Diseño de Subsistema

“LAUREL”

25/09/2021

—

Integrantes:

- Tomás Herceg
- Javiera Naranjo
- Scarlett Ojeda
- Álvaro Navarro

Artefacto de análisis

1. Modelo de clases conceptuales

1.1. Identificar conceptos

Categoría de clases conceptuales	Ejemplo
Objetos tangibles	Carta
Sistemas informáticos	PagoCredito, BoletaElectronica, RegistroCuenta, IngresoCuenta
Especificaciones o Descripciones de las cosas	DescripcionProducto
Lugares	Cocina, Barra, Terraza, Comedor, Restaurante
Transacciones	Pago, Reserva, Reembolso
Roles de las personas	Mesero, Cliente, Administrador
Cosas en el restaurante	Mesa, Silla, Utensilios de mesa
Reglas y Políticas	PoliticaDeDevolucion

1.2. Incorporar asociaciones

Categoría	Ejemplo
A es una parte física de B	Carta - Comedor
A es una parte lógica de B	PagoCredito - Pago
A esta fisicamente contenido en B	Mesa - Comedor
A esta lógicamente contenido en B	RegistroOnline - Reserva
A es una descripción de B	DescripciónProducto - Carta

A es un elemento de línea de una transacción o informe de B	PagoCredito - Pago
A se conoce/introduce/registra/presenta/captura en B	Pago - BoletaElectronica
A es miembro de B	Mesero - Comedor
A es una subunidad organizacional de B	Cocina - Restaurante
A usa o dirige B	Mesero - Carta
A se comunica con B	Cliente - Mesero
A se relaciona con una transacción B	BoletaElectronica - Pago
A es una transacción relacionada con otra transacción B	Pago - Reserva

1.3. Agregar atributos necesarios

Usuario:

- Rut: números
- Contraseña: texto

Productos/platos:

- Nombre: texto
- Grupo: texto
- Valor: números
- Ingredientes: texto

Carta:

- descripción: productos/platos
- fecha: fecha
- id: números

Inventario:

- descripción: producto/plato
- cantidad: números
- id: números

Pedido:

- NumeroMesa: números
- NumeroPedido: números

- Monto: números
- Descripción: producto/plato

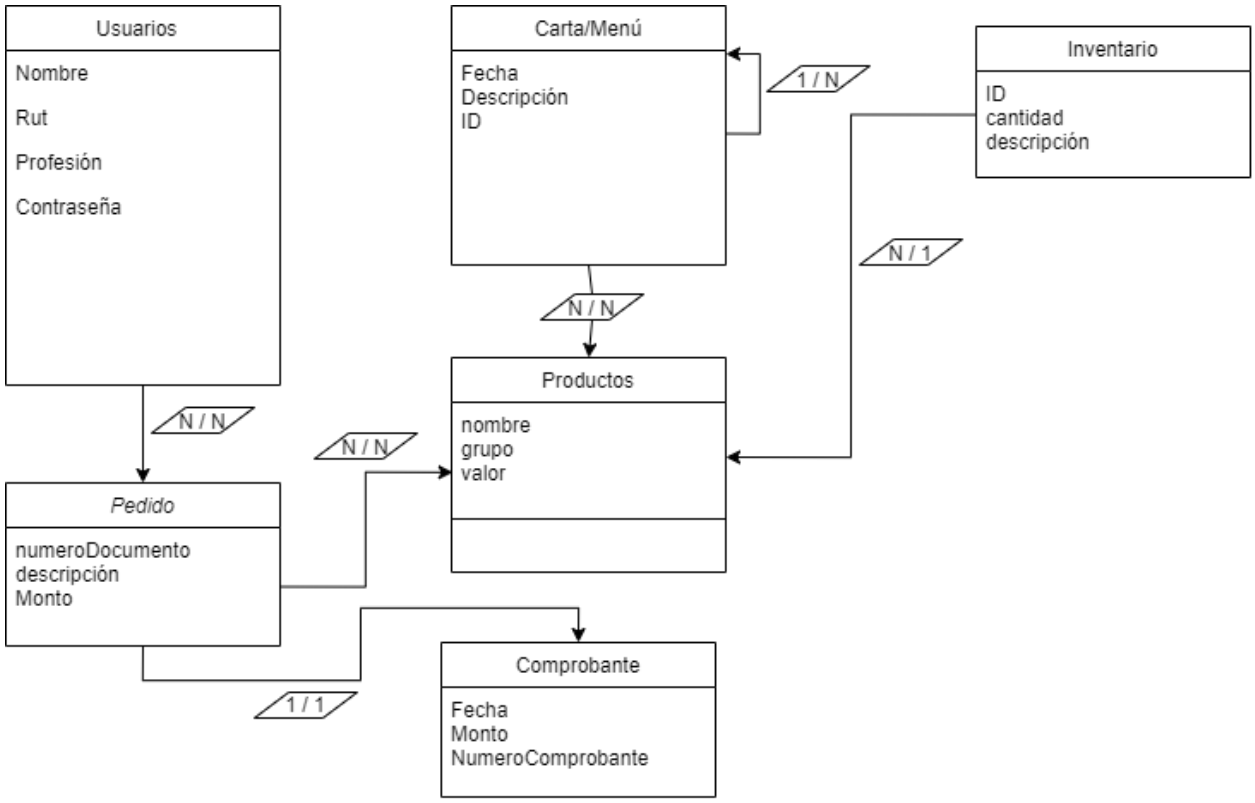
Pago:

