



Software Requirements Specification [SRS]

(IEEE Std. 830-1998)

INTEGRANTES DEL GRUPO:

ALBERTO ALEJANDRO RIVAS FERNÁNDEZ
BRYAN EDUARDO CÓRDOVA ASCURRA
CARLOS GÓMEZ LÓPEZ
CARLOS FORRIOL MOLINA
GABRIEL TORRES MARTÍN
JAVIER DE HOYOS PINO
JOSÉ LUIS BARTHA DE LAS PEÑAS
JUAN ROMO IRIBARREN
LUIS ENRIQUE BARRERO PEÑA
MARIO GONZÁLEZ DE SANTOS
PABLO GAMO GONZÁLEZ
RUBÉN MARTÍN CASTRO

1. Introducción	3
1.1 Propósito	3
1.2 Alcance	3
1.3 Definiciones, acrónimos y abreviaturas	4
1.4 Referencias.....	5
1.5 Resumen.....	6
2. Descripción general	6
2.1 Perspectiva del producto.....	6
2.2 Funciones del producto	6
2.2.1 Producto	6
2.2.2 Sección	7
2.2.3 Trabajador.....	7
2.2.4 Cliente (Supermercado)	7
2.2.5 Compra.....	8
2.2.6 Proveedor.....	8
2.2.7 Marca.....	8
2.2.8 Cliente (Restaurante)	8
2.2.9 Ingrediente	9
2.2.10 Pedido.....	9
2.2.11 Personal	9
2.2.12 Plato	10
2.2.13 Turno	10
2.3 Características del usuario.....	10
2.4 Restricciones	10
2.4.1 Restricción principal.....	10
2.4.2 Restricciones de tiempo	11
2.4.3 Requisitos de lenguajes de alto nivel	11
2.4.4 Requisitos de Bases de datos	11
2.4.5 Restricciones del proyecto	11
2.4.6 Funciones de auditoría.....	12
2.4.7 Requisitos de Fiabilidad	12
2.5 Supuestos y dependencias	12
3. Requisitos Específicos	13
3.1 Interfaces Externas (Inicio).....	13
3.1 Interfaces Externas (Supermercado)	14
3.1 Interfaces Externas (Restaurante)	16
3.2 Funciones.....	17
3.2.1 Módulo de Productos.....	17
3.2.2 Módulo de Secciones.....	21
3.2.3 Módulo de Trabajadores.....	24

3.2.4 Módulo de Clientes (Supermercado).....	28
3.2.5 Módulo de Compras	31
3.2.6 Módulo de Proveedores	35
3.2.7 Módulo de Marcas	38
3.2.8 Módulo de Clientes (Restaurante).....	41
3.2.9 Módulo de Ingredientes	44
3.2.10 Módulo de Pedidos.....	47
3.2.11 Módulo de Personal	51
3.2.12 Módulo de Plato	55
3.2.13 Módulo de Turno.....	59
3.3 Requisitos de Rendimiento	61
3.4 Logical Database Requirements (Modelo del Dominio).....	62
3.4.1 Supermercado	62
3.4.2 Restaurante	62
3.5 Restricciones de Diseño	62
3.6 Atributos del Sistema Software	62

1. Introducción

El proyecto por desarrollar se basa en la unión de dos proyectos de manera que sean totalmente cohesivos entre sí y además conseguir que la gestión de datos sea transaccional.

Los dos proyectos que vamos a utilizar son los siguientes, un software de gestión de auto compra de supermercados y un software de gestión de un restaurante.

1.1 Propósito

Este documento pretende proporcionar información sobre los límites y requerimientos del producto final, junto a la información necesaria para el entendimiento de este.

Está dirigido a los empleados de ambos establecimientos para conseguir una buena gestión de ambas aplicaciones.

El propósito de este documento es hacer la especificación de requisitos (SRS) siguiendo el estándar IEEE Std.830-1998 para que nuestro sistema permita el correcto funcionamiento de un gestor de auto compra de un supermercado junto a un gestor de un restaurante.

1.2 Alcance

Este manual se centrará en detallar el proyecto de “Supermercado” y sobre el restaurante asociado “Restaurante”.

Es importante comentar que estos dos softwares son ajenos entre ellos y el objetivo de este proyecto es conseguir una cohesión entre ellos y se entiendan como una sola unidad.

El software del supermercado proporciona un control sobre todas las compras que se realicen, ya sean efectuadas por los trabajadores o por los clientes mediante auto compra. También controlara los productos, las marcas de estos y los proveedores que las distribuyen.

El software del restaurante proporciona el control de un restaurante basado en pedidos, éstos contienen unos platos y cada uno de estos es atendido por el personal del restaurante.

El proyecto final “Supermercados” ha sido creado con el propósito de facilitar la gestión de un supermercado con un restaurante asociado, pero este producto puede ser de gran utilidad dentro del mercado actual debido a las numerosas fusiones entre empresas.

Este producto puede servir como posibles prototipos para uniones que sean parecidas a este proyecto.

Por parte del sistema del supermercado se quiere proporcionar al cliente una forma rápida de realizar su compra sin la necesidad de que le atienda un cajero, mediante una BD que contenga la información de los clientes, proveedores, marcas, productos, empleados. El programa podrá ser utilizado por el administrador que, entre otras cosas,

podrá gestionar los clientes, las secciones, los productos y sus marcas y sus respectivos proveedores.

Por parte del sistema del restaurante se quiere proporcionar al cliente una forma para gestionar sus pedidos al restaurante de manera automática, en la que los datos se guardarán en una BD que contendrá la información de los clientes, sus pedidos, los platos que tendrá éste, con sus ingredientes específicos y además al personal que se le asigne dicho pedido, los turnos que tienen estos empleados.

1.3 Definiciones, acrónimos y abreviaturas

SRS: Software Requirements Specification, es una descripción completa del comportamiento del sistema que se va a desarrollar.

BD: Base de datos, es un conjunto de datos pertenecientes a un mismo contexto que son almacenados conjuntamente para su posterior uso.

IEEE (Institute of Electrical and Electronics Engineers): El estándar que se utiliza en este documento es el Std. 830-1998 del IEEE.

S.O: Sistema operativo.

Cliente: Organización que ha contratado nuestros servicios para producir este software.

Usuario: Toda persona que va a interactuar con el software.

UCM: Universidad Complutense de Madrid.

IBM (International Business Machines) RSAD (Rational Software Architect Designer): Herramienta Case utilizada para desarrollar parte de los requisitos de la SRS.

Diagrama de actividades: Es un diagrama que muestra un flujo de actividades, está formado por nodos y arcos. Los utilizamos para describir las acciones de las funciones.

Herramienta CASE: Son herramientas que facilitan el almacenamiento de requisitos, la gestión del cambio y la gestión de la traza.

SCV (Sistema de Control de Versiones): Es un programa que ayuda a crear y mantener las diversas versiones de un conjunto de ficheros.

RDBMS (Sistema de gestión de bases de datos relacionales): es una base de datos que almacena datos de forma estructurada.

MySQL: Sistema de gestión de bases de datos relacional, es una de las más populares, sobre todo para entornos de desarrollo web.

Backup: Consiste en una copia de seguridad de los datos originales que se realiza con el fin de disponer de un medio para recuperarlos en caso de su pérdida.

Revisiones técnicas formales (RTFs): El objetivo básico de las RTFs es detectar errores antes de que se conviertan en defectos.

Elementos de configuración software (ECS): Son los resultados de cada tarea de trabajo.

JPA (Java Persistence API): Es la propuesta estándar que ofrece Java para implementar un Framework Object Relational Mapping (ORM), que permite interactuar con la base de datos por medio de objetos, de esta forma, JPA es el encargado de convertir los objetos Java en instrucciones para el Manejador de Base de Datos (MDB).

Framework: Es una estructura base utilizada como punto de partida para elaborar un proyecto con objetivos específicos.

ORM: Modelo de programación que permite mapear las estructuras de una base de datos relacional (SQL Server, Oracle, MySQL, etc.), sobre una estructura lógica de entidades con el objeto de simplificar y acelerar el desarrollo de nuestras aplicaciones.

Marca: Signo, que permite diferenciar y hacer únicos los productos o servicios de una empresa.

Compra: Acción de comprar unos productos por un precio.

Trabajador: Persona que realiza un trabajo a cambio de un salario.

Producto: Cosa producida natural o artificialmente, o resultado de un trabajo u operación.

Proveedor: Que se dedica a proveer o abastecer de productos necesarios a una persona o empresa.

Perecedero: Que tiene duración limitada, está destinado a perecer, perder su utilidad o validez, o estropearse en un determinado plazo de tiempo.

Sección: Parte que junto con otras constituye una empresa, un establecimiento comercial, un organismo, una agrupación, etc.

Pedido: Encargo realizado por un cliente que contiene de uno a varios platos.

Plato: Preparación culinaria específica servida para ingerirla.

Ingrediente: Elemento que forma un compuesto, en especial de un compuesto destinado a la ingestión.

Turno: Intervalo de tiempo que trabaja un trabajador dentro de un día.

1.4 Referencias.

Apuntes del profesor ANTONIO NAVARRO MARTIN del Campus Virtual de la UCM:
https://cvmdp.ucm.es/moodle/pluginfile.php/687418/mod_resource/content/1/6.%20Ingenier%C3%ADa%20de%20requisitos.pdf

https://cvmdp.ucm.es/moodle/pluginfile.php/1941171/mod_resource/content/1/jpa.pdf

Wikipedia:

-https://es.wikipedia.org/wiki/Base_de_datos

-https://es.wikipedia.org/wiki/Especificaci%C3%B3n_de_requisitos_de_software

1.5 Resumen.

Este documento sigue el estándar 830-1998 y está dividido en tres apartados:

- El primer apartado, introducción de la SRS consiste en una descripción breve del producto que vamos a desarrollar, nuestros propósitos y alcance y una pequeña guía para entender los próximos apartados.
- El segundo apartado es la descripción general. En este apartado se describen los factores generales que afectan al producto y sus requisitos. Es una descripción breve y no define requisitos específicos ya que se realiza de manera más comprensible.
- Por último, se representan los requisitos específicos los cuales contienen los requisitos del sistema de forma extensa, pero sin dejar de lado la legibilidad. Este punto referencia a los anteriores.

2.Descripción general

2.1 Perspectiva del producto

La plataforma será un software que se desarrolle para poder ser utilizada por el supermercado y el restaurante para fomentar una mayor actividad comercial de auto compra o autoservicio además de la realización de pedidos para recoger del restaurante.

Se realizará habitualmente y de forma automática, copias de seguridad o backups en la propia base de datos para evitar riesgos en caso de pérdida de información o caída del sistema y así poder restaurarla posteriormente.

2.2 Funciones del producto

Las funciones del producto son:

2.2.1 Producto

Añadir un nuevo producto: Crea un nuevo producto, con los datos introducidos, en la base de datos.

Vincular producto a proveedor: Vincula un producto a un proveedor específico.

Buscar datos de todos los productos: Muestra los datos de todos los productos.

Buscar datos de un solo producto: Busca y muestra los datos del producto solicitado.

Modificar un producto: Actualizar o modificar el stock/información de un producto en la base de datos.

Eliminar un producto: Elimina de forma lógica el producto seleccionado en la base de datos.

Desvincular producto de proveedor: Desvincula un producto de un proveedor específico.

2.2.2 Sección

Añadir nueva sección: Crea una sección, con los datos introducidos, en la base de datos.

Buscar información de las secciones: Muestra los datos de todas las secciones.

