

**Alumno:** Luis Alberto Mancillas Robles

**Profesor:** Lic. Cristian Bernal Ramírez

**Asignatura**: Calidad en el desarrollo de software.

**Unidad:** I.

**Tema:** Práctica sobre PSP0

**Grupo y grado:** TIC51 TSU

**Cuatrimestre:** 5

**Lugar y fecha:** Playa del Carmen, Quintana Roo,

miércoles 15 de febrero del 2016

**Planeación**

**Objetivo**

Gestionar los datos del cliente y acomodarlos por categorías por tipos de cliente.

**Alcance**

La presente aplicación contempla procedimientos relacionados a la ejecución de las distintas fases del proceso de desalación por ósmosis inversa, describiendo la mejor solución para cada tipo de instalación a diseñar.

**Requerimientos**

**Requisitos funcionales:**

* Nombres y apellidos son obligatorios.
* No poder eliminar una categoría, si está relacionada con el cliente.
* El nombre de la categoría no puede quedar vacío.

**Requisitos no funcionales:**

* Corre en la línea de sistemas operativos Windows.
* Requiere tener instalado MySQL Server para la base de datos.

**Diseño**

**Diseño de base de datos**

Categorías

* Id
* Nombre

Clientes

* Id
* Nombre
* Apellidos
* Teléfono
* Correo\_electronico
* Categoria\_id

**Definición de líneas de código**

**Convención de nombres**

**Nombres de archivos**

* persona.cs
* cliente.cs

**Nombres de clases**

* Categorías : Form
* Clientes: Form

**Codigo**

**Clientes.cs:**

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;

using MySql.Data.MySqlClient;

namespace TI51\_LAMR\_AplicacionClientes

{

public partial class Clientes : Form

{

public Clientes()

{

//Se inicia el form.

InitializeComponent();

}

MySqlConnection conexion = new MySqlConnection("server=127.0.0.1; database=AplicacionClientes; Uid=root; pwd=root;"); //Cadena de conexion de la base de datos

DataSet dataset; //Definicion del dataset

private void button4\_Click(object sender, EventArgs e) //Boton para listar

{

try

{

//Se conecta a la base de datos de MySQL y se utiliza el comando de seleccionar.

conexion.Open();

MySqlCommand seleccionar = new MySqlCommand("select \* from clientes", conexion);

MySqlDataAdapter conn = new MySqlDataAdapter(seleccionar);

dataset = new DataSet();

conn.Fill(dataset);

dataGridView1.DataSource = dataset.Tables[0];

conexion.Close();

}

catch (Exception)

{

{

MySqlCommand seleccionar1 = new MySqlCommand("select \* from clientes", conexion);

MySqlDataAdapter conn = new MySqlDataAdapter(seleccionar1);

dataset = new DataSet();

conn.Fill(dataset);

dataGridView1.DataSource = dataset.Tables[0];

conexion.Close();

}

}

}

private void button6\_Click(object sender, EventArgs e) //Boton para seleccionar (es necesario primero seleccionar el dato en el DataGridView)

{

try

{

textBox1.Text = this.dataGridView1.CurrentRow.Cells[1].Value.ToString();

textBox2.Text = this.dataGridView1.CurrentRow.Cells[2].Value.ToString();

textBox3.Text = this.dataGridView1.CurrentRow.Cells[3].Value.ToString();

textBox4.Text = this.dataGridView1.CurrentRow.Cells[4].Value.ToString();

textBox5.Text = this.dataGridView1.CurrentRow.Cells[5].Value.ToString();

}

catch (Exception)

{

}

}

private void button1\_Click(object sender, EventArgs e) //Boton de insertar

{

try

{ //Se define las cadena nombre y apellido para referencia al TextBox donde se llenara el campo "nombre" y "apellido" del formulario.

string nombre;

nombre = textBox1.Text;

string apellidos;

apellidos = textBox2.Text;

//Se necesita llenar el campo, si no, se publicara nada en el DataGridView.

if (nombre == "")

{

MessageBox.Show("Debes de llenar el campo 'Nombre");

}

//Se necesita llenar el campo, si no, se publicara nada en el DataGridView.

else if (apellidos == "")

{

MessageBox.Show("Debes de llenar el campo 'Apellidos'");

}

else

{

//Se conecta a la base de datos de MySQL y se utiliza el comando de insertar.

conexion.Open();

MySqlCommand añadir = new MySqlCommand("insert into clientes(nombre,apellidos,telefono,correo\_electronico,categoriaid) values ('" + textBox1.Text + "','" + textBox2.Text + "','" + textBox3.Text + "','" + textBox4.Text + "','" + textBox5.Text + "')", conexion);

añadir.ExecuteNonQuery();

//Se borra los textBox automaticamente despues de hacer la insercion.

