La Forja Gaming

Cándido Nicolás Pérez Verwer

1ºA Desarrollo de Aplicaciones Web – BAE

1- Introducción

1.1- De qué trata

En este proyecto vamos a tratar la base de datos de un cibercafé, que está dedicado a los E-sports y ofrece también servicio de tienda/taller, con nombre de “La Forja Gaming”

Con lo anterior en mente y para desarrollarlo mejor, separamos para poder explicarlo por partes.

Un cliente puede ir al taller, para bien contratar un servicio de reparación/montaje, así como para realizar una compra a través del taller, ya que este sirve de intermediario y como punto de recogida para las webs de Amazon, Pccomponentes, El Corte Inglés y MediaMarkt, ya que a pesar de que ofrecen servicios de envío puede que no esté el cliente en su domicilio en el momento de recibirlo. Para ofrecer el mejor precio hay una comparativa de precios, en la que se puede buscar por características deseadas o bien por un artículo en concreto.

Para la parte de cibercafé, el cliente puede alquilar el equipo por horas, así como apuntarse a los distintos torneos que se realizan para animar la afluencia. También se puede alquilar las salas con consolas para el caso de que no se quiera jugar en PC.

En la zona de descanso se implementan pantallas cuando se están organizando torneos tanto internos como externos (OW league o de la LVPES, así como eventos como la Blizzcon). En dicha zona hay una máquina distribuidora de comida y de bebida.

1.2- Cómo funciona

Para el proyecto hemos elegido una mezcla de negocios, siendo un híbrido entre tienda/taller y un cibercafé, con un par de salas de juego separadas y una pequeña zona de descanso con máquinas expendedoras de comidas.

El funcionamiento sería el siguiente, un **cliente** puede acceder al comercio para acceder al **taller/tienda** (es una pequeña tienda de reparaciones o donde se hacen servicios de microinformática, así como donde el cliente puede comprar una o varias piezas a través del comercio para que en vez de que le llegue a su domicilio llegue al local y después recogerlo. En caso de estar esperando una

reparación puede esperar en la **zona de descanso**, o bien alquilar un **equipo** o **sala de juegos** con **consolas** para poder jugar mientras espera. También para mover la parte de **cibercafé** (que estará orientada hacia gaming) se organizan diversos eventos para promover la afluencia(**torneos**).

2- Personal de la empresa

La empresa se sitúa en un **local** en la isla de Gran Canaria de la cual necesitamos saber su dirección, teléfono, localidad y horario. Dentro del local podemos identificar al **personal** cuyo elemento identificativo será su DNI, obteniendo también su nombre, primer apellido, segundo apellido, correo electrónico, número de teléfono y el cargo que ostenta (dicho cargo tendrá asociado un horario), pudiendo ostentar un solo cargo en la empresa.

- **Gerente:** Se encarga de la gestión de personal y cubre también en caso de necesidad el puesto de administrativo.
- **Administrativo:** Se encarga de realizar el alta de los clientes en el sistema, tomando su DNI, su nombre, primer apellido, segundo apellido, correo electrónico, número de teléfono, tras lo cual se le daría un código identificativo al cliente para usar el sistema y rellena los partes de incidencia para el compañero de mantenimiento, una vez que estas tareas están realizadas ayudarán a los clientes en caso de que sea necesario con el proceso de compra, al igual que darle el paquete una vez haya llegado.
- **Técnico:** Se encarga de realizar las reparaciones pertinentes, así como proveer ayuda al cliente durante el proceso de compra, usando el comparador provisto para ello, al igual que darle el paquete una vez haya llegado.
- **Limpiador:** se encarga de limpiar el local
- **Auxiliar:** Se encarga de realizar las funciones complementando al administrativo y al técnico, siendo de este último solo la función de ayudar en la compra del cliente usando el comparador, al igual que darle el paquete una vez haya llegado.
- **Mantenimiento:** se encarga de solucionar los partes de incidencia que le pasa el administrativo.

3- Distribución y horario

El horario del comercio es de lunes a viernes 9 a 21 estando distribuidos los horarios de las siguientes maneras:

Gerente, no tiene un horario fijo, ya que es un autónomo que le hecha más horas que toda la clase junta a los trabajos que nos mandan.

Administrativo, de 9 a 13 y de 16 a 20 de lunes a domingo, (al no poder trabajar durante toda la semana por la ilegalidad de acuerdo a la jurisdicción habrá dos administrativos que se rotaran los fines de semana y librando de forma alterna los dos días o entre semana o fines de semana).

Limpiador, viene dos horas por la mañana (9 a 11) y dos horas por la tarde (14 a 16).

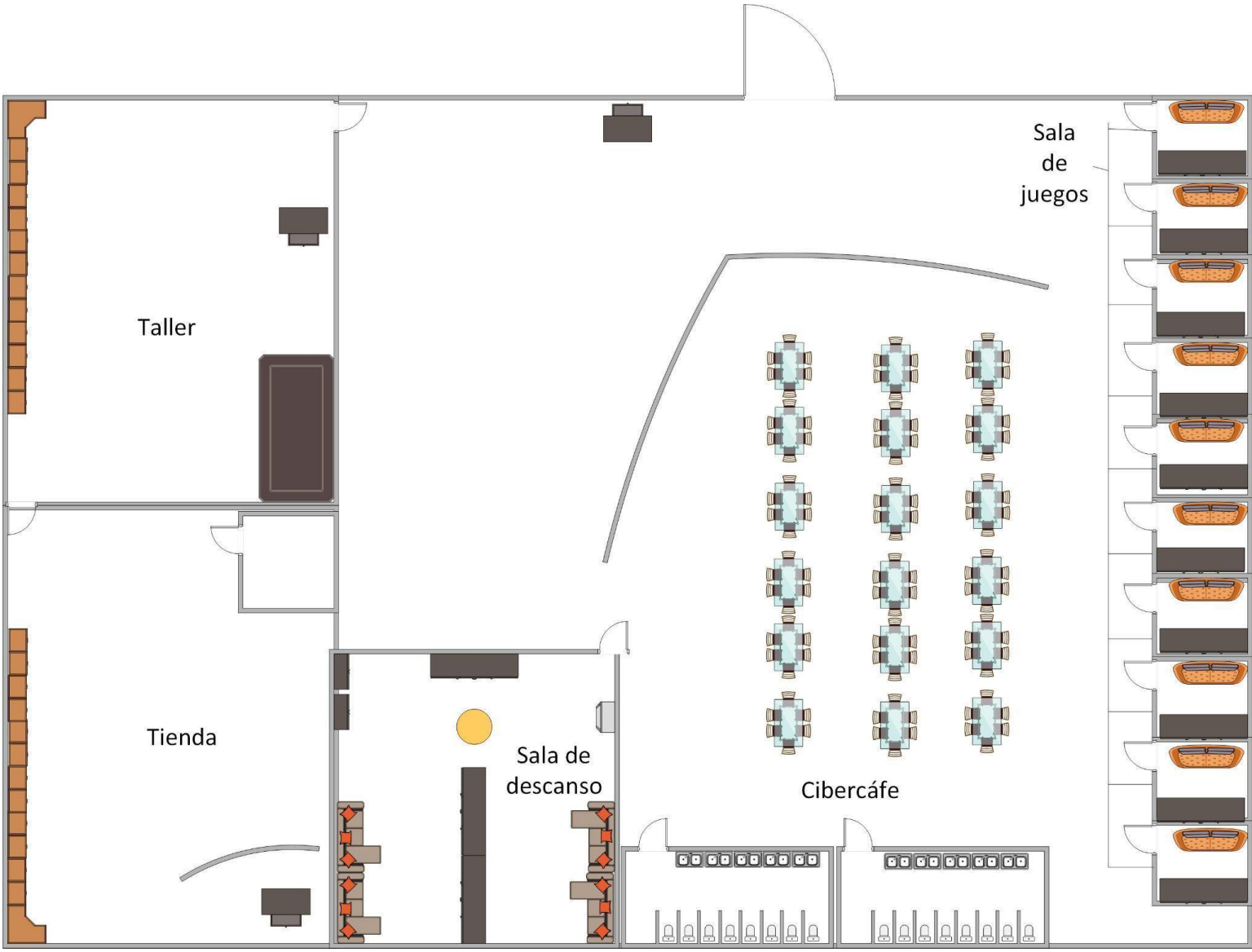
Auxiliar, de 13 a 16 y de 20 a 21 todos los días, al igual que el administrativo, los asistentes se rotan de forma alterna.

Técnico, de 9 a 13 y de 16 a 20 de lunes a viernes.

Mantenimiento, en principio sería de dos horas cada mañana para solucionar los problemas que requieran de su atención, en caso de que tarde más tiempo de su horario se compensa con otro día, en caso de no tener partes de incidencia se dedica al mantenimiento de las instalaciones.

4- Distribución del local

Local 1



4.1- Tabla de entidades, relaciones, grado y correspondencia

Entidades	Diminutivo	Relación	Grado	Cardinalidad y correspondencia
Cliente	cl	cl acude a loc	2	1,n uno o varios clientes acuden al local 1,1 a un único local acude como mínimo un cliente (n:1)
		cl solicita reparación en el tall	2	1,n uno o varios clientes solicitan reparación en el taller 1,1 un taller repara un aparato a la vez (n:1)
		cl compra en la td	2	1,n uno o varios clientes compran en la tienda 1,1 una tienda vende a un cliente (n:1)
		cl alquila sjueg	2	1,n uno o varios clientes alquilan una o varias salas de juegos 1,n una o varias salas de juegos son alquiladas por uno o

				varios clientes (n:m)
		cl acude a tor	2	1,n uno o varios clientes acuden a uno o varios torneos 1,n uno o varios torneos pueden darse a la vez (n:m)
		cl interactúa con el pers	2	1,n uno o varios clientes interactúan con uno o varios empleados 1,n uno o varios empleados interactúan con uno o varios clientes (n:m)
personal	pers	pers organizator	2	1,n uno o varios miembros del personal organizan uno o varios torneos 1,n uno o varios torneos son organizados por uno o varios empleados (n:m)
		pers trabaja en loc	2	1,n uno o varios miembros del personal trabajan en el local

				1,1 solo hay un local en el que trabaja como mínimo un empleado (n:1)
gerente	ger	ger gestiona pers	2	1-n un gerente gestiona a uno o varios miembros del personal 1,1 un miembro del personal es gestionado por un gerente (n:1)
		ger cubre al adm	2	1-1 un gerente cubre a un administrativo 1,1 un administrativo es cubierto por un gerente (1:1)
administrativo	adm	adm da de alta al cl	2	1,1 un administrativo da de alta a un cliente 1,n uno o varios clientes son dados de alta por un administrativo (1:n)
		adm rellena pinci	2	1,1 un administrativo rellena uno o varios

				partes de incidencia 1,n uno o varios partes de incidencia son rellenados por un administrativo (1:n)
tecnico	tec	tec repara aparatos de cl	2	1,1 un técnico repara un aparato 1,n uno o varios aparatos son reparados por un tecnico (1:n)
limpiador	lim	lim limpia el loc	2	1,1 un limpiador limpia un local 1,1 un local es limpiado por un limpiador (1:1)
mantenimiento	man	man repara pinci	2	1,1 mantenimiento repara un parte de incidencia 1,n uno o varios partes de incidencia son reparados por mantenimiento (1:n)
auxiliar	aux	aux cubre al adm	2	1-1 un auxiliar cubre a un administrativo 1,1 un administrativo es cubierto por un

				auxiliar (1:1)
equipos	equi	equi están colocados en el ccafe	2	1,n uno o varios equipos estan colocados en el cibercafe 1,1 solo hay un cibercafe (n:1)
consolas	cons	cons se posicionan en las sjueg	2	1,n una o varias consolas se posicionan en una o varias salas de juego 1,n una o varias salas de juegos tienen posicionadas una o varias consolas (n:m)
local	loc			
torneos	tor			
taller	tall			
tienda	td			
sala de juegos	sjueg			
partes de incidencia	pinci			
cibercafe	ccafe			

sala de descanso	sdes			

Tabla 1: Entidades, relaciones, grado y correspondencia

4.2- Análisis de atributos

Entidad	Atributos
Cliente	ID_usuario*, DNI, nombre, apellido1, apellido2, email, teléfono
Personal	DNI_personal*, nombre, apellido1, apellido2, email, telefono, cargo
Gerente	ID_ger*, cargo
Administrativo	ID_adm*, cargo
Auxiliar	ID_aux*, cargo
Mantenimiento	ID_man*, cargo
Tecnico	ID_tec*, cargo
Limpiador	ID_lim*, cargo
equipos	ID_equipo*, Num_serie, tipo _equipo, unidades
consolas	ID_consola*, cod_local
ordenadores	ID_ordenador*, cod_local
local	CIF*, dirección, telefono, web, horario, cod_local
torneos	id_torneo*, fecha_torneo, premio
taller	id_ taller*, cod_local, horario
tienda	id_tienda*, cod_local, horario
sala de juegos	id_salajuegos*, cod_local, horario
incidencias	id_parte*, especificaciones, cod_local
cibercafe	id_cibercafe*, cod_local, horario
sala de descanso	id_saladescanso*, cod_local, horario

Tabla 2: Análisis de atributos.

4.3- Tabla de especializaciones

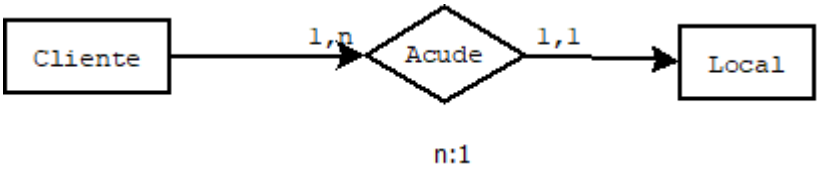
Especializaciones
En este caso tenemos una especialización, que de la superclase personal salen las subclases de gerente, administrativo, técnico, auxiliar, limpiador, y mantenimiento, teniendo los atributos puestos en personal, con la única diferencia de cargo
También tenemos una especialización en local, ya que se divide en varias partes, como el taller, el cibercafé, sala de juegos y sala de espera.
También en equipos, que se dividen en ordenadores y consolas

Tabla 3: tabla de especializaciones.

4.4- Cardinalidades y su explicación

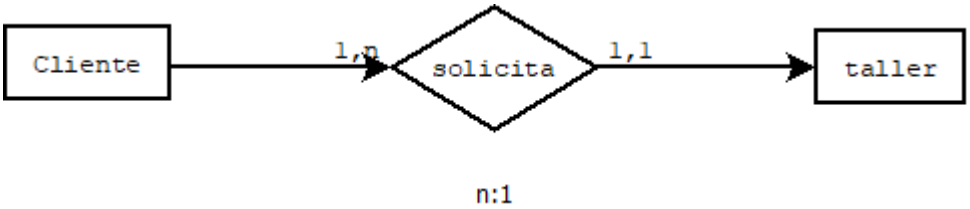
- 1- El cliente acude al local, la cardinalidad para cliente es 1,n, ya que pueden ir varios clientes a la vez. La cardinalidad para la entidad local es 1,1.

Relaciones 1: cliente acude al local



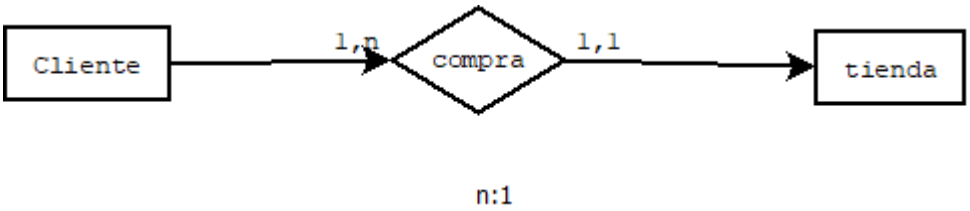
- 2- El cliente solicita una reparación en el taller, la cardinalidad para cliente es 1,n, ya que pueden solicitarla varios clientes a la vez. La cardinalidad para la entidad taller es 1,1.

Relaciones 2: cliente solicita reparación en taller



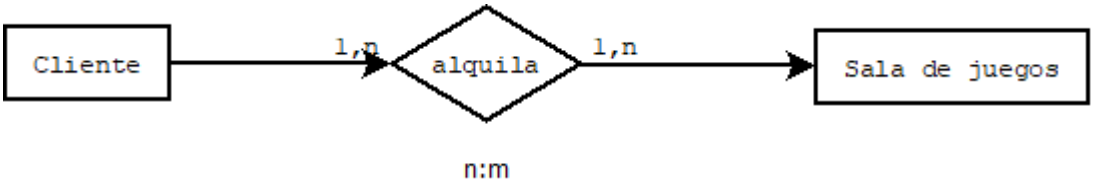
- 3- El cliente compra en la tienda, la cardinalidad para el cliente es 1,n ya que pueden comprar varios clientes, la cardinalidad para la entidad tienda es 1,1.

Relaciones 3: cliente compra en la tienda



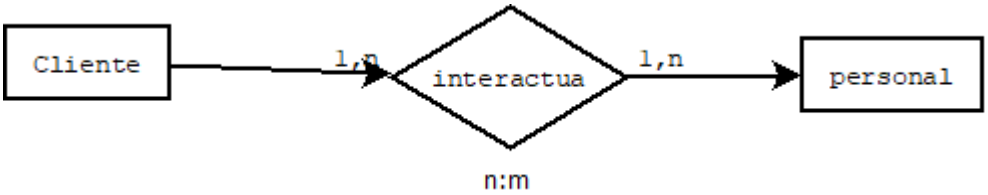
- 4- El cliente alquila una sala de juegos, la cardinalidad es 1,n ya que varios clientes pueden alquilar una sala de juegos, y desde la entidad sala de juegos es 1,n ya que pueden alquilar varias salas a la vez

Relaciones 4: alquila una sala de juegos



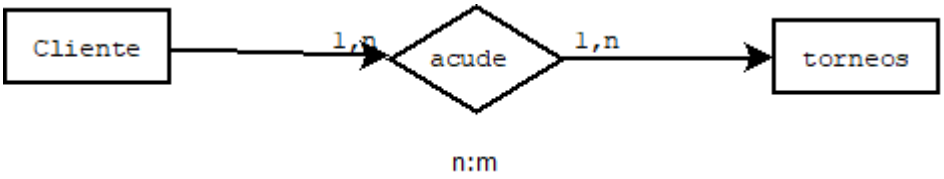
- 5- el cliente interactúa con el personal, la cardinalidad para cliente es 1,n y para personal seria 1,n, ya que pueden interactuar con uno o con varios respectivamente

Relaciones 5: cliente interactúa con el personal



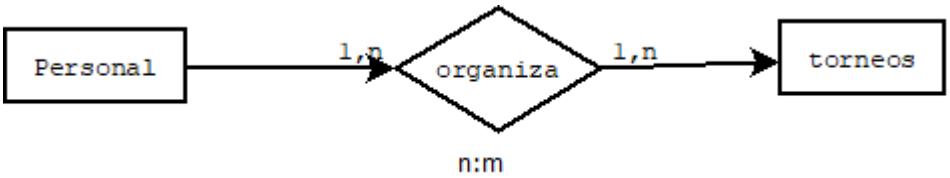
- 6- cliente acude a torneos, la relación desde cliente es 1,n porque pueden asistir varios clientes, así como torneos tiene una relación 1,n porque puede haber varios torneos a la vez

Relaciones 6: cliente acude a torneos



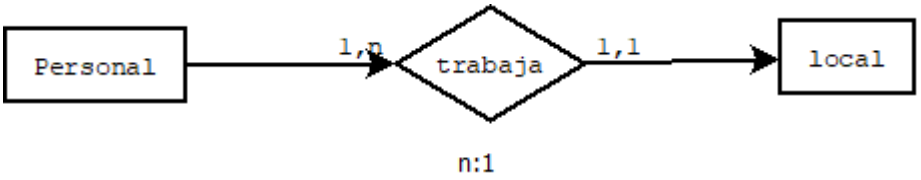
- 7- personal organiza torneos, la relación de personal es 1,n ya que puede ser que varios organizan distintos torneos como que uno organice uno nada más, y desde el punto de vista de torneo seria 1,n porque puede haber varios torneos a la vez

Relaciones 7: personal organiza torneos



- 8 - personal trabaja en local, el personal tiene una cardinalidad de 1,n y el local de 1,1

Relaciones 8: personal trabaja en local



n:1

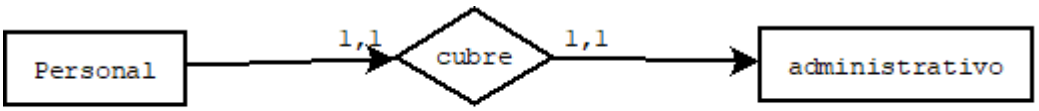
- 9 - gerente gestiona personal, tiene una cardinalidad de 1,n ya que 1 gerente puede gestionar a uno o a varios empleados, mientras que un empleado es gestionado por un gerente por lo que la cardinalidad desde empleado 1,1.

Relaciones 9: gerente gestiona personal

n:1

- 10 - gerente cubre al administrativo, con una cardinalidad dual de 1,1, ya que solo cubre un puesto y no a la persona, al igual que el puesto es cubierto por el gerente

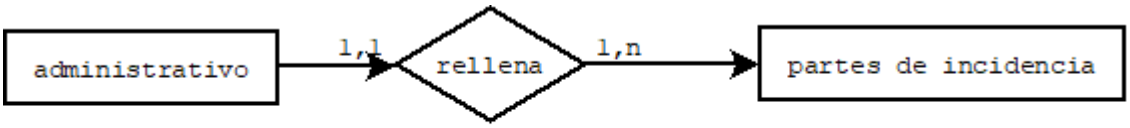
Relaciones 10: gerente cubre al administrativo



1:1

- 11- administrativo rellena partes de incidencia, con cardinalidad 1,1 en administrativo ya que solo uno escribe el parte de incidencia y con cardinalidad 1,n, ya que puedes escribir más de un parte de incidencia.

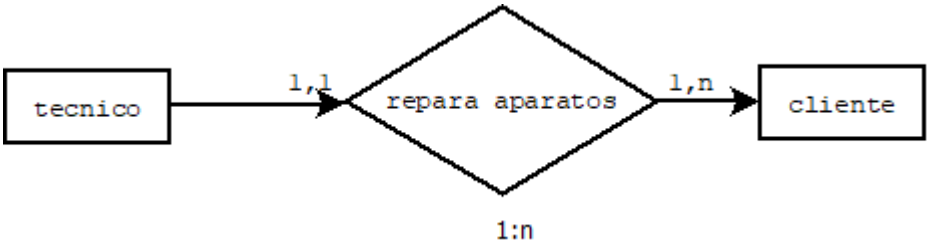
Relaciones 11: administrativo rellena partes de incidencia



1:n

- 12- técnico repara aparatos de cliente. desde la parte del técnico la cardinalidad sería 1,1 ya que solo hay uno y desde aparatos de cliente seria 1,n ya que puede haber varios aparatos, pero mínimo 1.

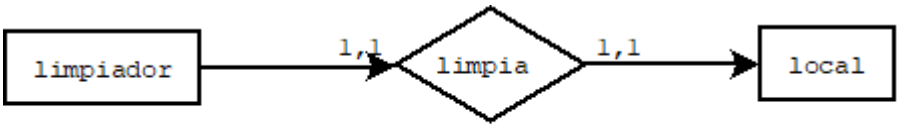
Relaciones 12: técnico repara aparatos de cliente



1:n

- 13- El limpiador limpia el local, la cardinalidad para la entidad limpiador es 1,1. La oficina es limpiada por el limpiador, la cardinalidad para la entidad local es 1,1

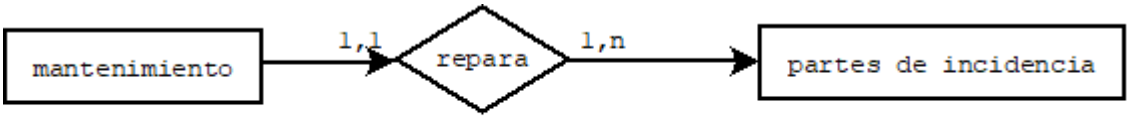
Relaciones 13: limpiador limpia el local



1:1

- 14- mantenimiento repara partes de incidencia, la cardinalidad para la entidad mantenimiento es 1,1, mantenimiento repara partes de incidencia, siendo la cardinalidad de partes de incidencia 1,n

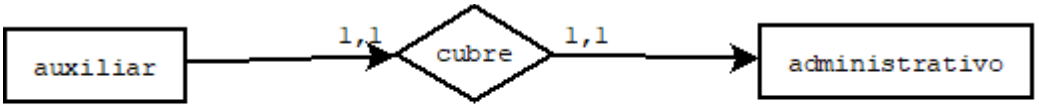
Relaciones 14: mantenimiento repara partes de incidencia



1:n

- 15- auxiliar cubre al administrativo, con una cardinalidad dual de 1,1, ya que solo cubre un puesto y no a la persona, al igual que el puesto es cubierto por el auxiliar

Relaciones 15: auxiliar cubre al administrativo



1:1

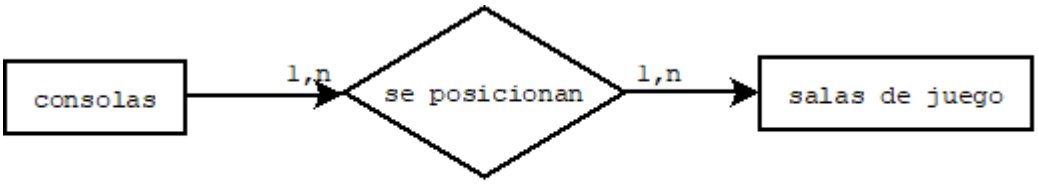
- 16- los equipos están colocados en el cibercafé, con una cardinalidad de 1,n ya que pueden ser más equipos y con cardinalidad 1,1 para cibercafé, ya que solo hay uno

Relaciones 16: equipos están colocados en cibercafé

n:1

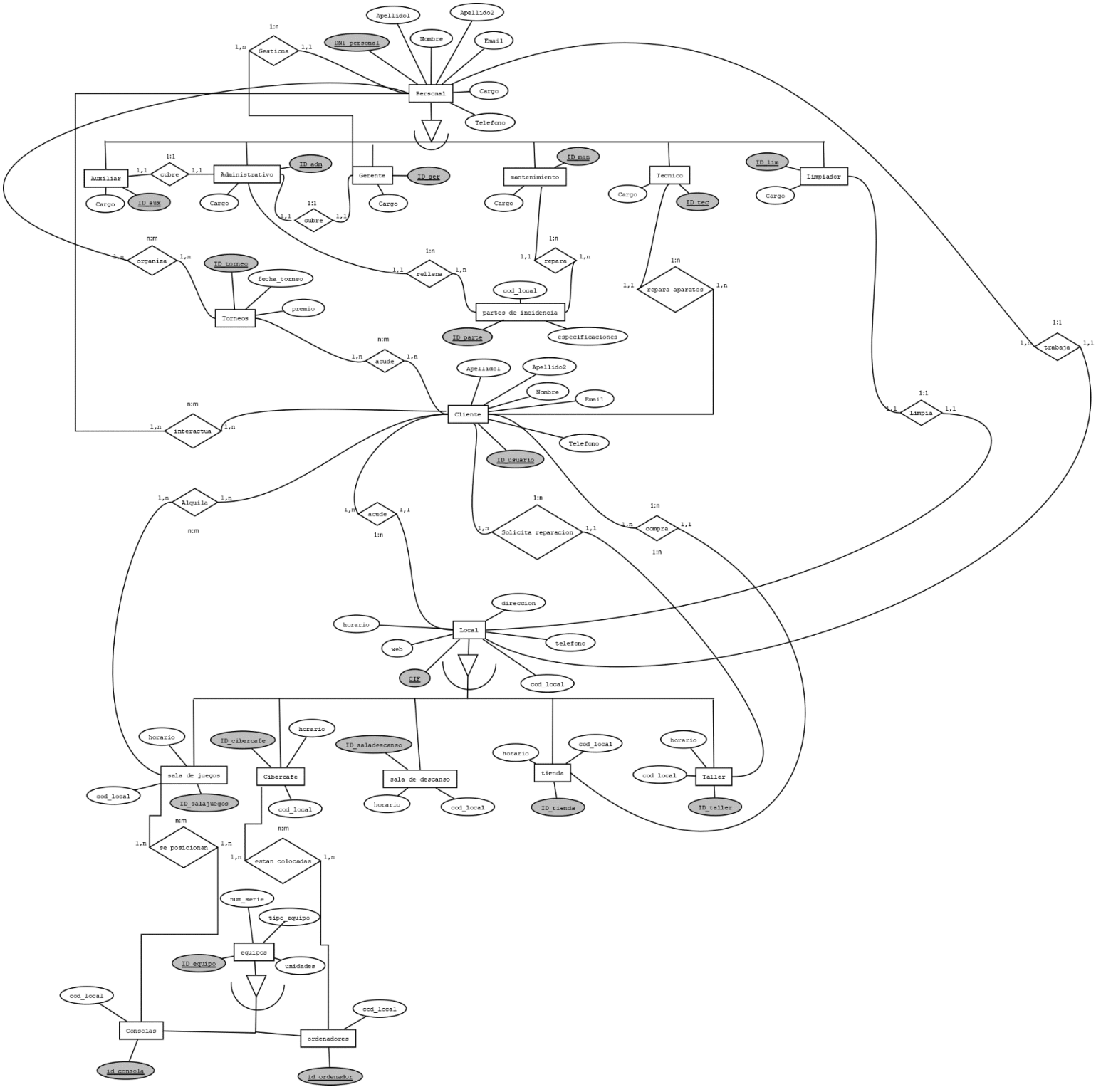
- 17- las consolas se posicionan en las salas de juego, con cardinalidad 1,n ya que hay varias consolas y también en salas de juego con 1,n ya que hay varias salas de juego.

