

☺ **LEER POR LO MENOS DOS VECES LOS ENUNCIADOS ANTES DE RESOLVER**

Leé con cuidado y por lo menos dos veces el enunciado para detectar las clases y estructuras necesarias para cumplir con lo pedido. Leé **todo**: el enunciado, el mismo termina al final de la hoja. Pensá bien la estrategia de resolución antes de comenzar el desarrollo.

El objetivo de este examen es **evaluar la correcta aplicación de los conceptos** que se asimilaron a lo largo del cuatrimestre en un ejercicio práctico que los combina. Estos conceptos son:

- Detección de clases, atributos, métodos y relaciones (asociativas y de uso).
- Modularización reutilizable y mantenible usando métodos con correcta parametrización.
- Asignación de responsabilidades a cada clase y encapsulamiento con *setters* y *getters* con visualización adecuada.
- Manejo del concepto de *instancia* y de la interacción entre objetos.
- Manipulación de listas de objetos (ArrayLists) y su uso en ciclos condicionales y *for-each*.
- Manejo de diagramas Nassi-Schneiderman y UML de clases.

Para aprobar el examen se requiere:

- Un diseño UML aceptable que permita resolver lo pedido.
- La implementación de los métodos pedidos.
- La correcta declaración de las firmas/encabezados de todos los métodos pedidos.
- La resolución implementada no debe contradecir el diseño de clases presentado.

Enunciado:

Una empresa que fabrica y vende todo tipo de ropa, posee varias filiales, que tiene cada una de ellas vendedores que se dedican a vender dichos productos por el gran Buenos Aires y por el interior del país. Cada sucursal tiene su propio stock de artículos, contando a su vez con muchos vendedores y cada uno de ellos genera pedidos. Un vendedor pertenece a una única sucursal y cada sucursal tiene sus números propios de pedido. Cada vendedor, aparte de sus datos, tiene un código (cada sucursal numera a sus vendedores), y una lista de clientes, cada cliente puede tener solo a un único vendedor. Cada pedido tiene un código, una marca si pudo o no satisfacerse, y puede constar de muchos artículos que lo componen con sus respectivas cantidades.

Se pide:

- Diagrama de Clases que represente el enunciado descripto.
- Desarrollar el método `DespacharPedido`, de tipo entera, que dada el código de la sucursal y el código de pedido debe devolver lo siguiente:
 - “1” El pedido pudo satisfacerse perfectamente.
 - “2” El pedido no existe como pedido en la sucursal.
 - “3” El pedido tiene al menos un artículo que no puede satisfacerse por falta de stock.Si el pedido puede satisfacerse, se le descontará a cada artículo pedido de la sucursal la cantidad pedida y se marcará a dicho pedido como satisfecho.
- Desarrollar el método `Agregar_Articulo_A_Pedido`: que recibe un número de pedido, código de artículo y cantidad pedida y se debe agregar a dicho artículo en la lista de artículos que tiene el pedido. Teniendo en cuenta si el artículo ya existía previamente en dicho pedido, se le sumará dicha cantidad a la cantidad que tenía dicho artículo en dicho pedido.