**Sistema Chimera para la gestión de un restaurante**

# 1.Descripción

Nuestro sistema se va a encargar de la gestión de un restaurante controlado por dos tipos de usuarios, el usuario **administrador** (gestor o encargado) y el usuario **personal** del restaurante (camareros), también contaremos con distintos **proveedores**, los cuales serán los encargados de proporcionar todo tipo de **productos** para hacer que el servicio ofrecido a los **clientes** sea lo más efectivo posible y de calidad, agilizando el proceso de ubicarlos en las **mesas** para que esperen el menor tiempo posible, y a su vez dicho sistema se encargará de hacer la gestión de **pagos** muy sencilla para que no haya ningún problema. Esta gestión se va a dividir en cinco partes donde cada una de esas partes va a tener sus propias funciones y requisitos.

En primer lugar trataremos la gestión del personal del restaurante, el cual es el encargado de gestionar todo lo relacionado acerca del personal que va a trabajar en el restaurante, datos, horarios, funciones a realizar...

Nos encontraremos con dos tipos de usuario. El usuario personal y el usuario administrador del sistema.

En este subsistema vamos a enumerar cinco requisitos funcionales, para **dar de alta** un usuario, el usuario deberá introducir el nombre (hasta 20 caracteres), sus apellidos (hasta 40 caracteres), su teléfono (hasta 20 caracteres), su DNI (hasta 10 caracteres) y el cargo que van a tomar (hasta 20 caracteres). Si algún campo mencionado está vacío se devolverá un error y no se introducirá nada.

Además debe de tener en cuenta que campo DNI y teléfono solo corresponden a un único usuario.

Para **dar de baja** a un usuario, deberá de hacerlo el administrador mediante la inserción del DNI, y se mostrará un error sí ese DNI no se encuentra almacenado.

Para **establecer un horario** a un usuario, el usuario o usuario jefe, deberá introducir los días de la semana que trabaja ese usuario(hasta 10 caracteres), la fecha de entrada y la fecha de salida(ambos hasta 5 caracteres). Estos campos deberán introducirse en un formato correcto, en caso contrario no se introducirán y se devolverá un error.

Para **mostrar el horario** de un usuario, el sistema mostrará un listado con los horarios del usuario que lo solicite mediante la inserción de su DNI.

Y por último para **asignar una mesa** el camarero encargado deberá comprobar si esa mesa identificada por un número de mesa (hasta 3 caracteres) está ocupada o no mediante el dato ocupada (de tipo booleano). En el caso de intentar asignar una mesa que ya se encuentra ocupada se devolverá un error. En el caso de no estarlo se asignará a dicho número de mesa y se cambiará el estado a ocupada.

En segundo lugar, nos encontramos con la gestión de productos, mediante este subsistema seremos capaces de administrar todos los productos disponibles del inventario y conocer su vez la fecha de caducidad de cada uno de ellos. Además, el cliente tendrá la opción de establecer una valoración de cada producto.

En este subsistema vamos a contar con cinco requisitos funcionales, el empleado podrá **añadir un producto** al inventario siempre y cuando el nombre del producto no esté vacío y exista, la cantidad a añadir sea mayor que 0 y la fecha de caducidad sea posterior a la fecha actual.

El empleado de igual forma podrá **eliminar un producto**, simplemente introduciendo el nombre del producto (nombre existente) y la cantidad que desee eliminar (que deberá ser igual o mayor que 1).

En cualquier momento, el empleado tendrá la posibilidad de realizar una consulta para **mostrar el inventario** de productos, y aparecerá en pantalla un listado con todos los productos disponibles, junto con la cantidad disponible.

De igual forma, el empleado podrá realizar una consulta para **mostrar la fecha de caducidad de un producto**, simplemente introduciendo el nombre del producto, obtendremos dicho producto, la cantidad existente de dicho producto y su fecha de caducidad asociada. Si el nombre del producto del producto no existe o está vacío obtendremos un error.