- MetodoPago: tarjeta, efectivo
- Monto: números
- Descripción: Pedido

Comprobante:

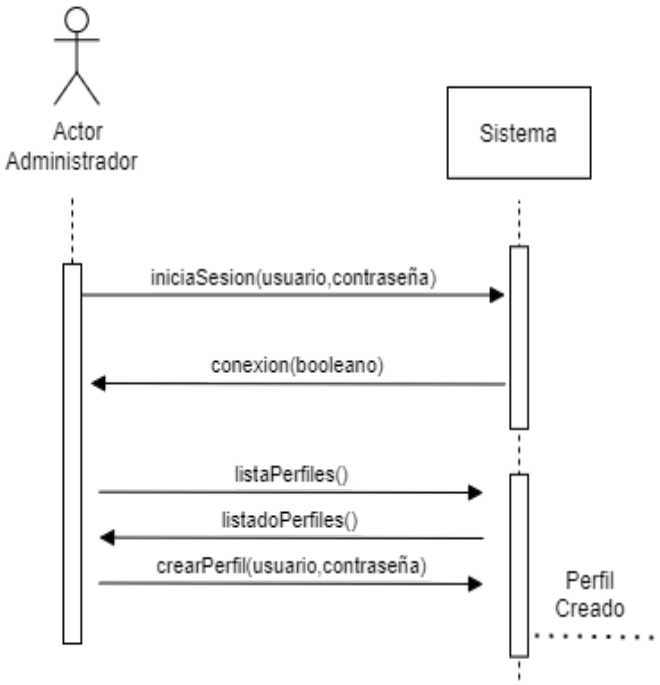
- Fecha: fecha
- Monto: números
- NumeroComprobante: números

