A red and white logo

Description automatically generated

**Ingeniería de Software**

**SISTEMA DE GESTION CENTRALIZADO PARA RESTAURANTES (SISFOOD)**

**Apellido y nombre: Jiménez, Daniel**

**Localización: Centro**

**Comisión: 3-A**

**Turno: Mañana**

**Profesor: Jiménez Gamboa, Leonel.**

**Fecha: 08/07/2024**

Contenido

[Historial de Revisión 5](#_Toc176820678)

[G00. Descripción Global del Producto 6](#_Toc176820679)

[G01. Propósito 6](#_Toc176820680)

[G02. Descripción funcional del producto y Alcance 7](#_Toc176820681)

[Requisitos Funcionales 7](#_Toc176820682)

[G03. Definiciones, Acrónimos, y Abreviaciones 9](#_Toc176820683)

[Definiciones: 9](#_Toc176820684)

[Acrónimos 9](#_Toc176820685)

[Abreviaciones 9](#_Toc176820686)

[G04. Descripción de las personas participantes en el desarrollo del sistema de información y los usuarios (Roles) 10](#_Toc176820687)

[G05. Otros Requisitos 12](#_Toc176820688)

[Requisitos del producto 12](#_Toc176820689)

[Estándares aplicables 12](#_Toc176820690)

[Requisitos del sistema 12](#_Toc176820691)

[Requisitos de desempeño 12](#_Toc176820692)

[Requisitos de entorno 12](#_Toc176820693)

[Requisitos de documentación 13](#_Toc176820694)

[Manual de usuario 13](#_Toc176820695)

[Ayuda en línea 13](#_Toc176820696)

[Guias de instalación, configuración, y fichero README. 13](#_Toc176820697)

[Pantalla inicial del sistema 13](#_Toc176820698)

[G06. Diagramas de clases generales por capas 14](#_Toc176820699)

[Diagrama de clases global del negocio 14](#_Toc176820700)

[Diagrama de clases de BE 15](#_Toc176820701)

[Diagrama de clases de BLL 16](#_Toc176820702)

[Diagrama de clases de DAL 17](#_Toc176820703)

[Diagrama de clases de Servicios 18](#_Toc176820704)

[Diagrama de clases de Interfaces 19](#_Toc176820705)

[G07. Modelo de datos parcial de todos los módulos implementados 20](#_Toc176820706)

[N00. Procesos de negocio 21](#_Toc176820707)

[N01. Especificación funcional por proceso de negocio 21](#_Toc176820708)

[N01.A Proceso de negocio “Gestionar Pedido” 21](#_Toc176820709)

[Roles intervinientes 21](#_Toc176820710)

[Diagrama de roles 22](#_Toc176820711)

[Diagrama de secuencia del proceso de negocio “Gestionar Pedido” 23](#_Toc176820712)

[Diagrama Entrada-Comportamiento-Salida 24](#_Toc176820713)

[Diagrama de actividades del proceso de negocio 25](#_Toc176820714)

[Modelo conceptual 26](#_Toc176820715)

[N01.B Proceso de negocio “Comprar productos” 27](#_Toc176820716)

[Roles intervinientes 27](#_Toc176820717)

[Diagrama de roles 27](#_Toc176820718)

[Diagrama de secuencia del proceso de negocio “Comprar insumos” 28](#_Toc176820719)

[Diagrama Entrada-Comportamiento-Salida 29](#_Toc176820720)

[Diagrama de actividades del proceso de negocio 30](#_Toc176820721)

[Primer Modelo conceptual 31](#_Toc176820722)

[N02. Especificaciones de Casos de Uso del sistema 32](#_Toc176820723)

[N02.A Especificaciones de casos de uso del sistema para “Gestionar Pedido” 32](#_Toc176820724)

[Diagrama general de casos de uso 32](#_Toc176820725)

[Caso de uso del sistema CU01 Registrar pedido 33](#_Toc176820726)

[Caso de uso del sistema CU02 Verificar disponibilidad 42](#_Toc176820727)

[Caso de uso del sistema CU03 Cobrar pedido 51](#_Toc176820728)

[Caso de uso del sistema CU04 Ordenar preparación 60](#_Toc176820729)

[Caso de uso del sistema CU05 Notificar pedido listo 69](#_Toc176820730)

[Caso de uso del sistema CU06 Cerrar pedido 78](#_Toc176820731)

[N02.B Especificaciones de casos de uso del sistema para “Comprar insumos” 87](#_Toc176820732)

[Caso de uso del sistema CU06 Cerrar pedido 87](#_Toc176820733)

[T00. Documentos de aspectos técnicos que provee el sistema de información. 92](#_Toc176820734)

[T01. Arquitectura Base 92](#_Toc176820735)

[Persistencia 93](#_Toc176820736)

[Mapa de navegación 96](#_Toc176820737)

[Explorador de soluciones 96](#_Toc176820738)

[T02. Gestión de Log In / Log Out del Sistema 97](#_Toc176820739)

[Objetivo 97](#_Toc176820740)

[Descripción detallada de cómo funciona 97](#_Toc176820741)

[Diagrama de clases 98](#_Toc176820742)

[Diagrama de secuencia Login 99](#_Toc176820743)

[Diagrama de secuencia Logout 100](#_Toc176820744)

[T03. Gestión de Encriptado 101](#_Toc176820745)

[Objetivo 101](#_Toc176820746)

[Descripción detallada de cómo funciona 101](#_Toc176820747)

[Diagrama de clases 101](#_Toc176820748)

[T04. Gestión de perfiles de usuario 102](#_Toc176820749)

[Objetivo 102](#_Toc176820750)

[Descripción detallada de cómo funciona 102](#_Toc176820751)

[Diagrama de clases 105](#_Toc176820752)

[DER 106](#_Toc176820753)

[T05. Gestión de Múltiples Idiomas 107](#_Toc176820754)

[Objetivo 107](#_Toc176820755)

[Descripción detallada de cómo funciona 107](#_Toc176820756)

[Diagrama de clases 108](#_Toc176820757)

[DER 109](#_Toc176820758)

[T06. Gestión de Bitácora y Control de cambios 110](#_Toc176820759)

[T06a. Gestión de Bitácora 110](#_Toc176820760)

[Objetivo 110](#_Toc176820761)

[Descripción detallada de cómo funciona 110](#_Toc176820762)

[Diagrama de clases 110](#_Toc176820763)

[DER 110](#_Toc176820764)

[GUI 111](#_Toc176820765)

[T06b. Control de cambios 111](#_Toc176820766)

Historial de Revisión

|  |  |  |  |
| --- | --- | --- | --- |
| **Fecha** | **Versión** | **Autor** | **Descripción** |
| 27/05/2024 | 1 | Daniel Jiménez |  |
| 24/06/2024 | 2 | Daniel Jiménez |  |

G00. Descripción Global del Producto

# G01. Propósito

El sistema de gestión para restaurantes es una solución informática diseñada específicamente para ayudar a los restaurantes pequeños y medianos a gestionar eficientemente sus operaciones diarias. Su objetivo principal es centralizar todos los procesos relacionados con las ventas, el inventario y las finanzas en un solo sistema de software, proporcionando herramientas integrales para optimizar la eficiencia y mejorar la rentabilidad del negocio.

El **objetivo general** de este trabajo es desarrollar un sistema de gestión para restaurantes que ayude a los propietarios y gerentes de establecimientos gastronómicos a mejorar la eficiencia operativa, aumentar la rentabilidad y ofrecer un mejor servicio a los clientes por medio de la centralización de los procesos de negocios en un solo sistema de software, fácil de usar, pudiendo ser accedido por todos los usuarios que correspondan.

Los objetivos específicos son los siguientes:

* Analizar y estudiar las necesidades principales de las operaciones diarias del negocio que requiere del software.
* Diseñar módulos específicos que faciliten el manejo de los distintos tipos de pedidos, gestión del inventario y de las finanzas del restaurante.
* Realizar un diseño pensado para el usuario común, que funcione de forma intuitiva.

Un restaurante tiene diversas necesidades que debe afrontar todos los días; debe atender los pedidos de los clientes, ya sea por mesa o por mostrador, manejar los productos que ofrece y el inventario de cada uno de sus ingredientes, así como también la compra de dichos productos para reabastecerse, debe administrar el personal rotativo por turnos y debe controlar las finanzas diarias del local.

A partir del estudio de los flujos de actividades comunes en los negocios gastronómicos, se encontró que estos, si son agrupados en un solo sistema mejoran drásticamente el funcionamiento del negocio. Aumentando la productividad y permitiendo tener una mirada centrada en cada uno de los distintos procesos que se ejecutan diariamente en un restaurante.

Debido a esto, el desarrollo de un sistema de gestión centralizado para restaurantes viene motivado por la necesidad de mejorar la eficiencia de una rama de negocios que puede resultar bastante caótica e ineficaz sin tener un sistema de apoyo para sus tareas del día a día.

# G02. Descripción funcional del producto y Alcance

## Requisitos Funcionales

**RFN1: Gestión de pedidos en el restaurante.** En cada restaurante se realiza el proceso de gestión de pedidos de los clientes que llegan físicamente al local a comprar. Esta gestión involucra el registro del cliente, verificación de la existencia de los ingredientes de los productos solicitados en el pedido, la preparación de dichos productos, el cobro del pedido y su respectiva entrega.

**PN1. Gestión de pedido**:

1. El **cliente** le indica al cajero que **productos** desea pedir (*Nombre, cantidad, etc.*). En este punto el cliente también brinda sus datos personales (*DNI, nombre, apellido y teléfono*) para realizar el **pedido** (*número, fecha, estado, etc*.).
2. El cajero registra los datos del **cliente** (*DNI, nombre, apellido y teléfono*) y los productos que desea, creando así la orden del cliente **(pedido)** con la siguiente informacion: *Número, fecha, estado, etc*. en un *estado por verificar*, pasando el pedido al jefe de cocina.
3. El jefe de cocina procede a verificar la disponibilidad de los productos contenidos en el **pedido** (*número, fecha, estado, etc*.). En este caso, chequea en sistema que haya en existencia la cantidad de **ingredientes** (*código, cantidad, nombre*) necesarios para la preparación de todo el pedido. En caso de que todos los ingredientes se encuentren disponibles, se le notifica al cajero que el pedido fue *aceptado*, y en caso contrario, se le notifica su *rechazo*. El jefe de cocina reserva en sistema la *cantidad* de cada ingrediente requerido para la preparación del pedido.
4. El jefe de cocina ordena la preparación del **pedido** (*número, fecha, estado, etc*.) al **cocinero** (nombre, apellido, DNI, disponibilidad, etc.) de turno que se encuentre *disponible*. Para esto, el jefe de cocina genera una **comanda** (*pedido,* *descripción, cocinero*) con los productos a realizar y las instrucciones especiales en caso de existir, y esta es asignada al cocinero.
5. Paralelamente a su preparación, el cajero puede gestionar el cobro del **pedido** (*número, fecha, estado, etc*.). Para esto, le comunica al cliente el *monto* a pagar y le ofrece los **métodos de pago** (*tipo*): *Efectivo* o *tarjeta*. En caso de que sea tarjeta, el cajero le solicitara los datos de esta: *Número de tarjeta, fecha de vencimiento, titular y CVV*, para registrar el pago. En caso de ser cobro en efectivo, el cajero ingresa el *monto recibido* del cliente.
6. Finalmente, el cajero emite un *documento (recibo de pago)* constatando *la fecha, los productos pedidos, su cantidad, precio y total* del **pedido** (*número, fecha, estado, etc*.), el cual es entregado al cliente.
7. Una vez que el **pedido** (*número, fecha, estado, etc*.) está *listo* y ha sido *pagado*, el cajero procederá a *cerrar* el pedido en el sistema.

**RFN2: Gestión de compras de insumos**. El sistema permitirá la realización de orden de compra, que será enviada al proveedor seleccionado indicando los productos, con su cantidad, que se desean comprar.

**PN2. Comprar insumos**

1. El jefe de cocina procede a realizar el chequeo de inventario físico, para determinar la *cantidad de productos faltantes*, comparando con los **ingredientes** (*nombre, cantidad, stockMínimo, stockMáximo, etc.)* registrados en sistema.
2. Con la informacion obtenida, el jefe de cocina genera una **solicitud de compra** (*nroSolicitud, fecha, ingredientes, estado*) y esta es enviada al encargado.
3. El encargado procede a verificar la **solicitud de compra** (*nroSolicitud, fecha, ingredientes, estado*). En caso de que no sea *aprobada*, este le informa al jefe de cocina su *rechazo*. En caso contrario, es enviada al jefe de suministros.
4. El jefe de suministros pide **cotizaciones** (*fecha, nroCotizacion, ingredientes con cantidad y precios*) a distintos **proveedores** (*nombre, email, telefono, direccion, cuit*) en base a la **solicitud de compra** (*nroSolicitud, fecha, ingredientes, estado*)., solicitando *la cotización* de los productos necesarios.
5. El jefe de suministros evalúa *las* **cotizaciones** recibidas (*fecha, nroCotizacion, ingredientes con cantidad y precios*), seleccionando la más conveniente de acuerdo con los criterios requeridos. En caso de no seleccionar alguna, se procede a solicitar nuevas *cotizaciones* a distintos **proveedores** (*nombre, email, telefono, direccion, cuit*).
6. El jefe de suministros emite una **orden de compra** (*nroOrden, fecha, condicionDePago)* en base *a la* **cotización** (*fecha, nroCotizacion, ingredientes con cantidad y precios*) *seleccionada* y la envía al **proveedor** (*nombre, email, telefono, direccion, cuit*).
7. El proveedor despacha la **orden de compra** (*nroOrden, fecha, condicionDePago)* y emite una factura.
8. El jefe de suministro verifica la compra recibida de acuerdo con la **orden de compra** (*nroOrden, fecha, condicionDePago)*. En caso de que haya discrepancias, se informa la no conformidad al proveedor.
9. El jefe de suministros procede a realizar el **pago** de la **orden de compra** (*nroOrden, fecha, condicionDePago*). Registra dicho pago (*orden, monto, fechaEmision, fechaPago*).
10. El jefe de cocina se encarga de actualizar las cantidades de los **ingredientes** (*nombre, cantidad, stockMínimo, stockMáximo, etc.*) recibidos en el sistema.

