UNIVERSIDAD TÉCNICA DE ORURO

FACULTAD NACIONAL DE INGENIERÍA

INGENIERÍA INF-SIS

**Construcción del Sistema de control de Ventas de la “Tienda X”**

Autor: Colque



Contenido

[**CAPÍTULO I** 3](#_Toc7284312)

[**1.** **INTRODUCCIÓN** 3](#_Toc7284313)

[**2.** **DESCRIPCIÓN DE LA ORGANIZACIÓN Y SITUACIÓN ACTUAL** 3](#_Toc7284314)

[**Aspectos Generales de la Empresa** 3](#_Toc7284315)

[**Antecedentes** 3](#_Toc7284316)

[**Misión** 3](#_Toc7284317)

[**Visión** 3](#_Toc7284318)

[**Descripción de la Situación Actual** 3](#_Toc7284319)

[**3.** **PROBLEMAS DE LA SITUACIÓN ACTUAL** 4](#_Toc7284320)

[**Oportunidad** 4](#_Toc7284321)

[**4.** **DEFINICIÓN DEL PROYECTO** 4](#_Toc7284322)

[**Solución Planteada** 4](#_Toc7284323)

[**5.** **OBJETIVOS** 4](#_Toc7284324)

[**Objetivo General** 4](#_Toc7284325)

[**Objetivos Específicos** 4](#_Toc7284326)

[**6.** **LIMITACIONES DEL PROYECTO** 5](#_Toc7284327)

[**CAPITULO II Diseño de la base de datos** 5](#_Toc7284328)

[**2.1 Descripción (Univ. Del Discurso)** 5](#_Toc7284329)

[**2.2 Diagrama entidad relación** 5](#_Toc7284330)

[**2.3 Modelo relacional** 5](#_Toc7284331)

[**2.3.1 Modelo relacional global** 5](#_Toc7284332)

[**2.3.2 1FN** 5](#_Toc7284333)

[**2.3.2 2FN** 5](#_Toc7284334)

[**2.3.2 3FN** 5](#_Toc7284335)

[**CAPITULO III Implementación** 5](#_Toc7284336)

[**3.1 tablas (esquemas)** 5](#_Toc7284337)

[**3.2 Consultas** 5](#_Toc7284338)

[**3.3 Procedimientos** 5](#_Toc7284339)

[**3.4 Disparadores** 5](#_Toc7284340)

[**CAPITULO IV Salidas** 5](#_Toc7284341)

[**4.1 Listados** 5](#_Toc7284342)

[**4.2 Repostes** 5](#_Toc7284343)

