

# Ejercicio Almacén Deportivo

---

*Francisco Blázquez Matías*

**Grupo PB1**

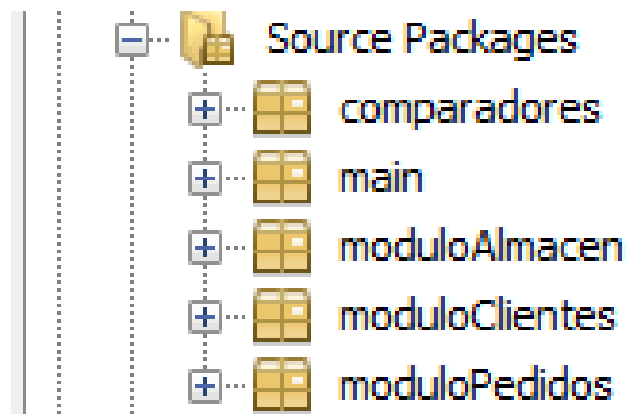
Equipo Echo

## Introducción:

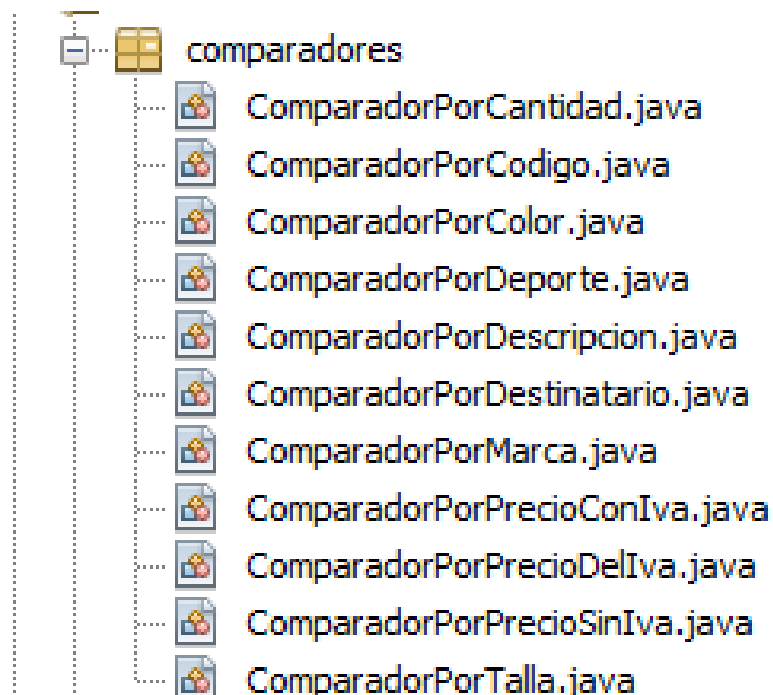
Programa de Java realizado en NetBeans que simula el almacén on-line de una tienda de deportes.

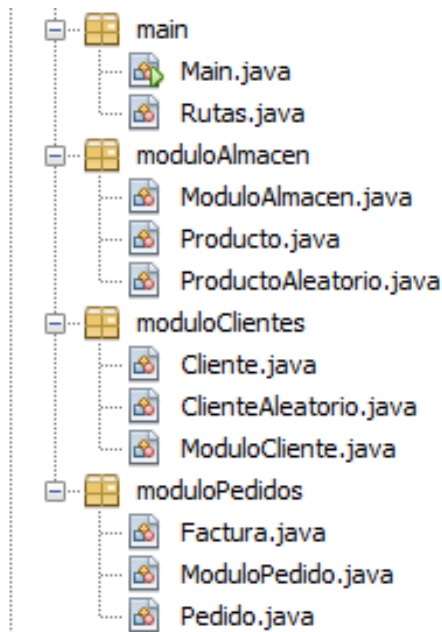
## Estructura:

El programa está dividido en 5 paquetes:



Cada paquete tiene sus respectivas clases:





- En el paquete de “comparadores” se encuentran las clases que realizan una comparación para ordenar el almacén por un campo especificado.
- En el paquete “main” se encuentra el main del programa y además una clase que contiene las rutas para enlazar además de los métodos setter y getter con los nombres de los directorios.

```

public class Rutas {

    private static String nombreCarpeta;
    private final String carpetaSeguridad = "backup";
    private final String carpetaAlmacen = "articulos";
    private final String carpetaPedidos = "pedidos";

    /** Proporciona la ruta al escritorio ...4 lines */
    public static Path pathToDesktop() {...15 lines }

    /** Proporciona la ruta a un directorio a situada en el es
    public static Path pathToFolderOnDesktop(String nombreDire

    /** Proporciona la ruta a un fichero situado en el escrito
    public static Path pathToFileOnDesktop(String nombreFicher

    /** Proporciona la ruta a un fichero situado en un directo
    public static Path pathToFileOnFolderOnDesktop(String nomb

    /** Proporciona la ruta a un fichero situado en un subdire
    public static Path pathToFileOnFolderOnFolderOnDesktop(Str

    /** Crea o enlaza las rutas a la carpeta del escritorio .
    public void crearCarpetaPrincipal() {...43 lines }
  
