Proyecto Final
SQL

Rodrigo Casas

CODERHOUSE

Índice

1. Introducción

Introducción

El presente proyecto final trata 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.

2. Objetivos

Objetivo

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.

3.
Situación problemática

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.

4.
Modelo de negocio

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.

Modelo de negocio

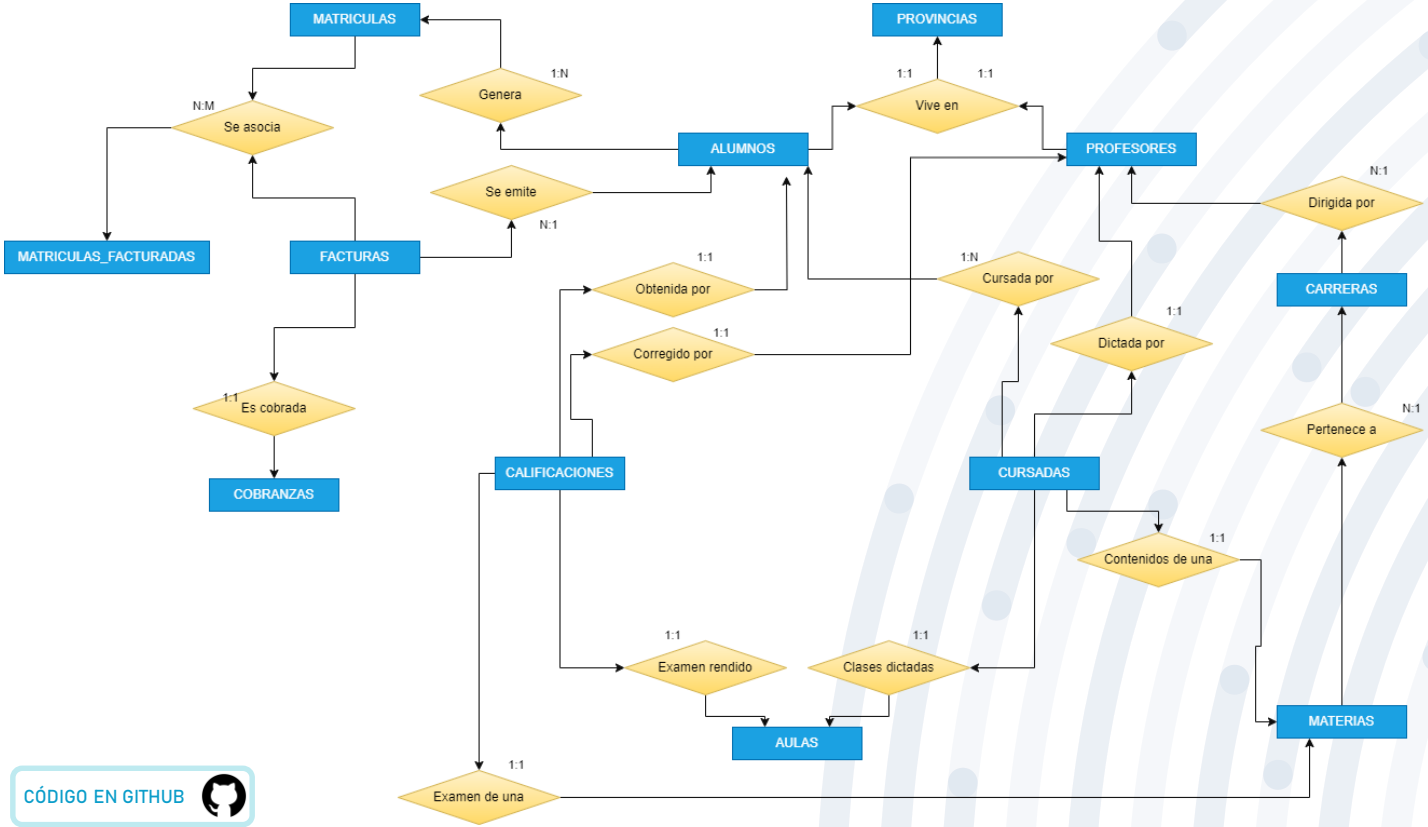
Los alumnos por cada vez que rinden un examen, obtiene una calificación.

