



DDSI 3ºC

DISEÑO Y DESARROLLO DE SISTEMAS DE INFORMACIÓN

REALIZADO POR:

María Martín Arias
Beatriz Cortés Contreras
Francisco Fernández Millán
Víctor Vallecillo Morilla

INDICE

1. Objetivos.....	2
2. Requisitos funcionales.....	2
3. Requisitos de datos.....	4
4. Restricciones semánticas.....	5
5. Diseño de diagramas.....	6
5.1. Diagrama caja negra	
5.2. DFD0	
5.3. Diagramas del área funcional: Gestión de datos de restaurantes	
5.4. Diagramas del área funcional: Gestión de datos de usuarios	
5.5. Diagramas del área funcional: Gestión de Menús y Platos	
5.6. Diagramas del área funcional: Gestión de pedidos y pagos.	
6. Diseño lógico	
E/R.....	16
7. Paso a tablas.....	16
8. Diagramas DDL.....	17
9. Código BASE DE DATOS: Creación de tablas.....	20
10. Código BASE DE DATOS: Creación de insert.....	21
11. Código BASE DE DATOS: Disparadores.....	22
12. Seguimiento de la implementación.....	22

Sistema de información y gestión de comida rápida/online

La idea general del proyecto será la gestión de pedidos de comida online. Se trata de diseñar un software que se encargue de la gestión de pedidos de comida a través de una aplicación. Comenzaremos definiendo los requisitos funcionales, así como requisitos de datos y restricciones semánticas.

Seguidamente diseñaremos los diagramas que, a través de un refinamiento gradual, nos darán la visión exacta de la aplicación y sus funciones.

Finalmente implementaremos dicho código, utilizando en este caso **Sql Developer** de Oracle para la base de datos y utilizando **NetBeans** para programar en lenguaje Java.

En cuanto a la división de responsabilidades, nuestras áreas funcionales han sido divididas de la siguiente forma:

Francisco Fernández Millán → *Gestión de restaurantes*

Beatriz Cortés Contreras → *Gestión de usuarios*

María Martín Arias → *Gestión de menús y platos*

Víctor Vallecillo Morilla → *Gestión de pedidos y pagos*

1. Objetivos

El sistema deberá gestionar y almacenar la información de los restaurantes que se hayan inscrito como usuarios, y de los clientes.

Gestión de reservas en restaurantes y de pedidos de comida a domicilio.

Solo el administrador del sistema podrá dar de alta a los restaurantes (usuarios).

Solo el administrador del sistema podrá dar de baja a los clientes.

Cualquier persona podrá entrar en nuestro sistema y consultar la información de los restaurantes (usuarios).

Solo los clientes registrados podrán realizar reservas y pedidos de comida.

2. Requisitos funcionales

R.F 1 Gestión de restaurantes. *Se permitirá dar de alta/baja a restaurantes en el sistema, así como consultar y/o modificar cualquiera de sus datos*

R.F 1.1 Alta restaurante. *Se dará de alta a un nuevo restaurante, con la información proporcionada.*

- Entrada: **RD1**
- Manejo: Se da de alta un restaurante en el sistema.
- Salida: Ninguna.

R.F 1.2 Baja restaurante. *Se eliminará del sistema toda la información relativa al restaurante.*

- Entrada: Identificación restaurante.
- Manejo: Se da de baja un restaurante en el sistema.
- Salida: Ninguna.

R.F 1.3 Consulta de restaurantes. *Se mostrarán los restaurantes que cumplen los requisitos de búsqueda.*

- Entrada: **RD5**
- Manejo: Se mostrará los restaurantes relacionados con dicha búsqueda. **RD5.1, RD5.2**
- Salida: Listado de restaurantes.

R.F 2 Gestión de clientes. *Se permitirá dar de alta/baja clientes, que usaran la aplicación en el sistema.*

R.F 2.1 Alta clientes. *Se dará de alta a nuevos usuario, con la información proporcionada en la fase de registro.*

- Entrada: **RD2**
- Manejo: Se da de alta un cliente en el sistema.
- Salida: Ninguna.

R.F 2.2 Baja de cliente. *Se eliminará del sistema toda la información relativa al cliente.*

- Entrada: Identificación cliente.
- Manejo: Se da de baja un cliente en el sistema.
- Salida: Ninguna.

R.F 2.3 Modificación de información de cliente. *El cliente podrá modificar su información.*

- Entrada: **RD2**
- Manejo: Se mostrará los campos con posibilidad de modificación.
- Salida: Ninguna.

R.F 3 Gestión de pedidos y pagos. *Se permitirá la solicitud, cancelación y seguimiento de pedidos.*

R.F 3.2 Solicitud de pedidos. *El usuario, que esté registrado, podrá realizar pedidos de comida a restaurantes del sistema.*

- Entrada: Ninguna.
- Manejo: Elección de pedido.
- Salida: Número de pedido.

R.F 3.3 Cancelación de pedidos. *El usuario podrá cancelar los pedidos que haya realizado. Solo si no se ha iniciado la elaboración del pedido.*

- Entrada: Número de pedido proporcionado.
- Manejo: Cancelar un pedido.
- Salida: Ninguna.

R.F. 3.4 Pago en efectivo. *Se preguntará por adelantado si necesita cambio.*

- Entrada: **RD4.1**
- Manejo: Pago de pedido en efectivo.
- Salida: Ninguna

R.F 3.5 Pago electrónico. Tarjeta de crédito, PayPal, etc.

- Entrada: **RD4.2**
- Manejo: Pago de pedido electrónico.
- Salida: Ticket de pago.

R.F. 5 Gestión de menús y platos. Se podrá gestionar la carta de cada restaurante añadiendo menús y platos.

R.F 5.1 Alta de menú. El restaurante como usuario registrado podrá añadir menús a la carta de su restaurante.

- Entrada: **RD3**
- Manejo: El restaurante añade los menús.
- Salida: Ninguna

R.F 5.2 Baja de menú. El restaurante como usuario registrado podrá eliminar menús de la carta de su restaurante.

- Entrada: **RD3**
- Manejo: El restaurante elimina el menú.
- Salida: Ninguna

R.F 5.3 Consulta de menú. Se solicita la carta con los menús y platos que tenga el restaurante.

- Entrada: **RD3**
- Manejo: Ninguno.
- Salida: Carta del restaurante.

3. Requisitos de datos.

R.D 1 Almacenamiento/Modificación de restaurantes

- Nombre restaurante (una cadena de 20 caracteres).
- CIF restaurante (una cadena de 20 caracteres alfanuméricos).
- Nombre propietario (una cadena de 10 caracteres).
- Apellidos propietario (una cadena de 15 caracteres).
- Teléfono (cadena de 20 caracteres numéricos).
- Localización (cadena de 20 caracteres).
- Correo electrónico (una cadena de 50 caracteres alfanuméricos).

R.D 2 Almacenamiento/Modificación de datos de clientes

- Correo electrónico (una cadena de 50 caracteres alfanuméricos).
- Nombre cliente (una cadena de 10 caracteres).
- Apellidos cliente (una cadena de 15 caracteres).
- Teléfono (cadena de 20 caracteres numéricos).
- Edad (cadena de 2 caracteres numéricos)

R.D 3 Almacenamiento/Consulta de carta de menús

Datos necesarios para añadir un nuevo menú a la carta:

- Número de menú (cadena de 3 caracteres numéricos)
- Descripción menú(campo de texto alfanumérico de 255 caracteres)
- Precio(cadena de 5 caracteres numéricos)

R.D 4 Almacenamiento de formas de pago

R.D 4.1 Datos necesarios para realizar el pago en efectivo

- Dinero de pago(cadena de 5 caracteres numéricos)

R.D 4.2 Datos necesarios para realizar el pago electrónico

- Tipo de tarjeta(PAYPAL, MASTERCARD, SAFECARD, VISA)
- Número de tarjeta(cadena de 20 caracteres numéricos)
- Fecha caducidad (de tipo date)
- CVC (cadena de 3 caracteres numéricos)

R.D 5 Realización/Consulta sobre pedidos.

R.D.5.1 Datos necesarios para la correcta localización

- Código postal (cadena de 5 caracteres numérica)
- Dirección

R.D.5.2 Datos sobre el tipo de comida a elegir.

- Asiática, Europea, Africana y Americana

4. Restricciones semánticas

R.S 1 Se necesita un mínimo de 16 años para poder registrarse en el sistema.

