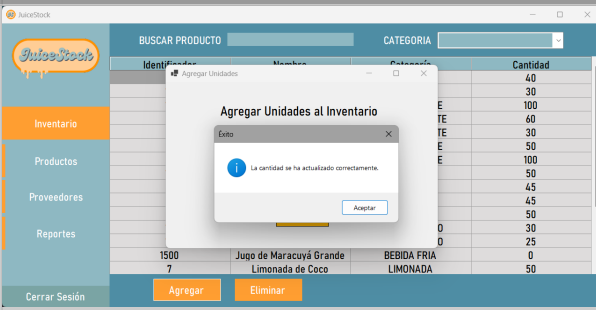
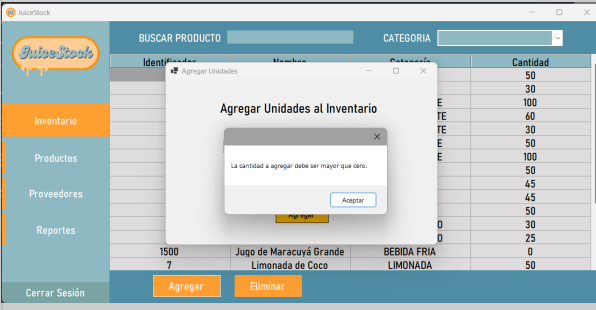
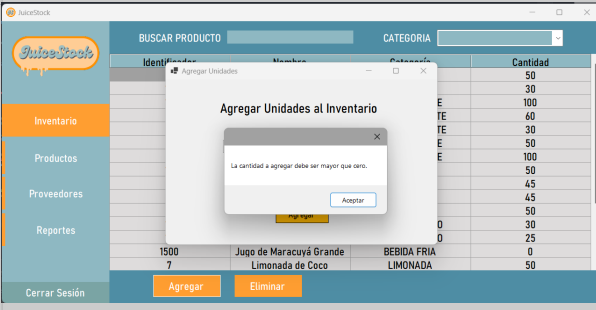


ID-TC	Elemento	Datos de prueba	Resultado Esperado	Resultado Real	Tipo
HE5-HU1-CA1	<pre>private void btnAgregar_Click(object sender, EventArgs e) { try { // Verificar si se ha seleccionado un producto if (cmbProducto.SelectedIndex == -1) { MessageBox.Show("Por favor, seleccione un producto."); return; // Salir del método si no hay un producto seleccionado } // Obtener el nombre del producto seleccionado string nombreProductoSeleccionado = cmbProducto.SelectedItem.ToString(); int cantidadASumar = Convert.ToInt32(numCantidad.Value); // Verificar que la cantidad a sumar sea mayor que cero if (cantidadASumar <= 0) { MessageBox.Show("La cantidad a agregar debe ser mayor que cero."); return; // Salir del método si la cantidad es cero o negativa } // Crear la conexión using (OracleConnection conexion = Conexion.GetInstance().CrearConexion()) { // Crear el comando para llamar al procedimiento almacenado using (OracleCommand comando = new OracleCommand("agregar_cantidad_procedimiento", conexion)) { comando.CommandType = CommandType.StoredProcedure; // Añadir los parámetros al comando comando.Parameters.Add(new OracleParameter("p_nombre_prod", nombreProductoSeleccionado)); comando.Parameters.Add(new OracleParameter("p_cantidad_agregar", cantidadASumar)); // Abrir la conexión y ejecutar el comando conexion.Open(); comando.ExecuteNonQuery(); } } // Mensaje de éxito MessageBox.Show("La cantidad se ha actualizado correctamente.", "Éxito", MessageBoxButtons.OK, MessageBoxIcon.Information); // Cerrar el formulario de agregar inventario this.Close(); // Actualizar el formulario de inventario var frmInventario = (frmInventario)Owner; // Obtener el formulario padre (frmInventario) frmInventario.ActualizarTabla(); // Llamar al método para actualizar la tabla } catch (Exception ex) { MessageBox.Show("Error al actualizar la cantidad: " + ex.Message); } }</pre>	<div><div>nombreProductoSeleccionado</div><div>cantidadASumar</div></div> <div>"Avena Fría"</div> <div>10</div>			+
	<pre>private void btnAgregar_Click(object sender, EventArgs e) { try { // Verificar si se ha seleccionado un producto if (cmbProducto.SelectedIndex == -1) { MessageBox.Show("Por favor, seleccione un producto."); return; // Salir del método si no hay un producto seleccionado } // Obtener el nombre del producto seleccionado string nombreProductoSeleccionado = cmbProducto.SelectedItem.ToString(); int cantidadASumar = Convert.ToInt32(numCantidad.Value); // Verificar que la cantidad a sumar sea mayor que cero if (cantidadASumar <= 0) { MessageBox.Show("La cantidad a agregar debe ser mayor que cero."); return; // Salir del método si la cantidad es cero o negativa } // Crear la conexión using (OracleConnection conexion = Conexion.GetInstance().CrearConexion()) { // Crear el comando para llamar al procedimiento almacenado using (OracleCommand comando = new OracleCommand("agregar_cantidad_procedimiento", conexion)) { comando.CommandType = CommandType.StoredProcedure; // Añadir los parámetros al comando comando.Parameters.Add(new OracleParameter("p_nombre_prod", nombreProductoSeleccionado)); comando.Parameters.Add(new OracleParameter("p_cantidad_agregar", cantidadASumar)); // Abrir la conexión y ejecutar el comando conexion.Open(); comando.ExecuteNonQuery(); } } // Mensaje de éxito MessageBox.Show("La cantidad se ha actualizado correctamente.", "Éxito", MessageBoxButtons.OK, MessageBoxIcon.Information); // Cerrar el formulario de agregar inventario this.Close(); // Actualizar el formulario de inventario var frmInventario = (frmInventario)Owner; // Obtener el formulario padre (frmInventario) frmInventario.ActualizarTabla(); // Llamar al método para actualizar la tabla } catch (Exception ex) { MessageBox.Show("Error al actualizar la cantidad: " + ex.Message); } }</pre>	<div><div>nombreProductoSeleccionado</div><div>cantidadASumar</div></div> <div>"Helado de Chocolate"</div> <div>0</div>			+
HE5-HU1-CA2	<pre>private void btnAgregar_Click(object sender, EventArgs e) { try { // Verificar si se ha seleccionado un producto if (cmbProducto.SelectedIndex == -1) { MessageBox.Show("Por favor, seleccione un producto."); return; // Salir del método si no hay un producto seleccionado } // Obtener el nombre del producto seleccionado string nombreProductoSeleccionado = cmbProducto.SelectedItem.ToString(); int cantidadASumar = Convert.ToInt32(numCantidad.Value); // Verificar que la cantidad a sumar sea mayor que cero if (cantidadASumar <= 0) { MessageBox.Show("La cantidad a agregar debe ser mayor que cero."); return; // Salir del método si la cantidad es cero o negativa } // Crear la conexión using (OracleConnection conexion = Conexion.GetInstance().CrearConexion()) { // Crear el comando para llamar al procedimiento almacenado using (OracleCommand comando = new OracleCommand("agregar_cantidad_procedimiento", conexion)) { comando.CommandType = CommandType.StoredProcedure; // Añadir los parámetros al comando comando.Parameters.Add(new OracleParameter("p_nombre_prod", nombreProductoSeleccionado)); comando.Parameters.Add(new OracleParameter("p_cantidad_agregar", cantidadASumar)); // Abrir la conexión y ejecutar el comando conexion.Open(); comando.ExecuteNonQuery(); } } // Mensaje de éxito MessageBox.Show("La cantidad se ha actualizado correctamente.", "Éxito", MessageBoxButtons.OK, MessageBoxIcon.Information); // Cerrar el formulario de agregar inventario this.Close(); // Actualizar el formulario de inventario var frmInventario = (frmInventario)Owner; // Obtener el formulario padre (frmInventario) frmInventario.ActualizarTabla(); // Llamar al método para actualizar la tabla } catch (Exception ex) { MessageBox.Show("Error al actualizar la cantidad: " + ex.Message); } }</pre>	<div><div>nombreProductoSeleccionado</div><div>cantidadASumar</div></div> <div>"Helado de Chocolate"</div> <div>0</div>			+

