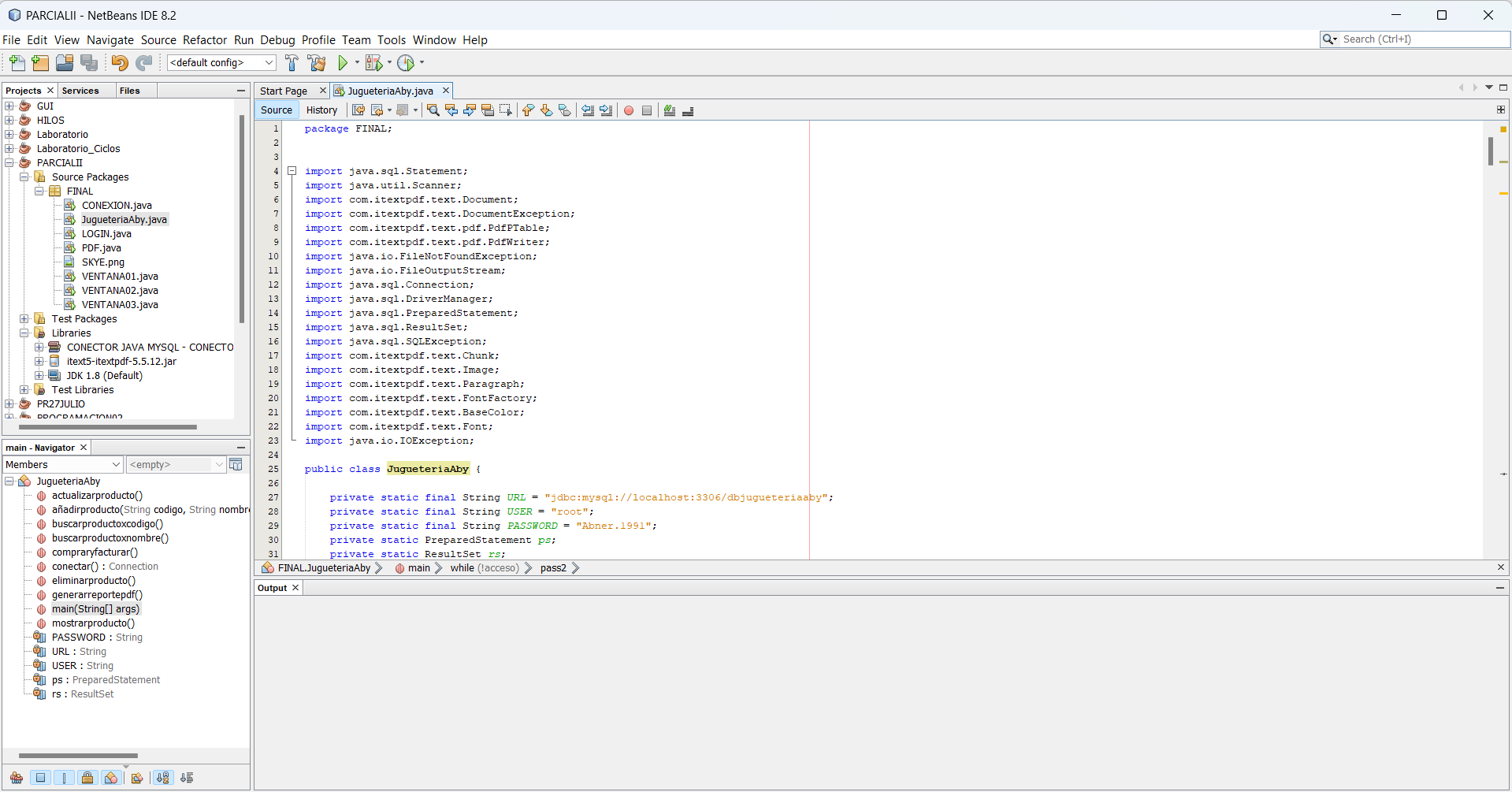
Al tener descargado el archivo procedemos a buscar la carpeta de nuestro proyecto y lo colocamos en la misma, podemos reemplazar el nombre en mi caso yo le puse CONECTOR

Al haber hecho el paso siguiente es añadirlo a nuestro IDE desde la carpeta Libraries presionas click derecho y seleccionamos la opción Add Jar/Folder y nos mostrara el explorador de archivos y ubicaremos el archivo que anteriormente copiamos que es nuestro CONECTOR lo seleccionamos damos abrir y nos aparecerá añadido en nuestras librerias



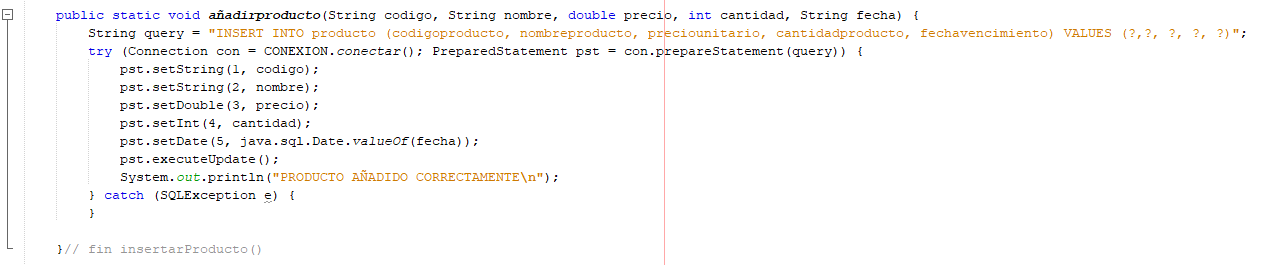
Previamente se creo nuestra base de datos en MYSQL WORKBENCH 8.0 a la cual añadimos 5 columnas con codigoproducto, nombreproducto, preciounitario, cantidadproducto y fechavencimiento, cabe mencionar que nuestra tabla se llama producto. Todo esto debe coincidir con los datos que ingresaremos en nuestros siguientes métodos que crearemos, también debemos validar la credenciales como el puerto, el nombre de la base de datos, el usuario y la contraseña de MYSQL.



