

Trabajo Práctico Final - Base de Datos

Integrantes: Guemes Mario, Llampá Paola, Casimiro Camila, Benitez Fidel

Índice

| | |
|-------------------------------------------------|-----------|
| Índice | 1 |
| Introducción | 1 |
| Objetivos | 2 |
| Alcance y Limitaciones | 2 |
| Descripción | 4 |
| Modelo Entidad-Relación Extendido (MERE) | 6 |
| Análisis (Consultas) | 6 |
| Conclusión | 10 |

Introducción

El presente trabajo tiene como objetivo desarrollar y documentar una base de datos relacional destinada a la gestión integral del funcionamiento académico de una institución universitaria. El sistema fue pensado para organizar información vinculada a facultades, carreras, planes de estudio, materias, comisiones, estudiantes, docentes, calificaciones, asistencia y matrículas. Toda la estructura se diseñó priorizando la coherencia, la integridad de los datos y la posibilidad de escalar el sistema en futuras etapas.

La base de datos busca representar cómo operan las universidades: los estudiantes se inscriben en una carrera, cursan materias dentro de distintas comisiones, rinden evaluaciones, pagan matrículas y avanzan en su trayectoria académica. Al mismo tiempo, los docentes se asignan a comisiones, se registran horarios, aulas y correlatividades entre materias. Contar con un sistema de este tipo facilita la gestión administrativa y académica, permite generar reportes y brinda una base sólida para sistemas más complejos como plataformas de inscripciones online o sistemas de analíticas académicas.

Objetivos

El trabajo tiene los siguientes objetivos:

- **Diseño lógico y físico de la base de datos:**
Crear un modelo de datos completo y ordenado, respetando las reglas de una institución académica real. Para ello se utilizaron claves primarias, foráneas, restricciones CHECK, campos UNIQUE y relaciones con cardinalidades adecuadas.
- **Normalización y consistencia:**
Asegurar que la base de datos esté en Tercera Forma Normal (3FN) para evitar redundancias innecesarias, incoherencias y facilitar futuras modificaciones.
- **Simulación de datos reales:**
Generar registros de ejemplo realistas para todas las tablas (estudiantes, carreras, materias, comisiones, profesores, calificaciones, etc.) de manera que la base de datos pueda probarse y utilizarse inmediatamente.
- **Preparación para consultas y reportes:**
Dejar la base estructuralmente lista para realizar consultas SQL complejas (por ejemplo, estudiantes con materias aprobadas, promedio por alumno, asistencia, profesores por comisión, deuda de matrículas).

Alcance y Limitaciones

Alcances

La base de datos desarrollada permite gestionar la información académica y administrativa de una universidad.

Entre sus principales alcances se destacan:

- **Gestión académica:** permite registrar facultades, carreras, planes de estudio, materias, correlatividades, comisiones, horarios y periodos lectivos.
- **Administración de estudiantes:** se pueden almacenar datos personales, inscripciones a cursadas, calificaciones, asistencias y matrícula anual.

- **Seguimiento docente:** incluye la asignación de profesores a comisiones y el cálculo de su carga horaria total.
- **Control de infraestructura:** contempla aulas, su capacidad y estado, permitiendo analizar la utilización de espacios.
- **Análisis y reportes:** mediante consultas SQL es posible obtener estadísticas sobre rendimiento académico, asistencia, aprobación de materias y distribución docente entre otras .
- **Integridad referencial:** el uso de claves foráneas garantiza la coherencia de los datos y evita la eliminación o modificación de información relacionada.
- **Escalabilidad:** la estructura puede ampliarse fácilmente para incorporar nuevas funcionalidades, como evaluaciones más detalladas, registros administrativos o reportes financieros.

En resumen, el modelo relacional propuesto cubre los aspectos esenciales de la administración universitaria y sirve como base para futuros sistemas de gestión educativa o plataformas académicas.

