

Manual Técnico

Autor (s):

Luis Roberto Carlos Reyes Rayas
Rosendo Ignacio Partida de León
Erick Martínez Dueñas
José Guadalupe Jiménez Ceja
Isaac Mora Ortega

**Universidad de Guadalajara
Guadalajara, Jalisco. México
11, 2021**

Índice

1. Introducción.....	3
2. Confección	3
2.1 Nombre del Sistema.....	3
2.2 Versión del Sistema	3
2.3 Tipo de Manual	3
2.4 Poner una Imagen	3
2.5 Fecha de Elaboración	3
2.6 Área donde fue elaborado	3
2.7 Índice del Contenido del Manual.....	4
2.7.1. <i>Introducción</i>	4
2.7.2. <i>Objetivos Generales y Específicos del Sistema</i>	4
2.7.3. <i>Normas, Políticas y Procedimientos</i>	4
2.7.4. <i>Definición de las Reglas del Negocio Implementadas</i>	4
2.7.5. <i>Fundamentación de la Tecnología Utilizada</i>	5
2.7.6. <i>Descripción de los Actores del Sistema</i>	5
2.7.7. <i>Especificación de Requisitos</i>	5
2.7.8. <i>Vista Funcional</i>	6
2.7.9 <i>Vista Lógica</i>	8
2.7.10. <i>Modelo Lógico de Datos</i>	10
2.7.11. <i>Modelo Físico de Datos</i>	11
2.7.12. <i>Descripción Detallada de los Algoritmos</i>	11
2.7.13. <i>Diseño de Pantallas y Reportes</i>	11
2.7.14. <i>Descripción de Campos Requeridos por Pantalla</i>	11
2.7.15. <i>Vista de Implementación</i>	11
2.7.16. <i>Vista de Despliegue</i>	12
2.7.17. <i>Diagrama de Navegación del Sistema</i>	12
2.7.18. <i>Controles de Auditoría Implementados en el Sistema</i>	12
2.7.19. <i>Glosario de Términos</i>	12
3. Estándares de Elaboración del Manual	13

1. Introducción

El Manual Técnico, como su nombre lo indica, contiene las especificaciones técnicas más importantes del sistema desarrollado. Constituye una guía especializada para la realización de las operaciones de mantenimiento de la aplicación. Se encuentra dirigido fundamentalmente a la dirección de Tecnologías de la Información, al administrador del sistema, a otros desarrolladores, así como al departamento de calidad y auditoría de sistemas.

2. Confección

Para la elaboración de un manual de técnico se deben de considerar los siguientes apartados normativos.

Nota: Los acápites que aparecen en color rojo y con descripción en letra cursiva son de carácter opcional.

2.1 Nombre del Sistema: Software Gestor de Restaurantes (Soft Restaurant)

2.2 Versión del Sistema: Versión 1.0

2.3 Tipo de Manual: El tipo de manual es técnico, el cual vendrán los objetivos generales del sistema, así como un mapa de navegación en el sistema, todo lo relevante a la tecnología utilizada al desarrollar este sistema y los protocolos, normas y políticas del mismo sistema.

2.4 Poner una Imagen: Es recomendable ilustrar el manual con una imagen representativa del sistema.



2.5 Fecha de Elaboración: 21 de noviembre de 2021.

2.6 Área donde fue elaborado: Área de documentación.

2.7 Índice del Contenido del Manual: Deberá contar con un índice y/o contenido del manual para facilitar su manejo e identificación de los puntos importantes, pues si sólo se busca un punto en específico con el índice es fácil identificarlo.

2.7.1. Introducción

Bienvenido al sistema Soft Restaurant, el sistema para restaurantes y bares, desarrollado para el sistema operativo Windows® que le ayudará a administrar de una manera práctica y sencilla su establecimiento de alimentos y/o bebidas.

El programa incorpora el punto de venta en servicio comedor, servicio a domicilio, servicio rápido, Comandero, facturación, caja, inventarios, costos y todo un conjunto de reportes y estadísticas.

Soft Restaurant cubre las operaciones de su restaurante y le ayuda a ofrecer un mejor servicio al reducir el tiempo de entrega de cuentas al cliente, evitando devoluciones, omisiones en la cuenta, sumas mal realizadas, cálculos manuales, etc. Soft Restaurant opcionalmente imprime las comandas en diversos puntos de su restaurante como la cocina o barra o los lugares que usted defina, limitando así, el tráfico de meseros en las áreas mencionadas solo para recoger las ordenes preparadas y agilizando la preparación de las órdenes.

Mediante los reportes de Soft Restaurant, obtenga al instante información como ventas, existencias de inventario, ventas por mesero, productos vendidos, etc.

2.7.2. Objetivos Generales y Específicos del Sistema

Breve descripción de los objetivos generales y específicos que se cumplieron con el desarrollo del sistema.

Con este documento hemos contenido toda la información acerca de los recursos utilizados por este proyecto, además de incluir las pautas de configuración y lógica para su desarrollo. Se describen las características físicas y técnicas de cada elemento que compone el sistema.

2.7.3. Normas, Políticas y Procedimientos

El sistema de Soft Restaurant se basa en normas como ISO/IEC9126 para el diseño de la interfaz, la IEEE 29119 para el diseño del plan de pruebas y la ISO/IEC 14598. Base metodológica para la evaluación del producto software. El desarrollo de este proyecto siguió la estrategia establecida por nuestro equipo de trabajo haciendo uso de estándares para la realización de software.

2.7.4. Definición de las Reglas del Negocio Implementadas

Estos son los lineamientos contemplados para garantizar la satisfacción de las

restricciones que existen en el negocio

- Hemos seguido el planteamiento que nos propone la metodología de desarrollo RUP-UML para poder asignar las tareas y responsabilidades de forma disciplinada, según las restricciones propuestas, además de poder hacer el mantenimiento más sencillo y mejorar los tiempos de desarrollo para poder cumplir con las restricciones de tiempo.
- La reutilización para una mejor organización del proyecto y menos utilización de recursos y tiempo.
- La utilización de distintas vistas para mostrar los distintos aspectos del programa desarrollado y de esta forma se pueda comprender mejor la estructura y comportamiento del sistema.

2.7.5. Fundamentación de la Tecnología Utilizada

Realizar una breve descripción, con sus correspondientes referencias bibliográficas, acerca de las tendencias y tecnologías actuales sobre las que se apoya la propuesta, además de incluir la justificación de las seleccionadas para el desarrollo de la aplicación.

Para la realización de este software hemos utilizado JAVA para poder desarrollar la parte funcional del programa, MYSQL para poder diseñar y construir una base de datos que se pueda utilizar con nuestro programa y Windows 10 como sistema operativo para poder hacer las pruebas, ya que en este entorno probablemente sea donde más se utilice el software.

Hemos seguido estas tecnologías debido a que se han estado fortaleciendo mucho en la nube, creando nuevas listas de herramientas y servicios en distintas plataformas relacionadas, Windows sigue siendo el sistema operativo más utilizado y MySQL la que mayor tasa de utilización tiene. (evolbit, 2007) (programadorclic, 2019) (EUROINNOVA BUSINESS SCHOOL, s.f.)

2.7.6. Descripción de los Actores del Sistema

Tabla1.Definición Resumida de los Actores del Sistema

Tipo de usuario	Administrador
Formación	Preparatoria / Licenciatura
Habilidades	<ul style="list-style-type: none">- Manejo de dispositivos electrónicos de nivel intermedio- Administración de la empresa- Visión general más completa de las actividades realizadas dentro del establecimiento.
Actividades	<ul style="list-style-type: none">- Registro de empleado- Consulta de información del empleado- Edición de la información de un empleado- baja de un empleado- alta de platillos- modificar de platillos- baja de platillos- alta de ingredientes- Modificación de ingredientes- baja de ingredientes

Tipo de usuario	Mesero
Formación	Secundaria / Preparatoria / Licenciatura
Habilidades	<ul style="list-style-type: none">- Retención de información- Mayor uso de dispositivos electrónicos.- Aprendizaje rápido- Fácil adaptación
Actividades	<ul style="list-style-type: none">- Registro de orden- consulta de orden- Edición de orden- cierre de orden

Tipo de usuario	Cocinero
Formación	Secundaria / Preparatoria / Licenciatura
Habilidades	<ul style="list-style-type: none"> - Retención de información - Poco uso o nulo uso de dispositivos electrónicos - Versatilidad con utensilios - Fácil aprendizaje
Actividades	<ul style="list-style-type: none"> - Consulta de orden - Modificación de estatus de la orden - Notificación de orden

2.7.7. **Especificación de Requisitos**

I. *Descripción de los Requisitos Funcionales.*

Requisito funcional 1: Poder crear de forma organizada órdenes.

El sistema debe ser capaz de crear de forma completamente organizada órdenes para las mesas, siendo este requisito uno de los principales requerimientos a desarrollarse y testearse, la información introducida se guardará en la Base de datos.