Debemos tener en cuenta que cada vez que vayamos añadiendo nuestros métodos debemos importar todas la librerías correspondientes para que nuestro proyecto funcione correctamente

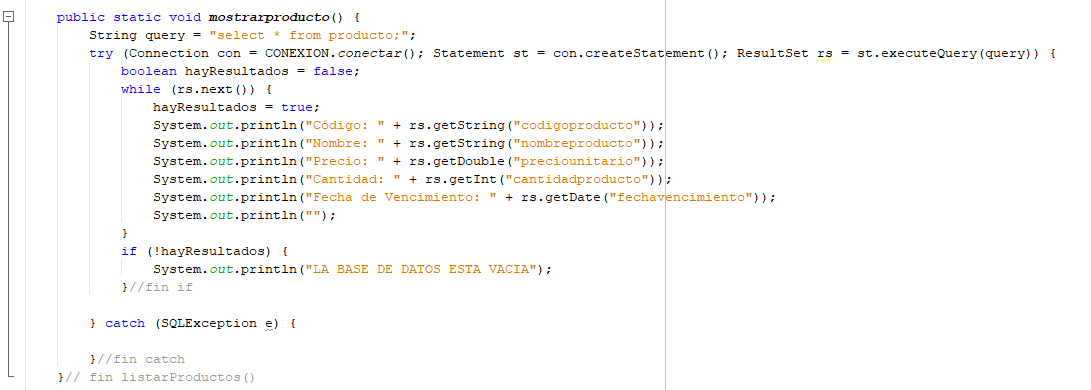
Ahora añadiremos el método para Crear un registro en la base de datos para realiza nuestro CRUD

CREATE



Se añade el método para Leer o mostrar todos nuestros registros de nuestra base de datos

READ



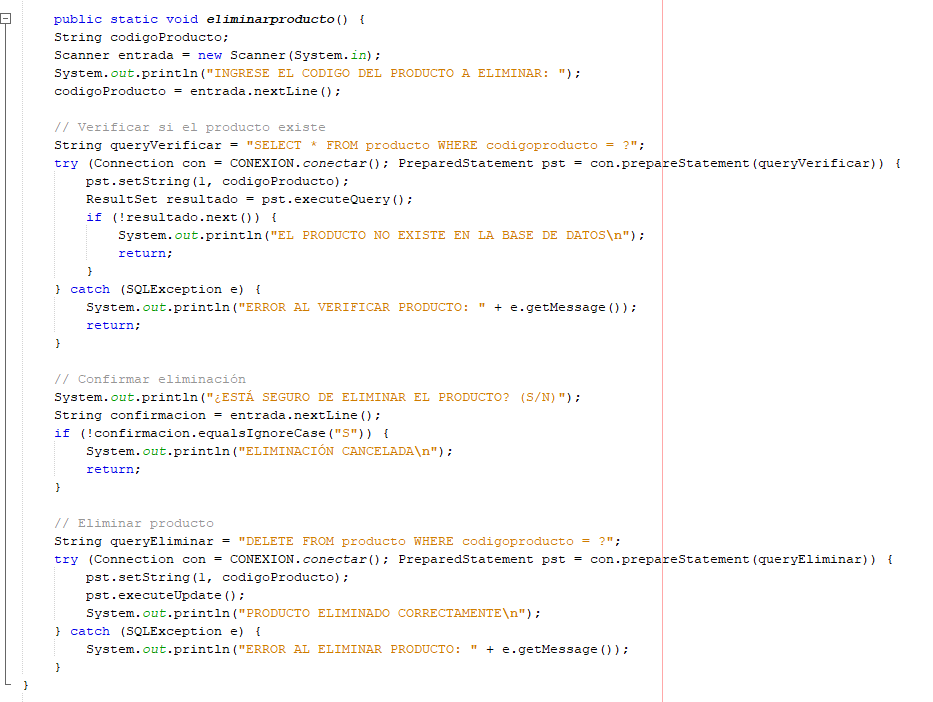
Se añade el método para Actualizar nuestros registros previamente añadidos

