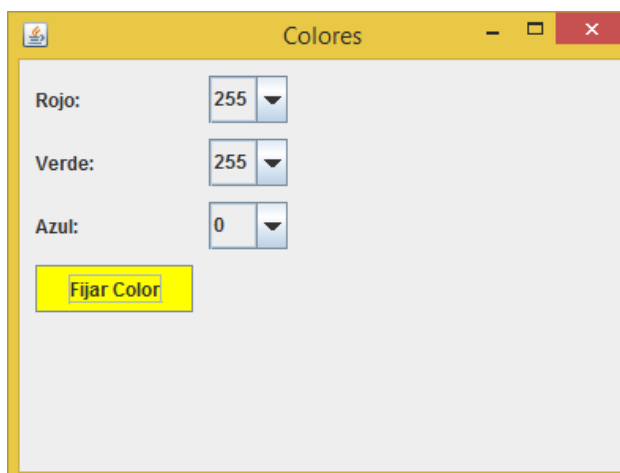


## EJERCICIOS 3

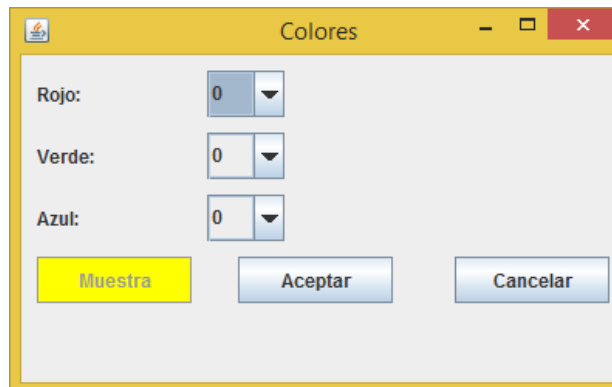
### Ejercicios avanzados

1. Crea un formulario llamado **BotonesColores** que tenga el siguiente aspecto:



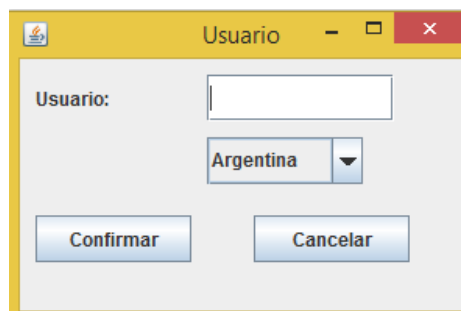
El formulario está formado por 3 listas con los números del 0 al 255 y un botón. Al pulsar el botón, el color de fondo del mismo se establecerá con el color indicado por las listas.

2. Modifica la aplicación anterior (llámala **BotonesColores2**) para que sea un **diálogo** con el siguiente aspecto (se trata de una funcionalidad que podemos lograr con un **JColorChooser**, pero vamos a simularla):
  - a. El botón de la izquierda estará deshabilitado (**setEnabled(false)**) y su color cambiará automáticamente cuando cambiemos los valores de las listas.
  - b. Si pulsamos Aceptar se guardará el color escogido en el diálogo, pero si pulsamos Cancelar el color que se guardará será el blanco.



Prueba el diálogo desde un frame que lo invoque y cambie su color de fondo con el color escogido en el diálogo.

3. Crea un diálogo como el siguiente que se llame **Usuario2**. El diálogo deberá pedir un nombre de usuario y un país de entre 5. Internamente, la información se guardará en un objeto de clase **Usuario**, que encapsulará los datos metidos.
4. En lugar de mostrar el diálogo en el constructor, crea el método **muestraDialogo()** que devuelva un objeto **Usuario** con los datos introducidos (y null si se pulsó *Cancelar*).



Crea un frame que llame al diálogo y muestre los datos introducidos en sendas etiquetas.