MessageBox.Show("Ingresado correctamente");

textBox1.Clear();

textBox2.Clear();

textBox3.Clear();

textBox4.Clear();

textBox5.Clear();

}

}

catch (Exception)

{

}

}

private void button2\_Click(object sender, EventArgs e) //Boton para modificar.

{

try

{

//Se define las cadena nombre y apellido para referencia al TextBox donde se llenara el campo "nombre" y "apellido" del formulario.

string nombre;

nombre = textBox1.Text;

string apellidos;

apellidos = textBox2.Text;

//Se necesita llenar el campo, si no, se publicara nada en el DataGridView.

if (nombre == "")

{

MessageBox.Show("Debes de llenar el campo 'Nombre");

}

//Se necesita llenar el campo, si no, se publicara nada en el DataGridView.

else if (apellidos == "")

{

MessageBox.Show("Debes de llenar el campo 'Apellidos'");

}

else

{

//Se conecta a la base de datos de MySQL y se utiliza el comando de modificar.

conexion.Open();

MySqlCommand modificar = new MySqlCommand("update clientes set nombre='" + textBox1.Text + "',apellidos='" + textBox2.Text + "',telefono='" + textBox3.Text + "',correo\_electronico='" + textBox4.Text + "',categoriaid='" + textBox5.Text + "' where id=" + dataGridView1.CurrentRow.Cells[0].Value.ToString(), conexion);

modificar.ExecuteNonQuery();

MessageBox.Show("Modificado correctamente");

textBox1.Clear();

textBox2.Clear();

textBox3.Clear();

textBox4.Clear();

textBox5.Clear();

conexion.Close();

}

}

catch (Exception)

{

}

}

private void button3\_Click(object sender, EventArgs e) //Boton para eliminar.

{

try

{

//Se conecta a la base de datos de MySQL y se utiliza el comando de eliminar.

conexion.Open();

MySqlCommand eliminar= new MySqlCommand("delete from clientes where id=" + dataGridView1.CurrentRow.Cells[0].Value.ToString(), conexion);

eliminar.ExecuteNonQuery();

MessageBox.Show("Eliminado correctamente.");

conexion.Close();

textBox1.Clear();

textBox2.Clear();

textBox3.Clear();

textBox4.Clear();

textBox5.Clear();

}

catch

{

}

}

private void button5\_Click(object sender, EventArgs e) //Boton para ir a form de Categorias.

{

Categorias frm = new Categorias();

frm.Show();

}

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

{

}

private void label10\_Click(object sender, EventArgs e)

{

}

private void button7\_Click(object sender, EventArgs e)

{

}

private void groupBox2\_Enter(object sender, EventArgs e)

{

}

private void dataGridView1\_CellContentClick(object sender, DataGridViewCellEventArgs e)

{

}

private void textBox6\_TextChanged(object sender, EventArgs e)

{

}

private void comboBox1\_SelectedIndexChanged(object sender, EventArgs e)

{

}

}

}

**Categorias.cs:**

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;

using MySql.Data.MySqlClient;

namespace TI51\_LAMR\_AplicacionClientes

{

public partial class Categorias : Form

{

public Categorias()

{

//Se inicia el form.

InitializeComponent();

}

MySqlConnection conexion = new MySqlConnection("server=127.0.0.1; database=AplicacionClientes; Uid=root; pwd=root;"); //Cadena de conexion de la base de datos

DataSet dataset; //Definicion del dataset

private void button1\_Click(object sender, EventArgs e) //Boton para seleccionar (es necesario primero seleccionar el dato en el DataGridView)

{

try

{

textBox1.Text = this.dataGridView1.CurrentRow.Cells[1].Value.ToString();

}

catch (Exception)

{

}

}

private void button2\_Click(object sender, EventArgs e) //Boton para listar.