UPDATE

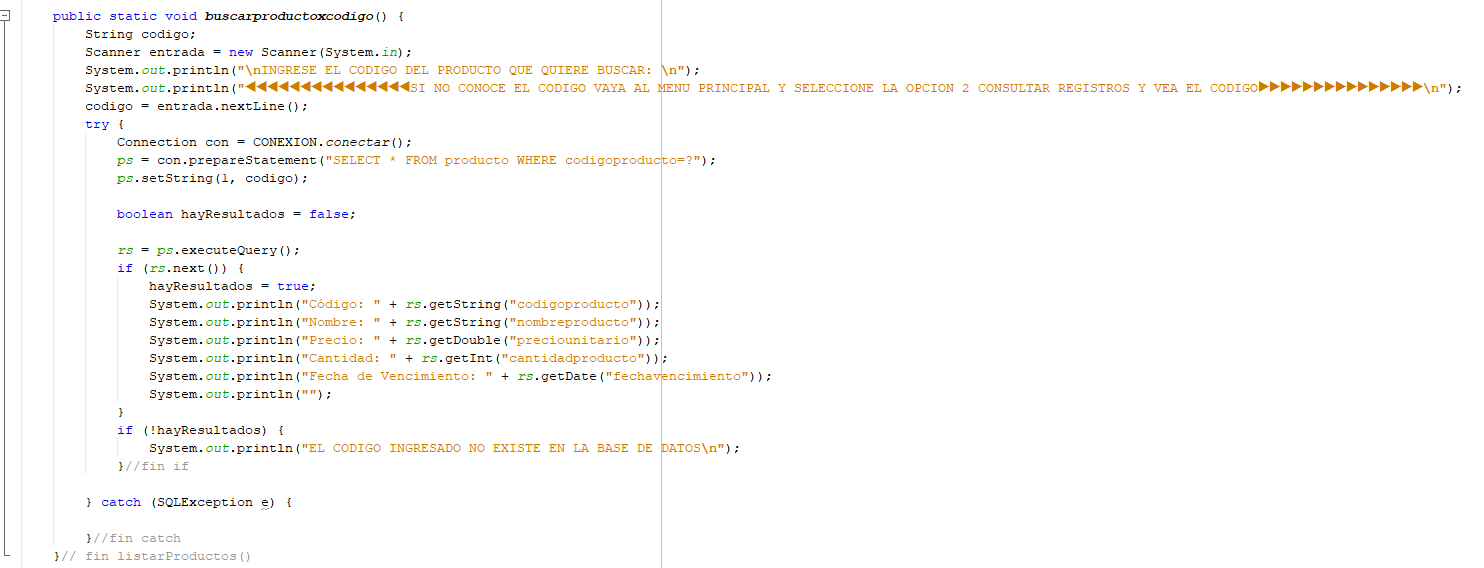


Se añade el método para Eliminar nuestros registros previamente añadidos

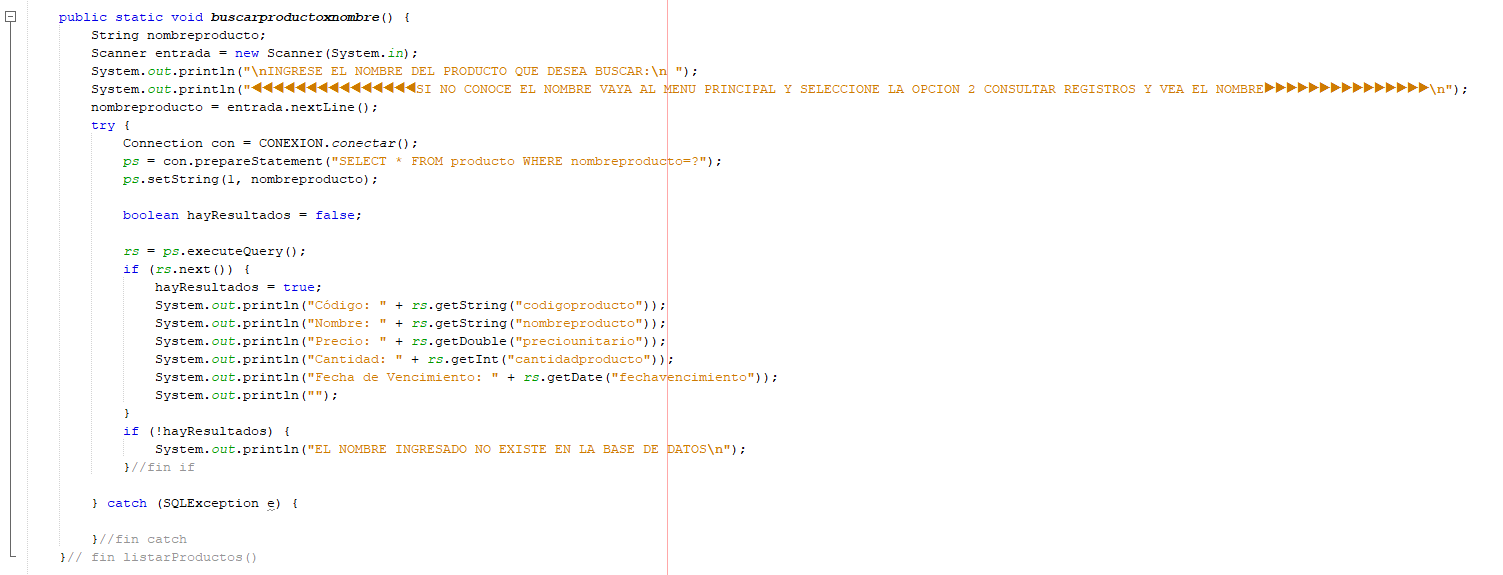
DELETE



Aparte del CRUD se añade otros métodos como el siguiente que nos permite buscar un registro mediante su código o llave primaria



También se añade otro método para buscar solo que este busca mediante el nombre del producto



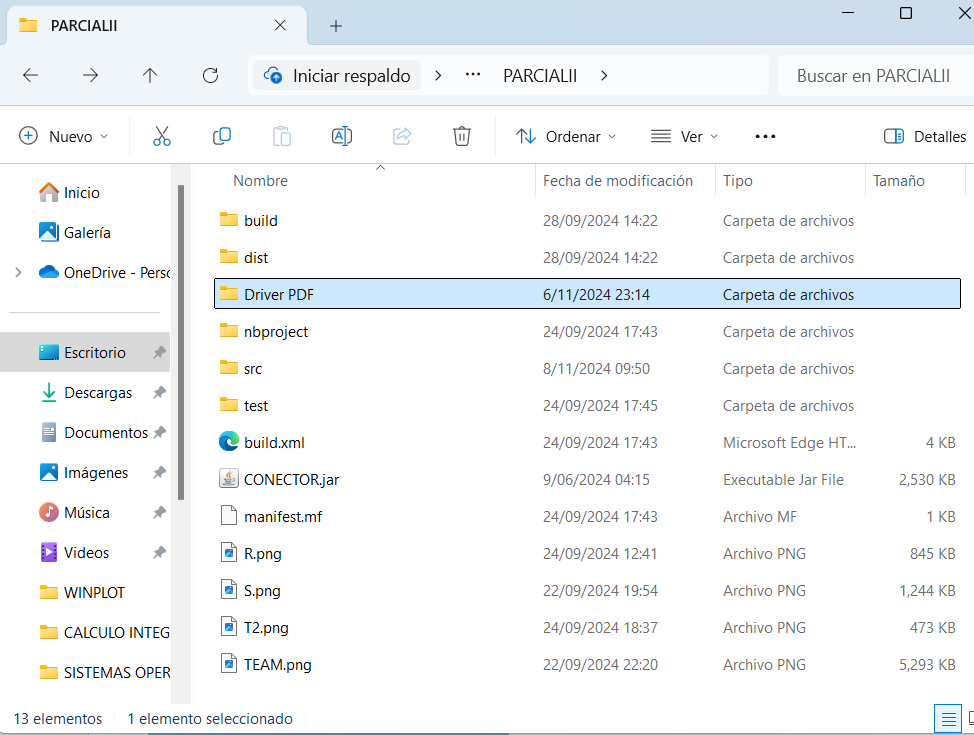
También utilizamos otro método que nos permite crear un reporte de Stock en existencia en un documento PDF con dirección en el escritorio