Limitaciones

A pesar de su alcance amplio, el diseño presenta ciertas limitaciones que deben tenerse en cuenta:

- **Datos generados artificialmente:** la información contenida en las tablas fue creada de manera automática mediante herramientas de inteligencia artificial, por lo que puede contener errores, inconsistencias o situaciones que no se ajustan a la realidad. Los nombres, fechas, correos, calificaciones y relaciones entre entidades son ejemplos ficticios que solo buscan simular un entorno real para fines académicos.
- **Simplificación de procesos administrativos:** las tablas de matrícula, inscripción y cursado se representan de forma general, sin incluir particularidades como reincorporaciones, equivalencias o situaciones excepcionales de alumnos.

- **Dependencia del contexto educativo:** las estructuras y relaciones fueron pensadas para un modelo universitario genérico; su aplicación directa en otra institución podría requerir ajustes.

Descripción

El diseño abarca varias áreas clave de la institución:

1. Estructura académica

- **FACULTAD:** representa cada unidad académica (ej., Facultad de Ingeniería).
- **CARRERA:** depende de una facultad y define una propuesta educativa concreta (ej., Ingeniería en Sistemas).
- **PLANESTUDIO:** especifica qué materias integran una carrera y en qué año se implementó ese plan. Cada carrera puede tener varios planes (vigentes o históricos).

2. Materias y correlatividades

- **MATERIA:** contiene todas las materias posibles, con su nombre, código, horas semanales y tipo (obligatoria, optativa o electiva).
- **MATERIA_CARRERA:** relaciona las materias con el plan de estudios y la carrera y define en qué año y cuatrimestre se cursan.
- **CORRELATIVA:** establece qué materias deben aprobarse o regularizarse antes de cursar o rendir otra.

3. Gestión de cursadas

- **COMISION:** representa cada grupo de cursada de una materia, con modalidad, aula, cupo máximo y cupo ocupado.
- **HORARIO:** añade días y horas semanales para esa comisión.

- **ASIGNACION_DOCENTE:** indica qué profesor dicta esa comisión y cuáles su rol (Titular, Adjunto, Ayudante, etc.).