{

try

{

//Se conecta a la base de datos de MySQL y se utiliza el comando de seleccionar.

conexion.Open();

MySqlCommand seleccionar = new MySqlCommand("select \* from categorias", conexion);

MySqlDataAdapter conn = new MySqlDataAdapter(seleccionar);

dataset = new DataSet();

conn.Fill(dataset);

dataGridView1.DataSource = dataset.Tables[0];

conexion.Close();

}

catch (Exception)

{

MySqlCommand seleccionar1 = new MySqlCommand("select \* from categorias", conexion);

MySqlDataAdapter conn = new MySqlDataAdapter(seleccionar1);

dataset = new DataSet();

conn.Fill(dataset);

dataGridView1.DataSource = dataset.Tables[0];

conexion.Close();

}

}

private void button3\_Click(object sender, EventArgs e) //Boton para ir al form de Clientes.

{

Clientes frm = new Clientes();

frm.Show();

}

private void button4\_Click(object sender, EventArgs e) //Boton para insertar.

{

try

{

//Se conecta a la base de datos de MySQL y se utiliza el comando de insertar.

conexion.Open();

MySqlCommand insertar1 = new MySqlCommand("insert into categorias(nombre) values ('" + textBox1.Text + "')", conexion);

insertar1.ExecuteNonQuery();

MessageBox.Show("Ingresado correctamente.");

textBox1.Clear();

conexion.Close();

}

catch (Exception error)

{

MessageBox.Show("Error.." + error.Message);

conexion.Close();

}

}

private void button5\_Click(object sender, EventArgs e) //Boton para modificar.

{

try

{

//Se conecta a la base de datos de MySQL y se utiliza el comando de modificar.

conexion.Open();

MySqlCommand modificar1 = new MySqlCommand("update categorias set nombre='" + textBox1.Text + "' where id=" + dataGridView1.CurrentRow.Cells[0].Value.ToString(), conexion);

modificar1.ExecuteNonQuery();

MessageBox.Show("Modificado correctamente!");

textBox1.Clear();

conexion.Close();

}

catch (Exception)

{

}

}

private void button6\_Click(object sender, EventArgs e) //Boton para eliminar.

{

try

{

//Se conecta a la base de datos de MySQL y se utiliza el comando de eliminar.

conexion.Open();

MySqlCommand eliminar1 = new MySqlCommand("delete from categorias where id=" + dataGridView1.CurrentRow.Cells[0].Value.ToString(), conexion);

eliminar1.ExecuteNonQuery();

MessageBox.Show("Eliminado correctamente.");

conexion.Close();

textBox1.Clear();

}

catch (Exception)

{

}

}

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

{

}

private void groupBox1\_Enter(object sender, EventArgs e)

{

}

private void textBox1\_TextChanged(object sender, EventArgs e)

{

}

private void button7\_Click(object sender, EventArgs e)

{

}

private void dataGridView1\_CellContentClick(object sender, DataGridViewCellEventArgs e)