Buscar información de una sección: Busca y muestra los datos de la sección solicitada.

Editar datos de sección: Modificar los datos de la sección en la base de datos.

Eliminar una sección: Elimina de forma lógica la sección introducida de la base de datos.

2.2.3 Trabajador

Dar de alta a un nuevo: Añade o activa un empleado en la base de datos.

Buscar datos de un trabajador por ID: Busca un empleado y muestra sus datos a través de su ID.

Buscar datos de un trabajador por DNI: Busca un empleado y muestra sus datos a través de su DNI.

Buscar datos de todos los trabajadores: Muestra los datos de todos los empleados

Actualizar/modificar datos de un trabajador: Modifica los datos de un empleado en la base de datos.

Dar de baja a un trabajador: Introduce un empleado y lo elimina de forma lógica de la base de datos.

Identificar al trabajador: Identifica al trabajador en el sistema para que pueda realizar las respectivas acciones.

2.2.4 Cliente (Supermercado)

Dar de alta a un nuevo cliente: Añade un nuevo cliente a la base de datos.

Buscar información general de los clientes: Muestra los datos de todos los clientes.

Buscar datos específicos de un cliente: Busca un cliente y muestra sus datos.

Editar información del cliente: Modifica los datos del cliente solicitado en la base de datos.

Dar de baja al cliente registrado: Introduce un cliente y lo elimina de forma lógica de la base de datos.

2.2.5 Compra

Abrir compra: Un cliente inicia una compra.

Añadir producto al carrito: Lee el ID y si existe se añade al carrito.

Eliminar productos del carrito: Se elimina el producto seleccionado del carrito.

Cerrar compra: El cliente debe introducir su ID y proceder al pago.

Buscar compra: Dado un ID de compra la muestra con toda su información.

Buscar todas las compras: Muestra todas las compras realizadas.

Buscar las compras de un cliente: Muestra todas las compras realizadas por un cliente.

Realizar devolución: Permite devolver uno o más productos de una compra.

2.2.6 Proveedor

Añadir nuevo proveedor: Registra y añade un nuevo proveedor, con los datos introducidos en la base de datos.

Buscar proveedores: Busca y muestra los datos de todos los proveedores.

Buscar proveedor: Busca y muestra los datos del proveedor deseado.

Editar información de un proveedor: Edita la información del proveedor que se desee.

Dar de baja un proveedor: Elimina de forma lógica a un proveedor de la base de datos.

2.2.7 Marca

Añadir nuevas marcas: Registra una nueva marca, con los datos introducidos, en la base de datos.

Buscar información de una marca: Busca y muestra los datos de la marca solicitada.

Editar datos específicos de una marca: Permite editar los datos de una marca-

Buscar información de las marcas: Muestra los datos de todas las marcas.

Eliminar una marca: Elimina de forma lógica la marca solicitada de la base de datos.

2.2.8 Cliente (Restaurante)

Alta cliente: Añade un nuevo cliente en la base de datos.

Buscar cliente: Busca y muestra la información del cliente solicitado.

Listado clientes: Muestra un listado de todos los clientes.

Modificar cliente: Modifica los atributos del cliente solicitado en la base de datos.

Baja cliente: Da de baja el cliente solicitado de forma lógica de la base de datos.

2.2.9 Ingrediente

Añadir ingrediente nuevo: Añade un nuevo ingrediente en la base de datos.

Ver lista de ingredientes disponibles: Muestra la lista de ingredientes disponibles en la base de datos.

Buscar ingrediente: Busca un ingrediente solicitado y muestra su información.

Modificar ingrediente: Modifica los atributos del ingrediente solicitado.

Eliminar ingrediente: Elimina el ingrediente especificado de forma lógica de la base de datos.

2.2.10 Pedido

Abrir pedido nuevo: Añade un nuevo pedido a la base de datos.

Buscar pedidos de un cliente: Muestra la lista de pedidos realizados por un cliente.

Buscar todos los pedidos: Muestra todos los pedidos de la base de datos.

Modificar pedido: Modifica los atributos de un pedido ya realizado.

Eliminar pedido: Elimina el pedido indicado de forma lógica de la base de datos.

Cerrar pedido: Se cerrará el pedido de un cliente.

Añadir plato a pedido: Se añade un plato al pedido indicado.

Eliminar plato de pedido: Se elimina un plato del pedido indicado.

2.2.11 Personal

Añadir empleado: Añade a un nuevo empleado al personal, con los datos introducidos en la base de datos.

Buscar empleados: Lista los datos de todos los empleados del restaurante.

Buscar un empleado por ID: Busca y muestra los datos del empleado deseado, filtrado por su ID.

Buscar un empleado por DNI: Busca y muestra los datos del empleado deseado, filtrado por su DNI.

Editar información de un empleado: Edita la información del empleado que se solicite.

Dar de baja a un empleado: Elimina de forma lógica a un empleado de la base de datos.

Identificar empleado: Iniciar sesión en el sistema como empleado del restaurante, para realizar las respectivas acciones.

2.2.12 Plato

Crear nuevo plato: Crea un nuevo plato y lo añade a la base de datos.

Añadir ingrediente a un plato: Añade un nuevo ingrediente a un plato.

Eliminar ingrediente a un plato: Elimina un ingrediente del plato.

Buscar plato: Busca y muestra el plato en la base de datos a través de su nombre.

Buscar todos los platos: Muestra todos los platos que hay dentro de la base de datos.

Modificar plato: Modifica los atributos del plato indicado.

Eliminar plato: Elimina el plato solicitado de forma lógica de la base de datos.

2.2.13 Turno

Crear nuevo turno: Crea un nuevo turno.

Buscar turno: Busca y muestra el turno indicado.

Ver listas turnos: Muestra la lista de turnos.

Modificar turno: Modifica los atributos de un turno.

Eliminar turno: Elimina el turno seleccionado.

2.3 Características del usuario

El software será usado por los empleados del supermercado y por los clientes. No se necesita ninguna formación ni experiencia para utilizar la aplicación ya que tendrá una interfaz de usuario simple y fácil de usar.

Los empleados tendrán acceso a más funcionalidades, ya que pueden iniciar sesión y gestionar los datos del supermercado.

El idioma de la aplicación será el español exclusivamente, por lo que los usuarios deberán tener un nivel básico de español.

2.4 Restricciones

2.4.1 Restricción principal

Es obligatorio utilizar técnicas de Ingeniería del software a nivel de diseño y gestión de este software para lograr un producto de calidad y cumplir con los requisitos. También debemos utilizar una arquitectura multicapa.

2.4.2 Restricciones de tiempo

El proyecto constara de tres entregas:

- La primera entrega del proyecto (SRS) será el día 30.09.2021. Consistirá en la especificación de requisitos software de la aplicación según formato IEEE 830-1998.
- La segunda entrega (parte DAO) del proyecto será el día 11.11.2021. Consistirá en la primera versión de la aplicación, incluyendo su modelo UML 2.x y las pruebas JUnit.
- La tercera entrega del proyecto (parte JPA) será el día 16.12.2021. Consistirá en la segunda versión de la aplicación, incluyendo su modelo UML 2.x y las pruebas JUnit.

2.4.3 Requisitos de lenguajes de alto nivel

El diseño se realizará con el lenguaje UML 2.x el cual desarrollaremos desde la herramienta CASE IBS RSAD.

En cuanto al lenguaje que se usará para la codificación será Java y la aplicación será de escritorio.

Utilizaremos el SCV de la Facultad de Informática (SVN) para gestionar tanto el código como la documentación.

Para gestionar la base de datos utilizaremos el lenguaje de MySQL.

2.4.4 Requisitos de Bases de datos

Utilizaremos una base de datos relacional, en este caso MySQL 5x.

2.4.5 Restricciones del proyecto

- El proyecto va a realizarse en equipo, el cual constará de 12 personas exactamente.
- Cada entrega es prerequisite de la siguiente.
- Todas las entregas deben hacerse como un tag en el SVN.
- En las entregas segunda y tercera deben incluirse:
 - Los archivos del diseño en formato IBM RSAD 9.6
 - Los modelos de despliegue y componentes UML
 - El código y las pruebas JUnit
 - Una memoria
 - Un informe del trabajo desarrollado por cada alumno con su factor de calificación
- Los alumnos son los únicos responsables del contenido almacenado en el SCV. La presencia de contenido inadecuado en el SCV conllevará su borrado y la invalidación del proyecto.

En la primera versión de la aplicación deben aplicarse los siguientes patrones:

- Server to worker
- Transfer Object
- Transfer Object Assembler
- Application Service
- Data Access Object
- Alguna query

En la segunda versión, se debe añadir los siguientes patrones:

- Bussiness Object
- Domain Store (implementado con JPA 2.x.)

2.4.6 Funciones de auditoría

Vamos a realizar revisiones técnicas formales (RTF) para garantizar la calidad del trabajo y asegurarnos de que no haya errores.

2.4.7 Requisitos de Fiabilidad

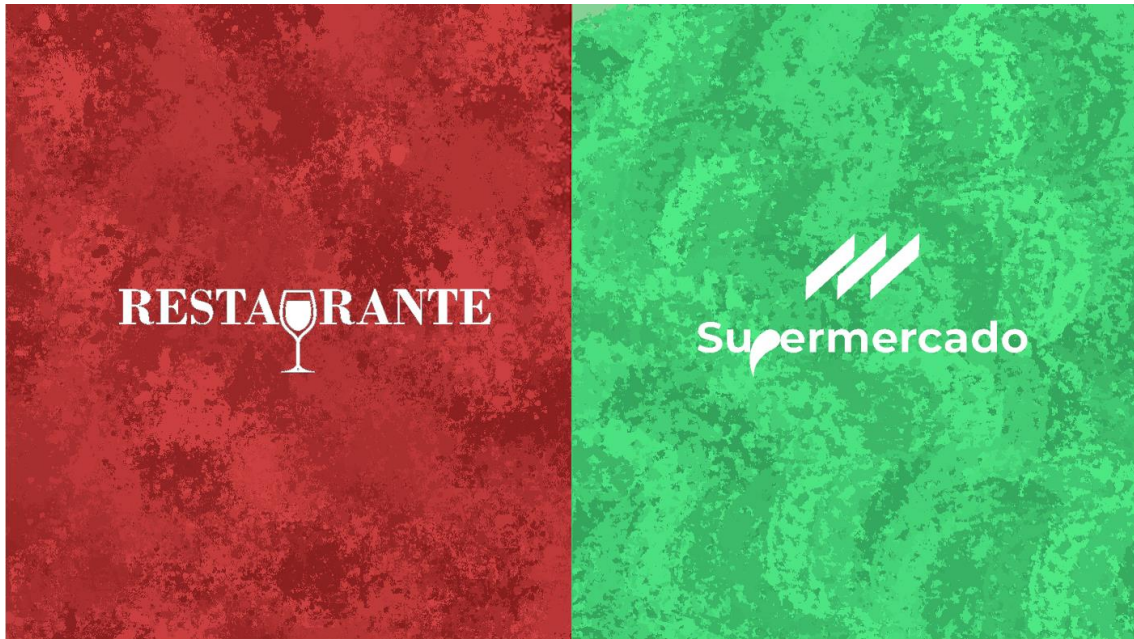
Debemos implementar un sistema de autenticación de los usuarios para garantizar que solo los empleados puedan tener acceso y modificar los datos. Almacenaremos la información en una base de datos MySQL para que sea persistente en el tiempo y puedan ser accedidos en cualquier momento.