Relaciones 17: consolas se posicionan en salas de juego



n:m

4.5- Diagrama general



Relaciones 18: Diagrama entidad-relación

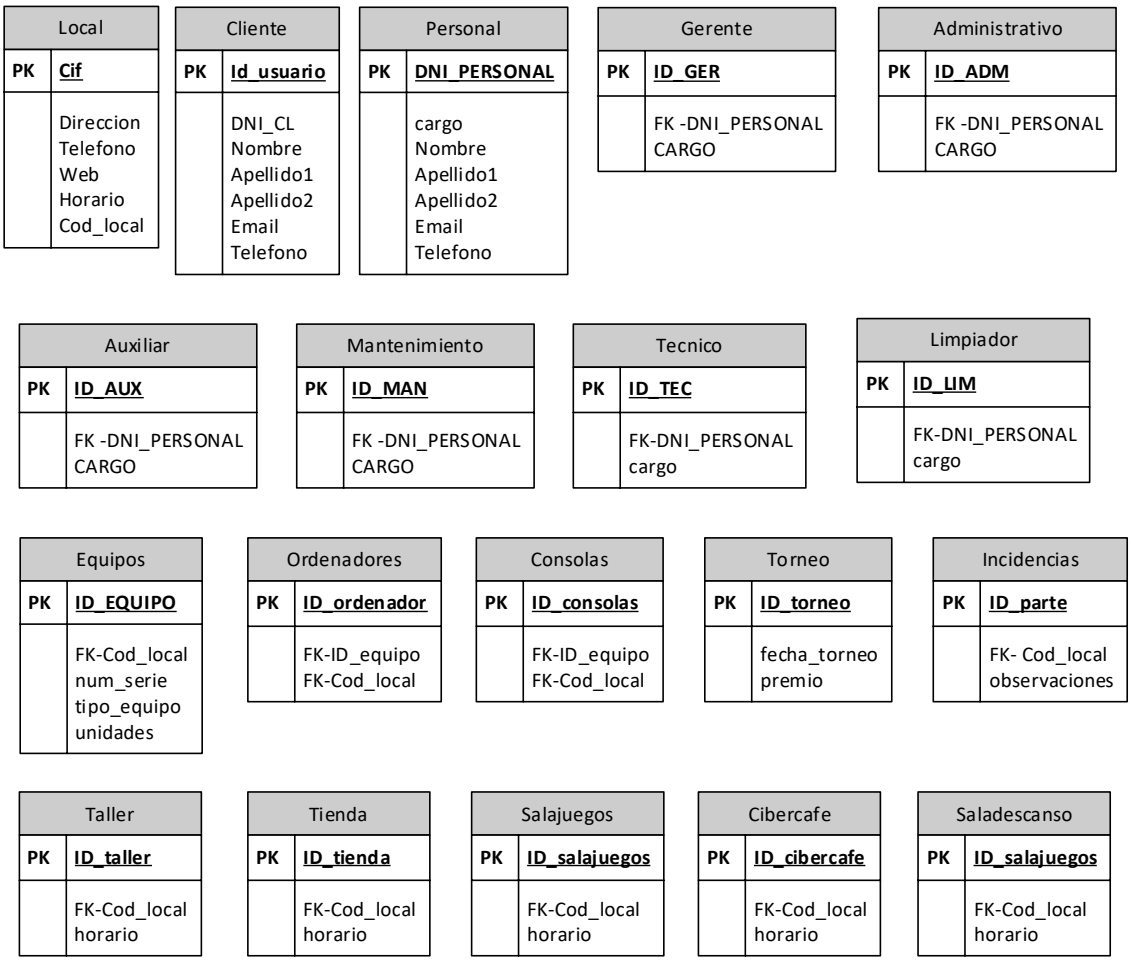
5- Modelo Físico Relacional:

5.1- A)-Paso Previo -

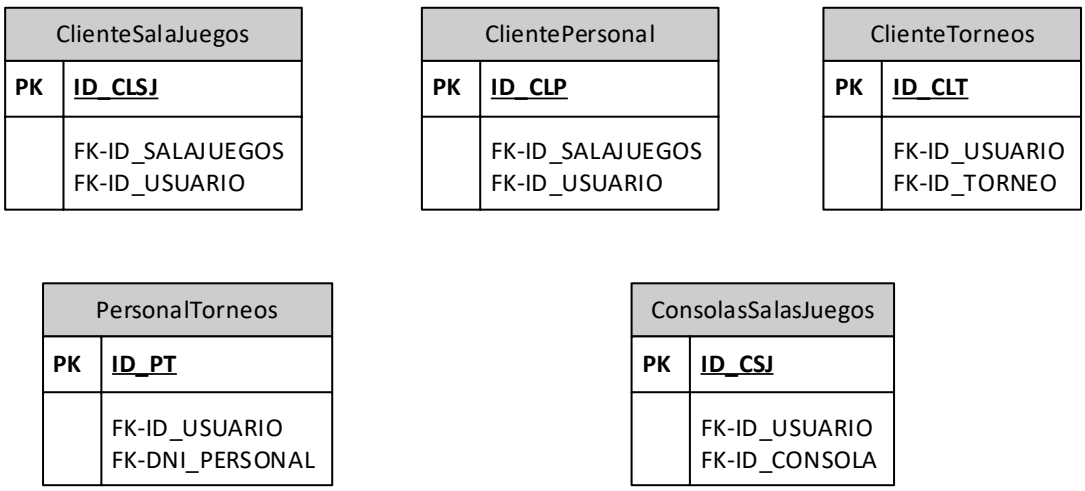
-Tabla LOCAL= **CIF**, DIRECCION,TELEFONO,WEB,HORARIO, COD_LOCAL
-Tabla CLIENTE=**ID_USUARIO**, DNI_CL, NOMBRE_CL, APELLIDO1_CL, APELLIDO2_CL, EMAIL, TELÉFONO,
-Tabla PERSONAL=**DNI_PERSONAL**, CARGO,NOMBRE,APELLIDO1, APELLIDO2,EMAIL,TELÉFONO,
-Tabla GERENTE =**ID_GER, *DNI_PERSONAL***, CARGO
-Tabla ADMINISTRATIVO =**ID_ADM, *DNI_PERSONAL***, CARGO
-Tabla AUXILIAR= **ID_AUX, *DNI_PERSONAL***, CARGO
-Tabla MANTENIMIENTO= **ID_MAN, *DNI_PERSONAL***, CARGO
-Tabla TECNICO= **ID_TEC, *DNI_PERSONAL***, CARGO
-Tabla LIMPIADOR = **ID_LIM**, ***DNI_PERSONAL***, CARGO
-Tabla EQUIPOS= **ID_EQUIPO**, NUM_SERIE, TIPO_EQUIPO, UNIDADES, ***COD_LOCAL***
-Tabla ORDENADORES=**ID_ORDENADOR**, ***ID_EQUIPO,COD_LOCAL***
-Tabla CONSOLAS= **ID_CONSOLA**, ***ID_EQUIPO, COD_LOCAL***
-Tabla TORNEO= **ID_TORNEO**, FECHA_TORNEO, PREMIO
-Tabla INCIDENCIAS= **ID_PARTE, *COD_LOCAL***, OBSERVACIONES,
-Tabla TALLER=**ID_TALLER, *COD_LOCAL***, HORARIO
-Tabla TIENDA= **ID_TIENDA, *COD_LOCAL***, HORARIO
-Tabla SALAJUEGOS= **ID_SALAJUEGOS, *COD_LOCAL***, HORARIO
-Tabla CIBERCAFE= **ID_CIBERCAFE, *COD_LOCAL***, HORARIO
-Tabla SALADESCANSO= **ID_SALADESCANSO**, ***COD_LOCAL***, HORARIO
-Tabla ClienteSalaJuegos = **ID_CLSJ, *ID_SALAJUEGOS, ID_USUARIO***,
-Tabla ClientePersonal = **ID_CLP, *ID_USUARIO, DNI_PERSONAL***.
-Tabla ClienteTorneos = **ID_CLT*, ID_USUARIO, ID_TORNEO*.**
-Tabla PersonalTorneos = **ID_PT, *ID_TORNEO, DNI_PERSONAL*.**
-TConsolasSalasJuegos = **ID_CSJ. *ID_SALAJUEGOS, ID_CONSOLA***

en mayúscula las entidades, en minúscula las relaciones que tenían que tener tabla, en negrita las Primary Key y en negrita y cursiva las Foreign Key

5.2- B)-Tablas -

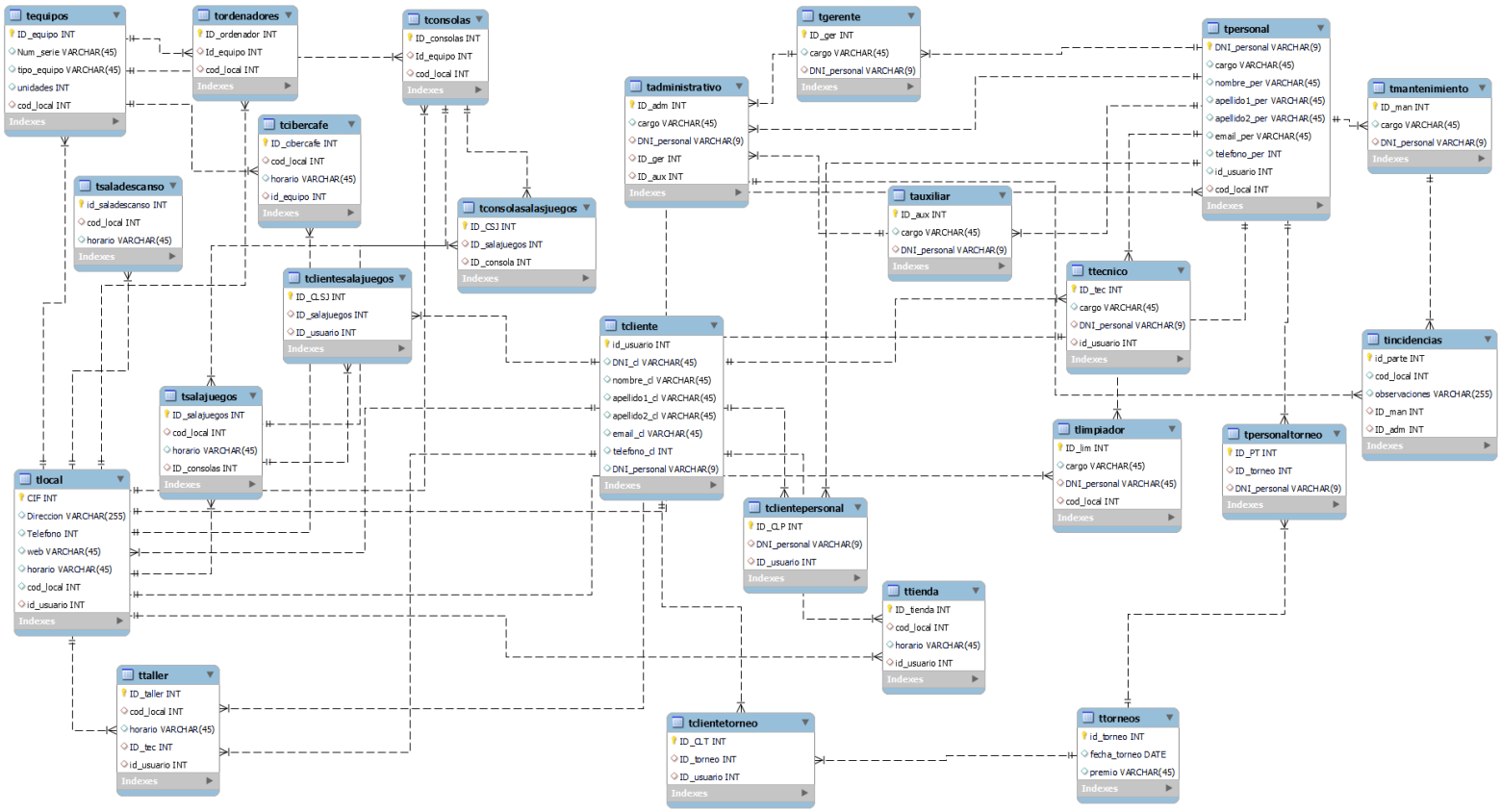


Modelo Físico 1: Entidades



Modelo Físico 2: Relaciones entre entidades

5.3- C)-Diagrama de Tablas Relacional –



MYSQL

5.4- A)-Creación de tablas

Una vez planificado la organización de la base de datos, la creamos con comandos en MySQL command line.

Creamos la base de datos y nos metemos dentro para añadir las tablas

```
mysql> CREATE DATABASE LAFORJA;
Query OK, 1 row affected (0.02 sec)

mysql> USE laforja;
Database changed
mysql>
```

MySQL 1: Creación y uso de la Base de Datos

```
mysql> CREATE TABLE Tcliente (
    ->   id_usuario INT NOT NULL AUTO_INCREMENT,
    ->   DNI_cl VARCHAR(45) NULL,
    ->   nombre_cl VARCHAR(45) NULL,
    ->   apellido1_cl VARCHAR(45) NULL,
    ->   apellido2_cl VARCHAR(45) NULL,
    ->   email_cl VARCHAR(45) NULL,
    ->   telefono_cl INT NULL,
    ->   DNI_personal VARCHAR(9) NULL,
    ->   PRIMARY KEY (id_usuario));
Query OK, 0 rows affected (0.06 sec)

mysql>
```

MySQL 2: Cliente

```
mysql> CREATE TABLE Tlocal (
    ->   CIF INT NOT NULL,
    ->   Direccion VARCHAR(255) NULL,
    ->   Telefono INT NULL,
    ->   web VARCHAR(45) NULL,
    ->   horario VARCHAR(45) NULL,
    ->   cod_local INT NULL,
    ->   id_usuario INT NULL,
    ->   PRIMARY KEY (CIF),
    ->   INDEX cod_local (cod_local ASC) VISIBLE,
    ->   INDEX id_usuario_idx (id_usuario ASC) VISIBLE,
    ->   CONSTRAINT id_usuario FOREIGN KEY (id_usuario) REFERENCES Tcliente (id_usuario)
    ->     ON DELETE NO ACTION
    ->     ON UPDATE NO ACTION);
Query OK, 0 rows affected (0.04 sec)

mysql>
```

MySQL 3: Local

```
mysql>   CREATE TABLE Tpersonal (
    ->   DNI_personal VARCHAR(9) NOT NULL,
    ->   cargo VARCHAR(45) NULL,
    ->   nombre_per VARCHAR(45) NULL,
    ->   apellido1_per VARCHAR(45) NULL,
    ->   apellido2_per VARCHAR(45) NULL,
    ->   email_per VARCHAR(45) NULL,
    ->   telefono_per INT NULL,
    ->   id_usuario INT NULL,
    ->   cod_local INT NULL,
    ->   PRIMARY KEY (DNI_personal),
    ->   INDEX cod_local_idx (cod_local ASC) VISIBLE,
    ->   CONSTRAINT cod_local
    ->     FOREIGN KEY (cod_local)
    ->     REFERENCES Tlocal (cod_local)
    ->     ON DELETE NO ACTION
    ->     ON UPDATE NO ACTION);
Query OK, 0 rows affected (0.03 sec)

mysql>
```

MySQL 4: Personal

```
mysql> CREATE TABLE Tgerente (
    ->   ID_ger INT NOT NULL,
    ->   cargo VARCHAR(45) NULL,
    ->   DNI_personal VARCHAR(9) NULL,
    ->   PRIMARY KEY (ID_ger),
    ->   INDEX DNI_personal_idx (DNI_personal ASC) VISIBLE,
    ->   CONSTRAINT DNI_personal
    ->     FOREIGN KEY (DNI_personal)
    ->     REFERENCES Tpersonal (DNI_personal)
    ->     ON DELETE NO ACTION ON UPDATE NO ACTION);
Query OK, 0 rows affected (0.03 sec)
```

MySQL 5: Gerente

```
mysql> CREATE TABLE Tauxiliar (
    ->   ID_aux INT NOT NULL,
    ->   cargo VARCHAR(45) NULL,
    ->   DNI_personal VARCHAR(9) NULL,
    ->   PRIMARY KEY (ID_aux),
    ->   INDEX DNI_personal_idx (DNI_personal ASC) VISIBLE,
    ->   CONSTRAINT DNI_personal1
    ->     FOREIGN KEY (DNI_personal) REFERENCES Tpersonal (DNI_personal)
    ->     ON DELETE NO ACTION ON UPDATE NO ACTION);
Query OK, 0 rows affected (0.03 sec)

mysql>
```

MySQL 6: Auxiliar

```
mysql> CREATE TABLE Tadministrativo (
    ->   ID_adm INT NOT NULL,
    ->   cargo VARCHAR(45) NULL,
    ->   DNI_personal VARCHAR(9) NULL,
    ->   ID_ger INT NULL,
    ->   ID_aux INT NULL,
    ->   PRIMARY KEY (ID_adm),
    ->   INDEX DNI_personal_idx (DNI_personal ASC) VISIBLE,
    ->   INDEX ID_ger_idx (ID_ger ASC) VISIBLE,
    ->   INDEX ID_aux_idx (ID_aux ASC) VISIBLE,
    ->   CONSTRAINT DNI_personal2
    ->     FOREIGN KEY (DNI_personal)
    ->     REFERENCES Tpersonal (DNI_personal)
    ->     ON DELETE NO ACTION
    ->     ON UPDATE NO ACTION,
    ->   CONSTRAINT ID_ger
    ->     FOREIGN KEY (ID_ger)
    ->     REFERENCES Tgerente (ID_ger)
    ->     ON DELETE NO ACTION
    ->     ON UPDATE NO ACTION,
    ->   CONSTRAINT ID_aux
    ->     FOREIGN KEY (ID_aux)
    ->     REFERENCES Tauxiliar (ID_aux)
    ->     ON DELETE NO ACTION
    ->     ON UPDATE NO ACTION);
Query OK, 0 rows affected (0.04 sec)
```

MySQL 7: Administrativo

```
mysql> CREATE TABLE Tmantenimiento (
    ->   ID_man INT NOT NULL,
    ->   cargo VARCHAR(45) NULL,
    ->   DNI_personal VARCHAR(9) NULL,
    ->   PRIMARY KEY (ID_man),
    ->   INDEX DNI_persoanl_idx (DNI_personal ASC) VISIBLE,
    ->   CONSTRAINT DNI_personal3
    ->     FOREIGN KEY (DNI_personal)
    ->     REFERENCES Tpersonal (DNI_personal)
    ->     ON DELETE NO ACTION
    ->     ON UPDATE NO ACTION);
Query OK, 0 rows affected (0.03 sec)
```

MySQL 8: Mantenimiento

```
mysql> CREATE TABLE Ttecnico (
    ->   ID_tec INT NOT NULL,
    ->   cargo VARCHAR(45) NULL,
    ->   DNI_personal VARCHAR(9) NULL,
    ->   id_usuario INT NULL,
    ->   PRIMARY KEY (ID_tec),
    ->   INDEX DNI_persoanl_idx (DNI_personal ASC) VISIBLE,
    ->   INDEX id_usuario_idx (id_usuario ASC) VISIBLE,
    ->   CONSTRAINT DNI_personal4
    ->     FOREIGN KEY (DNI_personal)
    ->     REFERENCES Tpersonal (DNI_personal)
    ->     ON DELETE NO ACTION
    ->     ON UPDATE NO ACTION,
    ->   CONSTRAINT id_usuario1
    ->     FOREIGN KEY (id_usuario)
    ->     REFERENCES Tcliente (id_usuario)
    ->     ON DELETE NO ACTION
    ->     ON UPDATE NO ACTION);
Query OK, 0 rows affected (0.03 sec)
```

MySQL 9: Técnico

```
mysql> CREATE TABLE Tlimpiador (
    ->   ID_lim INT NOT NULL,
    ->   cargo VARCHAR(45) NULL,
    ->   DNI_personal VARCHAR(45) NULL,
    ->   cod_local INT NULL,
    ->   PRIMARY KEY (ID_lim),
    ->   INDEX DNI_personal_idx (DNI_personal ASC) VISIBLE,
    ->   INDEX cod_local_idx (cod_local ASC) VISIBLE,
    ->   CONSTRAINT DNI_personal5
    ->     FOREIGN KEY (DNI_personal)
    ->     REFERENCES Tpersonal (DNI_personal)
    ->     ON DELETE NO ACTION
    ->     ON UPDATE NO ACTION,
    ->   CONSTRAINT cod_local1
    ->     FOREIGN KEY (cod_local)
    ->     REFERENCES Tlocal (cod_local)
    ->     ON DELETE NO ACTION
    ->     ON UPDATE NO ACTION);
Query OK, 0 rows affected (0.03 sec)
```

MySQL 10: Limpiador

```
mysql> CREATE TABLE Tequipos (
    ->   ID_equipo INT NOT NULL,
    ->   Num_serie VARCHAR(45) NULL,
    ->   tipo_equipo VARCHAR(45) NULL,
    ->   unidades INT NULL,
    ->   cod_local INT NULL,
    ->   PRIMARY KEY (ID_equipo),
    ->   INDEX cod_local_idx (cod_local ASC) VISIBLE,
    ->   CONSTRAINT cod_local2
    ->     FOREIGN KEY (cod_local)
    ->     REFERENCES Tlocal (cod_local)
    ->     ON DELETE NO ACTION
    ->     ON UPDATE NO ACTION);
Query OK, 0 rows affected (0.02 sec)
```

MySQL 11: Equipos

```
mysql> CREATE TABLE Tordenadores (
    ->   ID_ordenador INT NOT NULL,
    ->   Id_equipo INT NULL,
    ->   cod_local INT NULL,
    ->   PRIMARY KEY (ID_ordenador),
    ->   INDEX ID_equipos_idx (Id_equipo ASC) VISIBLE,
    ->   INDEX Cod_local_idx (cod_local ASC) VISIBLE,
    ->   CONSTRAINT ID_equipos
    ->     FOREIGN KEY (Id_equipo)
    ->     REFERENCES Tequipos (ID_equipo)
    ->     ON DELETE NO ACTION
    ->     ON UPDATE NO ACTION,
    ->   CONSTRAINT Cod_local3
    ->     FOREIGN KEY (cod_local)
    ->     REFERENCES Tlocal (cod_local)
    ->     ON DELETE NO ACTION
    ->     ON UPDATE NO ACTION);
Query OK, 0 rows affected (0.03 sec)
```

MySQL 12: Ordenadores

```
mysql> CREATE TABLE Tconsolas (
    ->   ID_consolas INT NOT NULL,
    ->   Id_equipo INT NULL,
    ->   cod_local INT NULL,
    ->   PRIMARY KEY (ID_consolas),
    ->   INDEX ID_equipos_idx (Id_equipo ASC) VISIBLE,
    ->   INDEX cod_local_idx (cod_local ASC) VISIBLE,
    ->   CONSTRAINT ID_equipos1
    ->     FOREIGN KEY (Id_equipo)
    ->     REFERENCES Tequipos (ID_equipo)
    ->     ON DELETE NO ACTION
    ->     ON UPDATE NO ACTION,
    ->   CONSTRAINT cod_local4
    ->     FOREIGN KEY (cod_local)
    ->     REFERENCES Tlocal (cod_local)
    ->     ON DELETE NO ACTION
    ->     ON UPDATE NO ACTION);
Query OK, 0 rows affected (0.03 sec)
```

MySQL 13: Consolas

```
mysql> CREATE TABLE Ttorneos (
    ->   id_torneo INT NOT NULL,
    ->   fecha_torneo DATE NULL,
    ->   premio VARCHAR(45) NULL,
    ->   PRIMARY KEY (id_torneo));
Query OK, 0 rows affected (0.01 sec)
```

MySQL 14: Torneos

```
mysql> CREATE TABLE Tincidencias (
    ->   id_parte INT NOT NULL,
    ->   cod_local INT NULL,
    ->   observaciones VARCHAR(255) NULL,
    ->   ID_man INT NULL,
    ->   ID_adm INT NULL,
    ->   PRIMARY KEY (id_parte),
    ->   INDEX ID_man_idx (ID_man ASC) VISIBLE,
    ->   INDEX ID_adm_idx (ID_adm ASC) VISIBLE,
    ->   CONSTRAINT ID_man
    ->     FOREIGN KEY (ID_man)
    ->     REFERENCES Tmantenimiento (ID_man)
    ->     ON DELETE NO ACTION
    ->     ON UPDATE NO ACTION,
    ->   CONSTRAINT ID_adm
    ->     FOREIGN KEY (ID_adm)
    ->     REFERENCES Tadministrativo (ID_adm)
    ->     ON DELETE NO ACTION
    ->     ON UPDATE NO ACTION);
Query OK, 0 rows affected (0.03 sec)
```

MySQL 15: Incidencias

```
mysql> CREATE TABLE Ttaller (
    ->   ID_taller INT NOT NULL,
    ->   cod_local INT NULL,
    ->   horario VARCHAR(45) NULL,
    ->   ID_tec INT NULL,
    ->   id_usuario INT NULL,
    ->   PRIMARY KEY (ID_taller),
    ->   INDEX ID_tec_idx (ID_tec ASC) VISIBLE,
    ->   INDEX cod_local_idx (cod_local ASC) VISIBLE,
    ->   INDEX id_usuario_idx (id_usuario ASC) VISIBLE,
    ->   CONSTRAINT ID_tec
    ->     FOREIGN KEY (ID_tec)
    ->     REFERENCES Ttecnico (ID_tec)
    ->     ON DELETE NO ACTION
    ->     ON UPDATE NO ACTION,
    ->   CONSTRAINT cod_local5
    ->     FOREIGN KEY (cod_local)
    ->     REFERENCES Tlocal (cod_local)
    ->     ON DELETE NO ACTION
    ->     ON UPDATE NO ACTION,
    ->   CONSTRAINT id_usuario3
    ->     FOREIGN KEY (id_usuario)
    ->     REFERENCES Tcliente (id_usuario)
    ->     ON DELETE NO ACTION
    ->     ON UPDATE NO ACTION);
Query OK, 0 rows affected (0.03 sec)
```

MySQL 16: Taller

```
ysql> CREATE TABLE Ttienda (
    ->   ID_tienda INT NOT NULL,
    ->   cod_local INT NULL,
    ->   horario VARCHAR(45) NULL,
    ->   id_usuario INT NULL,
    ->   PRIMARY KEY (ID_tienda),
    ->   INDEX cod_local_idx (cod_local ASC) VISIBLE,
    ->   INDEX id_usuario_idx (id_usuario ASC) VISIBLE,
    ->   CONSTRAINT cod_local6
    ->     FOREIGN KEY (cod_local)
    ->     REFERENCES Tlocal (cod_local)
    ->     ON DELETE NO ACTION
    ->     ON UPDATE NO ACTION,
    ->   CONSTRAINT id_usuario4
    ->     FOREIGN KEY (id_usuario)
    ->     REFERENCES Tcliente (id_usuario)
    ->     ON DELETE NO ACTION
    ->     ON UPDATE NO ACTION);
uery OK, 0 rows affected (0.03 sec)
```

MySQL 17: Tienda

```
mysql> CREATE TABLE Tsalajuegos (
    ->   ID_salajuegos INT NOT NULL,
    ->   cod_local INT NULL,
    ->   horario VARCHAR(45) NULL,
    ->   ID_consolas INT NULL,
    ->   PRIMARY KEY (ID_salajuegos),
    ->   INDEX cod_local_idx (cod_local ASC) VISIBLE,
    ->   INDEX id_consolas_idx (ID_consolas ASC) VISIBLE,
    ->   CONSTRAINT cod_local7
    ->     FOREIGN KEY (cod_local)
    ->     REFERENCES Tlocal (cod_local)
    ->     ON DELETE NO ACTION
    ->     ON UPDATE NO ACTION,
    ->   CONSTRAINT id_consolas
    ->     FOREIGN KEY (ID_consolas)
    ->     REFERENCES Tconsolas (ID_consolas)
    ->     ON DELETE NO ACTION
    ->     ON UPDATE NO ACTION);
Query OK, 0 rows affected (0.03 sec)
```

MySQL 18: Sala de Juegos

```
mysql> CREATE TABLE Tcibercafe (
    ->   ID_cibercafe INT NOT NULL,
    ->   cod_local INT NULL,
    ->   horario VARCHAR(45) NULL,
    ->   id_equipo INT NULL,
    ->   PRIMARY KEY (ID_cibercafe),
    ->   INDEX cod_local_idx (cod_local ASC) VISIBLE,
    ->   INDEX id_equipo_idx (id_equipo ASC) VISIBLE,
    ->   CONSTRAINT cod_local8
    ->     FOREIGN KEY (cod_local)
    ->     REFERENCES Tlocal (cod_local)
    ->     ON DELETE NO ACTION
    ->     ON UPDATE NO ACTION,
    ->   CONSTRAINT id_equipo
    ->     FOREIGN KEY (id_equipo)
    ->     REFERENCES Tequipos (ID_equipo)
    ->     ON DELETE NO ACTION
    ->     ON UPDATE NO ACTION);
Query OK, 0 rows affected (0.03 sec)
```

MySQL 19: Cibercafe

```
mysql> CREATE TABLE Tsaladescanso (
    ->   id_saladescanso INT NOT NULL,
    ->   cod_local INT NULL,
    ->   horario VARCHAR(45) NULL,
    ->   PRIMARY KEY (id_saladescanso),
    ->   INDEX cod_local_idx (cod_local ASC) VISIBLE,
    ->   CONSTRAINT cod_local9
    ->     FOREIGN KEY (cod_local)
    ->     REFERENCES Tlocal (cod_local)
    ->     ON DELETE NO ACTION
    ->     ON UPDATE NO ACTION);
Query OK, 0 rows affected (0.03 sec)
```

