**U**

**P**

**T**

**Universidad Politécnica de Tulancingo**

**Plantilla Evaluación de casos de prueba.**

**V1SCU: Guía metodológica para el proceso de validación y verificación de casos de uso**

Por:

1730608, Mario Etzael Francisco Melo

**Ingeniería en Sistemas Computacionales**

Módulo

**Métodos y Herramientas de Ingeniería de Software**

Nombre del Catedrático:

**Roberto Arturo Sanchez Herrera**

Tulancingo de Bravo, Hidalgo

Mayo-agosto 2019



Contenido

[Introducción 3](#_Toc15573877)

[Propósito 3](#_Toc15573878)

[Alcance 3](#_Toc15573879)

[Documentos Referenciados 3](#_Toc15573880)

[Definiciones, abreviaciones y acrónimos 3](#_Toc15573881)

[Caso de Prueba 1. 3](#_Toc15573882)

[Bateria de Prueba. 4](#_Toc15573883)

[Codigo. 4](#_Toc15573884)

[Evidencia de ejecución de caso 1. 7](#_Toc15573885)

[Caso de Prueba 2. 7](#_Toc15573886)

[Bateria de Pruebas. 8](#_Toc15573887)

[Codigo 9](#_Toc15573888)

[Evidencia de ejecución de caso 2. 29](#_Toc15573889)

[Caso de Prueba 3. 30](#_Toc15573890)

[Bateria de Pruebas. 30](#_Toc15573891)

[Codigo: 31](#_Toc15573892)

[Evidencia de ejecución de caso 3. 45](#_Toc15573893)

[Documento Diseño de Prueba: 45](#_Toc15573894)

[Matriz de Trazabilidad de Pruebas. 45](#_Toc15573895)

[Referencias 46](#_Toc15573896)

Documento de casos de prueba

# Introducción

En el siguiente documento se definen los diferentes tipos de pruebas que se aplicarán al primer módulo del sistema de Laboratorios Bustillos el cual consta de una pantalla que contiene la validación de un usuario, la primera pantalla de menú, el registro de un nuevo usuario, además del reporte por fecha de las ventas.

# Propósito

El propósito es planear y diseñar pruebas para validar y verificar el acceso de usuario, registro de usuarios así como la integración completa del sistema.

# Alcance

El plan de pruebas solo verificará el funcionamiento de las pantallas: Acceso (Login), crear registro de Usuarios junto con la integración completa del sistema, no se evaluará ningún otro modulo.

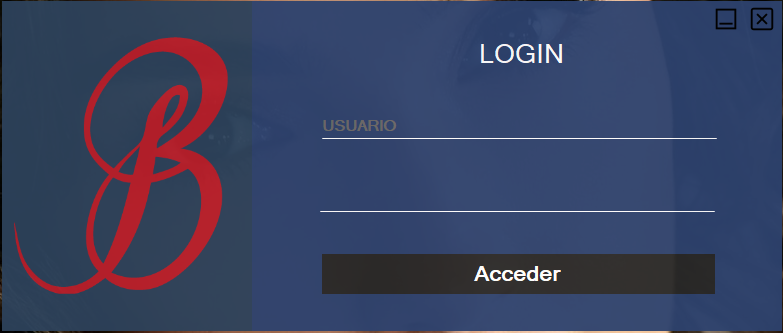
# Documentos Referenciados

Documento de especificación de requerimientos y Documento de Requisitos de Caso de Uso.

# Definiciones, abreviaciones y acrónimos

|  |  |
| --- | --- |
| Concepto | Descripción |
| *CP* | *Caso de prueba* |

# **Caso de Prueba 1.**



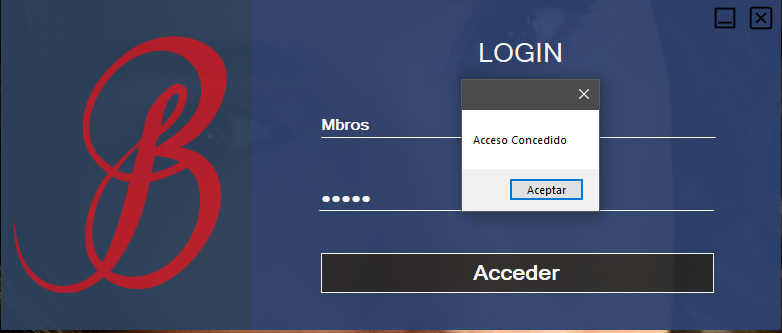
## Bateria de Prueba.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Caso de prueba | *Acceso* | | | | |
| Identificador caso de prueba | *CP02LOGIN* | | | | |
| Función probar | *Acceso al sistema.* | | | | |
| Objetivo | *Seguridad en el Sistema.* | | | | |
| Descripción | *Permitir el acceso al sistema mediante el logueo*  *Al presionar el botón de “Acceder” el sistema verifica que el usuario y la contraseña son correctos si lo son, verifica que tipo de rol tiene asignado el usuario* | | | | |
| Criterios de éxito | *Acceder al sistema y poder hacer uso del mismo* | | | | |
| Criterios de falla | *Loguearte si el usuario aun no existe* | | | | |
| Precondiciones | *Ser Usuario registrado previamente* | | | | |
| Necesidades para el caso de prueba | *Usuario y contraseña para dar acceso* | | | | |
| Autor | *Mario Etzael Francisco Melo* | | | | |
| Fecha de creación | *20 de junio del 2019* | | | | |
| Flujo del caso de prueba | No paso | Usuario del sistema | No paso | Sistema | Resultado obtenido |
| *1* | *Se ingresa un usuario y una contraseña al sistema* |  | *Permite el acceso a los servicios que ofrece dependiendo del rol del usuario* | *Éxito* |
|  |  |  |  |  |
|  |  |  |  |  |
| Post condiciones | *Una vez logueado se tiene acceso al sistema* | | | | |