1.4. Modelo Conceptual

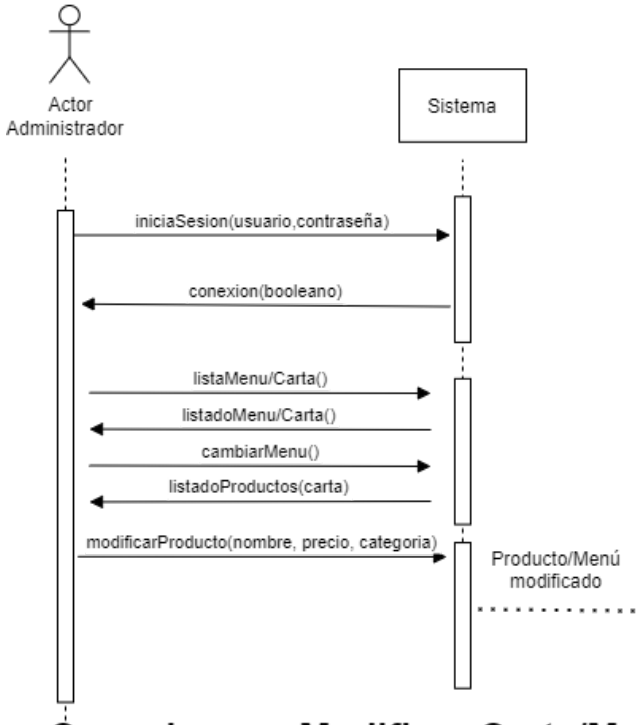


-

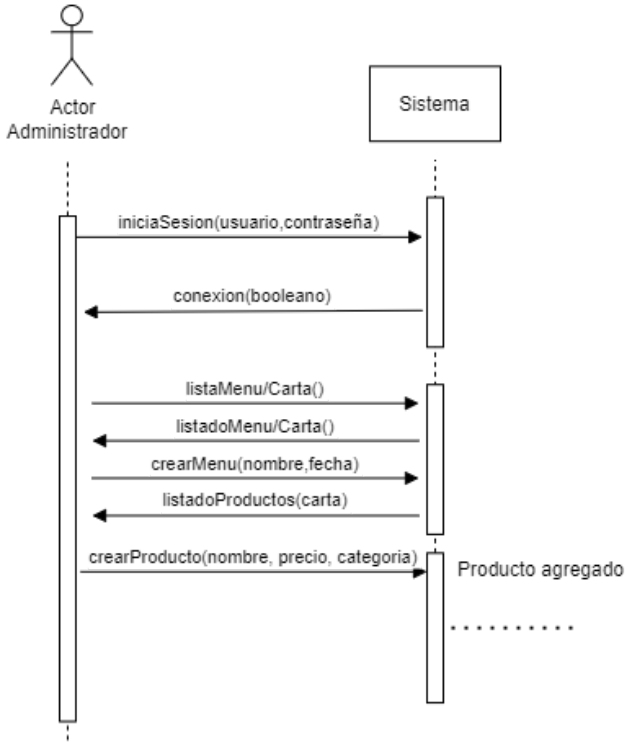
2. Diagrama de secuencia del sistema



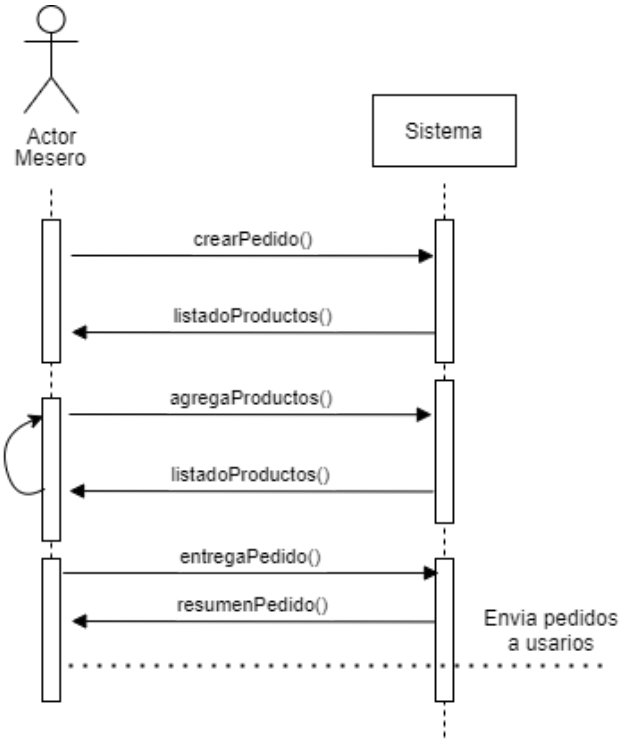
Caso de uso: Crear Usuario



Caso de uso: Modificar Carta/Menú



Caso de uso: Ingreso de productos a base de datos nueva



Caso de uso: Tomar un pedido

3. Contratos

3.1. Tomar pedidos

Nombre	**crearPedido()**
Responsabilidades	Tomar pedido del usuario para agregarlo al sistema.
Referencias cruzadas	Caso de Uso: Tomar Pedido
Precondiciones	Un usuario mesero debe estar tomando un pedido.
Poscondiciones	- Se crea la instancia de creación de pedido cdp (creación de instancias) - se asigna cdp con el pedido actual. - cdp.cantidad aumenta en 1 (modificación de atributos). - cdp.id se le asigna al número de mesa (modificación de atributos).

crearPedido(num_mesa) [Laurel] createPedido(id_pedido,num_mesa) [pedido]

Nombre	**agregarProducto(id_pedido,id_producto)**
Responsabilidades	Que se pueda agregar un producto al pedido.
Referencias cruzadas	Caso de Uso: Tomar Pedido
Precondiciones	Debe estar una de las instancias de creación de pedido, modificación de pedido, agregar a base de datos activa.
Poscondiciones	- Se busca producto.id_producto y se asocia con instancia de pedido.id_pedido. *(asociación formada)* - Se agrega a pedido.producto el producto asociado.

Agregarproducto(id.producto) [Laurel] addProducto(pt_pedido,pt_prod) [pedido]

pt_prod = :buscarProd(producto.id)

[Producto]

Nombre	**listadoProductos()**
Responsabilidades	Que se pueda mostrar todos los productos del sistema.
Referencias cruzadas	Caso de Uso: Tomar Pedido, Caso de uso: Agregar producto a base de datos
Precondiciones	Se debe haber abierto un menú o un mesero debe estar haciendo un pedido
Poscondiciones	- Se crea instancia de listadoProductos (Creación de instancia) - Se buscan todos los productos existentes en sistema y se asocia con instancia listadoProductos *(asociación formada)* - Se muestra la instancia listadoProductos.

