



# PROYECTO FINAL BASE DE DATOS II



**Base de Datos II**

**Unifranz sede el ALTO - 2022**

**Integrantes del equipo:**

**1. Delgado Mamani Christian Miguel**

# 1. Introducción.

## **Especificar cuál es el objetivo del proyecto**

El objetivo del proyecto es estructurar una base de datos para el registro y modificación de datos académicos ("notas, tutores, materias, docentes") de forma eficiente

# 2. Diseño de la base de Datos.

Dada la situación de almacenar y llenar los datos del estudiante se creó esta base de datos para poder actualizarlos en el registro académico

## 2.1. Análisis y definición de Tablas.

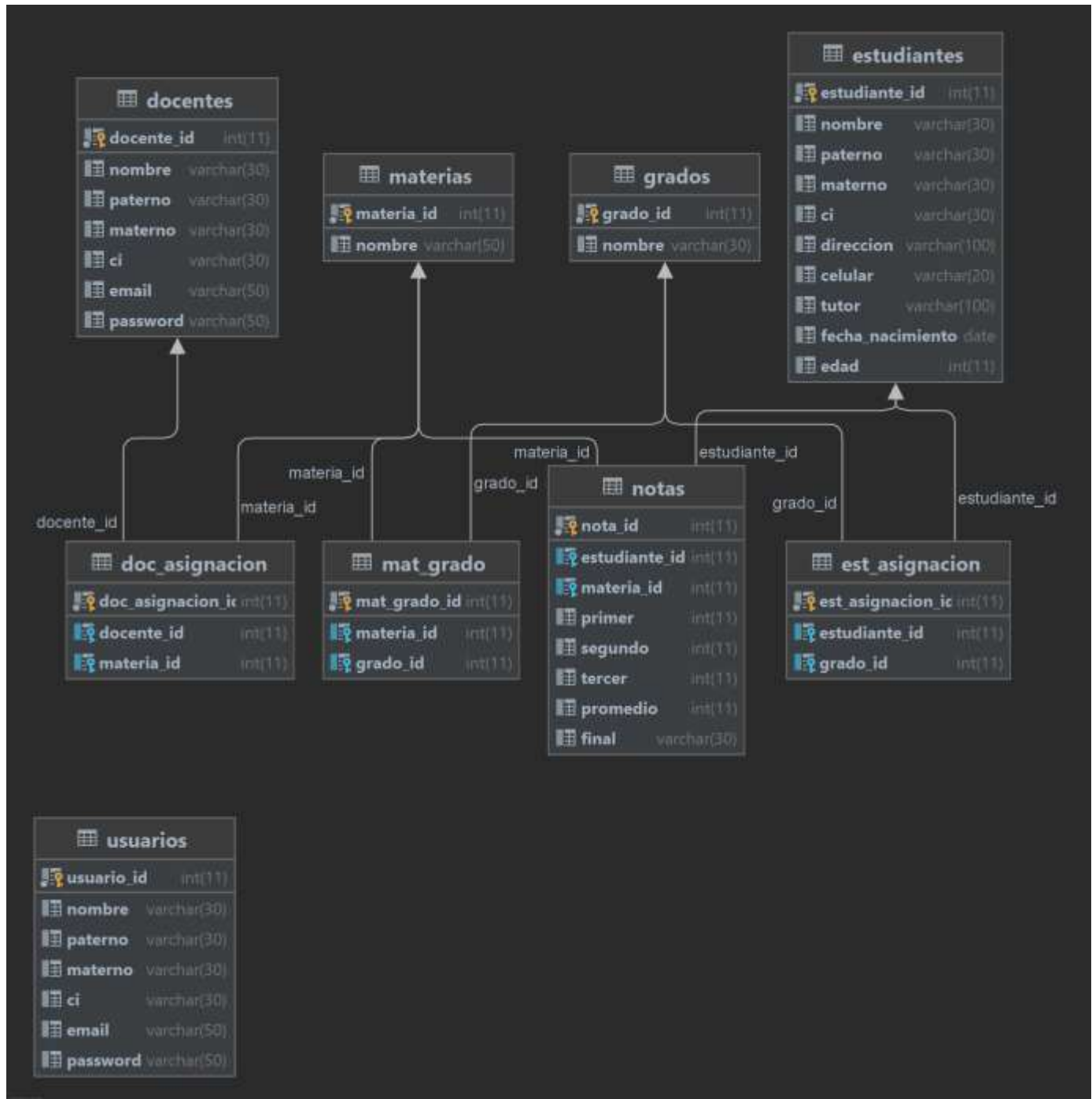
NOMBRE DE LA TABLA	Descripción
Usuarios	Administradores de la base de datos
Docentes	Almacena los datos del maestro y materia impartida
Estudiantes	Esta tabla se relaciona con la tabla estudiantes y materias para así hacer el control de las actividades del estudiante
Docente asignación	Esta tabla se relaciona con la tabla docente y materia.
Materia	Esta tabla almacena los datos de las materias de los estudiantes
Asignación	Almacena los id de estudiantes, materias, docentes
Notas	Almacena todos los datos de materia, estudiantes
Usuarios	<pre>create table usuarios(   usuario_id integer auto_increment primary key,   nombre varchar(30),   paterno varchar(30),   materno varchar(30),   ci varchar(30),   email varchar(50),   password varchar(50) );</pre>