Procedimiento para seguir:

- 1.- En el menú principal el usuario (Mesero) da clic en el apartado "crear orden".
- 2.- El sistema muestra en pantalla el formulario "NUEVA ORDEN". 3.- El usuario (Mesero) Ingresa en el formulario "NUEVA ORDEN" el Id de la mesa.
- 4.- El sistema muestra en pantalla el menú disponible para ese día / turno.
- 5.- El usuario (Mesero) selecciona del menú mostrado en pantalla lo que cada comensal le solicitó al tomar la orden en la mesa.
- 6.- El usuario (Mesero) selecciona la cantidad de cada platillo solicitado.
- 7.- En caso de que haya alguna "peculiaridad" en el platillo el usuario (Mesero) anota en el apartado de comentarios tal especificación.
- 8.- Una vez terminado de cargar la información en el formulario "NUEVA ORDEN", el usuario (Mesero) da clic en el botón "ENVIAR ORDEN".
- 9.- El sistema muestra en pantalla el mensaje "Orden guardada exitosamente".

Respuesta a situaciones anormales:

- Mesa ya tiene una orden abierta
- Orden no permitida
- Fallo de comunicación
- Mesa no disponible

Requisito funcional 2: Guardar platillos con sus costos.

El sistema debe ser capaz de guardar la información de los platillos ofertados en el establecimiento de manera eficiente.

Procedimiento a seguir:

- 1.- El usuario Administrador (Gerente) ingresa al menú **"MENU"** en el sistema (previamente logeado para ello).
- 2.- El sistema muestra en pantalla las opciones en ese apartado.
- 3.- El usuario Administrador (Gerente) ingresa al apartado **"Alta Platillo"**.
- 4.- El sistema muestra en pantalla el formulario "Alta Platillo".
- 5.- El usuario Administrador (Gerente) llena el (los) campo(s) solicitado(s) por el sistema para agregar un platillo al menú.
- 6.- El usuario Administrador (Gerente) una vez que termina de llenar TODOS los campos solicitados en el formulario "Alta Platillo", da clic en el botón **GUARDAR**.
- 7.- El sistema muestra en pantalla el mensaje "Información guardada correctamente".

Requisito funcional 3: Saber fácilmente el total de las órdenes de los comensales/mesas.

El sistema debe ser amigable con el usuario de manera que al querer saber el total de las ordenes, pueda ser de una forma fácil y eficaz, desplegando que platillo se pidió, cuanto costo, cuantos platillos pidió y sus comentarios adicionales si es que existieron.

Procedimiento para seguir:

1. En el menú principal el usuario da clic en el apartado "consultar orden".
2. El sistema requiere a través de un formulario los datos de fecha y número de orden.
3. El sistema despliega las órdenes de la fecha seleccionada si el usuario no ingresa el número de orden y da clic en aceptar.
4. El usuario deberá seleccionar un número de orden de la lista desplegada por el sistema, si el usuario no agregó manualmente el número de orden.
5. El sistema muestra los datos a verificar por el usuario de la orden a buscar.
6. El usuario deberá dar clic en el botón aceptar.
7. El sistema mostrará en pantalla los datos de la orden seleccionada.

Requisito funcional 4: Control de inventarios

El sistema debe ser capaz de llevar un control de inventarios sobre lo que hay en existencia en el restaurante.

- 1.- En el menú principal, el usuario da clic en el apartado "Inventario"
- 2.- Se despliega una tabla con todos lo que hay actualmente disponible en el inventario.

Requisito funcional 5: Perfiles de seguridad con accesos restringidos para el colaborador.

El sistema debe ser seguro, siendo capaz de tener un control de permisos de usuario y así evitar comprometer la información a la que otros usuarios no debe estar disponible.

Funcionalidades que el sistema debe poseer.
II. *Descripción de los Requisitos No Funcionales.*

Requisitos de rendimiento

- el número de terminales: se estima contar con 6 terminales
- el número esperado de usuarios simultáneamente conectados: hasta 10 usuarios conectados
- número de transacciones por segundo que deberá soportar el sistema: el 85% de las transacciones deben realizarse en menos de 1 segundo

Seguridad

- Empleo de técnicas criptográficas.
- Registro de ficheros con "logs" de actividad.
- Asignación de determinadas funcionalidades a determinados módulos.
- Restricciones de comunicación entre determinados módulos.
- Comprobaciones de integridad de información crítica.

Fiabilidad

- probabilidad
- un funcionamiento adecuado
- calificación con respecto al entorno
- tiempo
- total de incidentes permisible

Disponibilidad

La disponibilidad del sistema debe ser continua con un nivel de servicio para los usuarios de 7 días por 24 horas, garantizando un esquema adecuado que permita la posible falla en cualquiera de sus componentes, contar con una contingencia, generación de alarmas.

Mantenibilidad

Capacidad para ser modificado. Capacidad del producto que permite que sea modificado de forma efectiva y eficiente sin introducir defectos o degradar el desempeño.

1. El número de peticiones de cambio del sistema. Los cambios al sistema, por lo general, corrompen la estructura del sistema y dificultan cambios futuros. Cuanto más alto sea este valor acumulado, más baja será la calidad del sistema. (usuario)
2. El número de interfaces de usuario. Éste es un factor importante en los sistemas basados en formas, donde cada forma puede considerarse como una interfaz de usuario separada. Cuantas más interfaces haya, más probabilidad habrá de que existan inconsistencias y redundancias en dichas interfaces. (usuario)
3. El volumen de datos usados por el sistema. Cuanto más alto sea el volumen de datos (número de archivos, tamaño de base de datos, etcétera), más probable será que haya inconsistencias de datos que reduzcan la calidad del sistema. (Desarrollador)
4. Número de peticiones de cambio pendientes. Con el tiempo, un aumento en este número implicaría un declive en la mantenibilidad. (desarrollador)

- generación de estadísticas de acceso mensuales.

Portabilidad

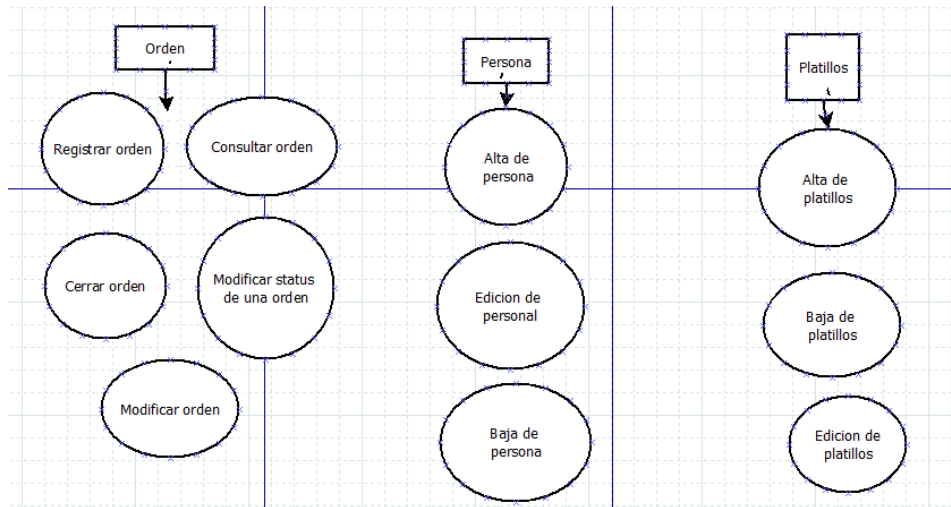
- Guardado de un ejecutable de instalación en una unidad de almacenamiento.
- Sistema operativo Windows 7 o posterior para su instalación de software.

2.7.8. Vista Funcional

Diagrama de los casos de uso del sistema estructurado que comprometen la arquitectura, incluyendo los de mayor importancia para el usuario. Identificar relaciones de inclusión, extensión o generalización entre los casos de uso en caso de existir, así como, generalización entre actores si está presente.

Vista del Modelo de Casos de Uso

Diagrama de Casos de Uso relevantes a la Arquitectura



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.

