
Especificación de Requisitos según el estándar de IEEE 830

IEEE Std. 830-1998

Hecho por: Kernel Panic

Alberto Ramos Suárez

Álvaro del Campo Gragera

Antonio Trenado Martín

Carlos Forriol Molina

Fan Ye

Ignacio Redondo Navarro

Juan Embid Sánchez

Juan Montero Gómez

Juan Romo Iribarren

Juan José Encina Fernández

Samuel Parra Calderón

Victor Ortego Martín

28 de Septiembre de 2022

CONTENIDOS

INTRODUCCIÓN.....	3
1.1 PROPÓSITO	3
1.2 ALCANCE.....	3
1.3 DEFINICIONES, ACRÓNIMOS & ABREVIATURAS.....	4
1.4 REFERENCIAS.....	5
1.5 VISIÓN GENERAL DEL DOCUMENTO	5
DESCRIPCIÓN GENERAL	6
2.1 PERSPECTIVA DE PRODUCTO.....	6
<i>Interfaz De Usuario</i>	<i>6</i>
2.2 FUNCIONES DE PRODUCTO	8
2.3 CARACTERÍSTICAS DE LOS USUARIOS	10
2.4 RESTRICCIONES.....	10
REQUISITOS ESPECÍFICOS	11
3.1 INTERFACES EXTERNOS.....	11
3.2 FUNCIONES.....	12
3.3 REQUISITOS DE RENDIMIENTO	13
3.4 REQUERIMIENTOS LÓGICOS DE LA BASE DE DATOS.....	65
3.5 RESTRICCIONES DE DISEÑO.....	66
3.6 ATRIBUTOS DEL SISTEMA SOFTWARE.....	66
APÉNDICES.....	67

Introducción

1.1 Propósito

El objeto de este documento es especificar los requisitos del proyecto de Modelado Software (SRS) de manera que se detallarán las funcionalidades de la aplicación y cada una de sus entidades. También se mostrarán diagramas de actividades para aclarar gráficamente el comportamiento de todas las funcionalidades, así como otros detalles importantes a fin de clarificar rigurosamente la especificación software del proyecto. En esta especificación de requisitos, trataremos las características a desarrollar en la asignatura de Modelado Software.

Esto no sólo ayudará al equipo de desarrollo a comenzar su elaboración, sino que permitirá al cliente (como viene a ser el profesor de la asignatura) a comprobar que la aplicación está en concordancia con sus intereses y condiciones.

1.2 Alcance

El proyecto, de nombre “Kernel Panic”, está formado por dos partes. Primero, tiene como objetivo ayudar a los empleados de los restaurantes a gestionar las facturas y los menús de los clientes. Esto contribuirá a abaratar costes y agilizará los tiempos del restaurante. En segunda parte, se encargará de ayudar con la gestión de un hotel con el mismo objetivo de agilizar cada uno de sus procesos.

1.3 Definiciones, Acrónimos & Abreviaturas

En este punto se proporcionan las definiciones de todos los términos necesarios para comprender el contenido del documento.

CONCEPTO	DEFINICIÓN / ACRÓNIMO / ABREVIATURA
SRS	Software Requeriments Specification
ID	Identificador
JRE	Java Runtime Enviroment
JPA	Java Persistance API
IEEE	Institute of Electrical and Electronic Engineers
Interfaz	Zona de comunicación o acción de un sistema con otro
Stock	Conjunto de mercancías o productos que se tienen almacenados en espera de su venta o comercialización
String	Secuencia de caracteres
Autodescriptivo	Que su propio nombre describe su funcionalidad
Crear	Sinónimo de dar de alta
Eliminar	Sinónimo de dar de baja
UML	Unified Modeling Language
Empleado	El usuario final del sistema del restaurante
Factura	Conjunto de Menús asignados a una mesa con un precio total.
Menú	Conjunto de productos
Producto	Conjunto de ingredientes
Mesa	Lugar físico donde se sitúan los clientes.
Ingrediente	Componentes primos que conforman los productos.
Recepcionista	Personal del hotel encargado de la gestión de las reservas
Reserva	Entidad que relaciona un cliente del hotel con un recepcionista y al que se le asignan ciertos servicios y habitaciones
Habitación	Espacio físico del hotel en el que se hospedan los clientes
Servicio	Privilegios que pueden contratar los clientes con su reserva
Cliente	El usuario final del sistema del hotel
Turno	Horario de comidas o desayunos asignados a un cliente

1.4 Referencias

En esta subsección se mostrará una lista completa de todos los documentos referenciados en la SRS:

- IEEE Std 830-1998. *IEEE Recommended Practice for Software Requirements Specifications*.
- Material teórico publicado en el Campus Virtual de la asignatura de IS1
- Modelo SRS:
 - <https://versiones.fdi.ucm.es/svn/MS/2223E/kernelpanic/mod>
- Descripción de las secciones:
 - <https://www.fdi.ucm.es/profesor/gmendez/docs/is0809/ieee830.pdf>
- Definiciones:
 - <https://languages.oup.com/google-dictionary-es/>

1.5 Visión General Del Documento

Este documento se apoya en el estándar IEEE para especificaciones y requisitos. Consta de tres partes, una introducción, una descripción general y los requerimientos específicos.

La primera parte describe brevemente la propia SRS y consta de varias partes: propósito, alcance, definiciones, referencias y la visión general del documento.

La descripción general describe los factores generales que afectan al producto y sus requisitos, no define requisitos específicos, proporciona un contexto para estos requisitos y los hace más comprensibles. La descripción general a su vez se subdivide en perspectiva de producto, funciones de producto, características de usuario, restricciones y supuestos y dependencias.

La última parte contiene todos los requerimientos software al detalle incluyendo los diagramas de uso. Los puntos por tratar en esta sección son: interfaces externas, funciones, requisitos de rendimiento, modelo de dominio, restricciones de diseño y atributo de sistema software.

Descripción General

2.1 Perspectiva De Producto

Con el avance de la ingeniería del software se han ido desarrollando varias aplicaciones orientadas a la gestión de recursos y organización distintos tipos de comercios. La mayoría de las tiendas, por ejemplo, utilizan aplicaciones para llevar el stock o los horarios de los empleados. Al igual, los restaurantes y hoteles también se están modernizando. Donde antes se utilizaban pizarras para mostrar el menú ahora se usan pantallas, y donde antes se usaba papel para apuntar la carta de los clientes o las habitaciones y reservas de los huéspedes, ahora se seleccionan en aplicaciones. De aquí surge el proyecto Kernel Panic.

Interfaz De Usuario

Al iniciar el programa nos encontramos con dos botones:



Si presionamos RESTAURANTE, nos lleva a la ventana principal dedicada al restaurante.

En esta pantalla inicial de la interfaz de usuario de RESTAURANTE nos encontramos con un menú que consta de tres botones:



Si seleccionamos el botón de Mesa se mostrará los botones de crear, editar, eliminar, mostrar una y mostrar todas. Pulsar en el botón de crear, por ejemplo, te da la opción de introducir el número de mesa, y especificar su capacidad. Acto seguido se podrá dar a un botón para confirmar los cambios y otro para cancelar. Mostrar Una dará la opción de mostrar todos los datos en una tabla al introducir un id.

Si seleccionamos el botón de Comida y mesas se abrirá un submenú como el siguiente:



Ingredientes mostrará los mismos botones que para la mesa. Estos botones son ya autodescriptivos.

Productos e Ingredientes mostrará una pantalla muy similar a la de Mesas, pero con dos nuevos botones, uno para añadir un ingrediente y otro para quitarlo.

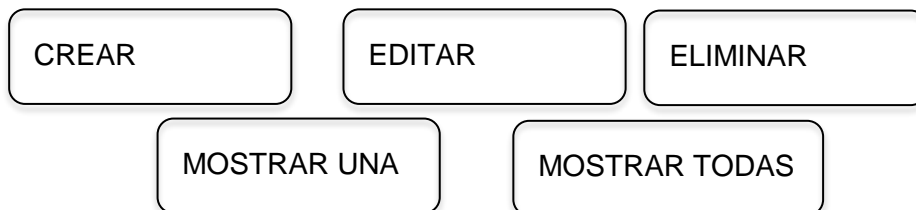
Menús mostrará una pantalla muy similar a la de productos.

Si seleccionamos el botón de Empleados aparecerá una vista también muy similar a la de los menús o la de los productos.

Sin embargo. Si en la vista principal presionamos HOTEL, se nos abre otra ventana, dedicada al hotel.



Si seleccionamos el botón RESERVAS, nos aparece una ventana con los siguientes botones.



Todas las pantallas tendrán dos botones con la función de volver al menú que le antecede y con la función de volver a la pantalla en la que estaba el usuario anteriormente.

2.2 Funciones De Producto

Las funciones del producto tienen como receptor final tanto a los empleados del restaurante como a los del hotel.

- **Mesa**
 - Crear mesa
 - Eliminar mesa
 - Editar mesa
 - Mostrar una o todas las mesas
- **Menú**
 - Crear menú
 - Eliminar menú
 - Editar menú
 - Mostrar uno o todos los menús
 - Funciones menú-producto: añadir y borrar producto
 - Mostrar menú por rango de precio.
- **Producto**
 - Crear producto
 - Eliminar producto
 - Editar producto
 - Mostrar uno o todos los productos
 - Funciones producto-ingrediente: añadir y borrar ingrediente
- **Ingredientes**
 - Crear ingrediente
 - Eliminar ingrediente
 - Editar ingrediente
 - Mostrar uno o todos los ingredientes
- **Facturas**
 - Abrir factura
 - Cerrar factura
 - Devolver menú
 - Editar factura
 - Atributos de la factura
 - Importe de menús
 - Cantidad de menús
 - Fecha de cobro
 - Eliminar factura
 - Mostrar una o todas las facturas
- **Empleados**
 - Crear empleado
 - Eliminar empleado
 - Editar empleado
 - Mostrar uno o todos los empleados
 - Mostrar empleados despedidos por rango salarial.
- **Recepcionista**
 - Crear recepcionista

- Eliminar recepcionista
- Editar recepcionista
- Mostrar por id
- Mostrar por nombre
- Mostrar todos
- **Reserva**
 - Crear reserva
 - Eliminar reserva
 - Editar reserva
 - Mostrar una o todas las reservas
 - Funciones reserva-servicio
 - Vincular reserva-servicio
 - Editar reserva-servicio
 - Desvincular reserva-servicio
 - Mostrar las reservas de un servicio
 - Mostrar servicios de una reserva
 - Mostrar todos
- **Habitación**
 - Crear habitación
 - Eliminar habitación
 - Editar habitación
 - Mostrar por id
 - Mostrar por número
 - Mostrar todas
 - Funciones habitación-reserva
 - Vincular habitación a reserva
 - Desvincular habitación de reserva
 - Mostrar habitaciones de una reserva
 - Mostrar reservas de una habitación
 - Mostrar todos
- **Servicio**
 - Crear servicio
 - Eliminar servicio
 - Editar servicio
 - Mostrar por id
 - Mostrar por tipo de servicio
 - Mostrar todos
- **Cliente**
 - Crear cliente
 - Eliminar cliente
 - Editar cliente
 - Mostrar uno o todos los clientes
 - Calcular precio de turnos
- **Turno**
 - Crear turno
 - Eliminar turno
 - Editar turno

- Mostrar uno o todos los turnos
- Funciones turno-cliente:
 - Crear turno-cliente
 - Editar turno-cliente
 - Eliminar turno-cliente
 - Mostrar turnos de un cliente
 - Mostrar clientes de un turno

2.3 Características De Los usuarios

Esta aplicación está destinada tanto a los empleados que tienen trato con los clientes y a los cocineros del restaurante, como a los recepcionistas y sirvientes del hotel. La aplicación es muy intuitiva y puede ser comprendida por cualquier usuario sin importar su historial.

Como lo que se busca tanto en un restaurante como en un hotel es la eficiencia y rapidez del trabajo, Kernel Panic ha creado este programa sencillo de manejar destinado a cualquier tipo de usuario sin tener en cuenta su formación.

2.4 Restricciones

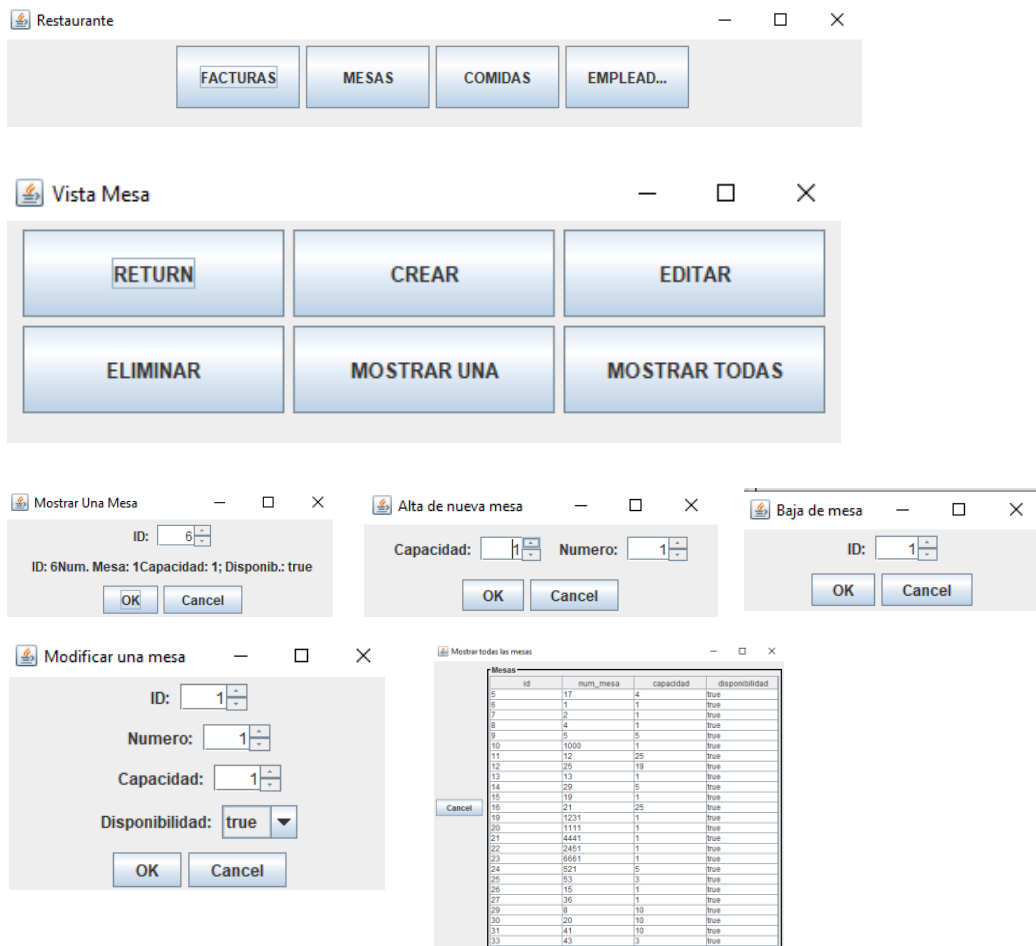
Esta subsección describe aquellas limitaciones que se imponen sobre los desarrolladores del producto.

