

# Proyecto Final

Primera entrega

## 1. Descripción de la temática

---

### Introducción

El proyecto final tratará sobre la base de datos utilizada por una Universidad que posee clases presenciales, y diversas carreras, para registrar los hechos relacionados con asuntos académicos y de los alumnos inscriptos.

### Objetivos

Brindar un medio para la persistencia de los datos que el negocio requiera para la toma de decisiones y para la registración de todas las transacciones relevantes para la Universidad.

### Situación problemática

Una universidad cuenta con muchos flujos de información y muchas personas que interactúan constantemente para obtener un estado actualizado de diversa índole. El presente proyecto busca brindar una solución que permita a todos los actores conocer el estado actual de cierta información (cómo los resultados de exámenes y de materias cursadas, entre otros asuntos académicos), así como sobre la situación financiera de la universidad y de los alumnos, conociendo aquellos conceptos facturados, aquellos adeudados y aquellos abonados.

### Modelo de negocio

En esta universidad existen distintos actores:

- Alumnos
- Profesores

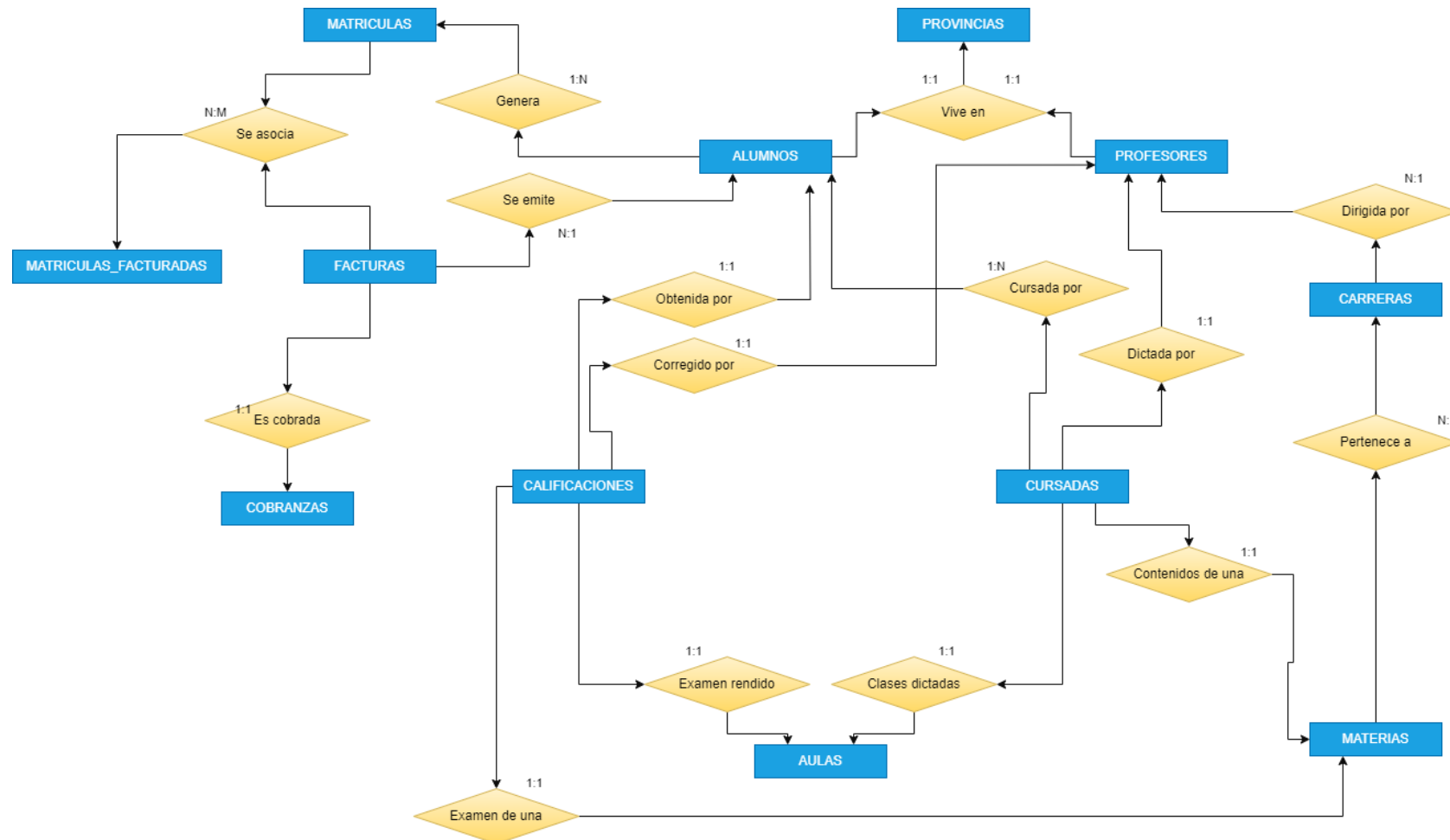
Los alumnos se inscriben en carreras para cursar un grupo de materias. Dichas materias son dictadas por profesores. Asimismo, al finalizar cada cursada, se obtiene un estado, que permitirá o no (según las reglas de negocio definidas), cursar las siguientes materias del plan de estudio.

