

Software Gestor de Restaurantes (Soft Restaurant).

Descripción de la Arquitectura – Diseño del Software.

IEEE 1471-2000 *Recommended Practice for Architectural Description of Software.* Descripción de la Arquitectura - Diseño de Software.

Versión [1.0]

Historia de revisiones

Fecha	Versión	Descripción	Autor(es)
20/10/2021	1.0	Primera revisión del documento	Equipo de Desarrollo

Contenido

SOFTWARE GESTOR DE RESTAURANTES (SOFT RESTAURANT).	1
DESCRIPCIÓN DE LA ARQUITECTURA – DISEÑO DEL SOFTWARE.	1
IEEE 1471-2000 RECOMMENDED PRACTICE FOR ARCHITECTURAL DESCRIPTION OF SOFTWARE. DESCRIPCIÓN DE LA ARQUITECTURA - DISEÑO DE SOFTWARE.	1
VERSIÓN [1.0]	1
HISTORIA DE REVISIONES	1
1. INTRODUCCIÓN	5
1.1. PROPÓSITO	5
1.2. ALCANCE	5
1.3. DEFINICIONES, SIGLAS Y ABREVIATURAS	5
1.4. REFERENCIAS	6
1.5. VISIÓN GENERAL	6
2. VISTA DEL MODELO DE CASOS DE USO	6
2.1. DIAGRAMA DE CASOS DE USO RELEVANTES A LA ARQUITECTURA	6
2.2. CASOS DE USO RELEVANTES A LA ARQUITECTURA	6
2.2.1. Registrar una orden	8
2.2.2. Edición de una orden	8
2.2.3. Cerrar una orden	9
2.2.4. Consultar una orden y su contenido	10
2.2.5. Modificar el estatus de una orden (contenido)	11
2.2.6. Alta de usuario	11
2.2.7. Edición de usuario	12
2.2.8. Borrado o baja de usuario	12
2.2.9. Alta de platillo	13
2.2.10. Edición de platillo	13
2.2.11. Baja o eliminación de platillo	14
3. TRAZABILIDAD DESDE EL MODELO DE CASOS DE USO AL MODELO DE DISEÑO	14
3.1.1. Registro de ordenes	14
3.1.2. Edición de ordenes	15
3.1.3. Gestión de usuarios	15
3.1.4. Gestión de platillos	16
4. VISTA DEL MODELO DE DISEÑO	16
4.1. DESCOMPOSICIÓN EN SUBSISTEMAS	16
4.1.1. Subsistema de gestión de dudas y sugerencias	16
4.1.2. Subsistema de gestión de empleados	17
4.1.3. Subsistema de gestión de usuarios	17
4.1.4. Subsistema de gestión de ordenes	17
4.1.5. Subsistema de gestión de platillos	18
4.2. DISEÑO DE CLASES	19
4.2.1. Clase Usuarios	19
4.2.2. Clase Seguridad	20
4.2.3. Clase Ordenes	21
4.2.4. Clase Orden_Detalles	22
4.2.5. Clase Orden_Estados	22
4.2.6. Mesas	23
4.2.7. Platillos	23
4.3. DISEÑO DE OBJETOS	24
4.3.1. Objeto Usuario	24
4.3.2. Objeto Orden	25
4.4. DIAGRAMAS DE ESTADOS	26

4.4.1.	Diagrama de Estados de Orden	26
4.4.2.	Diagrama de Estados Usuario	27
4.5.	DIAGRAMAS DE SECUENCIAS	27
4.5.1.	Diagrama de secuencias Alta de una orden.	27
4.5.2.	Diagrama de secuencias Consultar una orden.....	28
4.5.3.	Diagrama de secuencias Cerrar una orden.....	28
4.5.4.	Diagrama de secuencias Modificar una orden.....	28
4.5.5.	Diagrama de secuencias Modificar el estatus de una orden.	28
4.5.6.	Diagrama de secuencias Dar de alta un nuevo usuario.	29
4.5.7.	Diagrama de secuencias de edición de usuario.	29
4.5.8.	Diagrama de secuencias de baja de usuario.	29
4.5.9.	Diagrama de secuencias de Alta de platillo.	31
4.5.10.	Diagrama de secuencias de Edición de platillo.	31
4.5.11.	Diagrama de secuencias de Baja de platillo.	32
4.6.	DIAGRAMAS DE ACTIVIDADES	33
4.6.1.	Diagrama de actividad Registrar Orden.	33
4.6.2.	Diagrama de actividad Consultar una orden.	34
4.6.3.	Diagrama de actividad Cerrar una orden.	35
4.6.4.	Diagrama de actividad Modificar una orden.	36
4.6.5.	Diagrama de actividad Modificar el estatus de una orden.	37
4.6.6.	Diagrama de actividad Dar de alta un nuevo usuario.	38
4.6.7.	Diagrama de actividad de edición de usuario.	39
4.6.8.	Diagrama de actividad de baja de usuario.	40
4.6.9.	Diagrama de actividad de Alta de platillo.....	41
4.6.10.	Diagrama de actividad de Edición de platillo.	42
4.6.11.	Diagrama de actividad de Baja de platillo.	43
5.	TRAZABILIDAD DESDE EL MODELO DE DISEÑO AL MODELO DE IMPLEMENTACIÓN.....	44
6.	VISTA DEL MODELO DE IMPLEMENTACIÓN.....	45
6.1.	SUBSISTEMAS	45
6.1.1.	Gestión de dudas y sugerencias.....	45
6.1.2.	Gestión de empleados.....	45
6.1.3.	Gestión de usuarios	45
6.1.4.	Gestión de ordenes	45
6.1.5.	Gestión de platillos.....	45
6.2.	COMPONENTES	46
6.2.1.	Aplicación web	46
6.2.2.	Aplicación de escritorio.....	46
6.3.	INTERFACES	46
6.3.1.	Interfaz web	46
6.3.2.	Interfaz de usuario.....	46
6.3.3.	Interfaz de administrador.....	46
7.	VISTA DEL MODELO DE DISTRIBUCIÓN	47
7.1.	DIAGRAMA DE DISTRIBUCIÓN	47
7.2.	NODOS	47
8.	DESCRIPCIÓN DE DISEÑO DETALLADO	50
8.1.	DISEÑO DETALLADO DEL ELEMENTO 1	50
8.1.1.	Vistas estructurales	50
8.1.2.	Vistas de comportamiento.....	50
8.1.3.	Otras vistas relevantes.....	51
8.1.4.	Razón fundamental.....	51
8.2.	DISEÑO DETALLADO DEL ELEMENTO 2	51
8.2.1.	Vistas estructurales	51
8.2.2.	Vistas de comportamiento.....	51
8.2.3.	Razón fundamental	51

1. Introducción

Los elementos de la Arquitectura de Software típico son: descripción de procesos, actividades, tareas, roles y productos, plantillas, lista de verificación, ejemplo y referencia a estándares y modelos, y herramientas.

El propósito de este documento es proveer con guías y materiales personalizables y fáciles de usar para poder implementar un buen *Diseño de Software*.

Invertir esfuerzo en las **actividades** de diseño asegura que para la solución propuesta (por ejemplo, software a construir) se habrá meditado alguna idea previa a su implementación (por ejemplo, codificación). Construir algo sin diseñarlo típicamente produce una solución que no cumple con los requerimientos, se entrega tarde, excede el presupuesto o es de baja calidad.

Invertir esfuerzos en **documentación** explícita habilita la comunicación y negociación entre los stakeholders del proyecto, más específicamente a aquellos que tienen interés en el diseño. Capturando un diseño en alguna forma (documento electrónico, documento en papel, modelos...) no es solo útil mientras el proyecto de software está activo, sino también para el futuro mantenimiento y mejoras.

La actividad de Arquitectura de Software y Diseño Detallado produce un documento denominado el *Diseño de Software* que permite a los stakeholders entender las interacciones en el software y la trazabilidad de los elementos diseñados hacia los requerimientos. Esto permite una forma de verificar que cada requerimiento ha sido atendido (por ejemplo, completitud de diseño). El Diseño de Software también se usa para el mantenimiento del software porque describe los componentes y sus interfaces.

1.1. Propósito

Este documento proporciona una apreciación global y comprensible de la Arquitectura - Diseño del Software usando diferentes puntos de vista para mostrar distintos aspectos del sistema. Intenta capturar y llegar a las decisiones de arquitectura críticas que han sido hechas en el sistema.

Así mismo también como dispersar cualquier duda acerca de la construcción, definición, desarrollo del proyecto.

1.2. Alcance

El sistema de gestión y control Soft Restaurant tiene como objetivo ayudar a la gestión del restaurante "Tacos plaza" con el manejo de órdenes, control de empleados, platillos en el menú, platillos en ordenes, control de inventarios, y emisión de comprobantes de compras estilo tickets.

El alcance de este proyecto incluye y excluye los siguientes elementos:

1.2.1. Dentro del Alcance:

- Gestión de platillos
- Generación de menús
- Administración de cuentas

1.2.2. Fuera del Alcance:

- Renovación de contratos con empresas proveedoras
- Diseño de los menús
- Decisión sobre contratos de inventario

1.3. Definiciones, siglas y abreviaturas.

- ERS: Especificación de Requerimientos de Software.
- IEEE: Instituto de Ingeniería Eléctrica y Electrónica.
- SACP: Sistema de Administración de Clientes y Proveedores.
- SO: Sistema Operativo.
- RF: Requerimientos Funcionales.
- RNF: Requerimientos No Funcionales.
- RAM: Random Access Memory.
- Stakeholders: Personas, grupos y entidades que tienen intereses de cualquier tipo en una empresa.
- Trazabilidad: Serie de procedimientos que permiten seguir el proceso de evolución de un producto en cada una de sus etapas.

1.4. Referencias

Referencia	Título	Ruta	Fecha	Autor
IEEE	Estándar IEEE 830/98		22/10/2008	J.O.D.R.
Ingeniería de Requerimientos	Software Requirements. 2nd edition.		02/02/2003	Karl Wiegers
Diseño de arquitectura	Ingeniería del software Un enfoque practico		2010	Roger S. Pressman

1.5. Visión general

Este documento ofrece una descripción general del sistema y la información necesaria para conocer el funcionamiento del programa soft restaurant, además se especifican requisitos del sistema, diseño y beneficios previstos.

Este documento está organizado en 4 partes principales introducción descripción general, especificación de requisitos y apéndice, cada una con la información anteriormente descrita y con datos relevantes para cada uno de los elementos del contenido.

El programa gestor de restaurantes (soft restaurant) está estimado como un sistema de vanguardia que funciona en un entorno de escritorio con un buen rendimiento en los computadores, siendo clave que tenga una interfaz clara, amigable e intuitiva con el usuario final.

2. Vista del Modelo de Casos de Uso

2.1. Diagrama de Casos de Uso relevantes a la Arquitectura

[En esta sección se muestra un diagrama de Casos de Uso incluyendo solo los casos de uso relevantes a la Arquitectura - Diseño del Software.]

2.2. Casos de Uso relevantes a la Arquitectura

En esta sección se detallan los Casos de Uso y actores más significantes para definir la Arquitectura - Diseño del Software, estos diagramas ayudan a mitigar riesgos más serios, aclaran la funcionalidad tanto a equipo de desarrollo.

Estos casos de uso son:

- Registrar una orden.
- Consultar una orden.
- Cerrar una orden.
- Modificar una orden.
- Modificar el status de una orden.

- Alta de persona.
- Edicion de personal.
- Baja de personal.
- Alta de platillos.
- Baja de platillos.
- Edición de platillos.

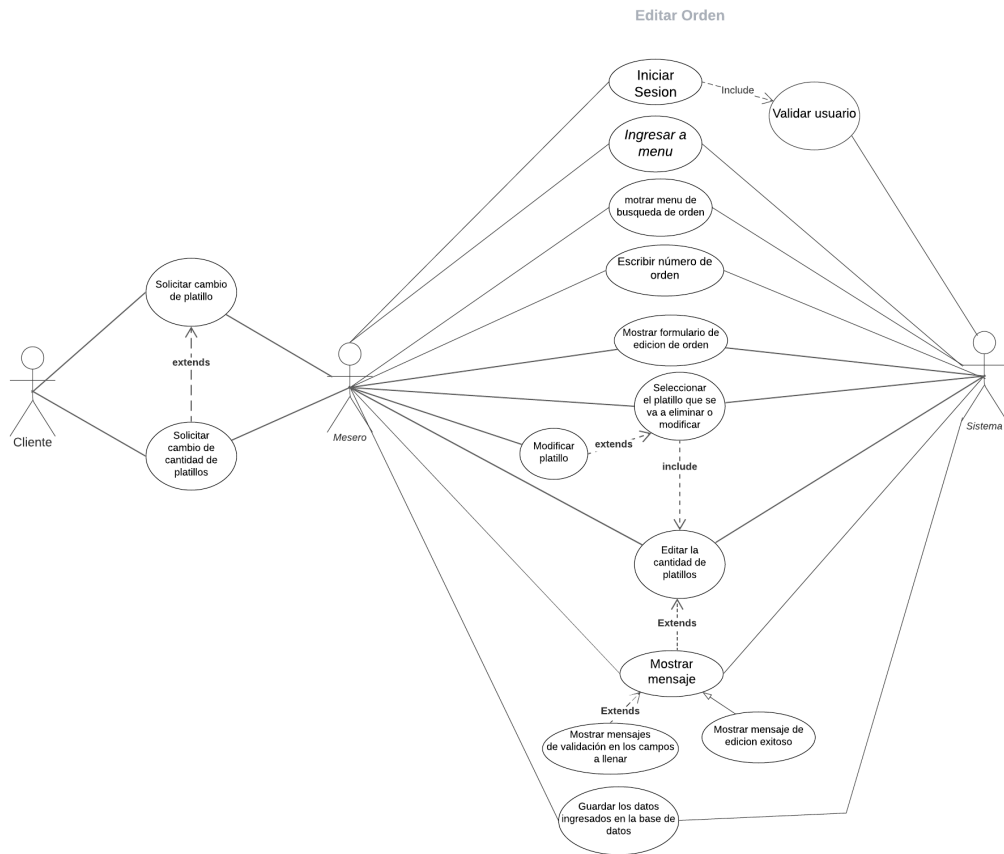
2.2.1. Registrar una orden.

El sistema registra la orden de alimentos / bebidas que el cliente requiere en la mesa en que se encuentra, dicho pedido es registrado por el mesero de manera directa en el sistema.



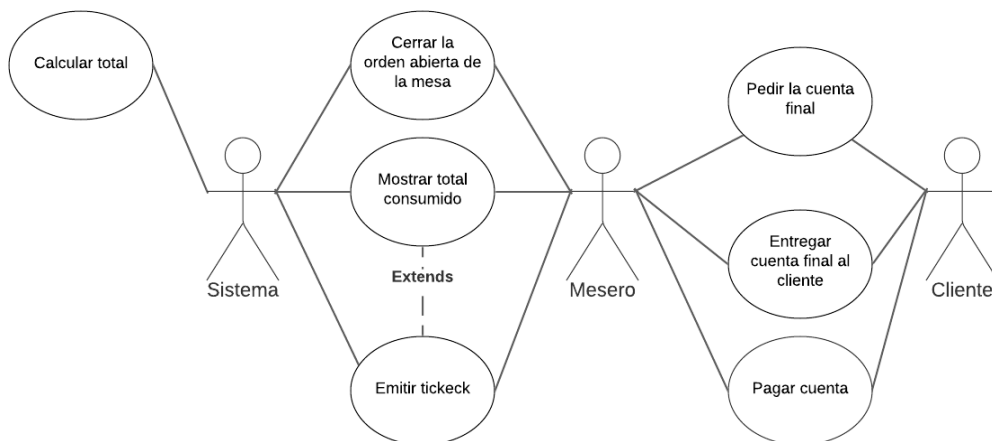
2.2.2. Edición de una orden

El sistema permitirá modificar el contenido de una orden (previamente guardada en el sistema). Cantidad, platillo solicitado, cerrar la orden. Este proceso lo pueden realizar todos los usuarios con los permisos correspondientes (Gerente, mesero, cocina).



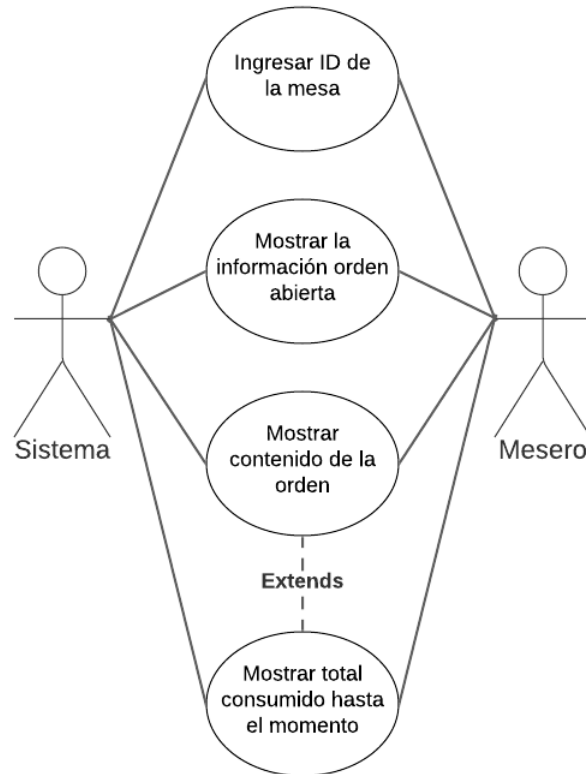
2.2.3. Cerrar una orden

Un usuario con los privilegios necesarios es capaz de dar por concluida una orden, esto para que ya no sea nadie capaz de agregar o eliminar elementos extras a la orden y así mantener la integridad de los datos.