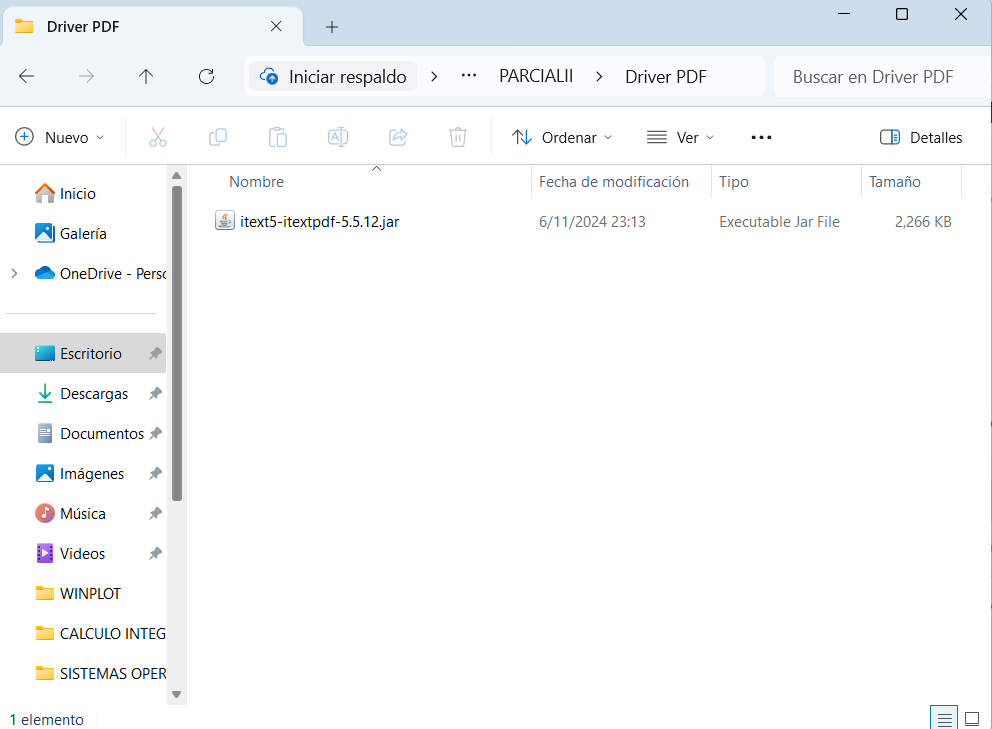
Debemos añadir otra biblioteca para que podamos generar un pdf en java, para eso ingresamos a la pagina del desarollador y descargamos iText.



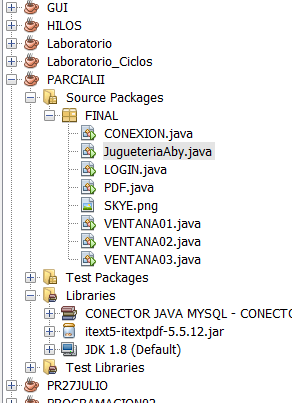
En este caso se añadió el archivo en una capeta con el nombre Driver DPF en la ubicación de nuestro proyecto



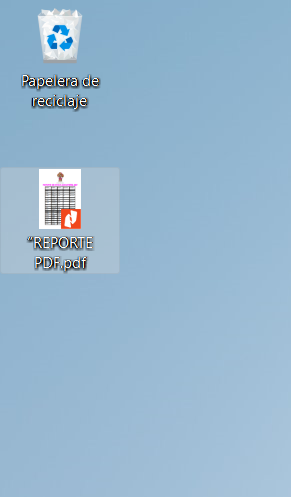
Acá mostramos que se añadió el archivo en la carpeta luego repetimos el mismo proceso como lo realizamos con el CONECTOR DE MYSQL



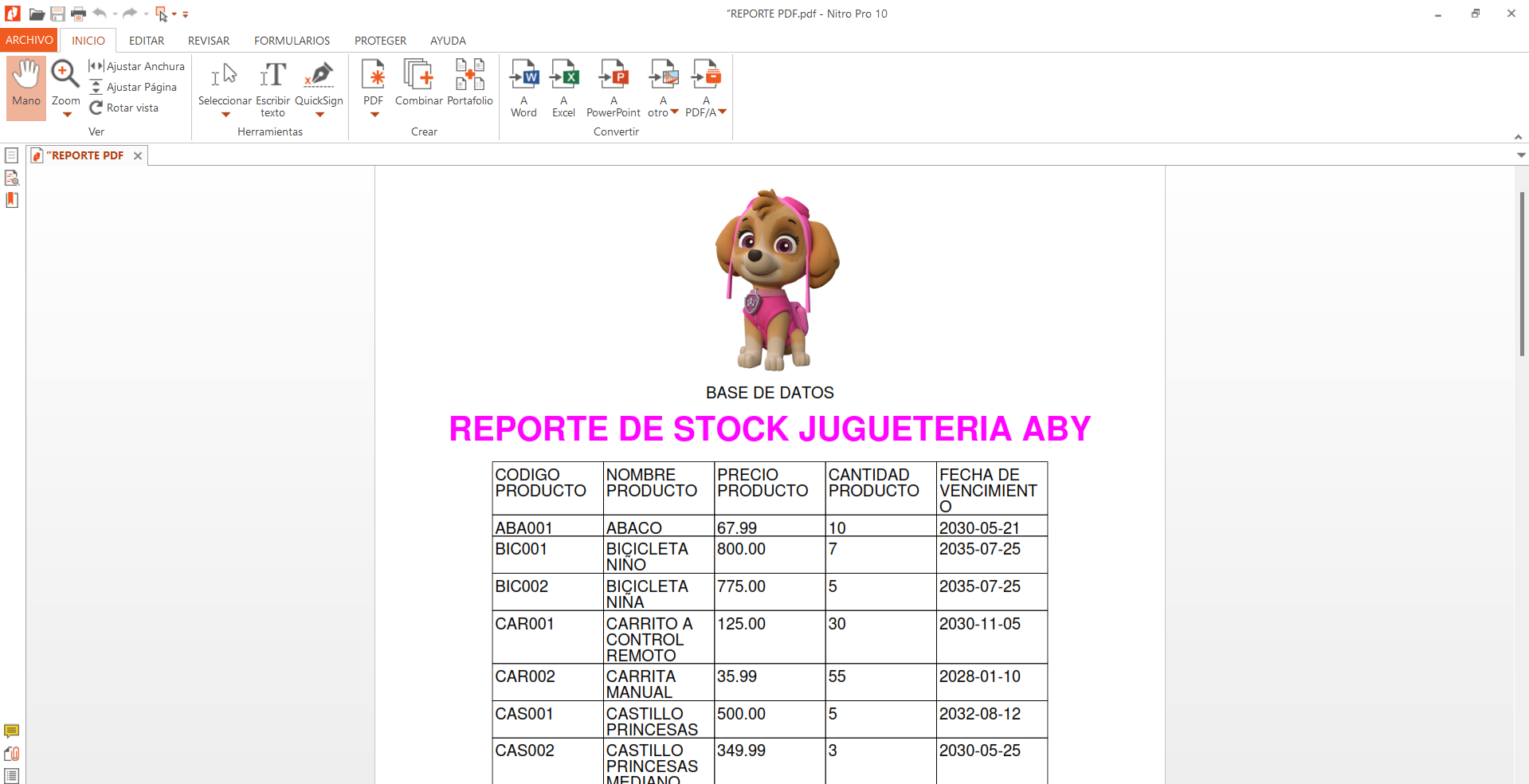
Nos debería mostra el archivo en las librerías de nuestro proyecto



