Codificacion de "Sistema de Gestión de Pedido”.

Para el desarrollo de la aplicación de escritorio se utiliza la herramienta Visual estudio 2010. La codificación se realiza con lenguajes de c#. A continuación se muestra el código desarrollado:

## Estructura de paquetes: a continuación se presenta la distribución de paquetes en realizados en la codificación.

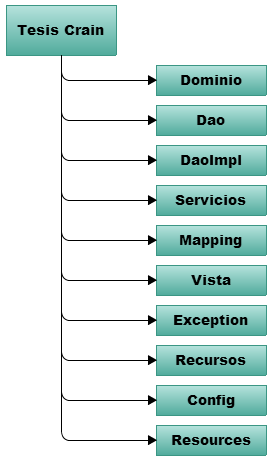


Figura x.x: Diagrama de paquetes de sistema Crain

Paquetes: A continuación se definen el contenido de cada uno de los paquetes:

**Dominio**: clases de entidades identificadas en el sistema. Contiene las clases:

* Administrador
* Categoria
* Diario
* Menu
* Mesa
* Pedido
* Reporte
* TeaMenuPedido
* Video

**Dao**: son interfaces las cuales implementaran las clases. Las interfaces son:

* AdministradorDao
* CategoriaDao
* DiarioDao
* MenuDao
* MesaDao
* PedidoDao
* ReporteDao
* TeaMenuPedidoDao
* VideoDao

**DaoImpl**: contiene las clases que implementan las interfaces definidas en el paquete dao. Las clases son:

* AdministradorDaoImpl
* CategoriaDaoImpl
* DiarioDaoImpl
* MenuDaoImpl
* MesaDaoImpl
* PedidoDaoImpl
* ReporteDaoImpl
* TeaMenuPedidoDaoImpl
* VideoDaoImpl

**Exception**: clases que manejarán los errores.

* ValorNumericoException

**Servicios**: se implementa el patrón facade

* Facade

**Mapping**: se encuentran los archivos de mapeo de hibernate.

* administrador
* categoria
* diario
* menu
* mesa
* pedido
* reporte
* teamenupedido
* video

**Vista**: contiene las pantallas del sistema

* Login
* PantallaPrincipal
* AgregarDiario
* AgregarVideo
* ModificarPrecio

**Recursos:** archivos adicionales necesarios en visual estudio para lograr el mapeo.

* nhibernate-configuration.xsd
* nhibernate-mapping.xdsd

**Config:** contiene los archivos que permiten el **mapeo** entre entidades de la base de datos y los objetos de la aplicación.

* hibernate.cfg.xml
* NHibernateManager

**Resources:**  contiene las imagenes utilizadas en el sistema.

* numerMesa2-2.png
* textField2b.png

Elementos del dominio

Clase Menu:

**public** class Menu

**{**

**private** int idMenu**;**

**private** string nombrePlatoMenu**;**

**private** double precioMenu**;**

**private** string descripcionMenu**;**

**private** ISet teaMenuPedidoMap **=** **new** HashedSet**();**

**private** ISet complementoMap **=** **new** HashedSet**();**

**private** Categoria categoria**;**

**public** Menu**()** **{** **}**

**public** Menu**(**int idMenu**,** string nombrePlatoMenu**,** double precioMenu**,** string descripcionMenu**,** Categoria categoria**)** **{**

**this.**idMenu **=** idMenu**;**

**this.**nombrePlatoMenu **=** nombrePlatoMenu**;**

**this.**precioMenu **=** precioMenu**;**

**this.**descripcionMenu **=** descripcionMenu**;**

**this.**categoria **=** categoria**;**

**}**

**}**

Clase Pedido:

**public** class Pedido

**{**

**private** int idPedido**;**

**private** string fechaPedido**;**

**private** double totalPedido**;**

**private** Mesa mesa**;**

**private** ISet teaMenuPedidoMap **=** **new** HashedSet**();**

**public** Pedido**()** **{** **}**

**public** Pedido**(**int idPedido**,** string fechaPedido**,** double totalPedido**,** Mesa mesa**)** **{**

**this.**idPedido **=** idPedido**;**

**this.**fechaPedido **=** fechaPedido**;**

**this.**totalPedido **=** totalPedido**;**

**this.**mesa **=** mesa**;**

**}**

**}**

Elementos de Dao:

Interfaz MenuDao

public interface MenuDao

{

void guardarMenu(Menu menu);

IList<Menu> getAllMenu();

void modificarMenu(Menu menu);

}

Interfaz PedidoDao

public interface PedidoDao

{

void guardarPedido(Pedido pedido);

IList<Pedido> getAllPedido();

}

Elementos de DaoImpl:

