/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package pruebasmovil;

import Objetos.Bar;

import Objetos.Camarero;

import Objetos.Carta;

import Objetos.Comanda;

import Objetos.Producto;

import Objetos.ProductoPrecio;

import Objetos.lineaComanda;

import Objetos.Oferta;

import java.sql.Connection;

import java.sql.Date;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.util.List;

import java.util.Vector;

import java.util.logging.Level;

import java.util.logging.Logger;

/\*\*

\*

\* @author appComanda

\*/

public class ServiciosAppComanda {

//-------------------------------------------------------------BAR-------------------------------------------------------------------------------------

public boolean insertarBar(int id, String nombre, String direccion ,int telefono, String correo,double latitud, double longitud, String provincia, String municipio, String contraseña){

boolean exito=false;

Connection con = null;

String sql ="INSERT INTO `bar`(`Id\_Bar`, `Nombre`, `Direccion`, `Telefono`, `Longitud`, `Latitud`, `Provincia`, `Municipio`, `Correo`, `Contraseña`) VALUES (?,?,?,?,?,?,?,?,?,?)";

PreparedStatement ps;

try {

con = ConexionDB.GetConnection();

con.setAutoCommit(false);

ps= con.prepareStatement(sql);

ps.setInt(1, id);

ps.setString(2, nombre);

ps.setString(3, direccion);

ps.setInt(4, telefono);

ps.setDouble(5, longitud);

ps.setDouble(6, latitud);

ps.setString(7, provincia);

ps.setString(8, municipio);

ps.setString(9, correo);

ps.setString(10, contraseña);

ps.executeUpdate();

ps.close();

con.commit();

exito=true;

} catch (SQLException ex) {

Logger.getLogger(ConexionDB.class.getName()).log(Level.SEVERE, null, ex);

if (con!=null) try {

con.rollback();

} catch (SQLException ex1) {

Logger.getLogger(ServiciosAppComanda.class.getName()).log(Level.SEVERE, null, ex1);

}

}finally{

if (con !=null) try {

con.close();

} catch (SQLException ex) {

Logger.getLogger(ServiciosAppComanda.class.getName()).log(Level.SEVERE, null, ex);

}

return exito;

}

}

public List<Bar> listaBar(){

List<Bar> lb = new Vector<Bar>();

Connection con;

PreparedStatement ps = null;

ResultSet rs = null;

Bar b = null;

con= ConexionDB.GetConnection();

String sql = "SELECT `Id\_Bar`,`Nombre` ,`Direccion`,`Telefono`, `Longitud`,`Latitud`,`Provincia`, `Municipio`,`Correo`, `Contraseña` FROM `bar` ";

try {

ps=con.prepareStatement(sql);

rs=ps.executeQuery();

while (rs.next()){

b = new Bar(rs.getInt(1), rs.getString(2), rs.getString(3), rs.getInt(4), rs.getString(9), rs.getDouble(5), rs.getDouble(6), rs.getString(7), rs.getString(8), rs.getString(10));

lb.add(b);

}

rs.close();

ps.close();

} catch (SQLException ex) {

Logger.getLogger(ServiciosAppComanda.class.getName()).log(Level.SEVERE, null, ex);

}finally{

try {

con.close();

} catch (SQLException ex) {

Logger.getLogger(ServiciosAppComanda.class.getName()).log(Level.SEVERE, null, ex);

}

return lb;

}

}

public boolean borrarBar(int id){

Connection con;

boolean exito = false;

PreparedStatement ps = null;

ResultSet rs = null;

con= ConexionDB.GetConnection();

String sql ="DELETE FROM `appcomanda`.`bar` WHERE `comanda`.`Id\_Bar` = ?";

try {

ps=con.prepareStatement(sql);

ps.setInt(1, id);

ps.executeUpdate();

ps.close();

con.commit();

exito=true;

} catch (SQLException ex) {

con.rollback();

Logger.getLogger(ServiciosAppComanda.class.getName()).log(Level.SEVERE, null, ex);

}finally{

try {

con.close();