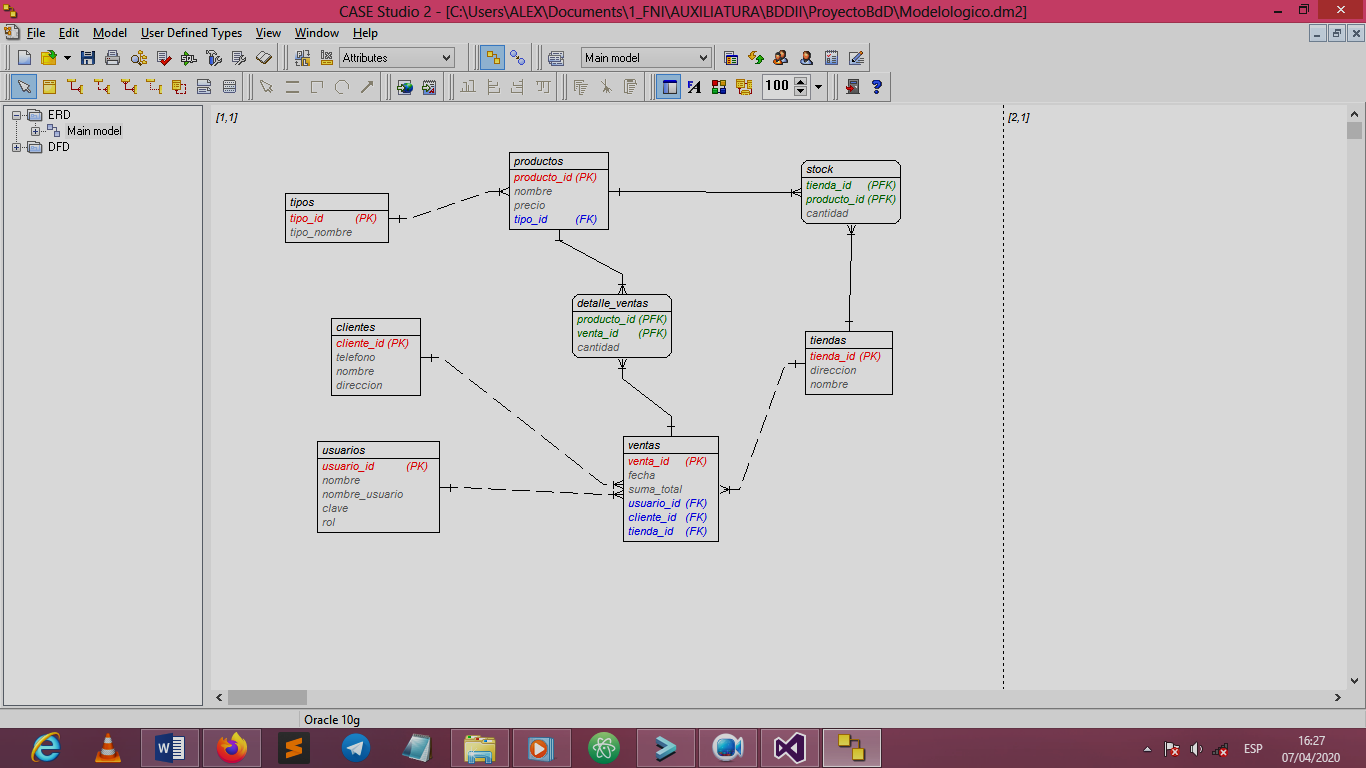
[**4.3 Gráficos** 5](#_Toc7284344)

[**Bibliografía** 5](#_Toc7284345)

