



GRUPO 3

Artefacto 6

Rubén Adarve

Walter Huatay

Adrián Marín

Ismael Ruiz

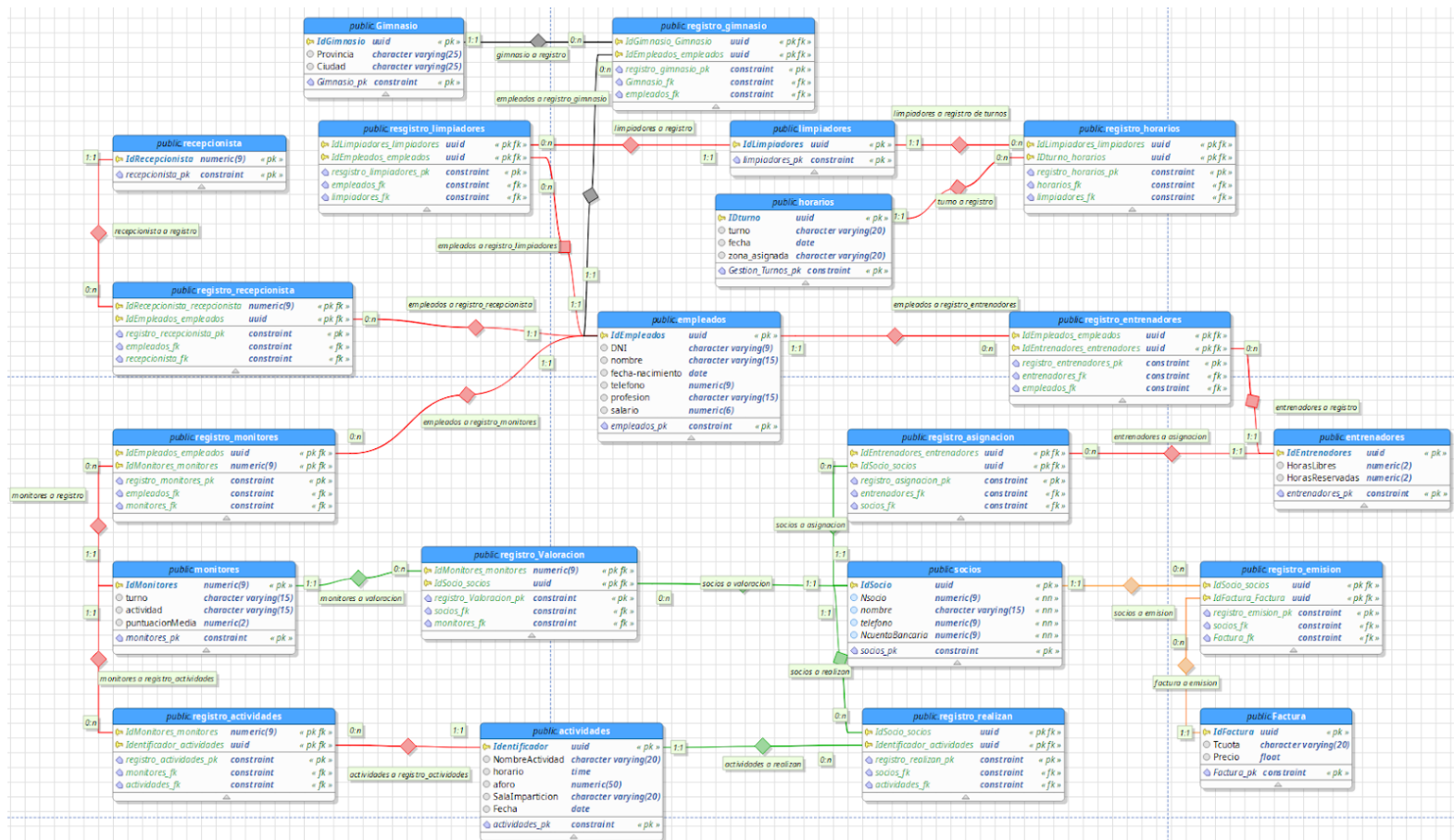
Ivan Sánchez



6. Modelo de datos

6.1. Diagramas de tablas

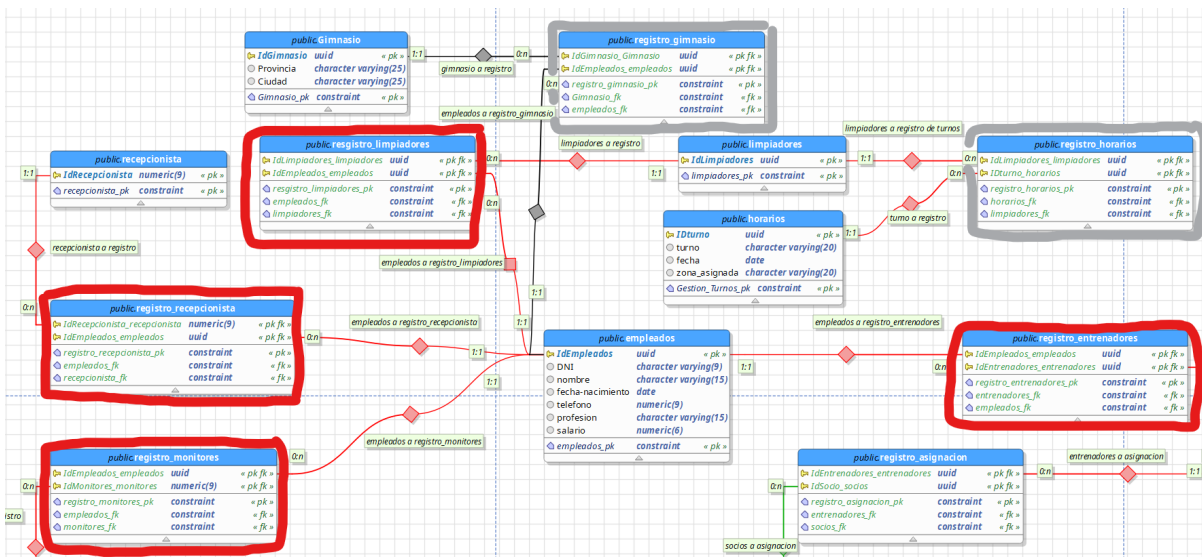
<Copia de la imagen del diagrama de tablas realizado en Toad. El diagrama se generará automáticamente a partir del diagrama E/R.>



6.1.1. Explicación del diagrama

<Explicación de los motivos que han llevado a tomar cada una de las decisiones de la conversión: relaciones de herencia , 1:1 y desnormalización.>

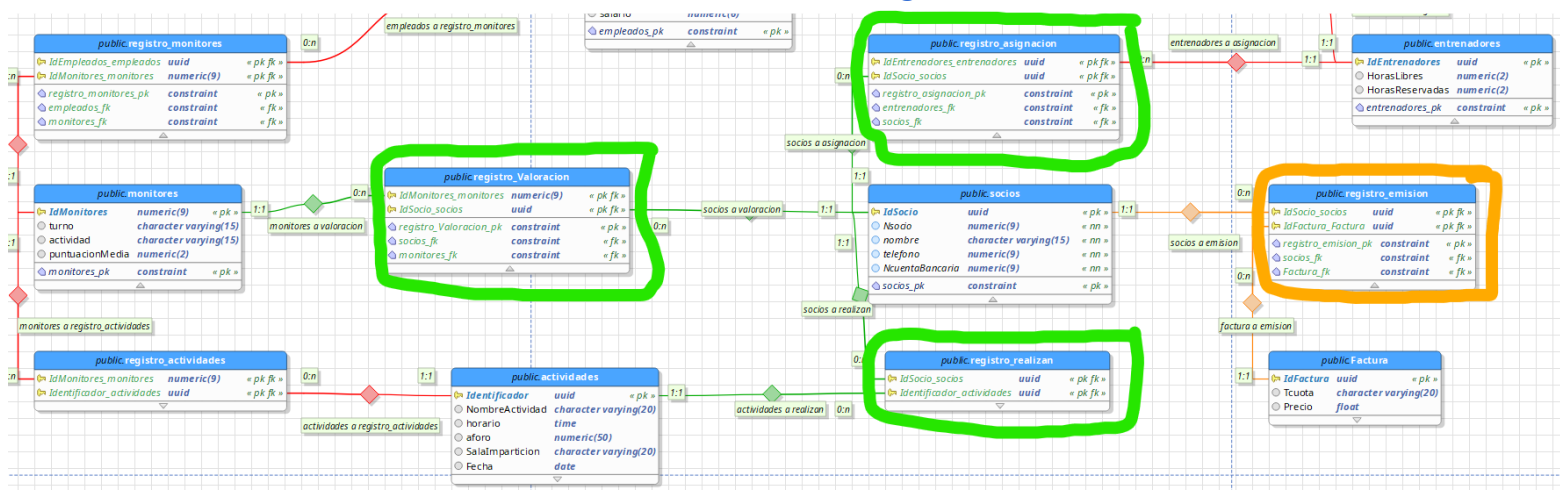
Nosotros desde un principio en el artefacto 4 ya convertimos las relaciones a tablas. Básicamente sabíamos que íbamos a tener que trabajar con postgres, entonces convertimos a tablas las relaciones porque sino más adelante a la hora de trabajar con postgres no pueden relacionarse unas con otras. Comentaremos a continuación las tablas convertidas:



Desde la tabla empleados podemos observar estas 4 tablas señaladas en rojo, son las que relacionan empleados con el puesto de trabajo y sus respectivas características (limpiadores, recepcionistas, monitores y entrenadores).

