

# Universidad- Proyecto Curso SQL

## Introducción

Este proyecto trata de la creación de una base de datos de una universidad para facilitar y agilizar la administración de la misma. Esta base de datos contiene información sobre facultades, materias, profesores, alumnos y aulas.

Elegí este proyecto inspirándome en mi universidad y su sistema de administración que es bastante defectuoso y lento, por lo tanto, la idea de mi proyecto es hacer un sistema más rápido para la gestión de los objetos que componen a una universidad.

## Objetivo

El objetivo principal de este proyecto es la optimización del sistema actual de gestión de recursos y seguimiento académico de los alumnos pertenecientes a cada facultad de la universidad, facilitando el acceso a la información que se requiera por parte de docentes o alumnos y además brindando datos útiles para la administración de cada facultad.

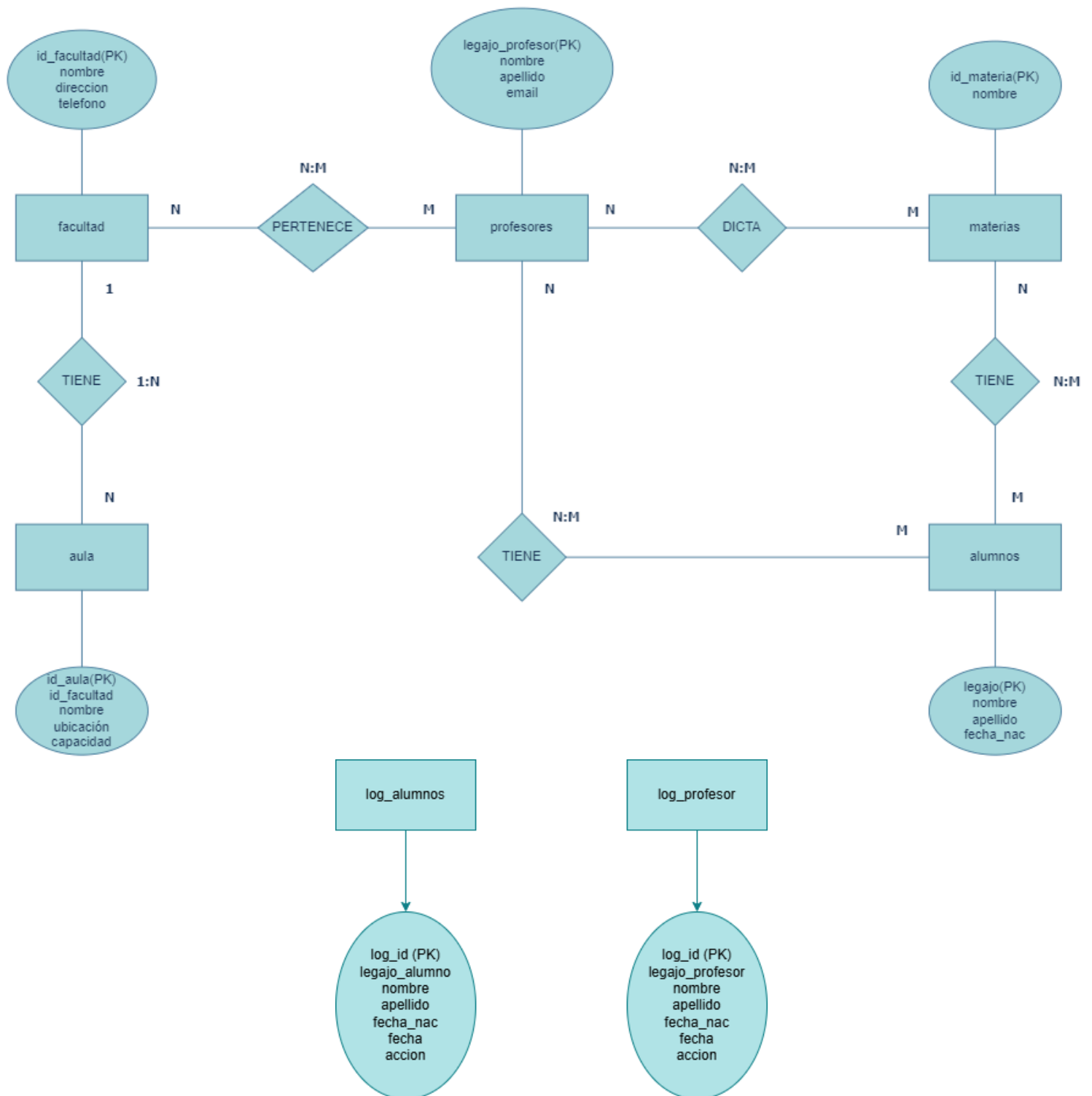
## Situación Problemática

Los problemas observados en el sistema actual de la universidad son la lentitud de las gestiones y solicitudes de certificados solicitados por profesores y alumnos, además de la mala utilización de los espacios de las facultades pertenecientes a la universidad. Estos problemas generan malestar en los empleados, alumnos y profesores de la universidad y pueden ser solucionados con la creación de una base de datos organizada y optimizada.