# G03. Definiciones, Acrónimos, y Abreviaciones

## Definiciones:

* **Cliente**: Persona natural de la que se recibe un pedido.
* **Pedido**: Conjunto de productos solicitados por un cliente.
* **Catálogo**: Lista de productos/ingredientes ofrecidos en el restaurante.
* **Producto**: Son los ítems ofrecidos por el restaurante, estos pueden ser platos o bebidas.
* **Ingrediente**: Componente básico que conforma a un plato.
* **Insumo**: Todo elemento que forma parte en la preparación de los productos.
* **Stock**: Inventario de ingredientes necesarios para realizar los productos ofrecidos por el restaurante.
* **Orden**: Solicitud realizada por un miembro del personal de restaurante, puede ser una orden resultante de un pedido realizada por un cajero, o una orden de compra realizada por el jefe de suministros.
* **Comanda**: Orden de preparación asignada a un cocinero por cada pedido.
* **Proveedor**: Persona no natural a la cual se le envía una solicitud de compra pidiendo presupuesto, y se reciben cotizaciones, facturas de compra e insumos.
* **Factura**: Documento enviado por un proveedor que detalla los insumos enviados.

## Acrónimos

* **POS**: Point of sale. Se refiere al Sistema de transacciones de ventas utilizados para procesar los pagos con tarjetas.
* **QA**:

## Abreviaciones

* **TC**: Tarjeta de Crédito.
* **TD**: Tarjeta de Débito.
* **Ing**: Ingrediente.
* **Prod**: Producto.

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

Primero, se encuentran aquellas personas que tienen poder de decisión, financiamiento o tienen la capacidad de colaborar directamente en el desarrollo del sistema de software:

|  |  |  |
| --- | --- | --- |
| **Nombre** | **Descripción** | **Responsabilidad** |
| Barnique, Andrés | Gerente del restaurante | Define las necesidades del restaurante que debe satisfacer el sistema |
| Jiménez, Daniel | Analista de sistemas | Releva los requerimientos del sistema, los documenta y los implementa. |
| Salas, Mariajose | QA analyst | Diseña e implementa todas las pruebas necesarias para cumplir con los requerimientos del sistema. |
| Kass Hanna, Maria | Lead Developer | Desarrollador lider del Proyecto. Supervisa y dirige las etapas mas importantes del desarrollo del sistema |
| Isturiz, Eduardo | Frontend developer | Se encarga de diseñar la interfaz de usuario y mejorar la experiencia del usuario. |

Los usuarios de sistema serán los siguientes:

|  |  |  |
| --- | --- | --- |
| **Nombre** | **Rol** | **Acceso** |
| Tankian, Serj | Administrador | * Stock de ingredientes * Stock de productos * Compra * Reabastecimiento * Gestión de usuarios y permisos * Registro de pedidos * Solicitud de presupuesto * Orden de Compra |
| Smith, Alan | Encargado | * Apertura y cierre de caja * Arqueo de caja automático * Arqueo de caja manual * Stock de ingredientes * Stock de productos * Solicitud de presupuesto * Orden de Compra * Facturación * Registro de pedidos |
| Ramsay, Gordon | Jefe de cocina | * Stock de ingredientes * Catálogo de productos * Reabastecimiento |
| Diaz, Esteban | Cajero | * Catálogo de productos * Registro de pedido |
| Mendoza, Lola | Cocinera | * Stock de ingredientes * Catálogo de productos |
| Martínez, Dina | Cocinera | * Stock de ingredientes * Catálogo de productos |

# G05. Otros Requisitos

## Requisitos del producto

### Estándares aplicables

* **Facturación electrónica**: Bajo las normativas del AFIP (Administración Federal de Ingresos Públicos) el sistema debe estar preparado para realización facturación electrónica para todas sus operaciones comerciales conforme a las regulaciones vigentes.
* **Impuestos y contribuciones**: El sistema debe cumplir con las obligaciones fiscales, como el pago de impuestos sobre las ventas.
* **Registro de movimientos financieros**: El sistema, al gestionar las ventas, estará obligado a llevar registros detallados de todas las ventas y movimientos financieros, incluyendo ingresos, egresos y pago a proveedores.
* **Propiedad intelectual**: La propiedad intelectual del sistema corresponde a los autores durante su vida, nadie tiene derecho a publicar, sin permiso de los autores, la producción de esta, estando protegida por la Ley 11.723 (Ley de Propiedad Intelectual).

### Requisitos del sistema

El sistema será compatible con el sistema operativo Windows 10 en adelante al ser desarrollado sobre .NET framework 4.8 usando Winforms. Este funcionará con SQL server 16.01115.1. No se garantiza compatibilidad con versiones futuras de dichos entornos de desarrollo, pero se podrán hacer modificaciones para soportar estas.

El sistema ademas debe tener compatibilidad con sistemas de pago POSNET, y la utilización de QR como MODO y Mercado Pago.

### Requisitos de desempeño

El sistema debe enfocarse en la robustez y fluidez para que este se encuentre siempre funcional durante la jornada laboral del local sin que haga esperar a los usuarios. No deberá tardar más de 4 segundos en la carga de catálogos, ni más de 5 segundos realizando los procesamientos de pago de los pedidos.

### Requisitos de entorno

Se deberá contar con Windows 10 de 32 o 64 bits para las PCs, Microsoft SQL server 2022 Standard Edition y conexión LAN. Se deberá contar con al menos 8GB de RAM para el correcto desempeño del sistema y contar con 1GB de almacenamiento. Deberá tener acceso a internet para el procesamiento de los pagos y las consultas a realizar con los proveedores.

## Requisitos de documentación

### Manual de usuario

Se entregará un manual de usuario en formato pdf que consistirá de la explicacion detallada de cada módulo del sistema, instrucciones para su uso, ademas de guias visuales con capturas de pantalla del software señalando las partes importantes con las que el usuario podrá interactuar.

### Ayuda en línea

Consistirá en videos explicativos y zona de preguntas y respuestas frecuentas acerca de los módulos del sistema ya explicados en el manual de usuario. Este podra ser accedido libremente por cualquier usuario.

### Guias de instalación, configuración, y fichero README.

Junto al manual de usuario, se entregará una guia específica para la instalación del sistema con sus configuraciones iniciales por las que puede optar el administrador del sistema. Ademas, poseerá un archivo README que tendrá esta misma informacion de manera concisa junto a links que lo lleve a la ayuda en línea.

## Pantalla inicial del sistema

A screenshot of a computer

Description automatically generated

# G06. Diagramas de clases generales por capas

## Diagrama de clases global del negocio

A diagram of a computer program

Description automatically generated with medium confidence

## Diagrama de clases de BE

A computer screen shot of a computer flowchart

Description automatically generated

## Diagrama de clases de BLL

A screenshot of a computer flowchart

Description automatically generated

## Diagrama de clases de DAL

A diagram of a computer code

Description automatically generated with medium confidence

## Diagrama de clases de Servicios

A screenshot of a computer program

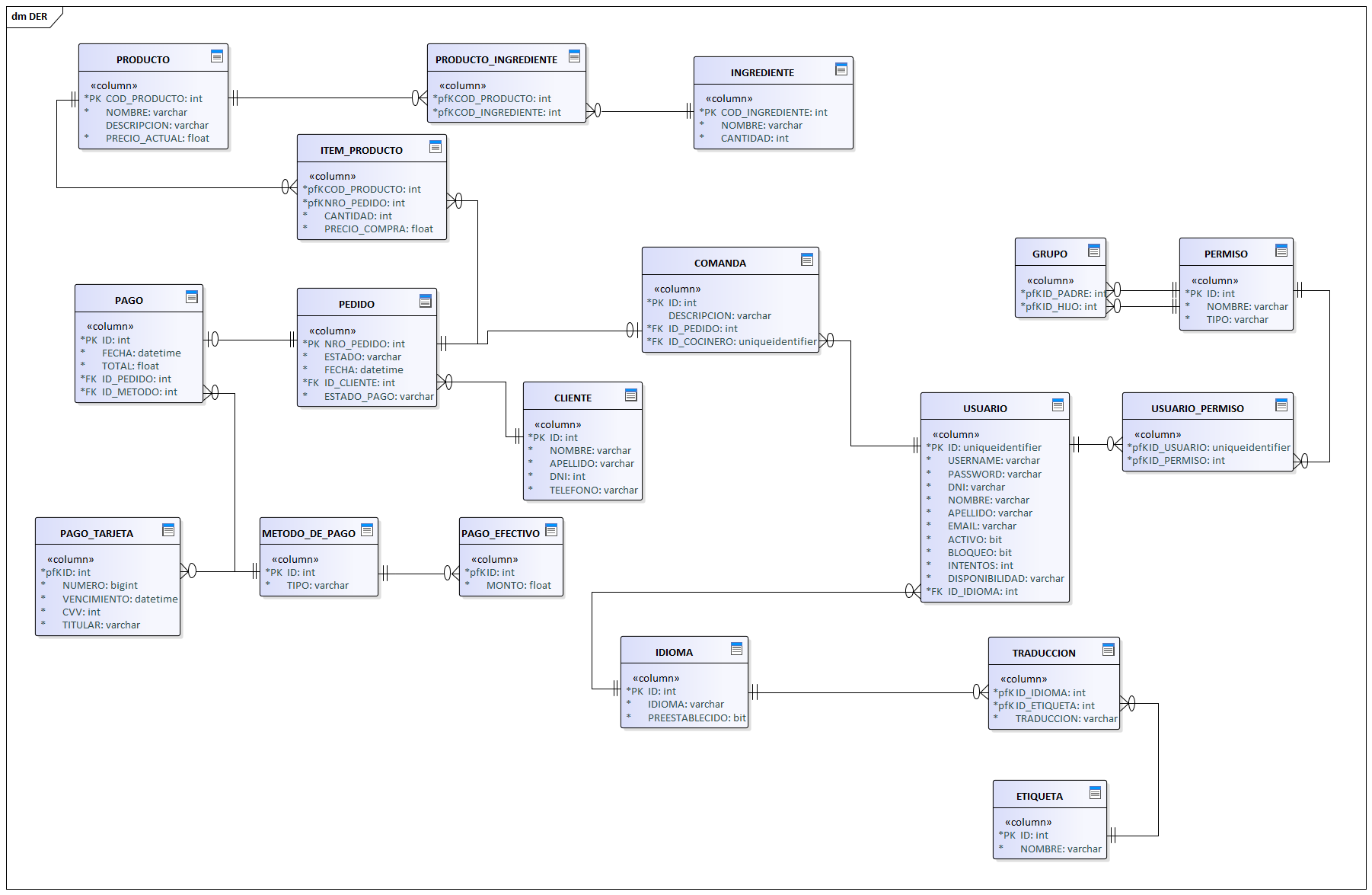
Description automatically generated

## Diagrama de clases de Interfaces

A screenshot of a computer program

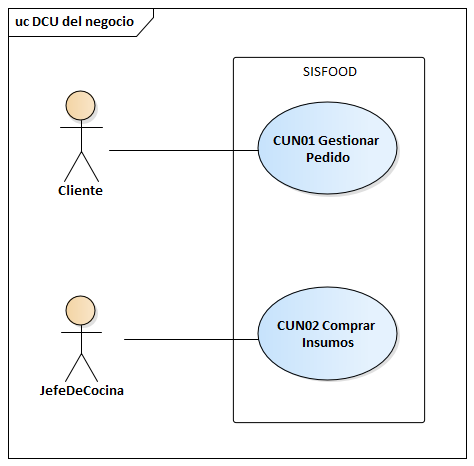
Description automatically generated

# G07. Modelo de datos parcial de todos los módulos implementados



N00. Procesos de negocio

El sistema se encuentra dividido por 2 procesos de negocio, los cuales se puede apreciar en el siguiente diagrama de casos de uso del sistema:



# N01. Especificación funcional por proceso de negocio

## N01.A Proceso de negocio “Gestionar Pedido”

### Roles intervinientes

* **Cliente**: Es el iniciador, no interviene con la GUI, y genera la informacion de entrada del proceso:
* **Cajero**: Actor primario, usuario del sistema
* **Jefe de Cocina**: Actor primario, usuario del sistema
* **Cocinero**: Actor secundario

### Diagrama de roles

A diagram of a company

Description automatically generated

### Diagrama de secuencia del proceso de negocio “Gestionar Pedido”

A diagram of a structure

Description automatically generated

### Diagrama Entrada-Comportamiento-Salida

### Diagrama de actividades del proceso de negocio

A diagram of a computer

Description automatically generated

### Modelo conceptual

A diagram of a company

Description automatically generated

## N01.B Proceso de negocio “Comprar productos”

### Roles intervinientes

* **Jefe de cocina**: Actor primario. Es quien inicia el proceso. Fuente inicial de información. Usuario del sistema.
* **Encargado**: Actor primario. Usuario del sistema.
* **Jefe de suministros**: Actor primario. Usuario principal del sistema.
* **Proveedor**: No es actor directo, es una fuente de información. Agente externo al sistema.

