


	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); } } 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>	Se agrega la cantidad correctamente al producto seleccionado y se muestra el mensaje "La cantidad se ha actualizado correctamente."		+
	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>	El programa muestra el mensaje de que la cantidad a agregar debe ser mayor que 0		+

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();
        // 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);
    }
}
```

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));

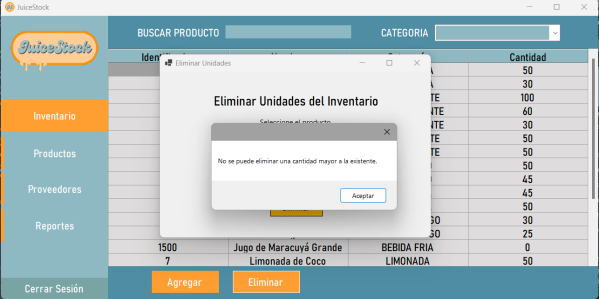
                // Abrir la conexión y ejecutar el comando
                conexion.Open();
                comando.ExecuteNonQuery();
            }
        }
        // Mensaje de éxito
        MessageBox.Show("La cantidad se ha actualizado correctamente.",
            "Éxito",
            MessageBoxButtons.OK,
  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

        // Cerrar el formulario de agregar inventario
        this.Close();
        // 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);
    }
}
```

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 || comboBoxCategoria.SelectedIndex == -1 || textBoxNombre.Text == "" || textBoxPrecio.Text == "")
        {
            MessageBox.Show("Faltan datos", "Error", MessageBoxButtons.OK, MessageBoxIcon.Warning);
            return; // Salir del método si no hay un producto seleccionado
        }

        // 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 nombreCategoriaSeleccionada = comboBoxCategoria.SelectedItem.ToString();

        // Validar que el precio sea mayor a 0
        if (precioProd <= 0)
        {
            MessageBox.Show("El precio debe ser mayor a 0", "Error", MessageBoxButtons.OK, MessageBoxIcon.Warning);
            return;
        }

        // Crear la conexión
        using (OracleConnection conexion = Conexion.getConnection())
        {
            // Preparar la consulta
            string sql = "INSERT INTO productos (nombre_producto, precio_producto, proveedor_producto, categoria_producto) VALUES (:nombre_producto, :precio_producto, :proveedor_producto, :categoria_producto)";
            using (OracleCommand comando = new OracleCommand(sql, conexion))
            {
                comando.CommandType = CommandType.StoredProcedure;

                // Agregar 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("proveedor_producto", nombreProveedorSeleccionado));
                comando.Parameters.Add(new OracleParameter("categoria_producto", nombreCategoriaSeleccionado));

                // Ejecutar la consulta
                comando.ExecuteNonQuery();

                // Mostrar mensaje de éxito
                MessageBox.Show("Producto agregado correctamente", "Éxito", MessageBoxButtons.OK, MessageBoxIcon.Information);
            }
        }

        if (comboBoxAgregar == 0)
        {
            // Mensaje de éxito
            MessageBox.Show("Producto se ha agregado correctamente", "Éxito", MessageBoxButtons.OK, MessageBoxIcon.Information);
        }
    }
    catch (Exception ex)
    {
        // Mensaje de error
        MessageBox.Show("Error al agregar producto: " + ex.Message, "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}

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 || comboBoxCategoria.SelectedIndex == -1 || textBoxNombre.Text == "" || textBoxPrecio.Text == "")
        {
            MessageBox.Show("Faltan datos", "Error", MessageBoxButtons.OK, MessageBoxIcon.Warning);
            return; // Salir del método si no hay un producto seleccionado
        }

        // 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 nombreCategoriaSeleccionada = comboBoxCategoria.SelectedItem.ToString();

        // Validar que el precio sea mayor a 0
        if (precioProd <= 0)
        {
            MessageBox.Show("El precio debe ser mayor a 0", "Error", MessageBoxButtons.OK, MessageBoxIcon.Warning);
            return;
        }