Los alumnos por cada vez que rinden un examen, obtiene una calificación.

Para poder anotarse a cursar un semestre, los alumnos deberán abonar una matrícula. Cada matrícula (o grupo en caso que se desee abonar más de una), será facturada según la normativa vigente, y al ser cobradas, se emitirá un recibo interno al alumno, como constancia de pago.

También, se registraran en este modelo, distintos datos geográficos que permitirán tomar decisiones según diversas dimensiones, como las provincias o aulas de la universidad.

## 2. Diagramas entidad relación



**Nota:** Por cuestiones de espacio en la hoja, se eliminaron los atributos de cada tabla, sin embargo, se muestran en la sección siguiente.

### 3. Listado de tablas

calificaciones			
Tabla que registra las calificaciones obtenidas por los alumnos al rendir los exámenes.			
Nombre del campo	Abreviatura	Tipo de datos	Tipos de claves
ID Calificación	id	INT	PK
Alumno ID	alumno_legajo	INT	FK
Materia ID	materia_id	SMALLINT	FK
Profesor ID	profesor_legajo	MEDIUMINT	FK
Aula ID	aula_id	SMALLINT	FK
Fecha de examen	fecha	DATETIME	
Aprobado	aprobado	BOOLEAN	

cursadas			
Tabla que registra las inscripciones de alumnos al cursado de las materias.			
Nombre del campo	Abreviatura	Tipo de datos	Tipos de claves
ID Cursada	id	INT	PK
Alumno ID	alumno_legajo	INT	PK, FK
Materia ID	materia_id	SMALLINT	PK, FK
Profesor ID	profesor_legajo	MEDIUMINT	FK
Aula ID	aula_id	SMALLINT	FK
Fecha de inscripción	fecha	DATETIME	
Semestre del cursado	semestre	ENUM	PK
Periodo Lectivo (año)	per_lectivo	SMALLINT	PK

cursadas			
Tabla que registra las inscripciones de alumnos al cursado de las materias.			
Nombre del campo	Abreviatura	Tipo de datos	Tipos de claves
Condición al fin del cursado	condicion	ENUM	

matriculas			
Tabla que registra los tickets que los alumnos deben pagar por la matrícula del semestre en que se anotan a cursar.			
Nombre del campo	Abreviatura	Tipo de datos	Tipos de claves
ID Matricula	id	INT	PK
Alumno ID	alumno_legajo	INT	PK, FK
Fecha de inscripción	fecha	DATETIME	
Semestre del cursado	semestre	ENUM	PK
Periodo Lectivo (año)	per_lectivo	SMALLINT	PK
Importe	importe	NUMERIC	

facturas			
Tabla que registra los tickets que los alumnos deben pagar por la matrícula del semestre en que se anotan a cursar.			
Nombre del campo	Abreviatura	Tipo de datos	Tipos de claves
N° factura	factura	VARCHAR(13)	PK
Alumno ID	alumno_legajo	INT	FK
Fecha de emisión	fecha	DATETIME	
Tipo de factura	tipo_factura	ENUM	

### matriculas\_facturadas

Tabla intermedia que detalla las distintas matrículas incluidas en un mismo comprobante (factura).

Nombre del campo	Abreviatura	Tipo de datos	Tipos de claves
ID matricula facturada	id	INT	PK
Matricula ID	matricula_id	INT	FK
Factura ID	factura_id	VARCHAR(13)	FK

### cobranzas

Tabla que registra las cobranzas realizadas de las facturas emitidas.

Nombre del campo	Abreviatura	Tipo de datos	Tipos de claves
N° recibo	id	BIGINT	PK
Factura ID	factura_id	VARCHAR(13)	FK
Fecha de pago	fecha	DATETIME	
Medio de pago	medio_pago	ENUM	

### alumnos

Tabla de dimensión que registra los alumnos que se encuentran inscriptos en la universidad.

