



# Prueba de Desempeño ASP.NET

Clan: Linus

## Reglas de la prueba

- **Comunicación:** Está prohibido hablar o comunicarse de cualquier forma con otros estudiantes durante el examen.
- **Integridad Académica:** Cualquier forma de trampa, incluido el plagio, copia o uso de material no autorizado, resultará en una calificación de cero en el examen y puede llevar a sanciones adicionales según las políticas de RIWI.
- **Material Permitido:** Solo se permite ver material de apoyo como lo son diapositivas o ejercicios realizados en clase, si se requiere ver estos materiales de apoyo se debe comunicar al Trainer.
- **Permanencia en el Aula:** Una vez iniciado el examen, no se permite salir del aula hasta haber entregado el examen y, de preferencia, hasta que haya transcurrido al menos la mitad del tiempo asignado.
- **Entrega:** Una vez finalizada la prueba se debe subir el entregable en moodle, donde debe estar el código comprimido, más la respectiva documentación y el link del repositorio de GitHub.

## Introducción:

Las escuelas enfrentan el desafío de gestionar de manera eficiente la información de sus alumnos, profesores, cursos y matrículas. Un sistema manual o fragmentado puede generar ineficiencias, errores y pérdida de datos importantes. Un sistema centralizado y automatizado permitiría optimizar la gestión de la escuela y mejorar la experiencia de los alumnos, profesores.

## Objetivo:



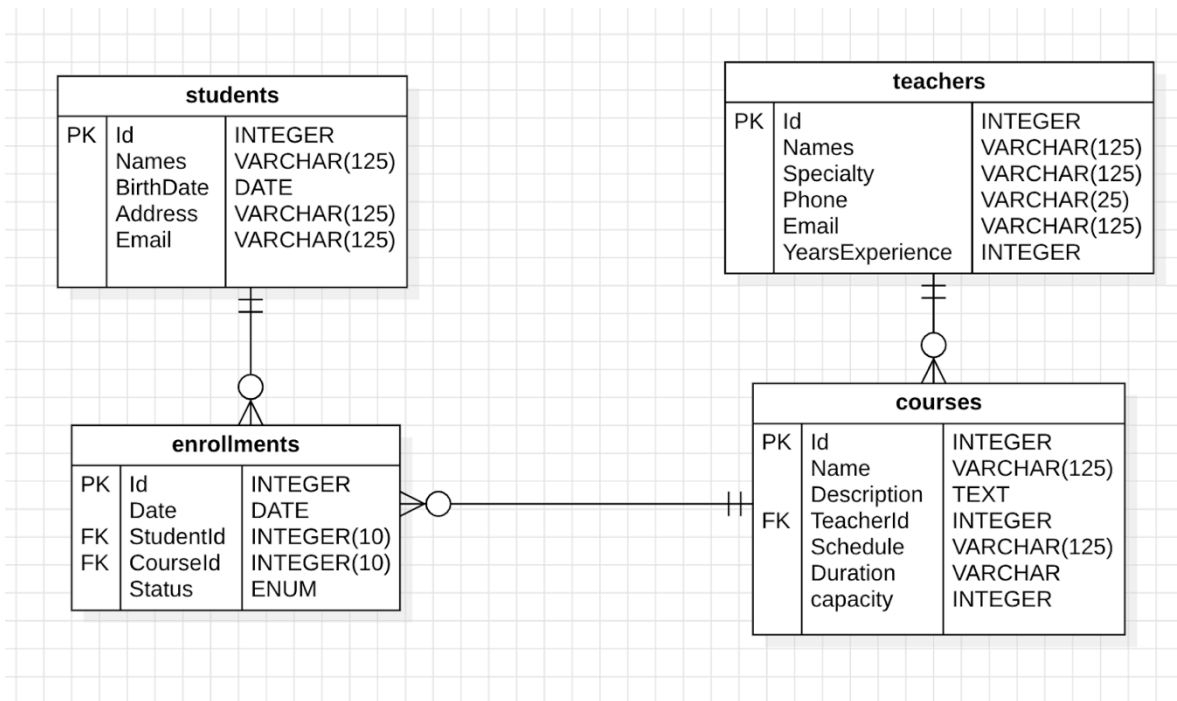
Desarrollar servicios web para la gestión integral de una escuela, incluyendo la administración de alumnos, profesores, cursos y matrículas. El sistema debe facilitar el almacenamiento, consulta y actualización de datos, así como mantener la integridad de las relaciones entre las diferentes entidades.

#### Entidades principales:

- **Alumno:** Información sobre los alumnos de la escuela, incluyendo nombre completo, fecha de nacimiento, dirección, teléfono, correo electrónico.
- **Profesor:** Información sobre los profesores de la escuela, incluyendo nombre completo, especialidad, teléfono, correo electrónico y años de experiencia.
- **Curso:** Información sobre los cursos que se ofrecen en la escuela, incluyendo nombre del curso, descripción, profesor a cargo, horario, duración y capacidad máxima.
- **Matrícula:** Información sobre las matrículas de los alumnos en los cursos, incluyendo fecha de matrícula, alumno matriculado, curso matriculado y estado de la matrícula (**pagada, pendiente de pago, cancelada**).

#### Relaciones:

- Un **alumno** puede estar **matriculado** en uno o más **cursos**.
- Un **curso** puede tener uno o más **alumnos matriculados**.
- Un **curso** está a cargo de un único **profesor**.
- Una **matrícula** está asociada a un **alumno** y a un **curso** específico.



## Criterios de aceptación generales.

1. Se debe realizar la prueba en **ASP.NET , Entity Framework Core**
2. Todos los Endpoints deben estar documentados utilizando **Postman**.
3. Se deben realizar relaciones entre modelos. (1:1) (1:N) (N:1) (N:M)
4. La información debe venir de la base de datos **MySQL**, por lo tanto debe existir una conexión.
5. Todas las entradas **deben tener validaciones** para que no afecte a la base de datos.
6. Si no se encuentra algún registro, se debe manejar el error y responder con su mensaje y su respectivo estatus de error, evitando los Errores 500.
7. Se debe Aplicar Principio S de SOLID (principio de responsabilidad única o SRP).
8. Repositorio en GitHub.



## Entregables

- Código fuente comprimido en zip de la solución
- Archivo leeme.txt con el enlace del repositorio en GITHUB
- Export de la colección en Postman

Se recomienda dejar datos de prueba en la base de datos de clever cloud y no eliminarla ya que se utilizará para evaluar el funcionamiento de los endpoints.