- Políticas de la empresa: el programa tiene que poder ser usado por cualquier tipo de empleado del restaurante y del hotel.
- Limitaciones hardware: solo se puede acceder al programa a través de un solo terminal.
- Lenguaje de programación: será programado en Java.
- Limitaciones software: el ordenador debe tener instalado JRE 8.
- Limitaciones de tiempo: el tiempo de ejecución del programa tiene que ser el menor posible.

Requisitos Específicos

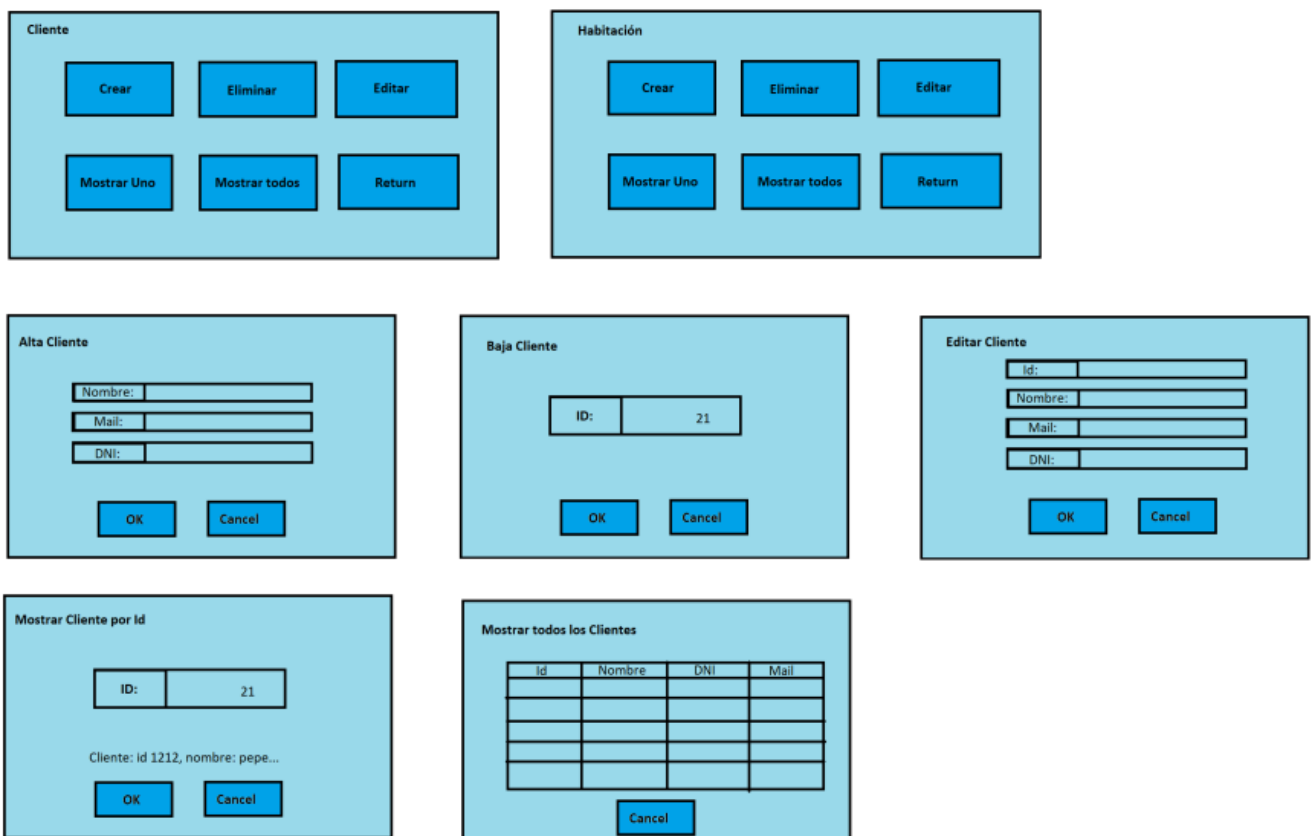
3.1 Interfaces Externas

A continuación se muestran algunas de las capturas de la interfaz de usuario:





Y de Hotel:

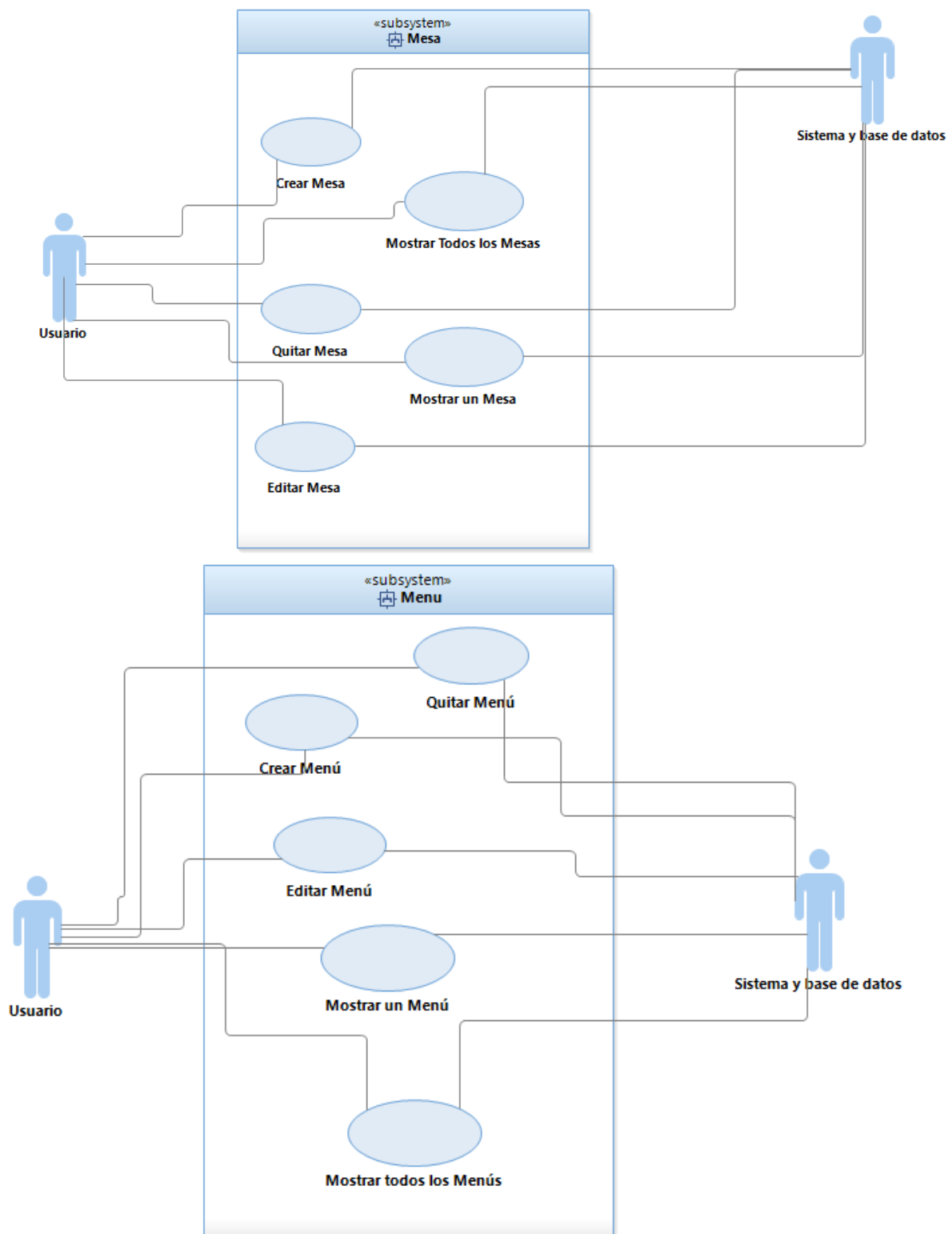


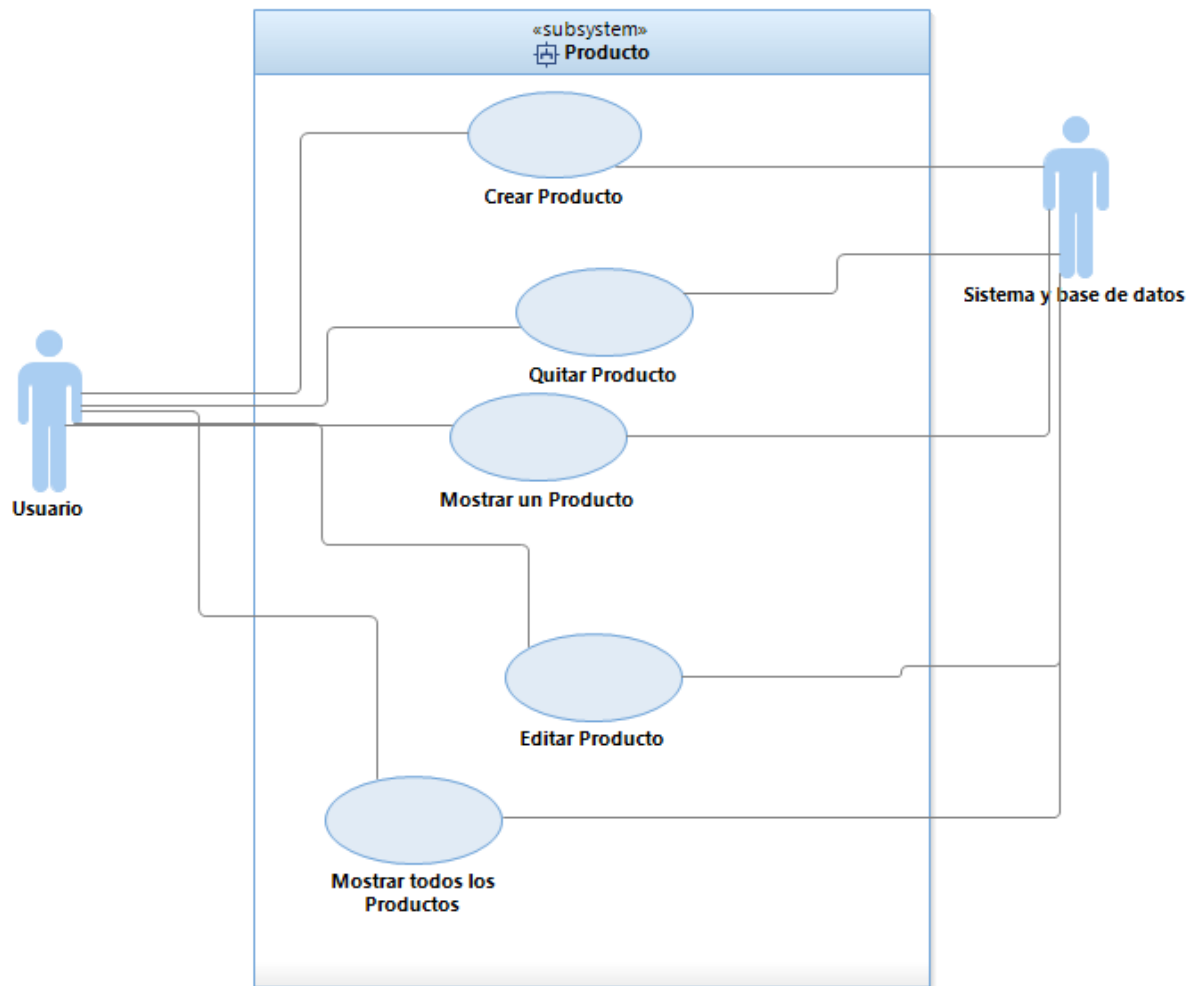
3.2 Funciones

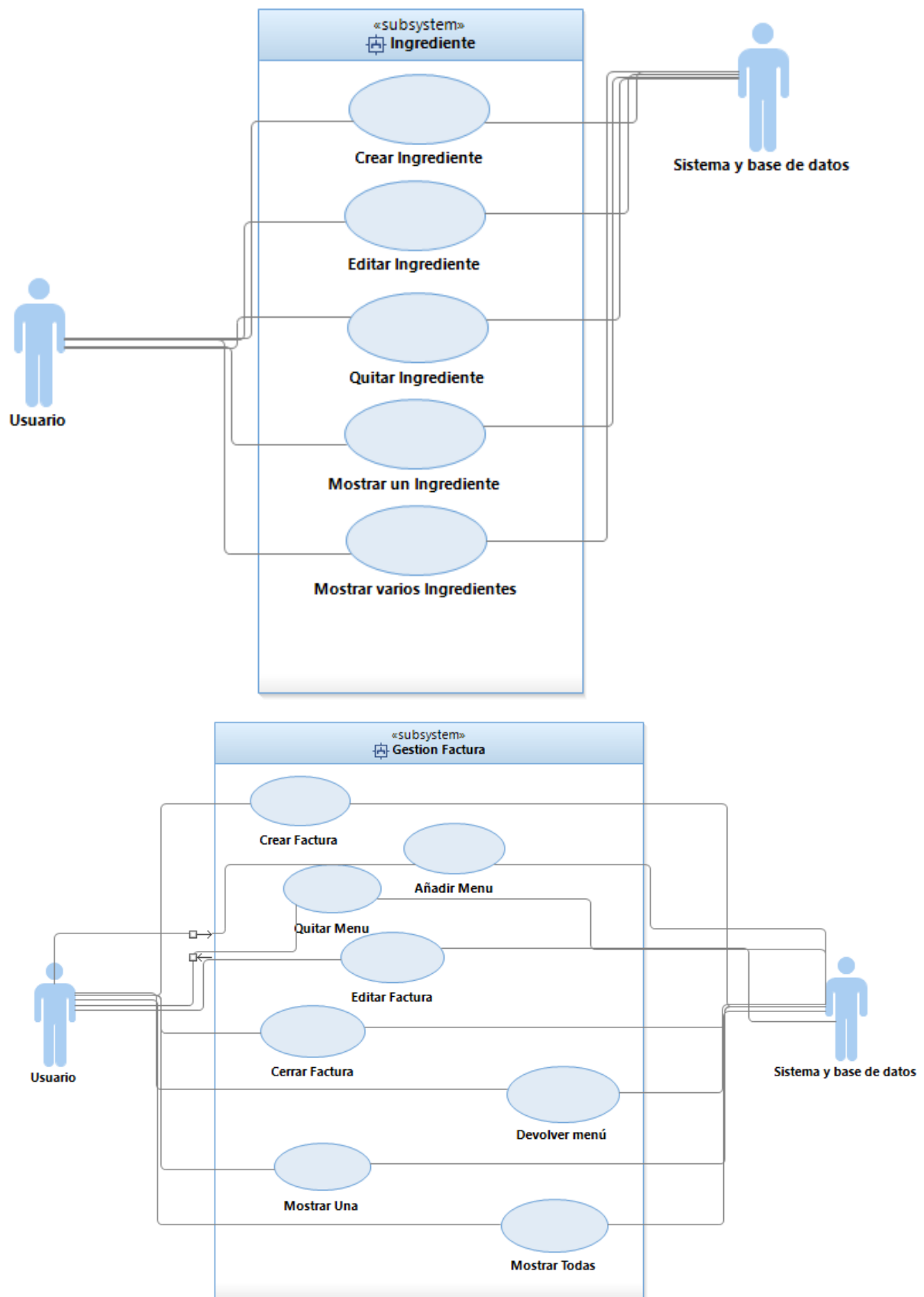
Deberá especificar todas aquellas acciones fundamentales que tiene que llevar a cabo el software durante la aceptación y procesamiento de la entrada y durante el procesamiento y generación de la salida.

