**Hamburguesería**

Nombre del sistema: **Puerco Araña**

Nombre del documento: **TP-Ingeniería-De-Software**

Apellido y Nombre del Alumno: **Gomez, Gonzalo**

Localización: **Sede Centro**

Comisión: **3-J**

Turno: **Noche**

Año de cursada: **3ro**

Apellido y Nombre del Docente: **Chamula, Christian Gabriel**

[**G00. Descripción Global del Producto 5**](#_58e4b6k541gh)

[**G01. Propósito 5**](#_sm5dicvpp4z1)

[**G02. Descripción funcional del producto y Alcance 6**](#_utcfq248kl1)

[Introducción 6](#_7nrblnlgh43g)

[Objetivos del Sistema 6](#_hlptr8rvugb8)

[Funcionalidades Principales 7](#_7dthq4z05ep0)

[**G03. Definiciones, Acrónimos, y Abreviaciones 7**](#_mu3xb8t6iz3y)

[**G04. Descripción de las personas participantes en el desarrollo del sistema de información y los usuarios (Roles) 9**](#_u8to3slrdg6)

[G04.01 Diagrama de Roles: 12](#_sxjd99j02cll)

[G04.03 Diagrama de Secuencia de Roles: 13](#_1j0vwkauaukv)

[G05. Otros Requisitos 13](#_zmu6q5zufsg)

[**N01 Proceso de negocio 14**](#_elemwghrbhp5)

[N01.1 Identificar roles intervinientes 14](#_kdhn0yuqjz9a)

[N01.2 Descripción funcional del proceso de negocio 14](#_b6zmt3lt4cy9)

[N01.3 Esquema ECS 17](#_cdjl6hqjc8wj)

[N01.4 Diagrama de proceso 17](#_urbl1olm89v0)

[N01.4 Modelo conceptual 19](#_ythainr4r267)

[**N02. Especificaciones de Casos de Uso 20**](#_yneq3znfjjqv)

[N02.1 Diagrama de Casos de Uso General 20](#_lauyilxocxhd)

[N02.2 CU 001 “Gestionar mesas” 20](#_y9cvks3obahh)

[N02.2.1 Especificación CU 001 “Gestionar mesas” 20](#_bwb6wflx0j8)

[N02.2.2 Diagrama de secuencia CU 001 “Gestionar mesas” 21](#_2vbnpiaoagc)

[N02.2.3 Diagrama de clases CU 001 “Gestionar mesas” 22](#_4qym0e7xbwqg)

[N02.2.4 Modelo de datos CU 001 “Gestionar mesas” 23](#_kdtxl0jj5qtc)

[N02.3 CU 002 “Realizar comanda” 24](#_h5j8cdqvovdl)

[N02.3.1 Especificación CU 002 “Realizar comanda” 24](#_pf6vpdl93j01)

[N02.3.2 Diagrama de secuencia CU 002 “Realizar comanda” 25](#_3rxdl5syw32j)

[N02.3.3 Diagrama de clases CU 002 “Realizar comanda” 26](#_uvc5of17fq39)

[N02.3.4 Modelo de datos CU 002 “Realizar comanda” 26](#_8oldixfcqzyp)

[N02.4 CU 003 “Preparar comanda” 27](#_ies498ii87wd)

[N02.4.1 Especificación CU 003 “Preparar comanda” 27](#_lnrp993dwg3y)

[N02.4.2 Diagrama de secuencia CU 003 “Preparar comanda” 28](#_p3hta34jf10h)

[N02.4.3 Diagrama de clases CU 003 “Preparar comanda” 29](#_qz8n26z0kx8w)

[N02.4.4 Modelo de datos CU 003 “Preparar comanda” 29](#_5ij95wlgvpxq)

[N02.5 CU-004: Notificar comanda lista 30](#_tsria5np8lrk)

[N02.5.1 Especificación CU-004: “Notificar comanda lista” 30](#_s8qu7tl7t4hb)

[N02.5.2 Diagrama de secuencia CU-004: “Notificar comanda lista” 30](#_nqtq1fwpajdg)

[N02.5.3 Diagrama de clases CU-004: “Notificar comanda lista” 31](#_fib5dkh5d81z)

[N02.5.4 Modelo de datos CU-004: “Notificar comanda lista” 32](#_6u5ktx20d5p9)

[N02.6 CU-005: ”Cobrar y generar factura” 32](#_a26go8m1hxmz)

[N02.6.1 Especificación CU-005: ”Cobrar y generar factura” 33](#_5xtj51qkehiz)

[N02.6.2 Diagrama de secuencia CU-005: ”Cobrar y generar factura” 34](#_egdptlkro30j)

[N02.6.3 Diagrama de clases CU-005: ”Cobrar y generar factura” 34](#_vbwuojozseco)

[N02.6.4 Modelo de datos CU-005: ”Cobrar y generar factura” 35](#_whkmkldobgyq)

[**N00.1 Procesos de negocio 1 (Entrega 2) 36**](#_7rqfn2ib8eug)

[**N00.2 Procesos de negocio 1 (Entrega 3) 36**](#_wjxis83s26bk)

[**T01. Arquitectura Base 36**](#_j1mhi3bocqxs)

[T01.1 Diseño de la arquitectura (Diagrama de capas) 37](#_6e8jwy1dvz63)

[T01.2 - Captura de pantalla de las capas creadas en el IDE 37](#_wy5slx6dbvuo)

[T01.3 - Esquema de Persistencia de Datos 39](#_339lr75w9k4d)

[T01.3 - Esquema de Consulta de Datos 40](#_qbiehl58ekdv)

[**T02. Gestión de Log In / Log Out del Sistema 41**](#_7j9udj14fqe7)

[T02.1. CU 006 Gestionar usuario “Crear usuario” 41](#_9tux94io3tbq)

[T02.1.1 Descomposición Funcional: Crear Usuario 41](#_ijwzvhxk2wwn)

[T02.1.2 Diagrama de caso de uso CU 006 “Crear usuario” 41](#_oxloxzv1v1pe)

[T02.1.3 Especificación CU 006 “Crear usuario” 42](#_iaguwl27it81)

[T02.1.4 Diagrama de secuencia CU 006 “Crear usuario” 43](#_g0ju1kpexixb)

[T02.1.4 Diagrama de clases CU 006 “Crear usuario” 45](#_s9bcfbne8fhg)

[T02.1.4 Modelo de datos CU 006 “Crear usuario” 45](#_uchpka3rxqil)

[T02.2. CU 007 “Realizar login” 46](#_x35f723s9n57)

[T02.2.1 Descomposición Funcional: “Realizar login” 46](#_a95zlsmlv42k)

[T02.2.2 Diagrama de caso de uso CU 007 “Realizar login” 46](#_ijom5ido9ddj)

[T02.2.3 Especificación CU 007 “Realizar login” 47](#_67maoy82i690)

[T02.2.4 Diagrama de secuencia CU 007 “Realizar login” 48](#_k0jcjqx16mmi)

[T02.2.4 Diagrama de clases CU 007 “Realizar login” 50](#_6dj77k3mjt5c)

[T02.2.4 Modelo de datos CU 007 “Realizar login” 50](#_3retxrzbkoc2)

[T02.3. CU 008 “Realizar Logout” 51](#_24rxwifv2p3r)

[T02.3.1 Descomposición Funcional: “Realizar logout” 51](#_tid3cveniw6o)

[T02.3.2 Diagrama de caso de uso CU 008 “Realizar Logout” 51](#_xhuae21xi6ku)

[T02.3.3 Especificación CU 008 “Realizar Logout” 52](#_28oj9fwp485l)

[T02.3.4 Diagrama de secuencia CU 008 “Realizar Logout” 53](#_se4m11m40u80)

[T02.3.4 Diagrama de clases CU 008 “Realizar Logout” 54](#_7s7z49ukjk6t)

[**T03. Gestión de Encriptado 56**](#_yw8l3oey9sf2)

[T03.1 Criptografía irreversible 56](#_xs8eodhpzex8)

[T03.2 Criptografía reversible 58](#_fpviulmzff6r)

[**T04. Gestión de Perfiles de Usuario (Entrega 2) 60**](#_b0frl94relei)

[**T05. Gestión de Múltiples Idiomas (Entrega 2) 60**](#_7t0iie516kv8)

Historial de revisión

| Fecha | Versión | Autor | Descripción |
| --- | --- | --- | --- |
|  |  |  |  |

# G00. Descripción Global del Producto

En este grupo se describe el producto con un conjunto de características de alto nivel que permiten comprender tanto el negocio como el producto de forma global.

# G01. Propósito

*El establecimiento “Puerco Araña”, fundado en 2018 en la localidad de San Fernando, ha destacado por su estilo moderno y ambiente que combina la gastronomía, arquitectura, tecnología y música con la mística del rock ´n roll, creando un gran vínculo con el público. Durante estos últimos años su clientela creció de forma exponencial, lo que llevó a grandes filas para ingresar al local. Por dicha complicación para la gestión de la demanda, se generaron desafíos operativos que requieren soluciones eficientes.*

Con el propósito de mejorar la experiencia del cliente y facilitar los procesos internos, la compañía FSSC (Fast Solution Software Company) desarrollará un sistema integral para la gestión de compra y venta.

El sistema propuesto permitirá la gestión informática de todas las tareas que se dan en el local, haciendo foco en la asignación de mesas, realizar pedidos y el cobro de los mismos. Dentro de la asignación de mesas, será posible la asignación de clientes a las mesas disponibles. Al realizar pedido, se permitirá a los clientes el poder solicitar alimentos y bebidas. Además, en el proceso de pago, se brindará flexibilidad de pago para la comodidad de los consumidores.

Adicionalmente, el sistema contará con funcionalidades para gestionar el inventario del restaurante, permitiendo monitorear el stock de productos. Cuando los niveles de inventario sean bajos, el sistema podrá comunicarse automáticamente con proveedores para realizar pedidos de reabastecimiento, asegurando que nunca falten productos esenciales en la cocina y el bar. Esto optimizará no solo la operación diaria del restaurante, sino también la relación con proveedores, reduciendo la probabilidad de interrupciones en el servicio debido a la falta de insumos.

Con estas mejoras, Puerco Araña busca optimizar su operación, reducir errores en pedidos y pagos, mejorar la gestión de inventario y abastecimiento, y ofrecer una experiencia de servicio más fluida y satisfactoria para sus clientes.

# 

# G02. Descripción funcional del producto y Alcance

### **Introducción**

El sistema integral de gestión propuesto para el establecimiento “Puerco Araña” busca optimizar y automatizar los procesos operativos en el restaurante, enfocándose en la gestión de mesas, pedidos, pagos y control de inventario. Ante el crecimiento exponencial de la clientela y los desafíos operativos resultantes, este sistema permitirá mejorar la experiencia del cliente y facilitar la operación interna del restaurante, asegurando un servicio más eficiente y satisfactorio.

### **Objetivos del Sistema**

* **Optimizar la Gestión de Mesas**: Asignar mesas a los clientes de manera eficiente, minimizando el tiempo de espera y maximizando la capacidad del local.
* **Facilitar la Toma de Pedidos**: Permitir a los meseros gestionar los pedidos de forma ágil, incluyendo la personalización de los productos solicitados por los clientes.
* **Mejorar el Proceso de Pagos**: Ofrecer múltiples opciones de pago, incluyendo la posibilidad de dividir cuentas, para adaptarse a las preferencias de los clientes.
* **Controlar el Inventario de Manera Eficiente**: Monitorear el stock de productos en tiempo real, asegurando la disponibilidad de insumos y evitando interrupciones en el servicio.
* **Analizar el Comportamiento del Cliente**: Recopilar datos sobre las preferencias y hábitos de consumo de los clientes para mejorar la oferta de productos y servicios.

### **Funcionalidades Principales**

1. **Gestión de Mesas**:
   * Asignación y visualización de mesas ocupadas y desocupadas en tiempo real.
   * Registro de la cantidad de clientes por mesa.
2. **Gestión de Pedidos**:
   * Creación, modificación y envío de pedidos a la cocina mediante dispositivos posnet.
   * Posibilidad de personalizar pedidos y registrar notas adicionales.
3. **Gestión de Pagos**:
   * Opciones de pago flexibles: efectivo, transferencia, Mercado Pago, tarjetas de crédito y débito.
   * Capacidad de dividir cuentas entre los clientes y emitir facturas electrónicas.
4. **Control de Inventario**:
   * Monitoreo continuo del stock y alertas automáticas para reabastecimiento.
   * Integración con proveedores para facilitar pedidos de productos.

# 

# G03. Definiciones, Acrónimos, y Abreviaciones

**Definiciones:**

**Comanda:** orden escrita que se genera en un establecimiento.

**Horno de convección:** Horno que utiliza un sistema de ventilación para distribuir el calor de manera uniforme.

**Punto de cocción:** Momento en el cual un producto horneado se considera cocido y listo para ser retirado del horno.

**Gluten:** Proteína presente en la harina que da elasticidad y estructura a la masa.

**Masa madre:** Mezcla de harina y agua fermentada usada naturalmente para hacer pan.

**Cocinero:** Experto en la preparación y cocción de los productos ofrecidos, dirige el proceso de producción y garantiza la calidad de los productos.

**Acrónimos:**

**POS:** Punto de Venta. Sistema utilizado para realizar transacciones y gestionar las ventas en la panadería.

**IVA:** Impuesto al Valor Agregado. Impuesto sobre el consumo que se aplica en productos y servicios.

**SAT:** Servicio de Atención al Cliente. Departamento encargado de brindar asistencia y solucionar consultas de los clientes.

**ERP:** Planificación de Recursos Empresariales. Software que ayuda a gestionar procesos como inventario, producción y finanzas.

**AFIP:** (Administración Federal de Ingresos Públicos) es la agencia gubernamental encargada de la administración y control de los impuestos, aduanas y regímenes de seguridad social en Argentina.

**Abreviaturas:**

**kg:** Kilogramo

**gr:** Gramo

**ml:** Mililitro

**cm:** Centímetro

**uni:** Unidad

**c/u:** Cada uno

**°C:** Grados Celsius (temperatura)

**horno:** hno. (abreviatura de "horno")

**min:** Minuto(s)

**hr:** Hora(s)

**pdto:** Producto

**rec:** Receta

**ingr:** Ingrediente

**env:** Envase

**desc:** Descuento

**promo:** Promoción

**fact:** Factura

**dep:** Departamento

**dir:** Dirección

**tel:** Teléfono

# G04. Descripción de las personas participantes en el desarrollo del sistema de información y los usuarios (Roles)

| **PERSONAL** | **ROL / DESCRIPCIÓN** | **RESPONSABILIDAD / ACCESO** |
| --- | --- | --- |
| **Paul McCarne** | **Gerente del Bar –** Gestión general del bar, supervisión del personal, inventarios y operaciones diarias | **RFN1- RFN2 - RFN3**  Tendrá acceso a todas las  funcionalidades del sistema. |
| **Freddie Mercookie** | **Cliente –** Persona que solicita productos para el consumo | **RFN1**  Podrá visualizar el catálogo de productos así como el pago de sus pedidos. |
| **Bread Pitt** | **Mesero –** Persona encargada de gestionar las mesas y entregar los pedidos | **RFN1**  Tendrá acceso a todas las funcionalidades del sistema relacionadas a gestionar mesas y pedidos. |
| **Danny Drinkwater** | **Bartender –** Encargado de preparar y servir bebidas alcohólicas y no alcohólicas | **RFN1 – RFN3**  Podrá visualizar los pedidos de las distintas mesas. |
| **Warren Buffet** | **Caja –** Encargado de gestionar la caja, manteniendo balance en la misma y cobrando los pedidos en efectivo | **RFN1 – RFN2**  Tiene acceso a todas las funcionalidades del sistema relacionadas con la gestión de las mesas y el cobro de los pedidos. Así también como la gestión de las compras y gastos del local. |
| **Gordon Ramsay** | **Cocinero –** Persona encargada de elaborar las comandas. | **RFN1**  Podrá visualizar las comandas, así como modificarlos, eliminarlos marcarlos como “En preparación” o “Listo”. |
| **Clean Eastwood** | **Limpieza –** Persona encargada de limpiar y mantener orden en la vajilla | **RFN1** |
| **Gustavo Cerati** | **Analista Funcional** | **RFN1 - RFN2 - RFN3** Tendrá acceso a todas las  funcionalidades del sistema |
| **Gonzalo Gomez** | **Analista Programador** | **RFN1 - RFN2 - RFN3** Analista y el desarrollador del  sistema de gestión de consultorio. |