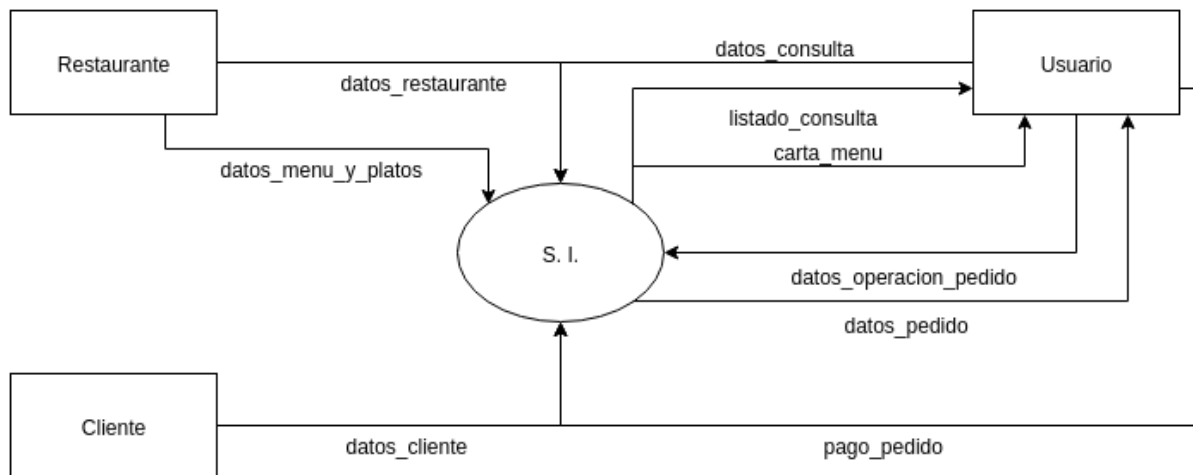
Afecta a RF1.1, RF2.1

R.S 2 A la hora de realizar un pago la tarjeta no puede estar caducada.

Afecta a RF4.2

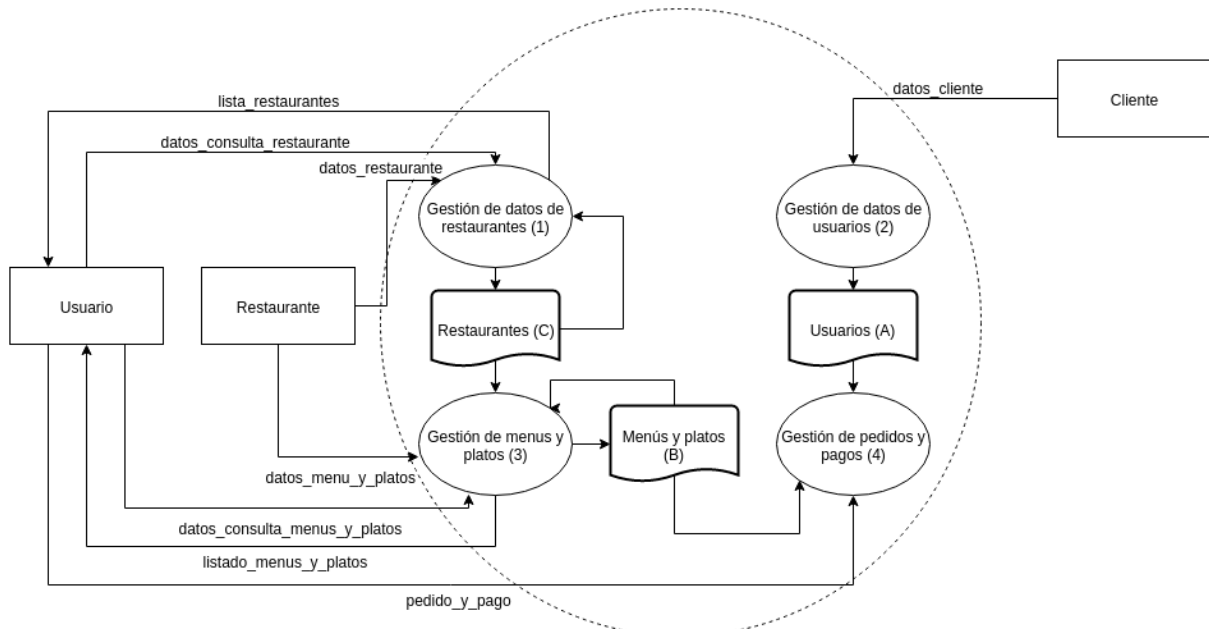
5. Diseño de diagramas

Esquema caja negra:



El restaurante proporciona al sistema sus datos y los de los menús y platos. El usuario proporciona al sistema los datos de consulta y los de operación de pedido, el sistema le devuelve un listado de consulta o los datos del pedido según le hayamos solicitado. También proporciona el usuario el pago del pedido al sistema. El cliente sólo le proporciona al sistema sus datos.

DFDO:



Ya tenemos los flujos planteados en el esquema de caja negra, ahora el sistema desaparece porque lo estamos mirando por dentro, es un divide y vencerás, vamos tratando cada parte de forma independiente enfocándonos específicamente en lo que hay. Se hace desde cero usando las primitivas ascendentes. Las interfaces están fuera de la frontera del sistema (Usuario, Restaurante...)

1.-Gestión de datos de restaurantes (proceso 1), todo lo referido a los restaurantes está aquí dentro. Todos los datos de restaurante van a parar aquí, un usuario ya registrado puede consultar datos sobre los restaurantes que son los *datos_consulta_restaurante* y el sistema le devuelve al usuario una *lista_restaurantes* según su consulta.

2.-Gestión de datos de usuarios (proceso 2), todo lo referido a los clientes está aquí dentro, todos los datos de cliente van a parar aquí. Un usuario ya registrado puede consultar datos sobre los menús y platos que son los *datos_consulta_menús_platos* y el sistema le devuelve al usuario una *lista_menus_platos* según su consulta.

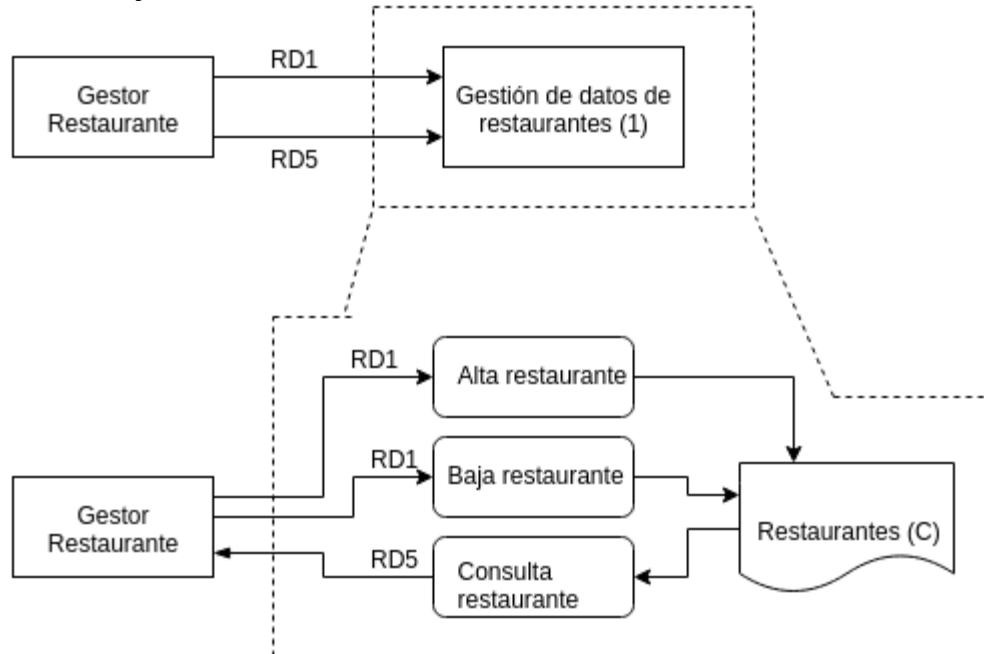
3.-Gestión de menús y platos (proceso 3), todo lo referido a los menús y platos está aquí dentro, todos los datos *datos_menús_platos* van a parar aquí.

4.-Gestión de pedidos y pagos (Proceso 4), todo lo referido a los pedidos y pagos está aquí dentro, aquí el usuario realiza todo lo referente a la cancelación de pedidos etcétera va dentro de pedido y pago sólo creamos un único flujo.

