

Prueba SQL - UAPA

Cristhian David Mora Uribe

Links al Contenido

[Links al Contenido](#)

[Entregables](#)

[Diagramas](#)

[Código](#)

[Etapas de diseño](#)

[Diagrama lógico](#)

[Diagrama Físico](#)

[Diagrama en workbench](#)

[Creates](#)

[Triggers](#)

[Procedimientos](#)

1. [¿Cuántos docentes activos hay en la Facultad de Ingeniería?](#)
2. [¿Cuáles son los cursos que se dictaron en un periodo dado?](#)
3. [¿Cuál es la lista de estudiantes para un curso y grupo dados en el periodo actual?](#)
4. [¿Cuál es el promedio de las calificaciones obtenidas en un curso los últimos 5 años?](#)
5. [¿Cuál sería el procedimiento para crear un nuevo curso y asignarle un docente?](#)

[Resultados:](#)

[Contenido inicial de las tablas:](#)

[Pruebas ejecutadas y resultados:](#)

[Aspectos a mejorar](#)

Entregables

Todo está subido en github y a la vez con copia en archivo.zip enviado al correo.

Diagramas

Diseño inicial de la base de datos en formato png, para visualizarlos en mejor calidad.

Código

create.sql = archivo con secuencia de create, para crear todo desde cero.

triggers.sql = Un par de disparadores para facilitarme los inserts y que calculaban columnas en base a otras

inserts.sql = datos base para generar pruebas

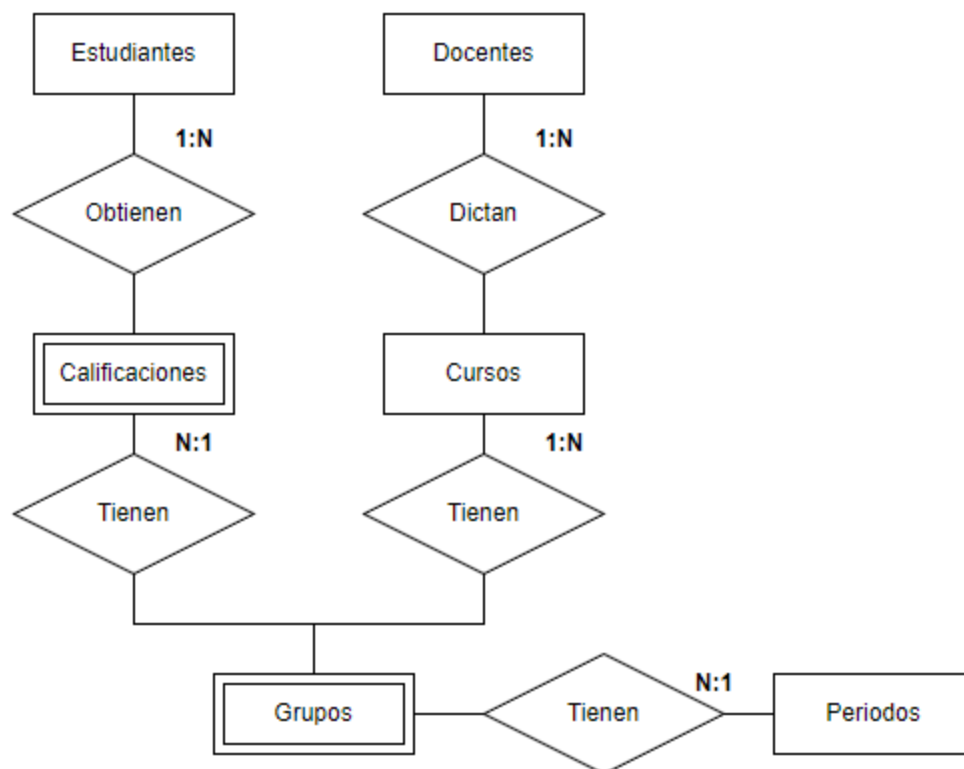
procedimientos = procedures con comentarios que responden las preguntas 1 a 1.

all.sql = Junte todo lo anterior en un solo archivo para una **ejecución más cómoda**.

