



Documento
DIAGRAMA ENTIDAD-RELACIÓN

Plataforma para Clubes Deportivos

TracksYours

[code=TRACKS]



Nombre_fichero:	DAW_PWR_TRACKS_UT01.3_Diagrama E-R.odt
Fecha de esta versión:	12/02/2026
Propiedad:	RHMS Sporting Developments

HISTORIAL DE REVISIONES

Fecha	Descripción	Autor
12/02/2026	Primera versión del documento <i>Diagrama de entidad/relación</i>	Roberto Himar Medina Sosa

Sumario

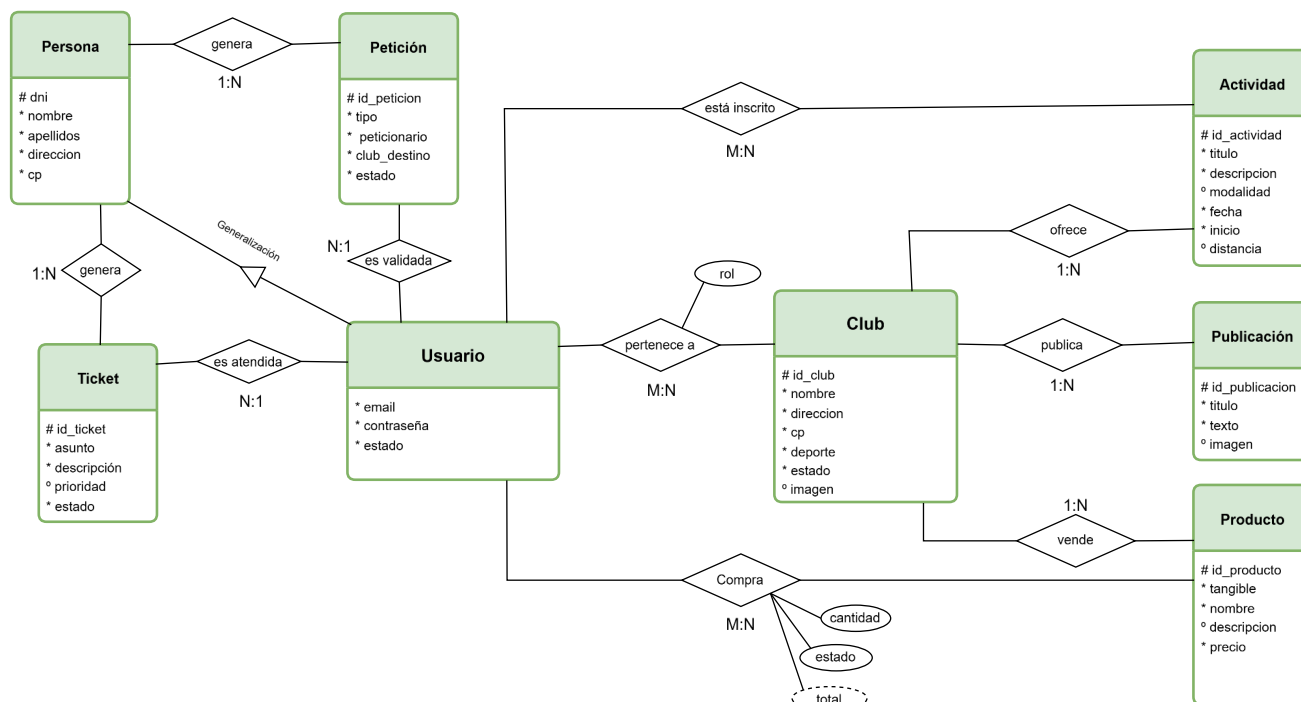
INTRODUCCIÓN.....	4
DIAGRAMA DE ENTIDAD/RELACIÓN.....	4
MODELO RELACIONAL.....	5
Modelo lógico relacional:.....	5
Modelo relacional implementable:.....	5
PROPUESTA DE SCRIPT DDL.....	6

1 INTRODUCCIÓN

Este documento forma se constituye como uno de los entregables de la fase de análisis del proyecto *TracksYours*. Pretende, una vez analizadas las historias de usuario, determinados los requisitos funcionales e identificados los casos de uso de la solución a implementar, establecer que entidades se detectan en el sistema. Esto servirá de base para establecer el modelo de datos de la aplicación.