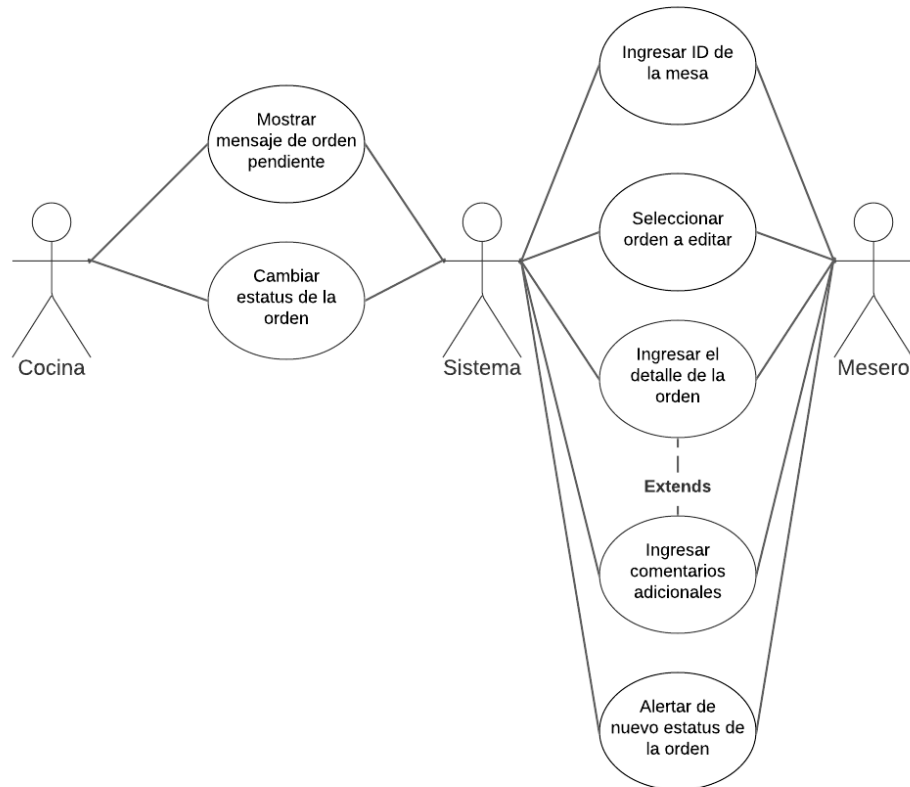
2.2.4. Consultar una orden y su contenido

El usuario mesero podrá consultar la información de la orden de alguna mesa y su contenido a través del sistema.



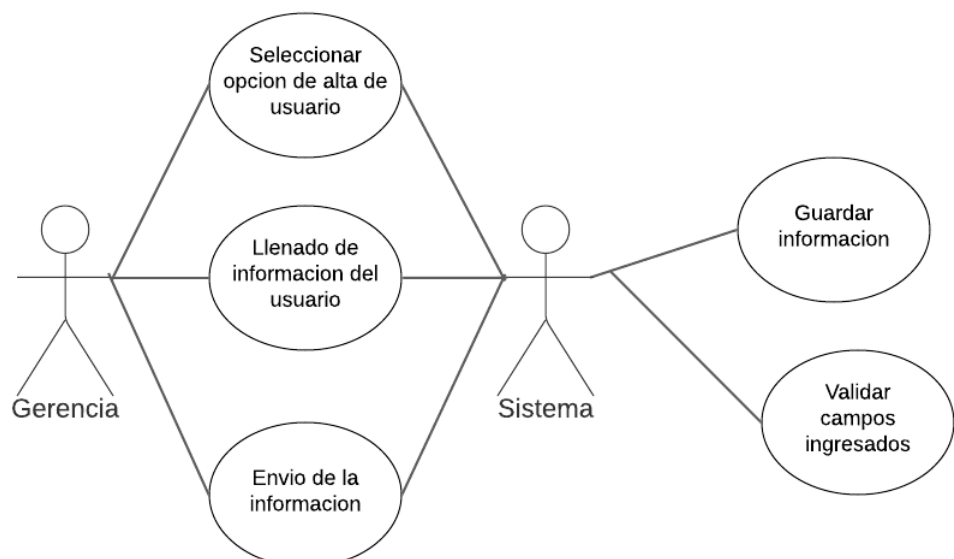
2.2.5. Modificar el estatus de una orden (contenido).

Al momento de hacer un pedido en la orden, el pedido llegará a la cocina, donde esta área comenzará la preparación del pedido (alimento/bebida)



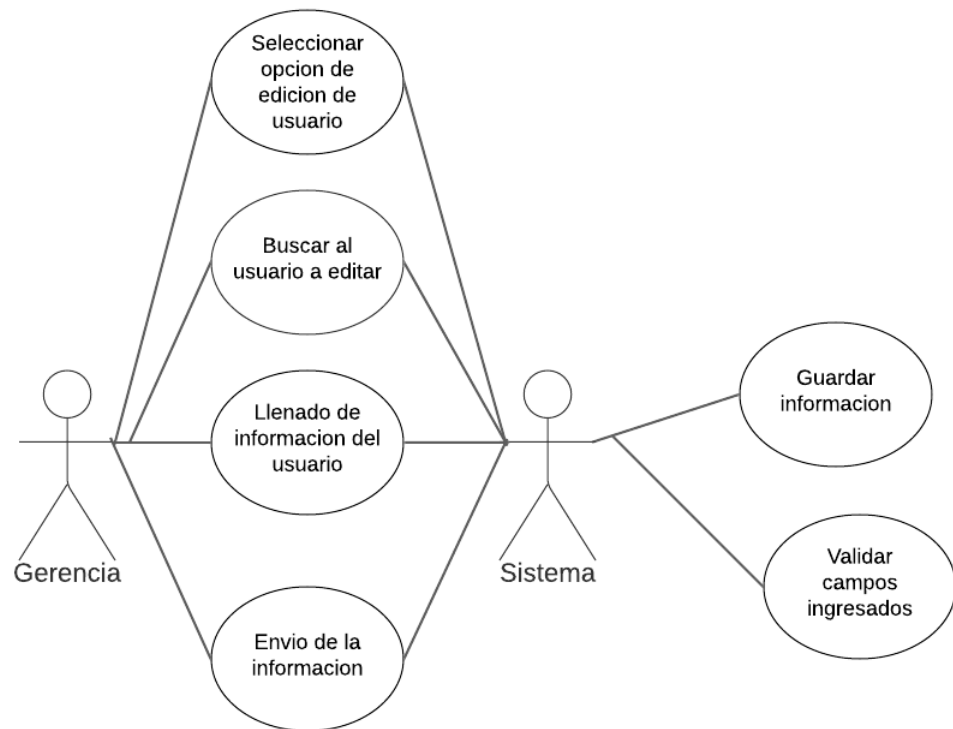
2.2.6. Alta de usuario

Solamente el usuario con más permisos dentro del sistema puede dar de alta a otros usuarios, por lo que este proceso está restringido a personal autorizado.



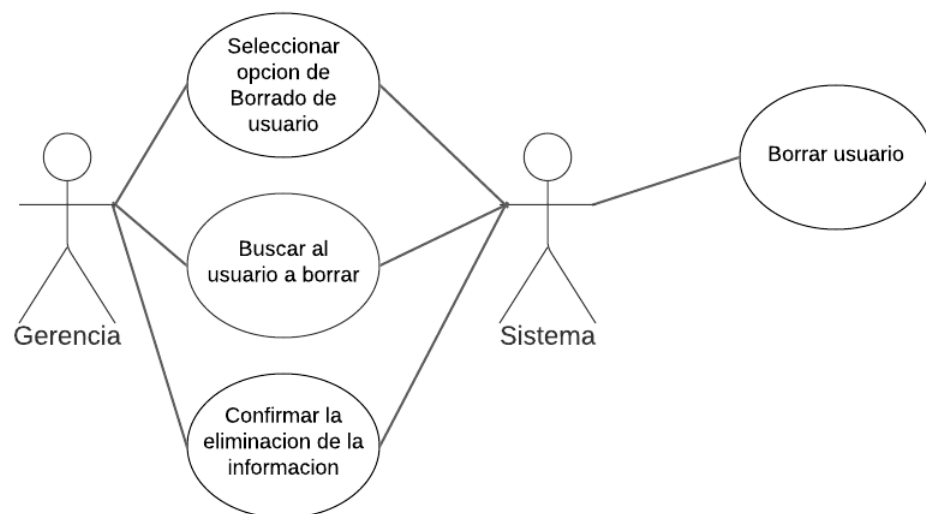
2.2.7. Edición de usuario.

La edición de usuarios solo se llevará a cabo por usuarios con mayor permiso en el sistema, por lo que este proceso aplica solamente a usuarios en la gerencia del restaurante.



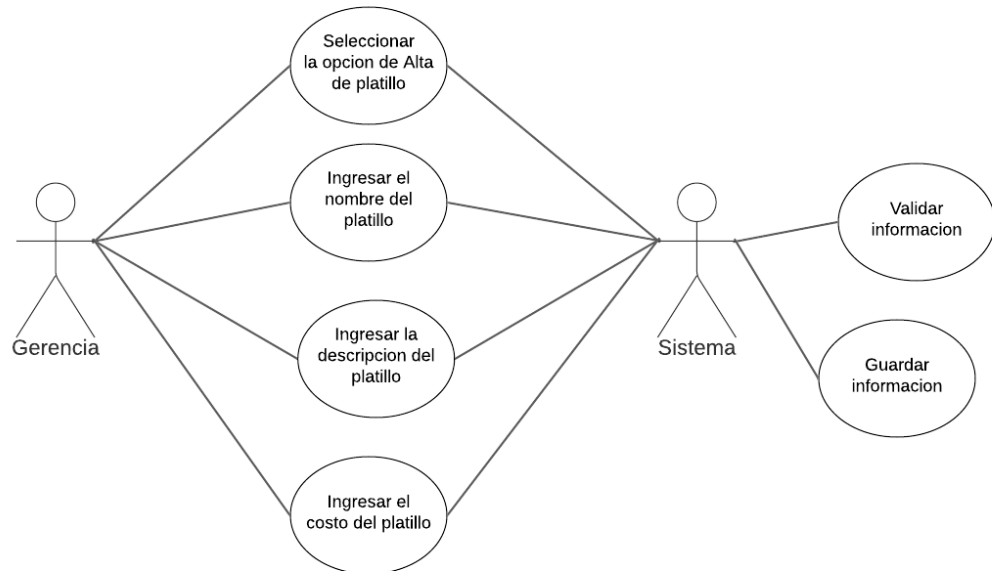
2.2.8. Borrado o baja de usuario.

EL borrado de un usuario es un proceso delicado, por lo que, al hacerlo, se debe confirmar la acción, estas operaciones solo se pueden realizar por la gerencia del restaurante.



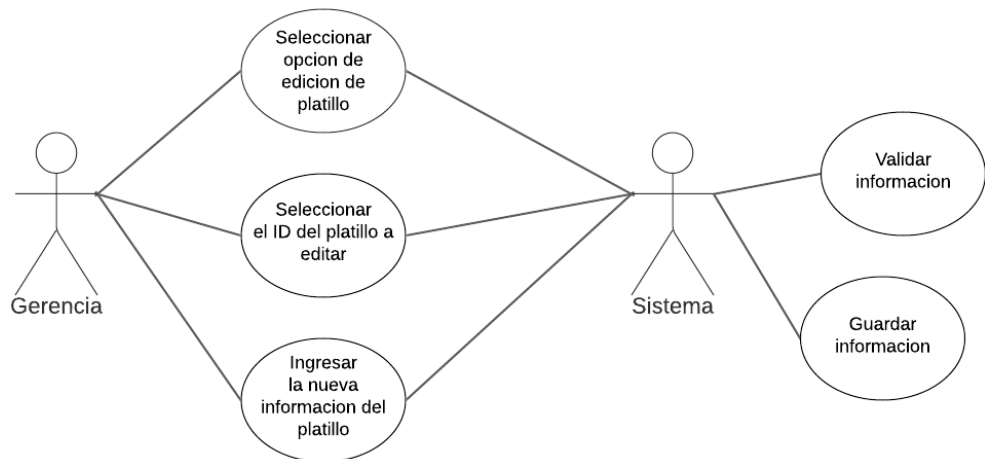
2.2.9. Alta de platillo

EL alta de un platillo lo realiza solo los usuarios con permisos de gerencia, esto permite añadir nuevos platillos al menú del restaurante.



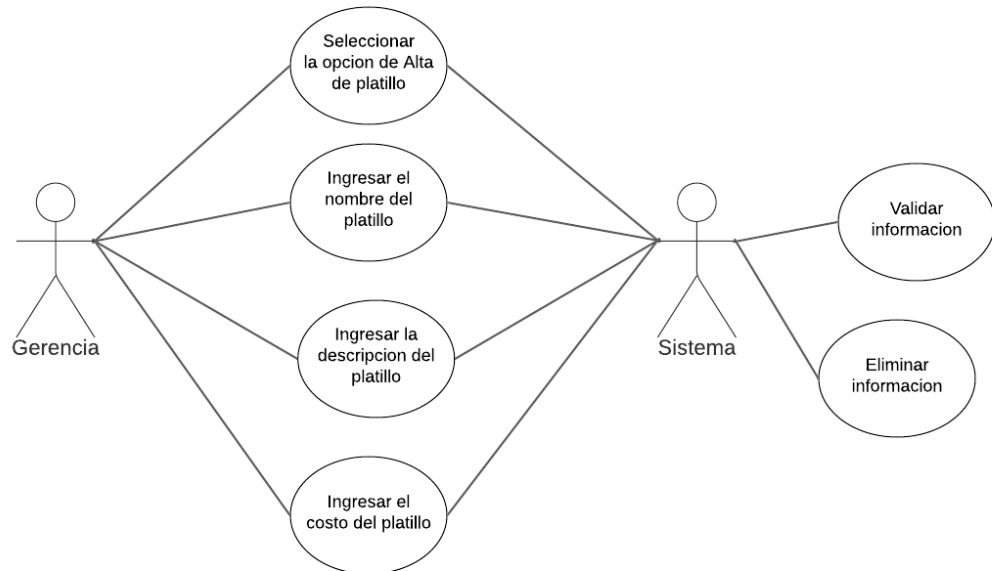
2.2.10. Edición de platillo.

Permite que la información de un platillo previamente guardado en el sistema sea modificada, ya sea su título, su disponibilidad, su precio o su descripción.



2.2.11. Baja o eliminación de platillo.

Permite eliminar permanentemente un platillo del sistema, esto sin la posibilidad de recuperación.



3. Trazabilidad desde el Modelo de Casos de Uso al Modelo de Diseño

[Describir la trazabilidad entre el Modelo de Casos de Uso y el Modelo de Diseño. Identificando los objetos y subsistemas de diseño que intervienen en el caso de uso y sus relaciones (Diagrama de paquetes).]

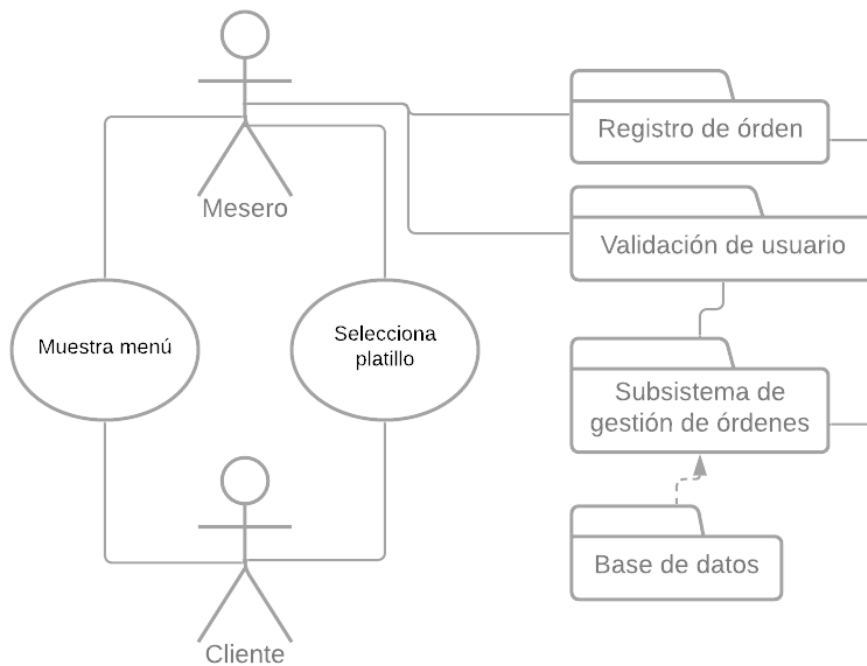
La trazabilidad entre el modelo de casos de uso y el modelo de diseño identifica los subsistemas de interfaz de usuario y administrador según el actor que los utilice, se hace una separación entre las capas de la lógica del negocio y la capa de presentación al usuario.

En la capa de la lógica de negocio hay subsistemas para la redacción de dudas y sugerencias, gestión de empleados, gestión de usuarios, gestión de órdenes y gestión de platillos.

Cada uno de estos subsistemas de la capa lógica son influidos o dependientes de la base de datos por lo que se identifica también como un paquete que tiene relación con esa capa.