## G04.01 Diagrama de Roles:

#### 

#### 

#### 

#### 

#### 

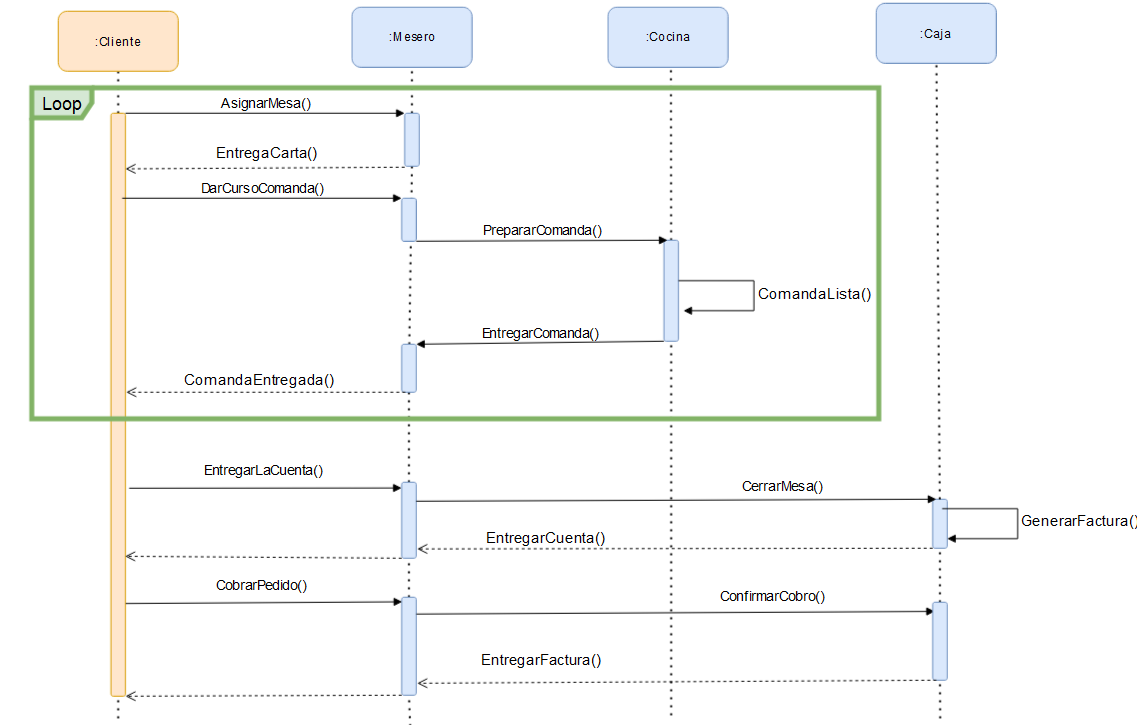
#### 

#### 

#### 

#### 

## G04.03 Diagrama de Secuencia de Roles:



## G05. Otros Requisitos

* Estándares aplicables:
  + El sistema debe tener una interfaz de usuario amigable.
  + El sistema debe tener el logo de la empresa en algún lugar.
  + El sistema debe ser veloz.
* Requisitos de Desempeño: Se espera que FSSC garantice tiempos de carga rápidos, con una velocidad de respuesta elevada para agilizar los procesos y una experiencia fluida para los usuarios. Además, priorizará el uso eficiente de la memoria para maximizar el rendimiento del sistema. En caso de, por ejemplo, tardar más de 30 segundos en cargar información de un evento o de cualquier acción solicitada, se lo devolverá a la página de inicio.
* Requisitos del Sistema: FSSC estará diseñado para funcionar en hardware estándar de uso común, con requisitos mínimos que incluyen un procesador compatible, memoria ram suficiente para permitir fluidez, y suficiente capacidad de almacenamiento. El hardware en el que se utilizará el sistema no es de alto nivel, por lo que no es deseado que esto sea un impedimento a la hora de trabajar.
* Requisitos de Entorno: El sistema será compatible con Windows, el cual es el SO montado en las máquinas. Se deberá disponer de una conexión a internet estable para acceder al sistema y realizar transacciones en línea, por ejemplo, a la hora de los pagos.

# N01 Proceso de negocio

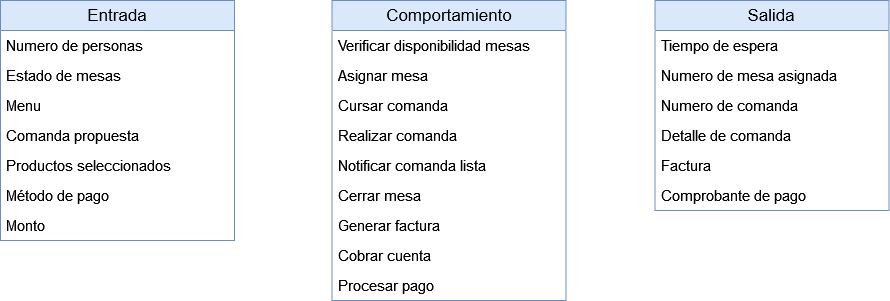
## N01.1 Identificar roles intervinientes

* **Comensal (Persona – Actor directo – Fuente de información)**
* **Mesero (Persona – Primario - Usa GUI)**
* **Caja (Persona - Primario - Usa GUI)**
* **Cocina (Persona - Secundario - Usa GUI)**
* **Banco (Sistema – Se conecta con el Sistema de Ventas – Remoto)**

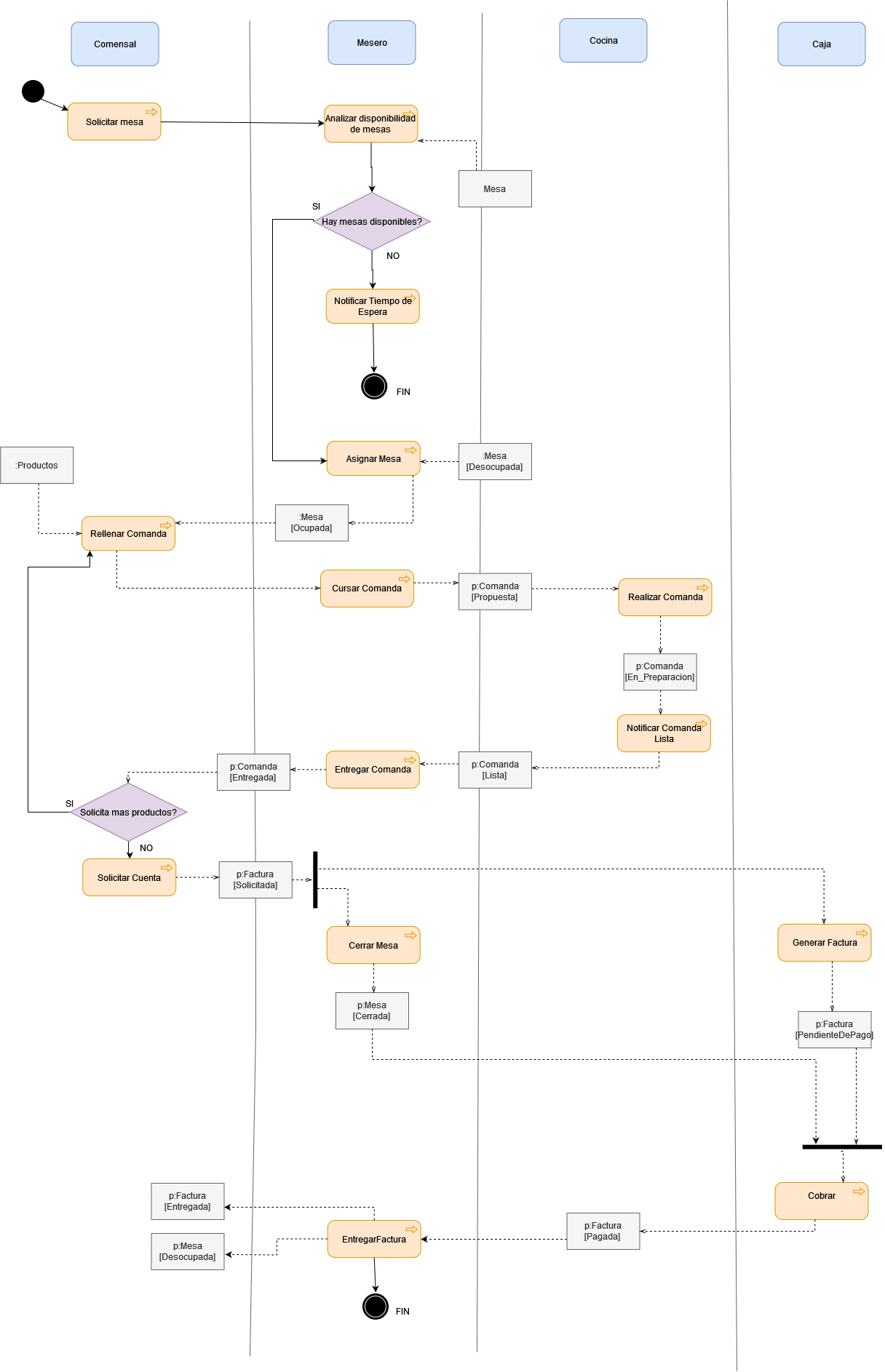
## N01.2 Descripción funcional del proceso de negocio

1. **Solicitud de Mesa:** El comensal solicita una mesa al mesero, indicando el número de personas. El sistema verifica la disponibilidad de mesas y asigna una si hay lugar. En caso contrario, se informa al comensal el tiempo estimado de espera.
   * **Campos:** Número de personas.
2. **Verificación de Disponibilidad:** El sistema consulta la base de datos para verificar qué mesas están libres y cuáles ocupadas.
   * **Campos:** Estado de la mesa (desocupada/ocupada), capacidad de la mesa.
3. **Asignación de Mesa:** Se asigna una mesa libre a los comensales. El sistema actualiza el estado de la mesa a "ocupada".
   * **Campos:** Número de mesa, estado de la mesa
4. **Toma de Pedido:** El mesero toma el pedido del comensal y lo introduce en el sistema.
   * **Campos:** Número de pedido, mesa asignada, código de producto, cantidad, precio unitario, personalizaciones (observaciones especiales del cliente).
5. **Preparación de Pedido:** El sistema envía el pedido a la cocina, que comienza a prepararlo.
   * **Campos:** Estado del pedido (en espera, en preparación, listo).
6. **Entrega de Pedido:** El mesero lleva el pedido a la mesa asignada.
   * **Campos:** Estado del pedido (entregado).
7. **Consumo y Nuevos Pedidos:** Si el comensal desea pedir más, se repite el proceso de toma de pedido, asignando el nuevo pedido a la misma mesa.
   * **Campos:** (Mismos que en el paso de "Toma de Pedido").
8. **Solicitud de Cuenta:** El comensal solicita la cuenta. El sistema busca la comanda asociada a la mesa.
9. **Cierre de Mesa:** El sistema marca la mesa como “desocupada” y genera la factura correspondiente.  
   **Campos:** Estado de la mesa (desocupada).
10. **Generación de Factura:** Se crea una factura detallada con todos los productos consumidos, sus precios y el total a pagar.
    * **Campos:** Número de factura, fecha, mesa asignada, productos (código, descripción, cantidad, precio unitario), subtotal, total, impuestos, forma de pago, número de cliente (si está registrado), datos de contacto del cliente (si están disponibles).
11. **Cobro:** El cliente realiza el pago y el sistema registra la transacción.
    * **Campos:** Número de pago, método de pago, monto, fecha.
12. **Entrega de Factura:** Se entrega una copia de la factura al cliente, ya sea en formato físico o electrónico.
    * **Campos:** Número de factura.

