**Proyecto Bases de Datos**

Jefferson Montenegro

Santiago Tobón

Junio 2022.

Universidad del Valle

Valle del Cauca

Bases de Datos

# Descripción del Proyecto

Este proyecto implementa una herramienta para la gestión de asistencia de los diferentes cursos de la universidad del valle a través de Bienestar Universitario en donde su sistema de almacenamiento es físico desperdiciando recursos y desperdiciando mucho tiempo y espacio.

Con la herramienta buscamos brindar una mejor gestión de este proceso y tener una eficiencia a la hora de almacenar y guardar los datos.

# Temas

Los temas previstos que vamos a utilizar serán:

- Modelo Entidad-Relación

- Modelo Relacional

- Transformación de modelo Entidad-Relación a modelo Relacional

- Lenguaje de Consulta SQL

- Acceso programado a bases de datos

# Objetivos

## Objetivo General

* Desarrollar un programa de Software que permita realizar el control de asistencia y seguimiento de los estudiantes y profesores a los cursos

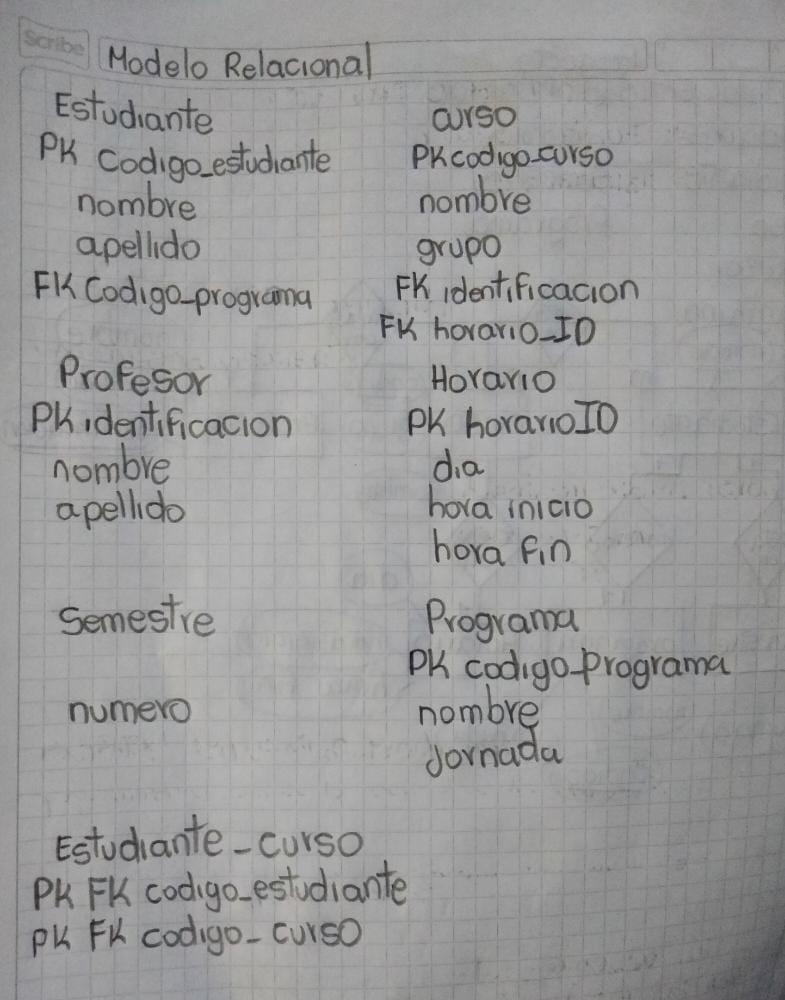
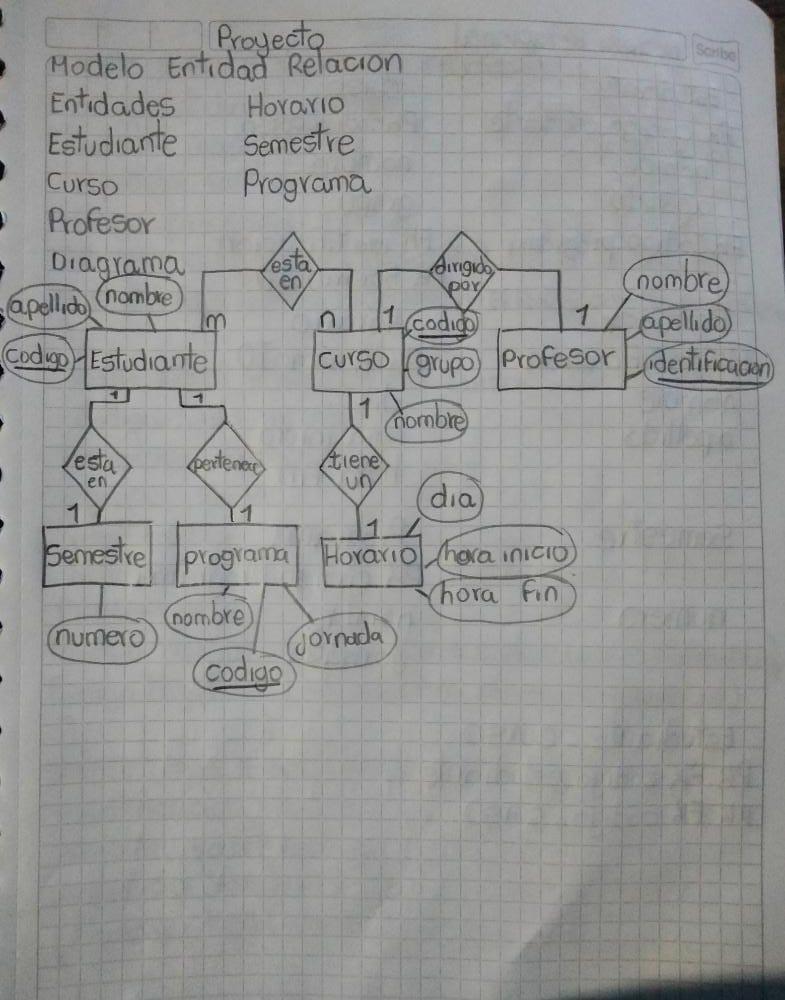
## Objetivos Específicos