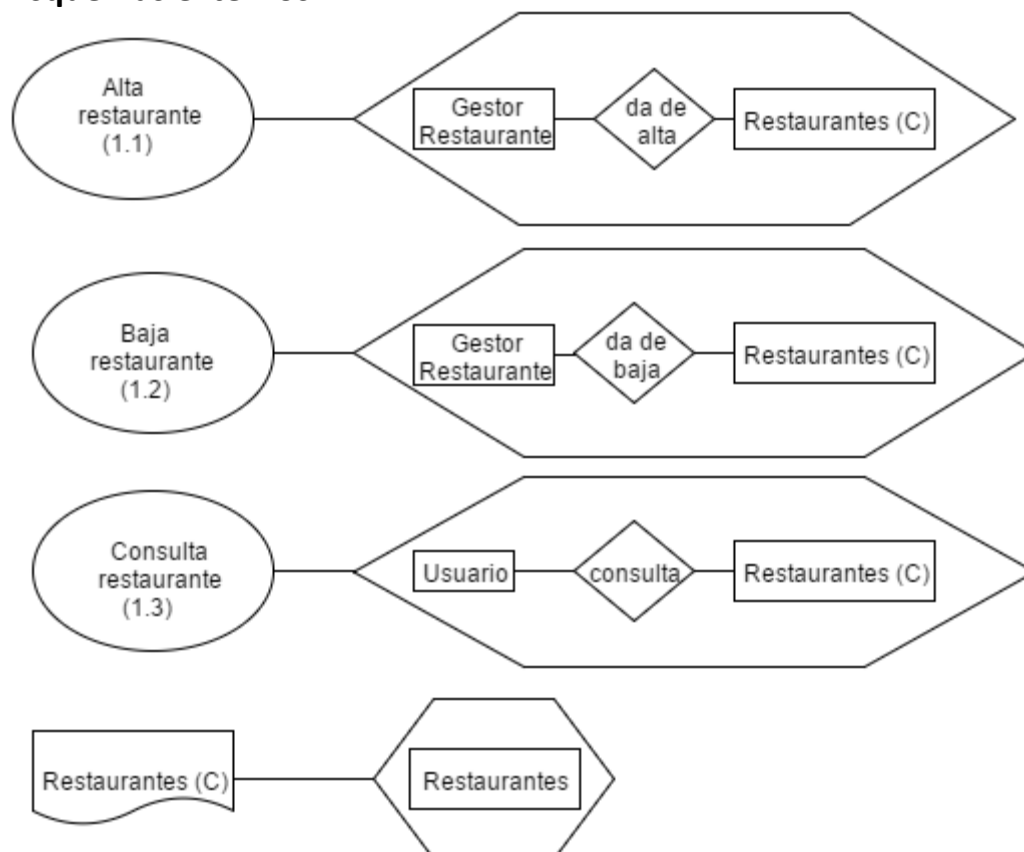
Ahora determinamos específicamente como se comunican unos procesos con otros, es decir, que información tienen unos que requieren los otros. Cuando un usuario se registra hace pedidos y pagos, entonces entre el proceso 2 y el proceso 4 hay un intercambio de información que son los usuarios. Cuando un cliente se registra se convierte en un usuario, cuando un usuario quiere hacer un pedido hay que saber que usuario es, por eso usamos el almacén Usuarios(A) para asociarla al pedido, ya que el pedido asocia un usuario con un menú o con un plato. Cuando se hace un pedido se hace sobre un menú o sobre un plato por eso usamos el almacén Menús y Platos(B) entre el proceso 3 y el proceso 4, con esto se puede dar de alta un menú, dar de alta el plato y leer el menú para asociar el menú al plato, es decir, para registrar la relación. Entre el proceso 1 y el proceso 2 no hay comunicación porque no tienen nada que ver entre ellos salvo satisfacer los pedidos. Entre el proceso 1 y el proceso 3 hay un intercambio que es el almacén Restaurantes(C), lleva un flujo al proceso 1 para poder consultar los restaurantes para el listado.

Área Funcional: Gestión de datos de restaurantes.

Diagrama de flujo de datos:

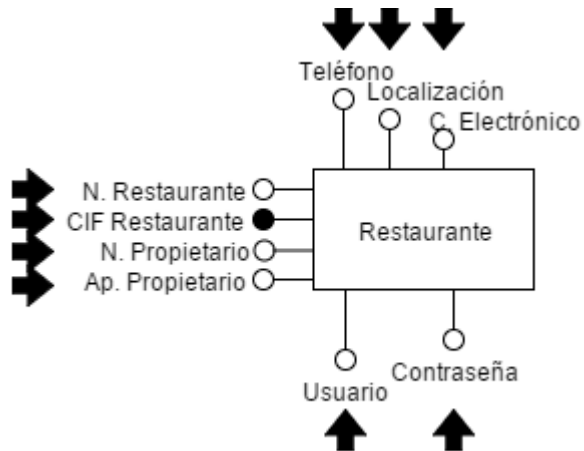


Esquemas externos:

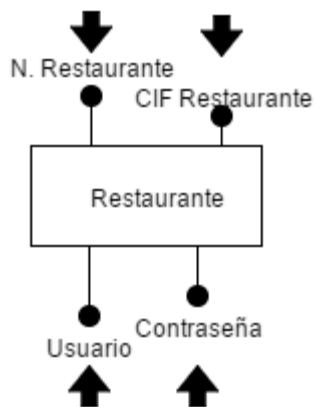


Operaciones de datos:

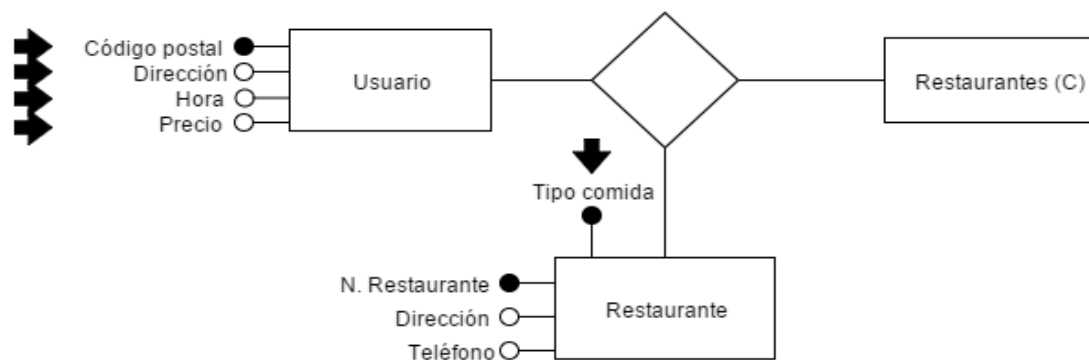
-O1: Alta restaurante.



-O2: Baja restaurante.

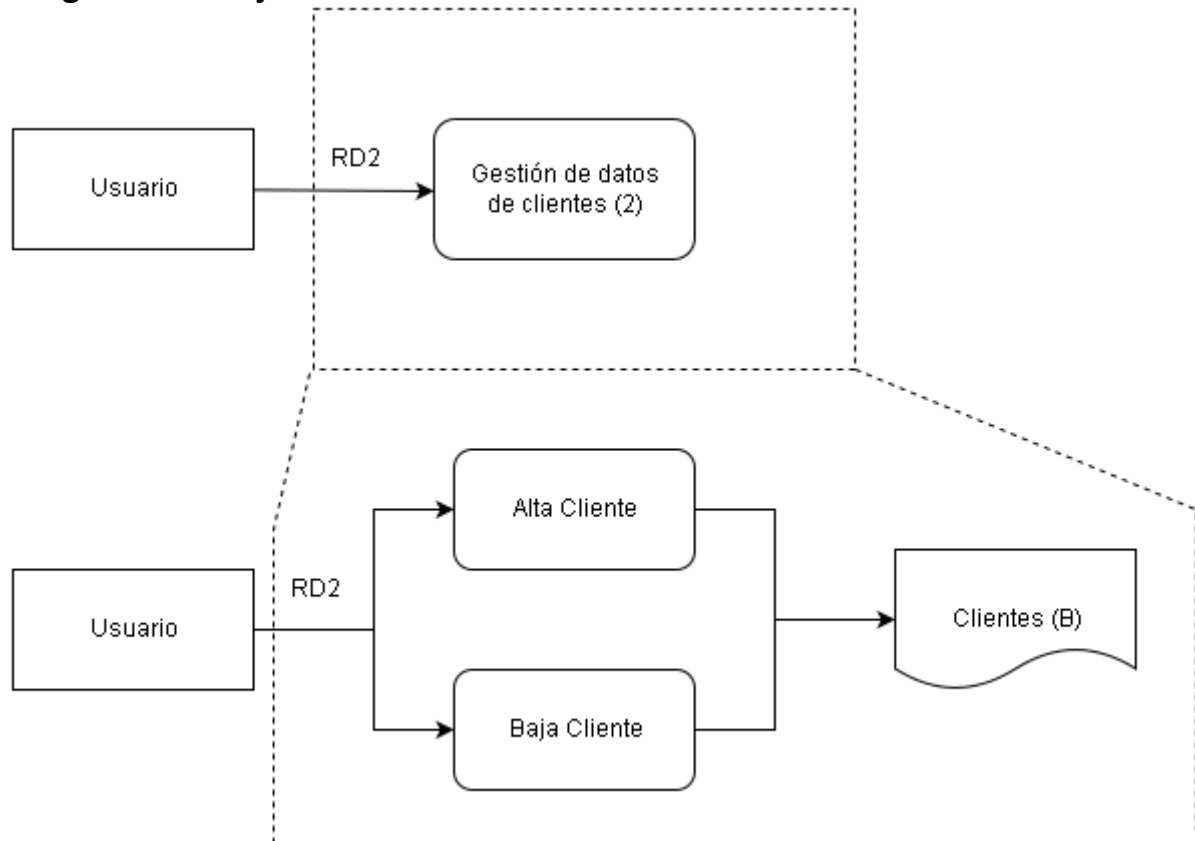


-O3: Consulta restaurante.

