

31 - Swing - JFrame

[Listado completo de tutoriales](#)

La componente básica que requerimos cada vez que implementamos una interfaz visual con la librería Swing es la clase JFrame. Esta clase encapsula una Ventana clásica de cualquier sistema operativo con entorno gráfico (Windows, OS X, Linux etc.)

Hemos dicho que esta clase se encuentra en el paquete javax.swing y como generalmente utilizamos varias clases de este paquete luego para importarla utilizamos la sintaxis:

```
import javax.swing.*;
```

Podemos hacer una aplicación mínima con la clase JFrame:

```
import javax.swing.JFrame;
public class Formulario {
    public static void main(String[] ar) {
        JFrame f=new JFrame();
        f.setBounds(10,10,300,200);
        f.setVisible(true);
        f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }
}
```

Como vemos importamos la clase JFrame del paquete javax.swing:

```
import javax.swing.JFrame;
```

y luego en la main definimos y creamos un objeto de la clase JFrame (llamando luego a los métodos setBounds, setVisible y setDefaultCloseOperation):

```
public static void main(String[] ar) {
    JFrame f=new JFrame();
    f.setBounds(10,10,300,200);
    f.setVisible(true);
    f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
}
```

Pero esta forma de trabajar con la clase JFrame es de poca utilidad ya que rara vez necesitemos implementar una aplicación que muestre una ventana vacía.

Lo más correcto es plantear una clase que herede de la clase JFrame y extienda sus responsabilidades agregando botones, etiquetas, editores de línea etc.

Entonces la estructura básica que emplearemos para crear una interfaz visual será:

Programa:

```
import javax.swing.*;
public class Formulario extends JFrame{
    public Formulario() {
        setLayout(null);
    }

    public static void main(String[] ar) {
        Formulario formulario1=new Formulario();
        formulario1.setBounds(10,20,400,300);
        formulario1.setVisible(true);
        formulario1.setDefaultCloseOperation(JFrame.E
    }
}
```

Importamos el paquete donde se encuentra la clase JFrame:

```
import javax.swing.*;
```

Planteamos una clase que herede de la clase JFrame:

```
public class Formulario extends JFrame{
```

En el constructor indicamos que ubicaremos los controles visuales con coordenadas absolutas mediante la desactivación del Layout heredado (más adelante veremos otras formas de ubicar los controles visuales dentro del JFrame):

```
    public Formulario() {
        setLayout(null);
    }
```

En la main creamos un objeto de la clase y llamamos a los métodos setBounds, setVisible y setDefaultCloseOperation:

```
    public static void main(String[] ar) {
        Formulario formulario1=new Formulario();
        formulario1.setBounds(10,20,400,300);
        formulario1.setVisible(true);
```

```
formulario1.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
}
```

El método `setBounds` ubica el `JFrame` (la ventana) en la columna 10, fila 20 con un ancho de 400 píxeles y un alto de 300.

Debemos llamar al método `setVisible` y pasarle el valor `true` para que se haga visible la ventana.

Mediante el método `setDefaultCloseOperation` y pasando la constante `JFrame.EXIT_ON_CLOSE` se le indica al `JFrame` que cuando se presione el botón de cerrado de la ventana proceda a finalizar por completo la aplicación.

Problemas propuestos

1. Crear una ventana de 1024 píxeles por 800 píxeles. Luego no permitir que el operador modifique el tamaño de la ventana. Sabiendo que hacemos visible al `JFrame` llamando la método `setVisible` pasando el valor `true`, existe otro método llamado `setResizable` que también requiere como parámetro un valor `true` o `false`.

[Ver video](#)

```
import javax.swing.*;  
public class Formulario extends JFrame{  
    Formulario() {  
        setLayout(null);  
    }  
  
    public static void main(String[] ar) {  
        Formulario formulario1=new Formulario();  
        formulario1.setBounds(0,0,1024,800);  
        formulario1.setResizable(false);  
        formulario1.setVisible(true);  
        formulario1.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
    }  
}
```

[Retornar](#)