2 DIAGRAMA DE ENTIDAD/RELACIÓN.

A continuación se muestran las entidades que se han podido establecer como presentes en la solución y las relaciones entre ellas. Se muestran de manera visual en el siguiente diagrama de E/R:



**Apuntar que se ha empleado una notación para los atributos de las entidades un poco diferenciada de la común en el marco de referencia UML a fin de hacer el diagrama más legible.

La decisión de diseño del modelo de datos, y de las entidades en el presentes, está basada en los requisitos que se establecen para el producto *TrackYours*.

Si bien todas las entidades tienen la consideración de entidades fuertes, no habiendo ninguna débil, existen dos entidades de central relevancia que son el **Club** y el **Usuario**. Notar la entidad **Persona** que viene a representar a un visitante de la aplicación que aún no tiene la consideración de usuario, pero que desde esa condición de visitante puede, además de ver el *Home* de la aplicación web, generar **Petición** de alta de club o de socio. También podrá generar **Ticket** de soporte que será atendido por un administrador de la aplicación.

Un **Usuario** podrá tener tres roles asociados, el de “administrador”, establecido por el sistema inicialmente, siendo único, o los roles de “gestor” de un club o “socio” de un club. Todos éstos serán los que podrán autenticarse en la aplicación. El administrador tendrá la potestad de validar las peticiones de altas de club con su correspondiente gestor, y los gestores tendrán la potestad de validar las peticiones de altas de socio en un club.

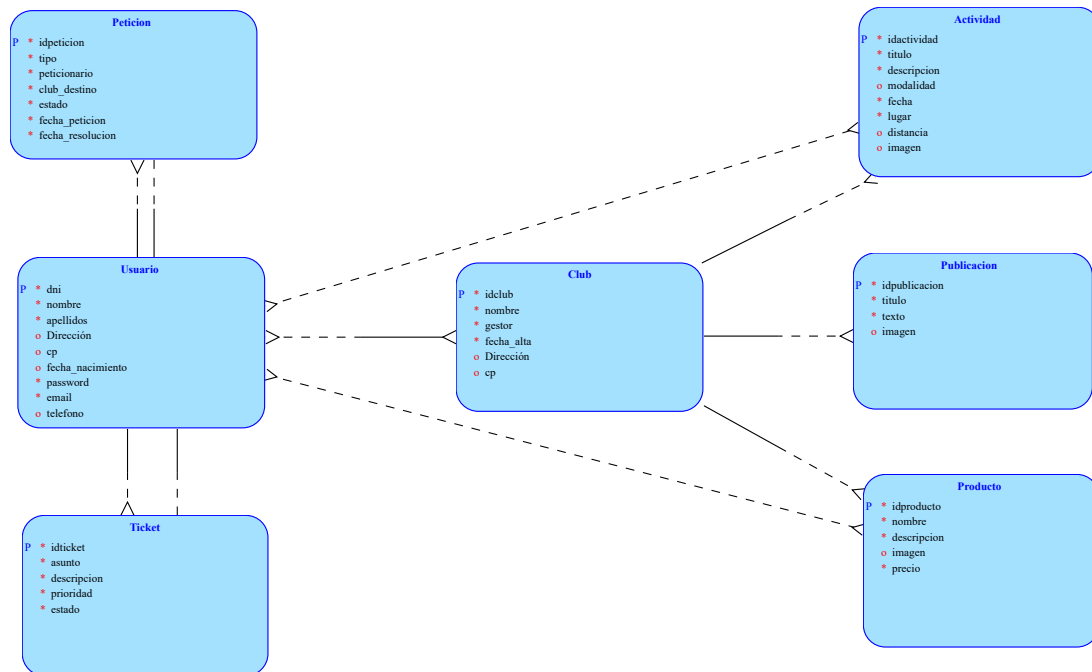
Un **Club** ofrece **Actividades** para su realización, publica **Publicaciones** para mantener informados a sus socios y vende **Productos** a sus socios. De ahí las relaciones 1:N del club con estos tres tipos de entidades.