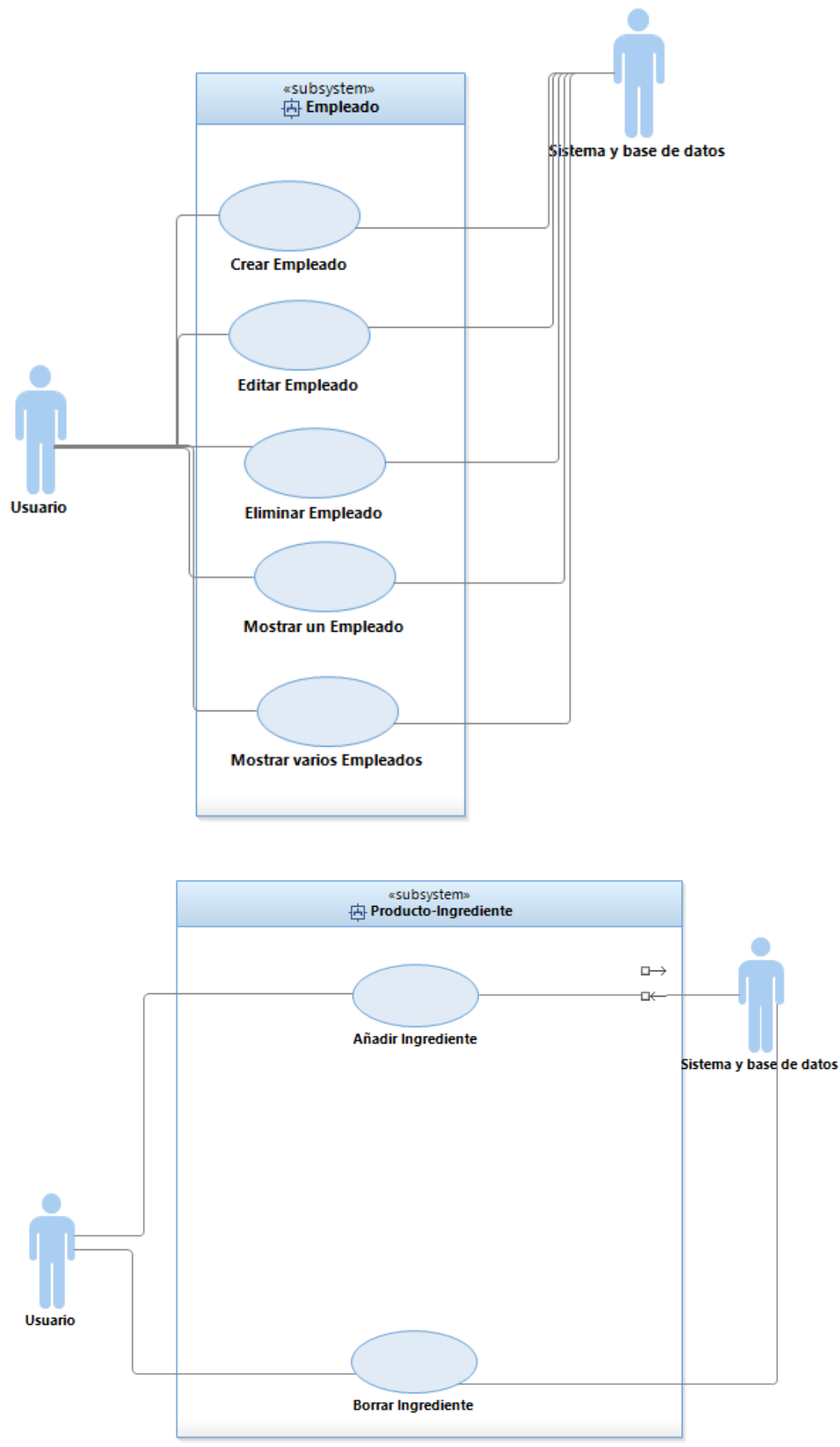
Diagramas De Casos De Uso

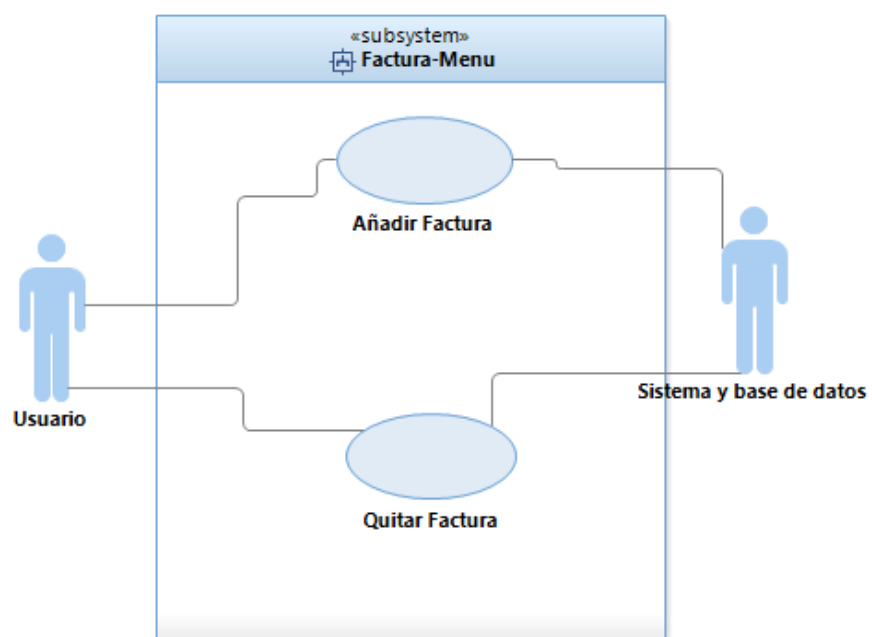
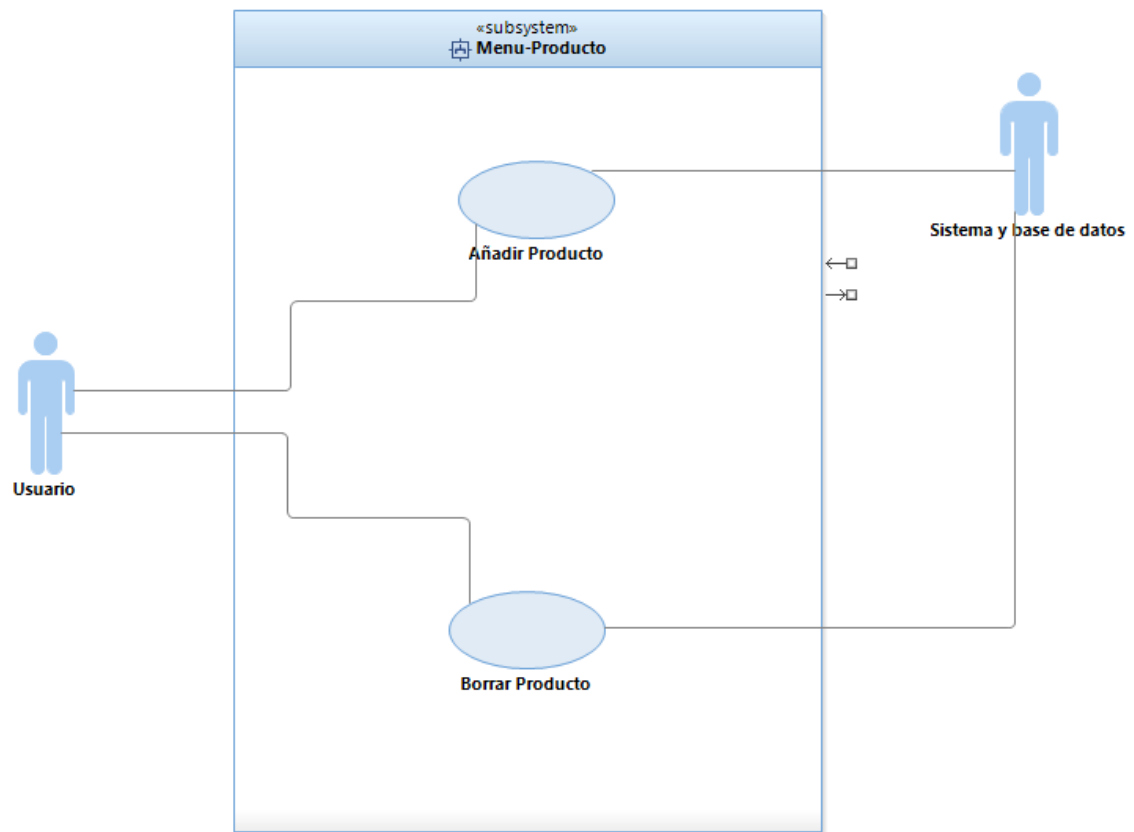
Se refiere a la forma de diagrama de comportamiento UML.











Casos De Uso

Descripción de una acción o actividad que se deberá realizar para poder llevar a cabo los distintos procesos.

CASO 1: CREAR MESA	
Descrita por	Alberto Ramos
Prioridad	Alta
Estabilidad	Alta
Descripción	Creación de una nueva mesa en el restaurante
Actor	Usuario
Precondición	No hay identificadores ni números de mesa repetidos en la base de datos de mesa
Entrada	Capacidad de la mesa, identificador y número
Salida	Identificador de la mesa
Origen	Interfaz de usuario de mesas
Destino	Sistema
Necesita	Base de datos de mesas
Postcondición	No hay identificadores ni números de mesa repetidos en la base de datos de mesa, se inserta una nueva mesa con los datos introducidos
Acciones	Comprobar que los datos son sintácticamente correctos y que no hay otra mesa con el mismo número. En caso positivo, crear la mesa (o la activa si estaba inactiva) y devolver el identificador

CASO 2: ELIMINAR MESA	
Descrita por	Alberto Ramos
Prioridad	Media
Estabilidad	Alta
Descripción	Eliminación de una de las mesas previamente existentes en el restaurante
Actor	Usuario
Precondición	No hay identificadores ni números de mesa repetidos en la base de datos de mesa
Entrada	Identificador de la mesa que se desea eliminar
Salida	Identificador de la mesa eliminada
Origen	Interfaz de usuario de mesas
Destino	Sistema
Necesita	Base de datos de mesas
Postcondición	No hay identificadores ni números de mesa repetidos en la base de datos de mesa. Se marca como inactiva la mesa con el identificador introducido
Acciones	Comprobar que la mesa que se desea eliminar existe. En caso positivo, marcar la mesa como no activa y devolver el identificador

CASO 3: EDITAR MESA	
Descrita por	Alberto Ramos
Prioridad	Media
Estabilidad	Alta
Descripción	Editar los atributos de una mesa, excepto el identificador
Actor	Usuario
Precondición	No hay identificadores ni números de mesa repetidos en la base de datos de mesa
Entrada	El identificador de la mesa a editar, luego los datos que se desean modificar sobre la mesa (excepto identificador)
Salida	Identificador de la mesa
Origen	Interfaz de usuario de mesas
Destino	Sistema
Necesita	Base de datos de mesas
Postcondición	No hay identificadores ni números de mesa repetidos en la base de datos de mesa, se modifica la mesa con el identificador introducido
Acciones	Se busca la mesa a editar y se verifica que los cambios realizados son sintácticamente correctos. Si son correctos y está activo, se realizan los cambios en la base de datos. El usuario puede cancelar en cualquier momento.

CASO 4: MOSTRAR UNA MESA	
Descrita por	Alberto Ramos
Prioridad	Alta
Estabilidad	Alta
Descripción	Mostrar una mesa con sus atributos
Actor	Usuario
Precondición	No hay identificadores ni números de mesa repetidos en la base de datos de mesa
Entrada	Identificador de la mesa que se desea mostrar
Salida	Atributos de la mesa pedida
Origen	Interfaz de usuario de mesas
Destino	Sistema
Necesita	Base de datos de mesas
Postcondición	No hay identificadores ni números de mesa repetidos en la base de datos de mesa, se muestra la mesa con el identificador introducido
Acciones	Busca el identificador de la mesa en la base de datos. Si está, se muestra

CASO 5: MOSTRAR TODAS LAS MESAS	
Descrita por	Alberto Ramos
Prioridad	Alta
Estabilidad	Alta
Descripción	Mostrar todas las mesas de la base de datos
Actor	Usuario
Precondición	No hay identificadores ni números de mesa repetidos en la base de datos de mesa
Entrada	Ninguna entrada
Salida	Atributos de todas las mesas de la base de datos
Origen	Interfaz de usuario de mesas
Destino	Sistema
Necesita	Base de datos de mesas
Postcondición	No hay identificadores ni números de mesa repetidos en la base de datos de mesa, se muestran todas las mesas con sus atributos
Acciones	Se muestran todas las mesas activas de la base de datos

CASO 6: CREAR MENÚ	
Descrita por	Alberto Ramos
Prioridad	Alta
Estabilidad	Alta
Descripción	Creación de un nuevo menú
Actor	Usuario
Precondición	No hay identificadores ni nombres de menú repetidos en la base de datos de menú
Entrada	Nombre del menú, identificador, precio y stock
Salida	Confirmación de creación de un nuevo menú, identificador del menú
Origen	Interfaz de usuario de menú
Destino	Sistema
Necesita	Base de datos de menú
Postcondición	No hay identificadores ni nombres de menú repetidos en la base de datos de menú, se inserta un nuevo menú con los datos introducidos
Acciones	Comprobar que los datos son sintácticamente correctos y que no hay otro menú con el mismo nombre. En caso positivo, crear el menú (o lo activa si estaba inactivo) y devolver el identificador

CASO 7: ELIMINAR MENÚ	
Descrita por	Alberto Ramos
Prioridad	Media
Estabilidad	Alta
Descripción	Eliminación de uno de los menús previamente existentes
Actor	Usuario
Precondición	No hay identificadores ni nombres de menú repetidos en la base de datos de menú
Entrada	Identificador del menú que se desea eliminar
Salida	Identificador del menú eliminado
Origen	Interfaz de usuario de menús
Destino	Sistema
Necesita	Base de datos de menú
Postcondición	No hay identificadores ni nombres de menú repetidos en la base de datos de menú. Se marca el menú con el identificador introducido como inactivo
Acciones	Comprobar que el menú que se desea eliminar existe. En caso positivo, marcar el menú como no activo y devolver el identificador

CASO 8: EDITAR MENÚ	
Descrita por	Alberto Ramos
Prioridad	Media
Estabilidad	Alta
Descripción	Editar el stock y precio de un menú
Actor	Usuario
Precondición	No hay identificadores ni nombres de menú repetidos en la base de datos de menú
Entrada	El identificador del menú a editar, luego el nuevo stock y/o precio del menú
Salida	Identificador del menú
Origen	Interfaz de usuario de menús
Destino	Sistema
Necesita	Base de datos de menú
Postcondición	No hay identificadores ni nombres de menú repetidos en la base de datos de menú, se modifica el menú con el identificador introducido
Acciones	Se busca el menú a editar y se verifican los cambios realizados. Si son correctos, se realizan los cambios en la base de datos. El usuario puede cancelar en cualquier momento.