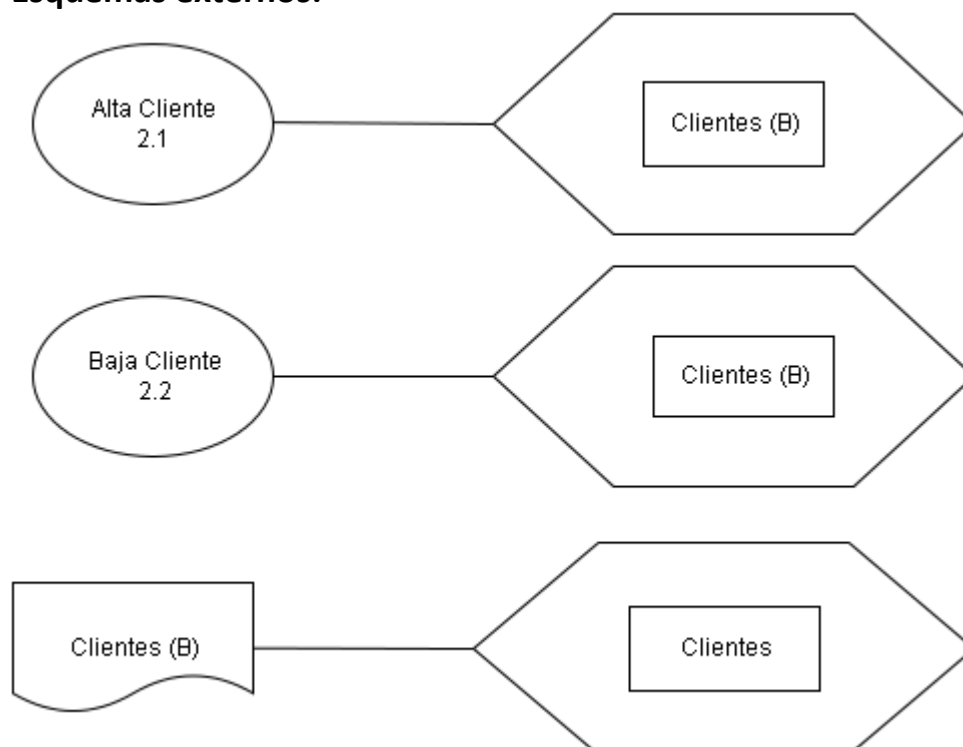


Área Funcional: Gestión de datos de usuarios

Diagrama de flujo de datos:

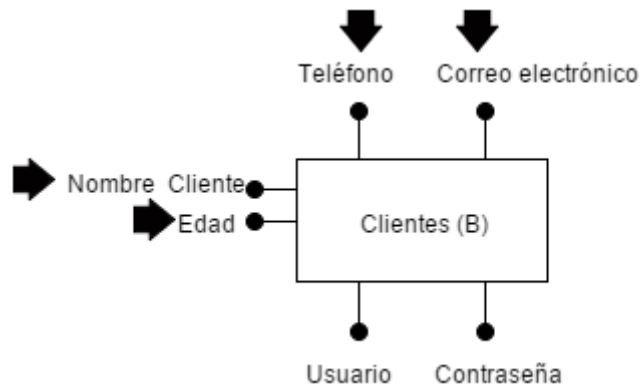


Esquemas externos:

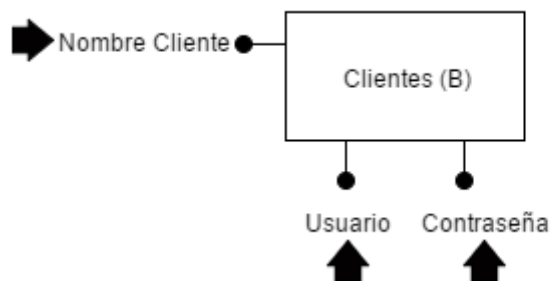


Operaciones de datos:

-O1: Alta cliente.

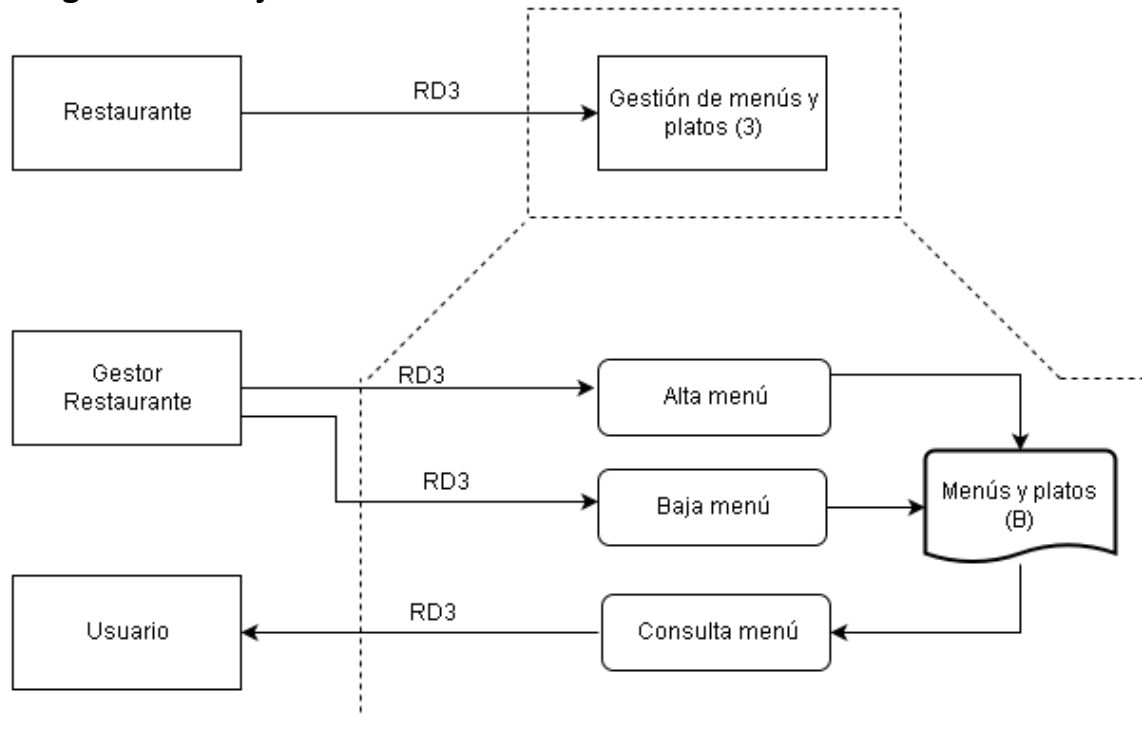


-O2: Baja cliente.