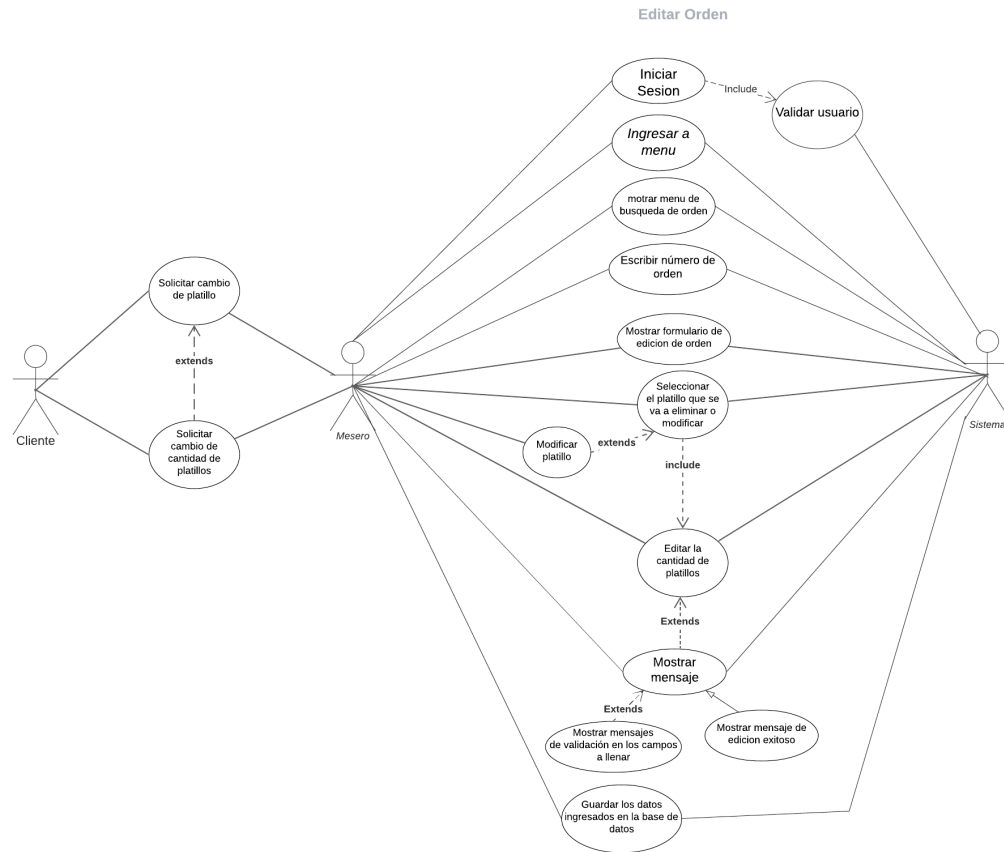
Registrar una orden.

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.



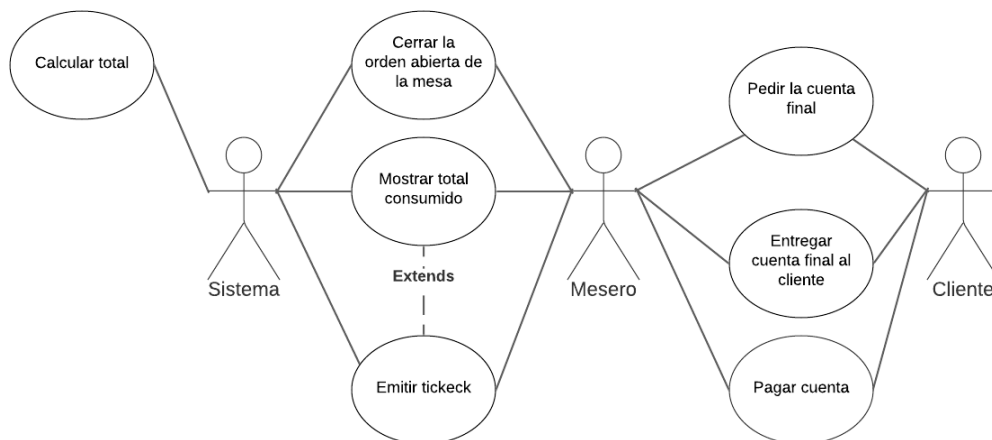
Edición de una orden

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).



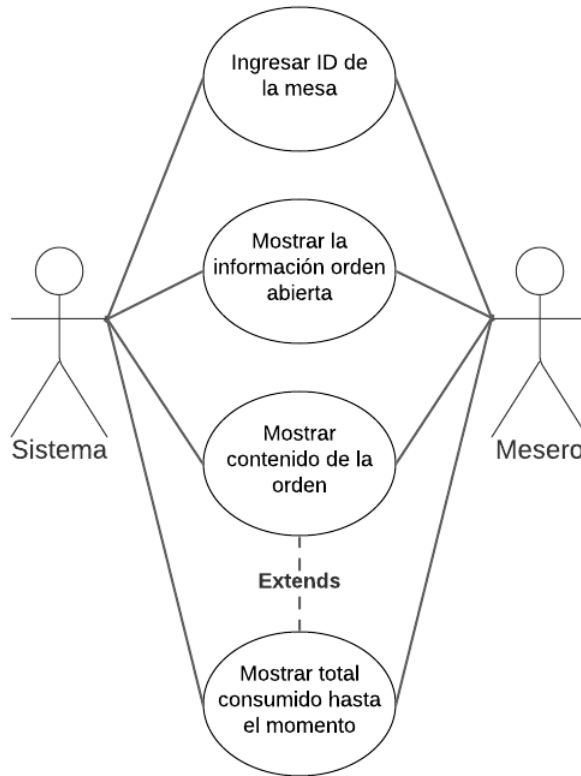
Cerrar una orden

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.



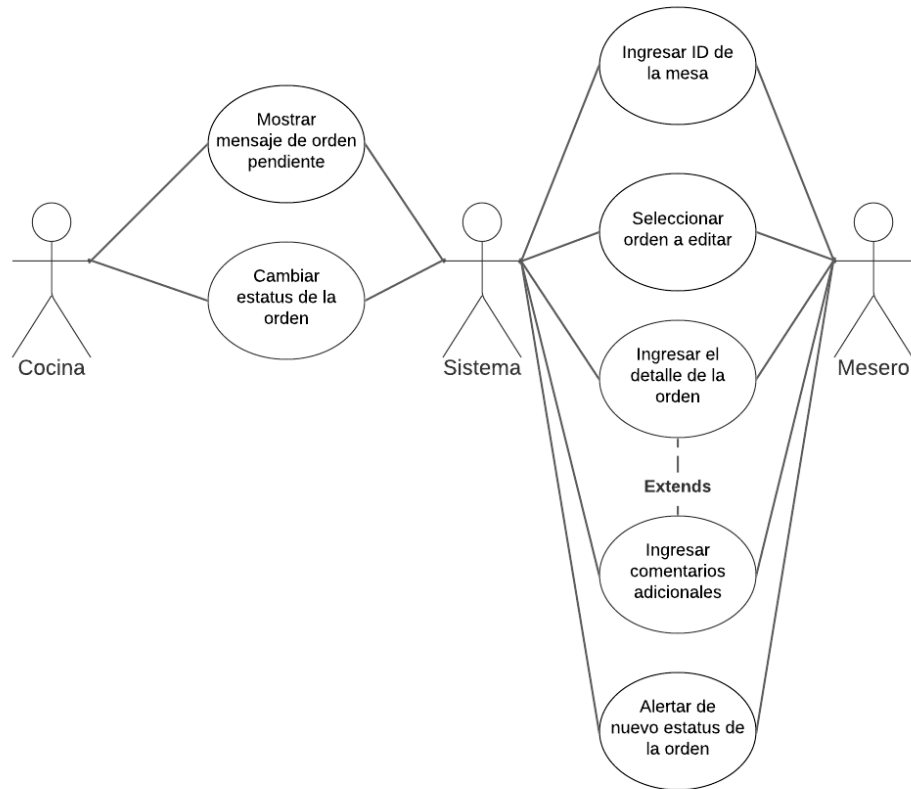
Consultar una orden y su contenido

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



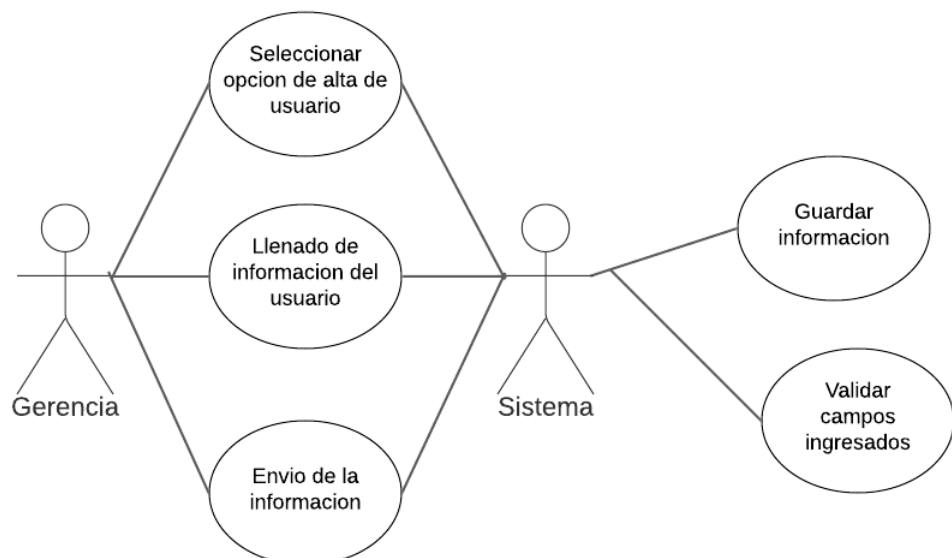
Modificar el estatus de una orden (contenido).

