

Marco Teórico:

1. ¿Por qué un Objeto debería tener una sola Responsabilidad?
2. Que entiende por Extensión y modificación.
3. ¿Que es Inversión de Dependencia, que permite?, como se implementa?
4. ¿Que es GRASP? ¿Por qué debe aplicarlo al diseño?
5. ¿Dar 3 maneras de Evitar Memory Leaks?
6. ¿Que solucionan los Smart Pointers?
7. ¿Cuándo realizaría una refactorización?
8. ¿Que entiende por pruebas unitarias?
9. ¿Explicar la idea de TDD?
10. ¿Cuándo debería usar un Doble de Prueba?

Marco Practico

Una Empresa que vende “cosméticos” nos convoca para la realización de una App que permita registrar los Pedidos de Artículos de los clientes.

La misma debe permitir:

1. Cargar Artículos.
2. Cargar Clientes
3. Cargar Pedidos de Artículos Asignados a los Clientes.
4. Mostrar los pedidos de los Clientes.

Realizar el Diagrama de Clases.

La aplicación deberá realizarse Implementando los Objetos Correspondientes.

Deberá mostrar por consola los pedidos de los clientes.

Se aconseja usar clases y un Array de Objetos.

Aplicar RAI, IOC, SOLID y GRASP

Desarrollar primero una Prueba Unitaria (puede prescindir del GTest, es a efectos evaluativos.)

Luego implementar la Solución y Refactorizar.

Nota: Tome nota de lo que se pide. Responda las consignas como se han establecido. No “haga” de más, tampoco de “Menos”,

Para aprobar será necesario contar con más del 60% de las consignas aprobadas.

La cátedra