### Diagrama de roles

A diagram of a company

Description automatically generated

### Diagrama de secuencia del proceso de negocio “Comprar insumos”

A diagram of a project

Description automatically generated

### Diagrama Entrada-Comportamiento-Salida

### Diagrama de actividades del proceso de negocio

A diagram of a flowchart

Description automatically generated

### Primer Modelo conceptual

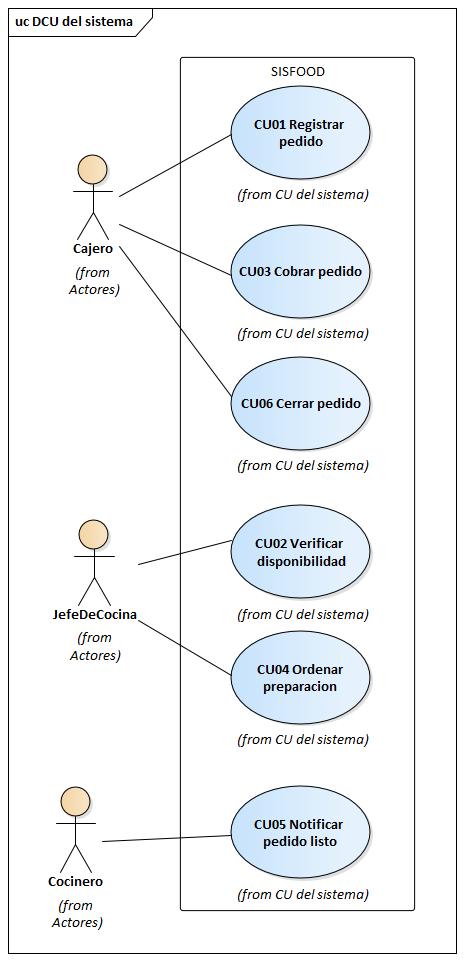
A diagram of a computer

Description automatically generated

# N02. Especificaciones de Casos de Uso del sistema

## N02.A Especificaciones de casos de uso del sistema para “Gestionar Pedido”

### Diagrama general de casos de uso



### Caso de uso del sistema CU01 Registrar pedido

#### Historial de revisión de la especificación

|  |  |  |  |
| --- | --- | --- | --- |
| Fecha | Versión | Descripción | Autor |
| 27/05/2024 | 1.0 | Versión inicial | Daniel Jiménez |

#### Objetivo

Capturar la informacion del pedido emitido por el cliente, almacenarla en el sistema y enviarla al jefe de cocina, asegurando que esté disponible para su posterior procesamiento y gestión dentro del sistema de gestión del restaurante.

#### Precondiciones

Cajero ya inició sesión en el sistema

#### Postcondiciones

Pedido registrado en el sistema.

Su estado se actualiza a “Registrado”

#### Eventos disparadores del caso de uso

Cliente realiza un pedido

#### Diagrama del caso de uso

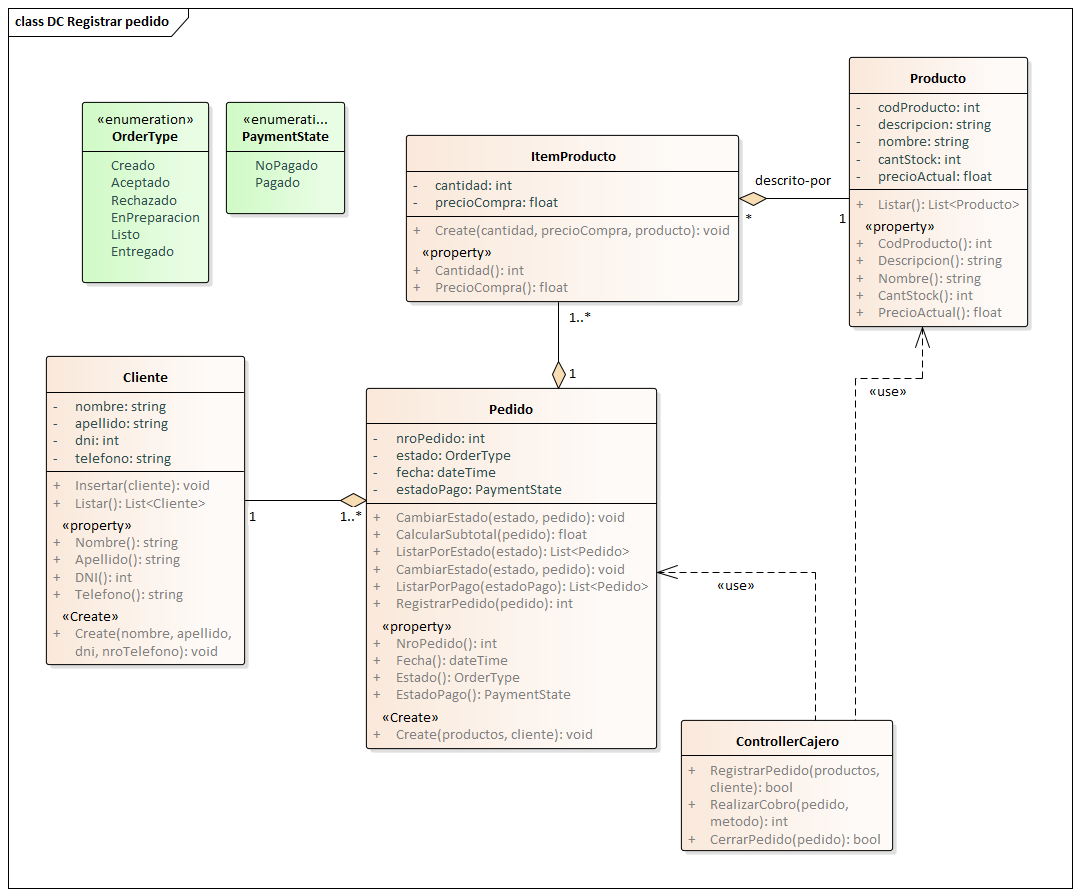
A diagram of a person

Description automatically generated

#### Descripción analítica del caso de uso

|  |
| --- |
| **ID y Nombre**: CU01 Registrar pedido |
| **Estado**: Pendiente. |
| **Descripción**: El usuario registra un pedido en el sistema |
| **Actor Principal**: Cajero |
| **Actor Secundario**: - |
| **Puntos de Extensión**: - |
| **Puntos de inclusión:** Paso 2. CU09 Registrar Cliente |
| **Escenario Principal**:   1. El cajero ingresa al sistema en la sección “Crear pedido”. 2. El sistema muestra una pantalla para ingresar los datos pertinentes al pedido: Nombre y apellido del cliente, número de teléfono, DNI, productos escogidos, cantidad de cada uno. 3. El cajero ingresa los datos del cliente. 4. El cajero ingresa cada producto seleccionándolo del catálogo y presiona el botón “Agregar”. 5. El sistema muestra el producto agregado, junto a su cantidad y precio. 6. Se repite el paso 4 cuantas veces se requiera. 7. El cajero finaliza la carga de productos y presiona el botón “Crear”. 8. El sistema almacena los datos del pedido y del cliente. 9. El sistema actualiza el estado de pedido a “Registrado”. |
| **Flujos Alternativos**:   * 1. Mensaje de error: Por favor selecciona un producto a agregar.   2. Mensaje de error: Debe seleccionar al menos un producto antes de crear el pedido.   3. Mensaje de error: Debe completar todos los datos del cliente.   4. Mensaje de error: El DNI debe ser un número válido. |

#### Diagrama de clase



#### Diagrama de secuencia

A screenshot of a computer

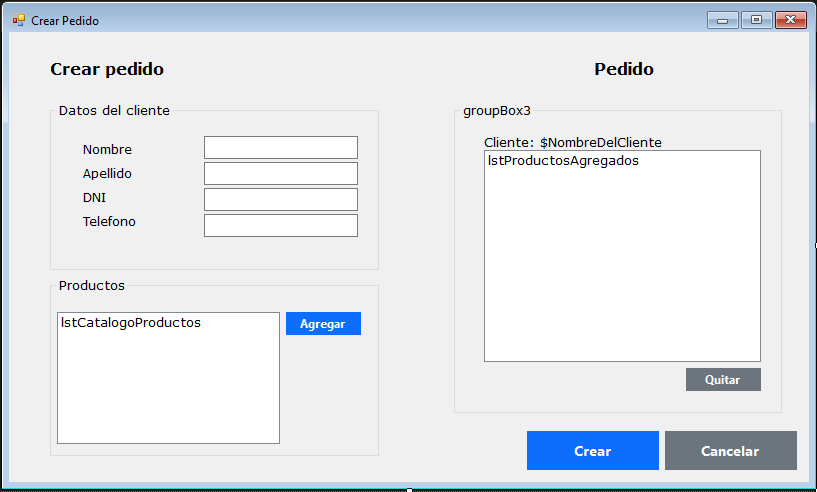
Description automatically generated

#### DER

A computer screen shot of a diagram

Description automatically generated

#### Prototipo de interfaz de usuario



### Caso de uso del sistema CU02 Verificar disponibilidad

#### Historial de revisión de la especificación

|  |  |  |  |
| --- | --- | --- | --- |
| Fecha | Versión | Descripción | Autor |
| 27/05/2024 | 1.0 | Versión inicial | Daniel Jiménez |

#### Objetivo

Verificar disponibilidad de los productos que conforman al pedido registrado, para luego notificar su resultado al cajero.

#### Precondiciones

Jefe de cocina ya inicio sesión en el sistema.

#### Postcondiciones

El pedido es verificado. Su estado en el sistema pasa a “Verificado”

#### Eventos disparadores del caso de uso

Cajero registra un pedido.