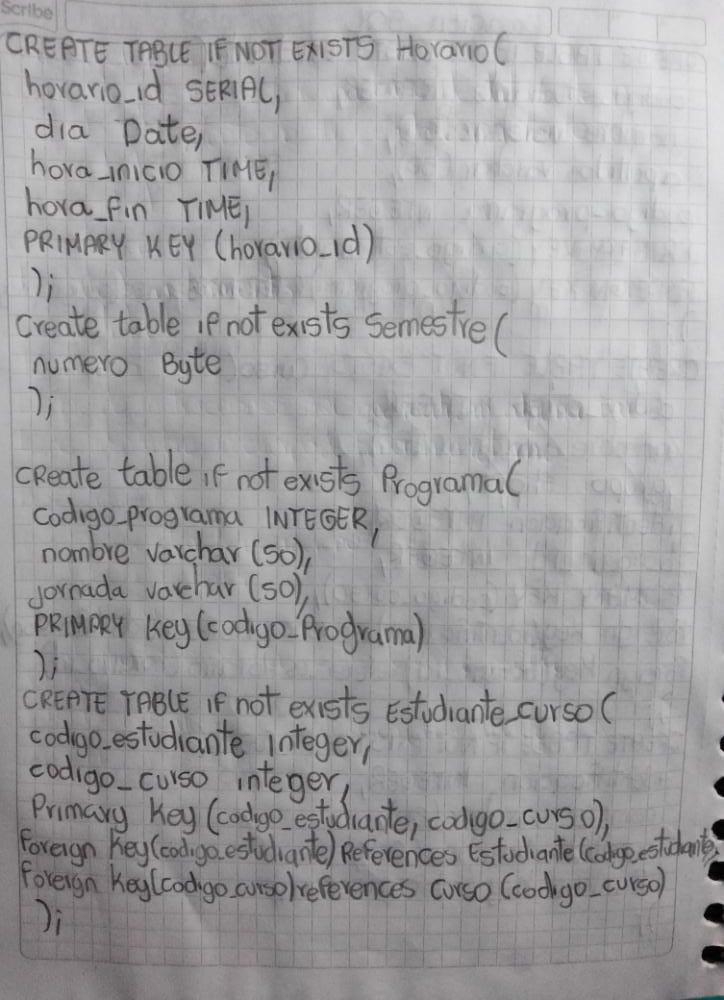
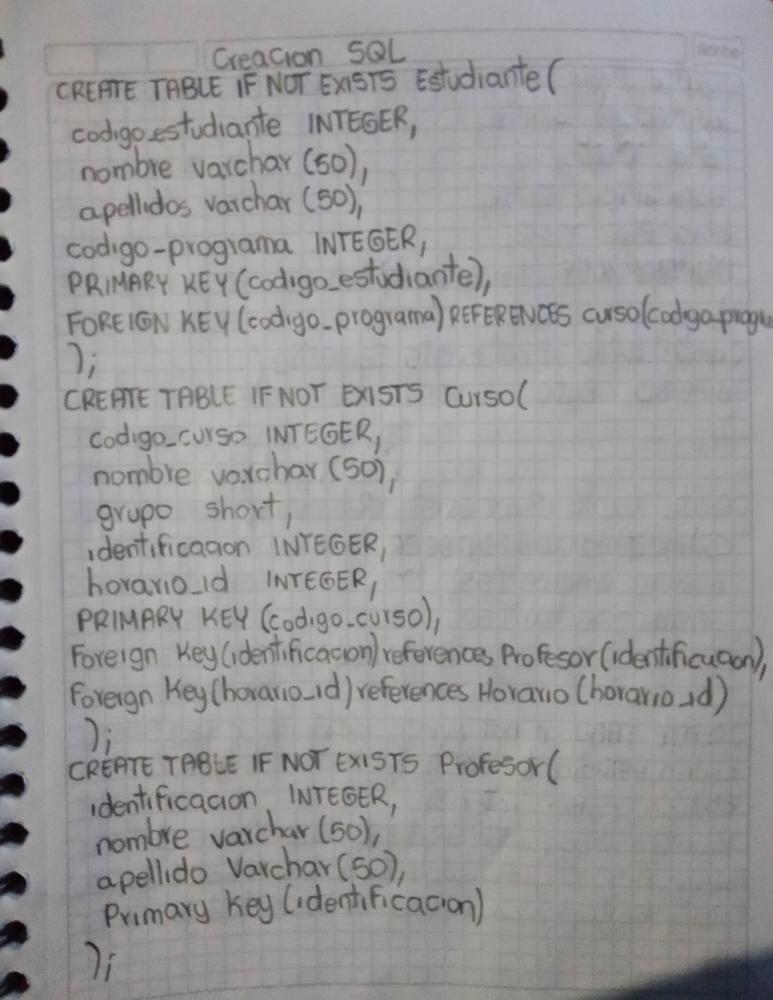
* Hacer la toma de requerimientos para la identificación del problema
* Analizar la información suministrada para hacer la descripción detallada de lo que se desea implementar
* Diseñar el programa de Software teniendo en cuenta los requerimientos
* Construir el programa de acuerdo a la arquitectura propuesta