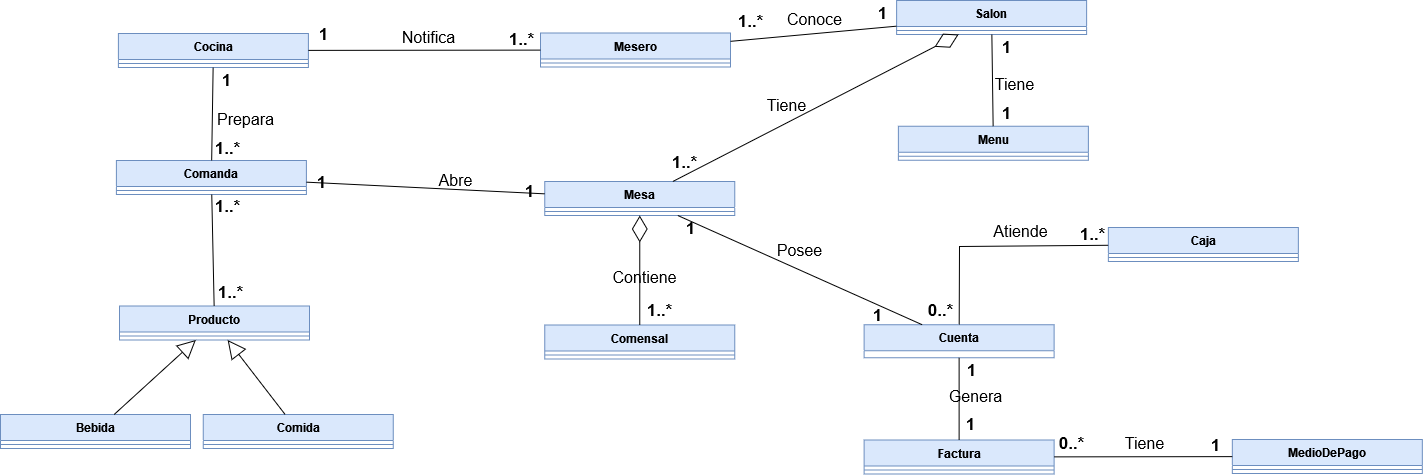
## N01.3 Esquema ECS



## N01.4 Diagrama de proceso

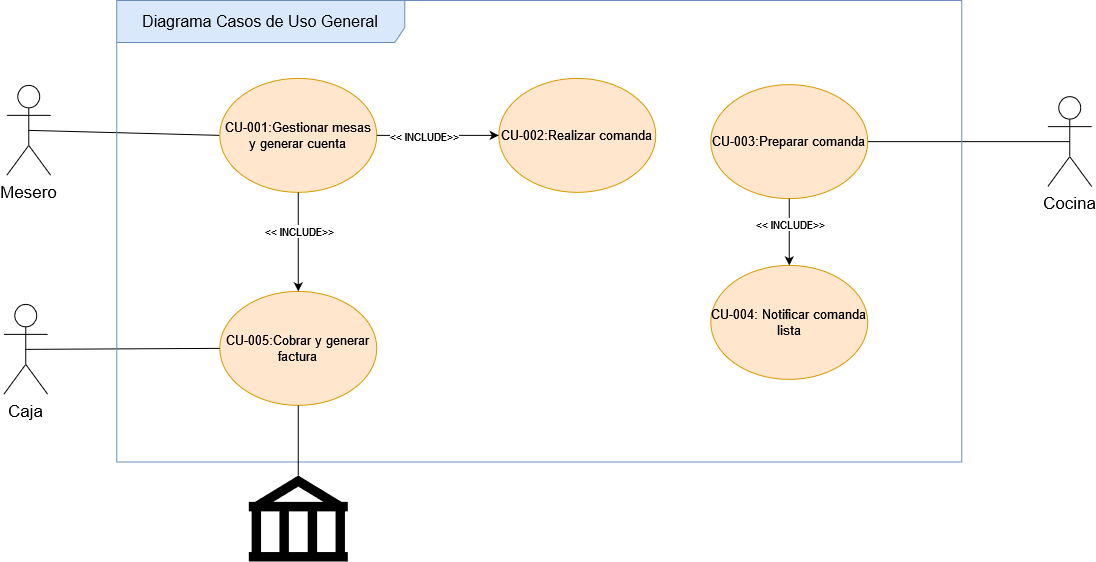


## N01.4 Modelo conceptual



# **N02. Especificaciones de Casos de Uso**

## N02.1 Diagrama de Casos de Uso General

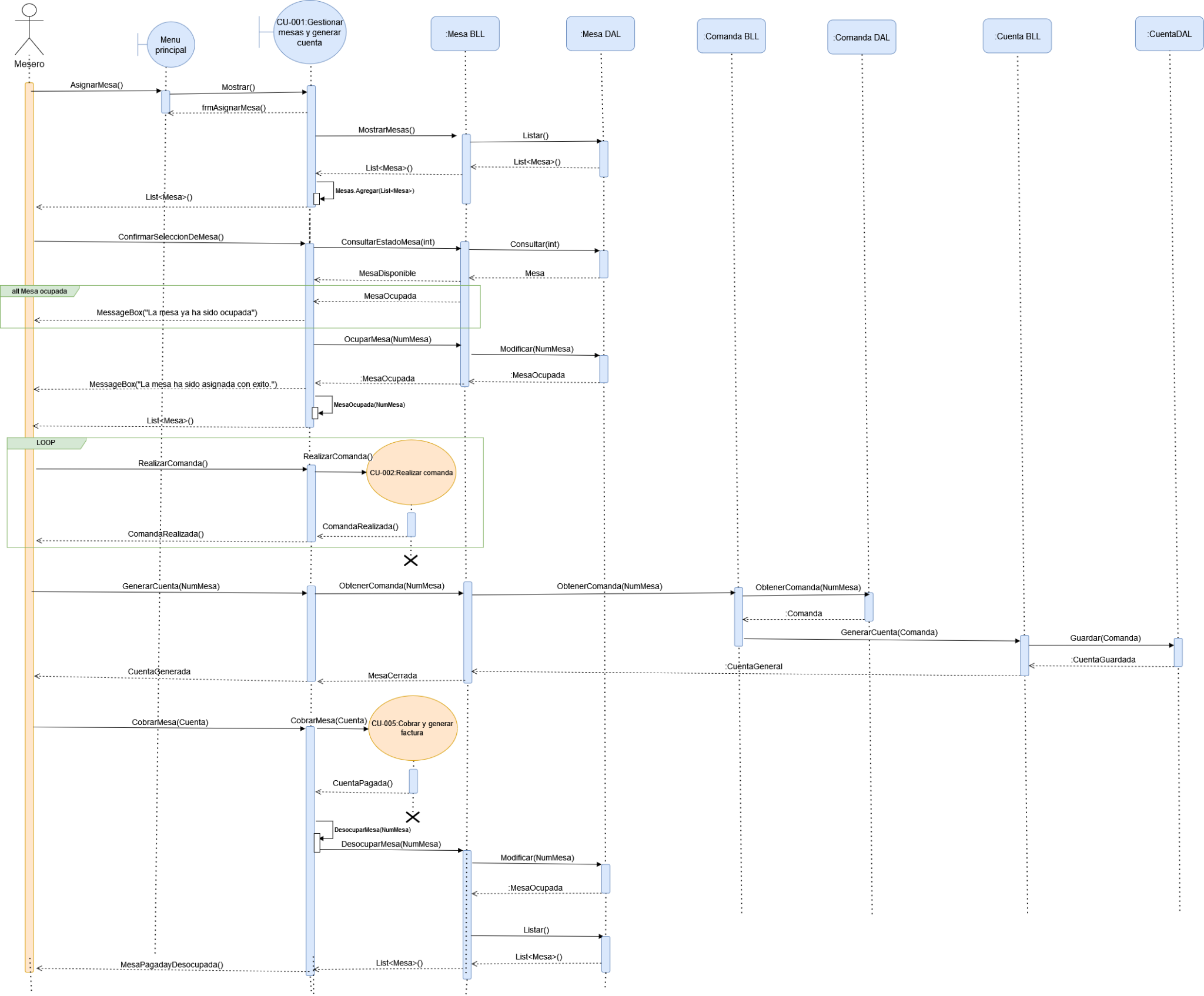


## **N02.2 CU 001 “Gestionar mesas”**

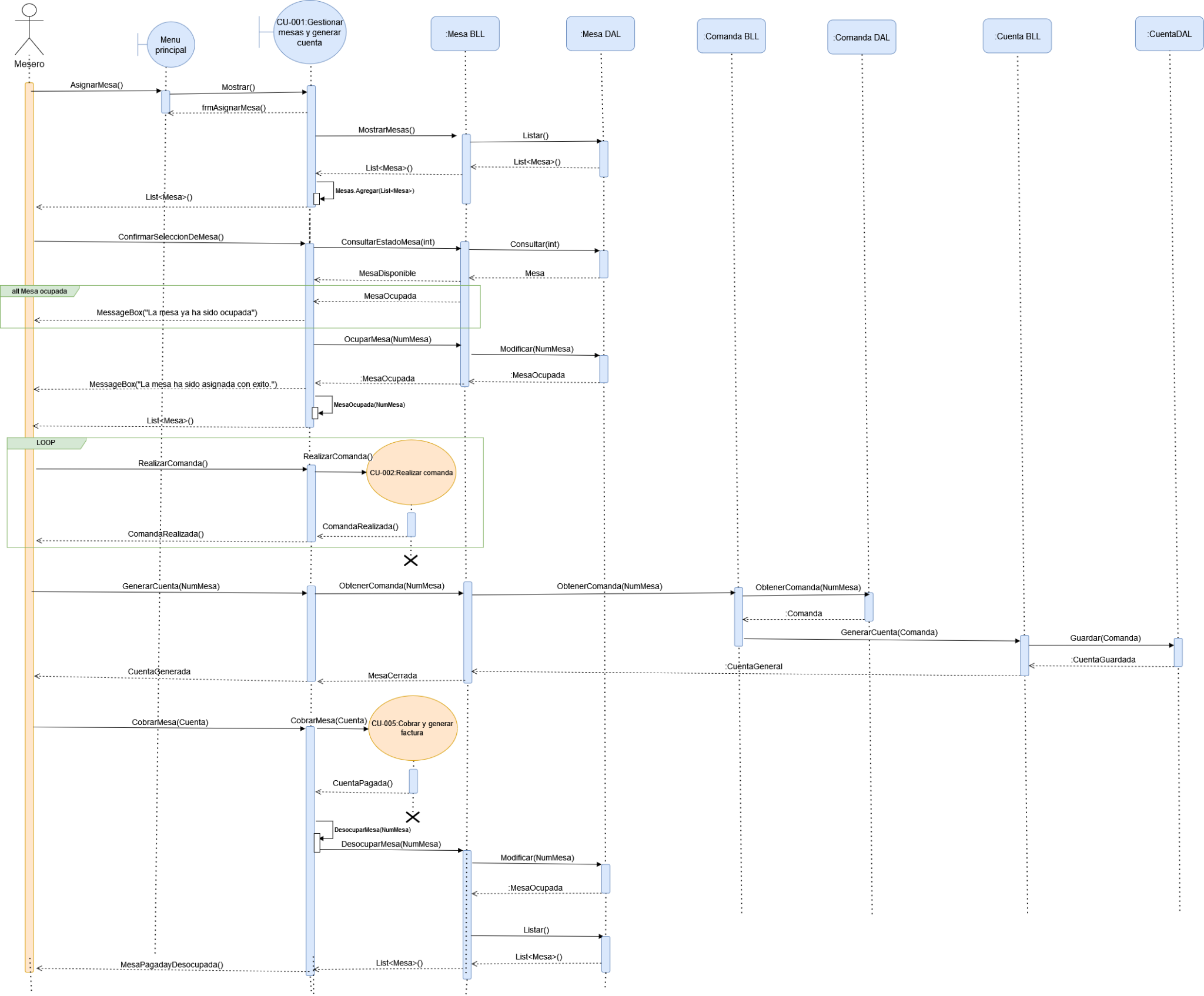
### N02.2.1 Especificación CU 001 “Gestionar mesas”

| **Nombre:** CU-001 “Gestionar mesas” |
| --- |
| **Objetivo:** Encargarse de asignar mesa a los comensales, abrir y cerrar las mesas. |
| **Actor principal:** Mesero |
| **Actor Secundario:** Comensal |
| **Precondiciones:** El actor debe estar logueado en el sistema |
| **Disparador:** El mesero ingresa a la pantalla “Gestionar mesa” |
| **Postcondiciones:** La cuenta se encuentra pagada y la mesa desocupada para ser asignada a otros comensales |
| **Flujo principal:**   1. El usuario selecciona la opción “Gestionar mesas” en el menú principal. 2. El sistema abre una pantalla con el listado de mesas con su estado actual, su número de mesa y la cantidad de comensales que tiene asignado. 3. El usuario selecciona una mesa desocupada. 4. El usuario ingresa cuantos comensales estarán en la mesa. 5. El usuario presiona el botón “Confirmar asignación y abrir mesa”. 6. El sistema coloca como ocupada la mesa. 7. El sistema valida que la mesa no fue ocupada antes. 8. El sistema guarda el estado ocupado de la mesa. 9. El sistema actualiza el estado de las mesas y lo muestra en pantalla. 10. El usuario selecciona una mesa y presiona el botón “Realizar comanda”. 11. El sistema incluye el CU-002 “Realizar comanda”. 12. El usuario selecciona una mesa y presiona el botón “Cerrar Mesa”. 13. El sistema genera una cuenta donde se detallan los productos que fueron solicitados en la comanda, el precio unitario de cada producto, el total y el número de la mesa 14. El usuario selecciona una mesa y presiona el botón “Cobrar mesa”. 15. El sistema incluye el CU-006 “Cobrar y generar factura”. 16. El sistema desocupa la mesa que fue cerrada y cobrada. 17. El sistema muestra el listado de las mesas actualizadas. |
| **Flujo alternativo:**   * 1. No hay mesas disponibles, se le notifica al cliente el tiempo de espera.   5.1. No hay espacios suficientes en la mesa para la cantidad de comensales ingresados.  7.1. La mesa fue ocupada mientras el usuario asignaba la mesa, se vuelve al paso 3.  10.1 El usuario selecciona una mesa y presiona el botón “Visualizar comanda”.  10.1.1 El sistema muestra un listado de productos que pertenecen a la comanda asignada a la mesa.  Del producto detallará su nombre, precio, estado (Pendiente, En preparación, Entregado).  12.1 La mesa no tiene una comanda abierta con productos. Se vuelve al paso 9. |

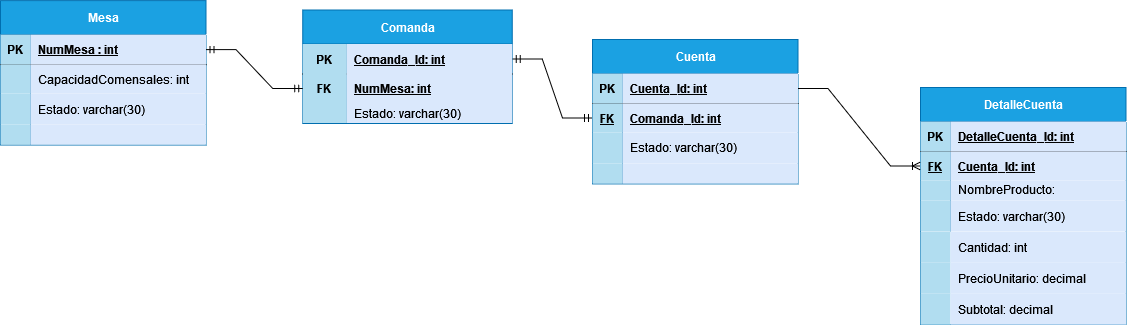
### N02.2.2 Diagrama de secuencia CU 001 “Gestionar mesas”