#### Diagrama del caso de uso

A diagram of a person with a person in the middle

Description automatically generated

#### Descripción analítica del caso de uso

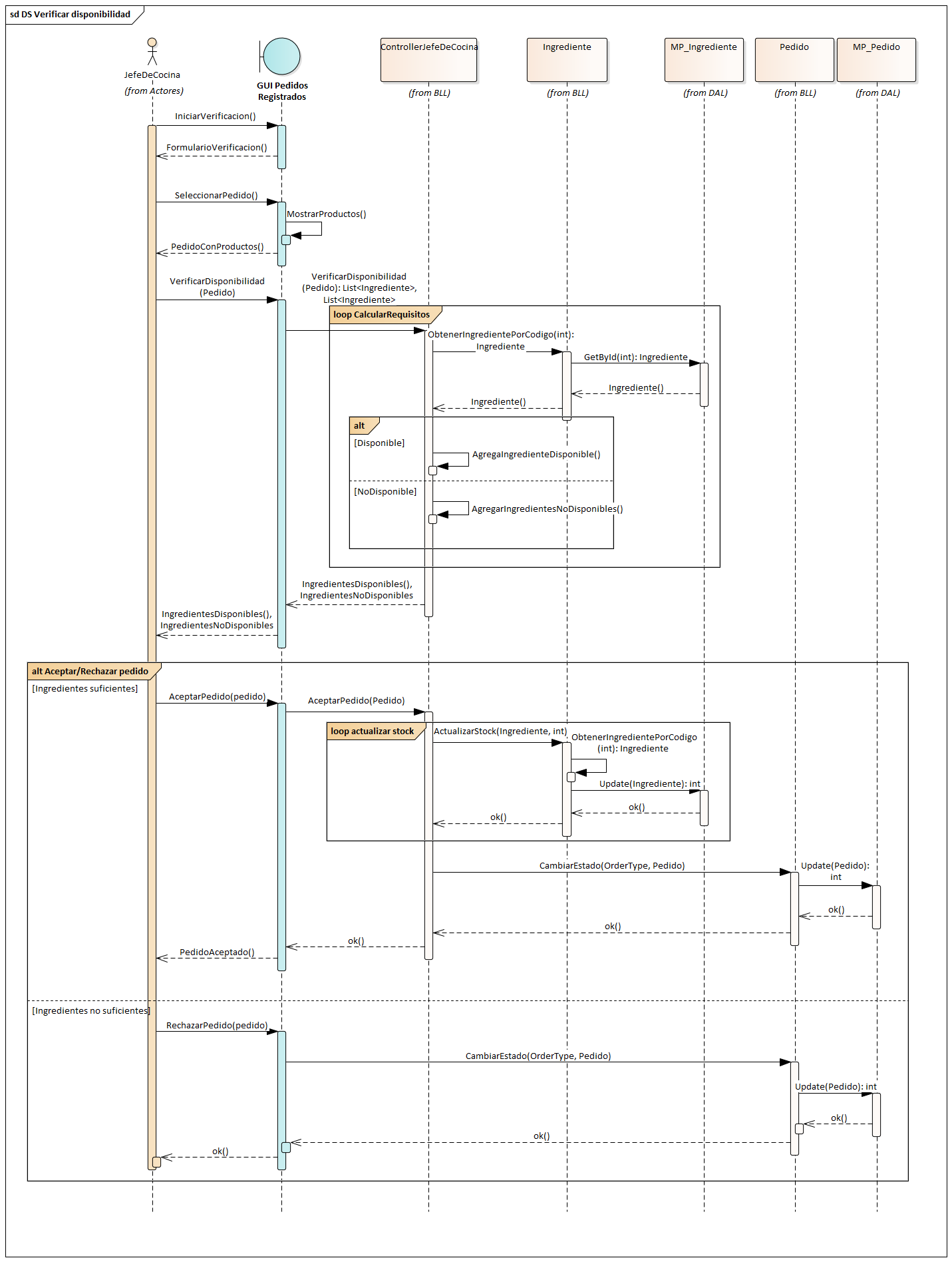
|  |
| --- |
| **ID y Nombre**: CU02 Verificar disponibilidad |
| **Estado**: Pendiente. |
| **Descripción**: Se verifica la disponibilidad de los productos del pedido |
| **Actor Principal**: Jefe de cocina |
| **Actor Secundario**: |
| **Puntos de Extensión**: Paso 6. CU04 Notificar pedido aceptado, CU03 Notificar pedido rechazado |
| **Puntos de Inclusión**: |
| **Escenario Principal**:   1. El jefe de cocina ingresa a la sección de Pedidos registrados. 2. El sistema muestra una lista de todos los pedidos creados sin verificar. 3. El jeje de cocina selecciona un pedido. 4. El sistema muestra una lista con los productos del pedido seleccionado. 5. El jefe de cocina selecciona el botón “verificar”. 6. El sistema separa en 2 listas los productos: Los disponibles y los no disponibles. 7. El jefe de cocina acepta o rechaza el pedido en base a lo mostrado por el sistema. 8. El sistema actualiza el pedido al estado que corresponda. |
| **Flujos Alternativos**:   * 1. Mensaje de información: Algunos ingredientes no están disponibles.   2. Mensaje de información: Todos los ingredientes están disponibles.   3. Pedido Aceptado: Ver CU04 Notificar pedido aceptado.   4. Pedido rechazado: Ver CU03 Notificar pedido rechazado. |

#### Diagrama de clase

A screenshot of a computer

Description automatically generated

#### Diagrama de secuencia

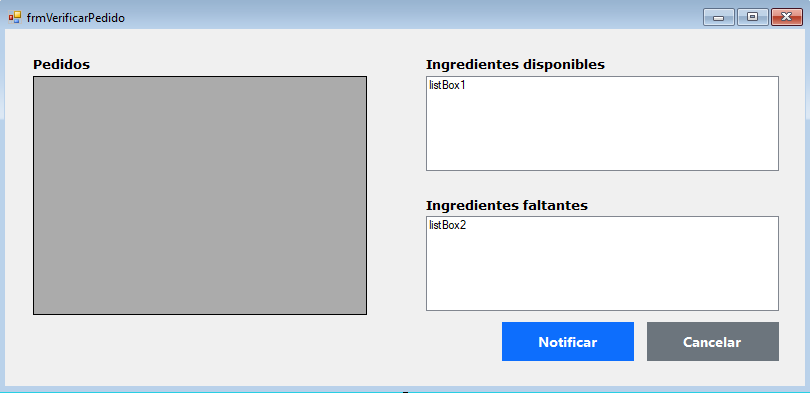


#### DER

A diagram of a product

Description automatically generated

#### Prototipo de interfaz de usuario



### Caso de uso del sistema CU03 Cobrar pedido

#### Historial de revisión de la especificación

|  |  |  |  |
| --- | --- | --- | --- |
| Fecha | Versión | Descripción | Autor |
| 27/05/2024 | 1.0 | Versión inicial | Daniel Jiménez |

#### Objetivo

Facilitar el proceso de pago de los pedidos en el restaurante, generando una factura detallada y actualizando el estado del pedido una vez completado el pago.

#### Precondiciones

Pedido registrado en el sistema.

Pedido verificado y aceptado.

#### Postcondiciones

Se registró el pago del pedido en el sistema.

#### Eventos disparadores del caso de uso

Cajero notificado de pedido aceptado.

#### Diagrama del caso de uso

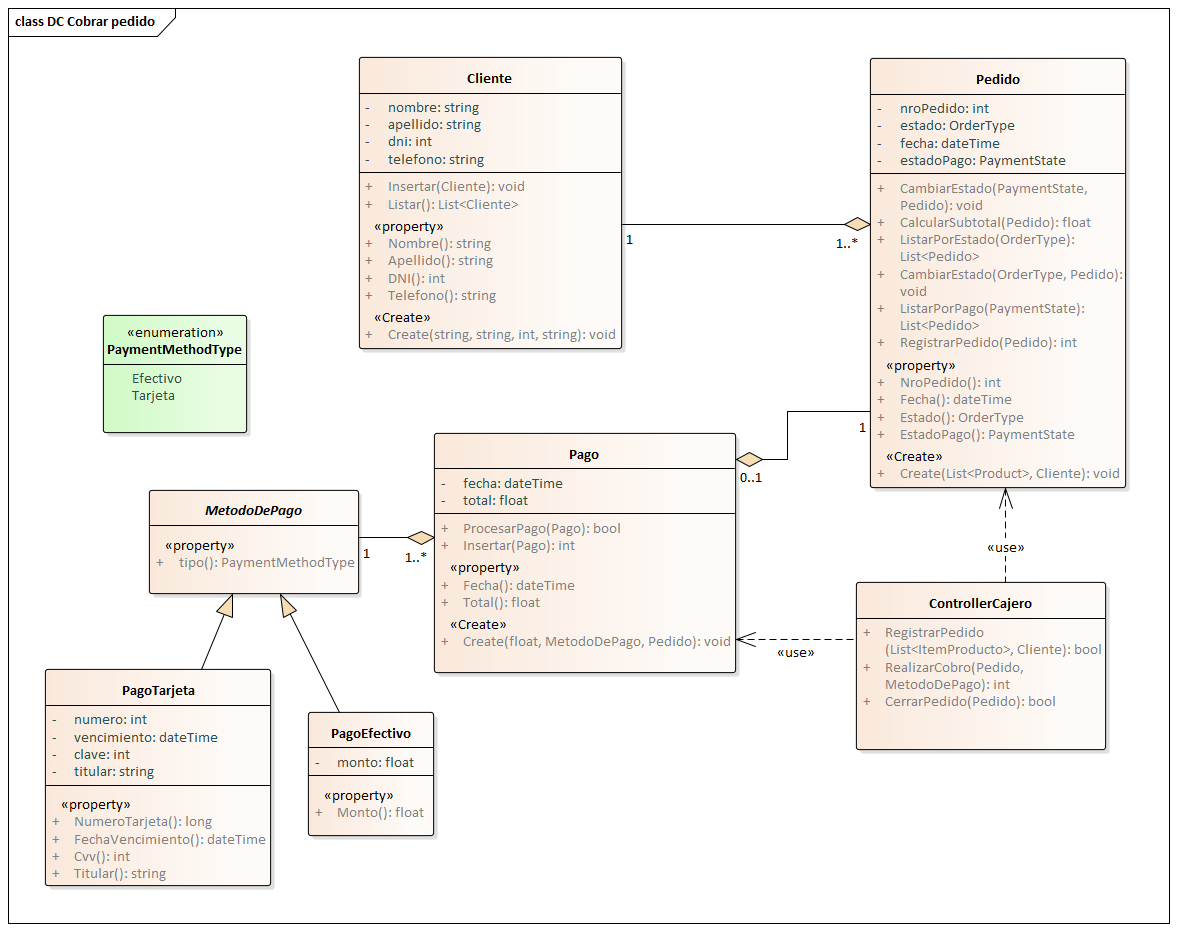
A diagram of a person with a person's body

Description automatically generated

#### Descripción analítica del caso de uso

|  |
| --- |
| **ID y Nombre**: CU03 Cobrar pedido |
| **Estado**: Pendiente. |
| **Descripción**: Realizar cobro del pedido al cliente correspondiente |
| **Actor Principal**: Cajero |
| **Actor Secundario**: |
| **Puntos de Extensión**: |
| **Puntos de Inclusión**: |
| **Escenario Principal**:   1. El cajero ingresa a la sección “Cobrar pedido” 2. El sistema muestra al cajero los pedidos verificados que no han sido pagados. 3. El cajero selecciona el pedido que se desee cobrar 4. El sistema muestra el total a cobrar. 5. El cajero selecciona el metodo de pago (Tarjeta o efectivo). 6. El sistema muestra las secciones para guardar los datos correspondientes al metodo de pago elegido. 7. El cajero solicita los datos al cliente y los ingresa al sistema. 8. El sistema verifica el cobro y concluye el proceso. |
| **Flujos Alternativos**:   * 1. Mensaje de error: Seleccione un pedido valido.   2. Mensaje de error: Seleccione un metodo de pago.   3. Mensaje de error: Complete todos los campos de la tarjeta.   4. Mensaje de error: Ingrese un monto en efectivo   5. Mensaje de error: Pago rechazado. Error al registrar el pago. |

#### Diagrama de clase



#### Diagrama de secuencia

A diagram of a computer program

Description automatically generated

#### DER

A computer screen shot of a computer

Description automatically generated

#### Prototipo de interfaz de usuario

A screenshot of a computer

Description automatically generated

### Caso de uso del sistema CU04 Ordenar preparación

#### Historial de revisión de la especificación

|  |  |  |  |
| --- | --- | --- | --- |
| Fecha | Versión | Descripción | Autor |
| 27/05/2024 | 1.0 | Versión inicial | Daniel Jiménez |

#### Objetivo

Preparar una orden detallada de productos solicitados por los clientes.

#### Precondiciones

El jefe de cocina inicio sesión en el sistema.

#### Postcondiciones

Se ordenó la preparación de un pedido al personal de cocina.

#### Eventos disparadores del caso de uso

El pedido fue verificado y aceptado.

#### Diagrama del caso de uso

A diagram of a person

Description automatically generated

#### Descripción analítica del caso de uso

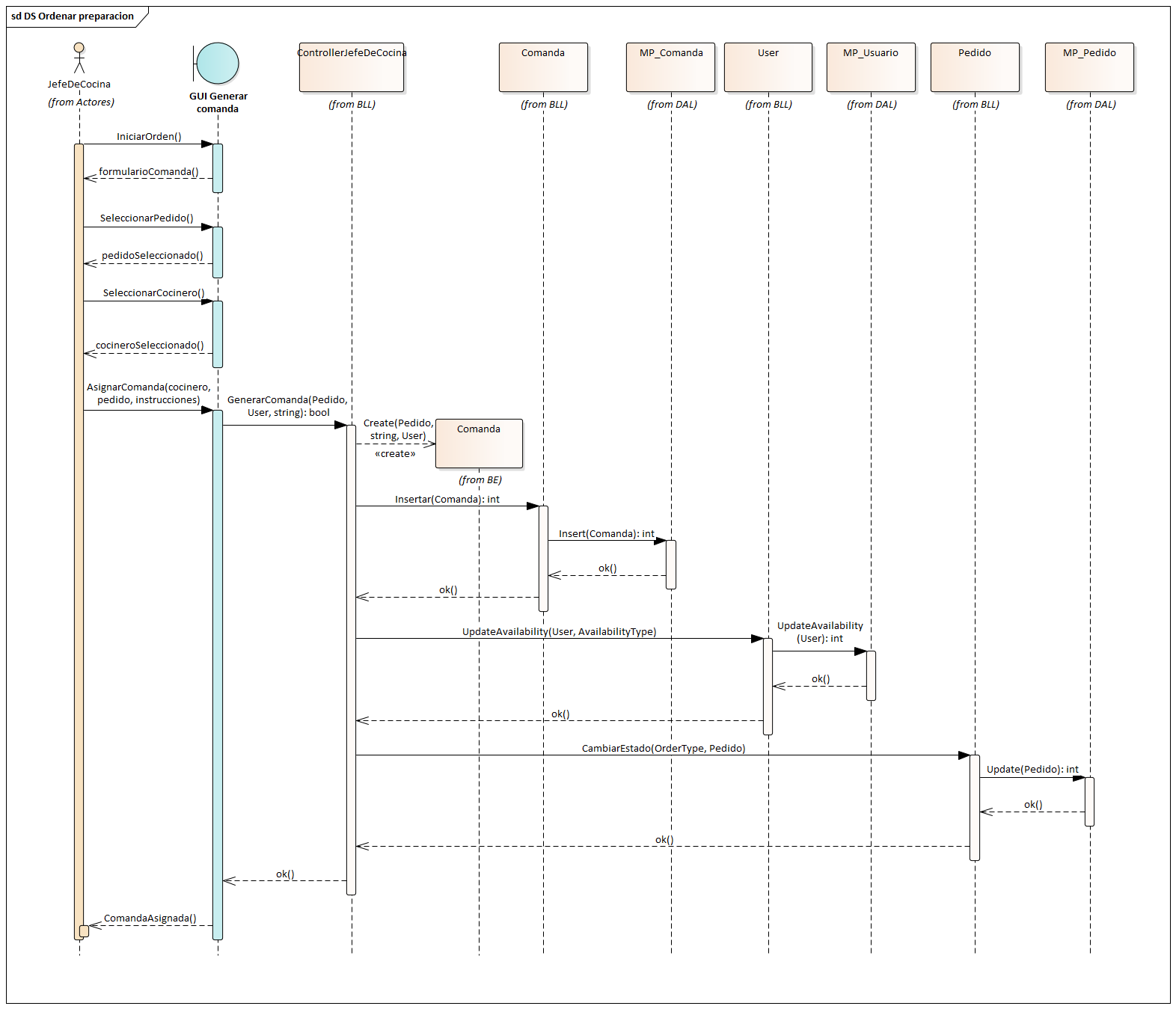
|  |
| --- |
| **ID y Nombre**: CU04 Ordenar preparación |
| **Estado**: Pendiente. |
| **Descripción**: Se ordena al personal de cocina la preparación de un producto por medio de una comanda |
| **Actor Principal**: Jefe De Cocina |
| **Actor Secundario**: |
| **Puntos de Extensión**: |
| **Escenario Principal**:   1. El jefe de cocina ingresa a la sección “Generar comanda” 2. El sistema muestra la lista de pedidos aceptados. 3. El jefe de cocina selecciona un pedido que desee ordenar su preparación. 4. El sistema muestra los productos que conforman al pedido y muestra una lista con los cocineros disponibles actualmente. 5. El usuario selecciona al cocinero a quien desee asignarle el pedido, ingresa los detalles de instrucciones adicionales (si es que las hay) y selecciona “Generar”. 6. El sistema envía dicha comanda al personal de cocina para que sea preparada. |
| **Flujos Alternativos**:   * 1. Mensaje de error: Por favor, seleccione un pedido.   2. Mensaje de error: Por favor, seleccione un cocinero.   3. Mensaje de error: No se pudo generar la comanda. |