        // Crear la conexión
        using (OracleConnection conexion = Conexion.getConnection())
        {
            // Preparar la consulta
            string sql = "DELETE FROM productos WHERE nombre_producto = :nombre_producto AND precio_producto = :precio_producto AND proveedor_producto = :proveedor_producto AND categoria_producto = :categoria_producto";
            using (OracleCommand comando = new OracleCommand(sql, conexion))
            {
                comando.CommandType = CommandType.StoredProcedure;

                // Agregar 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("proveedor_producto", nombreProveedorSeleccionado));
                comando.Parameters.Add(new OracleParameter("categoria_producto", nombreCategoriaSeleccionado));

                // Ejecutar la consulta
                comando.ExecuteNonQuery();

                // Mostrar mensaje de éxito
                MessageBox.Show("Producto eliminado correctamente", "Éxito", MessageBoxButtons.OK, MessageBoxIcon.Information);
            }
        }

        if (comboBoxEliminar == 0)
        {
            // Mensaje de éxito
            MessageBox.Show("Producto se ha eliminado correctamente", "Éxito", MessageBoxButtons.OK, MessageBoxIcon.Information);
        }
    }
    catch (Exception ex)
    {
        // Mensaje de error
        MessageBox.Show("Error al eliminar producto: " + ex.Message, "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}
```

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 || comboBoxCategoria.SelectedIndex == -1 || textBoxNombre.Text == "" || textBoxPrecio.Text == "")
        {
            MessageBox.Show("Faltan datos", "Error", MessageBoxButtons.OK, MessageBoxIcon.Warning);
            return; // Salir del método si no hay un producto seleccionado
        }

        // 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 nombreCategoriaSeleccionada = comboBoxCategoria.SelectedItem.ToString();

        // Validar que el precio sea mayor a 0
        if (precioProd <= 0)
        {
            MessageBox.Show("El precio debe ser mayor a 0", "Error", MessageBoxButtons.OK, MessageBoxIcon.Warning);
            return;
        }

        // Crear la conexión
        using (OracleConnection conexion = Conexion.getConnection())
        {
            // Preparar la consulta
            string sql = "INSERT INTO productos (nombre_producto, precio_producto, proveedor_producto, categoria_producto) VALUES (:nombre_producto, :precio_producto, :proveedor_producto, :categoria_producto)";
            using (OracleCommand comando = new OracleCommand(sql, conexion))
            {
                comando.CommandType = CommandType.StoredProcedure;

                // Agregar 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("proveedor_producto", nombreProveedorSeleccionado));
                comando.Parameters.Add(new OracleParameter("categoria_producto", nombreCategoriaSeleccionado));

                // Ejecutar la consulta
                comando.ExecuteNonQuery();

                // Mostrar mensaje de éxito
                MessageBox.Show("Producto agregado correctamente", "Éxito", MessageBoxButtons.OK, MessageBoxIcon.Information);
            }
        }

        if (comboBoxAgregar == 0)
        {
            // Mensaje de éxito
            MessageBox.Show("Producto se ha agregado correctamente", "Éxito", MessageBoxButtons.OK, MessageBoxIcon.Information);
        }
    }
    catch (Exception ex)
    {
        // Mensaje de error
        MessageBox.Show("Error al agregar producto: " + ex.Message, "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}

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 || comboBoxCategoria.SelectedIndex == -1 || textBoxNombre.Text == "" || textBoxPrecio.Text == "")
        {
            MessageBox.Show("Faltan datos", "Error", MessageBoxButtons.OK, MessageBoxIcon.Warning);
            return; // Salir del método si no hay un producto seleccionado
        }

        // 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 nombreCategoriaSeleccionada = comboBoxCategoria.SelectedItem.ToString();

        // Validar que el precio sea mayor a 0
        if (precioProd <= 0)
        {
            MessageBox.Show("El precio debe ser mayor a 0", "Error", MessageBoxButtons.OK, MessageBoxIcon.Warning);
            return;
        }

        // Crear la conexión
        using (OracleConnection conexion = Conexion.getConnection())
        {
            // Preparar la consulta
            string sql = "DELETE FROM productos WHERE nombre_producto = :nombre_producto AND precio_producto = :precio_producto AND proveedor_producto = :proveedor_producto AND categoria_producto = :categoria_producto";
            using (OracleCommand comando = new OracleCommand(sql, conexion))
            {
                comando.CommandType = CommandType.StoredProcedure;

                // Agregar 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("proveedor_producto", nombreProveedorSeleccionado));
                comando.Parameters.Add(new OracleParameter("categoria_producto", nombreCategoriaSeleccionado));

                // Ejecutar la consulta
                comando.ExecuteNonQuery();

                // Mostrar mensaje de éxito
                MessageBox.Show("Producto eliminado correctamente", "Éxito", MessageBoxButtons.OK, MessageBoxIcon.Information);
            }
        }

        if (comboBoxEliminar == 0)
        {
            // Mensaje de éxito
            MessageBox.Show("Producto se ha eliminado correctamente", "Éxito", MessageBoxButtons.OK, MessageBoxIcon.Information);
        }
    }
    catch (Exception ex)
    {
        // Mensaje de error
        MessageBox.Show("Error al eliminar producto: " + ex.Message, "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}
```