MySQL 20: Sala de descanso

```
mysql> CREATE TABLE TClienteSalajuegos (
    ->   ID_CLSJ INT NOT NULL,
    ->   ID_salajuegos INT NULL,
    ->   ID_usuario INT NULL,
    ->   PRIMARY KEY (ID_CLSJ),
    ->   INDEX ID_salajuegos_idx (ID_salajuegos ASC) VISIBLE,
    ->   INDEX ID_usuario_idx (ID_usuario ASC) VISIBLE,
    ->   CONSTRAINT ID_salajuegos
    ->     FOREIGN KEY (ID_salajuegos)
    ->     REFERENCES Tsalajuegos (ID_salajuegos)
    ->     ON DELETE NO ACTION
    ->     ON UPDATE NO ACTION,
    ->   CONSTRAINT ID_usuario6
    ->     FOREIGN KEY (ID_usuario)
    ->     REFERENCES Tcliente (id_usuario)
    ->     ON DELETE NO ACTION
    ->     ON UPDATE NO ACTION);
Query OK, 0 rows affected (0.03 sec)
```

MySQL 21: Cliente Sala Juegos

```
mysql> CREATE TABLE TClientePersonal (
    ->   ID_CLP INT NOT NULL,
    ->   DNI_personal VARCHAR(9) NULL,
    ->   ID_usuario INT NULL,
    ->   PRIMARY KEY (ID_CLP),
    ->   INDEX DNI_personal_idx (DNI_personal ASC) VISIBLE,
    ->   INDEX ID_usuario_idx (ID_usuario ASC) VISIBLE,
    ->   CONSTRAINT DNI_personal6
    ->     FOREIGN KEY (DNI_personal)
    ->     REFERENCES Tpersonal (DNI_personal)
    ->     ON DELETE NO ACTION
    ->     ON UPDATE NO ACTION,
    ->   CONSTRAINT ID_usuario7
    ->     FOREIGN KEY (ID_usuario)
    ->     REFERENCES Tcliente (id_usuario)
    ->     ON DELETE NO ACTION
    ->     ON UPDATE NO ACTION);
Query OK, 0 rows affected (0.03 sec)
```

MySQL 22: Cliente Personal

```
mysql> CREATE TABLE TClienteTorneo (
    ->   ID_CLT INT NOT NULL,
    ->   ID_torneo INT NULL,
    ->   ID_usuario INT NULL,
    ->   PRIMARY KEY (ID_CLT),
    ->   INDEX ID_torneo_idx (ID_torneo ASC) VISIBLE,
    ->   INDEX ID_usuario_idx (ID_usuario ASC) VISIBLE,
    ->   CONSTRAINT ID_torneo
    ->     FOREIGN KEY (ID_torneo)
    ->     REFERENCES Ttorneos (id_torneo)
    ->     ON DELETE NO ACTION
    ->     ON UPDATE NO ACTION,
    ->   CONSTRAINT ID_usuario8
    ->     FOREIGN KEY (ID_usuario)
    ->     REFERENCES Tcliente (id_usuario)
    ->     ON DELETE NO ACTION
    ->     ON UPDATE NO ACTION);
Query OK, 0 rows affected (0.03 sec)
```

MySQL 23: Cliente Torneo

```
mysql> CREATE TABLE TPersonalTorneo (
    ->   ID_PT INT NOT NULL,
    ->   ID_torneo INT NULL,
    ->   DNI_personal VARCHAR(9) NULL,
    ->   PRIMARY KEY (ID_PT),
    ->   INDEX Id_torneo_idx (ID_torneo ASC) VISIBLE,
    ->   INDEX DNI_personal_idx (DNI_personal ASC) VISIBLE,
    ->   CONSTRAINT Id_torneo2
    ->     FOREIGN KEY (ID_torneo)
    ->     REFERENCES Ttorneos (id_torneo)
    ->     ON DELETE NO ACTION
    ->     ON UPDATE NO ACTION,
    ->   CONSTRAINT DNI_personal10
    ->     FOREIGN KEY (DNI_personal)
    ->     REFERENCES Tpersonal (DNI_personal)
    ->     ON DELETE NO ACTION
    ->     ON UPDATE NO ACTION);
Query OK, 0 rows affected (0.03 sec)
```

MySQL 24: Personal Torneo

```
mysql> CREATE TABLE TConsolaSalasJuegos (
    ->   ID_CSJ INT NOT NULL,
    ->   ID_salajuegos INT NULL,
    ->   ID_consola INT NULL,
    ->   PRIMARY KEY (ID_CSJ),
    ->   INDEX ID_salajuegos_idx (ID_salajuegos ASC) VISIBLE,
    ->   INDEX ID_consolas_idx (ID_consola ASC) VISIBLE,
    ->   CONSTRAINT ID_salajuegos1
    ->     FOREIGN KEY (ID_salajuegos)
    ->     REFERENCES Tsalajuegos (ID_salajuegos)
    ->     ON DELETE NO ACTION
    ->     ON UPDATE NO ACTION,
    ->   CONSTRAINT ID_consolas1
    ->     FOREIGN KEY (ID_consola)
    ->     REFERENCES Tconsolas (ID_consolas)
    ->     ON DELETE NO ACTION
    ->     ON UPDATE NO ACTION);
Query OK, 0 rows affected (0.03 sec)
```

MySQL 25: Consolas Salas de Juegos

```
mysql> SHOW tables;
+---------------------+
| Tables_in_laforja   |
+---------------------+
| tadministrativo     |
| tauxiliar           |
| tcibercafe          |
| tcliente            |
| tclientepersonal    |
| tclientesalajuegos  |
| tclientetorneo      |
| tconsolas           |
| tconsolasalasjuegos |
| tequipos            |
| tgerente            |
| tincidencias        |
| tlimpiador          |
| tlocal              |
| tmantenimiento      |
| tordenadores        |
| tpersonal           |
| tpersonaltorneo     |
| tsaladescanso       |
| tsalajuegos         |
| ttaller             |
| ttecnico            |
| ttienda             |
| ttorneos            |
+---------------------+
24 rows in set (0.01 sec)
```

MySQL 26: Tablas Creadas

6- Cambios realizados de Fase 1,2 y 3 a Fase 4

Se han modificado un par de relaciones, ya que se estaban duplicando, también se han modificado los atributos, ya que había redundancia de datos a la hora de crear las tablas. Se ha añadido también una especialización de equipos, para aglutinar consolas y ordenadores.

Al final se ha modificado el diagrama entidad-relación por las modificaciones creadas.

7- Tratamiento de Datos

- 7.1- Modificar dos tablas. Eliminando, modificando o añadiendo algún atributo

En mi caso tengo que modificar la mayoría de tablas donde la primary key es tipo int, pero no le puse auto increment, por lo que tendría que meterlo constantemente, por lo que empezamos con esa modificación y mato dos pájaros de un tiro. Ya que son claves primarias y en algunas son claves foráneas, empezamos poniendo SET FOREIGN_KEY_CHECKS=0;, para que nos permita realizar la modificación y así no tener que desenlazarlo

la idea es ir tabla por tabla cambiando la clave primaria, siendo las tablas que tenemos que modificar las siguientes, TCLIENTE, TGERENTE, TADMINISTRATIVO, TAUXILIAR, TMANTENIMIENTO, TTECNICO, TLIMPIADOR, TEQUIPOS, TORDENADORES, TCONSOLAS, TTORNEOS, TINCIDENCIAS, TTALLER, TTIENDA, TSALAJUEGOS, TCIBERCAFE, TSALADESCANSO, TLOCAL, TTORNEOS, TClienteSalaJuegos, TClientePersonal, TClienteTorneo, TPersonalTorneo, TConsolaSalasJuegos, también por comodidad hemos modificado el primary key de la tabla local por un varchar.

El Script de los cambios sería el siguiente

```
SET FOREIGN_KEY_CHECKS=0;

ALTER TABLE TGERENTE MODIFY ID_GER INT AUTO_INCREMENT;

ALTER TABLE TADMINISTRATIVO MODIFY ID_ADM INT
AUTO_INCREMENT;

ALTER TABLE TAUXILIAR MODIFY ID_AUX INT AUTO_INCREMENT;

ALTER TABLE TMANTENIMIENTO MODIFY ID_MAN INT
AUTO_INCREMENT;

ALTER TABLE TTECNICO MODIFY ID_TEC INT AUTO_INCREMENT;

ALTER TABLE TLIMPIADOR MODIFY ID_LIM INT AUTO_INCREMENT;

ALTER TABLE TEQUIPOS MODIFY ID_EQUIPO INT AUTO_INCREMENT;
```

```
ALTER TABLE TORDENADORES MODIFY ID_ORDENADOR INT
AUTO_INCREMENT;

ALTER TABLE TCONSOLAS MODIFY ID_CONSOLAS INT
AUTO_INCREMENT;

ALTER TABLE TTORNEOS MODIFY ID_TORNEO INT AUTO_INCREMENT;

ALTER TABLE TINCIDENCIAS MODIFY ID_PARTE INT AUTO_INCREMENT;

ALTER TABLE TTALLER MODIFY ID_TALLER INT AUTO_INCREMENT;

ALTER TABLE TTIENDA MODIFY ID_TIENDA INT AUTO_INCREMENT;

ALTER TABLE TSALAJUEGOS MODIFY ID_SALAJUEGOS INT
AUTO_INCREMENT;

ALTER TABLE TCIBERCAFE MODIFY ID_CIBERCAFE INT
AUTO_INCREMENT;

ALTER TABLE TSALADESCANSO MODIFY ID_SALADESCANSO INT
AUTO_INCREMENT;

ALTER TABLE TClienteSalaJuegos MODIFY ID_CLSJ INT
AUTO_INCREMENT;

ALTER TABLE TClientePersonal MODIFY ID_CLP INT AUTO_INCREMENT;

ALTER TABLE TClienteTorneo MODIFY ID_CLT INT AUTO_INCREMENT;

ALTER TABLE TPersonalTorneo MODIFY ID_PT INT AUTO_INCREMENT;

ALTER TABLE TConsolaSalasJuegos MODIFY ID_CSJ INT
AUTO_INCREMENT;

ALTER TABLE tlocal MODIFY CIF VARCHAR(255);

ALTER TABLE ttorneos MODIFY fecha_torneo VARCHAR(255);

SET FOREIGN_KEY_CHECKS=1;
```

lo introducimos con el CLI de Mysql y saco captura de pantalla, no pongo todas, porque se pilla cerca de tres hojas para los cambios, pero se podrán verificar los cambios realizados en las siguientes capturas de las inserciones

```
mysql> SET FOREIGN_KEY_CHECKS=0;
Query OK, 0 rows affected (0.00 sec)

mysql> ALTER TABLE TGERENTE MODIFY ID_GER INT AUTO_INCREMENT;
Query OK, 0 rows affected (0.04 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> ALTER TABLE TADMINISTRATIVO MODIFY ID_ADM INT AUTO_INCREMENT;
Query OK, 0 rows affected (0.04 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> ALTER TABLE TAUXILIAR MODIFY ID_AUX INT AUTO_INCREMENT;
Query OK, 0 rows affected (0.04 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> ALTER TABLE TMANTENIMIENTO MODIFY ID_MAN INT AUTO_INCREMENT;
Query OK, 0 rows affected (0.04 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> ALTER TABLE TTECNICO MODIFY ID_TEC INT AUTO_INCREMENT;
Query OK, 0 rows affected (0.03 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> ALTER TABLE TLIMPIADOR MODIFY ID_LIM INT AUTO_INCREMENT;
Query OK, 0 rows affected (0.03 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> ALTER TABLE TEQUIPOS MODIFY ID_EQUIPO INT AUTO_INCREMENT;
Query OK, 0 rows affected (0.04 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> ALTER TABLE TORDENADORES MODIFY ID_ORDENADOR INT AUTO_INCREMENT;
Query OK, 0 rows affected (0.04 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> ALTER TABLE TCONSOLAS MODIFY ID_CONSOLAS INT AUTO_INCREMENT;
Query OK, 0 rows affected (0.04 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> ALTER TABLE TTORNEOS MODIFY ID_TORNEO INT AUTO_INCREMENT;
Query OK, 0 rows affected (0.03 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> ALTER TABLE TINCIDENCIAS MODIFY ID_PARTE INT AUTO_INCREMENT;
Query OK, 0 rows affected (0.05 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> ALTER TABLE TTALLER MODIFY ID_TALLER INT AUTO_INCREMENT;
Query OK, 0 rows affected (0.07 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

Cambios 1: se hace cambio en las tablas creadas en fase anterior

- 7.2- Realizar la inserción de tres registros por tabla.

comenzaremos con la tabla cliente, dejando por ahora el campo de DNI_personal vacío, que lo rellenaremos a posterior en la parte de modificación cuando ya tengamos al personal introducido con un update

```
mysql> INSERT INTO tcliente(DNI_cl,nombre_cl,apellido1_cl,apellido2_cl,email_cl,telefono_cl)
    -> VALUES
    -> ("78639466M","Pilar","Guigou","Suarez","pguigousuarez@gmail.com",674143240),
    -> ("42194407X","Almudena","Hernandez","Hernandez","almu31.12.1987@gmail.com",652428457),
    -> ("45767987B","Rosa Maria","Moreno","Gutierrez","rosamg1979@gmail.com",676530483);
Query OK, 3 rows affected (0.00 sec)
Records: 3  Duplicates: 0  Warnings: 0

mysql> select * from tcliente;
+------------+-----------+------------+--------------+--------------+--------------------------+-------------+--------------+
| id_usuario | DNI_cl    | nombre_cl  | apellido1_cl | apellido2_cl | email_cl                 | telefono_cl | DNI_personal |
+------------+-----------+------------+--------------+--------------+--------------------------+-------------+--------------+
|          1 | 78639466M | Pilar      | Guigou       | Suarez       | pguigousuarez@gmail.com  |   674143240 | NULL         |
|          2 | 42194407X | Almudena   | Hernandez    | Hernandez    | almu31.12.1987@gmail.com |   652428457 | NULL         |
|          3 | 45767987B | Rosa Maria | Moreno       | Gutierrez    | rosamg1979@gmail.com     |   676530483 | NULL         |
+------------+-----------+------------+--------------+--------------+--------------------------+-------------+--------------+
3 rows in set (0.00 sec)
```

Inserción 1: insertamos en clientes

Pasamos a rellenar la tabla de local, estableciendo ya los ids de usuarios introducidos anteriormente

```
mysql> INSERT INTO tlocal(CIF,Direccion,telefono,web,horario, id_usuario)
    -> VALUES
    -> ("35713575-Taller","Calle Jamaica 16",928142611,"HTTPS://nodomainyet.com","10 a 22",1),
    -> ("35713575-Tienda","Calle Jamaica 16",928142611,"HTTPS://nodomainyet.com","10 a 22",2),
    -> ("35713575-Cibercafe","Calle Jamaica 16",928142611,"HTTPS://nodomainyet.com","10 a 22",3);
Query OK, 3 rows affected (0.00 sec)
Records: 3  Duplicates: 0  Warnings: 0

mysql> select * from tlocal;
+--------------------+------------------+-----------+-------------------------+---------+-----------+------------+
| CIF                | Direccion        | Telefono  | web                     | horario | cod_local | id_usuario |
+--------------------+------------------+-----------+-------------------------+---------+-----------+------------+
| 35713575-Cibercafe | Calle Jamaica 16 | 928142611 | HTTPS://nodomainyet.com | 10 a 22 |         3 |          3 |
| 35713575-Taller    | Calle Jamaica 16 | 928142611 | HTTPS://nodomainyet.com | 10 a 22 |         1 |          1 |
| 35713575-Tienda    | Calle Jamaica 16 | 928142611 | HTTPS://nodomainyet.com | 10 a 22 |         2 |          2 |
+--------------------+------------------+-----------+-------------------------+---------+-----------+------------+
3 rows in set (0.00 sec)

mysql>
```

Inserción 2: insertamos en local

pasamos a rellenar la tabla personal, aquí tengo que meter bastantes para después separarlos en sus especializaciones

```
mysql> INSERT INTO tpersonal(DNI_personal,cargo,nombre_per,apellido1_per,apellido2_per,email_per,telefono_per)
    -> VALUES
    -> ("42227069N","Gerente","Cándido","Pérez","Verwer","candidonpv@gmail.com",656342086),
    -> ("45749494H","Gerente","Lucia","Morales","Pérez","luciambito@gmail.com",65242052),
    -> ("12345678S","Gerente","Iris","Pérez","Morales","Iris2023@gmail.com",32165498),
    -> ("631751698","Administrativo","Eadmund","Prover","Bertolaccini","ebertolaccini0@upenn.edu",17835453),
    -> ("788769809","Administrativo","Elisabeth","Banger","Daunay","edaunay1@icq.com",29860293),
    -> ("751253619","Administrativo","Germaine","Haig","Scurrey","gscurrey2@a8.net",18448679),
    -> ("572078928","Auxiliar","Dani","Armatage","Braycotton","dbraycotton3@sohu.com",22460131),
    -> ("182410414","Auxiliar","Clarissa","Miguet","Larkings","clarkings4@dot.gov",12856342),
    -> ("864069323","Auxiliar","Kamila","Crickmore","Garrigan","kgarrigan5@dedecms.com",57284730),
    -> ("396157927","Mantenimiento","Alisha","Tomanek","Stabbins","astabbins6@reddit.com",48724671),
    -> ("802845095","Mantenimiento","Rhodie","Prestie","Le-Good","rlegood7@npr.org",91875996),
    -> ("510284963","Mantenimiento","Charin","Kment","Streeting","cstreeting8@vistaprint.com",14473110),
    -> ("343715166","Tecnico","Donnamarie","Djekovic","Deerness","ddeerness9@shinystat.com",57143061),
    -> ("569020898","Tecnico","Barth","Knevit","Foden","bfodena@hugedomains.com",3506547),
    -> ("881459082","Tecnico","Patton","Tidmas","Semiraz","psemirazb@i2i.jp",80219784),
    -> ("792614188","Limpiador","Maximilian","Adnet","Mathey","mmatheyc@about.me",32342406),
    -> ("852540320","Limpiador","Neilla","Croot","Riach","nriachd@timesonline.co.uk",25639512),
    -> ("896330468","Limpiador","Inness","Kilkenny","Pockey","ipockeye@sourceforge.net",91538029);
Query OK, 18 rows affected (0.00 sec)
Records: 18  Duplicates: 0  Warnings: 0

mysql> select * from tpersonal;
+--------------+--------------+-------------+--------------+--------------+-------------------------+--------------+------------+-----------+
| DNI_personal | cargo        | nombre_per  | apellido1_per | apellido2_per | email_per               | telefono_per | id_usuario | cod_local |
+--------------+--------------+-------------+--------------+--------------+-------------------------+--------------+------------+-----------+
| 12345678S    | Gerente      | Iris        | Pérez        | Morales      | Iris2023@gmail.com      |     32165498 |       NULL |      NULL |
| 182410414    | Auxiliar     | Clarissa    | Miguet       | Larkings     | clarkings4@dot.gov      |     12856342 |       NULL |      NULL |
| 343715166    | Tecnico      | Donnamarie  | Djekovic     | Deerness     | ddeerness9@shinystat.com |     57143061 |       NULL |      NULL |
| 396157927    | Mantenimiento | Alisha      | Tomanek      | Stabbins     | astabbins6@reddit.com   |     48724671 |       NULL |      NULL |
| 42227069N    | Gerente      | Cándido     | Pérez        | Verwer       | candidonpv@gmail.com    |    656342086 |       NULL |      NULL |
| 45749494H    | Gerente      | Lucia       | Morales      | Pérez        | luciambito@gmail.com    |     65242052 |       NULL |      NULL |
| 510284963    | Mantenimiento | Charin      | Kment        | Streeting    | cstreeting8@vistaprint.com |     14473110 |       NULL |      NULL |
| 569020898    | Tecnico      | Barth       | Knevit       | Foden        | bfodena@hugedomains.com |      3506547 |       NULL |      NULL |
| 572078928    | Auxiliar     | Dani        | Armatage     | Braycotton   | dbraycotton3@sohu.com   |     22460131 |       NULL |      NULL |
| 631751698    | Administrativo | Eadmund     | Prover       | Bertolaccini | ebertolaccini0@upenn.edu |     17835453 |       NULL |      NULL |
| 751253619    | Administrativo | Germaine    | Haig         | Scurrey      | gscurrey2@a8.net        |     18448679 |       NULL |      NULL |
| 788769809    | Administrativo | Elisabeth   | Banger       | Daunay       | edaunay1@icq.com        |     29860293 |       NULL |      NULL |
| 792614188    | Limpiador    | Maximilian  | Adnet        | Mathey       | mmatheyc@about.me       |     32342406 |       NULL |      NULL |
| 802845095    | Mantenimiento | Rhodie      | Prestie      | Le-Good      | rlegood7@npr.org        |     91875996 |       NULL |      NULL |
| 852540320    | Limpiador    | Neilla      | Croot        | Riach        | nriachd@timesonline.co.uk |     25639512 |       NULL |      NULL |
| 864069323    | Auxiliar     | Kamila      | Crickmore    | Garrigan     | kgarrigan5@dedecms.com  |     57284730 |       NULL |      NULL |
| 881459082    | Tecnico      | Patton      | Tidmas       | Semiraz      | psemirazb@i2i.jp        |     80219784 |       NULL |      NULL |
| 896330468    | Limpiador    | Inness      | Kilkenny     | Pockey       | ipockeye@sourceforge.net |     91538029 |       NULL |      NULL |
+--------------+--------------+-------------+--------------+--------------+-------------------------+--------------+------------+-----------+
18 rows in set (0.00 sec)
```

Inserción 3: insertamos en personal

Insertamos datos en la tabla Gerente

```
mysql> INSERT INTO tgerente(cargo,DNI_personal)
    -> VALUES
    -> ("Gerente","42227069N"),
    -> ("Gerente","45749494H"),
    -> ("Gerente","12345678S");
Query OK, 3 rows affected (0.00 sec)
Records: 3  Duplicates: 0  Warnings: 0

mysql> select * from tgerente;
+--------+---------+--------------+
| ID_GER | cargo   | DNI_personal |
+--------+---------+--------------+
|      1 | Gerente | 42227069N    |
|      2 | Gerente | 45749494H    |
|      3 | Gerente | 12345678S    |
+--------+---------+--------------+
3 rows in set (0.00 sec)
```

Inserción 4: insertamos en gerente

Insertamos datos en la tabla auxiliar

```
mysql> INSERT INTO tauxiliar(cargo,DNI_personal)
    -> VALUES
    -> ("Auxiliar","572078928"),
    -> ("Auxiliar","182410414"),
    -> ("Auxiliar","864069323");
Query OK, 3 rows affected (0.00 sec)
Records: 3  Duplicates: 0  Warnings: 0

mysql> select * from tauxiliar;
+--------+----------+--------------+
| ID_AUX | cargo    | DNI_personal |
+--------+----------+--------------+
|      1 | Auxiliar | 572078928    |
|      2 | Auxiliar | 182410414    |
|      3 | Auxiliar | 864069323    |
+--------+----------+--------------+
3 rows in set (0.00 sec)
```

Inserción 5: insertamos en auxiliar

Insertamos datos en la tabla administrativo

```
mysql> INSERT INTO tadministrativo(cargo,dni_personal,id_ger,id_aux)
    -> VALUES
    -> ("Administrativo","631751698",1,2),
    -> ("Administrativo","788769809",2,1),
    -> ("Administrativo","751253619",3,3);
Query OK, 3 rows affected (0.00 sec)
Records: 3  Duplicates: 0  Warnings: 0

mysql> select * from tadministrativo;
+--------+---------------+--------------+--------+--------+
| ID_ADM | cargo         | DNI_personal | ID_ger | ID_aux |
+--------+---------------+--------------+--------+--------+
|      1 | Administrativo | 631751698    |      1 |      2 |
|      2 | Administrativo | 788769809    |      2 |      1 |
|      3 | Administrativo | 751253619    |      3 |      3 |
+--------+---------------+--------------+--------+--------+
3 rows in set (0.00 sec)
```

Inserción 6: insertamos en administrativo

Insertamos datos en la tabla mantenimiento

```
mysql> INSERT INTO tmantenimiento(cargo,DNI_personal)
    -> VALUES
    -> ("Mantenimiento","396157927"),
    -> ("Mantenimiento","802845095"),
    -> ("Mantenimiento","510284963");
Query OK, 3 rows affected (0.00 sec)
Records: 3  Duplicates: 0  Warnings: 0

mysql> select * from tmantenimiento;
+--------+---------------+--------------+
| ID_MAN | cargo         | DNI_personal |
+--------+---------------+--------------+
|      1 | Mantenimiento | 396157927    |
|      2 | Mantenimiento | 802845095    |
|      3 | Mantenimiento | 510284963    |
+--------+---------------+--------------+
3 rows in set (0.00 sec)
```

Inserción 7: insertamos mantenimiento

Insertamos datos en la tabla técnico

```
mysql> INSERT INTO ttecnico(cargo,dni_personal,id_usuario)
    -> VALUES
    -> ("Tecnico","343715166",3),
    -> ("Tecnico","569020898",1),
    -> ("Tecnico","881459082",2);
Query OK, 3 rows affected (0.00 sec)
Records: 3  Duplicates: 0  Warnings: 0

mysql> select * from ttecnico;
+--------+---------+--------------+------------+
| ID_TEC | cargo   | DNI_personal | id_usuario |
+--------+---------+--------------+------------+
|      1 | Tecnico | 343715166    |          3 |
|      2 | Tecnico | 569020898    |          1 |
|      3 | Tecnico | 881459082    |          2 |
+--------+---------+--------------+------------+
3 rows in set (0.00 sec)
```

Inserción 8: insertamos en técnico

Insertamos datos en la tabla limpiador

```
mysql> INSERT INTO tlimpiador(cargo,DNI_personal)
    -> VALUES
    -> ("Limpiador","792614188"),
    -> ("Limpiador","852540320"),
    -> ("Limpiador","896330468");
Query OK, 3 rows affected (0.00 sec)
Records: 3  Duplicates: 0  Warnings: 0

mysql> select * from tlimpiador;
+--------+-----------+--------------+-----------+
| ID_LIM | cargo     | DNI_personal | cod_local |
+--------+-----------+--------------+-----------+
|      1 | Limpiador | 792614188    |      NULL |
|      2 | Limpiador | 852540320    |      NULL |
|      3 | Limpiador | 896330468    |      NULL |
+--------+-----------+--------------+-----------+
3 rows in set (0.00 sec)
```

Inserción 9: insertamos en limpiador

Insertamos datos dentro de equipos

```
mysql> INSERT INTO tequipos(Num_serie,tipo_equipo,unidades,cod_local)
    -> VALUES
    -> ("pc01","Ordenador",5,3),
    -> ("PS501","Play Station",5,3),
    -> ("XBOX-X","Xbox One X",5,3);
Query OK, 3 rows affected (0.00 sec)
Records: 3  Duplicates: 0  Warnings: 0

mysql>
mysql> select * from tequipos;
+-----------+-----------+--------------+----------+-----------+
| ID_EQUIPO | Num_serie | tipo_equipo  | unidades | cod_local |
+-----------+-----------+--------------+----------+-----------+
|         1 | pc01      | Ordenador    |        5 |         3 |
|         2 | PS501     | Play Station |        5 |         3 |
|         3 | XBOX-X    | Xbox One X   |        5 |         3 |
+-----------+-----------+--------------+----------+-----------+
3 rows in set (0.00 sec)
```

Inserción 10: insertamos en equipos

Insertamos datos dentro de ordenadores

```
mysql> INSERT INTO tordenadores(id_equipo,cod_local)
    -> VALUES
    -> (1,3),
    -> (2,2),
    -> (3,2);
Query OK, 3 rows affected (0.00 sec)
Records: 3  Duplicates: 0  Warnings: 0

mysql> select * from tordenadores;
+--------------+-----------+-----------+
| ID_ORDENADOR | Id_equipo | cod_local |
+--------------+-----------+-----------+
|            1 |         1 |         3 |
|            2 |         2 |         2 |
|            3 |         3 |         2 |
+--------------+-----------+-----------+
3 rows in set (0.00 sec)
```

Inserción 11: insertamos en ordenadores

Insertamos datos dentro de consolas

```
mysql> INSERT INTO tconsolas(id_equipo,cod_local)
    -> VALUES
    -> (2,2),
    -> (3,2),
    -> (2,3);
Query OK, 3 rows affected (0.00 sec)
Records: 3  Duplicates: 0  Warnings: 0

mysql> select * from tconsolas;
+-------------+-----------+-----------+
| ID_CONSOLAS | Id_equipo | cod_local |
+-------------+-----------+-----------+
|           1 |         2 |         2 |
|           2 |         3 |         2 |
|           3 |         2 |         3 |
+-------------+-----------+-----------+
3 rows in set (0.00 sec)
```

Inserción 12: insertamos en consolas

Insertamos datos dentro de incidencias

```
mysql> INSERT INTO tincidencias(cod_local, observaciones, id_man,id_adm)
    -> VALUES
    -> (3,"cable pelado al aire",2,3),
    -> (1,"enchufe roto",3,1),
    -> (2,"aire roto",1,2);
Query OK, 3 rows affected (0.00 sec)
Records: 3  Duplicates: 0  Warnings: 0

mysql> select * from tincidencias;
+----------+-----------+---------------------+--------+--------+
| ID_PARTE | cod_local | observaciones       | ID_man | ID_adm |
+----------+-----------+---------------------+--------+--------+
|        1 |         3 | cable pelado al aire |      2 |      3 |
|        2 |         1 | enchufe roto        |      3 |      1 |
|        3 |         2 | aire roto           |      1 |      2 |
+----------+-----------+---------------------+--------+--------+
3 rows in set (0.00 sec)
```

Inserción 13: insertamos en incidencias

Insertamos datos dentro de taller

