**GRUPO DE MONTAÑA ARABA**

PROYECTO FIN DE CURSO 2019-2020

1º DESARROLLO DE APLICACIONES MULTIPLATAFORMA

TAREAS

[1. Análisis del proyecto. 3](#_Toc42452962)

[2. Diseño. 4](#_Toc42452963)

[*a)* *Modelo entidad relación* 4](#_Toc42452964)

[*b)* *Diagrama de clases* 6](#_Toc42452965)

[*c)* *Diagramas de secuencia* 7](#_Toc42452966)

[3. Desarrollo. 9](#_Toc42452967)

[*a)* *Fichero DDL.* 9](#_Toc42452968)

[ Tabla PERFIL: 9](#_Toc42452969)

[ Tabla SOCIO: 9](#_Toc42452970)

[ Tabla CUOTA: 10](#_Toc42452971)

[ Tabla CUOTAPAGADA: 10](#_Toc42452972)

[ Tabla TIPOACTIVIDAD: 11](#_Toc42452973)

[ Tabla DIFICULTAD: 11](#_Toc42452974)

[ Tabla ACTIVIDAD: 11](#_Toc42452975)

[ Tabla SOCIOPARTICIPA: 12](#_Toc42452976)

[ Tabla JUNTA: 12](#_Toc42452977)

[ Tabla ACTIVIDADJUNTA: 13](#_Toc42452978)

[ TRIGGER 1: BAJAS 14](#_Toc42452979)

[ TRIGGER 2: MENOR 15](#_Toc42452980)

[*b)* *Vista.* 16](#_Toc42452981)

# Análisis del proyecto.

El proyecto a realizar tiene como objetivo la creación de una aplicación para un club de montaña.

En dicha aplicación se recogerán los siguientes datos:

* **Socios** del club. Estos tendrán que ser mayores de 4 años.
  + **Socios adultos:** mayores de 18 años.
  + **Socios menores:** de 4 a 18 años. Tendrán un **responsable** que será un socio adulto.
* **Cuotas.** Las cuotas vendrán determinadas por la edad de los socios (adultos/menores).
* **Actividades**:
  + Los socios organizarán al menos 1 actividad y participan en varias actividades.
  + Estas actividades se deciden en las **juntas.**
  + Las actividades tendrán definido un **tipo de actividad.**
  + Cada actividad es única.
* **Juntas**:
  + Formadas por socios, los cuales cada uno desempeñará un cargo.

# Diseño.

# Modelo entidad relación

RELACIONES:

1. SOCIOS-CUOTAS:

M-N: Los socios pagan 1 o más cuotas. Entiendo que cada año pagan 1 cuota, con lo cual si están asociados al club pagarán más de una cuota.

Hay dos tipos de cuotas, que son pagadas por distintos socios.

Esta relación tiene atributos propios (pagado y fecha).

1. SOCIOS-ACTIVIDADES
   1. ORGANIZA:

1-N: 1 socio debe organizar al menos 1 o más actividades. 1 actividad es organizada por 1 socio.

* 1. PARTICIPA:

M-N: 1 socio puede participar en varias actividades. 1 actividad puede ser realizada por varios socios.

1. SOCIOS-CARGOS\_JUNTA:

1-1: 1 socio en 1 junta solo puede desempeñar un cargo. Hay 9 cargos y cada cargo solo es desempeñado por 1 socio.