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.

5. Diagrama de E-R

Diagrama de Entidad-Relación



CÓDIGO EN GITHUB

6. Listado de tablas

Calificaciones

CÓDIGO EN GITHUB

Tabla que registra las calificaciones obtenidas por los alumnos al rendir los exámenes.

Nombre	Abreviatura	Tipo de dato	Primary Key	Foreign Key
ID Calificación	id	INT	✓	
Alumno ID	alumno_legajo	INT		✓
Materia ID	materia_id	SMALLINT		✓
Profesor ID	profesor_legajo	MEDIUMINT		✓
Aula ID	aula_id	SMALLINT		✓
Fecha de examen	fecha	DATETIME		
Aprobado	aprobado	BOOLEAN		

Cursadas

CÓDIGO EN GITHUB

Tabla que registra las inscripciones de alumnos al cursado de las materias.

Nombre	Abreviatura	Tipo de dato	Primary Key	Foreign Key
ID Calificación	id	INT	✓	
Alumno ID	alumno_legajo	INT	✓	✓
Materia ID	materia_id	SMALLINT	✓	✓
Profesor ID	profesor_legajo	MEDIUMINT		✓
Aula ID	aula_id	SMALLINT		✓
Fecha de inscripción	fecha	DATETIME		
Semestre del cursado	semestre	ENUM	✓	
Periodo lectivo (año)	per_lectivo	SMALLINT	✓	
Condición al fin del cursado	condicion	ENUM		

Matrículas

CÓDIGO EN GITHUB

Tabla que registra los tickets que los alumnos deben pagar por la matrícula del semestre en que se anotan a cursar.

Nombre	Abreviatura	Tipo de dato	Primary Key	Foreign Key
ID Matricula	id	INT	✓	
Alumno ID	alumno_legajo	INT	✓	✓
Fecha de inscripción	fecha	DATETIME		
Semestre del cursado	semestre	ENUM	✓	
Periodo lectivo (año)	per_lectivo	SMALLINT	✓	
Importe	importe	NUMERIC		

Facturas

CÓDIGO EN GITHUB

Tabla que registra los tickets que los alumnos deben pagar por la matrícula del semestre en que se anotan a cursar.

Nombre	Abreviatura	Tipo de dato	Primary Key	Foreign Key
N° factura	factura	VARCHAR(13)	✓	
Alumno ID	alumno_legajo	INT		✓
Fecha de emisión	fecha	DATETIME		
Tipo de factura	tipo_factura	ENUM		

Matrículas facturadas

CÓDIGO EN GITHUB

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

Nombre	Abreviatura	Tipo de dato	Primary Key	Foreign Key
ID matricula facturada	id	INT	✓	
Matricula ID	matricula_id	INT		✓
Factura ID	factura_id	VARCHAR(13)		✓

Cobranzas

CÓDIGO EN GITHUB

Tabla que registra las cobranzas realizadas de las facturas emitidas.

Nombre	Abreviatura	Tipo de dato	Primary Key	Foreign Key
N° recibo	id	BIGINT	✓	
Factura ID	factura_id	VARCHAR(13)		✓
Fecha de pago	fecha	DATETIME		
Medio de pago	medio_pago	ENUM		

Alumnos

CÓDIGO EN GITHUB

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

Nombre	Abreviatura	Tipo de dato	Primary Key	Foreign Key
ID Alumno	legajo	INT	✓	
Nombre	nombre	VARCHAR(50)		
Apellido	apellido	VARCHAR(50)		
Fecha de nacimiento	fecha_nac	DATE		
Sexo	sexo	ENUM		
Email	Email	VARCHAR(50)		
Teléfono	telefono	BIGINT		
Provincia	provincia	TINYINT		✓

Profesores

CÓDIGO EN GITHUB

Tabla de dimensión que registra los profesores que imparten materias en la universidad.

Nombre	Abreviatura	Tipo de dato	Primary Key	Foreign Key
ID Profesor	legajo	MEDIUMINT	✓	
Nombre	nombre	VARCHAR(50)		
Apellido	apellido	VARCHAR(50)		
Fecha de nacimiento	fecha_nac	DATE		
Sexo	sexo	ENUM		
Email	Email	VARCHAR(50)		
Profesión	profesion	VARCHAR(50)		
Tipo de jornada	tipo_jornada	ENUM		
Dirección	direccion	VARCHAR(50)		
Provincia	provincia	TINYINT		✓

Materias

CÓDIGO EN GITHUB

Tabla de dimensión que registra las materias de cada carrera.