{

}

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

{

}

}

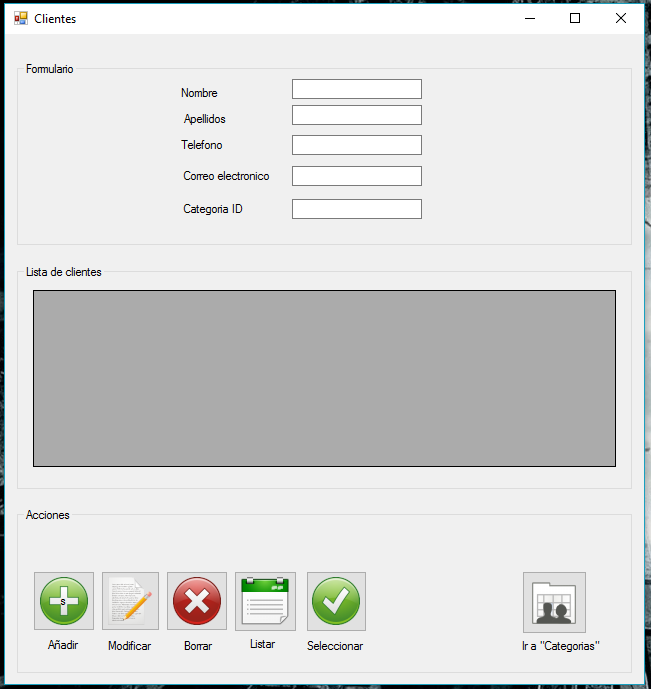
}

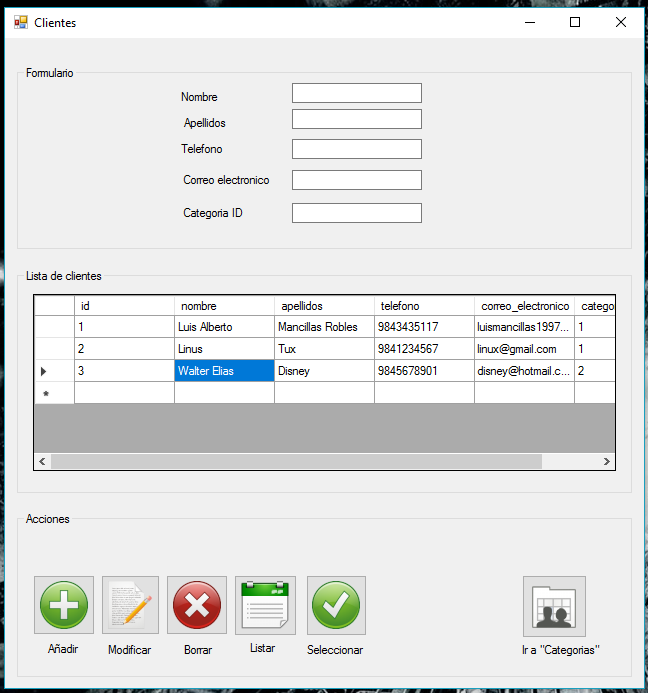
**Compilacion**

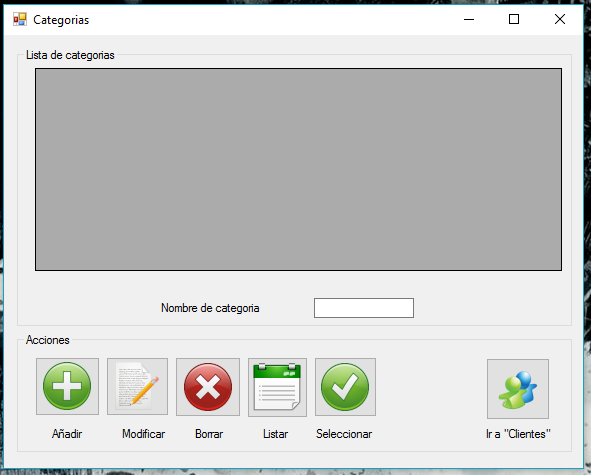
Muchos lenguajes interpretados (como el caso del C#, que es el lenguaje utilizado para el desarrollo de esta aplicacion) son primero compilados a código byte, que luego es normalmente interpretado por la máquina virtual usando la compilación en tiempo de ejecución, del código byte a código nativo. Sin embargo, algunas veces, el código byte también puede ser compilado a un binario nativo usando un compilador Ahead-of-time compilation (compilación por adelantado), o ejecutado nativamente, por el procesador hardware.

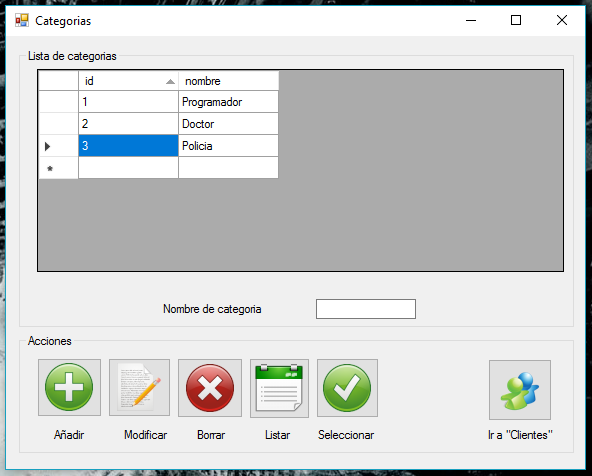
**Pruebas**

Se hizo pruebas del funcionamiento de ambas ventanas (clientes y categorías), donde es necesario que aparezca en el DataGriedView de la lista de clientes/categorías, los datos a insertar, modificar y eliminar, los cuales funcionan de manera correcta y para mostrar los cambios realizados se muestran con el botón “Listar”