1. ACTIVIDADES-JUNTAS:

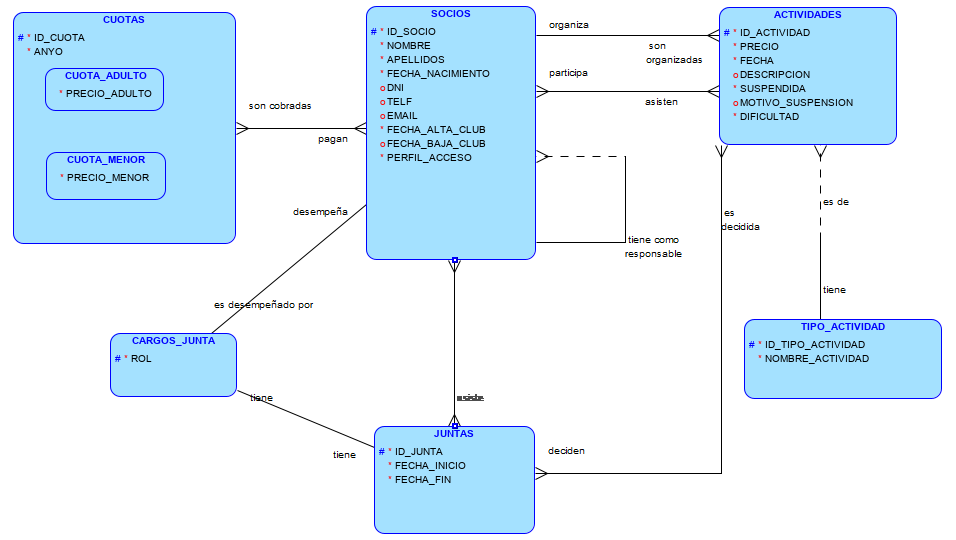
M-N: una actividad es programada en una junta. Una junta puede programar distintas actividades.

1. ACTIVIDADES-TIPO\_ACTIVIDAD:

1-N: una actividad debe ser de un tipo de actividad, puede que un tipo de actividad nunca se lleve a cabo.