Nombre del campo	Abreviatura	Tipo de datos	Tipos de claves
ID Alumno	legajo	INT	PK
Nombre	nombre	VARCHAR(50)	
Apellido	apellido	VARCHAR(50)	
Fecha de nacimiento	fecha_nac	DATE	
Sexo	sexo	ENUM	
Email	email	VARCHAR(50)	
Telefono	telefono	BIGINT	

alumnos			
Tabla de dimensión que registra los alumnos que se encuentran inscriptos en la universidad.			
Nombre del campo	Abreviatura	Tipo de datos	Tipos de claves
Provincia	provincia_id	TINYINT	FK

profesores			
Tabla de dimensión que registra los profesores que imparten materias en la universidad.			
Nombre del campo	Abreviatura	Tipo de datos	Tipos de claves
ID Profesor	legajo	MEDIUMINT	PK
Nombre	nombre	VARCHAR(50)	
Apellido	apellido	VARCHAR(50)	
Fecha de nacimiento	fecha_nac	DATE	
Sexo	sexo	ENUM	
Email	email	VARCHAR(50)	
Telefono	telefono	BIGINT	
Profesión	profesion	VARCHAR(50)	
Tipo de jornada	tipo_jornada	ENUM	
Dirección	dirección	VARCHAR(50)	
Provincia	provincia_id	TINYINT	FK

materias			
Tabla de dimensión que registra las materias de cada carrera.			
Nombre del campo	Abreviatura	Tipo de datos	Tipos de claves
ID Materia	id	SMALLINT	PK
Nombre	nombre	VARCHAR(50)	

<b>materias</b>			
Tabla de dimensión que registra las materias de cada carrera.			
Nombre del campo	Abreviatura	Tipo de datos	Tipos de claves
Puntos que otorga	puntos	TINYINT	
Semestre en que se dicta	sem_dict	ENUM	
Carrera a la que pertenece	carrera_id	TINYINT	FK

<b>carreras</b>			
Tabla de dimensión que registra las carreras dictadas por la universidad.			
Nombre del campo	Abreviatura	Tipo de datos	Tipos de claves
ID Carrera	id	TINYINT	PK
Nombre	nombre	VARCHAR(50)	
Puntos necesarios	puntos	TINYINT	
Director de carrera	director	MEDIUMINT	FK
Plan de estudios vigente	plan_vigente	VARCHAR(20)	

<b>aulas</b>			
Tabla de dimensión que registra las aulas de la universidad.			
Nombre del campo	Abreviatura	Tipo de datos	Tipos de claves
ID Aula	id	SMALLINT	PK
Nombre	nombre	VARCHAR(20)	
Edificio	edificio	VARCHAR(20)	
Capacidad de personas	capacidad	SMALLINT	

provincias			
Tabla de dimensión que registra las provincias de Argentina.			
Nombre del campo	Abreviatura	Tipo de datos	Tipos de claves
ID Provincia	id	TINYINT	PK
Nombre	nombre	VARCHAR(20)	
Cantidad de habitantes	habitantes	BIGINT	



