

CONTENIDOS

- Implementación de POO
- Asociación entre clases, manejo de composición, agregación
- Niveles de acceso, abstracción y ocultamiento de la información
- Métodos, atributos y propiedades

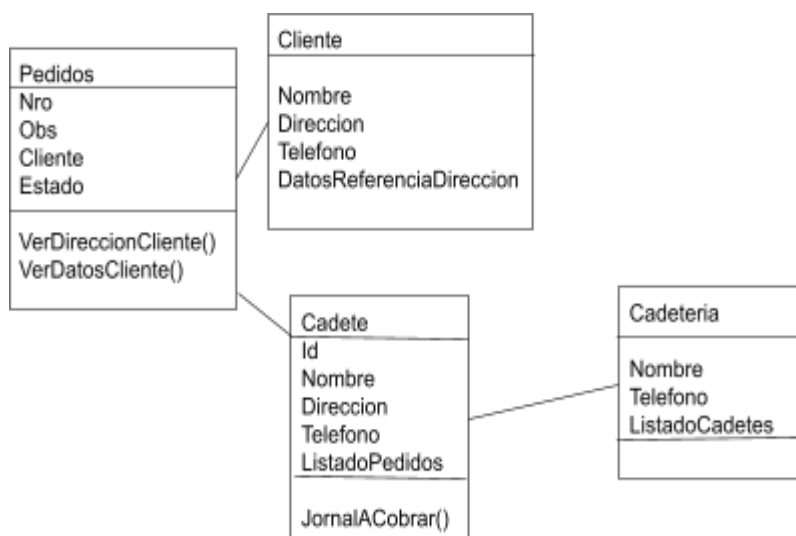
1) Copie el siguiente enlace en su navegador: <https://classroom.github.com/a/yIIzTnUy> esto creará el repositorio para poder subir el **Trabajo Práctico Nro. 1**, realice los pasos ya aprendidos para *clonar* el repositorio en su máquina y poder comenzar a trabajar de forma *local*.

2) Sistema para una Cadetería

Una empresa de cadetería necesita implementar un sistema para asignar pedidos a sus cadetes y poder luego saber cuántos pedidos despachó cada uno para poder así pagarles su correspondiente jornal (\$500 por cada pedido Entregado)

Tenga en cuenta que: Cada Pedido tiene un Cliente y cada Cadete puede tener uno o más pedidos. Si se elimina un pedido entonces el cliente tiene que eliminarse también. Un pedido puede reasignarse a otro Cadete. Es necesario generar informes sobre la actividad de la cadetería.

2.a A partir del siguiente diseño de clases *incompleto*, responda las preguntas planteadas a continuación en el archivo Readme.md:



- ¿Cuál de estas relaciones considera que se realiza por composición y cuál por agregación?
- ¿Qué métodos considera que debería tener la clase Cadetería y la clase Cadete?
- Teniendo en cuenta los principios de abstracción y ocultamiento, que atributos, propiedades y métodos deberían ser públicos y cuáles privados.
- ¿Cómo diseñaría los constructores de cada una de las clases?
- ¿Se le ocurre otra forma que podría haberse realizado el diseño de clases?

2.b Implemente el sistema de cadetería solicitado utilizando como base el diseño de clases sugerido, tenga en cuenta que:

- A) Debe agregar los métodos faltantes a las clases en función de sus respuestas del punto anterior.
- B) Los datos de la cadetería y de sus cadetes deberán ser cargados automáticamente a partir de 2 archivos csv, uno por cada entidad.
- C) El sistema posee una interfaz de consola para gestión de pedidos para realizar las siguientes operaciones:
 - a) dar de alta pedidos
 - b) asignarlos a cadetes
 - c) cambiarlos de estado
 - d) reasignar el pedido a otro cadete.
- D) Mostrar un informe de pedidos al finalizar la jornada que incluya el monto ganado y la cantidad de envíos de cada cadete y el total. Muestre también la cantidad de envíos promedio por cadete.

Nota 1: Utilice listas para guardar y manipular los datos correspondientes. Investigue sobre el uso de sentencias **Linq** para realizar las operaciones del informe.