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)



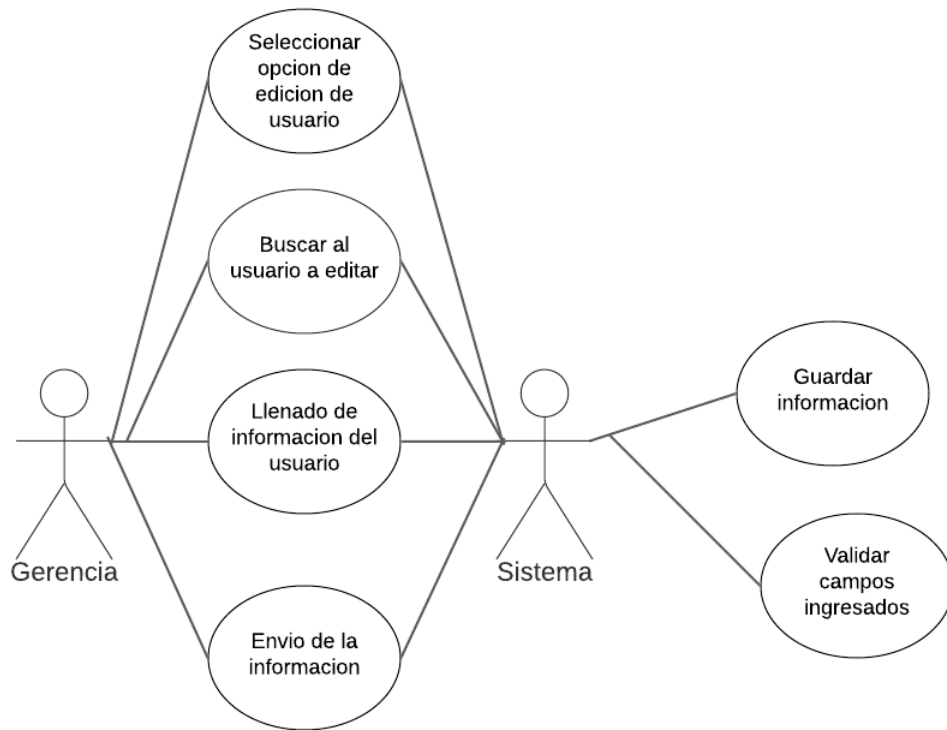
Alta de usuario

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.



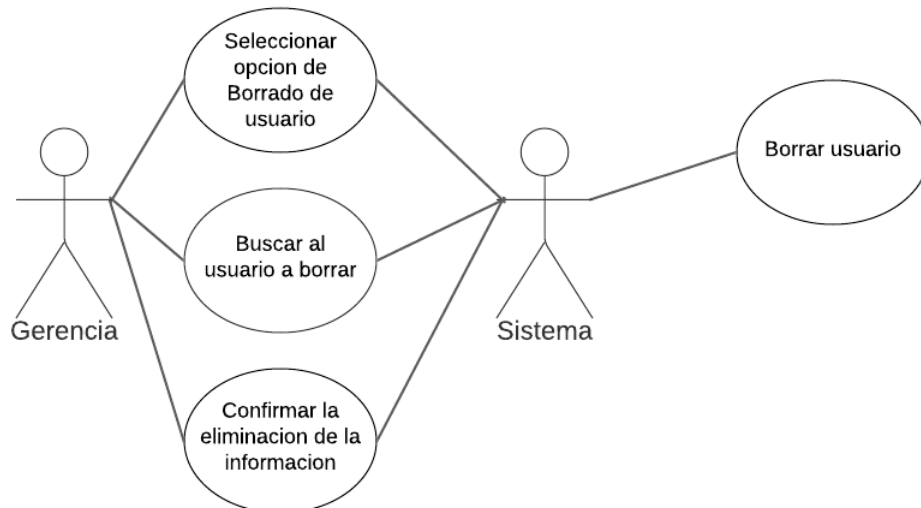
Edición de usuario.

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.



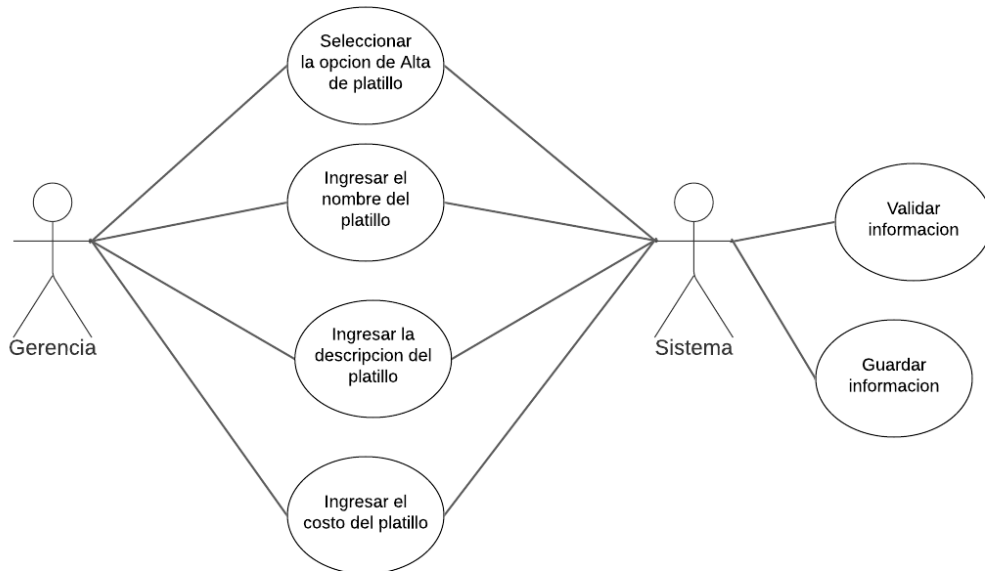
Borrado o baja de usuario.

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.



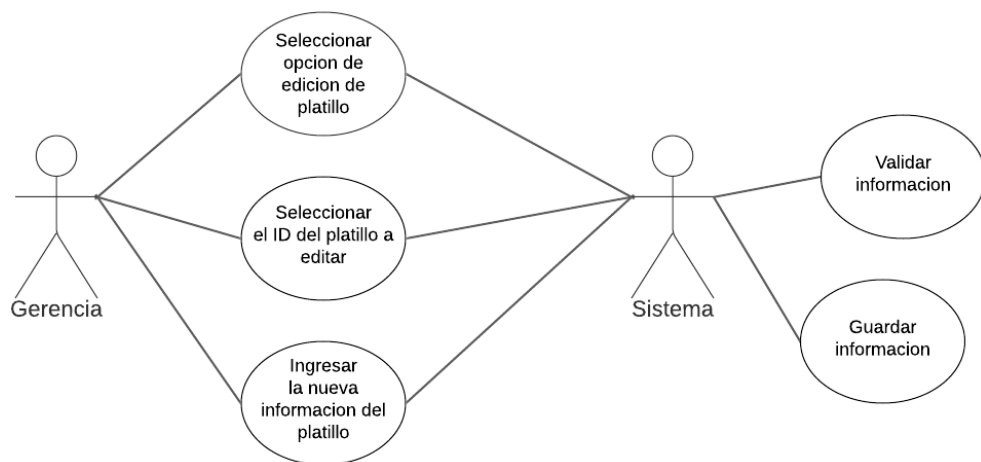
Alta de platillo

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



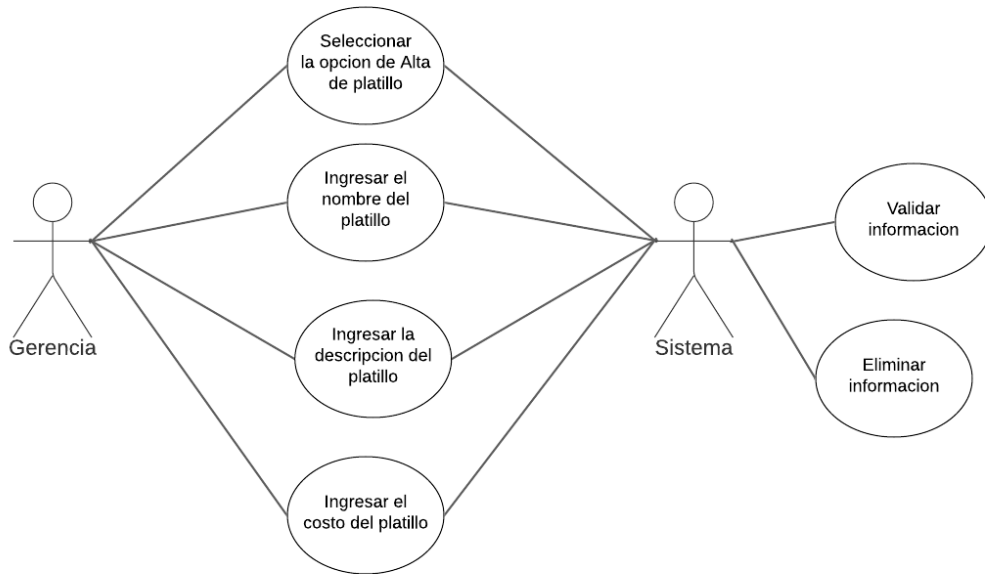
Edición de platillo.

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.



Baja o eliminación de platillo.