## 4. Archivo SQL

---

```
-- Creación esquema:
CREATE SCHEMA IF NOT EXISTS Universidad;
USE Universidad;

-- Creación de tablas:
CREATE TABLE IF NOT EXISTS PROVINCIAS (
    id TINYINT AUTO_INCREMENT,
    nombre VARCHAR(20) NOT NULL,
    habitantes BIGINT,
    PRIMARY KEY (id)
);

CREATE TABLE IF NOT EXISTS AULAS (
    id SMALLINT AUTO_INCREMENT,
    nombre VARCHAR(20),
    edificio VARCHAR(20),
    capacidad SMALLINT,
    PRIMARY KEY (id)
);

CREATE TABLE IF NOT EXISTS PROFESORES (
    legajo MEDIUMINT AUTO_INCREMENT,
    nombre VARCHAR(50) NOT NULL,
    apellido VARCHAR(50) NOT NULL,
    fecha_nac DATE,
    sexo ENUM('F', 'M', 'X'),
    email VARCHAR(50),
    telefono BIGINT,
    profesion VARCHAR(50),
    tipo_jornada ENUM('Full-Time', 'Part-Time'),
    dirección VARCHAR(50),
    provincia_id TINYINT,
    PRIMARY KEY (legajo),
    FOREIGN KEY (provincia_id) REFERENCES provincias(id)
);

CREATE TABLE IF NOT EXISTS ALUMNOS (
    legajo INT AUTO_INCREMENT,
    nombre VARCHAR(50) NOT NULL,
    apellido VARCHAR(50) NOT NULL,
    fecha_nac DATE,
    sexo ENUM('F', 'M', 'X'),
    email VARCHAR(50),
    telefono BIGINT,
    provincia_id TINYINT,
    PRIMARY KEY (legajo),
    FOREIGN KEY (provincia_id) REFERENCES provincias(id)
```

```
);

CREATE TABLE IF NOT EXISTS CARRERAS (
    id TINYINT AUTO_INCREMENT,
    nombre VARCHAR(50) NOT NULL,
    puntos TINYINT,
    director MEDIUMINT NOT NULL,
    plan_vigente VARCHAR(20),
    PRIMARY KEY (id),
    FOREIGN KEY (director) REFERENCES profesores(legajo)
);

CREATE TABLE IF NOT EXISTS MATERIAS (
    id SMALLINT AUTO_INCREMENT,
    nombre VARCHAR(50) NOT NULL,
    puntos TINYINT,
    sem_dict ENUM('1', '2'),
    carrera_id TINYINT NOT NULL,
    PRIMARY KEY (id),
    FOREIGN KEY (carrera_id) REFERENCES carreras(id)
);

CREATE TABLE IF NOT EXISTS MATRICULAS (
    id INT AUTO_INCREMENT,
    alumno_legajo INT,
    fecha DATETIME,
    semestre ENUM('1', '2'),
    per_lectivo SMALLINT,
    importe NUMERIC DEFAULT 0,
    PRIMARY KEY (id, alumno_legajo, semestre, per_lectivo),
    FOREIGN KEY (alumno_legajo) REFERENCES alumnos(legajo)
);

CREATE TABLE IF NOT EXISTS FACTURAS (
    factura VARCHAR(13),
    alumno_legajo INT NOT NULL,
    fecha DATETIME,
    tipo_factura ENUM('A', 'B'),
    PRIMARY KEY (factura),
    FOREIGN KEY (alumno_legajo) REFERENCES alumnos(legajo)
);

CREATE TABLE IF NOT EXISTS MATRICULAS_FACTURADAS (
    id INT AUTO_INCREMENT,
    matricula_id INT NOT NULL,
    factura_id VARCHAR(13) NOT NULL,
    PRIMARY KEY (id),
    FOREIGN KEY (matricula_id) REFERENCES matriculas(id),
    FOREIGN KEY (factura_id) REFERENCES facturas(factura)
);
```

```
CREATE TABLE IF NOT EXISTS COBRANZAS (  
    id BIGINT AUTO_INCREMENT,  
    factura_id INT NOT NULL,  
    fecha DATETIME,  
    medio_pago ENUM('Efectivo', 'Transferencia', 'Tarjeta de Crédito'),  
    PRIMARY KEY (id),  
    FOREIGN KEY (factura_id) REFERENCES facturas(factura)  
);
```

```
CREATE TABLE IF NOT EXISTS CURSADAS (  
    id INT AUTO_INCREMENT,  
    alumno_legajo INT NOT NULL,  
    materia_id SMALLINT NOT NULL,  
    profesor_legajo MEDIUMINT NOT NULL,  
    aula_id SMALLINT NOT NULL,  
    fecha DATETIME,  
    semestre ENUM('1', '2'),  
    per_lectivo SMALLINT,  
    condicion ENUM('Promoción', 'Regular', 'Libre'),  
    PRIMARY KEY (id, alumno_legajo, materia_id, semestre, per_lectivo),  
    FOREIGN KEY (alumno_legajo) REFERENCES alumnos(legajo),  
    FOREIGN KEY (materia_id) REFERENCES materias(id),  
    FOREIGN KEY (profesor_legajo) REFERENCES profesores(legajo),  
    FOREIGN KEY (aula_id) REFERENCES aulas(id)  
);
```

```
CREATE TABLE IF NOT EXISTS CALIFICACIONES (  
    id INT AUTO_INCREMENT,  
    alumno_legajo INT NOT NULL,  
    materia_id SMALLINT NOT NULL,  
    profesor_legajo MEDIUMINT NOT NULL,  
    aula_id SMALLINT NOT NULL,  
    fecha DATETIME,  
    aprobado BOOLEAN,  
    PRIMARY KEY (id),  
    FOREIGN KEY (alumno_legajo) REFERENCES alumnos(legajo),  
    FOREIGN KEY (materia_id) REFERENCES materias(id),  
    FOREIGN KEY (profesor_legajo) REFERENCES profesores(legajo),  
    FOREIGN KEY (aula_id) REFERENCES aulas(id)  
);
```

**Archivo:**



Coder  
Universidad.sql