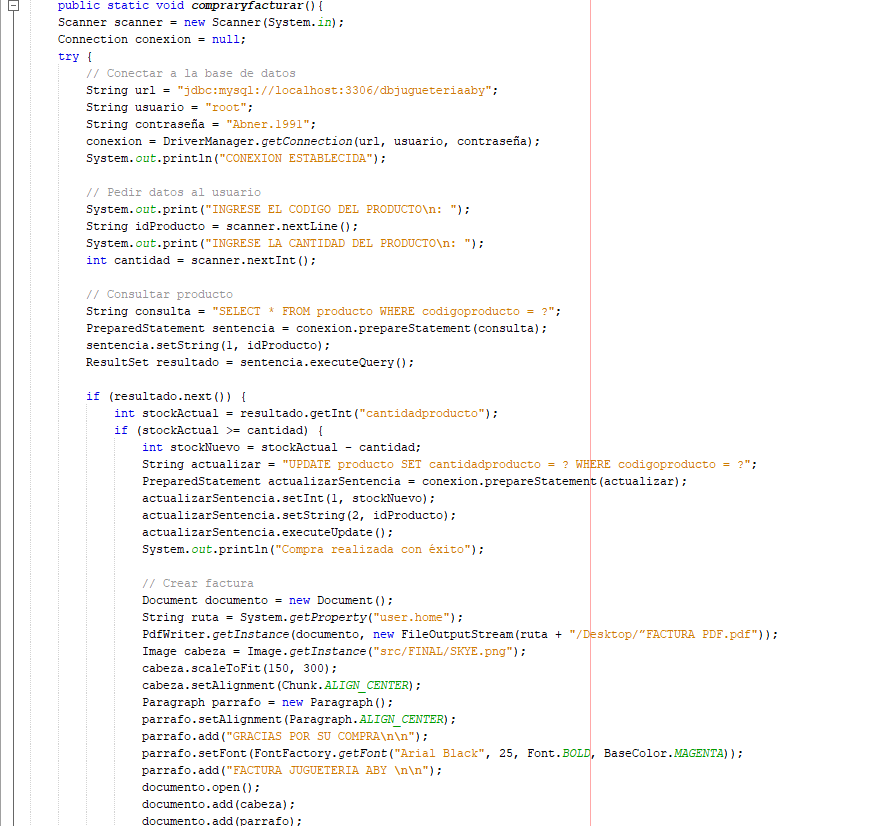
Ya habiendo realizado todos los procesos anteriores debe mostrarnos que se creo el archivo PDF en el escritorio

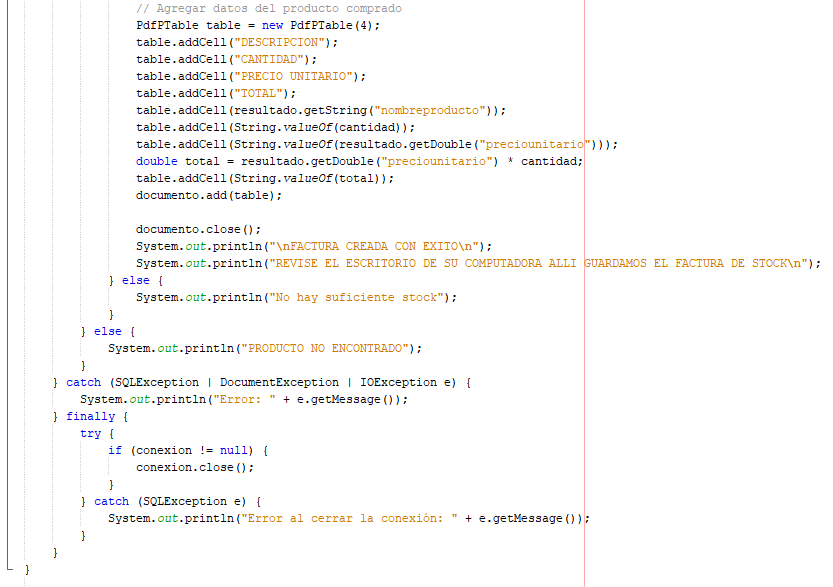


Al abrir el archivo nos muestra el nombre de nuestra empresa juntamente con la tabla de reportes de productos en existencia

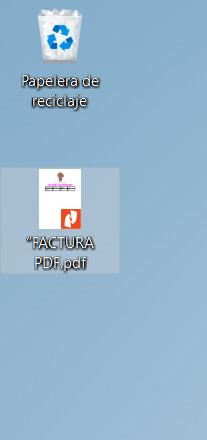


También añadimos un método que permite realizar una compra y genera factura en el escritorio en PDF