## Modelo de Negocio

Esta base de datos no solo mejora la operatividad de la universidad al gestionar eficazmente sus recursos, sino que también potencia su capacidad para ofrecer un servicio educativo de alta calidad. Facilita la toma de decisiones basada en datos y, por lo tanto, contribuye a la misión de la universidad de ser una institución educativa moderna y accesible.

## Diagrama de Entidad - Relación



## Listado de Tablas con descripción de Estructura

### 1. log\_alumnos

- log\_id: INT, Identificador único del log.
- legajo\_alumno: INT, Identificador del alumno relacionado.
- nombre: VARCHAR(255), Nombre del alumno.
- apellido: VARCHAR(255), Apellido del alumno.
- fecha\_nac: DATE, Fecha de nacimiento del alumno.
- fecha: DATETIME, Fecha y hora del registro de la acción.
- accion: VARCHAR(50), Descripción de la acción realizada.

### 2. log\_alumnos

- log\_id: INT, Identificador único del log.
- legajo\_profesor: INT, Identificador del alumno relacionado.
- nombre: VARCHAR(255), Nombre del alumno.
- apellido: VARCHAR(255), Apellido del alumno.
- email: VARCHAR(255), Mail del profesor.
- fecha: DATETIME, Fecha y hora del registro de la acción.
- accion: VARCHAR(50), Descripción de la acción realizada.

### 3. facultad

- id\_facultad: INT, Identificador único de la facultad.
- nombre: VARCHAR(255), Nombre de la facultad.
- direccion: VARCHAR(255), Dirección de la facultad.
- telefono: VARCHAR(30), Teléfono de contacto de la facultad.

### 4. aula

- id\_aula: INT, Identificador único del aula.
- id\_facultad: INT, Identificador de la facultad a la que pertenece el aula.
- nombre: VARCHAR(255), Nombre del aula.
- ubicacion: VARCHAR(255), Ubicación física del aula.
- capacidad: INT, Capacidad máxima del aula.

### 5. profesores

- legajo\_profesor: INT, Identificador único del profesor.
- nombre: VARCHAR(255), Nombre del profesor.
- apellido: VARCHAR(255), Apellido del profesor.
- email: VARCHAR(255), Correo electrónico del profesor.

### 6. profesores\_facultad

Nombre: Patricio Ferreyra

- legajo\_profesor: INT, Identificador del profesor.
- id\_facultad: INT, Identificador de la facultad donde enseña el profesor.

#### 7. materias

- id\_materia: INT, Identificador único de la materia.
- nombre: VARCHAR(255), Nombre de la materia.

#### 8. profesores\_materias

- legajo\_profesor: INT, Identificador del profesor que imparte la materia.
- id\_materia: INT, Identificador de la materia impartida.

#### 9. alumnos

- legajo\_alumno: INT, Identificador único del alumno.
- nombre: VARCHAR(255), Nombre del alumno.
- apellido: VARCHAR(255), Apellido del alumno.
- fecha\_nac: DATE, Fecha de nacimiento del alumno.

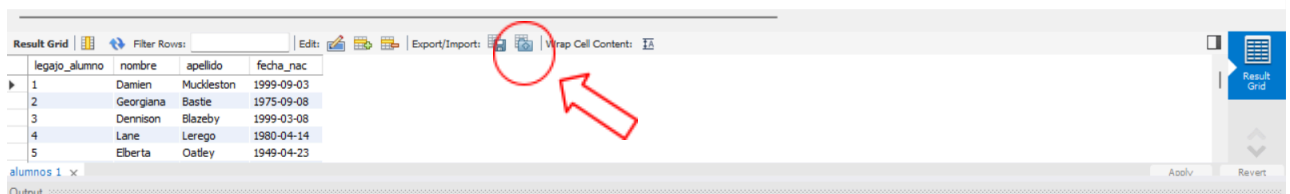
#### 10. materias\_alumnos

- legajo\_alumno: INT, Identificador del alumno que está inscrito en la materia.
- id\_materia: INT, Identificador de la materia en la que está inscrito el alumno.