#### Diagrama de clase

A diagram of a computer

Description automatically generated

#### Diagrama de secuencia



#### DER

A computer diagram of a computer

Description automatically generated with medium confidence

#### Prototipo de interfaz de usuario

A screenshot of a computer

Description automatically generated

### Caso de uso del sistema CU05 Notificar pedido listo

#### Historial de revisión de la especificación

|  |  |  |  |
| --- | --- | --- | --- |
| Fecha | Versión | Descripción | Autor |
| 27/05/2024 | 1.0 | Versión inicial | Daniel Jiménez |

#### Objetivo

Permitir al cocinero revisar la comanda que se le ha sido asignada para verificar sus instrucciones y poder marcar al pedido como “listo” una vez finalizada su preparación.

#### Precondiciones

El cocinero ha iniciado sesión en el sistema.

Se ha generado una comanda y se le ha asignado para su preparación.

#### Postcondiciones

El pedido pasa a estar listo.

#### Eventos disparadores del caso de uso

El cocinero termina la preparación del pedido

#### Diagrama del caso de uso

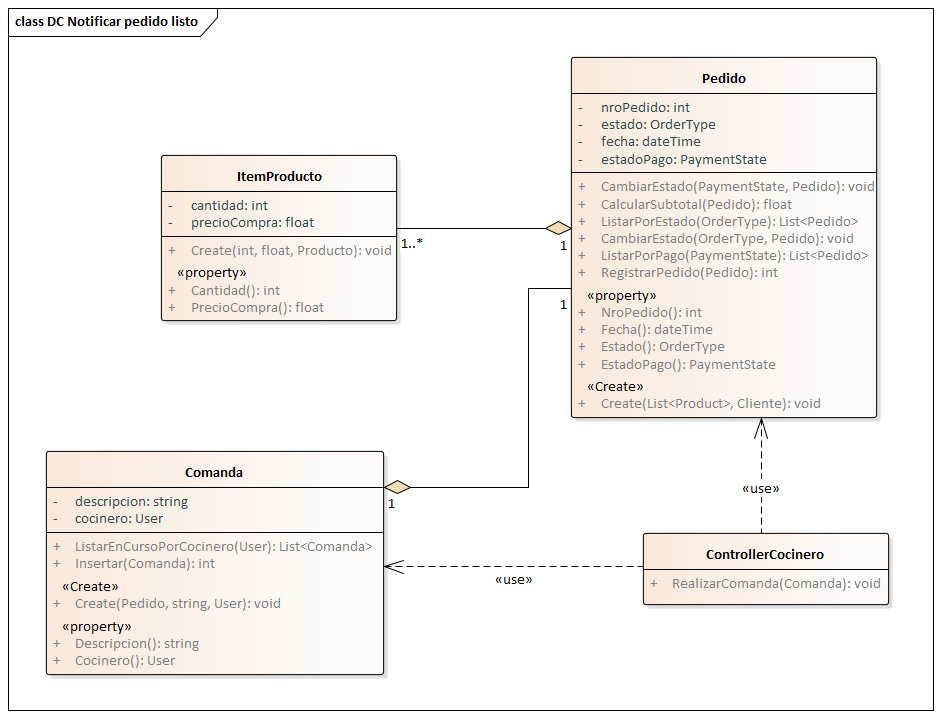
A diagram of a person with a person's face

Description automatically generated

#### Descripción analítica del caso de uso

|  |
| --- |
| **ID y Nombre**: CU05 Notificar pedido listo |
| **Estado**: Pendiente. |
| **Descripción**: EL usuario verifica el pedido y marca en el sistema que este está listo. |
| **Actor Principal**: Cocinero |
| **Actor Secundario**: |
| **Puntos de Extensión**: |
| **Escenario Principal**:   1. El cocinero ingresa a la sección de “Ver comanda”. 2. El sistema muestra un formulario donde se verá la comanda que tenga asignada el cocinero (usuario actual del sistema). 3. El cocinero selecciona la comanda. 4. El sistema muestra los productos a preparar. 5. El cocinero selecciona “Pasar a listo” una vez que haya preparado el pedido. 6. El sistema pasa el pedido a “listo” y el cocinero vuelve a estar disponible. |
| **Flujos Alternativos**:   * 1. Mensaje de error: Por favor seleccione una comanda. |

#### Diagrama de clase



#### Diagrama de secuencia

A diagram of a project

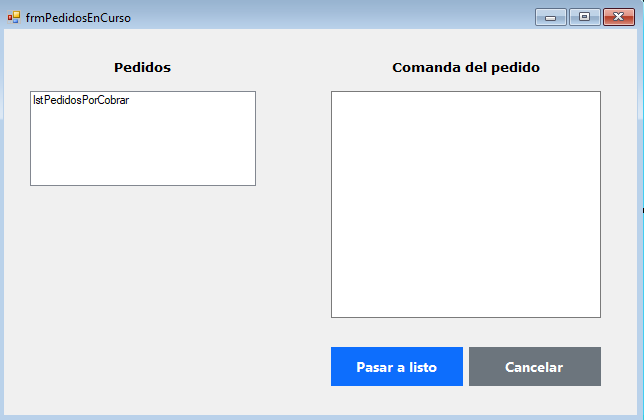
Description automatically generated

#### DER

A computer screen shot of a computer

Description automatically generated

#### Prototipo de interfaz de usuario



### Caso de uso del sistema CU06 Cerrar pedido

#### Historial de revisión de la especificación

|  |  |  |  |
| --- | --- | --- | --- |
| Fecha | Versión | Descripción | Autor |
| 27/05/2024 | 1.0 | Versión inicial | Daniel Jiménez |

#### Objetivo

Permitir al cajero dar por cerrado un pedido una vez que ya haya sido cobrado y terminado su preparación.

#### Precondiciones

El cajero ha iniciado sesión en el sistema.

#### Postcondiciones

El pedido es entregado al cliente

#### Eventos disparadores del caso de uso

Se marco como “listo” el pedido.

#### Diagrama del caso de uso

A diagram of a person with a circle and text

Description automatically generated

#### Descripción analítica del caso de uso

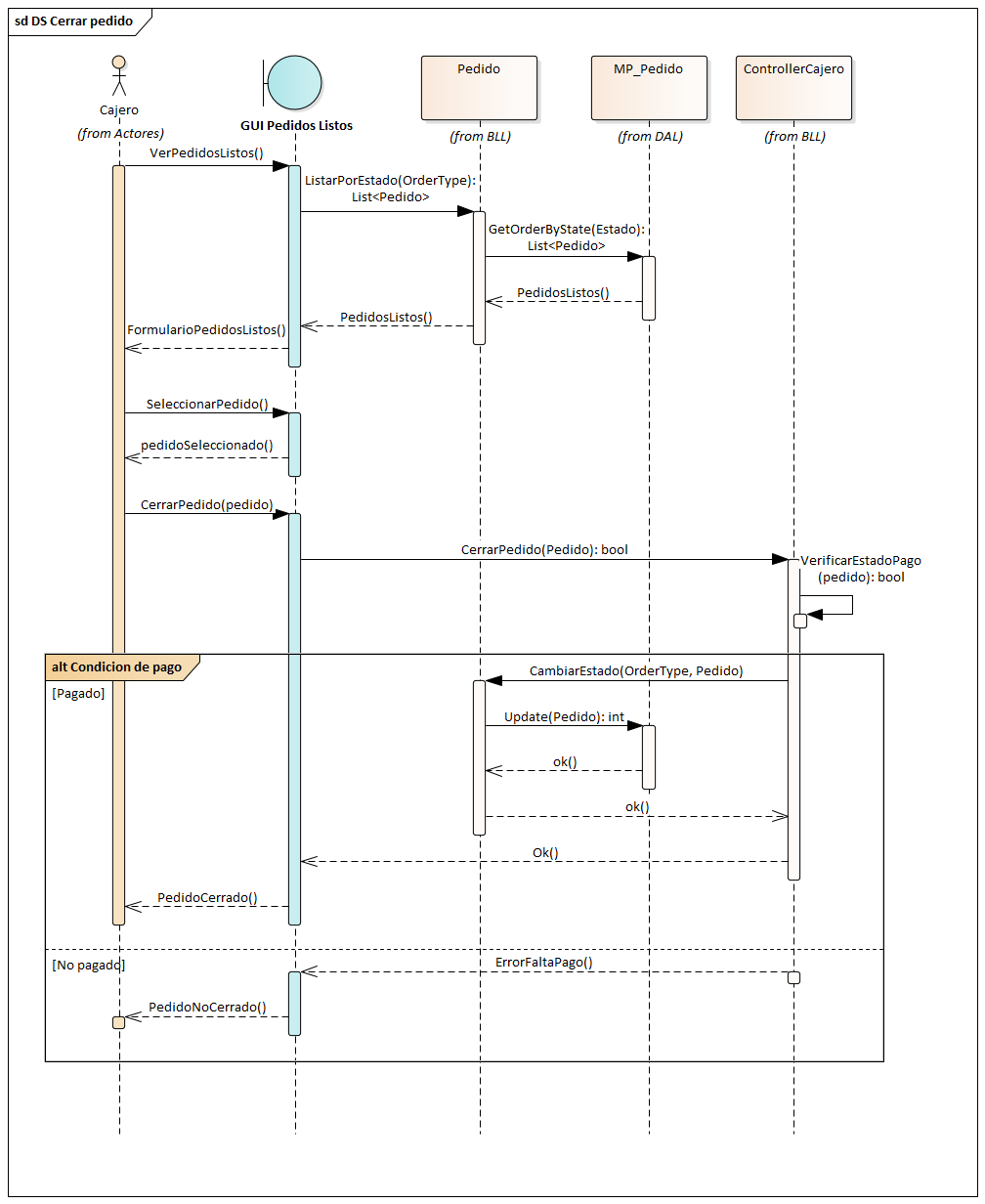
|  |
| --- |
| **ID y Nombre**: CU06 Cerrar pedido |
| **Estado**: Pendiente. |
| **Descripción**: EL usuario verifica el pedido por ultima vez y lo da por terminado. |
| **Actor Principal**: Cajero |
| **Actor Secundario**: |
| **Puntos de Extensión**: |
| **Escenario Principal**:   1. El cajero ingresa a la sección de “Pedidos Listos”. 2. El sistema muestra una lista de todos los pedidos marcados como “listo”. 3. El cajero selecciona el pedido que quiera entregar y selecciona “Cerrar pedido”. 4. El sistema procesa la solicitud e indica que el pedido ya puede ser entregado. |
| **Flujos Alternativos**:   * 1. Mensaje de error: Por favor seleccione un pedido a cerrar.   2. Mensaje de error: Error al cerrar el pedido. No ha sido pagado. |

#### Diagrama de clase

A screenshot of a computer

Description automatically generated

#### Diagrama de secuencia



#### DER

A screenshot of a computer

Description automatically generated

#### Prototipo de interfaz de usuario

A screenshot of a computer

Description automatically generated

## N02.B Especificaciones de casos de uso del sistema para “Comprar insumos”

### Diagrama general de casos de uso

A diagram of a diagram

Description automatically generated

### Caso de uso del sistema CU-RFN2-01 Buscar insumos faltantes

#### Historial de revisión de la especificación

|  |  |  |  |
| --- | --- | --- | --- |
| Fecha | Versión | Descripción | Autor |
| 30/09/2024 | 1.0 | Versión inicial | Daniel Jiménez |

#### Objetivo

Establecer la cantidad de insumos faltantes del stock para generar la solicitud de compra.