SPRINT 1

HE5-HU2-CA1

```
private void btnEliminarProducto_Click(object sender, EventArgs e)
{
    try
    {
        if (cmbProductos.SelectedIndex == -1)
        {
            MessageBox.Show("Por favor, seleccione un producto.");
            return; // Salir del método si no hay un producto seleccionado
        }
        // Obtener el nombre del producto seleccionado
        string nombreProductoSeleccionado = cmbProductos.SelectedItem.ToString();
        int cantidadRestar = Convert.ToInt32(numCantidad.Value);
        // Verificar que la cantidad a restar sea mayor que cero
        if (cantidadRestar <= 0)
        {
            MessageBox.Show("La cantidad a restar debe ser mayor que cero.");
            return; // Salir del método si la cantidad es cero o negativa
        }
        // Obtener la cantidad actual del producto
        int cantidadActual = ObtenerCantidadProducto(nombreProductoSeleccionado);
        // Verificar si la cantidad a eliminar es mayor que la cantidad actual
        if (cantidadRestar > cantidadActual)
        {
            MessageBox.Show("No se puede eliminar una cantidad mayor a la existente.");
            return; // Salir del método si la cantidad a eliminar es mayor que la cantidad actual
        }
        // Crear la conexión
        using (OracleConnection conexion = Conexion.GetInstance().CrearConexion())
        {
            // Crear el comando para llamar al procedimiento almacenado
            using (OracleCommand comando = new OracleCommand("eliminar_cantidad_procedimiento", conexion))
            {
                comando.CommandType = CommandType.StoredProcedure;

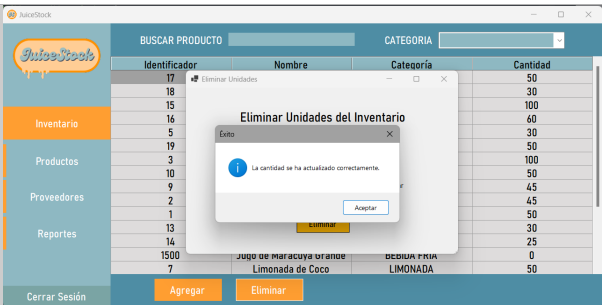
                // Añadir los parámetros al comando
                comando.Parameters.Add(new OracleParameter("p_nombre_prod", nombreProductoSeleccionado));
                comando.Parameters.Add(new OracleParameter("p_cantidad_eliminar", cantidadRestar));

                // Abrir la conexión y ejecutar el comando
                conexion.Open();
                comando.ExecuteNonQuery();
            }
        }
        // Mensaje de éxito
        MessageBox.Show("La cantidad se ha actualizado correctamente.",
            "Éxito",
            MessageBoxButtons.OK,
            MessageBoxIcon.Information);

        // Cerrar el formulario de agregar inventario
        this.Close();
    }
    catch (Exception ex)
    {
        MessageBox.Show("Error al actualizar la cantidad: " + ex.Message);
    }
}
```

nombreProductoSeleccionado	"Galletas de Avena"
cantidadARestar	50
cantidadActual	100

Se elimina la cantidad correctamente del producto seleccionado y se muestra el mensaje "La cantidad se ha actualizado correctamente."



+

HE5-HU2-CA3

```
private void btnEliminarProducto_Click(object sender, EventArgs e)
{
    try
    {
        if (cmbProductos.SelectedIndex == -1)
        {
            MessageBox.Show("Por favor, seleccione un producto.");
            return; // Salir del método si no hay un producto seleccionado
        }
        // Obtener el nombre del producto seleccionado
        string nombreProductoSeleccionado = cmbProductos.SelectedItem.ToString();
        int cantidadRestar = Convert.ToInt32(numCantidad.Value);
        // Verificar que la cantidad a restar sea mayor que cero
        if (cantidadRestar <= 0)
        {
            MessageBox.Show("La cantidad a restar debe ser mayor que cero.");
            return; // Salir del método si la cantidad es cero o negativa
        }
        // Obtener la cantidad actual del producto
        int cantidadActual = ObtenerCantidadProducto(nombreProductoSeleccionado);
        // Verificar si la cantidad a eliminar es mayor que la cantidad actual
        if (cantidadRestar > cantidadActual)
        {
            MessageBox.Show("No se puede eliminar una cantidad mayor a la existente.");
            return; // Salir del método si la cantidad a eliminar es mayor que la cantidad actual
        }
        // Crear la conexión
        using (OracleConnection conexion = Conexion.GetInstance().CrearConexion())
        {
            // Crear el comando para llamar al procedimiento almacenado
            using (OracleCommand comando = new OracleCommand("eliminar_cantidad_procedimiento", conexion))
            {
                comando.CommandType = CommandType.StoredProcedure;

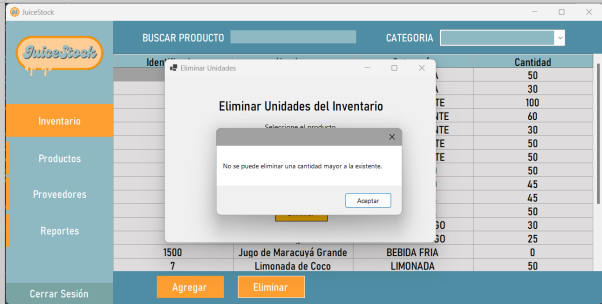
                // Añadir los parámetros al comando
                comando.Parameters.Add(new OracleParameter("p_nombre_prod", nombreProductoSeleccionado));
                comando.Parameters.Add(new OracleParameter("p_cantidad_eliminar", cantidadRestar));

                // Abrir la conexión y ejecutar el comando
                conexion.Open();
                comando.ExecuteNonQuery();
            }
        }
        // Mensaje de éxito
        MessageBox.Show("La cantidad se ha actualizado correctamente.",
            "Éxito",
            MessageBoxButtons.OK,
            MessageBoxIcon.Information);

        // Cerrar el formulario de agregar inventario
        this.Close();
    }
    catch (Exception ex)
    {
        MessageBox.Show("Error al actualizar la cantidad: " + ex.Message);
    }
}
```

