



Instituto Profesional AIEP

Proyecto Botillería con metodología cascada

NOMBRE ESTUDIANTE(S):

Sebastián Garay

Roberto Ibarra

Rodrigo Larraín

CARRERA:Técnico en programación y análisis de sistema

SEDE:San Fernando

MÓDULO:PRO204-1000-2022-METODOLOGÍAS DE DESARROLLO DE SOFTWARE

SECCIÓN:1000

DOCENTE:Sebastián Santana Valenzuela

San Fernando, Noviembre, 2022

Indice

1.	Análisis de requerimientos	2
a.	Problemática:	3
b.	Objetivos:	3
c.	Requerimientos:	3
	Funcionales:	3
	No funcionales:	3
2.	Fase de análisis:	3
a.	Casos de uso:	4
b.	Diagrama de Actividad:	4
c.	Diagrama de Secuencias:	6
d.	Casos de prueba:	9
3.	Fase de Diseño:	13
e.	Diagrama de Clases:	13
a.	Prototipo de interfaces	14
b.	Diagrama de modelo entidad-relación:	18
c.	Diagrama de paquetes:	18
d.	Diagrama de componentes:	19
e.	Diagrama de despliegue:	20
		20
4.	Implementación:	21
5.	Pruebas:	26

1. Análisis de requerimientos

a. Problemática:

La problemática que nos comentó el socio fue crear un sistema de botillería creando así un registro de mermas en el sistema, como también un registro de productos, y que se pueda acceder a estas listas de mermas y productos como también que los productos sean actualizables de precio y stock.

b. Objetivos:

Crear una lista de mermas visualizable.

Crear una lista de productos actualizable.

Mantener un registro de mermas.

Mantener un registro de inventario.

c. Requerimientos:

Funcionales:

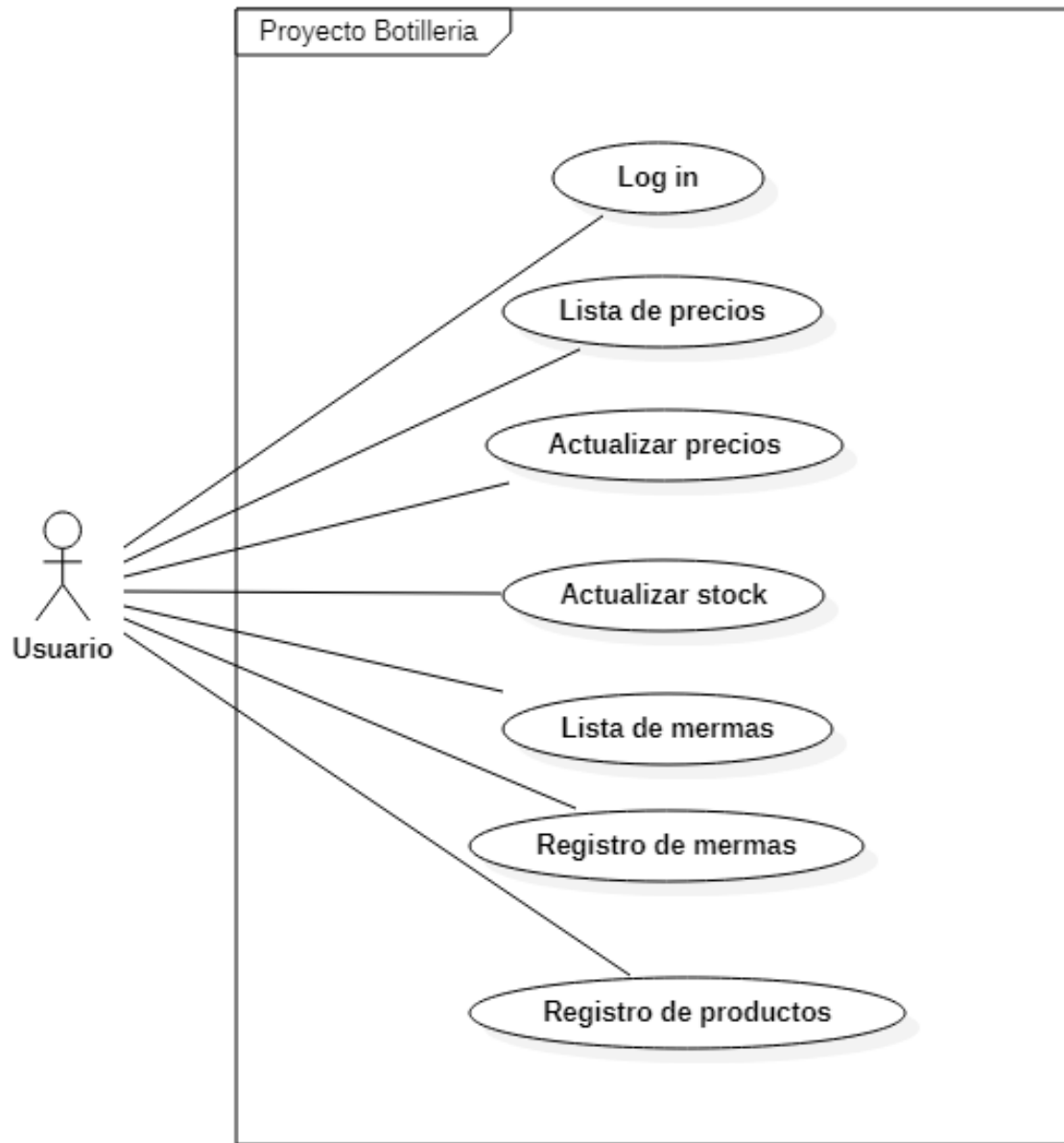
- I. Sistema deberá registrar mermas
- II. Sistema deberá mostrar lista de mermas
- III. Sistema deberá registrar productos
- IV. Sistema deberá mostrar lista de productos
- V. Sistema deberá actualizar/editar precios de los productos
- VI. Sistema deberá editar el stock de los productos

No funcionales:

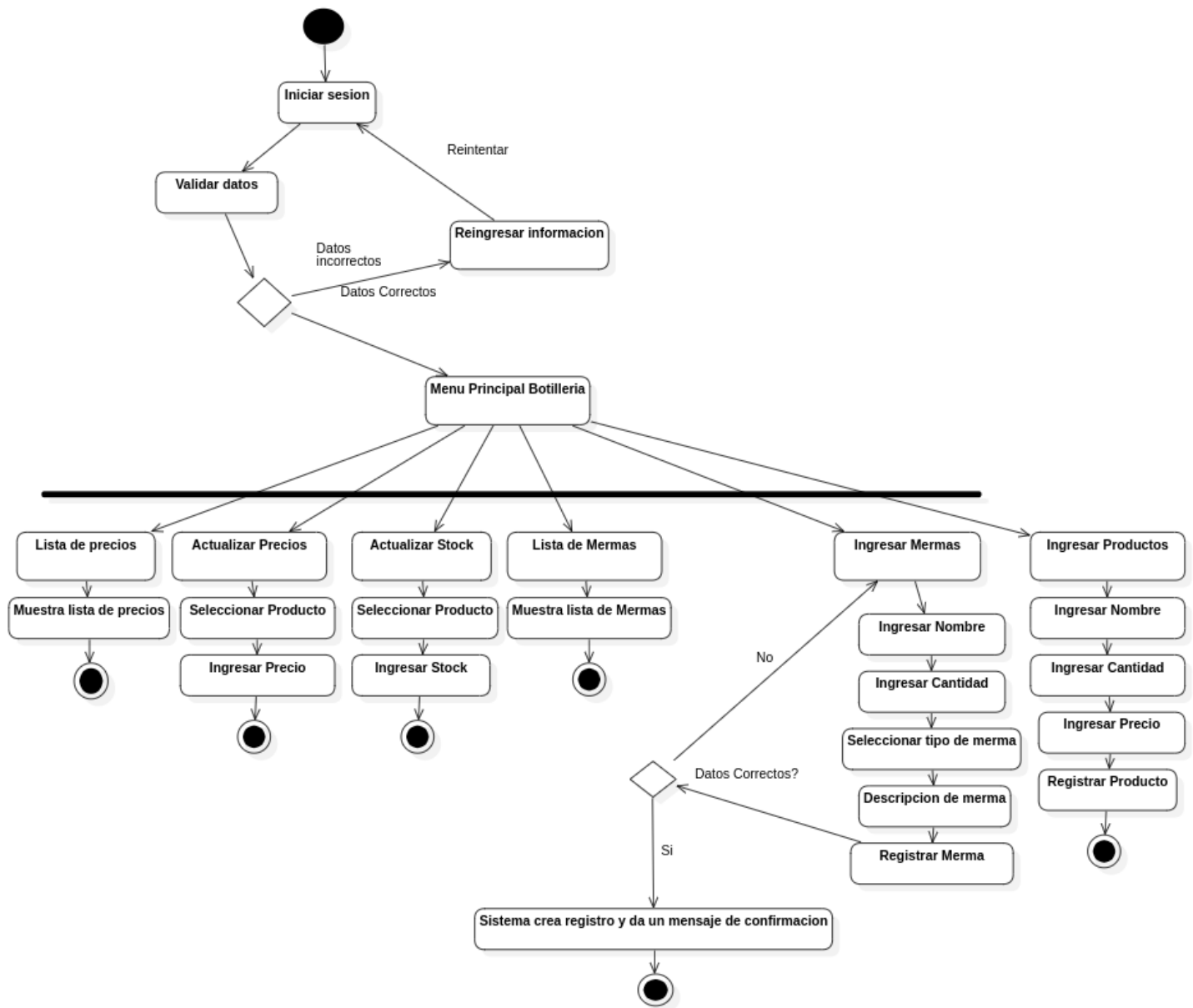
- I. Sistema deberá ser intuitivo para el usuario
- II. Sistema deberá tener acceso solo a personal autorizado

2. Fase de análisis:

a. Casos de uso:

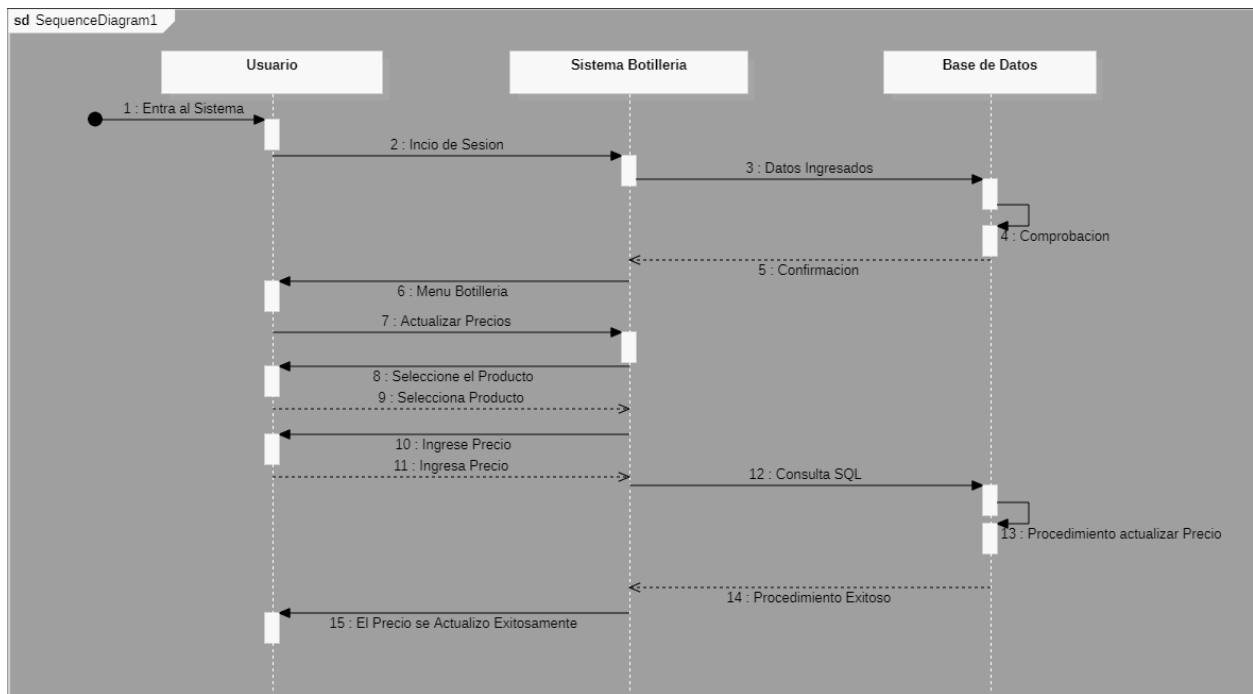


b. Diagrama de Actividad:

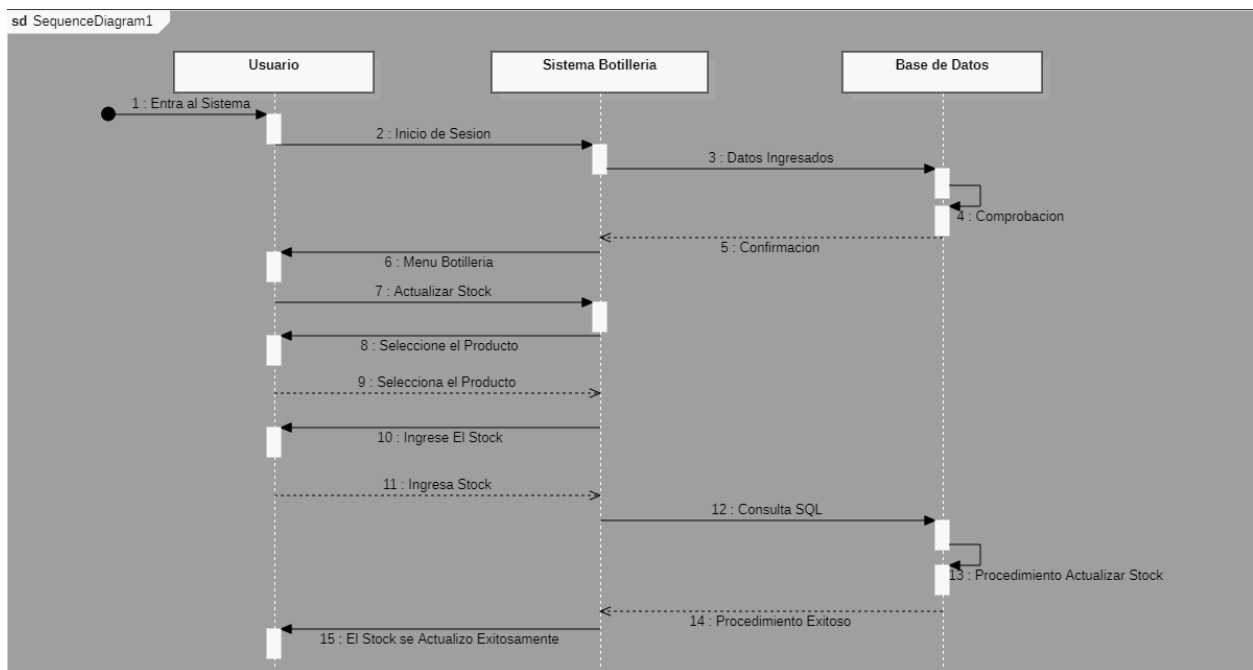


c. Diagrama de Secuencias:

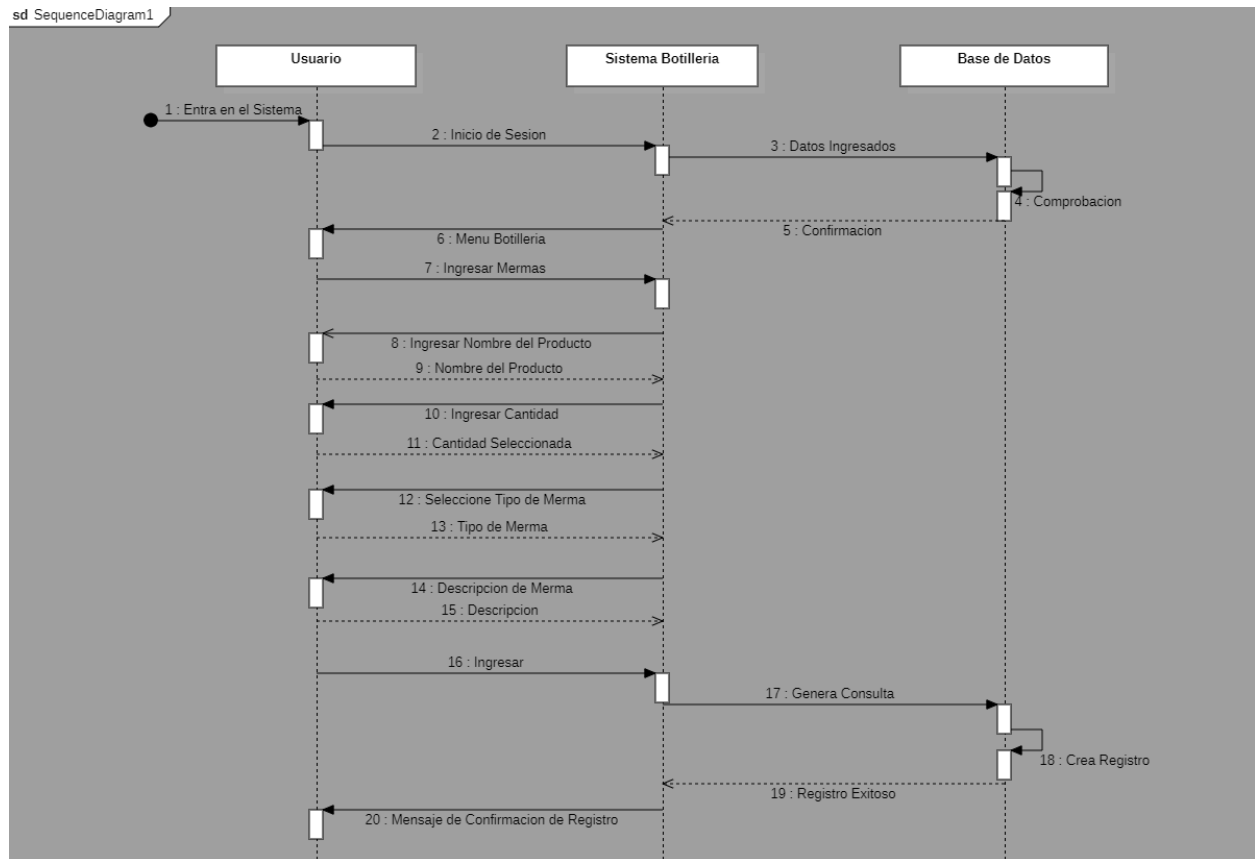
1. Actualizar Precios:



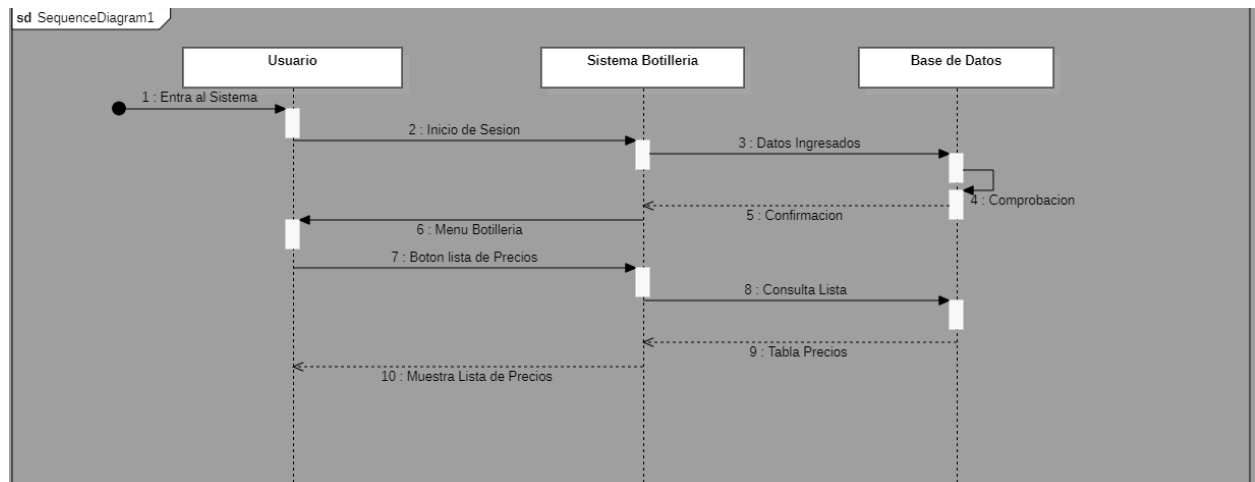
2. Actualizar Stock:



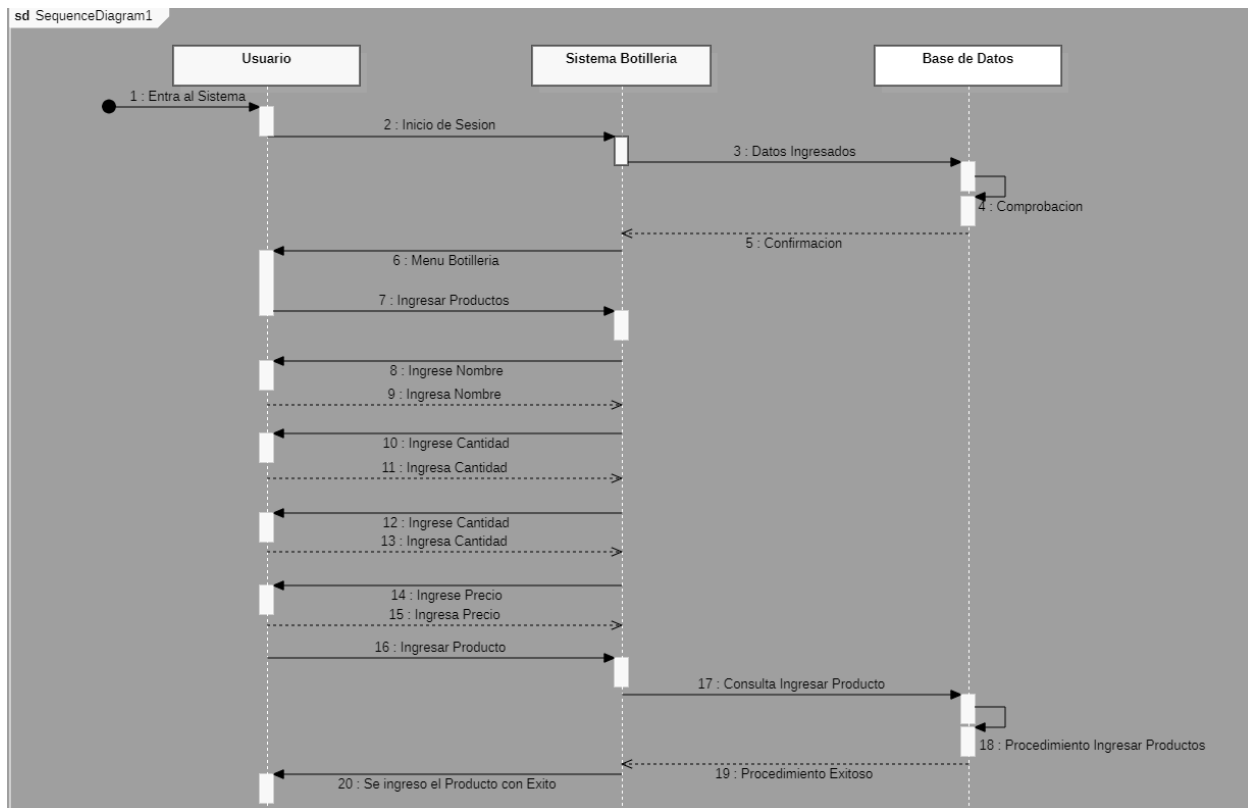
3. Ingresar Mermas (Implementado):



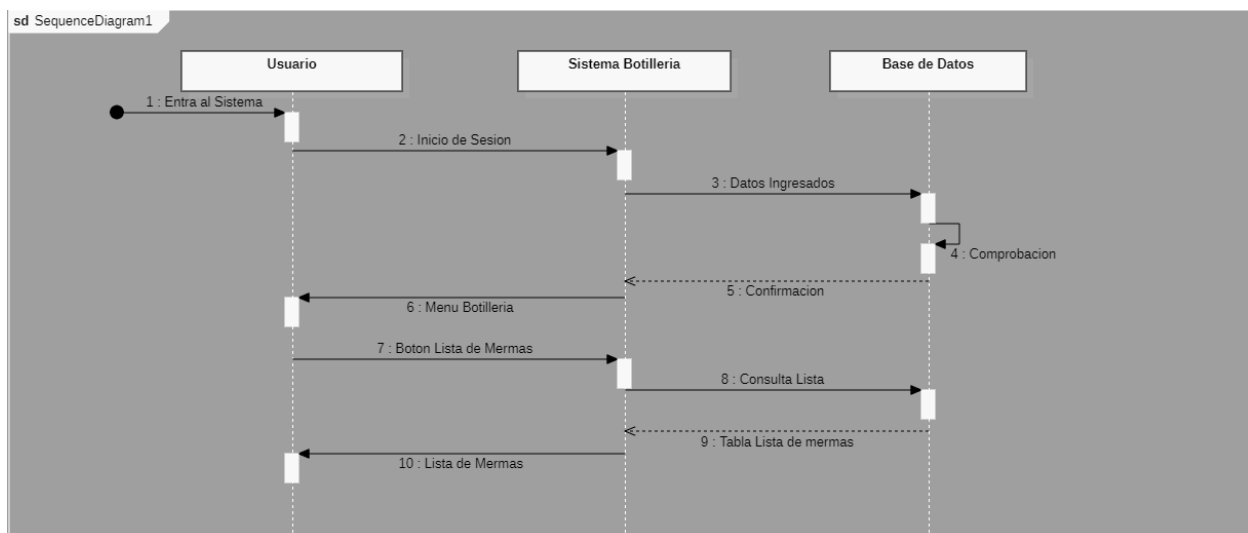
4. Lista de Precios:



5. Ingresar Productos:



6. Lista de Mermas:



d. Casos de prueba:

Transformación de caso de uso a un caso de prueba:

1. Escribir el detalle del caso de uso:

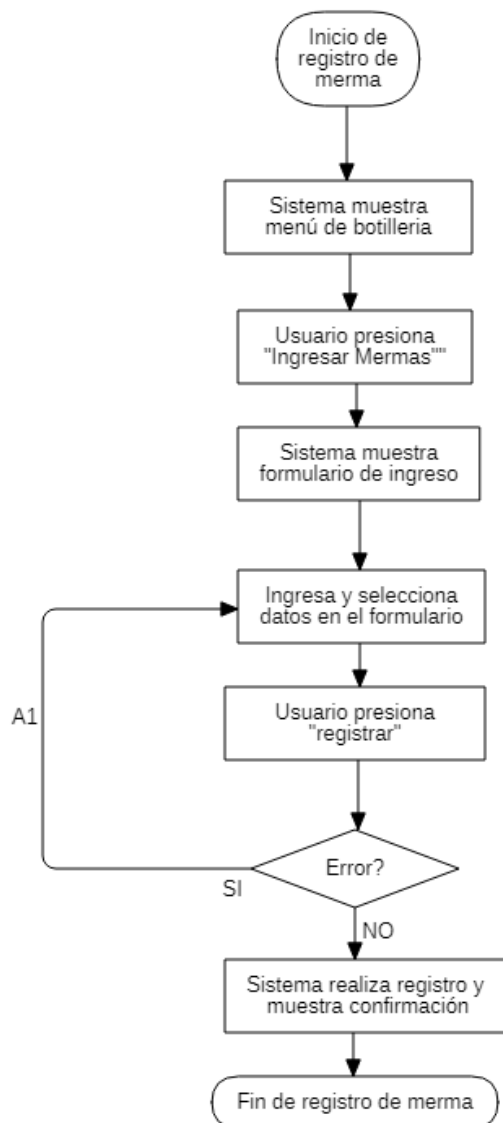
Caso de uso ID	Registro_001		
Caso de uso nombre	Registro de mermas		
Creado por		Ultima actualización por:	Roberto Ibarra
Fecha Creación	9/11/2022	Fecha de Ultima actualización:	9/11/2022
Actores	Usuario		
Descripción	Registro de una merma con la aplicación de escritorio		
Trigger	Usuario encuentra una merma y desea registrarla		
Condiciones previas	Usuario está registrado en la aplicación cuenta con el producto defectuoso y sus detalles.		
Condiciones posteriores	Se toma la información del producto, y se realiza el proceso de registro.		
Flujo Básico	<p>Registro:</p> <p>Sistema muestra menú de botillería</p> <p>Usuario presiona botón “ingresar mermas”</p> <p>Usuario ingresa nombre del producto al formulario</p> <p>Usuario selecciona la cantidad de producto al formulario</p> <p>Usuario selecciona el tipo de merma en el formulario</p> <p>Usuario ingresa descripción de merma al formulario</p> <p>Usuario presiona el botón “registrar”</p> <p>Sistema realiza el proceso de registrar y muestra al usuario un mensaje de “merma registrada exitosamente”</p> <p>Usuario cierra ventana de formulario</p>		
Flujo Alternativo	A1. Usuario presiona el botón “registrar”, pero tiene un error al no completar algún campo, usuario llena correctamente y se repite el (paso 7)		
Actores	Usuario		
Incluye	Ninguna		
Prioridad	Baja		
Frecuencia de uso	Cuando se necesite		