Clase MenuDaoImpl

public class MenuDaoImpl:MenuDao

{

public void guardarMenu(Menu menu) {

NHibernateManager manager = new NHibernateManager();

{

using (ITransaction transaction = manager.Session.BeginTransaction())

{

manager.Session.Save(menu);

transaction.Commit();

}

}

}

public void modificarMenu(Menu menu) {

NHibernateManager manager = new NHibernateManager();

{

using (ITransaction transaction = manager.Session.BeginTransaction())

{

manager.Session.Update(menu);

transaction.Commit();

}

}

}

public IList<Menu> getAllMenu() {

IList<Menu> lista = null;

NHibernateManager manager = new NHibernateManager();

using (ISession session = manager.Session)

{

lista = session.CreateCriteria(typeof(Menu)).List<Menu>();

session.Close();

}

return lista;

}

}

ClasePedidoDaoImpl

public class PedidoDaoImpl : PedidoDao

{

public void guardarPedido(Pedido pedido)

{

NHibernateManager manager = new NHibernateManager();

{

using (ITransaction transaction = manager.Session.BeginTransaction())

{

manager.Session.Save(pedido);

transaction.Commit();

}

}

}

public IList<Pedido> getAllPedido()

{

IList<Pedido> lista = null;

NHibernateManager manager = new NHibernateManager();

using (ISession session = manager.Session)

{

lista = session.CreateCriteria(typeof(Pedido)).List<Pedido>();

session.Close();

}

return lista;

}

}

Elementos de Mapping:

Mapeo de Menu

<class name="Menu" table="menu" lazy="true" >

<id name="IdMenu" column="MENU\_ID" type="int" unsaved- value="0">

<generator class="native"/>

</id>

<property name="NombrePlatoMenu" column="MENU\_NOMBRE\_PLATO" type="String">

</property>

<property name="PrecioMenu" column="MENU\_PRECIO" type="double">

</property>

<property name="DescripcionMenu" column="MENU\_DESCRIPCION" type="String">

</property>

<many-to-one name="Categoria" class="Categoria" fetch="join" lazy="false">

<column name="CATEGORIA\_ID" />

</many-to-one>

<set name="ComplementoMap" table="COMPLEMENTO" inverse="true" lazy="false" fetch="select">

<key>

<column name="MENU\_ID" />

</key>

<one-to-many class="ComplementoMenu" />

</set>

<set name="TeaMenuPedidoMap" table="tea\_menu\_pedido" inverse="true" lazy="false" fetch="select">

<key>

<column name="Menu\_ID" />

</key>

<one-to-many class="TeaMenuPedido" />

</set>

</class>

</hibernate-mapping>

Mapeo de Pedido

<class name="Pedido" table="pedido" lazy="true" >

<id name="IdPedido" column="PEDIDO\_ID" type="int" unsaved- value="0">

<generator class="native"/>

</id>

<property name="FechaPedido" column="PEDIDO\_FECHA" type="String">

</property>

<property name="TotalPedido" column="PEDIDO\_TOTAL" type="double">

</property>

<many-to-one name="Mesa" class="Mesa" fetch="join" lazy="false">

<column name="MESA\_ID" />

</many-to-one>

<set name="TeaMenuPedidoMap" table="tea\_menu\_pedido" inverse="true" lazy="false" fetch="select">

<key>

<column name="PEDIDO\_ID" />

</key>

<one-to-many class="TeaMenuPedido" />

</set>

</class>

</hibernate-mapping>

Elemento de Servicio:

Facade

class Facade

{

...

MenuDaoImpl menuDaoImpl = new MenuDaoImpl();

PedidoDaoImpl pedidoDaoImpl = new PedidoDaoImpl();

...

public Facade() { }

...

//Servicios de Menu

public IList<Menu> getAllMenu()

{

return menuDaoImpl.getAllMenu();

}

public void modificarMenu(Menu menu)

{

this.menuDaoImpl.modificarMenu(menu);

}

// Servicio de Pedido

public IList<Pedido> getAllPedido()

{

return pedidoDaoImpl.getAllPedido();

}

...

}

}

Elementos de Vista:

Pantalla Login

public partial class Login : Form

{ Facade facade = new Facade();

private bool bandera = false;

public Login()

{

InitializeComponent();

}

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

{

if (facade.verificarAdministrador(textBoxUsuario.Text, textBoxPassword.Text))

{ bandera = true;

}

else

{ MessageBox.Show("USUARIO O CONTRASEÑA INCORRECTA");

}

if (bandera)

{

this.DialogResult = DialogResult.OK;

}

}

private void Login\_Load(object sender, EventArgs e)

{ }

private void label2\_Click(object sender, EventArgs e)

{ }

private void label1\_Click(object sender, EventArgs e)

{ }

private void label3\_Click(object sender, EventArgs e)

{ }

}