```
mysql> INSERT INTO ttaller(cod_local, horario, id_tec,id_usuario)
    -> VALUES
    -> (3,"10 a 21",2,1),
    -> (3,"10 a 21",3,2),
    -> (3,"10 a 21",1,3);
Query OK, 3 rows affected (0.00 sec)
Records: 3  Duplicates: 0  Warnings: 0

mysql> select * from ttaller;
+-----------+-----------+---------+--------+------------+
| ID_TALLER | cod_local | horario | ID_tec | id_usuario |
+-----------+-----------+---------+--------+------------+
|         1 |         3 | 10 a 21 |      2 |          1 |
|         2 |         3 | 10 a 21 |      3 |          2 |
|         3 |         3 | 10 a 21 |      1 |          3 |
+-----------+-----------+---------+--------+------------+
3 rows in set (0.00 sec)
```

Inserción 14: insertamos en taller

Insertamos datos dentro de tienda

```
mysql> INSERT INTO ttienda(cod_local,horario,id_usuario)
    -> VALUES
    -> (3,"10 a 21",1),
    -> (2,"10 a 21",1),
    -> (1,"10 a 21",2);
Query OK, 3 rows affected (0.00 sec)
Records: 3  Duplicates: 0  Warnings: 0

mysql> select * from ttienda;
+-----------+-----------+---------+------------+
| ID_TIENDA | cod_local | horario | id_usuario |
+-----------+-----------+---------+------------+
|         1 |         3 | 10 a 21 |          1 |
|         2 |         2 | 10 a 21 |          1 |
|         3 |         1 | 10 a 21 |          2 |
+-----------+-----------+---------+------------+
3 rows in set (0.00 sec)
```

Inserción 15: insertamos en tienda

Insertamos datos dentro de salajuegos

```
mysql> INSERT INTO tsalajuegos(cod_local,horario,id_consolas)
    -> VALUES
    -> (3,"10 a 21",2),
    -> (3,"10 a 21",3),
    -> (3,"10 a 21",3);
Query OK, 3 rows affected (0.00 sec)
Records: 3  Duplicates: 0  Warnings: 0

mysql> select * from tsalajuegos;
+---------------+-----------+---------+-------------+
| ID_SALAJUEGOS | cod_local | horario | ID_consolas |
+---------------+-----------+---------+-------------+
|             1 |         3 | 10 a 21 |           2 |
|             2 |         3 | 10 a 21 |           3 |
|             3 |         3 | 10 a 21 |           3 |
+---------------+-----------+---------+-------------+
3 rows in set (0.00 sec)
```

Inserción 16: insertamos en salajuegos

Insertamos datos dentro de cibercafé

```
mysql> INSERT INTO tcibercafe(cod_local,horario,id_equipo)
    -> VALUES
    -> (3,"10 a 21",1),
    -> (3,"10 a 21",1),
    -> (3,"10 a 21",1);
Query OK, 3 rows affected (0.00 sec)
Records: 3  Duplicates: 0  Warnings: 0

mysql> select * from tcibercafe;
+--------------+-----------+---------+-----------+
| ID_CIBERCAFE | cod_local | horario | id_equipo |
+--------------+-----------+---------+-----------+
|            1 |         3 | 10 a 21 |         1 |
|            2 |         3 | 10 a 21 |         1 |
|            3 |         3 | 10 a 21 |         1 |
+--------------+-----------+---------+-----------+
3 rows in set (0.00 sec)
```

Inserción 17: insertamos en cibercafé

Insertamos datos dentro de saladescanso

```
mysql> INSERT INTO tsaladescanso(cod_local, horario)
    -> VALUES
    -> (3,"10 a 21"),
    -> (2,"10 a 21"),
    -> (1,"10 a 21");
Query OK, 3 rows affected (0.00 sec)
Records: 3  Duplicates: 0  Warnings: 0

mysql> select * from tsaladescanso;
+-----------------+-----------+---------+
| ID_SALADESCANSO | cod_local | horario |
+-----------------+-----------+---------+
|               1 |         3 | 10 a 21 |
|               2 |         2 | 10 a 21 |
|               3 |         1 | 10 a 21 |
+-----------------+-----------+---------+
3 rows in set (0.00 sec)
```

Inserción 18: insertamos en saladescanso

Insertamos datos dentro de clientesalajuegos

```
mysql> INSERT INTO tclientesalajuegos(id_salajuegos,id_usuario)
    -> VALUES
    -> (2,2),
    -> (3,3),
    -> (1,1);
Query OK, 3 rows affected (0.00 sec)
Records: 3  Duplicates: 0  Warnings: 0

mysql> select * from tclientesalajuegos;
+---------+---------------+------------+
| ID_CLSJ | ID_salajuegos | ID_usuario |
+---------+---------------+------------+
|       1 |             2 |          2 |
|       2 |             3 |          3 |
|       3 |             1 |          1 |
+---------+---------------+------------+
3 rows in set (0.00 sec)
```

Inserción 19: insertamos en clientesalajuegos

Insertamos datos dentro de clientepersonal

```
mysql> INSERT INTO tclientepersonal(DNI_personal,id_usuario)
    -> VALUES
    -> ("182410414",2),
    -> ("396157927",1),
    -> ("792614188",3);
Query OK, 3 rows affected (0.00 sec)
Records: 3  Duplicates: 0  Warnings: 0

mysql> select * from tclientepersonal;
+--------+--------------+------------+
| ID_CLP | DNI_personal | ID_usuario |
+--------+--------------+------------+
|      1 | 182410414    |          2 |
|      2 | 396157927    |          1 |
|      3 | 792614188    |          3 |
+--------+--------------+------------+
3 rows in set (0.00 sec)
```

Inserción 20: insertamos en clientepersonal

Insertamos datos dentro de clientetorneos

```
mysql> INSERT INTO tclientetorneo(id_torneo,id_usuario)
    -> VALUES
    -> (3,1),
    -> (2,1),
    -> (1,2);
Query OK, 3 rows affected (0.00 sec)
Records: 3  Duplicates: 0  Warnings: 0

mysql> select * from tclientetorneo;
+--------+-----------+------------+
| ID_CLT | ID_torneo | ID_usuario |
+--------+-----------+------------+
|      4 |         3 |          1 |
|      5 |         2 |          1 |
|      6 |         1 |          2 |
+--------+-----------+------------+
3 rows in set (0.00 sec)
```

Inserción 21: insertamos en clientetorneo

Insertamos datos dentro de personaltorneo

```
mysql> INSERT INTO tpersonaltorneo(id_torneo,DNI_personal)
    -> VALUES
    -> (2,"631751698"),
    -> (1,"12345678S"),
    -> (3,"569020898");
Query OK, 3 rows affected (0.00 sec)
Records: 3  Duplicates: 0  Warnings: 0

mysql> select * from tpersonaltorneo;
+-------+-----------+--------------+
| ID_PT | ID_torneo | DNI_personal |
+-------+-----------+--------------+
|     1 |         2 | 631751698    |
|     2 |         1 | 12345678S    |
|     3 |         3 | 569020898    |
+-------+-----------+--------------+
3 rows in set (0.00 sec)
```

Inserción 22: insertamos en personaltorneo

Insertamos datos dentro de consolasalasjuegos

```
mysql> INSERT INTO tconsolasalasjuegos(id_salajuegos,id_consola)
    -> VALUES
    -> (2,3),
    -> (1,2),
    -> (3,2);
Query OK, 3 rows affected (0.00 sec)
Records: 3  Duplicates: 0  Warnings: 0

mysql>
mysql> select * from tconsolasalasjuegos;
+--------+---------------+------------+
| ID_CSJ | ID_salajuegos | ID_consola |
+--------+---------------+------------+
|      1 |             2 |          3 |
|      2 |             1 |          2 |
|      3 |             3 |          2 |
+--------+---------------+------------+
3 rows in set (0.00 sec)
```

Inserción 23: insertamos en consolasalasjuegos

- 7.3- Realizar tres modificaciones de registros en tres tablas diferentes

Haremos un par de cambios en la tabla cliente, otro par en la tabla personal y en la tabla local

En la captura se verá el antes y el después

Cambios en la tabla cliente

```
mysql> select * from tcliente;
+------------+-----------+------------+--------------+--------------+--------------------------+------------+--------------+
| id_usuario | DNI_cl    | nombre_cl  | apellido1_cl | apellido2_cl | email_cl                 | telefono_cl | DNI_personal |
+------------+-----------+------------+--------------+--------------+--------------------------+------------+--------------+
|          1 | 78639466M | Pilar      | Guigou       | Suarez       | pguigousuarez@gmail.com  |   674143240 | NULL         |
|          2 | 42194407X | Almudena   | Hernandez    | Hernandez    | almu31.12.1987@gmail.com |   652428457 | NULL         |
|          3 | 45767987B | Rosa Maria | Moreno       | Gutierrez    | rosamg1979@gmail.com     |   676530483 | NULL         |
+------------+-----------+------------+--------------+--------------+--------------------------+------------+--------------+
3 rows in set (0.00 sec)

mysql> update tCliente set DNI_personal="42227069N" where (id_usuario=1);
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> update tCliente set DNI_personal="45779494H" where (id_usuario=2);
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> update tCliente set DNI_personal="12345678S" where (id_usuario=3);
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select * from tcliente;
+------------+-----------+------------+--------------+--------------+--------------------------+------------+--------------+
| id_usuario | DNI_cl    | nombre_cl  | apellido1_cl | apellido2_cl | email_cl                 | telefono_cl | DNI_personal |
+------------+-----------+------------+--------------+--------------+--------------------------+------------+--------------+
|          1 | 78639466M | Pilar      | Guigou       | Suarez       | pguigousuarez@gmail.com  |   674143240 | 42227069N    |
|          2 | 42194407X | Almudena   | Hernandez    | Hernandez    | almu31.12.1987@gmail.com |   652428457 | 45779494H    |
|          3 | 45767987B | Rosa Maria | Moreno       | Gutierrez    | rosamg1979@gmail.com     |   676530483 | 12345678S    |
+------------+-----------+------------+--------------+--------------+--------------------------+------------+--------------+
3 rows in set (0.00 sec)
```

Cambio 1: se cambia la tabla cliente

Cambios en la tabla personal

```
mysql> select * from tpersonal;
+-------------+---------------+-----------+-------------+-------------+---------------------------+--------------+------------+----------+
| DNI_personal | cargo         | nombre_per | apellido1_per | apellido2_per | email_per                 | telefono_per | id_usuario | cod_local |
+-------------+---------------+-----------+-------------+-------------+---------------------------+--------------+------------+----------+
| 12345678S   | Gerente       | Iris      | Pérez       | Morales     | Iris2023@gmail.com        |     32165498 |       NULL |      NULL |
| 182410414   | Auxiliar      | Clarissa  | Miguet      | Larkings    | clarkings4@dot.gov        |     12856342 |       NULL |      NULL |
| 343715166   | Tecnico       | Donnamarie | Djekovic    | Deerness    | ddeerness9@shinystat.com  |     57143061 |       NULL |      NULL |
| 396157927   | Mantenimiento | Alisha    | Tomanek     | Stabbins    | astabbins6@reddit.com     |     48724671 |       NULL |      NULL |
| 42227069N   | Gerente       | Cándido   | Pérez       | Verwer      | candidonpv@gmail.com      |    656342086 |       NULL |      NULL |
| 45749494H   | Gerente       | Lucía     | Morales     | Pérez       | luciambito@gmail.com      |     65242052 |       NULL |      NULL |
| 510284963   | Mantenimiento | Charin    | Kment       | Streeting   | cstreeting8@vistaprint.com |     14473110 |       NULL |      NULL |
| 569020898   | Tecnico       | Barth     | Knevit      | Foden       | bfodena@hugedomains.com   |      3506547 |       NULL |      NULL |
| 572078928   | Auxiliar      | Dani      | Armatage    | Braycotton  | dbraycotton3@sohu.com     |     22460131 |       NULL |      NULL |
| 631751698   | Administrativo | Eadmund   | Prover      | Bertolaccini | ebertolaccini0@upenn.edu  |     17835453 |       NULL |      NULL |
| 751253619   | Administrativo | Germaine  | Haig        | Scurrey     | gscurrey2@a8.net          |     18448679 |       NULL |      NULL |
| 788769809   | Administrativo | Elisabeth | Banger      | Daunay      | edaunay1@icq.com          |     29860293 |       NULL |      NULL |
| 792614188   | Limpiador     | Maximilian | Adnet       | Mathey      | mmatheyc@about.me         |     32342406 |       NULL |      NULL |
| 802845095   | Mantenimiento | Rhodie    | Prestie     | Le-Good     | rlegood7@npr.org          |     91875996 |       NULL |      NULL |
| 852540320   | Limpiador     | Neilla    | Croot       | Riach       | nriachd@timesonline.co.uk |     25639512 |       NULL |      NULL |
| 864069323   | Auxiliar      | Kamila    | Crickmore   | Garrigan    | kgarrigan5@dedecms.com    |     57284730 |       NULL |      NULL |
| 881459082   | Tecnico       | Patton    | Tidmas      | Semiraz     | psemirazb@i2i.jp          |     80219784 |       NULL |      NULL |
| 896330468   | Limpiador     | Inness    | Kilkenny    | Pockey      | ipockeye@sourceforge.net  |     91538029 |       NULL |      NULL |
+-------------+---------------+-----------+-------------+-------------+---------------------------+--------------+------------+----------+
18 rows in set (0.00 sec)

mysql> update tpersonal set id_usuario = 1 where (DNI_personal="45779494H");
Query OK, 0 rows affected (0.00 sec)
Rows matched: 0  Changed: 0  Warnings: 0

mysql> update tpersonal set id_usuario = 3 where (DNI_personal="788769809");
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> update tpersonal set id_usuario = 2 where (DNI_personal="864069323");
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select * from tpersonal;
+-------------+---------------+-----------+-------------+-------------+---------------------------+--------------+------------+----------+
| DNI_personal | cargo         | nombre_per | apellido1_per | apellido2_per | email_per                 | telefono_per | id_usuario | cod_local |
+-------------+---------------+-----------+-------------+-------------+---------------------------+--------------+------------+----------+
| 12345678S   | Gerente       | Iris      | Pérez       | Morales     | Iris2023@gmail.com        |     32165498 |       NULL |      NULL |
| 182410414   | Auxiliar      | Clarissa  | Miguet      | Larkings    | clarkings4@dot.gov        |     12856342 |       NULL |      NULL |
| 343715166   | Tecnico       | Donnamarie | Djekovic    | Deerness    | ddeerness9@shinystat.com  |     57143061 |       NULL |      NULL |
| 396157927   | Mantenimiento | Alisha    | Tomanek     | Stabbins    | astabbins6@reddit.com     |     48724671 |       NULL |      NULL |
| 42227069N   | Gerente       | Cándido   | Pérez       | Verwer      | candidonpv@gmail.com      |    656342086 |       NULL |      NULL |
| 45749494H   | Gerente       | Lucía     | Morales     | Pérez       | luciambito@gmail.com      |     65242052 |       NULL |      NULL |
| 510284963   | Mantenimiento | Charin    | Kment       | Streeting   | cstreeting8@vistaprint.com |     14473110 |       NULL |      NULL |
| 569020898   | Tecnico       | Barth     | Knevit      | Foden       | bfodena@hugedomains.com   |      3506547 |       NULL |      NULL |
| 572078928   | Auxiliar      | Dani      | Armatage    | Braycotton  | dbraycotton3@sohu.com     |     22460131 |       NULL |      NULL |
| 631751698   | Administrativo | Eadmund   | Prover      | Bertolaccini | ebertolaccini0@upenn.edu  |     17835453 |       NULL |      NULL |
| 751253619   | Administrativo | Germaine  | Haig        | Scurrey     | gscurrey2@a8.net          |     18448679 |       NULL |      NULL |
| 788769809   | Administrativo | Elisabeth | Banger      | Daunay      | edaunay1@icq.com          |     29860293 |          3 |      NULL |
| 792614188   | Limpiador     | Maximilian | Adnet       | Mathey      | mmatheyc@about.me         |     32342406 |       NULL |      NULL |
| 802845095   | Mantenimiento | Rhodie    | Prestie     | Le-Good     | rlegood7@npr.org          |     91875996 |       NULL |      NULL |
| 852540320   | Limpiador     | Neilla    | Croot       | Riach       | nriachd@timesonline.co.uk |     25639512 |       NULL |      NULL |
| 864069323   | Auxiliar      | Kamila    | Crickmore   | Garrigan    | kgarrigan5@dedecms.com    |     57284730 |          2 |      NULL |
| 881459082   | Tecnico       | Patton    | Tidmas      | Semiraz     | psemirazb@i2i.jp          |     80219784 |       NULL |      NULL |
| 896330468   | Limpiador     | Inness    | Kilkenny    | Pockey      | ipockeye@sourceforge.net  |     91538029 |       NULL |      NULL |
+-------------+---------------+-----------+-------------+-------------+---------------------------+--------------+------------+----------+
18 rows in set (0.00 sec)
```

Cambio 2: se cambia la tabla personal

Cambios en la tabla local

```
mysql> select * from tlocal;
+--------------------+------------------+-----------+-------------------------+---------+-----------+------------+
| CIF                | Direccion        | Telefono  | web                     | horario | cod_local | id_usuario |
+--------------------+------------------+-----------+-------------------------+---------+-----------+------------+
| 35713575-Cibercafe | Calle Jamaica 16 | 928142611 | HTTPS://nodomainyet.com | 10 a 22 |         3 |          3 |
| 35713575-Taller    | Calle Jamaica 16 | 928142611 | HTTPS://nodomainyet.com | 10 a 22 |         1 |          1 |
| 35713575-Tienda    | Calle Jamaica 16 | 928142611 | HTTPS://nodomainyet.com | 10 a 22 |         2 |          2 |
+--------------------+------------------+-----------+-------------------------+---------+-----------+------------+
3 rows in set (0.00 sec)

mysql> update tlocal set CIF = "35713575-Sala de juegos" where (cod_local=1);
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select * from tlocal;
+-------------------------+------------------+-----------+-------------------------+---------+-----------+------------+
| CIF                     | Direccion        | Telefono  | web                     | horario | cod_local | id_usuario |
+-------------------------+------------------+-----------+-------------------------+---------+-----------+------------+
| 35713575-Cibercafe      | Calle Jamaica 16 | 928142611 | HTTPS://nodomainyet.com | 10 a 22 |         3 |          3 |
| 35713575-Sala de juegos | Calle Jamaica 16 | 928142611 | HTTPS://nodomainyet.com | 10 a 22 |         1 |          1 |
| 35713575-Tienda         | Calle Jamaica 16 | 928142611 | HTTPS://nodomainyet.com | 10 a 22 |         2 |          2 |
+-------------------------+------------------+-----------+-------------------------+---------+-----------+------------+
3 rows in set (0.00 sec)
```

Cambio 3: se cambia la tabla local

- 7.4- Realizar tres eliminaciones de registros en tres tablas diferentes.

Realizamos 3 eliminaciones de las tablas consolasalasjuegos, incidencias y saladescanso, que pondremos al igual que en el ejercicio anterior el antes y el después.

Eliminación de datos en la tabla consolasalasjuegos

```
mysql> select * from tconsolasalasjuegos;
+--------+---------------+------------+
| ID_CSJ | ID_salajuegos | ID_consola |
+--------+---------------+------------+
|      1 |             2 |          3 |
|      2 |             1 |          2 |
|      3 |             3 |          2 |
+--------+---------------+------------+
3 rows in set (0.00 sec)

mysql> DELETE FROM tconsolasalasjuegos WHERE id_CSJ = 3;
Query OK, 1 row affected (0.00 sec)

mysql> select * from tconsolasalasjuegos;
+--------+---------------+------------+
| ID_CSJ | ID_salajuegos | ID_consola |
+--------+---------------+------------+
|      1 |             2 |          3 |
|      2 |             1 |          2 |
+--------+---------------+------------+
2 rows in set (0.00 sec)
```

Eliminación 1: eliminamos de consolasalasjuegos

Eliminación de datos en la tabla incidencias

```
mysql> select * from tincidencias;
+----------+-----------+----------------------+--------+--------+
| ID_PARTE | cod_local | observaciones        | ID_man | ID_adm |
+----------+-----------+----------------------+--------+--------+
|        1 |         3 | cable pelado al aire |      2 |      3 |
|        2 |         1 | enchufe roto         |      3 |      1 |
|        3 |         2 | aire roto            |      1 |      2 |
+----------+-----------+----------------------+--------+--------+
3 rows in set (0.00 sec)

mysql> DELETE FROM tincidencias WHERE ID_PARTE = 3;
Query OK, 1 row affected (0.00 sec)

mysql> select * from tincidencias;
+----------+-----------+----------------------+--------+--------+
| ID_PARTE | cod_local | observaciones        | ID_man | ID_adm |
+----------+-----------+----------------------+--------+--------+
|        1 |         3 | cable pelado al aire |      2 |      3 |
|        2 |         1 | enchufe roto         |      3 |      1 |
+----------+-----------+----------------------+--------+--------+
2 rows in set (0.00 sec)
```

Eliminación 2: eliminamos datos de incidencias

Eliminación de datos de la tabla saladescanso

```
mysql> select * from tsaladescanso;
+-----------------+-----------+---------+
| ID_SALADESCANSO | cod_local | horario |
+-----------------+-----------+---------+
|               1 |         3 | 10 a 21 |
|               2 |         2 | 10 a 21 |
|               3 |         1 | 10 a 21 |
+-----------------+-----------+---------+
3 rows in set (0.00 sec)

mysql> DELETE FROM tsaladescanso WHERE cod_local =1;
Query OK, 1 row affected (0.00 sec)

mysql> select * from tsaladescanso;
+-----------------+-----------+---------+
| ID_SALADESCANSO | cod_local | horario |
+-----------------+-----------+---------+
|               1 |         3 | 10 a 21 |
|               2 |         2 | 10 a 21 |
+-----------------+-----------+---------+
2 rows in set (0.00 sec)
```

Eliminación 3: eliminamos datos de saladescanso

- 7.5- Realizar tres transacciones con mínimo dos o más tablas implicadas.

Y aquí ponemos las transacciones

Transacción 1

Añadí el campo ganador a torneos y modifiqué al usuario que había ganado

```
mysql> SET AUTOCOMMIT=0;
Query OK, 0 rows affected (0.00 sec)

mysql> START TRANSACTION;
Query OK, 0 rows affected (0.00 sec)

mysql> LOCK TABLE ttorneos WRITE;
Query OK, 0 rows affected (0.00 sec)

mysql> ALTER TABLE ttorneos ADD ganador VARCHAR(255);
Query OK, 0 rows affected (0.01 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> UPDATE ttorneos SET ganador = "usuario 1" WHERE ID_torneo=1;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> UNLOCK TABLES;
Query OK, 0 rows affected (0.00 sec)

mysql> LOCK TABLE tcliente WRITE;
Query OK, 0 rows affected (0.00 sec)

mysql> UPDATE tcliente SET telefono_cl =687598736 WHERE id_usuario=1;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> UNLOCK TABLES;
Query OK, 0 rows affected (0.00 sec)

mysql> COMMIT;
Query OK, 0 rows affected (0.00 sec)
```

Transacción 1: modificamos torneos y usuario

Transacción 2

Aquí añadí a los equipos los portátiles y modifiqué la tabla ordenadores a posterior

```
mysql> SET AUTOCOMMIT =0;
Query OK, 0 rows affected (0.00 sec)

mysql> START TRANSACTION;
Query OK, 0 rows affected (0.00 sec)

mysql> LOCK TABLE tequipos WRITE;
Query OK, 0 rows affected (0.00 sec)

mysql> INSERT INTO tequipos(Num_serie,tipo_equipo,unidades,cod_local)
    -> VALUES
    -> ("PT01","Portátil",5,3);
Query OK, 1 row affected (0.00 sec)

mysql> UNLOCK TABLES;
Query OK, 0 rows affected (0.00 sec)

mysql> LOCK TABLE tordenadores WRITE;
Query OK, 0 rows affected (0.00 sec)

mysql> UPDATE tordenadores SET id_equipo=4 WHERE id_ordenador=3;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> UPDATE tordenadores SET id_equipo=4 WHERE id_ordenador=2;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> UNLOCK TABLES;
Query OK, 0 rows affected (0.00 sec)

mysql> COMMIT;
Query OK, 0 rows affected (0.00 sec)

mysql>
```

Transacción 2: añadimos portátiles y modificamos ordenadores

Transacción 3

Esta está dividida en 3 capturas porque no me convencía que se viera el cargo en la tabla de personal, ya que si estoy viendo la tabla de administrativo no es necesario que vea que su cargo es administrativo, así que lo cambie por el nombre, modificando el nombre de la columna y los datos introducidos, de las tablas que componen personal (gerente, administrativo, auxiliar, técnico, limpiador, mantenimiento)

```
mysql> SET AUTOCOMMIT=0;
Query OK, 0 rows affected (0.00 sec)

mysql> START TRANSACTION;
Query OK, 0 rows affected (0.00 sec)

mysql> LOCK TABLE tgerente WRITE;
Query OK, 0 rows affected (0.00 sec)

mysql> ALTER TABLE tgerente CHANGE cargo Nombre_ger VARCHAR(255);
Query OK, 3 rows affected (0.03 sec)
Records: 3  Duplicates: 0  Warnings: 0

mysql> UPDATE tgerente SET Nombre_ger="Cándido" WHERE id_ger =1;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> UPDATE tgerente SET Nombre_ger="Lucia" WHERE id_ger =2;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> UPDATE tgerente SET Nombre_ger="Iris" WHERE id_ger =3;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> UNLOCK TABLES;
Query OK, 0 rows affected (0.00 sec)

mysql> LOCK TABLE tadministrativo WRITE ;
Query OK, 0 rows affected (0.00 sec)

mysql> ALTER TABLE tadministrativo CHANGE cargo Nombre_adm VARCHAR(255);
Query OK, 3 rows affected (0.04 sec)
Records: 3  Duplicates: 0  Warnings: 0

mysql> UPDATE tadministrativo SET Nombre_adm="Eadmund" WHERE id_adm =1;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> UPDATE tadministrativo SET Nombre_adm="Elisabeth" WHERE id_adm =2;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> UPDATE tadministrativo SET Nombre_adm="Germaine" WHERE id_adm =3;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> UNLOCK TABLES;
Query OK, 0 rows affected (0.00 sec)
```

Transacción 3: Modificamos varias tablas de una sola vez

```
mysql> LOCK TABLE tauxiliar WRITE ;
Query OK, 0 rows affected (0.00 sec)

mysql> ALTER TABLE tauxiliar CHANGE cargo Nombre_aux VARCHAR(255);
Query OK, 3 rows affected (0.03 sec)
Records: 3  Duplicates: 0  Warnings: 0

mysql> UPDATE tauxiliar SET Nombre_aux="Dani" WHERE id_aux =1;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> UPDATE tauxiliar SET Nombre_aux="Clarissa" WHERE id_aux =2;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> UPDATE tauxiliar SET Nombre_aux="Kamila" WHERE id_aux =3;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> UNLOCK TABLES;
Query OK, 0 rows affected (0.00 sec)

mysql> LOCK TABLE tmantenimiento WRITE ;
Query OK, 0 rows affected (0.00 sec)

mysql> ALTER TABLE tmantenimiento CHANGE cargo Nombre_man VARCHAR(255);
Query OK, 3 rows affected (0.04 sec)
Records: 3  Duplicates: 0  Warnings: 0

mysql> UPDATE tmantenimiento SET Nombre_man="Alisha" WHERE id_man =1;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> UPDATE tmantenimiento SET Nombre_man="Rhodie" WHERE id_man =2;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> UPDATE tmantenimiento SET Nombre_man="Charin" WHERE id_man =3;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> UNLOCK TABLES;
Query OK, 0 rows affected (0.00 sec)
```

Transacción 4: continuación de transacción

```
mysql> LOCK TABLE tlimpiador WRITE ;
Query OK, 0 rows affected (0.00 sec)

mysql> ALTER TABLE tlimpiador CHANGE cargo Nombre_lim VARCHAR(255);
Query OK, 3 rows affected (0.03 sec)
Records: 3  Duplicates: 0  Warnings: 0

mysql> UPDATE tlimpiador SET Nombre_lim="Alisha" WHERE id_lim =1;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> UPDATE tlimpiador SET Nombre_lim="Rhodie" WHERE id_lim =2;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> UPDATE tlimpiador SET Nombre_lim="Charin" WHERE id_lim =3;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> UNLOCK TABLES;
Query OK, 0 rows affected (0.00 sec)

mysql> LOCK TABLE ttecnico WRITE ;
Query OK, 0 rows affected (0.00 sec)

mysql> ALTER TABLE ttecnico CHANGE cargo Nombre_tec VARCHAR(255);
Query OK, 3 rows affected (0.03 sec)
Records: 3  Duplicates: 0  Warnings: 0

mysql> UPDATE ttecnico SET Nombre_tec="Donnamarie" WHERE id_tec =1;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> UPDATE ttecnico SET Nombre_tec="Barth" WHERE id_tec =2;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> UPDATE ttecnico SET Nombre_tec="Patton" WHERE id_tec =3;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> UNLOCK TABLES;
Query OK, 0 rows affected (0.00 sec)

mysql> COMMIT;
Query OK, 0 rows affected (0.00 sec)
```

Transacción 5: continuación de transacción

8- Cambios realizados de Fase 4 a Fase 5 y 6

Los cambios realizados son los que se realizan durante la ejecución de los ejercicios, ya que se han aprovechado los ejercicios para cambiar la base de datos a una versión más legible por mi parte.

9- Fase 7 del proyecto, Consulta de Datos

9.1- Una consulta sobre una tabla, mostrando todos los datos de dicha tabla.

Mostramos la tabla de persona.