Reglas de negocio	Ninguna
Requisitos especiales	ninguno
Suposiciones	Exista un producto en mal estado
Notas y ediciones	Ninguna

2. Identificar los escenarios del caso de uso:

Escenario 1	Flujo Básico		
Escenario 2	Flujo Básico	Flujo Alternativo A1	

3. Diagrama de flujo:



4. Generar los casos de prueba:

Caso de prueba id	Escenario/Condición	Nombre del producto	Cantidad de producto	Tipo de merma	Descripción de merma	Registro exitoso	Resultado esperado
Registro_001	Escenario 1- Registro exitoso	V	V	V	V	V	El registro es exitoso
Registro_002	Escenario 2- Usuario no completa todos los campos del formulario	N/A	N/A	N/A	N/A	F	Usuario presiona botón "registrar" y si algún campo está vacío, muestra un mensaje de error.

5. Generar datos de prueba

Caso de prueba id	Escenario/Condición	Nombre del producto	Cantidad de producto	Tipo de merma	Descripción de merma	Registro exitoso	Resultado esperado
Registro_001	Escenario 1- Registro exitoso	"Yogurt Colun"	2	"Devolución"	"Cliente lo devuelve porque está vencido"	V	El registro es exitoso
Registro_002	Escenario 2- Usuario no completa todos los campos del formulario	"Manque"	N/A	"operacional"	"productos aplastados"	F	Usuario presiona botón "registrar" y no selecciona la cantidad, se muestra un mensaje indicando el campo faltante.

6. Documentar el caso de prueba:

Fecha: 09/11/2022 **Prueba por:** Roberto Ibarra

Sistema: Botillería **Entorno:** Aplicación de escritorio

Objetivo: Registrar una merma en el sistema **ID Prueba:** Registro_1 **Req ID:** Requerimiento_1

Función: Registrar una merma exitosamente en el sistema **Pantalla:** interfaz registro de merma

Versión: 1.0 **Tipo De Prueba:** Sistema (Caja Negra)

Condiciones de Prueba:

Usuario está registrado en la aplicación cuenta con el producto defectuoso y sus detalles.

Pasos a Realizar:

Presionar el botón “Registrar merma”, sistema muestra un formulario de registro, usuario llena el formulario de merma con nombre de producto, cantidad de producto, tipo de merma y descripción de merma para luego presionar el botón “registrar”, el sistema hace proceso de registrar la merma y usuario cierra la ventana.

Resultados Esperados:

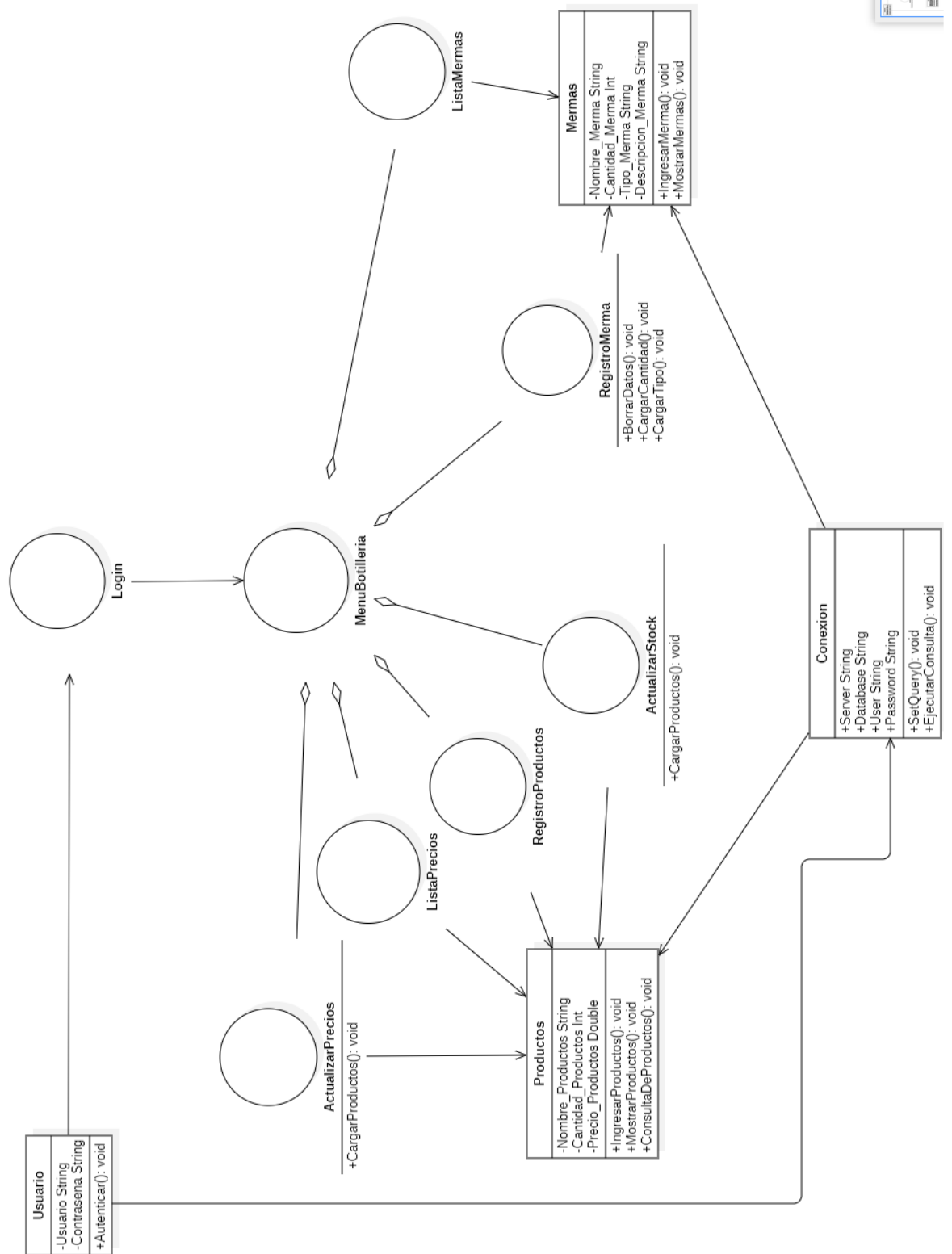
Registrar una merma en el sistema.

Resultados Actuales:

Exitoso, Se registra una merma en el sistema botillería.