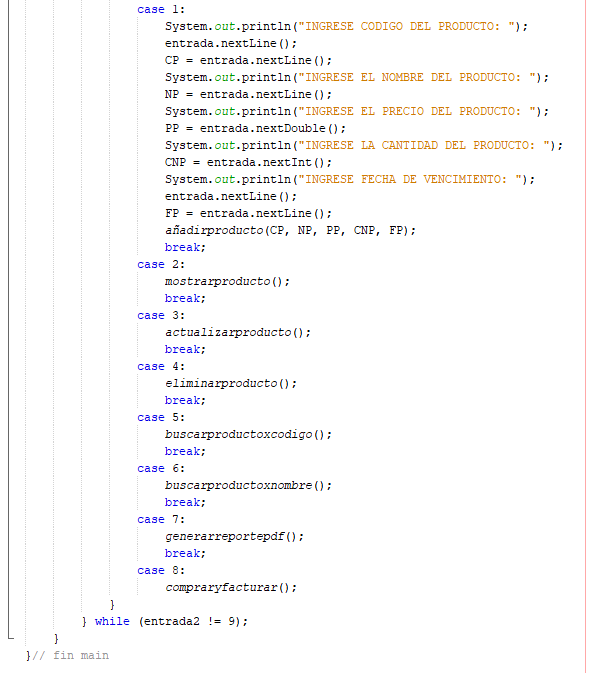
Al realizar una compra tenemos nuestro archivo generado



Al abrir nuestra factura nos muestra el producto elegido la cantidad que se solicito el precio unitario y el total



Por ultimo añadimos nuestro método principal el cual ejecutara nuestro menú, pero para poder ingresar nos solicitara un usuario y contraseña, los cuales podemos ver en el código fuente de nuestro proyecto, ya que este proyecto solo es de fines educativos y solo creo un login básico mediante consola

Por ultimo dejo el código en modo editable para que se pueda reutilizar, mejorar o corregir

package FINAL;

import java.sql.Statement;

import java.util.Scanner;

import com.itextpdf.text.Document;

import com.itextpdf.text.DocumentException;

import com.itextpdf.text.pdf.PdfPTable;

import com.itextpdf.text.pdf.PdfWriter;

import java.io.FileNotFoundException;

import java.io.FileOutputStream;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import com.itextpdf.text.Chunk;

import com.itextpdf.text.Image;

import com.itextpdf.text.Paragraph;

import com.itextpdf.text.FontFactory;

import com.itextpdf.text.BaseColor;

import com.itextpdf.text.Font;

import java.io.IOException;

public class JugueteriaAby {

private static final String URL = "jdbc:mysql://localhost:3306/dbjugueteriaaby";

private static final String USER = "root";

private static final String PASSWORD = "Abner.1991";

private static PreparedStatement ps;

private static ResultSet rs;

public static Connection conectar() {

Connection conexion = null;

try {

conexion = DriverManager.getConnection(URL, USER, PASSWORD);

System.out.println("\nCONEXION EXITOSA A LA BASE DE DATOS\n");

} catch (SQLException e) {

System.out.println("ERROR AL CONECTAR " + e.getMessage());

}

return conexion;

}// fin conectar()

public static void añadirproducto(String codigo, String nombre, double precio, int cantidad, String fecha) {

String query = "INSERT INTO producto (codigoproducto, nombreproducto, preciounitario, cantidadproducto, fechavencimiento) VALUES (?,?, ?, ?, ?)";

try (Connection con = CONEXION.conectar(); PreparedStatement pst = con.prepareStatement(query)) {

pst.setString(1, codigo);

pst.setString(2, nombre);

pst.setDouble(3, precio);

pst.setInt(4, cantidad);

pst.setDate(5, java.sql.Date.valueOf(fecha));

pst.executeUpdate();

System.out.println("PRODUCTO AÑADIDO CORRECTAMENTE\n");

} catch (SQLException e) {

}

}// fin insertarProducto()

public static void mostrarproducto() {

String query = "select \* from producto;";

try (Connection con = CONEXION.conectar(); Statement st = con.createStatement(); ResultSet rs = st.executeQuery(query)) {

boolean hayResultados = false;

while (rs.next()) {

hayResultados = true;

System.out.println("Código: " + rs.getString("codigoproducto"));

System.out.println("Nombre: " + rs.getString("nombreproducto"));

System.out.println("Precio: " + rs.getDouble("preciounitario"));

System.out.println("Cantidad: " + rs.getInt("cantidadproducto"));

System.out.println("Fecha de Vencimiento: " + rs.getDate("fechavencimiento"));

System.out.println("");

}

if (!hayResultados) {

System.out.println("LA BASE DE DATOS ESTA VACIA");

}//fin if

} catch (SQLException e) {

}//fin catch

}// fin listarProductos()

public static void buscarproductoxcodigo() {

String codigo;

Scanner entrada = new Scanner(System.in);

System.out.println("\nINGRESE EL CODIGO DEL PRODUCTO QUE QUIERE BUSCAR: \n");

System.out.println("◄◄◄◄◄◄◄◄◄◄◄◄◄◄◄SI NO CONOCE EL CODIGO VAYA AL MENU PRINCIPAL Y SELECCIONE LA OPCION 2 CONSULTAR REGISTROS Y VEA EL CODIGO►►►►►►►►►►►►►►►\n");

codigo = entrada.nextLine();

try {

Connection con = CONEXION.conectar();

ps = con.prepareStatement("SELECT \* FROM producto WHERE codigoproducto=?");

ps.setString(1, codigo);

boolean hayResultados = false;

rs = ps.executeQuery();

if (rs.next()) {

hayResultados = true;

System.out.println("Código: " + rs.getString("codigoproducto"));

System.out.println("Nombre: " + rs.getString("nombreproducto"));

System.out.println("Precio: " + rs.getDouble("preciounitario"));

System.out.println("Cantidad: " + rs.getInt("cantidadproducto"));

System.out.println("Fecha de Vencimiento: " + rs.getDate("fechavencimiento"));

System.out.println("");

}

if (!hayResultados) {

System.out.println("EL CODIGO INGRESADO NO EXISTE EN LA BASE DE DATOS\n");

}//fin if

} catch (SQLException e) {

}//fin catch

}// fin listarProductos()

public static void buscarproductoxnombre() {

String nombreproducto;

Scanner entrada = new Scanner(System.in);

System.out.println("\nINGRESE EL NOMBRE DEL PRODUCTO QUE DESEA BUSCAR:\n ");

System.out.println("◄◄◄◄◄◄◄◄◄◄◄◄◄◄◄SI NO CONOCE EL NOMBRE VAYA AL MENU PRINCIPAL Y SELECCIONE LA OPCION 2 CONSULTAR REGISTROS Y VEA EL NOMBRE►►►►►►►►►►►►►►►\n");

nombreproducto = entrada.nextLine();

try {

Connection con = CONEXION.conectar();

ps = con.prepareStatement("SELECT \* FROM producto WHERE nombreproducto=?");

ps.setString(1, nombreproducto);

boolean hayResultados = false;

rs = ps.executeQuery();

if (rs.next()) {

hayResultados = true;