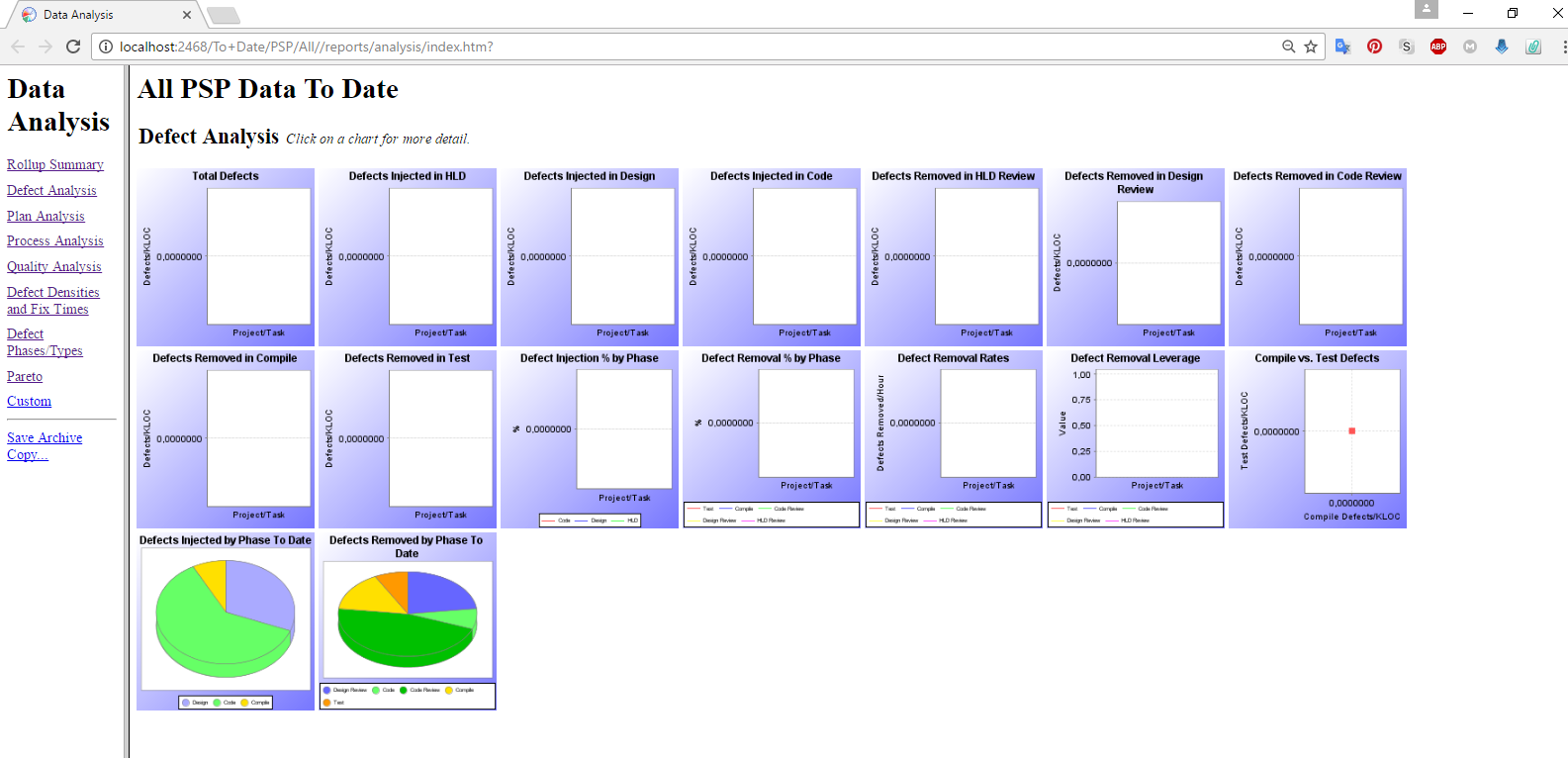