Finalmente, el cliente tendrá la oportunidad de realizar una **valoración del producto** o los productos que él desee, introduciendo el nombre del producto y una valoración, en caso de que no exista el producto o la valoración no esté comprendida en el rango de 0 a 5, el cliente recibirá un mensaje de error.

En tercer lugar tenemos el subsistema que se encargará de gestionar las mesas, este será gestionado por el usuario personal, es decir, los camareros. En este subsistema, se gestionará la reserva y el número de mesas dentro de nuestro sistema así como se podrá obtener la carta y los platos para que el usuario personal informe a los clientes.

El subsistema gestionará la **reserva de las mesas,** que el usuario modificará el estado de la mesa de reservado a no reservado según convenga.

También tendremos la funcionalidad de **mostrar plato** que mostrará por pantalla los ingredientes que contiene un plato también su precio las alergias que puede contener el plato para informar a los clientes, así como la funcionalidad de **mostrar la carta** donde saldrá por pantalla todo el repertorio de platos, bebidas, tapas y precios de estos.

Para **añadir mesa** el usuario personal tendrá que introducir el nuevo identificador de la mesa que será un entero, su estado si está reservada o no, estos dos parámetros tienen que estar completos o si no no se completara la funcionalidad y el número de clientes que tendrá la mesa, este último valor será un entero mayor o igual a 0.

Y por último el personal podrá **quitar mesa**, que eliminará los parámetros de una mesa, es decir eliminará la mesa, y saldrá por pantalla un mensaje informando de que se ha eliminado correctamente.

En cuarto lugar se tratará la gestión de proveedores. Este subsistema será gestionado por el administrador(gestor) y tratará el manejo de información de los diferentes proveedores del restaurante.

Este subsistema se encargará de **dar de alta** a los proveedores y para ello se deberá introducir el nombre del proveedor (cadena de hasta 20 caracteres), su teléfono de contacto (cadena de hasta 20 caracteres) y su correo electrónico (cadena de hasta 40 caracteres). El sistema le asignará a la entrada un IDProveedor automático que no puede ser nulo y no puede ser igual a otro ya existente en la base de datos, el cual devolverá para informar de que id se le ha asignado a ese proveedor.

**Para dar de baja** a un proveedor el gestor deberá proporcionar el IDProveedor que es un número entero y confirmar el borrado o dará un error.

El sistema también permitirá **mostrar la información de un proveedor** introduciendo el número entero de IDProveedor por el que se identifican los proveedores devolviendo un error si no existe el ID.

Asimismo, el sistema también permitirá **valorar a un proveedor** asignándole una valoración (RatingProveedor que es un entero comprendido entre el 0 y el 5 ambos incluidos) y un comentario (cadena larga de hasta 400 caracteres). Para ello el gestor deberá introducir el ID del proveedor, la valoración entre el 0 y el 5 y un comentario opcional. Si el ID no existe devolverá un error.

En quinto lugar, trataremos sobre la gestión de pagos, la cual podremos consultar con el sistema tanto el número de horas trabajadas por empleado, la cantidad de productos solicitados a los proveedores, el precio de cada producto para el cobro de mesas.

En este subsistema contaremos con cinco requisitos funcionales, el camarero puede proceder al **cobro de mesas** solicitando la cuenta al sistema siempre y cuando la mesa no haya sido cobrada por otro camarero o la cuenta de esta sea 0.

En relación al **pago a clientes y proveedores** el agente encargado de interactuar con el sistema se trata del gerente el cual consultará el número de horas trabajadas por cada empleado y la cantidad a pagar por hora, asimismo este se encargará de realizar el pago a los proveedores consultando la cantidad de productos solicitados a estos, estos dos requisitos funcionales no se podrán llevar a cabo si el número de horas trabajados por empleado o la cantidad de productos solicitados a los proveedores es igual a 0.

En cuanto a la situación en la que el cliente quiera **consultar la factura**, el camarero solicitará al sistema una factura en la cual figurarán el precio de cada producto solicitado por el cliente.