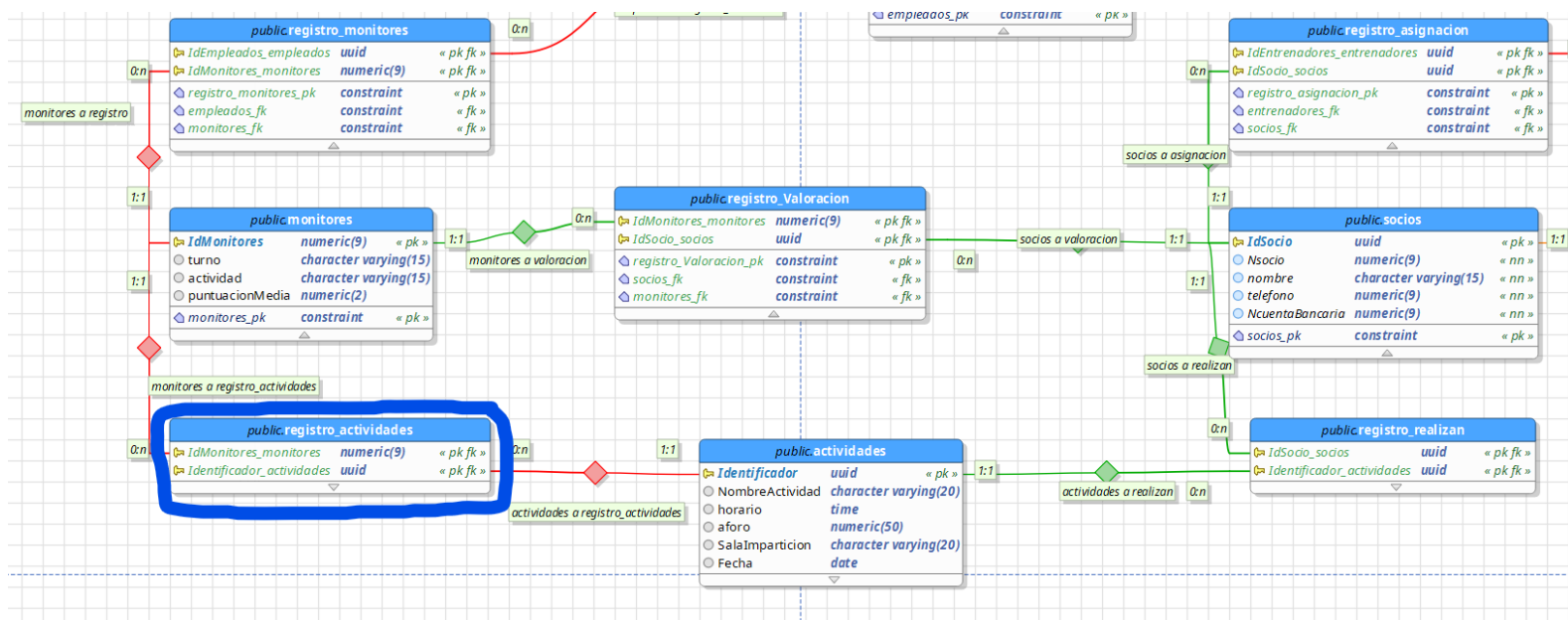
Las tablas que estan de color gris son las nuevas relaciones que hemos implementado, la primera **registro_gimnasio** relaciona los empleados con los gimnasios que hay guardados en la base de datos para saber en que gimnasio estan trabajando.

la segunda **registro_horarios** con ella podemos darle más variabilidad al horario que se le asigna a los limpiadores, ya que relaciona limpiadores con horarios.



las que estan redondeadas de color verde son las relaciones que unen, las actividades que realizan los socios, la asignacion de los entrenadores personales y la valoracion que se le pone a un monitor.

En cuanto a la tabla naranja relaciona la tabla socios y la factura. Básicamente asigna la factura para el cliente.



Por último, la tabla azul relaciona las actividades que tiene el gimnasio y se le asigna a los monitores dichas actividades.