CASO 9: MOSTRAR UN MENÚ	
Descrita por	Alberto Ramos
Prioridad	Alta
Estabilidad	Alta
Descripción	Mostrar un menú con sus atributos y productos
Actor	Usuario
Precondición	No hay identificadores ni nombres de menú repetidos en la base de datos de menú
Entrada	Identificador del menú que se desea mostrar
Salida	Atributos del menú pedido
Origen	Interfaz de usuario de menús
Destino	Sistema
Necesita	Base de datos de menú
Postcondición	No hay identificadores ni nombres de menú repetidos en la base de datos de menú, se muestra el menú con el identificador introducido
Acciones	Busca el identificador de la mesa en la base de datos. Si está, se muestra

CASO 10: MOSTRAR TODOS LOS MENÚS	
Descrita por	Alberto Ramos
Prioridad	Alta
Estabilidad	Alta
Descripción	Mostrar todos los menús de la base de datos
Actor	Usuario
Precondición	No hay identificadores ni nombres de menú repetidos en la base de datos de menú
Entrada	Sin entrada
Salida	Atributos de todos los menús de la base de datos
Origen	Interfaz de usuario de menús
Destino	Sistema
Necesita	Base de datos de menú
Postcondición	No hay identificadores ni nombres de menú repetidos en la base de datos de menú, se muestran todos los menús con sus atributos
Acciones	Se muestran todas las mesas

CASO 11: CREAR PRODUCTO	
Descrita por	Juan Embid
Prioridad	Media baja
Estabilidad	Alta
Descripción	Creación de un nuevo producto, con su nombre
Actor	Usuario y sistema
Precondición	No hay identificadores ni nombres de producto repetidos en la base de datos de producto
Entrada	Nombre del producto
Salida	El identificacor del producto
Origen	Interfaz de usuario
Destino	Sistema
Necesita	Acceso a la base de datos con todos los productos
Postcondición	No hay identificadores ni nombres de producto repetidos en la base de datos de producto, se inserta un nuevo producto con los datos introducidos
Acciones	Comprobar que los datos son sintácticamente correctos y que no hay otro producto con el mismo identificador y nombre. En caso positivo, crear el producto y devolver el identificador

CASO 12: ELIMINA PRODUCTO	
Descrita por	Juan Embid
Prioridad	Media baja
Estabilidad	Alta
Descripción	Eliminación de uno de los productos previamente existentes
Actor	Usuario
Precondición	No hay identificadores ni nombres de producto repetidos en la base de datos de producto
Entrada	Identificador del producto que se desea eliminar
Salida	El identificador del producto
Origen	Interfaz de usuario
Destino	Sistema
Necesita	Acceso a la base de datos con todos los productos
Postcondición	No hay identificadores ni nombres de producto repetidos en la base de datos de mesa. Se marca como inactivo el producto con el identificador introducido
Acciones	Comprobar que el producto que se desea eliminar existe, que no hay más de un producto con ese identificador y que no está vinculado a un menú. En caso positivo, marcar el producto como no activo y devolver el identificador

CASO 13: EDITA PRODUCTO	
Descrita por	Juan Embid
Prioridad	Media baja
Estabilidad	Alta
Descripción	Editar los atributos del producto, excepto el identificador
Actor	Usuario
Precondición	No hay identificadores ni nombres de producto repetidos en la base de datos de productos
Entrada	El identificador del producto a editar, luego los ingredientes y datos que se desean modificar sobre el producto
Salida	El identificador del producto
Origen	Interfaz de usuario
Destino	Sistema
Necesita	Acceso a los ingredientes del producto
Postcondición	No hay identificadores ni nombres de producto repetidos en la base de datos de producto, se modifica el producto con el identificador introducido
Acciones	Se busca el producto a editar y se verifica que los cambios realizados son sintácticamente correctos. Si son correctos, se realizan los cambios en la base de datos. El usuario puede cancelar en cualquier momento

CASO 14: MOSTRAR UN PRODUCTO	
Descrita por	Juan Embid
Prioridad	Media baja
Estabilidad	Alta
Descripción	Mostrar un producto con sus atributos e ingredientes
Actor	Usuario
Precondición	No hay identificadores ni nombres de producto repetidos en la base de datos de producto
Entrada	Identificador del producto que se desea mostrar
Salida	Atributos e ingredientes que contiene la mesa pedida
Origen	Interfaz de usuario
Destino	Sistema
Necesita	Acceso a los ingredientes del producto
Postcondición	No hay identificadores ni números de mesa repetidos en la base de datos de producto, se muestra el producto con el identificador introducido
Acciones	Busca el identificador del producto en la base de datos. Si está, se muestra

CASO 15: MOSTRAR TODOS LOS PRODUCTOS	
Descrita por	Juan Embid
Prioridad	Media baja
Estabilidad	Alta
Descripción	Mostrar todos los productos de la base de datos
Actor	Usuario
Precondición	No hay identificadores ni nombres de producto repetidos en la base de datos de producto
Entrada	No hay entrada
Salida	Atributos e ingredientes que contienen todos los productos de la base de datos
Origen	Interfaz de usuario
Destino	Sistema
Necesita	Acceso a los ingredientes del producto
Postcondición	No hay identificadores ni números de mesa repetidos en la base de datos de producto, se muestra el producto con el identificador introducido
Acciones	Se muestran todos los productos activos de la base de datos

CASO 16: CREAR INGREDIENTE	
Descrita por	Juan Montero Gómez
Prioridad	Alta
Estabilidad	Alta
Descripción	Creación de un nuevo ingrediente
Actor	Usuario
Precondición	No hay identificadores ni nombres de ingrediente repetidos en la base de datos de ingrediente
Entrada	Alérgeno y nombre
Salida	Confirmación de creación de nuevo ingrediente, identificador del ingrediente
Origen	Interfaz de usuario
Destino	Sistema
Necesita	Base de datos de ingrediente
Postcondición	No hay identificadores repetidos en la base de datos de ingrediente, se inserta un nuevo ingrediente
Acciones:	Comprobar que los datos son sintácticamente correctos y que no hay dos productos ni dos menús con el mismo ID. En caso positivo, se da de alta el producto dentro del menú

CASO 17: ELIMINAR INGREDIENTE	
Descrita por	Juan Montero Gómez
Prioridad	Media
Estabilidad	Alta
Descripción	Eliminación de uno de los ingredientes previamente existentes en el restaurante
Actor	Usuario
Precondición	No hay identificadores ni nombres repetidos en la base de datos de ingrediente
Entrada	Identificador del ingrediente que se desea eliminar
Salida	Confirmación de la eliminación del ingrediente
Origen	Interfaz de usuario
Destino	Sistema
Necesita	Base de datos de ingrediente
Postcondición	No hay identificadores repetidos en la base de datos de ingrediente, se desactiva el ingrediente con el identificador introducido
Acciones:	Buscar ID de entrada en la base de datos..Comprueba que el ingrediente no esté vinculado con un producto. En caso positivo, desactiva el ingrediente con el ID que coincida.

CASO 18: EDITAR INGREDIENTE	
Descrita por	Juan Montero Gómez
Prioridad	Media
Estabilidad	Alta
Descripción	Editar los atributos de un ingrediente, excepto el identificador
Actor	Usuario
Precondición	No hay identificadores repetidos en la base de datos de ingrediente
Entrada	El identificador del ingrediente a editar, luego los datos que se desean modificar sobre el ingrediente (excepto identificador)
Salida	Confirmación de la modificación del ingrediente
Origen	Interfaz de usuario
Destino	Sistema
Necesita	Base de datos de ingrediente
Postcondición si éxito	No hay identificadores repetidos en la base de datos de ingrediente se modifica el ingrediente con el identificador introducido
Acciones:	Se verifica que los cambios realizados son sintácticamente correctos y el identificador se corresponde con un ingrediente existente y activo. Si son correctos y está activo, se realizan los cambios en la base de datos. El usuario puede cancelar en cualquier momento.

CASO 19: MOSTRAR UN INGREDIENTE	
Descrita por	Juan Montero Gómez
Prioridad	Alta
Estabilidad	Alta
Descripción	Mostrar un ingrediente con sus atributos
Actor	Usuario
Precondición	No hay identificadores repetidos en la base de datos de ingrediente
Entrada	Identificador del ingrediente que se desea mostrar
Salida	Atributos del ingrediente pedida
Origen	Interfaz de usuario
Destino	Sistema
Necesita	Base de datos de ingrediente
Postcondición	No hay identificadores repetidos en la base de datos de ingrediente se muestra el ingrediente con el identificador introducido
Acciones:	Busca el identificador del ingrediente en la base de datos. Si existe y está activo, se muestra

CASO 20: MOSTRAR TODOS LOS INGREDIENTES	
Descrita por	Juan Montero Gómez
Prioridad	Alta
Estabilidad	Alta
Descripción	Mostrar todos los ingredientes con sus atributos
Actor	Usuario
Precondición	No hay identificadores repetidos en la base de datos de ingrediente
Entrada	No hay entrada
Salida	Atributos de todos los ingredientes de la base de datos
Origen	Interfaz de usuario
Destino	Sistema
Necesita	Base de datos de ingrediente
Postcondición	No hay identificadores repetidos en la base de datos de ingrediente Se muestran todos los ingredientes con sus atributos
Acciones:	Se muestran todos los ingredientes activos

CASO 21: Abrir factura	
Descrita por	Fan Ye y Juan Montero Gómez
Prioridad	Alta
Estabilidad	Alta
Descripción	Adición de una nueva factura a la base de datos de facturas
Actor	Usuario
Precondición	No hay IDs repetidos en la base de datos de facturas. Tampoco hay más de una factura abierta con el mismo ID de mesa
Entrada	El ID de la mesa y el ID del empleado
Salida	ID de la nueva factura
Origen	Interfaz de usuario de facturas
Destino	Sistema
Necesita	Base de datos de facturas, de mesas y de empleados
Postcondición	No hay IDs repetidos en la base de datos de facturas y no hay más de una factura activa con el mismo ID de mesa. Se añade una nueva factura a la base de datos de factura
Acciones	Comprobar que los datos son sintácticamente correctos y que no hay ningún otra factura con el mismo ID de mesa abierta. Abre la factura guardando los datos pero sin modificar el stock.

CASO 23: Cerrar factura	
Descrita por	Fan Ye y Juan Montero Gómez
Prioridad	Alta
Estabilidad	Alta
Descripción	Guarda la hora de cobro en la factura y se da de baja
Actor	Usuario
Precondición	No hay identificadores repetidos en la base de datos y los menús de la factura existen y su cantidad no sobrepasa el stock y los id de empleado y mesa vinculados sean correctos y existan.
Entrada	ID empleado, ID mesa, menús vinculados.
Salida	ID de la factura con la hora de cobro e importe, dada de baja.
Origen	Interfaz de usuario de facturas
Destino	Sistema
Necesita	Base de datos de facturas
Postcondición	No hay identificadores repetidos en la base de datos. Se da de baja la factura con una hora de cobro
Acciones	Comprobar que la factura que se desea cerrar está abierta. En caso positivo, se guarda la hora de cobro en la factura y se cierra, modificando el stock de los menús.

CASO 24: Devolver menú de una factura	
Descrita por	Fan Ye y Juan Montero Gómez
Prioridad	Media baja
Estabilidad	Alta
Descripción	Da de alta una factura que se ha dado de baja anteriormente para poder realizar cambios
Actor	Usuario
Precondición	No hay identificadores repetidos en la base de datos
Entrada	ID de la factura, ID del menú, cantidad del menú
Salida	La factura con el nuevo importe, y el menú con el nuevo stock
Origen	Interfaz de usuario de facturas
Destino	Sistema
Necesita	Base de datos de facturas
Postcondición	No hay identificadores repetidos en la base de datos. Se da de alta la factura seleccionada
Acciones	Comprobar validez sintáctica de los datos. Comprobar que la factura está cerrada y que el menú está en la factura. Si todo correcto se resta el mínimo entre la cantidad especificada a devolver y la que está en la factura. Esta se mantiene cerrada y se modifica el stock.

CASO 25: Editar factura	
Descrita por	Fan Ye y Juan Montero Gómez
Prioridad	Media
Estabilidad	Alta
Descripción	Editar la información de una factura, excepto identificador
Actor	Usuario
Precondición	No hay identificadores repetidos en la base de datos
Entrada	ID de la factura, ID de la mesa si quiere cambiar la mesa asignada o ID del empleado si quiere cambiar el empleado asignado
Salida	La factura actualizada
Origen	Interfaz de usuario de facturas
Destino	Sistema
Necesita	Base de datos de facturas, de mesas y de empleados
Postcondición	No hay identificadores repetidos en la base de datos. Se actualiza la factura
Acciones	Se busca la factura a editar y se verifica que los cambios realizados son sintácticamente correctos. En caso positivo, se realiza los cambios que se han introducido

CASO 26: Mostrar una factura	
Descrita por	Fan Ye y Juan Montero Gómez
Prioridad	Alta
Estabilidad	Alta
Descripción	Mostrar una factura con su información y atributos
Actor	Usuario
Precondición	No hay identificadores repetidos en la base de datos
Entrada	El ID de la factura está activa
Salida	La factura seleccionada
Origen	Interfaz de usuario de facturas
Destino	Sistema
Necesita	Base de datos de facturas
Postcondición	No hay identificadores repetidos en la base de datos. Se muestra la factura seleccionada
Acciones	Busca el identificador de la factura en la base de datos. Si está, se muestra

Caso 27: Mostrar todas las facturas	
Descrita por	Fan Ye y Juan Montero Gómez
Prioridad	Media Baja
Estabilidad	Alta
Descripción	Muestra una lista de todas las facturas, primero las que están dadas de alta y luego las que ya están dadas de baja
Actor	Usuario
Precondición	No hay identificadores repetidos en la base de datos de la factura
Entrada	No hay entrada
Salida	El listado de facturas
Origen	Interfaz de usuario de facturas
Destino	Sistema
Necesita	Base de datos de facturas
Postcondición	No hay identificadores repetidos en la base de datos. Se muestran las facturas
Acciones	Se comprueba que los datos son sintácticamente correctos. En caso positivo, se muestra la lista de facturas, las que están abiertas primero y las que están cerradas después.

Caso 27: Eliminar factura	
Descrita por	Juan Montero Gómez
Prioridad	Media Baja
Estabilidad	Alta
Descripción	Elimina una factura
Actor	Usuario
Precondición	No hay identificadores repetidos en la base de datos de la factura
Entrada	No hay entrada
Salida	El listado de facturas
Origen	Interfaz de usuario de facturas
Destino	Sistema
Necesita	Base de datos de facturas
Postcondición	No hay identificadores repetidos en la base de datos.
Acciones	Se comprueba que los datos son sintácticamente correctos. Se comprueba que una factura con ese ID existe. Si está cerrada, se devuelven los menús que estaban en la factura y después se elimina. Si está abierta solo se elimina la factura.