# Resumen del problema

Según lo analizado esto es un resumen del problema:

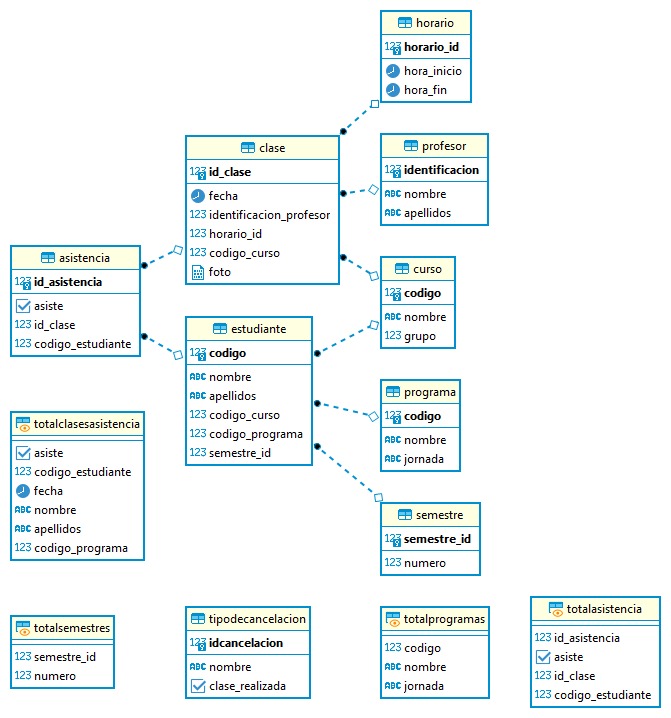
| El problema es | Al almacenar la información de los asistentes de los cursos de las hojas al Excel toma mucho tiempo y el uso de hojas que posteriormente serán desechadas hace un total desperdicio de recursos |
| --- | --- |
| Afecta a | Universidad del Valle sección Bienestar Universitario |
| Cuyo Impacto es | Pérdida de tiempo  Desperdicio de recursos  Dificultades para obtener información |
| Una solución satisfactoria trae | Ahorro de tiempo  Organización de las bases de datos para obtener información |

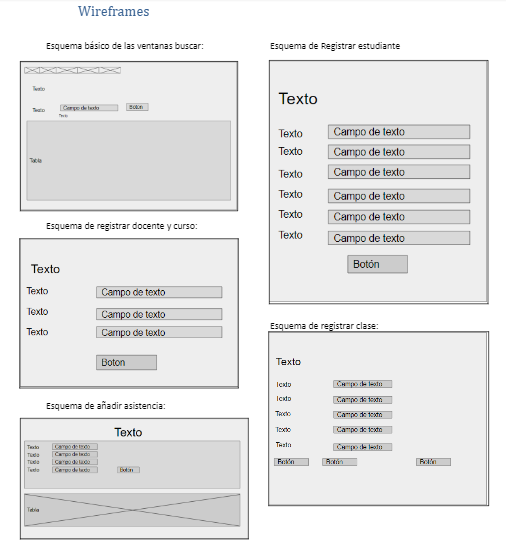


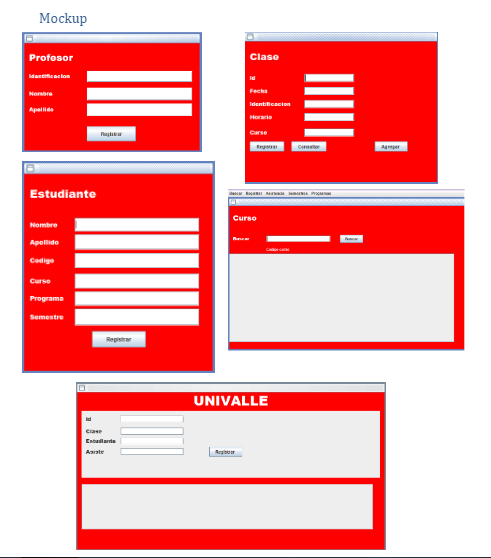


# Diagrama de entidad relación

# Modelo de la base de datos







# Código de creación de tablas

Create table if not exists Curso(

codigo INTEGER,

nombre VARCHAR(50),

grupo INTEGER,

Primary key(codigo)

);

Create table if not exists Horario(

horario\_id integer,

hora\_inicio time,

hora\_fin time,

Primary key(horario\_id)

);

Create table if not exists Profesor(

identificacion integer,

nombre varchar(50),

apellidos varchar(50),

Primary key(identificacion)

);

Create table if not exists Programa(

codigo integer,

nombre varchar(50),

jornada varchar(50),

primary key(codigo)

);

Create table if not exists Semestre(

semestre\_id integer,

numero integer,

Primary key(semestre\_id)