Nombre	Abreviatura	Tipo de dato	Primary Key	Foreign Key
ID Materia	id	SMALLINT	✓	
Nombre	nombre	VARCHAR(50)		
Puntos que otorga	puntos	TINYINT		
Semestre en que se dicta	sem_dict	ENUM		
Carrera a la que pertenece	carrera_id	TINYINT		✓

Carreras

CÓDIGO EN GITHUB

Tabla de dimensión que registra las carreras dictadas por la universidad.

Nombre	Abreviatura	Tipo de dato	Primary Key	Foreign Key
ID Carrera	id	TINYINT	✓	
Nombre	nombre	VARCHAR(50)		
Puntos necesarios	puntos	TINYINT		
Director de carrera	director	MEDIUMINT		✓
Plan de estudios vigente	plan_vigente	VARCHAR(20)		

Aulas

CÓDIGO EN GITHUB

Tabla de dimensión que registra las aulas de la universidad.

Nombre	Abreviatura	Tipo de dato	Primary Key	Foreign Key
ID Aula	id	SMALLINT	✅	
Nombre	nombre	VARCHAR(20)		
Edificio	edificio	VARCHAR(20)		
Capacidad de personas	capacidad	SMALLINT		

Provincias

CÓDIGO EN GITHUB

Tabla de dimensión que registra las provincias de Argentina.

Nombre	Abreviatura	Tipo de dato	Primary Key	Foreign Key
ID Provincia	id	TINYINT	✅	
Nombre	nombre	VARCHAR(20)		
Capacidad de habitantes	habitantes	BIGINT		

7.
Inserción de datos

Sólo se muestra la inserción de 5 registros...

Provincias

CÓDIGO EN GITHUB

```
INSERT INTO PROVINCIAS (nombre, habitantes)
VALUES
  ('Buenos Aires', 17446000),
  ('Catamarca', 406000),
  ('Chaco', 1356000),
  ('Chubut', 615000),
  ('Ciudad Autónoma de Buenos Aires', 2890000),
  (…)
;
```

Aulas

CÓDIGO EN GITHUB

```
INSERT INTO AULAS (nombre, edificio, capacidad)
VALUES
  ('Juan Domingo Perón', 'Principal', 131),
  ('Eva Perón', 'Principal', 281),
  ('Manuel Belgrano', 'Principal', 267),
  ('José de San Martín', 'Principal', 309),
  ('Domingo Faustino Sarmiento', 'Principal', 474),
  (…)
;
```

Profesores

CÓDIGO EN GITHUB 

```
INSERT INTO PROFESORES (nombre, apellido, fecha_nac, sexo, email, profesion, tipo_jornada, direccion, provincia_id)
VALUES
  ('Juan', 'García', '19811024', 'M', 'j.garcia@universidad.com', 'Médico', 'Full-Time', 'Calle Rosario, 123', 16),
  ('María', 'López', '19901111', 'F', 'm.lopez@universidad.com', 'Ingeniero', 'Full-Time', 'Avenida del Sol, 456', 17),
  ('Carlos', 'Rodríguez', '19620607', 'M', 'c.rodriguez@universidad.com', 'Abogado', 'Part-Time', 'Calle Primavera, 789', 14),
  ('Laura', 'Martínez', '19850930', 'F', 'l.martinez@universidad.com', 'Profesor', 'Full-Time', 'Paseo de los Pinos, 234', 15),
  ('José', 'González', '19671012', 'M', 'j.gonzalez@universidad.com', 'Arquitecto', 'Full-Time', 'Avenida Libertad, 567', 12),
  (…)
;
```