} catch (SQLException ex) {

Logger.getLogger(ServiciosAppComanda.class.getName()).log(Level.SEVERE, null, ex);

}

return exito;

}

}

//-----------------------------Producto--------------------------------------------------------------------------------

public boolean insertarProducto(int id, String nombre, String descripcion ,String categoria){

boolean exito=false;

Connection con = null;

String sql ="INSERT INTO `producto`(`Id\_Producto`, `Nombre`, `Descripcion`, `Categoria`) VALUES (?,?,?,?)";

PreparedStatement ps;

try {

con = ConexionDB.GetConnection();

con.setAutoCommit(false);

ps= con.prepareStatement(sql);

ps.setInt(1, id);

ps.setString(2, nombre);

ps.setString(3, descripcion);

ps.setString(4, categoria);

ps.executeUpdate();

ps.close();

con.commit();

exito=true;

} catch (SQLException ex) {

Logger.getLogger(ConexionDB.class.getName()).log(Level.SEVERE, null, ex);

if (con!=null) try {

con.rollback();

} catch (SQLException ex1) {

Logger.getLogger(ServiciosAppComanda.class.getName()).log(Level.SEVERE, null, ex1);

}

}finally{

if (con !=null) try {

con.close();

} catch (SQLException ex) {

Logger.getLogger(ServiciosAppComanda.class.getName()).log(Level.SEVERE, null, ex);

}

return exito;

}

}

public List<Producto> listaProducto(){

List<Producto> lb = new Vector<>();

Connection con;

PreparedStatement ps = null;

ResultSet rs = null;

Producto p = null;

con= ConexionDB.GetConnection();

String sql = "SELECT `Id\_Producto`,`Nombre`,`Descripcion`,`Categoria` FROM `producto`";

try {

ps=con.prepareStatement(sql);

rs=ps.executeQuery();

while (rs.next()){

p = new Producto(rs.getInt(1), rs.getString(2), rs.getString(3), rs.getString(4));

lb.add(p);

}

rs.close();

ps.close();

} catch (SQLException ex) {

Logger.getLogger(ServiciosAppComanda.class.getName()).log(Level.SEVERE, null, ex);

}finally{

try {

con.close();

} catch (SQLException ex) {

Logger.getLogger(ServiciosAppComanda.class.getName()).log(Level.SEVERE, null, ex);

}

return lb;

}

}

public Producto buscarProducto(int id){

Connection con;

PreparedStatement ps = null;

ResultSet rs = null;

Producto p = null;

con= ConexionDB.GetConnection();

String sql = "SELECT `Id\_Producto`,`Nombre`,`Descripcion`,`Categoria` FROM `producto` WHERE `Id\_Producto`=?";

try {

ps=con.prepareStatement(sql);

ps.setInt(1, id);

rs=ps.executeQuery();

if (rs.next()){

p = new Producto(rs.getInt(1), rs.getString(2), rs.getString(3), rs.getString(4));

}

rs.close();

ps.close();

} catch (SQLException ex) {

Logger.getLogger(ServiciosAppComanda.class.getName()).log(Level.SEVERE, null, ex);

}finally{

try {

con.close();

} catch (SQLException ex) {

Logger.getLogger(ServiciosAppComanda.class.getName()).log(Level.SEVERE, null, ex);

}

return p;

}

}

public boolean borrarProducto(int id){

Connection con;

boolean exito = false;

PreparedStatement ps = null;

con= ConexionDB.GetConnection();

String sql ="DELETE FROM `appcomanda`.`producto` WHERE `comanda`.`Id\_Producto` = ?";

try {

ps=con.prepareStatement(sql);

ps.setInt(1, id);

ps.executeUpdate();

ps.close();

con.commit();

exito=true;

} catch (SQLException ex) {

con.rollback();

Logger.getLogger(ServiciosAppComanda.class.getName()).log(Level.SEVERE, null, ex);

}finally{

try {

con.close();

} catch (SQLException ex) {

Logger.getLogger(ServiciosAppComanda.class.getName()).log(Level.SEVERE, null, ex);