#### Precondiciones

El usuario ya inició sesión en el sistema.

#### Postcondiciones

Se determinaron los insumos faltantes en el inventario.

#### Eventos disparadores del caso de uso

El jefe de cocina detecta faltantes en el inventario físico de insumos.

#### Diagrama del caso de uso

A diagram of a person with a circle and text

Description automatically generated

#### Descripción analítica del caso de uso

|  |
| --- |
| **ID y Nombre**: CU06 Cerrar pedido |
| **Estado**: Pendiente. |
| **Descripción**: EL usuario verifica el pedido por ultima vez y lo da por terminado. |
| **Actor Principal**: Cajero |
| **Actor Secundario**: |
| **Puntos de Extensión**: |
| **Escenario Principal**:   1. El cajero ingresa a la sección de “Pedidos Listos”. 2. El sistema muestra una lista de todos los pedidos marcados como “listo”. 3. El cajero selecciona el pedido que quiera entregar y selecciona “Cerrar pedido”. 4. El sistema procesa la solicitud e indica que el pedido ya puede ser entregado. |
| **Flujos Alternativos**:   * 1. Mensaje de error: Por favor seleccione un pedido a cerrar.   2. Mensaje de error: Error al cerrar el pedido. No ha sido pagado. |

T00. Documentos de aspectos técnicos que provee el sistema de información.

# T01. Arquitectura Base

Para el desarrollo del software se decidió utilizar una arquitectura de 6 capas, siendo estas las que se muestran en el siguiente diagrama:

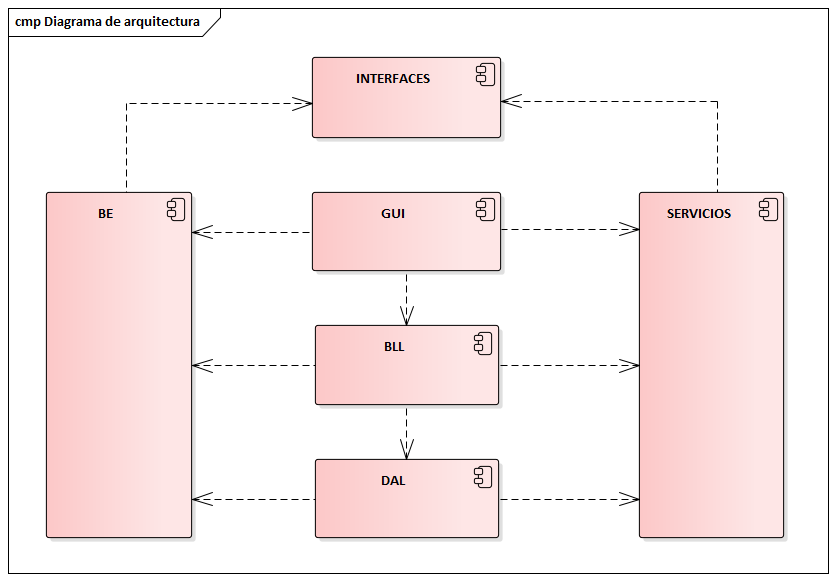


Figura 1 Diagrama de arquitectura

El uso de este tipo de arquitectura permite tener un software con bajo acoplamiento entre sus componentes, y una alta cohesión. Las capas son las que se describen a continuación:

**GUI:** Es la capa de presentación. Aquí se encuentran todos los componentes referidos a la interfaz gráfica con la que van a interactuar los usuarios directamente. Esta capa depende de la BE, la BLL y la capa de servicios.

**BE:** Es la capa de dominio. Se encuentran todas las entidades del dominio que serán utilizadas por las demás, por lo que todas dependen de esta mientras que ella solo depende de la capa de interfaces.

**INTERFACES**: Acá se encuentran interfaces que contienen atributos que no son propio del dominio pero que igual deben ser implementados por las entidades. Esta capa no depende de nadie, y de ella dependen la BE y la capa de servicios.

**BLL:** Es la capa de aplicación o negocio. Se encuentran todos los procesos y métodos referidos a las reglas del negocio del sistema. Esta capa depende también de la BE, la de servicios y la DAL.

**SERVICIOS:** Esta es una capa donde se encuentran todos los métodos que debe tener el sistema que no son referidos a las reglas del negocio. Se encuentra la logica del inicio y cierre de sesión, y principalmente se encuentra todo lo referido a la seguridad de los datos. Todas las capas dependen de esta, pero esta depende de la capa de interfaces.

**DAL:** Es la capa de acceso a datos. En esta capa se encuentra todo lo referente al acceso directo de la base de datos. Esta capa también depende de la BE y de la de servicios.

## Persistencia

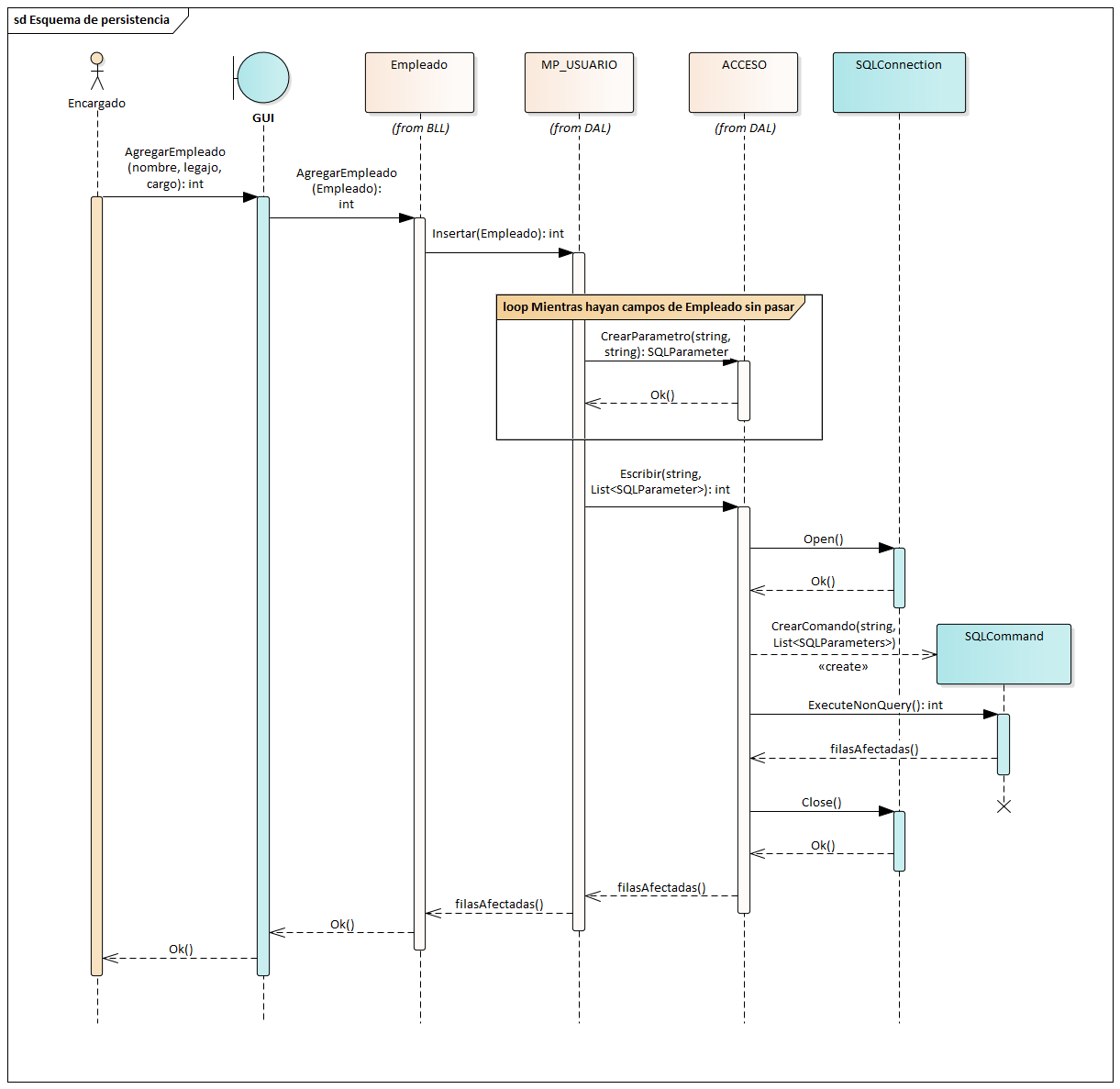
Con el objetivo de persistir los datos, se emplea el sistema de ADO desconectado con SQL server 16.01115.1.

Dentro de la capa de acceso de datos, tendremos una clase ACCESO la cual se encargará de toda la comunicación interna entre las clases de ADO.NET. La conexión y ejecución de comandos sobre la base de datos se realizará por medio de procedimientos almacenados con pase de parámetros. Se tendrá una clase abstracta MAPPER, la cual será implementada por todos los mappers de cada entidad que se desee persistir en la base de datos. Dichos mappers son clases que se encargan de hacer la transformación entre una entidad del modelo relacional al modelo orientado a objetos y viceversa. Estos tendrán 5 métodos fundamentales:

* Insertar
* Listar por id
* Listar todos
* Actualizar
* Modificar

Al ser ADO desconectado, implica que los datos son leídos de la base de datos, se almacenan en un objeto DataTable a través del método Fill de la clase SQLDataAdapter y son manipulados luego de forma offline. Por lo que el tiempo que estamos ocupando la base de datos es muy bajo, logrando un mayor rendimiento al evitar una conexión activa con dicha base.

A continuación, se presenta un par de ejemplos de cómo sería la secuencia para realizar una escritura sobre la base de datos:



A diagram of a project

Description automatically generated

## Mapa de navegación

## Explorador de soluciones

A screenshot of a menu

Description automatically generated

## Menú principal

A screenshot of a computer

Description automatically generated

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

## Objetivo

Garantizar la seguridad y la privacidad del sistema al verificar la identidad de los usuarios, mediante el ingreso de un nombre de usuario junto a su contraseña y controlar el acceso a las utilidades del sistema en base al perfil asignado a dichos usuarios. Se hará uso del patrón singleton para mantener la sesión.

## Descripción detallada de cómo funciona

Con el fin de poder acceder a la mayoria de las funciones del sistema, el usuario deberá proveer sus credenciales al sistema (usuario y contraseña).

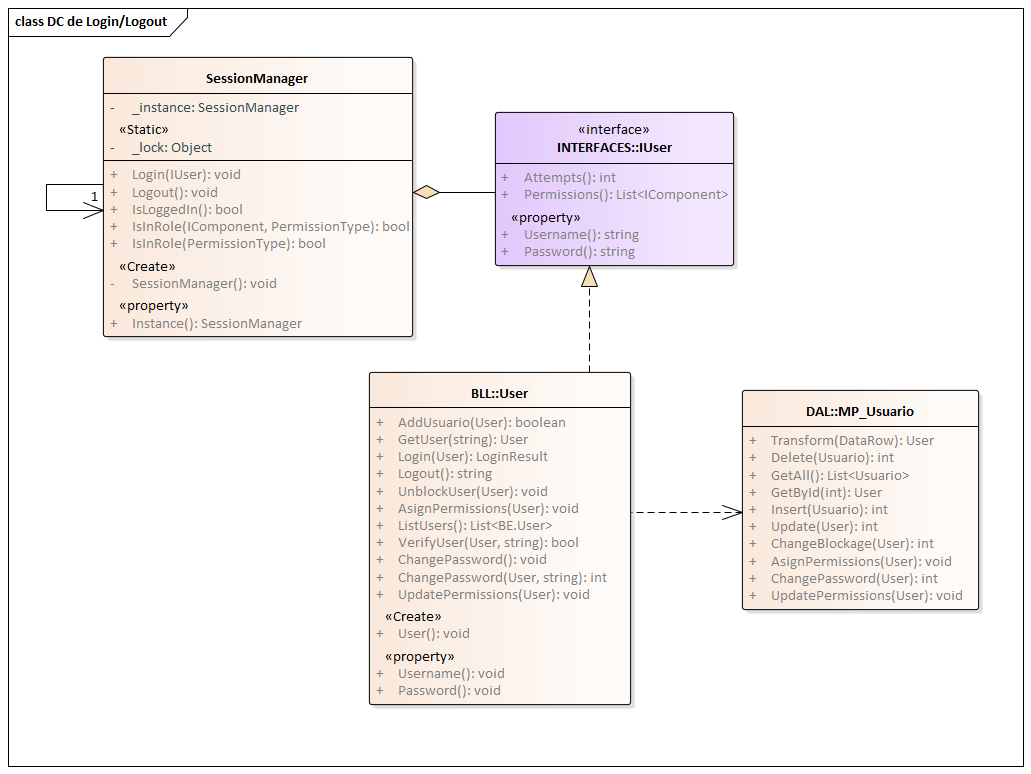
El sistema inicia verificando que el nombre de usuario exista en la base de datos, y luego procede a verificar el estado del usuario. Este último puede ser tanto activo como bloqueado, siendo el estado inicial como “activo”.