CASO 28: Añadir menú a factura	
Descrita por	Juan Montero Gómez
Prioridad	Alta
Estabilidad	Alta
Descripción	Añadir una cantidad de un menú a una factura
Actor	Usuario
Precondición	
Entrada	Id factura, Id menú, cantidad
Salida	Factura con cantidad de menú añadido
Origen	Interfaz de usuario
Destino	Sistema
Necesita	Base de datos de empleado
Postcondición	
Acciones:	Comprobar que los datos son sintácticamente correctos, comprobar que la factura existe y está abierta, comprobar que el menú existe y está activo, si no estaba previamente vinculado se añade el menú con la cantidad especificada a la cesta, si ya estaba previamente vinculado, simplemente se suma la cantidad a la ya existente.

CASO 28: Quitar menú de factura	
Descrita por	Juan Montero Gómez
Prioridad	Alta

Estabilidad	Alta
Descripción	Quitar una cantidad de un menú de una factura
Actor	Usuario
Precondición	
Entrada	Id factura, Id del menú y la cantidad a quitar
Salida	Factura con la cantidad del menú quitada
Origen	Interfaz de usuario
Destino	Sistema
Necesita	Base de datos de empleado
Postcondición	
Acciones:	Comprobar que los datos son sintácticamente correctos, comprobar que la factura existe y está abierta, comprobar que el <u>menú</u> existe y está activo, si todo correcto se quita el mínimo entre la cantidad solicitada y la que había en la factura.

CASO 28: CREAR EMPLEADO	
Descrita por	Juan Montero Gómez
Prioridad	Alta
Estabilidad	Alta
Descripción	Creación de un nuevo empleado, con su identificador, nombre, sueldo, DNI y horario. Se distinguen entre Camareros, que tienen un idioma, y Cocineros, que tienen un título
Actor	Usuario
Precondición	No hay identificadores repetidos en la base de datos de empleados
Entrada	Identificador, nombre, sueldo, DNI, horario, Camarero/Cocinero, idioma/título
Salida	Confirmación de creación de nuevo empleado, identificador del empleado
Origen	Interfaz de usuario
Destino	Sistema
Necesita	Base de datos de empleado
Postcondición	No hay identificadores repetidos en la base de datos de empleado, se inserta un nuevo empleado
Acciones:	Comprobar que los datos son sintácticamente correctos, comprobar que no hay otro empleado con el mismo DNI ya activo. En caso positivo, guardar empleado en la base de datos, devolver ID. Si el DNI ya existía y estaba desactivado, activarlo.

CASO 29: EDITAR EMPLEADO	
Descrita por	Juan Montero Gómez
Prioridad	Alta

Estabilidad	Alta
Descripción	Editar los atributos y/o tipo de empleado y sus atributos, excepto el identificador
Actor	Usuario
Precondición	No hay identificadores repetidos en la base de datos de empleado
Entrada	El identificador del empleado a editar, luego los datos que se desean modificar sobre el ingrediente (excepto identificador)
Salida	Confirmación de la modificación del empleado
Origen	Interfaz de usuario
Destino	Sistema
Necesita	Base de datos de empleado
Postcondición si éxito	No hay identificadores repetidos en la base de datos de empleado se modifica el empleado con el identificador introducido
Acciones:	Se verifica que los cambios realizados son sintácticamente correctos y el identificador se corresponde con un empleado existente y activo. Si son correctos y está activo, se realizan los cambios en la base de datos. El usuario puede cancelar en cualquier momento.

CASO 30: ELIMINAR EMPLEADO	
Descrita por	Juan Montero Gómez
Prioridad	Media
Estabilidad	Alta
Descripción	Eliminación de uno de los empleados previamente existentes en el restaurante
Actor	Usuario
Precondición	No hay identificadores repetidos en la base de datos de empleado
Entrada	Identificador del empleado que se desea eliminar
Salida	Confirmación de la eliminación del empleado
Origen	Interfaz de usuario
Destino	Sistema
Necesita	Base de datos de empleado
Postcondición si éxito	No hay identificadores repetidos en la base de datos de empleado, se elimina el empleado con el identificador introducido
Acciones:	Buscar ID de entrada en la base de datos. Si existe y está activo, desactiva el empleado con el ID que coincide.

CASO 31: MOSTRAR UN EMPLEADO	
Descrita por	Juan Montero Gómez
Prioridad	Alta
Estabilidad	Alta
Descripción	Mostrar un empleado con sus atributos
Actor	Usuario

Precondición	No hay identificadores repetidos en la base de datos de empleado
Entrada	Identificador del empleado que se desea mostrar
Salida	Atributos del empleado pedido
Origen	Interfaz de usuario
Destino	Sistema
Necesita	Base de datos de empleado
Postcondición si éxito	No hay identificadores repetidos en la base de datos de empleado se muestra el empleado con el identificador introducido
Acciones:	Busca el identificador del empleado en la base de datos. Si existe y está activo, se muestra

CASO 32: MOSTRAR TODOS LOS EMPLEADOS	
Descrita por	Juan Montero Gómez
Prioridad	Alta
Estabilidad	Alta
Descripción	Mostrar todos los empleados con sus atributos
Actor	Usuario
Precondición	No hay identificadores repetidos en la base de datos de empleado
Entrada	Petición para mostrar todos los empleados
Salida	Atributos de todos los ingredientes de la base de datos
Origen	Interfaz de usuario
Destino	Sistema
Necesita	Base de datos de empleado
Postcondición si éxito	No hay identificadores repetidos en la base de datos de empleado Se muestran todos los empleados con sus atributos
Acciones:	Se muestran todos los empleados activos

CASO 33: FUNCIONES MENÚ-PRODUCTO AÑADIR PRODUCTO

Descrita por	Juan Embid
Prioridad	Alta
Estabilidad	Alta
Descripción	Creación de un nuevo producto
Actor	Usuario
Precondición	No hay identificadores ni nombres repetidos y activos en la base de datos de menús ni en la de productos
Entrada	Identificador del nuevo producto y del menú en el que se quiere introducir
Salida	El identificador de menú
Origen	Interfaz de usuario
Destino	Sistema
Necesita	Acceso a la base de datos de menú y productos
Postcondición	No hay identificadores repetidos en la base de datos de ingrediente ni menú, se inserta un nuevo producto en un menú
Acciones	Comprobar que los datos son sintácticamente correctos y que no hay dos productos ni dos menús con el mismo ID. En caso positivo, se da de alta el producto dentro del menú

CASO 34: FUNCIONES MENÚ-PRODUCTO BORRAR PRODUCTO

Descrita por	Juan Embid
Prioridad	Media
Estabilidad	Alta
Descripción	Borrar un producto de un menú
Actor	Usuario
Precondición	No hay identificadores ni nombres repetidos y activos en la base de datos de menús ni en la de productos
Entrada	Identificador del producto a borrar y el del menú del que se quiere borrar
Salida	El identificador del menú
Origen	Interfaz de usuario
Destino	Sistema
Necesita	Acceso a la base de datos de menú y productos
Postcondición	No hay identificadores repetidos en la base de datos de menú, se borran los productos que contiene un menú
Acciones	Comprobar que el producto que se desea eliminar y el menú de donde se desea eliminar existen y que no hay más de un producto y menú con ese identificador. En caso positivo, dar de baja el producto del menú

CASO 35: FUNCIÓN PRODUCTO-INGREDIENTE AÑADIR INGREDIENTE A PRODUCTO	
Descrita por	Fan YE
Prioridad	Alta
Estabilidad	Alta
Descripción	Se da de alta un ingrediente en un producto
Actor	Usuario
Precondición	No hay ID repetido en la base datos de ingredientes ni la de productos
Entrada	El ID del ingrediente y el ID del producto
Salida	El producto con su contenido modificado
Origen	Interfaz de usuario Productos-Ingrediente
Destino	Sistema
Necesita	Base de datos de ingredientes y de productos
Postcondición	No hay identificadores repetidos en la base de datos. Se actualiza la información del producto
Acciones	Se comprueba que los datos son sintácticamente correctos y que no hay dos productos ni dos ingredientes con el mismo ID. En caso positivo, se da de alta el ingrediente dentro del producto

CASO 36: FUNCIÓN PRODUCTO-INGREDIENTE ELIMINAR INGREDIENTE DE PRODUCTO	
Descrita por	Fan Ye
Prioridad	Media
Estabilidad	Alta
Descripción	Se da de baja un ingrediente de producto
Actor	Usuario
Precondición	No hay ID repetido en la base datos de ingredientes ni la de productos
Entrada	El ID del ingrediente y el ID del producto
Salida	El producto con su contenido modificado
Origen	Interfaz de usuario Productos-Ingrediente
Destino	Sistema
Necesita	Base de datos de ingredientes y de productos
Postcondición	No hay identificadores repetidos en la base de datos. Se actualiza la información del producto
Acciones	Se comprueba que los datos son sintácticamente correctos y que no hay dos productos ni dos ingredientes con el mismo ID. En caso positivo, se da de baja el ingrediente dentro del producto

CASO 37: FUNCIÓN CLIENTE CREAR UN CLIENTE	
Descrita por	Victor Ortego
Prioridad	Alta
Estabilidad	Alta
Descripción	Se da de alta un cliente
Actor	Usuario
Precondición	No hay ID repetido en la base datos de clientes
Entrada	Se recoge el dni, nombre y email del cliente
Salida	Confirmación de la creación del cliente
Origen	Interfaz de usuario
Destino	Sistema
Necesita	Base de datos de clientes
Postcondición	No hay identificadores repetidos en la base de datos. Se actualiza la información del producto
Acciones	Comprobar que los datos son sintácticamente correctos, comprobar que no hay otro cliente con el mismo DNI ya activo. En caso positivo, guardar cliente en la base de datos, devolver ID. Si el DNI ya existía y estaba desactivado, activarlo.

CASO 38: FUNCIÓN CLIENTE EDITAR CLIENTE	
Descrita por	Victor Ortego
Prioridad	Alta
Estabilidad	Alta
Descripción	Se edita un cliente
Actor	Usuario
Precondición	No hay ID repetido en la base datos de clientes
Entrada	El ID del cliente
Salida	Confirmación de la modificación del cliente
Origen	Interfaz de usuario
Destino	Sistema
Necesita	Base de datos de clientes
Postcondición	No hay identificadores repetidos en la base de datos. Se actualiza la información del producto
Acciones	Se verifica que los cambios realizados son sintácticamente correctos y el identificador se corresponde con un empleado existente y activo. Si son correctos y está activo, se realizan los cambios en la base de datos. El usuario puede cancelar en cualquier momento.

CASO 39: FUNCIÓN CLIENTE ELIMINAR CLIENTE

Descrita por	Victor Ortego
Prioridad	Alta
Estabilidad	Alta
Descripción	Se da de baja un cliente
Actor	Usuario
Precondición	No hay ID repetido en la base datos de clientes
Entrada	El ID del cliente
Salida	Confirmación de la eliminación del cliente
Origen	Interfaz de usuario
Destino	Sistema
Necesita	Base de datos de clientes
Postcondición	No hay identificadores repetidos en la base de datos. Se actualiza la información del producto
Acciones	Se comprueba que los datos son sintácticamente correctos y que no hay dos productos ni dos ingredientes con el mismo ID. En caso positivo, se da de baja el cliente

CASO 40: FUNCIÓN CLIENTE MOSTRAR CLIENTE	
Descrita por	Victor Ortego
Prioridad	Alta
Estabilidad	Alta
Descripción	Se muestra un cliente
Actor	Usuario
Precondición	No hay ID repetido en la base datos de clientes
Entrada	El ID del cliente
Salida	Los datos del cliente solicitado
Origen	Interfaz de usuario
Destino	Sistema
Necesita	Base de datos de clientes
Postcondición	No hay identificadores repetidos en la base de datos. Se actualiza la información del producto
Acciones	Busca el identificador del cliente en la base de datos. Si existe y está activo, se muestra

CASO 41: FUNCIÓN CLIENTE MOSTRAR TODOS LOS CLIENTES	
Descrita por	Victor Ortego

Prioridad	Alta
Estabilidad	Alta
Descripción	Se muestran todos los clientes
Actor	Usuario
Precondición	No hay ID repetido en la base datos de clientes
Entrada	El ID del cliente
Salida	Los datos de todos los clientes
Origen	Interfaz de usuario
Destino	Sistema
Necesita	Base de datos de clientes
Postcondición	No hay identificadores repetidos en la base de datos. Se actualiza la información del producto
Acciones	Se muestran todos los clientes activos

CASO 42: FUNCIÓN CLIENTE CALCULAR PRECIO TURNOS	
Descrita por	Victor Ortego
Prioridad	Alta
Estabilidad	Alta
Descripción	Se calcula el coste de los turnos de un cliente
Actor	Usuario
Precondición	No hay ID repetido en la base datos de clientes
Entrada	El ID del cliente
Salida	El resultado del cálculo del precio
Origen	Interfaz de usuario
Destino	Sistema
Necesita	Base de datos de clientes
Postcondición	No hay identificadores repetidos en la base de datos. Se actualiza la información del producto
Acciones	Se comprueba que existe el identificador del cliente y en relación a los turnos que tenga, se calcula el precio final mostrándolo por pantalla.

CASO 43: FUNCIÓN TURNO CREAR TURNO	
Descrita por	Álvaro del Campo
Prioridad	Alta
Estabilidad	Alta
Descripción	Se crea un turno para el desayuno/comida del hotel
Actor	Usuario
Precondición	Existe el cliente al que se asocia el turno
Entrada	Día y hora del turno
Salida	Confirmación de la creación del turno
Origen	Interfaz de usuario
Destino	Sistema
Necesita	Base de datos de clientes
Postcondición	No hay identificadores repetidos en la base de datos. Se guarda el turno en la base de datos
Acciones	Comprobar que los datos son sintácticamente correctos, comprobar que el cliente al que se asocia el turno existe. En caso positivo, guardar turno en la base de datos, devolver ID. En caso negativo, informar del error.

CASO 44: FUNCIÓN TURNO EDITAR TURNO	
Descrita por	Álvaro del Campo
Prioridad	Alta
Estabilidad	Alta
Descripción	Se edita un turno
Actor	Usuario
Precondición	No hay ID repetido en la base datos de turno
Entrada	El ID del turno
Salida	Confirmación de la modificación del turno
Origen	Interfaz de usuario
Destino	Sistema
Necesita	Base de datos de clientes
Postcondición	No hay identificadores repetidos en la base de datos. Se actualiza la información del turno
Acciones	Se verifica que los cambios realizados son sintácticamente correctos, que no cause conflicto y que el identificador se corresponde con un empleado turno existente y activo. De ser así, se realizan los cambios en la base de datos. El usuario puede cancelar en cualquier momento.