}

return exito;

}

}

//-----------------------------------------------------CAMARERO----------------------------------------

public boolean insertarCamarero(int id, String nombre, String apellidos ,String contraseña, String nick ,int id\_bar){

boolean exito=false;

Connection con = null;

String sql ="INSERT INTO `camarero`(`Id\_Camarero`, `Nombre`, `Apellidos`, `Contraseña`, `nick`, `Id\_Bar`) VALUES (?,?,?,?,?,?)";

PreparedStatement ps;

try {

con = ConexionDB.GetConnection();

con.setAutoCommit(false);

ps= con.prepareStatement(sql);

ps.setInt(1, id);

ps.setString(2, nombre);

ps.setString(3, apellidos);

ps.setString(4, contraseña);

ps.setString(5, nick);

ps.setInt(6, id\_bar);

ps.executeUpdate();

ps.close();

con.commit();

exito=true;

} catch (SQLException ex) {

Logger.getLogger(ConexionDB.class.getName()).log(Level.SEVERE, null, ex);

if (con!=null) try {

con.rollback();

} catch (SQLException ex1) {

Logger.getLogger(ServiciosAppComanda.class.getName()).log(Level.SEVERE, null, ex1);

}

}finally{

if (con !=null) try {

con.close();

} catch (SQLException ex) {

Logger.getLogger(ServiciosAppComanda.class.getName()).log(Level.SEVERE, null, ex);

}

return exito;

}

}

public List<Camarero> listaCamareroBar(int id){

Connection con;

PreparedStatement ps = null;

ResultSet rs = null;

List<Camarero> lc = new Vector<Camarero>();

Camarero c = null;

con= ConexionDB.GetConnection();

String sql = "SELECT `Id\_Camarero`, `Nombre`, `Apellidos`, `Contraseña`, `nick`, `Id\_Bar` FROM `camarero` WHERE `Id\_Bar`=?";

try {

ps=con.prepareStatement(sql);

ps.setInt(1, id);

rs=ps.executeQuery();

while (rs.next()){

c = new Camarero(rs.getInt(1), rs.getString(2), rs.getString(3), rs.getString(4),rs.getString(5),id);

lc.add(c);

}

rs.close();

ps.close();

} catch (SQLException ex) {

Logger.getLogger(ServiciosAppComanda.class.getName()).log(Level.SEVERE, null, ex);

}finally{

try {

con.close();

} catch (SQLException ex) {

Logger.getLogger(ServiciosAppComanda.class.getName()).log(Level.SEVERE, null, ex);

}

return lc;

}

}

public boolean borrarCamarero(int id){

Connection con;

boolean exito = false;

PreparedStatement ps = null;

ResultSet rs = null;

con= ConexionDB.GetConnection();

String sql ="DELETE FROM `appcomanda`.`camarero` WHERE `comanda`.`Id\_Camarero` = ?";

try {

ps=con.prepareStatement(sql);

ps.setInt(1, id);

ps.executeUpdate();

ps.close();

con.commit();

exito=true;

} catch (SQLException ex) {

con.rollback();

Logger.getLogger(ServiciosAppComanda.class.getName()).log(Level.SEVERE, null, ex);

}finally{

try {

con.close();

} catch (SQLException ex) {

Logger.getLogger(ServiciosAppComanda.class.getName()).log(Level.SEVERE, null, ex);

}

return exito;

}

}

//------------------------------------------COMANDA-----------------------------------------------------------------

public boolean insertarComanda(int id, int id\_bar, Date fecha ,int mesa, double coste ,int id\_camarero){

boolean exito=false;

Connection con = null;

String sql ="INSERT INTO `comanda`(`Id\_Comanda`, `Id\_Bar`, `Fecha`, `Mesa`, `Coste\_Total`, `Id\_Camarero`) VALUES (?,?,?,?,?,?)";