System.out.println("Código: " + rs.getString("codigoproducto"));

System.out.println("Nombre: " + rs.getString("nombreproducto"));

System.out.println("Precio: " + rs.getDouble("preciounitario"));

System.out.println("Cantidad: " + rs.getInt("cantidadproducto"));

System.out.println("Fecha de Vencimiento: " + rs.getDate("fechavencimiento"));

System.out.println("");

}

if (!hayResultados) {

System.out.println("EL NOMBRE INGRESADO NO EXISTE EN LA BASE DE DATOS\n");

}//fin if

} catch (SQLException e) {

}//fin catch

}// fin listarProductos()

public static void actualizarproducto() {

String codigoProducto, nombre, fecha;

double precio;

int cantidad;

Scanner entrada = new Scanner(System.in);

System.out.println("INGRESE CODIGO DEL PRODUCTO A MODIFICAR:");

codigoProducto = entrada.nextLine();

System.out.println("INGRESE NUEVO NOMBRE");

nombre = entrada.nextLine();

System.out.println("INGRESE NUEVO PRECIO");

precio = entrada.nextDouble();

System.out.println("INGRESE NUEVA CANTIDAD");

cantidad = entrada.nextInt();

System.out.println("INGRESE NUEVA FECHA");

fecha = entrada.next();

entrada.nextLine();

String query = "UPDATE producto SET nombreproducto = ?, preciounitario = ?, cantidadproducto = ?, fechavencimiento = ? WHERE codigoproducto = ?";

try (Connection con = CONEXION.conectar(); PreparedStatement pst = con.prepareStatement(query)) {

pst.setString(1, nombre);

pst.setDouble(2, precio);

pst.setInt(3, cantidad);

pst.setDate(4, java.sql.Date.valueOf(fecha));

pst.setString(5, codigoProducto);

pst.executeUpdate();

System.out.println("\nPRODUCTO ACTUALIZADO CORRECTAMENTE\n");

} catch (SQLException e) {

}

}// fin actualizarProducto

public static void eliminarproducto() {

String codigoProducto;

Scanner entrada = new Scanner(System.in);

System.out.println("INGRESE EL CODIGO DEL PRODUCTO A ELIMINAR: ");

codigoProducto = entrada.nextLine();

// Verificar si el producto existe

String queryVerificar = "SELECT \* FROM producto WHERE codigoproducto = ?";

try (Connection con = CONEXION.conectar(); PreparedStatement pst = con.prepareStatement(queryVerificar)) {

pst.setString(1, codigoProducto);

ResultSet resultado = pst.executeQuery();

if (!resultado.next()) {

System.out.println("EL PRODUCTO NO EXISTE EN LA BASE DE DATOS\n");

return;

}

} catch (SQLException e) {

System.out.println("ERROR AL VERIFICAR PRODUCTO: " + e.getMessage());

return;

}

// Confirmar eliminación

System.out.println("¿ESTÁ SEGURO DE ELIMINAR EL PRODUCTO? (S/N)");

String confirmacion = entrada.nextLine();

if (!confirmacion.equalsIgnoreCase("S")) {

System.out.println("ELIMINACIÓN CANCELADA\n");

return;

}

// Eliminar producto

String queryEliminar = "DELETE FROM producto WHERE codigoproducto = ?";

try (Connection con = CONEXION.conectar(); PreparedStatement pst = con.prepareStatement(queryEliminar)) {

pst.setString(1, codigoProducto);

pst.executeUpdate();

System.out.println("PRODUCTO ELIMINADO CORRECTAMENTE\n");

} catch (SQLException e) {

System.out.println("ERROR AL ELIMINAR PRODUCTO: " + e.getMessage());

}

}

public static void generarreportepdf() {

Document documento = new Document();

try {

String ruta = System.getProperty("user.home");

PdfWriter.getInstance(documento, new FileOutputStream(ruta + "/Desktop/”REPORTE PDF.pdf"));

Image cabeza = Image.getInstance("src/PARCIAL02/SKYE.png");

cabeza.scaleToFit(150, 300);

cabeza.setAlignment(Chunk.ALIGN\_CENTER);

Paragraph parrafo = new Paragraph();

parrafo.setAlignment(Paragraph.ALIGN\_CENTER);

parrafo.add("BASE DE DATOS\n\n");

parrafo.setFont(FontFactory.getFont("Arial Black", 25, Font.BOLD, BaseColor.MAGENTA));

parrafo.add("REPORTE DE STOCK JUGUETERIA ABY \n\n");

documento.open();

documento.add(cabeza);

documento.add(parrafo);

PdfPTable table = new PdfPTable(5);

table.addCell("CODIGO PRODUCTO");

table.addCell("NOMBRE PRODUCTO");

table.addCell("PRECIO PRODUCTO");

table.addCell("CANTIDAD PRODUCTO");

table.addCell("FECHA DE VENCIMIENTO");

try {

Connection cn = DriverManager.getConnection("jdbc:mysql://localhost/dbjugueteriaaby", "root", "Abner.1991");

PreparedStatement pst = cn.prepareStatement("select \* from producto");

rs = pst.executeQuery();

if (rs.next()) {

do {

table.addCell(rs.getString(1));

table.addCell(rs.getString(2));

table.addCell(rs.getString(3));

table.addCell(rs.getString(4));

table.addCell(rs.getString(5));

} while (rs.next());

documento.add(table);

}

} catch (DocumentException | SQLException e) {

}

documento.close();

System.out.println("\nREPORTE CREADO CON EXITO\n");

System.out.println("REVISE EL ESCRITORIO DE SU COMPUTADORA ALLI GUARDAMOS EL REPORTE DE STOCK\n");

} catch (DocumentException | FileNotFoundException e) {

System.out.println("Error en PDF" + e);

} catch (IOException e) {

System.out.println("error en la imagen" + e);

}

}

public static void compraryfacturar(){

Scanner scanner = new Scanner(System.in);

Connection conexion = null;

try {

// Conectar a la base de datos

String url = "jdbc:mysql://localhost:3306/dbjugueteriaaby";

String usuario = "root";

String contraseña = "Abner.1991";

conexion = DriverManager.getConnection(url, usuario, contraseña);