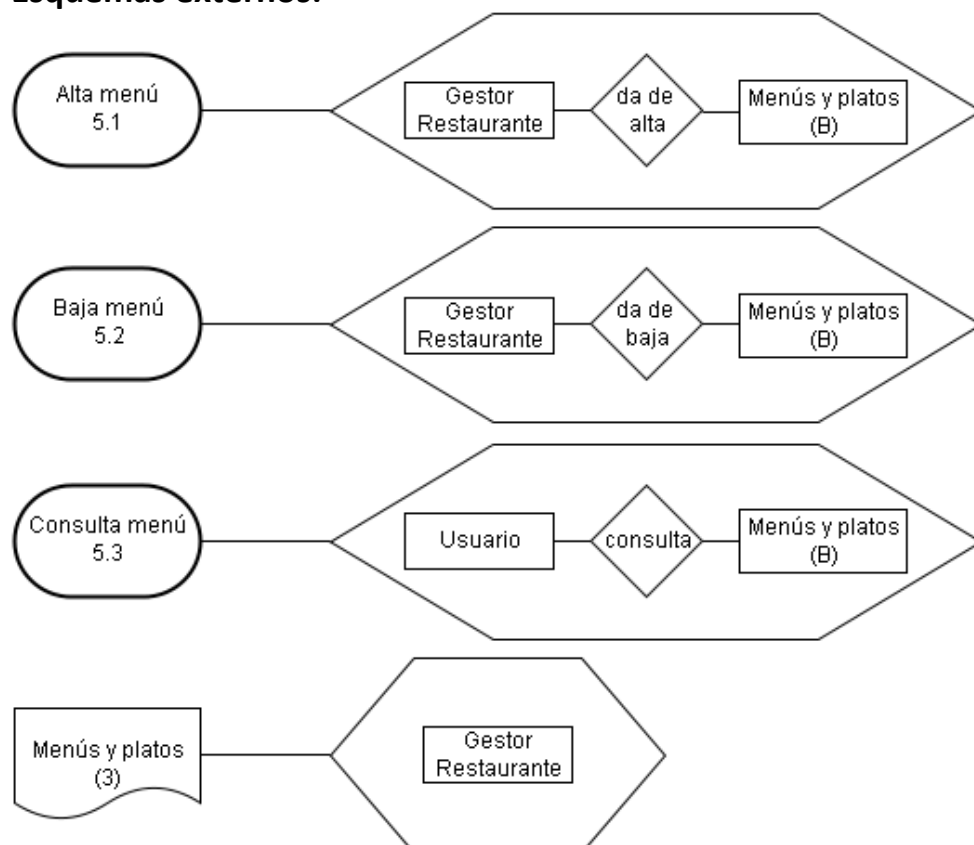


Área Funcional: Gestión de menús y platos

Diagrama de flujo de datos:

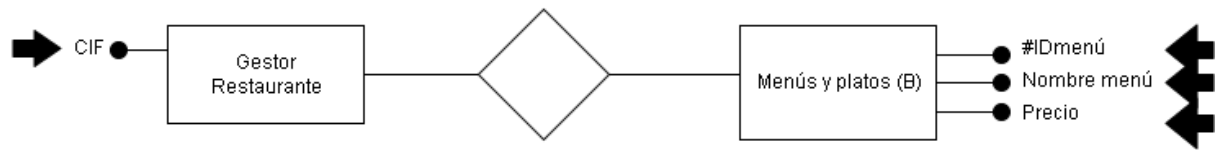


Esquemas externos:

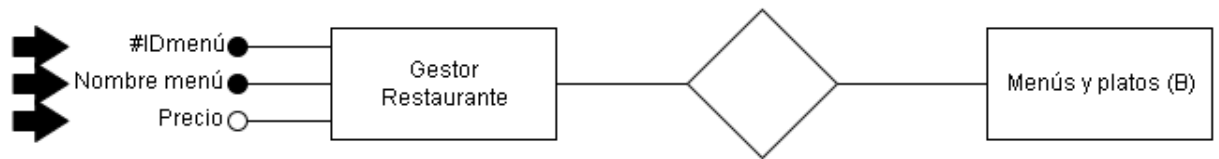


Operaciones de datos:

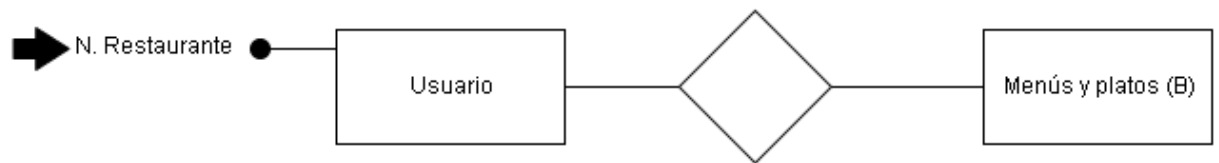
-O1: Alta menú.



-O2: Baja menú.

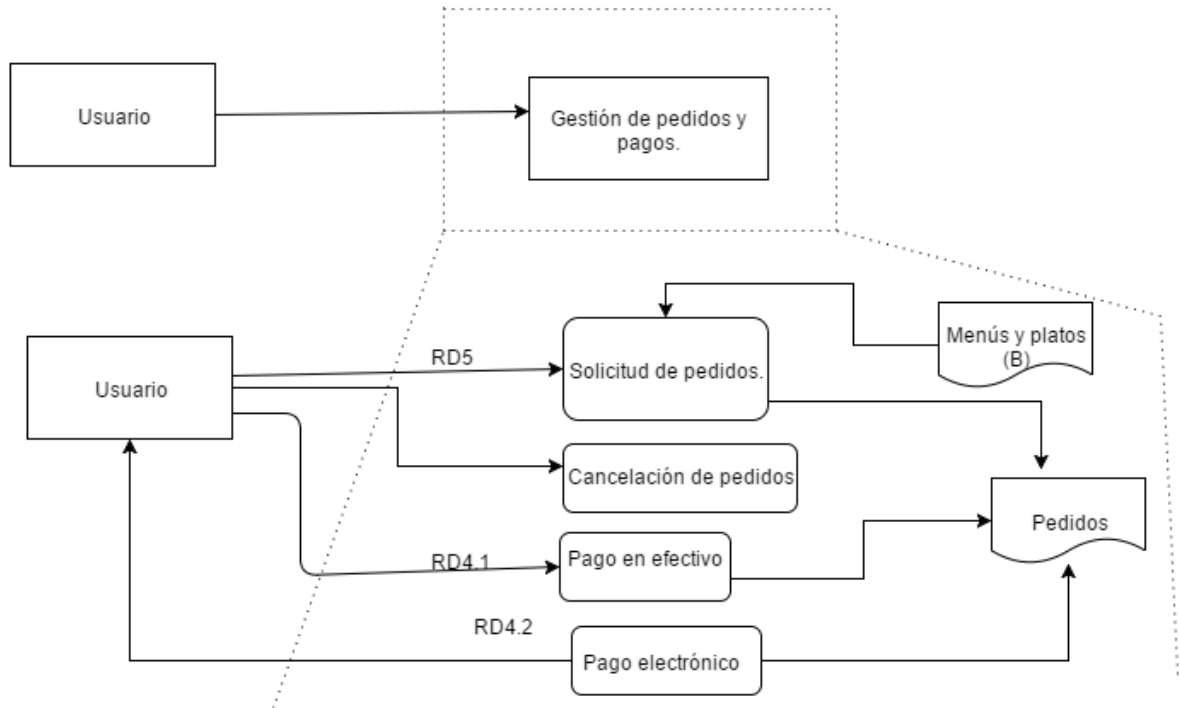


-O3: Consulta menú.

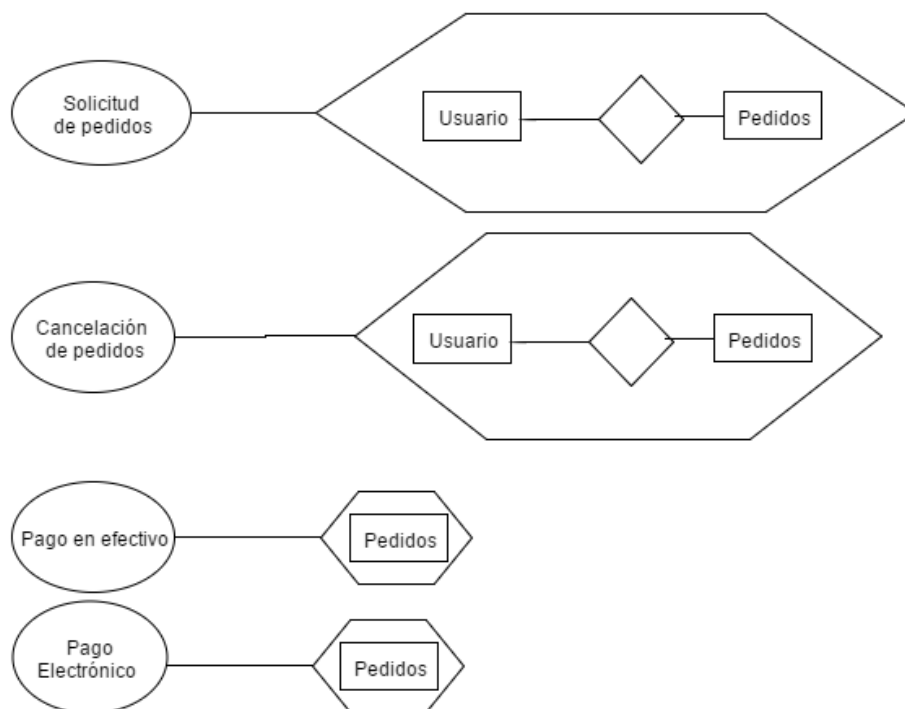


Área Funcional: Gestión de pedidos y pagos

Diagrama de flujo de datos:

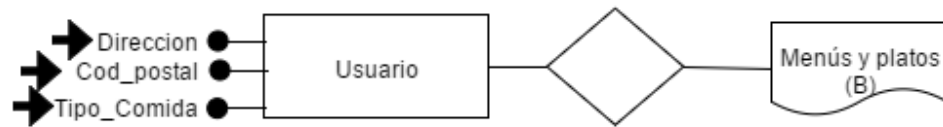


Esquemas externos:

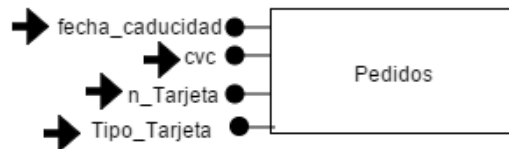


Operaciones de datos:

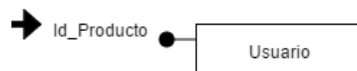
-O1: Consulta menú



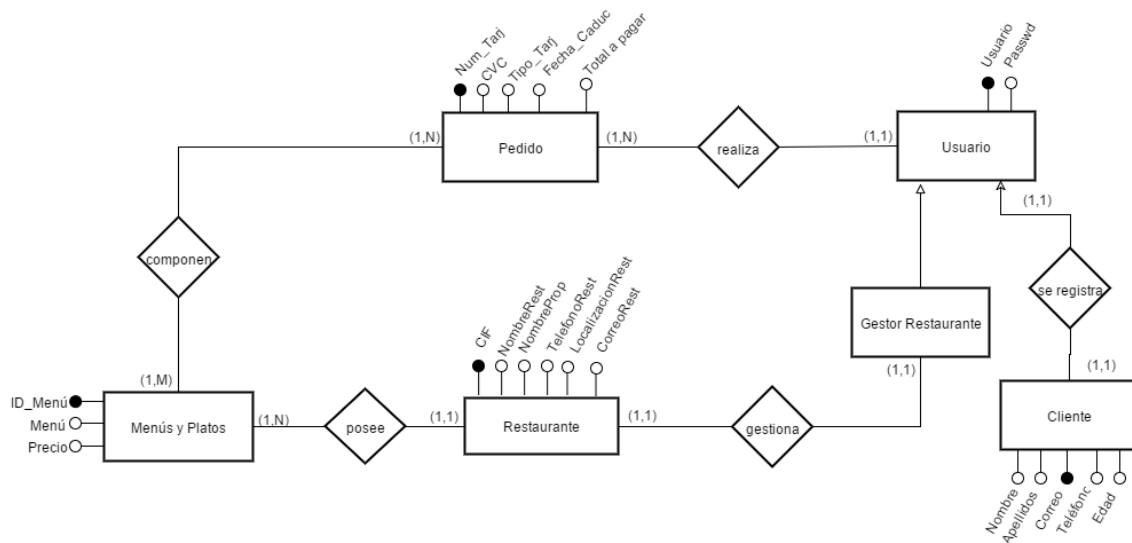
-O2: Realización pedido.



-O3: Cancelación pedido.



DISEÑO LÓGICO: Diagrama E/R



Finalmente y tras todo el refinamiento, llegamos al diagrama entidad-relación final. El diagrama engloba todas las entidades y las respectivas relaciones entre sus entidades. Diferenciamos una entidad *Usuario* que engloba a todos los usuarios registrados en el sistema, estos usuarios se diferencian en dos categorías, ya sean *Clientes* o *Gestor Restaurante*. Se trata de una relación de herencia total y exclusiva, es decir, todos los usuarios deben ser *Clientes* o *Gestor Restaurante*, sin poder ser ambas categorías a la vez.

Por otro lado, cada *Gestor Restaurante* gestiona un único *Restaurante* y dicho restaurante posee uno o varios menús.

Además de todo esto, tenemos otra funcionalidad en la cual un usuario puede realizar un pedido el cual está compuesto por uno o varios menús.

PASO A TABLAS

USUARIO (Usuario, Passwd)

RESTAURANTE (CIF, UsuarioR, NombreRest, NombreProp, TelefonoRest, LocalizacionRest, CorreoRest)

-*UsuarioR*: clave externa de USUARIO

POSEE_MENUSPLATOS (ID_Menu, CIF_Rest, Menú, Precio)

-*CIF*: clave externa de RESTAURANTE

REALIZA_PEDIDO (Num_tarj, CVC, UsuarioPed, Tipo_tarj, Fecha_caduc, TotalAPagar)

-*UsuarioPed*: clave externa de USUARIO

COMPONEN (ID_Menu_Rest, Num_tarj_Cliente)

-*Num_tarj_Cliente*: clave externa de REALIZA_PEDIDO

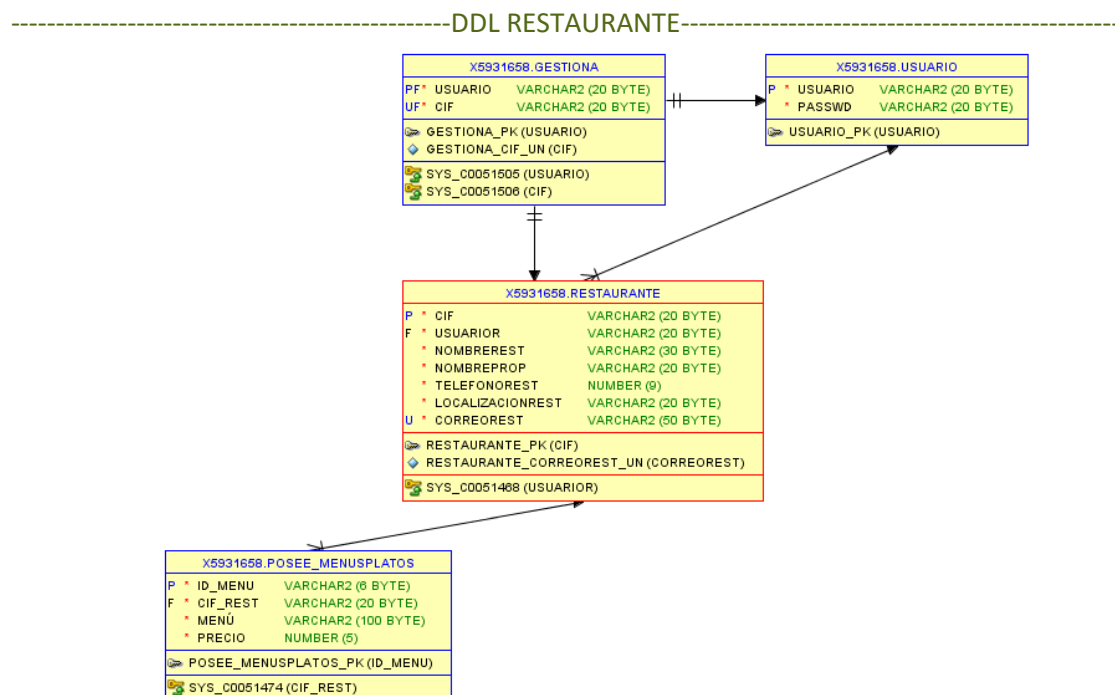
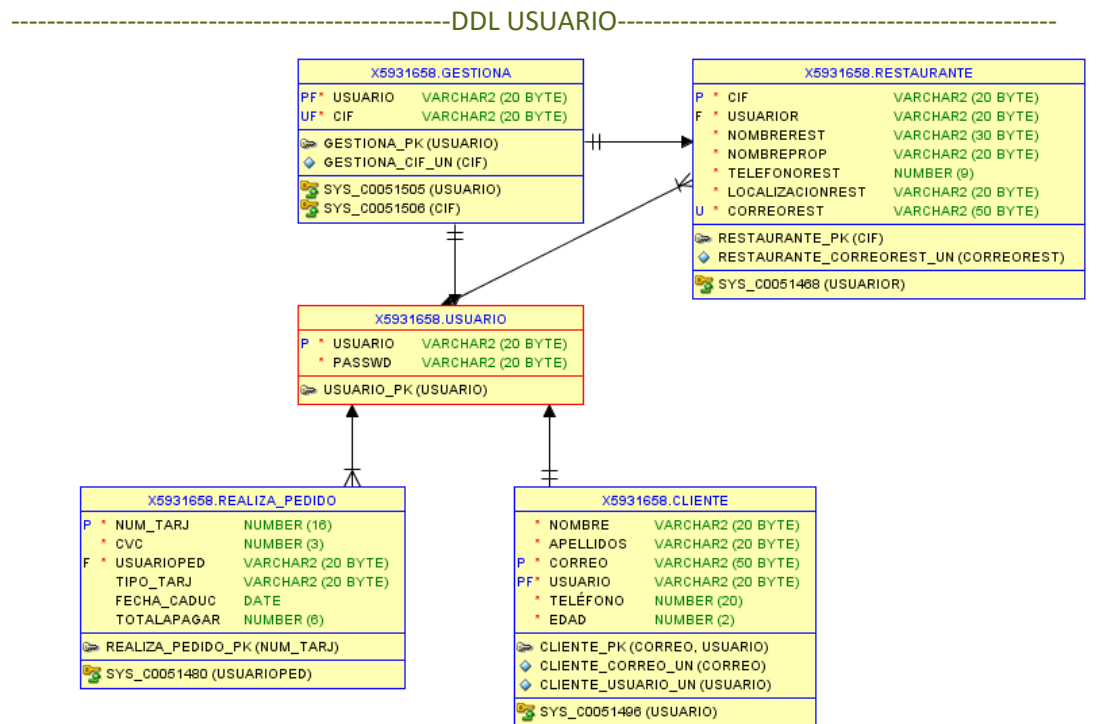
-*ID_Menu_Rest*: clave externa de POSEE_MENUSPLATOS