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



2.7.9 Vista Lógica

Patrones de Diseño

Un patrón de diseño es un conjunto de reglas que describen como afrontar tareas y solucionar problemas que surgen durante el desarrollo del software. Existen varias clasificaciones de los patrones, una de ellas es según su finalidad, dentro de la cual se consideran 3 conjuntos fundamentales:

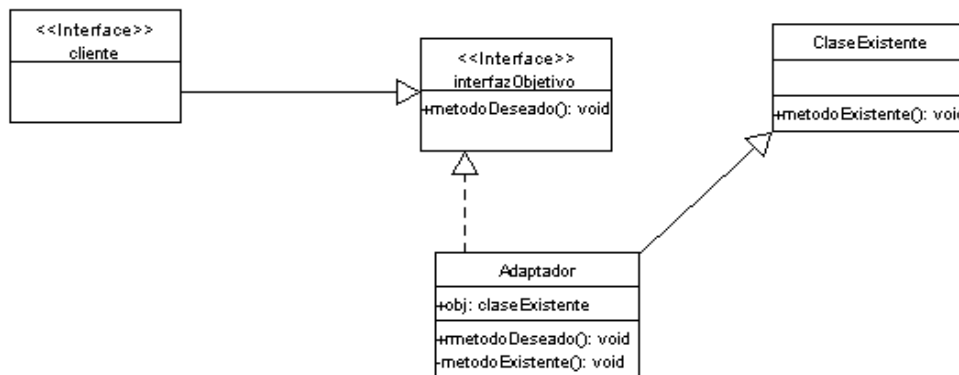
- Patrones de Creación

Estos patrones crean objetos, evitando dicha función y la instanciación directa por parte de los desarrolladores. Ello proporciona a nuestros programas una mayor flexibilidad para decidir que objetos usar. Dentro de este grupo los más conocidos son: Factory, Abstract Factory, Builder, Prototype y Singleton.

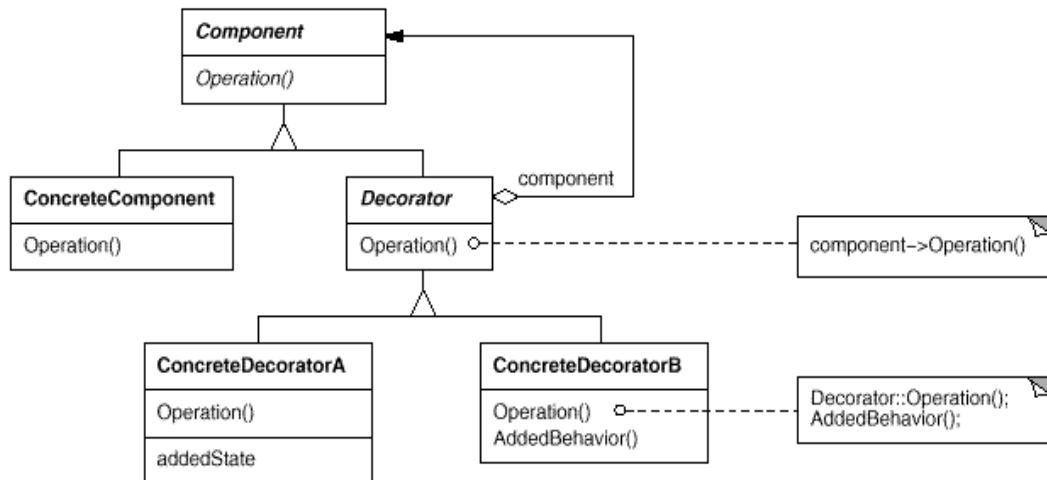
- Patrones Estructurales

Guían el desarrollo de la aplicación bajo una determinada estructura. Entre ellos se encuentran: Adapter, Bridge, Composite, Decorator, Facade, Flayweight y Proxy.

✓ Estructura del Patrón *Adapter*



✓ Estructura del Patrón *Decorator*



- Patrones de Comportamiento

Enfocados principalmente en algoritmos y en la asignación de responsabilidades entre los objetos. Describen el comportamiento de objetos, clases y la comunicación entre ellos.

Se clasifican

- ✓ **A nivel de clases:** Usan la herencia como vía para distribuir el comportamiento. El Interpreter y el Template Method constituyen ejemplos de este tipo de patrón.
- ✓ **A nivel de objetos:** Usan la composición, más que la herencia para llevar a cabo las tareas. Muestra de ellos son el Command, Iterator, Chain of Responsibility, Strategy y el State.

Patrones de Diseño Detallados

El patrón de diseño usado en el desarrollo de este proyecto consta del modelo MVC. MVC es un patrón de diseño que se estructura mediante tres componentes: modelo, vista y controlador. Este patrón tiene como principio que cada uno de los componentes esté separado en diferentes objetos, esto quiere decir que los componentes no se pueden combinar dentro de una misma clase.

Antes que nada, MVC separa la lógica de negocios y la capa de representación entre sí. Tradicionalmente se utilizaba para interfaces gráficas de usuario de escritorio. Hoy en día, la arquitectura MVC se ha hecho popular para el diseño de aplicaciones Web y aplicación móvil. Como se mencionó anteriormente, el patrón de diseño MVC es constituido por tres componentes:

Modelo: Contiene sólo los datos de aplicación más puros, no contiene ninguna lógica que describa cómo presentar los datos a un usuario.

Vista: presenta los datos del modelo al usuario. La vista sabe cómo acceder a los datos del modelo, pero no sabe qué significa esta información o qué puede hacer el usuario para manipularla.

Controlador: Existe entre la vista y el modelo. Escucha los eventos desencadenados por la

vista y ejecuta el procedimiento adecuado a estos eventos. Por ejemplo, un controlador puede actualizar un modelo cambiando los atributos de un personaje en un videojuego, así como también puede modificar la vista mostrando el personaje actualizado en el juego. Las tres partes de MVC están interconectadas; la vista muestra el modelo para el usuario, después el controlador acepta la entrada del usuario y actualiza el modelo y debido a esa acción la vista vuelve a tener un cambio con los datos actualizados.

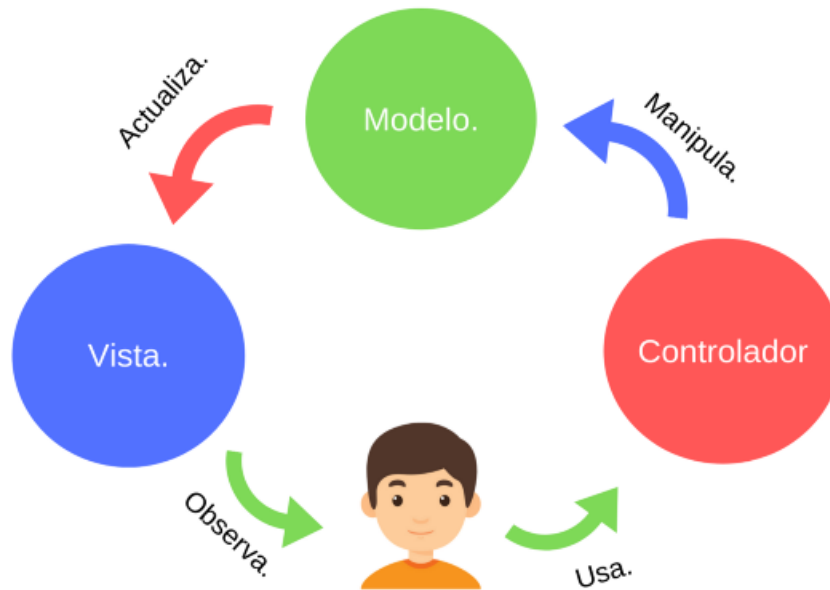
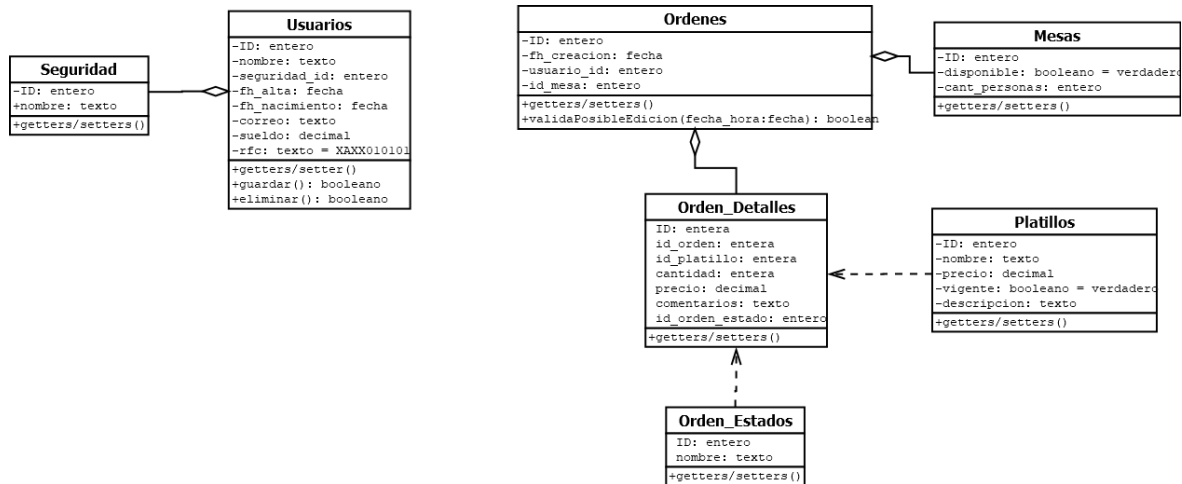


Imagen ejemplo simplificado del modelo MVC

Diseño de Clases

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



Clase Usuarios

La clase Usuarios es una representación de los usuarios que manipularán 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.