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

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

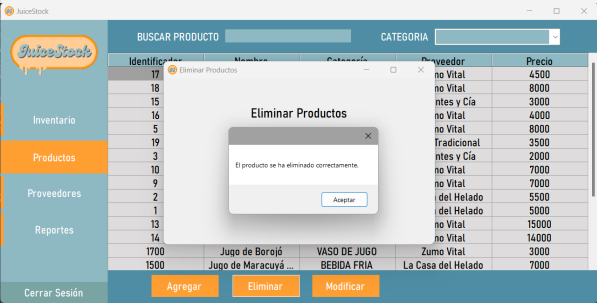
+

SPRINT 2

HE2-HU2-CA1

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 (comboBoxProducto.SelectedIndex == -1)
        {
            MessageBox.Show("Por favor, seleccione un producto.");
            return; // Si no se ha seleccionado un producto, no hay que eliminar nada
        }
        string nombreProductoSeleccionado = comboBoxProducto.SelectedItem.ToString();
        int cantidadEliminar = 0;

        // Muestra 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;
        }
        // Crear la conexión
        using (OracleConnection conexion = Conexion.getInstancia())
        {
            // Crear el comando para borrar el producto(s) seleccionado(s)
            using (OracleCommand comando = new OracleCommand("delete from producto where nombre_producto = '" + nombreProductoSeleccionado + "'", conexion))
            {
                comando.CommandType = CommandType.StoredProcedure;

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

                // Ejecutar el comando de borrar
                comando.ExecuteNonQuery();

                // Mostrar mensaje de éxito
                MessageBox.Show("Producto eliminado correctamente.");
            }
        }

        // Actualizar la cantidad y eliminar el comando
        comando.CommandText = "UPDATE producto SET cantidad = cantidad - " + cantidadEliminar + " WHERE nombre_producto = '" + nombreProductoSeleccionado + "'";
        comando.ExecuteNonQuery();

        // Mostrar el valor del parámetro de salida
        if (comando.Parameters["n_cantidad_eliminada"].Value != DBNull.Value)
        {
            cantidadEliminar = (int)comando.Parameters["n_cantidad_eliminada"].Value;
        }

        if (cantidadEliminar == 0)
        {
            // Mensaje de éxito
            MessageBox.Show("El producto se ha eliminado correctamente.");
        }
        else if (cantidadEliminar == 1)
        {
            // Mensaje de error: 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

```
private void txtBuscar_TextChanged(object sender, EventArgs e)
{
    // Cuando el texto ingresado en el TextBox
    string textoBusqueda = txtBuscar.Text;
    string consultaFiltrada;

    if (string.IsNullOrWhiteSpace(textoBusqueda))
    {
        // Si el TextBox está vacío, mostrar todos los productos
        consultaFiltrada = "SELECT ID_PROD 'Identificador', NOMBRE_PROD 'Nombre', NOMBRE_CATEGORIA 'Categoría', CANTIDAD_PROD 'Cantidad' " +
            "FROM VISTA_PRODUCTOS";
    }
    else
    {
        // Si hay texto, buscar los productos que contengan la cadena ingresada
        consultaFiltrada = "SELECT ID_PROD 'Identificador', NOMBRE_PROD 'Nombre', NOMBRE_CATEGORIA 'Categoría', CANTIDAD_PROD 'Cantidad' " +
            "FROM VISTA_PRODUCTOS " +
            "WHERE upper(NOMBRE_PROD) LIKE '%" + textoBusqueda.ToUpper() + "%'"; // Permite los que coinciden con el texto
    }

    // Llamar al método que ejecutará el procedimiento con los productos filtrados
    g.Productos Datos = new g.Productos();
    g.ListadoInventario DetalleData = Datos.Listado_producto(consultaFiltrada);
}
```