2.5 Supuestos y dependencias

- El software está creado para utilizarlo únicamente en ordenadores cuyo sistema operativo sea Windows 7, Windows 8 o Windows 10, así como las principales distribuciones de Linux como Ubuntu, Debian, CentOS o Canaima. Este software no está disponible para macOS.
- Los módulos utilizados por el software utilizarán una de las marcas más populares de software RDBMS (MySQL) ya que implementa un modelo cliente-servidor, útil para nuestro software.

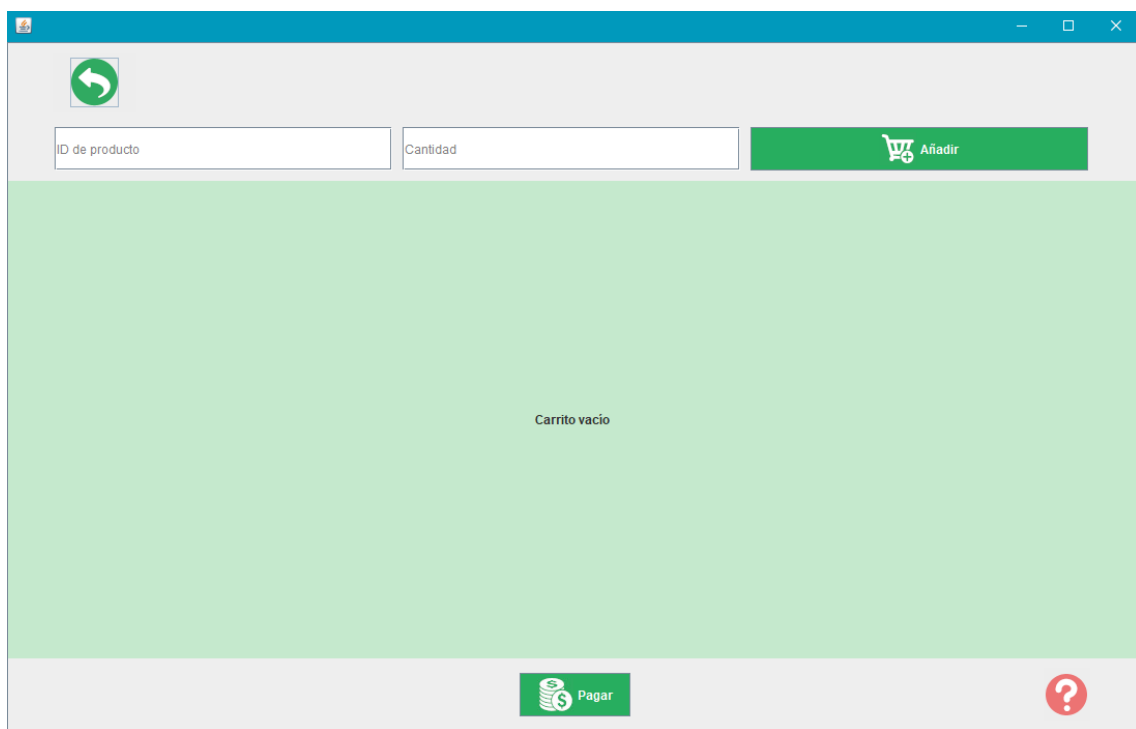
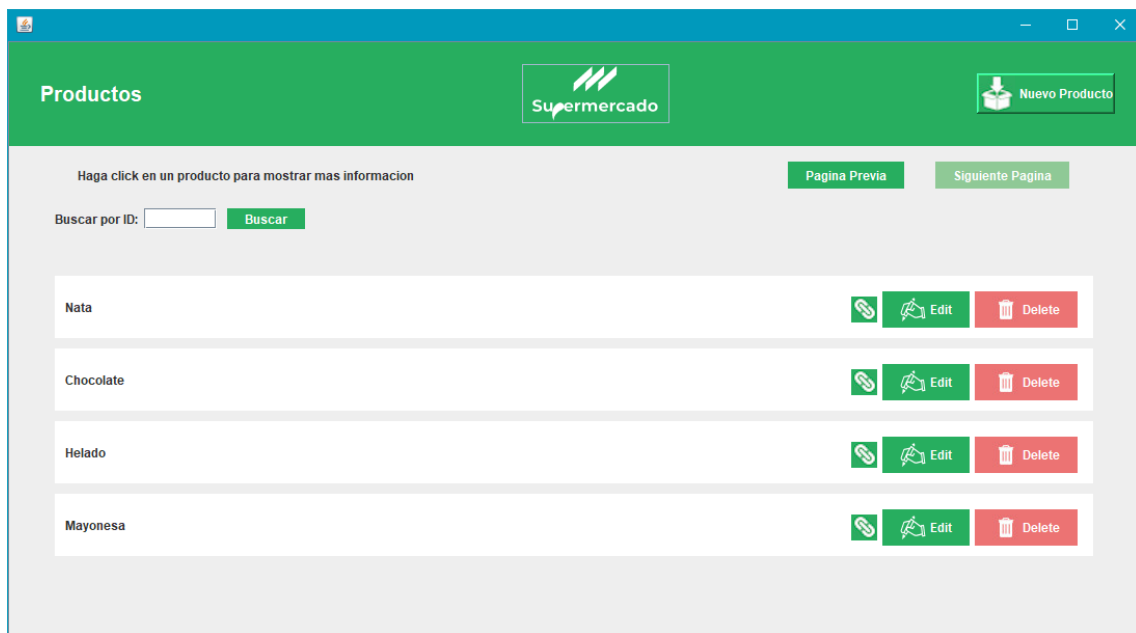
3. Requisitos Específicos

3.1 Interfaces Externas (Inicio)

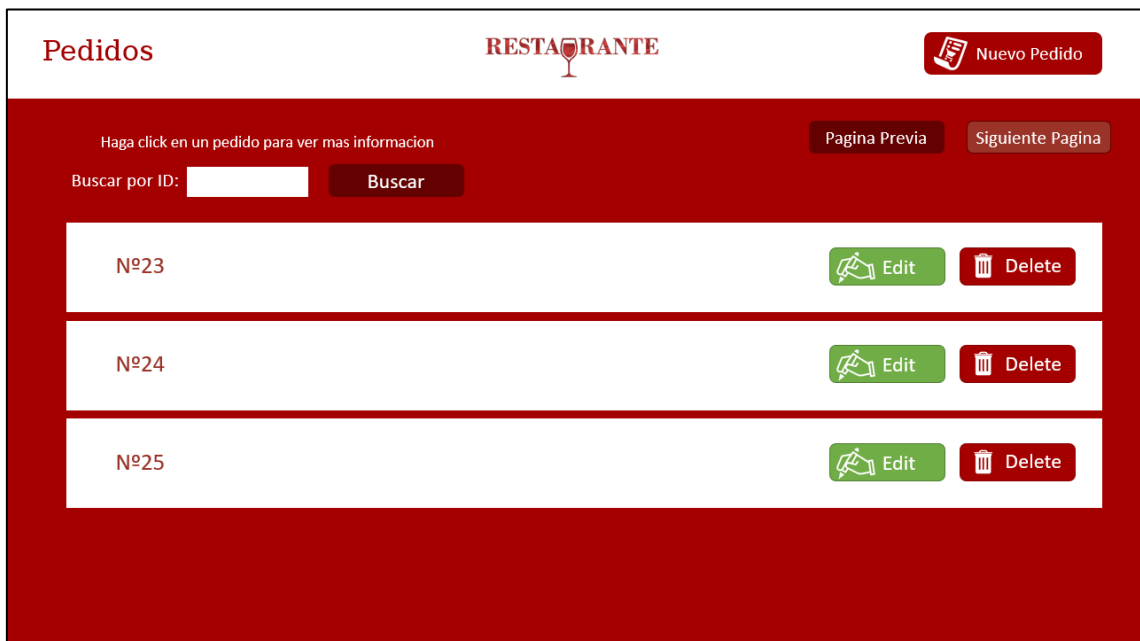


3.1 Interfaces Externas (Supermercado)





3.1 Interfaces Externas (Restaurante)



3.2 Funciones

3.2.1 Módulo de Productos

Función 3.2.1.1 Añadir un nuevo producto	
Prioridad	Alta.
Estabilidad	Alta.
Descripción	Añade un nuevo producto a la BD de productos.
Entrada	Precio, Stock, Nombre, ID_Marca, ID_Seccion, Tipo.
Salida	Muestra ID_Producto y su nombre.
Origen	Interfaz Usuario – Añadir producto.
Destino	Sistema.
Necesita	Base de datos de productos.
Acción	Comprobar que los datos introducidos son sintácticamente correctos y que su marca exista y este activa, que no haya ningún producto con el mismo nombre y marca. En caso afirmativo añadir el producto y devolver ID_Producto.
Precondición	Los datos introducidos sean válidos, la marca y la sección existan y estén activos, y si existe el producto, que no esté activo.
Postcondición	El producto será creado en la BD y se quedará activo.
Efectos laterales	Si falla el procedimiento avisar mediante un mensaje para repetir el proceso.

Función 3.2.1.2 Vincular Producto a Proveedor	
Prioridad	Media.
Estabilidad	Media.
Descripción	Vincula un producto a un proveedor específico.
Entrada	ID_Producto e Id_Proveedor.
Salida	Muestra ID_Producto, su nombre y su proveedor asignado con su ID.
Origen	Interfaz Usuario – Vincular producto.
Destino	Sistema.
Necesita	Base de datos de productos.
Acción	Comprobar que los IDs sean correctos, que existan y estén activos, que el producto no esté vinculado a otro proveedor y si se cumple, se asignara el producto a un proveedor en concreto.
Precondición	El producto y el proveedor existan y estén activos.
Postcondición	Vinculará el producto al proveedor.
Efectos laterales	Ninguno.

Función 3.2.1.3	Buscar datos de todos los productos
Prioridad	Media.
Estabilidad	Alta.
Descripción	Busca y muestra los datos de todos los productos.
Entrada	Ninguna.
Salida	Todos los atributos del producto.
Origen	Interfaz Usuario – Buscar todos productos
Destino	Sistema.
Necesita	Base de datos de productos.
Acción	Accede a la BD, obtienes todos los productos y los muestra en forma de lista.
Precondición	Ninguna.
Postcondición	Los productos se mostrarán en forma de lista.
Efectos laterales	Ninguno.

Función 3.2.1.4	Buscar datos de un solo producto
Prioridad	Media.
Estabilidad	Alta.
Descripción	Busca y muestra los datos del producto indicado.
Entrada	ID_Producto.
Salida	Todos los atributos del producto.
Origen	Interfaz Usuario – Buscar producto.
Destino	Sistema.
Necesita	Base de datos de productos.
Acción	Dado un ID_Producto busca en la BD el producto y lo muestra con toda su información.
Precondición	El producto debe existir en la BD.
Postcondición	El producto será mostrado con todos sus atributos.
Efectos laterales	Ninguno.

Función 3.2.1.5		Modificar un producto
Prioridad		Alta.
Estabilidad		Alta.
Descripción		Se modificarán los productos.
Entrada		ID_Producto.
Salida		Mensaje de confirmación.
Origen		Interfaz Usuario – Modificar producto.
Destino		Sistema.
Necesita		Base de datos de productos.
Acción		Busca el producto en la base de datos, en caso de que exista y este activo, cambiará los datos.
Precondición		El producto exista en la BD y esté activo.
Postcondición		Se quedarán registrados los cambios del producto.
Efectos laterales		Ninguno.

