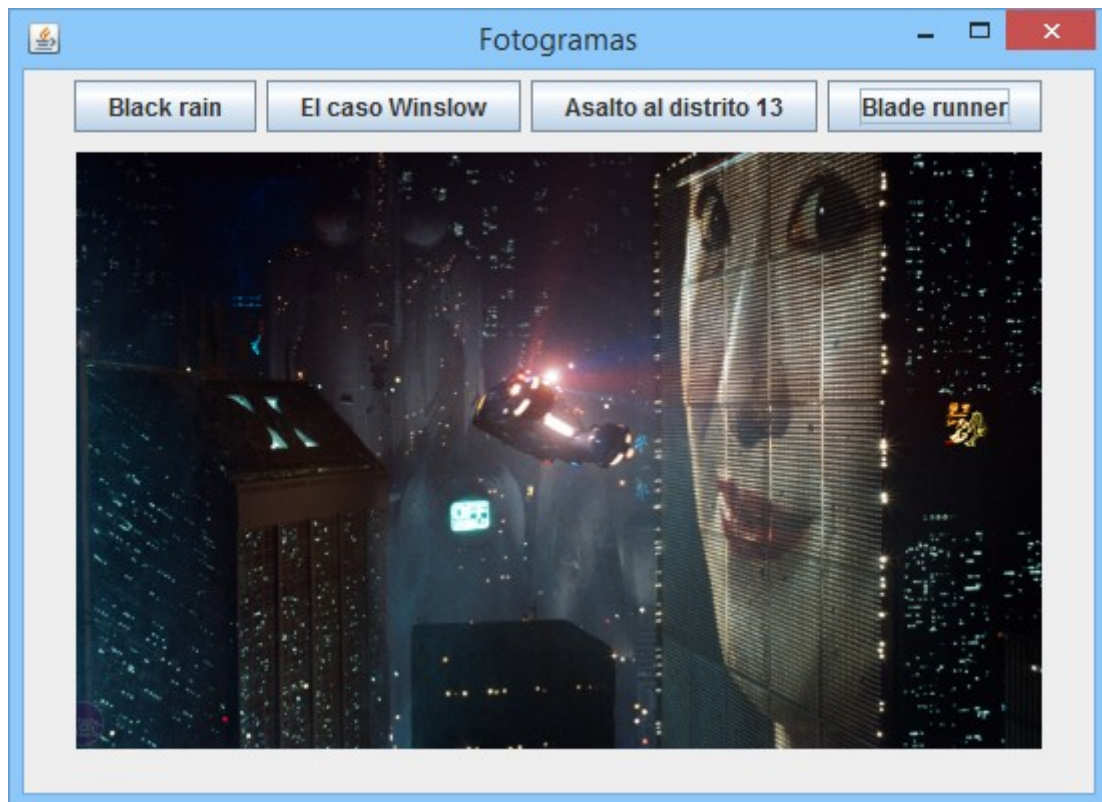


Actividades de Interfaces gráficas de usuario 4

Manejo de eventos

1.- Crea una aplicación que muestre en la parte superior cuatro botones con títulos de películas. Al pulsar uno de los botones, se mostrará una imagen de dicha película.



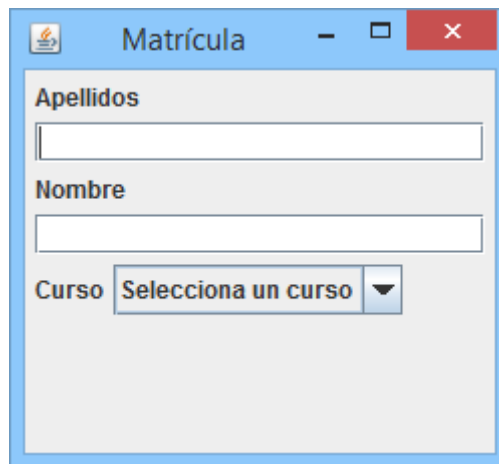
Ayuda:

- Prepara cuatro imágenes con el mismo tamaño, con un editor tipo GIMP. Así la ventana no cambiará de tamaño al cambiar de imagen.
- La imagen la puedes mostrar en la ventana usando una etiqueta, a la que le puedes asignar un *ImageIcon* con el método *setIcon*. Por ejemplo:

```
lblImagen.setIcon(new ImageIcon("iconos/pelis/blackrain.jpg"));
```

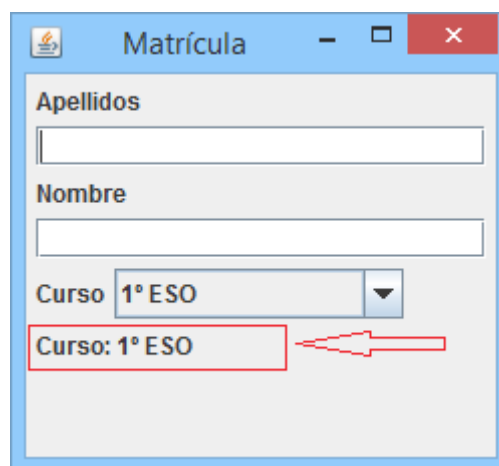
- Utiliza *BorderLayout* para maquetar la ventana: los botones irán en la parte superior y la imagen en la parte central.
- En la parte superior, crea un *JPanel*, asígnale un *FlowLayout* y añade dentro los botones. Así quedarán como en el ejemplo.
- En la parte central, crea un *JPanel*, asígnale un *FlowLayout* y un *JLabel* que muestre la primera imagen.