nombreProductoSeleccionado	"Granizado de Fresa"
cantidadARestar	51
cantidadActual	50

El programa muestra el mensaje de que la cantidad a eliminar debe ser menor a la cantidad disponible en el inventario



+

HE2-HU1-CA1

```
private void BotonAgregarProducto_Click(object sender, EventArgs e)
{
    try
    {
        // Verificar si se ha ingresado un nombre, un precio, se ha seleccionado un proveedor o una categoría
        if (comboBoxProveedor.SelectedIndex == -1 || textBoxCategoria.SelectedIndex == -1 || textBoxNombre.Text == "" || textBoxPrecio.Text == "")
        {
            MessageBox.Show("Error", "Faltan datos", MessageBoxButtons.OK, MessageBoxIcon.Warning);
            return;
        }
        // Obtener el nombre del producto nuevo
        string nombreProd = textBoxNombre.Text;

        // Obtener el precio del producto nuevo
        int precioProd = int.Parse(textBoxPrecio.Text);

        // Obtener el nombre del proveedor seleccionado
        string nombreProveedorSeleccionado = comboBoxProveedor.SelectedItem.ToString();

        // Obtener el nombre de la categoría seleccionada
        string nombreCategoriaSeleccionado = comboBoxCategoria.SelectedItem.ToString();

        // Crear la conexión
        using (OracleConnection conexion = Conexion.getConnection())
        {
            // Preparar la consulta
            // Crear el comando para llamar al procedimiento almacenado
            using (OracleCommand comando = new OracleCommand("agregar_producto_nuevo", conexion))
            {
                comando.CommandType = CommandType.StoredProcedure;

                // Añadir los parámetros al comando
                comando.Parameters.Add(new OracleParameter("nombre_producto", nombreProd));
                comando.Parameters.Add(new OracleParameter("precio_producto", precioProd));
                comando.Parameters.Add(new OracleParameter("nombre_proveedor", nombreProveedorSeleccionado));
                comando.Parameters.Add(new OracleParameter("nombre_categoria", nombreCategoriaSeleccionado));

                // Ejecutar la consulta de salida
                OracleParameterCollection parametros = new OracleParameterCollection();
                parametros.Add(new OracleParameter("mensaje_salida", OracleDbType.Varchar2));
                comando.Parameters.AddRange(parametros);

                // Ejecutar la consulta y ejecutar el comando
                comando.ExecuteNonQuery();

                // Mostrar el valor del parámetro de salida
                if (comando.Parameters["mensaje_salida"].Value != null)
                {
                    MessageBox.Show(comando.Parameters["mensaje_salida"].Value.ToString());
                }
            }
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}

private void BotonEliminarProducto_Click(object sender, EventArgs e)
{
    try
    {
        // Verificar si se ha ingresado un nombre, un precio, se ha seleccionado un proveedor o una categoría
        if (comboBoxProveedor.SelectedIndex == -1 || textBoxCategoria.SelectedIndex == -1 || textBoxNombre.Text == "" || textBoxPrecio.Text == "")
        {
            MessageBox.Show("Error", "Faltan datos", MessageBoxButtons.OK, MessageBoxIcon.Warning);
            return;
        }
        // Obtener el nombre del producto nuevo
        string nombreProd = textBoxNombre.Text;

        // Obtener el precio del producto nuevo
        int precioProd = int.Parse(textBoxPrecio.Text);

        // Obtener el nombre del proveedor seleccionado
        string nombreProveedorSeleccionado = comboBoxProveedor.SelectedItem.ToString();

        // Obtener el nombre de la categoría seleccionada
        string nombreCategoriaSeleccionado = comboBoxCategoria.SelectedItem.ToString();

        // Crear la conexión
        using (OracleConnection conexion = Conexion.getConnection())
        {
            // Preparar la consulta
            // Crear el comando para llamar al procedimiento almacenado
            using (OracleCommand comando = new OracleCommand("eliminar_producto_nuevo", conexion))
            {
                comando.CommandType = CommandType.StoredProcedure;

                // Añadir los parámetros al comando
                comando.Parameters.Add(new OracleParameter("nombre_producto", nombreProd));
                comando.Parameters.Add(new OracleParameter("precio_producto", precioProd));
                comando.Parameters.Add(new OracleParameter("nombre_proveedor", nombreProveedorSeleccionado));
                comando.Parameters.Add(new OracleParameter("nombre_categoria", nombreCategoriaSeleccionado));

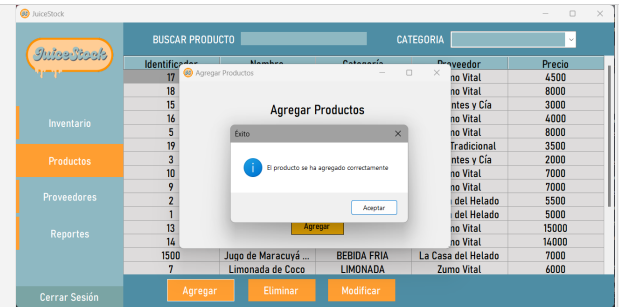
                // Ejecutar la consulta de salida
                OracleParameterCollection parametros = new OracleParameterCollection();
                parametros.Add(new OracleParameter("mensaje_salida", OracleDbType.Varchar2));
                comando.Parameters.AddRange(parametros);

                // Ejecutar la consulta y ejecutar el comando
                comando.ExecuteNonQuery();

                // Mostrar el valor del parámetro de salida
                if (comando.Parameters["mensaje_salida"].Value != null)
                {
                    MessageBox.Show(comando.Parameters["mensaje_salida"].Value.ToString());
                }
            }
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}
```

nombreProd	"Jugo de Borojó"
precioProd	3000
nombreProveedorSeleccionado	"Zumo Vital"
nombreCategoriaSeleccionado	"VASO DE JUGO"