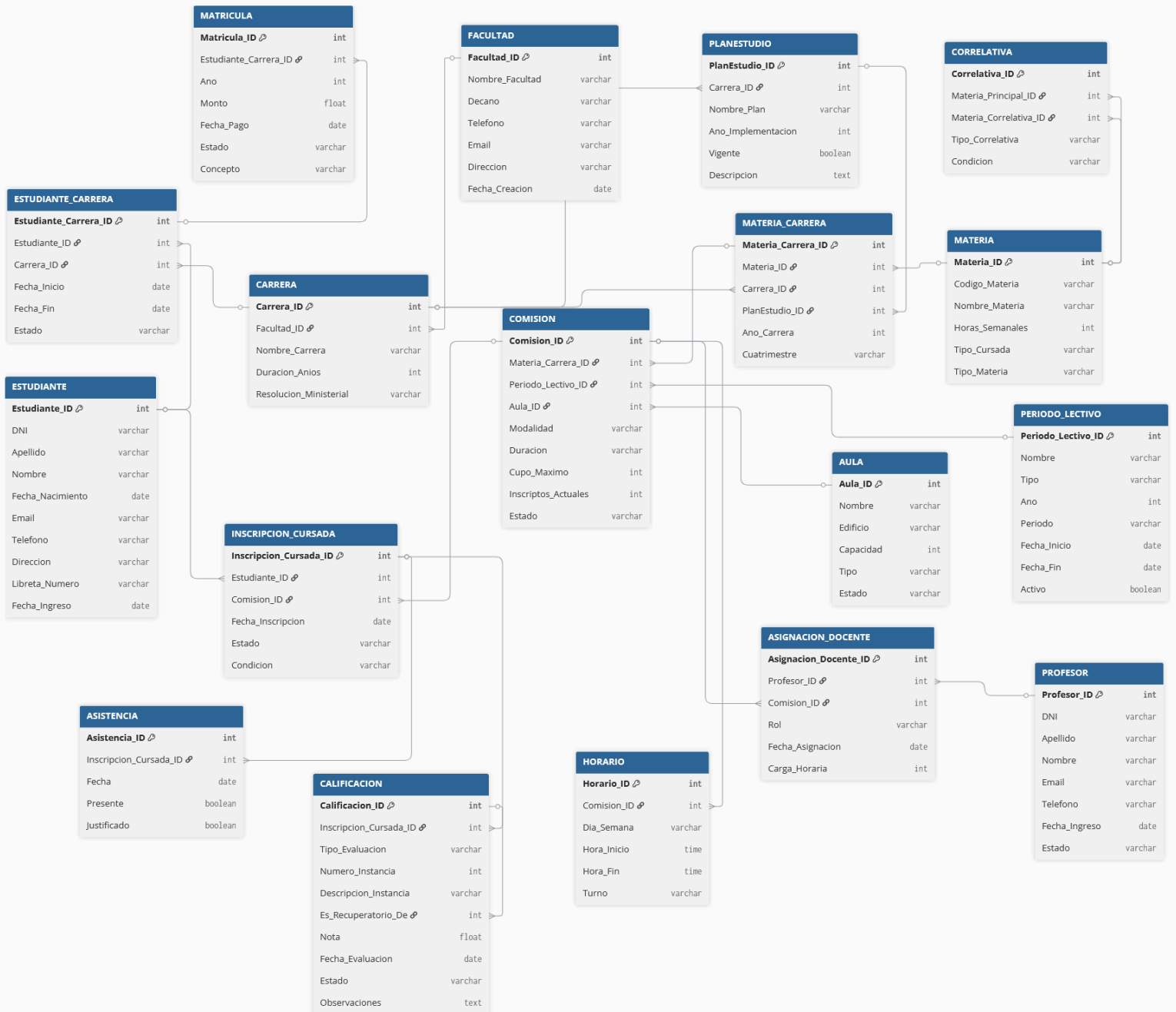
4. Estudiantes y su trayectoria

- **ESTUDIANTE:** almacena los datos personales de cada alumno, incluyendo DNI, fecha de nacimiento, contacto, libreta universitaria y fecha de ingreso.
- **ESTUDIANTE_CARRERA:** registra en qué carrera está inscripto cada estudiante, con fecha de inicio, posible egreso o abandono.
- **INSCRIPCION_CURSADA:** representa la inscripción del estudiante a una comisión específica de una materia.

5. Evaluaciones, asistencia y matrícula

- **CALIFICACION:** registra todas las notas de parciales, finales, TPs, recuperatorios, etc.
- **ASISTENCIA:** lleva el registro de presentismo por fecha, incluyendo ausencias justificadas.
- **MATRICULA:** registra el pago o deuda de la matrícula anual, con estado (Pagado, Pendiente, Moroso).

Modelo Entidad-Relación Extendido (MERE)



Análisis (Consultas)

Listar todos los estudiantes

SELECT Estudiante_ID, Nombre || ' ' || Apellido as Nombre_Completo, Email

FROM ESTUDIANTE

ORDER BY Estudiante_ID

Obtener los datos de un estudiante por DNI

SELECT *

FROM ESTUDIANTE

WHERE DNI = '22000000'

Contar cuántos estudiantes hay por carrera

SELECT c.Nombre_Carrera, COUNT(ec.Estudiante_Carrera_ID) AS
Cantidad_Estudiantes

FROM ESTUDIANTE_CARRERA ec

JOIN CARRERA c ON ec.Carrera_ID = c.Carrera_ID

GROUP BY c.Carrera_ID

ORDER BY Cantidad_Estudiantes DESC

Listar comisiones activas del periodo lectivo actual

SELECT com.Comision_ID, m.Nombre_Materia, c.Nombre_Carrera

FROM COMISION com

JOIN MATERIA_CARRERA mc ON com.Materia_Carrera_ID =
mc.Materia_Carrera_ID

JOIN MATERIA m ON mc.Materia_ID = m.Materia_ID

JOIN CARRERA c ON mc.Carrera_ID = c.Carrera_ID

JOIN PERIODO_LECTIVO p ON com.Periodo_Lectivo_ID =
p.Periodo_Lectivo_ID

WHERE p.Activo = 1

ORDER BY c.Nombre_Carrera

Alumnos inscriptos en una comisión específica (por Comision_ID)

SELECT ic.Inscripcion_Cursada_ID, ic.Estudiante_ID, e.Nombre || ' ' || e.Apellido
AS Estudiante, ic.Estado