Usuarios
-ID: entero -nombre: texto -seguridad_id: entero -fh_alta: fecha -fh_nacimiento: fecha -correo: texto -sueldo: decimal -rfc: texto = XAXX010101
+getters/setter() +guardar(): booleano +eliminar(): booleano

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).

Seguridad
-ID: entero +nombre: texto
+getters/setters()

Clase Ordenes

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

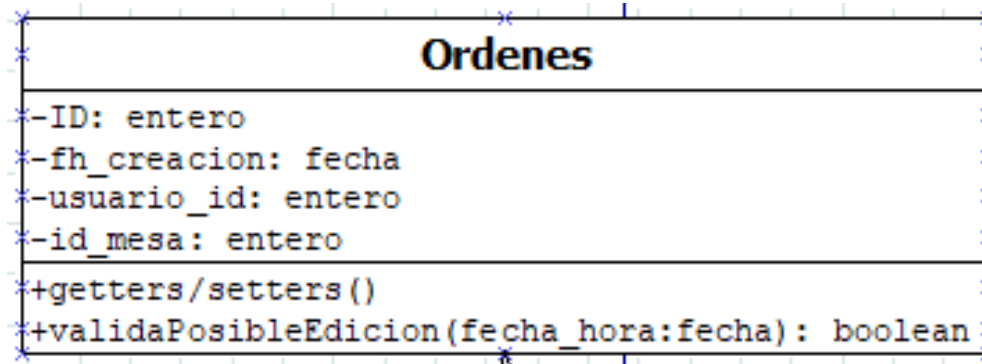
Cuenta con los siguientes atributos:

- 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.



Clase Orden_Detalles

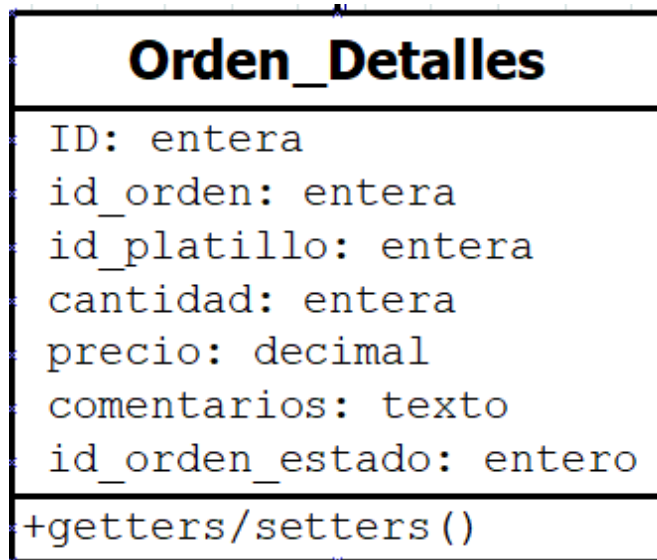
Clase representativa del contenido de las ordenes, es el modelo de como serian el contenido de las órdenes.

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).



Clase Orden_Estados

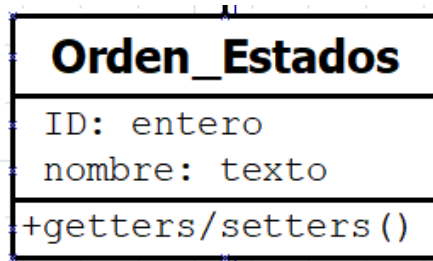
Clase simple que almacenará los diferentes estados de la orden dentro del sistema.

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 contara con sus métodos de acceso de datos getters y setters.

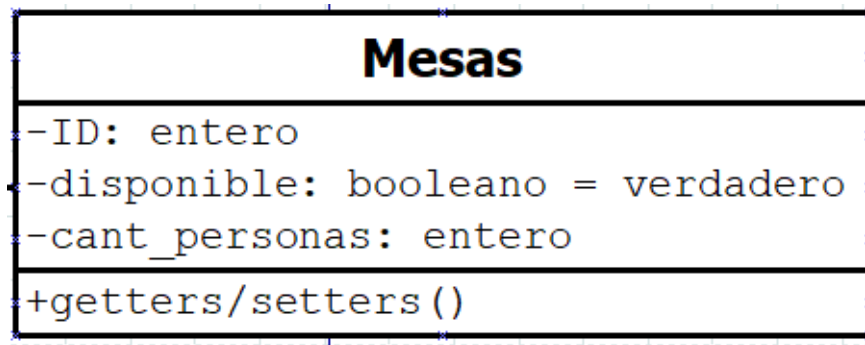


Clase que modelará las mesas dentro del restaurante.

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).



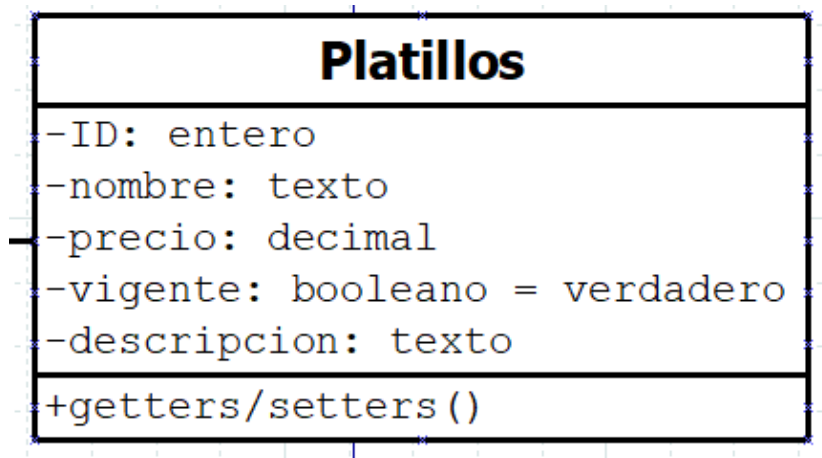
Platillos

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

Cuenta con los siguientes atributos:

- ID: Identificador único del platillo
- Nombre: nombre del platillo ofertado en el menú.
- Precio: Precio del platillo en el restaurante.
- Vigente: Campo que servirá para saber que platillos 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 platillo y sus componentes para facilitar a los usuarios sobre el platillo.

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



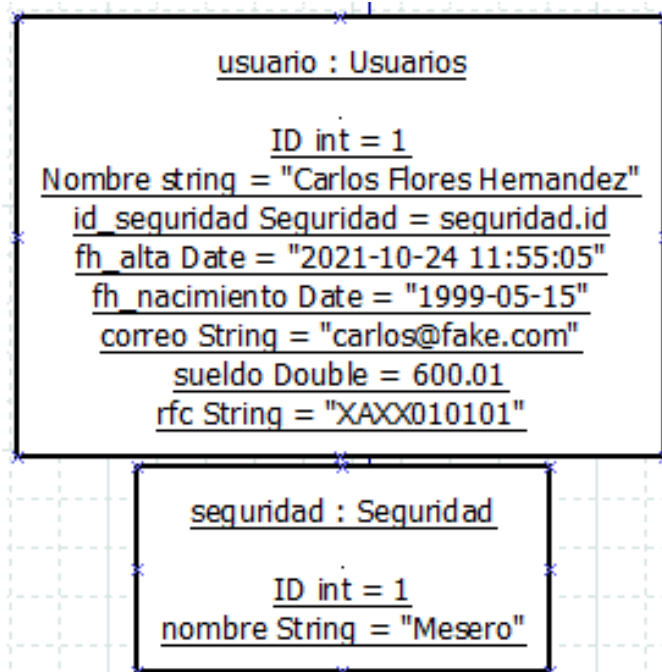
Mecanismos de Diseño Detallados

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.