5. Crea una aplicación Swing llamada **Colores** que debería tener el siguiente aspecto:

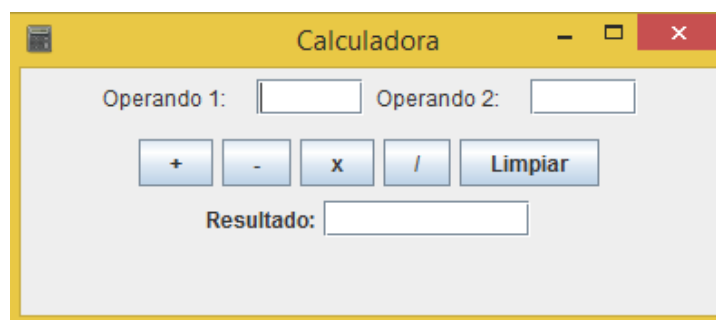


La ventana tiene 3 botones en la parte superior y un panel en la parte inferior.

Al pulsar cada uno de los botones, el panel de la parte inferior debería cambiar de color:

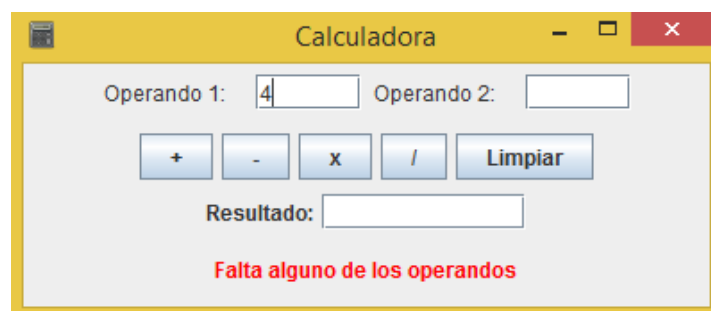


6. Implementa una aplicación Swing llamada **Calculadora** que tenga el siguiente aspecto:



La interfaz tiene cuatro filas:

- La primera fila tiene dos cajas de texto destinadas a almacenar los operandos de la operación.
- La segunda fila tiene 5 botones destinados a representar las 4 operaciones matemáticas fundamentales (+, -, x, /) y un botón que limpia las cajas de texto.
- La tercera fila tiene una caja de texto que almacena el resultado de la operación correspondiente.
- La cuarta fila tiene una caja de texto donde se mostrará errores asociados a la calculadora:
  - Alguno de los operandos está vacío
  - Se intenta dividir entre cero.



Utiliza un icono de la colección [Tango Icon](#) para usar como icono de la aplicación. Aunque el propio icono debería de ser de 16x16, ten en cuenta que el mismo icono se utiliza cuando tenemos la aplicación minimizada, luego usaremos el icono

de 32x32 (pues si usáramos el pequeño la calidad del mismo al aumentar sería muy baja):



Los errores se mostrarán al pulsar los botones correspondientes. Ten en cuenta que la siguiente vez que pulsamos el botón, si no hay error debería borrarse el mismo.

7. Modifica el ejercicio anterior para eliminar la etiqueta de error y mostrar los errores mediante un diálogo.
8. Implementa la una aplicación Swing llamada **Banderas** que tenga el siguiente aspecto:



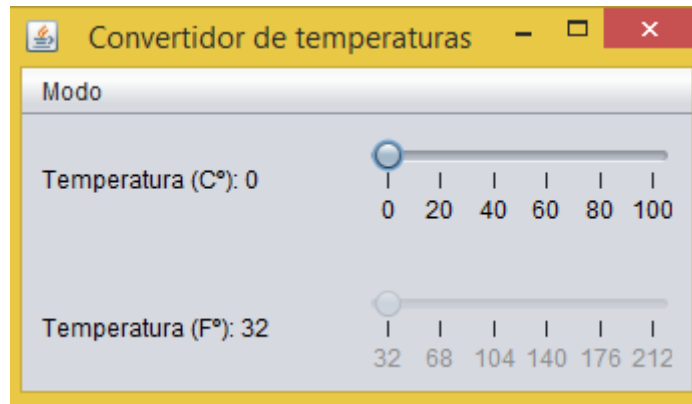
La aplicación tiene un grupo de botones de radio en la parte izquierda y una etiqueta en la parte derecha, que tiene como icono la imagen de una bandera. Al cambiar el botón de radio, cambiará la imagen de la bandera y la del título de la aplicación:



Utiliza un **ActionListener** para implementar las acciones asociadas a la pulsación de un **JRadioButton**.