);

Create table if not exists Estudiante(

codigo integer,

nombre varchar(50),

apellidos varchar(50),

codigo\_curso integer,

codigo\_programa integer,

semestre\_id integer,

primary key(codigo),

foreign key(codigo\_curso) references Curso(codigo),

foreign key(codigo\_programa) references Programa(codigo),

foreign key(semestre\_id) references Semestre(semestre\_id)

);

Create table if not exists TipodeCancelacion(

idcancelacion integer,

nombre varchar(50),

clase\_realizada boolean,

primary key(idcancelacion)

);

Create table if not exists Clase(

id\_clase integer,

fecha date,

identificacion\_profesor integer,

horario\_id integer,

codigo\_curso integer,

primary key(id\_clase),

foreign key(identificacion\_profesor) references Profesor(identificacion),

foreign key(horario\_id) references Horario(horario\_id),

foreign key(codigo\_curso) references Curso(codigo)

);

Create table if not exists Asistencia(

id\_asistencia integer,

asiste boolean,

id\_clase integer,

codigo\_estudiante integer,

Primary key(id\_asistencia),

Foreign key(id\_clase) references Clase(id\_clase),

Foreign key(codigo\_estudiante) references Estudiante(codigo)

);

# Código de inserción

create view totalclasesasistencia as

Select asistencia.asiste,asistencia.codigo\_estudiante,clase.fecha,estudiante.nombre,estudiante.apellidos,estudiante.codigo\_programa From asistencia

inner join clase on asistencia.id\_clase = clase.id\_clase

inner join estudiante on asistencia.codigo\_estudiante = estudiante.codigo;

create view totalasistencia as

SELECT asistencia.id\_asistencia,

asistencia.asiste,

asistencia.id\_clase,

asistencia.codigo\_estudiante

FROM asistencia;

create view totalprogramas as

SELECT programa.codigo,

programa.nombre,

programa.jornada

FROM programa;

create view totalsemestres as

SELECT semestre.semestre\_id,

semestre.numero

FROM semestre;

CREATE OR REPLACE Function insertarestudiante (codigo integer, nombre varchar, apellidos varchar, codigocurso integer, codigoprograma integer, semestreid integer) returns void

AS

$$

insert into estudiante(codigo,nombre,apellidos,codigo\_curso,codigo\_programa,semestre\_id) values(codigo,nombre,apellidos,codigocurso,codigoprograma,semestreid);

$$

language SQL

CREATE OR REPLACE Function insertarcurso (codigo integer, nombre varchar, grupo integer) returns void

AS

$$

insert into curso(codigo,nombre,grupo) values (codigo, nombre,grupo);

$$

language SQL

CREATE OR REPLACE Function insertarprofesor (identificacion integer, nombre varchar, apellidos varchar) returns void

AS

$$

insert into profesor(identificacion,nombre,apellidos) VALUES(identificacion, nombre, apellidos);

$$

language SQL

CREATE OR REPLACE Function insertarasistencia (idasist integer, asiste boolean, idclase integer, codigoest integer) returns void

AS

$$

insert into asistencia(id\_asistencia,asiste,id\_clase,codigo\_estudiante) values (idasist, asiste, idclase, codigoest);

$$

language SQL

Alter table clase add column foto bytea;

CREATE OR REPLACE Function insertarclase (idc integer, fecha date, identificacion integer, horario integer, curso integer, foto bytea) returns void

AS

$$

insert into clase(id\_clase,fecha,identificacion\_profesor,horario\_id,codigo\_curso,foto) values (idc,fecha,identificacion,horario,curso,foto);

$$

language SQL

CREATE OR REPLACE Function insertarsemestre (semestre\_id integer, numero integer) returns void

AS

$$

insert into semestre(semestre\_id, numero) values (semestre\_id, numero);

$$

language SQL

CREATE OR REPLACE Function insertarprograma (codigo integer, nombre varchar, jornada varchar) returns void

AS

$$

insert into programa(codigo, nombre, jornada) values (codigo, nombre, jornada);

$$

language SQL