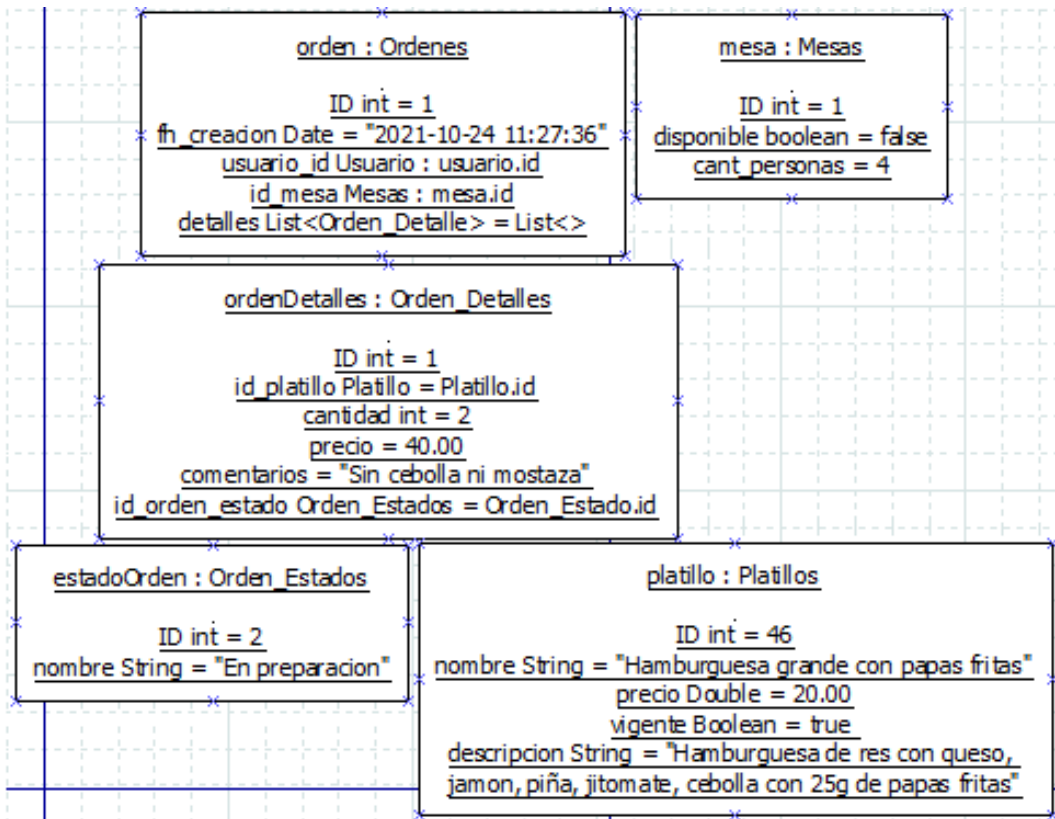
Objeto Usuario

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.



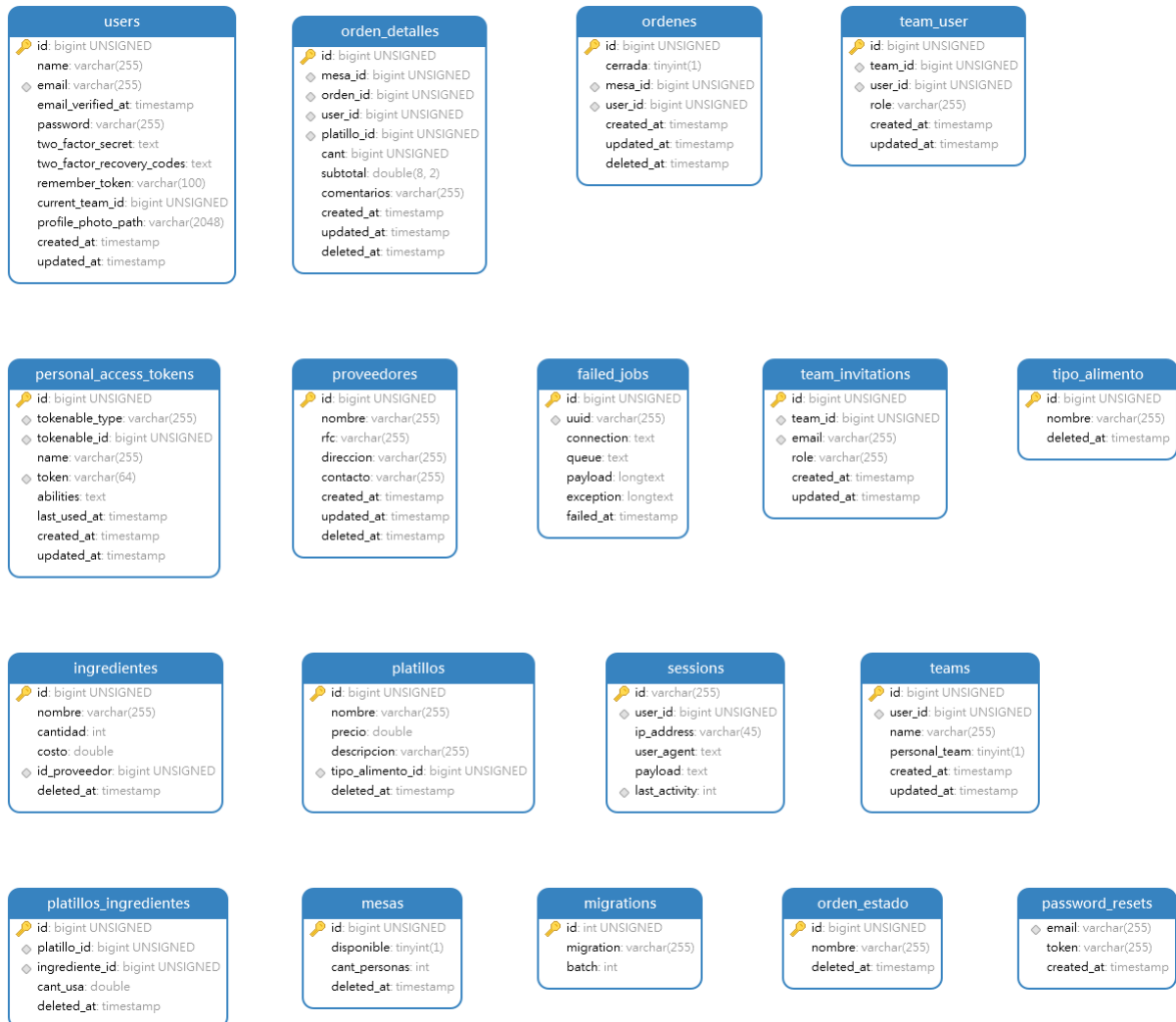
Objeto Orden

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.

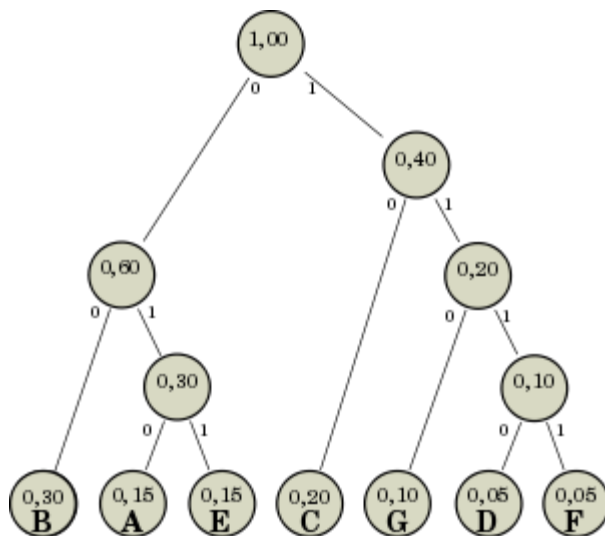


2.7.10. Modelo Lógico de Datos

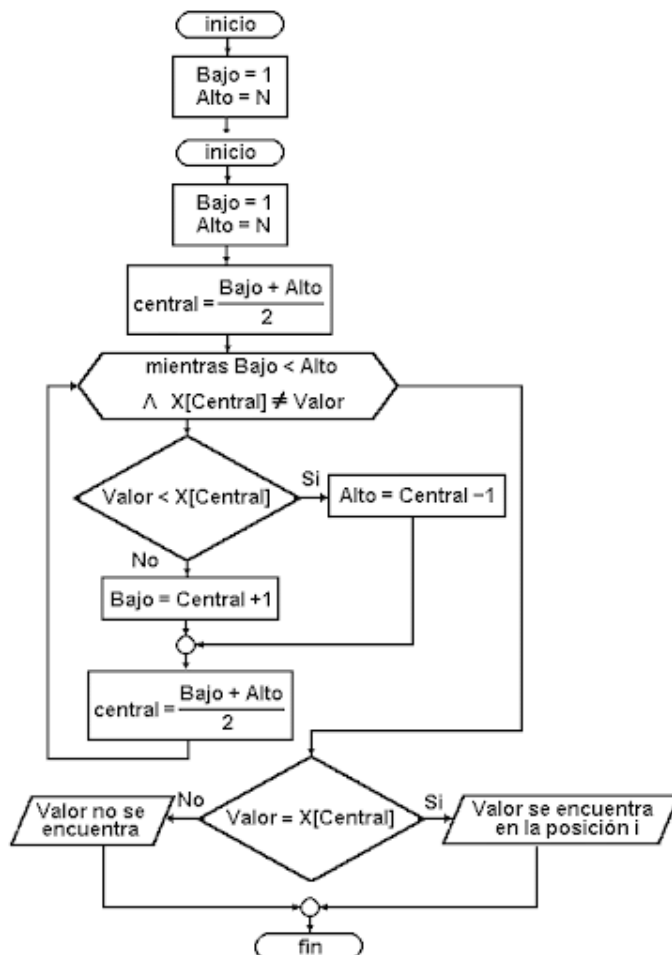
En el siguiente apartado se muestra e modelo de la base de datos en forma de tablas, para una fácil identificación de campos y relaciones entre las entidades. Esto con el fin de conocer el esqueleto de la información del sistema.