Por último lugar en relación al **pago del alquiler** del local, se solicitará al sistema la cantidad a pagar y la fecha del cobro por el alquiler del local, este no se podrá llevar a cabo si la cantidad a pagar es igual a 0.

# 2.Subsistemas

## **1.Gestión de personal: (Jesus)**

**RF1.1: Dar de alta**

Entrada: agente externo: usuario. Acción: solicitar inserción.

Requisito de datos de entrada RDE1.1.

Bd: requisito de datos de escritura RDW1.1.

Salida: agente externo: usuario. Acción: confirmación del resultado.

Requisito de datos de salida ninguno.

RDE1.1: Datos de entrada de alta de empleado.

DNI: Cadena de caracteres (10)

Nombre: Cadena de caracteres (20)

Apellidos: Cadena de caracteres (40)

Teléfono: Cadena de caracteres (20)

Cargo: Cadena de caracteres (20)

RDW1.1: Datos almacenados de empleados. Los mismos datos que RDE1.

**RF1.2: Dar de baja**

Entrada: agente externo: usuario. Acción: solicitar borrado.

Requisitos de datos de entrada RDE1.2.

Bd: requisito de datos de escritura RDW1.2.

Salida: agente externo: usuario. Acción: confirmar borrado.

Requisito de datos de salida ninguno.

RDE1.2: Datos de empleado de baja. Mismos datos que RDE.1.1

RDW1.2: Datos almacenados de empleado seleccionado en RDE1.1.

**RF1.3: Establecer horario**

Entrada: agente externo: usuario. Acción: insertar horario de empleado.

Requisito de datos de entrada RDE1.3.

Bd: requisito de datos de escritura RDW1.3.

Salida: agente externo: usuario. Acción: confirmación de inserción de horario.

Requisito de datos de salida ninguno.

RDE1.3: Datos de entrada con fecha de entrada y salida y dni del usuario a establecer horario.

Dni: mismo dni que en RDE1.1

Dia de la semana: cadena de caracteres(10)

Hora de entrada: cadena de caracteres (5)

Hora de salida: cadena de caracteres (5)

RDW1.3: Datos iguales a RDE1.3 menos dni

**RF1.4: Mostrar horario**

Entrada: agente externo: usuario. Acción: solicitar ver horario.

Requisito de datos de entrada: RDE1.4

Bd: requisito de datos de lectura: RDR1.1,RDR1.4.

Salida: agente externo: usuario. Acción: Mostrar listado de horario.

Requisitos de datos de entrada: Dni del usuario a mostrar, mismo DNI que en RDE1.1

Requisito de datos de salida RDS1.4.

RDR1.4: datos del horario almacenado. Mismos datos que en RDW1.3.

RDS1.4: datos de horario, mismos que datos de RDR1.4.

**RF1.5: Asignar mesas**

Entrada: agente externo: usuario. Acción: asignar una mesa a un cliente.

Requisito de datos de entrada RDE1.5.

Bd: requisito de datos de escritura RDW1.5.RDR1.5

Salida: agente externo: usuario. Acción: confirmación de asignación de mesa.

Requisito de datos de salida ninguno.

RDE1.5: Datos de entrada con el número de mesa.

Número de mesa: cadena de caracteres (3)

Ocupada : booleano

RDW1.5: Mismos datos que en RDE1.5

RDR1.5: mismos datos que RDW1.5

**Restricciones semánticas del subsistema 1.**

RS1.1: Dni,nombre, apellidos, teléfono y cargo no pueden ser vacíos.

RF: RF1.1

RD(s): RDW1.1

**Descripción**: sí alguno de estos campos está vacío no se introduce nada y se devuelve un error.

RS1.2: Un dni corresponde a un único usuario.

RF: RF1.1

RD(s): RDW1.1

**Descripción**: si se introduce un dni ya existente, no se introduce y se devuelve un error.

RS1.3: Un teléfono corresponde a un solo usuario.