6.1.2. Script BD

<Copia del contenido del fichero obtenido tras la generación automática en Toad del script de BD. Se utiliza la opción Model / Generate DDL script / Run.>

```
-- Database generated with pgModeler (PostgreSQL Database  
Modeler).
```

```
-- pgModeler version: 0.9.1
```

```
-- PostgreSQL version: 10.0
```

```
-- Project Site: pgmodeler.io
```

```
-- Model Author: ---
```

```
-- Database creation must be done outside a multicommand file.
```

```
-- These commands were put in this file only as a convenience.
```

```
-- -- object: new__database | type: DATABASE --
```

```
-- -- DROP DATABASE IF EXISTS new__database;
```

```
-- CREATE DATABASE new__database;
```

```
-- -- ddl-end --
```

```
--
```

```
-- object: public.empleados | type: TABLE --
```

```
-- DROP TABLE IF EXISTS public.empleados CASCADE;
```

```
CREATE TABLE public.empleados(
```

```
    "IdEmpleados" uuid NOT NULL,
```

```
"DNI" character varying(9),
nombre character varying(15),
"fecha-nacimiento" date,
telefono numeric(9),
profesion character varying(15),
salario numeric(6),
CONSTRAINT empleados_pk PRIMARY KEY ("IdEmpleados")

);
-- ddl-end --

ALTER TABLE public.empleados OWNER TO postgres;
-- ddl-end --

-- object: public.monitores | type: TABLE --
-- DROP TABLE IF EXISTS public.monitores CASCADE;
CREATE TABLE public.monitores(
    "IdMonitores" numeric(9) NOT NULL,
    turno character varying(15),
    actividad character varying(15),
    "puntuacionMedia" numeric(2),
    CONSTRAINT monitores_pk PRIMARY KEY ("IdMonitores")

);
```

```
-- ddl-end --
```

```
ALTER TABLE public.monitores OWNER TO postgres;
```

```
-- ddl-end --
```

```
-- object: public.registro_monitores | type: TABLE --
```

```
-- DROP TABLE IF EXISTS public.registro_monitores CASCADE;
```

```
CREATE TABLE public.registro_monitores(
```

```
    "IdEmpleados_empleados" uuid NOT NULL,
```

```
    "IdMonitores_monitores" numeric(9) NOT NULL,
```

```
    CONSTRAINT registro_monitores_pk PRIMARY KEY  
("IdEmpleados_empleados", "IdMonitores_monitores")
```

```
);
```

```
-- ddl-end --
```

```
-- object: empleados_fk | type: CONSTRAINT --
```

```
-- ALTER TABLE public.registro_monitores DROP CONSTRAINT IF  
EXISTS empleados_fk CASCADE;
```

```
ALTER TABLE public.registro_monitores ADD CONSTRAINT  
empleados_fk FOREIGN KEY ("IdEmpleados_empleados")
```

```
REFERENCES public.empleados ("IdEmpleados") MATCH FULL
```

```
ON DELETE CASCADE ON UPDATE CASCADE;
```

```
-- ddl-end --
```

```
-- object: monitores_fk | type: CONSTRAINT --

-- ALTER TABLE public.registro_monitores DROP CONSTRAINT IF
EXISTS monitores_fk CASCADE;

ALTER TABLE public.registro_monitores ADD CONSTRAINT
monitores_fk FOREIGN KEY ("IdMonitores_monitores")
REFERENCES public.monitores ("IdMonitores") MATCH FULL
ON DELETE CASCADE ON UPDATE CASCADE;

-- ddl-end --


-- object: public.limpiadores | type: TABLE --

-- DROP TABLE IF EXISTS public.limpiadores CASCADE;

CREATE TABLE public.limpiadores(
    "IdLimpiadores" uuid NOT NULL,
    CONSTRAINT limpiadores_pk PRIMARY KEY ("IdLimpiadores")

);

-- ddl-end --

ALTER TABLE public.limpiadores OWNER TO postgres;

-- ddl-end --


-- object: public.resregistro_limpiadores | type: TABLE --

-- DROP TABLE IF EXISTS public.resregistro_limpiadores CASCADE;

CREATE TABLE public.resregistro_limpiadores(
    "IdLimpiadores_limpiadores" uuid NOT NULL,
```



```
"IdEmpleados_empleados" uuid NOT NULL,  
  
CONSTRAINT resgistro_limpiadores_pk PRIMARY KEY  
("IdEmpleados_empleados","IdLimpiadores_limpiadores")  
  
);  
  
-- ddl-end --  
  
-- object: empleados_fk | type: CONSTRAINT --  
  
-- ALTER TABLE public.resgistro_limpiadores DROP CONSTRAINT IF  
EXISTS empleados_fk CASCADE;  
  
ALTER TABLE public.resgistro_limpiadores ADD CONSTRAINT  
empleados_fk FOREIGN KEY ("IdEmpleados_empleados")  
REFERENCES public.empleados ("IdEmpleados") MATCH FULL  