## Codigo.

1. using System;
2. using System.Data;
3. using System.Data.SqlClient;
4. using System.Windows.Forms;
5. namespace Lab\_Bustillos
6. {
7. public partial class Login : Form
8. {
9. //Se instancia la clase conexion para poder acceder a la base de datos
10. Conexion cn = new Conexion();
11. SqlDataReader leer;
12. DataTable tabla = new DataTable();
13. SqlCommand comando = new SqlCommand();
14. public Login()
15. {
16. InitializeComponent();
17. }
18. //El metodo \_enter sirve para poner el placeholder del campo de texto al igual que en \_leave
19. private void txtUsu\_Enter(object sender, EventArgs e)
20. {
21. if (txtUsu.Text == "USUARIO")
22. {
23. txtUsu.Text = "";
24. txtUsu.ForeColor = System.Drawing.Color.White;
25. }
26. }
27. private void txtUsu\_Leave(object sender, EventArgs e)
28. {
29. if (txtUsu.Text=="")
30. {
31. txtUsu.Text = "USUARIO";
32. txtUsu.ForeColor = System.Drawing.Color.DimGray;
33. }
34. }
35. private void txtContra\_Enter(object sender, EventArgs e)
36. {
37. if (txtContra.Text == "CONTRASEÑA")
38. {
39. txtContra.Text = "";
40. txtContra.ForeColor = System.Drawing.Color.White;
41. txtContra.UseSystemPasswordChar = true;
42. }
43. }
44. private void txtContra\_Leave(object sender, EventArgs e)
45. {
46. if (txtContra.Text == "")
47. {
48. txtContra.Text = "CONTRASEÑA";
49. txtContra.ForeColor = System.Drawing.Color.DimGray;
50. txtContra.UseSystemPasswordChar = false;
51. }
52. }
53. //El boton de acceder verifica en la base de datos que el usuario y la contraseña
54. //ingresados sean los correctos.
55. private void btnAcceder\_Click\_1(object sender, EventArgs e)
56. {
57. try
58. {
59. //Primero abre la conexion para acceder al servidor
60. comando.Connection = cn.AbrirConexion();
61. //selecciona los datos pedidos de la base de datos.
62. comando.CommandText = "select Personal.Nombre,Personal.Rol, Clicks.Clicks from Personal inner join Clicks on Personal.Nombre=Clicks.Nombre where Personal.Usuario='"+txtUsu.Text+"' and Personal.Contraseña='"+txtContra.Text+"'";
63. leer = comando.ExecuteReader();
64. int ed = 0;
65. string rol = "";
66. string clicks="";
67. //si la sentencia cumple con la codicion se ejecuta un while donde se saca el rol del usuario
68. while (leer.Read())
69. {
70. ed++;
71. rol = leer.GetString(1);
72. Personal.nombreP= leer.GetString(0);
73. Personal.click= leer.GetInt32(2);
74. }
75. Principal p = new Principal();
76. if (ed == 1)
77. {
78. //si es administrador tiene acceso a todas las funcionalidades del programa
79. if (rol.Equals("Administrador"))
80. {
81. MessageBox.Show("Acceso Concedido");
82. this.Visible = false;
83. p.Show();
84. }
85. else
86. {
87. //el empleado solo tiene acceso a la ventana de ventas.
88. MessageBox.Show("Bienvenido");
89. new Ventas().Show();
90. this.Visible = false;
91. }
92. }
93. else
94. {
95. MessageBox.Show("Acceso Denegado");
96. }
97. }
98. catch (Exception ex)
99. {
100. MessageBox.Show("Error: " + ex);
101. throw;
102. }
103. comando.Connection = cn.CerrarConexion();
104. }
105. private void btnCerrar\_Click(object sender, EventArgs e)
106. {
107. System.Environment.Exit(0);
108. }
109. private void btnMinimizar\_Click(object sender, EventArgs e)
110. {
111. this.WindowState = FormWindowState.Minimized;
112. }
113. }
114. }

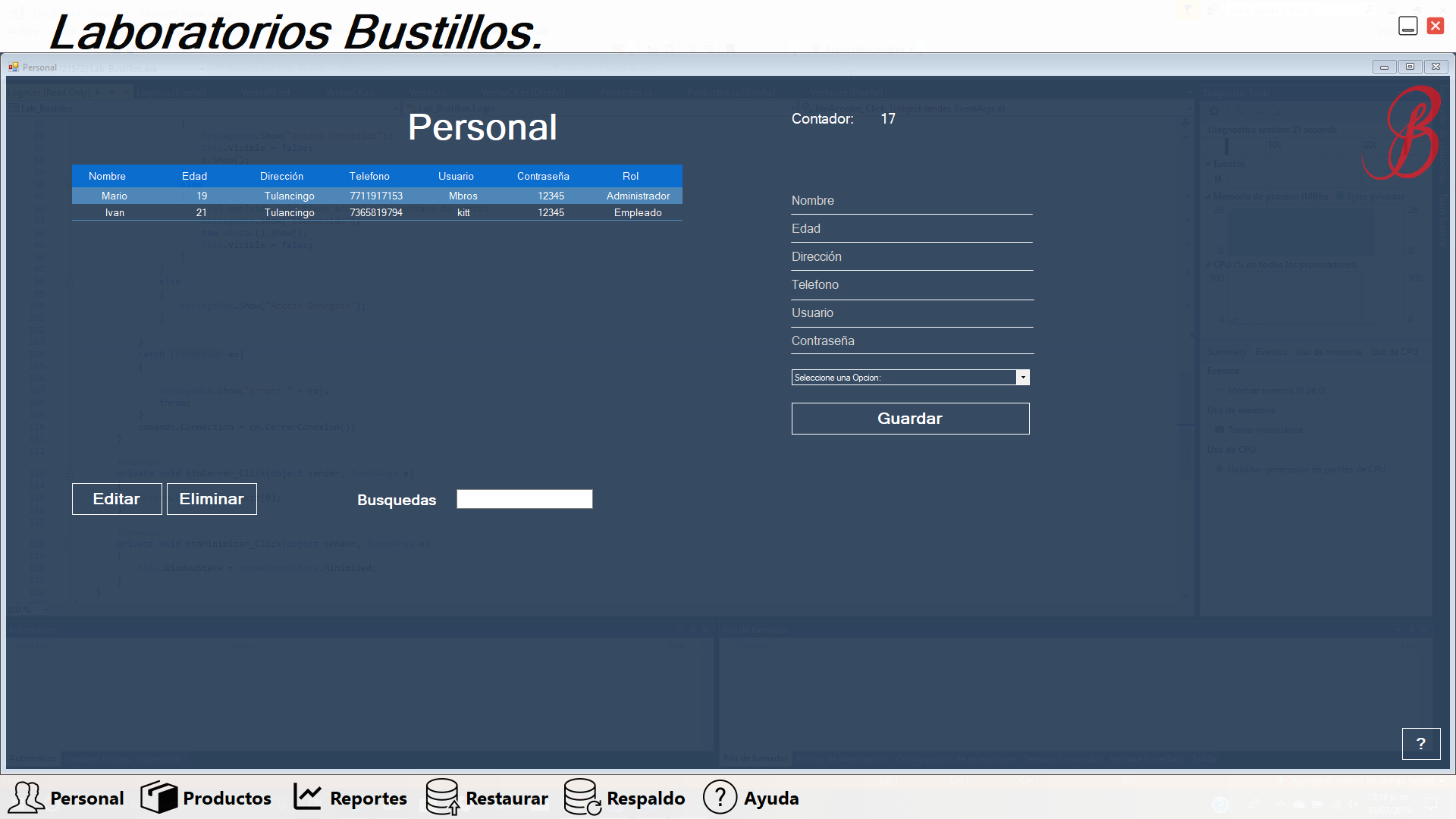
# Evidencia de ejecución de caso 1.