RF: RF1.1

RD(s): RDW1.1

**Descripción**: Si existe un contacto igual no se introduce y se devuelve un error.

RS1.4: Debe existir un usuario con el dni introducido a borrar.

RF: RF1.2

RD(s): RDW1.2

**Descripción**: Si no existe un dni introducido del usuario a borrar, se devuelve un error.

RS1.5: No se debe asignar la misma mesa que ya se encuentra asignada

RF: RF1.5

RD(s): RDW1.5,RDR1.5

**Descripción**: Sí una mesa ya asignada se intenta asignar de nuevo antes de ser liberada se devuelve un error.

## **2.Gestión de productos: (Andrés Arco López)**

**RF2.1: Añadir producto**

Entrada:Agente Externo: empleado Acción: Añadir una determinada cantidad de productos y su fecha de caducidad en el inventario.

BD: Requisito de datos de escritura RDW2.1

Salida: Agente externo: empleado. Acción : confirmación resultado. Requisito de datos de salida: ninguno.

RDE2.1: Datos de entrada de un nuevo producto

NombreProducto: Cadena de caracteres (20)

Cantidad: Entero

FechaCaducidad: Fecha (dd/mm/yyyy)

RDW2.1: Los mismos que RDE2.1

**RF2.2: Eliminar producto**

Entrada**:** Agente Externo: empleado Acción: Eliminar una determinada cantidad de productos del inventario.

BD: Requisito de datos de escritura RDW2.2

Salida: Agente externo: empleado. Acción : confirmación resultado. Requisito de datos de salida: ninguno.

RDE2.2:Datos de entrada de un producto a eliminar

NombreProducto: Cadena de caracteres (20)

Cantidad: Entero

RDW2.2: Los mismos que RDE2.2

**RF2.3: Mostrar inventario**

Entrada:Agente Externo: empleado Acción: Solicitar la visualización de los productos existentes en el inventario.

BD: Requisito de datos de lectura RDR2.3

Salida: Agente externo: empleado. Acción : muestra del listado de productos del inventario. Requisito de datos de salida: ninguno.

RDR2.3: Se leen los datos de los productos registrados en el inventario

NombreProducto: Cadena de caracteres (20)

Cantidad: Entero

Valoracion: Entero

RDS2.3: Se muestran los datos de los productos existentes en el inventario, junto con su cantidad y su valoración (en caso de que exista). Mismos que RDR2.3.

**RF2.4: Mostrar fecha caducidad de producto**

Entrada:Agente Externo: empleado Acción: Mostrar la fecha de caducidad de un producto.

BD: Requisito de datos de lectura RDR2.4

Salida: Agente externo: empleado. Acción : muestra de producto y fecha caducidad. Requisito de datos de salida: ninguno.

RDE2.4:Datos de un producto

NombreProducto: Cadena de caracteres (20)

RDR2.4: Se leen los datos del producto indicado

NombreProducto: cadena de caracteres (20)

FechaCaducidad: Fecha (dd/mm/yyyy)

RDS2.4: Se muestra el producto y su fecha de caducidad. Mismos datos que RDR2.4.

**RF2.5: Valoración producto**

Entrada:Agente Externo: cliente Acción: Establecer una valoración de un producto en un rango de 0 a 5 (estrellas).

BD: Requisito de datos de escritura RDW2.5

Salida: Agente externo: cliente. Acción : confirmación resultado. Requisito de datos de salida: ninguno.

RDE2.5: Datos de un producto y su valoración

NombreProducto: Cadena de caracteres (20)

RatingProducto: Entero entre el 1 y el 5

RDW2.5: Mismos que RDE2.5

**Restricciones semánticas del subsistema 2.**

RS2.1: NombreProducto, Cantidad y FechaCaducidad

RF: RF2.1

RD(s): RDW2.1

**Descripción**: sí NombreProducto no existe o está vacío ó Cantidad es menor que 1 ó FechaCaducidad es anterior a la fecha actual ó el formato es incorrecto no se realiza ninguna inserción y se devuelve un error.