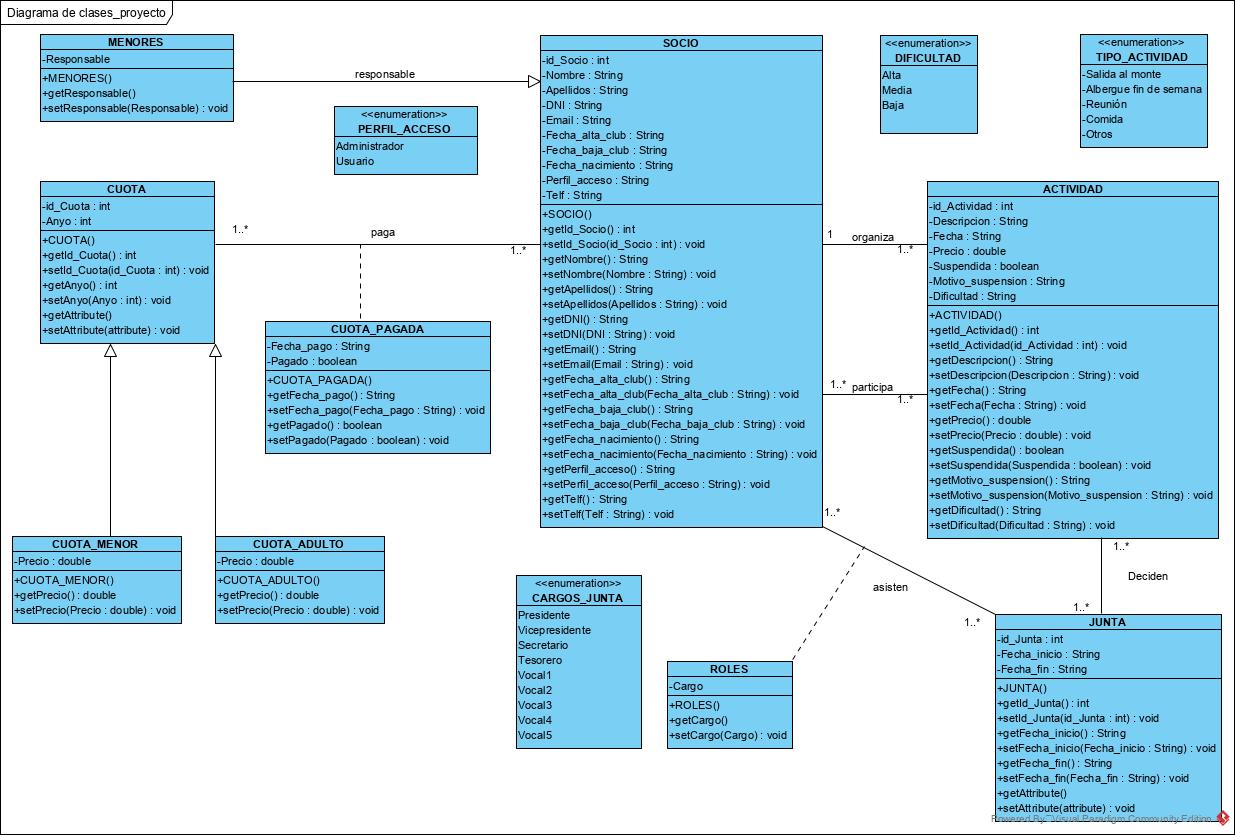
1. SOCIOS-SOCIOS

1-N: 1 socio menor tiene siempre que tener un responsable. 1 socio adulto puede no tener que ser responsable de ningún menor.



# Diagrama de clases

En un diagrama de clases tratamos de ver una imagen estática de los elementos básicos de nuestra aplicación y las relaciones establecidas entre los elementos.



*\*los métodos incluidos en las clases nos permiten añadir, buscar, modificar datos. Cuando se programe la aplicación aparecerán métodos nuevos.*

HERENCIAS:

1. CUOTA\_ADULTO y CUOTA\_MENOR heredan de CUOTA.
2. MENORES hereda de SOCIO.

LISTAS ENUMERADAS:

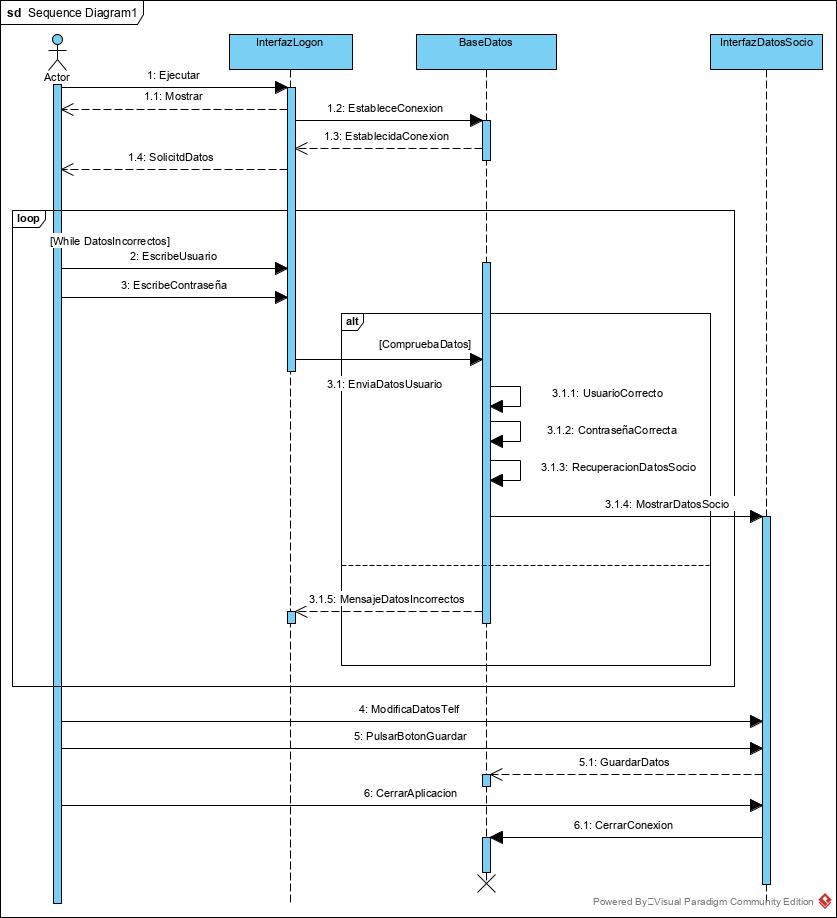
1. PERFIL DE ACCESO
2. DIFICULTAD
3. TIPO ACTIVIDAD
4. CARGOS JUNTA

Las uniones entre clases son ASOCIACIONES.