El programa muestra el mensaje de que el producto se agregó correctamente



HE2-HU1-CA2

```
private void BotonAgregarProducto_Click(object sender, EventArgs e)
{
    try
    {
        // Verificar si se ha ingresado un nombre, un precio, se ha seleccionado un proveedor o una categoría
        if (comboBoxProveedor.SelectedIndex == -1 || textBoxCategoria.SelectedIndex == -1 || textBoxNombre.Text == "" || textBoxPrecio.Text == "")
        {
            MessageBox.Show("Error", "Faltan datos", MessageBoxButtons.OK, MessageBoxIcon.Warning);
            return;
        }
        // Obtener el nombre del producto nuevo
        string nombreProd = textBoxNombre.Text;

        // Obtener el precio del producto nuevo
        int precioProd = int.Parse(textBoxPrecio.Text);

        // Obtener el nombre del proveedor seleccionado
        string nombreProveedorSeleccionado = comboBoxProveedor.SelectedItem.ToString();

        // Obtener el nombre de la categoría seleccionada
        string nombreCategoriaSeleccionado = comboBoxCategoria.SelectedItem.ToString();

        // Crear la conexión
        using (OracleConnection conexion = Conexion.getConnection())
        {
            // Preparar la consulta
            // Crear el comando para llamar al procedimiento almacenado
            using (OracleCommand comando = new OracleCommand("agregar_producto_nuevo", conexion))
            {
                comando.CommandType = CommandType.StoredProcedure;

                // Añadir los parámetros al comando
                comando.Parameters.Add(new OracleParameter("nombre_producto", nombreProd));
                comando.Parameters.Add(new OracleParameter("precio_producto", precioProd));
                comando.Parameters.Add(new OracleParameter("nombre_proveedor", nombreProveedorSeleccionado));
                comando.Parameters.Add(new OracleParameter("nombre_categoria", nombreCategoriaSeleccionado));

                // Ejecutar la consulta de salida
                OracleParameterCollection parametros = new OracleParameterCollection();
                parametros.Add(new OracleParameter("mensaje_salida", OracleDbType.Varchar2));
                comando.Parameters.AddRange(parametros);

                // Ejecutar la consulta y ejecutar el comando
                comando.ExecuteNonQuery();

                // Mostrar el valor del parámetro de salida
                if (comando.Parameters["mensaje_salida"].Value != null)
                {
                    MessageBox.Show(comando.Parameters["mensaje_salida"].Value.ToString());
                }
            }
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}

private void BotonEliminarProducto_Click(object sender, EventArgs e)
{
    try
    {
        // Verificar si se ha ingresado un nombre, un precio, se ha seleccionado un proveedor o una categoría
        if (comboBoxProveedor.SelectedIndex == -1 || textBoxCategoria.SelectedIndex == -1 || textBoxNombre.Text == "" || textBoxPrecio.Text == "")
        {
            MessageBox.Show("Error", "Faltan datos", MessageBoxButtons.OK, MessageBoxIcon.Warning);
            return;
        }
        // Obtener el nombre del producto nuevo
        string nombreProd = textBoxNombre.Text;

        // Obtener el precio del producto nuevo
        int precioProd = int.Parse(textBoxPrecio.Text);

        // Obtener el nombre del proveedor seleccionado
        string nombreProveedorSeleccionado = comboBoxProveedor.SelectedItem.ToString();

        // Obtener el nombre de la categoría seleccionada
        string nombreCategoriaSeleccionado = comboBoxCategoria.SelectedItem.ToString();

        // Crear la conexión
        using (OracleConnection conexion = Conexion.getConnection())
        {
            // Preparar la consulta
            // Crear el comando para llamar al procedimiento almacenado
            using (OracleCommand comando = new OracleCommand("eliminar_producto_nuevo", conexion))
            {
                comando.CommandType = CommandType.StoredProcedure;

                // Añadir los parámetros al comando
                comando.Parameters.Add(new OracleParameter("nombre_producto", nombreProd));
                comando.Parameters.Add(new OracleParameter("precio_producto", precioProd));
                comando.Parameters.Add(new OracleParameter("nombre_proveedor", nombreProveedorSeleccionado));
                comando.Parameters.Add(new OracleParameter("nombre_categoria", nombreCategoriaSeleccionado));

                // Ejecutar la consulta de salida
                OracleParameterCollection parametros = new OracleParameterCollection();
                parametros.Add(new OracleParameter("mensaje_salida", OracleDbType.Varchar2));
                comando.Parameters.AddRange(parametros);

                // Ejecutar la consulta y ejecutar el comando
                comando.ExecuteNonQuery();

                // Mostrar el valor del parámetro de salida
                if (comando.Parameters["mensaje_salida"].Value != null)
                {
                    MessageBox.Show(comando.Parameters["mensaje_salida"].Value.ToString());
                }
            }
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}
```

nombreProd	"Jugo de Piña"
precioProd	2500
nombreProveedorSeleccionado	"Zumo Vital"
nombreCategoriaSeleccionado	null

El programa muestra el mensaje de que se deben ingresar todos los datos.