### N02.2.3 Diagrama de clases CU 001 “Gestionar mesas”



### N02.2.4 Modelo de datos CU 001 “Gestionar mesas”

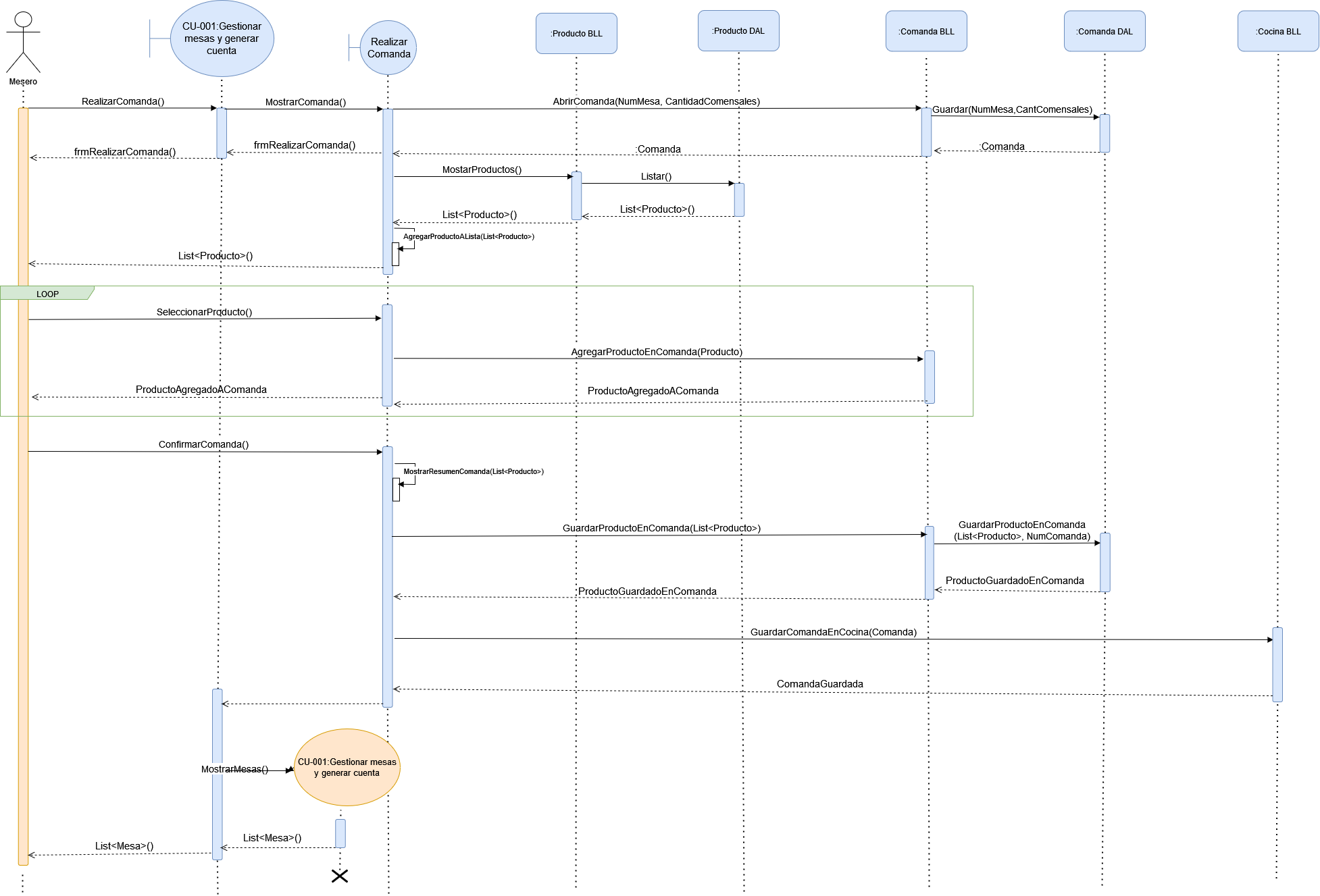


## **N02.3 CU 002 “Realizar comanda”**

### N02.3.1 Especificación CU 002 “Realizar comanda”

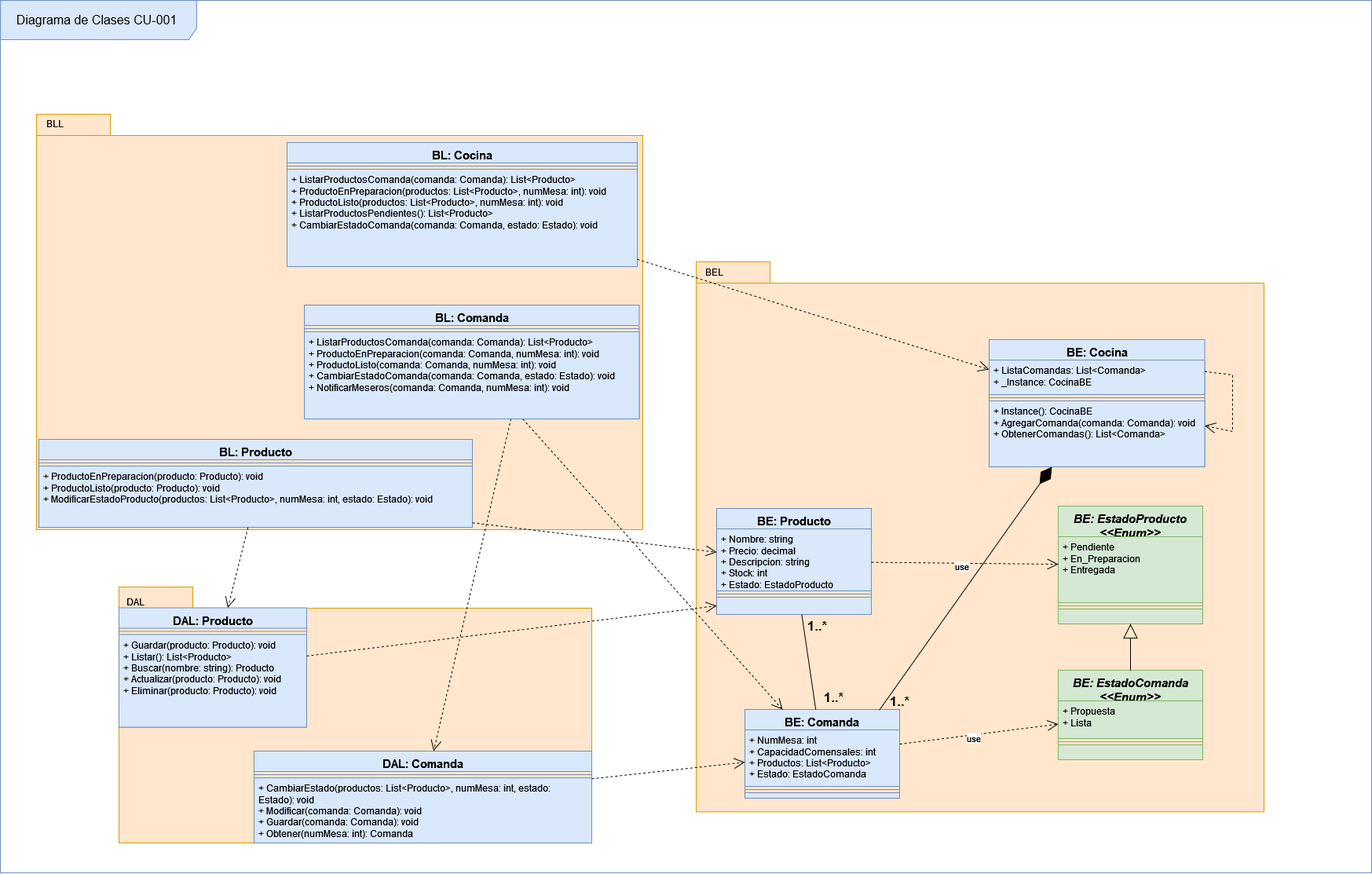
| **Nombre:** CU-002 “Realizar comanda” |
| --- |
| **Objetivo:** El actor añade productos a la comanda para que luego se pueda realizar la carga de la factura |
| **Actor principal:** Mesero |
| **Actor Secundario:** Comensal |
| **Precondiciones:** La mesa debe encontrarse ocupada por el o los comensales. |
| **Disparador:** El mesero selecciona una mesa dentro de la pantalla “Asignar mesa” y selecciona la opción “realizar comanda”. |
| **Postcondiciones:** La mesa tiene una comanda cerrada con los productos solicitados por los comensales. |
| **Flujo principal:**   1. El usuario selecciona una mesa en la pantalla “Gestionar mesas” y presiona el botón “Realizar comanda” 2. El sistema muestra una pantalla con el listado de productos con su código de producto, nombre de producto, descripción y su precio. Además muestra el número de mesa al que pertenece la comanda. 3. El usuario selecciona los productos los cuales se irán listando en la comanda de la mesa. 4. El usuario presiona el botón “Confirmar comanda”. 5. El sistema muestra un resumen de la comanda. 6. El usuario presiona el botón “Aceptar”. 7. El sistema envía a cocina la lista de productos de la comanda. 8. El sistema regresa a la pantalla “Gestionar Mesas”. |
| **Flujo alternativo:**  1.1 La mesa asignada se encuentra desocupada.  1.1.1 Se muestra un aviso “Debe abrir mesa y asignar comensales para realizar una comanda”.  1.1.2 Se repite el paso 1.  6.1 El usuario presiona el botón “Cancelar”.  6.2 Vuelta al paso 2. |

### N02.3.2 Diagrama de secuencia CU 002 “Realizar comanda”

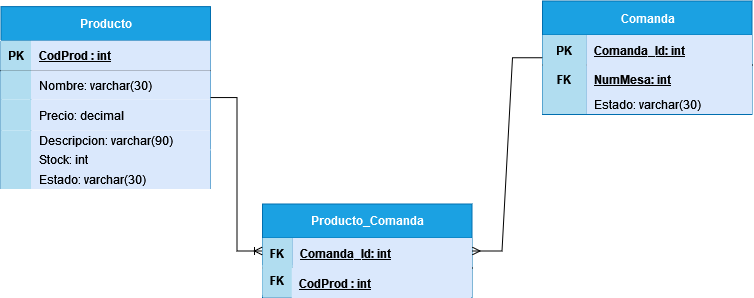


### 

### N02.3.3 Diagrama de clases CU 002 “Realizar comanda”



### N02.3.4 Modelo de datos CU 002 “Realizar comanda”

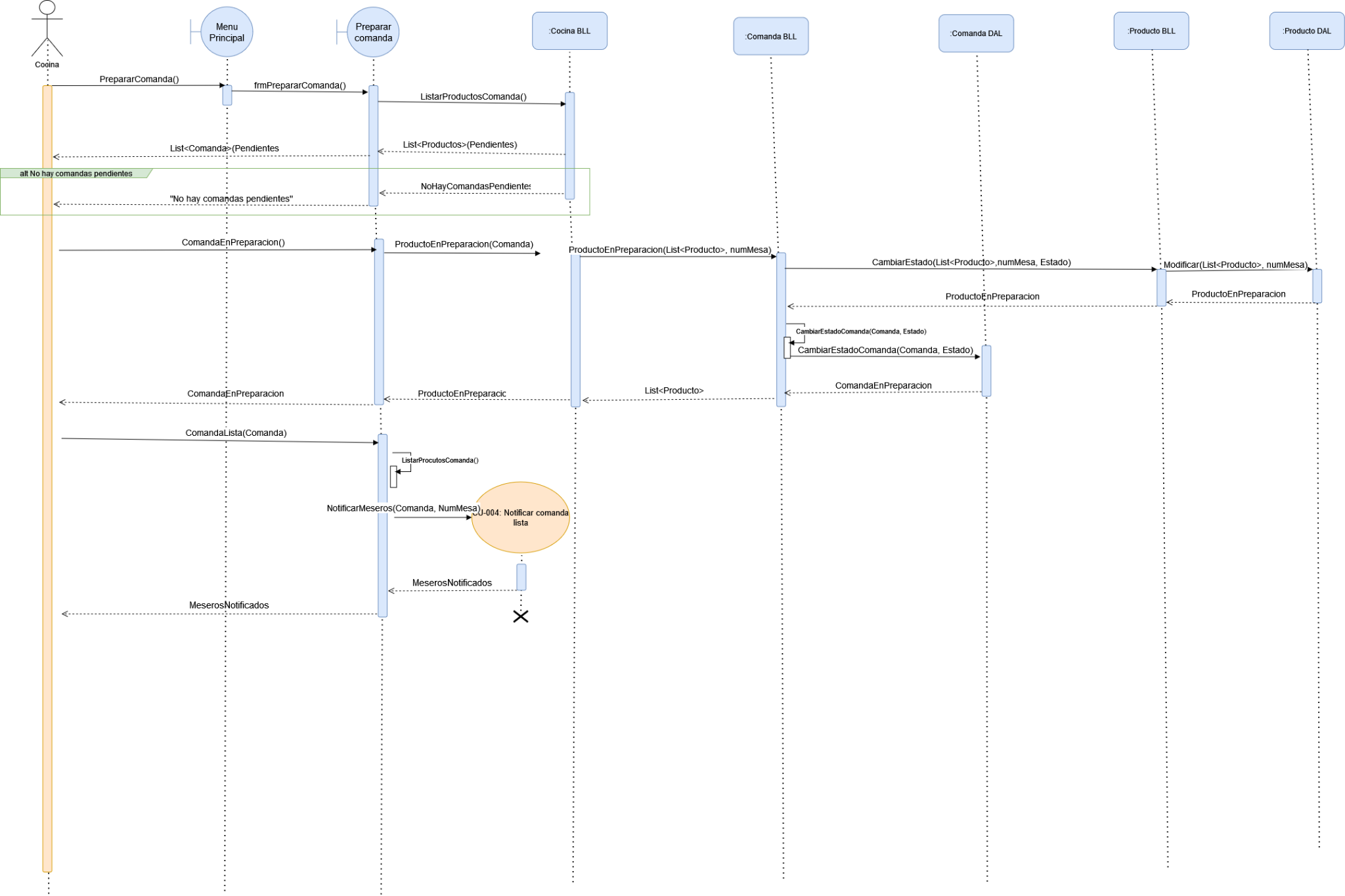


## **N02.4 CU 003 “Preparar comanda”**

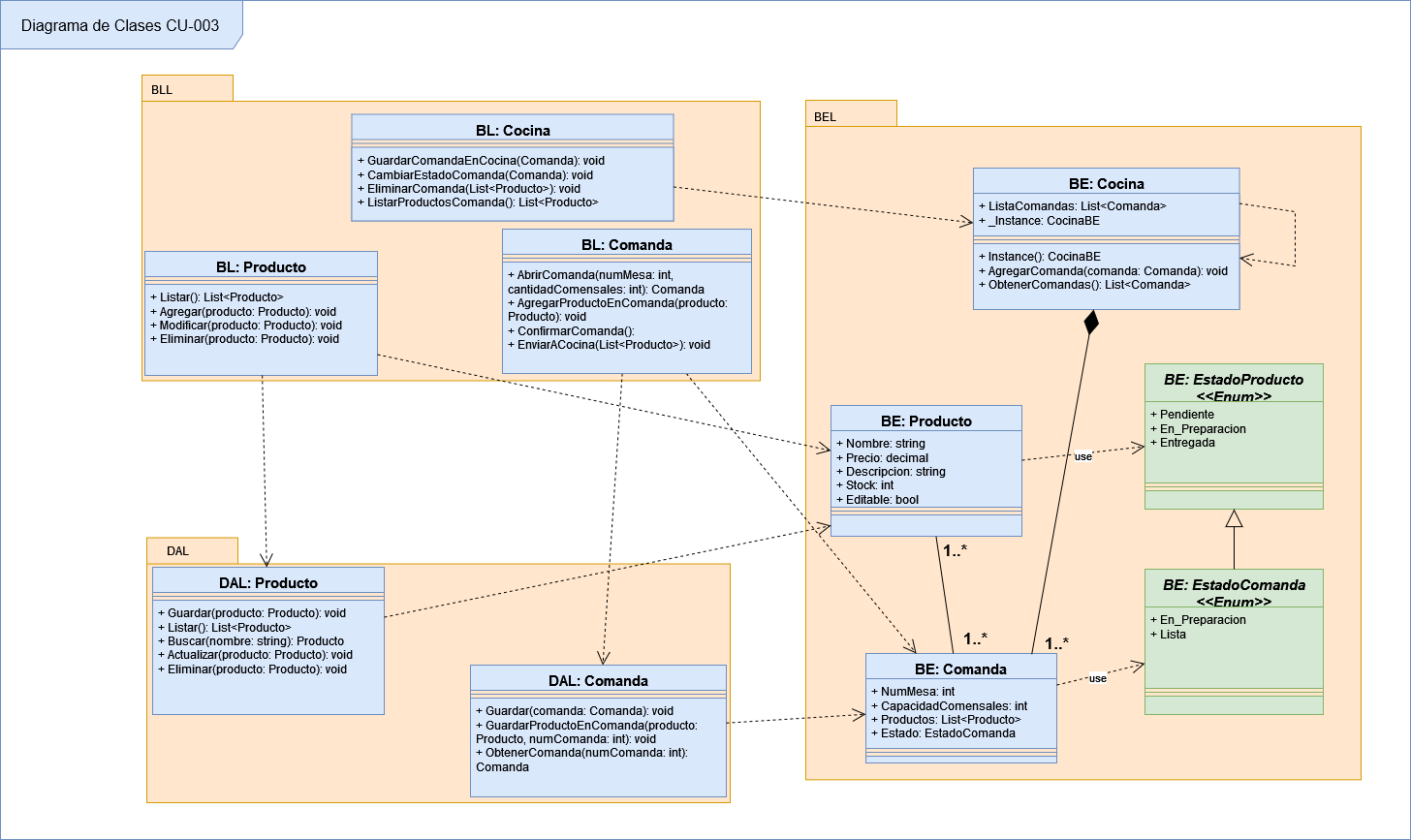
### N02.4.1 Especificación CU 003 “Preparar comanda”

| **Nombre:** CU 003 “Preparar comanda” |
| --- |
| **Objetivo:** El actor visualiza los productos no preparados de las comandas, las marca como entregado y notifica a los meseros. |
| **Actor principal:** Cocina |
| **Actor Secundario:** Mesero |
| **Precondiciones:** Se debe haber realizado una comanda. |
| **Disparador:** En cocina se loguea e ingresa a “Preparar Comanda” |
| **Postcondiciones:** Los meseros serán notificados de que una comanda se encuentra lista. |
| **Flujo principal:**   1. El usuario ingresa a la pantalla “Preparar Comanda”. 2. El sistema muestra un listado de las distintas comandas con los productos pendientes. 3. El usuario selecciona una comanda y la marca como “En Preparación”. 4. El sistema actualiza el estado de la comanda y de los productos que esten pendientes y lo muestra. 5. El usuario selecciona una comanda y la marca como “Lista”. 6. El sistema Incluye el CU-004 “Notificar Comanda Lista”. 7. El sistema vuelve a listar las comandas y los productos que se encuentren en preparación, con su nuevo estado. |
| **Flujo alternativo:**  2.1 No hay comandas con productos pendientes, se muestra un cartel “No hay comandas pendientes en  este momento”.  3.1 El usuario selecciona una comanda y presiona el botón “Cancelar productos de esta comanda”.  3.2 El sistema elimina los productos de la comanda que fue seleccionada.  3.2 El sistema vuelve a listar las comandas con su nuevo estado. |

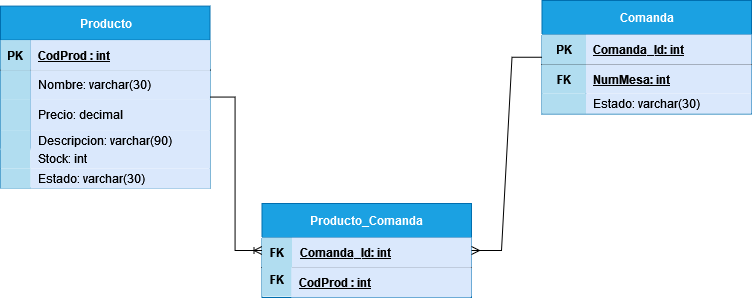
### N02.4.2 Diagrama de secuencia CU 003 “Preparar comanda”