Alumnos

CÓDIGO EN GITHUB 

```
INSERT INTO ALUMNOS (nombre, apellido, fecha_nac, sexo, email, telefono, provincia_id)
VALUES
  ('Sofía', 'García', '20010821', 'F', 'sofia_garcia@gmail.com', 3516864233, 9),
  ('Mateo', 'Rodríguez', '19961022', 'M', 'mateo_rodriguez@gmail.com', 3516611277, 11),
  ('Valentina', 'López', '20010819', 'F', 'valentina_lopez@gmail.com', 3515846713, 10),
  ('Santiago', 'Martínez', '20010605', 'M', 'santiago_martinez@gmail.com', 3515788754, 4),
  ('Isabella', 'Gómez', '20010516', 'F', 'isabella_gomez@gmail.com', 3515113189, 7),
  (…)
;
```

Carreras

CÓDIGO EN GITHUB

```
INSERT INTO CARRERAS (nombre, puntos, director, plan_vigente)
VALUES
  ('Medicina', 136, 6, 'Plan 2010 - Res. HCD 2675/2010'),
  ('Ingeniería Civil', 144, 18, 'Plan 2022 - Res. HCD 2891/2022'),
  ('Derecho', 105, 19, 'Plan 2015 - Res. HCD 2623/2015'),
  ('Psicología', 122, 14, 'Plan 2023 - Res. HCD 2488/2023'),
  ('Administración de Empresas', 147, 14, 'Plan 2009 - Res. HCD 3051/2009'),
  (…)
;
```

Materias

CÓDIGO EN GITHUB

```
INSERT INTO MATERIAS (nombre, puntos, sem_dict, carrera_id)
VALUES
  ('Anatomía', 36, '1', 1),
  ('Farmacología I', 28, '1', 1),
  ('Materiales constructivos', 45, '1', 2),
  ('Leyes impositivas', 24, '2', 3),
  ('Derecho procesal', 50, '2', 3),
(…)
;
```

Matrículas

CÓDIGO EN GITHUB

```
INSERT INTO MATRICULAS (alumno_legajo, fecha, semestre, per_lectivo, importe)
VALUES
  (2, '20220321', '1', 2022, 67679.27),
  (10, '20220314', '1', 2022, 63533.81),
  (16, '20220326', '1', 2022, 78551.32),
  (13, '20220323', '1', 2022, 98423.47),
  (9, '20220301', '1', 2022, 69033.59),
  (…)
;
```

Facturas

CÓDIGO EN GITHUB

```
INSERT INTO FACTURAS (factura, alumno_legajo, fecha, tipo_factura)
VALUES
  ('0001-00007062', 2, '20220702', 'B'),
  ('0001-00005046', 5, '20230316', 'B'),
  ('0001-00002829', 18, '20220406', 'B'),
  ('0001-00001715', 20, '20220617', 'B'),
  ('0001-00002883', 9, '20220614', 'A'),
  (…)
;
```

Matrículas facturadas

CÓDIGO EN GITHUB

```
INSERT INTO MATRICULAS_FACTURADAS (matricula_id,  factura_id)
VALUES
  (8, '0001-00007062'),
  (14, '0001-00007062'),
  (6, '0001-00005046'),
  (35, '0001-00005046'),
  (10, '0001-00001715'),
  (…)
;
```

Cobranzas

CÓDIGO EN GITHUB

```
INSERT INTO COBRANZAS (factura_id, fecha, medio_pago)
VALUES
  ('0001-00001383', '20230322', 'Transferencia'),
  ('0001-00002829', '20220406', 'Tarjeta de Crédito'),
  ('0001-00002883', '20220618', 'Transferencia'),
  ('0001-00003297', '20230401', 'Tarjeta de Crédito'),
  ('0001-00005046', '20230318', 'Efectivo'),
  (…)
;
```

Cursadas

CÓDIGO EN GITHUB

```
INSERT INTO CURSADAS (alumno_legajo, materia_id, profesor_legajo, aula_id, fecha_inscripcion, semestre, per_lectivo, condicion)
VALUES
  (16, 5, 19, 7, '20220402', '1', 2022, 'Promoción'),
  (19, 1, 20, 13, '20220406', '1', 2022, 'Libre'),
  (3, 10, 3, 9, '20220326', '1', 2022, 'Regular'),
  (8, 7, 4, 12, '20220325', '1', 2022, 'Regular'),
  (1, 10, 17, 6, '20220323', '1', 2022, 'Libre'),
  (…)
;
```