ON DELETE CASCADE ON UPDATE CASCADE;  
  
-- ddl-end --  
  
-- object: limpiadores_fk | type: CONSTRAINT --  
  
-- ALTER TABLE public.resgistro_limpiadores DROP CONSTRAINT IF  
EXISTS limpiadores_fk CASCADE;  
  
ALTER TABLE public.resgistro_limpiadores ADD CONSTRAINT  
limpiadores_fk FOREIGN KEY ("IdLimpiadores_limpiadores")  
REFERENCES public.limpiadores ("IdLimpiadores") MATCH FULL  
ON DELETE CASCADE ON UPDATE CASCADE;  
  
-- ddl-end --
```

```
-- object: public.recepcionista | type: TABLE --  
  
-- DROP TABLE IF EXISTS public.recepcionista CASCADE;  
  
CREATE TABLE public.recepcionista(  
    "IdRecepcionista" numeric(9) NOT NULL,  
    CONSTRAINT recepcionista_pk PRIMARY KEY  
    ("IdRecepcionista")  
  
);  
  
-- ddl-end --  
  
ALTER TABLE public.recepcionista OWNER TO postgres;  
  
-- ddl-end --  
  
-- object: public.registro_recepcionista | type: TABLE --  
  
-- DROP TABLE IF EXISTS public.registro_recepcionista CASCADE;  
  
CREATE TABLE public.registro_recepcionista(  
    "IdRecepcionista_recepcionista" numeric(9) NOT NULL,  
    "IdEmpleados_empleados" uuid NOT NULL,  
    CONSTRAINT registro_recepcionista_pk PRIMARY KEY  
    ("IdEmpleados_empleados", "IdRecepcionista_recepcionista")  
  
);  
  
-- ddl-end --  
  
-- object: empleados_fk | type: CONSTRAINT --
```

```
-- ALTER TABLE public.registro_recepcionista DROP CONSTRAINT IF  
EXISTS empleados_fk CASCADE;
```

```
ALTER TABLE public.registro_recepcionista ADD CONSTRAINT  
empleados_fk FOREIGN KEY ("IdEmpleados_empleados")
```

```
REFERENCES public.empleados ("IdEmpleados") MATCH FULL
```

```
ON DELETE CASCADE ON UPDATE CASCADE;
```

```
-- ddl-end --
```

```
-- object: recepcionista_fk | type: CONSTRAINT --
```

```
-- ALTER TABLE public.registro_recepcionista DROP CONSTRAINT IF  
EXISTS recepcionista_fk CASCADE;
```

```
ALTER TABLE public.registro_recepcionista ADD CONSTRAINT  
recepcionista_fk FOREIGN KEY ("IdRecepcionista_recepcionista")
```

```
REFERENCES public.recepcionista ("IdRecepcionista") MATCH FULL
```

```
ON DELETE CASCADE ON UPDATE CASCADE;
```

```
-- ddl-end --
```

```
-- object: public.entrenadores | type: TABLE --
```

```
-- DROP TABLE IF EXISTS public.entrenadores CASCADE;
```

```
CREATE TABLE public.entrenadores(  
    "IdEntrenadores" uuid NOT NULL,
```

```
    "HorasLibres" numeric(2),
```

```
    "HorasReservadas" numeric(2),
```

```
    CONSTRAINT entrenadores_pk PRIMARY KEY  
("IdEntrenadores")
```

```
);
```

```
-- ddl-end --
```

```
ALTER TABLE public.entrenadores OWNER TO postgres;
```

```
-- ddl-end --
```

```
-- object: public.registro__entrenadores | type: TABLE --
```

```
-- DROP TABLE IF EXISTS public.registro__entrenadores CASCADE;
```

```
CREATE TABLE public.registro__entrenadores(
```

```
    "IdEmpleados__empleados" uuid NOT NULL,
```

```
    "IdEntrenadores__entrenadores" uuid NOT NULL,
```

```
    CONSTRAINT registro__entrenadores_pk PRIMARY KEY  
("IdEntrenadores__entrenadores", "IdEmpleados__empleados")
```

```
);
```

```
-- ddl-end --
```

```
-- object: entrenadores_fk | type: CONSTRAINT --
```

```
-- ALTER TABLE public.registro__entrenadores DROP CONSTRAINT IF  
EXISTS entrenadores_fk CASCADE;
```

```
ALTER TABLE public.registro__entrenadores ADD CONSTRAINT  
entrenadores_fk FOREIGN KEY ("IdEntrenadores__entrenadores")
```

```
REFERENCES public.entrenadores ("IdEntrenadores") MATCH FULL  
ON DELETE CASCADE ON UPDATE CASCADE;
```

```
-- ddl-end --
```

```
-- object: empleados_fk | type: CONSTRAINT --
```

```
-- ALTER TABLE public.registro_entrenadores DROP CONSTRAINT IF  
EXISTS empleados_fk CASCADE;
```

```
ALTER TABLE public.registro_entrenadores ADD CONSTRAINT  
empleados_fk FOREIGN KEY ("IdEmpleados_empleados")
```

```
REFERENCES public.empleados ("IdEmpleados") MATCH FULL  
ON DELETE CASCADE ON UPDATE CASCADE;
```

```