PreparedStatement ps;

try {

con = ConexionDB.GetConnection();

con.setAutoCommit(false);

ps= con.prepareStatement(sql);

ps.setInt(1, id);

ps.setInt(2, id\_bar);

ps.setDate(3, fecha);

ps.setInt(4, mesa);

ps.setDouble(5, coste);

ps.setInt(6, id\_camarero);

ps.executeUpdate();

ps.close();

con.commit();

exito=true;

} catch (SQLException ex) {

Logger.getLogger(ConexionDB.class.getName()).log(Level.SEVERE, null, ex);

if (con!=null) try {

con.rollback();

} catch (SQLException ex1) {

Logger.getLogger(ServiciosAppComanda.class.getName()).log(Level.SEVERE, null, ex1);

}

}finally{

if (con !=null) try {

con.close();

} catch (SQLException ex) {

Logger.getLogger(ServiciosAppComanda.class.getName()).log(Level.SEVERE, null, ex);

}

return exito;

}

}

public List<Comanda> listaComandasCamarero(int id){

Connection con;

PreparedStatement ps = null;

ResultSet rs = null;

List<Comanda> lc = new Vector<Comanda>();

Comanda c = null;

con= ConexionDB.GetConnection();

String sql = "SELECT `Id\_Comanda`, `Id\_Camarero`, `Id\_Bar`, `Fecha`, `Mesa`, `Coste\_Total` FROM `comanda` WHERE `Id\_Camarero`=?";

try {

ps=con.prepareStatement(sql);

ps.setInt(1, id);

rs=ps.executeQuery();

while (rs.next()){

c = new Comanda(rs.getInt(1), rs.getInt(3) ,rs.getDate(4), rs.getInt(5), rs.getDouble(6),id);

lc.add(c);

}

rs.close();

ps.close();

} catch (SQLException ex) {

Logger.getLogger(ServiciosAppComanda.class.getName()).log(Level.SEVERE, null, ex);

}finally{

try {

con.close();

} catch (SQLException ex) {

Logger.getLogger(ServiciosAppComanda.class.getName()).log(Level.SEVERE, null, ex);

}

return lc;

}

}

public List<Comanda> listaComandasBar(int id){

Connection con;

PreparedStatement ps = null;

ResultSet rs = null;

List<Comanda> lc = new Vector<Comanda>();

Comanda c = null;

con= ConexionDB.GetConnection();

String sql = "SELECT `Id\_Comanda`, `Id\_Camarero`, `Id\_Bar`, `Fecha`, `Mesa`, `Coste\_Total` FROM `comanda` WHERE `Id\_Bar`=?";

try {

ps=con.prepareStatement(sql);

ps.setInt(1, id);

rs=ps.executeQuery();

while (rs.next()){

c = new Comanda(rs.getInt(1), id ,rs.getDate(4), rs.getInt(5), rs.getDouble(6),rs.getInt(2));

lc.add(c);

}

rs.close();

ps.close();

} catch (SQLException ex) {

Logger.getLogger(ServiciosAppComanda.class.getName()).log(Level.SEVERE, null, ex);

}finally{

try {

con.close();

} catch (SQLException ex) {

Logger.getLogger(ServiciosAppComanda.class.getName()).log(Level.SEVERE, null, ex);

}

return lc;

}

}

public boolean borrarComanda(int id){

boolean exito=false;

Connection con = null;

String sql ="DELETE FROM `appcomanda`.`comanda` WHERE `comanda`.`Id\_Comanda` = ?";

PreparedStatement ps;

try {

con = ConexionDB.GetConnection();

con.setAutoCommit(false);

ps= con.prepareStatement(sql);

ps.setInt(1, id);

ps.executeUpdate();

ps.close();

con.commit();

exito=true;

} catch (SQLException ex) {

Logger.getLogger(ConexionDB.class.getName()).log(Level.SEVERE, null, ex);

if (con!=null) try {

con.rollback();

} catch (SQLException ex1) {

Logger.getLogger(ServiciosAppComanda.class.getName()).log(Level.SEVERE, null, ex1);

}

}finally{

if (con !=null) try {

con.close();

} catch (SQLException ex) {

Logger.getLogger(ServiciosAppComanda.class.getName()).log(Level.SEVERE, null, ex);

