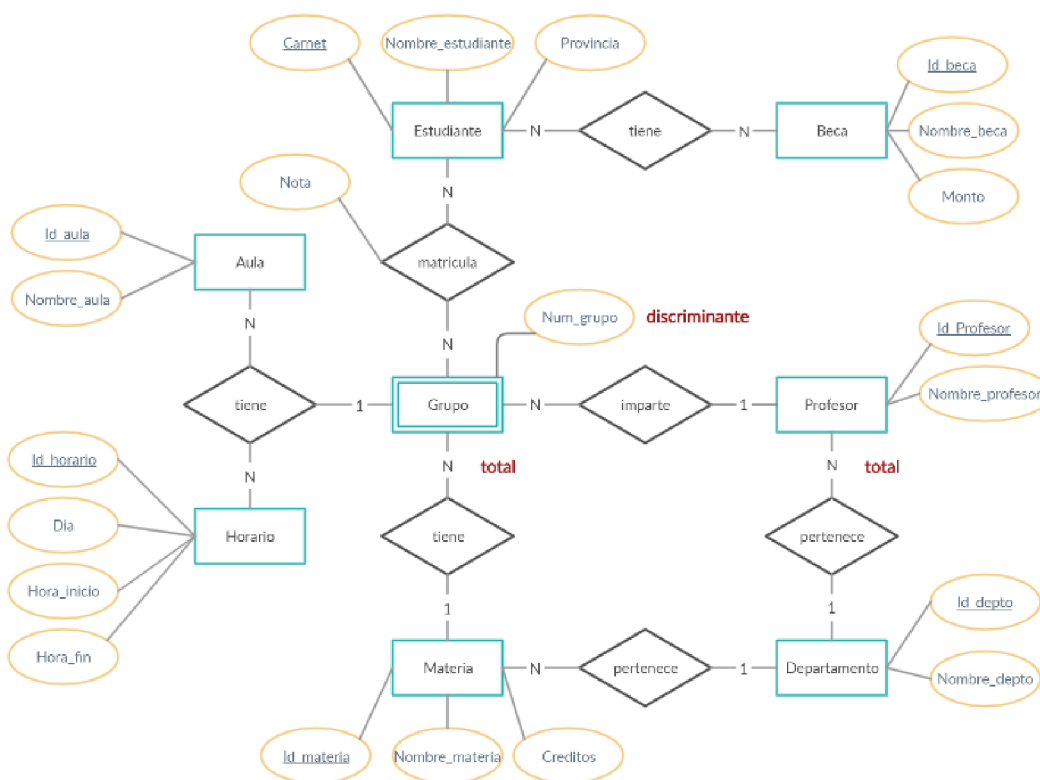


## Especificación de Proyecto

### Sistema de Matrícula

#### Motivación

Con este proyecto se pretende complementar los casos que se han venido trabajando para un Sistema de Matrícula (similar al del TEC) en donde poco a poco se ha ido incluyendo diversos requerimientos para cubrir varias de las principales funcionalidades.



#### Objetivos Formativos

La idea es que hagan uso de todos los conocimientos adquiridos a lo largo del curso para implementar los requerimientos solicitados en este proyecto y así terminar de darle forma al modelo y programar ciertas funcionalidades.

Entre los elementos que podría utilizar están:

- Procedimientos almacenados
- Funciones
- Triggers
- Tablas temporales
- Vistas

## Especificación del proyecto

El modelo de datos del sistema de matrícula que hemos desarrollado hasta el momento solo maneja la información de un periodo (semestre) por lo que se quiere hacer los ajustes necesarios para que permita el manejo de múltiples periodos de modo que se pueda mantener la información de varios años. Por ejemplo, periodo 1-2018, periodo 2-2018, periodo 1-2019, etc.

La otra mejora que se desea incorporar en el modelo es la posibilidad de poder manejar el esquema de requisitos para las materias/cursos, pero solo se tomaran en cuenta los requisitos, no los correquisitos. De modo de que, si algún estudiante intenta matricular un curso, se debe validar si cumple con los requisitos.