ListadoProducto() [Laurel] listProducto(pt_lista) [Lista]

pt_lista = :findProducts() [Producto]

Nombre	**resumenPedido(id_pedido)**
Responsabilidades	Crear una lista con todos los productos agregados al pedido asociado.
Referencias cruzadas	Caso de Uso: Tomar Pedido
Precondiciones	Un usuario mesero debe estar tomando pedido
Poscondiciones	- Se asocia id_pedido a pedido.id_pedido (asociación formada) - Se muestra productos.id_productos asociados a pedido.id_pedido

resumenPedido(id_pedido) [Laurel] findPedido(pt_pedido) [pedido]

pt_pedido= :buscarPedido(pedido.id)

[Pedido]

3.2. Ingresar productos a base de datos

Nombre	**iniciarSesion(id_usuario, contraseña)**
Responsabilidades	Medio de autenticación para los trabajadores del local.
Referencias cruzadas	Caso de Uso: Ingresar productos a base de datos
Excepciones	Si el usuario no está registrado, se rechaza.
Salida	Confirmación: Booleano
Precondiciones	Un usuario necesita acceder a la aplicación
Poscondiciones	- Se asocia id_usuario a usuarios.id_usuario - Se comprueba que usuarios.id_usuario.contraseña sea igual que contraseña - Si lo anterior es verdadero se inicia sesión

iniciarSesion(usuario, contraseña) [Laurel] checkUser(pt_usuario, pt_contraseña) [Usuario]

pt_usuario= :buscarUsuario(usuario.id)

pt_contraseña =:comprobarContraseña(pt_usuario)

[Usuario]

Nombre	**listadoMenú/Cartas()**
Responsabilidades	Que se pueda mostrar todos los menús existentes en sistema.
Referencias cruzadas	Caso de Uso: Ingresar productos a base de datos
Precondiciones	Usuario administrador debe estar comprobado y pidendo ver el listado para su modificación o para agregar.
Poscondiciones	

	- Se buscan todos los menu.id_menu en sistema y se asocian con instancia listadoMenú *(asociación formada)* - Se muestra la instancia listadoMenú.

listadoMenu() [Laurel] findMenu(pt_menú) [Menú]

pt_menú= :buscarMenú(menú.id)

[Menú]

Nombre	**crearMenú/Cartas(Nombre, Fecha)**
Responsabilidades	Que se pueda crear un nuevo menú en el sistema.
Referencias cruzadas	Caso de Uso: Ingresar productos a base de datos
Precondiciones	Usuario administrador debe pedir crear un menú.
Poscondiciones	- Se crea una instancia menú *(creación de instancias)*. - Se asigna una id a menu.id - Se asigna Nombre a menu.nombre *(modificacion de atributos)*. - Se asigna a menu.fecha Fecha.*(modificacion de atributos)*.

crearMenu(Nombre, Fecha) [Laurel] addMenu(pt_menú) [Menu]

pt_menu =: createMenú(Nombre, Fecha)

[Menú]

Nombre	**listadoProductos(menú_id)**
Responsabilidades	Que se pueda mostrar todos los productos del sistema asociados al menú.
Referencias cruzadas	Caso de uso: Agregar producto a base de datos
Excepciones	Si no hay stock de algún producto, hay que modificar la carta.
Precondiciones	Se debe haber abierto un menú o un mesero debe estar

	haciendo un pedido
Poscondiciones	- Se crea instancia de listadoProducto *(Creación de instancia)* - Se asocia menú.id a la instancia listadoProducto.menú*(asociación formada)* - Se asocia productos.id_producto en sistema que correspondan a menu.id con instancia listadoProductos *(asociación formada)*

ListadoProducto(menú_id) [Laurel] listProducto(pt_menu) [Lista]

pt_menu =: findProducts(menu.id)

[Lista]

Nombre	**agregarProducto(nombre_id,precio_id,categoria_id)**
Responsabilidades	Que se pueda agregar un producto al pedido.
Referencias cruzadas	Caso de Uso: Ingresar producto a base de datos
Precondiciones	Debe estar una de las instancias de creación de pedido, modificación de pedido, agregar a base de datos activa.
Poscondiciones	- Se crea instancia producto (creación de instancias). - Se asocia menú.id a la instancia agregarProductos() *(asociación formada)* - Se asigna a producto.nombre el nombre dado por el usuario *(modificación de atributos).* - Se asigna a producto.precio el precio dado por el usuario *(modificación de atributos).* - Se asigna a producto.categoría la categoría dada por el usuario *(modificación de atributos).* - Se agrega producto al menú actual (menú.id)

agregarProducto(nombre_id,precio_id,categoria_id) [Laurel] createProducto(pt_producto) [Productos]

pt_producto =: crearProducto(nombre_id,precio_id,categoria_id) [Producto]