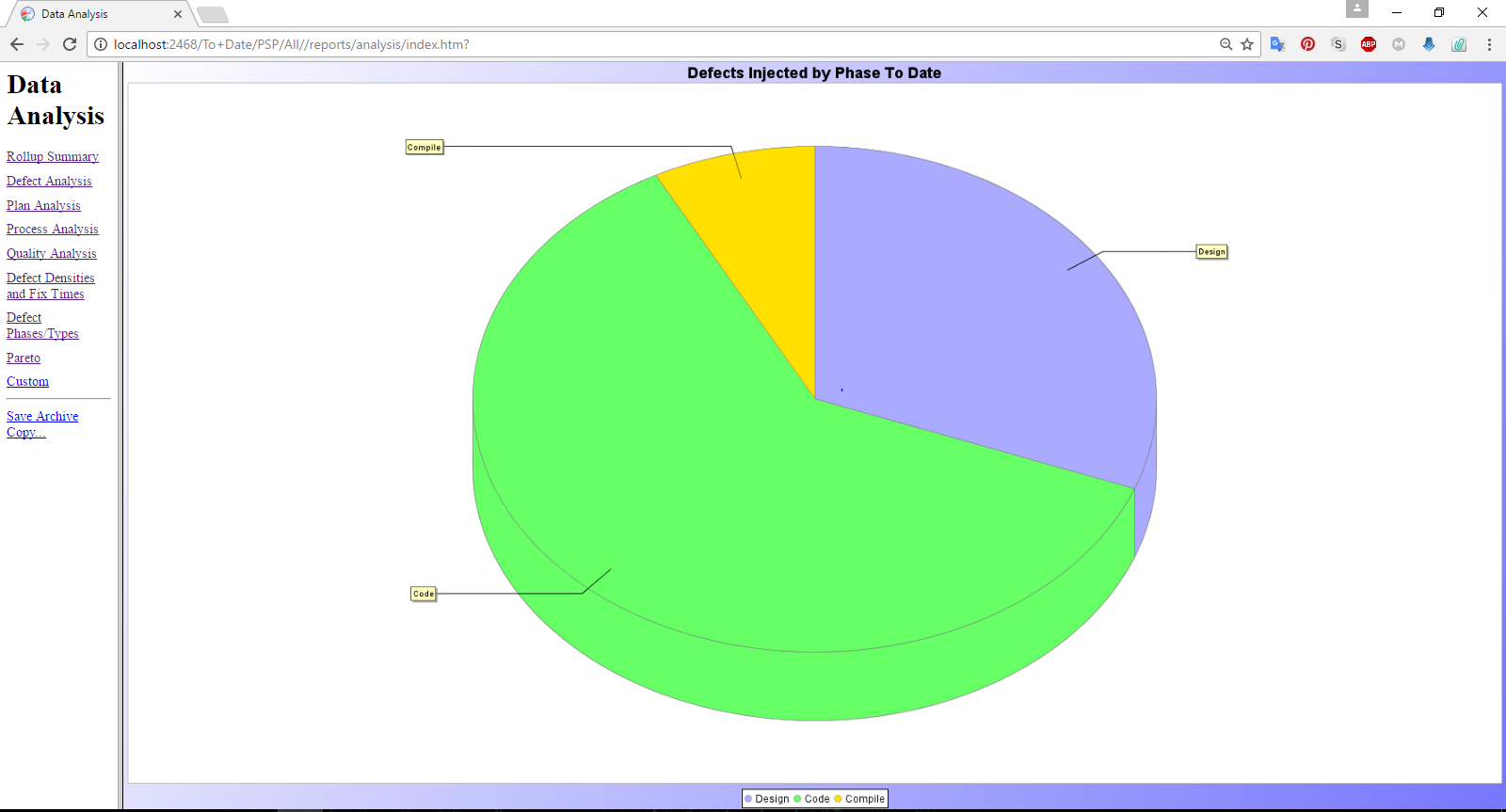


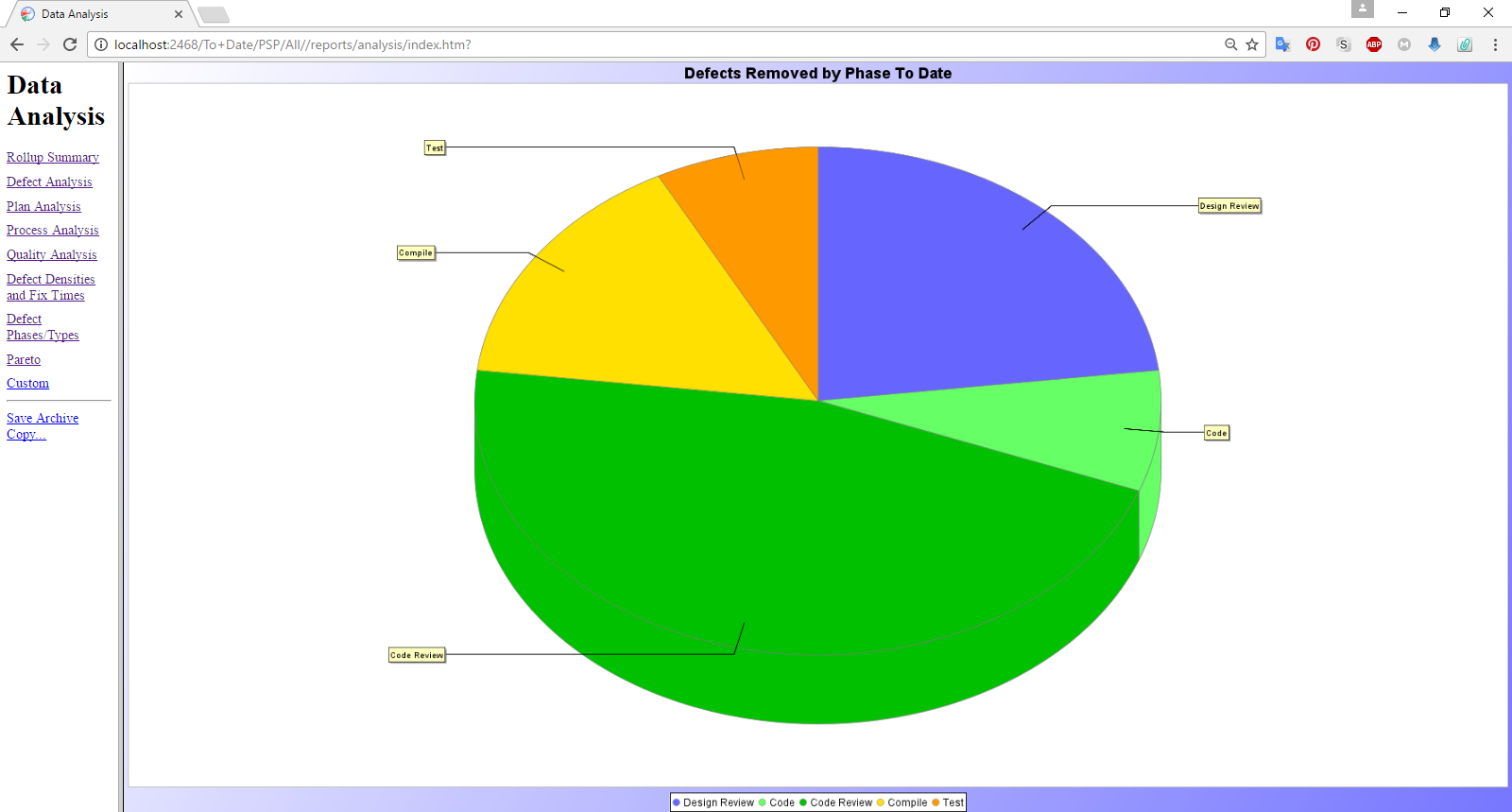




**Postmorten**







**Reporte del Process Dashboard**