Etapa de diseño

Diagrama lógico

Para comenzar con el diseño comencé con un diagrama simple para tener todas las entidades que probablemente tuviera que tener en cuenta para crear las tablas y responder las preguntas sin detallar demasiado.



En este diseño inicial consideré tener tablas adicionales, cómo la de inscripciones ya que pareciera que los únicos estudiantes que existirían en los grupos serían quienes estuvieran en la tabla clasificaciones. sin embargo decidí que ambas tablas podrían contenerse en calificaciones y los estudiantes que no tuvieran calificación contarían cómo inscritos, permitiendo simplemente el espacio nulo en tal lugar, mientras posteriormente se subiera la nota.

Diagrama Físico

Al diseñar este diagrama como detalle adicional cree una tabla de facultades y ya que en el numeral 1 se demandaba contar los docentes activos específicamente de la facultad de ingeniería. De resto deje todo de igual forma y detalle las columnas necesarias para cada tabla.

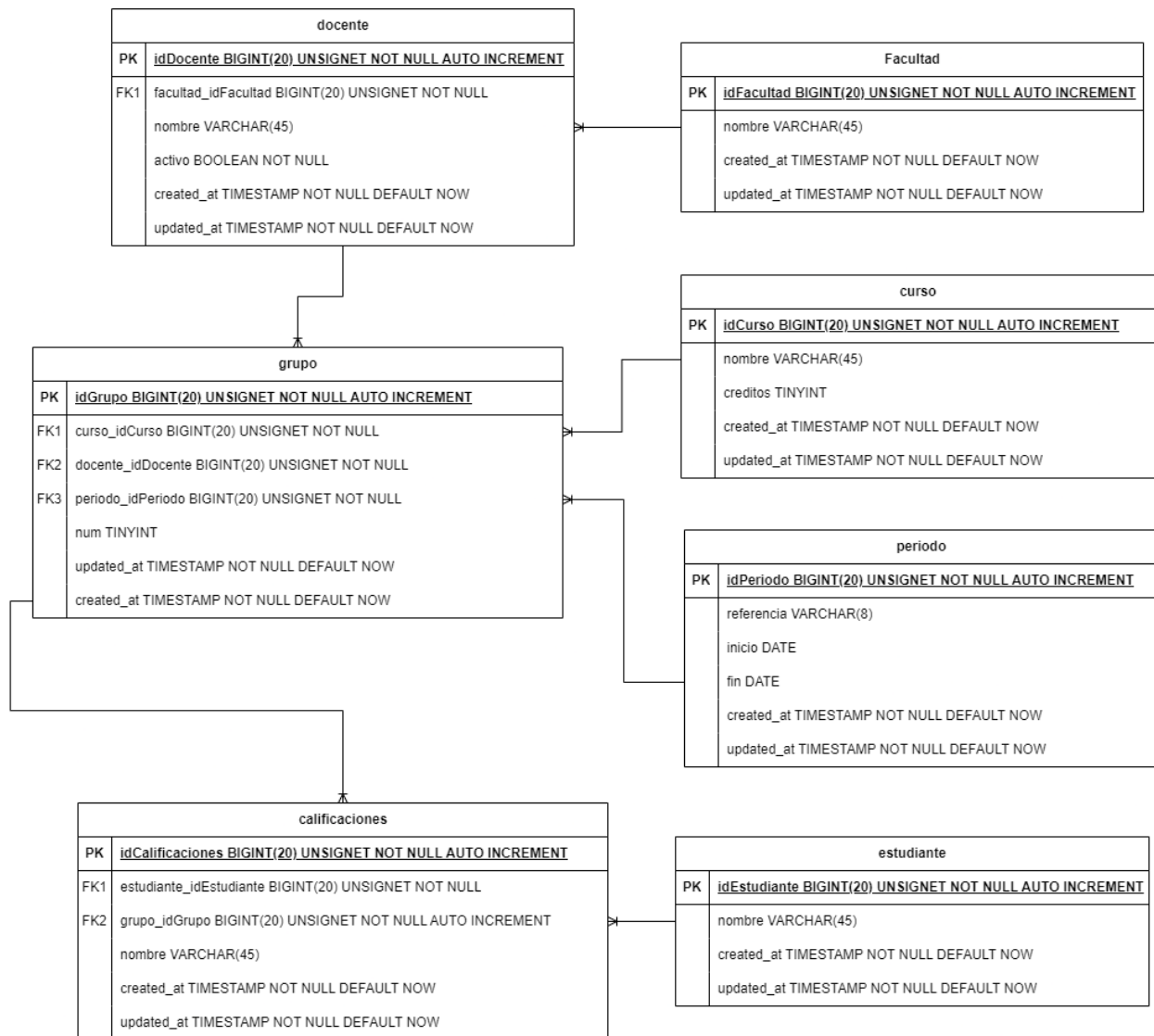
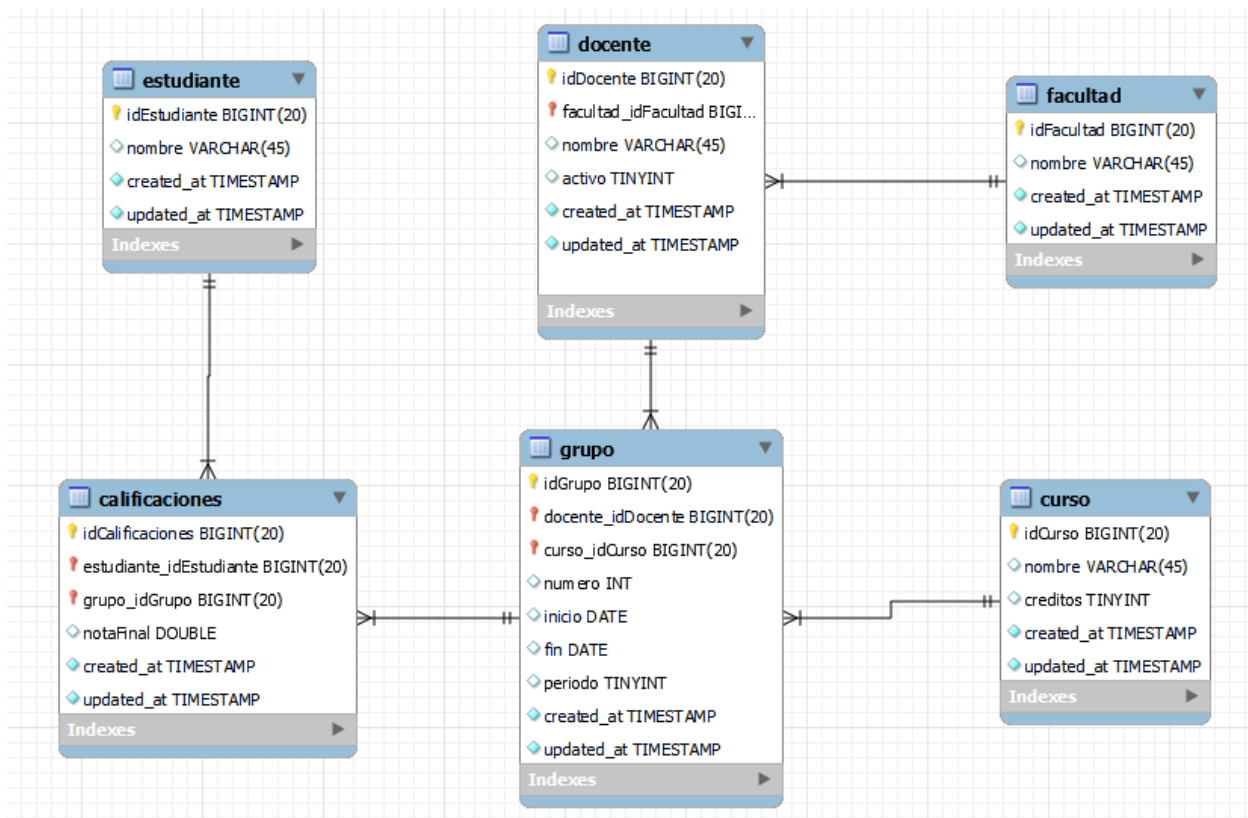