```
mysql> select * from tpersonal;
+-------------+---------------+------------+--------------+--------------+---------------------------+--------------+------------+-----------+
| DNI_personal | cargo         | nombre_per | apellido1_per | apellido2_per | email_per                 | telefono_per | id_usuario | cod_local |
+-------------+---------------+------------+--------------+--------------+---------------------------+--------------+------------+-----------+
| 12345678S   | Gerente       | Iris       | P├@rez       | Morales      | Iris2023@gmail.com        |     32165498 |       NULL |      NULL |
| 182410414   | Auxiliar      | Clarissa   | Miguet       | Larkings     | clarkings4@dot.gov        |     12856342 |       NULL |      NULL |
| 343715166   | Tecnico       | Donnamarie | Djekovic     | Deerness     | ddeerness9@shinystat.com  |     57143061 |       NULL |      NULL |
| 396157927   | Mantenimiento | Alisha     | Tomanek      | Stabbins     | astabbins6@reddit.com     |     48724671 |       NULL |      NULL |
| 42227069N   | Gerente       | C├índido   | P├@rez       | Verwer       | candidonpv@gmail.com      |    656342086 |       NULL |      NULL |
| 45749494H   | Gerente       | Lucia      | Morales      | P├@rez       | luciambito@gmail.com      |     65242052 |       NULL |      NULL |
| 510284963   | Mantenimiento | Charin     | Kment        | Streeting    | cstreeting8@vistaprint.com|     14473110 |       NULL |      NULL |
| 569020898   | Tecnico       | Barth      | Knevit       | Foden        | bfodena@hugedomains.com   |      3506547 |       NULL |      NULL |
| 572078928   | Auxiliar      | Dani       | Armatage     | Braycotton   | dbraycotton3@sohu.com     |     22460131 |       NULL |      NULL |
| 631751698   | Administrativo| Eadmund    | Prover       | Bertolaccini | ebertolaccini0@upenn.edu  |     17835453 |       NULL |      NULL |
| 751253619   | Administrativo| Germaine   | Haig         | Scurrey      | gscurrey2@a8.net          |     18448679 |       NULL |      NULL |
| 788769809   | Administrativo| Elisabeth  | Banger       | Daunay       | edaunay1@icq.com          |     29860293 |          3 |      NULL |
| 792614188   | Limpiador     | Maximilian | Adnet        | Mathey       | mmatheyc@about.me         |     32342406 |       NULL |      NULL |
| 802845095   | Mantenimiento | Rhodie     | Prestie      | Le-Good      | rlegood7@npr.org          |     91875996 |       NULL |      NULL |
| 852540320   | Limpiador     | Neilla     | Croot        | Riach        | nriachd@timesonline.co.uk |     25639512 |       NULL |      NULL |
| 864069323   | Auxiliar      | Kamila     | Crickmore    | Garrigan     | kgarrigan5@dedecms.com    |     57284730 |          2 |      NULL |
| 881459082   | Tecnico       | Patton     | Tidmas       | Semiraz      | psemirazb@i2i.jp          |     80219784 |       NULL |      NULL |
| 896330468   | Limpiador     | Inness     | Kilkenny     | Pockey       | ipockeye@sourceforge.net  |     91538029 |       NULL |      NULL |
+-------------+---------------+------------+--------------+--------------+---------------------------+--------------+------------+-----------+
18 rows in set (0.00 sec)
```

Consulta 1: Mostramos la tabla personal

9.2- Dos consultas en las que se realizará exclusivamente sobre dos tablas y tendrá el uso de dos funciones diferentes (CUALQUIER FUNCIÓN EXPLICADA EN CLASE, SUM, MIN, MAX, AVG etc.).

1ª consulta: Sacamos las unidades de equipos que hay en el cibercafé.

```
mysql> SELECT SUM(unidades) AS unidades
    -> FROM tequipos
    -> JOIN tcibercafe ON tcibercafe.id_equipo=tequipos.id_equipo;
+-----------+
| unidades |
+-----------+
|       15 |
+-----------+
1 row in set (0.00 sec)
```

Consulta 2: Sacamos las unidades de equipos que hay en el cibercafé

2ª consulta: Sacamos las unidades totales y las medias de cada consola que están en la sala de juegos.

```
mysql> SELECT SUM(unidades) AS unidades
    -> FROM tequipos
    -> LEFT JOIN tconsolas ON tequipos.id_equipo=tconsolas.id_equipo
    -> LEFT JOIN tsalajuegos ON tconsolas.id_consolas=tsalajuegos.id_consolas;
+----------+
| unidades |
+----------+
|       30 |
+----------+
1 row in set (0.00 sec)

mysql> SELECT AVG(unidades) AS unidades
    -> FROM tequipos
    -> LEFT JOIN tconsolas ON tequipos.id_equipo=tconsolas.id_equipo
    -> LEFT JOIN tsalajuegos ON tconsolas.id_consolas=tsalajuegos.id_consolas;
+----------+
| unidades |
+----------+
|   5.0000 |
+----------+
1 row in set (0.00 sec)
```

Consulta 3: Sacamos las unidades totales y las medias de cada consola que están en la sala de juegos.

9.3- Dos consultas que tendrán que ser implementadas mediante subconsultas (anidamiento). La consulta y subconsulta debe realizarse sobre dos tablas diferentes. También, se debe utilizar la cláusula BETWEEN en una de ellas.

1ª consulta: verificamos que empleado creo el torneo numero 1

```
mysql> SELECT nombre_per
    -> FROM tpersonal
    -> WHERE dni_personal IN (
    -> SELECT dni_personal FROM tpersonaltorneo
    -> WHERE id_torneo =1);
+------------+
| nombre_per |
+------------+
| Iris       |
+------------+
1 row in set (0.00 sec)
```

Consulta 4: verificamos el empleado que creo el torneo con ID 1

2ª consulta: que nos muestre el id de torneo que se ha celebrado entre unas fechas y tiene un ganador

```
mysql> SELECT id_torneo
    -> FROM tclientetorneo
    -> WHERE id_torneo in
    -> (SELECT fecha_torneo FROM ttorneos WHERE fecha_torneo BETWEEN "02-04-2023" AND "23-08-2023");
+-----------+
| id_torneo |
+-----------+
|         3 |
+-----------+
1 row in set, 1 warning (0.00 sec)
```

Consulta 5: que nos muestre el id de torneo que se ha celebrado entre unas fechas y tiene un ganador

9.4- Dos consultas que tendrán que ser implementadas mediante JOIN. (INNER JOIN, LEFT JOIN, RIGHT JOIN u OUTER JOIN).

1ª consulta: Consultamos los clientes que han sido atendidos por algún miembro del personal.

```
mysql> SELECT nombre_cl,apellido1_cl,cargo,nombre_per,apellido1_per
    -> FROM tcliente
    -> LEFT JOIN tpersonal
    -> ON tcliente.id_usuario=tpersonal.id_usuario
    -> WHERE tpersonal.id_usuario IS NOT NULL;
+------------+--------------+----------------+------------+---------------+
| nombre_cl  | apellido1_cl | cargo          | nombre_per | apellido1_per |
+------------+--------------+----------------+------------+---------------+
| Rosa Maria | Moreno       | Administrativo | Elisabeth  | Banger        |
| Almudena   | Hernandez    | Auxiliar       | Kamila     | Crickmore     |
+------------+--------------+----------------+------------+---------------+
2 rows in set (0.00 sec)
```

Consulta 6: Consultamos los clientes que han sido atendidos por algún miembro del personal.

2ª consulta: Consultamos el ganador del ganador que hay en los torneos realizados.

```
mysql> SELECT ttorneos.id_torneo,ganador,fecha_torneo,dni_personal
    -> FROM ttorneos LEFT JOIN tpersonaltorneo
    -> ON tpersonaltorneo.id_torneo=ttorneos.id_torneo
    -> WHERE ganador IS NOT NULL;
+-----------+-----------+--------------+--------------+
| id_torneo | ganador   | fecha_torneo | dni_personal |
+-----------+-----------+--------------+--------------+
|         1 | usuario 1 | 03-05-2023   | 12345678S    |
+-----------+-----------+--------------+--------------+
1 row in set (0.00 sec)
```

Consulta 7: Consultamos el ganador del ganador que hay en los torneos realizados.

9.5- Dos consultas que tendrán que ser implementadas mediante GROUP BY con WHERE o HAVING o ambas a la vez.

1ª consulta: sacamos al personal que ha tenido contacto con el cliente número 2.

```
mysql> SELECT tcliente.nombre_cl, tpersonal.nombre_per, tpersonal.id_usuario
    -> from tpersonal
    -> left join tcliente ON tcliente.id_usuario=tpersonal.id_usuario
    -> where tpersonal.id_usuario = 2
    -> GROUP BY tpersonal.DNI_personal ;
+-----------+------------+------------+
| nombre_cl | nombre_per | id_usuario |
+-----------+------------+------------+
| Almudena  | Kamila     |          2 |
+-----------+------------+------------+
1 row in set (0.00 sec)
```

Consulta 8: sacamos al personal que ha tenido contacto con el cliente número 2

2ª consulta: verificamos que técnico este asignado a la incidencia en el código de local 2 y que administrativo creo la incidencia.

```
mysql> SELECT tadministrativo.nombre_adm, tincidencias.observaciones, tmantenimiento.nombre_man
    -> from tadministrativo
    -> INNER JOIN tincidencias on tincidencias.id_adm=tadministrativo.id_adm
    -> LEFT JOIN tmantenimiento on tmantenimiento.id_man=tincidencias.id_man
    -> WHERE tincidencias.cod_local=3;
+------------+---------------------+------------+
| nombre_adm | observaciones       | nombre_man |
+------------+---------------------+------------+
| Germaine   | cable pelado al aire | Rhodie     |
+------------+---------------------+------------+
1 row in set (0.00 sec)
```

Consulta 9: verificamos que técnico este asignado a la incidencia en el código de local 2 y que administrativo creo la incidencia

10- Cambios realizados de Fase 5 y 6 a la fase 7

Para poder cumplir con todas las consultas tuve que realizar una modificación de datos de la tabla equipos.

update tequipos set unidades = 15 where (id_equipo=1);

update tequipos set unidades = 7 where (id_equipo=2);

update tequipos set unidades = 5 where (id_equipo=3);

update tequipos set unidades = 20 where (id_equipo=4);

aunque al final no los use para la consulta que iba a hacer y cree otra distinta.

11- Fase 8 Aplicación BD en PHP

Para esta parte, use una versión de mi base de datos en miniatura, para así centrarme en lo que iba a hacer y no distraerme, por lo que están las siguientes tablas:

```
mysql> use trabajobae;
Database changed
mysql> show tables;
+----------------------+
| Tables_in_trabajobae |
+----------------------+
| tadministrativo      |
| tauxiliar            |
| tcliente             |
| tgerente             |
| tlimpiador           |
| tmantenimiento       |
| tpersonal            |
| ttecnico             |
+----------------------+
8 rows in set (0.00 sec)
```

Fase 8: 1 Tablas a utilizar

Todas ellas están conectadas por dni personal a tpersonal.

La idea principal era mezclar estas dos partes en una

• Las acciones (CRUD) Listado, Inserción, Modificación y Borrado de 2

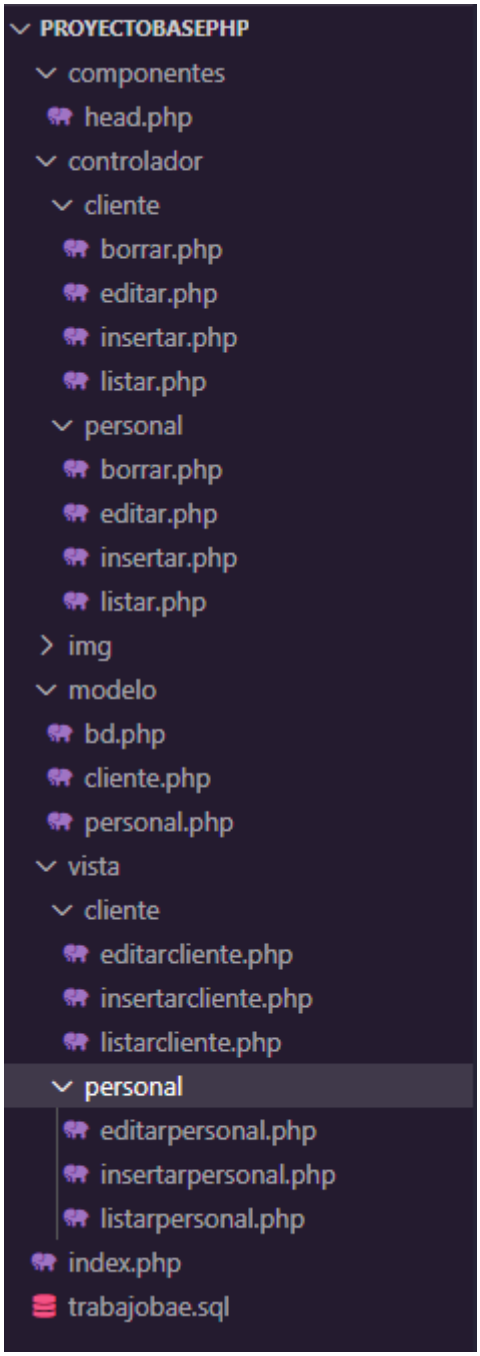
tablas de la base de datos, estas tablas deben estar relacionadas.

• Se deben realizar además dos transacciones, aparte de las acciones

CRUD.

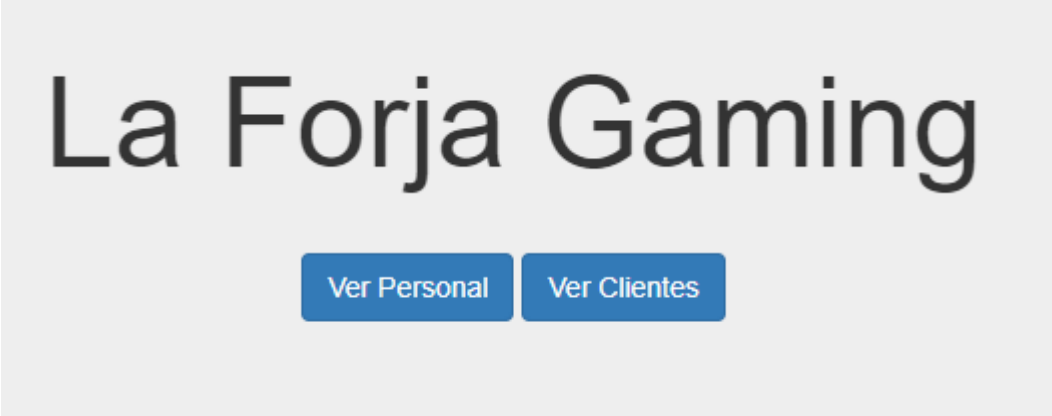
, pero como se tienen que realizar las transacciones aparte de las acciones CRUD, añadí la tabla clientes para hacer las CRUD directamente y dejar las transacciones para las tablas del personal.

Lo primero fue crear de forma correcta las carpetas del proyecto para tener separadas las vistas, modelos y controladores en función de si eran para el personal o para el cliente



Fase 8: 2 organización ficheros

En la parte de índex se ha hecho de forma sencilla, que muestre el nombre de la empresa y dos botones, que nos redirige a la parte de personal o clientes



Fase 8: 3 Web índex

```
<!DOCTYPE html>
<html lang="es">
<head>
    <?php
include "componentes/head.php";
?>

</head>
<body>
    <div class="container-fluid">
        <div class="jumbotron">
            <h1 align="center">La Forja Gaming</h1>
            <br>
            <div align="center">
            <a href="vista/personal/listarpersonal.php"><button class="btn btn-primary">Ver Personal</button></a>
            <a href="vista/cliente/listarcliente.php"><button class="btn btn-primary">Ver Clientes</button></a>
            </div>
        </div>
    </div>

</body>
</html>
```

Fase 8: 4 código índex

Lo primero que necesitamos para poder realizar las acciones es conectarnos a la base de datos, por lo que se creó el modelo bd.php para poder conectarnos

```php
<?php

class bd
{

    private $cadenaConexion = "mysql:dbname=trabajobae;host=localhost;port=3306";

    private $usuario = "root";

    private $clave = "1234";


    function conectarBD()
    {
        try {
            return new PDO($this->cadenaConexion, $this->usuario, $this->clave);
        } catch (PDOException $e) {
            echo "Error ocurrio un problema en Base de datos: " . $e->getMessage();
            return null;
        }
    }
}
```

Fase 8: 5 PDO para conectar a base de datos

En mi caso en vez de subirlo a phpmyadmin use Mysql con el Workbench y el command line para poder ir haciendo las verificaciones y comprobarlo de igual manera por si la visualización diera fallos

Empezamos con la parte de personal, ya que no se puede insertar un cliente sin tener un personal creado.

Le creamos un modelo con los atributos de las tablas de personal, con sus respectivos getter y setter y con la conexión a la base de datos

```php
<?php

require_once "bd.php";

14 references | 0 implementations
class personal{

    35 references
    private $db;
    3 references
    private $DNI_personal;
    2 references
    private $cargo;
    2 references
    private $nombre_per;
    2 references
    private $apellido1_per;
    2 references
    private $apellido2_per;
    2 references
    private $email_per;
    2 references
    private $telefono_per;
    2 references
    private $id_usuario;
    2 references
    private $cod_local;


    5 references | 0 overrides
    function __construct(){
        $bd=new bd();
        $this->db=$bd->conectarBD();
    }
```

Fase 8: 6 modelo personal con atributos

También creamos un controlador con una función para listar los empleados y poder devolverlos

```php
<?php
require_once "../../modelo/personal.php";
$personalModelo = new personal();
$personal = $personalModelo->obtenerPersonal();

foreach ($personal as $p) {
    echo "<tr>
        <th>".$p->getDni_personal()."</th>
        <th>".$p->getNombre_per()."</th>
        <th>".$p->getApellido1_per()."</th>
        <th>".$p->getApellido2_per()."</th>
        <th>".$p->getEmail_per()."</th>
        <th>".$p->getTelefono_per()."</th>
        <th>".$p->getCargo()."</th>
        <th>
            <a href=editarpersonal.php?DNI_personal=".$p->getDni_personal()."><button>Editar</button></a>
            <a href=../../controlador/personal/borrar.php?DNI_personal=".$p->getDni_personal()."><button>Borrar</button></a>
        </th>
    </tr>
    ";
}

?>
```

Fase 8: 7 controlador listar personal

```php
function obtenerPersonal(){
    try {
        $querySelect="SELECT * FROM tpersonal ORDER BY cargo";
        $listapersonal=$this->db->prepare($querySelect);
        $listapersonal->execute();
    } catch (Exception $ex) {
        echo "Te equivocaste en: ".$ex->getMessage();
    }
    return $listapersonal->fetchALL(PDO::FETCH_CLASS,"personal");
}
```

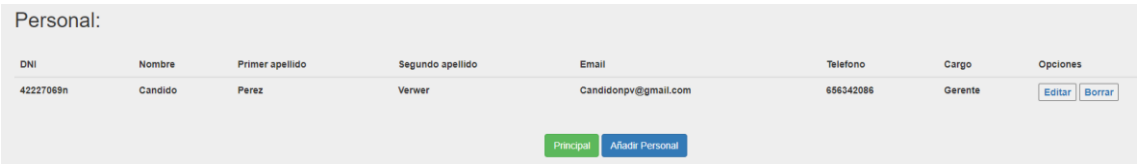
Fase 8: 8 Función listar personal

Y finalmente tenemos la vista para poder mostrarlo

```php
<!DOCTYPE html>
<html>

<?php
include "../../componentes/head.php";
?>

<body>
    <div class="container-fluid">
        <div class="jumbotron">
            <h2>Personal: </h2>
            <br>

            <table class='table'>
                <thead class='table-dark'>
                    <tr>
                        <th>DNI</th>
                        <th>Nombre</th>
                        <th>Primer apellido</th>
                        <th>Segundo apellido</th>
                        <th>Email</th>
                        <th>Telefono</th>
                        <th>Cargo</th>
                        <th>Opciones</th>
                    </tr>
                </thead>/.table-dark
                <tbody>
                <?php
                include "../../controlador/personal/listar.php";
                ?>
                </tbody>
            </table>/.table

            <br>
            <div align="center">
            <a href="../../index.php"><button class="btn btn-success">Principal </button></a>
            <a href="insertarPersonal.php"><button class="btn btn-primary">Añadir Personal</button></a>
            </div>
        </div>/.jumbotron
    </div>/.container-fluid
</body>

</html>
```

Fase 8: 9 vista listar personal

Que quedaría de la siguiente manera (acabo de insertar los datos para no mostrarlo vacío



Fase 8: 10 Web listar personal

En la parte de insertar los datos, tenemos el siguiente formulario

```
body>
    <div class="container-fluid">
        <div class="jumbotron">
            '/

            <form id="insertarPersonal" target="listarpersonal.php" method="post">
                <div class="form-group">
                    <label>DNI</label>
                    <input type="text" class="form-control" name="DNI_personal" required>
                </div>/.form-group
                <div class="form-group">
                    <label>Nombre</label>
                    <input type="text" class="form-control" name="nombre_per" required>
                </div>/.form-group

                <div class="form-group">
                    <label>Apellido 1</label>
                    <input type="text" class="form-control" name="apellido1_per" required>
                </div>/.form-group
                <div class="form-group">
                    <label>Apellido 2</label>
                    <input type="text" class="form-control" name="apellido2_per" required>
                </div>/.form-group
                <div class="form-group">
                    <label>Email</label>
                    <input type="text" class="form-control" name="email_per" required>
                </div>/.form-group
                <div class="form-group">
                    <label>Telefono</label>
                    <input type="text" class="form-control" name="telefono_per" required>
                </div>/.form-group
                <div>
                    <label>Cargo</label>
                    <br>
                    <select id="Cargo" name="cargo">
                        <option value="Administrativo">Administrativo</option>
                        <option value="Tecnico">Tecnico</option>
                        <option value="Auxiliar">Auxiliar</option>
                        <option value="Gerente">Gerente</option>
                        <option value="Mantenimiento">Mantenimiento</option>
                        <option value="Limpiador">Limpiador</option>
                    </select>/#Cargo
                </div>
                <br>

                <button type="submit" target="listarpersonal.php" class="btn btn-primary">Añadir Personal</button>
```

Fase 8: 11 vista insertar personal

Con este controlador

```php
<?php
require_once "../../modelo/personal.php";
if (
    isset($_POST['DNI_personal'])
    && isset($_POST['nombre_per'])
    && isset($_POST['apellido1_per'])
    && isset($_POST['apellido2_per'])
    && isset($_POST['email_per'])
    && isset($_POST['telefono_per'])
    && isset($_POST['cargo'])
) {
    $DNI_personal =$_POST['DNI_personal'];
    $nombre_per = $_POST['nombre_per'];
    $apellido1_per = $_POST['apellido1_per'];
    $apellido2_per = $_POST['apellido2_per'];
    $email_per = $_POST['email_per'];
    $telefono_per = $_POST['telefono_per'];
    $cargo =$_POST['cargo'];

    $persona = new personal();
    $persona->setDni_personal($DNI_personal);
    $persona->setNombre_per($nombre_per);
    $persona->setApellido1_per($apellido1_per);
    $persona->setApellido2_per($apellido2_per);
    $persona->setEmail_per($email_per);
    $persona->setTelefono_per($telefono_per);
    $persona->setCargo($cargo);
    echo $persona->addPersona(
        $_POST['DNI_personal'],
        $_POST['nombre_per'],
        $_POST['apellido1_per'],
        $_POST['apellido2_per'],
        $_POST['email_per'],
        $_POST['telefono_per'],
        $_POST['cargo']
    );
}

?>
```

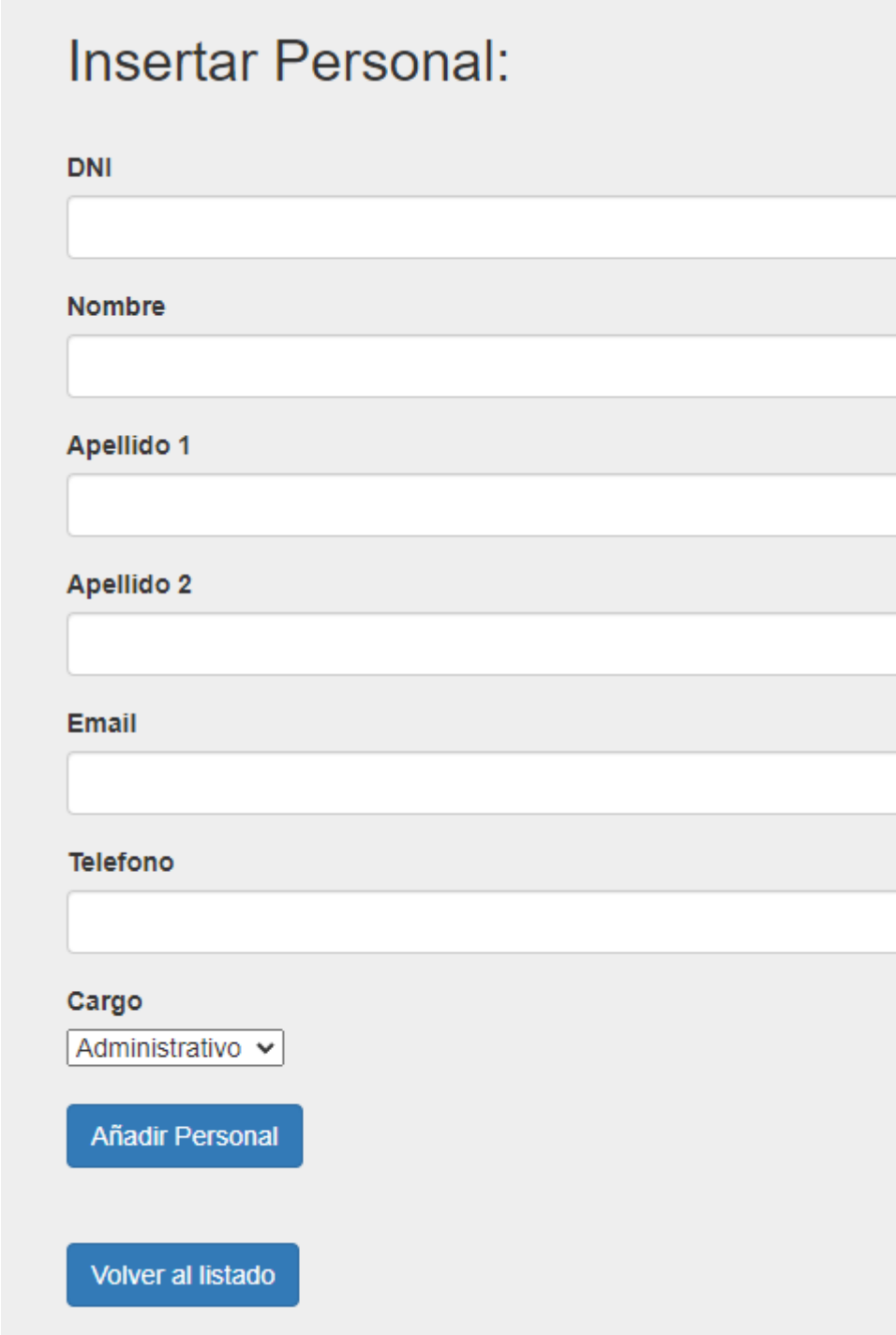
Fase 8: 12 controlador insertar personal

Y esta función (muestro una parte porque es demasiado grande para sacar una captura completa)

```php
function addPersona($DNI_personal, $nombre_per, $apellido1_per, $apellido2_per, $email_per, $telefono_per, $cargo) {
        $this->db->beginTransaction();
        try {
            $insertarPersona = "INSERT INTO Tpersonal (DNI_personal, nombre_per, apellido1_per, apellido2_per, email_per, telefono_per, cargo)
                VALUES (:DNI_personal, :nombre_per, :apellido1_per, :apellido2_per, :email_per, :telefono_per, :cargo)";
            $queryPersona = $this->db->prepare($insertarPersona);
            $queryPersona->bindParam(":DNI_personal", $DNI_personal);
            $queryPersona->bindParam(":nombre_per", $nombre_per);
            $queryPersona->bindParam(":apellido1_per", $apellido1_per);
            $queryPersona->bindParam(":apellido2_per", $apellido2_per);
            $queryPersona->bindParam(":email_per", $email_per);
            $queryPersona->bindParam(":telefono_per", $telefono_per);
            $queryPersona->bindParam(":cargo", $cargo);
            $queryPersona->execute();

            if ($cargo == 'Administrativo') {
                $insertarAdmin = "INSERT INTO Tadministrativo (Nombre_adm, DNI_personal)
                    VALUES (:nombre_per, :DNI_personal)";
                $queryAdmin = $this->db->prepare($insertarAdmin);
                $queryAdmin->bindParam(":nombre_per", $nombre_per);
                $queryAdmin->bindParam(":DNI_personal", $DNI_personal);
                $queryAdmin->execute();

            } else if ($cargo == 'Gerente') {
                $insertarGerente = "INSERT INTO Tgerente (Nombre_ger, DNI_personal)
                    VALUES (:nombre_per, :DNI_personal)";
                $queryGerente = $this->db->prepare($insertarGerente);
                $queryGerente->bindParam(":nombre_per", $nombre_per);
                $queryGerente->bindParam(":DNI_personal", $DNI_personal);
                $queryGerente->execute();