### N02.4.3 Diagrama de clases CU 003 “Preparar comanda”



### N02.4.4 Modelo de datos CU 003 “Preparar comanda”

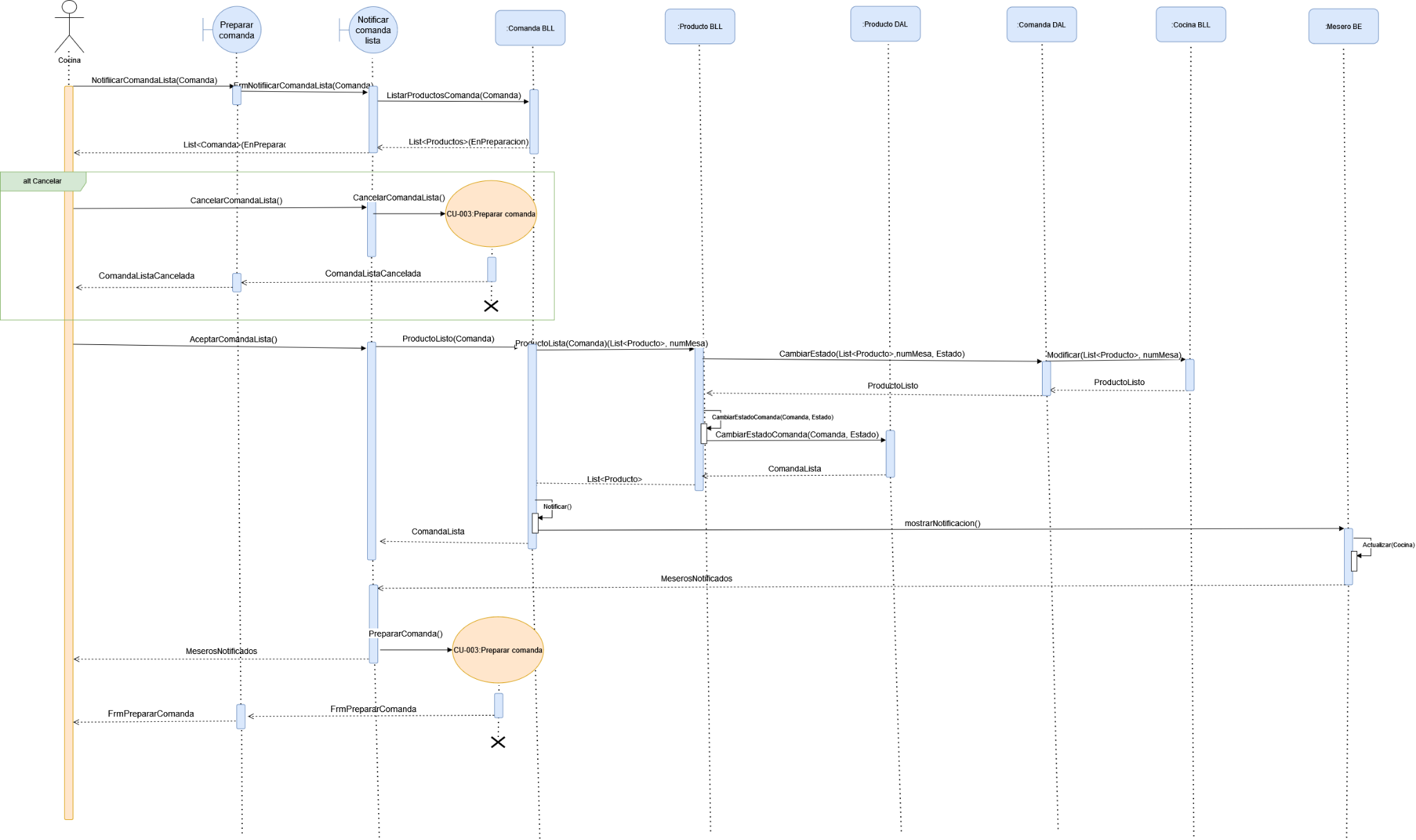


## **N02.5 CU-004: Notificar comanda lista**

### N02.5.1 Especificación CU-004: “Notificar comanda lista”

| **Nombre:** CU-004: Notificar comanda lista |
| --- |
| **Objetivo:** Notificar a los meseros de que una comanda se encuentra lista para su entrega |
| **Actor principal:** Cocina |
| **Actor Secundario:** Mesero |
| **Precondiciones:** La comanda se encuentra lista |
| **Disparador:** El actor presiona el botón “Notificar comanda lista”. |
| **Postcondiciones:** Los meseros se encuentran notificados de la comanda lista. |
| **Flujo principal:**   1. El usuario presiona el botón “Notificar comanda lista” y accede a la pantalla. 2. El sistema muestra un aviso con la comanda y su lista de productos que serán notificados como listos. 3. El usuario presiona el botón “Aceptar”. 4. El sistema notifica a los meseros mostrándoles el número de mesa. 5. El sistema retorna a la pantalla “Preparar comanda”. |
| **Flujo alternativo:**  3.1 El usuario presiona el botón “Cancelar”.  3.1.1 Se retorna al paso 5. |

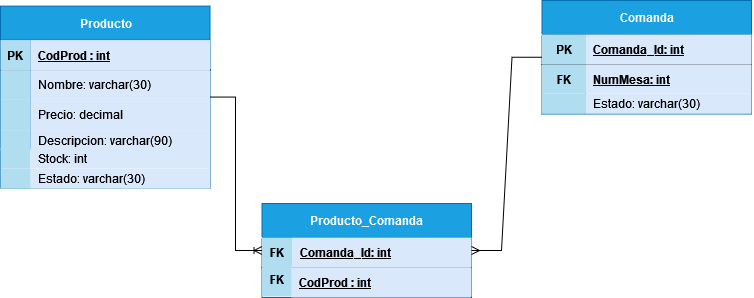
### N02.5.2 Diagrama de secuencia CU-004: “Notificar comanda lista”



### N02.5.3 Diagrama de clases CU-004: “Notificar comanda lista”

### 

### N02.5.4 Modelo de datos CU-004: “Notificar comanda lista”

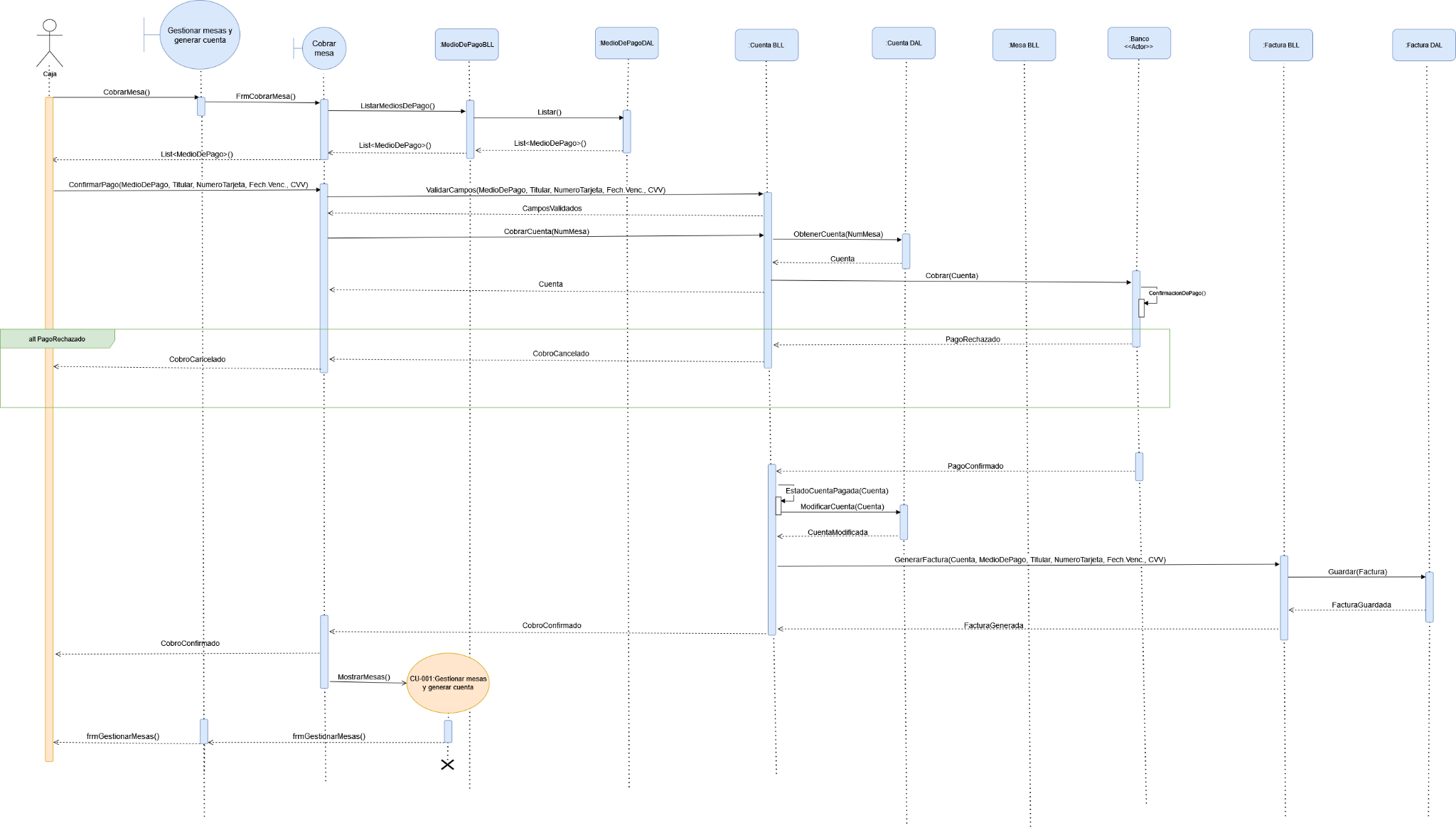


## **N02.6 CU-005: ”Cobrar y generar factura”**

### N02.6.1 Especificación CU-005: ”Cobrar y generar factura”

| **Nombre:** CU-005: ”Cobrar y generar factura” |
| --- |
| **Objetivo:** El mesero realiza el cobro de la cuenta y genera una factura. |
| **Actor principal:** Caja |
| **Actor Secundario:** Mesero |
| **Precondiciones:** El mesero generó una cuenta y presiona el botón “Cobrar mesa”. |
| **Disparador:** El actor presiona el botón “Cobrar mesa”. |
| **Postcondiciones:** El cuenta se encuentra pagada y la factura generada |
| **Flujo principal:**   1. El usuario selecciona una mesa y presiona el botón “Cobrar mesa”. 2. El sistema muestra un formulario con los datos necesarios para realizar el cobro. Se detalla método de pago, titular de la tarjeta, número de la tarjeta, fecha de vencimiento, CVV. Se visualiza el subtotal y monto total. 3. El usuario selecciona el método de pago. 4. El usuario ingresa datos de pago. 5. El usuario presiona el botón “Confirmar pago”. 6. El sistema valida formato de campos. 7. El usuario toca el botón de confirmar. 8. El sistema valida el pago. 9. El sistema genera la factura correspondiente. 10. El sistema vuelve a la pantalla “Gestionar mesas y generar cuenta”. |
| **Flujo alternativo:**  3.1 El usuario presiona el botón “Cancelar”. Se retorna al paso 5.  4.1 En caso de ser efectivo, se ingresan sólo el nombre, apellido, DNI y correo electrónico de la persona.  Saltar al paso 5.  5.1 El formato es incorrecto. Vuelta al paso 4.  7.1 Se rechazó el medio de pago. Vuelta al paso 2. |

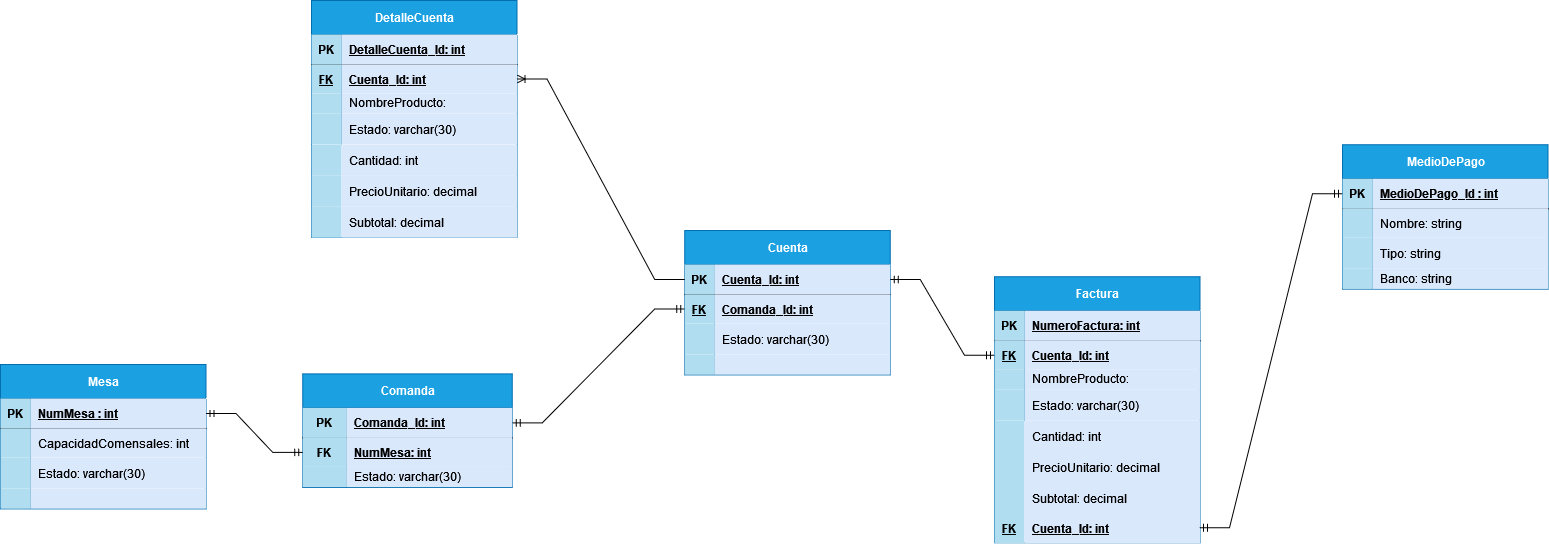
### N02.6.2 Diagrama de secuencia CU-005: ”Cobrar y generar factura”



### N02.6.3 Diagrama de clases CU-005: ”Cobrar y generar factura”

### 

### N02.6.4 Modelo de datos CU-005: ”Cobrar y generar factura”



# N00.1 Procesos de negocio 1 (Entrega 2)

# N00.2 Procesos de negocio 1 (Entrega 3)

# T01. Arquitectura Base

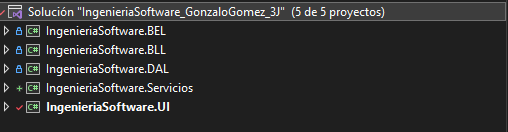
## T01.1 Diseño de la arquitectura (Diagrama de capas)

#### 

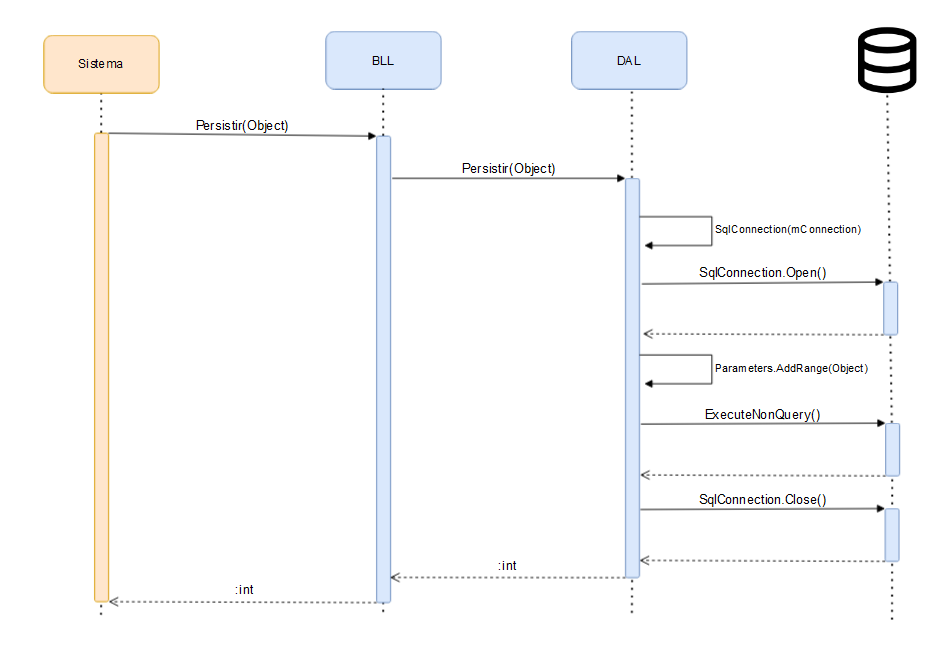
#### 

#### 

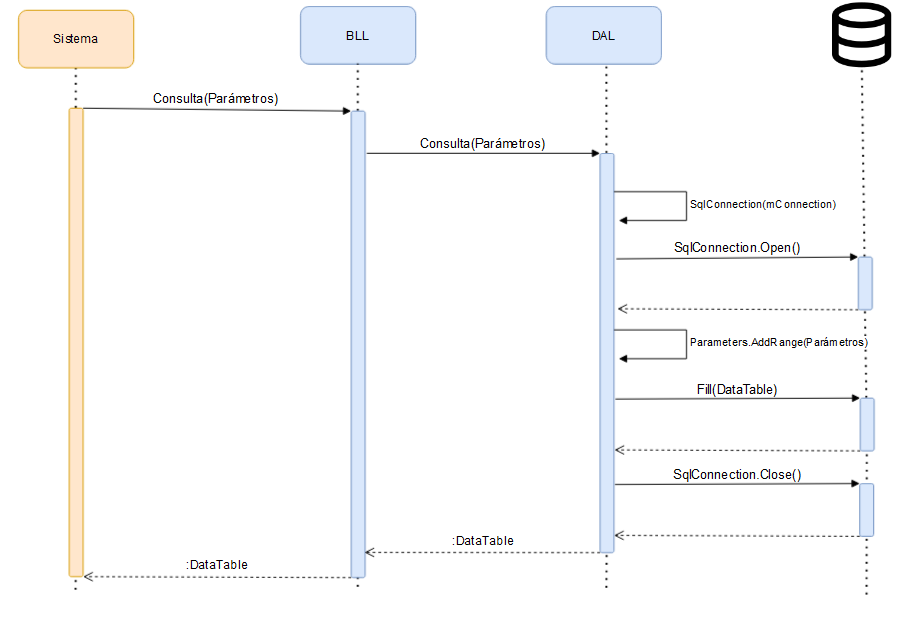
## T01.2 - Captura de pantalla de las capas creadas en el IDE



## T01.3 - Esquema de Persistencia de Datos



## T01.3 - Esquema de Consulta de Datos



### 

### 

# T02. Gestión de Log In / Log Out del Sistema

## T02.1. CU 006 Gestionar usuario “Crear usuario”

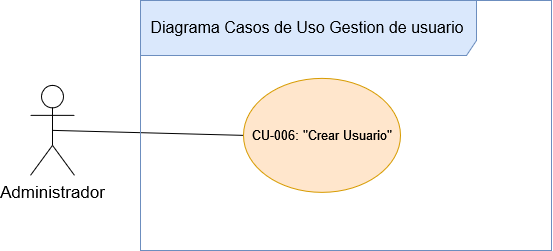
### **T02.1.1 Descomposición Funcional: Crear Usuario**

1. El administrador ingresa al “Menú principal”, va a la sección “Admin” y presiona el botón “Gestionar usuarios”.
2. El administrador selecciona "Gestión Usuario" en el submenú de Administrador.
3. El sistema muestra una lista de usuarios y opciones disponibles.
4. El administrador presiona el botón "Agregar".
5. El sistema presenta un formulario para ingresar los datos del nuevo usuario.
6. El administrador completa los campos requeridos.
7. El sistema valida los datos ingresados.
8. Si son válidos, se crea el usuario:

* Se genera automáticamente la contraseña (DNI + Apellido).
* Se genera automáticamente el nombre de usuario (Nombre + Apellido).