Diagrama en workbench

Algo que suelo hacer es apoyarme de la funcionalidad para crear diagramas de workbench para modificar y ver de manera más grafica y veloz cómo se vería el diseño, además de hacer ingeniería inversa al mismo para generar código sql a partir del diseño y terminar más rápido los create correspondientes. Claro que esto sería solo una guía inicial que seria bastante modificada mientras avanzaba en el diseño.



Aquí no se ve la tabla periodo, tuve bastantes dudas si incluirla o no en el diseño, al final la deje, más adelante hablaré de ello. El diseño que más representa el resultado final es el **diagrama fisico**.

Creates

Los create son un reflejo del diseño casi de manera general, excepto por la fila activo en la tabla profesores, decidí calcular ese estado en el momento que se pida para ahorrar procesamiento de actualizarlo cada vez que se consulte la base de datos o se cambie de periodo. El detalle de cómo se ve cada atributo se puede apreciar más en el código en si.

Triggers

Despues de terminar de detallar los create procedí a hacer un par de disparadores para poder hacer los inserts más sencillo en particular 2.

set_group_number: Una vez ingresado el grupo procede a comprobar si ya existían otros grupos con el mismo nombre, en el mismo periodo y cuantos eran, para hacer que el valor de el numero de grupo corresponda a lo ya previamente ingresado.

set_period_dates: Con el periodo era para agregar fechas de inicio y fin por defecto por ejemplo al 2010-1, esto para facilitar calculos a futuro al intentar saber si algo pertenecía al periodo actual.

Procedimientos

1. ¿Cuántos docentes activos hay en la Facultad de Ingeniería?

"GetActiveDocentesInFacultad"

Las condiciones para que un docente esté activo es que debe de encontrarse en la tabla de grupos (donde se relaciona con un curso) y que dicho curso tenga el periodo actual.

Para esto genere un código que revisa en la tabla periodos el periodo donde la fecha actual esté entre la fecha de inicio y la fecha fin de dicho periodo. Y luego busca y cuenta todos los id de los profesores que tienen asignado un grupo en un periodo que contenga la fecha actual.

Consulta para obtener el conteo de docentes activos:

- Utiliza una subconsulta para obtener el número de docentes activos en la facultad. Para ello, realiza las siguientes acciones:
 - En la tabla `grupo`, filtra los registros en función de los siguientes criterios:
 - El ID del período (`periodo_idPeriodo`) coincide con el período actual.
 - Utiliza la función `CURDATE()` para determinar si la fecha actual está dentro del rango de fechas (inicio y fin) del período.

- El ID del docente (`docente_idDocente`) en la tabla `grupo` debe estar presente en la subconsulta siguiente.
- La subconsulta dentro del segundo `IN` selecciona los IDs de docentes de la tabla `docente` que pertenecen a la facultad con el ID 1 (puedes cambiarlo según corresponda).

En resumen, este procedimiento te permite obtener el número de docentes activos en una facultad específica durante el período actual. El resultado será la cantidad de docentes que están asignados a grupos en la facultad y que tienen un período de actividad que incluye la fecha actual.

2. ¿Cuáles son los cursos que se dictaron en un periodo dado?

"GetCoursesByPeriodReference"

Filtro los id de cursos en la tabla grupos que contienen el periodo dado, y uso `distinct` para eliminar cursos repetidos que tienen varios grupos.

1. Búsqueda del ID de Período:

- El procedimiento toma como entrada una referencia de período (`reference`) en el formato especificado (por ejemplo, "2022-1").
- Utiliza una consulta para encontrar el ID del período correspondiente en la tabla `periodo` , basado en la referencia proporcionada. El resultado se almacena en la variable `period_id` .

