

### CONTENIDOS

- Implementación de ASP WebApi
- Separación de responsabilidades

1) Copie el siguiente enlace en su navegador: <https://classroom.github.com/a/ty0fKk0T> esto creará el repositorio para poder subir el **Trabajo Práctico Nro. 4**, realice los pasos ya aprendidos para *clonar* el repositorio en su máquina y poder comenzar a trabajar de forma *local*.

#### 2) Migración del Sistema para Cadetería

Continuando con el requisito solicitado por el cliente, en esta etapa se migrará la aplicación a un servidor web, utilizando API REST para acceder a las funcionalidades del sistema.

Para poder cumplir con dicho requisito se decidió utilizar una Web Api de ASP.

- a) Cree un nuevo proyecto ASP Web API (**dotnet new webapi**)
- b) Copie al proyecto todas las clases del TP 3 en una nueva carpeta llamada Models.
- c) Cree un Controlador Para la cadetería llamado **CadeteriaController**, y en el Implemente un endpoint para cada una de las operaciones ya existentes, siguiendo la siguiente forma:
  - [Get] GetPedidos() => Retorna una lista de Pedidos
  - [Get] GetCadetes() => Retorna una lista de Cadetes
  - [Get] GetInforme() => Retorna un objeto Informe
  - [Post] AgregarPedido(Pedido pedido)
  - [Put] AsignarPedido(int idPedido, int idCadete)
  - [Put] CambiarEstadoPedido(int idPedido, int NuevoEstado)
  - [Put] CambiarCadetePedido(int idPedido, int idNuevoCadete)

Recuerde que debe siempre debe retornar una respuesta a cada petición, respetando los códigos de estado vistos en clase.

Para la persistencia puede emplear una clase estática como la vista en clases.

3) Opcional: Descargue la aplicación para probar APIs Postman (<https://www.postman.com/downloads/>) para probar los endpoints implementados en el punto anterior.