

# Ejercicio “Practica 7”

---

Juan Bautista Arellano Bruno  
Francisco Blázquez Matías  
Víctor Castilla Garrido  
David Flores Barbero

**Grupo PB1**

Equipo Echo

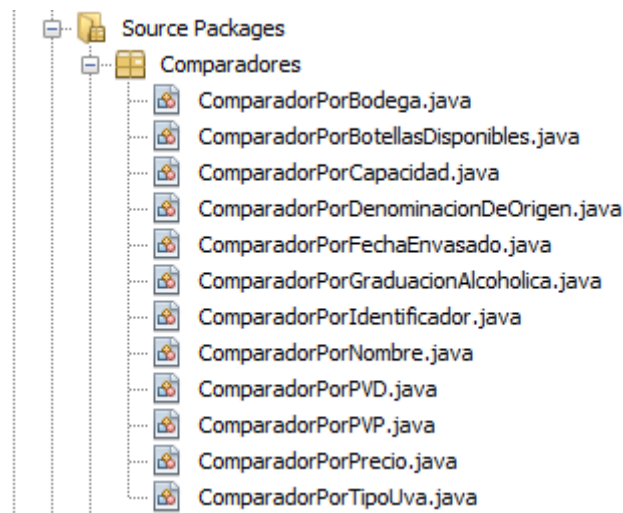
## Introducción:

Programa en Java que consiste en leer desde un directorio una serie de ficheros con extensión “.csv” y hacer un análisis de ellos. Los datos de estos ficheros son de un almacén de vinos.

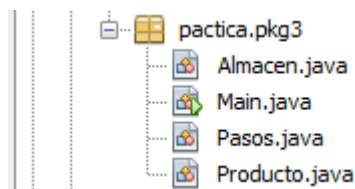
## Estructura:

Este programa consta de 2 paquetes (comparadores y pactica.pkg3)

comparadores: Contiene los comparadores de cada campo de cada almacén



pactica.pkg3: Contiene las clases “almacen” y “producto”, además del “main” y de las clases auxiliares.



Dentro de “Producto” se encuentran los atributos de cada uno de los productos con sus getter y setter.

Dentro de “Almacen” se encuentra un arrayList de objetos “Producto”, la ruta del archivo del almacen y su precio total. Además de los métodos setter y getter

Dentro de “Pasos” se encuentran las clases que ejecutan el “main” además de los getter y setter de la propia clase.

```
/** Aporta la ruta al directorio donde se va a enco
public void rutaAlEscritorio() { ...16 lines }

/** Pide al usuario que introduzca por teclado el r
public void pedirCarpeta() { ...50 lines }

/** Inicia el atributo ruta de cada almacen para pe
public void iniciarRutaALosArchivos() { ...15 lines }

/** Leemos uno a uno cada uno de los productos que
public void leerDatos() { ...21 lines }

/** Presenta todos los datos de todos los almacenes
public void presentarTodosAlmacenes() { ...82 lines }

/** Ordena la salida de datos por pantalla mediante
public void ordenarAlmacen() { ...121 lines }

/** Presenta por pantalla un unico almacen ...4 lin
public void presentarUnAlmacen(int x) { ...30 lines }

/** Ordena todos los almacenes por PVP ...3 lines *
public void ordenarTodoPorPVP() { ...10 lines }

/** Ordena todos los almacenes por precio ...3 line
public void ordenarPorPrecio() { ...8 lines }

/** Presenta por pantalla los atributos que se pide
void presentarSegundaForma() { ...54 lines }

/** Presenta por pantalla los almacenes ya ordenado
void presentarTerceraForma() { ...9 lines }

/** Presenta el menu para la eleccion de un almacen
void contarDenominacionDeOrigen() { ...48 lines }
```

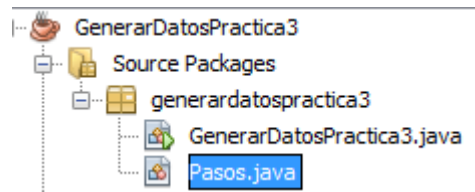
Y por último el “main” en el cual no cabe destacar nada interesante.

La ejecución del programa visualiza los siguientes objetivos en forma de tabla encolumnada:

- Visualización de los productos de todos los almacenes ordenados por almacenes.
- Visualización de los productos de un único almacén ordenados a gusto del usuario.
- Visualización de algunos campos de los productos de los almacenes ordenados por PVP
- Visualización de los almacenes ordenados por precio.

Además de el número de botellas de una denominación de origen a elegir por el usuario.

NOTA IMPORTANTE: Para generar los datos hemos hecho un programa suplementario el cual genera archivos “.csv” aleatorios dentro de un directorio.



```
/** Metodo con el que vamos a crear una ruta a nuestro escritorio ...3 lines */
public void crearRutaAlEscritorio() {...16 lines }

/** Pide al usuario que introduzca por teclado la carpeta donde desea crear todos los datos
public void seleccionarCarpeta() {...17 lines }

/** Pide el numero de almacenes a crear y crea un numero de productos por cada uno de ellos
public void pedirInformacion() {...38 lines }

/** Crea los archivos "*" ...3 lines */
public void crearArchivos() {...15 lines }

/** Modificamos la ruta para poder acceder a la carpeta donde vamos a crear los datos ...3 l
public void crearRutaACarpeta() {...5 lines }

/** Escribe los datos en los ficheros "*" ...3 lines */
public void iniciarArchivos() {...37 lines }

/** Genera el dato a guardar en el archivo aleatoriamente escogiendo un valor ...6 lines */
public String devolverCampo(String[] vector) {...6 lines }
```

Para más información leer los comentarios del código del programa.