# **CAPITULO I Implementación**

## **1 tablas (esquemas)**

**Modelo lógico.**

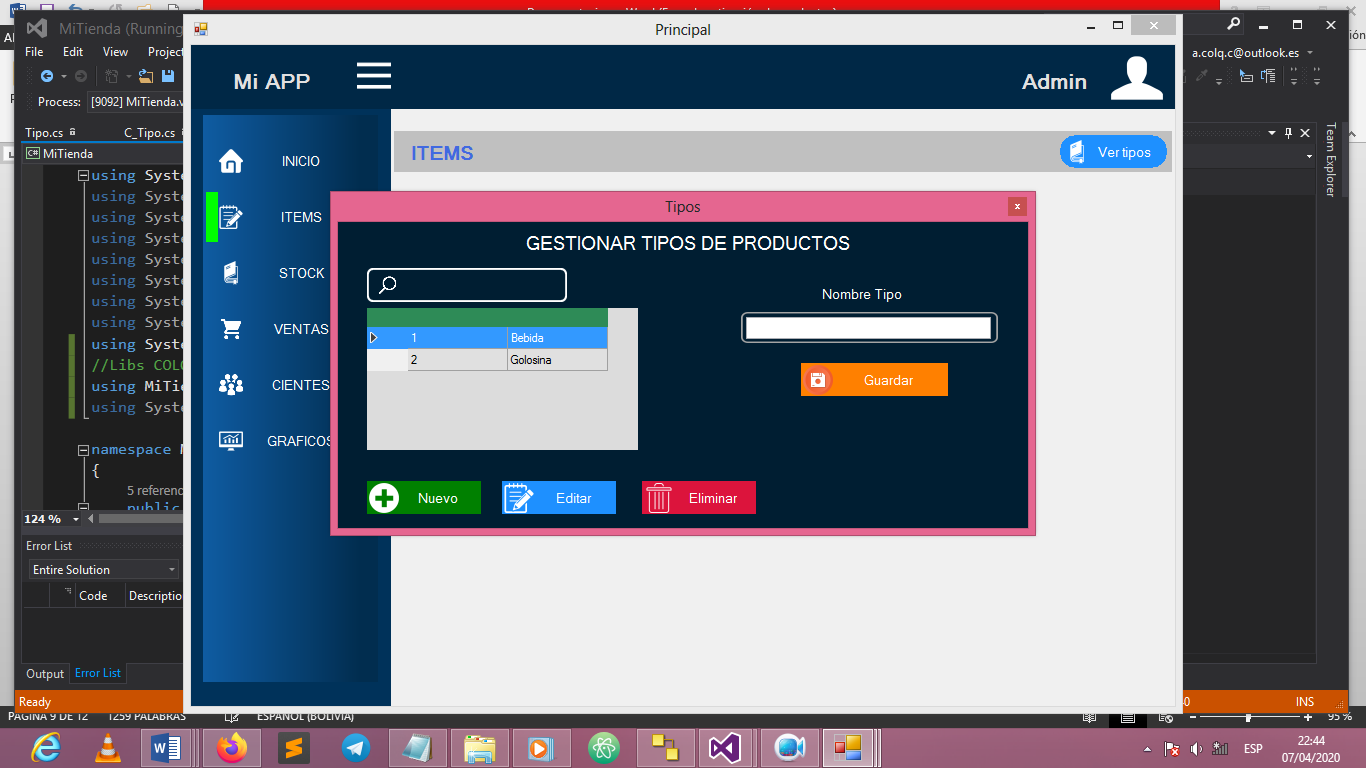


## **Código para crear la base de datos**

Crear la base de datos

|  |
| --- |
| drop table usuarios  CREATE TABLE usuarios(  usuario\_id number(8) NOT NULL,  nombre varchar2(50),  nombre\_usuario varchar2(30),  clave varchar2(30),  rol number(2),  PRIMARY KEY (usuario\_id)  )  INSERT ALL  INTO usuarios VALUES(1,'Admin','admin','admin',1)  INTO usuarios VALUES(2,'Admin2','admin2','admin2',2)  SELECT \* FROM dual;  select \* from usuarios  CREATE TABLE tipos(  tipo\_id number(8) NOT NULL,  tipo\_nombre varchar2(30),  PRIMARY KEY (tipo\_id)  )  INSERT ALL  INTO tipos VALUES(1,'Bebida')  INTO tipos VALUES(2,'Golosina')  SELECT \* FROM dual;  select \* from tipos  CREATE TABLE productos(  producto\_id number(8) NOT NULL,  nombre varchar2(50),  precio number(8,2),  tipo\_id number(8),  PRIMARY KEY (producto\_id),  FOREIGN KEY (tipo\_id) REFERENCES tipos  )  INSERT ALL  INTO productos VALUES(1,'Cocacola',8.5,1)  INTO productos VALUES(2,'Bonbonbun',10.5,2)  SELECT \* FROM dual;  select \* from productos  CREATE TABLE clientes(  cliente\_id number(8) NOT NULL,  nombre varchar2(50),  telefono varchar2(30),  direccion varchar2(50),  PRIMARY KEY (cliente\_id)  )  INSERT ALL  INTO clientes VALUES(1,'Juan','61454552','Calle bal #3')  INTO clientes VALUES(2,'Sam','60000060','Calle hulk #4')  SELECT \* FROM dual;  select \* from clientes  CREATE TABLE tiendas(  tienda\_id number(8) NOT NULL,  nombre varchar2(50),  direccion varchar2(50),  PRIMARY KEY (tienda\_id)  )  INSERT ALL  INTO tiendas VALUES(1,'Juan de Dios','Plaza miel #323')  INTO tiendas VALUES(2,'La Estrella','Calle Bolivar #423')  SELECT \* FROM dual;  select \* from tiendas  CREATE TABLE ventas(  venta\_id number(8) NOT NULL,  fecha date,  suma\_total number(8,2),  usuario\_id number(8),  cliente\_id number(8),  tienda\_id number(8),  PRIMARY KEY (venta\_id),  FOREIGN KEY (usuario\_id) REFERENCES usuarios,  FOREIGN KEY (cliente\_id) REFERENCES clientes,  FOREIGN KEY (tienda\_id) REFERENCES tiendas  )  INSERT ALL  INTO ventas VALUES(1,'1/1/2020',200.5,1,1,1)  SELECT \* FROM dual;  select \* from ventas  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*PFK\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  CREATE TABLE stock(  tienda\_id number(8) NOT NULL,  producto\_id number(8) NOT NULL,  cantidad number(8),  PRIMARY KEY (tienda\_id, producto\_id),  FOREIGN KEY (tienda\_id) REFERENCES tiendas,  FOREIGN KEY (producto\_id) REFERENCES productos  )  INSERT ALL  INTO stock VALUES(1,1,50)  INTO stock VALUES(1,2,100)  SELECT \* FROM dual;  select \* from stock  CREATE TABLE detalle\_ventas(  venta\_id number(8) NOT NULL,  producto\_id number(8) NOT NULL,  cantidad number(8),  PRIMARY KEY (venta\_id, producto\_id),  FOREIGN KEY (venta\_id) REFERENCES ventas,  FOREIGN KEY (producto\_id) REFERENCES productos  )  INSERT ALL  INTO detalle\_ventas VALUES(1,1,5)  INTO detalle\_ventas VALUES(1,2,5)  SELECT \* FROM dual;  select \* from detalle\_ventas |