Reporte

|  |  |  |
| --- | --- | --- |
| Sub-Flujo | Cumple | Observaciones |
| 1 | Correcto | Funciona Correctamente |

# Caso de Prueba 2.



## Bateria de Pruebas.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Caso de prueba | *Registro de Usuarios.* | | | | |
| Identificador caso de prueba | *CP02REGISTRO* | | | | |
| Función probar | *Registrar Usuarios* | | | | |
| Objetivo | *Registro de usuarios para el uso del sistema* | | | | |
| Descripción | *Registrar los usuarios asi como la modificación y la eliminación de los mismos*  *Al presionar el botón de Guardar dependiendo de la función dada, elimina, registra o modifica un usuario* | | | | |
| Criterios de éxito | *Acceder al sistema y poder hacer uso del mismo* | | | | |
| Criterios de falla | *Loguearte si el usuario aun no existe* | | | | |
| Precondiciones | *Ser Usuario registrado previamente* | | | | |
| Necesidades para el caso de prueba | *Usuario y contraseña para dar acceso* | | | | |
| Autor | *Mario Etzael Francisco Melo* | | | | |
| Fecha de creación | *20 de junio del 2019* | | | | |
| Flujo del caso de prueba | No paso | Usuario del sistema | No paso | Sistema | Resultado obtenido |
| *1* | *Se hace uso del sistema dependiendo del rol* |  | *Permite el acceso a los servicios que ofrece dependiendo del rol del usuario* | *Éxito* |
|  |  |  |  |  |
|  |  |  |  |  |
| Post condiciones | *Una vez logueado se tiene acceso al sistema* | | | | |

## Codigo

using System;

using System.Data;

using System.Data.SqlClient;

using System.Windows.Forms;

namespace Lab\_Bustillos

{

public partial class Personal : Form

{

Conexion cn = new Conexion();

SqlDataReader leer;

DataTable tabla = new DataTable();

SqlDataAdapter da;

SqlCommand comando = new SqlCommand();

private static int suma=0;

public static string nombreP = "";

public static int click = 0;

private static int opcion=0;

string nombre, edad, direccion, telefono, usuario, contraseña, rol;

public Personal()

{

InitializeComponent();

ComboRol.SelectedIndex = 0;

Usuarios(dataGridView1);

cont.Text = Convert.ToString(click);

}

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

{

ClickB();

if (opcion == 0)

{

try

{

int i = 0;

//Se abre la conexion a la base de datos y se ejecuta el procedimiendo almacenado con el paso de parametros dado

comando.Connection = cn.AbrirConexion();

comando.CommandText = "Exec AgregarU '" + Nombretxt.Text + "','" + Edadtxt.Text + "','" + Direcciontxt.Text + "','" + Telefonotxt.Text + "','" + Usuariotxt.Text + "','" + Contraseñatxt.Text + "','" + ComboRol.SelectedItem.ToString() + "';";

comando.ExecuteNonQuery();

Usuarios(dataGridView1);

i++;

if (i == 1)//Si el procedimiendo se ejecuto con exito da como resultado uno

{

MessageBox.Show("Se agrego correctamente");//Todo OK

Limpiar();

}

else

{

//Si los datos ya existen lanza un mensaje de datos duplicados

MessageBox.Show("Datos Duplicados");

}

}

catch (SqlException ex)

{

MessageBox.Show("Datos Duplicados");

Limpiar();

}

}

else {

try

{

int i = 0;

//Se abre la conexion a la base de datos y se ejecuta el procedimiendo almacenado con el paso de parametros dado

comando.Connection = cn.AbrirConexion();

comando.CommandText = "Exec ModificarU '" + Nombretxt.Text + "','" + Edadtxt.Text + "','" + Direcciontxt.Text + "','" + Telefonotxt.Text + "','" + Usuariotxt.Text + "','" + Contraseñatxt.Text + "','" + ComboRol.Text + "','" + nombre + "','" + edad + "','" + direccion + "','" + telefono + "','" + usuario + "','" + contraseña + "','" + rol + "';";

comando.ExecuteNonQuery();

Usuarios(dataGridView1);

i++;

if (i == 1)//Si el procedimiendo se ejecuto con exito da como resultado uno

{

MessageBox.Show("Se Modifico correctamente");

Limpiar();

}

else

{

MessageBox.Show("Error");

}

}

catch (SqlException ex)

{

//Si hubo algun problema con la base de datos el programa lanzara un error

MessageBox.Show("Error: " + ex.ToString());

}

opcion = 0;