Función 3.2.1.6		Eliminar Producto
Prioridad		Alta.
Estabilidad		Media.
Descripción		Se elimina un producto del sistema.
Entrada		ID_Producto.
Salida		Mensaje de que se ha eliminado.
Origen		Interfaz Usuario – Eliminar producto.
Destino		Sistema.
Necesita		Base de datos de productos.
Acción		Busca el producto en la base de datos, en caso de que exista, lo elimina de forma lógica.
Precondición		El producto debe existir en la BD y estar activo.
Postcondición		El producto quedará inactivo.
Efectos laterales		Ninguno.

Función 3.2.1.7 Desvincular Producto de Proveedor	
Prioridad	Media.
Estabilidad	Media.
Descripción	Desvincula un producto de un proveedor específico.
Entrada	ID_Producto e Id_Proveedor.
Salida	Muestra mensaje de confirmación.
Origen	Interfaz Usuario – Desvincular producto.
Destino	Sistema.
Necesita	Base de datos de productos y de proveedores.
Acción	Comprobar que los IDs sean correctos y desasignar el producto a un proveedor en concreto.
Precondición	El producto y el proveedor existan y estén activos.
Postcondición	Se desvinculará el producto del proveedor.
Efectos laterales	Ninguno.

3.2.2 Módulo de Secciones

Función 3.2.2.1		Añadir nueva sección
Prioridad		Alta.
Estabilidad		Alta.
Descripción		Crea una sección en la BD de secciones.
Entrada		Zona y pasillo.
Salida		Mensaje de confirmación.
Origen		Interfaz Usuario – Añade sección.
Destino		Sistema.
Necesita		Base de datos secciones.
Acción		Comprueba si están mal introducidos los datos. En caso de estar bien introducidos, comprueba que no haya una sección ya con ese nombre, si la hay mandara un mensaje de error, si es nueva, se mandará un mensaje de confirmación.
Precondición		Los datos introducidos sean válidos y si existe, que no esté activo.
Postcondición		La sección será creada en la BD y quedará activada.
Efectos laterales		Ninguno.

Función 3.2.2.2		Buscar información de las secciones
Prioridad		Media.
Estabilidad		Alta.
Descripción		Busca y muestra los datos de todas las secciones.
Entrada		Ninguna.
Salida		Todos los atributos de todas las secciones.
Origen		Interfaz Usuario – Buscar todas secciones.
Destino		Sistema.
Necesita		Base de datos de secciones.
Acción		Comprueba todas las secciones que existen para mostrar los atributos, en caso de que no exista ninguna, muestra mensaje de error.
Precondición		Ninguna.
Postcondición		Las secciones serán mostradas en forma de lista.
Efectos laterales		Ninguno.

Función 3.2.2.3		Buscar información de una sección
Prioridad		Media.
Estabilidad		Alta.
Descripción		Muestra los datos de la sección indicada.
Entrada		ID_Seccion.
Salida		Todos los atributos de la sección solicitada.
Origen		Interfaz Usuario – Buscar sección.
Destino		Sistema.
Necesita		Base de datos de secciones.
Acción		Comprueba si existe la sección que se quiere buscar. En caso de existir muestra todos sus atributos, si no, muestra mensaje de error.
Precondición		La sección exista en la BD y esté activa.
Postcondición		La sección será mostrada con todos sus atributos.
Efectos laterales		Ninguno.

Función 3.2.2.4		Editar datos de sección
Prioridad		Media.
Estabilidad		Media.
Descripción		Se modificará la sección.
Entrada		ID_Seccion.
Salida		Mensaje de confirmación.
Origen		Interfaz Usuario – Editar sección.
Destino		Sistema.
Necesita		Base de datos de secciones.
Acción		Busca la sección en la base de datos, en caso de que exista, comprobará su validez.
Precondición		La sección debe existir en la BD y estar activa.
Postcondición		Se quedarán registrados los cambios de la sección.
Efectos laterales		Ninguno.

Función 3.2.2.5 Eliminar una sección	
Prioridad	Alta.
Estabilidad	Media.
Descripción	Se elimina de forma lógica una sección del sistema.
Entrada	ID_Seccion.
Salida	Mensaje de que se ha eliminado.
Origen	Interfaz Usuario – Eliminar sección.
Destino	Sistema.
Necesita	Base de datos de secciones.
Acción	Busca la sección en la base de datos, en caso de que exista, la inactiva (baja lógica).
Precondición	La sección debe existir en la BD y estar activa.
Postcondición	La sección se quedará inactiva.
Efectos laterales	Eliminar una Sección implica que esta no debe tener ningún producto activo asociado.

3.2.3 Módulo de Trabajadores

Función 3.2.3.1 Dar de alta a un nuevo trabajador	
Prioridad	Alta.
Estabilidad	Alta.
Descripción	Da de alta a un trabajador del supermercado.
Entrada	Nombre, DNI, mail (puede ser vacío).
Salida	ID del trabajador.
Origen	Interfaz Usuario – Alta trabajador.
Destino	Sistema.
Necesita	Base de datos de trabajadores.
Acción	Se verifica que el DNI tiene exactamente 9 dígitos y no está repetido en nuestra base de datos. En caso de que el trabajador exista, pero no esté activo, se vuelve a activar. En caso de que no exista, se da de alta. En caso negativo, se mostrará un mensaje de error.
Precondición	Los valores introducidos deben ser válidos o si existe, no tiene que estar activo.
Postcondición	El trabajador se creará en la BD y se quedará activo.
Efectos laterales	Si falla el procedimiento avisar mediante un mensaje para repetir el proceso.

Función 3.2.3.2 Buscar datos de un trabajador por ID	
Prioridad	Media.
Estabilidad	Alta.
Descripción	Buscar y mostrar los datos de un trabajador filtrado por ID.
Entrada	ID_Trabajador.
Salida	Todos los atributos del trabajador.
Origen	Interfaz Usuario –Buscar trabajador por ID.
Destino	Sistema.
Necesita	Base de datos de trabajadores.
Acción	Se comprueba que el ID del trabajador existe y está activo en la BD y en caso afirmativo, se muestra toda la información correspondiente a esa fila.
Precondición	El trabajador debe existir en la BD y estar activo.
Postcondición	Se mostrará la información del trabajador.
Efectos laterales	Ninguno.

Función 3.2.3.3	Buscar datos de un trabajador por DNI
Prioridad	Media.
Estabilidad	Alta.
Descripción	Buscar y mostrar los datos de un trabajador filtrado por DNI.
Entrada	ID_Trabajador.
Salida	Todos los atributos del trabajador.
Origen	Interfaz Usuario – Buscar trabajador por DNI.
Destino	Sistema.
Necesita	Base de datos de trabajadores.
Acción	Se comprueba que el DNI del trabajador introducido existe en la BD y está activo, en caso afirmativo se muestra toda la información correspondiente a la fila.
Precondición	El trabajador debe existir en la BD y estar activo.
Postcondición	Se mostrará la información del trabajador.
Efectos laterales	Ninguno.

Función 3.2.3.4	Buscar datos de todos los trabajadores
Prioridad	Alta.
Estabilidad	Alta.
Descripción	Buscar y mostrar los datos de todos los trabajadores.
Entrada	Ninguna.
Salida	Todos los atributos de los trabajadores.
Origen	Interfaz Usuario – Buscar todos trabajadores.
Destino	Sistema.
Necesita	Base de datos de trabajadores.
Acción	Se muestra un listado con todos los trabajadores activos y su información.
Precondición	Ninguna.
Postcondición	Muestra un listado de todos los trabajadores.
Efectos laterales	Ninguno.

Función 3.2.3.5 Modificar datos de un trabajador	
Prioridad	Alta.
Estabilidad	Media.
Descripción	Modificar los datos de un trabajador.
Entrada	Datos del usuario que se desean modificar.
Salida	Mensaje de éxito.
Origen	Interfaz Usuario – Modificar trabajador.
Destino	Sistema.
Necesita	Base de datos de trabajadores.
Acción	Tras seleccionar manualmente al trabajador que se desea modificar, se permitirá editar los campos no únicos (ID y DNI) de dicho trabajador. Tras la edición, se almacenan los cambios y se imprime un mensaje por pantalla.
Precondición	El trabajador debe existir en la BD y estar activo.
Postcondición	Se quedarán guardados en la BD los cambios del trabajador.
Efectos laterales	Ninguno.

Función 3.2.3.6 Dar de baja a un trabajador	
Prioridad	Alta.
Estabilidad	Alta.
Descripción	Da de baja a un trabajador del supermercado.
Entrada	ID del trabajador.
Salida	Mensaje de éxito.
Origen	Interfaz Usuario – Baja trabajador
Destino	Sistema.
Necesita	Base de datos de trabajadores.
Acción	Buscará al trabajador por ID y se eliminará de la BD.
Precondición	El trabajador debe existir en la BD y estar activo.
Postcondición	El trabajador se quedará inactivo.
Efectos laterales	Si falla el procedimiento avisar mediante un mensaje para repetir el proceso.

Función 3.2.3.7 Identificar al trabajador	
Prioridad	Alta.
Estabilidad	Alta.
Descripción	Iniciar sesión en el sistema como trabajador.
Entrada	ID_Trabajador.
Salida	Mensaje de éxito en registro del trabajador.
Origen	Interfaz Usuario – Empleado.
Destino	Sistema.
Necesita	Base de datos de trabajadores.
Acción	Se introduce el ID del trabajador. En caso de éxito, la pantalla se actualizará mostrando opciones adicionales para el trabajador que ha iniciado sesión.
Precondición	El trabajador debe existir en la BD y estar activo.
Postcondición	El trabajador habrá iniciado sesión.
Efectos laterales	Ninguno.

3.2.4 Módulo de Clientes (Supermercado)

Función 3.2.4.1 Dar de alta a un nuevo cliente	
Prioridad	Alta.
Estabilidad	Alta.
Descripción	Da de alta a un nuevo cliente.
Entrada	Nombre, mail, DNI.
Salida	ID del cliente.
Origen	Interfaz Usuario – Nuevo Cliente.
Destino	Sistema.
Necesita	Base de datos de clientes.
Acción	Comprobar que los datos sean correctos y que no hay otros clientes con el mismo DNI en caso afirmativo dar de alta al cliente y devolver ID.
Precondición	Los datos introducidos deben ser válidos y que si existe, no esté activo.
Postcondición	El cliente será creado en la BD y se quedará activo.
Efectos laterales	Si falla el procedimiento avisar mediante un mensaje para repetir el proceso.

Función 3.2.4.2 Buscar datos de un cliente	
Prioridad	Media.
Estabilidad	Alta.
Descripción	Buscar y mostrar los datos de un cliente.
Entrada	ID_Socio.
Salida	Todos los atributos del cliente.
Origen	Interfaz Usuario – Buscar cliente.
Destino	Sistema.
Necesita	Base de datos de clientes.
Acción	Comprueba si existe el cliente que se quiere buscar. En caso de existir muestra todos sus atributos, si no, muestra mensaje de error.
Precondición	El cliente debe existir en la BD y estar activo.
Postcondición	El cliente será mostrado con su información.
Efectos laterales	Ninguno.

Función 3.2.4.3		Buscar datos de todos los clientes
Prioridad		Media.
Estabilidad		Alta.
Descripción		Buscar y mostrar los datos de todos los clientes.
Entrada		Ninguna.
Salida		Todos los atributos de los clientes.
Origen		Interfaz Usuario – Buscar todos los clientes.
Destino		Sistema.
Necesita		Base de datos de clientes.
Acción		Busca y muestra en forma de lista todos los clientes.
Precondición		Ninguna.
Postcondición		Se muestra por pantalla la información de los clientes en formato lista.
Efectos laterales		Ninguno.

