

**Instituto FOC**  
**Módulo de desarrollo de aplicaciones web**  
Programación

**BRUNO MARENCO CERQUEIRA**

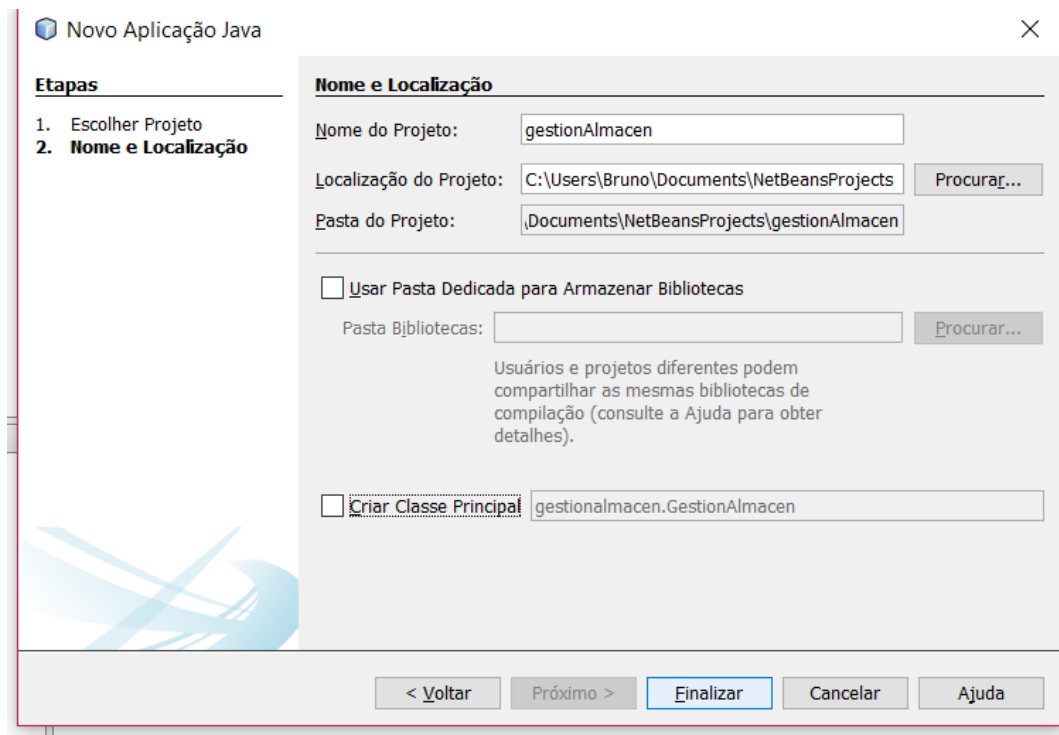
**Tarea Individual 5: Colecciones**

Marzo/2018

## Índice

Índice .....	2
Crear un proyecto en NetBeans denominado, "gestionAlmacen" .....	3
<b>Dentro de dicho proyecto, crear una clase denominada "Producto" con las siguientes características: .....</b>	<b>3</b>
<b>Dentro de dicho proyecto, crear una clase denominada "gestionAlmacen", que contenga un único método main y donde se lleven a cabo las siguientes acciones: .....</b>	<b>5</b>
Crear 4 objetos de tipo producto, con los siguientes valores en sus atributos: .....	5
Crear una colección de tipo ArrayList denominada "almacen" que permita gestionar los distintos objetos productos.....	5
Añadir los 4 objetos producto a la colección "almacen" creada con anterioridad. ....	6
Consultar el tamaño de la colección después de añadir los objetos. ....	6
Listar todos los objetos almacenados en el ArrayList "almacen" y mostrar sus atributos por pantalla. ....	6
Determinar si en el ArrayList "almacen" hay objetos cuyo nombre es "Tuerca" .....	7
Borrar el objeto cuyo nombre es "Alambre" del ArrayList "almacen". .....	7
<b>Código. ....</b>	<b>8</b>

## Crear un proyecto en NetBeans denominado, "gestionAlmacen".



Dentro de dicho proyecto, crear una clase denominada "Producto" con las siguientes características:

- Atributos (Todos los atributos de la clase Producto deben tener visibilidad privada):
  - codProducto: número entero que representa el identificador único del tipo de producto.
  - nombre: cadena de caracteres que representa el nombre del producto.
  - cantidad: número entero que representa la cantidad en stock de dicho producto.
  - descripcion: cadena de texto que representa una descripción breve del producto.
- Métodos (Todos los métodos de la clase Producto deben tener visibilidad pública):
  - constructor sin parámetros: constructor que inicializa todos los atributos de tipo cadenas de caracteres al valor null y los números enteros a 0.
  - constructor con parámetros: constructor que tienen tantos parámetros como atributos tiene la clase, y que inicializa cada uno de los atributos con el valor de los parámetros correspondientes.

```
Producto.java x
Código-Fuente  Histórico
1  /*
2  * To change this license header, choose License Headers in Project Properties.
3  * To change this template file, choose Tools | Templates
4  * and open the template in the editor.
5  */
6
7  /**
8   *
9   * @author Bruno
10  */
11  public class Producto {
12
13      /*
14       * Declaramos los atributos privados codProducto, nombre, cantidad y descripcion.
15       */
16      private int codProducto;
17      private String nombre;
18      private int cantidad;
19      private String descripcion;
20
21      /*
22       * Creamos los dos constructores, uno sin parámetros con los String inicializados en null y los integer en 0.
23       * Y otro constructor con parámetros , inicializando cada atributo por el valor de los parámetros.
24       */
25      public Producto () {
26          codProducto = 0;
27          nombre = null;
28          cantidad = 0;
29          descripcion = null;
30      }
31
32      public Producto (int cod, String nom, int cant, String des) {
33          codProducto = cod;
34          nombre = nom;
35          cantidad = cant;
36          descripcion = des;
37      }
38  }
```

- get y set: Métodos get y set para poder consultar y modificar cada uno de los atributos desde fuera de la clase, al tener visibilidad privada.

```
/* Definimos los métodos get y set de cada atributo. */
/*
public int getCodigo () {
    return codProducto;
}

public void setCodigo (int cod) {
    codProducto = cod;
}

public String getNombre () {
    return nombre;
}

public void setNombre (String nom) {
    nombre = nom;
}

public int getCantidad () {
    return cantidad;
}

public void setCantidad (int can) {
    cantidad = can;
}

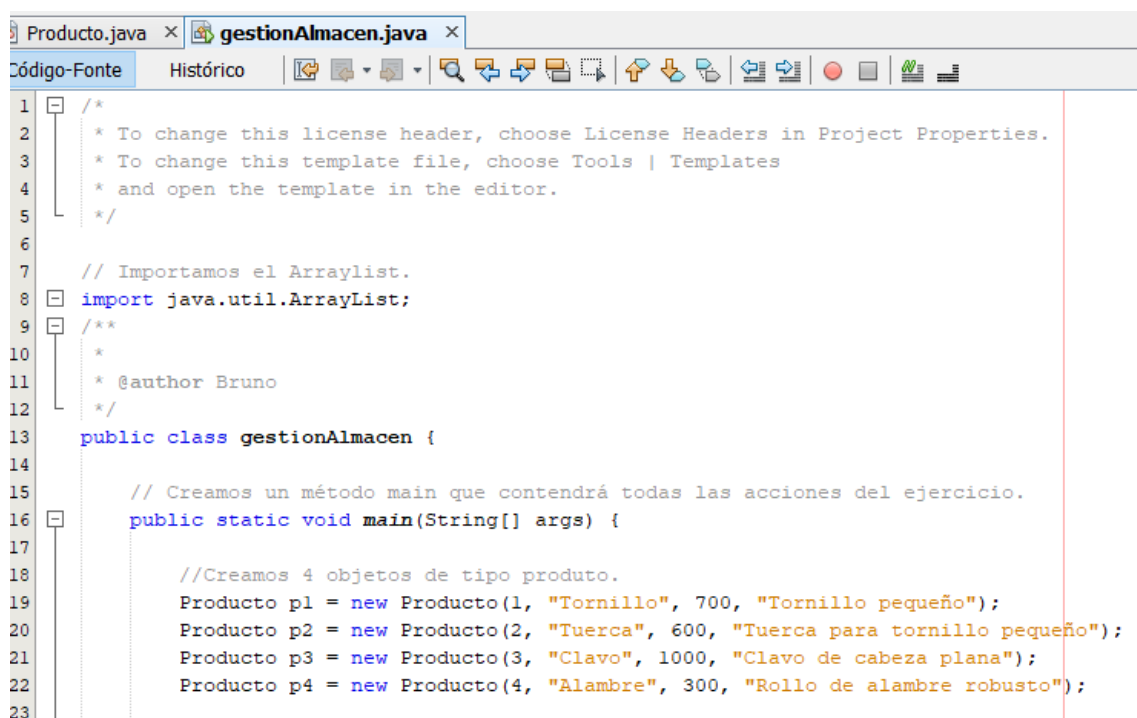
public String getDescripcion () {
    return descripcion;
}

public void setDescripcion (String des) {
    descripcion = des;
}
*/
```