}

}

//Estos metodos llaman al metodo ClickB para llevar el conteo de los clicks

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

{

ClickB();

}

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

{

ClickB();

}

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

{

ClickB();

}

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

{

ClickB();

}

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

{

ClickB();

}

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

{

ClickB();

}

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

{

ClickB();

}

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

{

ClickB();

}

//Los metodos \_Enter sirven para el placeholder del campo de texto al igual que los metodos \_Leave

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

{

if (Nombretxt.Text == "Nombre")

{

Nombretxt.Text = "";

Nombretxt.ForeColor = System.Drawing.Color.White;

}

}

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

{

if (Edadtxt.Text == "Edad")

{

Edadtxt.Text = "";

Edadtxt.ForeColor = System.Drawing.Color.White;

}

}

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

{

if (Direcciontxt.Text == "Dirección")

{

Direcciontxt.Text = "";

Direcciontxt.ForeColor = System.Drawing.Color.White;

}

}

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

{

if (Telefonotxt.Text == "Telefono")

{

Telefonotxt.Text = "";

Telefonotxt.ForeColor = System.Drawing.Color.White;

}

}

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

{

if (Usuariotxt.Text == "Usuario")

{

Usuariotxt.Text = "";

Usuariotxt.ForeColor = System.Drawing.Color.White;

}

}

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

{

if (Contraseñatxt.Text == "Contraseña")

{

Contraseñatxt.Text = "";

Contraseñatxt.ForeColor = System.Drawing.Color.White;

}

}

private void Nombretxt\_Leave(object sender, EventArgs e)

{

if (Nombretxt.Text == "")

{

Nombretxt.Text = "Nombre";

Nombretxt.ForeColor = System.Drawing.Color.LightGray;

}

}

private void Edadtxt\_Leave(object sender, EventArgs e)

{

if (Edadtxt.Text == "")

{

Edadtxt.Text = "Edad";

Edadtxt.ForeColor = System.Drawing.Color.LightGray;

}

}

private void Direcciontxt\_Leave(object sender, EventArgs e)

{

if (Direcciontxt.Text == "")

{

Direcciontxt.Text = "Dirección";

Direcciontxt.ForeColor = System.Drawing.Color.LightGray;

}

}

private void Telefonotxt\_Leave(object sender, EventArgs e)

{

if (Telefonotxt.Text == "")

{

Telefonotxt.Text = "Telefono";

Telefonotxt.ForeColor = System.Drawing.Color.LightGray;

}

}

private void Usuariotxt\_Leave(object sender, EventArgs e)

{

if (Usuariotxt.Text == "")

{

Usuariotxt.Text = "Usuario";

Usuariotxt.ForeColor = System.Drawing.Color.LightGray;

}

}

private void Contraseñatxt\_Leave(object sender, EventArgs e)

{

if (Contraseñatxt.Text == "")

{

Contraseñatxt.Text = "Contraseña";

Contraseñatxt.ForeColor = System.Drawing.Color.LightGray;

}

}

//-------------------------------------------------------------------

//Estos metodos reciben un parametro de tipo datagridview para el llenado de las tablas

public void Usuarios(DataGridView dgv)

{

dgv.Rows.Clear();

try

{

string nombre = "";

int edad = 0;

string direccion = "";

string telefono = "";

string usuario = "";

string contraseña = "";

string rol = "";

comando = new SqlCommand("select \* from Personal;", cn.AbrirConexion());

leer = comando.ExecuteReader();

while (leer.Read())

{

//Los valores de la base de datos se pasan a variables creadas anteriormente

nombre = leer.GetString(0);

edad = leer.GetInt32(1);

direccion = leer.GetString(2);

telefono = leer.GetString(3);

usuario = leer.GetString(4);

contraseña = leer.GetString(5);

rol = leer.GetString(6);

//Los datos de agregan a la tabla en la interfaz

dgv.Rows.Add(nombre, edad, direccion, telefono, usuario, contraseña, rol);

}

leer.Close();

cn.CerrarConexion();

}

catch (Exception ex)

{

MessageBox.Show("No se puedo realizar la busqueda: " + ex.ToString());

}

}

//Este metodo busca todas la coincidencias que encuentre con lo que se bsuca en el campo de texto de BUSQUEDA

public void PersonalB(DataGridView dgv)

{

dgv.Rows.Clear();

try

{

string nombre = "";

int edad = 0;

string direccion = "";

string telefono = "";

string usuario = "";

string contraseña = "";

string rol = "";

comando = new SqlCommand("select \* from Personal where Nombre like '" + txtBuscar.Text + "%'", cn.AbrirConexion());

leer = comando.ExecuteReader();

while (leer.Read())

{

//Los valores de la base de datos se pasan a variables creadas anteriormente

nombre = leer.GetString(0);

edad = leer.GetInt32(1);

direccion = leer.GetString(2);

telefono = leer.GetString(3);

usuario = leer.GetString(4);

contraseña = leer.GetString(5);

rol = leer.GetString(6);

//Los datos de agregan a la tabla en la interfaz

dgv.Rows.Add(nombre, edad, direccion, telefono, usuario, contraseña, rol);

}

leer.Close();

cn.CerrarConexion();

}

catch (Exception ex)

{

MessageBox.Show("No se puedo realizar la busqueda: " + ex.ToString());

}

}

//-----------------------------------------------------------------------------------------------------------------

//Los keyPress nos sirvieron para validar si son numero o letras dependiendo del tipo de campo

private void Nombretxt\_KeyPress(object sender, KeyPressEventArgs e)

{

//IsLetter(e.KeyChar) permite cualquier caracter entre la a y la z tanto mayusculas como minusculas

//IsWhiteSpace permite espacio en blanco.

//'\b' permite poder eliminar caracteres con BackSpace

if (!char.IsLetter(e.KeyChar) && !char.IsWhiteSpace(e.KeyChar) && e.KeyChar != '\b')

{

e.Handled = true;

}

}

private void Direcciontxt\_KeyPress(object sender, KeyPressEventArgs e)

{

//IsLetter(e.KeyChar) permite cualquier caracter entre la a y la z tanto mayusculas como minusculas

//IsWhiteSpace permite espacio en blanco.