2.- Escribe un programa con la siguiente interfaz:



En el menú desplegable aparecerán las opciones: "1º ESO", "2º ESO", "3º ESO", "4º ESO".

Al seleccionar una opción aparecerá un rótulo debajo indicando el curso seleccionado:

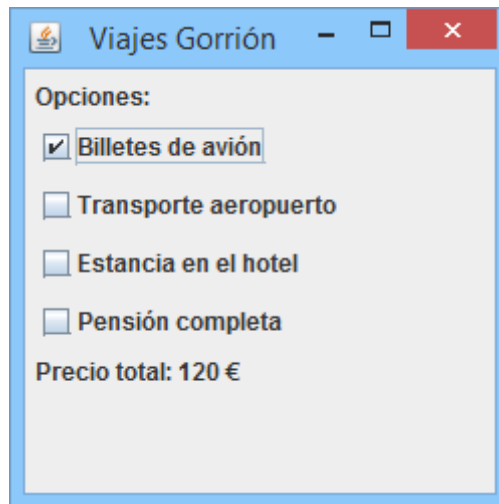


En cambio, si no se selecciona ninguno, no se mostrará nada.

Ayuda:

- Puedes hacer este ejercicio usando FlowLayout, con alineación a la izquierda. No es necesario usar JPanels.

3.- Crea una aplicación para una agencia de viajes, que ofrezca las siguientes opciones para contratar. Cada vez que se seleccione o deseleccione una opción deberá actualizarse el precio total del paquete contratado.



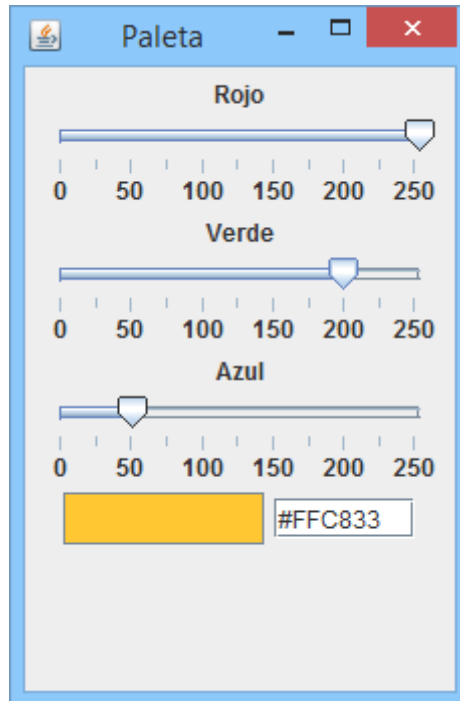
Los precios concretos de cada opción son:

- Billetes de avión: 120 €
- Transporte aeropuerto: 40 €
- Estancia en hotel: 180 €
- Pensión completa: 70 €

Ayuda:

- Puedes hacer este ejercicio usando `FlowLayout`, con alineación a la izquierda. No es necesario usar `JPanels`.
- Cuando se selecciona o deselecciona un checkbox se produce un evento de tipo `ItemEvent`, igual que al seleccionar una opción en un `ComboBox`. Por tanto la clase tiene que implementar `ItemListener` y el método que se ejecutará al clicar una opción es `itemStateChanged()`.

4.- Construye una aplicación que nos permita seleccionar la cantidad de cada color primario mediante controles deslizantes y nos coloree un botón con el color combinado. Además nos mostrará el color resultante en hexadecimal.



Ayuda:

- Puedes hacer este ejercicio usando `FlowLayout`. No es necesario usar `JPanels`.
- El recuadro coloreado es un botón al que se le pone un color de fondo usando el método `setBackground()`. Aeste método hay que pasarle un objeto `Color`, que podemos crear pasándole las componentes de cada color (rojo, verde y azul):

```
btnColor.setBackground(new Color(rojo, verde, azul));
```

- Cuando se mueve un `Slider` se produce un evento de tipo `ChangeEvent`. La clase tiene que implementar `ChangeListener` y el método que se ejecutará al clicar una opción es `stateChanged()`.
- Para la casilla que muestra el color resultante en formato hexadecimal, crea un método

```
private String hexadecimal(int rojo, int verde, int azul)
```

que reciba las tres componentes de color y devuelva una cadena con el formato `#FFC833`.