2. Obtención de Cursos:

- Utiliza una consulta con una cláusula `INNER JOIN` para combinar la tabla `curso` y la tabla `grupo` utilizando el campo `idCurso` .
- Filtra los resultados para obtener cursos que estén relacionados con el ID del período obtenido en el paso anterior.
- Utiliza la cláusula `DISTINCT` para asegurarse de que cada curso se muestre solo una vez en el resultado.

- Selecciona los campos `idCurso` y `nombre` de la tabla `curso` para obtener información sobre los cursos dictados en el período dado.

En resumen, este procedimiento te permite obtener una lista de cursos que se dictaron en un período específico, utilizando la referencia del período como entrada. Los cursos se obtienen mediante la combinación de las tablas `curso` y `grupo`, y se filtran según el ID del período.

3. ¿Cuál es la lista de estudiantes para un curso y grupo dados en el periodo actual?

"GetStudentsByCourseGroupAndPeriod":

1. Parámetros de entrada:

- `IN course_id INT`: Este parámetro de entrada representa el ID del curso para el cual se desea obtener la lista de estudiantes.
- `IN group_id INT`: Este parámetro de entrada representa el ID del grupo para el cual se desea obtener la lista de estudiantes.

2. Declaración de variables:

- `period_id INT`: Esta variable se utilizará para almacenar el ID del período actual.

3. Consulta para obtener el ID del período actual:

- Utilizando la función `CURDATE()`, se determina si la fecha actual está dentro de las fechas de inicio y fin de algún período.
- El ID del período correspondiente se guarda en la variable `period_id`.

4. Consulta para obtener la lista de estudiantes:

- Selecciona los campos `idEstudiante` y `nombre` de la tabla `estudiante`.
- Se utiliza una serie de uniones (`INNER JOIN`) para relacionar las tablas `estudiante`, `calificaciones` y `grupo`.
- Se filtra la información en función de los siguientes criterios:

- El ID del curso (`curso_idCurso`) en la tabla `grupo` coincide con el valor proporcionado `course_id` .
- El ID del grupo (`idGrupo`) en la tabla `grupo` coincide con el valor proporcionado `group_id` .
- El ID del período (`periodo_idPeriodo`) en la tabla `grupo` coincide con el valor almacenado en la variable `period_id` .

En resumen, este procedimiento te permite obtener la lista de estudiantes que están inscritos en un curso y grupo específico durante el período actual. Esto puede ser útil para tener un registro de los estudiantes que están tomando un curso específico en un grupo particular durante el semestre actual.

4. ¿Cuál es el promedio de las calificaciones obtenidas en un curso los últimos 5 años?

"CalculateAverageGradeByCourse":

1. **Parámetro de entrada** (`IN course_id INT`): Este procedimiento toma un parámetro de entrada `course_id` , que representa el ID del curso para el cual se desea calcular el promedio de calificaciones.
2. **Declaración de variables:**
 - `current_year` : Se asigna el año actual mediante la función `YEAR(CURDATE())` .
 - `start_year` : Se calcula el año de inicio del período de los últimos 5 años restando 5 años del año actual.
 - `end_year` : Se asigna el año actual.
3. **Consulta SQL:**
 - Selecciona el promedio de las calificaciones (`notaFinal`) de la tabla `calificaciones` .
 - La cláusula `WHERE` filtra las calificaciones basadas en los siguientes criterios:

- `grupo_idGrupo` está en la lista de ID de grupos que corresponden al curso proporcionado (`course_id`).
- Se utiliza una subconsulta para seleccionar los ID de grupos de la tabla `grupo` que cumplen con los siguientes criterios:
 - `curso_idCurso` coincide con el `course_id` proporcionado.
 - Existe un período (tabla `periodo`) en el que el grupo haya tenido lugar durante los últimos 5 años.
- `notaFinal` no es nula.

En resumen, este procedimiento calcula el promedio de las calificaciones obtenidas en un curso específico durante los últimos 5 años. Ayuda a proporcionar una visión general de cómo les ha ido a los estudiantes en ese curso a lo largo del tiempo.

5. ¿Cuál sería el procedimiento para crear un nuevo curso y asignarle un docente?