//'\b' permite poder eliminar caracteres con BackSpace

if (!char.IsLetter(e.KeyChar) && !char.IsWhiteSpace(e.KeyChar) && e.KeyChar != '\b')

{

e.Handled = true;

}

}

private void Edadtxt\_KeyPress\_1(object sender, KeyPressEventArgs e)

{

if (Char.IsDigit(e.KeyChar))

{

e.Handled = false;

}

else if (Char.IsControl(e.KeyChar))

{

e.Handled = false;

}

else if (Char.IsSeparator(e.KeyChar))

{

e.Handled = false;

}

else

{

e.Handled = true;

}

}

private void Telefonotxt\_KeyPress(object sender, KeyPressEventArgs e)

{

if (Char.IsDigit(e.KeyChar))

{

e.Handled = false;

}

else if (Char.IsControl(e.KeyChar))

{

e.Handled = false;

}

else if (Char.IsSeparator(e.KeyChar))

{

e.Handled = false;

}

else

{

e.Handled = true;

}

}

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

{

AyudaP.opcion = 0;

new AyudaP().Show();

}

private void Usuariotxt\_KeyPress(object sender, KeyPressEventArgs e)

{

//IsLetter(e.KeyChar) permite cualquier caracter entre la a y la z tanto mayusculas como minusculas

//IsWhiteSpace permite espacio en blanco.

//'\b' permite poder eliminar caracteres con BackSpace

if (!char.IsLetter(e.KeyChar) && !char.IsWhiteSpace(e.KeyChar) && e.KeyChar != '\b')

{

e.Handled = true;

}

}

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

{

PersonalB(dataGridView1);

}

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

{

ClickB();

foreach (DataGridViewRow row in dataGridView1.SelectedRows)

{

nombre = row.Cells[0].Value.ToString();

edad = row.Cells[1].Value.ToString();

direccion = row.Cells[2].Value.ToString();

telefono = row.Cells[3].Value.ToString();

usuario = row.Cells[4].Value.ToString();

contraseña = row.Cells[5].Value.ToString();

rol = row.Cells[6].Value.ToString();

}

try

{

int i = 0;

comando.Connection = cn.AbrirConexion();

comando.CommandText = "Exec EliminarU '" + nombre + "','" + edad + "','" + direccion + "','" + telefono + "','" + usuario + "','" + contraseña + "','" + rol + "';";

comando.ExecuteNonQuery();

Usuarios(dataGridView1);

i++;

if (i == 1)

{

MessageBox.Show("Se Elimino correctamente");

Limpiar();

}

else

{

MessageBox.Show("Error");

}

}

catch (SqlException ex)

{

MessageBox.Show("Error: " + ex.ToString());

}

}

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

{

ClickB();

opcion = 1;

//Saca los valores de la fila seleccionada y los pasa a los campos de texto

foreach (DataGridViewRow row in dataGridView1.SelectedRows)

{

Nombretxt.Text = row.Cells[0].Value.ToString();

Edadtxt.Text = row.Cells[1].Value.ToString();

Direcciontxt.Text = row.Cells[2].Value.ToString();

Telefonotxt.Text = row.Cells[3].Value.ToString();

Usuariotxt.Text = row.Cells[4].Value.ToString();

Contraseñatxt.Text = row.Cells[5].Value.ToString();

ComboRol.SelectedItem = row.Cells[6].Value.ToString();

nombre = row.Cells[0].Value.ToString();

edad = row.Cells[1].Value.ToString();

direccion = row.Cells[2].Value.ToString();

telefono = row.Cells[3].Value.ToString();

usuario = row.Cells[4].Value.ToString();

contraseña = row.Cells[5].Value.ToString();

rol = row.Cells[6].Value.ToString();

}

Nombretxt.Enabled = false;

}

public void Limpiar()

{

Nombretxt.Text = "Nombre";

Edadtxt.Text = "Edad";

Direcciontxt.Text = "Dirección";

Telefonotxt.Text = "Telefono";

Usuariotxt.Text = "Usuario";

Contraseñatxt.Text = "Contraseña";

ComboRol.SelectedIndex = 0;

Nombretxt.Enabled = true;

Usuarios(dataGridView1);

}

//Este metodo sirve para llevar el conteo de los clicks que el usuario da.

public void ClickB() {

click += 1;

try

{

comando.Connection = cn.AbrirConexion();

comando.CommandText = "update Clicks set Clicks='"+click+"' where Nombre='" + nombreP + "'";

comando.ExecuteNonQuery();

}

catch (Exception e)

{

throw;

}

cn.CerrarConexion();

try

{

comando.Connection = cn.AbrirConexion();

comando.CommandText = "select \* from Clicks where Nombre='"+nombreP+"'";

leer = comando.ExecuteReader();

int ed = 0;

while (leer.Read())

{

ed++;

cont.Text = Convert.ToString(leer.GetInt32(1));

}

}

catch (Exception e)

{

throw;