FROM INSCRIPCION_CURSADA ic

JOIN ESTUDIANTE e ON ic.Estudiante_ID = e.Estudiante_ID

WHERE ic.Comision_ID = 10

ORDER BY Estudiante

Promedio de notas por estudiante

SELECT e.Estudiante_ID, e.Nombre || ' ' || e.Apellido AS Estudiante,
ROUND(AVG(ca.Nota), 2) AS Promedio_General, COUNT(ca.Calificacion_ID)
AS CantNotas

FROM ESTUDIANTE e

JOIN INSCRIPCION_CURSADA ic ON e.Estudiante_ID = ic.Estudiante_ID

JOIN CALIFICACION ca ON ic.Inscripcion_Cursada_ID =
ca.Inscripcion_Cursada_ID

GROUP BY e.Estudiante_ID

HAVING CantNotas > 0

ORDER BY Promedio_General DESC

Estudiantes por carrera con estado actual

SELECT c.Nombre_Carrera, f.Nombre_Facultad, COUNT(*) as
Total_Estudiantes, SUM(CASE WHEN ec.Estado = 'Activo' THEN 1 ELSE 0
END) as Activos, SUM(CASE WHEN ec.Estado = 'Egresado' THEN 1 ELSE 0
END) as Egresados

FROM ESTUDIANTE_CARRERA ec

JOIN CARRERA c ON ec.Carrera_ID = c.Carrera_ID

JOIN FACULTAD f ON c.Facultad_ID = f.Facultad_ID

GROUP BY c.Carrera_ID

ORDER BY Total_Estudiantes DESC

Materias con mayor número de comisiones y cupo total disponible

SELECT m.Materia_ID, m.Nombre_Materia, COUNT(com.Comision_ID) AS
Cant_Comisiones, SUM(com.Cupo_Maximo) AS Cupo_Total,
SUM(com.Inscriptos_Actuales) AS Inscriptos_Totales

FROM MATERIA m

JOIN MATERIA_CARRERA mc ON m.Materia_ID = mc.Materia_ID

LEFT JOIN COMISION com ON mc.Materia_Carrera_ID =
com.Materia_Carrera_ID

GROUP BY m.Materia_ID

ORDER BY Cant_Comisiones DESC, Inscriptos_Totales DESC

Profesores con más comisiones asignadas

SELECT p.Profesor_ID, p.Nombre || ' ' || p.Apellido AS Profesor,
COUNT(ad.Comision_ID) AS Comisiones_Asignadas

FROM PROFESOR p

JOIN ASIGNACION_DOCENTE ad ON p.Profesor_ID = ad.Profesor_ID

GROUP BY p.Profesor_ID

ORDER BY Comisiones_Asignadas DESC

LIMIT 10

Obtener la lista de correlativas (prerrequisitos) de una materia dada

```
SELECT m.Materia_ID, m.Nombre_Materia, mc.Materia_Correlativa_ID AS  
Correlativa_ID, m2.Nombre_Materia AS Correlativa_Nombre,  
mc.Tipo_Correlativa, mc.Condicion  
  
FROM CORRELATIVA mc  
  
JOIN MATERIA m ON mc.Materia_Principal_ID = m.Materia_ID  
  
JOIN MATERIA m2 ON mc.Materia_Correlativa_ID = m2.Materia_ID  
  
WHERE m.Materia_ID = 12
```

Análisis de ocupación de aulas

```
SELECT a.Nombre as Aula, a.Edificio, a.Capacidad, COUNT(co.Comision_ID)  
as Comisiones_Asignadas, SUM(co.Cupo_Maximo) as Cupo_Total_Asignado  
  
FROM AULA a  
  
LEFT JOIN COMISION co ON a.Aula_ID = co.Aula_ID  
  
GROUP BY a.Aula_ID  
  
ORDER BY Comisiones_Asignadas DESC
```

Conclusión

Se logró construir una base de datos completa, coherente y adaptable al funcionamiento de una institución académica real. El modelo respeta las reglas y relaciones del sistema universitario, evitando redundancias y asegurando la integridad de los datos. Además, se cargaron datos de prueba suficientes para validar la estructura y dejar lista la base para consultas, reportes o integración con sistemas más grandes en el futuro.