## 2.2. Diseño de la Base de Datos.

est_asignacion	<pre>create table est_asignacion(   est_asignacion_id integer auto_increment primary key,   estudiante_id integer,   grado_id integer,   foreign key (estudiante_id) references estudiantes(estudiante_id),   foreign key (grado_id) references grados(grado_id) );</pre>
----------------	---

DOC_ASIGNACION	<pre> create table doc_asignacion(     doc_asignacion_id integer auto_increment primary key,     docente_id integer,     materia_id integer,     foreign key (docente_id) references docentes(docente_id),     foreign key (materia_id) references materias(materia_id) ); </pre>
DOCENTES	<pre> create table docentes(     docente_id integer auto_increment primary key,     nombre varchar(30),     paterno varchar(30),     materno varchar(30),     ci varchar(30),     email varchar(50),     password varchar(50) ); </pre>
NOTAS	<pre> create table notas(     nota integer IDENTITY(1,1) primary key,     estudiante_id integer null,     doc_asignacion_id integer null,     primer integer null,     segundo integer null,     tercer integer null,     promedio integer default 0,     foreign key (estudiante_id) references estudiantes(estudiante_id),     foreign key (doc_asignacion_id) references doc_asignacion(doc_asignacion_id) ); </pre>
ESTUDIANTES	<pre> create table estudiantes(     estudiante_id integer auto_increment primary key,     nombre varchar(30),     paterno varchar(30),     materno varchar(30),     ci varchar(30),     direccion varchar(100),     celular varchar(20),     tutor varchar(100),     fecha_nacimiento date,     edad integer ); </pre>
MATERIAS	<pre> create table materias(     materia_id integer auto_increment primary key,     nombre varchar(50) ); </pre>

### 2.2.1 Modelo lógico de la Base de Datos.



### 3. Funciones aplicadas

```
/* Trigger para saber que edad tiene un estudiante */
CREATE TRIGGER calcularEdad BEFORE INSERT
  ON estudiantes
  FOR EACH ROW
  BEGIN
    SET NEW.edad = TIMESTAMPDIFF(YEAR,NEW.fecha_nacimiento,CURDATE());
  END;

CREATE TRIGGER calcularEdadUpdate BEFORE UPDATE
  ON estudiantes
  FOR EACH ROW
  BEGIN
    SET NEW.edad = TIMESTAMPDIFF(YEAR,NEW.fecha_nacimiento,CURDATE());
  END;

/* Nota */
/* Al insertar datos se valida solo la primera nota y materia */
CREATE TRIGGER insertNota BEFORE INSERT
  ON notas
  FOR EACH ROW
  BEGIN
    SET NEW.segundo = 0;
    SET NEW.tercer = 0;
    SET NEW.promedio = 0;
    SET NEW.final = 0;
  END;

/* Trigger Validación de Segunda y Tercera Nota */
/* Segunda Nota: se verifica si ya se tiene la Primera nota registrada */
CREATE TRIGGER updateNotaSegundo BEFORE UPDATE
  ON notas
  FOR EACH ROW
  BEGIN
    IF OLD.primer = '' AND OLD.primer = 0 THEN
      SET NEW.segundo = 0;
    END IF;
  END;

/* Tercera Nota: se verifica si ya se tiene la Segunda nota registrada */
/* Trigger Promedio Final */
CREATE TRIGGER updateNotaTercera BEFORE UPDATE
  ON notas
  FOR EACH ROW
  BEGIN
    IF OLD.primer = '' AND OLD.primer = 0 AND OLD.segundo = '' AND OLD.segundo = 0 THEN
      SET NEW.tercer = 0;
    ELSE
      SET NEW.promedio = (NEW.primer + NEW.segundo + NEW.tercer)/3;
      IF NEW.promedio > 0 AND NEW.promedio < 40 THEN
        SET NEW.final = 'Bajo';
      END IF;
      IF NEW.promedio > 39 AND NEW.promedio < 70 THEN
        SET NEW.final = 'Medio';
      END IF;
    END IF;
  END;
```

```
END IF;  
IF NEW.promedio > 69 AND NEW.promedio < 90 THEN  
    SET NEW.final = 'Intermedio';  
END IF;  
IF NEW.promedio > 89 THEN  
    SET NEW.final = 'Excelente';  
END IF;  
  
END IF;  
END;
```

## 4. Conclusión

Según lo avanzado se logró estructurar e implementar la base de datos para el registro de datos académicos tanto para docentes y alumnado. Para así hacer el llenado de las distintas tablas requeridas por el docente

### Notas a considerarse:

- **En github crear una nueva carpeta de nombre PROYECTO\_FINAL.**
- **Los documentos que deben estar presente en github son los siguientes.**
  - **Informe (documento solicitado en esta documentación)**
  - **Un archivo excel(u otro) con todos los participantes (el título del archivo debe ser el nombre del grupo)**
  - **Diagrama entidad relación (Imagen).**
  - **Diagrama modelo lógico (Imagen).**
  - **Archivo .sql con todo el código sql de la Base de Datos.**