

9. PROGRAMACIÓN GRÁFICA

9.1 SWING

Swing es un conjunto robusto de herramientas creadas por **Sun Microsystems** para la creación de interfaces gráficas en java. Básicamente swing es un grupo de clases que proporciona alternativas poderosas y flexibles a los componentes estándar de los **AWT** (Kits de Herramientas de Ventana Abstracta); por ejemplo, además de los componentes que siempre utiliza como las etiquetas, botones, casillas de verificación; **Swing** proporciona otros componentes importantes como paneles desplazables, paneles de fichas, árboles y tablas los cuales permiten a los programadores un mejor trabajo en el desarrollo de una aplicación [11].

9.2 CREACIÓN DE UN FRAME

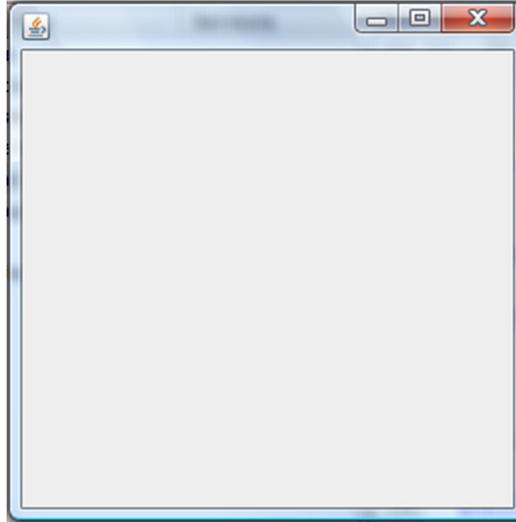
Un frame es un componente Swing capaz de contener a otros componentes (botones, etiquetas, cajas de texto, etc). Para crear un Frame es necesario hacer uso de la clase JFrame contenida en el paquete *javax.swing* mediante la instrucción extends. Una vez esté definido el uso del JFrame es necesario establecer un tamaño del Frame a través del constructor. Finalmente se crea el objeto desde el método main() de la clase principal y se le establecen las acciones que este ejecutará; en este caso, visualizarse y terminar el programa cuando se presione el botón de cierre del Frame. A continuación se muestra el código y la salida gráfica que se produce al ejecutar el programa.

```
package framesimple;
import javax.swing.JFrame;

public class FrameVacio extends JFrame {
    public FrameVacio(){
        setSize(300,300);
    }
}

package framesimple;
import javax.swing.JFrame;
```

```
public class Main {  
    public static void main(String[] args) {  
        FrameVacio fv1 = new FrameVacio();  
        fv1.setVisible(true);  
        fv1.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
    }  
}
```



9.3 ASIGNAR TÍTULO A UN FRAME

El método `setTitle()` permite asignarle un título a la barra de títulos de un frame. Su sintaxis es:

```
nombreFrame.setTitle("Título del frame");
```

9.4 LOCALIZACIÓN DE UN FRAME

Al momento de crear un frame, este queda posicionado en las coordenadas (0,0) de la pantalla del computador. Para ubicar un frame ubicado en una posición diferente, se utiliza el método **`setLocation()`**, cuya sintaxis es la siguiente:

```
nombreComponente.setLocation(posición_x, posición_y);
```

Los valores para `posición_x` y `posición_y` deben ser enteros y corresponden al número de píxeles desde el borde izquierdo y número de píxeles desde el borde superior de la pantalla respectivamente en donde se ubicará la esquina superior izquierda del frame. Para hacer que el frame creado en el programa anterior se ubique en la posición (300,300) se debe escribir la siguiente instrucción una vez creado el objeto:

```
fv1.setLocation(300,300);
```

9.5 MOVER Y CAMBIAR EL TAMAÑO DE UN FRAME

A través del método **`setBounds()`** es posible cambiar la posición y tamaño de un componente Swing. Su sintaxis es:

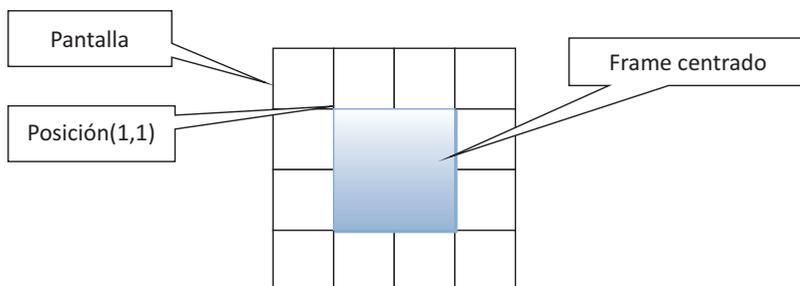
```
nombreComponente.setBounds(posición_x, posición_y, ancho, alto);
```

Los argumentos `posición_x`, `posición_y`, `ancho` y `alto` deben ser enteros y corresponden a la distancia en píxeles desde el borde izquierdo, borde derecho, ancho del componente y alto del componente respectivamente. Los valores `ancho` o `alto` serán modificados automáticamente si cualquiera de los dos está por debajo del mínimo especificado previamente mediante el método **`setMinimumSize()`**. Es importante tener en cuenta que si no especifican las coordenadas de posicionamiento y el tamaño del frame, este se posicionara en las coordenadas (0,0) y tendrá un tamaño de 0 x 0 píxeles. También se debe tener en cuenta que aplicaciones software de calidad deben ser capaces de adaptarse a la configuración de pantalla de los usuarios al momento de visualizar formularios con diferentes tipos de información; esto requiere de los programadores la agilidad para capturar la resolución de video de los usuarios y ajustar según estas la posición y tamaño de los formularios de captura y/o visualización de datos.

9.6 CENTRAR UN FRAME

Para centrar un frame en la pantalla de un usuario determinado, es necesario partir por obtener la configuración de la resolución en píxeles de la pantalla del usuario y posteriormente establecer un frame que ocupe una cuarta parte del total de la pantalla. A continuación se ilustra gráficamente el problema planteado; se parte de la suposición de tener

una pantalla de 4 x 4 pixeles en la que se centra un frame de 2 x 2 pixeles, la esquina superior izquierda del frame debe ser ubicada en la posición (1,1) de la pantalla.



Observando la figura anterior podemos observar que el ancho del frame que se pretende centrar en la pantalla corresponde a la mitad del ancho de la pantalla, el alto del frame corresponde a la mitad del alto de la pantalla y la posición de la esquina superior izquierda en la que se ubicará el frame para que quede centrado en la pantalla corresponde a la cuarta parte del ancho de la pantalla para la posición *x* y a la cuarta parte del alto de la pantalla para la posición *y*.

Para poder obtener la resolución en pixeles a la cual está configurada la pantalla del usuario es necesario recurrir al sistema operativo de su computador. En java es posible hacer esto gracias a la clase **Toolkit** la cual cuenta con el método **getScreenSize()** para retornar un objeto de tipo **Dimension** con el tamaño en pixeles (**ancho, alto**) de la pantalla del computador.

A continuación se adjunta el código en java de un programa que centra un frame en la pantalla de un computador:

```
import java.awt.Dimension;
import java.awt.Toolkit;
import javax.swing.JFrame;
public class FrameCentrado extends JFrame{
    int ancho, alto, posicion_x, posicion_y;
    Toolkit tk1 = Toolkit.getDefaultToolkit();
    Dimension pantallaActual = tk1.getScreenSize();
    void darTamanno(){
        ancho = pantallaActual.width/2;
```

```

        alto = pantallaActual.height/2;
        setSize(ancho,alto);
    }
    void centrarFrame(){
        posicion_x = pantallaActual.width/4;
        posicion_y = pantallaActual.height/4;
        setLocation(posicion_x,posicion_y);
    }
}
public class Main {
    public static void main(String[] args) {
        FrameCentrado fc1 = new FrameCentrado();
        fc1.darTamanno();
        fc1.centrarFrame();
        fc1.setVisible(true);
    }
}