SPRINT 2

HE2-HU2-CA1

```
private void btnEliminarProducto_Click(object sender, EventArgs e)
{
    try
    {
        if (cmbProductos.SelectedItem == null)
        {
            MessageBox.Show("Por favor, seleccione un producto.");
            return; // Salir del método si no hay un producto seleccionado
        }
        string nombreProductoSeleccionado = cmbProductos.SelectedItem.ToString();
        int cantidadEliminada = 0;

        // Mensaje de confirmación
        DialogResult confirmacion = MessageBox.Show(
            "Se eliminará el producto: " + nombreProductoSeleccionado + ". ¿Está seguro?",
            "Confirmar eliminación",
            MessageBoxButtons.YesNo,
            MessageBoxIcon.Warning
        );

        if (confirmacion == DialogResult.No)
        {
            // Si selecciona "No", se cancela la eliminación
            return;
        }
        // Cerrar la conexión
        using (OracleConnection conexion = Conexion.getInstancia()) { Conexion.cerrar(); }

        // Crear el comando para borrar el procedimiento almacenado
        using (OracleCommand comando = new OracleCommand("eliminar_producto_pro", conexion))
        {
            comando.CommandType = CommandType.StoredProcedure;
            comando.Parameters.AddWithValue(OracleParameterType.VarChar, nombreProductoSeleccionado);

            // Mensaje de advertencia de salud
            OracleParameter parametroSalud = new OracleParameter("p_salud", OracleType.VarChar, 100);
            parametroSalud.Direction = ParameterDirection.Output;
            comando.Parameters.Add(parametroSalud);

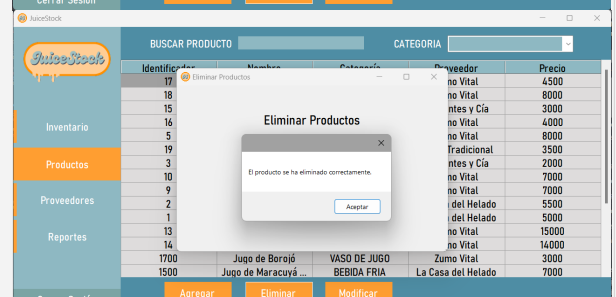
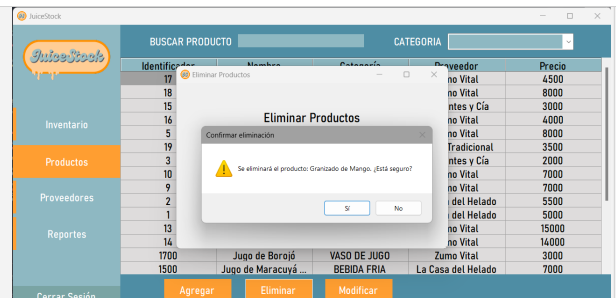
            // Ejecutar la consulta y ejecutar el comando
            comando.ExecuteNonQuery();

            // Mostrar el valor del parámetro de salud
            if (parametroSalud.Value != null)
            {
                MessageBox.Show("El producto se ha eliminado correctamente.");
            }
        }

        if (cantidadEliminada == 0)
        {
            // Mensaje de éxito
            MessageBox.Show("El producto se ha eliminado correctamente.");
        }
        else if (cantidadEliminada == 1)
        {
            // Mensaje de éxito (producto tiene cantidad en inventario)
            MessageBox.Show("El producto tiene cantidad en inventario.");
        }
        else
        {
            // Mensaje de error (producto no tiene cantidad en inventario)
            MessageBox.Show("No se pudo eliminar el producto.");
        }
    }
    catch { }
}
```

nombreProductoSeleccionado "Granizado de Mango"

El programa muestra el mensaje de que el producto se eliminó correctamente



HE2-HU2-CA2

```
private void btnEliminarProducto_Click(object sender, EventArgs e)
{
    try
    {
        if (cmbProductos.SelectedItem == null)
        {
            MessageBox.Show("Por favor, seleccione un producto.");
            return; // Salir del método si no hay un producto seleccionado
        }
        string nombreProductoSeleccionado = cmbProductos.SelectedItem.ToString();
        int cantidadEliminada = 0;

        // Mensaje de confirmación
        DialogResult confirmacion = MessageBox.Show(
            "Se eliminará el producto: " + nombreProductoSeleccionado + ". ¿Está seguro?",
            "Confirmar eliminación",
            MessageBoxButtons.YesNo,
            MessageBoxIcon.Warning
        );

        if (confirmacion == DialogResult.No)
        {
            // Si selecciona "No", se cancela la eliminación
            return;
        }
        // Cerrar la conexión
        using (OracleConnection conexion = Conexion.getInstancia()) { Conexion.cerrar(); }

        // Crear el comando para borrar el procedimiento almacenado
        using (OracleCommand comando = new OracleCommand("eliminar_producto_pro", conexion))
        {
            comando.CommandType = CommandType.StoredProcedure;
            comando.Parameters.AddWithValue(OracleParameterType.VarChar, nombreProductoSeleccionado);

            // Mensaje de advertencia de salud
            OracleParameter parametroSalud = new OracleParameter("p_salud", OracleType.VarChar, 100);
            parametroSalud.Direction = ParameterDirection.Output;
            comando.Parameters.Add(parametroSalud);

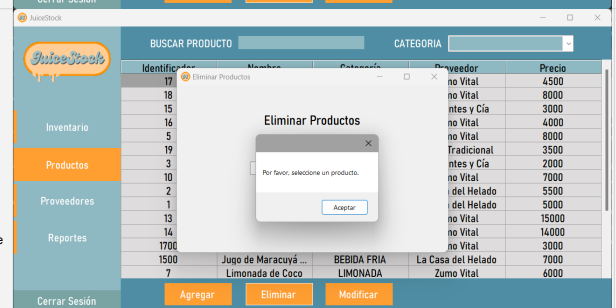
            // Ejecutar la consulta y ejecutar el comando
            comando.ExecuteNonQuery();

            // Mostrar el valor del parámetro de salud
            if (parametroSalud.Value != null)
            {
                MessageBox.Show("El producto se ha eliminado correctamente.");
            }
        }

        if (cantidadEliminada == 0)
        {
            // Mensaje de éxito
            MessageBox.Show("El producto se ha eliminado correctamente.");
        }
        else if (cantidadEliminada == 1)
        {
            // Mensaje de éxito (producto tiene cantidad en inventario)
            MessageBox.Show("El producto tiene cantidad en inventario.");
        }
        else
        {
            // Mensaje de error (producto no tiene cantidad en inventario)
            MessageBox.Show("No se pudo eliminar el producto.");
        }
    }
    catch { }
}
```

nombreProductoSeleccionado null

El programa deberá mostrar un mensaje indicándole al usuario que debe seleccionar un producto



HE5-HU3-CA1

```
consultaFiltrada = "SELECT ID_PROD 'Identificador',  
NOMBRE_PROD 'Nombre', NOMBRE_CATEGORIA  
'Categoria', CANTIDAD_PROD 'Cantidad' FROM  
VISTA_PRODUCTOS WHERE upper(NOMBRE_PROD)  
LIKE '%GRANIZADO%' ORDER BY CASE WHEN upper  
(NOMBRE_PROD) LIKE '%GRANIZADO%' THEN 1 WHEN  
upper(NOMBRE_PROD) LIKE '%GRANIZADO%' THEN 2  
ELSE 3 END, NOMBRE_PROD ASC"
```

+

[illegible]

```
consultaFiltrada = "SELECT ID_PROD 'Identificador',  
NOMBRE_PROD 'Nombre', NOMBRE_CATEGORIA  
'Categoria', CANTIDAD_PROD 'Cantidad' FROM  
VISTA_PRODUCTOS WHERE upper  
(NOMBRE_CATEGORIA) LIKE '%HELADERIA%' ORDER  
BY NOMBRE_CATEGORIA ASC"
```