"CreateCourseAndAssignDocent":

1. Creación del Curso:

- El procedimiento toma como entrada el nombre del curso (`course_name`) y el número de créditos (`creditos_value`) que el curso tendrá.
- Utiliza una sentencia `INSERT INTO` para crear un nuevo registro en la tabla `curso` con el nombre y los créditos proporcionados. El `LAST_INSERT_ID()` se utiliza para capturar el ID del curso recién creado.

2. Obtención del Año y Semestre Actual:

- Utiliza la función `YEAR(CURDATE())` para obtener el año actual y `MONTH(CURDATE())` para obtener el mes actual.
- Determina si el semestre actual es 1 (enero a junio) o 2 (julio a diciembre) en función del mes actual.

3. Búsqueda o Creación del Período:

- Utiliza una consulta para buscar en la tabla `periodo` si ya existe un período con la referencia correspondiente al año y semestre actual.
- Si no se encuentra, crea un nuevo registro en la tabla `periodo` con la referencia calculada y captura el ID del período recién creado.

4. Creación del Grupo:

- Utiliza una sentencia `INSERT INTO` para crear un nuevo grupo en la tabla `grupo`, asignando el ID del docente, el ID del curso y el ID del período obtenido.

5. Mensaje de Resultado:

- Finalmente, se construye un mensaje `result_message` que informa que el curso fue creado y el docente asignado al grupo, junto con el ID del grupo creado.

En resumen, este procedimiento te permite crear un nuevo curso, asignarle un docente y crear un grupo asociado al curso y período especificados. El mensaje de resultado proporciona información sobre la operación realizada.

Resultados:

Contenido inicial de las tablas:

Periodo

	idPeriodo	referencia	inicio	fin	created_at	updated_at
	11	2015-1	2015-02-01	2015-06-01	2023-08-21 00:35:13	2023-08-21 00:35:13
	12	2015-2	2015-08-01	2015-12-01	2023-08-21 00:35:13	2023-08-21 00:35:13
	13	2016-1	2016-02-01	2016-06-01	2023-08-21 00:35:13	2023-08-21 00:35:13
	14	2016-2	2016-08-01	2016-12-01	2023-08-21 00:35:13	2023-08-21 00:35:13
	15	2017-1	2017-02-01	2017-06-01	2023-08-21 00:35:13	2023-08-21 00:35:13
	16	2017-2	2017-08-01	2017-12-01	2023-08-21 00:35:13	2023-08-21 00:35:13
	17	2018-1	2018-02-01	2018-06-01	2023-08-21 00:35:13	2023-08-21 00:35:13
	18	2018-2	2018-08-01	2018-12-01	2023-08-21 00:35:13	2023-08-21 00:35:13
	19	2019-1	2019-02-01	2019-06-01	2023-08-21 00:35:13	2023-08-21 00:35:13
	20	2019-2	2019-08-01	2019-12-01	2023-08-21 00:35:13	2023-08-21 00:35:13
	21	2020-1	2020-02-01	2020-06-01	2023-08-21 00:35:13	2023-08-21 00:35:13
	22	2020-2	2020-08-01	2020-12-01	2023-08-21 00:35:13	2023-08-21 00:35:13
	23	2021-1	2021-02-01	2021-06-01	2023-08-21 00:35:13	2023-08-21 00:35:13
	24	2021-2	2021-08-01	2021-12-01	2023-08-21 00:35:13	2023-08-21 00:35:13
	25	2022-1	2022-02-01	2022-06-01	2023-08-21 00:35:13	2023-08-21 00:35:13
	26	2022-2	2022-08-01	2022-12-01	2023-08-21 00:35:13	2023-08-21 00:35:13
	27	2023-1	2023-02-01	2023-06-01	2023-08-21 00:35:13	2023-08-21 00:35:13
	28	2023-2	2023-08-01	2023-12-01	2023-08-21 00:35:13	2023-08-21 00:35:13
*	NULL	NULL	NULL	NULL	NULL	NULL