}

return exito;

}

}

//-----------------------------------------CARTA----------------------------------------------------------------------------------

public boolean insertarCarta(int id, int id\_producto ,int id\_bar, double precio , String foto){

boolean exito=false;

Connection con = null;

String sql ="INSERT INTO `carta`(`Id\_Producto`, `Id\_Bar`, `Precio`, `Foto`, `Id\_Carta`) VALUES (?,?,?,?,?)";

PreparedStatement ps;

try {

con = ConexionDB.GetConnection();

con.setAutoCommit(false);

ps= con.prepareStatement(sql);

System.out.println("EEE+ "+id+foto+precio+id\_bar+id\_producto);

ps.setInt(5, id);

ps.setInt(2, id\_bar);

ps.setInt(1, id\_producto);

ps.setDouble(3, precio);

ps.setString(4, foto);

ps.executeUpdate();

ps.close();

con.commit();

exito=true;

} catch (SQLException ex) {

Logger.getLogger(ConexionDB.class.getName()).log(Level.SEVERE, null, ex);

if (con!=null) try {

con.rollback();

} catch (SQLException ex1) {

Logger.getLogger(ServiciosAppComanda.class.getName()).log(Level.SEVERE, null, ex1);

}

}finally{

if (con !=null) try {

con.close();

} catch (SQLException ex) {

Logger.getLogger(ServiciosAppComanda.class.getName()).log(Level.SEVERE, null, ex);

}

return exito;

}

}

public List<ProductoPrecio> listaProductosBar(int id){

Connection con;

PreparedStatement ps = null;

ResultSet rs = null;

List<ProductoPrecio> lpp = new Vector<ProductoPrecio>();

ProductoPrecio pp = null;

con= ConexionDB.GetConnection();

String sql = "SELECT producto.Id\_Producto, producto.Nombre, producto.Descripcion, producto.Categoria, carta.Precio, carta.Id\_Carta FROM `carta` as carta JOIN Producto as producto on carta.Id\_Producto = producto.Id\_Producto where carta.Id\_Carta=?";

try {

ps=con.prepareStatement(sql);

ps.setInt(1, id);

rs=ps.executeQuery();

while (rs.next()){

pp = new ProductoPrecio(rs.getInt(1),rs.getString(2), rs.getString(3), rs.getString(4), rs.getDouble(5));

lpp.add(pp);

}

rs.close();

ps.close();

} catch (SQLException ex) {

Logger.getLogger(ServiciosAppComanda.class.getName()).log(Level.SEVERE, null, ex);

}finally{

try {

con.close();

} catch (SQLException ex) {

Logger.getLogger(ServiciosAppComanda.class.getName()).log(Level.SEVERE, null, ex);

}

return lpp;

}

}

public Carta cartaBar(int id){

Connection con;

PreparedStatement ps = null;

ResultSet rs = null;

Carta c = null;

con= ConexionDB.GetConnection();

String sql = "SELECT `Id\_Carta`,Ìd\_Bar` ,`Id\_Producto`, `Precio`, `Foto` FROM `carta` WHERE `Id\_Bar`=?";

try {

ps=con.prepareStatement(sql);

ps.setInt(1, id);

rs=ps.executeQuery();

if (rs.next()){

c = new Carta(rs.getInt(1),rs.getInt(2), rs.getInt(3), rs.getDouble(4),rs.getString(5));

}

rs.close();

ps.close();

} catch (SQLException ex) {

Logger.getLogger(ServiciosAppComanda.class.getName()).log(Level.SEVERE, null, ex);

}finally{

try {

con.close();

} catch (SQLException ex) {

Logger.getLogger(ServiciosAppComanda.class.getName()).log(Level.SEVERE, null, ex);

}

return c;

}

}

public boolean borrarCarta(int id){

boolean exito=false;

Connection con = null;

String sql ="DELETE FROM `appcomanda`.`carta` WHERE `comanda`.`Id\_Carta` = ?";