            } else if ($cargo == 'Auxiliar') {
                $insertarAuxiliar = "INSERT INTO Tauxiliar (DNI_personal, Nombre_aux)
                    VALUES (:DNI_personal, :nombre_per)";
```

Fase 8: 13 Función insertar personal

Y en la vista web se vería de la siguiente manera



Fase 8: 14 Web insertar personal

Para editarlo tenemos este controlador

```php
<?php
    require_once "../../modelo/personal.php";

    if(isset ($_GET['DNI_personal'])&& !empty($_GET['DNI_personal'])){
        $p = new personal();
        $p->setDni_personal($_GET['DNI_personal']);
        $p=$p->obtenerPersonaindividual();

    }
    if (
        isset($_POST['DNI_personal'])
        && isset($_POST['nombre_per'])
        && isset($_POST['apellido1_per'])
        && isset($_POST['apellido2_per'])
        && isset($_POST['email_per'])
        && isset($_POST['telefono_per'])
        && isset($_POST['cargo'])
    ) {
        $DNI_personal =$_POST['DNI_personal'];
        $nombre_per = $_POST['nombre_per'];
        $apellido1_per = $_POST['apellido1_per'];
        $apellido2_per = $_POST['apellido2_per'];
        $email_per = $_POST['email_per'];
        $telefono_per = $_POST['telefono_per'];
        $cargo =$_POST['cargo'];

        $p = new personal();
        $p->setDni_personal($DNI_personal);
        $p->setNombre_per($nombre_per);
        $p->setApellido1_per($apellido1_per);
        $p->setApellido2_per($apellido2_per);
        $p->setEmail_per($email_per);
        $p->setTelefono_per($telefono_per);
        $p->setCargo($cargo);
        echo $p->editarPersona($DNI_personal, $nombre_per, $apellido1_per, $apellido2_per, $email_per, $telefono_per, $cargo);
    }
```

Fase 8: 15 controlador editar personal

Esta vista

```php
<html>

<body>
    <div class="container-fluid">
        <div class="jumbotron">
            <h2>Editar Personal: </h2>
            <br>
            <!-- hay que terminar esta parte e ir probandola -->
            <?php
                include "../../controlador/personal/editar.php";
            ?>

            <form id="editarPersonal" method="post">
                <div class="form-group">
                    <label>DNI</label>
                    <input type="text" class="form-control" name="DNI_personal" value="<?php echo $p->getDni_personal() ?>" readonly>
                </div>/.form-group
                <div class="form-group">
                    <label>Nombre</label>
                    <input type="text" class="form-control" name="nombre_per" value="<?php echo $p->getNombre_per() ?>">
                </div>/.form-group

                <div class="form-group">
                    <label>Apellido 1</label>
                    <input type="text" class="form-control" name="apellido1_per" value="<?php echo $p->getApellido1_per() ?>">
                </div>/.form-group
                <div class="form-group">
                    <label>Apellido 2</label>
                    <input type="text" class="form-control" name="apellido2_per" value="<?php echo $p->getApellido2_per() ?>">
                </div>/.form-group
                <div class="form-group">
                    <label>Email</label>
                    <input type="text" class="form-control" name="email_per" value="<?php echo $p->getEmail_per() ?>">
                </div>/.form-group
                <div class="form-group">
                    <label>Telefono</label>
                    <input type="text" class="form-control" name="telefono_per" value="<?php echo $p->getTelefono_per() ?>">
                </div>/.form-group
                <div>
                    <label>Cargo</label>
                    <br>
                    <select id="Cargo" name="cargo" >
                        <option value="Administrativo">Administrativo</option>
                        <option value="Tecnico">Tecnico</option>
                        <option value="Auxiliar">Auxiliar</option>
                        <option value="Gerente">Gerente</option>
                        <option value="Mantenimiento">Mantenimiento</option>
                        <option value="Limpiador">Limpiador</option>
```

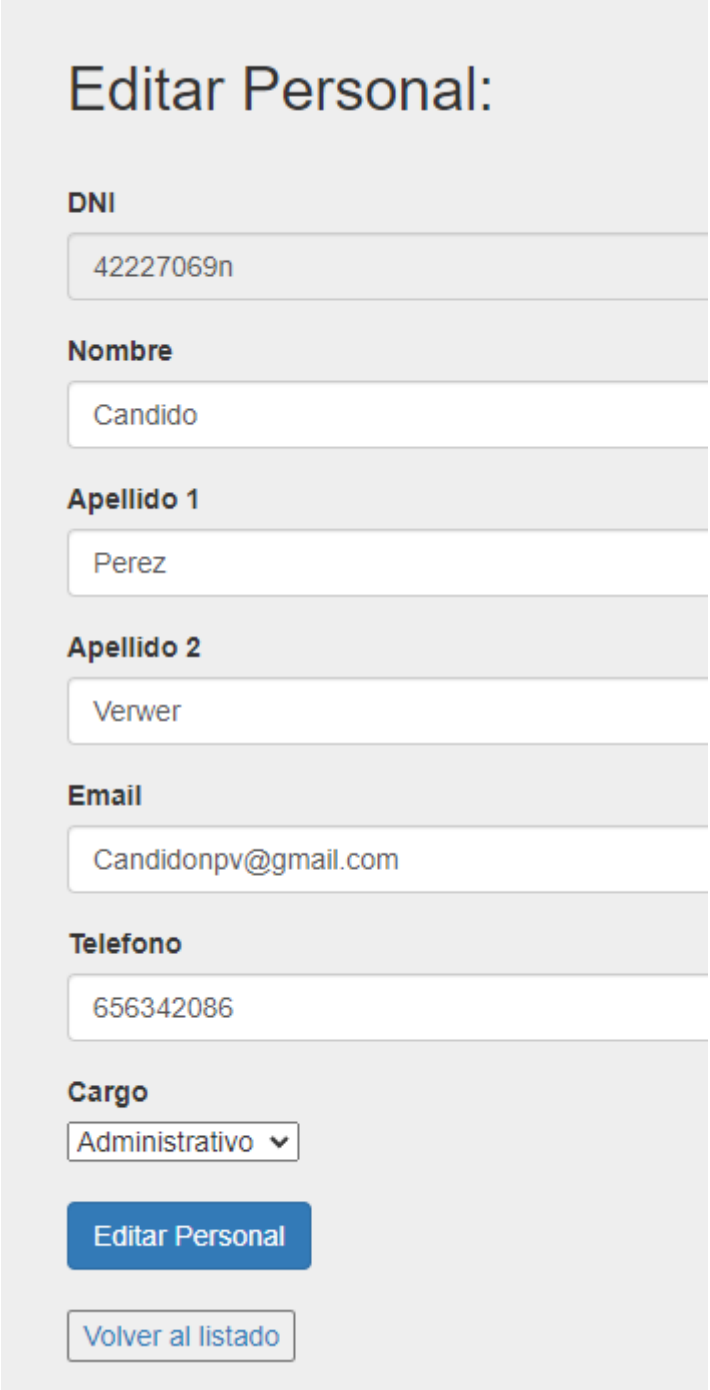
Fase 8: 16 vista editar personal

Y esta función

```php
function editarPersona($DNI_personal, $nombre_per, $apellido1_per, $apellido2_per, $email_per, $telefono_per,
    $this->db->beginTransaction();

    try {

        $eliminarAdmin = "DELETE FROM Tadministrativo WHERE DNI_personal = :DNI_personal";
        $queryAdmin = $this->db->prepare($eliminarAdmin);
        $queryAdmin->bindParam(":DNI_personal", $DNI_personal);
        $queryAdmin->execute();


        $eliminarGerente = "DELETE FROM Tgerente WHERE DNI_personal = :DNI_personal";
        $queryGerente = $this->db->prepare($eliminarGerente);
        $queryGerente->bindParam(":DNI_personal", $DNI_personal);
        $queryGerente->execute();


        $eliminarAuxiliar = "DELETE FROM Tauxiliar WHERE DNI_personal = :DNI_personal";
        $queryAuxiliar = $this->db->prepare($eliminarAuxiliar);
        $queryAuxiliar->bindParam(":DNI_personal", $DNI_personal);
        $queryAuxiliar->execute();


        $eliminarMantenimiento = "DELETE FROM Tmantenimiento WHERE DNI_personal = :DNI_personal";
        $queryMantenimiento = $this->db->prepare($eliminarMantenimiento);
        $queryMantenimiento->bindParam(":DNI_personal", $DNI_personal);
        $queryMantenimiento->execute();


        $eliminarTecnico = "DELETE FROM Ttecnico WHERE DNI_personal = :DNI_personal";
        $queryTecnico = $this->db->prepare($eliminarTecnico);
        $queryTecnico->bindParam(":DNI_personal", $DNI_personal);
        $queryTecnico->execute();


        $modificarPersona = "UPDATE Tpersonal SET nombre_per = :nombre_per, apellido1_per = :apellido1_per, ap
        $queryPersona = $this->db->prepare($modificarPersona);
        $queryPersona->bindParam(":nombre_per", $nombre_per);
        $queryPersona->bindParam(":apellido1_per", $apellido1_per);
```

Fase 8: 17 Función editar personal

En la versión web se quedaría de la siguiente manera



Fase 8: 18 Web editar personal

Para borrar el personal tengo el siguiente controlador

```php
<?php
require_once "../../modelo/personal.php";

if (isset($_GET['DNI_personal']) && !empty($_GET['DNI_personal'])) {
    $DNI_personal = $_GET['DNI_personal'];
    $p = new personal();
    $p->setDni_personal($DNI_personal);
    echo $p->delPersona($DNI_personal);
    header("Location:../../vista/personal/listarpersonal.php");
} else {
    echo "Error: DNI de personal no encontrado.";
}
?>
```

Fase 8: 19 controlador borrar personal

Y la siguiente función

```php
function delPersona($DNI_personal) {
    $this->db->beginTransaction();
    try {

        $eliminarAdmin = "DELETE FROM Tadministrativo WHERE DNI_personal = :DNI_personal";
        $queryAdmin = $this->db->prepare($eliminarAdmin);
        $queryAdmin->bindParam(":DNI_personal", $DNI_personal);
        $queryAdmin->execute();


        $eliminarGerente = "DELETE FROM Tgerente WHERE DNI_personal = :DNI_personal";
        $queryGerente = $this->db->prepare($eliminarGerente);
        $queryGerente->bindParam(":DNI_personal", $DNI_personal);
        $queryGerente->execute();


        $eliminarAuxiliar = "DELETE FROM Tauxiliar WHERE DNI_personal = :DNI_personal";
        $queryAuxiliar = $this->db->prepare($eliminarAuxiliar);
        $queryAuxiliar->bindParam(":DNI_personal", $DNI_personal);
        $queryAuxiliar->execute();


        $eliminarMantenimiento = "DELETE FROM Tmantenimiento WHERE DNI_personal = :DNI_personal";
        $queryMantenimiento = $this->db->prepare($eliminarMantenimiento);
        $queryMantenimiento->bindParam(":DNI_personal", $DNI_personal);
        $queryMantenimiento->execute();


        $eliminarTecnico = "DELETE FROM Ttecnico WHERE DNI_personal = :DNI_personal";
        $queryTecnico = $this->db->prepare($eliminarTecnico);
        $queryTecnico->bindParam(":DNI_personal", $DNI_personal);
        $queryTecnico->execute();

        $eliminarPersona = "DELETE FROM Tpersonal WHERE DNI_personal = :DNI_personal";
        $queryPersona = $this->db->prepare($eliminarPersona);
        $queryPersona->bindParam(":DNI_personal", $DNI_personal);
        $queryPersona->execute();

        $this->db->commit();
        $mensaje="Se ha eliminado la persona con DNI ".$DNI_personal." correctamente";
        return $mensaje;
    } catch (Exception $e) {
        $this->db->rollBack();
        $Fallido ="No se ha podido eliminar la persona";
```

Fase 8: 20 Función borrar personal

No tengo una vista porque hice que se ejecutara al darle al botón de borrar

Todas las funciones relacionadas al personal están con transacciones, ya que al insertarlo en tpersonal, también lo inserto en la tabla correspondiente en función del cargo, si lo elimino también lo tengo que eliminar de la tala correspondiente, y si lo actualizo lo borro de la tabla relacionada al cargo y lo vuelvo a meter en la tabla relacionada al cargo, de manera que si hay algún error me haga un rollback y no me haga el commit.

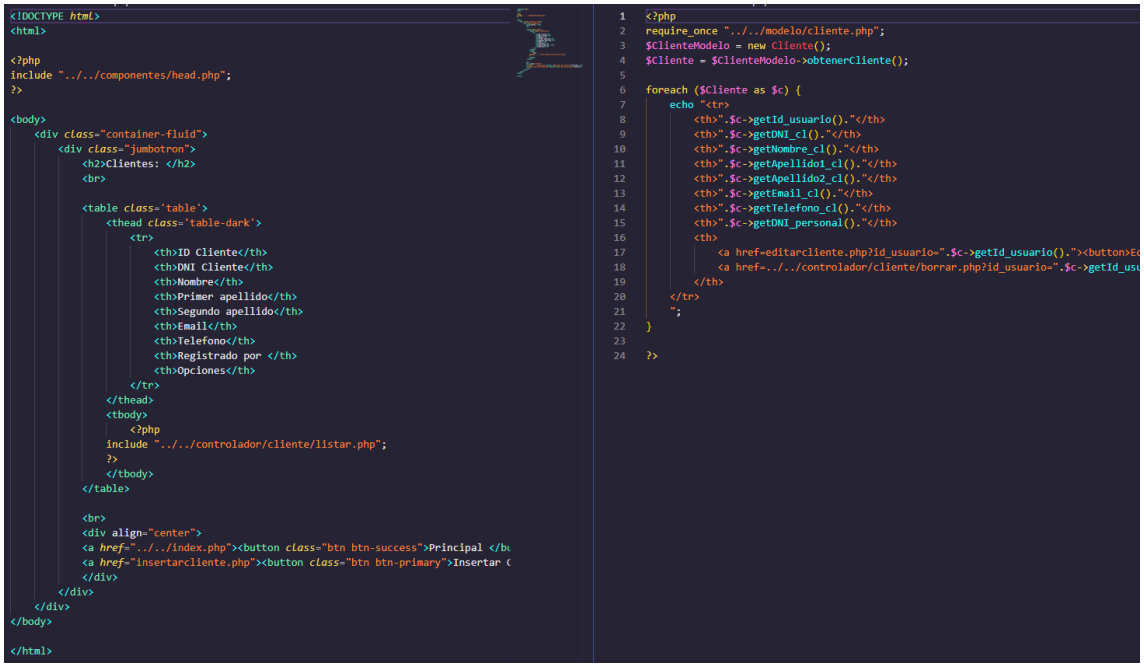
Ya en la parte de cliente, necesitas a alguien en la tabla de personal, ya que para darle de alta tienes que meter el DNI del que le dio de alta.

Repetimos el mismo proceso que hicimos con el personal, pero con cliente

```php
<?php

require_once "bd.php";

13 references | 0 implementations
class Cliente {

    6 references
    private $db;
    4 references
    private $id_usuario;
    2 references
    private $DNI_cl;
    2 references
    private $nombre_cl;
    2 references
    private $apellido1_cl;
    2 references
    private $apellido2_cl;
    2 references
    private $email_cl;
    2 references
    private $telefono_cl;
    2 references
    private $DNI_personal;

    5 references | 0 overrides
    function __construct(){
        $bd = new bd();
        $this->db = $bd->conectarBD();
    }
```

Fase 8: 21 modelo cliente con atributos

Tengo la siguiente vista y controlador de cliente

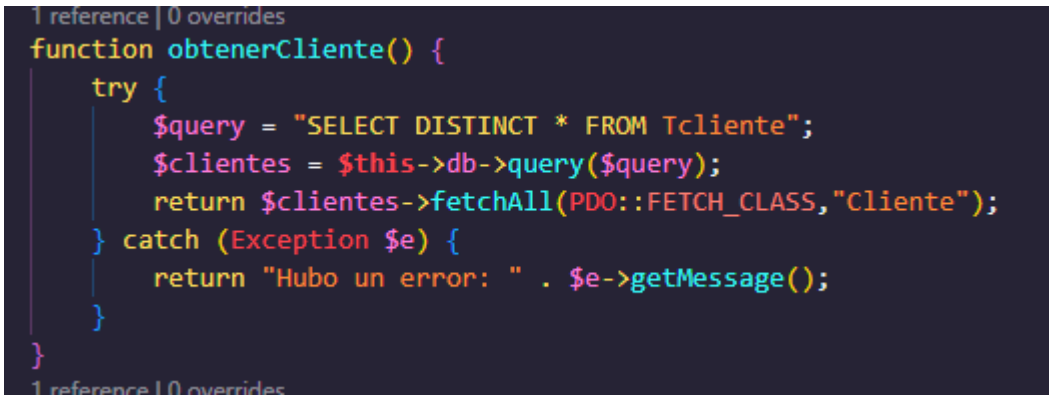


```php
<!DOCTYPE html>
<html>

<?php
include "../../componentes/head.php";
?>

<body>
    <div class="container-fluid">
        <div class="jumbotron">
            <h2>Clientes: </h2>
            <br>

            <table class='table'>
                <thead class='table-dark'>
                    <tr>
                        <th>ID Cliente</th>
                        <th>DNI Cliente</th>
                        <th>Nombre</th>
                        <th>Primer apellido</th>
                        <th>Segundo apellido</th>
                        <th>Email</th>
                        <th>Telefono</th>
                        <th>Registrado por </th>
                        <th>Opciones</th>
                    </tr>
                </thead>
                <tbody>
                    <?php
                include "../../controlador/cliente/listar.php";
                ?>
                </tbody>
            </table>

            <br>
            <div align="center">
            <a href="../../index.php"><button class="btn btn-success">Principal </bu
            <a href="insertarcliente.php"><button class="btn btn-primary">Insertar c
            </div>
        </div>
    </div>
</body>

</html>
```

```php
<?php
require_once "../../modelo/cliente.php";
$ClienteModelo = new Cliente();
$Cliente = $ClienteModelo->obtenerCliente();

foreach ($Cliente as $c) {
    echo "<tr>
        <th>".$c->getId_usuario()."</th>
        <th>".$c->getDNI_cl()."</th>
        <th>".$c->getNombre_cl()."</th>
        <th>".$c->getApellido1_cl()."</th>
        <th>".$c->getApellido2_cl()."</th>
        <th>".$c->getEmail_cl()."</th>
        <th>".$c->getTelefono_cl()."</th>
        <th>".$c->getDNI_personal()."</th>
        <th>
            <a href=editarcliente.php?id_usuario=".$c->getId_usuario()."><button>Ed
            <a href=../../controlador/cliente/borrar.php?id_usuario=".$c->getId_usu
        </th>
    </tr>
    ";
}

?>
```

Fase 8: 22 Vista y controlador para listar clientes

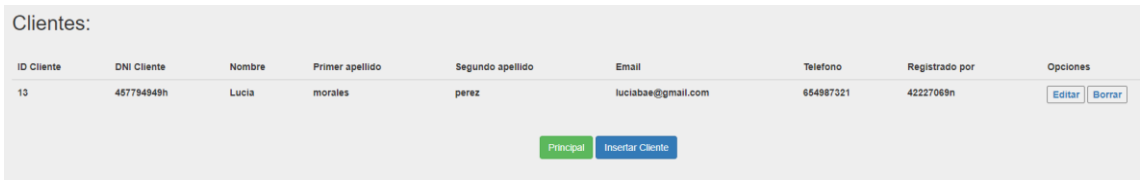
Con esta función para poder obtenerlos y listarlos



```php
1 reference | 0 overrides
function obtenerCliente() {
    try {
        $query = "SELECT DISTINCT * FROM Tcliente";
        $clientes = $this->db->query($query);
        return $clientes->fetchAll(PDO::FETCH_CLASS,"Cliente");
    } catch (Exception $e) {
        return "Hubo un error: " . $e->getMessage();
    }
}
1 reference | 0 overrides
```

Fase 8: 23 Función obtener cliente

Y se vería de esta manera en la página web (de nuevo se ha metido un usuario para mostrar el listado)



Fase 8: 24 Web listado clientes

Para dar de alta a un cliente tenemos la siguiente función

```php
function addCliente($DNI_cl, $nombre_cl, $apellido1_cl, $apellido2_cl, $email_cl, $telefono_cl, $DNI_personal){
    try {
        $insertarCliente= "INSERT INTO tcliente (DNI_cl, nombre_cl, apellido1_cl, apellido2_cl, email_cl, telefono_cl, DNI_personal)
        VALUES(:DNI_cl, :nombre_cl, :apellido1_cl, :apellido2_cl, :email_cl, :telefono_cl, :DNI_personal)";
        $queryCliente=$this->db->prepare($insertarCliente);
        $queryCliente->bindParam(":DNI_cl",$DNI_cl);
        $queryCliente->bindParam(":nombre_cl",$nombre_cl);
        $queryCliente->bindParam(":apellido1_cl",$apellido1_cl);
        $queryCliente->bindParam(":apellido2_cl",$apellido2_cl);
        $queryCliente->bindParam(":email_cl",$email_cl);
        $queryCliente->bindParam(":telefono_cl",$telefono_cl);
        $queryCliente->bindParam(":DNI_personal",$DNI_personal);
        $queryCliente->execute();
        $mensaje="se ha insertado un cliente correctamente con nombre ".$nombre_cl;
        return $mensaje;
    } catch (Exception $ex) {
        echo "Hay un error en ".$ex->getMessage();
        $Fallido ="No se ha podido insertar el cliente.";
        return $Fallido;
    }
}
```

Fase 8: 25 Función insertar cliente

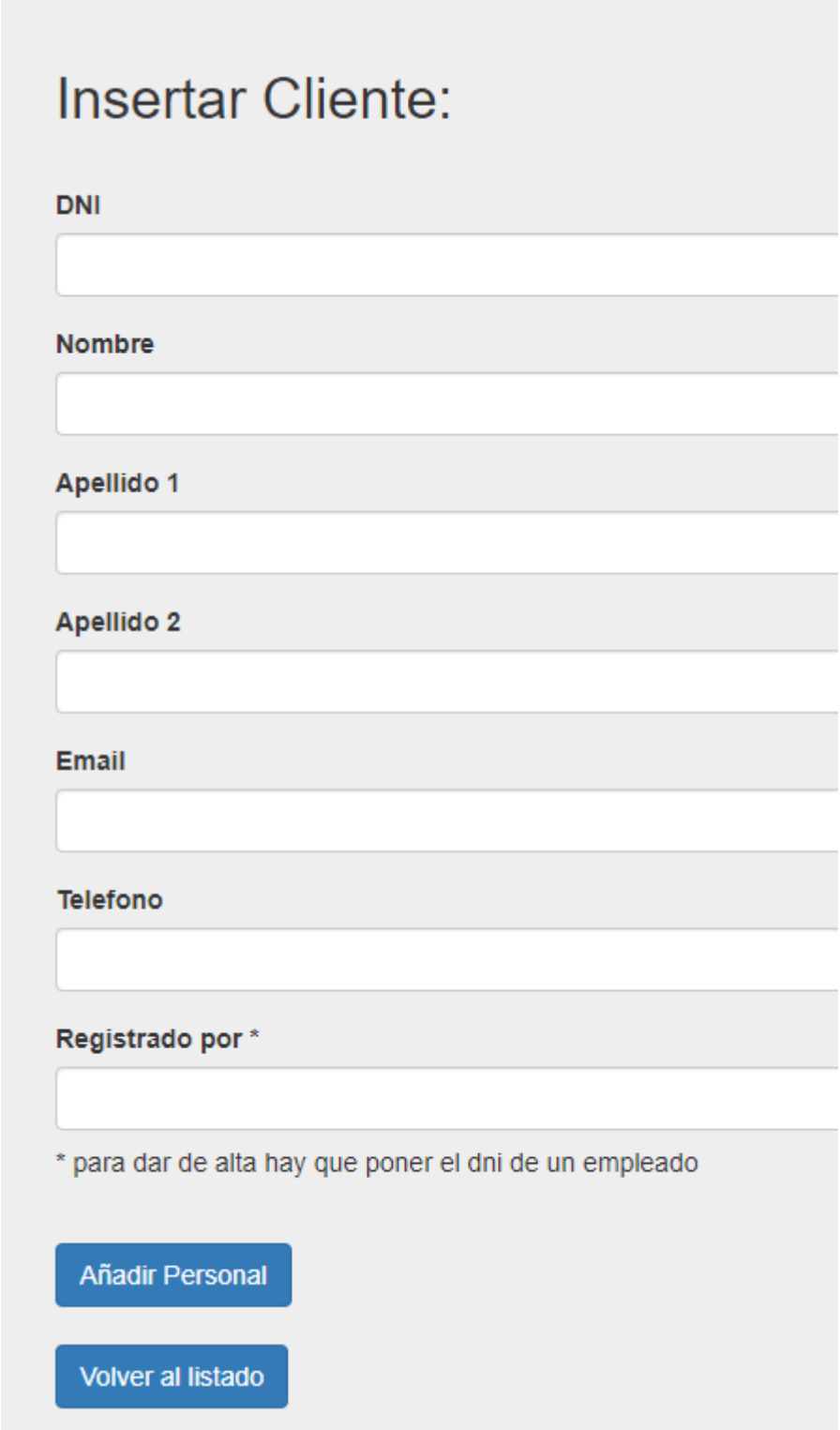
Y tengo el siguiente formulario con el siguiente controlador

```php
<body>
    <div class="container-fluid">
        <div class="jumbotron">
            <h2>Insertar Cliente: </h2>
            <br>
            <?php
                include "../../controlador/cliente/insertar.php";
            ?>

            <form id="insertarPersonal" method="post">
                <div class="form-group">
                    <label>DNI</label>
                    <input type="text" class="form-control" name="DNI_cl">
                </div>
                <div class="form-group">
                    <label>Nombre</label>
                    <input type="text" class="form-control" name="nombre_cl">
                </div>
                <div class="form-group">
                    <label>Apellido 1</label>
                    <input type="text" class="form-control" name="apellido1_cl">
                </div>
                <div class="form-group">
                    <label>Apellido 2</label>
                    <input type="text" class="form-control" name="apellido2_cl">
                </div>
                <div class="form-group">
                    <label>Email</label>
                    <input type="text" class="form-control" name="email_cl">
                </div>
                <div class="form-group">
                    <label>Telefono</label>
                    <input type="text" class="form-control" name="telefono_cl">
```

```php
require_once "../../modelo/cliente.php";
if (
    isset($_POST['DNI_cl'] )
    && isset($_POST['nombre_cl'])
    && isset($_POST['apellido1_cl'])
    && isset($_POST['apellido2_cl'])
    && isset($_POST['email_cl'])
    && isset($_POST['telefono_cl'])
    && isset($_POST['DNI_personal'])
) {
    $dni_cl=$_POST['DNI_cl'];
    $nombre_cl = $_POST['nombre_cl'];
    $apellido1_cl = $_POST['apellido1_cl'];
    $apellido2_cl = $_POST['apellido2_cl'];
    $email_cl = $_POST['email_cl'];
    $telefono_cl = $_POST['telefono_cl'];
    $DNI_personal = $_POST['DNI_personal'];

    $c = new Cliente();
    $c->setDni_cl($dni_cl);
    $c->setNombre_cl($nombre_cl);
    $c->setApellido1_cl($apellido1_cl);
    $c->setApellido2_cl($apellido2_cl);
    $c->setEmail_cl($email_cl);
    $c->setTelefono_cl($telefono_cl);
    $c->setDni_personal($DNI_personal);
    echo $c->addCliente($dni_cl, $nombre_cl, $apellido1_cl, $apellido2_cl, $email_cl,
)
```

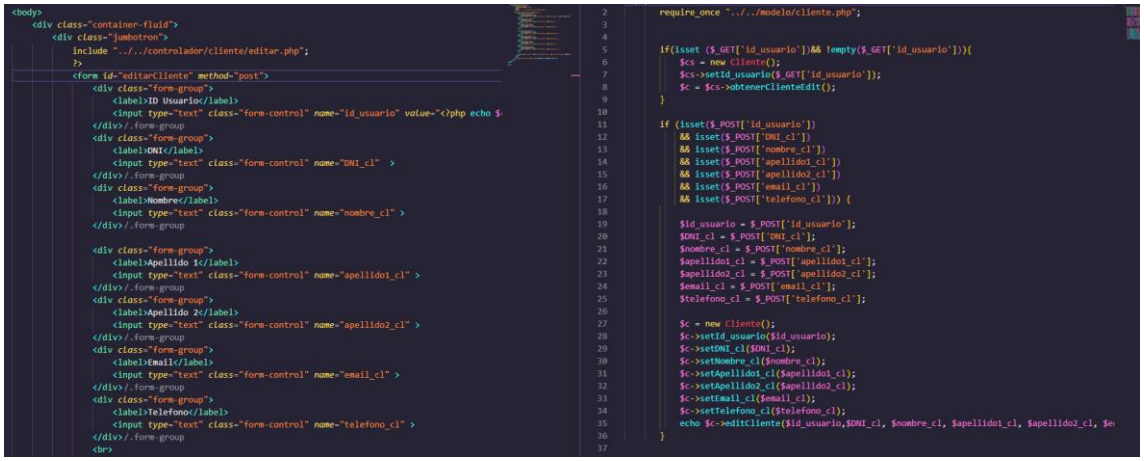
Fase 8: 26 vista y controlador insertar cliente

De manera que se vería así en la web



Fase 8: 27 Web añadir cliente

Para editar al cliente tenemos el siguiente controlador y vista