-- ddl-end --
```

```
-- object: public.socios | type: TABLE --
```

```
-- DROP TABLE IF EXISTS public.socios CASCADE;
```

```
CREATE TABLE public.socios(  
    "IdSocio" uuid NOT NULL,
```

```
    "Nsocio" numeric(9) NOT NULL,
```

```
    "nombre" character varying(15) NOT NULL,
```

```
    "telefono" numeric(9) NOT NULL,
```

```
    "telefono" numeric(9) NOT NULL,
```

```
    "NcuentaBancaria" numeric(9) NOT NULL,
```

```
    CONSTRAINT socios_pk PRIMARY KEY ("IdSocio")
```

```
);
```

```
-- ddl-end --
```

```
ALTER TABLE public.socios OWNER TO postgres;
```

```
-- ddl-end --
```

```
-- object: public.actividades | type: TABLE --
```

```
-- DROP TABLE IF EXISTS public.actividades CASCADE;
```

```
CREATE TABLE public.actividades(
```

```
    "Identificador" uuid NOT NULL,
```

```
    "NombreActividad" character varying(20),
```

```
    horario time,
```

```
    aforo numeric(50),
```

```
    "SalaImparticion" character varying(20),
```

```
    "Fecha" date,
```

```
    CONSTRAINT actividades__pk PRIMARY KEY ("Identificador")
```

```
);
```

```
-- ddl-end --
```

```
ALTER TABLE public.actividades OWNER TO postgres;
```

```
-- ddl-end --
```

```
-- object: public."Factura" | type: TABLE --
```

```
-- DROP TABLE IF EXISTS public."Factura" CASCADE;
```

```
CREATE TABLE public."Factura" (
```

```
    "IdFactura" uuid NOT NULL,
```

```
    "Tcuota" character varying(20),
```

```
"Precio" float,  
  
CONSTRAINT "Factura_pk" PRIMARY KEY ("IdFactura")  
  
);  
  
-- ddl-end --  
  
ALTER TABLE public."Factura" OWNER TO postgres;  
  
-- ddl-end --  
  
-- object: public.registro_actividades | type: TABLE --  
  
-- DROP TABLE IF EXISTS public.registro_actividades CASCADE;  
CREATE TABLE public.registro_actividades(  
    "IdMonitores_monitores" numeric(9) NOT NULL,  
    "Identificador_actividades" uuid NOT NULL,  
    CONSTRAINT registro_actividades_pk PRIMARY KEY  
    ("IdMonitores_monitores","Identificador_actividades")  
  
);  
  
-- ddl-end --  
  
-- object: monitores_fk | type: CONSTRAINT --  
  
-- ALTER TABLE public.registro_actividades DROP CONSTRAINT IF  
EXISTS monitores_fk CASCADE;  
ALTER TABLE public.registro_actividades ADD CONSTRAINT  
monitores_fk FOREIGN KEY ("IdMonitores_monitores")
```

```
REFERENCES public.monitores ("IdMonitores") MATCH FULL
```

```
ON DELETE CASCADE ON UPDATE CASCADE;
```

```
-- ddl-end --
```

```
-- object: actividades_fk | type: CONSTRAINT --
```

```
-- ALTER TABLE public.registro_actividades DROP CONSTRAINT IF  
EXISTS actividades_fk CASCADE;
```

```
ALTER TABLE public.registro_actividades ADD CONSTRAINT  
actividades_fk FOREIGN KEY ("Identificador_actividades")
```

```
REFERENCES public.actividades ("Identificador") MATCH FULL
```

```
ON DELETE CASCADE ON UPDATE CASCADE;
```

```
-- ddl-end --
```

```
-- object: public."registro_valoracion" | type: TABLE --
```

```
-- DROP TABLE IF EXISTS public."registro_valoracion" CASCADE;
```

```
CREATE TABLE public."registro_valoracion"(  
    "IdMonitores_monitores" numeric(9) NOT NULL,
```

```
    "IdSocio_socios" uuid NOT NULL,
```

```
    CONSTRAINT "registro_valoracion_pk" PRIMARY KEY  
("IdSocio_socios", "IdMonitores_monitores")
```

```
);
```

```
-- ddl-end --
```



```
-- object: socios_fk | type: CONSTRAINT --

-- ALTER TABLE public."registro_Valoracion" DROP CONSTRAINT IF
EXISTS socios_fk CASCADE;

ALTER TABLE public."registro_Valoracion" ADD CONSTRAINT
socios_fk FOREIGN KEY ("IdSocio_socios")
REFERENCES public.socios ("IdSocio") MATCH FULL
ON DELETE CASCADE ON UPDATE CASCADE;

-- ddl-end --


-- object: monitores_fk | type: CONSTRAINT --

-- ALTER TABLE public."registro_Valoracion" DROP CONSTRAINT IF
EXISTS monitores_fk CASCADE;

ALTER TABLE public."registro_Valoracion" ADD CONSTRAINT
monitores_fk FOREIGN KEY ("IdMonitores_monitores")
REFERENCES public.monitores ("IdMonitores") MATCH FULL