Las banderas se proporcionan junto con este ejercicio. Intenta aprovechar el hecho de que los nombres de los países están en inglés y coinciden con el nombre de la imagen.

9. Implementa una interfaz Swing llamada **Temperaturas** que tenga el siguiente aspecto:



La aplicación está formada por dos etiquetas y dos barras de desplazamiento:

- Las barras de desplazamiento muestran una temperatura expresada tanto en grados Celsius (barra superior) como Farenheit (barra inferior). Los rangos de los sliders son 0-100 y 32-212 respectivamente. Se trata de los valores de congelación y ebullición del agua en las respectivas escalas.
- Cada una de las etiquetas mostrará el valor que contiene su barra de desplazamiento correspondiente.
- Por defecto, la barra superior está activada y la inferior desactivada (método **setEnabled(false)**)
- Al mover la barra superior se actualizarán tanto el valor de la barra inferior como ambas etiquetas. Esto lo podemos lograr implementando un **ChangeListener** asociado a la barra superior.

Ten en cuenta que las fórmulas para la conversión de Celsius a Farenheit y viceversa son las siguientes:

**Fahrenheit a Celsius:**

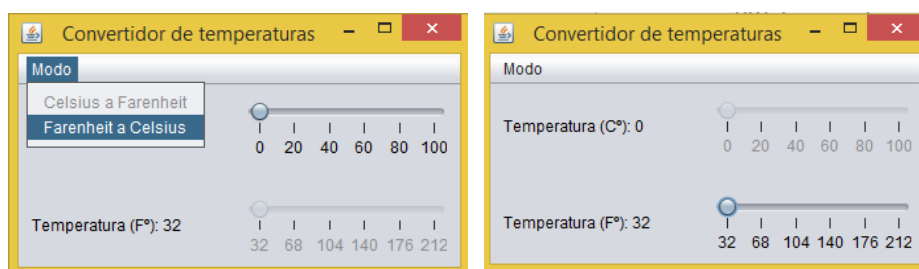
$$C = (F - 32) \cdot \frac{5}{9}$$

**Celsius a Fahrenheit:**

$$F = C \cdot \frac{9}{5} + 32$$

*NOTA: Las fracciones decimales tendremos que implementarlas como los números 0.55 y 1.8 respectivamente.*

- El menú de la parte superior permite escoger el modo en que vamos a convertir la temperatura:
  - Si se escoge la opción “Fahrenheit a Celsius”, se deshabilitará el slider correspondiente a Celsius y se habilitará el de Fahrenheit. También se habilitará la opción de menú “Celsius a Fahrenheit” y se deshabilitará “Fahrenheit a Celsius” (que acabamos de escoger).
  - Si se escoge la opción “Celsius a Fahrenheit” se realizarán las operaciones opuestas.



- Ten en cuenta que al deshabilitar una barra de desplazamiento también tendremos que desvincular el `ChangeListener` usado su método **`removeChangeListener()`** pues de lo contrario caeremos en un bucle (cada vez que cambiamos el valor del slider, aunque sea de manera programática, estamos activando el evento). Cuando activemos de nuevo el slider volveremos a escuchar los cambios de estado.

10. Modifica el ejercicio de la calculadora para aplicar el MVC. Para ello:

- a. Crea una clase **`Calculadora`** que implemente las operaciones.
- b. Crea una clase llamada **`CalculadoraListener`** que actúe como controlador.
- c. Crea las clases que creas necesario para implementar la interfaz.