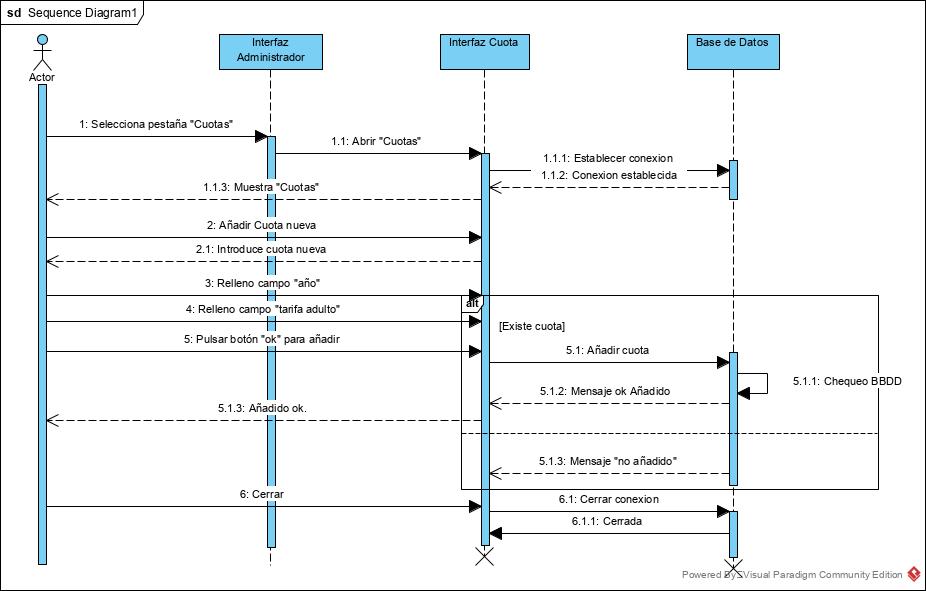
# Diagramas de secuencia

Un diagrama de secuencia nos muestra un proceso que se lleva a cabo en nuestra aplicación, revisando que objetos interactúan entre sí en el tiempo.

A continuación se muestra un proceso en el cual un socio desea modificar su número de teléfono en el aplicativo.



En el siguiente diagrama de secuencia llevamos a cabo los pasos necesarios para dar de alta una nueva cuota por un administrador del sistema.



# Desarrollo.

# Fichero DDL.

Todas las “insert” de todas las tablas las incluyo en el archivo “DATOS\_TABLAS.SQL”.

* Tabla PERFIL:

CREATE TABLE PERFIL(

PERFIL VARCHAR2(13) PRIMARY KEY

);

* Tabla SOCIO:

He creado una secuencia auto incremental para el código de socio:

CREATE SEQUENCE INCREMENTO\_IDSOCIOS

INCREMENT BY 1

START WITH 0

MINVALUE 0;

CREATE TABLE SOCIO (

ID\_SOCIO NUMBER NOT NULL PRIMARY KEY,

NOMBRE VARCHAR2(25) NOT NULL,

APELLIDOS VARCHAR2(25) NOT NULL,

FECHA\_NACIMIENTO DATE NOT NULL,

DNI VARCHAR2(10),

TELF VARCHAR2(14),

EMAIL VARCHAR2(25),

FECHA\_ALTA\_CLUB DATE NOT NULL,

FECHA\_BAJA\_CLUB DATE,

PERFIL\_ACCESO VARCHAR2(13),

REPRESENTANTE NUMBER,

CONSTRAINT SOC\_IDS\_FK FOREIGN KEY (ID\_SOCIO) REFERENCES SOCIO (ID\_SOCIO),

CONSTRAINT SOC\_PER\_FK FOREIGN KEY(PERFIL\_ACCESO) REFERENCES PERFIL(PERFIL)

);

* Tabla CUOTA:

He creado una secuencia auto incremental para el código de cuota:

CREATE SEQUENCE INCREMENTO\_CUOTAS

INCREMENT BY 1

START WITH 0

MINVALUE 0;

CREATE TABLE CUOTA(

ID\_CUOTA NUMBER NOT NULL PRIMARY KEY,

PRECIO\_ADULTO NUMBER,

PRECIO\_MENOR NUMBER,

ANYO DATE

);

* Tabla CUOTAPAGADA:

CREATE TABLE CUOTAPAGADA (

ID\_SOCIO NUMBER NOT NULL,

ID\_CUOTA NUMBER NOT NULL,

FECHA DATE,

PAGADO VARCHAR2(2),

CONSTRAINT CUOPA\_PK PRIMARY KEY(ID\_SOCIO, ID\_CUOTA),

CONSTRAINT SO\_IDS\_FK FOREIGN KEY (ID\_SOCIO) REFERENCES SOCIO(ID\_SOCIO),

CONSTRAINT CUO\_IDC\_FK FOREIGN KEY (ID\_CUOTA) REFERENCES CUOTA (ID\_CUOTA)

);

* Tabla TIPOACTIVIDAD:

CREATE TABLE TIPOACTIVIDAD(

ID\_TIPO\_ACTIVIDAD NUMBER PRIMARY KEY,

DESCRIPCION VARCHAR2(25)

);

* Tabla DIFICULTAD:

CREATE TABLE DIFICULTAD(

DIFICULTAD VARCHAR2(5) PRIMARY KEY

);

* Tabla ACTIVIDAD:

He creado una secuencia auto incremental para id de actividad:

*CREATE SEQUENCE INCREMENTO\_IDACTIVIDAD*

*INCREMENT BY 1*

*START WITH 0*

*MINVALUE 0;*

CREATE TABLE ACTIVIDAD (

ID\_ACTIVIDAD NUMBER NOT NULL PRIMARY KEY,

DESCRIPCION VARCHAR2(50) NOT NULL,

FECHA DATE NOT NULL,

PRECIO NUMBER(4,2),

SUSPENDIDA VARCHAR2(2),

MOTIVO\_SUSPENSION VARCHAR2(50),

DIFICULTAD VARCHAR2(5),

ORGANIZADOR NUMBER NOT NULL,

TIPOACTIVIDAD NUMBER NOT NULL,

CONSTRAINT ACT\_ORG\_FK FOREIGN KEY (ORGANIZADOR) REFERENCES SOCIO (ID\_SOCIO),

CONSTRAINT ACT\_TIP\_FK FOREIGN KEY (TIPOACTIVIDAD) REFERENCES TIPOACTIVIDAD(ID\_TIPO\_ACTIVIDAD),

CONSTRAINT ACT\_DIF\_FK FOREIGN KEY (DIFICULTAD) REFERENCES DIFICULTAD (DIFICULTAD)

);

* Tabla SOCIOPARTICIPA:

CREATE TABLE SOCIOPARTICIPA (

ID\_ACTIVIDAD NUMBER NOT NULL,

ID\_PARTICIPANTE NUMBER NOT NULL,

CONSTRAINT SP\_PK PRIMARY KEY(ID\_ACTIVIDAD, ID\_PARTICIPANTE),

CONSTRAINT SOC\_ID\_FK FOREIGN KEY (ID\_PARTICIPANTE) REFERENCES SOCIO(ID\_SOCIO),

CONSTRAINT ACT\_ID\_FK FOREIGN KEY (ID\_ACTIVIDAD) REFERENCES ACTIVIDAD (ID\_ACTIVIDAD)

);

* Tabla JUNTA:

CREATE SEQUENCE INCREMENTO\_JUNTAS

INCREMENT BY 1

START WITH 0

MINVALUE 0;

CREATE TABLE JUNTA(

ID\_JUNTA NUMBER NOT NULL PRIMARY KEY,

PRESIDENTE NUMBER CONSTRAINT SOC\_IDP\_FK REFERENCES SOCIO,

VICEPRESIDENTE NUMBER CONSTRAINT SOV\_IDS\_FK REFERENCES SOCIO,

SECRETARIO NUMBER CONSTRAINT SOC\_ISE\_FK REFERENCES SOCIO,

TESORERO NUMBER CONSTRAINT SOC\_IDT\_FK REFERENCES SOCIO,

VOCAL1 NUMBER CONSTRAINT SOC\_ID1\_FK REFERENCES SOCIO,

VOCAL2 NUMBER CONSTRAINT SOC\_ID2\_FK REFERENCES SOCIO,

VOCAL3 NUMBER CONSTRAINT SOC\_ID3\_FK REFERENCES SOCIO,

VOCAL4 NUMBER CONSTRAINT SOC\_ID4\_FK REFERENCES SOCIO,

VOCAL5 NUMBER CONSTRAINT SOC\_ID5\_FK REFERENCES SOCIO,

FECHA\_INICIO DATE,

FECHA\_FIN DATE

);