Calificaciones

CÓDIGO EN GITHUB

```
INSERT INTO CALIFICACIONES (alumno_legajo, materia_id, profesor_legajo, aula_id, fecha_examen, nota, aprobado)
VALUES
  (7, 12, 14, 7, '20230215', 9, true),
  (14, 1, 18, 6, '20220916', 10, true),
  (16, 15, 10, 10, '20230511', 2, false),
  (20, 5, 11, 4, '20220708', 6, true),
  (1, 15, 1, 1, '20221103', 4, true),
  (…)
;
```

8.
Creación de objetos

Vistas

Promedio de alumnos

Objetivo

Obtener una tabla que posea los datos de los alumnos y los promedios obtenidos en el total de sus calificaciones.

Tablas/Datos

alumnos, calificaciones

Descripción

Se realiza un join entre las tablas alumnos y calificaciones para obtener las calificaciones que los alumnos han obtenido, luego se calcula el promedio agrupando por cada alumno.

CÓDIGO EN GITHUB 

Alumnos por provincia

Objetivo

Obtener una tabla que muestre la provincia de origen de la totalidad de alumnos, puntualmente, la cantidad de alumnos asociados a cada provincia.

Tablas/Datos

provincias, alumnos

Descripción

Se realiza un join entre la tabla provincias y alumnos para obtener todos los alumnos relacionados con cada provincia.

Luego se agrupa por provincia, contando la cantidad de registros. Se realiza un LEFT JOIN para que el resultado contengan la totalidad de provincias, sin importar si tienen o no alumnos.

CÓDIGO EN GITHUB 

Facturación histórica por alumno

Objetivo

Obtener el total facturado por alumno.

Tablas/Datos

matriculas_facturadas, facturas, matriculas, alumnos

Descripción

Se realiza un join de las tablas mencionadas anteriormente, para lograr obtener las matriculas_facturadas y los datos de los alumnos (como nombre y apellido).

La unión se realizó en 4 tablas debido a que las mismas se relacionan en cadena.

Con los datos obtenidos, se agrupa por alumno y se obtiene la suma del importe obtenido de la tabla matriculas_facturadas.

CÓDIGO EN GITHUB 

Matrículas pendientes de facturar

Objetivo

Obtener un listado de aquellas matrículas generadas por los alumnos, pero que aún no fueron facturadas a los mismos (es decir, no se emitió un comprobante de factura).

Tablas/Datos

matriculas, matriculas_facturadas

Descripción

Se realizó un join entre las matrículas y las matrículas_facturadas, y luego se descartaron aquellas matriculas que ya se encontraban facturadas, quedando solamente las pendientes de facturar.

Luego, se totaliza por periodo lectivo, semestre y se obtiene la suma del importe pendiente de facturar.

CÓDIGO EN GITHUB 

Facturas impagas

Objetivo

Obtener aquellas facturas que a la fecha no poseen cobranzas asociadas.

Tablas/Datos

facturas, cobranzas

Descripción

Se realizó un join entre ambas tablas para obtener todas las cobranzas asociadas a facturas, y se descartaron aquellas facturas que tenían cobranzas.

No se usaron criterios de agregación o agrupamiento, sino que se trata de una lista detallada de facturas pendiente de cobro.

CÓDIGO EN GITHUB 

Facturación duplicada

Objetivo

Obtener un detalle de aquellas matrículas que fueron incluidas erróneamente en más de una factura.

Tablas/Datos

matriculas_facturadas, facturas

Descripción

Se procedió a realizar un join entre las tablas mencionadas para tener detalle que matrícula se incluye en cada factura.

Luego se filtraron aquellas que figuraban en más de una factura.

Para realizar dicho filtro, se utilizó una subconsulta que obtiene aquellos id de matrículas que se encuentran duplicados.

CÓDIGO EN GITHUB 

Alumnos promocionados

Objetivo

Listar todos los alumnos que hayan obtenido la condición de “Promoción” en sus materias cursadas.

Tablas/Datos

cursadas, alumnos, profesores, materias

Descripción

Se unieron las tablas mencionadas, y se filtró aquellos alumnos que poseían la condición de “Promoción”.

Se reportan datos del alumno y del profesor, así como periodo lectivo, y semestre.