CASO 45: FUNCIÓN TURNO ELIMINAR TURNO	
Descrita por	Álvaro del Campo
Prioridad	Alta
Estabilidad	Alta
Descripción	Se da de baja un turno
Actor	Usuario
Precondición	Existe el ID en la base datos de turno
Entrada	El ID del turno
Salida	ID de la mesa eliminada
Origen	Interfaz de usuario
Destino	Sistema
Necesita	Base de datos de turno
Postcondición	No hay identificadores repetidos en la base de datos. Se actualiza la información del turno
Acciones	Se comprueba que el turno que se desea eliminar existe y que no esté vinculado a un cliente. En caso positivo, marcar el turno como no activo y devolver el identificador

CASO 46: FUNCIÓN TURNO MOSTRAR TURNO	
Descrita por	Álvaro del Campo
Prioridad	Alta
Estabilidad	Alta
Descripción	Se muestra un turno
Actor	Usuario
Precondición	No hay ID repetido en la base datos de turno
Entrada	El ID del cliente
Salida	Los datos del turno solicitado
Origen	Interfaz de usuario
Destino	Sistema
Necesita	Base de datos de turno
Postcondición	Existe el turno en la base de datos. Se muestra la información de este
Acciones	Busca el identificador del cliente en la base de datos. Si existe, se muestra

CASO 47: FUNCIÓN TURNO MOSTRAR TODOS LOS TURNOS

Descrita por	Álvaro del Campo
Prioridad	Alta
Estabilidad	Alta
Descripción	Se muestran todos los turnos
Actor	Usuario
Precondición	No hay ID repetido en la base datos de turno
Entrada	-
Salida	Los datos de todos los turnos
Origen	Interfaz de usuario
Destino	Sistema
Necesita	Base de datos de clientes
Postcondición	No hay identificadores repetidos en la base de datos de turno, se muestran todos los turnos con sus atributos
Acciones	Se muestra la información de cada turno

CASO 48: FUNCIÓN RESERVA CREAR UNA RESERVA	
Descrita por	Ignacio Redondo
Prioridad	Alta
Estabilidad	Alta
Descripción	Se crea una reserva
Actor	Usuario
Precondición	No hay ids ni códigos de reserva repetidos en la BBDD
Entrada	Se recoge el código, fecha de inicio, número de días de la reserva, id cliente e id recepcionista
Salida	Confirmación de la creación de la reserva o mensaje de error
Origen	Interfaz de usuario
Destino	Sistema
Necesita	Base de datos de reserva
Postcondición	No hay identificadores ni códigos de reserva repetidos en la base de datos
Acciones	Comprobar que los datos son sintácticamente correctos, comprobar que no haya otra reserva activa con el mismo código, comprobar que el id de cliente y recepcionista se corresponda con entidades activas. Si todo se cumple, guarda una nueva reserva en la BBDD

CASO 49: FUNCIÓN RESERVA ELIMINAR UNA RESERVA	
Descrita por	Ignacio Redondo
Prioridad	Alta
Estabilidad	Alta
Descripción	Se da de baja lógica una reserva
Actor	Usuario
Precondición	No hay ids de reserva repetidos en la BBDD, la reserva no tienen ninguna relación activa con habitación ni servicio
Entrada	Se recoge el ID de la reserva a eliminar
Salida	Confirmación de la eliminación de la reserva o mensaje de error
Origen	Interfaz de usuario
Destino	Sistema
Necesita	Base de datos de reserva
Postcondición	No hay identificadores ni códigos de reserva repetidos en la base de datos
Acciones	Comprobar que el ID se corresponde con una reserva existente y activa, comprobar que no tiene relación activa con habitación o servicio. Marcar la reserva como inactiva

CASO 50: FUNCIÓN RESERVA EDITAR UNA RESERVA	
Descrita por	Ignacio Redondo
Prioridad	Alta
Estabilidad	Alta
Descripción	Se edita una reserva
Actor	Usuario
Precondición	No hay ids de reserva repetidos en la BBDD
Entrada	Se recoge el ID de la reserva a modificar y después se recogen los datos modificados. Se pueden modificar todos los atributos excepto "id" y "activa"
Salida	Confirmación de reserva editada o mensaje de error
Origen	Interfaz de usuario
Destino	Sistema
Necesita	Base de datos de reserva
Postcondición	No hay identificadores ni códigos de reserva repetidos en la base de datos, el id de cliente y recepcionista de la reserva se corresponden con entidades activas.
Acciones	Comprobar que el ID de reserva se corresponde con entidad activa, mostrar todos los datos de esa reserva, recoger datos modificados, comprobar validez sintáctica, comprobar que ID recepcionista y cliente se corresponden con entidades activas. Guardar cambios

CASO 51: FUNCIÓN RESERVA MOSTRAR UNA RESERVA	
Descrita por	Ignacio Redondo
Prioridad	Alta
Estabilidad	Alta
Descripción	Se muestra una reserva
Actor	Usuario
Precondición	No hay ids de reserva repetidos en la BBDD
Entrada	Se recoge el ID de la reserva a mostrar
Salida	Datos de la reserva o mensaje de error
Origen	Interfaz de usuario
Destino	Sistema
Necesita	Base de datos de reserva
Postcondición	No hay identificadores ni códigos de reserva repetidos en la base de datos.
Acciones	Comprobar ID reserva se corresponde con entidad activa, mostrar datos

CASO 52: FUNCIÓN RESERVA MOSTRAR TODAS LAS RESERVAS	
Descrita por	Ignacio Redondo
Prioridad	Alta
Estabilidad	Alta
Descripción	Se muestran todas las reservas
Actor	Usuario
Precondición	No hay ids de reserva repetidos en la BBDD
Entrada	Ninguna
Salida	Datos de la reservas
Origen	Interfaz de usuario
Destino	Sistema
Necesita	Base de datos de reserva
Postcondición	No hay identificadores ni códigos de reserva repetidos en la base de datos.
Acciones	Recoger datos de todas las reservas y mostrarlos

CASO 53: FUNCIÓN TURNO-CLIENTE CREAR UN NUEVO TURNO-CLIENTE	
Descrita por	Antonio Trenado Martin
Prioridad	Alta

Estabilidad	Alta
Descripción	Se crea un turno-cliente
Actor	Usuario
Precondición	No hay ids repetidos de cliente o turno y el turno no está ocupado
Entrada	Se recogen los ids del cliente y turno, y el turno que quiere reservar
Salida	Confirmación de la creación del turno-cliente o mensaje de error
Origen	Interfaz de usuario
Destino	Sistema
Necesita	Base de datos de turno-cliente
Postcondición	No hay identificadores ni códigos de cliente o turno repetidos en la base de datos
Acciones	Comprobar que los datos son sintácticamente correctos, comprobar que el id de cliente y turno se corresponda con entidades activas, y que se pueda coger el turno seleccionado. Si todo se cumple, guarda una nueva relación turno-cliente en la BBDD

CASO 55: FUNCIÓN TURNO-CLIENTE ELIMINAR UN TURNO-CLIENTE	
Descrita por	Antonio Trenado Martin
Prioridad	Alta
Estabilidad	Alta
Descripción	Se da de baja lógica un turno-cliente
Actor	Usuario
Precondición	No hay ids de clientes repetidos en la BBDD. Y el turno que quiere eliminar existe y está activo.
Entrada	Se recoge el ID del cliente con el turno a eliminar.
Salida	Confirmación de la eliminación de turno-cliente o mensaje de error
Origen	Interfaz de usuario
Destino	Sistema
Necesita	Base de datos de turno-cliente
Postcondición	No hay identificadores ni códigos de clientes y turno repetidos en la base de datos
Acciones	Comprobar que el ID se corresponde con un cliente existente y activo, y el turno a eliminar corresponde al ese cliente existe y está activo. Marcar la relación como inexistente.

CASO 56: FUNCIÓN TURNO-CLIENTE MOSTRAR TODAS LAS RELACIONES TURNO-CLIENTE	
Descrita por	Antonio Trenado Martin
Prioridad	Alta
Estabilidad	Alta
Descripción	Se muestran todas las relaciones turno-cliente
Actor	Usuario
Precondición	No hay ids de turno o cliente repetidos en la BBDD
Entrada	Ninguna
Salida	Datos de las relaciones turno-cliente
Origen	Interfaz de usuario
Destino	Sistema
Necesita	Base de datos de turno-cliente
Postcondición	No hay identificadores ni códigos de turno o cliente repetidos en la base de datos.
Acciones	Recoger datos de todas las relaciones entre turno-cliente y mostrarlas

CASO 57: FUNCIÓN TURNO-CLIENTE MOSTRAR TODOS LOS TURNOS DE UN CLIENTE	
Descrita por	Antonio Trenado Martin
Prioridad	Alta
Estabilidad	Alta
Descripción	Se muestran todos los turnos de un cliente
Actor	Usuario
Precondición	No hay ids de turno o cliente repetidos en la BBDD. Y el cliente existe y está activo.
Entrada	Ninguna
Salida	Datos de los turnos de un cliente
Origen	Interfaz de usuario
Destino	Sistema
Necesita	Base de datos de turno-cliente
Postcondición	No hay identificadores ni códigos de turno o cliente repetidos en la base de datos.
Acciones	Recoger el id del cliente que quiere mostrar todos sus turnos y comprobar que este esté activo y exista.

CASO 58: FUNCIÓN TURNO-CLIENTE MOSTRAR TODOS LOS CLIENTES DE UN TURNO

Descrita por	Antonio Trenado Martin
Prioridad	Alta
Estabilidad	Alta
Descripción	Se muestran todos los clientes de un turno.
Actor	Usuario
Precondición	No hay ids de turno o cliente repetidos en la BBDD. Y el turno existe y está activo.
Entrada	Ninguna
Salida	Datos de los clientes de un turno
Origen	Interfaz de usuario
Destino	Sistema
Necesita	Base de datos de turno-cliente
Postcondición	No hay identificadores ni códigos de turno o cliente repetidos en la base de datos.
Acciones	Recoger el id de turno que quiere mostrar a todos sus clientes y comprobar que este esté activo y exista.

CASO 59: FUNCION RESERVA-SERVICIO VINCULAR RESERVA-SERVICIO	
Descrita por	Antonio Trenado Martín, Ignacio Redondo, Álvaro del Campo, Víctor Ortego
Prioridad	Alta
Estabilidad	Alta
Descripción	Se vinculan una reserva y un servicio
Actor	Usuario
Precondición	No hay ids de reservas ni de servicios repetidos en la BBDD
Entrada	El ID de la reserva y del servicio a vincular y la fecha
Salida	Mensaje de éxito o de error
Origen	Interfaz de usuario
Destino	Sistema
Necesita	Base de datos
Postcondición	No hay ya una vinculación entre esa misma reserva y servicio para esa misma fecha
Acciones	Comprobar que ids recibidos se corresponden con entidades activas y existentes. Comprobar que no existe ya esa misma vinculación para esa misma fecha. Guardar en base de datos

CASO 60: FUNCION RESERVA-SERVICIO EDITAR RESERVA-SERVICIO	
Descrita por	Antonio Trenado Martín, Ignacio Redondo, Álvaro del Campo, Víctor Ortego
Prioridad	Alta
Estabilidad	Alta

Descripción	Se crea una relación y un reserva-servicio
Actor	Usuario
Precondición	No hay ids de reserva o servicio repetidos en la BBDD. Y tanto la reserva como el servicio existe y está activo
Entrada	El id de la reserva y servicio vinculado.
Salida	Datos de la reserva-servicio.
Origen	Interfaz de usuario
Destino	Sistema
Necesita	Base de datos de reserva-servicio.
Postcondición	No hay identificadores ni códigos de reserva o servicio repetidos en la base de datos.
Acciones	Recoger el id de la reserva que quiere modificar su fecha y el id del servicio vinculado y comprobar que existan y estén activos Comprobar que no existe ya esa vinculación. Guardar los cambios en la BBDD.

CASO 61: FUNCION RESERVA-SERVICIO DESVINCULAR RESERVA-SERVICIO	
Descrita por	Antonio Trenado Martín, Ignacio Redondo, Álvaro del Campo, Víctor Ortego
Prioridad	Alta
Estabilidad	Alta
Descripción	Se da de baja un vínculo de reserva con servicio
Actor	Usuario
Precondición	No hay ids de reservas repetidos en la BBDD. Y tanto la reserva como el servicio existen y están activos
Entrada	Se recoge el ID de la reserva con el servicio vinculado.
Salida	Confirmación de desvinculación de reserva con servicio
Origen	Interfaz de usuario
Destino	Sistema
Necesita	Base de datos de reserva-servicio
Postcondición	No hay identificadores ni códigos de reserva o de servicio repetidos en la base de datos.
Acciones	Comprobar que la reserva existe y está activa, y que el servicio está vinculado a esta. Marcar la relación entre ambos como inexistente.

CASO 62: FUNCION RESERVA-SERVICIO MOSTRAR RESERVAS DE UN SERVICIO	
Descrita por	Antonio Trenado Martín, Ignacio Redondo, Álvaro del Campo, Víctor Ortego

Prioridad	Alta
Estabilidad	Alta
Descripción	Se muestran todas las reservas vinculadas a un servicio
Actor	Usuario
Precondición	No hay ids de servicios repetidos ni vinculaciones clónicas en la BBDD
Entrada	El ID del servicio
Salida	Mensaje de error o se muestran todas las reservas vinculadas a este servicio
Origen	Interfaz de usuario
Destino	Sistema
Necesita	Base de datos
Postcondición	No hay ids de servicios repetidos ni vinculaciones clónicas en la BBDD
Acciones	Comprobar que id recibido se corresponde con entidad activa y existente, acceder a la BBDD y recoger todas las reservas vinculadas a este servicio, mostrar datos por pantalla

CASO 63: FUNCION RESERVA-SERVICIO MOSTRAR SERVICIOS DE UNA RESERVA	
Descrita por	Antonio Trenado Martín, Ignacio Redondo, Álvaro del Campo, Víctor Ortego
Prioridad	Alta
Estabilidad	Alta
Descripción	Se muestran todas los servicios vinculados a una reserva.
Actor	Usuario
Precondición	No hay ids de servicios repetidos ni vinculaciones clónicas en la BBDD
Entrada	El ID del servicio
Salida	Mensaje de error o se muestran todas los servicios vinculados a esta reserva
Origen	Interfaz de usuario
Destino	Sistema
Necesita	Base de datos
Postcondición	No hay ids de reservas repetidas ni vinculaciones clónicas a la BBDD.
Acciones	Comprobar que id recibido se corresponde con entidad activa y existente, acceder a la BBDD y recoger todas los servicios vinculados a esta reserva, mostrar datos por pantalla.

CASO 64: FUNCION RESERVA-SERVICIO MOSTRAR TODOS	
Descrita por	Antonio Trenado Martín, Ignacio Redondo, Álvaro del Campo, Víctor Ortego

Prioridad	Alta
Estabilidad	Alta
Descripción	Se muestran todas las vinculaciones reserva-servicio
Actor	Usuario
Precondición	No hay vinculaciones clónicas en la BBDD
Entrada	-
Salida	Mostrar la información de todas las relaciones reserva-servicio
Origen	Interfaz de usuario
Destino	Sistema
Necesita	Base de datos
Postcondición	No hay vinculaciones clónicas en la BBDD
Acciones	Recoger todos los datos de la BBDD y mostrarlos por pantalla

CASO 65: CREAR RECEPCIONISTA	
Descrita por	Juan Romo
Prioridad	Alta
Estabilidad	Alta
Descripción	Dar de alta a un recepcionista para que pueda atender las reservas del hotel.
Actor	Usuario
Precondición	No hay ni identificadores ni números de recepcionista repetidos.
Entrada	Salario, identificador, nombre y número recepcionista.
Salida	Identificador del recepcionista.
Origen	Interfaz de usuario de recepcionistas.
Destino	Sistema
Necesita	Base de datos de recepcionista.
Postcondición	No hay identificadores ni números de recepcionista repetidos en la base de datos. Se inserta un nuevo recepcionista con los datos introducidos.
Acciones	Comprobar que los datos son sintácticamente correctos y que no hay otro recepcionista con el mismo número. En caso positivo, crear el recepcionista (o reactivar si estaba inactivo) y devolver el identificador.