Puede usar como referencia el plan de la carrera:

<https://www.tec.ac.cr/planes-estudio/bachillerato-ingenieria-computacion>

⊖ IC3101 - ARQUITECTURA DE COMPUTADORES			
Créditos	Horas	Requisitos	Correquisitos
4	4	IC1400, IC1803	NO HAY CORREQUISITOS EN EL PLAN

⊖ IC2001 - ESTRUCTURAS DE DATOS			
Créditos	Horas	Requisitos	Correquisitos
4	12	NO HAY REQUISITOS EN EL PLAN	IC2101

Tomando como ejemplo las imágenes anteriores, si el estudiante X desea matricular el curso IC3101, debe cumplir el requisito de haber aprobado (nota  $\geq 67.5$ ) los cursos IC1400 e IC1803.

Sobre este modelo se deben construir lo siguiente:

1. Diagrama Entidad-Relación del modelo con los cambios necesarios para que permita (**20 pts.**):
  - a. el manejo de los requisitos de las materias/cursos.
  - b. el manejo de la información de varios períodos.
2. Procedimiento almacenado para realizar la matrícula de un de un estudiante en un grupo específico. En este procedimiento es donde se debe hacer la validación de los requisitos. En caso de que el estudiante no cumpla los requisitos de la materia que desea matrícula entonces se debe mostrar un mensaje significativo en donde se indique cuál es el requisito que tiene pendiente. Para la implementación de esta validación puede usar cualquiera de los elementos vistos en clase. **40 pts.**
3. El promedio ponderado se utiliza para asignar la cita de matrícula, este se calcula de la siguiente forma:
  - Se multiplica la cantidad de créditos de cada curso por la nota obtenida.
  - Se suman todos los resultados del punto anterior y se divide entre la cantidad total de créditos.
$$\sum \text{Nota X Créditos} / \text{Total de Créditos} = \text{Promedio Ponderado}$$

Crear una función que calcule el promedio ponderado (basado en el último período matriculado) para un estudiante dado. Dicha función debe ser invocada por un procedimiento que reciba como parámetro un año e inserte en una tabla temporal la información de todos estudiantes que tengan carné con ese año (carnet, nombre, ponderado). Dicha tabla temporal se consultará después de la ejecución del procedimiento. **20pts.**
4. Scripts para la creación de los objetos de la base de datos junto con todas sus restricciones de integridad, así como la inserción de los datos de prueba. También se deben incluir los scripts para el borrado de todos los objetos creados (en el orden correcto para que no de errores). **10 pts.**

Como entregable en este proyecto se requieren:

1. Diagrama del modelo Entidad-Relación con todos los ajustes necesarios para modelar lo mencionado anteriormente. Detalle de cardinalidad (límites y final), restricciones de integridad, participación, tipos de entidades (fuertes y débiles), tipos de atributos, nombres de las relaciones.
2. Código PL/SQL para la creación de todos los objetos (tablas, vistas, procedimientos almacenados, funciones, triggers) así como para la inserción de los datos de prueba. Los scripts deben incluir todas las restricciones de integridad necesarias.

## Metodología

Analizando los requerimientos planteados se deben utilizar todos los conocimientos adquiridos para adaptar el modelo de la base de datos y desarrollar las funcionalidades solicitadas.

Debe asegurarse de proveer un conjunto de datos para validar apropiadamente las funcionalidades solicitadas. Por ejemplo, en la validación de requisitos, debe haber datos para los diferentes semestres de la carrera. De manera similar para el promedio ponderado.

## Rúbrica

Rubro	Valor
Modelo Entidad-Relación	25%
Código PL/SQL	75%
<b>Total</b>	<b>100%</b>

## Estimación de tiempo

Rubro	Horas
Modelo Entidad-Relación	4
Código PL/SQL	11
<b>Total</b>	<b>15</b>

Proyecto **grupo (máximo 3 personas)**.

Fecha de entrega: 17 de noviembre antes de las 11:59 PM

Desarrollo por parte de los estudiantes fuera del horario de clases.

## Aspectos Generales

- Los entregables deben hacerse en formato .PDF y el código en un .SQL
- No se aceptan entregas después de la fecha y hora especificada.