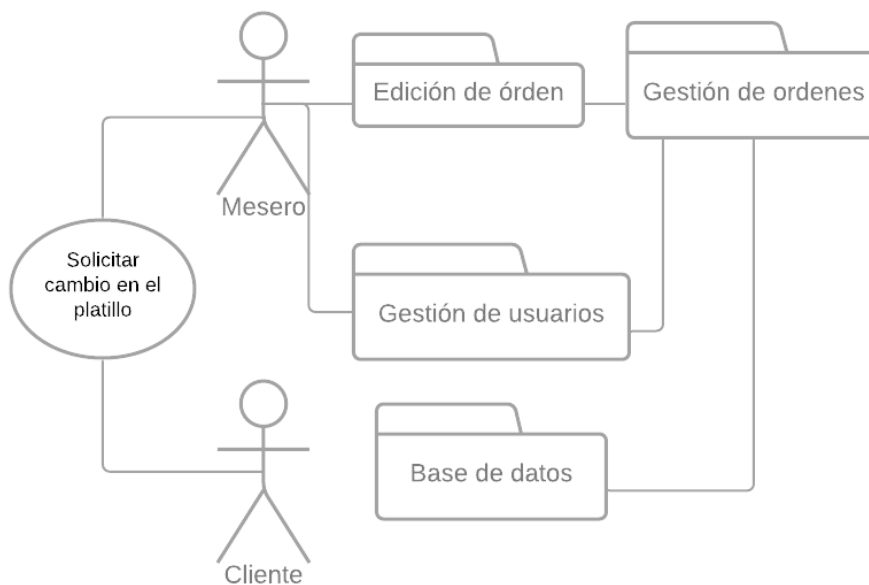
3.1.1.Registro de ordenes

Registro de órdenes



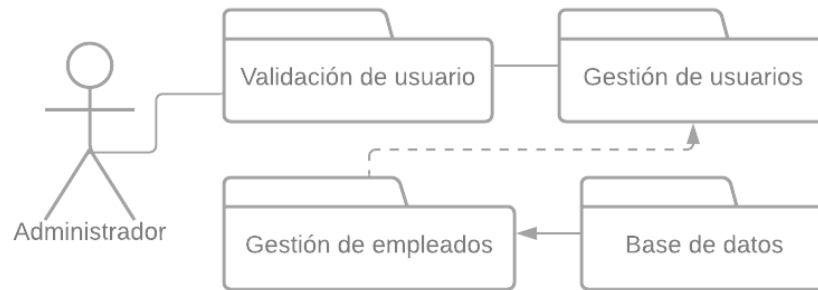
3.1.2. Edición de ordenes

Edición de órdenes



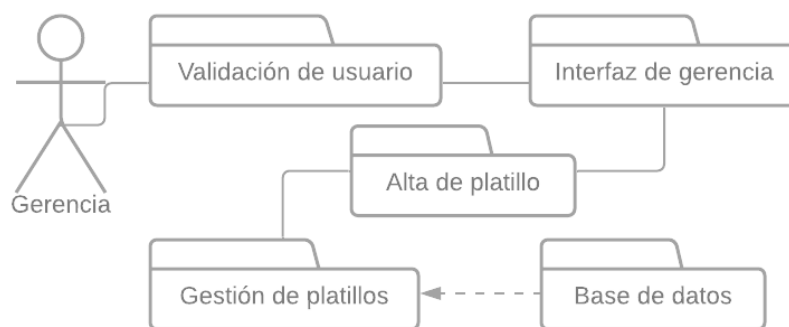
3.1.3. Gestión de usuarios

Gestión de usuarios



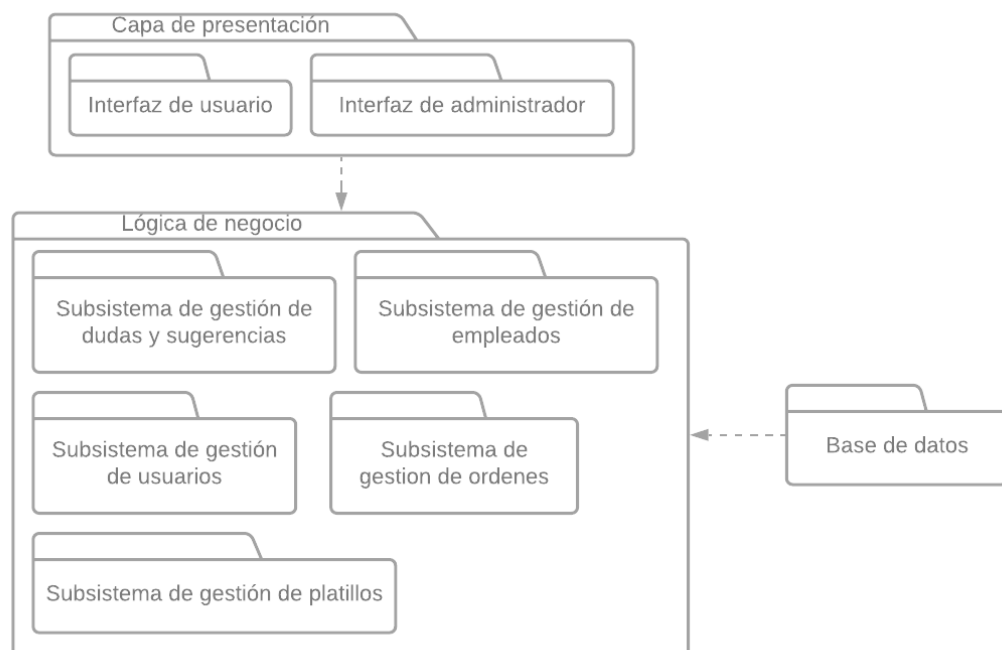
3.1.4. Gestión de platillos

Gestión de platillos



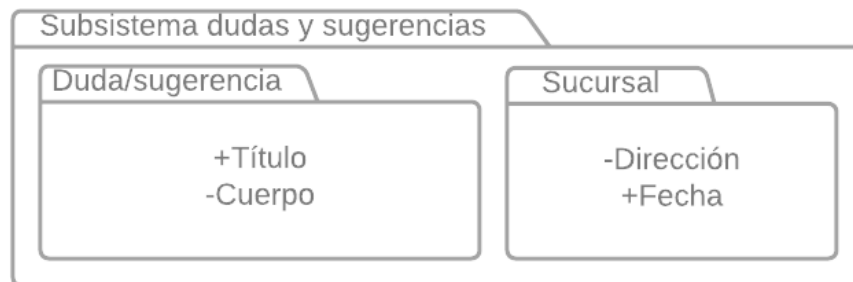
4. Vista del Modelo de Diseño

4.1. Descomposición en Subsistemas



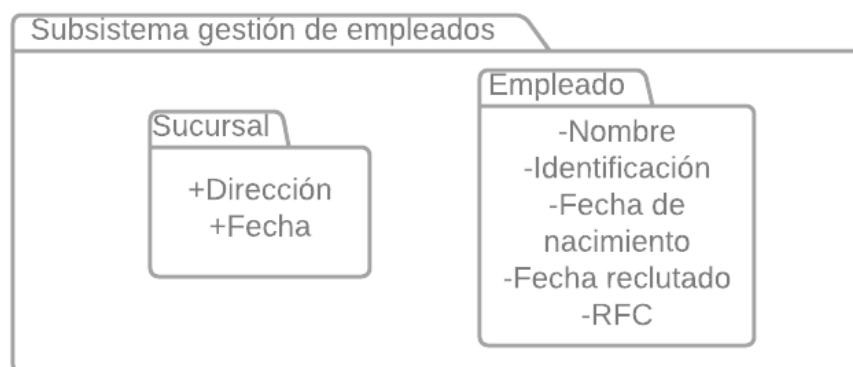
4.1.1. Subsistema de gestión de dudas y sugerencias

El subsistema de gestión de dudas y sugerencias contiene la clase de duda/sugerencia que contiene los atributos de, título y cuerpo. También contiene la clase Sucursal que está compuesta por destinatario y fecha.



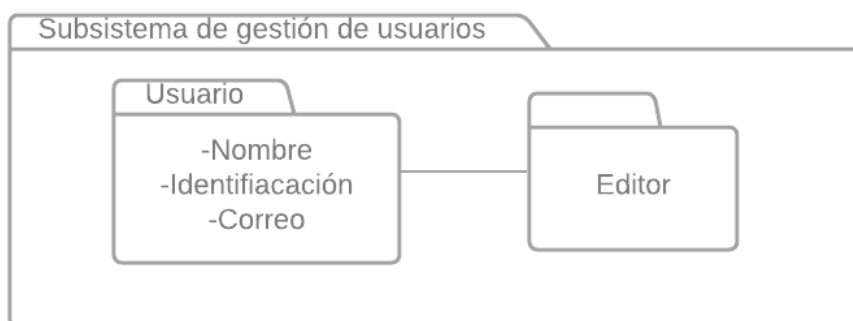
4.1.2. Subsistema de gestión de empleados

El subsistema de gestión de empleados contiene la clase empleado y Sucursal, la clase empleado contiene los atributos de Nombre, Identificación, Fecha de nacimiento, Fecha reclutado, sueldo y RFC, la clase sucursal incluye Fecha y dirección.



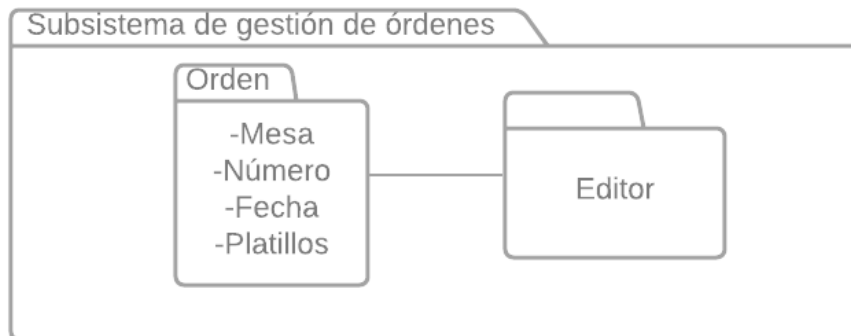
4.1.3. Subsistema de gestión de usuarios

El subsistema de gestión de usuarios tiene la clase usuario y el paquete Editor. El editor configura al usuario y le otorga permisos. La clase usuario contiene Nombre, identificación y correo.



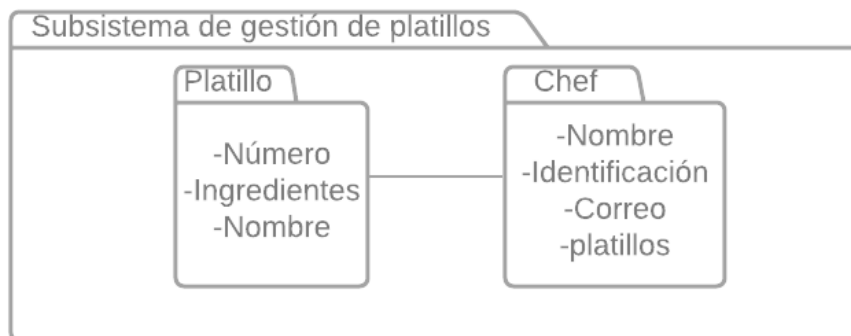
4.1.4. Subsistema de gestión de ordenes

El subsistema de Gestión de ordenes tiene la clase orden y el paquete editor que permite interactuar con la clase orden y modificarla.



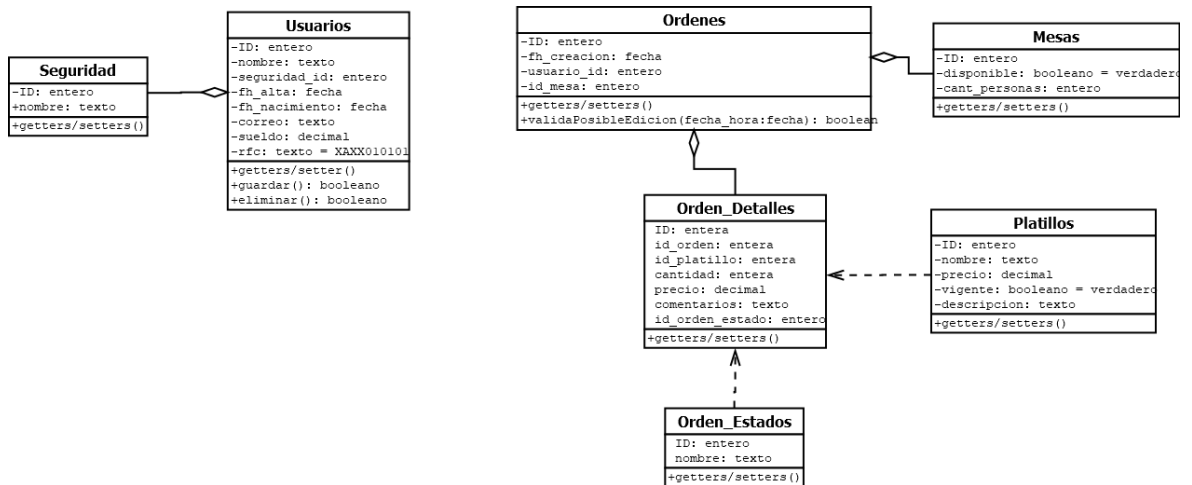
4.1.5. Subsistema de gestión de platillos

El subsistema de gestión de platillos contiene la clase Platillo y Chef. La clase platillo sirve para dar de alta platillos y sus ingredientes, y la clase Chef identifica los cocineros que pueden hacer dichos platillos, ambas interactúan en la creación y realización de órdenes.



4.2. Diseño de Clases

El siguiente diagrama es un acercamiento a las clases internas que se estarán interactuando entre sí en el sistema.



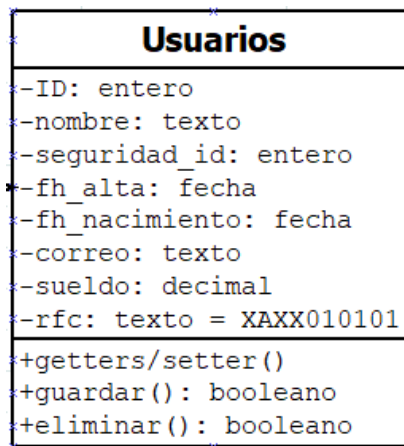
4.2.1. Clase Usuarios

La clase Usuarios es una representación de los usuarios que manipularan en el software final, teniendo como atributos privados:

- ID: es el campo identificador del usuario ingresado dentro del sistema.
- Nombre: es el nombre del usuario
- Seguridad_id: Es el campo que define que tipo de usuario es, desde gerencia, mesero o cocina.
- Fh_alta: Campo que guarda el día y la hora en la que el usuario fue dado de alta en el sistema.
- Fh_nacimiento: Campo donde se guarda la fecha de nacimiento del usuario registrado.
- Correo: correo electrónico del usuario.
- Sueldo: dinero que se le paga al usuario por tiempo definido por el negocio (diario/semanal/quincenal/mensual).
- Rfc: Clave única del usuario, se inicializa con la cadena "XAXX010101".

Con sus respectivos métodos de acceso a sus atributos (getters y setters) de la clase, además de los métodos correspondientes para guardar y eliminar usuarios en la base de datos.

Además, cuenta con una relación de agregación con la clase Seguridad, ya que esta usa su información para saber qué tipo de usuario es.



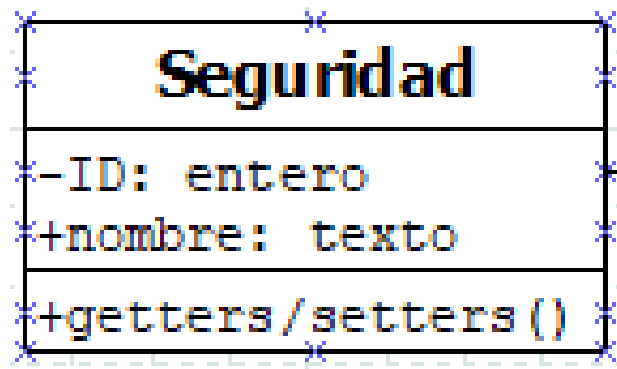
4.2.2. Clase Seguridad

Clase que define que tipos de usuarios existen dentro del sistema.

Cuenta con los siguientes atributos:

- ID: es campo identificador del objeto.
- Nombre: es el nombre del área del restaurante.

Y cuenta con sus métodos reglamentarios de acceso a sus atributos (getters y setters).



4.2.3. Clase Ordenes

Clase que representa como tal el "encabezado" de una orden, ya que con ella se identifica en sí la las ordenes, con un numero de orden, y el número de mesa asignada.

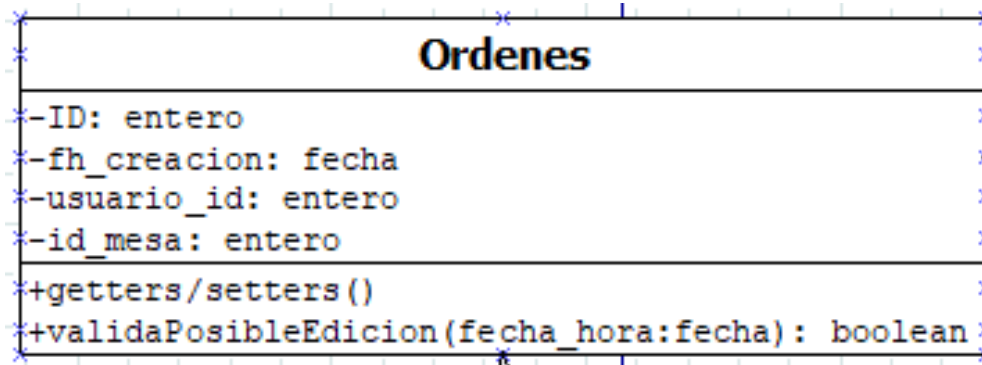
Cuenta con los siguientes atributos:

- ID: número de identificación de la orden.
- Fh_creacion: Campo que permite saber a qué hora y en qué fecha se creó la orden.
- Usuario_id: Campo para saber por cual usuario fue creada la orden.
- Id_mesa: Campo que relaciona la orden con la mesa que la abrió.

Y también cuenta con sus métodos reglamentarios para acceder a sus atributos (getters y setters) además de una función que valida si es posible realizar una modificación a esta, ya que si

Y también cuenta con sus métodos reglamentarios para acceder a sus atributos (getters y setters) además de una función que valida si es posible realizar una modificación a esta, ya que, si ha pasado mucho tiempo desde que se hizo un pedido, para evitar conflictos con el área de cocina, no se podrán hacer cambios en una orden que ya está en preparación o que ya está lista.

Además de tener una relación con la clase Mesas y la Clase Orden_detalle.



4.2.4. Clase Orden_Detalles

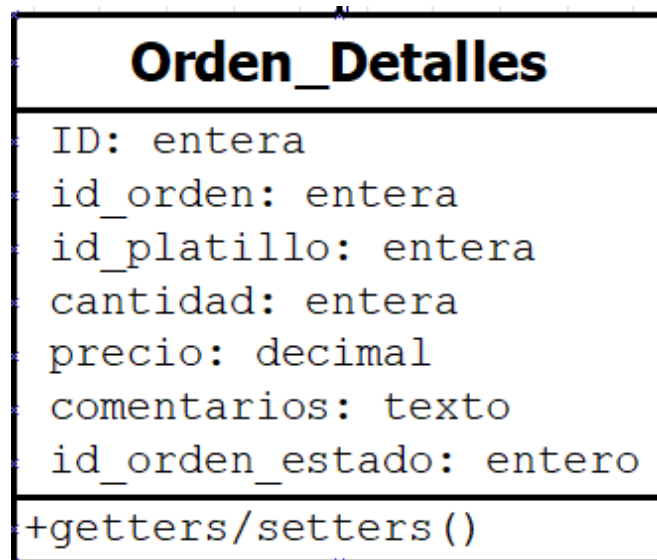
Clase representativa del contenido de las ordenes, es el modelo de como serian el contenido de las órdenes.

Cuenta con los siguientes campos atributo:

- ID: campo identificador único del registro.
- Id_orden: Campo que identifica a que orden está relacionada.
- Id_platillo: Campo que identifica que platillo integra esa orden.
- Cantidad: Campo que define la cantidad de platillos solicitados en la orden.
- Precio: Campo donde se almacenará el precio total de los platillos solicitados.
- Comentarios: Campo donde se almacenarán comentarios extras a los platillos solicitados, como especificaciones especiales hacia la cocina.
- Id_orden_estado: Campo que define cual es el estado de la orden.

Contando también con 2 relaciones de integración en su interior, dado que se necesita la clase Platillo y la clase Orden_estados para su existencia.

Además, cuenta con los métodos reglamentarios de acceso de datos (getters y setters).



