

UNIVERSIDAD AUTÓNOMA GABRIEL RENE MORENO

FACULTAD DE INGENIERÍA EN CIENCIAS DE LA COMPUTACIÓN Y TELECOMUNICACIONES

PROYECTO. BASE DE DATOS 1 : "COLEGIO"

MATERIA: Base de datos 1

DOCENTE: Ing. Josué Obed Veizaga G.

NOMBRE: Carrasco Dalba Daniela

REGISTRO: 219010641

GRUPO: SC

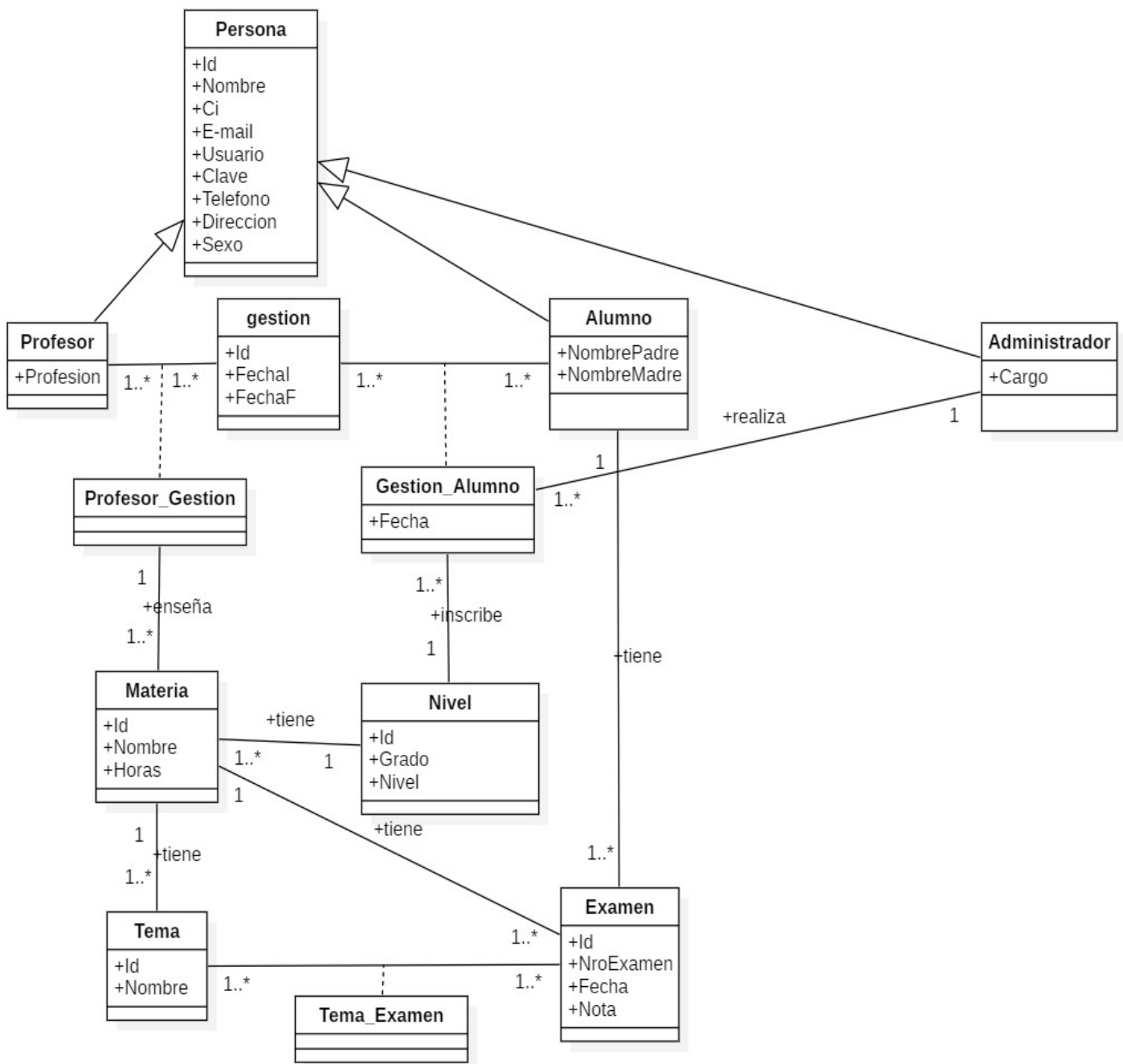
SANTA CRUZ - BOLIVIA

PROYECTO BASE DE DATOS I

Se quiere diseñar una aplicación para gestionar los datos de un colegio llamado “SAN MARTIN DE PORRES” desde inicial hasta bachiller de cada estudiante. Tras analizar el problema, la aplicación debe cumplir con los siguientes requisitos:

- En un colegio hay 3 tipos de usuarios : estudiantes, profesores y administrativos. De cada usuario se necesitan registrar sus datos personales como ser : nombre, CI , e-mail, dirección, teléfono, sexo. Además, cada usuario tendrá un nombre de usuario y una clave.
- Cada estudiante será registrado en el nivel que le corresponda, por un administrativo en una determinada gestión.
- De los estudiantes se quiere registrar el nombre del padre y de la madre.
- De los profesores también se registrará su profesión y de los administradores su cargo en el colegio.
- El estudiante a lo largo de su estancia en el establecimiento podrá ver las notas que tiene en cada examen de sus materias, como también los temas del cual se lo evaluó y en la fecha realizada.
- Un profesor en una determinada gestión puede enseñar una o varias materias, sin embargo, una materia es enseñada por un solo profesor.
- Cada materia tiene distintos temas y estos son evaluados en un examen. La materia tiene un id, un nombre y las horas totales de la clase en la gestión.
- En una materia se pueden desarrollar varios exámenes, de distintos temas.
- Cada examen tendrá un id que lo identifique como también cuantos exámenes se han tomado de cada materia, la fecha en que fue realizado y la nota.
- Cada gestión tendrá un id que lo identifique, una fecha de inicio y una fecha final.
- Cada nivel tendrá un nivel y un grado. Por ejemplo (id =01-P, grado = 1, nivel = primaria)
- Cada tema tendrá un id, y un nombre.

1. Diseño Conceptual



2. Diseño Lógico

Persona															
PK															
<u>Id</u>	Nombre	ci	E-mail	Usuario	Clave	Telefono	Direccion	Sexo	Profesion	NombrePadre	NombreMadre	Cargo	TipoProf	TipoAlum	TipoAdm
Gestion															
PK															
<u>Id</u>	Fechal	FechaF													
Profesor_Gestion															
PK / fk		PK / fk													
<u>IdProf</u>	<u>IdGestion</u>														
Gestion_Alumno															
PK / fk		PK / fk		fk		fk									
<u>IdGestion</u>	<u>IdAlumno</u>	Fecha	<u>IdAdm</u>	<u>IdNivel</u>											
Materia															
PK				fk		fk		fk							
<u>Id</u>	Nombre	Horas	<u>IdProf</u>	<u>IdGestion</u>	<u>IdNivel</u>										
Nivel															
PK															
<u>Id</u>	Grado	Nivel													
Tema															
PK		fk													
<u>Id</u>	Nombre	<u>IdMateria</u>													
Examen															
PK				fk		fk									
<u>Id</u>	NroExamen	Fecha	Nota	<u>IdMateria</u>	<u>IdAlumno</u>										
Tema_Examen															
PK / fk		PK /fk													
<u>IdTema</u>	<u>IdExamen</u>														

3. Diseño Físico

```
create database colegio;
use colegio;

create table persona(
    id integer not null,
    nombre varchar(50) not null,
    ci integer not null,
    email varchar(50),
    usuario varchar(20) not null,
    clave varchar(20) not null,
    telefono integer,
    direccion varchar(60),
    sexo char not null,
    profesion varchar(25),
    nombrePadre varchar(50),
    nombreMadre varchar(50),
    cargo varchar(25),
    TipoProf tinyint not null,
    TipoAlum tinyint not null,
    TipoAdm tinyint not null,
    primary key (id)
);

create table gestion(
    id integer not null,
    FechaI date not null,
    FechaF date not null,
    primary key (id)
);

create table profesor_gestion(
    primary key (idprof,idgestion),
    idprof integer not null,
    idgestion integer not null,
    foreign key (idprof) references persona(id)
        on update cascade
        on delete cascade,
    foreign key (idgestion) references gestion(id)
        on update cascade
        on delete cascade
);

create table nivel(
    id varchar(4) not null,
    grado tinyint not null,
    nivel varchar(10) not null,
    primary key (id)
);
```

```

create table gestion_alumno(
    primary key (idgestion,idalumno),
    idgestion integer not null,
    idalumno integer not null,
    fecha date not null,
    idadm integer not null,
    idnivel varchar(4) not null,
    foreign key (idgestion) references gestion(id)
        on update cascade
        on delete cascade,
    foreign key (idalumno) references persona(id)
        on update cascade
        on delete cascade,
    foreign key (idadm) references persona(id)
        on update cascade
        on delete cascade,
    foreign key (idnivel) references nivel(id)
        on update cascade
        on delete cascade
);

```

```

create table materia(
    primary key (id),
    id integer not null,
    nombre varchar(50) not null,
    horas int not null,
    idprof integer not null,
    idgestion integer not null,
    idnivel varchar(4) not null,
    foreign key (idprof) references persona(id)
        on update cascade
        on delete cascade,
    foreign key (idgestion) references gestion(id)
        on update cascade
        on delete cascade,
    foreign key (idnivel) references nivel(id)
        on update cascade
        on delete cascade
);

```

```

create table tema(
    primary key (id),
    id integer not null,
    nombre varchar(50) not null,
    idmateria integer not null,
    foreign key (idmateria) references materia(id)
        on update cascade
        on delete cascade
);

```

```

create table examen(
    primary key (id),
    id integer not null,
    nroexamen tinyint not null,
    fecha date not null,
    nota tinyint not null,
    idmateria integer not null,
    idalumno integer not null,
    foreign key (idmateria) references materia(id)
        on update cascade
        on delete cascade,
    foreign key (idalumno) references persona(id)
        on update cascade
        on delete cascade
);

create table tema_examen(
    primary key (idTema, idExamen),
    idTema integer not null,
    idExamen integer not null,
    foreign key (idTema) references tema(id)
        on update cascade
        on delete cascade,
    foreign key (idExamen) references examen(id)
        on update cascade
        on delete cascade
);

```