CÓDIGO EN GITHUB 

8.
Creación de objetos

Funciones

Nota en texto

Objetivo

Esta función devuelve un texto asociado a la nota obtenida por el alumno.
Se obtuvo la escala de ejemplo de la siguiente página (Facultad Ciencias Económicas).
https://www.eco.unc.edu.ar/reglamentaciones/#escala-de-calificaciones-ordenanza-n-482-2009

Tablas/Datos

No aplicable.

Descripción

Usando una estructura case, se evalúa en que rango se encuentra la nota obtenida, para retornar distintos textos.

CÓDIGO EN GITHUB

Cantidad de materias aprobadas

Objetivo

Esta función cuenta la cantidad de materias aprobadas para un alumno dado, en la determinada carrera. Se agrega la carrera porque el alumno puede estar cursando más de una carrera.

Tablas/Datos

alumnos, carreras, calificaciones

Descripción

En primer lugar, se verifica si los parámetros de legajo de alumno y el id de carrera existen en las tablas correspondientes.

En caso de que no, se lanza una excepción.

En caso de que existan, se guarda en una variable la cantidad de notas mayores a 4 (estado aprobado) que el alumno tiene en la carrera determinada.

CÓDIGO EN GITHUB 

8.
Creación de objetos

Procedimientos almacenados

Ordenar tabla

Objetivo

Consultar a una tabla pasada por parámetro, ordenando por un campo pasado también por parámetro. Asimismo, debe especificarse ASC o DESC para indicar el modo de orden. En caso de estar vacío el modo de orden, se realizará un orden ascendente.

Tablas/Datos

tabla a ordenar

Descripción

Se va creando de a partes una prepared statement, indicando la tabla, campo de ordenación y sentido de orden (ascendente/descendente).

Se validan los parámetros ingresados, en caso de que no sean los esperados, se lanza una excepción.

Una vez construida la sentencia preparada, se ejecuta y se libera de la memoria.

CÓDIGO EN GITHUB 

Calcular el promedio de un alumno

Objetivo

Devuelve en un parámetro de salida el promedio de las notas obtenidos por un alumno.

Tablas/Datos

calificaciones

Descripción

Se realiza una consulta a la tabla mencionada, filtrando por el alumno recibido por parámetro, y se guarda en una variable de salida, el promedio de dichas notas obtenidas.

CÓDIGO EN GITHUB 

Registrar cobranza

Objetivo

Registra una cobranza de una factura existente en la tabla facturas.

Tablas/Datos

cobranzas

Descripción

Realiza una inserción de un registro en la tabla mencionada, con los parámetros recibidos (fecha, factura que se cobró y medio de pago utilizado).

CÓDIGO EN GITHUB 

Cambiar director de carrera

Objetivo

Asigna otro profesor como director de carrera. Se le debe indicar la carrera a actualizar y el nombre y apellido del profesor que asumirá el rol de director.

Tablas/Datos

profesores, carreras

Descripción

Dado que se poseen nombre de alumno y descripción de carrera, ambos campos que no son claves primarias de sus tablas, se procede a determinar los mismos haciendo consultas a las tablas mencionada para tener el legajo y el id correspondiente.

Con dichos datos, se hace una actualización del registro de la carrera para asignar el campo director con el legajo del profesor que se obtuvo a través del nombre pasado por parámetro.

CÓDIGO EN GITHUB 

8.
Creación de objetos

Disparadores

Validez de calificación

Objetivo

Analiza la validez de la nota (rango entre 0 y 10), y setea el campo aprobado, considerando que con 4 ya se obtiene dicha condición.

Tablas/Datos

calificaciones

Descripción

En primera instancia se valida si la nota se encuentra fuera del rango, seteando alguno de los valores extremos (0 o 10) en caso de estar por fuera del intervalo.

Luego, si la nota es mayor o igual a 4, se setea el campo aprobado en 1, o 0 en su defecto.

Una vez realizados todos estas validaciones y definiciones, se continua el proceso de inserción en la tabla calificaciones.

CÓDIGO EN GITHUB 

Log modificación de alumnos

Objetivo

Registra sólo las modificaciones que sufre cada registro y campo de la tabla alumnos. Aclaración: no registra inserciones ni eliminaciones.