ON DELETE CASCADE ON UPDATE CASCADE;

-- ddl-end --


-- object: public.registro_emision | type: TABLE --

-- DROP TABLE IF EXISTS public.registro_emision CASCADE;

CREATE TABLE public.registro_emision(

    "IdSocio_socios" uuid NOT NULL,

    "IdFactura_Factura" uuid NOT NULL,

    CONSTRAINT registro_emision_pk PRIMARY KEY
("IdSocio_socios","IdFactura_Factura")
```

```
);
```

```
-- ddl-end --
```

```
-- object: socios_fk | type: CONSTRAINT --
```

```
-- ALTER TABLE public.registro_emision DROP CONSTRAINT IF  
EXISTS socios_fk CASCADE;
```

```
ALTER TABLE public.registro_emision ADD CONSTRAINT socios_fk  
FOREIGN KEY ("IdSocio_socios")
```

```
REFERENCES public.socios ("IdSocio") MATCH FULL
```

```
ON DELETE CASCADE ON UPDATE CASCADE;
```

```
-- ddl-end --
```

```
-- object: "Factura_fk" | type: CONSTRAINT --
```

```
-- ALTER TABLE public.registro_emision DROP CONSTRAINT IF  
EXISTS "Factura_fk" CASCADE;
```

```
ALTER TABLE public.registro_emision ADD CONSTRAINT  
"Factura_fk" FOREIGN KEY ("IdFactura_Factura")
```

```
REFERENCES public."Factura" ("IdFactura") MATCH FULL
```

```
ON DELETE CASCADE ON UPDATE CASCADE;
```

```
-- ddl-end --
```

```
-- object: public.registro_asignacion | type: TABLE --
```

```
-- DROP TABLE IF EXISTS public.registro_asignacion CASCADE;
```

```
CREATE TABLE public.registro_asignacion(
```

```
"IdEntrenadores_entrenadores" uuid NOT NULL,  
  
"IdSocio_socios" uuid NOT NULL,  
  
CONSTRAINT registro_asignacion_pk PRIMARY KEY  
("IdEntrenadores_entrenadores", "IdSocio_socios")  
  
);  
  
-- ddl-end --  
  
-- object: entrenadores_fk | type: CONSTRAINT --  
  
-- ALTER TABLE public.registro_asignacion DROP CONSTRAINT IF  
EXISTS entrenadores_fk CASCADE;  
  
ALTER TABLE public.registro_asignacion ADD CONSTRAINT  
entrenadores_fk FOREIGN KEY ("IdEntrenadores_entrenadores")  
REFERENCES public.entrenadores ("IdEntrenadores") MATCH FULL  
ON DELETE CASCADE ON UPDATE CASCADE;  
  
-- ddl-end --  
  
-- object: socios_fk | type: CONSTRAINT --  
  
-- ALTER TABLE public.registro_asignacion DROP CONSTRAINT IF  
EXISTS socios_fk CASCADE;  
  
ALTER TABLE public.registro_asignacion ADD CONSTRAINT  
socios_fk FOREIGN KEY ("IdSocio_socios")  
REFERENCES public.socios ("IdSocio") MATCH FULL  
ON DELETE CASCADE ON UPDATE CASCADE;  
  
-- ddl-end --
```

```
-- object: public.registro_realizan | type: TABLE --  
  
-- DROP TABLE IF EXISTS public.registro_realizan CASCADE;  
  
CREATE TABLE public.registro_realizan(  
    "IdSocio_socios" uuid NOT NULL,  
    "Identificador_actividades" uuid NOT NULL,  
    CONSTRAINT registro_realizan_pk PRIMARY KEY  
    ("IdSocio_socios","Identificador_actividades")  
  
);  
  
-- ddl-end --  
  
-- object: socios_fk | type: CONSTRAINT --  
  
-- ALTER TABLE public.registro_realizan DROP CONSTRAINT IF  
EXISTS socios_fk CASCADE;  
  
ALTER TABLE public.registro_realizan ADD CONSTRAINT socios_fk  
FOREIGN KEY ("IdSocio_socios")  
  
REFERENCES public.socios ("IdSocio") MATCH FULL  
  
ON DELETE CASCADE ON UPDATE CASCADE;  
  
-- ddl-end --  
  
-- object: actividades_fk | type: CONSTRAINT --  
  
-- ALTER TABLE public.registro_realizan DROP CONSTRAINT IF  
EXISTS actividades_fk CASCADE;
```

```
ALTER TABLE public.registro_realizan ADD CONSTRAINT
actividades_fk FOREIGN KEY ("Identificador_actividades")
REFERENCES public.actividades ("Identificador") MATCH FULL
ON DELETE CASCADE ON UPDATE CASCADE;
-- ddl-end --
```

```
-- object: public.horarios | type: TABLE --
-- DROP TABLE IF EXISTS public.horarios CASCADE;
CREATE TABLE public.horarios(
    "IDturno" uuid NOT NULL,
    turno character varying(20),
    fecha date,
    zona_asignada character varying(20),
    CONSTRAINT "Gestion_Turnos_pk" PRIMARY KEY
("IDturno")
);
-- ddl-end --
```

```
ALTER TABLE public.horarios OWNER TO postgres;
-- ddl-end --
```

```