Función 3.2.4.4		Modificar datos de un cliente
Prioridad		Media.
Estabilidad		Media.
Descripción		Modificar los datos de un cliente.
Entrada		ID del cliente a modificar y los atributos que se desean modificar.
Salida		Mensaje de confirmación.
Origen		Interfaz Usuario – Modificar cliente.
Destino		Sistema.
Necesita		Base de datos de clientes.
Acción		Buscará al cliente por ID y se modificarán los datos que haya introducido el usuario por los que había antes.
Precondición		El cliente tiene que existir en la BD y estar activo.
Postcondición		Se quedarán guardados los cambios realizados en la BD.
Efectos laterales		Ninguno.

Función 3.2.4.5 Dar de baja a un cliente	
Prioridad	Media
Estabilidad	Alta.
Descripción	Da de baja a un cliente.
Entrada	ID del cliente.
Salida	Mensaje de éxito.
Origen	Interfaz Usuario – Baja cliente.
Destino	Sistema.
Necesita	Base de datos de clientes.
Acción	Buscar al cliente por su ID en la BD y eliminarlo de ella.
Precondición	El cliente tiene que existir en la BD y estar activo.
Postcondición	El cliente quedará inactivo.
Efectos laterales	Si falla el procedimiento avisar mediante un mensaje para repetir el proceso.

3.2.5 Módulo de Compras

Función 3.2.5.1 Abrir Compra	
Prioridad	Alta.
Estabilidad	Media.
Descripción	Abre una nueva compra.
Entrada	ID_Trabajador (si es autocompra será siempre 0).
Salida	Mensaje de éxito o error.
Origen	Interfaz Usuario – Iniciar Compra.
Destino	Sistema.
Necesita	Base de datos de trabajadores.
Acción	Comprueba que el trabajador que abre la compra existe y crea la estructura necesaria OO para almacenar el carrito.
Precondición	El trabajador debe existir en la BD para iniciar una compra.
Postcondición	La compra estará iniciada.
Efectos laterales	Ninguno.

Función 3.2.5.2 Añadir productos al carrito	
Prioridad	Alta.
Estabilidad	Alta.
Descripción	Añade un producto al carrito de la compra.
Entrada	ID_Producto y cantidad.
Salida	Mensaje de éxito o error.
Origen	Interfaz usuario – Añadir producto al carrito.
Destino	Sistema.
Necesita	Un carrito previamente abierto.
Acción	Se guarda en el carrito el producto con el ID introducido.
Precondición	El carrito debe estar abierto .
Postcondición	El producto quedará añadido al carrito.
Efectos laterales	Ninguno.

Función 3.2.5.3 Eliminar producto del carrito	
Prioridad	Alta.
Estabilidad	Media.
Descripción	Elimina un producto del carrito de la compra.
Entrada	ID_Producto.
Salida	Mensaje de éxito o error.
Origen	Interfaz Usuario – Eliminar producto del carrito.
Destino	Sistema.
Necesita	Un carrito previamente abierto.
Acción	Dado el ID del producto se elimina del carrito.
Precondición	El producto debe de haber sido añadido previamente al carrito.
Postcondición	El producto quedará eliminado del carrito.
Efectos laterales	Ninguno.

Función 3.2.5.4 Cerrar compra	
Prioridad	Alta.
Estabilidad	Alta.
Descripción	Finaliza la compra y procede al pago.
Entrada	ID_Cliente.
Salida	Mensaje de éxito o error.
Origen	Interfaz Usuario – Cerrar compra.
Destino	Sistema.
Necesita	Base de datos de productos y de compras (ventas).
Acción	Comprobar que todos los productos tienen stock suficiente, modificarlo y guardar la compra con toda su información relevante.
Precondición	Los productos deben tener stock suficiente y el cliente debe existir.
Postcondición	La compra finalizará y quedará registrada en la BD.
Efectos laterales	Si alguno de los productos no existe o no tiene stock suficiente la compra no tendrá éxito.

Función 3.2.5.5		Buscar compra
Prioridad		Alta.
Estabilidad		Media.
Descripción		Muestra una compra con toda su información.
Entrada		ID_Compra.
Salida		Mensaje de éxito o error.
Origen		Interfaz Usuario – Buscar compra.
Destino		Sistema.
Necesita		Base de datos de compras (ventas).
Acción		Dado un ID_Compra busca en la BD la compra y la muestra con toda su información.
Precondición		La compra debe existir en la BD.
Postcondición		La compra será mostrada con toda su información.
Efectos laterales		Ninguno.

Función 3.2.5.6		Buscar todas las compras
Prioridad		Alta.
Estabilidad		Media.
Descripción		Muestra todas las compras realizadas.
Entrada		Ninguna.
Salida		Mensaje de éxito o error.
Origen		Interfaz Usuario – Buscar todas las compras.
Destino		Sistema.
Necesita		Base de datos de compras (ventas).
Acción		Accede a la BD, obtiene las compras (ventas) y las muestra en forma de lista.
Precondición		Ninguna.
Postcondición		Las compras se mostrarán en forma de lista.
Efectos laterales		Ninguno.

Función 3.2.5.7		Buscar todas las compras de un cliente
Prioridad		Alta.
Estabilidad		Media.
Descripción		Muestra todas las compras realizadas por un cliente.
Entrada		ID del cliente.
Salida		Mensaje de éxito o error.
Origen		Interfaz Usuario – Buscar compras de un cliente.
Destino		Sistema.
Necesita		Base de datos de compras (ventas).
Acción		Accede a la BD, obtiene las compras (ventas) del cliente solicitado y las muestra en forma de lista.
Precondición		El cliente debe existir en la BD y estar activo.
Postcondición		Las compras del cliente se mostrarán en forma de lista.
Efectos laterales		Ninguno.

Función 3.2.5.8		Realizar devolución
Prioridad		Alta.
Estabilidad		Media.
Descripción		Permite devolver uno o más productos de una compra.
Entrada		ID de la compra y los IDs de los productos a devolver.
Salida		Mensaje de éxito o error.
Origen		Interfaz Usuario – Realizar devolución.
Destino		Sistema.
Necesita		Base de datos de compras (ventas).
Acción		Elimina los productos que se quieren devolver de una compra en la BD.
Precondición		La compra debe existir en la BD y los productos deben permanecer a esa compra.
Postcondición		Se realizará la devolución con éxito.
Efectos laterales		Ninguno.

3.2.6 Módulo de Proveedores

Función 3.2.6.1		Añadir datos de un nuevo proveedor
Prioridad		Alta.
Estabilidad		Alta.
Descripción		Dar de alta a un nuevo proveedor.
Entrada		Nombre del proveedor.
Salida		ID_Proveedor.
Origen		Interfaz Usuario – Añadir nuevo proveedor.
Destino		Sistema.
Necesita		Base de datos de proveedores.
Acción		Añade un nuevo proveedor con los datos introducidos y se incorpora a la BD.
Precondición		Los datos introducidos sean válidos y que si existe que no esté activo.
Postcondición		El proveedor será creado en la BD y se quedará activo.
Efectos laterales		Si falla el procedimiento avisar mediante un mensaje para repetir el proceso.

Función 3.2.6.2		Búsqueda proveedores
Prioridad		Media.
Estabilidad		Media.
Descripción		Se buscarán y mostrarán los datos del proveedor.
Entrada		Ninguna.
Salida		Datos de todos los proveedores.
Origen		Interfaz Usuario – Buscar proveedores.
Destino		Sistema.
Necesita		Base de datos de proveedores.
Acción		Buscará y mostrará por pantalla una lista de los proveedores con su información.
Precondición		Ninguna.
Postcondición		Se mostrará una lista de proveedores.
Efectos laterales		Ninguno.

Función 3.2.6.3		Buscar datos de un proveedor
Prioridad		Media.
Estabilidad		Media.
Descripción		Busca y muestra los datos del proveedor solicitado.
Entrada		ID_Proveedor.
Salida		Datos del proveedor.
Origen		Interfaz Usuario – Buscar proveedor.
Destino		Sistema.
Necesita		Base de datos proveedores.
Acción		Buscará el proveedor y mostrará por pantalla su información.
Precondición		El proveedor debe existir en la BD y estar activo.
Postcondición		Se mostrarán todos los atributos del proveedor.
Efectos laterales		Ninguno.

Función 3.2.6.4		Editar la información de un proveedor
Prioridad		Media.
Estabilidad		Media.
Descripción		Se modificarán los datos que se quieran del proveedor seleccionado.
Entrada		ID_Proveedor.
Salida		Mensaje de confirmación.
Origen		Interfaz Usuario – Editar proveedor
Destino		Sistema.
Necesita		Base de datos de proveedores.
Acción		Buscará el proveedor deseado en la BD y lo modificará.
Precondición		El proveedor debe existir en la BD y estar activo.
Postcondición		Se quedarán registrados los cambios del proveedor.
Efectos laterales		Ninguno.

Función 3.2.6.5 Dar de baja un proveedor	
Prioridad	Media.
Estabilidad	Media.
Descripción	Se dará de baja a un proveedor.
Entrada	ID_Proveedor.
Salida	Mensaje de confirmación.
Origen	Interfaz Usuario – Eliminar proveedor.
Destino	Sistema.
Necesita	Base de datos de proveedores.
Acción	Se dará de baja lógica al proveedor de la BD.
Precondición	El proveedor debe existir en la BD y estar activo.
Postcondición	El proveedor se quedará inactivo.
Efectos laterales	Ninguno.

3.2.7 Módulo de Marcas

Función 3.2.7.1		Añadir nuevas marcas
Prioridad		Alta.
Estabilidad		Alta.
Descripción		Añadir marca a la BD.
Entrada		Nombre de la marca.
Salida		ID_Marca.
Origen		Interfaz Usuario – Añadir nueva marca.
Destino		Sistema.
Necesita		Base de datos de marcas.
Acción		Comprobar que la marca no existe o está inactiva, si no lo está, se activa y si no existe, la añade y activa.
Precondición		Los datos introducidos son válidos y que si existe que no esté activo.
Postcondición		La marca quedará añadida y quedará activa.
Efectos laterales		Si falla el procedimiento avisar mediante un mensaje para repetir el proceso.

Función 3.2.7.2		Busca información de una marca
Prioridad		Alta.
Estabilidad		Media.
Descripción		Busca y muestra el nombre de la marca.
Entrada		ID_Marca.
Salida		Datos de la marca.
Origen		Interfaz Usuario – Buscar marca.
Destino		Sistema.
Necesita		Base de datos de marcas.
Acción		Busca en BD de marcas los datos introducidos de la marca deseada y mostrará por pantalla la información.
Precondición		La marca debe existir en la BD y estar activa.
Postcondición		Se mostrará la marca con todos sus atributos.
Efectos laterales		Ninguno.

Función 3.2.7.3		Buscar Información de las marcas
Prioridad		Media.
Estabilidad		Media.
Descripción		Busca y muestra los nombres de las marcas.
Entrada		Ninguna.
Salida		Datos de las marcas.
Origen		Interfaz Usuario – Buscar todas marcas.
Destino		Sistema.
Necesita		Base de datos de marcas.
Acción		Buscará las marcas y mostrará por pantalla su información.
Precondición		Ninguna.
Postcondición		Se mostrará un listado de las marcas.
Efectos laterales		Ninguno.