textoBusqueda = "granizado"

consultaFiltrada = "SELECT ID_PROD 'Identificador', NOMBRE_PROD 'Nombre', NOMBRE_CATEGORIA 'Categoría', 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"

El programa mostrará en una grilla la información perteneciente al producto existente que esté nombrado de manera similar a la cadena de texto a buscar

SPRINT 3

HE5-HU3-CA2

```
private void cmbCategoria_SelectedIndexChanged(object sender, EventArgs e)
{
    // Cuando el texto ingresado en el TextBox
    string textoCategoria = cmbCategoria.Text;
    string consultaFiltrada;

    if (string.IsNullOrWhiteSpace(textoCategoria))
    {
        // Si el TextBox está vacío, mostrar todos los productos
        consultaFiltrada = "SELECT ID_PROD 'Identificador', NOMBRE_PROD 'Nombre', NOMBRE_CATEGORIA 'Categoría', CANTIDAD_PROD 'Cantidad' " +
            "FROM VISTA_PROD " +
            "WHERE upper(NOMBRE_PROD) LIKE '%" + textoCategoria.ToUpper() + "%'"; // Permite los que coinciden con el texto
    }
    else
    {
        // Si hay texto, buscar los productos que contengan la cadena ingresada
        consultaFiltrada = "SELECT ID_PROD 'Identificador', NOMBRE_PROD 'Nombre', NOMBRE_CATEGORIA 'Categoría', CANTIDAD_PROD 'Cantidad' " +
            "FROM VISTA_PROD " +
            "WHERE upper(NOMBRE_PROD) LIKE '%" + textoCategoria.ToUpper() + "%'"; // Permite los que coinciden con el texto
    }

    // Llamar al método que ejecutará el procedimiento con los productos filtrados
    g.Productos Datos = new g.Productos();
    g.ListadoInventario DetalleData = Datos.Listado_producto(consultaFiltrada);
}
```

textoCategoria = "HELADERIA"

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

El programa mostrará en una grilla los productos filtrados por la categoría seleccionada