-- object: public.registro_horarios | type: TABLE --
-- DROP TABLE IF EXISTS public.registro_horarios CASCADE;
CREATE TABLE public.registro_horarios(
```

```
"IdLimpiadores_limpiadores" uuid NOT NULL,  
"IDturno_horarios" uuid NOT NULL,  
CONSTRAINT registro_horarios_pk PRIMARY KEY  
("IDturno_horarios","IdLimpiadores_limpiadores")  
  
);  
  
-- ddl-end --  
  
-- object: horarios_fk | type: CONSTRAINT --  
  
-- ALTER TABLE public.registro_horarios DROP CONSTRAINT IF  
EXISTS horarios_fk CASCADE;  
  
ALTER TABLE public.registro_horarios ADD CONSTRAINT  
horarios_fk FOREIGN KEY ("IDturno_horarios")  
REFERENCES public.horarios ("IDturno") MATCH FULL  
ON DELETE CASCADE ON UPDATE CASCADE;  
  
-- ddl-end --  
  
-- object: limpiadores_fk | type: CONSTRAINT --  
  
-- ALTER TABLE public.registro_horarios DROP CONSTRAINT IF  
EXISTS limpiadores_fk CASCADE;  
  
ALTER TABLE public.registro_horarios ADD CONSTRAINT  
limpiadores_fk FOREIGN KEY ("IdLimpiadores_limpiadores")  
REFERENCES public.limpiadores ("IdLimpiadores") MATCH FULL  
ON DELETE CASCADE ON UPDATE CASCADE;  
  
-- ddl-end --
```

```
-- object: public."Gimnasio" | type: TABLE --  
  
-- DROP TABLE IF EXISTS public."Gimnasio" CASCADE;  
  
CREATE TABLE public."Gimnasio"(  
    "IdGimnasio" uuid NOT NULL,  
    "Provincia" character varying(25),  
    "Ciudad" character varying(25),  
    CONSTRAINT "Gimnasio_pk" PRIMARY KEY ("IdGimnasio")  
  
);  
  
-- ddl-end --  
  
ALTER TABLE public."Gimnasio" OWNER TO postgres;  
  
-- ddl-end --  
  
  
-- object: public.registro_gimnasio | type: TABLE --  
  
-- DROP TABLE IF EXISTS public.registro_gimnasio CASCADE;  
  
CREATE TABLE public.registro_gimnasio(  
    "IdGimnasio_Gimnasio" uuid NOT NULL,  
    "IdEmpleados_empleados" uuid NOT NULL,  
    CONSTRAINT registro_gimnasio_pk PRIMARY KEY  
("IdGimnasio_Gimnasio", "IdEmpleados_empleados")  
  
);  
  
-- ddl-end --
```

```
-- object: "Gimnasio_fk" | type: CONSTRAINT --  
  
-- ALTER TABLE public.registro_gimnasio DROP CONSTRAINT IF  
EXISTS "Gimnasio_fk" CASCADE;  
  
ALTER TABLE public.registro_gimnasio ADD CONSTRAINT  
"Gimnasio_fk" FOREIGN KEY ("IdGimnasio_Gimnasio")  
REFERENCES public."Gimnasio" ("IdGimnasio") MATCH FULL  
ON DELETE CASCADE ON UPDATE CASCADE;  
  
-- ddl-end --
```

```
-- object: empleados_fk | type: CONSTRAINT --  
  
-- ALTER TABLE public.registro_gimnasio DROP CONSTRAINT IF  
EXISTS empleados_fk CASCADE;  
  
ALTER TABLE public.registro_gimnasio ADD CONSTRAINT  
empleados_fk FOREIGN KEY ("IdEmpleados_empleados")  
REFERENCES public.empleados ("IdEmpleados") MATCH FULL  
ON DELETE CASCADE ON UPDATE CASCADE;  
  
-- ddl-end --
```


6.2. Otros elementos de almacenamiento

<Otros elementos archivados necesarios para la aplicación, por ejemplo ficheros de intercambio de información con otros sistemas o plantillas de correo electrónico.>.

6.2.1. Diseño y explicación elemento 1

<Diseño del elemento, es decir formato detallado de cabeceras, líneas, etc. y una explicación textual de su uso.>

Se mantendrán copias de las bases de datos originales (documentos excel) en una instalación externa de la infraestructura usada por la aplicación. Como medida de seguridad y copia de seguridad.

6.2.2. Diseño y explicación elemento 2

<Diseño del elemento, es decir formato detallado de cabeceras, líneas, etc. y una explicación textual de su uso.>

Aparte de las bases de datos en excel, también se hará una copia de seguridad en un sistema RAID de 16TB (**1 discos de WD My Book Duo 16TB 3.5**). Por si sufrimos algún tipo de ciberataque o ocurre algún error, poder usar las copias de seguridad para volver atrás y encontrar el fallo. Estos backups no estarán conectados en red por si sufrimos algún ataque no se puedan corromper. Además se revisará que funcione correctamente y una copia también la llevará una instalación externa a la empresa.