Tablas/Datos

alumnos, log_modif_alumnos

Descripción

Este trigger se ejecuta luego de cada actualización de un registro de la tabla alumnos, no teniendo efecto sobre inserciones o eliminaciones.

El procedimiento consiste en validar uno a uno los atributos de la tabla alumnos, y en caso de que el valor anterior difiera del actualizado, se procede a insertar un registro en la tabla log_modif_alumnos con detalle del campo cambiado, valor anterior, valor nuevo, fecha y hora de la actualización, y usuario que ejecutó la sentencia.

CÓDIGO EN GITHUB 

Log insert/update/select

Objetivo

Registra los eventos realizados sobre cada id de la tabla materias.
Aclaración: no registra valores anteriores y nuevos.

Tablas/Datos

log_tablas, y a modo de ejemplo: materias

Descripción

En este caso se trata de un conjunto de triggers (uno para cada evento de inserción, actualización y eliminación) que registran en la tabla log_tablas el detalle de que operación se hizo, para que registro de una entidad particular.

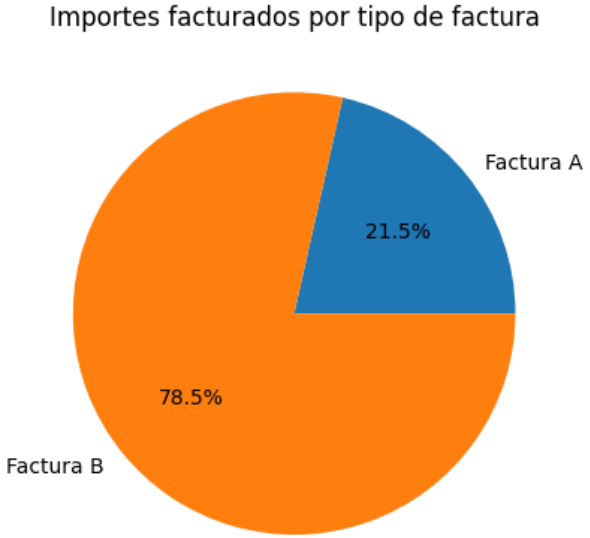
Con el objetivo de lograr este cometido, se procedió a crear un stored procedure denominado RegistrarEntradaLog que recibe por parámetro la entidad (tabla), el id de la novedad y la operación realizada (insert/update/delete). Este SP facilita la inserción de registros en la tabla de log.

Luego, como ejemplo, se creó para la tabla materias un trigger para cada operación, haciendo llamada del SP con los datos necesarios con posterioridad a cada operación.