3. Fase de Diseño:

e. Diagrama de Clases:



a. Prototipo de interfaces

I. Registro de merma (requerimiento implementado)

INGRESO DE MERMAS

Nombre Producto

Cantidad

Tipo de merma

Descripción

Registrar

II. Login:



Login

Usuario

Contraseña

Ingresar

III. Menú:



Menú Botilleria

Lista Precios
Actualizar Precios
Actualizar Stock
Lista Mermas
Ingresar Mermas
Ingresar Productos

IV. Lista de precios:

Lista de precios



V. Lista Mermas:

Lista de mermas



VI. Actualizar Precios:

Actualizar Precio

Seleccione un producto

Ingrese precio

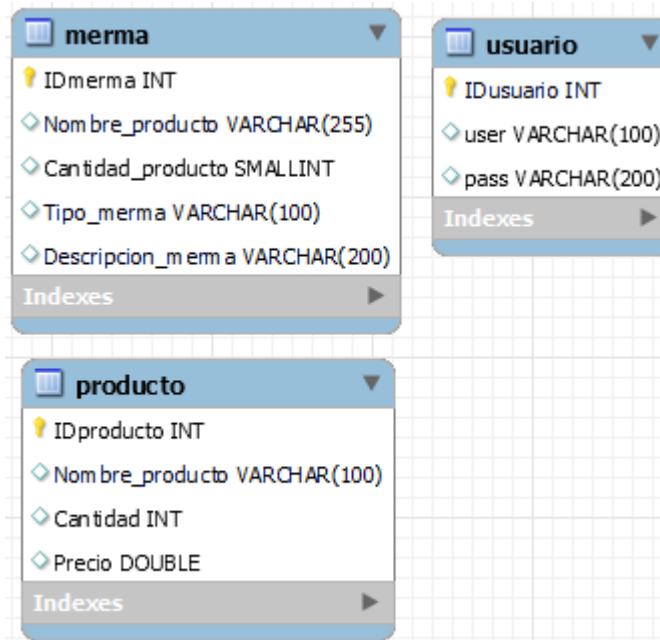
VII. Actualizar Stock:

Actualizar Stock

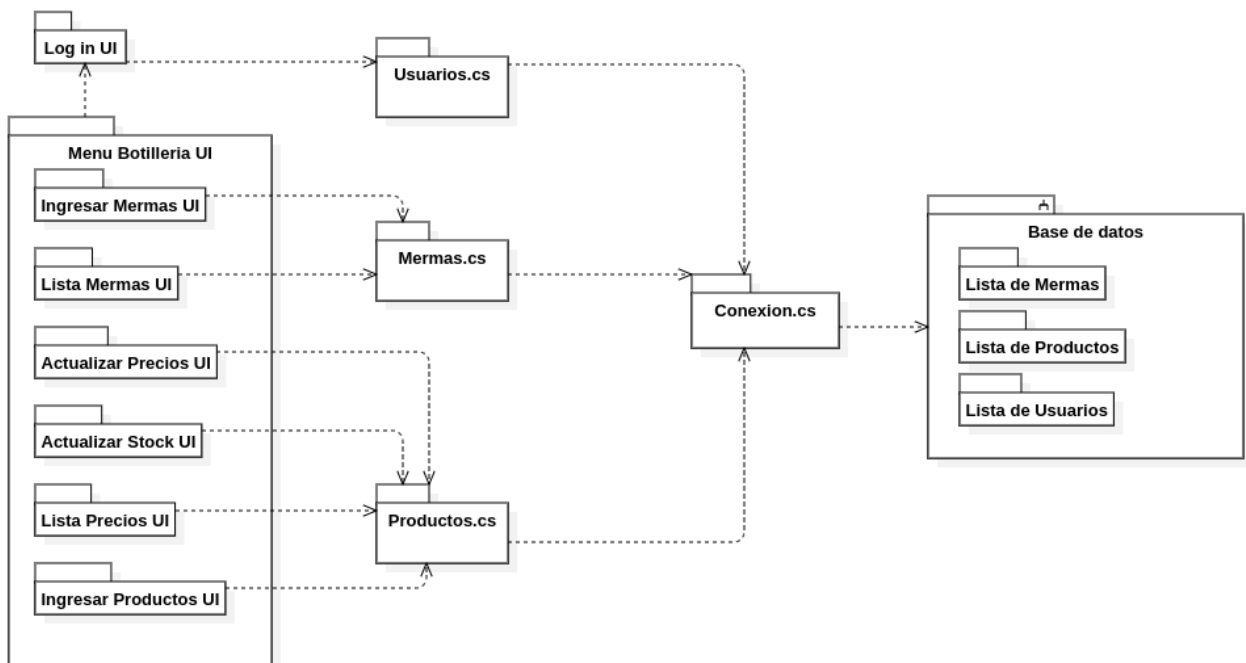
Seleccione un producto

Ingrese Stock

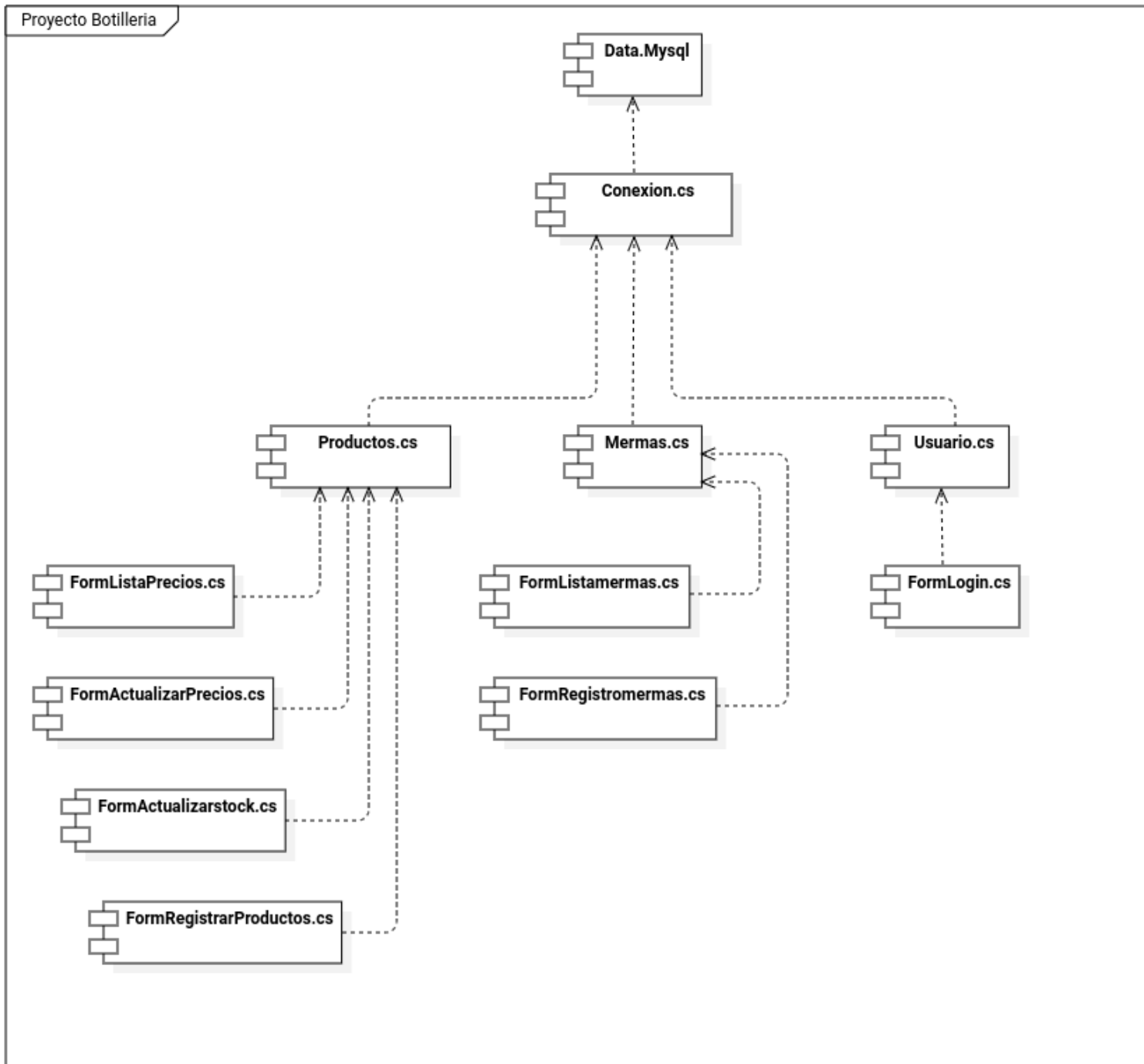
b. Diagrama de modelo entidad-relación:



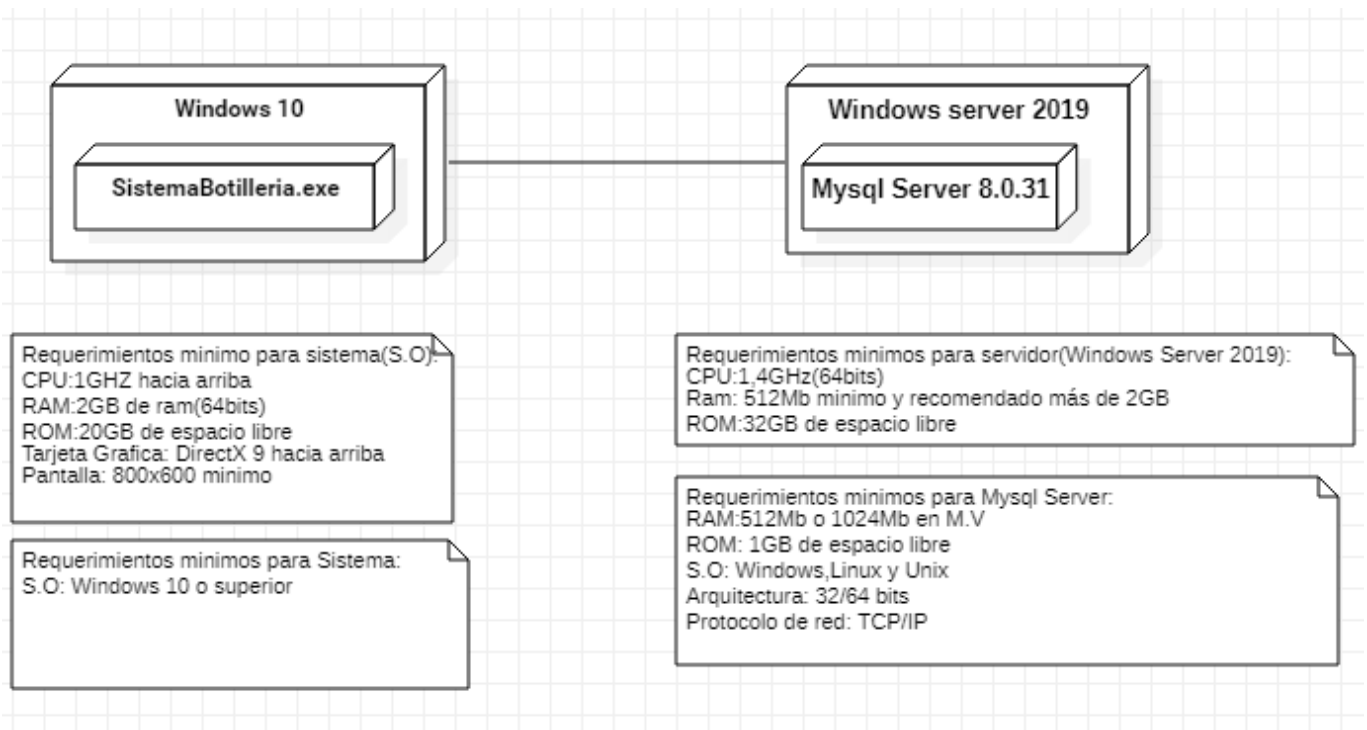
c. Diagrama de paquetes:



d. Diagrama de componentes:



e. Diagrama de despliegue:



4. Implementación:

I. Código Registro Mermas (Form)

```
12 namespace Proyecto_Botilleria
13 {
14     3 referencias
15     public partial class RegistroMerma : Form
16     {
17         1 referencia
18         public RegistroMerma()
19         {
20             //inicia el form y los metodos dentro de este
21             InitializeComponent();
22             cargarCantidad();
23             cargarTipo();
24         }
25
26         1 referencia
27         private void lbl_Click(object sender, EventArgs e)
28         {
29         }
30
31         1 referencia
32         private void btn_registrar_Click(object sender, EventArgs e)
33         {
34             // toma los datos tanto de los textbox, como los combobox seleccionados
35             string nombre_merma = txtb_nombreP.Text;
36             int cantidad_merma = cmb_cantidad.SelectedIndex;
37             string tipo_merma = cmb_TipoMerma.Text;
38             string descripcion_merma = txtb_descripcion.Text;
39
40             // instancia merma
41             Mermas merma = new Mermas(nombre_merma, cantidad_merma, tipo_merma, descripcion_merma);
42
43             //verifica cada campo buscando si esta vacio o no, si esta vacio muestra un error.
44             if (string.IsNullOrEmpty(cmb_TipoMerma.Text))
45             {
46                 MessageBox.Show("Error: Falta seleccionar el tipo de merma, debe seleccionar un tipo");
47             }
48             else if (string.IsNullOrEmpty(cmb_cantidad.Text))
49             {
50                 MessageBox.Show("Error: Falta seleccionar la cantidad de productos, debe seleccionar una cantidad");
51             }
52             else if (string.IsNullOrEmpty(txtb_nombreP.Text)) {
53                 MessageBox.Show("Error: Falta ingresar nombre de producto, debe ingresar un nombre");
54             }
55             else if (string.IsNullOrEmpty(txtb_descripcion.Text))
56             {
57                 MessageBox.Show("Error: Falta ingresar una descripción de la merma, debe ingresar una descripción");
58             }
59             else
60             {
61                 // llama el metodo utilizando los datos antes puestos
62                 merma.ingresarMerma();
63                 BorrarDatos();
64             }
65         }
66     }
67 }
```

```

47     MessageBox.Show("Error: Falta seleccionar la cantidad de productos, debe seleccionar una cantidad");
48 }
49 else if (string.IsNullOrEmpty(txtb_nombreP.Text)) {
50     MessageBox.Show("Error: Falta ingresar nombre de producto, debe ingresar un nombre");
51 }
52 else if (string.IsNullOrEmpty(txtb_descripcion.Text))
53 {
54     MessageBox.Show("Error: Falta ingresar una descripción de la merma, debe ingresar una descripción");
55 }
56 else
57 {
58     // llama el metodo utilizando los datos antes puestos
59     merma.ingresarMerma();
60     BorrarDatos();
61 }
62
63 }
64
65 // borra lo escrito y seleccionado en el formulario
66
67 1 referencia
68 public void BorrarDatos() {
69
70     txtb_nombreP.Text = "";
71     cmb_cantidad.SelectedItem = null;
72     cmb_TipoMerma.SelectedItem = null;
73     txtb_descripcion.Text = "";
74 }
75
76 //para rellenar el combobox de cantidad de productos
77 1 referencia
78 public void cargarCantidad() {
79     // un for para que se agreguen la cantidad de numero dentro del combobox, el maximo será 24 productos por registro de merma
80     for (int i = 1; i < 25; i++)
81     {
82         cmb_cantidad.Items.Add(i);
83     }
84 }
85
86 //para rellenar el combobox de tipo de merma
87 1 referencia
88 public void cargarTipo() {
89     cmb_TipoMerma.Items.Add("Operacional");
90     cmb_TipoMerma.Items.Add("Devolución");
91     cmb_TipoMerma.Items.Add("Caducado");
92 }