```

- Dentro del paquete “moduloAlmacen” hay tres clases:
  - “Producto” contiene todos los atributos que tendrá un producto, además de sus métodos setter y getter.

```
public class Producto {

    private String codNum;
    private String descripcion;
    private String marca;
    private String deporte;
    private String destinatario;
    private String color;
    private String talla;
    private String precioSinIva;
    private String cantidad;
    private String precioIva;
    private final String iva = "21";
    private String precioConIva;

    /*METODOS SETTER AND GETTER*/

    /**...4 lines */
    public String getCodNum() { ...3 lines }

}
```

- “ProductoAleatorio” contiene una serie de vectores con parametros aleatorios y un método para generar un campo aleatorio partiendo de uno de esos vectores.

```
public class ProductoAleatorio {

    private final String[] deporte = {"Futbol", "Tenis", "Basket", "Caza", "Hockey", "Ba",
                                     "Atletismo", "Gimnasia"};
    private final String[] destinatario = {"Hombre", "Mujer", "Junior", "Niño"};
    private final String[] talla = {"XS", "S", "L", "M", "L", "XL", "XXL", "XXXL"};
    private final String[] color = {"Rojo", "Azul", "Verde", "Amarillo", "Beige", "Negro",
                                   "Naranja", "Ocre", "Morado"};
    private final String[] descripcion = {"Botas", "Tutu", "Camiseta Corta", "Camiseta L",
                                          "Bañador", "Casco", "Zapatillas", "Brazalete", "Mallas", "Guantes"};

    private final String[] palabraInutil = {"Cuel", "RealTy", "Confit", "Etyras", "Tirk",
                                             "UnderT", "Xrit"};

    /** Retorna el una posicion aleatoria del vector que se le pasa por parametro ...5 1
    public String campoAleatorio(String[] campo) { ...6 lines }

    /*METODOS SETTER AND GETTER*/

    /** ... 4 lines */

}
```

- “ModuloAlmacen” contiene los métodos correspondientes a las opciones del menú del almacén además del ArrayList de productos.

```
public class ModuloAlmacen {

    private static ArrayList<Producto> listaProductos = new ArrayList<>(); //Lc
    private int numProductos;
    private final String[] productoAtributos = {"Codigo", "Descripcion", "Depor
    private String ficheroAlmacen;
    private final String delimitador = ";";

    public void menuAlmacen() { ...76 lines }

    /** Genera aleatoriamente productos para el almacen ...3 lines */
    public void almacenAleatorio() { ...59 lines }

    /** Genera un archivo "*" ...3 lines */
    public void exportarAlmacen() { ...54 lines }

    /** Importa desde un archivo "*" ...3 lines */
    public void importarAlmacen() { ...68 lines }

    /** Muestra todos los productos del almacen encolumnados ...3 lines */
    public void mostrarAlmacen() { ...43 lines }

    /** Ordena la salida de datos por pantalla mediante el campo especificado
    public void ordenarAlmacen() { ...88 lines }

    /*METODOS SETTER AND GETTER*/

    ...
}
```

- Dentro del paquete “moduloClientes” hay tres clases:
  - “Cliente” contiene todos los atributos que tendrá un cliente, además de sus métodos setter y getter.

```
public class Cliente {

    private String mail;
    private String pass;
    private String nombre;
    private String apellido1, apellido2;
    private String direccion;
    private String poblacion;
    private String provincia;
    private String cp;
    private String telefono;

    /*METODOS GETTER AND SETTER*/

    /**...4 lines */
    public String getMail() { ...3 lines }

    /**...4 lines */
}
```

- “ClienteAleatorio” contiene una serie de vectores con parámetros aleatorios y un método para generar un campo aleatorio partiendo de uno de esos vectores.

```
public class ClienteAleatorio {

    /** Vector de nombres aleatorios ...3 lines */
    private final String[] nombreA = {"Francisco", "Roberto", "Mar:
                                     "Alfonso", "Pedro", "Nacho", "Naiara",
                                     "Laura", "Sara", "Diego", "Juan", "Anto

    /** Vector de apellidos aleatorios ...3 lines */
    private final String[] apellidosA = {"Blazquez", "Martin", "Sar
                                     "Dominguez", "Barroso", "Tomas", "l
                                     "Garcia", "Fernandez", "Muñoz", "Pe

    /** Vector de emails aleatorios ...3 lines */
    private final String[] mailA = {"gmail.com", "hotmail.com", "ya

    /** Abecedario ...3 lines */
    private final String abecedario = "abcdefghijklmnopqrstuvwxyz";

    /** Vector de provincias aleatorias ...3 lines */
    private final String[] provinciaA = {"Salamanca", "Avila", "Le
                                     "Madrid", "Barcelona", "Cadiz", "O
                                     "Pontevedra", "Bilbao"};

    /** Vector de direcciones aleatorias ...3 lines */
    private final String[] direccionA = {"Alfareros", "Don Quijote'
                                     "Toro", "Valle", "Pintores", "Guer
                                     "Gran Capitan", "Almirante", "Orqu:

    /** Retorna el una posicion aleatoria del vector que se le pas
    public String campoAleatorio(String[] campo){...6 lines }
```