4.2.5. Clase Orden_Estados

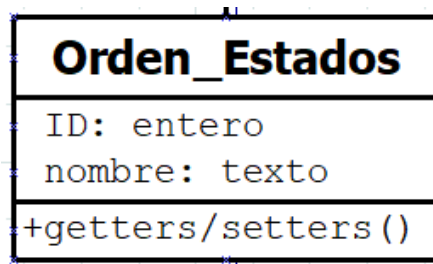
Clase simple que almacenará los diferentes estados de la orden dentro del sistema.

El nombre de los estados será de acuerdo con los que especifique el cliente.

Cuenta con los siguientes atributos:

- ID: Es el identificador único para el estado de la orden
- Nombre: Es el nombre que se le da al estado de la orden

Igualmente contará con sus métodos de acceso de datos getters y setters.



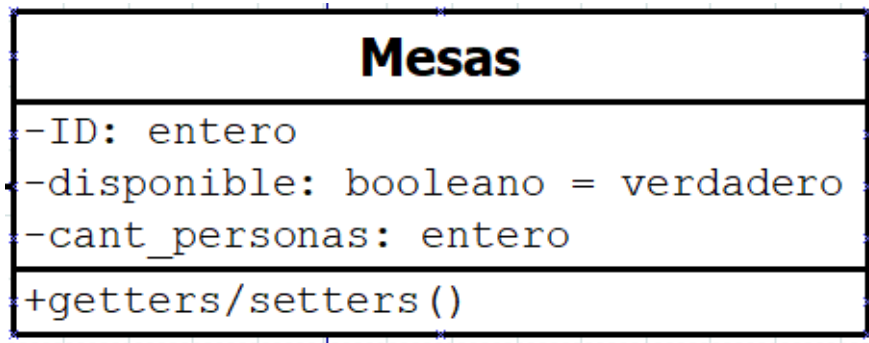
4.2.6. Mesas

Clase que modelará las mesas dentro del restaurante.

Cuenta con los siguientes atributos:

- ID: Identificador único de la mesa.
- Disponible: campo booleano que permite saber si la mesa está ocupada o está disponible actualmente.
- Cant_personas: Campo que define cuantas personas pueden usar la mesa en una misma interacción.

Además, cuenta con sus métodos de acceso a sus atributos (getters y setters).



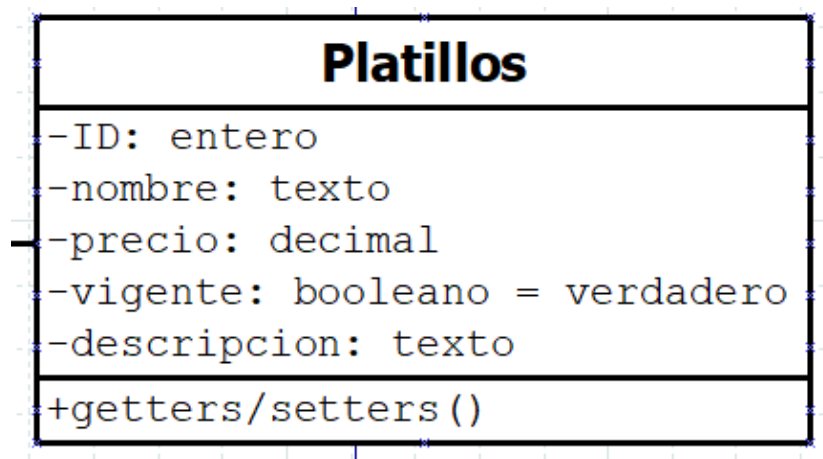
4.2.7. Platos

Clase que servirá como modelo de los platos ofertados dentro del establecimiento.

Cuenta con los siguientes atributos:

- ID: Identificador único del plato
- Nombre: nombre del plato ofertado en el menú.
- Precio: Precio del plato en el restaurante.
- Vigente: Campo que servirá para saber que platos están disponibles y cuales no en ese momento, se inicializará por defecto con un valor verdadero.
- Descripción: Campo donde se añadirá una breve descripción del plato y sus componentes para facilitar a los usuarios sobre el plato.

Además, cuenta con sus métodos reglamentarios de acceso a los atributos getters y setters.

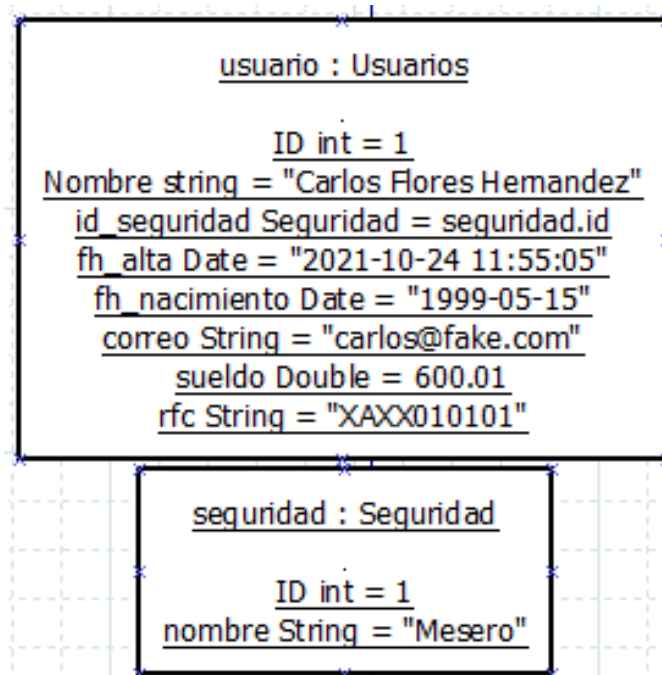


4.3. Diseño de objetos.

En este apartado se mostrarán ejemplos de cómo se poblarán los objetos en el sistema, creándose a partir de las clases anteriormente mencionadas.

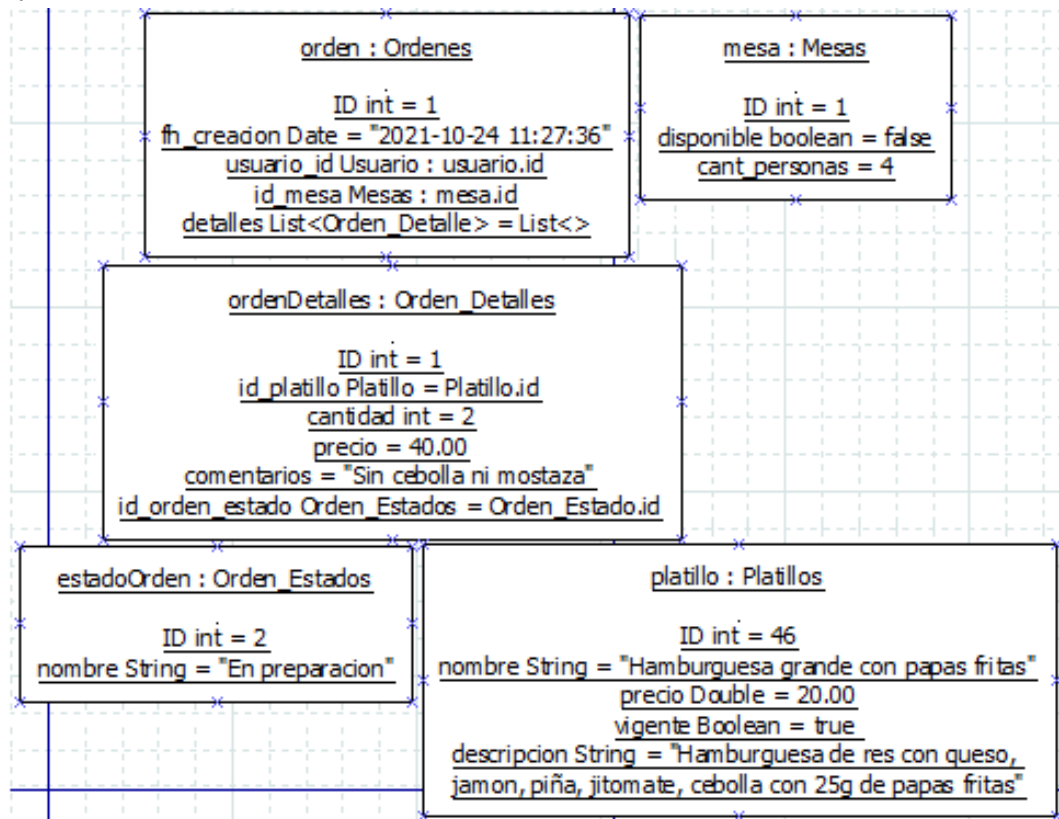
4.3.1. Objeto Usuario

El objeto usuario será el usuario que estará con la sesión activa dentro del sistema, dependiendo de los permisos asignados al usuario, serán las acciones que este podrá realizar dentro del sistema.



4.3.2. Objeto Orden

Este es el objeto más complejo del sistema, ya que integra muchas clases dentro de él, además de contener la información clave de los productos/platillos que solicitan los clientes del restaurante.

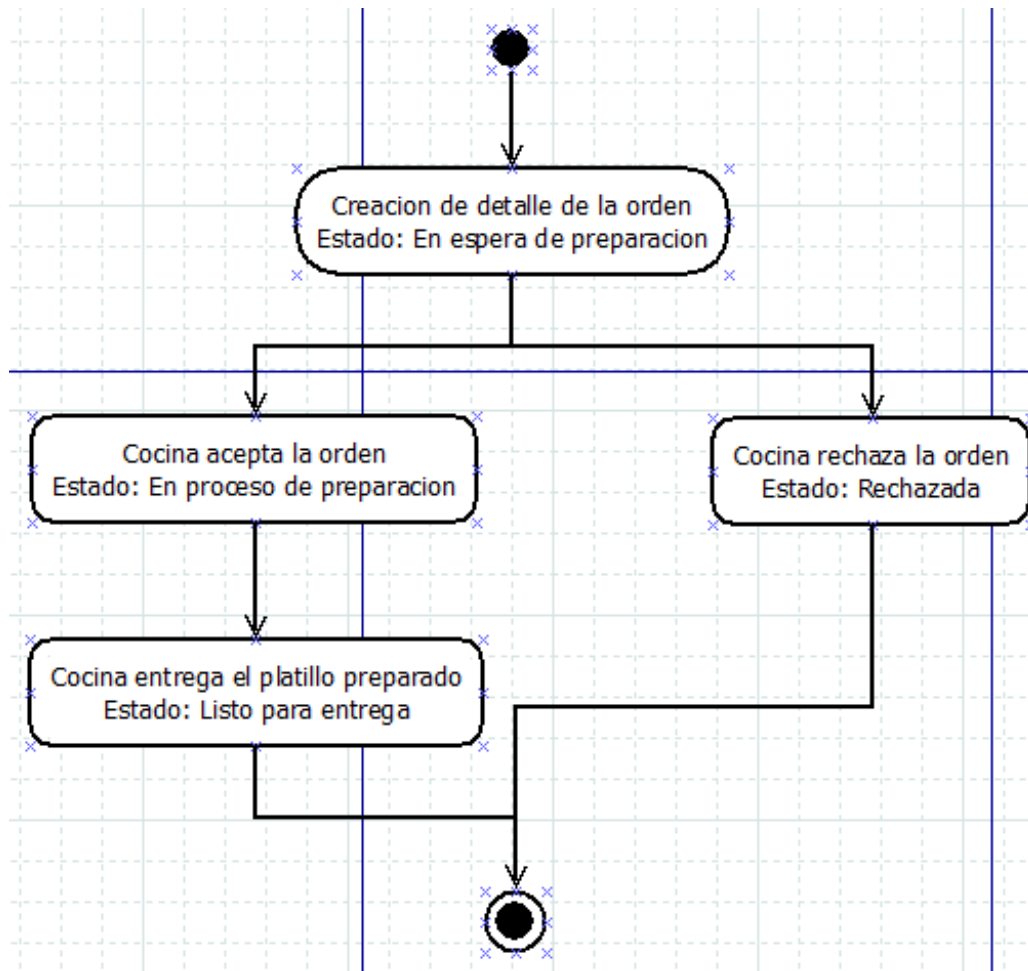


4.4. Diagramas de Estados.

En el actual apartado, se mostrará de forma gráfica los diferentes estados que pueden alcanzar los objetos dentro del sistema.

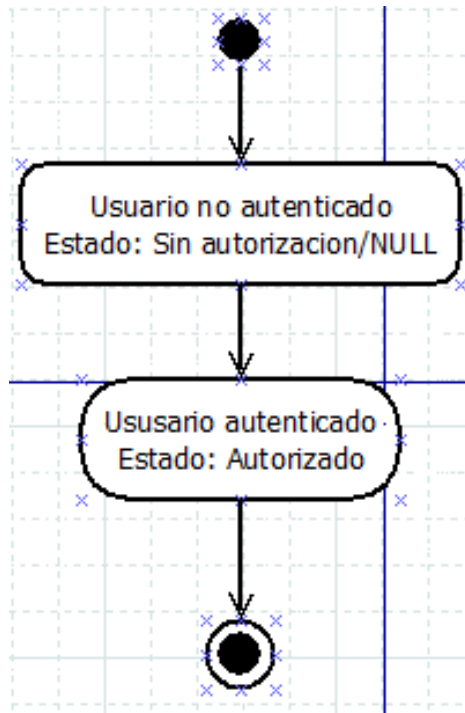
4.4.1. Diagrama de Estados de Orden

El siguiente diagrama de estados representara de forma gráfica como una orden puede pasar de un estado a otro, como se mencionó anteriormente, este solo puede ser efectuado por un usuario ingresado al sistema y con los permisos necesarios.



4.4.2. Diagrama de Estados Usuario

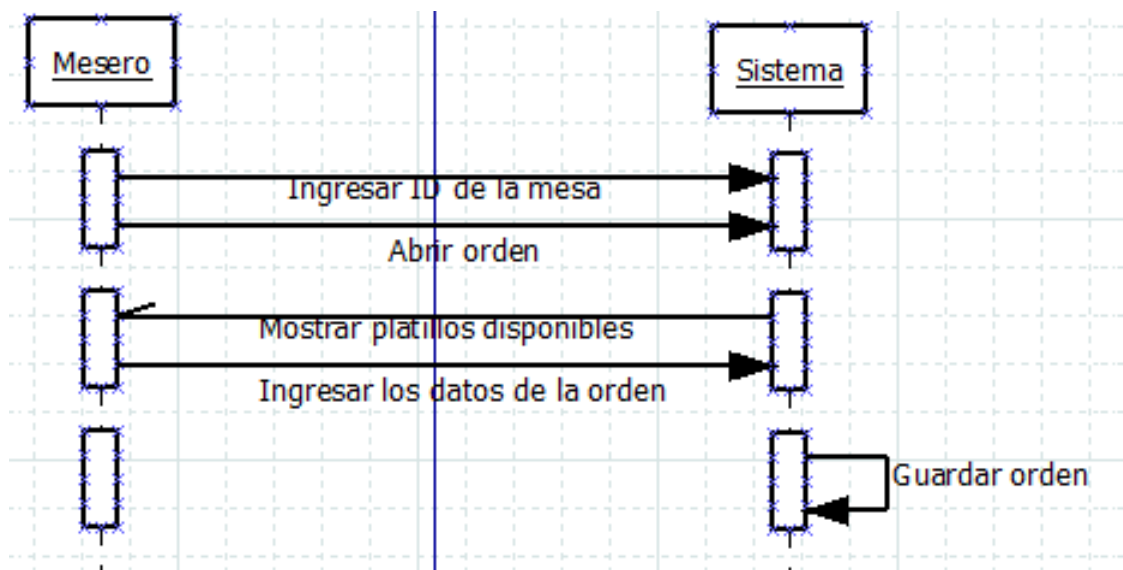
El siguiente diagrama muestra los estados del usuario dentro del sistema, es bastante básico y fácil de interpretar.



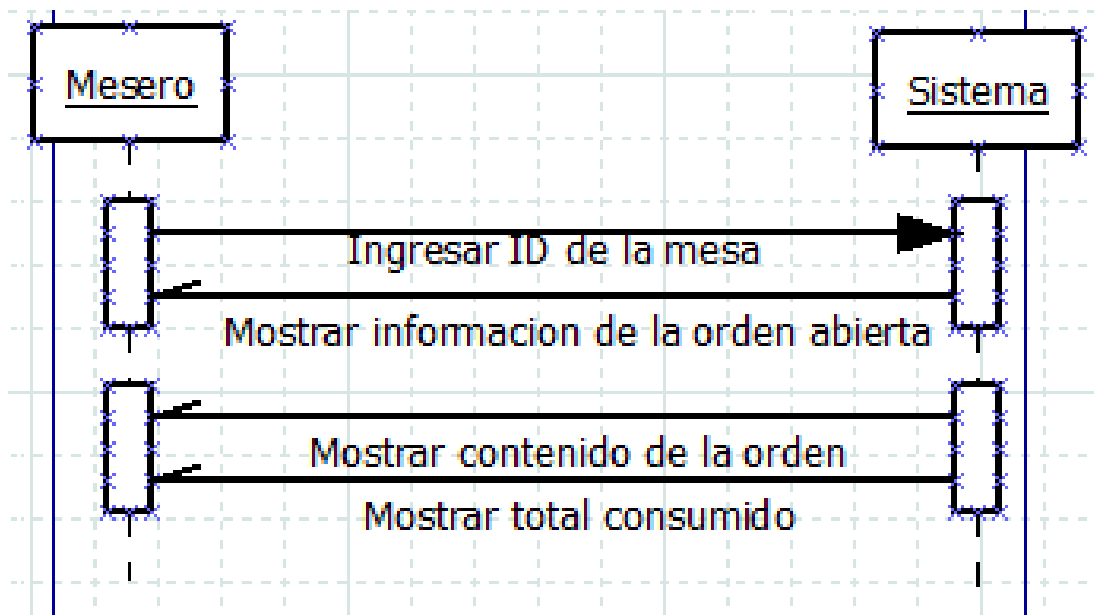
4.5. Diagramas de Secuencias

Los siguientes diagramas de secuencias buscan ilustrar de forma gráfica los procesos realizados en los diagramas de actividades, mostrando así en qué momento se realizan todas las acciones y en qué orden se realizan.