3.3. Modificar Menú/Carta

Nombre	**iniciarSesion(usuario, contraseña)**
Responsabilidades	Medio de autenticación para los trabajadores del local.
Referencias cruzadas	Caso de Uso: Modificar Menú
Excepciones	Si el usuario no está registrado, se rechaza.
Salida	Confirmación: Booleano
Precondiciones	Un usuario necesita acceder a la aplicación
Poscondiciones	- Se asocia id_usuario a usuarios.id_usuario - Se comprueba que usuarios.id_usuario.contraseña sea igual a contraseña - Si lo anterior es verdadero se inicia sesión

iniciarSesion(usuario, contraseña) [Laurel] checkUser(pt_usuario, pt_contraseña) [Usuario]

pt_usuario= :buscarUsuario(usuario.id)

pt_contraseña =:(pt_usuario)

[Usuario]

Nombre	**listadoMenú/Cartas()**
Responsabilidades	Que se pueda mostrar todos los menús existentes en sistema.
Referencias cruzadas	Caso de Uso: Modificar Menú
Precondiciones	Usuario administrador debe estar comprobado y pidiendo ver el listado para su modificación o para agregar.
Poscondiciones	- Se crea instancia de listadoMenú (Creación de instancia) - Se buscan todos los menu.id_menu en sistema y se asocian con instancia listadoMenú *(asociación formada)* - Se muestra la instancia listadoMenú.

listadoMenu() [Laurel] findMenu(pt_menú) [Menú]

pt_menú= :buscarMenú(menú.id)

[Menú]

Nombre	**cambiarMenú(id_menu,productos)**
Responsabilidades	Debe poder cambiar los productos dentro de un menú o sus atributos.
Referencias cruzadas	Caso de Uso: Modificar Menú
Precondiciones	Usuario Administrador debe estar requiriendo el cambio.
Poscondiciones	- Se asocia menú.id a id_menu. *(asociación formada)* - Se modifica los productos en el menu asociado

cambiarMenu(id_menu,productos) [Laurel] changeMenu(pt_menu) [Menu]

pt_menu =: changeProducts(productos)

[Menú]

Nombre	**listadoProductos(menú_id)**
Responsabilidades	Que se pueda mostrar todos los productos del sistema asociados al menú.
Referencias cruzadas	Caso de Uso: Modificar Menú
Precondiciones	Se debe haber abierto un menú.
Poscondiciones	- Se crea instancia de listadoProducto *(Creación de instancia)* - Se asocia menú.id a la instancia listadoProducto.menú *(asociación formada)* - Se buscan todos los productos.id_producto en sistema y se asocia con instancia listadoProductos *(asociación formada)* - Se muestra la instancia listadoProductos.

listadoProductos(menú_id) [Laurel] listProducto(pt_menu) [Lista]

pt_menu =: findProducts(menu.id)

[Lista]

Nombre	**modificarProducto(id_producto, atributos)**
Responsabilidades	Que se pueda mostrar todos los productos del sistema.
Referencias cruzadas	Caso de Uso: Modificar Menú
Precondiciones	Administrador debe requerir modificar un producto. Se debe haber seleccionado un menú.
Poscondiciones	- Se llama a la instancia producto. - Se asocia id_producto con instancia producto.id *(asociación formada)* - Se modifica el producto asociado con los nuevos atributos.

modificarProducto(pt.id) [Laurel] changeProduct(pt_producto) [Producto]

pt_producto =: ChangeProduct(atributos) [Producto]

3.4. Crear Usuario

Nombre	**iniciarSesion(usuario, contraseña)**
Responsabilidades	Medio de autenticación para los trabajadores del local.
Referencias cruzadas	Caso de Uso: Crear Usuario
Excepciones	Si el usuario no está registrado, se rechaza.
Salida	Confirmación: Booleano
Precondiciones	Un usuario necesita acceder a la aplicación
Poscondiciones	- Se asocia id_usuario a usuarios.id_usuario - Se comprueba que usuarios.id_usuario.contraseña sea igual a contraseña - Si lo anterior es verdadero se inicia sesión

iniciarSesion(usuario, contraseña) [Laurel] checkUser(pt_usuario, pt_contraseña) [Usuario]

pt_usuario= :buscarUsuario(usuario.id)

pt_contraseña =:(pt_usuario)

[Usuario]

Nombre	**listadoPerfiles()**
Responsabilidades	Que se pueda mostrar todos los productos del sistema.
Referencias cruzadas	Caso de Uso: Crear Usuario
Precondiciones	Usuario administrador pide el listado de perfiles.
Poscondiciones	- Se crea la instancia de listadoPerfiles *(Creación de instancia)* - Se asocia perfil.id existentes a listadoPerfiles.perfiles *(asociación formada)*.