1. El sistema guarda el nuevo usuario en la base de datos.
2. El sistema muestra un mensaje de confirmación al administrador.

### **T02.1.2 Diagrama de caso de uso CU 006 “Crear usuario”**

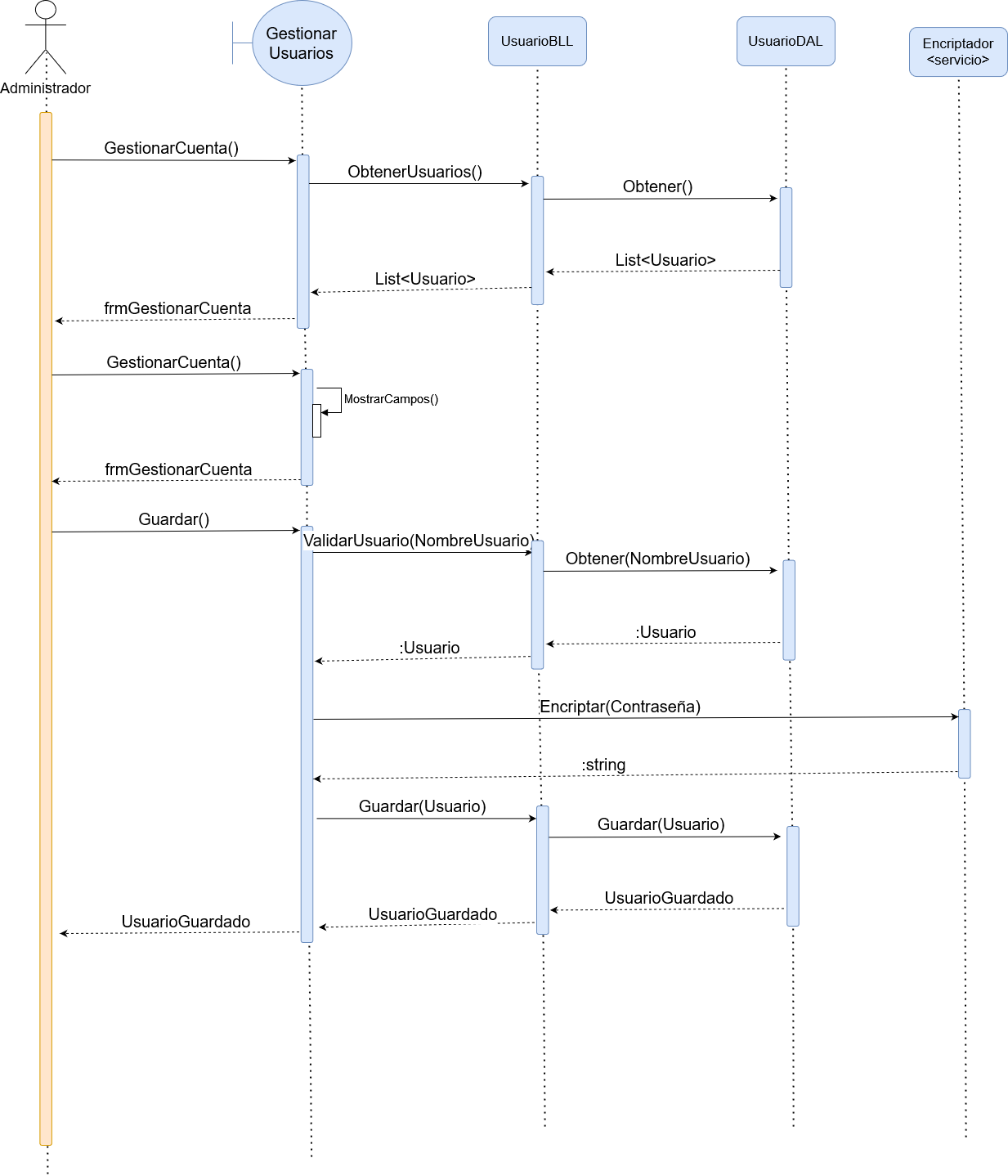


### **T02.1.3 Especificación CU 006 “Crear usuario”**

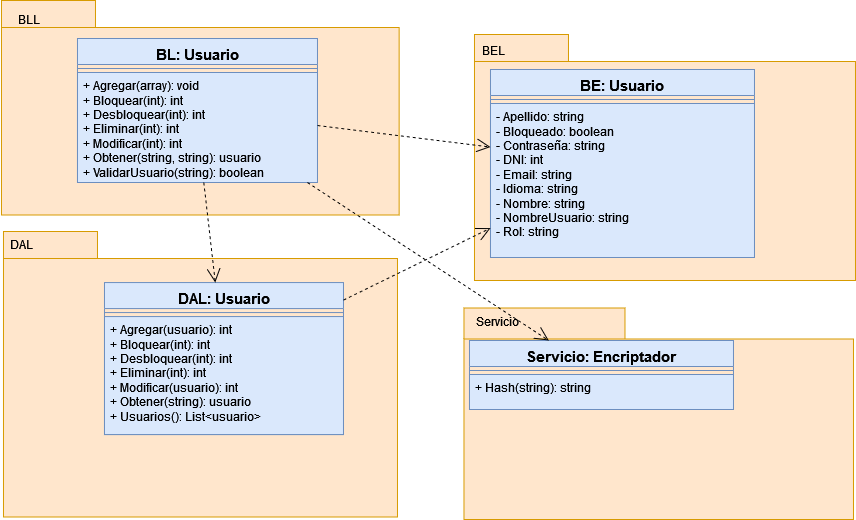
| **Nombre:** CU06 “Crear usuario” |
| --- |
| **Objetivo:** Crear una cuenta nueva de usuario en el sistema |
| **Actor principal:** Administrador |
| **Actor Secundario:** - |
| **Precondiciones:** El actor debe estar logueado en el sistema como administrador |
| **Disparador:** El administrador presiona el botón “Gestion de usuarios” |
| **Postcondiciones:** El administrador crea una nueva cuenta de usuario. |
| **Flujo principal:**   1. El administrador presiona el botón “Gestion de usuarios” en la sección “Admin”dentro del “Menú principal” 2. El sistema abre la pantalla de “Gestion de usuarios” donde se pueden visualizar todos los usuarios del sistema. 3. El administrador presiona el botón “Agregar”. 4. El sistema nos muestra una pantalla con los diferentes campos a agregar: Nombre, Apellido, DNI, Email, Rol y NombreUsuario. 5. El administrador completa los datos y presiona el botón “Aplicar”. 6. El sistema valida que el usuario no exista previamente. 7. El sistema crea la contraseña concatenando el nombre con el DNI del usuario. 8. El sistema encripta la contraseña y crea una nuevo registro en la tabla de usuarios de la base de datos. 9. El sistema actualiza la grilla y el administrador visualiza el nuevo usuario creado. |
| **Flujo alternativo:**  2.1 El administrador no va a continuar con la operación, por lo que presiona el botón “X” cerrando la pantalla.  2.1.1 El sistema muestra el menú principal.  6.1 El usuario ya existe en la base de datos por lo que el sistema le informa al administrador.  6.1.1 Se retorna al punto 2. |

### 

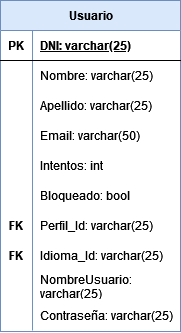
### **T02.1.4 Diagrama de secuencia CU 006 “Crear usuario”**



### **T02.1.4 Diagrama de clases CU 006 “Crear usuario”**



### **T02.1.4 Modelo de datos CU 006 “Crear usuario”**

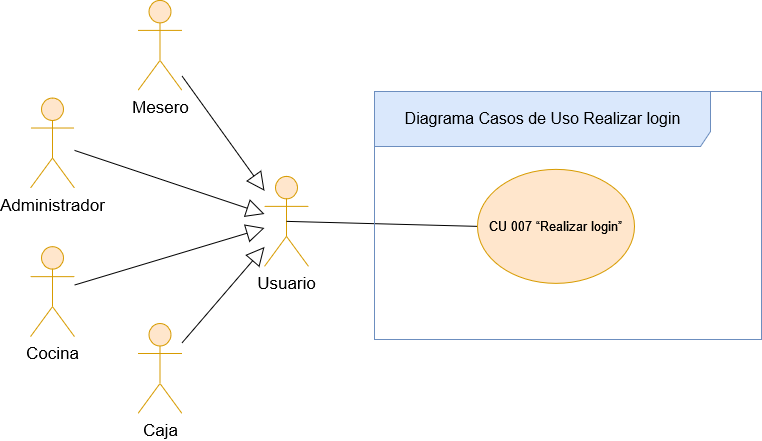


## T02.2. CU 007 “Realizar login”

### T02.2.1 Descomposición Funcional: “Realizar login”

1. El usuario inicia el programa.
2. El sistema muestra la pantalla “Login”.
3. El usuario ingresa los datos de Usuario y Contraseña.
4. El usuario presiona el botón “Ingresar”.
5. El sistema encripta la contraseña.
6. El sistema valida que el usuario exista, en caso de que muestra un mensaje de usuario inexistente.
7. Si el usuario existe el sistema verifica que no esté inactivo ni bloqueado.
8. Si el usuario existe el sistema valida que la contraseña coincida. En caso de que no coincida se le suma un intento, al llegar a los tres intentos queda bloqueado.
9. En caso de que la contraseña coincida el sistema inicia la sesión con el usuario.
10. El sistema muestra la pantalla “menú principal” con las funciones asociadas al rol de esa cuenta.

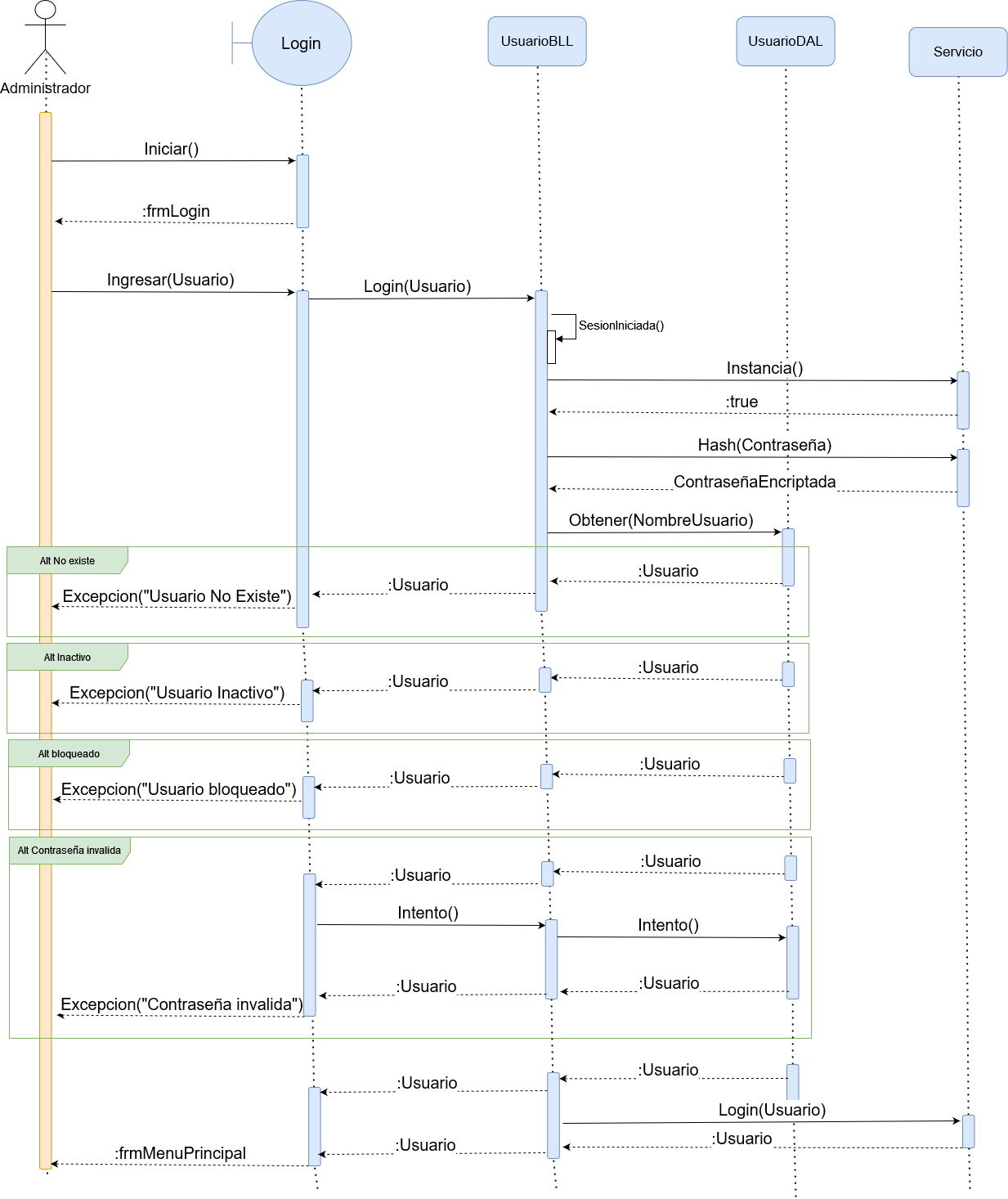
### **T02.2.2 Diagrama de caso de uso CU 007 “Realizar login”**



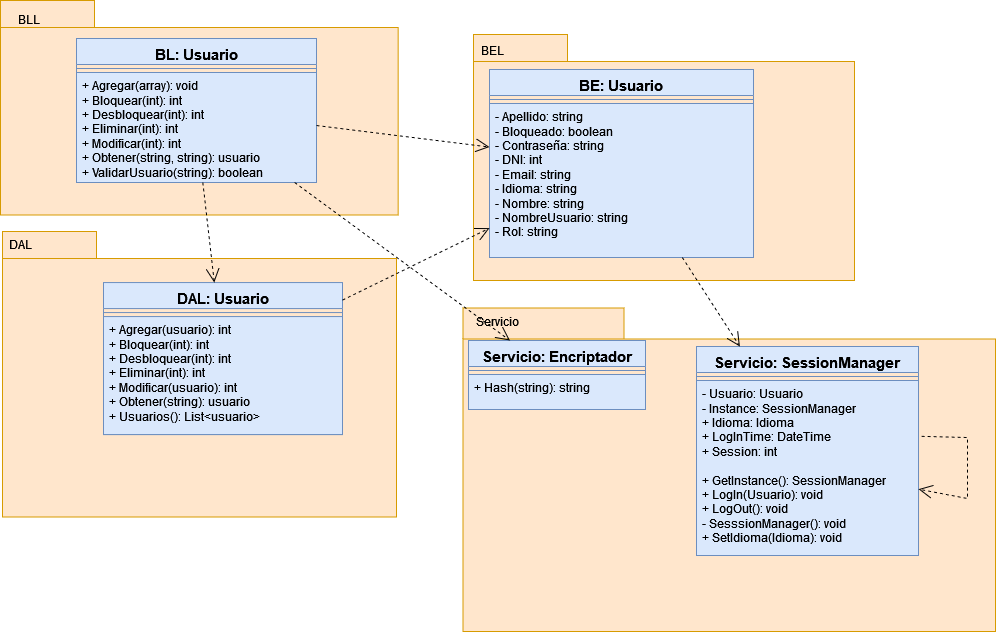
### **T02.2.3 Especificación CU 007 “Realizar login”**

| **Nombre:** CU 007 “Realizar login” |
| --- |
| **Objetivo:** El usuario se ingresa en el sistema |
| **Actor principal:** Administrador, Mesero, Caja, Cocina |
| **Actor Secundario:** - |
| **Precondiciones:** El usuario debe estar registrado y no estar bloqueado. |
| **Disparador:** El administrador presiona el botón “Gestion de usuarios” |
| **Postcondiciones:** El administrador crea una nueva cuenta de usuario. |
| **Flujo principal:**   1. El usuario inicia el programa. 2. El sistema abre la pantalla de “Inicio de sesión” donde se solicitan los datos de Usuario y Contraseña. 3. El usuario ingresa los campos y presiona el botón “Ingresar”. 4. El sistema verifica los datos del usuario. 5. El sistema muestra el menú principal con las funciones asociadas al perfil del usuario. |
| **Flujo alternativo:**  4.1 El sistema detecta que el usuario se encuentra bloqueado y muestra un mensaje en pantalla. Se retorna  al punto 2.  4.2 El sistema detecta que el usuario no existe. Se retorna al punto 2.  4.3 El sistema detecta que el usuario ingresó incorrectamente la contraseña por lo que se procede a sumar  un intento al usuario.  4.4 El sistema detecta que el usuario está inactivo, se muestra la excepción en pantalla y se vuelve al paso 2 |