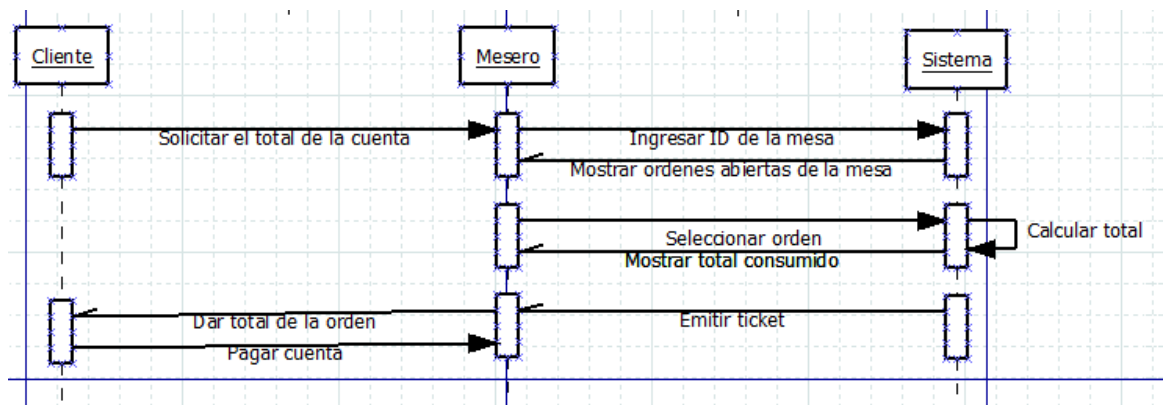
4.5.1. Diagrama de secuencias Alta de una orden.



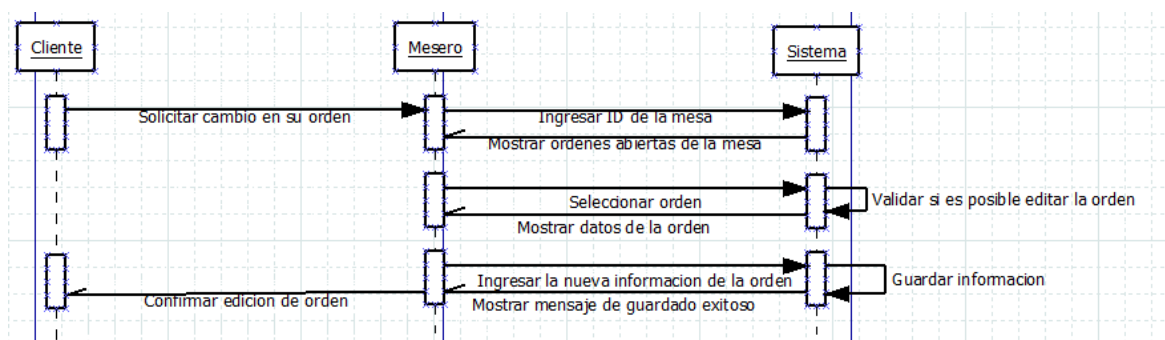
4.5.2. Diagrama de secuencias Consultar una orden.



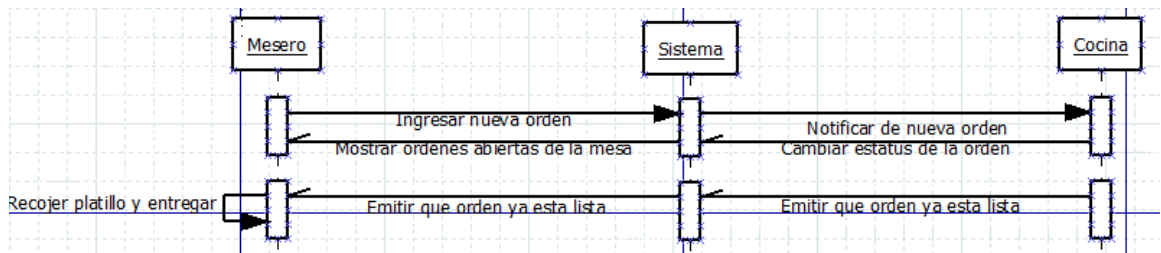
4.5.3. Diagrama de secuencias Cerrar una orden.



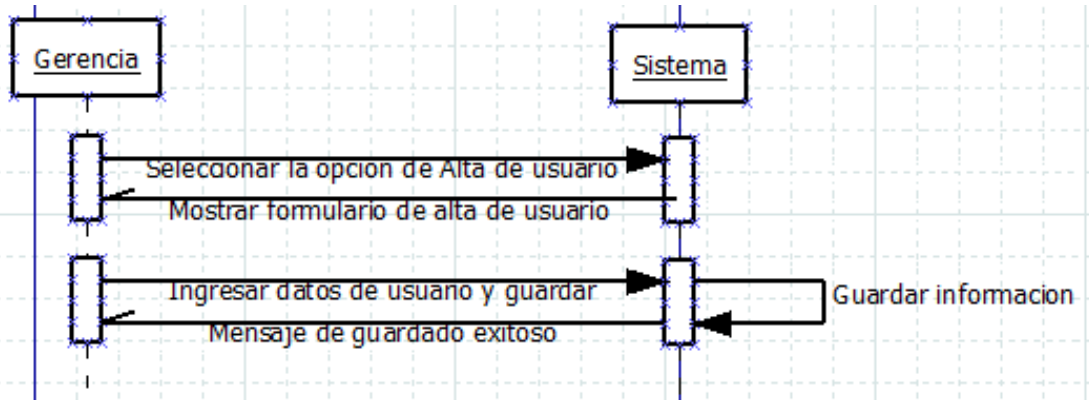
4.5.4. Diagrama de secuencias Modificar una orden.



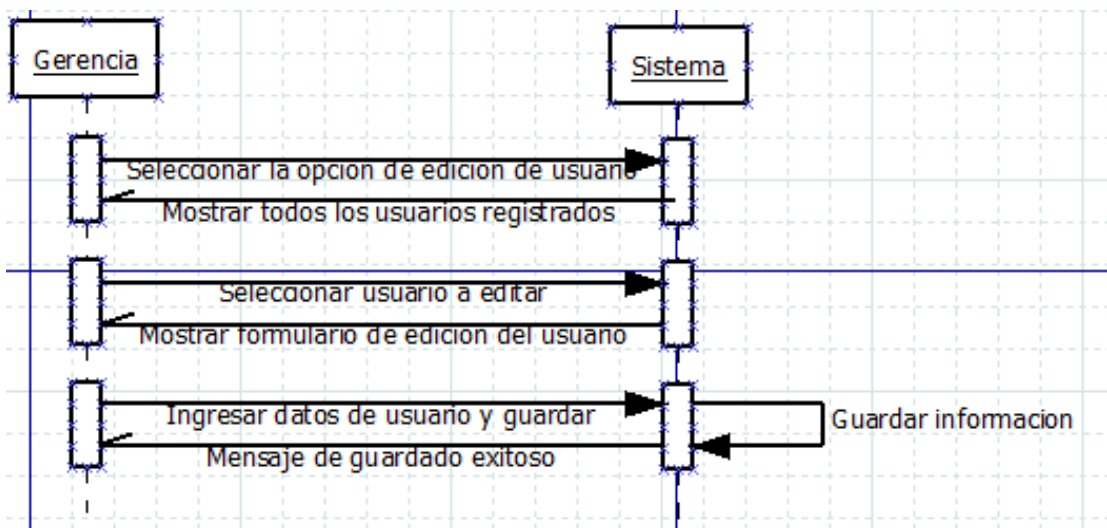
4.5.5. Diagrama de secuencias Modificar el estatus de una orden.



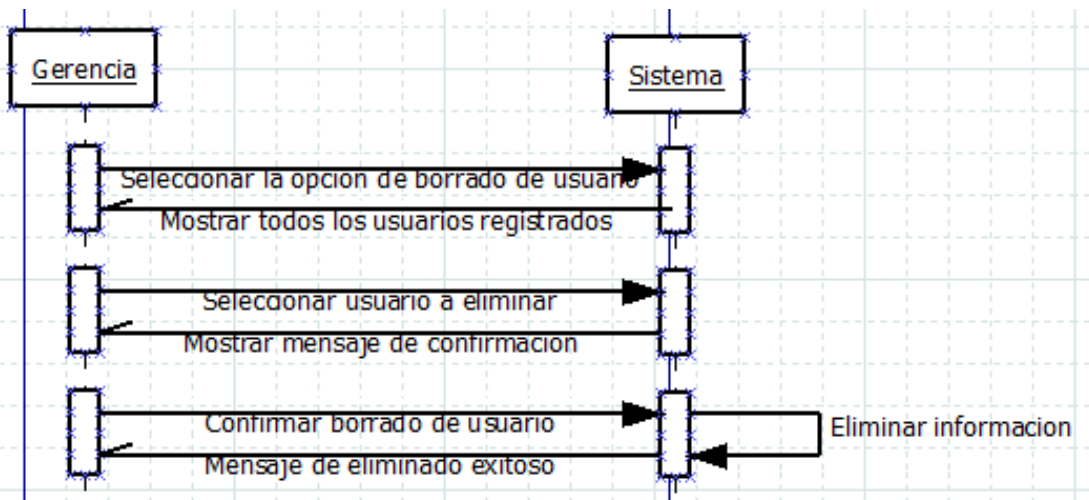
4.5.6. Diagrama de secuencias Dar de alta un nuevo usuario.



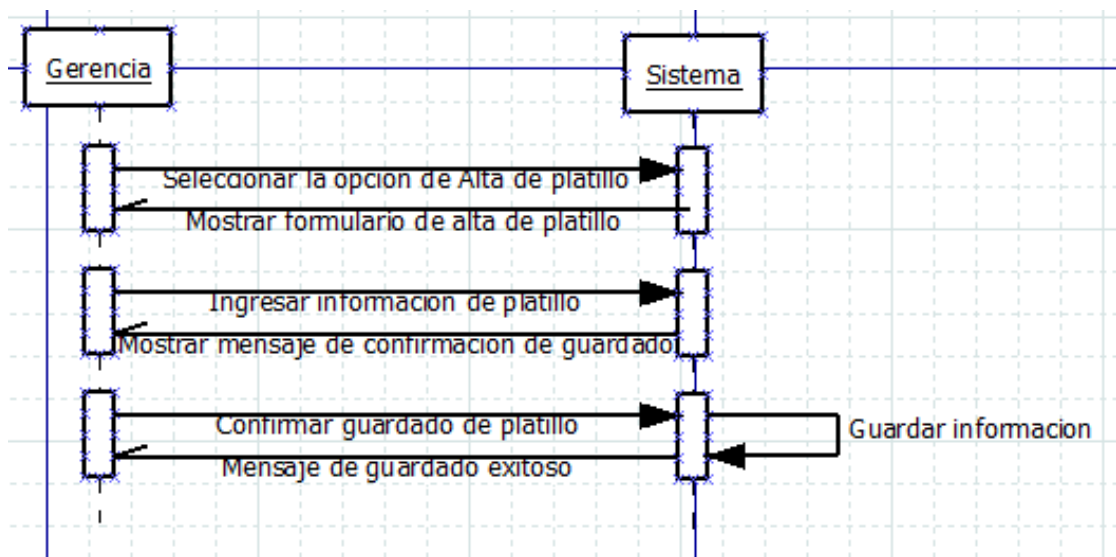
4.5.7. Diagrama de secuencias de edición de usuario.



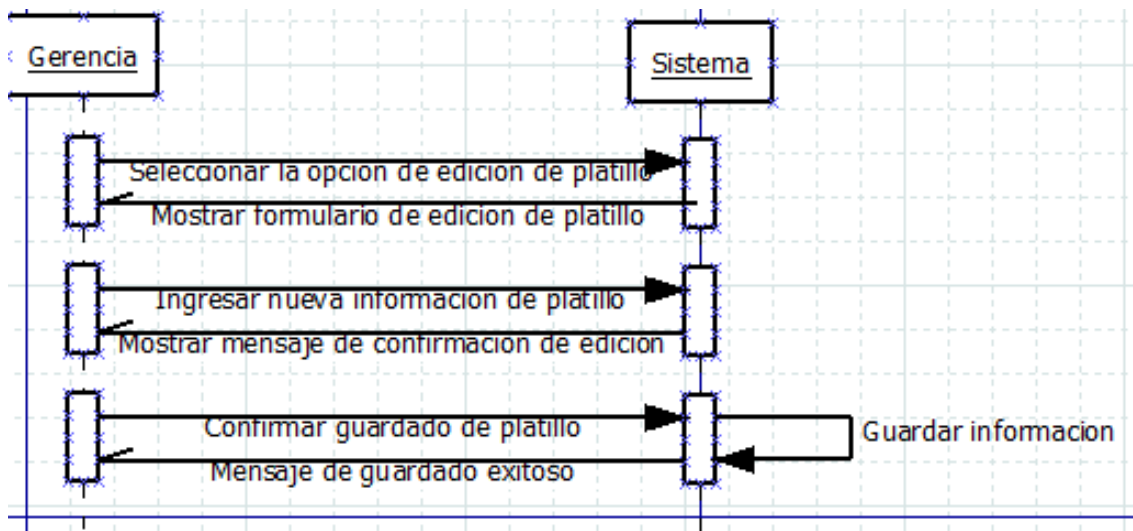
4.5.8. Diagrama de secuencias de baja de usuario.



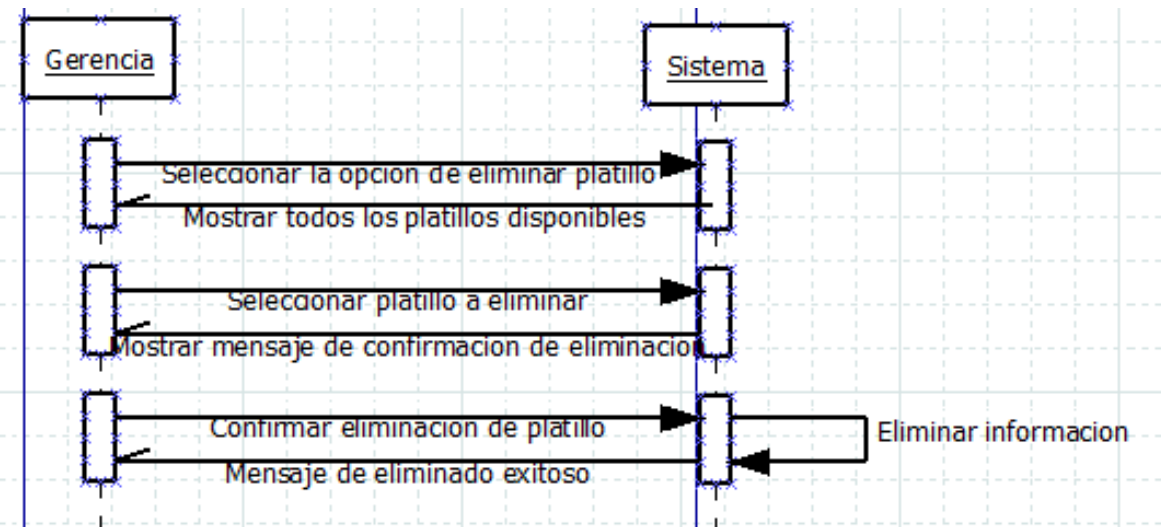
4.5.9. Diagrama de secuencias de Alta de platillo.



4.5.10. Diagrama de secuencias de Edición de platillo.



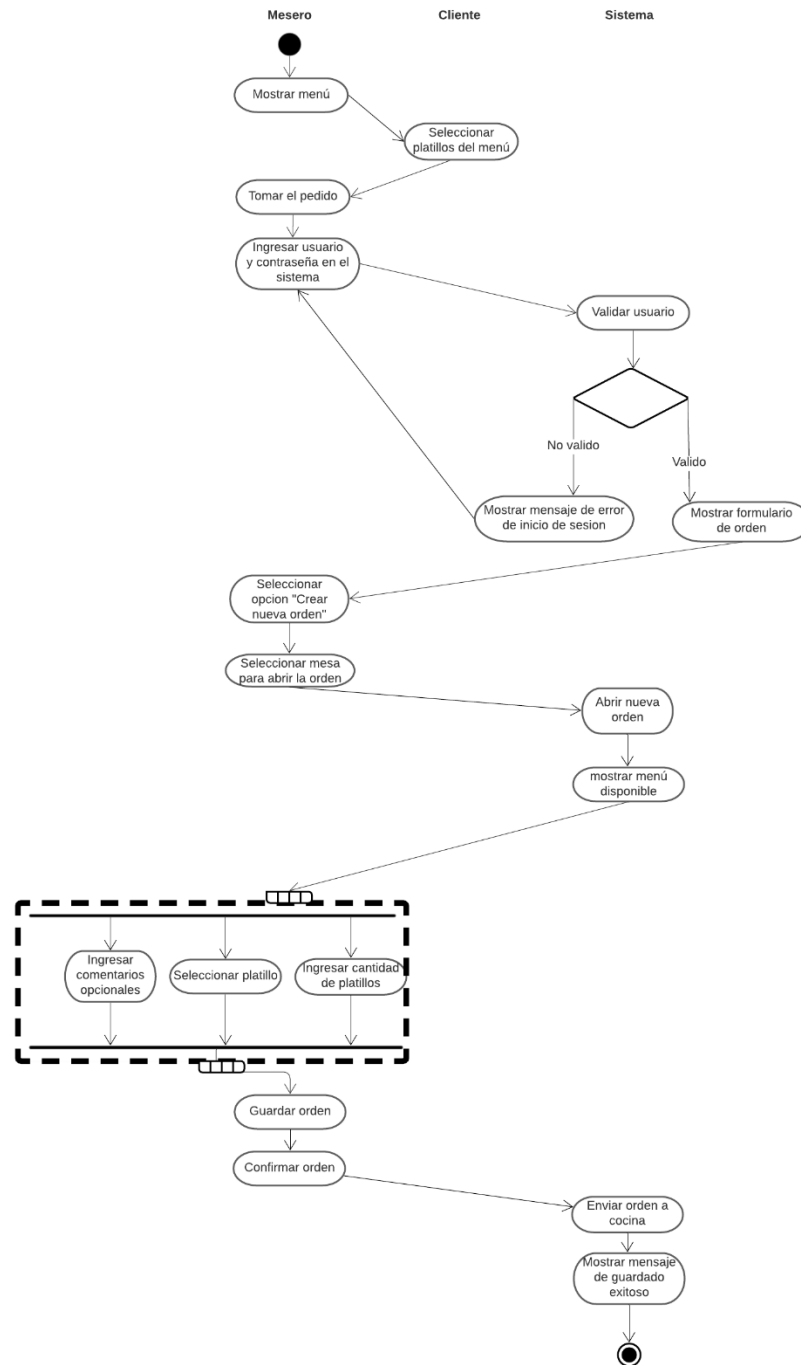
4.5.11. Diagrama de secuencias de Baja de platillo.