Un usuario puede leer una publicación cuantas veces quiera y no hay relación estricta entre estas dos entidades. Pero donde sí que se establecen relaciones de inscripción es entre Usuario (ya sea gestor o socio) y una Actividad, o relación de compra entre Usuario y un Producto.

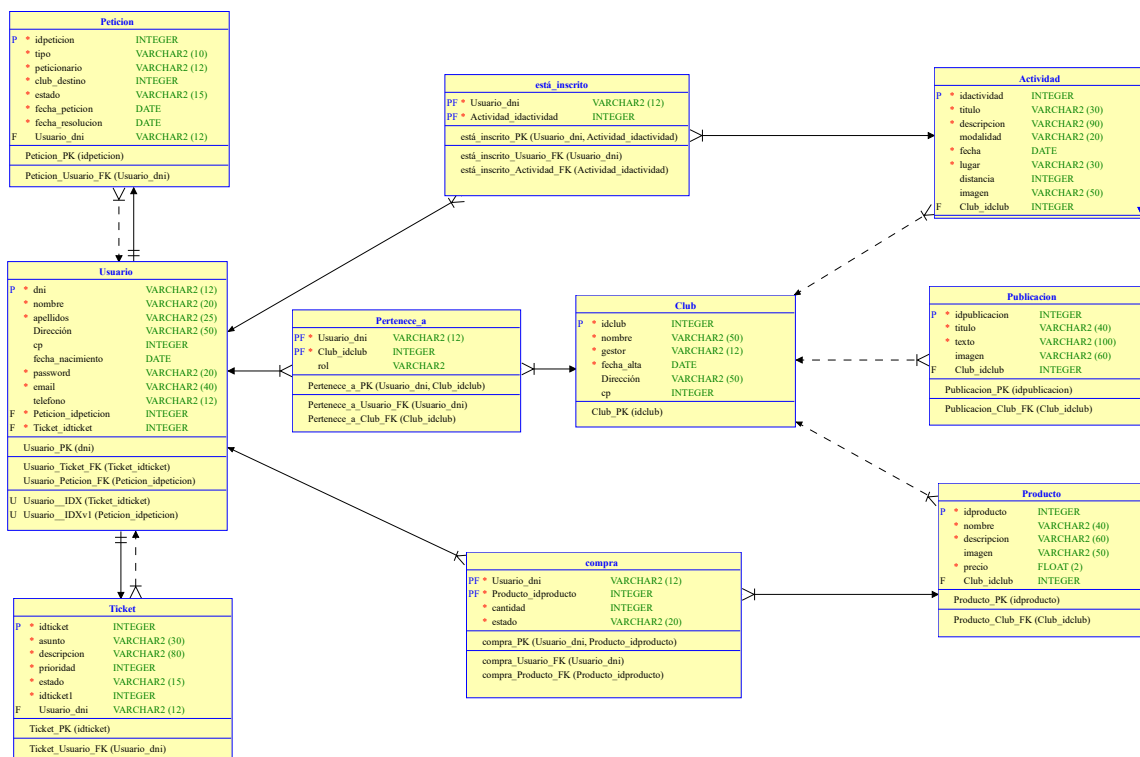
3 MODELO RELACIONAL.

En el apartado anterior se justificó y explicó la decisión de las entidades del sistema escogidas y sus relaciones. Pues basándonos en tal diagrama E/R resultante podemos definir el modelo lógico relacional en notación *Cross Foot* en la aplicación *Data Modeler* de Oracle, la cual a posteriori nos permite aplicar ingeniería y extraer un diagrama relacional implementable. Este último, a su vez, hace más sencillo materializar el modelo conceptual inicial a lenguaje DDL que nos sirva para lograr la implementación del modelo de datos.

3.1 Modelo lógico relacional:



3.2 Modelo relacional implementable:



4 PROPUESTA DE SCRIPT DDL.

A partir del modelo anterior se puede obtener una propuesta de script DDL que nos sirve para implementar o materializar el modelo de datos en un esquema de base de datos. En este punto si se debe indicar el *disclaimer* de que es una propuesta susceptible de modificaciones en fase de implementación del proyecto si se detectan necesidades o casuísticas que se deban cumplir y que impliquen modificaciones del esquema de base de datos fruto de este script propuesto.