```
<body>
    <div class="container-fluid">
        <div class="jumbotron">
            include "../../controlador/cliente/editar.php";
            ?>
            <form id="editarCliente" method="post">
                <div class="form-group">
                    <label>ID Usuario</label>
                    <input type="text" class="form-control" name="id_usuario" value="<?php echo $
                </div>/.form-group
                <div class="form-group">
                    <label>DNI</label>
                    <input type="text" class="form-control" name="DNI_cl"  >
                </div>/.form-group
                <div class="form-group">
                    <label>Nombre</label>
                    <input type="text" class="form-control" name="nombre_cl" >
                </div>/.form-group

                <div class="form-group">
                    <label>Apellido 1</label>
                    <input type="text" class="form-control" name="apellido1_cl" >
                </div>/.form-group
                <div class="form-group">
                    <label>Apellido 2</label>
                    <input type="text" class="form-control" name="apellido2_cl" >
                </div>/.form-group
                <div class="form-group">
                    <label>Email</label>
                    <input type="text" class="form-control" name="email_cl" >
                </div>/.form-group
                <div class="form-group">
                    <label>Telefono</label>
                    <input type="text" class="form-control" name="telefono_cl" >
                </div>/.form-group
                <br>
```

```
require_once "../../modelo/cliente.php";


if(isset ($_GET['id_usuario'])&& !empty($_GET['id_usuario'])){
    $cs = new Cliente();
    $cs->setId_usuario($_GET['id_usuario']);
    $c = $cs->obtenerClienteEdit();
}

if (isset($_POST['id_usuario'])
    && isset($_POST['DNI_cl'])
    && isset($_POST['nombre_cl'])
    && isset($_POST['apellido1_cl'])
    && isset($_POST['apellido2_cl'])
    && isset($_POST['email_cl'])
    && isset($_POST['telefono_cl'])) {

    $id_usuario = $_POST['id_usuario'];
    $DNI_cl = $_POST['DNI_cl'];
    $nombre_cl = $_POST['nombre_cl'];
    $apellido1_cl = $_POST['apellido1_cl'];
    $apellido2_cl = $_POST['apellido2_cl'];
    $email_cl = $_POST['email_cl'];
    $telefono_cl = $_POST['telefono_cl'];

    $c = new Cliente();
    $c->setId_usuario($id_usuario);
    $c->setDNI_cl($DNI_cl);
    $c->setNombre_cl($nombre_cl);
    $c->setApellido1_cl($apellido1_cl);
    $c->setApellido2_cl($apellido2_cl);
    $c->setEmail_cl($email_cl);
    $c->setTelefono_cl($telefono_cl);
    echo $c->editCliente($id_usuario,$DNI_cl, $nombre_cl, $apellido1_cl, $apellido2_cl, $e
}
```

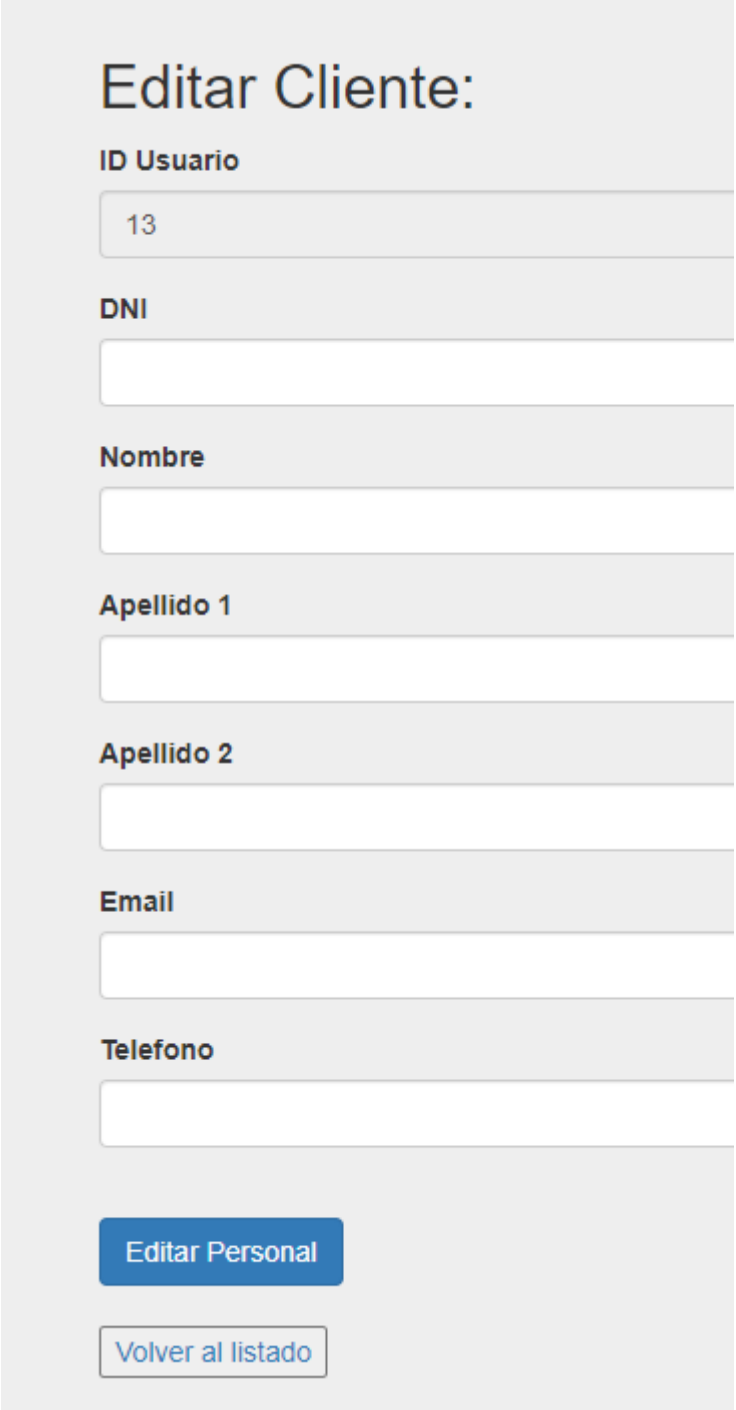
Fase 8: 28 vista y controlador para editar cliente

En este formulario solo recogemos el id de usuario y podemos modificar lo demás a través de la siguiente función

```
public function editCliente($id_usuario,$DNI_cl, $nombre_cl, $apellido1_cl, $apellido2_cl, $email_cl, $telefono_cl) {
    $editCliente="UPDATE tcliente set DNI_cl = :DNI_cl, nombre_cl = :nombre_cl, apellido1_cl = :apellido1_cl,
        apellido2_cl = :apellido2_cl, email_cl = :email_cl, telefono_cl = :telefono_cl WHERE id_usuario = :id_usuario";
    $queryCliente = $this->db->prepare($editCliente);
    $queryCliente->bindParam(":id_usuario", $id_usuario);
    $queryCliente->bindParam(":DNI_cl", $DNI_cl);
    $queryCliente->bindParam(":nombre_cl", $nombre_cl);
    $queryCliente->bindParam(":apellido1_cl", $apellido1_cl);
    $queryCliente->bindParam(":apellido2_cl", $apellido2_cl);
    $queryCliente->bindParam(":email_cl", $email_cl);
    $queryCliente->bindParam(":telefono_cl", $telefono_cl);
    $queryCliente->execute();

    $mensaje="Se ha editado el cliente correctamente.";
    return $mensaje;
}
```

Fase 8: 29 Función editar cliente

Y se vería de la siguiente manera en la web



Fase 8: 30 Web editar cliente

Para borrar al cliente, lo tengo igual que personal, le das al borrar y se ejecuta el controlador y la función de borrar

```
function delCliente(){
    $borrarCliente ="DELETE FROM tcliente WHERE id_usuario=:id_usuario";
    $queryBorrar =$this->db->prepare($borrarCliente);
    $queryBorrar->bindParam(":id_usuario", $this->id_usuario);
    $queryBorrar->execute();
    $mensaje="Se ha eliminado el cliente correctamente.";
    return $mensaje;
}
/**
 * @return mixed
 */
0 references | 0 overrides
```

```
require_once "../../modelo/cliente.php";

if (isset($_GET['id_usuario']) && !empty($_GET['id_usuario'])) {
    $id_usuario = $_GET['id_usuario'];
    $p = new cliente();
    $p->setId_usuario($id_usuario);
    echo $p->delCliente();
    header("Location:../../vista/cliente/listarcliente.php");
} else {
    echo "Error: id de usuario no encontrado.";
}
?>
```

Fase 8: 31 controlador y función borrar cliente

De la parte de diseño no se ha retocado por falta de tiempo simplemente se ha puesto en el head un icono para que no aparezca el de xampp y el título de la empresa

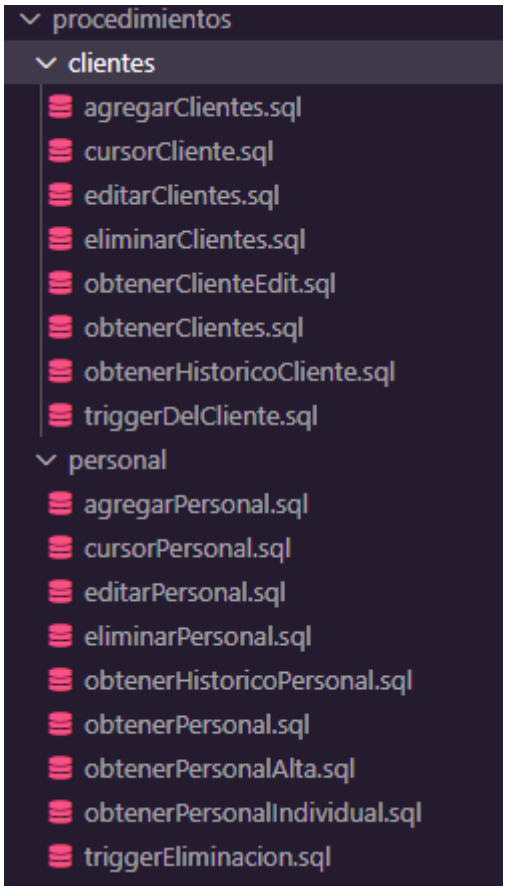
12- Cambios Fase 7 a Fase 8

Se ha cambiado la base de datos para dejarla más pequeña y enfocarla solo a las funcionalidades pensadas, aplicando algunos cambios que se habían hecho en las fases anteriores con transacciones. Esto lo hice literalmente para no distraerme y centrarme en realizar bien las funciones, solo para esta parte practica

13- Fase 9 Aplicación BD en PHP con procedimientos almacenados

Para esta fase del proyecto, lo primero que hay que hacer es pasar todas las funciones que teníamos para la fase anterior a procedimientos de almacenado, para que se realice las acciones CRUD de al menos 4 tablas, también hay que meter mínimo dos triggers, dos cursores y un total de 4 excepciones, 2 habituales y 2 personalizadas.

Empezamos con los procedimientos, pongo algunos ejemplos porque si no se alarga demasiado al poner captura de la función y del procedimiento, pero todos ellos se pueden comprobar en la carpeta de procedimientos



Fase 9: 1 carpeta con procedimientos

La idea es modificar la función para que llame al procedimiento

```php
function obtenerPersonal(){
    try {
        $querySelect = "CALL sp_obtenerPersonal()";
        $listapersonal=$this->db->prepare($querySelect);
        $listapersonal->execute();

    } catch (Exception $ex) {
        echo "Te equivocaste en: ".$ex->getMessage();
    }
    return $listapersonal->fetchALL(PDO::FETCH_CLASS,"personal");
}
```

Fase 9: 2 función obtenerPersonal

Y el procedimiento este en la base de datos,

```sql
-- procedimiento para el personal que da de alta
DELIMITER //
CREATE PROCEDURE `sp_obtenerPersonal`()
BEGIN
 SELECT * FROM tpersonal ORDER BY cargo ;
END //
DELIMITER ;
```

Fase 9: 3 procedimiento sp_obtenerPersonal

Después con los triggers que son acciones que se ejecutan en cuanto algo se ejecute en la base de datos (puede ser antes o después) como este por ejemplo que se ejecuta después del procedimiento de eliminar, que se ejecuta con la siguiente función

```php
function delPersona($DNI_personal) {
    $this->db->beginTransaction();
    try {
        $eliminarPersona = "CALL sp_delPersona(:DNI_personal)";
        $queryPersona = $this->db->prepare($eliminarPersona);
        $queryPersona->bindParam(":DNI_personal", $DNI_personal);
        $queryPersona->execute();

        $this->db->commit();
        $mensaje="Se ha eliminado la persona con DNI ".$DNI_personal." correctamente";
        return $mensaje;

    } catch (Exception $e) {
        $this->db->rollBack();
        $Fallido ="No se ha podido eliminar la persona";
        return $Fallido;
    }
}
```

Fase 9: 4 función delPersona

De esa función se llama al procedimiento

```
-- procedimiento para eliminar persona
DELIMITER //
CREATE PROCEDURE `sp_delPersona` (IN
_DNI_personal VARCHAR(10))
BEGIN
    DELETE FROM Tadministrativo WHERE DNI_personal = _DNI_personal;
    DELETE FROM Tgerente WHERE DNI_personal = _DNI_personal;
    DELETE FROM Tauxiliar WHERE DNI_personal = _DNI_personal;
    DELETE FROM Tmantenimiento WHERE DNI_personal = _DNI_personal;
    DELETE FROM Ttecnico WHERE DNI_personal = _DNI_personal;
    DELETE FROM Tpersonal WHERE DNI_personal = _DNI_personal;
END //
DELIMITER ;
```

Fase 9: 5 procedimiento sp_delPersona

Y una vez ejecutado entra la función del trigger que nos hace un guardado en otra tabla

```
-- trigger para el guardado del eliminado
DELIMITER //
CREATE TRIGGER trigger_delPersonal
   AFTER DELETE ON tpersonal
   FOR EACH ROW
   BEGIN
      INSERT INTO historicoPersonal(DNI_personal,estado)
                  VALUES (OLD.DNI_personal, 'Eliminado');
   END; //
DELIMITER ;
```

Fase 9: 6 trigger_delPersonal

En mi caso hice lo mismo cuando eliminaba un cliente

Se llama a la función para eliminar al cliente

```php
1 reference | 0 overrides
function delCliente(){
    $query = "CALL sp_eliminar_cliente(:id_usuario, @mensaje)";
    $queryCliente = $this->db->prepare($query);
    $queryCliente->bindParam(":id_usuario", $this->id_usuario);
    $queryCliente->execute();

    $output = $this->db->query("SELECT @mensaje as mensaje")->fetch(PDO::FETCH_ASSOC);
    $mensaje = $output['mensaje'];

return $mensaje;
}
/**
 * @return mixed
 */
1 reference | 0 overrides
public function getId_usuario() {
    return $this->id_usuario;
}
```

Fase 9: 7 función delCliente

Que ejecuta este procedimiento

```sql
DELIMITER //
CREATE PROCEDURE sp_eliminar_cliente(
    IN id_usuario INT,
    OUT mensaje VARCHAR(255)
)
BEGIN
    DECLARE EXIT HANDLER FOR SQLEXCEPTION
    BEGIN
        ROLLBACK;
        SELECT "No se ha podido eliminar el usuario " AS Errors;
    END;
    START TRANSACTION;
    UPDATE tcliente SET DNI_personal = NULL WHERE id_usuario = id_usuario;
    DELETE FROM tcliente WHERE id_usuario = id_usuario;
    COMMIT;
    SET mensaje = 'Se ha eliminado el cliente correctamente.';
END //
DELIMITER ;
```

Fase 9: 8 procedimiento sp_eliminar_cliente

y activa este trigger

```
    -- trigger para el guardado en historico tras su eliminacion
DELIMITER //
CREATE TRIGGER trigger_delCliente
   AFTER DELETE ON tcliente
   FOR EACH ROW
   BEGIN
      INSERT INTO historicocliente(DNI_cl,estado)
                  VALUES (OLD.DNI_cl, 'Eliminado');
   END; //
DELIMITER ;
```

Fase 9: 9 trigger_delCliente

Después los dos cursores que hay que crear, se crearon para llevar el control del personal y clientes creados en sus respectivas paginas

Creamos una función que contara los clientes

```
function contarCliente(){
    try{
        $query ="CALL cur_contar_clientes()";
        $stmt = $this->db->prepare($query);
        $stmt->execute();
        $data = $stmt->fetch(PDO::FETCH_ASSOC);
        $total_personas = $data['total_personas'];
    }catch (Exception $ex){
        echo "Ocurrió un error: ".$ex->getMessage();
    }
    return $total_personas;
}
```

Fase 9: 10 función contarCliente

Que llama a este cursor

```sql
DELIMITER //

CREATE PROCEDURE cur_contar_clientes()
BEGIN
    DECLARE done BOOLEAN DEFAULT FALSE;
    DECLARE total INT DEFAULT 0;
    DECLARE dni VARCHAR(9);

    -- Creamos el cursor que nos permitirá recorrer los registros de la tabla tcliente
    DECLARE cur CURSOR FOR SELECT id_usuario FROM Tcliente;

    -- Definimos el manejador de errores
    DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = TRUE;

    -- Abrimos el cursor y leemos los registros
    OPEN cur;

    read_loop: LOOP
        FETCH cur INTO dni;

        IF done THEN
            LEAVE read_loop;
        END IF;

        -- Incrementamos el contador de personas
        SET total = total + 1;
    END LOOP;

    -- Cerramos el cursor
    CLOSE cur;

    -- Mostramos el resultado
    SELECT CONCAT('Total de clientes creados: ', total) AS total_personas;
END //

DELIMITER ;
```

Fase 9: 11 cursor cur_contar_cliente

Y para integrarlo en la vista se creó este controlador

proyectoBasePHP > controlador > cliente > contar.php

```php
<?php
require_once "../../modelo/cliente.php";
    $cliente = new Cliente();
    $cuantoscli = $cliente->contarCliente();
    echo $cuantoscli;

?>
```

Fase 9: 12 controlador contar

Que se integró en la vista en la parte deseada

```php
<div align="center"><br>
<?php
include "../../controlador/cliente/contar .php";
?>

</div>
```

Fase 9: 13 llamado al controlador contar

Para que nos dijera cuantos clientes habían registrados en ese momento

Fase 9: 14 comprobación de total de clientes

Este procedimiento se realizó también en la parte de personal

Las transacciones aparte de las dos que había creadas de la fase anterior, se crearon dos nuevas, una dentro de la función de delpersona, para que no se borrara al cliente al eliminar al empleado

```php
function delPersona($DNI_personal) {
    $this->db->beginTransaction();
    try {
        $eliminarPersona = "CALL sp_delPersona(:DNI_personal)";
        $queryPersona = $this->db->prepare($eliminarPersona);
        $queryPersona->bindParam(":DNI_personal", $DNI_personal);
        $queryPersona->execute();

        $this->db->commit();
        $mensaje="Se ha eliminado la persona con DNI ".$DNI_personal." correctamente";
        return $mensaje;

    } catch (Exception $e) {
        $this->db->rollBack();
        $Fallido ="No se ha podido eliminar la persona";
        return $Fallido;
    }
}
```

Fase 9: 15 función delPersona

Y creamos la otra transacción dentro del procedimiento de eliminar cliente para lo mismo, que no se nos borre el personal al eliminar el cliente

```
DELIMITER //
CREATE PROCEDURE sp_eliminar_cliente(
    IN id_usuario INT,
    OUT mensaje VARCHAR(255)
)
BEGIN
    DECLARE EXIT HANDLER FOR SQLEXCEPTION
    BEGIN
        ROLLBACK;
        SELECT "No se ha podido eliminar el usuario " AS Errors;
    END;
    START TRANSACTION;
    UPDATE tcliente SET DNI_personal = NULL WHERE id_usuario = id_usuario;
    DELETE FROM tcliente WHERE id_usuario = id_usuario;
    COMMIT;
    SET mensaje = 'Se ha eliminado el cliente correctamente.';
END //
DELIMITER ;
```

Fase 9: 16 procedimiento sp_eliminar cliente con excepciones

Para las excepciones, aparte de las creadas en las funciones, dentro de los procedimientos se crearon las siguientes en añadir personal y añadir clientes respectivamente añadiendo 1 excepción de cada para ello

```
DELIMITER //
CREATE PROCEDURE `sp_addPersona` (
    IN _DNI_personal VARCHAR(10),
    IN _nombre_per VARCHAR(50),
    IN _apellido1_per VARCHAR(50),
    IN _apellido2_per VARCHAR(50),
    IN _email_per VARCHAR(100),
    IN _telefono_per VARCHAR(20),
    IN _cargo VARCHAR(50))
BEGIN

    DECLARE EXIT HANDLER FOR SQLSTATE '23000'
    BEGIN
        SELECT 'Datos duplicados' AS ERROR;
    END;
    IF _DNI_personal = null OR _DNI_personal = '' THEN
        SIGNAL SQLSTATE '45000'
            SET MESSAGE_TEXT = 'EL DNI NO PUEDE IR VACIO';
    end if;
```

Fase 9: 17 procedimiento sp_addPersona con excepciones

```
DELIMITER //
CREATE PROCEDURE sp_agregar_cliente(
    IN DNI_cl VARCHAR(20),
    IN nombre_cl VARCHAR(50),
    IN apellido1_cl VARCHAR(50),
    IN apellido2_cl VARCHAR(50),
    IN email_cl VARCHAR(50),
    IN telefono_cl VARCHAR(20),
    IN DNI_personal VARCHAR(20),
    OUT mensaje VARCHAR(100)
)
BEGIN
    DECLARE Fallido VARCHAR(100);
    DECLARE exito VARCHAR(100);

    DECLARE EXIT HANDLER FOR SQLSTATE '23000'
    BEGIN
        SELECT 'Datos duplicados' AS ERROR;
    END;
```

Fase 9: 18 procedimiento sp_agregar_cliente con excepciones

14- Cambios Fase 8 a Fase 9

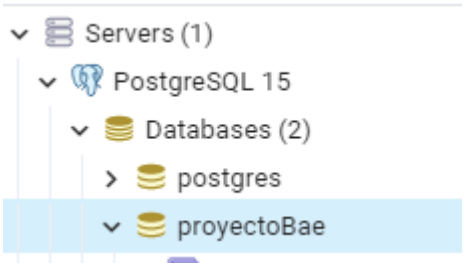
La mayor parte del cambio viene en la base de datos, que al incluirla por completo, para que no fuera demasiadas líneas se aplicaron directamente los cambios que se habían realizado con las transacciones anteriores (por ejemplo antes estaba la tabla de tadministrativo con id_adm, cargo y dni_personal, que en una transacción a posterior se cambió a id_adm, nombre y dni_personal), por lo que aquí se dejó directamente el cambio realizado en lugar de cargar primero la tabla de una manera y después hacerse la transacción, también se eliminó una tabla intermedia que era tpersonalcliente, ya que se estaba repitiendo la conexión en la tabla personal, en la tabla cliente y en la tabla eliminada, cuando solo al tenerlo en la tabla cliente servía para esa relación. Aparte se agregaron dos tablas de historial de personal y cliente, para los datos recogidos por el trigger.

También como se había indicado en la corrección de la fase anterior, se ha creado el selector a la hora de añadir al cliente, se ha solucionado el fallo de INDEX al importar la base de datos y se modificó el paso de variables de addpersona.

15-Fase 10: Aplicación BD en PHP con Base de datos Objeto-Relacionales

Para esta fase hubo que hacer la migración a la base de datos PostgreSQL, ajustando los CRUD que se habían creado anteriormente para la nueva base de datos.

En este caso se creo la base de datos proyectoBae en PostgreSQL



Fase 10: 1 Creación de la base de datos

y se ajusto el PDO de conexión a la base de datos nueva y se dejo comentada la parte del trabajo anterior

```php
<?php
16 references | 0 implementations
class bd
{
    // private $cadenaConexion = "mysql:dbname=trabajobae;host=localhost;port=3306";
    8 references
    private $cadenaConexion = "pgsql:dbname=proyectoBae;host=localhost;port=5432";
    // private $usuario = "root";
    8 references
    private $usuario = "postgres";
    8 references
    private $clave = "1234";

    16 references | 0 overrides | Codeium: Refactor | Explain | Generate Function Comment
    function conectarBD()
    {
        try {
            return new PDO($this->cadenaConexion, $this->usuario, $this->clave);
        } catch (PDOException $e) {
            echo "Error ocurrio un problema en Base de datos: " . $e->getMessage();
            return null;
        }
    }
}
```

Fase 10: 2 modificación del PDO

Después para pasar la base de datos o bien se podía migrar la base o bien crearla de nuevo, en mi caso como era una base de datos relativamente pequeña la fui creando desde 0, para ir ajustando lo que se solicitaba y que no se me quedara nada atrás. Estos requisitos era crear objetos de tipo personalizados y creaciones de tabla en la base de datos. En mi caso con las tablas que usaba para el trabajo tuve que crear 2 enumerados y un tipo, ya que después el resto de tablas usadas podían heredar de la tabla creada con el tipo

Primero creamos los dos enumerados que había que hacer como mínimo, que en mi caso son cargo y estado

```
CREATE TYPE cargo as ENUM(
    'Administrativo','Gerente','Tecnico','Auxiliar','Mantenimiento','Limpiador'
);

CREATE TYPE estado AS ENUM(
    'Activo','Inactivo'
);
```

Fase 10: 3 Enum creados

Después creamos el tipo personal que se usa para la tabla tpersonal, usando los dos enum que hicimos anteriormente

```
CREATE TYPE tipo_personal AS
(
  id integer,
    DNI_personal character varying(50),
    cargo cargo,
    nombre_per character varying(45),
    apellido1_per character varying(45),
    apellido2_per character varying(45),
    email_per character varying(45),
    telefono_per INTEGER,
    estado estado
);

CREATE SEQUENCE id_personal_seq;
CREATE TABLE tpersonal OF tipo_personal(
    id DEFAULT nextval('id_personal_seq')
);
```

Fase 10: 4 creación de tipo y de la tabla dependiente del tipo

Como las demás tablas usadas para la practica dependen de DNI_personal le establecemos un constraint para que su valor sea único y no se pueda repetir, y así poder usarlo para la herencia

```
ALTER TABLE tpersonal ADD CONSTRAINT DNI_personal_unique UNIQUE (DNI_personal);
```

Fase 10: 5 ponemos DNI_personal como único

Después empezamos a crear las tablas de las especializaciones de personal y la de cliente

```
CREATE TABLE Tgerente (
  ID_ger SERIAL PRIMARY KEY,
  Nombre_ger character varying(45),
  DNI_personal character varying(9) REFERENCES tpersonal (DNI_personal) ON DELETE SET NULL ON U
);

CREATE TABLE Tauxiliar (
  ID_aux SERIAL PRIMARY KEY,
  Nombre_aux character varying(45),
  DNI_personal character varying(9) REFERENCES tpersonal (DNI_personal) ON DELETE SET NULL ON U
);

CREATE TABLE Tadministrativo (
  ID_adm SERIAL PRIMARY KEY,
  Nombre_adm character varying(45),
  DNI_personal character varying(9) REFERENCES tpersonal (DNI_personal) ON DELETE SET NULL ON U
);

CREATE TABLE Tmantenimiento (
  ID_man SERIAL PRIMARY KEY,
  Nombre_man character varying(45),
  DNI_personal character varying(9) REFERENCES tpersonal (DNI_personal) ON DELETE SET NULL ON U
);

CREATE TABLE Ttecnico (
  ID_tec SERIAL PRIMARY KEY,
  Nombre_tec character varying(45),
  DNI_personal character varying(9) REFERENCES tpersonal (DNI_personal) ON DELETE SET NULL ON U
);

CREATE TABLE Tlimpiador (
  ID_lim SERIAL PRIMARY KEY,
  nombre_lim character varying(45),
  DNI_personal character varying(45) REFERENCES tpersonal (DNI_personal) ON DELETE SET NULL ON
);
```

Fase 10: 6 tablas que heredan de tpersonal

```
CREATE TABLE Tcliente (
  id_usuario SERIAL PRIMARY KEY,
  DNI_cl character varying(45),
  nombre_cl character varying(45),
  apellido1_cl character varying(45),
  apellido2_cl character varying(45),
  email_cl character varying(45),
  telefono_cl INTEGER,
  DNI_personal character varying(9) REFERENCES tpersonal (DNI_personal) ON DELETE SET NULL ON U
);
```

Fase 10: 7 tabla cliente

También las tablas de histórico de cliente y personal, en el que también usamos el enum de estado, pero no quise unirlas a las otras tablas para así dejarlo solo como una copia de seguridad

```
CREATE TABLE historicoPersonal (
  id SERIAL PRIMARY KEY,
  DNI_personal character varying(45),
  estado estado
);