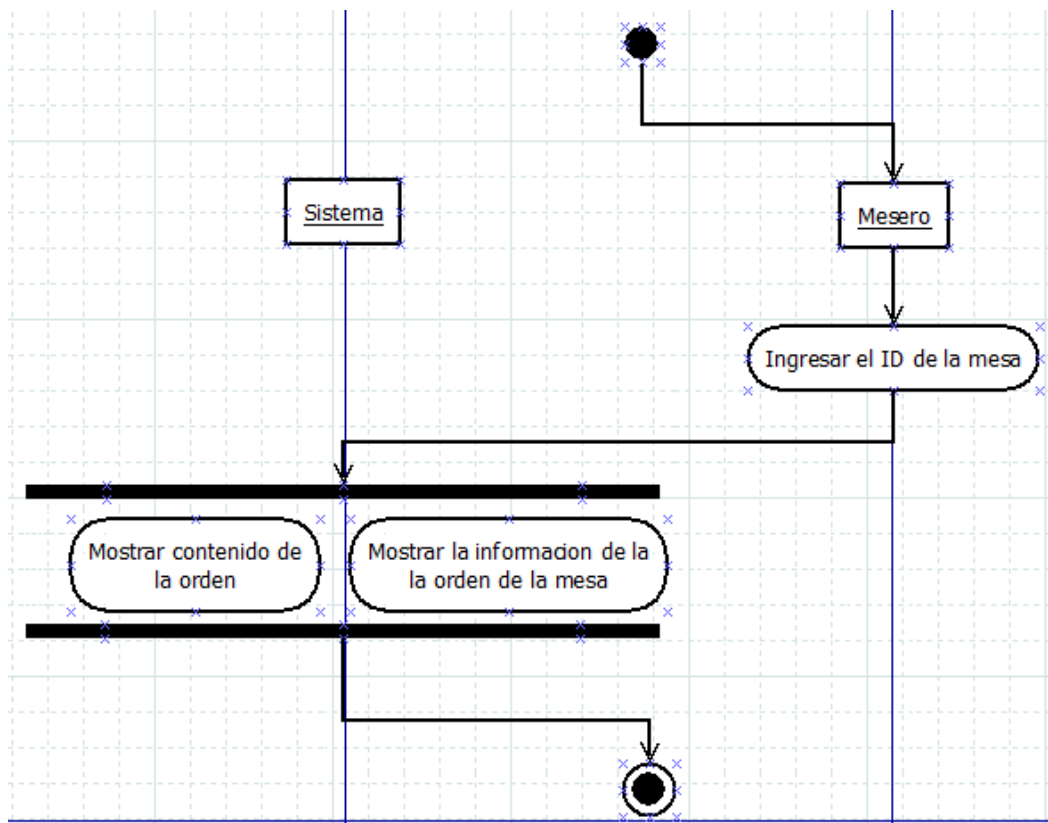
4.6. Diagramas de Actividades

En el siguiente apartado, se mostrarán los diagramas de actividades, que buscan mostrar de forma gráfica todos los procesos realizados a detalle, ilustrando la naturaleza dinámica de un sistema mediante el modelado del flujo ocurrente de actividad en actividad. Una actividad representa una operación en alguna clase del sistema y que resulta en un cambio en el estado del sistema. Típicamente, los diagramas de actividad son utilizados para modelar el flujo de trabajo interno de una operación.

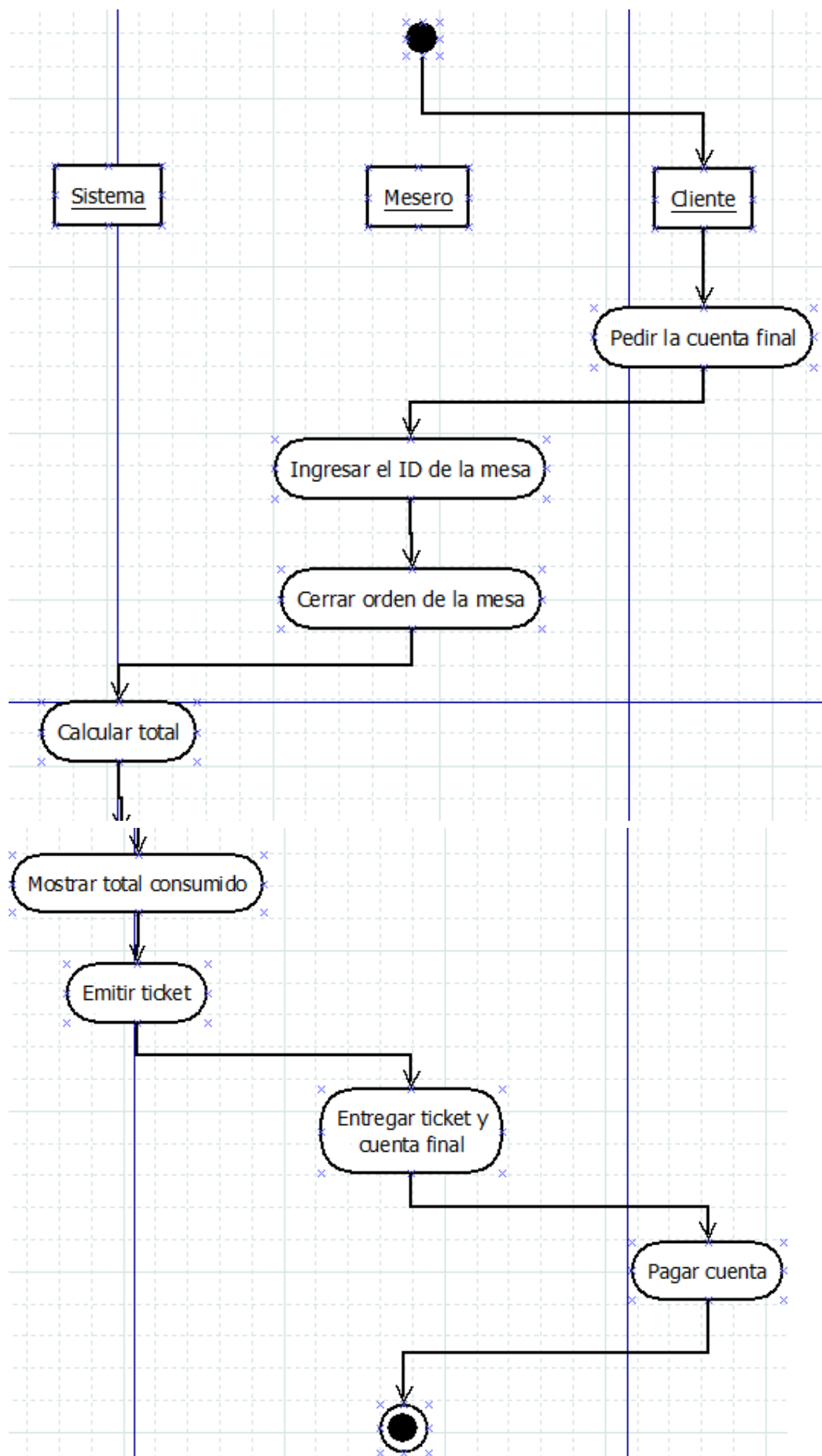
4.6.1. Diagrama de actividad Registrar Orden.



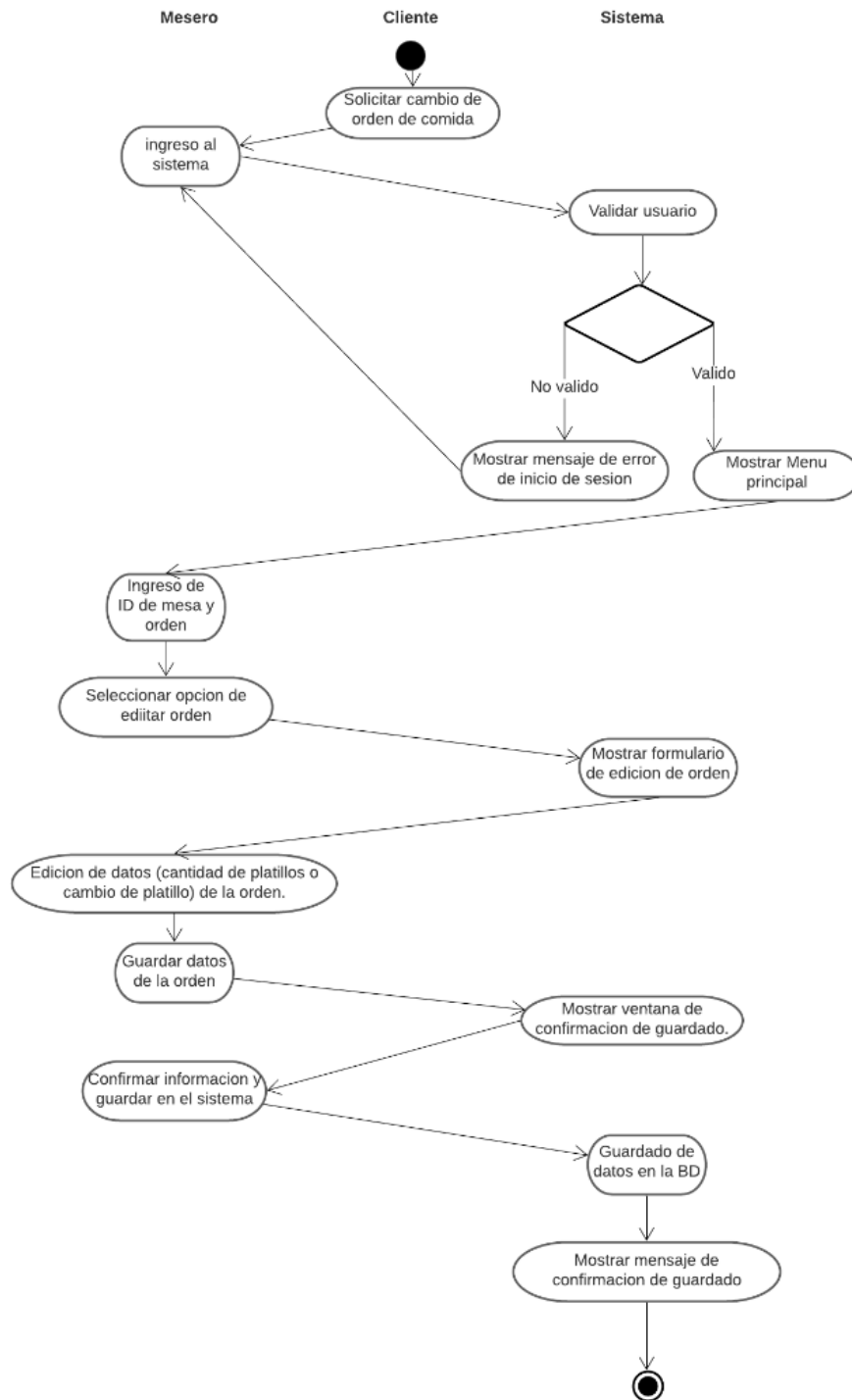
4.6.2. Diagrama de actividad Consultar una orden.



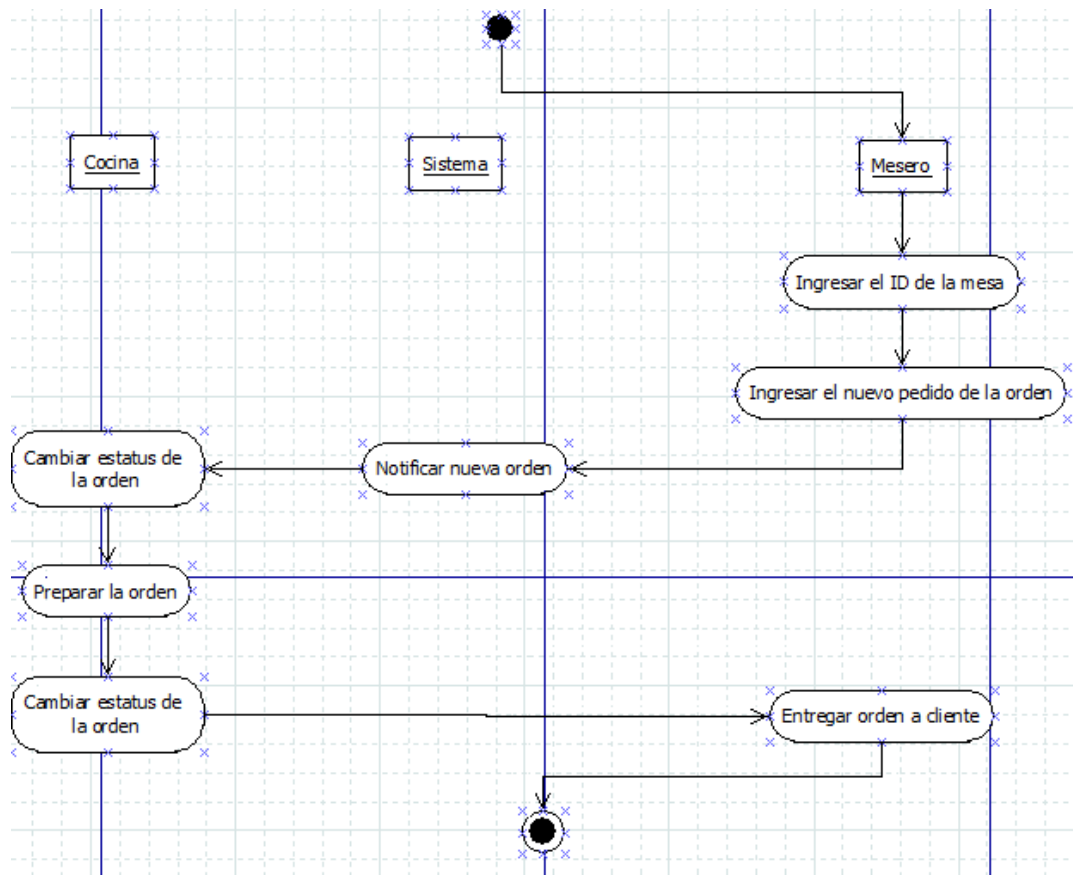
4.6.3. Diagrama de actividad Cerrar una orden.



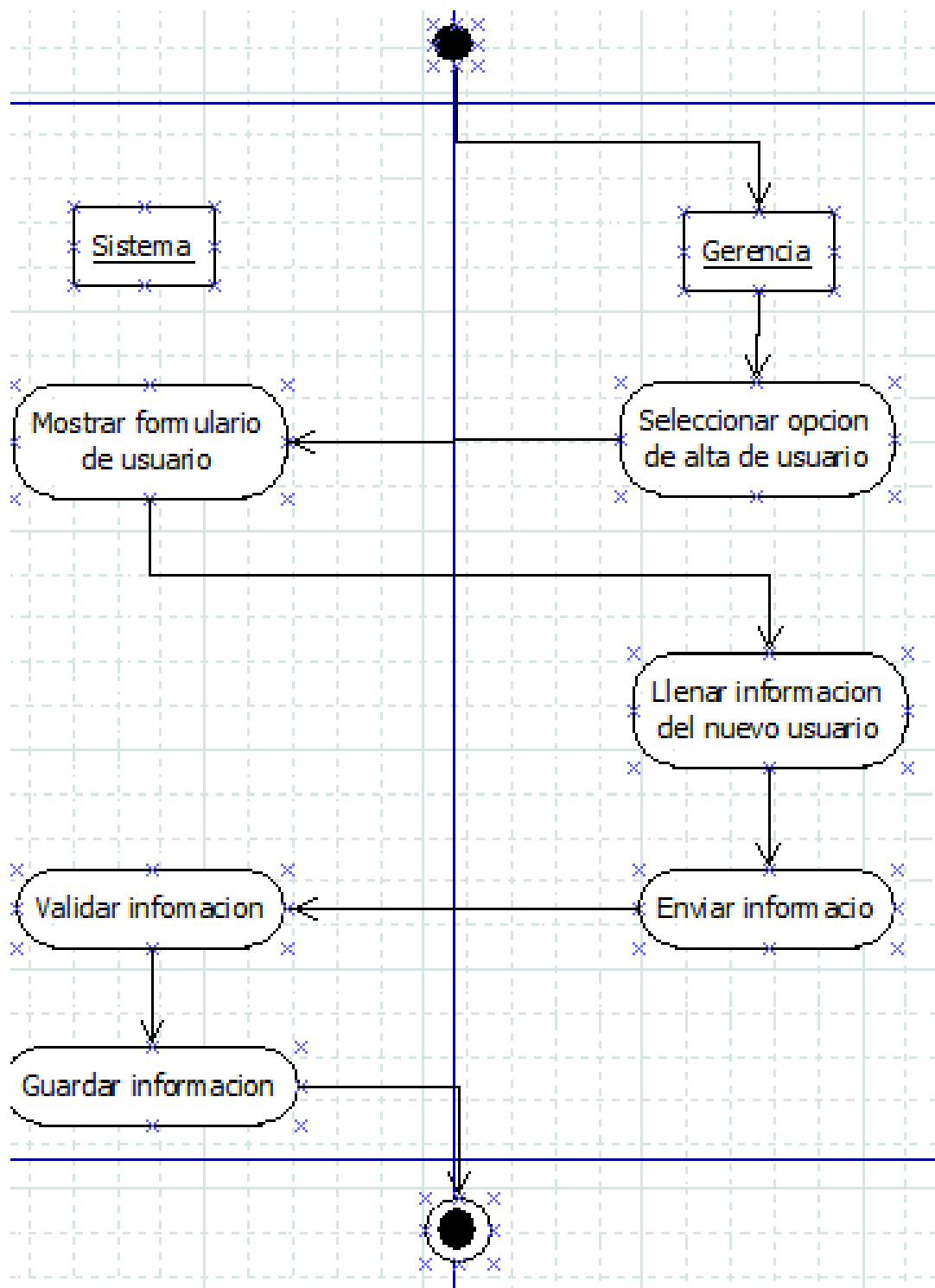
4.6.4. Diagrama de actividad Modificar una orden.