Función 3.2.7.4		Editar datos específicos de una marca
Prioridad		Media.
Estabilidad		Media.
Descripción		Modificar los datos de una marca.
Entrada		ID de la marca y los datos a modificar.
Salida		Mensaje de confirmación.
Origen		Interfaz Usuario – Editar marca.
Destino		Sistema.
Necesita		Base de datos de marca.
Acción		Busca la marca y se modifican los datos deseados.
Precondición		La marca debe existir en la BD y estar activa.
Postcondición		Se quedarán guardados en la BD los cambios de la marca.
Efectos laterales		Ninguno.

Función 3.2.7.5 Eliminar una marca	
Prioridad	Alta.
Estabilidad	Alta.
Descripción	Baja lógica de una marca de la BD.
Entrada	ID_Marca.
Salida	Mensaje de confirmación.
Origen	Interfaz Usuario - Eliminar marca.
Destino	Sistema.
Necesita	Base de datos de marcas.
Acción	Introducir datos de una marca, comprobando si existe, si su stock está vacío, y entonces la eliminará.
Precondición	La marca debe existir en la BD y estar activa.
Postcondición	Se dará de forma lógica la marca.
Efectos laterales	Ninguno.

3.2.8 Módulo de Clientes (Restaurante)

Función 3.2.8.1		Alta Cliente
Prioridad		Alta.
Estabilidad		Alta.
Descripción		Da de alta un nuevo cliente.
Entrada		DNI, nombre, apellidos.
Salida		Mensaje de éxito o error.
Origen		Interfaz de Usuario – Alta Cliente Restaurante.
Destino		Sistema.
Necesita		Base de datos de clientes.
Acción		Comprobar que los datos sean correctos y no haya ningún cliente con el mismo DNI, en caso afirmativo dar de alta.
Precondición		Los datos introducidos deben ser válidos y que si existe que no esté activo.
Postcondición		El cliente será creado en la BD y se quedará activo.
Efectos laterales		Si falla el procedimiento avisar mediante mensaje para repetir el proceso.

Función 3.2.8.2		Buscar cliente
Prioridad		Media.
Estabilidad		Alta.
Descripción		Buscar y mostrar los datos de un cliente.
Entrada		DNI.
Salida		Todos los atributos del cliente.
Origen		Interfaz Usuario – Buscar cliente restaurante
Destino		Sistema.
Necesita		Base de datos de clientes.
Acción		Insertar DNI del cliente, si existe, muestra todos sus atributos.
Precondición		El cliente debe existir en la BD y estar activo.
Postcondición		Muestra toda la información del cliente.
Efectos laterales		Ninguno.

Función 3.2.8.3		Listado clientes
Prioridad		Media.
Estabilidad		Alta.
Descripción		Busca y muestra un listado de todos los clientes.
Entrada		Ninguna.
Salida		Listado de todos los clientes.
Origen		Interfaz Usuario – Listado clientes
Destino		Sistema.
Necesita		Base de datos de clientes.
Acción		Busca y muestra todos los clientes de la BD en forma de lista.
Precondición		Ninguna.
Postcondición		Mostrará un listado de todos los clientes.
Efectos laterales		Ninguno.

Función 3.2.8.4		Modificar cliente
Prioridad		Media.
Estabilidad		Alta.
Descripción		Modificar los atributos de un cliente.
Entrada		DNI, Nombre, Apellido, N°Cuenta, pedidosPagados.
Salida		Mensaje de confirmación.
Origen		Interfaz de Usuario – Modificar cliente.
Destino		Sistema.
Necesita		Base de datos de clientes.
Acción		Comprueba si existe el cliente que se quiere modificar. En caso de existir y estar activo muestra todos sus atributos para modificarlos.
Precondición		El cliente debe existir en la BD y estar activo.
Postcondición		Se quedarán guardados los cambios realizados en la BD.
Efectos laterales		Ninguno.

Función 3.2.8.5		Baja cliente
Prioridad		Media.
Estabilidad		Alta.
Descripción		Dar de baja un cliente de la BD.
Entrada		DNI.
Salida		Mensaje de éxito.
Origen		Interfaz de Usuario – Baja cliente.
Destino		Sistema.
Necesita		Base de datos de clientes.
Acción		Busca al cliente por su ID en la BD y lo elimina de ella.
Precondición		El cliente debe existir en la BD y estar activo.
Postcondición		El cliente quedará inactivo.
Efectos laterales		Si falla el procedimiento avisar mediante un mensaje para repetir el proceso.

3.2.9 Módulo de Ingredientes

Función 3.2.9.1		Añadir ingrediente nuevo
Prioridad		Alta.
Estabilidad		Alta.
Descripción		Añade un nuevo ingrediente a la BD del ingrediente.
Entrada		Nombre ingrediente, cantidad.
Salida		ID_Ingrediente.
Origen		Interfaz Usuario – Añadir Ingrediente.
Destino		Sistema.
Necesita		Base de datos de ingredientes.
Acción		Busca si existe el ingrediente en la BD y si no existe lo añade a esta, si existe y está inactiva, la activa.
Precondición		Los datos introducidos tienen que ser válidos y que si existe que se active.
Postcondición		El ingrediente será añadido en la BD y se quedará activo.
Efectos laterales		Si falla el procedimiento avisar mediante un mensaje para repetir el proceso.

Función 3.2.9.2		Ver lista de ingredientes disponibles
Prioridad		Media.
Estabilidad		Alta.
Descripción		Muestra la lista de ingredientes disponibles en la BD.
Entrada		Ninguna.
Salida		Datos de todos los ingredientes.
Origen		Interfaz Usuario – Ver todos ingredientes.
Destino		Sistema.
Necesita		Base de datos de ingredientes.
Acción		Busca y muestra en forma de lista todos los ingredientes.
Precondición		Ninguna.
Postcondición		Los ingredientes serán mostrados en forma de lista.
Efectos laterales		Ninguno.

Función 3.2.9.3		Buscar ingrediente
Prioridad		Alta.
Estabilidad		Alta.
Descripción		Busca un ingrediente en la BD.
Entrada		Nombre ingrediente.
Salida		Mostrar datos del ingrediente.
Origen		Interfaz del usuario – Buscar ingrediente.
Destino		Sistema.
Necesita		Base de datos de ingredientes.
Acción		Introducir el ID de un ingrediente, comprobar si existe, y mostrar su información.
Precondición		El ingrediente debe existir en la BD y estar activo.
Postcondición		El ingrediente será mostrado con todos sus atributos.
Efectos laterales		Ninguno.

Función 3.2.9.4		Modificar ingrediente
Prioridad		Alta.
Estabilidad		Media.
Descripción		Modifica los atributos de un ingrediente.
Entrada		Nombre del ingrediente.
Salida		Mensaje de confirmación.
Origen		Interfaz del usuario – Modificar ingrediente.
Destino		Sistema.
Necesita		Base de datos de ingredientes.
Acción		Introducir el nombre del ingrediente, comprobar si existe, y cambiar los datos.
Precondición		El ingrediente debe existir en la BD y estar activo.
Postcondición		Se quedarán registrados los cambios del ingrediente.
Efectos laterales		Ninguno.

Función 3.2.9.5 Eliminar ingrediente	
Prioridad	Baja.
Estabilidad	Alta.
Descripción	Elimina de forma lógica un ingrediente de la BD.
Entrada	Nombre del ingrediente.
Salida	Mensaje de confirmación.
Origen	Interfaz de usuario – Eliminar ingrediente.
Destino	Sistema.
Necesita	Base de datos de ingredientes.
Acción	Introducir el nombre de un ingrediente, comprobará si existe, y le dará de baja lógica en la BD.
Precondición	El ingrediente debe existir en la BD y estar activo.
Postcondición	El ingrediente quedará inactivo.
Efectos laterales	Si falla el procedimiento avisar mediante un mensaje para repetir el proceso.

3.2.10 Módulo de Pedidos

Función 3.2.10.1		Abrir pedido nuevo
Prioridad		Alta.
Estabilidad		Alta.
Descripción		Abre un nuevo pedido.
Entrada		ID_Personal
Salida		Mensaje de confirmación.
Origen		Interfaz Usuario – Abrir nuevo pedido.
Destino		Sistema
Necesita		Base de datos de personal.
Acción		Primero, comprobar que los datos que el personal ha introducido son correctos, y abrir el nuevo pedido.
Precondición		El empleado exista en la BD y este activo.
Postcondición		Abre el pedido.
Efectos laterales		Si falla el procedimiento avisar mediante un mensaje para repetir el proceso.

Función 3.2.10.2		Buscar pedidos de un cliente
Prioridad		Media.
Estabilidad		Alta.
Descripción		Muestra todos los pedidos realizados por un cliente.
Entrada		DNI.
Salida		Todos los atributos del cliente y de los pedidos.
Origen		Interfaz Usuario – Buscar pedido de cliente.
Destino		Sistema.
Necesita		Base de datos de pedidos y de clientes.
Acción		Introduce el DNI del cliente, si existe se muestran todos los pedidos realizados por un cliente.
Precondición		El cliente debe existir en la BD y estar activo.
Postcondición		Se muestra el listado de pedidos del cliente.
Efectos laterales		Ninguno.

Función 3.2.10.3		Buscar todos los pedidos
Prioridad		Media.
Estabilidad		Alta.
Descripción		Muestra todos los pedidos de la BD.
Entrada		Ninguna.
Salida		Mostrará todos los pedidos en la BD, con sus IDs y sus platos.
Origen		Interfaz Usuario – Buscar pedidos.
Destino		Sistema.
Necesita		Base de datos de pedidos.
Acción		Buscará y mostrará los pedidos en forma de lista.
Precondición		Ninguna.
Postcondición		Se mostrará el listado de pedidos.
Efectos laterales		Ninguno.

Función 3.2.10.4		Modificar pedido
Prioridad		Media.
Estabilidad		Alta.
Descripción		Permite modificar un pedido ya realizado.
Entrada		ID_Pedido, nombrePlatos.
Salida		Mensaje de confirmación.
Origen		Interfaz Usuario – Modificar pedido.
Destino		Sistema.
Necesita		Base de datos de pedidos.
Acción		Comprueba que el pedido exista, y si existe permite modificarlo.
Precondición		El pedido debe existir en la BD y estar activo.
Postcondición		Se quedarán registrados los cambios del pedido.
Efectos laterales		Ninguno.

Función 3.2.10.5		Eliminar pedido
Prioridad		Media.
Estabilidad		Alta.
Descripción		Elimina de forma lógica un pedido de la BD.
Entrada		ID_Pedido.
Salida		Mensaje de confirmación.
Origen		Interfaz Usuario – Eliminar pedido.
Destino		Sistema.
Necesita		Base de datos de pedidos.
Acción		Se comprueba que existe el pedido, en ese caso se realizará la baja lógica de éste.
Precondición		El pedido debe existir en la BD y estar activo.
Postcondición		El pedido quedará inactivo.
Efectos laterales		Si falla el procedimiento avisar mediante un mensaje para repetir el proceso.

Función 3.2.10.6		Cerrar pedido
Prioridad		Alta.
Estabilidad		Alta.
Descripción		Se cerrará el pedido.
Entrada		DNI del cliente.
Salida		Mensaje de confirmación.
Origen		Interfaz Usuario – Cerrar pedido.
Destino		Sistema.
Necesita		Base de datos de pedidos.
Acción		Comprobará si existe el cliente en la BD, si existe, se cerrará el pedido y se guardará en la BD.
Precondición		El cliente debe existir en la BD y estar activo.
Postcondición		Se cerrará el pedido y se actualizará la BD.
Efectos laterales		Ninguno.