Se deben añadir los campos en cada tabla que permiten asegurar la trazabilidad acorde a lo que establece el ENS, esto es, campos de fecha de creación y modificación de registros de la base de datos y el actor ejecutándolas.

```
CREATE TABLE Actividad
(
    idactividad    INTEGER NOT NULL ,
    titulo        VARCHAR2 (30) NOT NULL ,
    descripcion    VARCHAR2 (90) NOT NULL ,
    modalidad     VARCHAR2 (20) ,
    fecha         DATE NOT NULL ,
    lugar         VARCHAR2 (30) NOT NULL ,
    distancia     INTEGER ,
    imagen        VARCHAR2 (50) ,
    Club_idclub   INTEGER
);

ALTER TABLE Actividad
ADD CONSTRAINT Actividad_PK PRIMARY KEY ( idactividad );

CREATE TABLE Club
(
    idclub        INTEGER NOT NULL ,
    nombre        VARCHAR2 (50) NOT NULL ,
    gestor        VARCHAR2 (12) NOT NULL ,
    fecha_alta    DATE NOT NULL ,
    Dirección     VARCHAR2 (50) ,
    cp            INTEGER
);

ALTER TABLE Club
ADD CONSTRAINT Club_PK PRIMARY KEY ( idclub );

CREATE TABLE compra
(
    Usuario_dni   VARCHAR2 (12) NOT NULL ,
    Producto_idproducto INTEGER NOT NULL ,
    cantidad      INTEGER NOT NULL ,
    estado        VARCHAR2 (20) NOT NULL
);

ALTER TABLE compra
ADD CONSTRAINT compra_PK PRIMARY KEY ( Usuario_dni, Producto_idproducto );

CREATE TABLE está_inscrito
(
    Usuario_dni   VARCHAR2 (12) NOT NULL ,
    Actividad_idactividad INTEGER NOT NULL
);

ALTER TABLE está_inscrito
ADD CONSTRAINT está_inscrito_PK PRIMARY KEY ( Usuario_dni, Actividad_idactividad );

CREATE TABLE Pertenece_a
(
    Usuario_dni   VARCHAR2 (12) NOT NULL ,
    Club_idclub   INTEGER NOT NULL ,
    rol           VARCHAR2 (12) NOT NULL
);

ALTER TABLE Pertenece_a
ADD CONSTRAINT Pertenece_a_PK PRIMARY KEY ( Usuario_dni, Club_idclub );

CREATE TABLE Peticion
(
    idpeticion    INTEGER NOT NULL ,
    tipo          VARCHAR2 (10) NOT NULL ,
    peticionario  VARCHAR2 (12) NOT NULL ,
    club_destino  INTEGER NOT NULL ,
    estado        VARCHAR2 (15) NOT NULL ,
    fecha_peticion DATE NOT NULL ,
    fecha_resolucion DATE NOT NULL ,
    Usuario_dni   VARCHAR2 (12)
);

ALTER TABLE Peticion
ADD CONSTRAINT Peticion_PK PRIMARY KEY ( idpeticion );

CREATE TABLE Producto
(
    idproducto    INTEGER NOT NULL ,
    nombre        VARCHAR2 (40) NOT NULL ,
    descripcion    VARCHAR2 (60) NOT NULL ,
    imagen        VARCHAR2 (50) ,
    precio        FLOAT (2) NOT NULL ,
    Club_idclub   INTEGER
);

ALTER TABLE Producto
ADD CONSTRAINT Producto_PK PRIMARY KEY ( idproducto );
```

```

CREATE TABLE Publicacion
(
    idpublicacion INTEGER NOT NULL ,
    titulo        VARCHAR2 (40) NOT NULL ,
    texto         VARCHAR2 (100) NOT NULL ,
    imagen        VARCHAR2 (60) ,
    Club_idclub   INTEGER
);

ALTER TABLE Publicacion
ADD CONSTRAINT Publicacion_PK PRIMARY KEY ( idpublicacion );

CREATE TABLE Ticket
(
    idticket      INTEGER NOT NULL ,
    asunto        VARCHAR2 (30) NOT NULL ,
    descripcion    VARCHAR2 (80) NOT NULL ,
    prioridad     INTEGER NOT NULL ,
    estado        VARCHAR2 (15) NOT NULL ,
    idticket1     INTEGER NOT NULL ,
    Usuario_dni   VARCHAR2 (12)
);

ALTER TABLE Ticket
ADD CONSTRAINT Ticket_PK PRIMARY KEY ( idticket );

CREATE TABLE Usuario
(
    dni            VARCHAR2 (12) NOT NULL ,
    nombre        VARCHAR2 (20) NOT NULL ,
    apellidos     VARCHAR2 (25) NOT NULL ,
    Dirección     VARCHAR2 (50) ,
    cp            INTEGER ,
    fecha_nacimiento DATE ,
    password      VARCHAR2 (20) NOT NULL ,
    email         VARCHAR2 (40) NOT NULL ,
    telefono      VARCHAR2 (12) ,
    Peticion_idpeticion INTEGER NOT NULL ,
    Ticket_idticket INTEGER NOT NULL
);

CREATE UNIQUE INDEX Usuario__IDX ON Usuario
(
    Ticket_idticket ASC
);

CREATE UNIQUE INDEX Usuario__IDXv1 ON Usuario
(
    Peticion_idpeticion ASC
);

ALTER TABLE Usuario
ADD CONSTRAINT Usuario_PK PRIMARY KEY ( dni );

ALTER TABLE Actividad
ADD CONSTRAINT Actividad_Club_FK FOREIGN KEY
(
    Club_idclub
)
REFERENCES Club
(
    idclub
);

ALTER TABLE compra
ADD CONSTRAINT compra_Producto_FK FOREIGN KEY
(
    Producto_idproducto
)
REFERENCES Producto
(
    idproducto
);

ALTER TABLE compra
ADD CONSTRAINT compra_Usuario_FK FOREIGN KEY
(
    Usuario_dni
)
REFERENCES Usuario
(
    dni
);

ALTER TABLE está_inscrito
ADD CONSTRAINT está_inscrito_Actividad_FK FOREIGN KEY
(
    Actividad_idactividad
)
REFERENCES Actividad
(
    idactividad
);

ALTER TABLE está_inscrito
ADD CONSTRAINT está_inscrito_Usuario_FK FOREIGN KEY
(
    Usuario_dni
)
REFERENCES Usuario
(
    dni
);