CLIENTE (Nombre, Apellidos, Correo, Usuario, Teléfono, Edad)
 -*Usuario*: clave externa de USUARIO

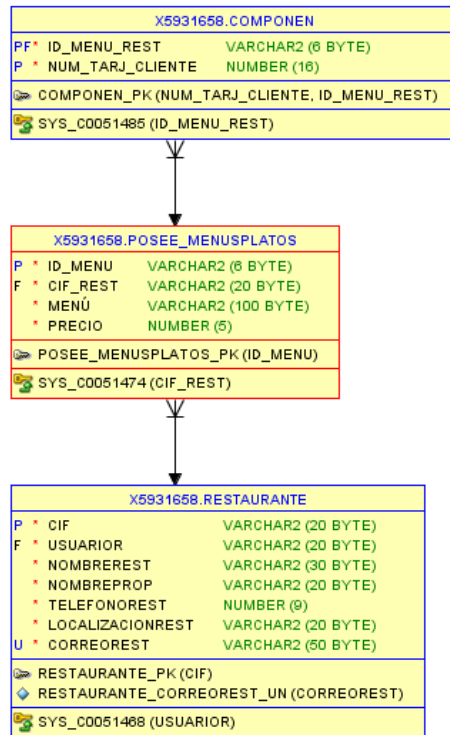
SE_REGISTRA (Usuario, Correo)
 -*Correo*: clave externa de CLIENTE

GESTIONA (Usuario, CIF)
 -*Usuario*: clave externa de USUARIO
 -*CIF*: clave externa de RESTAURANTE

