

Documentación con Javadoc

Ejercicio 1:

Para empezar, escribo los comentarios en las clases, añadiendo la versión y el autor. También uso las etiquetas HTML para escribir los comentarios.

```
/**
 * <h2>Clase FicheroAleatorioVentana</h2>
 * Clase principal para el programa.
 * @see VentanaDepart
 * @author David
 * @version v1.0
 */
public class FicheroAleatorioVentana {

    /**
     * <h3>Función main de la clase FicheroAleatorioVentana</h3>
     * @param args
     * @throws IOException
     */
    public static void main(String[] args) throws IOException {

        JFrame f = new JFrame("DEPARTAMENTOS.");
        // Nos aseguramos que el fichero exista para evitar posibles errores
        File fichero = new File("AleatorioDep.dat");
        RandomAccessFile file;
        file = new RandomAccessFile(fichero, "rw");
        file.close();

        VentanaDepart v = new VentanaDepart(f);
        v.setVisible(true);

    } //fin main
} //fin class
```

```
/**
 * <h3>Constructor</h3>
 * Para crear la ventana.
 * @param f
 */
public VentanaDepart(JFrame f )
{
    setTitle("GESTIÓN DE DEPARTAMENTOS.");

    JPanel p0 = new JPanel();
    c = Color.CYAN;
    p0.add(titulo);
    p0.setBackground(c);

    JPanel p1 = new JPanel();
    p1.setLayout (new FlowLayout());
    p1.add(lnum);
    p1.add(num); p1.add(consu);

    JPanel p2 = new JPanel();
```

Una vez escritos todos los comentarios, revisando que tengo una sintaxis correcta, generamos el javadoc. Para generarlo, iremos a la barra de menús superior, seleccionaremos Proyecto >

Generar Javadoc. La documentación se generará automáticamente en una carpeta llamada "doc". Para poder verla, solo tenemos que abrir index.html en nuestro navegador. Desde ahí podremos navegar por los HTML de una forma cómoda.

Class FicheroAleatorioVentana

java.lang.Object
FicheroAleatorioVentana

```
public class FicheroAleatorioVentana  
extends java.lang.Object
```

Clase FicheroAleatorioVentana

Clase principal para el programa.

Version:

V1.0

Author:

David

See Also:

[VentanaDepart](#)

Clase para el objeto VentanaDepart

Contiene la construcción de la ventana y las funcionalidades de los botones.

Version:

V1.0

Author:

David

See Also:

[Serialized Form](#)

Constructor Details

VentanaDepart

```
public VentanaDepart(javax.swing.JFrame f)
```

Constructor

Para crear la ventana.

Parameters:

f -

actionPerformed

```
public void actionPerformed(java.awt.event.ActionEvent e)
```

Función actionPerformed

Esta función va a detectar si se ha pulsado algún botón.

Specified by:

actionPerformed in interface `java.awt.event.ActionListener`

Parameters:

e -

verporconsola

```
public void verporconsola()  
    throws java.io.IOException
```

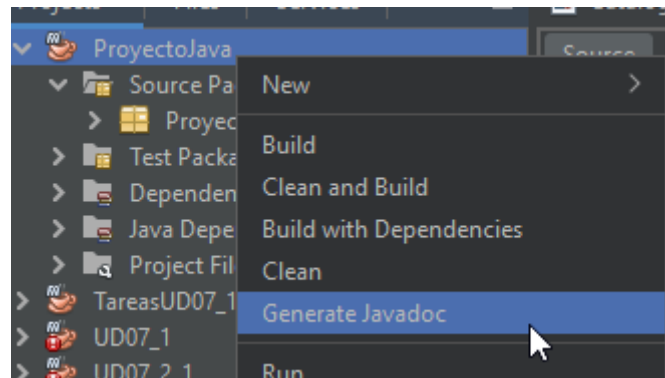
Función verporconsola

Esta función muestra por la consola del IDE los resultados de lo que hagamos.

Throws:

Realizamos las mismas operaciones en NetBeans. Para ello, importamos el mismo proyecto. Podemos añadir los mismos comentarios, o abrir el proyecto ya modificado. La diferencia es

que para generar el Javadoc, haremos clic derecho en el proyecto y seleccionaremos “Generar Javadoc”.



El resultado es exactamente el mismo, puesto que la herramienta usada, Javadoc, es la misma en los dos casos.

JavaScript is disabled on your browser.

PACKAGE CLASS USE TREE DEPRECATED INDEX HELP

PREV CLASS NEXT CLASS FRAMES NO FRAMES ALL CLASSES

SUMMARY: NESTED | FIELD | CONSTR | METHOD DETAIL: FIELD | CONSTR | METHOD

Class FicheroAleatorioVentana

java.lang.Object
FicheroAleatorioVentana

```
public class FicheroAleatorioVentana
extends java.lang.Object
```

Clase FicheroAleatorioVentana

Clase principal para el programa.

See Also:

[VentanaDepart](#)

Constructor Summary