RS2.2: NombreProducto y Cantidad

RF: RF2.2

RD(s): RDW2.2

**Descripción**: sí NombreProducto está vacío ó no existe ó Cantidad es menor que 1 no se realiza el borrado y se devuelve un error.

RS2.4: NombreProducto

RF: RF2.4

RD(s): RDR2.4

**Descripción**: sí NombreProducto está vacío ó no existe se devuelve un error.

RS2.5: NombreProducto y RatingProducto

RF: RF2.5

RD(s): RDW2.5

**Descripción**: sí NombreProducto está vacío ó no existe ó RatingProducto no tiene un valor entre 0-5 no se realiza ninguna inserción y se devuelve un error.

## **3.Gestión de mesas: (Amador Carmona Méndez)**

**RF3.1: Reservar/liberar de mesa**

Entrada: Agente Externo: Camarero. Acción: Solicitar información de mesa. Requisito de datos de entrada RDE3.1

BD: Requisito de datos de lectura RDR.3.1 Requisito de datos de escritura RDW3.1

Salida: Agente Externo: Camarero. Acción: Nuevo estado de la mesa. Requisito de datos de salida RDS3.1

RDE3.1: Identificador de la mesa

RDR.3.1: Obtener el estado de la mesa

Estado: booleano

RDW3.1: Cambiar estado de la mesa

RDS3.1: Nuevo estado de la mesa

**RF3.2: Mostrar un plato**

Entrada: Agente Externo: Camarero. Acción: Solicitar información del plato. Requisito de datos de entrada RDE3.2

BD: Requisito de datos de lectura RDR3.2

Salida: Agente Externo: Camarero. Acción: Información de un plato. Requisito de datos de salida RDS3.2

RDE3.2: Identificador del plato

RDR3.2: Informacion del plato

Ingredientes: cadena de caracteres(40)

Precio: entero

Valoración: entero

RDS3.2: Información del plato (ingredientes, precio,valoración)

**RF3.3: Mostrar carta (listado de productos)**

Entrada: Agente Externo: Camarero. Acción: Solicitar información de la carta. Requisito de datos de entrada RDE3.3

BD: Requisito de datos de lectura RDR3.3

Salida: Agente Externo: Camarero. Acción: Información de la carta. Requisito de datos de salida RDS3.3

RDE3.3: Identificador de la carta

RDR3.3: Información de la carta

ListadoPlatos: array de platos

RDS3.3: Información de la carta

**RF3.4: Quitar mesa**

Entrada: Agente Externo: Camarero. Acción: Solicitar borrado de una mesa al listado de mesas. Requisito de datos de entrada RDE3.4

BD: Requisito de datos de escritura RDW3.4

Salida: Agente Externo: Camarero.Acción: Confirmar borrado. Requisito de datos de salida: ninguno.

RDE3.4: Identificador de la mesa.

RDW3.4: Datos almacenados de la mesa, los mismos que el RDE3.5.

**RF3.5: Añadir mesa**

Entrada: Agente Externo: Camarero. Acción: Solicitar inserción de una mesa nueva al listado de mesas. Requisito de datos de entrada RDE3.5

BD: Requisito de datos de escritura RDW3.5

Salida: Agente Externo: Camarero. Acción: Confirmar inserción. Requisito de datos de salida: ninguno.

RDE3.5: Datos de entrada de la mesa:

Identificador de la mesa(un entero)

Número de clientes de la mesa(un entero mayor o igual que 0)

Estado(booleano (reservada o no reservada))

RDW3.5: Datos almacenados de la mesa, los mismos que el RDE3.5.

### **Restricciones semánticas del subsistema 3:**

RS3.1: Los identificadores de las mesas son únicos y no nulos, no puede haber dos mesas con el mismo identificador

RF: RF3.5.

RD(s): RDW3.5 Si lo comprueba el SGBD si no se añadiría RDE3.5

**Descripción**: Si ya hay una mesa con el mismo identificador, se devuelve un error y no se añade

RS3.2: El estado de las mesas será por defecto no reservado

RF:RF3.5.