PreparedStatement ps;

try {

con = ConexionDB.GetConnection();

con.setAutoCommit(false);

ps= con.prepareStatement(sql);

ps.setInt(1, id);

ps.executeUpdate();

ps.close();

con.commit();

exito=true;

} catch (SQLException ex) {

Logger.getLogger(ConexionDB.class.getName()).log(Level.SEVERE, null, ex);

if (con!=null) try {

con.rollback();

} catch (SQLException ex1) {

Logger.getLogger(ServiciosAppComanda.class.getName()).log(Level.SEVERE, null, ex1);

}

}finally{

if (con !=null) try {

con.close();

} catch (SQLException ex) {

Logger.getLogger(ServiciosAppComanda.class.getName()).log(Level.SEVERE, null, ex);

}

return exito;

}

}

//--------------------------------------------OFERTA--------------------------------------------------

public boolean insertarOferta(int id\_bar, int id\_producto, double precio){

boolean exito=false;

Connection con = null;

String sql ="INSERT INTO `oferta`(`Id\_Bar`,`Id\_Producto`, `Precio`) VALUES (?,?,?)";

PreparedStatement ps;

try {

con = ConexionDB.GetConnection();

con.setAutoCommit(false);

ps= con.prepareStatement(sql);

ps.setInt(1, id\_bar);

ps.setInt(2, id\_producto);

ps.setDouble(3, precio);

ps.executeUpdate();

ps.close();

con.commit();

exito=true;

} catch (SQLException ex) {

Logger.getLogger(ConexionDB.class.getName()).log(Level.SEVERE, null, ex);

if (con!=null) try {

con.rollback();

} catch (SQLException ex1) {

Logger.getLogger(ServiciosAppComanda.class.getName()).log(Level.SEVERE, null, ex1);

}

}finally{

if (con !=null) try {

con.close();

} catch (SQLException ex) {

Logger.getLogger(ServiciosAppComanda.class.getName()).log(Level.SEVERE, null, ex);

}

return exito;

}

}

public boolean borrarOferta(int id\_producto, int id\_bar){

boolean exito=false;

Connection con = null;

String sql ="DELETE FROM `appcomanda`.`oferta` WHERE `comanda`.`Id\_Bar` = ? and `comanda`.`Id\_Producto` = ? ";

PreparedStatement ps;

try {

con = ConexionDB.GetConnection();

con.setAutoCommit(false);

ps= con.prepareStatement(sql);

ps.setInt(1, id\_bar);

ps.setInt(2, id\_producto);

ps.executeUpdate();

ps.close();

con.commit();

exito=true;

} catch (SQLException ex) {

Logger.getLogger(ConexionDB.class.getName()).log(Level.SEVERE, null, ex);

if (con!=null) try {

con.rollback();

} catch (SQLException ex1) {

Logger.getLogger(ServiciosAppComanda.class.getName()).log(Level.SEVERE, null, ex1);

}

}finally{

if (con !=null) try {

con.close();

} catch (SQLException ex) {

Logger.getLogger(ServiciosAppComanda.class.getName()).log(Level.SEVERE, null, ex);

}

return exito;

}

}

public List<Oferta> listaOferta(){

Connection con;

PreparedStatement ps = null;

ResultSet rs = null;

List<Oferta> lo = new Vector<Oferta>();

Oferta o = null;

con= ConexionDB.GetConnection();

String sql = "SELECT `Id\_Bar`,`Id\_Producto`,`Precio` FROM `oferta`";

try {

ps=con.prepareStatement(sql);

rs=ps.executeQuery();

while (rs.next()){

o = new Oferta(rs.getInt(1),rs.getInt(2), rs.getDouble(3));

lo.add(o);

}

rs.close();

ps.close();

} catch (SQLException ex) {

Logger.getLogger(ServiciosAppComanda.class.getName()).log(Level.SEVERE, null, ex);

}finally{

try {

con.close();

} catch (SQLException ex) {

Logger.getLogger(ServiciosAppComanda.class.getName()).log(Level.SEVERE, null, ex);

}

return lo;