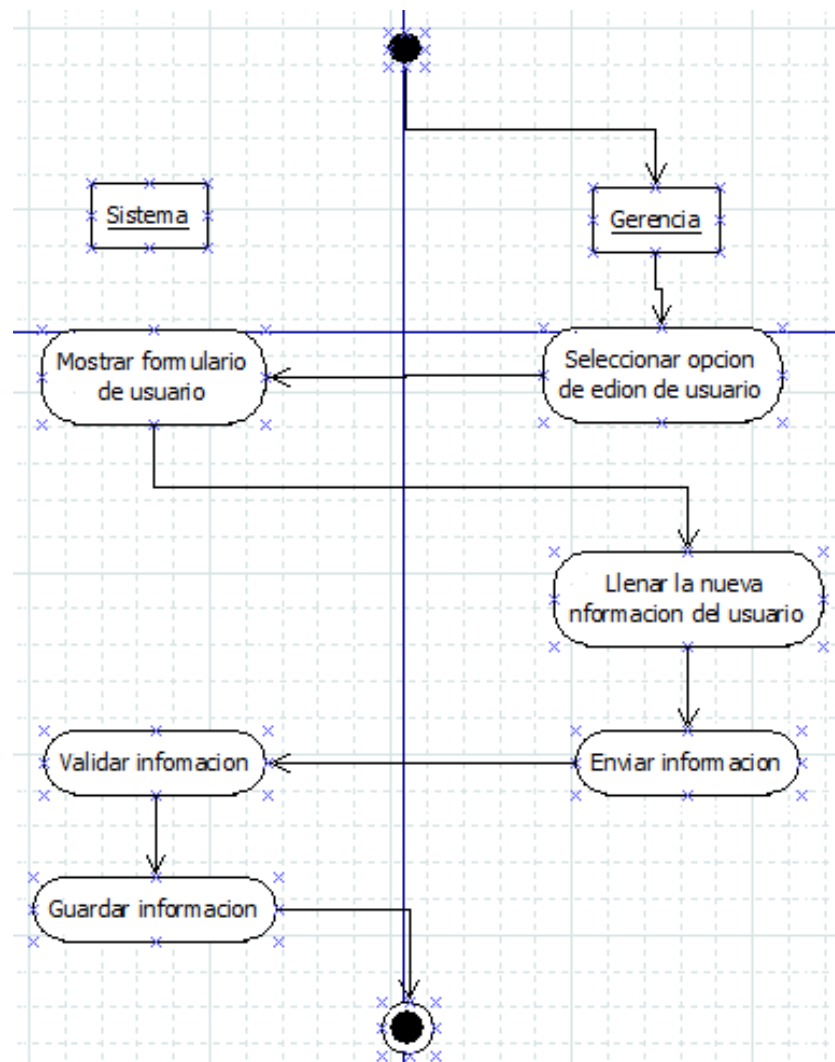
4.6.5. Diagrama de actividad Modificar el estatus de una orden.



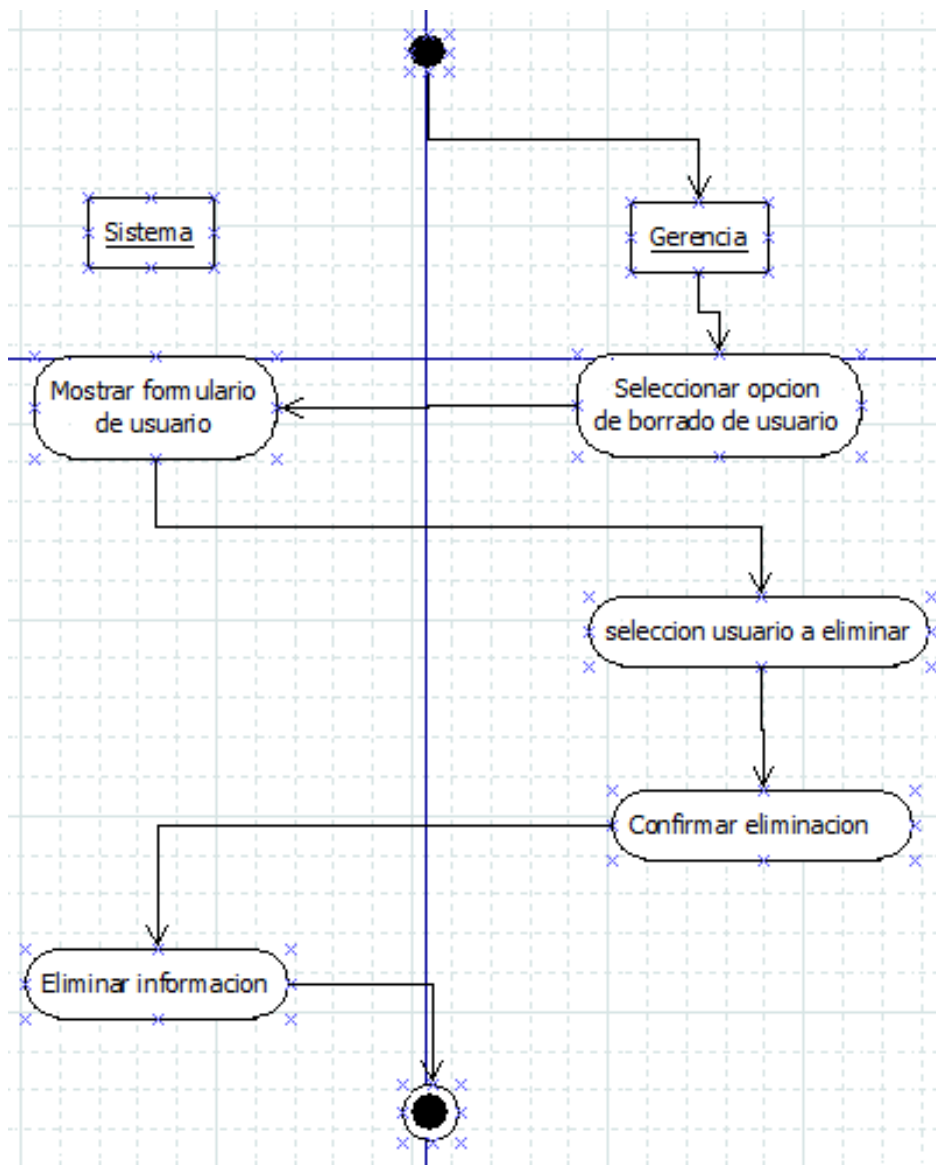
4.6.6. Diagrama de actividad Dar de alta un nuevo usuario.



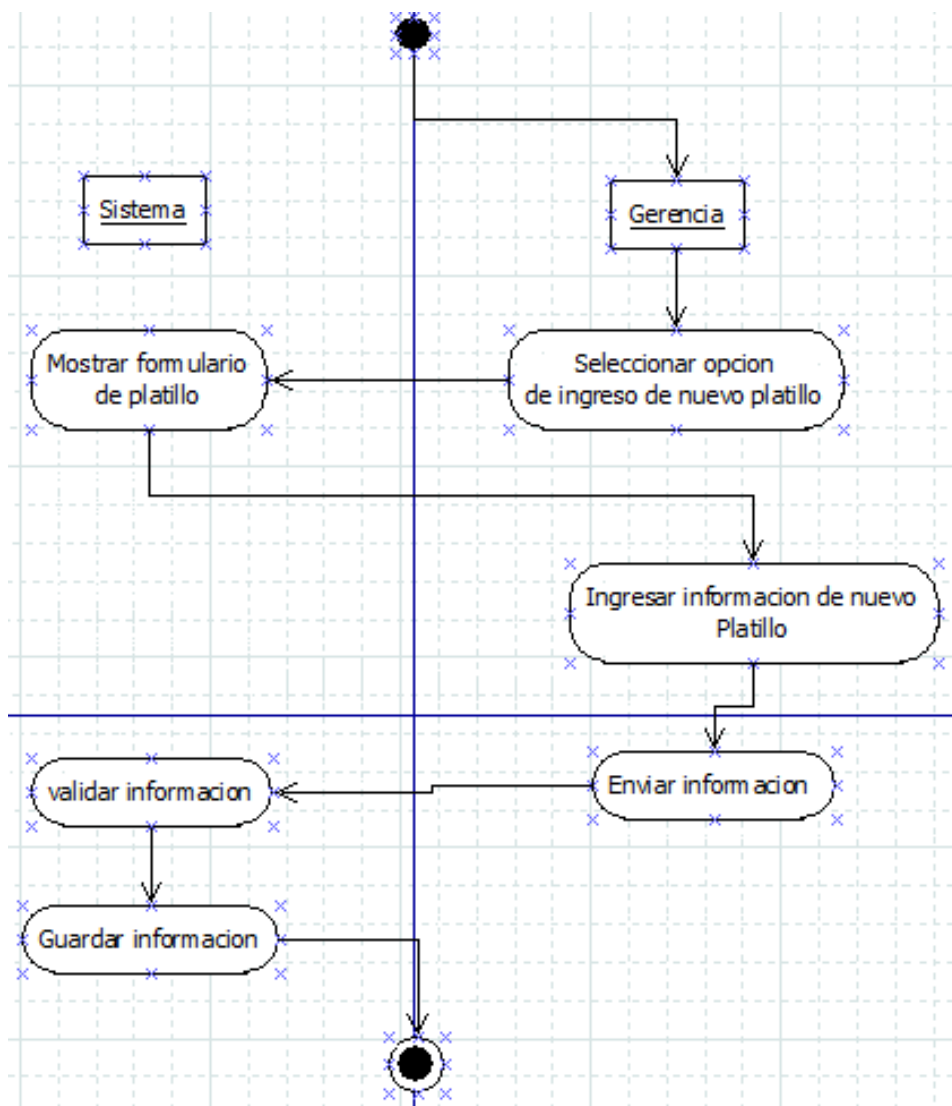
4.6.7. Diagrama de actividad de edición de usuario.



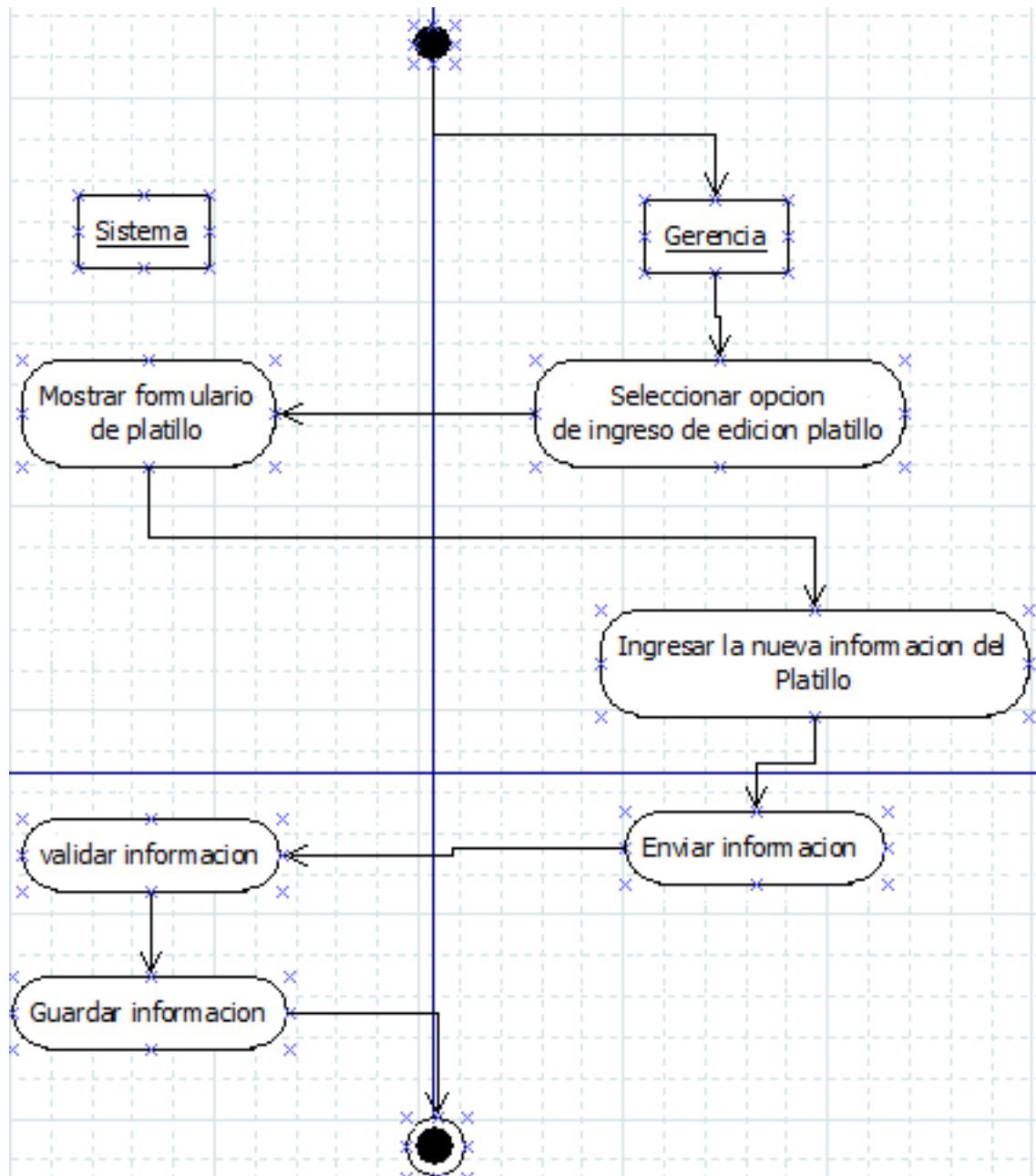
4.6.8. Diagrama de actividad de baja de usuario.



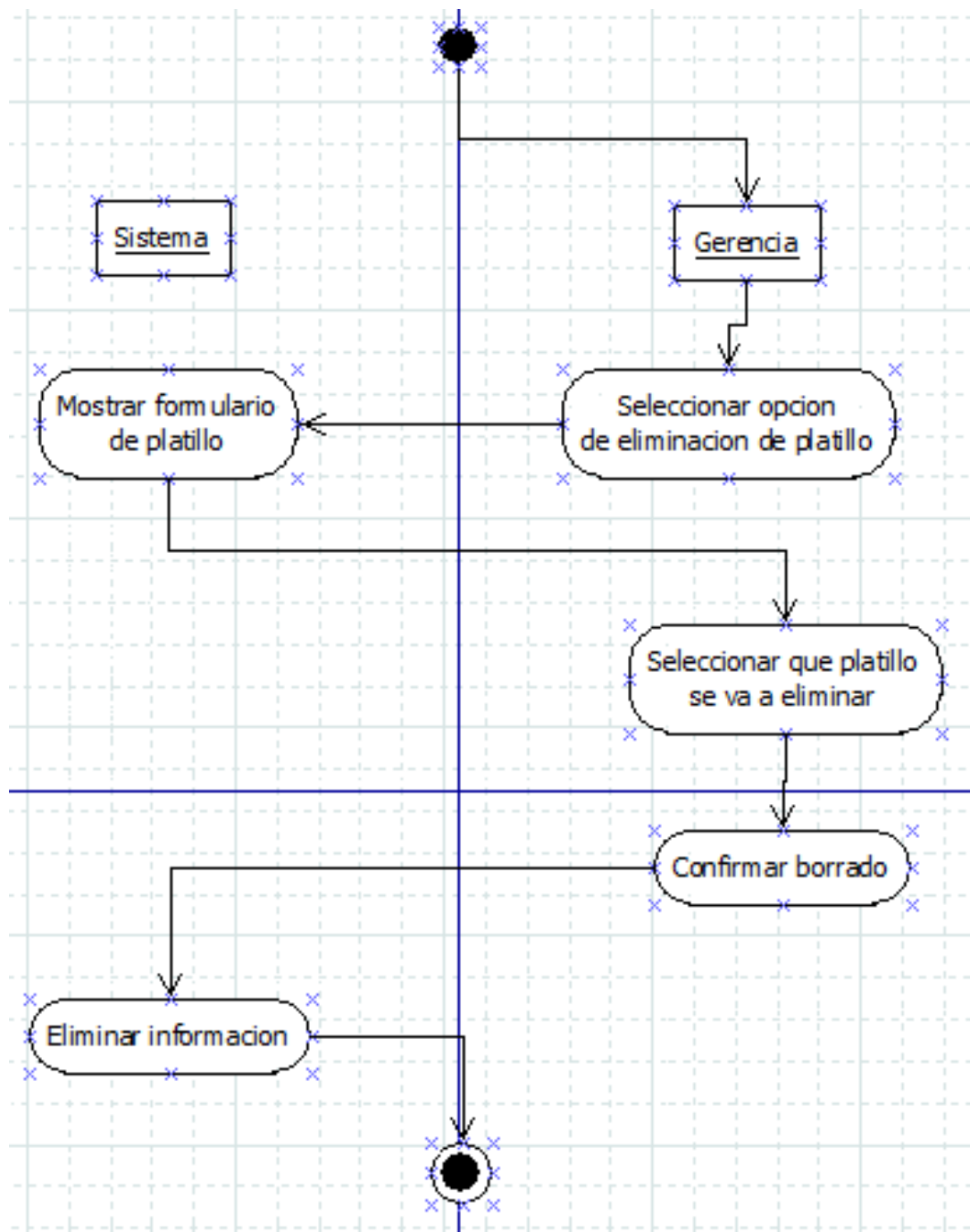
4.6.9. Diagrama de actividad de Alta de platillo.



4.6.10. Diagrama de actividad de Edición de platillo.

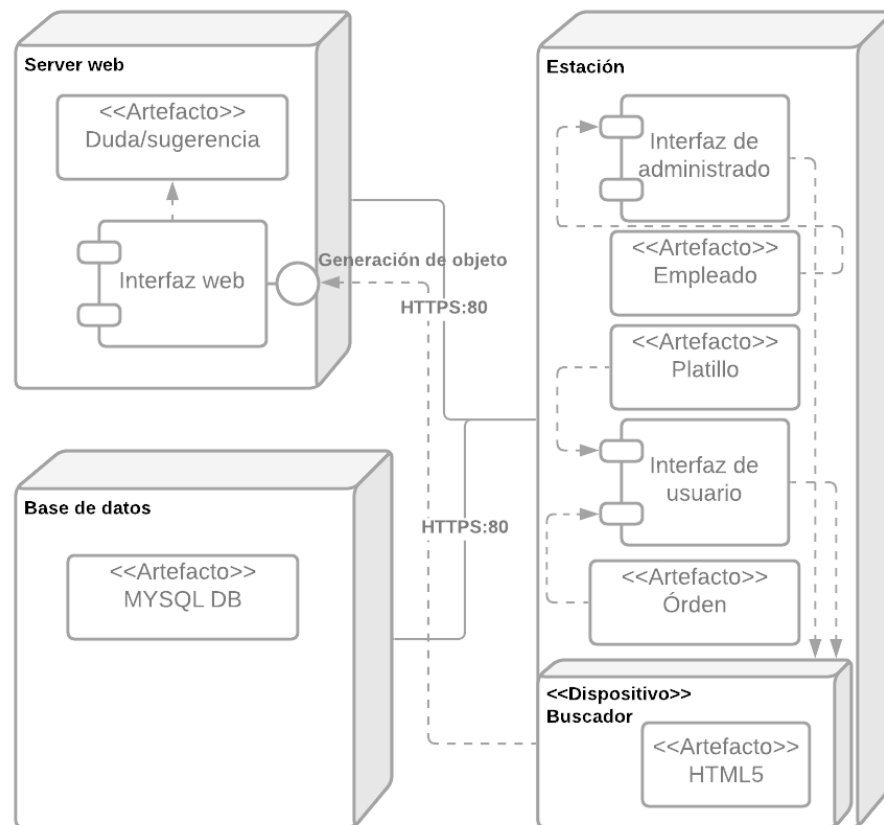


4.6.11. Diagrama de actividad de Baja de platillo.



5. Trazabilidad desde el Modelo de Diseño al Modelo de Implementación

El Proceso Unificado es un proceso de desarrollo adoptado por gran parte de las empresas desarrolladoras de software. Esto lleva a que atributos de calidad como la trazabilidad de requisitos deban estandarizarse para este proceso, con el fin de lograr los niveles de calidad exigidos por los clientes.



