

Leé con cuidado el enunciado y por lo menos dos veces para resolver lo pedido. Pensá bien la estrategia de resolución antes de comenzar el desarrollo de lo que te solicitan. El objetivo de este examen es **evaluar la correcta aplicación de los conceptos y técnicas** vistos hasta el momento:

- Correcta implementación de constructores.
- Modularización reutilizable y mantenible con uso de métodos con correcta parametrización y correcto encapsulamiento, publicando *setters* y *getters* sólo cuando corresponda.
- Manejo de clases, enumerados y colecciones.

Enunciado

YaEmpanadas es una aplicación que complementa el trabajo de una cadena existente en esta ciudad.

La aplicación registra y guarda cada uno de los pedidos que recibe. Cada pedido tiene el número de teléfono de quien hace el pedido y una dirección física donde se enviarán las empanadas.

También, cada pedido tiene uno o más ítems. En cada ítem se guarda la cantidad de empanadas que se piden de cada sabor. Los sabores disponibles son fijos: CARNE, POLLO, JAMÓN Y QUESO, VERDURA, HUMITA.

Se pide:

- Diseñar el diagrama UML completo que resuelva el enunciado propuesto.
- Desarrollar en NS+ los siguientes puntos (incluyendo los métodos derivados):
 - El constructor de **YaEmpanadas**.
 - Estos métodos públicos:
 - crearORecuperarPedido: Recibe un teléfono y una dirección. A partir de estos datos, busca un pedido existente para ese teléfono. Si no lo encuentra, lo registra. Devuelve el número del pedido, el cual será necesario para la carga de las empanadas y para mostrar el detalle del pedido. El número de pedido se debe incrementar automáticamente con cada pedido creado.
 - cargarEmpanadas: Recibe un número de pedido (el pedido debe existir), un sabor y una cantidad (debe ser mayor que cero).
 - Carga la cantidad de unidades y el sabor en un ítem del pedido según estas reglas:
 - Si el ítem existe (uno con el mismo sabor), **reemplaza** el valor actual por el nuevo. Solo acepta cantidades mayores que cero.
 - Si el ítem no existe, lo agrega a los existentes. También debe controlar la cantidad de unidades (como mínimo una empanada).
 - Si hay un error debe emitir un mensaje de error acorde al error producido. Si todo salió bien debe mostrar un aviso de que el pedido fue actualizado.
 - listarPedidoCompleto: Recibe un número de pedido y lista toda la información referida a este pedido. Si el pedido no existiera debe mostrarse un mensaje de error.
 - Desarrollar también los métodos privados y/o públicos de las clases existentes que creas necesarios para resolver lo pedido.