+


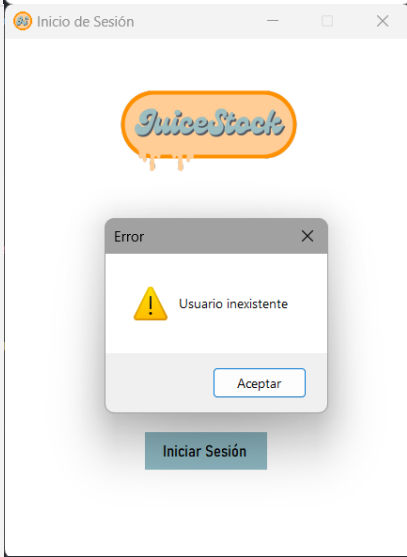
HE3-HU1-CA1

•

+

+

•

	<pre>private int verificarCredenciales(string usuario, string contraseña) { int resultado = 0; try { using (OracleConnection conexion = Conexion.GetInstance().CrearConexion()) { using (OracleCommand comando = new OracleCommand("verificar_credenciales", conexion)) { comando.CommandType = CommandType.StoredProcedure; // Parámetros de entrada comando.Parameters.Add(new OracleParameter("p_usuario", usuario)); comando.Parameters.Add(new OracleParameter("p_contraseña", contraseña)); // Parámetro de salida OracleParameter resultadoParam = new OracleParameter("p_resultado", OracleDbType.Int32); resultadoParam.Direction = ParameterDirection.Output; comando.Parameters.Add(resultadoParam); conexion.Open(); comando.ExecuteNonQuery(); if (resultadoParam.Value != DBNull.Value) { Oracle.ManagedDataAccess.Types.OracleDecimal oracleDecimal = (Oracle.ManagedDataAccess.Types.OracleDecimal)resultadoParam.Value; resultado = oracleDecimal.ToInt32(); } } } } catch (Exception ex) { MessageBox.Show(\$"Error al verificar credenciales: {ex.Message}", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error); } return resultado; }</pre>	<table><tr><td>usuario</td><td>"admin"</td></tr><tr><td>contraseña</td><td>null</td></tr></table>	usuario	"admin"	contraseña	null	El programa deberá mostrar un mensaje indicándole al usuario que debe llenar todos los datos para iniciar sesión		+
usuario	"admin"								
contraseña	null								
SPRINT 4	<pre>private int VerificarCredenciales(string usuario, string contraseña) { int resultado = 0; try { using (OracleConnection conexion = Conexion.GetInstance().CrearConexion()) { using (OracleCommand comando = new OracleCommand("verificar_credenciales", conexion)) { comando.CommandType = CommandType.StoredProcedure; // Parámetros de entrada comando.Parameters.Add(new OracleParameter("p_usuario", usuario)); comando.Parameters.Add(new OracleParameter("p_contraseña", contraseña)); // Parámetro de salida OracleParameter resultadoParam = new OracleParameter("p_resultado", OracleDbType.Int32); resultadoParam.Direction = ParameterDirection.Output; comando.Parameters.Add(resultadoParam); conexion.Open(); comando.ExecuteNonQuery(); if (resultadoParam.Value != DBNull.Value) { Oracle.ManagedDataAccess.Types.OracleDecimal oracleDecimal = (Oracle.ManagedDataAccess.Types.OracleDecimal)resultadoParam.Value; resultado = oracleDecimal.ToInt32(); } } } } catch (Exception ex) { MessageBox.Show(\$"Error al verificar credenciales: {ex.Message}", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error); } return resultado; }</pre>	<table><tr><td>usuario</td><td>"ines"</td></tr><tr><td>contraseña</td><td>"1234"</td></tr></table>	usuario	"ines"	contraseña	"1234"	El programa deberá mostrar un mensaje indicándole al usuario que el nombre de usuario no existe o no está registrado.		+
usuario	"ines"								
contraseña	"1234"								

HE3-HU1-CA4

```
private int VerificarCredenciales(string usuario, string contraseña)
{
    int resultado = 0;
    try
    {
        using (OracleConnection conexion = Conexion.getInstance().CrearConexion())
        {
            using (OracleCommand comando = new OracleCommand("vverificar_credenciales", conexion))
            {
                comando.CommandType = CommandType.StoredProcedure;

                // Parámetros de entrada
                comando.Parameters.Add(new OracleParameter("p_usuario", usuario));
                comando.Parameters.Add(new OracleParameter("p_contraseña", contraseña));

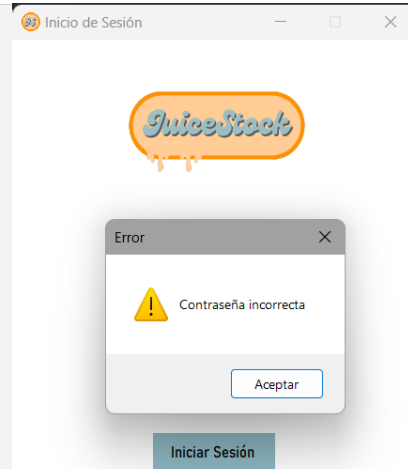
                // Parámetro de salida
                OracleParameter resultadoParam = new OracleParameter("p_resultado", OracleDbType.Int32);
                resultadoParam.Direction = ParameterDirection.Output;
                comando.Parameters.Add(resultadoParam);

                conexion.Open();
                comando.ExecuteNonQuery();

                if (resultadoParam.Value != DBNull.Value)
                {
                    Oracle ManagedDataAccess.Types.OracleDecimal oracleDecimal =
                        (Oracle ManagedDataAccess.Types.OracleDecimal)resultadoParam.Value;
                    resultado = oracleDecimal.ToInt32();
                }
            }
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show($"Error al verificar credenciales: {ex.Message}", "Error",
            MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
    return resultado;
}
```

usuario	"admin"
contraseña	"admin"

El programa deberá mostrar un mensaje indicándole al usuario que la contraseña es incorrecta