Usuarios y administradores

La capa de presentación muestra dos tipos distintos de interfaz, la interfaz que se tenga dependerá del tipo de usuario que sea, y este será implementado por un gerente en el subsistema de gestión de usuarios.

Subsistema de gestión de dudas y sugerencias

Este subsistema es implementado al ser llamado por la interfaz de usuario un objeto de la clase de duda/sugerencia y un objeto de sucursal, esta presenta un formato simple para el llenado y envío de la duda o sugerencia y la sucursal desde la que se hizo.

Subsistema de gestión de empleados

El subsistema de gestión de empleados tiene su implementación desde el módulo administrador, esta llama objetos de sucursal y crea un objeto de la clase empleado el cuál simplemente ingresa los datos pertinentes y registra al empleado en el objeto sucursal.

Subsistema de gestión de usuarios

Este subsistema es implementado por la clase de usuario que en el caso del administrador muestra los usuarios que se encuentran en la base de datos y toda su información además de poder configurar, editar y dar de alta nuevos usuarios en el sistema. De otra forma simplemente ingresa e implementa otros subsistemas de manipulación de información.

Subsistema de gestión de órdenes

Este es implementado por los módulos de usuario y administrador que pueden manipular órdenes y recuperar de la base de datos la información necesaria de una orden específica.

Subsistemas de gestión de platillos

Este subsistema tiene su implementación por parte del administrador y el tipo de usuario que puede agregar nuevos platillos, editarlos o eliminarlos además de poder ver cada uno.

6. Vista del Modelo de Implementación

El modelo de implementación identifica los componentes físicos de la implementación para que puedan comprenderse y gestionarse mejor.

6.1. Subsistemas

6.1.1. Gestión de dudas y sugerencias

Este subsistema es primeramente usado por la interfaz de usuario que lo ejecuta, luego se dirige al buscador web y envía la petición al servidor web, se conecta con el nodo del servidor web donde estará la creación de la duda y el envío al correo indicado.

6.1.2. Gestión de empleados

Los empleados solo pueden ser gestionados por la interfaz de administrador por lo que este se dirigiría al buscador web donde se conectará con el servidor web y se le presentará la interfaz del servidor donde ingresará al nuevo empleado y se agregará al servidor de la base de datos.

6.1.3. Gestión de usuarios

La gestión de usuarios es hecha por el sistema, aquí cada vez que es utilizada la interfaz de usuario se hace una validación de usuario en la base de datos y esta indicará el tipo de usuario que es.

6.1.4. Gestión de órdenes

Las órdenes son creadas y gestionadas por la interfaz de usuario, esta se conecta con el server web que a su vez se conecta con la base de datos y registra las órdenes.

6.1.5. Gestión de platillos

Los platillos son creados y gestionados por la interfaz de usuario, este se conecta con el server web y con la base de datos para obtener y registrar platillos.

6.2. Componentes

6.2.1. Aplicación web

La aplicación web se conecta con el subsistema de dudas y sugerencias.

6.2.2. Aplicación de escritorio

La aplicación de escritorio se conecta con el subsistema de usuarios, ordenes, platillos y empleados.

6.3. Interfaces

6.3.1. Interfaz web

Interfaz web, el buscador tiene conexión directa con el servidor mediante el protocolo HTTPS:80 y la aplicación está escrita en Java.

6.3.2. Interfaz de usuario

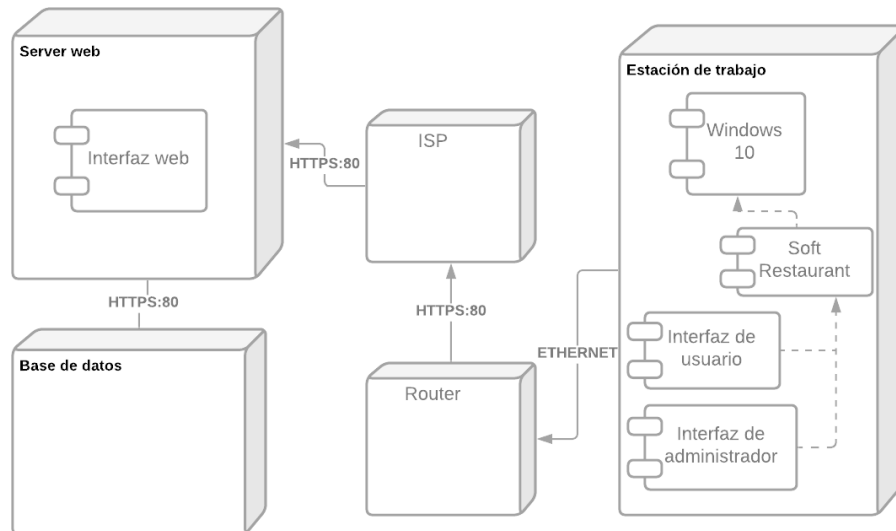
Interfaz de usuario, esta interfaz se conecta con todas las funciones de la aplicación de escritorio por lo que es capaz de gestionar órdenes y platillos además de tener que utilizar el subsistema de gestor de usuarios para poder relacionar a un usuario con sus acciones.

6.3.3. Interfaz de administrador

Interfaz de administrador, esta interfaz está conectada con todo el sistema de administración de la aplicación, tanto en el servidor web como en la base de datos es capaz de manipular los datos y la información de cada subsistema e interfaz.

7. Vista del Modelo de Distribución

7.1. Diagrama de Distribución



7.2. Nodos

7.2.1 Estación de trabajo

La estación de trabajo deberá contener los elementos necesarios para poder hacer uso del software, tener la capacidad de mantener conexión continua con el router, además, los subsistemas de gestión de usuarios, ordenes, platillos, empleados y de dudas/sugerencias.

7.2.2 Router

El router tiene la capacidad de mantener conexiones continuas entre el ISP y la estación de trabajo, deberá tener características necesarias para poder mantener un correcto funcionamiento entre los protocolos, los envíos, los puertos etc.

7.2.3 ISP

EL ISP será el proveedor de nuestro servicio de internet y tendrá que garantizarnos la fiabilidad de la conexión además de tener las características necesarias para poder tener una velocidad rápida y otros servicios de administración de redes.

7.2.4 Servidor web

El servidor web deberá de disponer de una interfaz web para su correcta administración, además de conexiones habilitadas para poder acceder a este, tendrá la capacidad de hacer correcciones, registros, bajas y modificaciones en la base de datos según sea necesario y pedido por el nodo de estación de trabajo.

7.2.5 Base de datos

La base de datos contendrá un espacio amplio para el almacenamiento de informes, reportes y registros de cada uno de los subsistemas integrados en los demás nodos.

7.3 Conexiones

7.3.1 Conexión estación de trabajo y router

La conexión entre el nodo estación de trabajo y router será por medio de cable ethernet y deberá tener una disponibilidad de 24/7 además de un ancho de banda de 360 Mb/s.

7.3.2 Conexión Router e ISP

La conexión entre el nodo Router e ISP será por medio de un cable de fibra óptica y deberá tener una disponibilidad de 24/7 además de un ancho de banda mínimo de 360 Mb/s.

7.3.3 Conexión ISP y server web

La conexión entre el nodo ISP y server web será por el medio que el ISP establezca, el protocolo será HTTPS y deberá tener una disponibilidad de 24/7 además de un ancho de banda mínimo de 360 Mb/s.

7.3.4 Conexión entre el nodo server web y base de datos

La conexión entre el nodo server web y base de datos será por medio de un cable ethernet y deberá tener una disponibilidad de 24/7 además de un ancho de banda mínimo de 360 Mb/s.

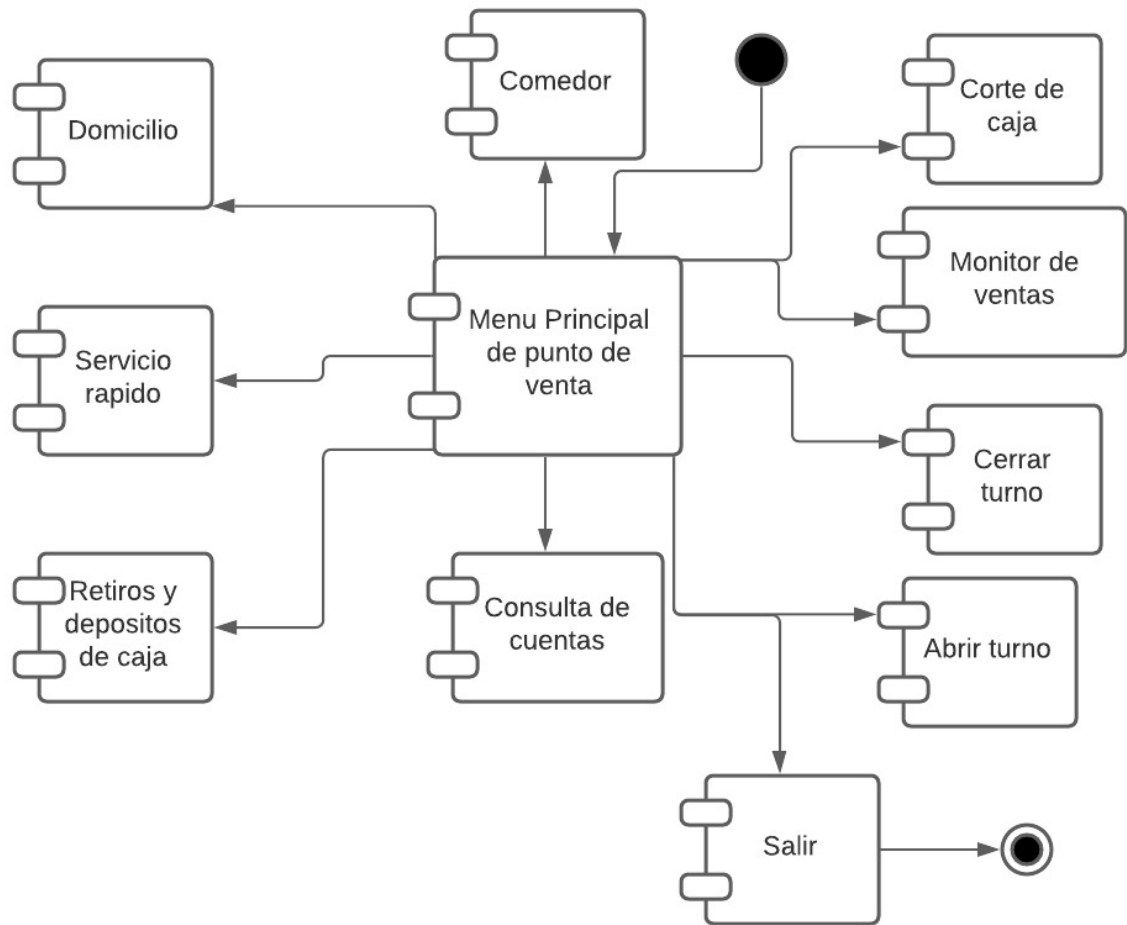


Diagrama de componentes de punto de venta

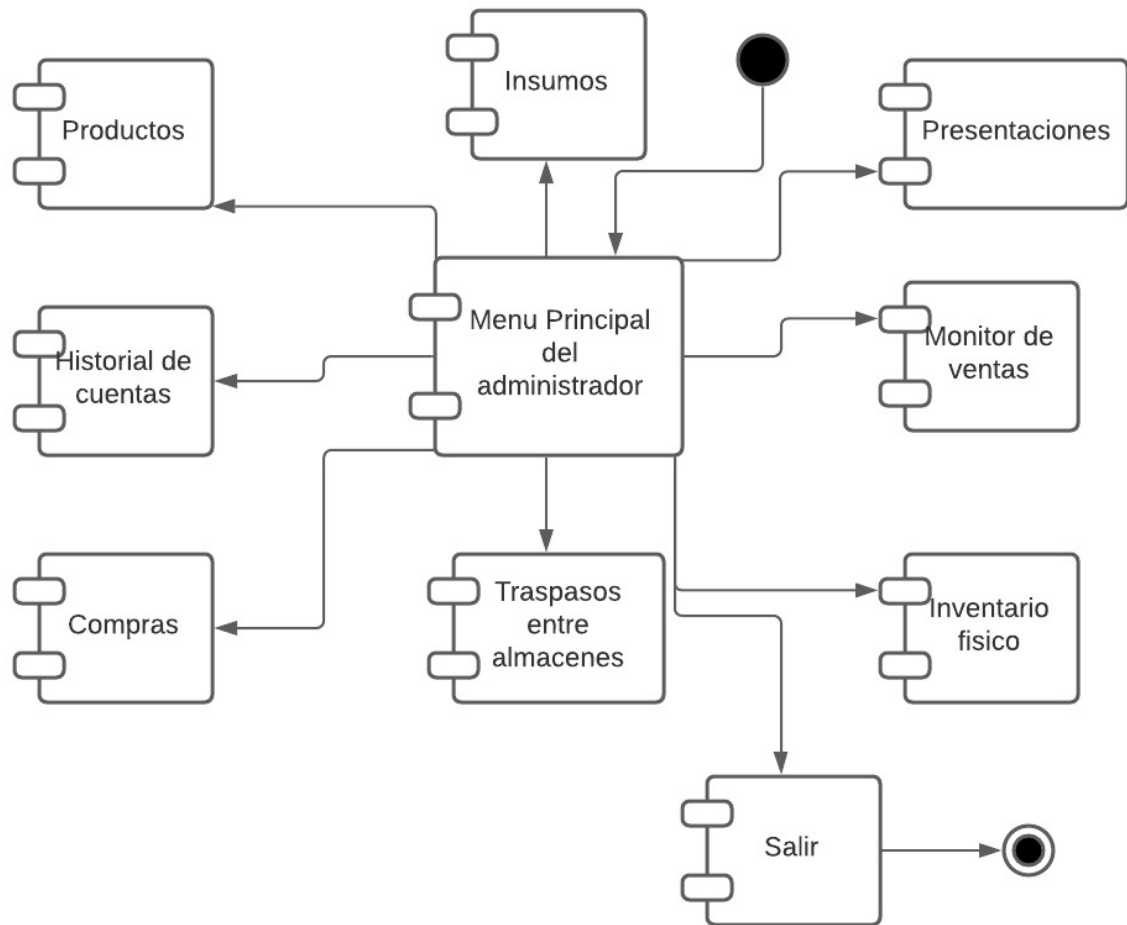


Diagrama de componentes de administrador

8. DESCRIPCIÓN DE DISEÑO DETALLADO

El diseño detallado separa la actividad de diseño a bajo nivel de la instrumentación, igual que las actividades de análisis y diseño aíslan las consideraciones de lo que se desea de la estructura que logrará los resultados deseados.

8.1. Diseño detallado del elemento 1

Subsistema de gestión de inventario

8.1.1. Vistas estructurales

La estructura de la gestión del inventario es una base de datos donde tenemos los botones para agregar un nuevo elemento al almacén y abajo se ve la base de datos del inventario.

8.1.2. Vistas de comportamiento

Al principio en el subsistema tenemos tres botones habilitados, agregar nuevo elemento, buscar elemento y cerrar la pestaña del subsistema. Al momento de elegir agregar elemento nos abre una ventana donde nos señala la hora, el número de existencias a ingresar la clave, descripción, aparte podemos agregarlo en el almacén general, la cocina y la barra, el que el usuario desee.

Después de agregar elementos se nos habilitan los otros botones que es guardar inventario, deshacer inventario, editar inventario, cancelar inventario e imprimir el inventario; los cuales son muy útiles para el subsistema de gestión de inventario.

8.1.3. Otras vistas relevantes

Otra vista relevante en el subsistema es que nos arroja los costos de almacén y nos lo calcula automáticamente, tenemos el inventario teórico, es el que se supone que debemos tener, el inventario físico, es el que tenemos en realidad, y la diferencia de estos dos inventarios. También nos da el valor total del inventario físico y por último tenemos un botón muy importante para buscar insumos.

8.1.4. Razón fundamental

La razón fundamental es que el software está dirigido para un restaurante o un bar y este subsistema es demasiado importante ya que es el inventario y en todo negocio debe de haber control del inventario. Con este subsistema se hace más sencillo y accesible.

8.2. Diseño detallado del elemento 2

Subsistemas de gestión de empleados

8.2.1. Vistas estructurales

El subsistema de gestión de empleados está compuesto por los nodos de estación d trabajo y la base de datos, en este diseño se especifican las entradas del usuario primeramente mostrada la interfaz del programa donde se inicia sesión, después se muestra la interfaz de un administrador para la gestión de los empleados y su dependencia con el componente de soft-restaurant. Aquí se hace una petición de alta de empleado y este es mandado al nodo base de datos que contendrá configurado el artefacto de la base de datos y se dará la respectiva alta de empleado.

8.2.2. Vistas de comportamiento

Las vistas de comportamiento muestran los estados por los que va cambiando la gestión de los empleados según los eventos que van ocurriendo, ya sea el alta de un empleado, la baja de un empleado o la modificación de uno. En estas vistas se muestran los actores como los meseros, los chefs y los gerentes que pueden administrar el sistema de soft-restaurant por completo y todo esto en los límites del sistema de la estación de trabajo.

8.2.3. Razón fundamental

La razón fundamental es que se necesita un subsistema para poder gestionar correctamente la nómina de la empresa y solamente ciertas personas pueden tener acceso a este tipo de información y control. Además, se necesita tener una organización de este tipo de registros que este separada de los demás subsistemas para que tenga un correcto funcionamiento y no cause problemas con los otros subsistemas.