2.7.11. Modelo Físico de Datos



Se uso la Búsqueda Binaria se asume que los elementos están en cualquier orden. En el peor de los casos deben hacerse n operaciones de comparación. Una búsqueda más eficiente puede hacerse sobre un arreglo ordenado. compara si el valor buscado está en la mitad superior o inferior. En la que esté, subdivido nuevamente, y así sucesivamente hasta encontrar el valor.



2.7.13. Diseño de Pantallas y Reportes

El diseño de pantallas es un menú principal en el cual tenemos dos botones para acceder a dos submenús “Administración” y “Puerto de venta”.

En cada uno de estos submenús tenemos diferentes botones que son requeridos para estas dos partes de los restaurantes, en cuales algunas son necesarias para hacer bases de datos ya sean de productos, personal o pedidos y en la mayoría de estas funciones tiene para crear reportes.

El diseño se basó para que sea fácil para el usuario y no tenga complicaciones, unas bases seguras y rápidas para administrar el negocio de manera eficaz.

2.7.14. Descripción de Campos Requeridos por Pantalla

En la primera imagen vemos el menú principal y ya nos colocamos en el puerto de venta donde tenemos todos sus botones necesarios para realizar actividades del frente del restaurante y en la segunda vemos lo que es el submenú de Administración donde de igual manera tenemos los botones necesarios con sus funciones para tener una buena administración del restaurante.



2.7.15. Vista de Implementación

Compuesta fundamentalmente por un diagrama que muestra las principales piezas desde el punto de vista físico (subsistemas de implementación) que conforman el sistema. Dicho diagrama representa la estructuración física del código, los distintos directorios que organizan el código fuente, librerías, ficheros ejecutables (si existen), entre otros elementos. Se deben describir los componentes mostrados en el diagrama, incluyendo su propósito y contenido. Además, han de considerarse como parte de la vista las dependencias con aplicaciones o componentes externos.

2.7.16. Vista de Despliegue

Abarca la representación gráfica del *Diagrama de Despliegue*. Se debe explicar la estructura de los nodos que se presentan e indicar la correspondencia de los mismos con la estructuración en capas implementada.

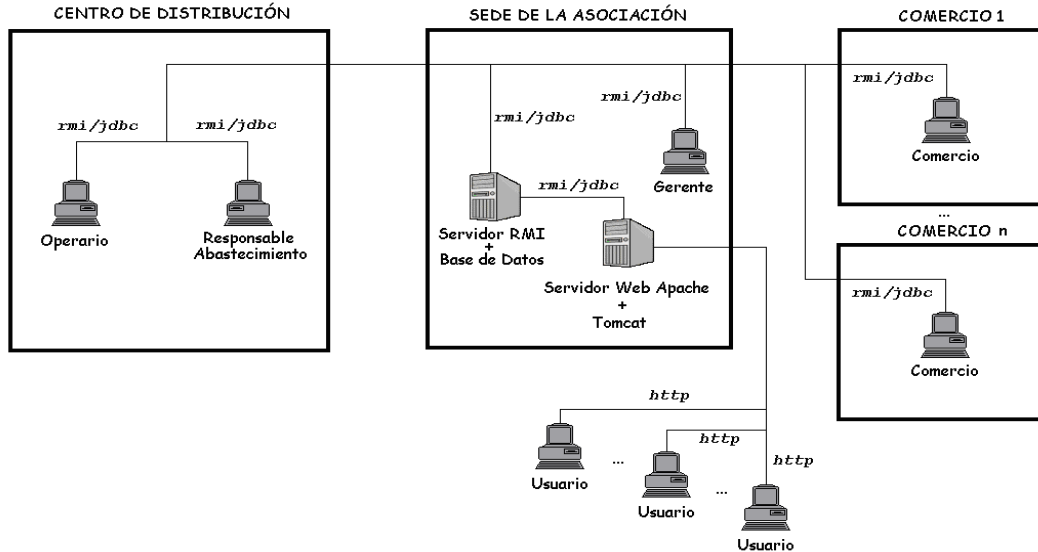
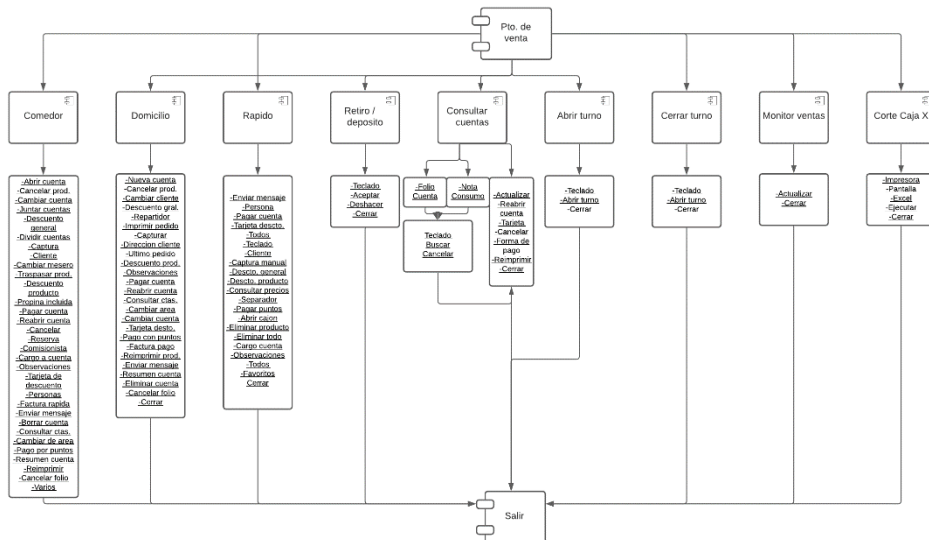
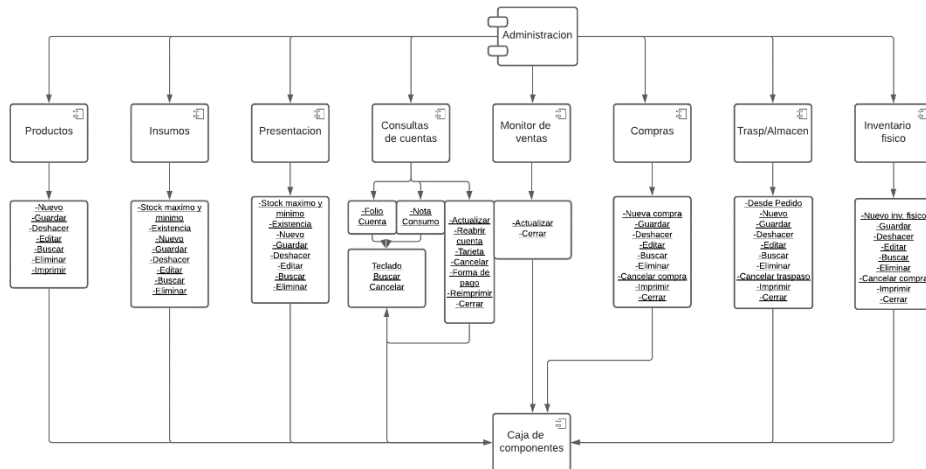


Figura1. Ejemplo de Diagrama de Despliegue

2.7.17. Diagrama de Navegación del Sistema

Muestra gráficamente la interconexión entre cada una de las pantallas del sistema, constituyendo de esta forma los diferentes caminos para llegar a determinada parte de la aplicación. Se deben mostrar los menús, submenús y pantallas a las que nos llevan cada uno de ellos.





2.7.18. Controles de Auditoría Implementados en el Sistema

La verificación de controles en el procesamiento de la información, desarrollo de sistemas e instalación con el objetivo de evaluar su efectividad y presentar recomendaciones a la Gerencia.

La actividad dirigida a verificar y juzgar información.

El examen y evaluación de los procesos del Área de Procesamiento automático de Datos (PAD) y de la utilización de los recursos que en ellos intervienen, para llegar a establecer el grado de eficiencia, efectividad y economía de los sistemas computarizados en una empresa y presentar conclusiones y recomendaciones encaminadas a corregir las deficiencias existentes y mejorarlas.

El proceso de recolección y evaluación de evidencia para determinar si un sistema automatizado.

2.7.19. Glosario de Términos

- 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.

3. Estándares de Elaboración del Manual

Toda la documentación que se relacione con un sistema, ya sea impresa o digital, sencilla o compleja, debe reunir los siguientes requisitos básicos:

Debe ser rotulada con claridad y bien organizada en carpetas e índice, con secciones claramente indicadas.

Los diagramas deberán ser claros, no aglomerados y la escritura manuscrita ha de ser legible

La documentación deberá ser completa.

Se incluirá una leyenda o explicación de los términos utilizados.

La documentación siempre se conserva actualizada.

El estilo de redacción de los manuales de documentación debe:

Ser concreto.

Definir los términos utilizados.

Utilizar títulos, subtítulos y párrafos cortos.

Emplear formas activas en lugar de pasivas.

Aplicar correctamente las referencias bibliográficas.

No usar frases largas que presenten hechos distintos.