RD(s)**:** RDW3.5 Si lo comprueba el SGBD si no se añadiría RDE3.5

**Descripción:** Siempre al crear una mesa, el estado de la mesa será no reservado.

RS3.3: El número de clientes de la mesa debe ser mayor o igual a 0

RF:RF3.5

RD(s): RDW3.5 Si lo comprueba el SGBD si no se añadiría RDE3.5

**Descripción:** el número de clientes por mesa debe ser 0 o mayor que cero

## **4.Gestión de proveedores: (Carlos Moreno Villarrubia)**

**RF4.1: Dar de alta proveedor**

Entrada: Agente externo: gestor. Acción: solicitar inserción. Requisito de datos de entrada RDE4.1.

BD: Requisito de datos de escritura RDW4.1.

Salida: Agente externo: gestor. Acción: confirmación resultado.

Requisito de datos de salida: RDS4.1.

RDE4.1: Datos de entrada de alta de proveedor

NombreProveedor: Cadena de caracteres (20)

Teléfono: Cadena de caracteres (20)

CorreoElectronico: Cadena de caracteres(40)

RDW4.1: Datos almacenados de contacto.

Los mismos datos que RDE4.1

IDProveedor: entero

RDS4.1: ID autogenerado.

IDProveedor: entero

**RF4.2: Dar de baja proveedor**

Entrada: Agente externo: gestor. Acción: solicitar borrado.

Requisito de datos de entrada RDE4.2.

BD: Requisito de datos de escritura RDW4.2.

Salida: Agente externo: gestor. Acción: confirmación resultado.

Requisito de datos de salida: ninguno.

RDE4.2: Datos de entrada de baja de contacto

IDProveedor: entero

RDW4.2: Datos almacenados de contacto.

Los mismos datos que RDW4.1

**RF4.3: Mostrar información proveedor**

Entrada: Agente externo: gestor. Acción: solicitar listado.

Requisito de datos de entrada: RDE4.3.

BD: Requisito de datos de lectura RDR4.3

Salida: Agente externo: gestor. Acción: confirmación resultado.

Requisito de datos de salida: RDS4.3

RDE4.3: Datos de identificacion de proveedor.

IDProveedor: entero

RDR4.3: Datos de contacto almacenado.

Los mismos datos que RDW4.1.

RDS4.3: Datos de contacto almacenados.

Los mismos datos que RDW4.1.

**RF4.4: Hacer pedido**

Entrada: Agente externo: gestor. Acción: solicitar inserción. Requisito de datos de entrada RDE4.4.

BD: Requisito de datos de escritura RDW4.4.

Salida: Agente externo: gestor. Acción: confirmación resultado.

Requisito de datos de salida: ninguno.

RDE4.4: Datos de entrada del pedido

NombreProveedor: Cadena de caracteres (20)

IDPedido: Cadena de caracteres (20)

Listado de registros cada uno con

NombreProducto: cadena de caracteres (20)

CantidadPedidoProducto: entero.

RDW4.4: Datos a almacenar del pedido.

Los mismos datos que RDE4.4

**RF4.5: valoración proveedor**

Entrada: Agente externo: gestor. Acción: solicitar inserción. Requisito de datos de entrada RDE4.5.

BD: Requisito de datos de escritura RDW4.5 y lectura RDR4.5.

Salida: Agente externo: gestor. Acción: confirmación resultado.

Requisito de datos de salida: ninguno.

RDE4.5: Datos de entrada de alta de proveedor

IDProveedor: entero

RatingProveedor: entero

ComentarioProveedor: cadena de caracteres larga (200)

RDR4.5: Datos de lectura de proveedor

IDProveedor: entero

RDW4.5: Datos de valoración y comentario

RatingProveedor: entero

ComentarioProveedor: cadena de caracteres larga (200)

### **Restricciones semánticas del subsistema 4.**

RS4.1: El identificador de proveedor es único y no nulo

RF:RF4.1

RD(S): RDW4.1

**Descripción**: IDProveedor es un identificador que debe ser único y es autogestionado por la base de datos. Si ya existe un proveedor con ese IDProveedor se devuelve un error.

