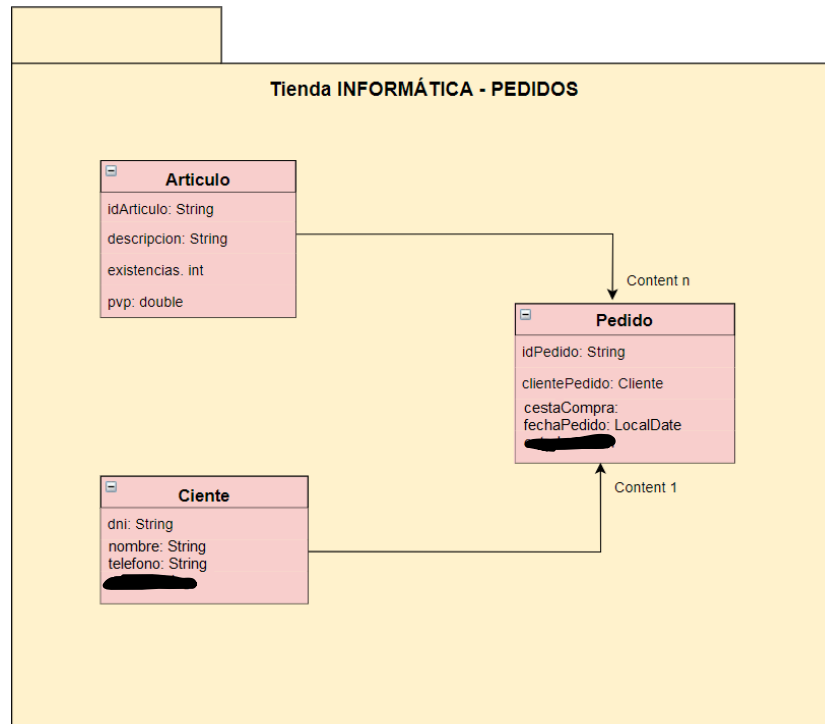


## TAREA CALIFICADA PROGRAMACIÓN – D1IFC303 – CUATRIMESTRE 2/2024

**ENUNCIADO:** Realizar un proyecto Java para gestionar Pedidos en una tienda de Informática. Las clases VO serán las de la figura, quedando abierta la forma en la que cada alumno/a implementa el atributo **cestaCompra** de la clase **Pedido**, teniendo en cuenta que un pedido puede incluir **varios artículos** y **varias unidades** de cada artículo pedido.



### GESTIÓN DE CLIENTES

- Hacer un submenú que permita añadir/listar/modificar/borrar **Clientes** (CRUD)
- **Modificar** se refiere únicamente a proporcionar la opción de modificar el **tel** o el **email** de un Cliente.
- La clase **Cliente** será **Comparable** y en los listados los clientes saldrán ordenados por **nombre**.

### GESTIÓN DE ARTÍCULOS

- Hacer un submenú que permita añadir/listar/borrar/**reponer** **Artículos** (CRUD).
- **reponer** se refiere a modificar el número de unidades en Stock del artículo por entrada de nuevas unidades al almacén. Al seleccionar esa opción, antes de solicitarnos los IdArticulo para añadir nuevas unidades, el programa debe mostrar un listado con todos los artículos con **0** unidades
- Para listar los artículos se ha de proporcionar la opción de listarlos **TODOS**, o hacerlo por **SECCIÓN**:  
**1.PERIFÉRICOS 2.ALMACENAMIENTO 3.IMPRESORAS 4.MONITORES 5.COMPONENTES**

No hay un atributo específico para indicar la **sección** de un artículo. No hace falta, pues esto lo vamos a obtener directamente con **el primer carácter del idArticulo**. Los artículos "1-01 1-12" pertenecen a la sección PERIFÉRICOS. "4-03 4-14" a la sección MONITORES, y así con el resto de secciones.

- La clase **Artículo** será **Comparable** y en los listados los artículos saldrán ordenados **por defecto** por **idArticulo**. No obstante, se dará la posibilidad en los listados de **Artículos** (todos o por sección) de cambiar ese criterio y ordenar los artículos por **precio**, de **menor a Mayor** o de **Mayor a menor**.

## GESTIÓN DE PEDIDOS

- Cada nuevo pedido tendrá un **idPedido** auto-generado consecutivamente, de la forma **0001/2023**, **0002/2023**, etc.
- Un pedido estará compuesto por un atributo **cestaCompra** de tipo **Colección** (preferentemente algún tipo de List) en dónde se irán añadiendo los **idArticulo** de los artículos pedidos y las **unidades** pedidas de cada artículo. Cuando se emite un pedido por primera vez siempre su estado inicial será "PREPARACION"

## EXCEPCIONES

- Hay que codificar clases para excepciones propias y hay que lanzar/avisar/manejar esas excepciones con **Throw, Throws y try-catch** en los siguientes casos.
  - a. **dni** de un cliente no encontrado en la tienda.
  - b. **idArticulo** de un Artículo no encontrado en la tienda.
  - c. **UnidadesInsuficientes** de un artículo para satisfacer la petición de un usuario. Cuando esta excepción suceda se debe de ofrecer al usuario la posibilidad de adquirir las unidades disponibles.
  - d. **StockAgotado de un artículo**. Cuando el número de unidades de un artículo sea 0, se debe lanzar un mensaje avisando de la necesaria reposición.