CREATE TABLE historicoCliente (
  id SERIAL PRIMARY KEY,
  DNI_cl character varying(45),
  estado estado
);
```

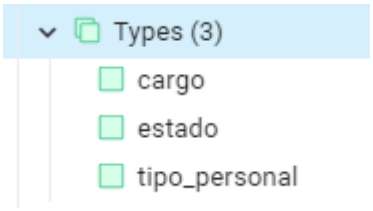
Fase 10: 8 tabla de históricos

El resto de tablas se ajusto para entrar en PostgreSQL, pero sin crear tipos

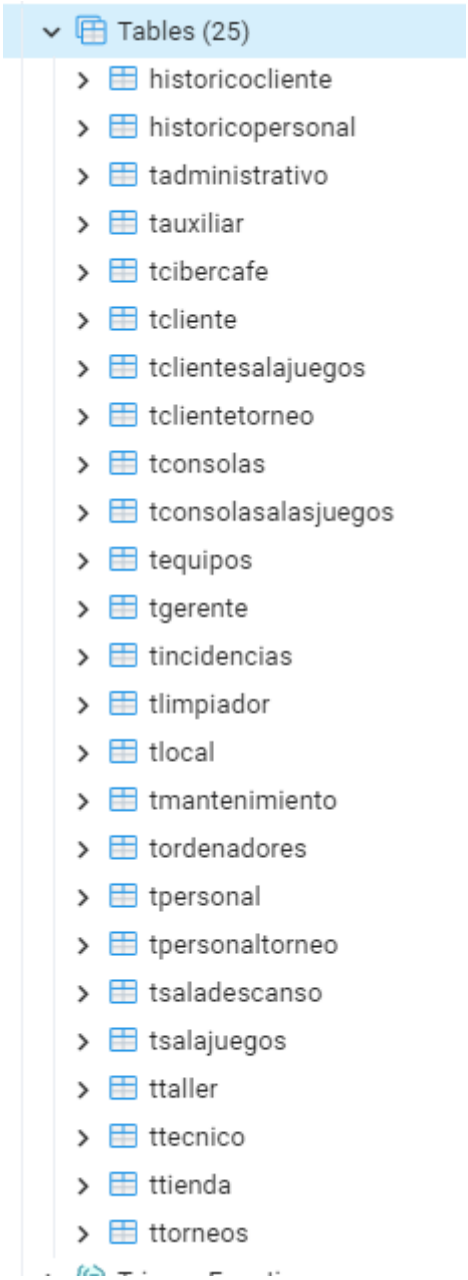
```
CREATE TABLE Tsaladescanso (
  id_saladescanso SERIAL PRIMARY KEY,
  cod_local INTEGER,
  horario character varying(45)
);

CREATE TABLE TClienteSalajuegos (
  ID_CLSJ SERIAL PRIMARY KEY,
  ID_salajuegos INTEGER REFERENCES Tsalajuegos (ID_salajuegos) ON DELETE SET NULL ON UPDATE SE
  ID_usuario INTEGER REFERENCES Tcliente (id_usuario) ON DELETE SET NULL ON UPDATE SET NULL
);

CREATE TABLE TClienteTorneo (
  ID_CLT SERIAL PRIMARY KEY,
  ID_torneo INTEGER REFERENCES Ttorneos (id_torneo) ON DELETE SET NULL ON UPDATE SET NULL,
  ID_usuario INTEGER REFERENCES Tcliente (id_usuario) ON DELETE SET NULL ON UPDATE SET NULL
);

CREATE TABLE TPersonalTorneo (
  ID_PT SERIAL PRIMARY KEY,
  ID_torneo INTEGER REFERENCES Ttorneos (id_torneo) ON DELETE SET NULL ON UPDATE SET NULL,
  DNI_personal character varying(9) REFERENCES tpersonal (DNI_personal) ON DELETE SET NULL ON
);

CREATE TABLE TConsolaSalasJuegos (
  ID_CSJ SERIAL PRIMARY KEY,
  ID_salajuegos INTEGER REFERENCES Tsalajuegos (ID_salajuegos) ON DELETE SET NULL ON UPDATE SE
  ID_consola INTEGER REFERENCES Tconsolas (ID_consolas) ON DELETE SET NULL ON UPDATE SET NULL
);
```

Fase 10: 9 tablas normales

Una vez que tengamos esto lo pasamos a PostgreSQL y nos quedaría algo así con respecto a las tablas y tipos



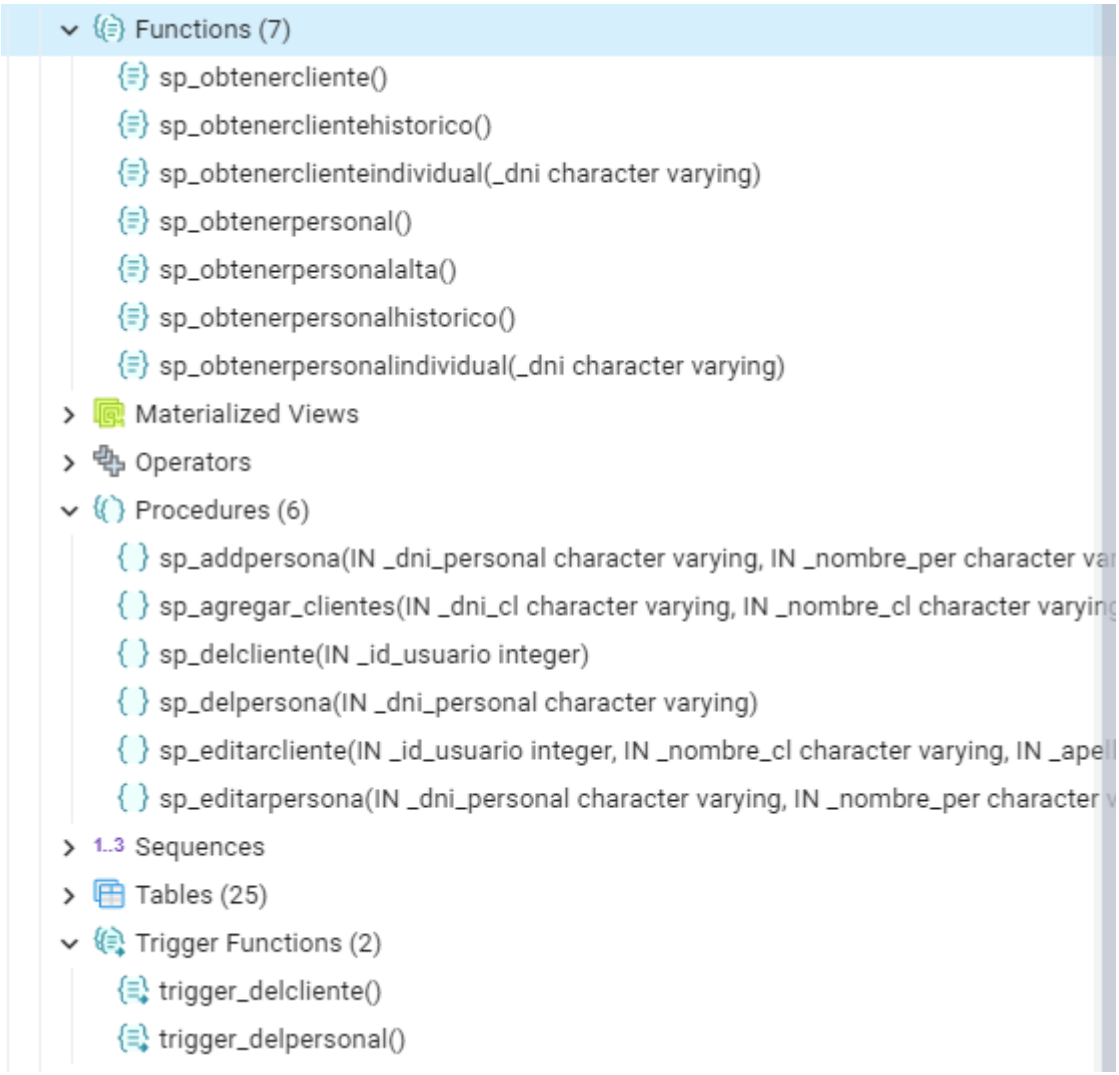
Fase 10: 10 tipos creados en PostgreSQL



Fase 10: 11 tablas creadas en PostgreSQL

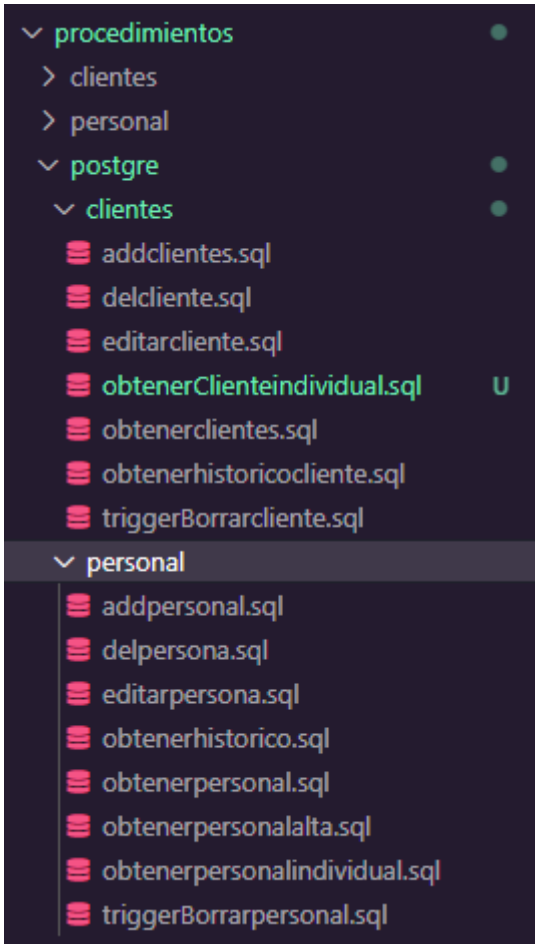
Con esto ya tendríamos la base de datos funcionando, pero nos faltarían los procedimientos y llamarlos desde PHP para así poder interactuar con la base de datos para ejecutar los CRUD necesarios.

En mi caso he creado los siguientes procedimientos, funciones y triggers



Fase 10: 12 funciones, procedimientos y trigger en PostgreSQL

Los procedimientos en SQL y como se llamaron desde las funciones fue de la siguiente manera



Fase 10: 13 procedimientos creados para PostgreSQL

Empezamos con listar el personal para así poder ver los datos que tengamos en las tablas

```
--procedimiento para listar el personal
CREATE OR REPLACE FUNCTION sp_obtenerPersonal(
    )
    RETURNS SETOF tpersonal
    LANGUAGE 'plpgsql'
AS $BODY$
    BEGIN
        RETURN QUERY SELECT * FROM tpersonal;
    END;
$BODY$;
```

Fase 10: 14 Procedimiento usado para listar personal

```php
1 reference | 0 overrides | Codeium: Refactor | Explain | Generate Function Comment
function obtenerPersonal(){
    try {
        $querySelect = "SELECT * FROM public.sp_obtenerpersonal()";
        $listapersonal=$this->db->prepare($querySelect);
        $listapersonal->execute();

        if($listapersonal){
            return $listapersonal->fetchAll(PDO::FETCH_CLASS,'personal');
        }else{
            echo "no se pudo obtener listado";
        }

    } catch (Exception $ex) {
        echo "Te equivocaste en: ".$ex->getMessage();
    }
    return $listapersonal->fetchALL(PDO::FETCH_CLASS,"personal");
}
```

Fase 10: 15 Función usada para listar el personal

Para añadir un miembro del personal

```php
1 reference | 0 overrides
function addPersona() {
    try {
        $insertarPersona = "CALL public.sp_addpersona($this)";
        $queryPersona = $this->db->query($insertarPersona);

        if ($queryPersona) {
            echo "Personal creado correctamente ";
            header("Location:/proyectoBasePHP/vista/personal/listarpersonal.php");
        } else {
            echo "Ocurrió un error inesperado al insertar el personal";
        }
    } catch (Exception $e) {
        echo "no se ha podido insertar ".var_dump($e);
        return null;
    }
}
```

Fase 10: 16 Función usada para añadir personal

```sql
-- --procedimiento para añadir personas

CREATE OR REPLACE PROCEDURE public.sp_addPersona(
  in persona tipo_personal
)
LANGUAGE plpgsql
AS $BODY$
BEGIN
  INSERT INTO tpersonal (dni_personal,cargo, nombre_per, apellido1_per,
  VALUES (persona.dni_personal,persona.cargo, persona.nombre_per, person
  IF (persona.cargo = 'Administrativo') THEN
    INSERT INTO tadministrativo (dni_personal, nombre_adm)
    VALUES (persona.dni_personal, persona.nombre_per);
  ELSIF (persona.cargo = 'Gerente') THEN
    INSERT INTO tgerente (dni_personal, nombre_ger)
    VALUES (persona.dni_personal, persona.nombre_per);
  ELSIF (persona.cargo = 'Auxiliar') THEN
    INSERT INTO tauxiliar (dni_personal, nombre_aux)
    VALUES (persona.dni_personal, persona.nombre_per);
  ELSIF (persona.cargo = 'Mantenimiento') THEN
    INSERT INTO tmantenimiento (dni_personal, nombre_man)
    VALUES (persona.dni_personal, persona.nombre_per);
  ELSIF (persona.cargo = 'Tecnico') THEN
    INSERT INTO ttecnico (dni_personal, nombre_tec)
    VALUES (persona.dni_personal, persona.nombre_per);
  ELSIF (persona.cargo = 'Limpiador') THEN
    INSERT INTO tlimpiador (dni_personal, nombre_lim)
    VALUES (persona.dni_personal, persona.nombre_per);
  END IF;
END
```

Fase 10: 17 Procedimiento usado para añadir personal

Para eliminar un personal

```
--procedimiento para eliminar persona
create procedure sp_delPersona(
 _dni_personal character varying
)
LANGUAGE plpgsql
AS $BODY$
BEGIN
    DELETE FROM tadministrativo WHERE dni_personal = _dni_personal;
    DELETE FROM tgerente WHERE dni_personal = _dni_personal;
    DELETE FROM tauxiliar WHERE dni_personal = _dni_personal;
    DELETE FROM tmantenimiento WHERE dni_personal = _dni_personal;
    DELETE FROM ttecnico WHERE dni_personal = _dni_personal;
    DELETE FROM tpersonal WHERE dni_personal = _dni_personal;
end
$BODY$
```

Fase 10: 18 Procedimiento usado para eliminar personal

```
1 reference | 0 overrides | Codeium: Refactor | Explain | Generate Function Comment
function delPersona($dni_personal) {
    try {
        $eliminarPersona = "CALL sp_delPersona(:dni_personal)";
        $queryPersona = $this->db->prepare($eliminarPersona);
        $queryPersona->bindParam(":dni_personal", $dni_personal);
        $queryPersona->execute();

        $mensaje="Se ha eliminado la persona con dni ".$dni_personal." correctamente";
        return $mensaje;

    } catch (Exception $e) {
        $Fallido ="No se ha podido eliminar la persona".$e;
        return $Fallido;
    }
}
1 reference | 0 overrides | Codeium: Refactor | Explain | Generate Function Comment
```

Fase 10: 19 Función para eliminar personal

Para poder editar un miembro del personal tenemos primero que elegirlo

```php
function obtenerPersonaindividual($dni_personal){
    try {
        $query = "SELECT * from sp_obtenerpersonalindividual(:dni_personal)"
        $instancia =$this->db->prepare($query);
        $instancia->bindParam(":dni_personal", $dni_personal);
        $instancia->execute();

        if ($instancia) {
            return $instancia->fetchALL(PDO::FETCH_CLASS,"personal")[0];
        } else {
            echo "Hubo un error";
        }

    } catch (Exception $ex) {
        echo "Hubo el siguiente fallo";
        echo var_dump($ex);
        return null;
    }
}
```

Fase 10: 20 Función para obtener un solo personal

```sql
--procedimiento para modificar solo una persona
CREATE OR REPLACE FUNCTION sp_obtenerpersonalindividual(
    in _dni character varying
    )
    RETURNS SETOF tpersonal
    LANGUAGE 'plpgsql'
AS $BODY$
    BEGIN
        RETURN QUERY SELECT * FROM tpersonal where dni_personal=_dni;
    END;
$BODY$;
```

Fase 10: 21 Procedimiento para obtener uno solo de personal

Después ya al haber obtenido al miembro del personal y haber modificado los datos procedemos a darle a editar y se ejecuta esta función que llama al siguiente procedimiento

```php
1 reference | 0 overrides
function editarPersona() {
    try {

        $modificarPersona = "CALL sp_editarpersona($this)";
        $querymodificar = $this->db->query($modificarPersona);
        header("Location:/proyectoBasePHP/vista/personal/listarpersonal.php");

    } catch (Exception $e) {
        throw new Exception("Hubo un error al modificar la persona: ".$e->getMessage());
    }
}
```

Fase 10: 22 Función de editar personal

```sql
--procedimiento para editar persona
CREATE PROCEDURE sp_editarpersona (
    in persona tipo_personal
    )
LANGUAGE plpgsql
AS $BODY$
BEGIN
    DELETE FROM tadministrativo WHERE dni_personal = persona.dni_personal;
    DELETE FROM tgerente WHERE dni_personal = persona.dni_personal;
    DELETE FROM tauxiliar WHERE dni_personal = persona.dni_personal;
    DELETE FROM tmantenimiento WHERE dni_personal = persona.dni_personal;
    DELETE FROM ttecnico WHERE dni_personal = persona.dni_personal;

    UPDATE tpersonal SET nombre_per = persona.nombre_per,
            apellido1_per = persona.apellido1_per,
            apellido2_per = persona.apellido2_per,
            email_per = persona.email_per,
            telefono_per = persona.telefono_per,
            cargo = persona.cargo
            estado=persona.estado
    WHERE dni_personal = persona.dni_personal;

    IF (persona.cargo  = 'Administrativo') THEN
        INSERT INTO tadministrativo (Nombre_adm, dni_personal)
        VALUES (persona.nombre_per, persona.dni_personal);

    ELSEIF (persona.cargo  = 'Gerente') THEN
        INSERT INTO tgerente (Nombre_ger, dni_personal)
        VALUES (persona.nombre_per, persona.dni_personal);

    ELSEIF (persona.cargo  = 'Auxiliar') THEN
        INSERT INTO tauxiliar (dni_personal, Nombre_aux)
        VALUES (persona.dni_personal, persona.nombre_per);

    ELSEIF (persona.cargo  = 'Mantenimiento') THEN
        INSERT INTO tmantenimiento (dni_personal, Nombre_man)
        VALUES (persona.dni_personal, persona.nombre_per);

    ELSEIF (persona.cargo  = 'Tecnico') THEN
        INSERT INTO ttecnico (dni_personal, Nombre_tec)
        VALUES (persona.dni_personal, persona.nombre_per);

    END IF;
END
$BODY$;
```

Fase 10: 23 Procedimiento para editar persona

También como solo pueden dar de alta a un cliente los administrativos o auxiliares tengo la función de obtener personal alta y el procedimiento para ello

```
yectoBasePHP > procedimientos > postgre > personal > obtenerpersonalalta.sql
1   --procedimiento para listar el personal
2   CREATE OR REPLACE FUNCTION sp_obtenerpersonalalta(
3       )
4       RETURNS SETOF tpersonal
5       LANGUAGE 'plpgsql'
6   AS $BODY$
7       BEGIN
8           RETURN QUERY SELECT * FROM tpersonal where cargo='Administrativo' OR cargo='Auxiliar' AND estado='Activo';
9       END;
10  $BODY$;
```

Fase 10: 24 procedimiento para obtener personal que da de alta

```
1 reference | 0 overrides | Codeium: Refactor | Explain | Generate Function Comment
function obtenerPersonalAlta(){
    try {
        $query = "Select * from sp_obtenerpersonalalta()";
        $listapersonal = $this->db->prepare($query);
        $listapersonal->execute();
    } catch (Exception $ex) {
        echo "Te equivocaste en: ".$ex->getMessage();
    }
    return $listapersonal->fetchALL(PDO::FETCH_CLASS,"personal");
}
```

Fase 10: 25 Función para mostrar personal que da de alta

Y después terminamos con la tabla de historicopersonal para cuando borremos a alguien

```
function verHistorico(){
    try {
        $querySelect = "select * from sp_obtenerpersonalhistorico()";
        $listapersonal=$this->db->prepare($querySelect);
        $listapersonal->execute();

    } catch (Exception $ex) {
        echo "Te equivocaste en: ".$ex->getMessage();
    }
    return $listapersonal->fetchALL(PDO::FETCH_CLASS,"personal");
}
```

Fase 10: 26 Función para ver histórico

```
CREATE OR REPLACE FUNCTION sp_obtenerPersonalHistorico(
    )
    RETURNS SETOF historicopersonal
    LANGUAGE 'plpgsql'
AS $BODY$
    BEGIN
        RETURN QUERY SELECT * FROM historicopersonal;
    END;
$BODY$;
```

Fase 10: 27 Procedimiento para ver histórico

Que para rellenarlo de forma automática usa el siguiente trigger

```
1  --trigger para llenar la tabla historico personal cuando se elimine algo de tpersonal
2  CREATE OR REPLACE FUNCTION trigger_delPersonal()
3  RETURNS TRIGGER
4  LANGUAGE plpgsql
5  AS $BODY$
6  BEGIN
7     INSERT INTO historicoPersonal(DNI_personal, estado)
8     VALUES (OLD.DNI_personal, 'Inactivo');
9
10    RETURN OLD;
11 END;
12 $BODY$;
13
14 CREATE TRIGGER personal_eliminado
15 AFTER DELETE ON tpersonal
16 FOR EACH ROW
17 EXECUTE FUNCTION trigger_delPersonal();
```

Fase 10: 28 Trigger para poblar la tabla histórico

Y con esto tendríamos la parte de personal, con cliente viene a ser prácticamente igual

Primero obtenemos el listado de cliente

```php
function obtenerCliente() {

    try {
        $querySelect = "SELECT * FROM public.sp_obtenercliente()";
        $listacliente=$this->db->prepare($querySelect);
        $listacliente->execute();

        if($listacliente){
            return $listacliente->fetchAll(PDO::FETCH_CLASS,'Cliente');
        }else{
            echo "no se pudo obtener listado";
        }

    } catch (Exception $e) {
        throw new Exception("Hubo un error: " . $e->getMessage());
    }
}
```

Fase 10: 29 Función para mostrar clientes

```sql
CREATE OR REPLACE FUNCTION sp_obtenercliente(
    )
    RETURNS SETOF tcliente
    LANGUAGE 'plpgsql'
AS $BODY$
    BEGIN
        RETURN QUERY SELECT * FROM tcliente;
    END;
$BODY$;
```

Fase 10: 30 Procedimiento para obtener los clientes

Después tenemos la parte de añadir a un cliente

```php
function addCliente(){
    try {
        $queryCliente = $this->db->prepare("CALL sp_agregar_clientes(?, ?, ?, ?, ?, ?, ?)");
        $queryCliente->bindParam(1, $this->dni_cl);
        $queryCliente->bindParam(2, $this->nombre_cl);
        $queryCliente->bindParam(3, $this->apellido1_cl);
        $queryCliente->bindParam(4, $this->apellido2_cl);
        $queryCliente->bindParam(5, $this->email_cl);
        $queryCliente->bindParam(6, $this->telefono_cl);
        $queryCliente->bindParam(7, $this->dni_personal);

        $queryCliente->execute();

        $mensaje = "Se ha introducido correctamente al cliente ".$this->dni_cl;

        return $mensaje;
    } catch (Exception $ex) {
        echo "Hay un error en ".$ex->getMessage();
        $Fallido = "No se ha podido insertar el cliente.";
        return $Fallido;
    }
}
```

Fase 10: 31 Función para añadir clientes

```sql
CREATE OR REPLACE PROCEDURE public.sp_agregar_clientes(
    _DNI_cl character varying(20),
    _nombre_cl character varying(50),
    _apellido1_cl character varying(50),
    _apellido2_cl character varying(50),
    _email_cl character varying(50),
    _telefono_cl INTEGER,
    _DNI_personal character varying(20)
)
LANGUAGE plpgsql
AS $BODY$
BEGIN
    INSERT INTO TCLIENTE (dni_cl,nombre_cl,apellido1_cl,apellido2_cl,email_cl,telefono_cl,dni_per
    values(_DNI_cl,_nombre_cl,_apellido1_cl,_apellido2_cl,_email_cl,_telefono_cl,_DNI_personal);
END
$BODY$
```

Fase 10: 32 Procedimiento para añadir clientes

La parte de borrar cliente

```php
1 reference | 0 overrides | Codeium: Refactor | Explain | Generate Function Comment
function delCliente($id_usuario){
    try{
     $query = "CALL sp_delcliente(:id_usuario)";
     $queryCliente = $this->db->prepare($query);
     $queryCliente->bindParam(":id_usuario", $id_usuario);
     $queryCliente->execute();
     $mensaje="Se ha eliminado la persona con id ".$id_usuario." correctamente";
     return $mensaje;

} catch (Exception $e) {
     $Fallido ="No se ha podido eliminar la persona".$e;
     return $Fallido;
}
}
```

Fase 10: 33 Función para eliminar clientes

```sql
--procedimiento para eliminar cliente
create procedure sp_delcliente(
    _id_usuario integer
)
LANGUAGE plpgsql
AS $BODY$
BEGIN
    update tcliente set dni_personal=null where id_usuario=_id_usuario;
    DELETE FROM tcliente WHERE id_usuario = _id_usuario;
end
$BODY$
```

Fase 10: 34 Procedimiento para eliminar clientes

Al igual que para editar personal, tenemos que crear una función y procedimiento para obtener solo el cliente a editar

```php
1 reference | 0 overrides | Codeium: Refactor | Explain | Generate Function Comment
function obtenerClienteEdit($id_usuario){
    try{
    $query = "SELECT * FROM sp_obtenerclienteindividual(:id_usuario)";
    $clientes = $this->db->prepare($query);
    $clientes->bindParam(":id_usuario",$id_usuario);
    $clientes->execute();

    if ($clientes) {
        return $clientes->fetch();
    } else {
        echo "Hubo un error";
    }
}
catch (Exception $e) {
    return "Hubo un error: " . $e->getMessage();
}
}
```

Fase 10: 35 Función para obtener solo un cliente

```sql
--procedimiento para modificar solo un cliente
CREATE OR REPLACE FUNCTION sp_obtenerclienteindividual(
    in _dni character varying
    )
    RETURNS SETOF tcliente
    LANGUAGE 'plpgsql'
AS $BODY$
    BEGIN
        RETURN QUERY SELECT * FROM tcliente where dni_cl=_dni;
    END;
$BODY$;
```

Fase 10: 36 Procedimiento para obtener solo un cliente

Y ya después de obtener al cliente seria editarlo

```php
1 reference | 0 overrides | Codeium: Refactor | Explain | Generate Function Comment
public function editCliente($id_usuario, $nombre_cl, $apellido1_cl, $apellido
    try{
     $query = "CALL sp_editarcliente(:id_usuario, :nombre_cl, :apellido1_cl, :
     $queryCliente = $this->db->prepare($query);
     $queryCliente->bindParam(":id_usuario", $id_usuario);
     $queryCliente->bindParam(":nombre_cl", $nombre_cl);
     $queryCliente->bindParam(":apellido1_cl", $apellido1_cl);
     $queryCliente->bindParam(":apellido2_cl", $apellido2_cl);
     $queryCliente->bindParam(":email_cl", $email_cl);
     $queryCliente->bindParam(":telefono_cl", $telefono_cl);
     $queryCliente->execute();
     $mensaje="se ha editado el cliente";
     return $mensaje;
    }catch (Exception $e) {
     throw new Exception("Hubo un error al modificar el cliente: ".$e->getMess
}
}
```

Fase 10: 37 Función para editar cliente

```sql
--procedimiento para editar cliente
CREATE OR REPLACE PROCEDURE sp_editarcliente(
    _id_usuario integer,
    _nombre_cl character varying(50),
    _apellido1_cl character varying(50),
    _apellido2_cl character varying(50),
    _email_cl character varying(50),
    _telefono_cl INTEGER
)
LANGUAGE plpgsql
AS $BODY$
BEGIN
    UPDATE tcliente set nombre_cl=_nombre_cl,apellido1_cl=_apellido1_cl,apellido2
    WHERE id_usuario =_id_usuario;
END
$BODY$;
```

Fase 10: 38 Procedimiento para editar cliente

Nuevamente para ver el histórico de clientes tenemos la función y su correspondiente procedimiento

```
1 reference | 0 overrides | Codeium: Refactor | Explain | Generate Function Comment
function verHistorico(){
    try {
        $querySelect = "SELECT * FROM  sp_obtenerclienteHistorico()";
        $listapersonal=$this->db->prepare($querySelect);
        $listapersonal->execute();

    } catch (Exception $ex) {
        echo "Te equivocaste en: ".$ex->getMessage();
    }
    return $listapersonal->fetchALL(PDO::FETCH_CLASS,"Cliente");
}
```

Fase 10: 39 Función para listar histórico

```
1  --procedimiento para obtener el historico de cliente
2  CREATE OR REPLACE FUNCTION sp_obtenerclienteHistorico(
3      )
4      RETURNS SETOF historicocliente
5      LANGUAGE 'plpgsql'
6  AS $BODY$
7      BEGIN
8          RETURN QUERY SELECT * FROM historicocliente;
9      END;
10 $BODY$;
```

Fase 10: 40 Procedimiento para obtener histórico

Y el trigger para poblarlo

```
--trigger para llenar la tabla historico cliente cuand
CREATE OR REPLACE FUNCTION trigger_delcliente()
RETURNS TRIGGER
LANGUAGE plpgsql
AS $BODY$
BEGIN
   INSERT INTO historicocliente(dni_cl, estado)
   VALUES (OLD.dni_cl, 'Inactivo');

   RETURN OLD;
END;
$BODY$;

CREATE TRIGGER cliente_eliminado
AFTER DELETE ON tcliente
FOR EACH ROW
EXECUTE FUNCTION trigger_delcliente();
```

Fase 10: 41 Trigger para poblar el histórico

16- Cambios Fase 9 a Fase 10

El principal cambio ha sido en la base de datos, ya que al cambiar de base de datos de MySQL o MariaDB a PostgreSQL hubo que ajustar la base de datos para meter los enumerados y el tipo usado para el ejercicio.

17- Índice de tablas

18- Índice de relaciones

19- Índice tablas MYSQL

20- Índice tablas Físicas

21- Índice de cambios

22- Índice de inserciones

23- Índice de modificaciones

24- Índice de eliminaciones

25- Índice de transacciones

26-Índice de consultas

27-índice fase 8

28- índice fase 9

29- Índice Fase 10