}

}

//-------------------------------LINEA COMANDA -------------------------------------------------------------

public boolean insertarLineaComada(int id\_comanda, int id\_producto, int cantidad, int id\_bar){

boolean exito=false;

Connection con = null;

String sql ="INSERT INTO `linea\_comanda`(`Id\_Comanda`,`Id\_Producto`, `Cantidad`,`Id\_Bar`) VALUES (?,?,?,?)";

PreparedStatement ps;

try {

con = ConexionDB.GetConnection();

con.setAutoCommit(false);

ps= con.prepareStatement(sql);

ps.setInt(1, id\_comanda);

ps.setInt(2, id\_producto);

ps.setDouble(3, cantidad);

ps.setDouble(4, id\_bar);

ps.executeUpdate();

ps.close();

con.commit();

exito=true;

} catch (SQLException ex) {

Logger.getLogger(ConexionDB.class.getName()).log(Level.SEVERE, null, ex);

if (con!=null) try {

con.rollback();

} catch (SQLException ex1) {

Logger.getLogger(ServiciosAppComanda.class.getName()).log(Level.SEVERE, null, ex1);

}

}finally{

if (con !=null) try {

con.close();

} catch (SQLException ex) {

Logger.getLogger(ServiciosAppComanda.class.getName()).log(Level.SEVERE, null, ex);

}

return exito;

}

}

public boolean borrarLineaComanda(int id\_comanda, int id\_producto,int id\_bar){

boolean exito=false;

Connection con = null;

String sql ="DELETE FROM `appcomanda`.`linea\_comanda` WHERE `comanda`.`Id\_Comanda` = ? and `comanda`.`Id\_Producto` = ? and `comanda`.`Id\_Bar` = ? ";

PreparedStatement ps;

try {

con = ConexionDB.GetConnection();

con.setAutoCommit(false);

ps= con.prepareStatement(sql);

ps.setInt(1, id\_comanda);

ps.setInt(2, id\_producto);

ps.setInt(3, id\_bar);

ps.executeUpdate();

ps.close();

con.commit();

exito=true;

} catch (SQLException ex) {

Logger.getLogger(ConexionDB.class.getName()).log(Level.SEVERE, null, ex);

if (con!=null) try {

con.rollback();

} catch (SQLException ex1) {

Logger.getLogger(ServiciosAppComanda.class.getName()).log(Level.SEVERE, null, ex1);

}

}finally{

if (con !=null) try {

con.close();

} catch (SQLException ex) {

Logger.getLogger(ServiciosAppComanda.class.getName()).log(Level.SEVERE, null, ex);

}

return exito;

}

}

public List<lineaComanda> listaLineaComanda(){

Connection con;

PreparedStatement ps = null;

ResultSet rs = null;

List<lineaComanda> lc = new Vector<lineaComanda>();

lineaComanda c = null;

con= ConexionDB.GetConnection();

//String sql = "SELECT lc.Id\_Comanda, lc.Id\_Producto, lc.Cantidad ,lc.Id\_Bar, p.Descripcion FROM linea\_comanda as lc join Producto as p on lc.Id\_Producto = p.Id\_Producto Group by lc.Id\_Comanda, lc.Id\_Producto,lc.Id\_Bar";

String sql = "SELECT lc.Id\_Comanda, lc.Id\_Producto, lc.Cantidad ,lc.Id\_Bar, p.Descripcion FROM linea\_comanda as lc join Producto as p on lc.Id\_Producto = p.Id\_Producto";

try {

ps=con.prepareStatement(sql);

rs=ps.executeQuery();

while (rs.next()){

c = new lineaComanda(rs.getInt(1),rs.getInt(2), rs.getInt(3), rs.getInt(4),rs.getString(5));

lc.add(c);

}

rs.close();

ps.close();

} catch (SQLException ex) {

Logger.getLogger(ServiciosAppComanda.class.getName()).log(Level.SEVERE, null, ex);

}finally{

try {

con.close();

} catch (SQLException ex) {

Logger.getLogger(ServiciosAppComanda.class.getName()).log(Level.SEVERE, null, ex);

}

return lc;

}

}

}