Grupo

	idGrupo	docente_idDocente	curso_idCurso	numero	periodo_idPeriodo	created_at	updated_at
▶	1	1	1	1	28	2023-08-21 00:35:13	2023-08-21 00:35:13
	2	2	1	2	28	2023-08-21 00:35:13	2023-08-21 00:35:13
	3	3	1	1	26	2023-08-21 00:35:13	2023-08-21 00:35:13
	4	4	1	1	24	2023-08-21 00:35:13	2023-08-21 00:35:13
	5	1	2	1	26	2023-08-21 00:35:13	2023-08-21 00:35:13
	6	2	2	1	27	2023-08-21 00:35:13	2023-08-21 00:35:13
	7	3	2	2	27	2023-08-21 00:35:13	2023-08-21 00:35:13
	8	1	9	1	28	2023-08-21 00:35:17	2023-08-21 00:35:17

Facultad

	idFacultad	nombre	created_at	updated_at
▶	1	Ingeniería	2023-08-21 00:35:13	2023-08-21 00:35:13
	2	Medicina	2023-08-21 00:35:13	2023-08-21 00:35:13
	3	Humanas	2023-08-21 00:35:13	2023-08-21 00:35:13
	4	Ciencias	2023-08-21 00:35:13	2023-08-21 00:35:13

Curso

	idCurso	nombre	creditos	created_at	updated_at
▶	1	Diferencial	4	2023-08-21 00:35:13	2023-08-21 00:35:13
	2	Programación	3	2023-08-21 00:35:13	2023-08-21 00:35:13
	3	Química	3	2023-08-21 00:35:13	2023-08-21 00:35:13
	4	materia 1	4	2023-08-21 00:35:13	2023-08-21 00:35:13
	5	materia 2	3	2023-08-21 00:35:13	2023-08-21 00:35:13
	6	materia 3	2	2023-08-21 00:35:13	2023-08-21 00:35:13
	7	materia 4	4	2023-08-21 00:35:13	2023-08-21 00:35:13
	8	materia 5	3	2023-08-21 00:35:13	2023-08-21 00:35:13

Docente

	idDocente	facultad_idFacultad	nombre	created_at	updated_at
▶	1	1	p0	2023-08-21 00:35:13	2023-08-21 00:35:13
	2	1	p01	2023-08-21 00:35:13	2023-08-21 00:35:13
	3	1	p02	2023-08-21 00:35:13	2023-08-21 00:35:13
	4	1	p03	2023-08-21 00:35:13	2023-08-21 00:35:13
	5	2	p1	2023-08-21 00:35:13	2023-08-21 00:35:13
	6	2	p2	2023-08-21 00:35:13	2023-08-21 00:35:13
	7	3	p3	2023-08-21 00:35:13	2023-08-21 00:35:13
	8	3	p4	2023-08-21 00:35:13	2023-08-21 00:35:13
	9	4	p5	2023-08-21 00:35:13	2023-08-21 00:35:13
	10	4	p6	2023-08-21 00:35:13	2023-08-21 00:35:13
	11	1	p7	2023-08-21 00:35:13	2023-08-21 00:35:13
	12	1	p8	2023-08-21 00:35:13	2023-08-21 00:35:13

Calificaciones

idCalificaciones	estudiante_idEstudiante	grupo_idGrupo	notaFinal	created_at	updated_at
1	1	1	39	2023-08-21 00:35:13	2023-08-21 00:35:13
2	2	1	45	2023-08-21 00:35:13	2023-08-21 00:35:13
3	3	1	23	2023-08-21 00:35:13	2023-08-21 00:35:13
4	4	1	36	2023-08-21 00:35:13	2023-08-21 00:35:13
5	5	1	43	2023-08-21 00:35:13	2023-08-21 00:35:13
6	6	1	34	2023-08-21 00:35:13	2023-08-21 00:35:13
7	7	2	20	2023-08-21 00:35:13	2023-08-21 00:35:13
8	8	2	25	2023-08-21 00:35:13	2023-08-21 00:35:13
9	9	2	15	2023-08-21 00:35:13	2023-08-21 00:35:13
10	11	3	40	2023-08-21 00:35:13	2023-08-21 00:35:13
11	12	3	45	2023-08-21 00:35:13	2023-08-21 00:35:13
12	1	5	30	2023-08-21 00:35:13	2023-08-21 00:35:13
13	2	5	30	2023-08-21 00:35:13	2023-08-21 00:35:13
14	3	5	30	2023-08-21 00:35:13	2023-08-21 00:35:13
15	4	6	50	2023-08-21 00:35:13	2023-08-21 00:35:13
16	5	6	50	2023-08-21 00:35:13	2023-08-21 00:35:13
17	6	7	50	2023-08-21 00:35:13	2023-08-21 00:35:13
18	7	7	40	2023-08-21 00:35:13	2023-08-21 00:35:13
19	8	7	48	2023-08-21 00:35:13	2023-08-21 00:35:13
NULL	NULL	NULL	NULL	NULL	NULL

