

---

## RELACIÓN DE EJERCICIOS 12 - Colecciones: ArrayList y HashSet

---

Resolver los siguientes problemas escribiendo el algoritmo con lenguaje Java.

Crear en GitHub un repositorio llamado **UD6-Colecciones** para subir cada uno de los ejercicios de la relación.

1. (ArrayList) Crea una clase **Menu**, que dispondrá de un atributo arrayList de String. Tendrá un método **creaOpcion** que permitirá crear las distintas opciones del menú, otro método **muestraMenu** que mostrará todas las opciones del menú, y un método **captura opcion** que devolverá la opción seleccionada por el usuario. Prueba dicha clase en otra clase denominada **TestMenu**. La clase **Menu** la utilizaremos en el resto de ejercicios que deban usar un menú de opciones
2. (ArrayList). Crea un nuevo proyecto denominado **EjerciciosArrayList**, donde se crearán las siguientes clases:
  - Crea la clase **Articulo**, con la siguiente información: código, descripción, precio de compra, precio de venta, stock. Los atributos serán privados y se crearán los métodos get, set necesarios. El constructor recibirá el código del artículo y la descripción.
  - Crea una clase **AlmArticulo**, dicha clase tendrá un atributo que será un arrayList de artículos. La clase dispondrá del método **almacena** que recibirá un artículo y lo insertará en el arrayList. Sobrecarga el método **toString** de la clase **Articulo** para que realice la impresión de todos los artículos del arrayList.
  - Crea la clase **GestSimAlm** para llevar el control de los artículos de un almacén. De cada artículo se debe saber el código, la descripción, el precio de compra, el precio de venta y el stock (número de unidades). Utilizaremos la clase **Articulo** del apartado anterior y almacenaremos todos los artículos de nuestro almacén en un arrayList. La clase dispondrá de los siguientes métodos:
    - **entradaAutomatizada**: creará 4 artículos de prueba del programa.
    - **lista**: Mostrará todos los datos de los artículos del almacén.
    - **alta**: Creará un nuevo artículo en el almacén.
    - **baja**: Dará de baja un artículo en el almacén.
    - **entradaMercancia**: Aumentará el stock de un determinado artículo
    - **salidaMercancia**: Disminuirá el stock de un determinado artículo. Hay que controlar que no se pueda sacar más mercancía de la que hay en el almacén.
  - Realiza la prueba de la clase anterior, para ello realiza un programa (clase **TestGestSimAlm**) que muestre el siguiente menú para que el usuario pueda seleccionar la opción elegida.
    0. Entrada automatizada (creará 4 artículos de prueba del programa)
    1. Listado (Mostrará todos los datos de los artículos del almacén)
    2. Alta (Crearé un nuevo artículo en el almacén)
    3. Baja (Daré de baja un artículo en el almacén)
    4. Entrada de mercancía (Aumentaré el stock de un determinado artículo)
    5. Salida de mercancía (Disminuiré el stock de un determinado artículo)
    6. Salir
3. (ArrayList). En el ejercicio 8 de la relación 11 construimos una biblioteca con libros y revistas. Ahora

---

## RELACIÓN DE EJERCICIOS 12 - Colecciones: ArrayList y HashSet

---

queremos que los objetos de las clases **Libro** y **Revista** puedan ser almacenados y ordenados en un arrayList por su título. Para ello debemos implementar la interfaz **Comparable** en dichas clases. Modifica el ejercicio 8 para que se cumplan estos requerimientos. Crea una clase **TestPublicacion**, donde realizaremos la prueba de este nuevo ejercicio. Debemos almacenar al menos tres publicaciones y ordenarlas dentro de un arrayList.

4. (HashSet). Realiza un pequeño programa que pregunte al usuario 5 números diferentes (almacenándolos en un HashSet), y que **DESPUÉS** calcule la suma de los mismos (usando un bucle for-each).