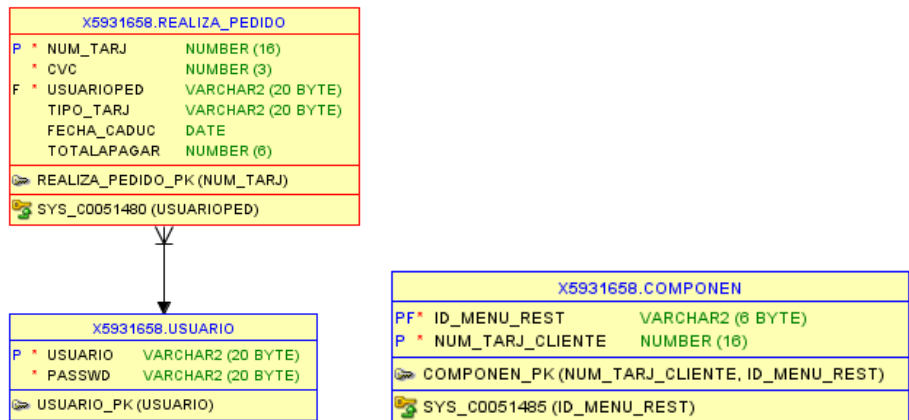
8. Diagramas DDL



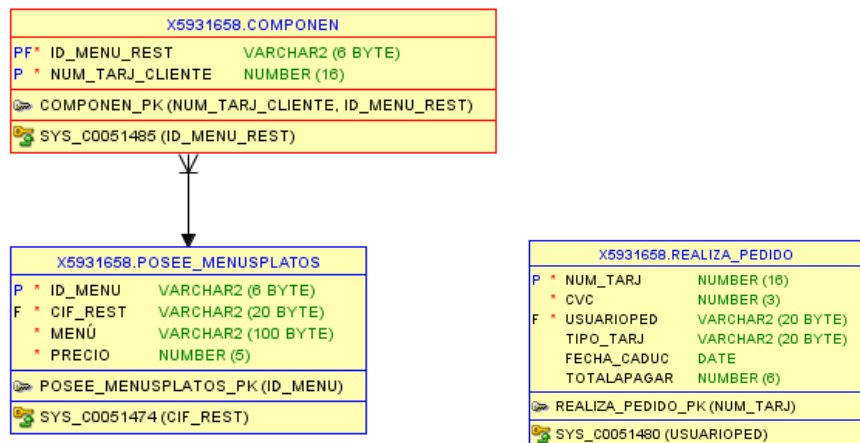
-----DDL POSEE_MENUPLATOS-----



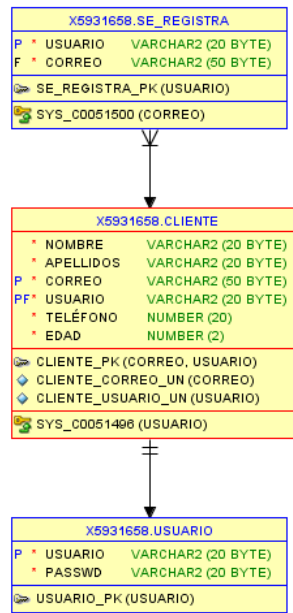
-----DDL REALIZA_PEDIDO-----



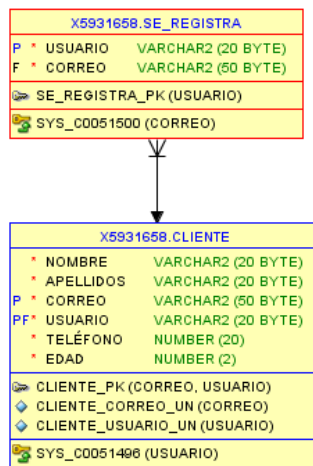
-----DDL COMPONEN-----



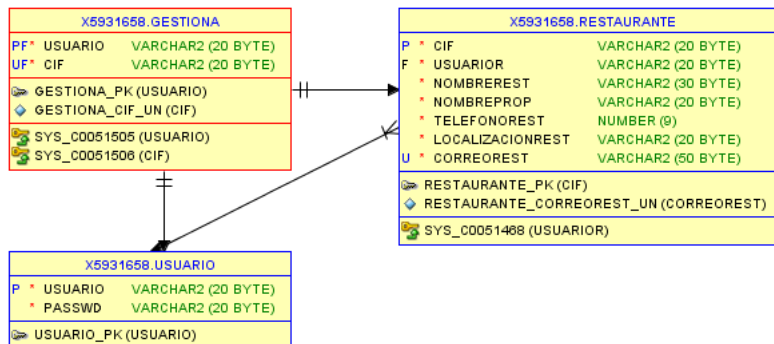
DDL CLIENTE



DDL SE_REGISTRA



DDL GESTIONA



9. Código BASE DE DATOS: Creación de tablas

```
CREATE TABLE USUARIO(  
  Usuario VARCHAR2(20) NOT NULL,  
  Passwd VARCHAR2(20) NOT NULL,  
  PRIMARY KEY (Usuario));
```

```
CREATE TABLE RESTAURANTE(  
  CIF VARCHAR2(20) NOT NULL,  
  UsuarioR VARCHAR2(20) NOT NULL,  
  NombreRest VARCHAR2(30) NOT NULL,  
  NombreProp VARCHAR2(20) NOT NULL,  
  TelefonoRest NUMBER(9) NOT NULL,  
  LocalizacionRest VARCHAR2(20) NOT NULL,  
  CorreoRest VARCHAR2(50) UNIQUE NOT NULL,  
  PRIMARY KEY (CIF),  
  FOREIGN KEY (UsuarioR) REFERENCES USUARIO (Usuario));
```

```
CREATE TABLE POSEE_MENUPLATOS(  
  ID_Menu VARCHAR2(6) NOT NULL,  
  CIF_Rest VARCHAR2(20) NOT NULL,  
  Menú VARCHAR2(100) NOT NULL,  
  Precio NUMBER(5) NOT NULL,  
  PRIMARY KEY (ID_Menu),  
  FOREIGN KEY (CIF_Rest) REFERENCES RESTAURANTE (CIF));
```

```
CREATE TABLE REALIZA_PEDIDO(  
  Num_tarj NUMBER(16) NOT NULL,  
  CVC NUMBER(3) NOT NULL,  
  UsuarioPed VARCHAR2(20) NOT NULL,  
  Tipo_tarj VARCHAR2(20) CHECK(Tipo_tarj in ('PAYPAL', 'MASTERCARD', 'SAFECARD', 'VISA')),  
  Fecha_caduc DATE,  
  TotalAPagar NUMBER(6),  
  PRIMARY KEY (Num_tarj),  
  FOREIGN KEY (UsuarioPed) REFERENCES USUARIO (Usuario));
```

```
CREATE TABLE COMPONENTE(  
  ID_Menu_Rest VARCHAR(6) NOT NULL,  
  Num_tarj_Cliente NUMBER(16) NOT NULL,  
  PRIMARY KEY (Num_Tarj_Cliente, ID_Menu_Rest),  
  FOREIGN KEY (Num_Tarj_Cliente) REFERENCES REALIZA_PEDIDO (Num_Tarj),  
  FOREIGN KEY (ID_Menu_Rest) REFERENCES POSEE_MENUPLATOS (ID_Menu));
```

```
CREATE TABLE CLIENTE (  
  Nombre VARCHAR2(20) NOT NULL,  
  Apellidos VARCHAR2(20) NOT NULL,  
  Correo VARCHAR2(50) UNIQUE NOT NULL,  
  Usuario VARCHAR2(20) UNIQUE NOT NULL,  
  Teléfono NUMBER(20) NOT NULL,  
  Edad NUMBER(2) NOT NULL CHECK (Edad >= 0),  
  PRIMARY KEY (Correo, Usuario),  
  FOREIGN KEY (Usuario) REFERENCES USUARIO (Usuario));
```

```
CREATE TABLE SE_REGISTRA(
Usuario VARCHAR2(20) NOT NULL,
Correo VARCHAR2(50) NOT NULL,
PRIMARY KEY (Usuario),
FOREIGN KEY (Correo) REFERENCES CLIENTE (Correo));
```

```
CREATE TABLE GESTIONA(
Usuario VARCHAR2(20) NOT NULL,
CIF VARCHAR2(20) UNIQUE NOT NULL,
PRIMARY KEY (Usuario),
FOREIGN KEY (Usuario) REFERENCES USUARIO (Usuario),
FOREIGN KEY (CIF) REFERENCES RESTAURANTE (CIF));
```

10. Código BASE DE DATOS: Creación de insert

```
Insert into USUARIO values ('seer24', 'seer658471');
Insert into USUARIO values ('NataliaPM', 'npm19051996');
Insert into USUARIO values ('antonio1998', '625487521');
Insert into USUARIO values ('AguilasRestaurant', 'AR1986');
Insert into USUARIO values ('SanNicolas26', 'SN_1990');
Insert into USUARIO values ('ElBuenGusto', '125BuenGusto');
```

```
Insert into RESTAURANTE values (74587412, 'AguilasRestaurant', 'Las Aguilas Restaurant', 'Paco', 958441236,
'Albolote', 'LasAguilas@gmail.es');
Insert into RESTAURANTE values (54785216, 'SanNicolas26', 'El mirador San Nicolás', 'Trinidad', 958432147,
'Granada', 'SanNicolas26@hotmail.com');
Insert into RESTAURANTE values (74596487, 'ElBuenGusto', 'El buen gusto', 'Maria Jesús', 958432655, 'Granada',
'ElBuenGusto@gmail.es');
```

```
Insert into POSEE_MENUSPLATOS values ('M001', 74587412, 'Primer plato: verduras; Segundo plato: carne; Postre y
bebida;', '15');
Insert into POSEE_MENUSPLATOS values ('M002', 74587412, 'Primer plato: pescado; Segundo plato: carne; Postre y
bebida;', '21');
Insert into POSEE_MENUSPLATOS values ('BG1', 74596487, 'Primer plato: carne; Segundo plato: guisado; Postre y
bebida;', '12');
Insert into POSEE_MENUSPLATOS values ('gr01', 54785216, 'Primer plato: pasta; Segundo plato: fritura; Postre y
bebida;', '9');
```

```
Insert into REALIZA_PEDIDO values ('4125478541111111', '745', 'seer24', 'PAYPAL', '10/12/2018', '25');
Insert into REALIZA_PEDIDO values ('1547154736589547', '124', 'NataliaPM', 'MASTERCARD', '02/12/2019', '18');
Insert into REALIZA_PEDIDO values ('3333254785479688', '995', 'antonio1998', 'VISA', '01/06/2020', '7');
```

```
Insert into COMPONENTEN values ('M001', '4125478541111111');
Insert into COMPONENTEN values ('BG1', '1547154736589547');
Insert into COMPONENTEN values ('M002', '3333254785479688');
```

```
Insert into CLIENTE values ('Sergio', 'Fernandez Lopez', 'seer24@gmail.es', 'seer24', 658471258, '20');
Insert into CLIENTE values ('Natalia', 'Perez Martin', 'NataliaP@hotmail.com', 'NataliaPM', 695474752, '32');
Insert into CLIENTE values ('Antonio', 'Torres Jimenez', 'antonio1998@gmail.es', 'antonio1998', 958406874, '18');
```

```
Insert into SE_REGISTRA values ('seer24', 'seer24@gmail.es');
Insert into SE_REGISTRA values ('NataliaPM', 'NataliaP@hotmail.com');
Insert into SE_REGISTRA values ('antonio1998', 'antonio1998@gmail.es');
```

```
Insert into GESTIONA values ('AguilasRestaurant', 74587412);
Insert into GESTIONA values ('SanNicolas26', 54785216);
Insert into GESTIONA values ('ElBuenGusto', 74596487);
```

11. Código BASE DE DATOS: Disparadores

DISPARADOR 1 → Nuestro primer disparador será “*seguridadTarjeta*”. Dicho disparador se encargará de no permitir que la tarjeta de crédito que se introduzca para pagar un pedido no esté caducada.

```
create or replace TRIGGER seguridadTarjeta
BEFORE INSERT ON REALIZA_PEDIDO FOR EACH ROW
BEGIN
    IF :new.FECHA_CADUC < SYSDATE then
        DBMS_OUTPUT.PUT_LINE('La tarjeta está caducada');
    END IF;
END;
```

DISPARADOR 2 → Nuestro segundo disparador será “*insertCliente*”. En este caso el disparador se encargará de evitar que un cliente sea insertado en su respectiva tabla si es menor de 16 años.

```
create or replace TRIGGER insertCliente
BEFORE INSERT ON CLIENTE FOR EACH ROW
BEGIN
    IF :new.EDAD < 16 then
        DBMS_OUTPUT.PUT_LINE('Error. Cliente NO insertado.');
```

12. Seguimiento de la implementación

Tal y como mencionamos al comienzo del trabajo, hemos utilizado una base de datos de Oracle por lo que tuvimos que utilizar un driver para poder conectarnos. El proceso de conexión de NetBeans a la base de datos consiste primeramente, en descargarnos dicho driver desde la página oficial de Oracle (<http://www.oracle.com/technetwork/database/features/jdbc/index-091264.html>). Una vez descargado simplemente añadir el “.jar” a los paquetes de dicho proyecto (cabe decir que debemos estar conectados a la facultad mediante conexión VPN).

Para conectarnos a la base de datos de nuestra facultad hemos utilizado el siguiente código:

```
Connection con = null;
public Connection conexion(){
    try {
        //Cargar el Driver
        Class.forName("oracle.jdbc.driver.OracleDriver");
        con=DriverManager.getConnection("jdbc:oracle:thin:@//oracle0.ugr.es:1521/practbd.oracle0.ugr.es", "x5958777", "x5958777");
        //System.out.println("Conexion establecida");
        JOptionPane.showMessageDialog(null, "Conexion establecida");
    } catch (Exception e) {
        System.out.println("Error de conexion");
        JOptionPane.showMessageDialog(null, "Error de conexion "+e);
    }
    return con;
}
```

En donde tendremos que sustituir “x5958777” por el login correspondiente de cada usuario en la base de datos.

Una vez conectado con la base de datos, y creadas las correspondientes tablas en SQLdeveloper, la dinámica de nuestro programa es que cada formulario dependiendo de la funcionalidad que tenga, se pueda comunicar con la base de datos mediante secuencias pl/sql. Para que esto sea posible hemos trabajado con tipos de datos entre otras como Statement, PreparedStatement, ResultSet, etc, tipos de datos utilizados para poder manejar los resultados de nuestras consultas PL/SQL.

Debido al poco tiempo disponible, en el sistema podemos tener dos roles. El usuario, donde tiene las funcionalidades de cualquier usuario como son realizar un pedido, cambiar su contraseña, consultar restaurantes, etc. Por otro lado, tenemos un administrador quien es responsable de todos los usuarios de diferentes restaurantes, que como su nombre indica tiene ciertos privilegios que un usuario no tiene.