RS4.2: El identificador de pedido debe corresponder a un único pedido

RF: RF4.4

RD(S): RDW4.4

**Descripción**: “Si ya existe un pedido con el mismo IDPedido, no se inserta de nuevo y se devuelve un error.”

RS4.3: El rating sera entre 0 y 5

RF: RF4.4

RD(S): RDW4.5

**Descripción**: “RatingProveedor debe ser un número entero comprendido entre el 0 y el 5 inclusive”

RS4.4: IDProveedor debe existir

RF: RF4.4

RD(s): RDR4.5 RDW4.5

**Descripción**: “IDProveedor debe existir, en caso contrario se devuelve un error y no se introduce nada.”

## **5.Gestión de pago: (Pelayo)**

**RF5.1: Cobro de mesas**

Entrada: Agente externo : Camarero. Accion : solicitar cuenta.

Requisito de datos de entrada RDE5.1

BD: Requisito de datos de lectura RDR5.1

Salida: Agente externo: Camarero. Accion : confirmación cuenta

Requisito de datos de salida: RDS5.1

RDE5.1: IdentificadorMesa: entero

ListadoProductos:lista de productos

IDPago : entero

RDW5.1: Mismos que RDE5.1

RDR5.1: Mismos que RDE5.1

**RF5.2: Pago personal**

Entrada: Agente externo : Gerente. Accion : solicitar listado de horas, cantidad a pagar por hora y fecha del pago.. Requisito de datos de entrada RDE5.2

BD: Requisito de datos de lectura RDR5.2

Salida: Agente externo: Gerente. Accion : confirmación resultado

Requisito de datos de salida: RDS5.2

RDE5.2: IdentificadorEmpleado : entero

HorasRegistradas : entero

IDPago : entero

RDW5.2: Mismos que RDE5.2

**RF5.3: Pago proveedores**

Entrada: Agente externo: Gerente. Accion: solicitar listado productos proveedor, indicar proveedor, cantidad y fecha del pago . Requisito de datos de entrada RDE5.3

BD: Requisitos de datos de lectura RDR5.3

Salida: Agente externo : Gerente. Acción : confirmación resultado

Requisito de datos de salida: RDS5.3

RDE5.3 : IdentificadorProveedor: entero

PrecioPedido : entero

FechaPedido: Fecha (dd/mm/yyyy)

IDPago : entero

RDW5.3: Mismos que RDE5.3

**RF5.4: Consulta factura**

Entrada: Agente externo: Camarero. Accion: solicitar listado cuenta. Requisito de datos de entrada RDE5.4

BD: Requisitos de dato de lectura RDR5.4

Salida: Agente externo: Camarero. Acción : confirmación resultado

Requisito de datos de salida:RDS5.4

RDE5.4 : IdentificadorMesa : entero

IDPago : entero

RDR5.4: Mismos que RDE5.4

RDS5.4: ListadoProductos: Lista de productos

PrecioTotal: entero

**RF5.5: Pago alquiler**

Entrada: Agente externo: Gerente. Acción : solicitar cantidad a pagar y fecha

del pago.

Requisito de datos de entrada RDE5.5

BD: Requisito de dato de lectura:RDR5.5

Salida: Agente externo: Gerente. Acción: confirmación resultado

Requisito de datos de salida: RDS5.5

RDE5.5 : FechaPago : Fecha (dd/mm/yyyy)