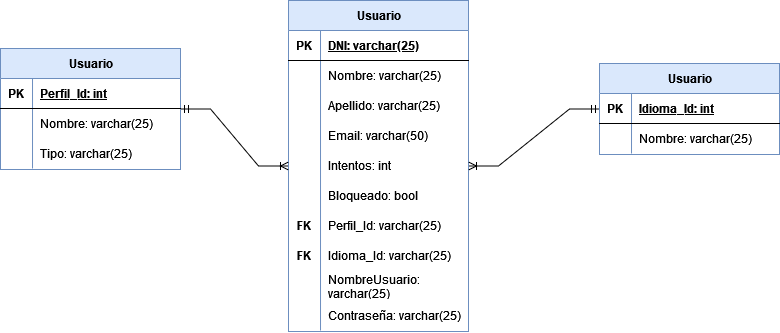
### **T02.2.4 Diagrama de secuencia CU 007 “Realizar login”**



### **T02.2.4 Diagrama de clases CU 007 “Realizar login”**



### **T02.2.4 Modelo de datos CU 007 “Realizar login”**

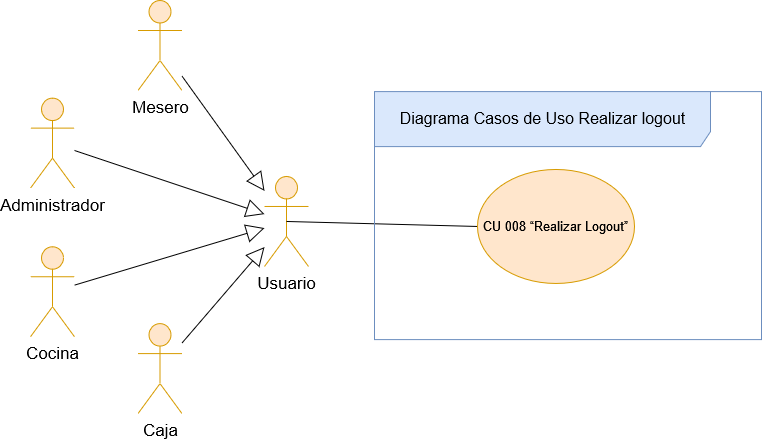


## T02.3. CU 008 “Realizar Logout”

### T02.3.1 Descomposición Funcional: “Realizar logout”

1. El usuario presiona el botón “Salir” dentro de la sección “Usuario” del menú principal.
2. El sistema le muestra un mensaje para verificar que quiere salir.
3. El usuario confirma su salida.
4. El sistema nos muestra la pantalla de “Login”.

### **T02.3.2 Diagrama de caso de uso CU 008 “Realizar Logout”**

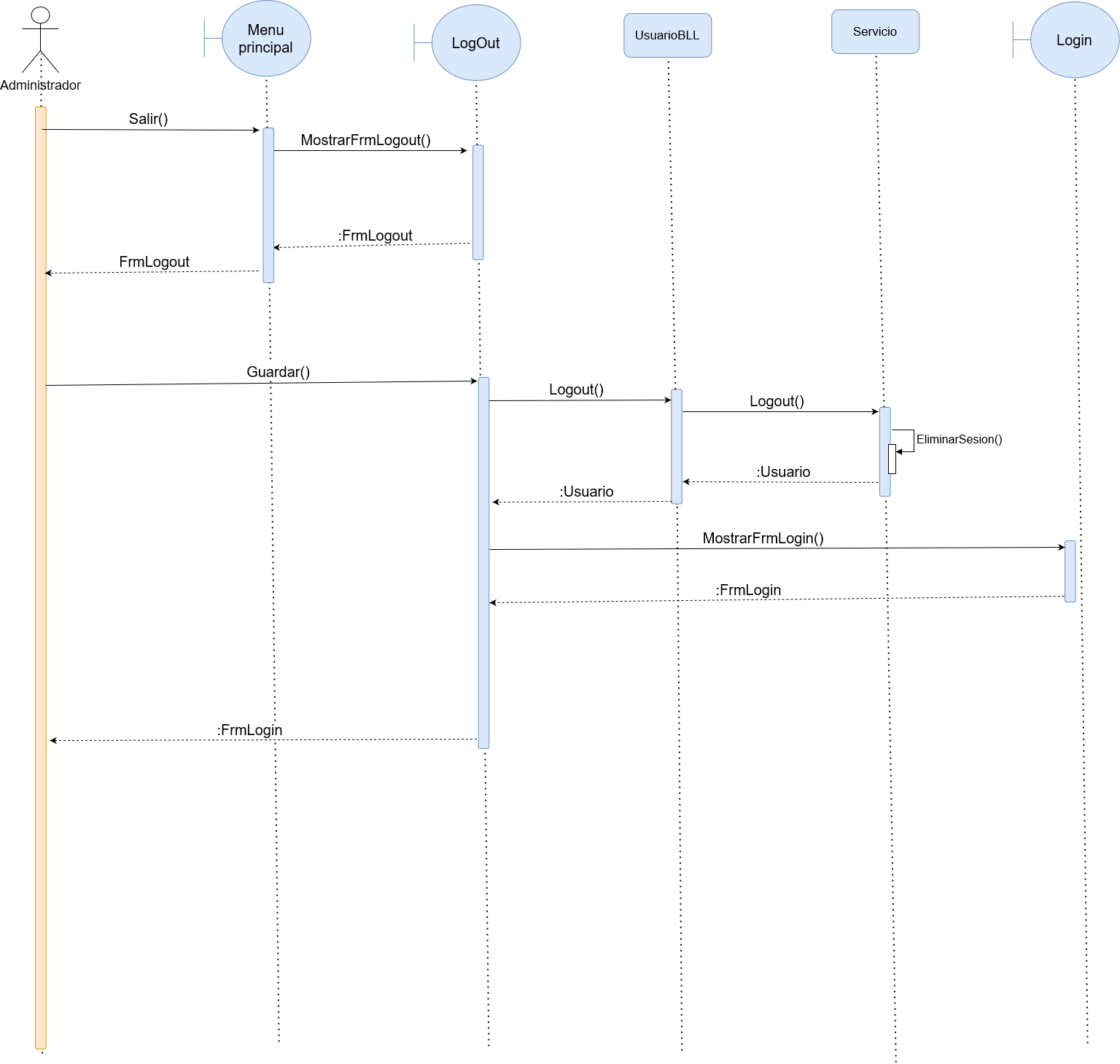


### **T02.3.3 Especificación CU 008 “Realizar Logout”**

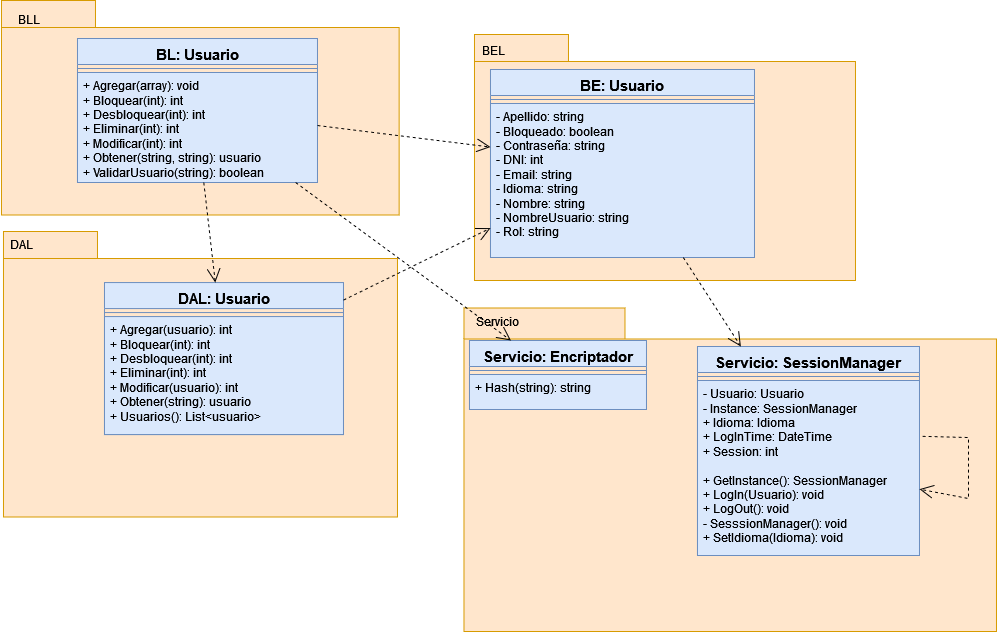
### 

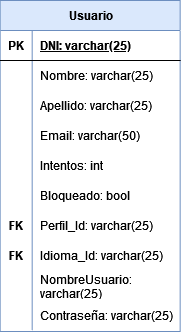
| **Nombre:** CU 008 “Realizar Logout” |
| --- |
| **Objetivo:** El usuario realiza sale de la sesión correctamente |
| **Actor principal:** Administrador, Mesero, Caja, Cocina |
| **Actor Secundario:** - |
| **Precondiciones:** El usuario debe estar logueado en el sistema |
| **Disparador:** El usuario presiona el botón “Salir” de la sección “Usuario”. |
| **Postcondiciones:** El usuario salio del sistema y este muestra la pantalla de “Iniciar Sesión” |
| **Flujo principal:**   1. El usuario selecciona el botón “Salir” en la sección de “Usuario” dentro del menú principal. 2. El sistema muestra un mensaje para confirmar la salida. 3. El usuario presiona “Aceptar”. 4. El sistema cierra la sesion y abre la pantalla de “Iniciar sesión” |
| **Flujo alternativo:**  3.1 El usuario se arrepiente y presiona el botón “Cancelar”.  3.1.1 El sistema vuelve al menú principal. |

### **T02.3.4 Diagrama de secuencia CU 008 “Realizar Logout”**



### **T02.3.4 Diagrama de clases CU 008 “Realizar Logout”**

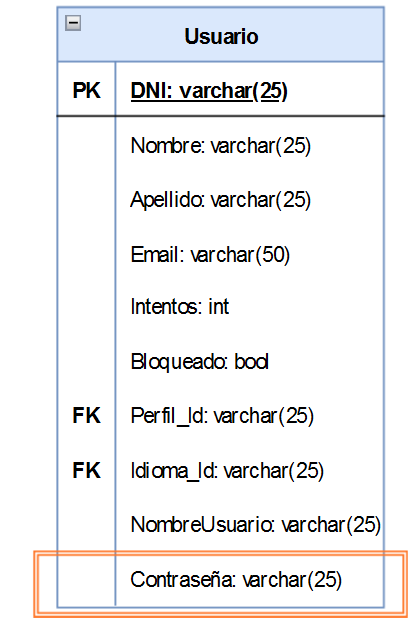
**T02.3.4 Modelo de datos CU 008 “Realizar Logout”**



# T03. Gestión de Encriptado

## T03.1 Criptografía irreversible

Dentro del sistema, se utilizará el algoritmo de hash criptográfico SHA-256 (Secure Hash Algorithm 256-bit) para el almacenamiento seguro de las contraseñas de usuario.



Este enfoque se basa en principios de seguridad sólidos y se justifica por varias razones fundamentales:

1. Seguridad Robusta:

SHA-256 es un algoritmo de hash criptográfico ampliamente reconocido por su seguridad robusta. Genera un resumen de mensaje de 256 bits que es extremadamente difícil de revertir o descifrar, incluso mediante métodos avanzados de criptoanálisis. Esta característica garantiza un alto nivel de protección para las contraseñas de nuestros usuarios contra intentos de intrusión y acceso no autorizado.

2. Resistencia a Ataques Conocidos:

SHA-256 ha resistido con éxito numerosos intentos de ataques criptográficos durante muchos años desde su introducción. Aunque ningún algoritmo de seguridad es completamente inmune a futuros avances tecnológicos, hasta la fecha no se ha encontrado ningún método efectivo para comprometer la integridad del algoritmo SHA-256.

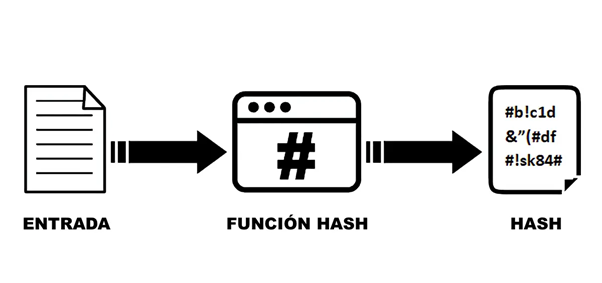
3. Amplia Adopción y Soporte:

SHA-256 es ampliamente utilizado en una variedad de aplicaciones de seguridad y criptografía, lo que garantiza su interoperabilidad y compatibilidad con otros sistemas y herramientas. Su estatus como un estándar de la industria respaldado por organizaciones como el NIST (Instituto Nacional de Normas y Tecnología) y la NSA (Agencia de Seguridad Nacional) aumenta aún más su credibilidad y fiabilidad.

4. Cumplimiento Normativo:

El uso de SHA-256 para el almacenamiento de contraseñas cumple con muchos estándares y regulaciones de seguridad de la información, incluidos aquellos establecidos por organizaciones gubernamentales y de la industria. Esto asegura que nuestras prácticas de seguridad estén alineadas con las mejores prácticas y directrices reconocidas a nivel internacional.

La elección de utilizar SHA-256 para el almacenamiento de contraseñas de usuario se basa en su sólida reputación de seguridad, resistencia comprobada y amplia adopción en la industria. Esto nos permite garantizar la protección confiable de las credenciales de nuestros usuarios y mantener la integridad y confidencialidad de sus datos en nuestro sistema.



## **T03.2 Criptografía reversible**

Criptografía AES (Advanced Encryption Standard)

AES (Advanced Encryption Standard) es uno de los algoritmos de cifrado simétrico más ampliamente utilizados y seguros en la actualidad. Fue establecido como un estándar por el Instituto Nacional de Estándares y Tecnología (NIST) de Estados Unidos en 2001 y es utilizado para proteger datos confidenciales en una variedad de aplicaciones.

Este tipo de criptografía se usará para el cliente

Características de AES

* Algoritmo Simétrico: Utiliza la misma clave para cifrar y descifrar la información.
* Tamaños de Clave: AES soporta tres tamaños de clave: 128 bits, 192 bits y 256 bits. Cuanto más larga es la clave, mayor es el nivel de seguridad.
* Bloques de Datos: Opera en bloques de datos de 128 bits.
* Rondas de Cifrado: El número de rondas de cifrado depende del tamaño de la clave:
  + AES-128: 10 rondas.
  + AES-192: 12 rondas.
  + AES-256: 14 rondas

Ventajas de AES

* Alta Seguridad: AES proporciona un nivel muy alto de seguridad y es resistente a todos los ataques conocidos hasta la fecha.
* Eficiencia: Es muy eficiente en términos de velocidad tanto en software como en hardware.
* Versatilidad: Su capacidad para soportar diferentes tamaños de clave lo hace adaptable a diversas necesidades de seguridad.