Dentro de dicho proyecto, crear una clase denominada "gestionAlmacen", que contenga un único método main y donde se lleven a cabo las siguientes acciones:

Crear 4 objetos de tipo producto, con los siguientes valores en sus atributos:

codProducto	Nombre	Cantidad	Descripción
1	Tornillo	700	Tornillo pequeño
2	Tuerca	600	Tuerca para tornillo pequeño
3	Clavo	1000	Clavo de cabeza plana
4	Alambre	300	Rollo de alambre robusto



```
1  /*
2   * To change this license header, choose License Headers in Project Properties.
3   * To change this template file, choose Tools | Templates
4   * and open the template in the editor.
5   */
6
7   // Importamos el ArrayList.
8   import java.util.ArrayList;
9   /**
10    *
11    * @author Bruno
12    */
13   public class gestionAlmacen {
14
15       // Creamos un método main que contendrá todas las acciones del ejercicio.
16       public static void main(String[] args) {
17
18           //Creamos 4 objetos de tipo producto.
19           Producto p1 = new Producto(1, "Tornillo", 700, "Tornillo pequeño");
20           Producto p2 = new Producto(2, "Tuerca", 600, "Tuerca para tornillo pequeño");
21           Producto p3 = new Producto(3, "Clavo", 1000, "Clavo de cabeza plana");
22           Producto p4 = new Producto(4, "Alambre", 300, "Rollo de alambre robusto");
23       }
```

Crear una colección de tipo ArrayList denominada "almacen" que permita gestionar los distintos objetos productos.

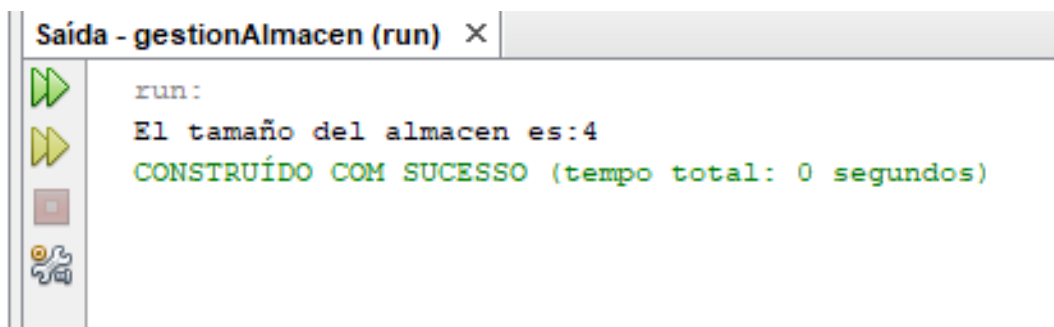
```
//Creamos una colección ArrayList almacén.
ArrayList<Producto> almacen = new ArrayList<Producto> ();
```

Añadir los 4 objetos producto a la colección "almacen" creada con anterioridad.

```
//Añadimos los productos a la colección almacén.  
almacen.add(p1);  
almacen.add(p2);  
almacen.add(p3);  
almacen.add(p4);
```

Consultar el tamaño de la colección después de añadir los objetos.

```
// Consultamos el tamaño del almacén.  
System.out.println("El tamaño del almacen es:" + almacen.size());
```



Listar todos los objetos almacenados en el ArrayList "almacen" y mostrar sus atributos por pantalla.

```
// Crear bucle para listar productos y mostrar atributos por pantalla.  
  
for (Producto pp : almacen ) {  
    System.out.println("El producto " + pp.getCodigo() + " es " + pp.getNombre() +  
        "; Cantidad: " + pp.getCantidad() + "; Descripción: " + pp.getDescripcion() + ".");  
}  
  
-----  
El producto 1 es Tornillo; Cantidad: 700; Descripción: Tornillo pequeño.  
El producto 2 es Tuerca; Cantidad: 600; Descripción: Tuerca para tornillo pequeño.  
El producto 3 es Clavo; Cantidad: 1000; Descripción: Clavo de cabeza plana.  
El producto 4 es Alambre; Cantidad: 300; Descripción: Rollo de alambre robusto.
```

Determinar si en el ArrayList "almacen" hay objetos cuyo nombre es "Tuerca".

```
// Verificar si existe objeto con nombre Tuerca.
for (Producto pp : almacen ) {
    if(pp.getNombre() == "Tuerca") {
        System.out.println("Existe el producto tuerca");
    }
}
```