}

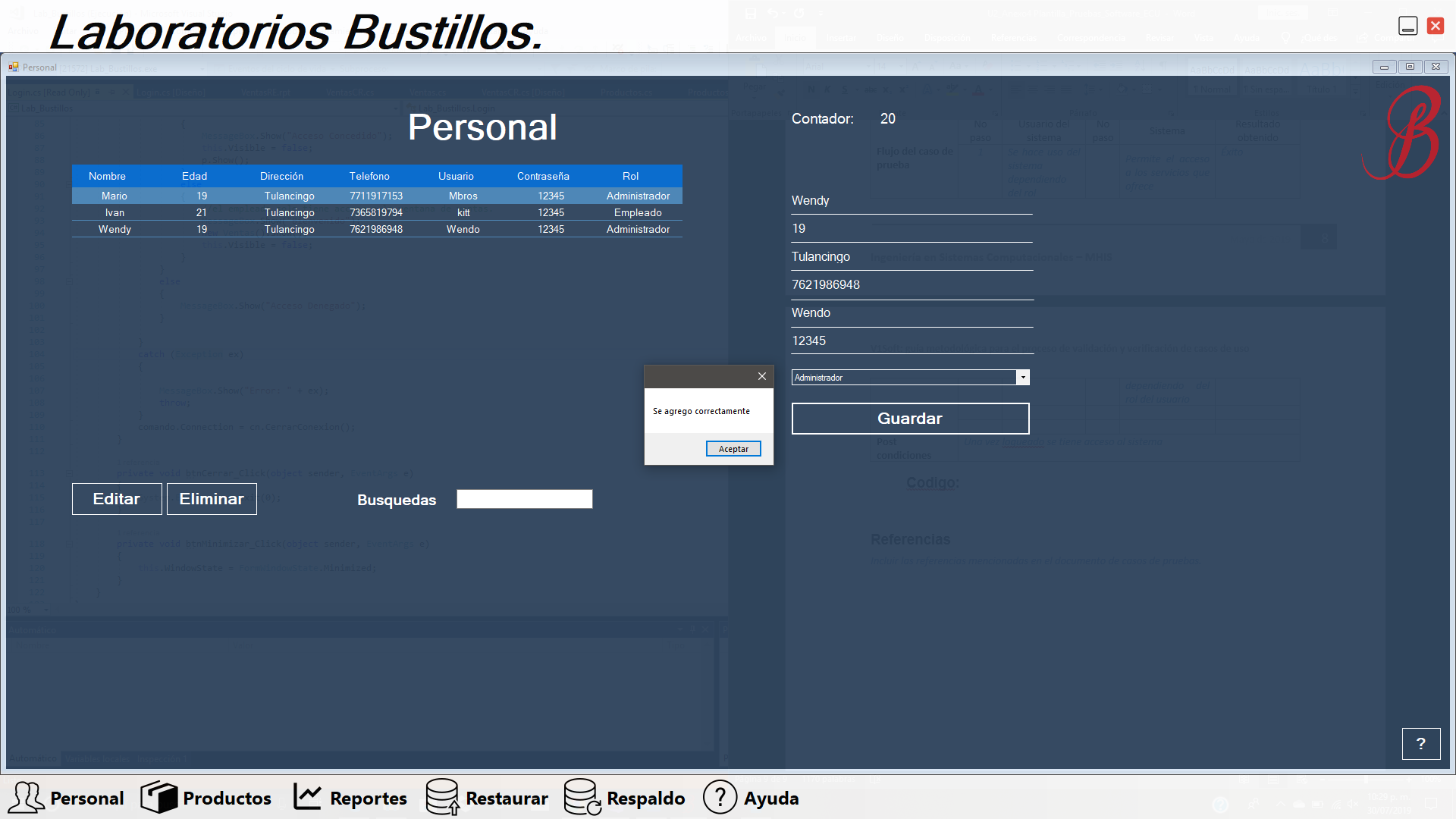
cn.CerrarConexion();

}

}

}

## Evidencia de ejecución de caso 2.



|  |  |  |
| --- | --- | --- |
| Sub-Flujo | Cumple | Observaciones |
| 2 | Correcto | Correcto |
|  |  |  |

# Caso de Prueba 3.



## Bateria de Pruebas.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Caso de prueba | *Registro de Productos.* | | | | |
| Identificador caso de prueba | *CP02REGISTROP* | | | | |
| Función probar | *Registrar Productos* | | | | |
| Objetivo | *Registro de Productos para la venta dentro del sistema* | | | | |
| Descripción | *Registrar los productis asi como la modificación y la eliminación de los mismos*  *Al presionar el botón de Guardar dependiendo de la función dada, elimina, registra o modifica un usuario* | | | | |
| Criterios de éxito | *Acceder al sistema y poder hacer uso del mismo* | | | | |
| Criterios de falla | *Loguearte si el usuario aun no existe* | | | | |
| Precondiciones | *Ser Usuario registrado previamente* | | | | |
| Necesidades para el caso de prueba | *Usuario y contraseña para dar acceso* | | | | |
| Autor | *Mario Etzael Francisco Melo* | | | | |
| Fecha de creación | *20 de junio del 2019* | | | | |
| Flujo del caso de prueba | No paso | Usuario del sistema | No paso | Sistema | Resultado obtenido |
| *1* | *Se hace uso del sistema dependiendo del rol* |  | *Permite el acceso a los servicios que ofrece dependiendo del rol del usuario* | *Éxito* |
|  |  |  |  |  |
|  |  |  |  |  |
| Post condiciones | *Una vez logueado se tiene acceso al sistema* | | | | |

## Codigo:

using System;

using System.Data;

using System.Data.SqlClient;

using System.Drawing;

using System.Windows.Forms;

namespace Lab\_Bustillos

{

public partial class Productos : Form

{

Conexion cn = new Conexion();

SqlDataReader leer;

DataTable tabla = new DataTable();

SqlDataAdapter da;

SqlCommand comando = new SqlCommand();

private static int opcion = 0;

public Productos()

{

InitializeComponent();

ProductosC(dataGridView1);

Random();

}

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

{

if (opcion == 0)

{

try

{

int i = 0;

comando.Connection = cn.AbrirConexion();

comando.CommandText = "exec AgregarP'" + txtCodigo.Text + "','" + txtDescripcion.Text + "','" + txtPrecio.Text + "','" + txtExist.Text + "'";

comando.ExecuteNonQuery();

//Usuarios(dataGridView1);

ProductosC(dataGridView1);

i++;

if (i == 1)

{

MessageBox.Show("Se agrego correctamente");

Limpiar();

}

else

{

MessageBox.Show("Error");

}

}

catch (SqlException ex)

{

MessageBox.Show("Error: " + ex.ToString());

}

}

else

{

try

{

int i = 0;

comando.Connection = cn.AbrirConexion();

comando.CommandText = "exec ModificarP '" + txtCodigo.Text + "','" + txtDescripcion.Text + "','" + txtPrecio.Text + "','" + txtExist.Text + "','" + codigo + "','" + desc + "','" + pre + "','" + exist + "'";

comando.ExecuteNonQuery();

//Usuarios(dataGridView1);

ProductosC(dataGridView1);

i++;

if (i == 1)