Pruebas ejecutadas y resultados:

```
CALL GetStudentsByCourseGroupAndPeriod('1','1');
```

	idEstudiante	nombre
▶	1	Al1
	2	al2
	3	al3
	4	al4
	5	al5
	6	al6

```
CALL GetActiveDocentesInFacultad;
```

	COUNT(DISTINCT docente_idDocente)
▶	2

```
CALL GetCoursesByPeriodReference('2023-2');
```

	idCurso	nombre
▶	1	Diferencial

```
CALL CalculateAverageGradeByCourse(1);
```

	promedio
▶	33.18181818181818

```
SET @message = '';
CALL CreateCourseAndAssignDocent('Nuevo Curso',4,1, 1, '2023-2', @message);
SELECT @message;
```

	idCurso	nombre	creditos	created_at	updated_at
▶	1	Diferencial	4	2023-08-21 00:35:13	2023-08-21 00:35:13
	2	Programación	3	2023-08-21 00:35:13	2023-08-21 00:35:13
	3	Química	3	2023-08-21 00:35:13	2023-08-21 00:35:13
	4	materia 1	4	2023-08-21 00:35:13	2023-08-21 00:35:13
	5	materia 2	3	2023-08-21 00:35:13	2023-08-21 00:35:13
	6	materia 3	2	2023-08-21 00:35:13	2023-08-21 00:35:13
	7	materia 4	4	2023-08-21 00:35:13	2023-08-21 00:35:13
	8	materia 5	3	2023-08-21 00:35:13	2023-08-21 00:35:13
	9	Nuevo Curso	4	2023-08-21 00:35:17	2023-08-21 00:35:17

	@message
▶	Curso "Nuevo Curso" creado y docente asignad...

Aspectos a mejorar

- Siento que hay ambigüedad respecto a lo que se debe considerar periodo actual y como debe ser almacenado, por ejemplo tuve muchas dudas si dejar una columna de estado en los docentes para verificar si están activos o no.
- En la solución particular asumí que un grupo particular de un curso era el que era asociado a un docente en particular, y no el curso directamente. Sin embargo sería interesante también implementar un docente asignado a un curso en general, cómo un responsable de la materia o algo así.
- Tuve varios problemas con los collate y tuve que recurrir a forzarlos para que algunos procedimientos funcionaran, sería bueno buscar evitar hacer eso.
- La forma en que se calculan los promedios de un curso los últimos 5 años podría mejorarse, ya que este promedio se veía afectado por la cantidad de estudiantes en un curso particular, es decir que había periodos que tenían más peso que otros por la cantidad de estudiantes, sería interesante ver cómo se vería un promedio donde cada periodo tuviera el mismo peso.
- Aún no estoy seguro si la relación de estudiantes y grupos, que era muchos a muchos debio almacenarse en conjunto a la sección de calificaciones, ya que los

estudiantes en el periodo actual no tenían calificación aún.

- Respecto al punto anterior, asumí que la calificación de un periodo particular era únicamente la calificación final, una mejora podría ser que por cada estudiante se tuviera una lista de calificaciones y porcentajes por cada grupo en el que estuvo y pues evidentemente mantener la calificación final.