CuotaMensual : entero

IDPago : entero

RDW5.5: Mismos que RDE5.5

### **Restricciones semánticas del subsistema 5:**

RS5.1:Una mesa solo puede ser cobrada por un único camarero

RF:RF5.1

RD(s): RDW5.1

**Descripción :** Un camarero no puede cobrar una mesa que ya ha sido cobrada por otro camarero

RS5.2:La cantidad a pagar no sea nulo

RF:RF5.2

RD(s): RDW5.2

**Descripción :** Un camarero no puede cobrar una cantidad que sea 0

RS5.3:La cantidad a pagar no sea nulo

RF:RF5.3

RD(s): RDW5.3

**Descripción :** Un proveedor no puede cobrar una cantidad que sea 0

RS5.5:La cantidad a pagar no sea nula

RF:RF5.5

RD(s): RDW5.5

**Descripción :** El precio del alquiler no puede ser 0

# 3.Esquemas

## DFD CAJA NEGRA

## 

## DFD0 Armazon

## 

### DFD1 Gestión de personal

### DFD2 Gestión de productos

### 

### DFD3 Gestión de mesas

### DFD4 Gestión de proveedores

### 

### DFD5 Gestion de pago

### 

### EE1 Gestión de personal

### 

### EE2 Gestion de productos

### EE3 Gestion de mesas

### 

### EE4 Gestión de proveedores

### 

### EE5 Gestión de pago

### 

## ESQUEMA EXTERNO GENERAL

## 

## ENTIDAD RELACIÓN

## 

## ﻿

# 4.Paso a Tablas

**Tablas de las Entidades:**

1. **Mesa**(IDMesa,numClientes,estado)
2. **Producto**(NombreProducto,valoracion,cantidad,listadoProductos,fechacaducidad)
3. **Empleado**(DNI,nombre,apellidos,horaEntrada,horaSalida,telefono,cargo)
4. **Pago**(idPago, cantidad)
5. **Proveedor**(idProveedor, nombreProveedor,rating,telefono,correoElectronico,comentarioProveedor)
6. **Local**(idLocal)

**Tablas de Relaciones:**

1. Tiene(IDMesa(CE:1), NombreProducto(CE:2))
2. sirve(IDMesa(CE:1), DNI(CE:3))
3. gestiona(NombreProducto(CE:2), DNI(CE:3))
4. cobra(DNI(CE:3),IDPago(CE:4))
5. obtiene(IDMesa(CE:1),IDPago(CE:4))
6. alquiler(IDPago(CE:4), IDLocal(CE:6))
7. recibe(IDPago(CE:4), IDProveedor(CE:5))
8. abastece(NombreProducto(CE:2),IDProveedor(CE:5))

**Relaciones:**

* Relación **tiene**: una mesa tiene muchos productos,un producto tiene una mesa. relación 1-M
* Relación **sirve**:una mesa es servida por un empleado y un empleado puede servir varias mesas. relación 1-M
* Relación **obtiene**: una mesa obtiene un pago y varios pagos se obtienen de una mesa . relación 1-M
* Relación **cobra**: un empleado cobra varios pagos y un pago se es cobrado a un empleado. relación M-1
* Relación **gestiona**: un empleado gestiona muchos productos y un producto es gestionado por un empleado. relación 1-M
* Relación **valora**: un cliente valora muchos producto y un producto es valorado por muchos clientes. relación M-M
* Relación **abastece**: un proveedor abastece muchos productos pero un producto solo es abastecido por un proveedor. relación 1-M
* Relación **recibe**: se reciben pagos de muchos proveedores y un proveedor paga múltiples veces. relación M-M
* Relación **alquiler**: se realiza un pago a un único local y local puede ser alquilado por varios pagos. relación 1-M

**Dependencias:**

Mesa:

IDMesa→numClientes, IDMesa→estado

Producto:

NombreProducto→valoracion, NombreProducto→cantidad, NombreProducto→listadoProductos, NombreProducto→fechacaducidad

Empleado:

DNI→nombre, DNI→apellidos, DNI→horaEntrada, DNI→horaSalida, DNI→telefono, DNI→cargo

Pago:

IDPago→cantidad

Proveedor:

IDProveedor→nombreProveedor, IDProveedor→rating, IDProveedor→telefono, IDProveedor→correoElectronico, IDProveedor→comentarioProveedor