Cada vez que se introduce la contraseña de forma incorrecta se aumenta un contador en 1 (inicia en 0), y cuando llega a 3 el estado pasa a ser “bloqueado” y no tendrá permitido realizar más intentos de inicios de sesión. En este caso deberá contactar a un administrados para reestablecer su usuario.

Retomando los pasos que sigue el sistema, luego de la verificación del estado como “activo” procede a verificar la contraseña. Para esto, encripta con un HASH a la contraseña introducida y la verifica con el valor guardado en la base de datos. Y si coinciden el usuario tendrá acceso al sistema correctamente con los permisos correspondientes.

Finalmente, una vez dentro del sistema, el usuario tendrá la opcion de cerrar su sesión y salir del sistema.

## Diagrama de clases



## Diagrama de secuencia Login

A computer screen shot of a diagram

Description automatically generated

## Diagrama de secuencia Logout

A computer screen shot of a diagram

Description automatically generated

# T03. Gestión de Encriptado

## Objetivo

Encriptar y desencriptar informacion sensible de los usuarios que se desee persistir en la base de datos del sistema.

## Descripción detallada de cómo funciona

Para proteger los datos importantes del sistema se utilizarán dos formas de encriptación diferentes. Un cifrado del tipo HASH para las contraseñas de los usuarios del sistema, y un metodo de cifrado simétrico para los demás datos que puedan necesitar recuperarse luego del encriptado.

Para el encriptado de las contraseñas, como ya mencionamos antes, se utilizará una función HASH. Específicamente, se utilizará el algoritmo criptográfico de 256 bits de salida SHA-256(Secure Hash Algorithm 256), el cual ofrece un buen equilibrio entre seguridad y rendimiento, siendo suficientemente rápido para muchas aplicaciones mientras mantiene un alto nivel de seguridad.

Como metodo de cifrado simétrico se utilizará la especificación de encriptación de data electrónica establecida por el instituto nacional de estándares y tecnologia de estados unidos (NIST), siendo este Estándar de Encriptación Avanzada (AES). Este funcionará mediante una llave única privada para la encriptación la cual será almacenada de forma segura en el equipo principal del restaurante. El modo de cifrado será el CBC (del inglés Cipher Block Chainning). Este funciona de la siguiente manera:

Antes de cada bloque de texto plano es encriptado, se combina con el texto cifrado del bloque anterior mediante una operación XOR. Asegurando que incluso si el texto contiene bloques idénticos, cada uno se encriptara a un diferente bloque de texto cifrado.

En AES también se incluye, ademas de la llave, un vector de inicialización, el cual agrega una capa extra de seguridad y este se combina con el primer bloque mediante un XOR antes de que este sea encriptado.

## Diagrama de clases

A screenshot of a computer program

Description automatically generated

# T04. Gestión de perfiles de usuario

## Objetivo

Permitir un control de acceso detallado y organizado, mejorando la seguridad al restringir funciones y datos sensibles a usuarios autorizados, y facilitando la administración al agrupar permisos en roles específicos. Este enfoque flexible y escalable simplifica la asignación y revocación de permisos, manteniendo una estructura clara y permitiendo una fácil delegación de responsabilidades dentro del sistema.

## Descripción detallada de cómo funciona

El sistema clasifica las acciones que puede realizar un usuario a través de permisos, y grupos (roles). Cada función del sistema tiene un permiso relacionado, y estos son asignados a los distintos usuarios dependiendo del rol que tengan por cumplir dentro del sistema.

Los grupos son, como su nombre lo indica, agrupaciones de permisos los cuales también se pueden asignar a los usuarios para simplicidad. Tanto los grupos como los permisos son considerados el mismo tipo de componente, por lo que cada usuario posee internamente un conjunto de grupos y permisos asociados a él.

Cada permiso o grupo se encuentra identificado unívocamente por un ID, y ademas, poseen un nombre.

En cuanto a la interacción usuario-sistema, al momento de iniciar sesión, se trae de la base de datos el conjunto de permisos y grupos que tiene asociado dicho usuario y en base a eso se le habilitan los módulos correspondientes del software.

El rol mas importante es el **admin**, siendo este quien posee acceso total a las funcionalidades del sistema y quien se encarga inicialmente de asignarle los permisos a los demás usuarios. Si bien los permisos atómicos, digase los permisos que habilitan individualmente una sección del software, son fijos, los grupos se pueden crear y asignarle los permisos y otros grupos que se ameriten.

Dentro del sistema, en total, se encuentran 16 permisos atómicos y 5 grupos. Son los siguientes:

|  |  |  |
| --- | --- | --- |
| **ID** | **NOMBRE** | **TIPO** |
| 1 | CrearPedido | P |
| 2 | VerPedidos | P |
| 3 | CobrarPedido | P |
| 4 | Cajero | G |
| 5 | JefeDeCocina | G |
| 6 | Admin | G |
| 7 | VerProductos | P |
| 8 | VerIngredientes | P |
| 9 | NotificarPedidoListo | P |
| 10 | VerificarPedido | P |
| 11 | GenerarComanda | P |
| 12 | VerComanda | P |
| 13 | GestorUsuario | P |
| 14 | GestorPerfil | P |
| 15 | GestorIdioma | P |
| 16 | GestionarAdmin | P |
| 17 | GestionarCatalogos | P |
| 18 | GestionarPedido | P |
| 19 | Encargado | G |
| 20 | Cocinero | G |
| 21 | GestionarComanda | P |
| 25 | VerPedidosRegistrados | P |
| 26 | VerPedidosEnCurso | P |
| 27 | VerPedidosCerrados | P |
| 28 | VerPedidosVerificados | P |
| 29 | VerPedidosListos | P |

Los grupos definidos por ahora son los siguientes:

* **Cajero:**
  + CrearPedido
  + VerPedidos
  + CobrarPedido
  + VerProductos
  + GestionarCatalogos
  + GestionarPedido
  + VerPedidosRegistrados
  + VerPedidosEnCurso
  + VerPedidosCerrados
  + VerPedidosVerificados
  + VerPedidosListos
* **JefeDeCocina:**
  + VerPedidos
  + VerProductos
  + VerIngredientes
  + NotificarPedidoListo
  + VerificarPedido
  + GenerarComanda
  + VerComanda
  + GestionarCatalogos
  + GestionarPedido
  + GestionarComanda
  + VerPedidosRegistrados
  + VerPedidosEnCurso
  + VerPedidosCerrados
  + VerPedidosVerificados
  + VerPedidosListos
* **Admin:**
  + **Cajero**
  + **JefeDeCocina**
  + GestorUsuario
  + GestorPerfil
  + GestorIdioma
  + GestionarAdmin
* **Cocinero**
  + VerPedidos
  + NotificarPedidoListo
  + VerComanda
  + GestionarPedido
  + GestionarComanda
  + VerPedidosEnCurso

El grupo “Encargado” todavía está por definir.

## Diagrama de clases

A diagram of a computer

Description automatically generated with medium confidence

## DER

A screenshot of a computer

Description automatically generated

# T05. Gestión de Múltiples Idiomas

## Objetivo

Proporcionar una experiencia de usuario adaptable y accesible para una audiencia diversa que habla diferentes idiomas por medio del cambio dinámico del lenguaje de la interfaz de usuario.

## Descripción detallada de cómo funciona

El sistema le asigna por default a cada un usuario un idioma a través del cual interactuara con la interfaz grafica del software.

Para la implementación de este sistema de traducción automático se utiliza el patrón observer por medio de la interfaz **IIdiomaObserver**, la cual es implementada por cada formulario del sistema. En este caso, el papel del observado lo maneja el **SessionManager** implementando los métodos:

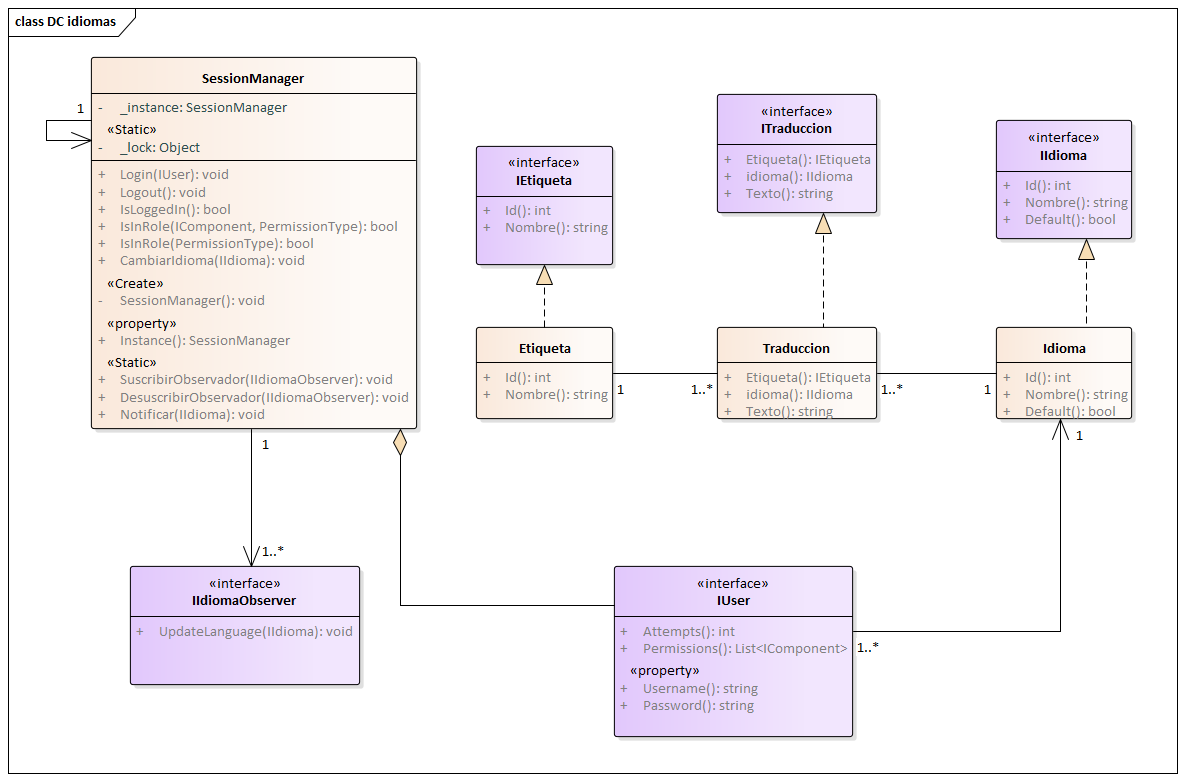
* **SuscribirObservador** (IIdiomaObserver): void
* **DesuscribirObservador** (IIdiomaObserver): void
* **Notificar** (IIdioma): void
* **CambiarIdioma** (IIdioma): void

Al cargar el form principal, este verifica si el usuario está loggeado o no, con el fin de cargar el idioma por default del sistema o, en caso de que haya iniciado sesión, se carga el idioma de preferencia del usuario.

El cambio de idioma se maneja de forma dinámica por medio del menú del form principal. A la opcion “**Cambiar idioma**” se le agrega en tiempo de ejecución la lista de idiomas guardados en base de datos, y al seleccionar alguno, el **SessionManager** notifica a cada form suscrito.

El proceso de traducción en sí se maneja por medio de etiquetas o “tags” que les son asignados a cada control que contenga texto dentro de los formularios. Estas etiquetas se encuentran asociadas dentro de la base de datos y poseen una traducción por cada idioma que maneje el sistema, así, al detectar el idioma actual al que se quiere traducir, el sistema trae de la base de datos la traducción adecuada para cada control.

## Diagrama de clases



## DER

A screenshot of a computer

Description automatically generated

# T06. Gestión de Bitácora y Control de cambios

## T06a. Gestión de Bitácora

### Registro de eventos (Backend)

El objetivo de una bitácora de eventos en un sistema es registrar todas las operaciones realizadas por los usuarios, permitiendo trazar sus actividades dentro de la aplicación. Esto incluye detalles como fecha, hora, usuario, actividad y la información relacionada, permitiendo la búsqueda combinada de estos datos.

#### Descripción detallada de cómo funciona

1. Un usuario realiza una acción sensible sobre el sistema.
2. El sistema genera un registro particular para dicha acción, donde recopila los siguientes datos: Fecha, Hora, Usuario, Operación, Módulo donde se realizó la operación, y Criticidad de la operación (dicha criticidad podrá tomar valores entre 1 y 5, donde 1 es la mayor y 5 la menor).
3. El sistema guarda automáticamente este registro en una tabla destinada a fines de control y auditoria, ningún otro proceso opera sobre ella para dar veracidad a los datos.