CASO 66: ELIMINAR RECEPCIONISTA	
Descrita por	Juan Romo
Prioridad	Media
Estabilidad	Alta
Descripción	Eliminación de uno de los recepcionistas previamente existentes en el hotel.
Actor	Usuario
Precondición	No hay identificadores ni números de recepcionista repetidos en la base de datos.
Entrada	Identificador del recepcionista que se desea eliminar.
Salida	Identificador del recepcionista eliminado.
Origen	Interfaz de usuario de recepcionistas.
Destino	Sistema
Necesita	Base de datos de recepcionistas.
Postcondición	No hay identificadores ni números de recepcionista repetidos en la base de datos. Se marca como inactivo el recepcionista con el identificador introducido.
Acciones	Comprobar que el recepcionista que se desea eliminar existe. En caso positivo, marcar el recepcionista como no activo y devolver el identificador.

CASO 67: EDITAR RECEPCIONISTA	
Descrita por	Juan Romo
Prioridad	Media
Estabilidad	Alta
Descripción	Editar los atributos de un recepcionista, excepto el identificador.
Actor	Usuario
Precondición	No hay identificadores ni números de recepcionista repetidos en la base de datos.
Entrada	El identificador del recepcionista a editar, luego los datos que se desean modificar sobre el recepcionista (excepto identificador).
Salida	Identificador del recepcionista.
Origen	Interfaz de usuario de recepcionistas.
Destino	Sistema
Necesita	Base de datos de recepcionista.
Postcondición	No hay identificadores ni números de recepcionista repetidos en la base de datos, se modifica al recepcionista con el identificador introducido.
Acciones	Se busca al recepcionista a editar y se verifica que los cambios realizados son sintácticamente correctos. Si son correctos y está activo, se realizan los cambios en la base de datos. El usuario puede cancelar en cualquier momento.

CASO 69: MOSTRAR RECEPCIONISTAS POR NOMBRE	
Descrita por	Juan Romo
Prioridad	Baja
Estabilidad	Alta
Descripción	Mostrar los recepcionistas con sus atributos, que coincidan con el nombre introducido.
Actor	Usuario
Precondición	No hay identificadores ni números de recepcionistas repetidos en la base de datos.
Entrada	Nombre del recepcionista que se desea mostrar.
Salida	Atributos del recepcionista pedido.
Origen	Interfaz de usuario de recepcionistas.
Destino	Sistema
Necesita	Base de datos de recepcionistas.
Postcondición	No hay identificadores ni números de recepcionista repetidos en la base de datos, se muestran los recepcionistas que coinciden con el nombre introducido.
Acciones	Busca el nombre del recepcionista en la base de datos. Si está, se muestra con los casos que coincida.

CASO 70: MOSTRAR TODOS LOS RECEPCIONISTAS	
Descrita por	Juan Romo
Prioridad	Alta
Estabilidad	Alta
Descripción	Mostrar todos los recepcionistas de la base de datos.
Actor	Usuario
Precondición	No hay identificadores ni números de recepcionista repetidos en la base de datos.
Entrada	Ninguna entrada.
Salida	Atributos de todos los recepcionistas de la base de datos.
Origen	Interfaz de usuario de recepcionistas.
Destino	Sistema
Necesita	Base de datos de recepcionistas.
Postcondición	No hay identificadores ni números de recepcionista repetidos en la base de datos, se muestran todos los recepcionistas con sus atributos.
Acciones	Se muestran todos los recepcionistas activos de la base de datos.

CASO 71: CREAR SERVICIO	
Descrita por	Juan Embid
Prioridad	Alta
Estabilidad	Alta

Descripción	Creación de un nuevo servicio en el Hotel
Actor	Usuario
Precondición	No hay identificadores en la base de datos de servicio
Entrada	Precio del servicio, tipo, identificador y descripción
Salida	Identificador del servicio
Origen	Interfaz de usuario de servicio
Destino	Sistema
Necesita	Base de datos de servicio
Postcondición	No hay identificadores de servicio repetidos en la base de datos de servicio, se crea un nuevo servicio con los datos introducidos
Acciones	Comprobar que los datos son sintácticamente correctos y que no hay otro servicio con el mismo precio y tipo. En caso positivo, crear el servicio (o lo activa si estaba inactivo) y devolver el identificador

CASO 72: ELIMINAR SERVICIO	
Descrita por	Juan Embid
Prioridad	Alta
Estabilidad	Alta
Descripción	Eliminación de un servicio en el Hotel
Actor	Usuario
Precondición	No hay identificadores en la base de datos de servicio repetidos
Entrada	Identificador del servicio que se desea eliminar
Salida	Identificador del servicio
Origen	Interfaz de usuario de servicio
Destino	Sistema
Necesita	Base de datos de servicio
Postcondición	No hay identificadores de servicio repetidos en la base de datos de servicio, se marca el servicio con el identificador introducido como inactivo
Acciones	Comprobar que el servicio que se desea eliminar existe. En caso positivo, marcar el servicio como no inactivo y devolver el identificador

CASO 73: EDITAR SERVICIO	
Descrita por	Juan Embid
Prioridad	Alta
Estabilidad	Alta

Descripción	Edición de un servicio en el Hotel
Actor	Usuario
Precondición	No hay identificadores en la base de datos de servicio repetidos
Entrada	Identificador, precio del servicio, tipo y descripción
Salida	Identificador del servicio
Origen	Interfaz de usuario de servicio
Destino	Sistema
Necesita	Base de datos de servicio
Postcondición	No hay identificadores de servicio repetidos en la base de datos de servicio, se edita el servicio con los datos introducidos
Acciones	Comprobar que los datos son sintácticamente correctos y que no hay otro servicio con el mismo precio y tipo. En caso positivo, crear el servicio (o lo activa si estaba inactivo) y devolver el identificador

CASO 74: MOSTRAR UN SERVICIO	
Descrita por	Juan Embid
Prioridad	Alta
Estabilidad	Alta
Descripción	Mostrar un servicio con sus atributos
Actor	Usuario
Precondición	No hay identificadores repetidos en la base de datos de servicio
Entrada	Identificador del servicio que se desea mostrar
Salida	Identificador del servicio
Origen	Interfaz de usuario de servicio
Destino	Sistema
Necesita	Base de datos de servicio
Postcondición	No hay identificadores de servicio repetidos en la base de datos de servicio, se muestra el servicio con el identificador introducido
Acciones	Busca el identificador del servicio en la base de datos. Si está, se muestra

CASO 75: MOSTRAR TODOS LOS SERVICIOS	
Descrita por	Juan Embid
Prioridad	Alta
Estabilidad	Alta
Descripción	Mostrar todos los servicios de su base de datos

Actor	Usuario
Precondición	No hay identificadores repetidos de servicio en la base de datos
Entrada	Sin entrada
Salida	Todos los servicios de la base de datos
Origen	Interfaz de usuario de servicio
Destino	Sistema
Necesita	Base de datos de servicio
Postcondición	No hay identificadores ni números de servicio repetidos en la base de datos de servicio, se muestran todos los servicios con sus atributos
Acciones	Se muestran todos los servicios

CASO 76: MOSTRAR POR TIPO DE SERVICIO

Descrita por	Juan Embid
Prioridad	Alta
Estabilidad	Alta
Descripción	Mostrar un servicio por tipo con todos sus atributos
Actor	Usuario
Precondición	No hay identificadores repetidos en la base de datos de servicio
Entrada	El tipo de servicio
Salida	Identificador del servicio
Origen	Interfaz de usuario de servicio
Destino	Sistema
Necesita	Base de datos de servicio
Postcondición	No hay identificadores ni números de servicio repetidos en la base de datos de servicio, se muestran todos los servicios de un tipo con sus atributos
Acciones	Se muestran todos los servicios de un tipo

CASO 77: FUNCIÓN HABITACIÓN-RESERVA VINCULAR HABITACIÓN A RESERVA

Descrita por	Samuel Parra Calderón
Prioridad	Alta
Estabilidad	Alta
Descripción	Se da de alta una habitación en una reserva

Actor	Usuario
Precondición	No hay ID repetido en la base datos de habitaciones ni las de reservas
Entrada	El ID de la habitación y el ID de la reserva
Salida	La reserva con su contenido modificado
Origen	Interfaz de usuario Habitación-Reserva
Destino	Sistema
Necesita	Base de datos de habitaciones y de reservas
Postcondición	No hay identificadores repetidos en la base de datos. Se actualiza la información de la reserva
Acciones	Se comprueba que los datos son sintácticamente correctos y que no hay dos habitaciones ni dos reservas con el mismo ID, además de que ambos existen y están activos y no están previamente vinculados. En caso positivo, se da de alta la habitación dentro de la reserva

CASO 78: FUNCIÓN HABITACIÓN-RESERVA DESVINCULAR HABITACIÓN DE RESERVA	
Descrita por	Samuel Parra Calderón
Prioridad	Alta
Estabilidad	Alta
Descripción	Se da de baja una habitación de una reserva
Actor	Usuario
Precondición	No hay ID repetido en la base datos de habitaciones ni las de reservas
Entrada	El ID de la habitación y el ID de la reserva
Salida	La reserva con su contenido modificado
Origen	Interfaz de usuario Habitación-Reserva
Destino	Sistema
Necesita	Base de datos de habitaciones y de reservas
Postcondición	No hay identificadores repetidos en la base de datos. Se actualiza la información de la reserva
Acciones	Se comprueba que los datos son sintácticamente correctos y que no hay dos habitaciones ni dos reservas con el mismo ID, además de que existen y ambos están activos. En caso positivo, se da de baja la habitación dentro de la reserva

CASO 79: FUNCIÓN HABITACIÓN-RESERVA MOSTRAR HABITACIONES DE UNA RESERVA	
Descrita por	Samuel Parra Calderón
Prioridad	Alta
Estabilidad	Alta
Descripción	Se muestran todas las habitaciones vinculadas a una reserva

Actor	Usuario
Precondición	No hay ID repetido en la base datos de habitaciones ni las de reservas
Entrada	El ID de la reserva
Salida	Mensaje de error o se muestran todas las habitaciones vinculadas a este reserva
Origen	Interfaz de usuario Habitación-Reserva
Destino	Sistema
Necesita	Base de datos de habitaciones y de reservas
Postcondición	No hay identificadores repetidos en la base de datos ni vinculaciones clónicas
Acciones	Comprobar que el id recibido existe y está activo, acceder a la BBDD y obtener las habitaciones vinculadas a la reserva para luego mostrar los datos por pantalla

CASO 80: FUNCIÓN HABITACIÓN-RESERVA MOSTRAR RESERVAS DE UNA HABITACION

Descrita por	Samuel Parra Calderón
Prioridad	Alta
Estabilidad	Alta
Descripción	Se muestran todas las reservas vinculadas a una habitación
Actor	Usuario
Precondición	No hay ID repetido en la base datos de habitaciones ni las de reservas
Entrada	El ID de la habitación
Salida	Mensaje de error o se muestran todas las reservas vinculadas a este habitación
Origen	Interfaz de usuario Habitación-Reserva
Destino	Sistema
Necesita	Base de datos de habitaciones y de reservas
Postcondición	No hay identificadores repetidos en la base de datos ni vinculaciones clónicas
Acciones	Comprobar que el id recibido existe y está activo, acceder a la BBDD y obtener las reservas vinculadas a la habitación para luego mostrar los datos por pantalla

CASO 81: FUNCIÓN HABITACIÓN-RESERVA MOSTRAR TODOS

Descrita por	Samuel Parra Calderón
Prioridad	Alta
Estabilidad	Alta
Descripción	Se muestran todas las vinculaciones habitación-reserva
Actor	Usuario
Precondición	No hay ID repetido en la base datos de habitaciones ni las de reservas

Entrada	-
Salida	Se muestran todas las vinculaciones habitación-reserva
Origen	Interfaz de usuario Habitación-Reserva
Destino	Sistema
Necesita	Base de datos de habitaciones y de reservas
Postcondición	No hay identificadores repetidos en la base de datos ni vinculaciones clónicas
Acciones	Obtener todas las vinculaciones Habitación-Reserva de la BBDD y luego mostrar los datos por pantalla

CASO 82: EDITAR HABITACIÓN	
Descrita por	Samuel Parra Calderón
Prioridad	Alta
Estabilidad	Alta
Descripción	Editar los atributos de habitación, excepto el ID
Actor	Usuario
Precondición	No hay ID repetido en la base datos de habitaciones
Entrada	El ID de la habitación, luego los atributos a editar menos el identificador
Salida	Identificador de la habitación
Origen	Interfaz de usuario Habitación
Destino	Sistema
Necesita	Base de datos de habitaciones
Postcondición	No hay identificadores de habitación repetidos, se modifica la habitación con los datos introducidos
Acciones	Se comprueba que los datos introducidos son sintácticamente correctos y que el ID de la habitación existe y está activo. De ser así, se realizan los cambios en la base de datos. El usuario puede cancelar en cualquier momento
CASO 83: MOSTRAR HABITACION POR NÚMERO	

Descrita por	Juan Embid Sánchez
Prioridad	Alta
Estabilidad	Alta
Descripción	Se muestra una habitación por número
Actor	Usuario
Precondición	No hay habitación con el número repetido en la base de datos de habitación
Entrada	El número de la habitación
Salida	Identificador de la habitación
Origen	Interfaz de usuario de habitación
Destino	Sistema
Necesita	Base de datos de habitación
Postcondición	No hay ni identificadores ni números de habitación repetidos en la base de datos, se muestra la habitación con todos sus atributos.
Acciones	Busca el número de habitación en la base de datos. Si está, se muestra.

CASO 84: MOSTRAR HABITACION POR IDENTIFICADOR	
Descrita por	Juan Embid Sánchez
Prioridad	Alta
Estabilidad	Alta
Descripción	Se muestra una habitación por identificador
Actor	Usuario
Precondición	No hay habitación con el identificador repetido en la base de datos de habitación
Entrada	El identificador de la habitación
Salida	Identificador de la habitación
Origen	Interfaz de usuario de habitación
Destino	Sistema
Necesita	Base de datos de habitación
Postcondición	No hay identificadores de habitación repetidos en la base de datos, se muestra la habitación con todos sus atributos.
Acciones	Busca el identificador de habitación en la base de datos. Si está, se muestra.
CASO 85: CREAR HABITACIÓN	

Descrita por	Carlos Forriol Molina
Prioridad	Alta
Estabilidad	Alta
Descripción	Creamos una nueva habitación con su id, número y planta
Actor	Usuario
Precondición	No hay ID ni número repetido en la base datos de habitaciones
Entrada	El ID de la habitación, su número y su planta
Salida	Confirmación de la nueva mesa creada, con su ID y su número
Origen	Interfaz de usuario Habitación
Destino	Sistema
Necesita	Base de datos de habitaciones
Postcondición	No hay identificadores repetidos en la base de datos. Se inserta nueva habitación
Acciones	Comprobar que los datos son sintácticamente correctos y que no hay 2 habitaciones con mismo ID ni número. En caso positivo, guardar habitación en la base de datos y devolver ID. Si el DNI ya existía y estaba desactivado, activarlo.