HE4-HU1-CA1

```
private void btnAgregarProveedor_Click(object sender, EventArgs e)
{
    try
    {
        // Verificar si se ha ingresado un nombre, un teléfono, un correo o una dirección
        if (textBox1.Text.Length == 0 || textBox2.Text.Length == 0 || textBox3.Text.Length == 0 || textBox4.Text.Length == 0)
        {
            MessageBox.Show("Debe ingresar los datos de los proveedores", "Error",
                MessageBoxButtons.OK, MessageBoxIcon.Warning);
            return; // Salir del método si no hay un producto seleccionado
        }

        // Obtener el nombre del proveedor nuevo
        string nombreProv = textBox1.Text;

        // Obtener el teléfono del proveedor nuevo
        string telefono = textBox2.Text;

        // Obtener el correo
        string correo = textBox3.Text;

        // Obtener el nombre de la corporación seleccionada
        string direccion = textBox4.Text;

        // Seleccionar la opción de agregar
        int btnAgregar = 0;

        // Crear la conexión
        using (OracleConnection conexion = Conexion.getInstance().CrearConexion())
        {
            // Preparar el comando
            // Crear el comando para llamar al procedimiento almacenado
            using (OracleCommand comando = new OracleCommand("agregar_proveedor", conexion))
            {
                comando.CommandType = CommandType.StoredProcedure;

                // Añadir los parámetros al comando
                comando.Parameters.Add(new OracleParameter("p_nombre", nombreProv));
                comando.Parameters.Add(new OracleParameter("p_telefono", telefono));
                comando.Parameters.Add(new OracleParameter("p_correo", correo));
                comando.Parameters.Add(new OracleParameter("p_direccion", direccion));

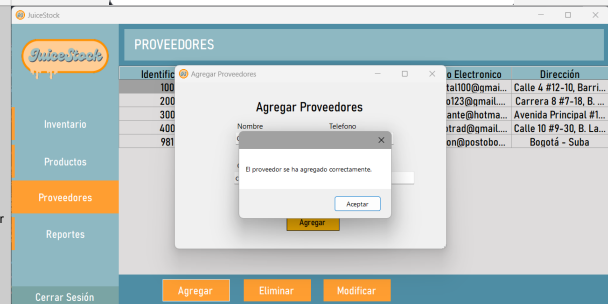
                // Añadir parámetros de salida
                OracleParameter resultadoParametro = new OracleParameter("p_resultado", OracleDbType.Int32);
                resultadoParametro.Direction = ParameterDirection.Output;
                comando.Parameters.Add(resultadoParametro);

                // Ejecutar la consulta y ejecutar el comando
                conexion.Open();
                comando.ExecuteNonQuery();

                // Recuperar el valor del parámetro de salida
                if (resultadoParametro.Value != DBNull.Value)
                {
                    Oracle ManagedDataAccess.Types.OracleDecimal oracleDecimal =
                        (Oracle ManagedDataAccess.Types.OracleDecimal)resultadoParametro.Value;
                    btnAgregar = oracleDecimal.ToInt32();
                }
            }
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show($"Error al agregar proveedor: {ex.Message}", "Error",
            MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
    else if (btnAgregar == 1)
    {
        // Mensaje de éxito
        MessageBox.Show("El proveedor se ha agregado correctamente.", "Éxito",
            MessageBoxButtons.OK, MessageBoxIcon.Information);
    }
}
```

nombreProv	"CocaCola"
telefono	"3004859347"
correo	"cocacola@cocacola.com"
direccion	"Manhattan"

El programa deberá indicar al usuario que el proveedor se ha podido agregar correctamente



SPRINT 5	<div>HE4-HU2-CA1</div> <pre>private void btnEliminarProveedor_Click(object sender, EventArgs e) { try { if (dataGridView1.SelectedRows.Count == 0) { MessageBox.Show("Por favor, seleccione un proveedor."); return; } string nombreProveedorSeleccionado = dataGridView1.SelectedRows[0].Cells[0].Text; int idProveedor = 0; // Mostrar mensaje de confirmación DialogResult confirmacion = MessageBox.Show("Se eliminará el proveedor: " + nombreProveedorSeleccionado + ". ¿Está seguro?", "Confirmar eliminación", MessageBoxButtons.YesNo, MessageBoxIcon.Warning); if (confirmacion == DialogResult.No) { // Si la selección fue "No", se cancela la eliminación return; } // Crear la conexión using (OracleConnection conexion = Conexion.GetInstance().CrearConexion()) { // Generar comando para eliminar el proveedor seleccionado using (OracleCommand comando = new OracleCommand("eliminar_proveedor", conexion)) { comando.CommandType = CommandType.StoredProcedure; // Añadir los parámetros de entrada comando.Parameters.Add(new OracleParameter("p_nombre_proveedor", nombreProveedorSeleccionado)); // Recibir parámetros de salida OracleParameter resultadoParametro = new OracleParameter("p_resultado", OracleDbType.Int32); resultadoParametro.Direction = ParameterDirection.Output; comando.Parameters.Add(resultadoParametro); // Ejecutar la consulta y asignar el comando comando.ExecuteNonQuery(); // Recuperar el valor del resultado de salida if (resultadoParametro.Value != DBNull.Value) { Oracle.ManagedDataAccess.Types.OracleInteger resultado = (Oracle.ManagedDataAccess.Types.OracleInteger)resultadoParametro.Value; idProveedor = resultado.ToInt32(); } } // Validar el resultado if (idProveedor == 0) { MessageBox.Show("El proveedor no ha sido eliminado correctamente.", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error); } else if (idProveedor != 0) { MessageBox.Show("El proveedor tiene productos asociados. No se puede eliminar.", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error); } else { MessageBox.Show("Se eliminó correctamente el proveedor.", "Éxito", MessageBoxButtons.OK, MessageBoxIcon.Information); } } } catch (Exception ex) { MessageBox.Show("Error al eliminar proveedor: " + ex.Message); } }</pre>	<div>nombreProveedorSeleccionado "CocaCola"</div> <div>El programa deberá indicar al usuario que el proveedor se ha podido eliminar correctamente</div>	<div>El programa deberá indicar al usuario que el proveedor se ha podido eliminar correctamente</div>
SPRINT 6	<div>HE1-HU1-CA1</div> <pre>private void btnFiltrarReportes_Click(object sender, EventArgs e) { try { // Formatear las fechas correctamente para Oracle string fecha_ini = dateTimePicker1.Value.ToString("dd/MM/yyyy", System.Globalization.CultureInfo.InvariantCulture); string fecha_fin = dateTimePicker2.Value.ToString("dd/MM/yyyy", System.Globalization.CultureInfo.InvariantCulture); // Construir la consulta SQL string consultaReportes = @" SELECT * FROM VISTA_MOVIMIENTOS_GENERAL WHERE TRUNC(FeCHA) BETWEEN TRUNC(TO_DATE(':" & fecha_ini & "', 'dd/mm/yyyy')) AND TRUNC(TO_DATE(':" & fecha_fin & "', 'dd/mm/yyyy'))"; // Llamar al método que carga los datos en el DataGridView D_Productos datos = new D_Productos(); dataGridView1.DataSource = datos.Llamar_datos(consultaReportes); catch (Exception ex) { MessageBox.Show("Error al filtrar reportes: " + ex.Message); } } }</pre>	<div>fecha_ini = "08/12/2024"</div> <div>fecha_fin = "10/12/2024"</div> <div>consultaReportes = "SELECT * FROM VISTA_MOVIMIENTOS_GENERAL WHERE TRUNC(FeCHA) BETWEEN TRUNC(TO_DATE('08/12/2024', 'DD/MM/YYYY')) AND TRUNC(TO_DATE('10/12/2024', 'DD/MM/YYYY'))"</div> <div>El programa mostrará la información correspondiente sobre una grilla de los movimientos realizados en un lapso específico de tiempo</div>	<div>El programa mostrará la información correspondiente sobre una grilla de los movimientos realizados en un lapso específico de tiempo</div>