## Inserción de Datos

La inserción de los datos de las tablas fue realizada por importación para facilitar la creación de los registros. El proceso fue el siguiente:

1.



2.



3.

**Table Data Import**

**Select Destination**

**Select destination table and additional options.**

☒ Use existing table: universidad.alumnos

☐ Create new table: universidad , TABLA\_AULAS

☐ Truncate table before import

4.

**Configure Import Settings**

Detected file format: csv

Encoding: utf-8

**Columns:**

Source Column	Dest Column
<input checked="" type="checkbox"/> id_aula	legajo_alumn
<input checked="" type="checkbox"/> id_facultad	nombre
<input checked="" type="checkbox"/> nombre	nombre
<input checked="" type="checkbox"/> ubicacion	fecha_nac
<input checked="" type="checkbox"/> capacidad	legajo_alumn

id_aula	id_facultad	nombre	ubicacion	capacidad
1	6	Colville Wil...	3-4-4	34
2	1	Rough Can...	2-4-1	29
3	1	Yellow Trite...	1-4-8	23
4	5	Eastern Lea...	3-2-8	43
5	2	Fewleaf Spi...	4-4-7	31

Siguiendo este proceso, complete todos los datos de las tablas de mi BASE DE DATOS.

## Herramientas y Tecnologías usadas

Para la creación de tablas y diseño de la base de datos se utilizó MySQL como sistema de gestión de base de datos y, para la creación y planificación del DER, se utilizó la página "DRAW.IO".

## Futuras Líneas

En el futuro planeo agregar un apartado de calificaciones para también controlar las evaluaciones de los alumnos como también sus notas y materias cursadas, aprobadas y por cursar.

## Funciones

Las funciones que cree son las siguientes:

- `cant_materias()`: Recibe el legajo del alumno y devuelve la cantidad de materias en las que el alumno está inscrito.
- `edad_persona()`: Recibe la fecha de nacimiento de una persona para determinar automáticamente la edad de la misma al compararla con la fecha actual y devuelve el número calculado.

## Vistas

Las vistas que cree son las siguientes:

- `cant_alumnos_profesores_vw`: Esta vista tiene la cantidad de alumnos que posee cada profesor y sirve para que las cátedras de cada materia de la facultad sepan la cantidad de alumnos que hay por comisión.
- `cant_profesores_facultad_vw`: Esta vista tiene la cantidad de profesores que posee cada facultad y sirve para reunir la información sobre la cantidad de profesores y presentarla de manera simplificada.

- `cant_alumnos_facultad_vw`: Esta vista tiene la cantidad de alumnos que posee cada facultad y sirve para tener un control de la cantidad de alumnos y controlar el funcionamiento de cada facultad.
- `cant_materias_alumno_vw`: Esta vista tiene la cantidad de materias en las que se inscribió cada alumno de la facultad y sirve para saber la cantidad de materias que el alumno esta cursando en el momento.
- `cant_aulas_facultad_vw`: Esta vista tiene la cantidad de aulas que posee cada facultad y sirve para dar a conocer el espacio disponible de las facultades y controlar su eficiencia e higiene.
- `info_facultades_vw`: Esta es una vista que reúne todas las vistas elaboradas previamente y presenta la información de cada facultad de manera simple y concentrada. Sirve para reunir toda la información importante de cada facultad en una misma vista.

## Stored Procedures

Los SP que cree son los siguientes:

- `inscribir_alumno()`: Es un procedimiento para inscribir un alumno que recibe los datos del alumno y realiza un control previo a la inserción en la tabla `alumnos` para que los campos de nombre y apellido no sean campos vacíos.
- `lista_alumnos()`: Es un procedimiento que recibe el legajo del profesor y devuelve la lista de alumnos que tiene con todos sus datos.

## Trigger

Los Triggers que cree son los siguientes:

- `log_insertar_alumno`: se ejecuta después de realizar una inserción en la tabla `alumnos` para tener un control de los cambios realizados en la misma. Los datos que se obtienen de cada inserción se almacenan en la tabla `"log_alumnos"` la que, además de almacenar los datos insertados en la tabla `alumnos`, guarda la fecha y hora de la operación y el tipo.
- `log_insertar_profesor`: Funciona de la misma manera que el Trigger de la inserción de alumnos solo que en este caso los datos de las inserciones de los datos se guardan en la tabla `"log_profesores"`.