Función 3.2.10.7		Añadir plato a pedido
Prioridad		Alta.
Estabilidad		Alta.
Descripción		Añade un plato a un pedido.
Entrada		nombrePlato, cantidad.
Salida		Mensaje de confirmación.
Origen		Interfaz Usuario – Añadir plato a pedido.
Destino		Sistema.
Necesita		Un pedido previamente abierto
Acción		Se guarda en el pedido el plato del nombre introducido.
Precondición		Debe estar el pedido abierto.
Postcondición		El plato se añadirá al pedido.
Efectos laterales		Ninguno.

Función 3.2.10.8		Eliminar plato de pedido
Prioridad		Alta.
Estabilidad		Alta.
Descripción		Elimina un plato de un pedido.
Entrada		nombrePlato.
Salida		Mensaje de confirmación.
Origen		Interfaz Usuario – Eliminar plato de pedido.
Destino		Sistema.
Necesita		Un pedido previamente abierto.
Acción		Dado el nombre del plato lo elimina del pedido.
Precondición		El plato debe de encontrarse ya en el pedido.
Postcondición		El plato se elimina del pedido.
Efectos laterales		Ninguno.

3.2.11 Módulo de Personal

Función 3.2.11.1		Añadir empleado
Prioridad		Alta.
Estabilidad		Alta.
Descripción		Da de alta a un empleado del restaurante.
Entrada		Nombre, email, DNI.
Salida		ID_Empleado.
Origen		Interfaz de usuario – Añadir empleado.
Destino		Sistema.
Necesita		Base de datos de empleados.
Acción		Añade un nuevo empleado del restaurante introduciéndolo en la base de datos.
Precondición		Los datos introducidos deben ser válidos y que si existe que se active.
Postcondición		El empleado se creará en la BD y se quedará activo.
Efectos laterales		Si falla el procedimiento avisar mediante un mensaje para repetir el proceso.

Función 3.2.11.2		Buscar empleados
Prioridad		Media.
Estabilidad		Media.
Descripción		Buscar y mostrar a todos los empleados del restaurante.
Entrada		Ninguna.
Salida		Datos de todos los empleados del restaurante.
Origen		Interfaz de usuario – Buscar empleados.
Destino		Sistema.
Necesita		Base de datos de empleados.
Acción		Buscará los empleados que existen y mostrará por pantalla su información.
Precondición		Ninguna.
Postcondición		Se mostrará un listado de los empleados.
Efectos laterales		Ninguno.

Función 3.2.11.3		Buscar un empleado por ID
Prioridad		Media.
Estabilidad		Alta.
Descripción		Buscar y mostrar los datos de un empleado del restaurante, filtrado por su ID.
Entrada		ID_Empleado
Salida		Datos del empleado del restaurante.
Origen		Interfaz del usuario – Buscar empleado por ID.
Destino		Sistema.
Necesita		Base de datos de empleados.
Acción		Buscará al empleado, filtrado por su ID, y mostrará por pantalla su información.
Precondición		El empleado debe existir en la BD y estar activo.
Postcondición		Se mostrarán todos los atributos del empleado.
Efectos laterales		Ninguno.

Función 3.2.11.4		Buscar un empleado por DNI
Prioridad		Media.
Estabilidad		Alta.
Descripción		Buscar y mostrar los datos de un empleado del restaurante, filtrado por su DNI.
Entrada		DNI del empleado.
Salida		Datos del empleado.
Origen		Interfaz de usuario – Buscar empleado por DNI.
Destino		Sistema.
Necesita		Base de datos de empleados.
Acción		Buscará al empleado, filtrado por su DNI, y mostrará por pantalla su información.
Precondición		El empleado debe existir en la BD y estar activo.
Postcondición		Se mostrarán los atributos del empleado.
Efectos laterales		Ninguno.

Función 3.2.11.5		Editar información de un empleado
Prioridad		Media.
Estabilidad		Media.
Descripción		Se modifican los datos deseados del empleado del restaurante.
Entrada		ID_Empleado.
Salida		Mensaje de confirmación.
Origen		Interfaz de usuario – Editar empleado.
Destino		Sistema.
Necesita		Base de datos de empleados.
Acción		Buscará el empleado del restaurante solicitado en la BD y modificará sus datos.
Precondición		El empleado debe existir en la BD y estar activo.
Postcondición		Se quedarán registrados los cambios del empleado.
Efectos laterales		Ninguno.

Función 3.2.11.6		Dar de baja a un empleado
Prioridad		Media.
Estabilidad		Media.
Descripción		Da de baja a un empleado del restaurante.
Entrada		ID_Empleado.
Salida		Mensaje de confirmación.
Origen		Interfaz de usuario – Baja empleado.
Destino		Sistema.
Necesita		Base de datos de empleados.
Acción		Se realizará la baja lógica de un empleado del restaurante de la BD.
Precondición		El empleado debe existir en la BD y estar activo.
Postcondición		El empleado se quedará inactivo.
Efectos laterales		Si falla el procedimiento avisar mediante un mensaje para repetir el proceso.

Función 3.2.11.7 Identificar empleado (Iniciar Sesión)	
Prioridad	Alta.
Estabilidad	Alta.
Descripción	Inicia sesión como empleado del restaurante.
Entrada	ID_Empleado.
Salida	Mensaje de éxito en la operación.
Origen	Interfaz de usuario – Identificar empleado
Destino	Sistema.
Necesita	Base de datos de empleados.
Acción	Se introduce el ID del empleado comprobando si existe, si existe se muestran las opciones a las que tiene acceso.
Precondición	El empleado debe existir en la BD y estar activo.
Postcondición	El empleado habrá iniciado sesión al software de restaurante.
Efectos laterales	Ninguno.

3.2.12 Módulo de Plato

Función 3.2.12.1		Crear nuevo plato
Prioridad		Alta.
Estabilidad		Alta.
Descripción		Crea un nuevo plato en la base de datos.
Entrada		Stock, Nombre, Descripción, Precio, ID_IngredienteInicial.
Salida		Mensaje de confirmación.
Origen		Interfaz de usuario – Crear plato.
Destino		Sistema.
Necesita		Base de datos de platos.
Acción		Comprobar que no existe ya un plato con ese nombre, si no existe se añade a la BD.
Precondición		Los datos introducidos sean válidos y que si existe que esté activo.
Postcondición		El plato se creará en la BD y se quedará activo.
Efectos laterales		Si falla el procedimiento avisar mediante un mensaje para repetir el proceso.

Función 3.2.12.2		Añadir ingrediente a un plato
Prioridad		Media.
Estabilidad		Alta.
Descripción		Añade un nuevo ingrediente a un plato.
Entrada		ID_Ingrediente, Cantidad, nombrePlato.
Salida		Mensaje de confirmación.
Origen		Interfaz de usuario – Añadir ingrediente a plato.
Destino		Sistema.
Necesita		Base de datos de platos e ingredientes.
Acción		Comprobar que existen el ingrediente y el plato, si existen y están activos, se añade el ingrediente.
Precondición		El ingrediente y el plato deben existir y estar activos.
Postcondición		El ingrediente se añadirá al plato.
Efectos laterales		Ninguno.

Función 3.2.12.3 Eliminar ingrediente a un plato	
Prioridad	Media.
Estabilidad	Alta.
Descripción	Elimina un ingrediente a un plato.
Entrada	ID_Ingrediente, nombrePlato.
Salida	Mensaje de confirmación.
Origen	Interfaz de usuario – Elimina ingrediente a plato.
Destino	Sistema.
Necesita	Base de datos de platos e ingredientes.
Acción	Comprobar que existen el ingrediente y el plato, si existen y están activos, se elimina el ingrediente.
Precondición	El ingrediente y el plato deben existir y estar activos.
Postcondición	El ingrediente se eliminará al plato.
Efectos laterales	Ninguno.

Función 3.2.12.4 Buscar plato	
Prioridad	Alta.
Estabilidad	Alta.
Descripción	Realiza la búsqueda de un plato.
Entrada	nombrePlato.
Salida	Los atributos del plato solicitado.
Origen	Interfaz Usuario – Buscar plato.
Destino	Sistema.
Necesita	Base de datos de platos.
Acción	Comprobar que existe el plato y está activo, si existe, se muestra el plato con sus atributos.
Precondición	El plato debe existir en la BD y estar activo.
Postcondición	Mostrará los atributos del plato solicitado.
Efectos laterales	Ninguno.

Función 3.2.12.5		Buscar todos los platos
Prioridad		Media.
Estabilidad		Alta.
Descripción		Muestra la lista de todos los platos disponibles, con sus atributos.
Entrada		Ninguna.
Salida		La lista de platos de la BD.
Origen		Interfaz Usuario – Buscar todos platos.
Destino		Sistema.
Necesita		Base de datos de platos.
Acción		Se mostrará por pantalla la lista completa de platos con todos sus atributos.
Precondición		Ninguna.
Postcondición		Se mostrará un listado de los platos.
Efectos laterales		Ninguno.

Función 3.2.12.6		Modificar plato
Prioridad		Media.
Estabilidad		Alta.
Descripción		Modifica los atributos del plato.
Entrada		nombrePlato, descripción, precio, stock.
Salida		Muestra el plato con sus atributos modificados.
Origen		Interfaz Usuario – Modificar plato.
Destino		Sistema.
Necesita		Base de datos de platos.
Acción		Buscar el plato, comprobando si existe y esta activo, y modificar los atributos solicitados.
Precondición		El plato debe existir en la BD y estar activo.
Postcondición		Se quedarán guardados en la BD del plato.
Efectos laterales		Ninguno.

Función 3.2.12.7 Eliminar plato	
Prioridad	Baja.
Estabilidad	Alta.
Descripción	Elimina de forma lógica de un plato, no siendo posible ya modificar o pedir ese plato.
Entrada	nombrePlato.
Salida	Mensaje de confirmación.
Origen	Interfaz Usuario – Eliminar plato.
Destino	Sistema.
Necesita	Base de datos de plato.
Acción	Buscar el plato, comprobando si existe y está activo, y lo dará de baja de forma lógica.
Precondición	El plato debe existir en la BD y estar activo.
Postcondición	El plato se quedará inactivo.
Efectos laterales	Si falla el procedimiento avisar mediante un mensaje para repetir el proceso.

3.2.13 Módulo de Turno

Función 3.2.13.1		Crear turno
Prioridad		Alta.
Estabilidad		Alta.
Descripción		Crea un nuevo turno.
Entrada		NombreTurno, horaInicio, horaFin.
Salida		Mensaje de confirmación.
Origen		Interfaz Usuario – Crea turno
Destino		Sistema.
Necesita		Base de datos de turnos.
Acción		Comprueba que no existe ningún turno con ese nombre y, si lo cumple, lo añade.
Precondición		Los datos introducidos deben ser válidos y si existe que no este activo.
Postcondición		El turno será creado en la BD y se quedará activo.
Efectos laterales		Si falla el procedimiento avisar mediante un mensaje para repetir el proceso.

Función 3.2.13.2		Buscar turno
Prioridad		Alta.
Estabilidad		Alta.
Descripción		Muestra un turno con toda su información.
Entrada		NombreTurno.
Salida		Todos los atributos del turno a mostrar.
Origen		Interfaz Usuario – Buscar turno
Destino		Sistema.
Necesita		Base de datos de turnos.
Acción		Comprobar que existe el nombre del turno, si se cumple se mostrará el turno con todos los atributos.
Precondición		El turno debe existir en la BD y estar activo.
Postcondición		Se muestran todos los atributos del turno.
Efectos laterales		Ninguno.