#### Diagrama de secuencia de registro de eventos.

Para mostrar la secuencia de operaciones que ocurren al momento de registrar un evento, se utilizara el siguiente diagrama de secuencia donde el administrador le da de alta a un usuario.

A diagram of a project

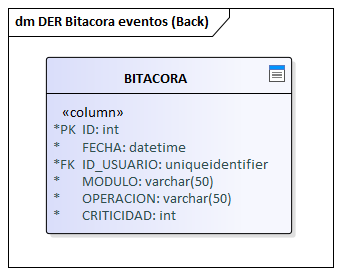
Description automatically generated

#### Diagrama de clases

A screenshot of a computer program

Description automatically generated

#### DER



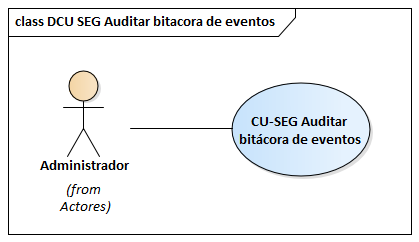
### Auditoria de eventos (Frontend)

La bitácora de eventos tiene como finalidad poder realizar auditorías de los eventos. Donde los únicos usuarios con acceso a estos registros serán los administradores, y en los que solamente podrán realizar operaciones de consultas en una pantalla.

#### Descripción detallada de cómo funciona

1. El administrador ingresa al módulo de Bitácora de eventos.
2. El sistema muestra en pantalla una grilla con todos los registros de la bitácora sin filtrar de cada una de las acciones tomadas en el sistema, conformadas por los siguientes campos: *Fecha de ocurrencia, usuario, modulo, operación, criticidad.*
3. El administrador podrá aplicar filtros en base a las siguientes opciones:
   1. Fecha de inicio
   2. Fecha de fin
   3. Usuario
   4. Modulo
   5. Operación
   6. Criticidad
4. El sistema, en base a los filtros aplicados, buscara en la base de datos y mostrara los registros de la bitácora que coincidan con dichos filtros.
5. Con los resultados, el administrador podrá imprimir todos los registros mostrados de la bitácora.

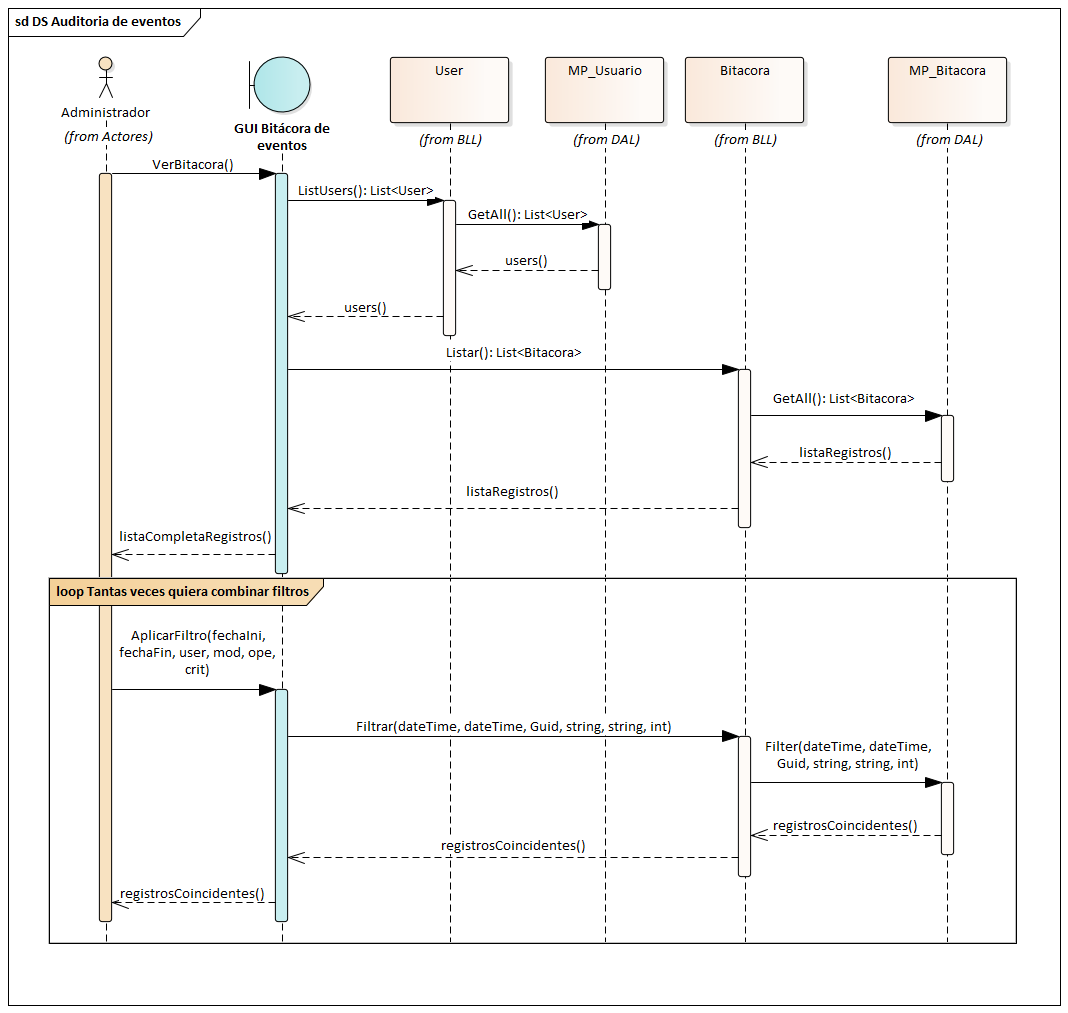
#### Diagrama del caso de uso



#### Especificación del caso de uso

|  |
| --- |
| **ID y Nombre**: CU-SEG Auditar bitácora de eventos |
| **Estado**: Pendiente. |
| **Descripción**: Se consultan los registros de la bitácora de eventos. |
| **Actor Principal**: Administrador. |
| **Actor Secundario**: - |
| **Precondiciones**: El usuario ha ingresado al sistema. |
| **Puntos de Extensión**: - |
| **Condición**: - |
| **Escenario Principal**:   1. El administrador ingresa al módulo de Bitácora de eventos. 2. El sistema muestra el módulo de la bitácora de eventos junto a todos sus registros actuales, permitiendo filtrar bajo distintos criterios (fecha de inicio, fecha de fin, usuario, modulo, operación y criticidad). 3. El administrador aplica los filtros que desee y selecciona “consultar”. 4. El sistema muestra los registros que coincidan con los filtros aplicados. |
| **Flujos Alternativos**:   * 1. No hay registros coincidentes. Vuelve al paso 3. |
| **Postcondiciones**: Se Auditó la bitácora de eventos. |

#### Diagrama de secuencia



#### Diagrama de clases

A diagram of a computer program

Description automatically generated with medium confidence

#### DER

A screenshot of a computer

Description automatically generated

#### GUI

A screenshot of a computer

Description automatically generated

## T06b. Control de cambios

### Registro de cambios (Backend)

#### Descripción detallada de cómo funciona

#### Aplicación de triggers para guardar los cambios.

#### Diagrama de actividad

### Auditoria de cambios (Frontend)

#### Descripción detalla de cómo funciona

#### Diagrama del caso de uso

#### Especificación del caso de uso

#### Diagrama de secuencia

#### Diagrama de clases

#### DER

#### GUI

# T07. Gestión de respaldos

## Descripción detallada de cómo funciona

1. El administrador ingresa al menú “Administrador”.
2. El sistema muestra las opciones que puede realizar en base a su perfil de Admin.
3. El administrador selecciona la opción de “Gestionar respaldos”
4. El sistema muestra una pantalla con:
   1. Botón para examinar y escoger una ruta para guardar una copia de seguridad de la base de datos.
   2. Botón para generar la copia de seguridad en la ruta escogida.
   3. Botón para examinar y escoger la ruta del archivo de copia de seguridad del cual se quiere restaurar la base de datos.
   4. Botón para restaurar la base de datos con el archivo escogido.
5. El administrador selecciona la ruta donde se va a generar la copia.
6. El sistema en base a la ruta escogida genera la copia de seguridad, identificándola con el nombre de la base de datos (TpRestaurante) y la fecha de creación en el formato: ddMMYY\_hhmm.
7. Ahora el administrador puede escoger la ruta donde se encuentra el archivo para restaurar la base de datos.
8. El sistema restaura la base datos utilizando el archivo escogido.

## Diagrama de caso de uso

A diagram of a person's work flow

Description automatically generated

## Especificación del caso de uso

|  |
| --- |
| **ID y Nombre**: CU-SEG Gestionar respaldos |
| **Estado**: Pendiente. |
| **Descripción**: Se genera una copia de seguridad de la base de datos y luego se restaura con una copia. |
| **Actor Principal**: Administrador |
| **Actor Secundario**: - |
| **Precondiciones**: El usuario ha iniciado sesión en el sistema. |
| **Puntos de Extensión**: - |
| **Condición**: - |
| **Escenario Principal**:   1. El administrador ingresa al módulo de gestión de respaldos. 2. El sistema muestra las opciones de generar una copia de seguridad y de restaurar la base de datos. 3. El administrador decide realizar generar una copia de seguridad, por lo que selecciona “Examinar” y escoge la ruta en la cual se creará la copia. 4. El sistema genera la copia de seguridad. 5. Ahora el administrador decir restaurar la base de datos, por lo que seleccionar “Examinar” y escoge el archivo de la copia de seguridad que utilizará para restaurar la base de datos. 6. El sistema restaura la base de datos. |
| **Flujos Alternativos**:   * 1. El administrador no desea realizar una copia de seguridad, pero sí restaurar la base de datos. Salta al paso 5.   2. El administrador no desea restaurar la base de datos. Solo generar una copia de seguridad. Finaliza el proceso. |
| **Postcondiciones**: Se generó una copia de seguridad y se restauró la base de datos. |

## Diagrama de secuencia

A screenshot of a computer

Description automatically generated

## Diagrama de clases

A diagram of a computer

Description automatically generated

## DER

A close-up of a computer screen

Description automatically generated

## GUI

A screenshot of a computer

Description automatically generated

A00. Características y funcionalidades adicionales

# A03. Serialización

La serialización tiene como objetivo principal guardar el estado de un objeto o lista de objetos en un archivo externo a la base de datos y poder consultarlo cuando sea necesario.

## Descripción funcional

1. Al ser la serialización y deserialización una extensión del modulo de maestro de clientes, el usuario ingresa a dicho módulo.
2. El sistema muestra una grilla con todos los clientes que se encuentren en estado *Activo*. Los campos que muestra son: *ID, Nombre, apellido, DNI, teléfono*. También, en modo Consulta muestra los filtros disponibles a aplicar sobre la grilla:
   1. Nombre
   2. Apellido
   3. DNI
   4. Teléfono
   5. Estado Activo
   6. Todos
3. El usuario aplica los filtros que desee. (*Nombre, apellido, DNI, telefono, activo*).
4. El sistema muestra los registros coincidentes en la grilla en caso de que existan.
5. El usuario podra escoger la ruta donde quiere guardar el archivo en donde se almacenarán el/los objeto/s a serializar y selecciona “serializar”.
6. El sistema genera un archivo XML con los objetos serializados.
7. Ahora, el usuario también tiene la opción de deserializar objetos almacenados en un archivo XML. Por lo que, en caso de querer realizar esta operación, selecciona la ruta del archivo que desea utilizar.
8. El sistema muestra en la grilla los datos de los objetos que están presentes en el archivo XML.

## Diagrama de caso de uso

A diagram of a structure

Description automatically generated

## Especificación del caso de uso

|  |
| --- |
| **ID y Nombre**: CU-EX Serializar/Deserializar |
| **Estado**: Pendiente. |
| **Descripción**: Se serializa una lista de objetos o se deserializar un archivo con una lista de objetos. |
| **Actor Principal**: Encargado |
| **Actor Secundario**: - |
| **Precondiciones**: El usuario ha iniciado sesión en el sistema |
| **Puntos de Extensión**: - |
| **Condición**: - |
| **Escenario Principal**:   1. El encargado ingresa al modulo de maestro de clientes. 2. El sistema muestra la lista completa de todos los clientes activos en el sistema. A su vez, muestra filtros que se pueden aplicar sobre dicha lista (activos, todos, nombre, apellido, DNI, telefono). 3. El encargado aplica los filtros que desee utilizar. 4. El sistema muestra los resultados coincidentes con la búsqueda. 5. El encargado selecciona la ruta donde va a guardar el archivo con la lista de objetos a serializar y selecciona “Serializar”. 6. El sistema genera el archivo serializado. 7. El encargado ahora selecciona la ruta de un archivo que quiere deserializar, y selecciona “Deserializar”. 8. El sistema deserializar el archivo y muestra en pantalla los datos de los objetos que estaban serializados. |
| **Flujos Alternativos**:   * 1. No hay coincidencias. Vuelve al paso 3. |
| **Postcondiciones**: Se serializó y deserializó un archivo. |

## Diagrama de secuencia

A screenshot of a computer

Description automatically generated

## Diagrama de clases

A diagram of a computer

Description automatically generated

## DER

A screenshot of a computer

Description automatically generated

## GUI