* Tabla ACTIVIDADJUNTA:

CREATE TABLE ACTIVIDADJUNTA(

ID\_ACTIVIDAD NUMBER NOT NULL,

ID\_JUNTA NUMBER NOT NULL,

CONSTRAINT AJ\_PK PRIMARY KEY(ID\_ACTIVIDAD, ID\_JUNTA),

CONSTRAINT ACT\_IDJ\_FK FOREIGN KEY (ID\_JUNTA) REFERENCES JUNTA(ID\_JUNTA),

CONSTRAINT ACT\_IDA\_FK FOREIGN KEY (ID\_ACTIVIDAD) REFERENCES ACTIVIDAD (ID\_ACTIVIDAD)

);

* TRIGGER 1: BAJAS

En este disparador debemos controlar que aquellos socios que no paguen la cuota antes del 31/12 del año correspondiente, se dará de baja.

create or replace trigger bajas

before insert on cuota

for each row

declare

v\_socio cuotapagada.id\_socio%type;

v\_fIni cuota.anyo%type:= :new.anyo-730;

v\_fFin cuota.anyo%type:= :new.anyo-365;

cursor cuotasImpagadas is

select id\_socio from cuotapagada, cuota

where pagado is null and

cuota.id\_cuota = cuotapagada.id\_cuota and

cuota.anyo >= v\_fIni and cuota.anyo <= v\_fFin;

begin

open cuotasImpagadas;

loop

fetch cuotasImpagadas into v\_socio;

exit when cuotasImpagadas%notfound;

update socio set fecha\_baja\_club = sysdate where v\_socio = id\_socio;

update cuotapagada set pagado = 'No' where v\_socio = id\_socio;

end loop;

close cuotasImpagadas;

end;

* TRIGGER 2: MENOR

Controlar que los socios menores de edad tengan un responsable, que será otro socio.

create or replace trigger menor

before insert on socio

for each roW

declare

annio number;

e\_falta\_responsable exception;

e\_no\_inscribir exception;

begin

select trunc(months\_between((sysdate,:neW.fecha\_nacimiento))/12) into annio from dual;

insert into error values(indiceerror.nextVal, 'trigger', annio, current\_timestamp);

-- si es menor 18 y mayor de 4

if annio >=4 and annio <18 then

insert into error values(indiceerror.nextVal, 'trigger', 'estoy en el if de annio', current\_timestamp);

raise e\_falta\_responsable;

elsif (annio <4) then

insert into error values(indiceerror.nextVal, 'trigger', 'estoy en el elsif de annio', current\_timestamp);

raise e\_no\_inscribir;

end if;

exception

when e\_falta\_responsable then

raise\_application\_error(-20100, 'falta introducir un responsable para el nuevo socio');

when e\_no\_inscribir then

raise\_application\_error(-20101, 'no se puede inscribir al nuevo socio, es menor de 4 años.');

end menor;

# Vista.

SELECT

actividad.fecha as "FECHA ACTIVIDAD",

tipoactividad.descripcion as "TIPO DE ACTIVIDAD",

ACTIVIDAD.DESCRIPCION AS "DESCRIPCION ACTIVIDAD",

ACTIVIDAD.PRECIO,

SOCIO.NOMBRE AS "NOMBRE ORGANIZADOR",

SOCIO.APELLIDOS AS "APELLIDOS ORGANIZADOR"

FROM

ACTIVIDAD, TIPOACTIVIDAD , SOCIO

WHERE

ACTIVIDAD.TIPOACTIVIDAD=TIPOACTIVIDAD.ID\_TIPO\_ACTIVIDAD AND

ACTIVIDAD.PRECIO>0 AND

ACTIVIDAD.ORGANIZADOR=SOCIO.ID\_SOCIO;