Función 3.2.13.3	Ver lista turnos
Prioridad	Alta.
Estabilidad	Alta.
Descripción	Muestra la lista de turnos.
Entrada	Ninguna.
Salida	Mensaje de confirmación.
Origen	Interfaz Usuario – Ver lista turnos
Destino	Sistema.
Necesita	Base de datos de turnos.
Acción	Busca y muestra todos los turnos en una lista.
Precondición	Ninguna.
Postcondición	Se mostrará el listado de todos los turnos.
Efectos laterales	Ninguno.

Función 3.2.13.3	Modificar turno
Prioridad	Media.
Estabilidad	Alta.
Descripción	Modifica los atributos de turno.
Entrada	NombreTurno.
Salida	Turno cambiado y mensaje de confirmación.
Origen	Interfaz Usuario – Modificar turno.
Destino	Sistema.
Necesita	Base de datos de turnos.
Acción	Comprueba el turno, y modifica los atributos deseados.
Precondición	El turno debe existir en la BD y estar activo.
Postcondición	Se quedarán guardados en la BD los cambios realizados.
Efectos laterales	Ninguno.

Función 3.2.13.4		Eliminar Turno
Prioridad		Media.
Estabilidad		Alta.
Descripción		Eliminar de forma lógica un turno asignado a personal.
Entrada		NombreTurno.
Salida		Mensaje de confirmación.
Origen		Interfaz Usuario – Eliminar turno
Destino		Sistema.
Necesita		Base de datos de turnos.
Acción		Comprobar que existe el nombre del turno. Si se cumple se eliminará el turno.
Precondición		El turno debe existir BD y estar activo.
Postcondición		El turno se quedará inactivo.
Efectos laterales		Ninguno.

3.3 Requisitos de Rendimiento

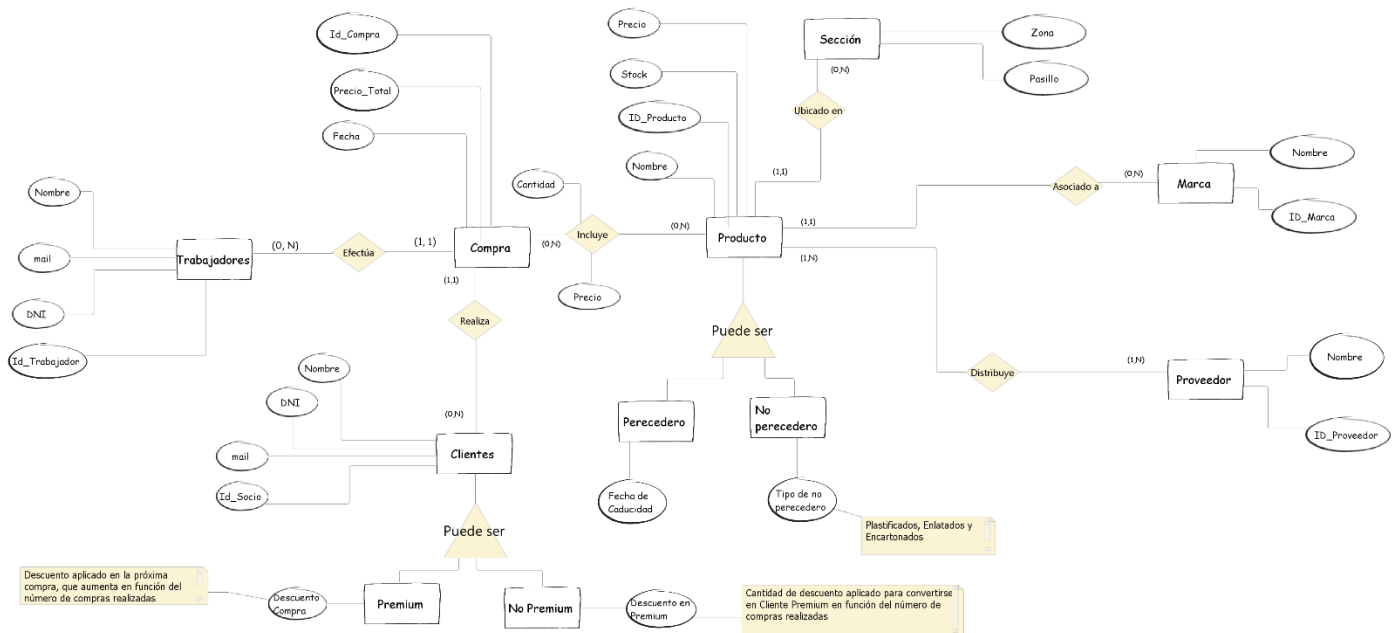
Este software de carácter empresarial está diseñado con el objetivo de reducir el consumo y recursos del sistema aportando gran fiabilidad, seguridad y robustez.

Requisitos mínimos. Especialmente pensado para procesadores compartidos o con recursos muy limitados, este sistema permite trabajar con procesadores, a partir de 1 GHz, de 32 bits. Con una memoria RAM de 1 GB y un almacenamiento disponible de 4 GB. En cuanto al sistema operativo, nuestro software soporta versiones de Windows a partir de Windows 7, así como las principales distribuciones de Linux como Ubuntu, Debian, CentOS o Canaima.

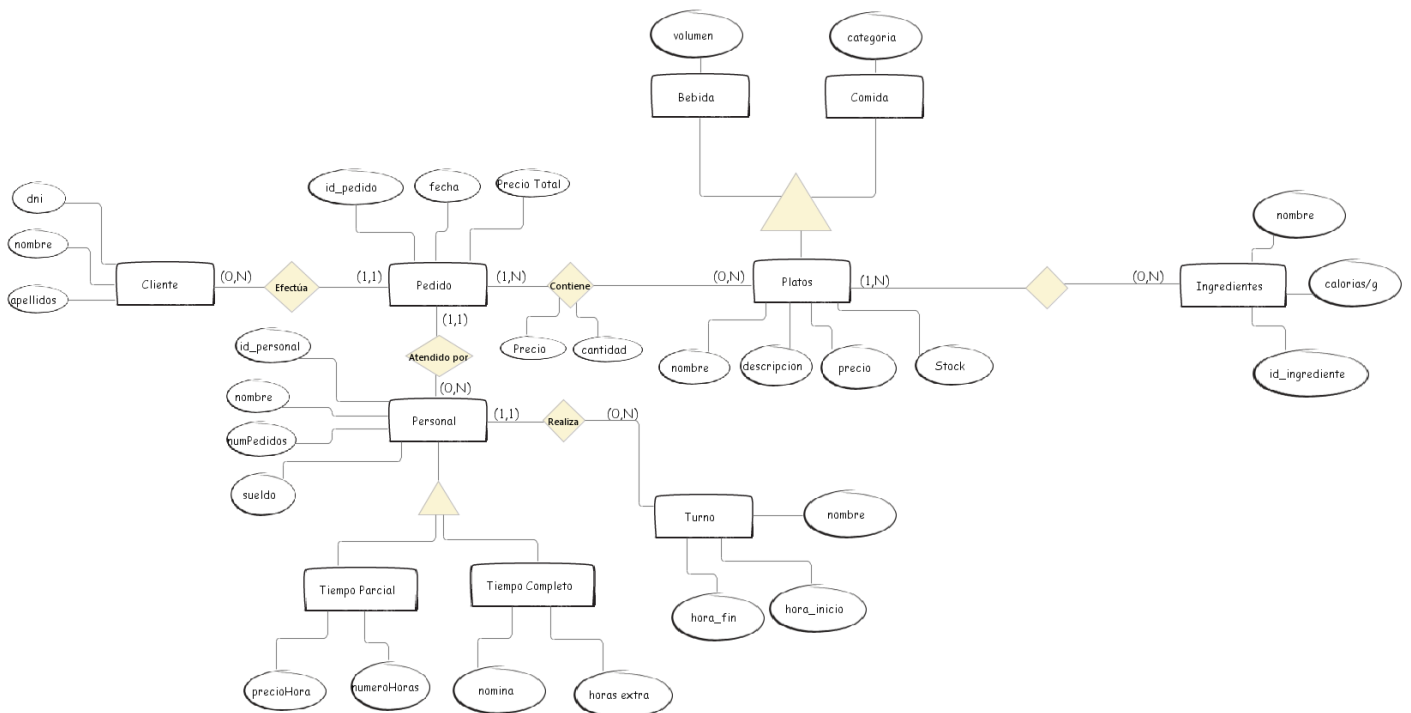
Requisitos recomendados. Para asegurar una experiencia satisfactoria a la hora de trabajar con este software, les recomendamos procesadores especializados para el entorno empresarial, como Intel Xeon E-2104G de 64 bits, con 3,20GHz, o superior. En cuanto a la memoria, se recomienda 4 GB de memoria RAM por sistema implementado, con un máximo de 128 GB. Con respecto al almacenamiento, 4GB para instalar el software es suficiente. Finalmente, recomendamos el uso de este software con Linux como sistema operativo, específicamente la distribución Debian, ya que ofrece gran rendimiento y comodidad tanto para los usuarios finales como para realizar tareas de mantenimiento y actualizaciones.

3.4 Logical Database Requirements (Modelo del Dominio)

3.4.1 Supermercado



3.4.2 Restaurante



3.5 Restricciones de Diseño

El diseño se realizará con el lenguaje UML 2.x el cual desarrollaremos desde la herramienta CASE IBM RSAD. En cuanto al lenguaje que se usará para la codificación será Java, y todo el código será desarrollado en el IDE eclipse que viene incluido en el IBM RSAD y el entorno de ejecución de Java será el JRE 1.8. También realizaremos pruebas en la codificación utilizando las librerías que proporciona JUnit. La aplicación será de escritorio.

Para la pertinencia de datos utilizaremos un formato relacional, el cual gestionaremos e implementaremos mediante el SGBDR MySQL. El servidor que alojará la base de datos será proporcionado por los servicios de la prueba gratuita de Google Cloud.

Utilizaremos el SCV SVN para gestionar el diseño, el código y la documentación de una manera efectiva y ágil para el desarrollo de la aplicación. Tanto los repositorios como los usuarios son proporcionados por la Facultad de Informática.

Tanto para el diseño como la codificación utilizaremos los patrones aprendidos en la asignatura de IS y durante el curso de MS.

3.6 Atributos del Sistema Software

Fiabilidad: Capacidad de un sistema o componente para desempeñar las funciones especificadas. Debe garantizar su correcto funcionamiento, evitar los posibles fallos que puedan producirse en el SGBD y la Base de Datos. Para el correcto uso y tratamiento de los datos han de evitarse fallos de JPA.

Disponibilidad: La disponibilidad de nuestro sistema está ligada y condicionada por la que nos ofrezcan nuestros proveedores. Debe estar disponible (operativo y accesible) el máximo tiempo posible, garantizando mantener el sistema en condiciones óptimas en caso de haber errores.

Seguridad: El sistema debe garantizar la información personal. Solo se debe acceder al sistema de forma autorizada y en un ámbito controlado de manera que personas o sistemas no autorizados no puedan leer o modificar datos.

Mantenibilidad: Se adapta a los posibles cambios durante su desarrollo y posteriores a su uso. La capacidad del sistema para ser modificado efectiva y eficientemente.

Portabilidad: Dado que el sistema será un programa Java, este podrá funcionar en todas las plataformas que dispongan de una versión reciente de la JVM. Además, el gestor de bases de datos MySQL también está disponible para las plataformas que usaremos, y no habrá problemas para la portabilidad del programa. JPA es el encargado de convertir los objetos Java en instrucciones para el Manejador de Base de Datos (MDB).