# T04. Gestión de Perfiles de Usuario (Entrega 2)

## T04.a Alta

**Objetivo**

El objetivo de la gestión de alta de usuarios es permitir la creación segura y eficaz de nuevas cuentas de usuario dentro del sistema. Este proceso no solo incluye la captura y el almacenamiento de datos del usuario, sino que también asegura que estos datos sean validados y cifrados adecuadamente para mantener la seguridad y la integridad del sistema.

#### **Descripción detallada de cómo funciona**

El proceso de alta de usuarios comienza cuando un usuario potencial llena un formulario proporcionando su correo electrónico y contraseña.

*1. Validación de entrada:*

Inicialmente, el sistema verifica que el correo electrónico y la contraseña cumplen con los criterios especificados, como el formato correcto del correo electrónico. Esta validación ayuda a prevenir errores comunes y aumenta la seguridad del usuario.

*2. Encriptación de Contraseña:*

Utilizando la clase Seguridad, la contraseña ingresada por el usuario se encripta utilizando el algoritmo SHA-256. Este paso es crucial para asegurar que las contraseñas almacenadas en la base de datos no sean legibles ni accesibles en su forma original.

*3. Verificación de Unicidad:*

Antes de proceder con el almacenamiento, el sistema, a través de la capa de servicio, verifica que no existe ya un usuario con el mismo correo electrónico. Esta comprobación asegura la unicidad de las cuentas y evita duplicaciones.

*4. Registro en la Base de Datos:*

Si la verificación es exitosa y no se encuentran duplicados, la capa de servicio procede a registrar el nuevo usuario en la base de datos. Este paso incluye pasar las propiedades del usuario, como el correo electrónico y la contraseña cifrada, a la función correspondiente de la capa de acceso a datos ( DAL), que finalmente guarda los datos en la base de datos.

*5. Confirmación:*

Una vez completado el registro, el sistema devuelve una confirmación al usuario indicando que la cuenta ha sido creada exitosamente. Además, se registra este evento en el sistema de bitácora a través de BitacoraBLL para auditorías y seguimiento de actividades.

## T04b.Baja

#### **Objetivo**

El objetivo de este proceso es permitir la eliminación segura y efectiva de los usuarios del sistema. Esto incluye asegurar que solo los usuarios que deben ser eliminados, y bajo las condiciones correctas, sean eliminados de la base de datos.

#### **Descripción detallada de cómo funciona**

El proceso de baja de usuarios en nuestro sistema es manejado cuidadosamente para garantizar la integridad y la seguridad de los datos del usuario. Este proceso implica varios pasos críticos para asegurar que la eliminación sea precisa y segura:

*1. Selección del Usuario:*

El proceso comienza cuando un administrador o usuario con permisos adecuados selecciona el usuario que desea eliminar del sistema. Esto generalmente se realiza a través de una interfaz gráfica donde se pueden listar y seleccionar los usuarios registrados.

*2. Confirmación y Validación:*

Antes de proceder con la eliminación, el sistema solicita una confirmación para asegurarse de que el usuario ha sido seleccionado intencionalmente para la eliminación.

*3. Llamada a UsuarioBLL para la Baja:*

Una vez confirmada la selección y la intención de eliminar, se invoca al método Borrar de la clase UsuarioBLL*.* Este método actúa como un intermediario entre la interfaz de usuario (UI) y las operaciones de bajo a nivel de base de datos (DAL).

*4. Operación de Baja en UsuarioBLL :*

El UsuarioBLL tiene la responsabilidad de la eliminación efectiva, la cual prepara y ejecuta la operación de baja. Esto incluye preparar el comando de SQL para eliminar el usuario de la base de datos, utilizando el identificador único del usuario para evitar errores de eliminación.

*5. Ejecución del Procedimiento Almacenado:*

UsuarioBLL llama al procedimiento almacenado pasando el Id del usuario como parámetro para asegurar que solo el usuario correcto sea eliminado de la base de datos. Si el usuario se encuentra disponible el campo Activo pasa a 0 indicando que se eliminó el registro.

*6. Registro en Bitácora:*

Después de realizar la eliminación, se registra esta acción en el sistema de bitácora mediante BitacoraBLL . Esto incluye registrar el evento de eliminación junto con detalles como el ID del usuario eliminado y la fecha y hora del evento, lo cual es crucial para auditorías de seguridad y cumplimiento normativo.

*7. Confirmación al Usuario:*

Finalmente, se informa al administrador o al usuario con permisos que la eliminación se ha completado exitosamente, proporcionando tranquilidad y cerrando el proceso de eliminación

### T04c. Modificar

#### **Objetivo**

El objetivo principal de la gestión de modificación de usuarios es permitir la actualización segura y eficaz de los datos de usuario dentro del sistema. Este proceso asegura que los cambios se realicen de manera controlada y coherente, manteniendo la integridad de los datos del usuario.

#### **Descripción detallada de cómo funciona**

La modificación de usuarios es manejada cuidadosamente para garantizar la precisión y la seguridad de la actualización de datos.

*1. Selección del Usuario:*

El proceso comienza cuando un administrador o usuario con permisos adecuados selecciona un usuario de la lista de usuarios activos, utilizando la función Listar() para obtener los datos actuales que se muestran en la interfaz de usuario.

*2. Introducción de nuevos datos:*

El usuario o el administrador introduce los nuevos datos que desean actualizar, como el correo electrónico y contraseña. El sistema puede validar estos nuevos datos para asegurar que cumplen con los estándares requeridos, como formato de correo y complejidad de la contraseña.

*3. Llamada UsuarioBLL para actualizar:*

Con los nuevos datos validados, se llama al método Modificar() de la clase BLL\_USUARIO. Este método es responsable de coordinar la lógica de negocio asociada con la actualización de los datos del usuario.

*4. Operación de actualización en UsuarioBLL:*

UsuarioBLL prepara y ejecuta la operación de actualización en la base de datos. Esto incluye preparar y ejecutar el procedimiento almacenado ActualizarUsuario, pasando los nuevos datos como parámetros.

*5. Ejecución del Procedimiento Almacenado:*

UsuarioBLL ejecuta el procedimiento almacenado Guardar\_Usuario en la base de datos, utilizando los parámetros actualizados para asegurar que solo los datos del usuario específico sean modificados.

*6. Confirmación al Usuario:*

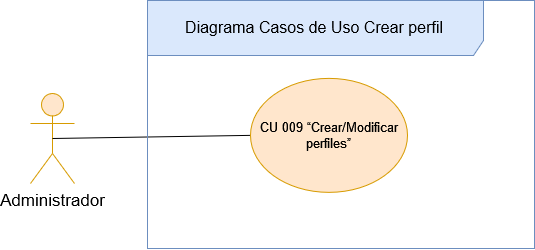
Finalmente, se notifica al usuario o administrador que la modificación se ha completado exitosamente, confirmando que los datos del usuario han sido actualizados en el sistema.

## T04.1 CU 009 “Crear/Modificar perfiles”

### T04.1.1 Descomposición Funcional: CU 009 “Crear/Modificar perfiles”

| **Nombre:** CU 009 “Crear/Modificar perfiles” |
| --- |
| **Objetivo:** El administrador crea un perfil |
| **Actor principal:** Administrador |
| **Actor Secundario:** - |
| **Precondiciones:** El usuario debe estar logueado en el sistema como administrador. |
| **Disparador:** El administrador presiona el botón “Crear usuario” en la sección “Gestión de perfiles”. |
| **Postcondiciones:** El administrador creó un perfil. |
| **Flujo principal:**   1. El administrador accede a la sección “Gestión de perfiles”. 2. El sistema muestra una pantalla con todos los perfiles. 3. El administrador presiona el botón “Crear perfil”. 4. El sistema abre una pantalla que muestra dos listas, una con los permisos y otra con las familias, y una lista en donde se coloca el resumen de los permisos y familias que tiene el perfil que se está creando. 5. El administrador ingresa el nombre del perfil y presiona el botón asignar nombre. 6. El sistema valida que el nombre del perfil no esté repetido. 7. El administrador selecciona y agrega los componentes para ese perfil. 8. El sistema valida que no se repitan componentes. 9. El administrador presiona el botón “Guardar”. |
| **Flujo alternativo:**  6.1 El sistema muestra un mensaje que ingrese otro nombre para el perfil debido a que ya existe.  6.1.1 El sistema muestra un mensaje que el componente que se quiere agregar ya se encuentra entre los  componentes de ese perfil.  8.1 El sistema muestra un mensaje que el componente que se quiere agregar ya se encuentra entre los  componentes de ese perfil. |

### T04.1.2 Diagrama de caso de uso CU 009 “Crear/Modificar perfiles”

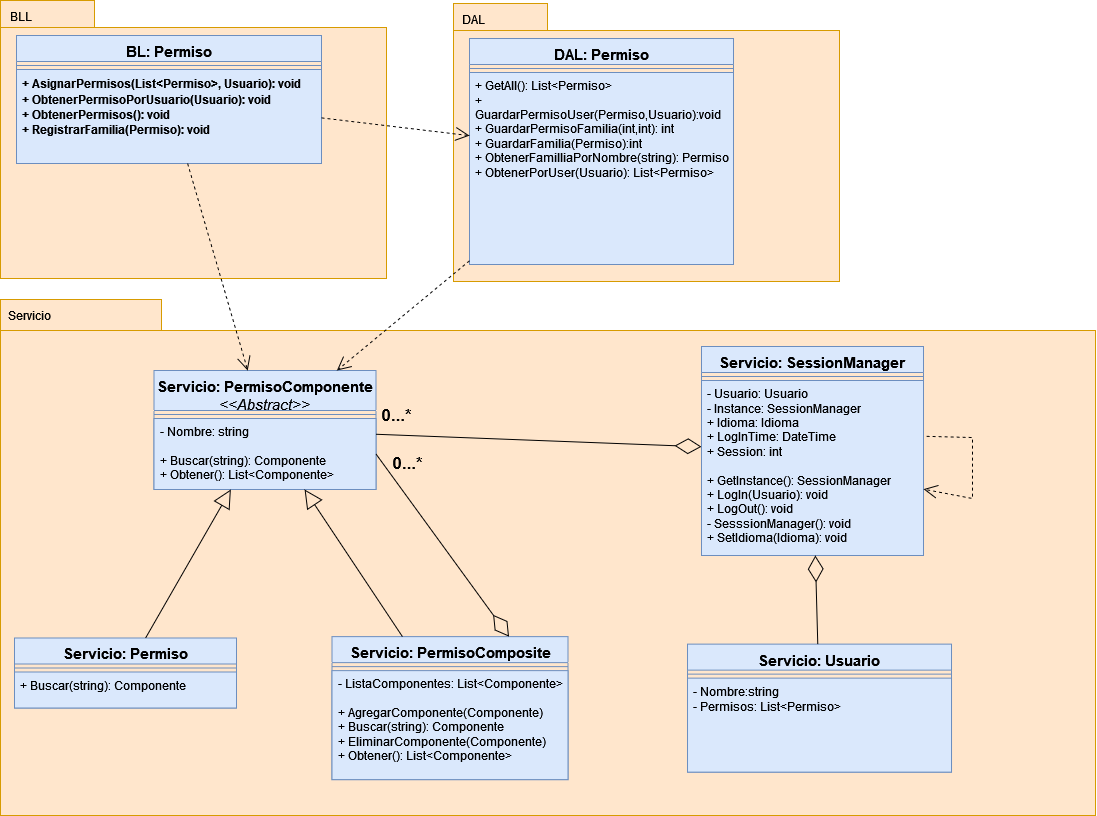


### T04.1.3 Especificación de caso de uso CU 009 “Crear/Modificar perfiles”

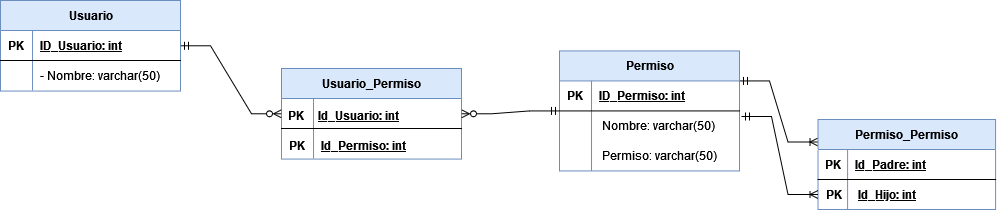
### T04.1.4 Diagrama de secuencia CU 009 “Crear/Modificar perfiles”



### T04.1.5 Diagrama de clases CU 009 “Crear/Modificar perfiles”



### T04.1.6 Modelo de datos CU 009 “Crear/Modificar perfiles”



### 

# T05. Gestión de Múltiples Idiomas (Entrega 2)

### **Objetivo**

El principal objetivo de la gestión de múltiples idiomas en nuestro sistema es proporcionar una interfaz de usuario adaptable a diferentes idiomas, mejorando así la accesibilidad y la experiencia del usuario. Esto permite a los usuarios seleccionar su idioma preferido y ver toda la interfaz en dicho idioma, facilitando su interacción con la aplicación.

### **Descripción detallada del funcionamiento**

La gestión de múltiples idiomas se implementa a través de varios componentes clave dentro del sistema, que incluyen la capa de presentación (UI), la capa de lógica de negocio (BLL) y la capa de acceso a datos (DAL). Aquí se explica cómo se realiza esta gestión paso a paso:

*1. Definición y Almacenamiento de Idiomas y Traducciones:*

Tablas y Procedimientos Almacenados:

- Idiomas almacena los diferentes idiomas disponibles con su respectivo identificador y nombre.

- Palabras almacena todas las palabras y frases que se pueden traducir, junto con un identificador único.

- Traducciones almacena las traducciones de cada palabra para cada idioma, relacionando los identificadores de Idiomas y Palabras con el texto traducido.

Procedimientos Almacenados:

- Guardar\_Traduccion: Este procedimiento se encarga de insertar o actualizar una traducción en la tabla Traducciones. Primero, verifica si la palabra ya existe en la tabla Palabras. Si no existe, la inserta y luego guarda la traducción.

- ListarTraduccionesPorIdioma: Este procedimiento recupera todas las traducciones para un idioma específico, uniendo las tablas Traducciones y Palabras para obtener el texto de la palabra y su traducción.

- Borrar\_Traduccion: Este procedimiento elimina una traducción específica de la tabla Traducciones.

*2. Carga de Idiomas y Traducciones:*

Formulario de Configuración de Idiomas:

En el formulario ConfigurarIdioma, se cargan todos los idiomas disponibles y se muestran en un ComboBox. El usuario puede seleccionar un idioma para visualizar y gestionar sus traducciones.

Las traducciones existentes para el idioma seleccionado se muestran en un DataGridView. Los usuarios pueden agregar, modificar o eliminar traducciones directamente desde esta interfaz.

Al agregar o modificar una traducción, se utiliza la capa de lógica de negocio (TraduccionBLL) para actualizar la base de datos.

*3. Actualización de la Interfaz:*

Patrón Observer:

El patrón Observer se implementa a través de SessionManager, que notifica a todos los formularios observadores cuando se cambia el idioma.

Los formularios registran observadores a SessionManager y actualizan sus textos cuando se selecciona un nuevo idioma. Este mecanismo asegura que cualquier cambio de idioma se propague inmediatamente a todas las interfaces de usuario abiertas.

Selección de Idioma:

Al seleccionar un idioma en el ComboBox de idiomas, se notifica a SessionManager que cambia el idioma actual. SessionManager entonces llama al método Actualizar en todos los observadores registrados.

El método Actualizar recorre todos los controles del formulario y cambia su texto según las traducciones disponibles para el idioma seleccionado, utilizando los tags definidos en cada control para identificar las palabras a traducir.

*4. Implementación adicional:*

Registro en la Bitácora:

Todas las acciones relacionadas con la gestión de idiomas y traducciones se registran en la bitácora del sistema utilizando BitacoraBLL. Esto incluye la creación, modificación y eliminación de traducciones, así como los cambios de idioma. Este registro proporciona una pista de auditoría que puede ser utilizada para revisar cambios y actividades sospechosas.

## T04.1 CU 010 “Crear/Modificar perfiles”

### T04.1.1 Descomposición Funcional: CU 010 “Crear/Modificar perfiles”

### T04.1.2 Diagrama de caso de uso CU 010 “Crear/Modificar perfiles”

### T04.1.3 Especificación de caso de uso CU 010 “Crear/Modificar perfiles”

### T04.1.4 Diagrama de secuencia CU 010 “Crear/Modificar perfiles”

### T04.1.5 Diagrama de clases CU 010 “Crear/Modificar perfiles”

### T04.1.6 Modelo de datos CU 010 “Crear/Modificar perfiles”