- “ModuloCliente” contiene los métodos correspondientes a las opciones del menú de los clientes además del ArrayList que contiene todos los clientes registrados.

```

private static ArrayList<Cliente> listaClientes = new ArrayList<>();

/** Opcion "Modulo clientes" en del menu principal, contiene todo lo re
public void menuCliente() {...76 lines }

/** Opcion que genera clientes aleatorios ...3 lines */
public void clientesAleatorios() {...94 lines }

/** Muestra por pantalla un listado de los clientes ...3 lines */
public void listarClientes() {...31 lines }

/** Opcion "Copias de seguridad" del menu de clientes, contiene lo rela
public void menuCopiaSeguridad() {...66 lines }

/** Opcion "Generar copia de seguridad" del menu de "copia de seguridad
public void generarCopiaClientes() {...47 lines }

/** Opcion "Generar copia de seguridad" del menu de "copia de seguridad
public void importarCopiaClientes() {...53 lines }

/** Opcion "Modificar cliente" del menu de clientes, contiene lo relati
public void menuModificarClientes() {...73 lines }

/** Opcion "Modificar cliente" del menu de modificar clientes, modifica
public void modificarCliente() {...83 lines }

/** Opcion "Registrar cliente" del menu de modificar clientes, añade un
public void registrarCliente() {...39 lines }

/** Opcion "Borrar cliente" del menu de modificar clientes, borra el cl
public void borrarCliente() {...18 lines }

/** Muestra los clientes con sus correos electronicos ...3 lines */
public void listarMailClientes() {...24 lines }

```

- Dentro del paquete “moduloPedidos” hay tres clases:
  - “Pedido” contiene todos los atributos que tendrá un pedido, cabe destacar que hay un ArrayList de productos dentro de esta clase y que tiene un constructor al que hay que pasarle el campo clave del cliente y genera la fecha actual del pedido, así como un código de pedido aleatorio. Además contiene sus métodos setter y getter.

```

public class Pedido {

    private String codPed = "";
    private String fecha;
    private String mailCli;
    private ArrayList<Producto> productoPed

    /** Constructor de los pedidos, genera
    public Pedido(String mailCli) {...22 lin

    /*METODOS SETTER AND GETTER*/

    /**...4 lines */
    public ArrayList<Producto> getProductoP

```

- “Factura” contiene todos los atributos que tendrá una factura, cabe destacar que tiene un constructor al que hay que pasarle el campo clave del pedido y el del cliente y genera un código de pedido aleatorio. Además contiene sus métodos setter y getter.

```
public class Factura {

    private String codFact = "";
    private String mailCli;
    private String codPed;
    private String totalBase;
    private String totalIva;
    private String total;

    /** Constructor de las facturas ...5 lines */
    public Factura(String codPed, String mailCli)

    /*METODOS SETTER AND GETTER*/

    /**...4 lines */
```

- “ModuloCliente” contiene los métodos correspondientes a las opciones del menú de los clientes además del ArrayList que contiene todos los clientes registrados

```
public class ModuloPedido {

    private ArrayList<Pedido> listaPedidos = new ArrayList<>();
    private final String delimitador = ";";

    public void menuPedidos() {...75 lines }

    /** Realiza los pedidos modificando los articulos del alma
    public void realizarPedido() {...111 lines }

    /** Anula un pedido realizado ...3 lines */
    public void anularPedido() {...51 lines }

    /** Genera el fichero de pedidos del dia ...3 lines */
    public void generarFicheroPedidos() {...58 lines }

    /** Genera las facturas correspondientes a los pedidos del
    public void generarFicheroFacturas() {...251 lines }

    /** Muestra la lista de los pedidos ...3 lines */
    public void mostrarListaPedidos() {...52 lines }
```

## Ejecución del programa:

- Se le pide al usuario el nombre de la carpeta del escritorio donde quiere guardar los datos de la aplicación.
- Se despliega un menú de navegación totalmente libre para el usuario.

Para más información leer los **comentarios del código del programa** o el **pdf de la documentación en Javadoc**.