```

```
ALTER TABLE Pertenece_a
ADD CONSTRAINT Pertenece_a_Club_FK FOREIGN KEY
(
Club_idclub
)
REFERENCES Club
(
idclub
);

ALTER TABLE Pertenece_a
ADD CONSTRAINT Pertenece_a_Usuario_FK FOREIGN KEY
(
Usuario_dni
)
REFERENCES Usuario
(
dni
);

ALTER TABLE Peticion
ADD CONSTRAINT Peticion_Usuario_FK FOREIGN KEY
(
Usuario_dni
)
REFERENCES Usuario
(
dni
);

ALTER TABLE Producto
ADD CONSTRAINT Producto_Club_FK FOREIGN KEY
(
Club_idclub
)
REFERENCES Club
(
idclub
);

ALTER TABLE Publicacion
ADD CONSTRAINT Publicacion_Club_FK FOREIGN KEY
(
Club_idclub
)
REFERENCES Club
(
idclub
);

ALTER TABLE Ticket
ADD CONSTRAINT Ticket_Usuario_FK FOREIGN KEY
(
Usuario_dni
)
REFERENCES Usuario
(
dni
);

ALTER TABLE Usuario
ADD CONSTRAINT Usuario_Peticion_FK FOREIGN KEY
(
Peticion_idpeticion
)
REFERENCES Peticion
(
idpeticion
);

ALTER TABLE Usuario
ADD CONSTRAINT Usuario_Ticket_FK FOREIGN KEY
(
Ticket_idticket
)
REFERENCES Ticket
(
idticket
);
```