listadoPerfiles(usuarios) [Laurel] listProfiles(pt_perfil) [Usuario]

pt_perfil= :buscarUsuario()

[Usuario]

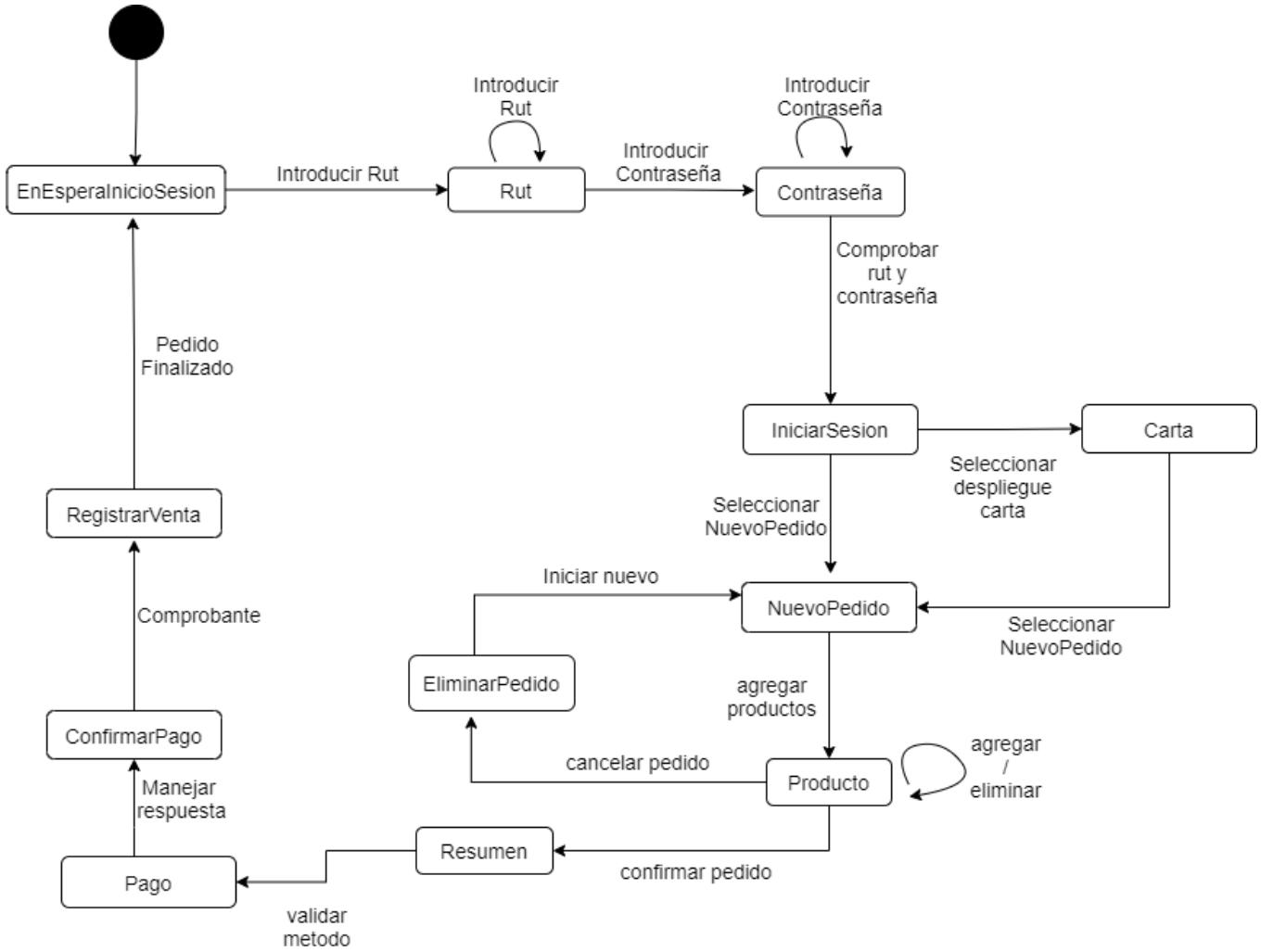
Nombre	**crearPerfil(usuario, contraseña, tipo)**
Responsabilidades	Que se pueda mostrar todos los productos del sistema.
Referencias cruzadas	Caso de Uso: Crear Usuario
Precondiciones	Usuario administrador va a crear un nuevo perfil.
Poscondiciones	- Se crea una instancia de perfil. - Se asigna usuario a perfil.usuario - Se asigna contraseña a perfil.contraseña - Se asigna tipo a perfil.tipo. - Se agrega perfil a base de datos.