CASO 86: MOSTRAR TODAS LAS HABITACIONES	
Descrita por	Juan Romo Iribarren
Prioridad	Alta
Estabilidad	Alta
Descripción	Mostrar todas las habitaciones con sus atributos.
Actor	Usuario
Precondición	No hay identificadores ni números de habitación repetidos en la base datos de habitaciones.
Entrada	Petición para mostrar todas las habitaciones
Salida	Atributos de todas las habitaciones de la base de datos
Origen	Interfaz de usuario Habitación.
Destino	Sistema
Necesita	Base de datos de habitaciones.
Postcondición	No hay identificadores ni números de habitación repetidos en la base de datos, se muestran todas las habitaciones activas con sus atributos.

Acciones	Se muestran todas las habitaciones activas
-----------------	--

CASO 87: ELIMINAR HABITACIÓN	
Descrita por	Juan Romo Iribarren
Prioridad	Media
Estabilidad	Alta
Descripción	Eliminación de una de las habitaciones previamente existentes en el hotel.
Actor	Usuario
Precondición	No hay identificadores repetidos en la base datos de habitaciones.
Entrada	El ID de la habitación que se desea eliminar.
Salida	El ID de la habitación eliminada.
Origen	Interfaz de usuario Habitación.
Destino	Sistema
Necesita	Base de datos de habitaciones.
Postcondición	No hay identificadores repetidos en la base de datos. Se marca como inactivo a la habitación con el identificador introducido
Acciones	Se comprueba que la habitación que se desea eliminar existe. En caso positivo, marcar la habitación como no activa y devolver el identificador.

CASO 88: Mostrar Menús por rango de precio

Descrita por	Juan Montero Gómez
Prioridad	Alta
Estabilidad	Alta
Descripción	Muestra los menús que tienen un precio dentro de un rango determinado
Actor	Usuario
Precondición	No hay identificadores repetidos en la base de datos de menús
Entrada	Un precio mínimo y un precio máximo
Salida	Los menús con precio en ese rango
Origen	Interfaz de usuario
Destino	Sistema
Necesita	Base de datos de menú
Postcondición	No hay identificadores repetidos en la base de datos de menú. No se produce ningún cambio en la base de datos de menú
Acciones:	Se verifica que los datos son sintácticamente correctos. Si lo son se muestran todos los menús que se encuentran en ese rango de precio. Si no lo son se informa al usuario de datos incorrectos.

CASO 89: Mostrar Empleados despedidos en un rango salarial	
Descrita por	Juan Montero Gómez
Prioridad	Alta
Estabilidad	Alta
Descripción	Muestra los empleados despedidos que tenían un salario dentro de un rango determinado
Actor	Usuario
Precondición	No hay identificadores repetidos en la base de datos de empleados
Entrada	Un salario mínimo y un salario máximo
Salida	Los empleados despedidos que tenían un salario dentro del rango de entrada
Origen	Interfaz de usuario
Destino	Sistema
Necesita	Base de datos de empleado
Postcondición si éxito	No hay identificadores repetidos en la base de datos de empleado. No se produce ningún cambio en la base de datos de empleado.
Acciones:	Se verifica que los datos son sintácticamente correctos. Si lo son se muestran todos los empleados despedidos cuyo salario estaba dentro del rango. En caso contrario se informa al usuario de datos no válidos.

CASO 90: MOSTRAR PRODUCTOS POR INGREDIENTE	
Descrita por	Juan José Encina Fernández
Prioridad	Media Baja
Estabilidad	Alta
Descripción	Muestra una lista de todos los productos que tiene un ingrediente
Actor	Usuario

Precondición	No hay identificadores repetidos en la base de datos de producto ni de ingredientes
Entrada	Id de ingrediente
Salida	Lista de productos vinculados a ingrediente o mensaje de error
Origen	Interfaz de usuario de producto-ingrediente
Destino	Sistema
Necesita	Base de datos de producto e ingredientes
Postcondición	No hay identificadores repetidos en la base de datos ni vinculaciones crónicas.
Acciones	Se comprueba si el id del ingrediente está activo y existe. En caso positivo, accede a la base de datos y obtiene y muestra la lista de productos vinculado al ingrediente. En caso contrario se informa al usuario

CASO 91: MOSTRAR MENÚ DE UN PRODUCTO	
Descrita por	Juan José Encina Fernández
Prioridad	Media Baja
Estabilidad	Alta
Descripción	Muestra una lista de todos los menús vinculados a un producto
Actor	Usuario
Precondición	No hay identificadores repetidos en la base de datos del id del producto, ni de los menús
Entrada	Id del producto
Salida	El listado de menús o mensaje de error
Origen	Interfaz de usuario de menú-producto
Destino	Sistema
Necesita	Base de datos de producto y menú
Postcondición	No hay identificadores repetidos en la base de datos ni vinculaciones crónicas. Se muestran los productos o mensaje de error
Acciones	Se comprueba si el id del producto está activo y existe. En caso positivo, se accede a la base de datos, se obtienen los menús vinculados al producto y se muestran. En caso contrario se informa al usuario

CASO 92: MOSTRAR FACTURAS POR MENÚ	
Descrita por	Juan José Encina Fernández
Prioridad	Media Baja
Estabilidad	Alta
Descripción	Muestra una lista de todas las facturas que están vinculadas a un menú
Actor	Usuario

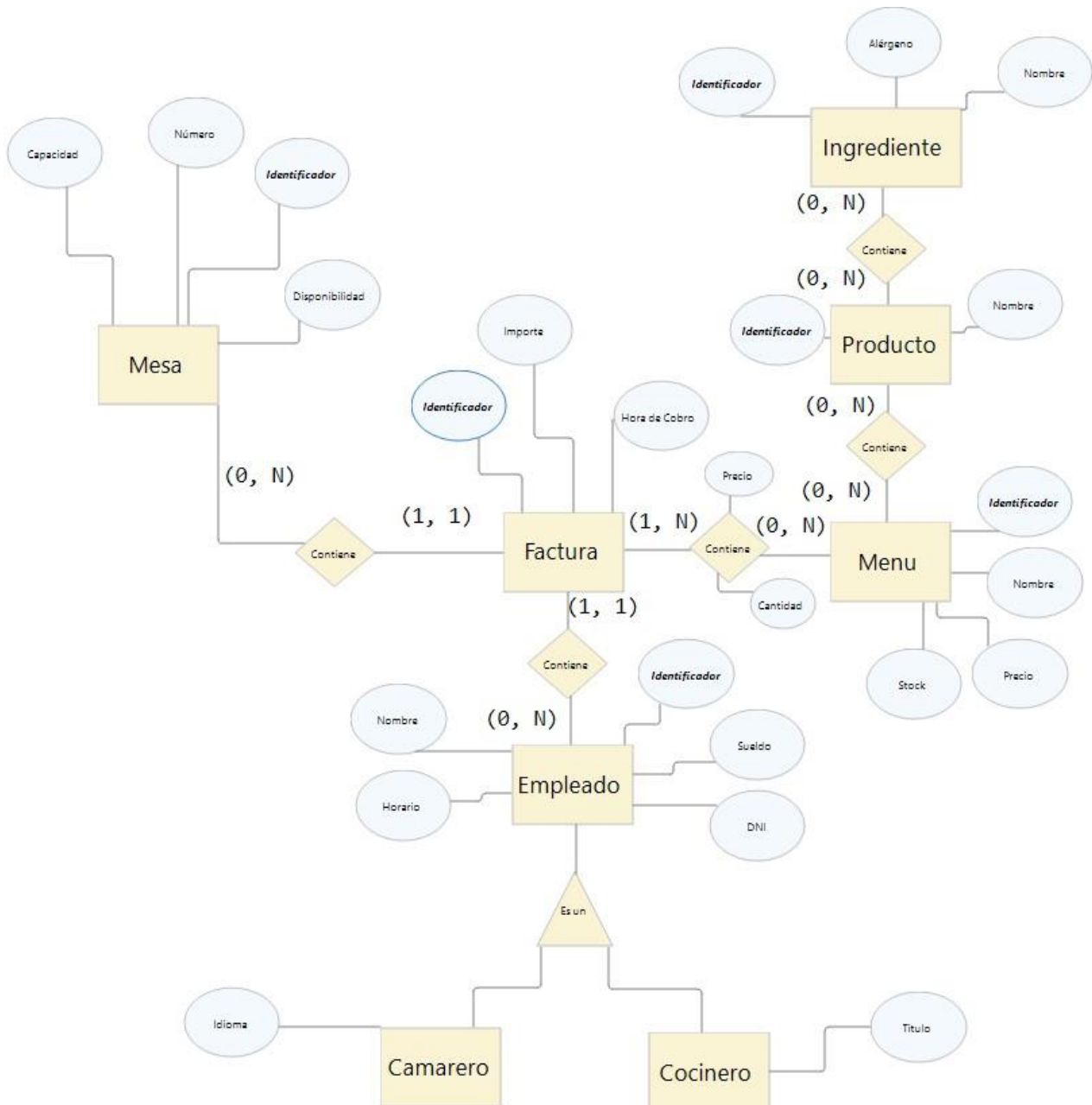
Precondición	No hay identificadores repetidos en la base de datos de menú ni factura
Entrada	Id del menú
Salida	El listado de facturas o mensaje de error
Origen	Interfaz de usuario de facturas-menú
Destino	Sistema
Necesita	Base de datos de factura y menú
Postcondición	No hay identificadores repetidos en la base de datos ni vinculaciones clónicas. Se muestran las facturas o mensaje error
Acciones	Se comprueba si el id del menú esta activo y existe. En caso positivo, accede a la base de datos, se buscan las facturas vinculadas a él menú se muestra. En caso contrario se informa al usuario

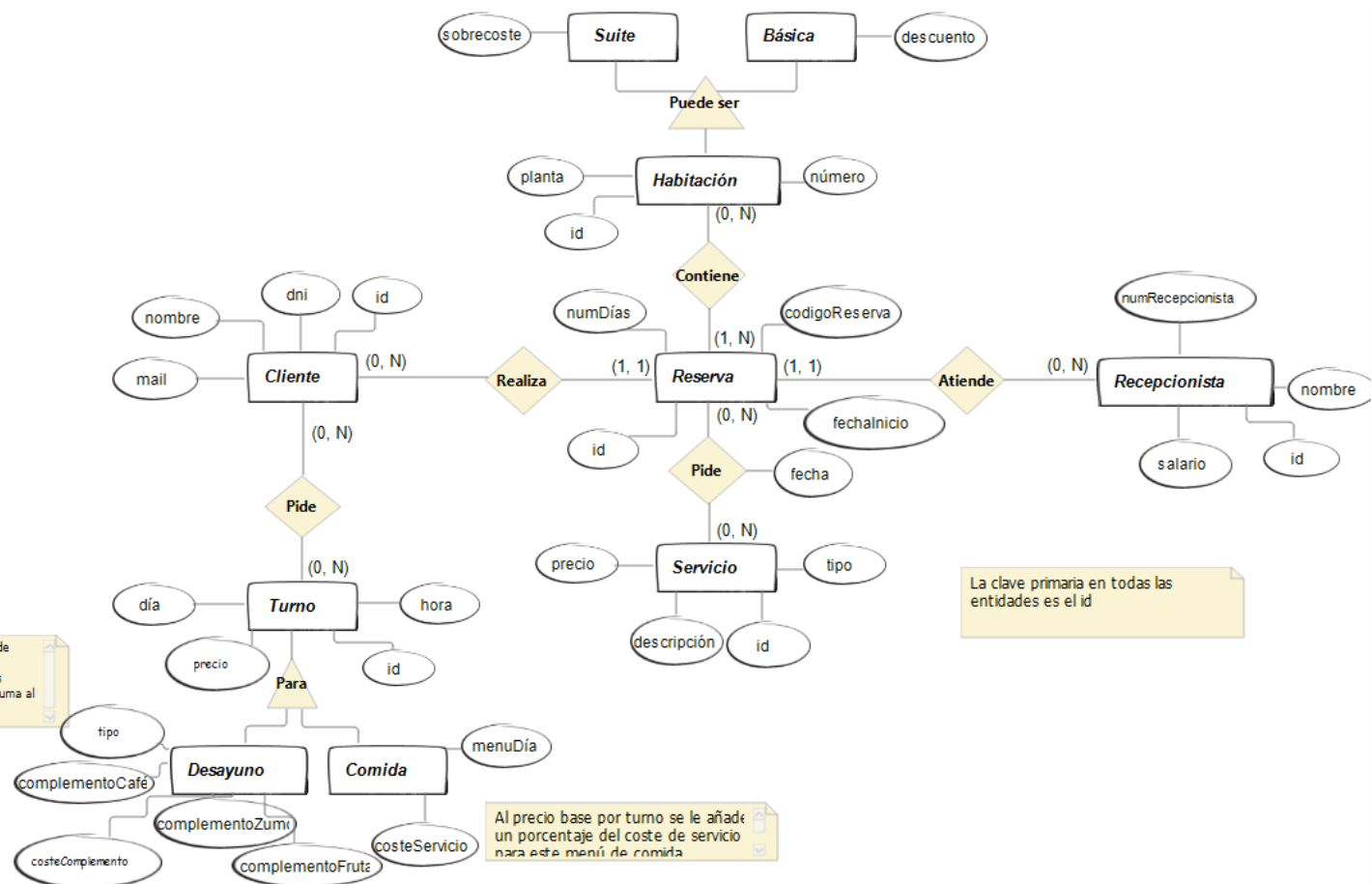
3.3 Requisitos De Rendimiento

Debido a la naturaleza rápida de un restaurante, es necesario que las operaciones realizadas se completen *rápido*. Se ejecuta en una sola terminal y un usuario.

3.4 Requerimientos Lógicos De La Base De Datos

Esta subsección hace referencia a los requisitos lógicos para cualquier información que resida en la base de datos del programa. El modelo del dominio es:





3.5 Restricciones De Diseño

Este programa está diseñado para ejecutarse en un único ordenador. Naturalmente son necesarios unos periféricos básicos (teclado, ratón y pantalla), un procesador, y una conexión a internet, ya que el programa debe acceder a una base de datos externa. También está diseñado para funcionar en Windows 10, con una instalación del Java Runtime Environment (JRE) para poder ejecutarse. Utiliza un sistema de gestión de bases de datos MySQL.

3.6 Atributos Del Sistema Software

Las que se aplican normalmente en un programa de gestión de datos prestando especial atención a la mantenibilidad y a la persistencia de los datos.

Apéndices