

2. Implementación del patrón **Strategy** para contar con la posibilidad de tener diferentes estrategias al momento de elegir un empleado para atender un cliente en un taller cerámico (2.5 puntos).

Use del archivo Repositorio.cs las clases *Cliente* y *Mecanico* y las interfaces *INegocio*, *IEmpleado* e *ICliente*.

**NOTA:** El archivo Repositorio.cs **NO DEBE** ser modificado.

Para ello:

- a. Implemente la estrategia *EstrategiaMejorAPeor* que elige, entre todos los empleados disponibles, el de mejor experiencia.
- b. Implemente la estrategia *EstrategiaPeorAMEjor* que elige, entre todos los empleados disponibles, el de peor experiencia.
- c. Implementar una subclase de *Negocio*, llamada *TallerChapaYPintura*, y reimplemente el método *algunEmpleado* para que devuelva el mejor o el peor empleado, según la estrategia.
- d. Implemente una clase *Programa* con la siguiente función estática:  
*void atender(INegocio taller, ICliente cliente)*  
que invoca al método *llegaCliente* del taller con el cliente pasado por parámetro.

Realice una función *Main* en la que el Programa que instancie un *TallerDeChapaYPintura* con seis *Mecanico* con experiencias distintas. Setee al *TallerDeChapaYPintura* la estrategia *EstrategiaMejorAPeor* e invoque a la función implementada en d) tres veces con tres *Cliente* distintos. Setee al *TallerChapaYPintura* la estrategia *EstrategiaPeorAMEjor*, invoque a la función implementada en d) 3 veces con tres *Cliente* distintos.