System.out.println("CONEXION ESTABLECIDA");

// Pedir datos al usuario

System.out.print("INGRESE EL CODIGO DEL PRODUCTO\n: ");

String idProducto = scanner.nextLine();

System.out.print("INGRESE LA CANTIDAD DEL PRODUCTO\n: ");

int cantidad = scanner.nextInt();

// Consultar producto

String consulta = "SELECT \* FROM producto WHERE codigoproducto = ?";

PreparedStatement sentencia = conexion.prepareStatement(consulta);

sentencia.setString(1, idProducto);

ResultSet resultado = sentencia.executeQuery();

if (resultado.next()) {

int stockActual = resultado.getInt("cantidadproducto");

if (stockActual >= cantidad) {

int stockNuevo = stockActual - cantidad;

String actualizar = "UPDATE producto SET cantidadproducto = ? WHERE codigoproducto = ?";

PreparedStatement actualizarSentencia = conexion.prepareStatement(actualizar);

actualizarSentencia.setInt(1, stockNuevo);

actualizarSentencia.setString(2, idProducto);

actualizarSentencia.executeUpdate();

System.out.println("Compra realizada con éxito");

// Crear factura

Document documento = new Document();

String ruta = System.getProperty("user.home");

PdfWriter.getInstance(documento, new FileOutputStream(ruta + "/Desktop/”FACTURA PDF.pdf"));

Image cabeza = Image.getInstance("src/PARCIAL02/SKYE.png");

cabeza.scaleToFit(150, 300);

cabeza.setAlignment(Chunk.ALIGN\_CENTER);

Paragraph parrafo = new Paragraph();

parrafo.setAlignment(Paragraph.ALIGN\_CENTER);

parrafo.add("GRACIAS POR SU COMPRA\n\n");

parrafo.setFont(FontFactory.getFont("Arial Black", 25, Font.BOLD, BaseColor.MAGENTA));

parrafo.add("FACTURA JUGUETERIA ABY \n\n");

documento.open();

documento.add(cabeza);

documento.add(parrafo);

// Agregar datos del producto comprado

PdfPTable table = new PdfPTable(4);

table.addCell("DESCRIPCION");

table.addCell("CANTIDAD");

table.addCell("PRECIO UNITARIO");

table.addCell("TOTAL");

table.addCell(resultado.getString("nombreproducto"));

table.addCell(String.valueOf(cantidad));

table.addCell(String.valueOf(resultado.getDouble("preciounitario")));

double total = resultado.getDouble("preciounitario") \* cantidad;

table.addCell(String.valueOf(total));

documento.add(table);

documento.close();

System.out.println("\nFACTURA CREADA CON EXITO\n");

System.out.println("REVISE EL ESCRITORIO DE SU COMPUTADORA ALLI GUARDAMOS EL FACTURA DE STOCK\n");

} else {

System.out.println("No hay suficiente stock");

}

} else {

System.out.println("PRODUCTO NO ENCONTRADO");

}

} catch (SQLException | DocumentException | IOException e) {

System.out.println("Error: " + e.getMessage());

} finally {

try {

if (conexion != null) {

conexion.close();

}

} catch (SQLException e) {

System.out.println("Error al cerrar la conexión: " + e.getMessage());

}

}

}

public static void main(String[] args) {

Scanner entrada = new Scanner(System.in);

int entrada2, CNP, PASS;

String NP, CP, FP;

double PP;

String user = "ABNER";

String pass = "123";

boolean acceso = false;

while (!acceso) {

System.out.println("♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦LOGIN♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦");

System.out.println("");

System.out.println("INGRESE SU USUARIO: ");

String user2 = entrada.nextLine();

System.out.println("INGRESE SU CONTRASEÑA: ");

String pass2 = entrada.nextLine();

if (user2.equals(user) && (pass2.equals(pass))) {

System.out.println("\n♥♥♥♥♥♥♥♥♥♥♥♥♥♥♥♥♥♥BIENVENIDO♥♥♥♥♥♥♥♥♥♥♥♥♥♥♥♥♥♥\n");

acceso = true;

} else {

System.out.println("\n ◄◄◄◄◄◄◄◄◄◄◄◄◄◄◄ ERROR ►►►►►►►►►►►►►►► \n");

System.out.println("\n ◄◄◄ USUARIO O CONTRASEÑA INCORRECTA, INTENTELO DE NUEVO ►►►\n");

}

}

do {

System.out.println("♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦MENU PRINCIPAL♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦\n");

System.out.println("1••••••••••••INGRESAR REGISTRO");

System.out.println("2••••••••••••CONSULTAR REGISTROS");

System.out.println("3••••••••••••ACTUALIZAR REGISTRO");

System.out.println("4••••••••••••ELIMINAR REGISTRO");

System.out.println("5••••••••••••BUSCAR REGISTRO POR CODIGO");

System.out.println("6••••••••••••BUSCAR REGISTRO POR NOMBRE");

System.out.println("7••••••••••••GENERAR REPORTE");

System.out.println("8••••••••••••COMPRAR Y FACTURAR");

System.out.println("9••••••••••••SALIR");

System.out.println("ELIGA UNA OPCION\n");

entrada2 = entrada.nextInt();

switch (entrada2) {

case 1:

System.out.println("INGRESE CODIGO DEL PRODUCTO: ");

entrada.nextLine();

CP = entrada.nextLine();

System.out.println("INGRESE EL NOMBRE DEL PRODUCTO: ");

NP = entrada.nextLine();

System.out.println("INGRESE EL PRECIO DEL PRODUCTO: ");

PP = entrada.nextDouble();

System.out.println("INGRESE LA CANTIDAD DEL PRODUCTO: ");

CNP = entrada.nextInt();

System.out.println("INGRESE FECHA DE VENCIMIENTO: ");

entrada.nextLine();

FP = entrada.nextLine();

añadirproducto(CP, NP, PP, CNP, FP);

break;

case 2:

mostrarproducto();

break;

case 3:

actualizarproducto();

break;

case 4:

eliminarproducto();

break;

case 5:

buscarproductoxcodigo();

break;

case 6:

buscarproductoxnombre();

break;

case 7:

generarreportepdf();

break;

case 8:

compraryfacturar();

}

} while (entrada2 != 9);

}

}// fin main