## **CÓDIGO DE CRUD PARA LA TABLA TIPOS**



## MODELO

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

//Referencias

using Oracle.DataAccess.Client;

using System.Data;

using System.Configuration;

using System.Windows.Forms;

using MiTienda.Controller;

namespace MiTienda.Model

{

public class Tipo

{

internal static OracleConnection conn = null;

private void abrirConexion()

{

string conexionstring = ConfigurationManager.ConnectionStrings["conexion"].ConnectionString;

conn = new OracleConnection(conexionstring);

try

{

conn.Open();

//MessageBox.Show("Conexion establecida");

}

catch (Exception ex)

{

MessageBox.Show("Error de conexion" + ex);

Application.Exit();

throw new Exception("Error !!!");

}

}

public List<C\_Tipo> ListarTipo(String lista)

{

List<C\_Tipo> Listar = new List<C\_Tipo>();

using (OracleCommand command = new OracleCommand())

{

StringBuilder Query = new StringBuilder();

abrirConexion();

Query.Append("SELECT tipo\_id, tipo\_nombre FROM tipos WHERE tipo\_nombre LIKE '"+lista+"%'");

command.CommandType = System.Data.CommandType.Text;

command.CommandText = Query.ToString();

OracleDataReader reader = null;

command.Connection = conn;

reader = command.ExecuteReader();

//List<C\_Tipo> Listar = new List<C\_Tipo>();

while (reader.Read())

{

Listar.Add(new C\_Tipo

{

Idtipo = reader.GetInt32(0),

Nombre = reader.GetString(1)

});

}

conn.Close();

reader.Close();

}

return Listar;

}

public void Insertar(C\_Tipo Tipo)

{

try

{

abrirConexion();

OracleCommand cmd1 = new OracleCommand("SELECT COUNT(\*) FROM tipos", conn);

string id = (cmd1.ExecuteScalar()).ToString();

if (Convert.ToInt32(id) > 0)

{

OracleCommand idmax = new OracleCommand("SELECT MAX(tipo\_id) FROM tipos", conn);

string \_id = (idmax.ExecuteScalar()).ToString();

OracleCommand cmd = new OracleCommand("INSERT INTO tipos VALUES ('" + (Convert.ToInt32(\_id) + 1) + "', '" + Tipo.Nombre + "')", conn);

cmd.ExecuteNonQuery();

}

else

{

OracleCommand cmd3 = new OracleCommand("INSERT INTO tipos VALUES ('" + 1 + "', '" + Tipo.Nombre + "')", conn);

cmd3.ExecuteNonQuery();

}

conn.Close();

}

catch (Exception ex)

{

MessageBox.Show("Error al registrar Tipo " + ex);

throw new Exception("Error !!!");

}

}

public void Editar(C\_Tipo Tipo)

{

try

{

abrirConexion();

OracleCommand cmd = new OracleCommand("UPDATE tipos SET tipo\_nombre = '" + Tipo.Nombre + "'WHERE tipo\_id = '" + Tipo.Idtipo + "'", conn);

cmd.ExecuteNonQuery();

conn.Close();

}

catch (Exception ex)

{

MessageBox.Show("Error al Actualizar Tipo" + ex);

throw new Exception("Error !!!");

}

}

public void Eliminar(C\_Tipo Tipo)

{

try

{

abrirConexion();

OracleCommand cmd = new OracleCommand("DELETE FROM tipos WHERE tipo\_id = '" + Tipo.Idtipo + "'", conn);

if (MessageBox.Show("¿Está seguro que que desea eliminar '" + Tipo.Nombre + "'?", "¡Advertencia!", MessageBoxButtons.YesNo, MessageBoxIcon.Warning) == DialogResult.Yes)