**Existe el producto tuerca**

Borrar el objeto cuyo nombre es "Alambre" del ArrayList "almacen".

```
// Borrar el objeto cuyo nombre es alambre de almacén.
for (Producto pp : almacen ) {
    if(pp.getNombre() == "Alambre") {
        productoBorrar = pp;
        System.out.println("Se elimina el producto Alambre.");
    }
}
almacen.remove(productoBorrar);

for (Producto pp : almacen ) {
    System.out.println("El producto " + pp.getCodigo() + " es " + pp.getNombre() +
        "; Cantidad: " + pp.getCantidad() + "; Descripción: " + pp.getDescripcion() + ".");
}
```

Eliminamos el producto y comprobamos que ya no está.

```
Se elimina el producto Alambre.
El producto 1 es Tornillo; Cantidad: 700; Descripción: Tornillo pequeño.
El producto 2 es Tuerca; Cantidad: 600; Descripción: Tuerca para tornillo pequeño.
El producto 3 es Clavo; Cantidad: 1000; Descripción: Clavo de cabeza plana.
CONSTRUIDO COM SUCESSO (tempo total: 0 segundos)
```

## Código.

Primer archivo. Producto.java

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */

/**
 *
 * @author Bruno
 */
public class Producto {

    /*
     * Declaramos los atributos privados codProducto, nombre, cantidad y descripcion.
     */
    private int codProducto;
    private String nombre;
    private int cantidad;
    private String descripcion;

    /*
     * Creamos los dos constructores, uno sin parámetros con los String inicializados en null y los integer en 0.
     * Y otro constructor con parámetros , inicializando cada atributo por el valor de los parámetros.
     */
    public Producto () {
        codProducto = 0;
        nombre = null;
```



```
    cantidad = 0;
    descripcion = null;
}
```

```
public Producto (int cod, String nom, int cant, String des) {
    codProducto = cod;
    nombre = nom;
    cantidad = cant;
    descripcion = des;
}
```

```
/*
 * Definimos los métodos get y set de cada atributo.
 */
public int getCodigo () {
    return codProducto;
}
```

```
public void setCodigo (int cod) {
    codProducto = cod;
}
```

```
public String getNombre () {
    return nombre;
}
```

```
public void setNombre (String nom) {
    nombre = nom;
}
```

```

public int getCantidad () {
    return cantidad;
}

public void setCantidad (int can) {
    cantidad = can;
}

public String getDescripcion () {
    return descripcion;
}

public void setDescripcion (String des) {
    descripcion = des;
}

}

```

Segundo archivo. gestionAlmacen.java

```

/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */

// Importamos el ArrayList.
import java.util.ArrayList;

/**
 *
 * @author Bruno

```

```

*/
public class gestionAlmacen {

    // Creamos un método main que contendrá todas las acciones del ejercicio.
    public static void main(String[] args) {

        //Creamos 4 objetos de tipo producto.
        Producto p1 = new Producto(1, "Tornillo", 700, "Tornillo pequeño");
        Producto p2 = new Producto(2, "Tuerca", 600, "Tuerca para tornillo pequeño");
        Producto p3 = new Producto(3, "Clavo", 1000, "Clavo de cabeza plana");
        Producto p4 = new Producto(4, "Alambre", 300, "Rollo de alambre robusto");
        Producto productoBorrar = new Producto();

        //Creamos una colección ArrayList almacén.
        ArrayList<Producto> almacen = new ArrayList<Producto> ();

        //Añadimos los productos a la colección almacén.
        almacen.add(p1);
        almacen.add(p2);
        almacen.add(p3);
        almacen.add(p4);

        // Consultamos el tamaño del almacén.
        System.out.println("El tamaño del almacen es: " + almacen.size());

        // Crear bucle para listar productos y mostrar atributos por pantalla.

        for (Producto pp : almacen ) {
            System.out.println("El producto " + pp.getCodigo() + " es " + pp.getNombre() +
                "; Cantidad: " + pp.getCantidad() + "; Descripción: " + pp.getDescripcion() + ".");
        }
    }
}

```

```
// Verificar si existe objeto con nombre Tuerca.
```

```
for (Producto pp : almacen ) {
```

```
    if(pp.getNombre() == "Tuerca") {
```

```
        System.out.println("Existe el producto tuerca");
```

```
    }
```

```
}
```

```
// Borrar el objeto cuyo nombre es alambre de almacén.
```

```
for (Producto pp : almacen ) {
```

```
    if(pp.getNombre() == "Alambre") {
```

```
        productoBorrar = pp;
```

```
        System.out.println("Se elimina el producto Alambre.");
```

```
    }
```

```
}
```

```
almacen.remove(productoBorrar);
```

```
for (Producto pp : almacen ) {
```

```
    System.out.println("El producto " + pp.getCodigo() + " es " + pp.getNombre() +
```

```
    "; Cantidad: " + pp.getCantidad() + "; Descripción: " + pp.getDescripcion() + ".");
```

```
}
```

```
}
```

```
}
```