{

MessageBox.Show("Se agrego correctamente");

Limpiar();

}

else

{

MessageBox.Show("Error");

}

}

catch (SqlException ex)

{

MessageBox.Show("Error: " + ex.ToString());

}

opcion = 0;

}

}

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

{

foreach (DataGridViewRow dat in dataGridView1.SelectedRows)

{

codigo = dat.Cells["Column1"].Value.ToString();

desc = dat.Cells["Column2"].Value.ToString();

pre = dat.Cells["Column3"].Value.ToString();

exist = dat.Cells["Column4"].Value.ToString();

}

try

{

int i = 0;

comando.Connection = cn.AbrirConexion();

comando.CommandText = "exec EliminarP'" + codigo + "','" + desc + "','" + pre + "','" + exist + "'";

comando.ExecuteNonQuery();

ProductosC(dataGridView1);

i++;

if (i == 1)

{

MessageBox.Show("Se agrego correctamente");

Limpiar();

}

else

{

MessageBox.Show("Error");

}

}

catch (SqlException ex)

{

MessageBox.Show("Error: " + ex.ToString());

}

}

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

{

opcion = 1;

foreach (DataGridViewRow dat in dataGridView1.SelectedRows)

{

txtCodigo.Text = dat.Cells["Column1"].Value.ToString();

txtDescripcion.Text = dat.Cells["Column2"].Value.ToString();

txtPrecio.Text = dat.Cells["Column3"].Value.ToString();

txtExist.Text = dat.Cells["Column4"].Value.ToString();

codigo = dat.Cells["Column1"].Value.ToString();

desc = dat.Cells["Column2"].Value.ToString();

pre = dat.Cells["Column3"].Value.ToString();

exist = dat.Cells["Column4"].Value.ToString();

txtCodigo.Enabled = false;

}

}

string codigo, desc, pre, exist;

private void dataGridView1\_CellMouseClick(object sender, DataGridViewCellMouseEventArgs e)

{

}

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

{

ProductosB(dataGridView1);

}

public void ProductosB(DataGridView dgv)

{

dgv.Rows.Clear();

try

{

string codigo = "";

string descrip = "";

int precio = 0;

int exist = 0;

comando = new SqlCommand("select \* from Productos where Codigo like '" + txtBuscar.Text + "%'", cn.AbrirConexion());

leer = comando.ExecuteReader();

while (leer.Read())

{

codigo = leer.GetString(0);

descrip = leer.GetString(1);

precio = leer.GetInt32(2);

exist = leer.GetInt32(3);

dgv.Rows.Add(codigo, descrip, precio, exist);

}

leer.Close();

cn.CerrarConexion();

}

catch (Exception ex)

{

MessageBox.Show("No se puedo realizar la busqueda: " + ex.ToString());

}

}

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

{

if (txtDescripcion.Text == "Descripción")

{

txtDescripcion.Text = "";

txtDescripcion.ForeColor = System.Drawing.Color.White;

}

}

private void txtDescripcion\_Leave(object sender, EventArgs e)

{

if (txtDescripcion.Text == "")

{

txtDescripcion.Text = "Descripción";

txtDescripcion.ForeColor = System.Drawing.Color.LightGray;

}

}

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

{

if (txtPrecio.Text == "Precio")

{

txtPrecio.Text = "";

txtPrecio.ForeColor = System.Drawing.Color.White;

}

}

private void txtPrecio\_Leave(object sender, EventArgs e)

{

if (txtPrecio.Text == "")

{

txtPrecio.Text = "Precio";

txtPrecio.ForeColor = System.Drawing.Color.LightGray;

}

}

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

{

if (txtExist.Text == "Existencias")

{

txtExist.Text = "";

txtExist.ForeColor = System.Drawing.Color.White;

}

}

private void txtExist\_Leave(object sender, EventArgs e)

{

if (txtExist.Text == "")

{

txtExist.Text = "Existencias";

txtExist.ForeColor = System.Drawing.Color.LightGray;

}

}

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

{

if (txtCodigo.Text == "Codigo")

{

txtCodigo.Text = "";

txtCodigo.ForeColor = System.Drawing.Color.White;

}

}

private void txtCodigo\_Leave(object sender, EventArgs e)

{

if (txtCodigo.Text == "")

{

txtCodigo.Text = "Existencias";

txtCodigo.ForeColor = System.Drawing.Color.LightGray;

}

}

private void txtDescripcion\_KeyPress(object sender, KeyPressEventArgs e)

{

//IsLetter(e.KeyChar) permite cualquier caracter entre la a y la z tanto mayusculas como minusculas

//IsWhiteSpace permite espacio en blanco.

//'\b' permite poder eliminar caracteres con BackSpace

if (!char.IsLetter(e.KeyChar) && !char.IsWhiteSpace(e.KeyChar) && e.KeyChar != '\b')

{

e.Handled = true;

}

}

private void txtPrecio\_KeyPress(object sender, KeyPressEventArgs e)

{

if (Char.IsDigit(e.KeyChar))

{

e.Handled = false;

}

else if (Char.IsControl(e.KeyChar))

{

e.Handled = false;

}

else if (Char.IsSeparator(e.KeyChar))

{

e.Handled = false;

}

else

{

e.Handled = true;

}

}

private void txtExist\_KeyPress(object sender, KeyPressEventArgs e)

{

if (Char.IsDigit(e.KeyChar))

{

e.Handled = false;

}

else if (Char.IsControl(e.KeyChar))

{

e.Handled = false;

}

else if (Char.IsSeparator(e.KeyChar))

{

e.Handled = false;

}

else

{

e.Handled = true;

}

}

private void dataGridView1\_CellFormatting(object sender, DataGridViewCellFormattingEventArgs e)

{

if (this.dataGridView1.Columns[e.ColumnIndex].Index == 3)

{

if (Convert.ToInt32(e.Value) <= 5)

{

e.CellStyle.ForeColor = Color.White;

e.CellStyle.BackColor = Color.Red;

}

}

}

public void ProductosC(DataGridView dgv)

