

# Prog. Orientada a Objetos

## CFL Programador full-stack

*Ejercicios para Practicar - Parte 2*

# Prog. Orientada a Objetos

## CFL Programador full-stack

*Ejercicios*

# Parte 1 - Diseño de Sistemas

- Implementar un sistema de gestión de legajos para un colegio. Dicho sistema debe llevar registro de los alumnos. A su vez, cada alumno, además de sus datos personales, tiene un listado de exámenes rendidos. El sistema debe proveer la siguiente funcionalidad:
  - Obtener el promedio general de un determinado alumno
  - Obtener el promedio de todos los alumnos inscriptos en el sistema
  - El alta de los alumnos a elección: a partir de teclado, o creandolos directo en el código

## Parte 2 - Documentación

- A partir del código TypeScript, plantear el diagrama de clases asociado
- Tener en cuenta las relaciones entre clases
  - Herencia
  - Implementación
  - Composición
- Tener en cuenta los modificadores de acceso en las variables y métodos
  - + → public
  - - → private
  - # → protected
- Partir del código disponible en el [repo](#)

# Parte 3 - Preguntas Teóricas

- Responder las preguntas en un TXT y subirlo a GitHub
  - ¿Qué pasa cuando se corre el comando “tsc”?
  - ¿Cómo está organizada una clase?
  - ¿Es correcto que un método auxiliar de una clase sea público? ¿Por qué?
  - Enumerar las características de una interfaz