crearPerfil(usuario,contraseña,tipo) [Laurel] addProfile (perf_usuario, perf_contraseña, perf_tipo)

perf_usuariol= : addProfile() [Usuario]

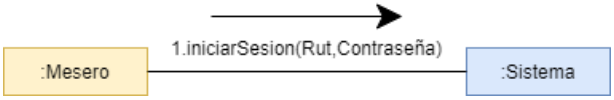
4. Diagrama de estado del caso de uso

Diagrama Tomar un pedido:

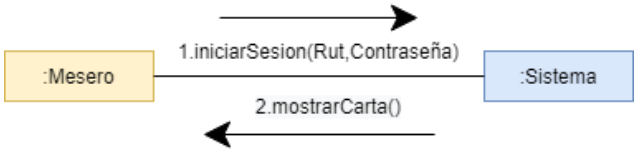


5. Diagrama de Colaboración

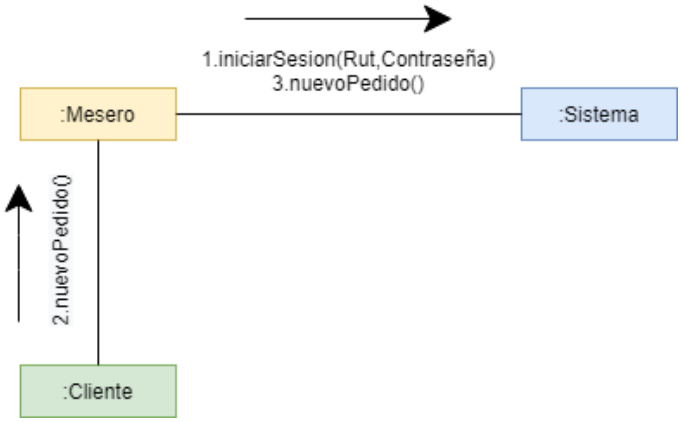
5.1 Iniciar Sesión



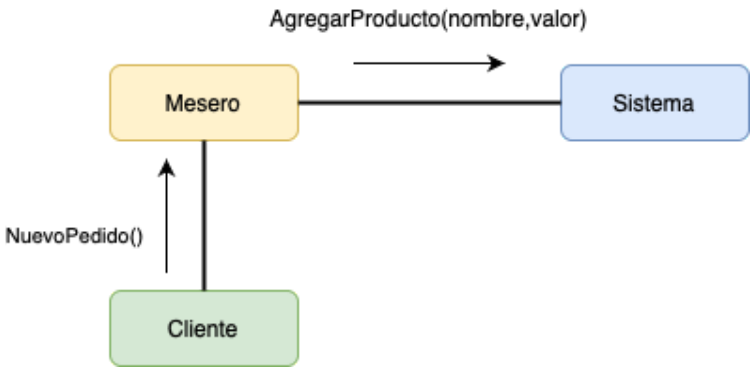
5.2 Mostrar Carta



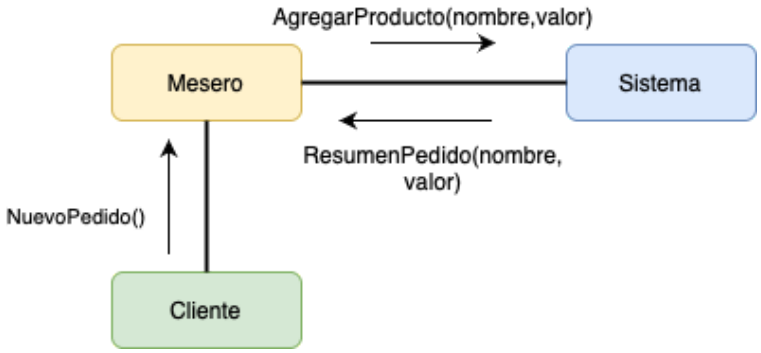
5.3 Nuevo Pedido



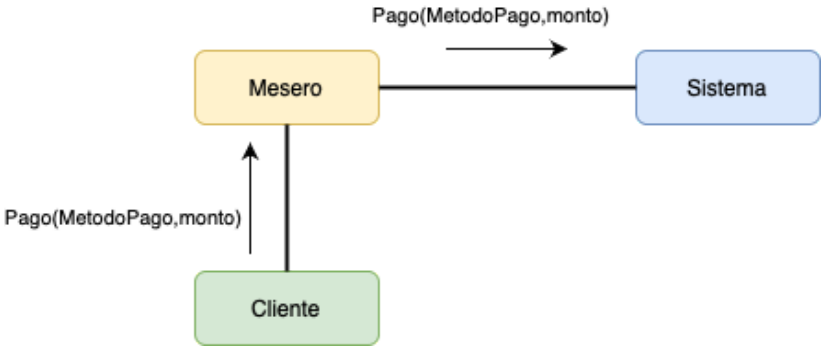
5.4 Agregar Producto



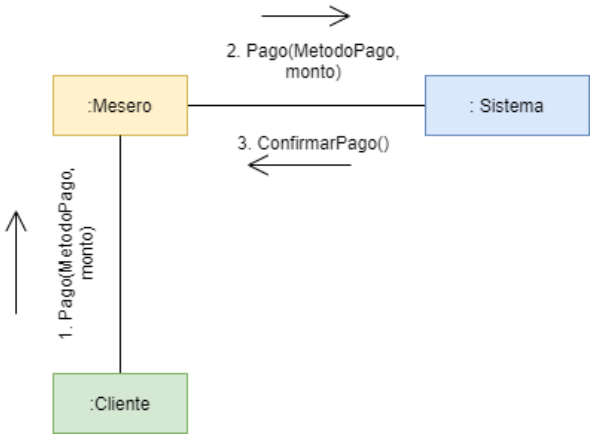