HE2-HU3-CA1

```
private void btnModificarProducto_Click(object sender, EventArgs e)
{
    if (cmbProductos.SelectedIndex == -1)
    {
        lblSeleccionaProd.Visible = true;
        return; // Salir del método si no hay un producto seleccionado
    }

    // Verificar si se ha ingresado un nombre, un precio, si se ha seleccionado un proveedor o una categoría
    if (cmbProveedor.SelectedIndex == -1 || cmbCategoria.SelectedIndex == -1 || tabNombre.Text == "" || tabPrecio.Text == "")
    {
        lblIncompleto.Visible = true;
        return; // Salir del método si no hay un producto seleccionado
    }

    string productoSeleccionado = "";
    productoSeleccionado = cmbProductos.SelectedItem.ToString();

    // Enviar nuevos valores a procedimiento para modificar productos
    int bandera = 0;
    int nuevoPrecio = Convert.ToInt32(tabPrecio.Text);
    string p_nuevoNombre = tabNombre.Text;
    string p_nuevoProveedor = cmbProveedor.SelectedItem.ToString();
    string p_nuevaCategoria = cmbCategoria.SelectedItem.ToString();
    using (OracleConnection conexion = Conexion.getInstance().CrearConexion())
    {
        using (OracleCommand comando = new OracleCommand("modificar_producto_pro", conexion))
        {
            comando.CommandType = CommandType.StoredProcedure;

            // Parámetro de entrada
            comando.Parameters.Add("nombre_actual", OracleDbType.Varchar22, Value = productoSeleccionado);
            comando.Parameters.Add("nuevo_nombre", OracleDbType.Varchar22, Value = p_nuevoNombre);
            comando.Parameters.Add("nueva_categoria", OracleDbType.Varchar22, Value = p_nuevaCategoria);
            comando.Parameters.Add("nuevo_proveedor", OracleDbType.Varchar22, Value = p_nuevoProveedor);
            comando.Parameters.Add("nuevo_precio", OracleDbType.Int32, Value = nuevoPrecio);

            // Parámetro de salida
            OracleParameter banderaParam = new OracleParameter("p_bandera", OracleDbType.Int32, 32);
            banderaParam.Direction = ParameterDirection.Output;
            comando.Parameters.Add(banderaParam);

            // Ejecutar el procedimiento
            conexion.Open();
            comando.ExecuteNonQuery();

            // Retornar valor de parámetro de salida
            if (banderaParam.Value != DBNull.Value)
            {
                bandera = Convert.ToInt32(banderaParam.Value.ToString());
            }

            // Caso de error según resultado de bandera
            if (bandera == 0)
            {
                // Mensaje de éxito
                MessageBox.Show("El producto se ha modificado correctamente.");
            }
        }
    }
}
```

productoSeleccionado	"Té de Manzana"
bandera	0
nuevoPrecio	7000
p_nuevoNombre	"Té de Durazno"
p_nuevoProveedor	"Zumo Vital"
p_nuevaCategoria	"BEBIDA CALIENTE"

El programa mostrará un mensaje confirmando que los cambios de modificación han sido efectuados.



SPRINT 7

HE2-HU3-CA2

```
private void btnModificarProducto_Click(object sender, EventArgs e)
{
    if (cmbProductos.SelectedIndex == -1)
    {
        lblSeleccionaProd.Visible = true;
        return; // Salir del método si no hay un producto seleccionado
    }

    // Verificar si se ha ingresado un nombre, un precio, si se ha seleccionado un proveedor o una categoría
    if (cmbProveedor.SelectedIndex == -1 || cmbCategoria.SelectedIndex == -1 || tabNombre.Text == "" || tabPrecio.Text == "")
    {
        lblIncompleto.Visible = true;
        return; // Salir del método si no hay un producto seleccionado
    }

    string productoSeleccionado = "";
    productoSeleccionado = cmbProductos.SelectedItem.ToString();

    // Enviar nuevos valores a procedimiento para modificar productos
    int bandera = 0;
    int nuevoPrecio = Convert.ToInt32(tabPrecio.Text);
    string p_nuevoNombre = tabNombre.Text;
    string p_nuevoProveedor = cmbProveedor.SelectedItem.ToString();
    string p_nuevaCategoria = cmbCategoria.SelectedItem.ToString();
    using (OracleConnection conexion = Conexion.getInstance().CrearConexion())
    {
        using (OracleCommand comando = new OracleCommand("modificar_producto_pro", conexion))
        {
            comando.CommandType = CommandType.StoredProcedure;

            // Parámetro de entrada
            comando.Parameters.Add("nombre_actual", OracleDbType.Varchar22, Value = productoSeleccionado);
            comando.Parameters.Add("nuevo_nombre", OracleDbType.Varchar22, Value = p_nuevoNombre);
            comando.Parameters.Add("nueva_categoria", OracleDbType.Varchar22, Value = p_nuevaCategoria);
            comando.Parameters.Add("nuevo_proveedor", OracleDbType.Varchar22, Value = p_nuevoProveedor);
            comando.Parameters.Add("nuevo_precio", OracleDbType.Int32, Value = nuevoPrecio);

            // Parámetro de salida
            OracleParameter banderaParam = new OracleParameter("p_bandera", OracleDbType.Int32, 32);
            banderaParam.Direction = ParameterDirection.Output;
            comando.Parameters.Add(banderaParam);

            // Ejecutar el procedimiento
            conexion.Open();
            comando.ExecuteNonQuery();

            // Retornar valor de parámetro de salida
            if (banderaParam.Value != DBNull.Value)
            {
                bandera = Convert.ToInt32(banderaParam.Value.ToString());
            }

            // Caso de error según resultado de bandera
            if (bandera == 0)
            {
                // Mensaje de éxito
                MessageBox.Show("El producto se ha modificado correctamente.");
            }
        }
    }
}
```

productoSeleccionado	"Torta de Zanahoria"
bandera	0
nuevoPrecio	0
p_nuevoNombre	"Torta de Zanahoria"
p_nuevoProveedor	null
p_nuevaCategoria	"ACOMPANANTE"

El programa mostrará un mensaje indicando al usuario llenar todos los datos necesarios para poder realizar la modificación correctamente