```

9.7 DIFERENCIA ENTRE UN FRAME Y UNA VENTANA

```
package javentanas;
```

```

public class Main {
    public static void main(String[] args) {
        Marco marco1 = new Marco();
        Ventana ventana1 = new Ventana();
        marco1.setVisible(true);
        ventana1.setVisible(true);
    }
}

```

CREACIÓN DE LA CLASE Marco

```

package javentanas;

import javax.swing.JFrame;

/**
 *
 * @author USUARIO

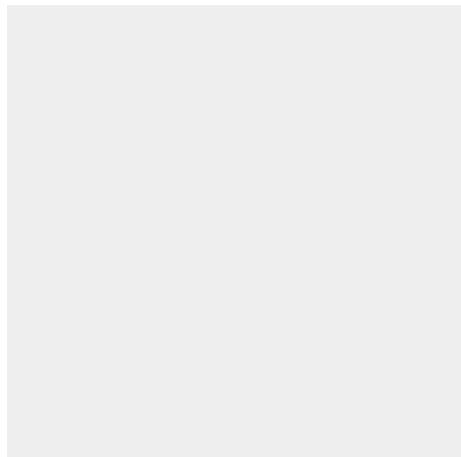
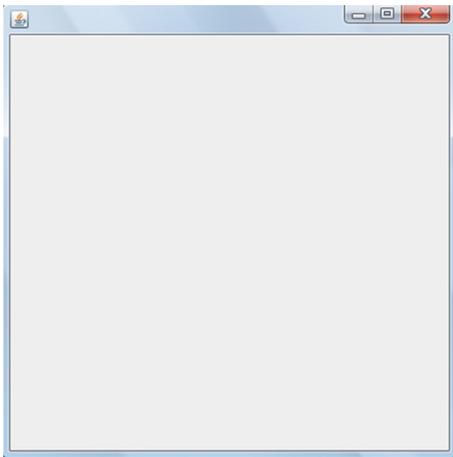
```

```
*/  
public class Marco extends JFrame {  
    Marco(){  
        setSize(400,400);  
        setLocation(100,100);  
    }  
}
```

CREACIÓN DE LA CLASE Ventana

```
package javentanas;  
  
import javax.swing.JWindow;  
  
/**  
 *  
 * @author USUARIO  
 */  
public class Ventana extends JWindow{  
    Ventana(){  
        setSize(400,400);  
        setLocation(600,100);  
    }  
}
```

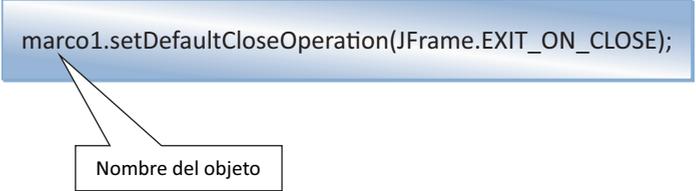
LA SALIDA SERÁ:



9.8 SALIDA DE LA APLICACIÓN DESDE UN FRAME

Un objeto de la clase JFrame posee entre otras propiedades: un ícono, una barra de título y los botones de control estándares para cualquier ventana (minimizar, maximizar y cerrar). Haciendo uso del botón cerrar de un objeto de la clase JFrame es posible finalizar con el funcionamiento de la aplicación que se esté ejecutando; para esto se hace utilizando la siguiente línea de código:

```
marco1.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
```



Nombre del objeto

Mediante esta línea de código, se establece la acción que seguirá el programa cuando el usuario de click sobre el botón cerrar del objeto JFrame; para este caso EXIT_ON_CLOSE, finaliza la ejecución de la aplicación en curso.

Es importante tener en cuenta que la acción por defecto para los JFrame es simplemente ocultar el JFrame sobre el que se efectuó la acción y no terminar la aplicación, sin embargo, en el programa del ejemplo, la instrucción puede considerarse como lógica, pues se está cerrando un único JFrame.