5.5 Resumen Pedido



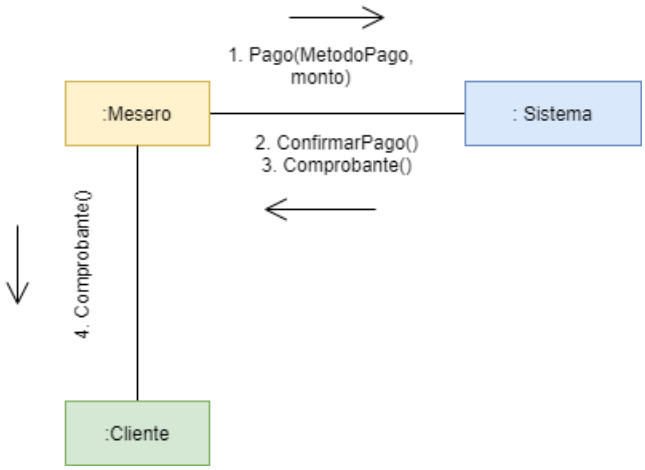
5.6 Pago



5.7 Confirmar Pago

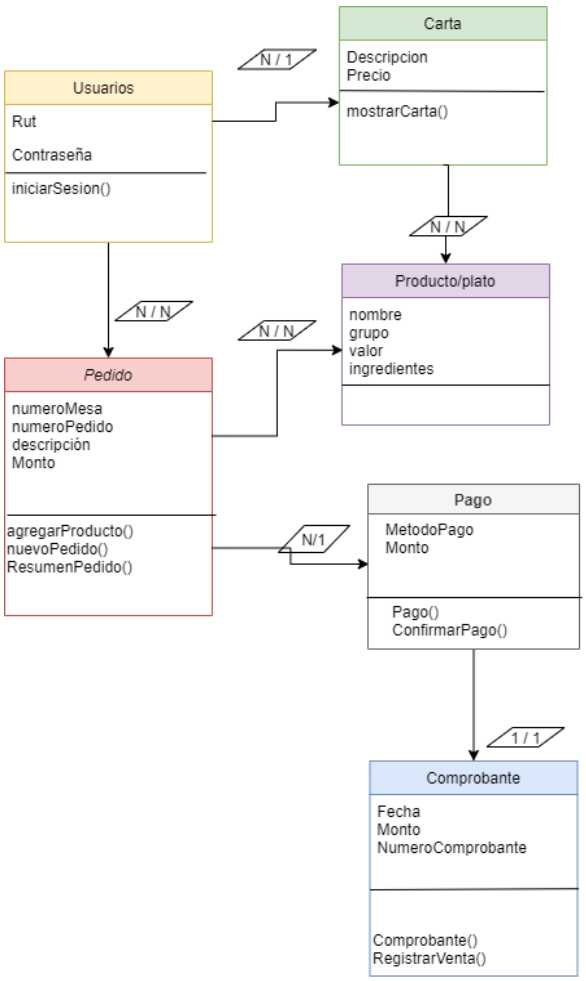


5.8 Comprobante



5.9 Registrar Venta

6. Diagrama de clases


Carta
Descripcion
Precio
mostrarCarta()
N / 1
Usuarios
Rut
Contraseña
iniciarSesion()
N / N
Producto/plato
nombre
grupo
valor
ingredientes
N / N
N / N
Pedido
numeroMesa
numeroPedido
descripción
Monto
agregarProducto()
nuevoPedido()
ResumenPedido()
N/1
Pago
MetodoPago
Monto
Pago()
ConfirmarPago()
1 / 1
Comprobante
Fecha
Monto
NumeroComprobante
Comprobante()
RegistrarVenta()