CÓDIGO EN GITHUB 

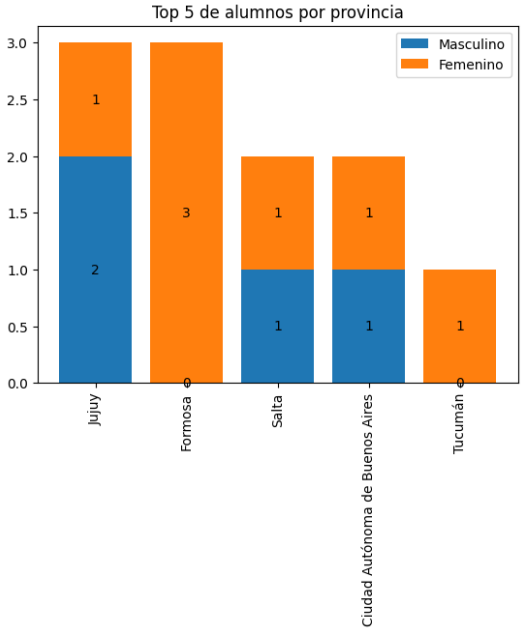
9.
Informes generados

Importes facturados por tipo de factura



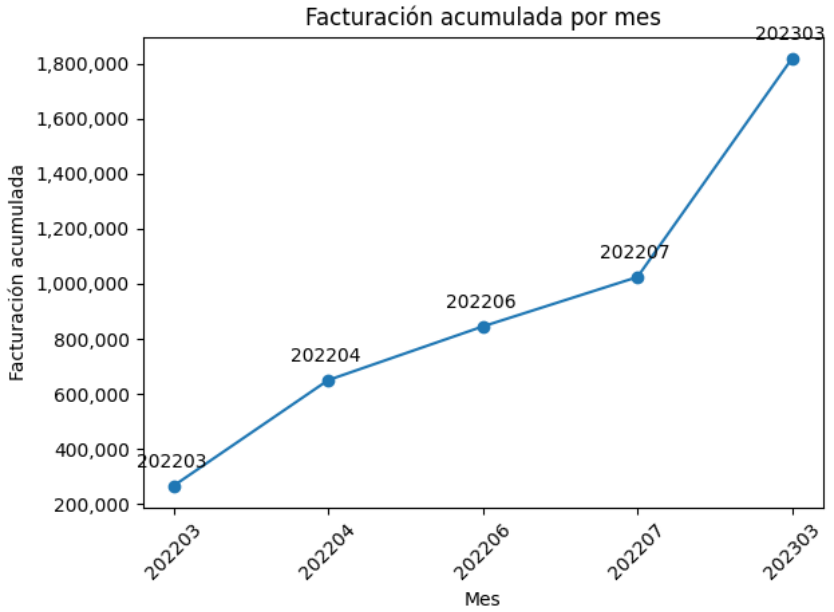
CÓDIGO EN GITHUB

Cantidad de alumnos por provincia (top 5)



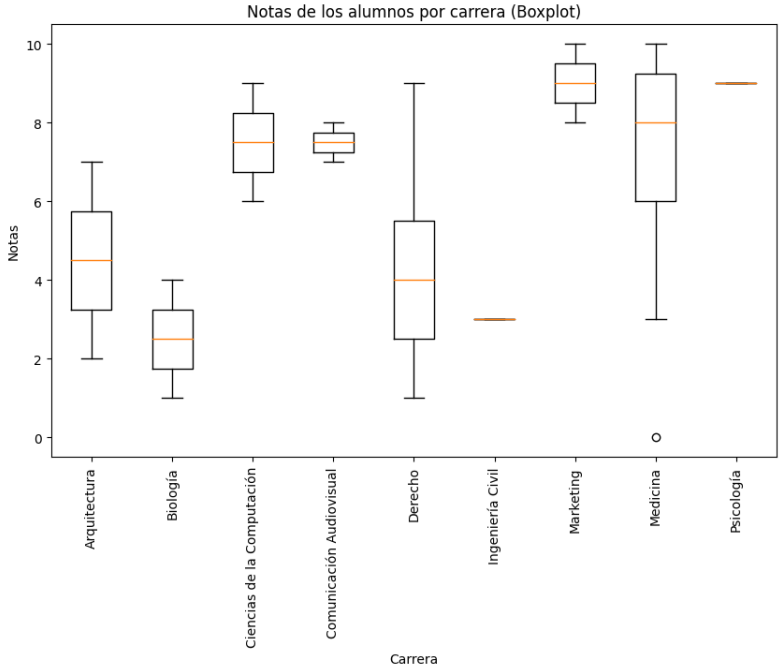
CÓDIGO EN GITHUB

Facturación acumulada por mes



CÓDIGO EN GITHUB

Boxplot de notas por carrera



CÓDIGO EN GITHUB

10. Herramientas y tecnologías

Herramientas y tecnologías

- **MySQL** – creación y manipulación de base de datos
- **draw.io** – diagrama de entidad-relación
- **ChatGPT** – generación de datos
- **Python / Jupyter Notebooks** – reportes de información
- **Git / GitHub** – control de cambios y publicación
- **Microsoft Office** – elaboración de informes y presentación





11.
Futuras líneas

Futuras líneas

- La base de datos podría utilizarse en un entorno real reflejando la totalidad de procesos asociados con la enseñanza en niveles terciario, universitario y/o post-grado.
- Esta base de datos podría trabajar en conjunto con diversos front-ends como ser una aplicación web tipo “Campus Online” (autogestión de alumnos) ya sea para plataformas web y/o mobile.
- Se podría incorporar entidades que reflejen otras operatorias como tesorería, recursos humanos, compras e insumos, entre otros.

Gracias!

Rodrigo Casas

casasrodri
casasrodrigo2307
cr.rodrigo.casas@gmail.com