{

dgv.Rows.Clear();

try

{

string codigo = "";

string descrip = "";

int precio = 0;

int exist = 0;

comando = new SqlCommand("select \* from Productos", cn.AbrirConexion());

leer = comando.ExecuteReader();

while (leer.Read())

{

codigo = leer.GetString(0);

descrip = leer.GetString(1);

precio = leer.GetInt32(2);

exist = leer.GetInt32(3);

dgv.Rows.Add(codigo, descrip, precio, exist);

}

leer.Close();

cn.CerrarConexion();

}

catch (Exception ex)

{

MessageBox.Show("No se puedo realizar la busqueda: " + ex.ToString());

}

}

public void Limpiar()

{

txtCodigo.Text = "Codigo";

txtDescripcion.Text = "Descripción";

txtPrecio.Text = "Precio";

txtExist.Text = "Existencias";

txtCodigo.Enabled = true;

ProductosC(dataGridView1);

Random();

}

public void Random()

{

Random r = new Random();

int num = r.Next(1000, 9999);

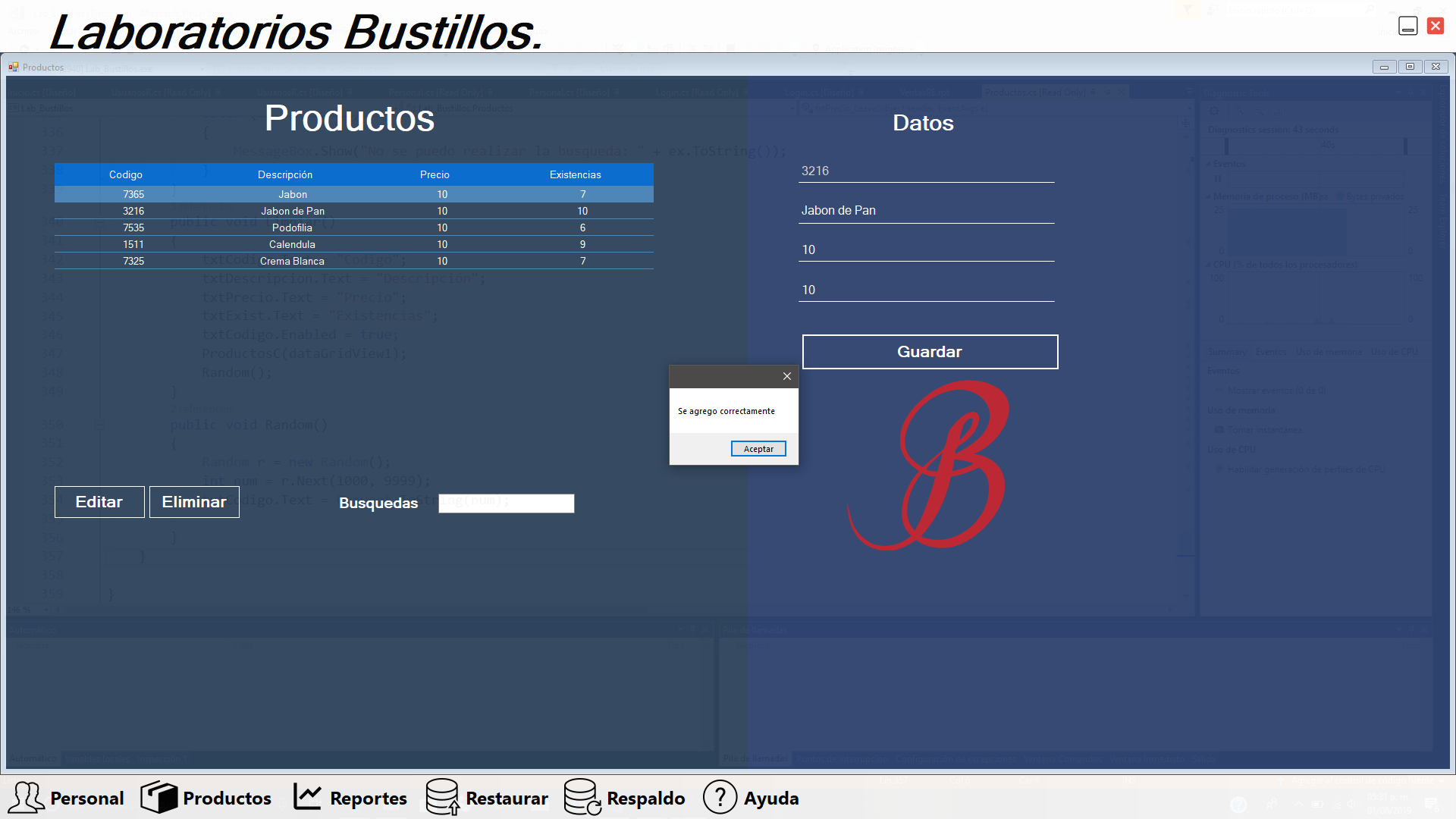
txtCodigo.Text = Convert.ToString(num);

}

}

}

## Evidencia de ejecución de caso 3.



# Documento Diseño de Prueba:

|  |  |  |
| --- | --- | --- |
| No. Escenario. | Escenario | Descripción |
| 1.- | Login | Acceso a los usuario |
| 2.- | Registro de Usuarios | Registro de los Usuarios para el acceso al sistema. |
| 3.- | Registro de Productos | Registro de los productos para su venta. |

# Matriz de Trazabilidad de Pruebas.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Id de Condicion de Pruebas. | Casos de Pruebas. | Requerimiento. | Regla de Negocio. | Requerimiento anterior. | Documento técnico. | Otro Documento. |
| 1 | Logueo de Usuarios | En la pantalla de incio hacia el sistema de información | La pantalla cuenta con el campo de usuario y contraseña | El usuario y contraseña deben ser validos | Todo usuario y contraseña deben estar previamen te almacenad os en el sistema |  |

# Referencias

El desarrollo de este documento fue posible gracias a las pruebas anteriormente realizadas durante la clase que sirvieron como base de este documento y así concretar las ideas de mejor manera, dando formalidad a las pruebas ya realizadas y documentadas.