{

cmd.ExecuteNonQuery();

}

conn.Close();

}

catch (Exception ex)

{

MessageBox.Show("Error al Eliminar tipos" + ex);

throw new Exception("Error !!!");

}

}

}

}

## CONTROLADOR

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

//Referencias

using MiTienda.Model;

namespace MiTienda.Controller

{

public class C\_Tipo

{

private int \_Idtipo;

private string \_Nombre;

public int Idtipo

{

get

{

return \_Idtipo;

}

set

{

\_Idtipo = value;

}

}

public string Nombre

{

get

{

return \_Nombre;

}

set

{

\_Nombre = value;

}

}

//Instanciar

Tipo obj = new Tipo();

//Metodos de CRUD

public List<C\_Tipo> Listado(string buscar)

{

return obj.ListarTipo(buscar);

}

public void Insertar(C\_Tipo Tipo)

{

obj.Insertar(Tipo);

}

public void Editar(C\_Tipo Tipo)

{

obj.Editar(Tipo);

}

public void Eliminar(C\_Tipo Tipo)

{

obj.Eliminar(Tipo);

}

}

}

## VISTA

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

//Libs COLQUE

using MiTienda.Controller;

using System.Configuration;

namespace MiTienda.View.Dialogs

{

public partial class Tipos : Form

{

private string id;//Ayuda Editar y eliminar

private bool editarse = false;

//Invocando a las clases claves

C\_Tipo objC = new C\_Tipo();

public Tipos()

{

InitializeComponent();

datostabla("");

}

public void AccionesTabla()

{

dgvTipo.Columns[0].Visible = false;

dgvTipo.Columns[1].Width = 60;

dgvTipo.Columns[2].Width = 150;

dgvTipo.ClearSelection();

}

private void limpiar()

{

editarse = false;

txtTipo.Text = "";

txtTipo.Focus();

}

private void datostabla(string buscar)

{

C\_Tipo obj = new C\_Tipo();

dgvTipo.DataSource = obj.Listado(buscar);

}

private void btnNuevo\_Click(object sender, EventArgs e)

{

limpiar();

}

private void btnEditar\_Click(object sender, EventArgs e)

{

if (dgvTipo.SelectedRows.Count > 0)

{

editarse = true;

id = dgvTipo.CurrentRow.Cells[0].Value.ToString();

txtTipo.Text = dgvTipo.CurrentRow.Cells[1].Value.ToString();

}

else

{

MessageBox.Show("Seleccione la tabla que desea editar");

}

}

private void btnEliminar\_Click(object sender, EventArgs e)

{

if (dgvTipo.SelectedRows.Count > 0)

{

objC.Idtipo = Convert.ToInt32(dgvTipo.CurrentRow.Cells[0].Value.ToString());

objC.Eliminar(objC);

datostabla("");

}

else

{

MessageBox.Show("Seleccione la fila que desee eliminar ");

}

}

private void btnGuardar\_Click(object sender, EventArgs e)

{

if (txtTipo.Text != "")

{

if (editarse == false)

{

try

{

objC.Nombre = txtTipo.Text;

objC.Insertar(objC);

MessageBox.Show("Se guardó el registro ");

datostabla("");

limpiar();

}

catch (Exception ex)

{

MessageBox.Show("No se pudo guardar el registro " + ex);

}

}

if (editarse == true)

{

try

{

objC.Idtipo = Convert.ToInt32(id);

objC.Nombre = txtTipo.Text;

objC.Editar(objC);

MessageBox.Show("Se guardó el registro ");

datostabla("");

limpiar();

editarse = false;

}

catch (Exception ex)

{

MessageBox.Show("No se pudo editar el registro " + ex);

}

}

}

else

{

MessageBox.Show("Llene los campos por favor");

}

}

private void txtBuscar\_TextChanged(object sender, EventArgs e)

{

datostabla(txtBuscar.Text);

}

}

}

## **3.2 Consultas**

## **3.3 Procedimientos**

|  |
| --- |
|  |

## **3.4 Disparadores**

|  |
| --- |
|  |

# **CAPITULO IV Salidas**

## **4.1 Listados**

## **4.2 Reportes**

## **4.3 Gráficos**