HE3-HU1-CA1

```
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)
                {
                    OracleManagedDataAccess.Types.OracleDecimal oracleDecimal =
                        (OracleManagedDataAccess.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	"1234"

El programa muestra el apartado de inicio de la aplicación

The screenshot shows the JuiceStock application window. At the top, there's a search bar with 'granizado' entered. Below it, a table displays the search results:

Identificador	Nombre	Categoría	Cantidad
10	Granizado de Fresa	GRANIZADO	50

On the left sidebar, 'Inventario' is selected. At the bottom, there are 'Agregar' and 'Eliminar' buttons.

The screenshot shows the JuiceStock application window. At the top, the category dropdown is set to 'HELADERIA'. Below it, a table displays the search results:

Identificador	Nombre	Categoría	Cantidad
2	Helado de Chocolate	HELADERIA	45
1	Helado de Vainilla	HELADERIA	50

On the left sidebar, 'Inventario' is selected. At the bottom, there are 'Agregar' and 'Eliminar' buttons.

+

+

+

SPRINT 4	HE3-HU1-CA2	<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; } 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									
	HE3-HU1-CA3	<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));

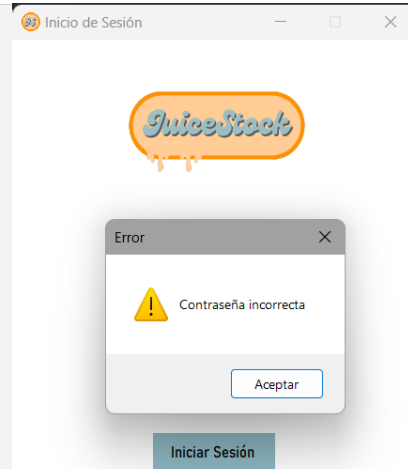
                // Parámetro de salida
                OracleParameter resultadoParam = new OracleParameter("p_resultado", OracleDbType.Int32);
                resultadoParam.Direction = ParameterDirection.Output;
                comando.Parameters.Add(resultadoParam);

                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 btnAgregarProveedores_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 fila de la tabla
        int indiceAgregar = 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));

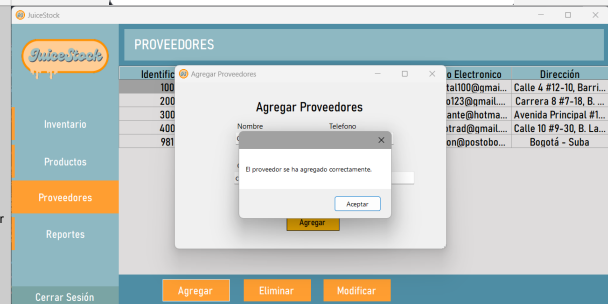
                // Añadir parámetros de salida
                OracleParameter resultadoParametro = new OracleParameter("p_resultado", OracleDbType.Int32);
                resultadoParametro.Direction = ParameterDirection.Output;
                comando.Parameters.Add(resultadoParametro);

                // 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;
                    indiceAgregar = oracleDecimal.ToInt32();
                }
            }
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show($"Error al agregar proveedor: {ex.Message}", "Error",
            MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
    else if (indiceAgregar == 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 data-bbox="315 437 398 451">HE4-HU2-CA1</div> <pre data-bbox="443 145 922 616">private void btnEliminarProveedor_Click(object sender, EventArgs e) { try { if (comboBoxProveedores.SelectedIndex == -1) { MessageBox.Show("Por favor, seleccione un proveedor."); return; } string nombreProveedorSeleccionado = comboBoxProveedores.SelectedItem.ToString(); int proveedor = 0; // 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 "No", se cancela la eliminación return; } // Crear la conexión using (OracleConnection conexion = Conexion.GetInstance().CrearConexion()) { // Crear el comando para eliminar el proveedor 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)); // Añadir los parámetros de salida OracleParameter resultadoEliminarProveedor = new OracleParameter("p_resultado", OracleDbType.Int32); comando.Parameters.Add(resultadoEliminarProveedor); comando.Parameters.Add(resultadoEliminarProveedor); // Ejecutar la consulta y obtener el resultado comando.ExecuteNonQuery(); // Recuperar el valor del parámetro de salida if (comando.Parameters["p_resultado"].Value != null) { OracleMessageDataAddress Types OracleMessageDataAddress = (OracleMessageDataAddress Types OracleMessageDataAddress)resultadoEliminarProveedor.Value; resultado = OracleMessageDataAddress.Types["p_resultado"]; } } } // Mostrar el resultado if (resultado == 0) { MessageBox.Show("Proveedor no ha eliminado correctamente.", "Error", MessageBoxButtons.OK, MessageBoxIcon.Information); } else if (resultado == 1) { MessageBox.Show("Proveedor tiene productos asociados. No se puede eliminar."); } else { MessageBox.Show("Se eliminó correctamente el proveedor."); } } catch (Exception ex) { MessageBox.Show("Error al eliminar proveedor: " + ex.Message); } }</pre>
----------	--

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, se ha seleccionado un proveedor o una categoría
    if (cmbProveedor.SelectedIndex == -1 || cmbCategoria.SelectedIndex == -1 || lblNombre.Text == "" || lblPrecio.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(lblPrecio.Text);
    string p_nuevoNombre = lblNombre.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_proce", 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, se ha seleccionado un proveedor o una categoría
    if (cmbProveedor.SelectedIndex == -1 || cmbCategoria.SelectedIndex == -1 || lblNombre.Text == "" || lblPrecio.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(lblPrecio.Text);
    string p_nuevoNombre = lblNombre.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_proce", 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