```

II. Clase Mermas

```
6  using static Proyecto_Botilleria.Conexion;
7  using System.Windows.Forms;
8
9  namespace Proyecto_Botilleria
10 {
11     3 referencias
12     class Mermas
13     {
14         private string nombre_merma;
15         private int cantidad_merma;
16         private string tipo_merma;
17         private string descripcion_merma;
18
19         1 referencia
20         public Mermas(string _nombreP, int _cantidadP, string _tipoM, string _descripcionM)
21         {
22             nombre_merma = _nombreP;
23             cantidad_merma = _cantidadP;
24             tipo_merma = _tipoM;
25             descripcion_merma = _descripcionM;
26         }
27
28         1 referencia
29         public void ingresarMerma()
30         {
31             Conexion connect = new Conexion();
32             // escribir la query
33             string query = string.Format("call sp_ingresarmerma('{0}','{1}','{2}','{3}');", nombre_merma, cantidad_merma, tipo_merma, descripcion_merma);
34             // tomar la query y ejecutarla con los metodos de conexion
35             connect.setQuery(query);
36             connect.EjecutarConsulta();
37             MessageBox.Show("Merma Registrada Exitosamente");
38
39         }
40     }
41
42 }
```

III. Clase Conexión

```
11 namespace Proyecto_Botilleria
12 {
13     5 referencias
14     class Conexion
15     {
16         public MySqlConnection con;
17         public string query;
18
19         static string server = "localhost";
20         static string database = "Botilleria";
21         static string user = "root";
22         static string password = "1234";
23
24         public string conString = string.Format("server={0};uid={1};pwd={2};database={3}",
25             server, user, password, database);
26
27         //Metodos
28         1 referencia
29         public void setQuery(string _query)
30         {
31             query = _query;
32         }
33
34         1 referencia
35         public void EjecutarConsulta()
36         {
37             try
38             {
39                 using (con = new MySqlConnection(conString))
40                 {
41                     con.Open();
42                     MySqlCommand cmd = new MySqlCommand(query, con);
43                     // que ejecute la query, pero no devuelve nada
44                     cmd.ExecuteNonQuery();
45                 }
46             }
47             // en caso de que falle sale un error
48             catch (MySqlException ex)
49             {
50                 MessageBox.Show("Error : {0}", ex.Message);
51             }
52         }
53     }
54 }
55
```








IV. Base de datos:

a. Crear base de datos y tablas:

```
1 • create database Botilleria;
2 • use Botilleria;
3
4 • create table usuario(
5     IDusuario int auto_increment primary key,
6     user varchar(100),
7     pass varchar(200)
8 );
9
10 • create table merma(
11     IDmerma int auto_increment primary key,
12     Nombre_producto varchar(255),
13     Cantidad_producto smallint,
14     Tipo_merma varchar(100),
15     Descripcion_merma varchar(200)
16 );
17
18 • create table producto(
19     IDproducto int auto_increment primary key,
20     Nombre_producto varchar(100),
21     Cantidad int,
22     Precio double
23 );
24
```

b. Insertar datos prueba:

```
31 -- Datos para probar BD
32 • insert into usuario(user,pass)
33     values("admin","admin");
34
35 • INSERT INTO producto(Nombre_producto,Cantidad,Precio)
36     VALUES("Yogurt",24,250),("Coca cola",24,800);
37
38 • insert into merma(Nombre_producto,Cantidad_producto,Tipo_merma,Descripcion_merma)
39     Values ("Yogurt Colun",1,"Devolución","Se rompio al momento de venta, cliente se le cayo al suelo");
40
41 • select * from usuario;
42 • select * from producto;
43 • select * from merma;
```

Result Grid					
Filter Rows: <input type="text"/>					
Edit:    Export/Import:   Wrap Cell Content: 					
	IDmerma	Nombre_producto	Cantidad_producto	Tipo_merma	Descripcion_merma
▶	1	Yogurt Colun	1	Devolución	Se rompio al momento de venta, cliente se le cayo al suelo
*	NULL	NULL	NULL	NULL	NULL

c. Procedimiento almacenado de registro de merma:

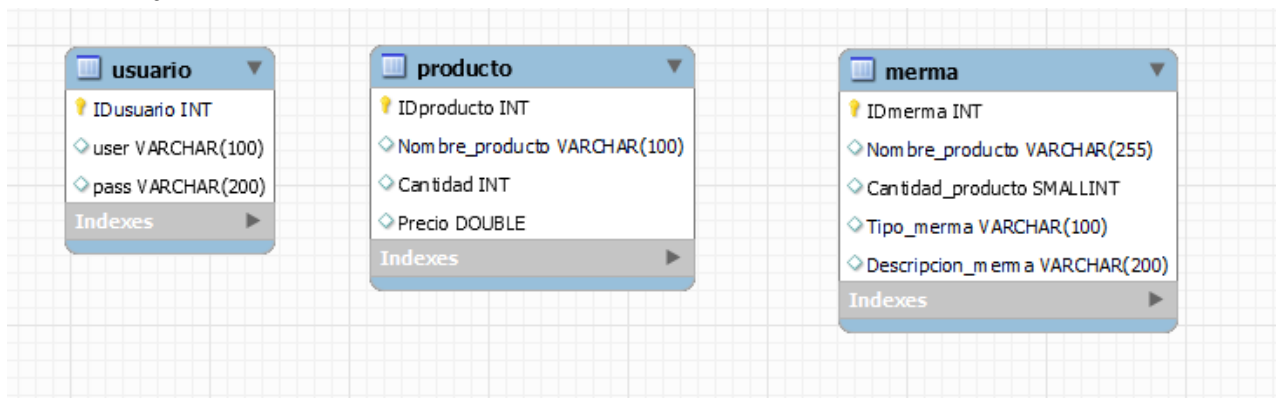
```

51  /*
52  Autor: Roberto Ibarra
53  Fecha:15/11/2022
54  Nombre:sp_ingresarmerma
55  Descripción: sirve para registrar una merma
56  */
57  delimiter $$
58  • create procedure sp_ingresarmerma(
59      IN _nombre varchar(255),
60      IN _cantidad smallint,
61      IN _tipo varchar(100),
62      IN _descripcion varchar(200) )
63  • Begin
64      insert into merma(nombre_producto,cantidad_producto,tipo_merma,descripcion_merma)
65      Values (_nombre,_cantidad,_tipo,_descripcion);
66      End
67      $$
68
69      -- datos de prueba del SP para ingresar mermas
70  • call sp_ingresarmerma("cerveza",4,"operacional","venia con la tapa media abierta");
71      call sp_ingresarmerma("bebida",2,"operacional","en mal estado");
72

```

Result Grid				
Filter Rows:				
Export:				
Wrap Cell Content:				
IDmerma	Nombre_producto	Cantidad_producto	Tipo_merma	Descripcion_merma
1	Yogurt Colun	1	Devolución	Se rompio al momento de venta, cliente se le cayo al suelo
2	cerveza	4	operacional	venia con la tapa media abierta
3	bebida	2	operacional	en mal estado

d. MER:



5. Pruebas:

Las únicas pruebas realizadas en el proyecto están detalladas en el caso de prueba.