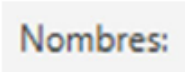
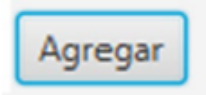

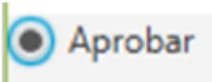
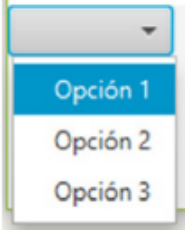
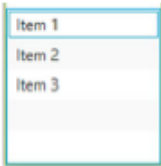
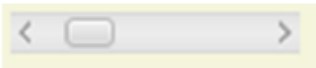

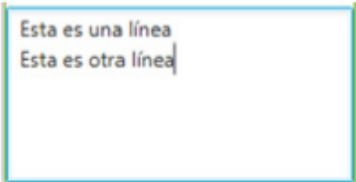



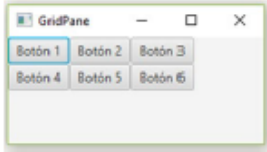
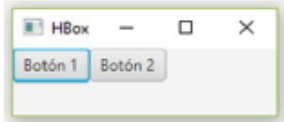
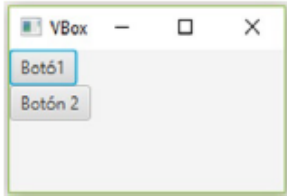
## Componentes gráficos

La siguiente tabla presenta un listado abreviado de diferentes componentes gráficos (nodos) incluidos en JavaFX.

Elementos	Descripción	Figura
<i>Label</i>	Control de texto no editable.	
<i>Button</i>	Un simple control de botón. El cual puede contener texto o un gráfico.	
<i>CheckBox</i>	Un control de selección que aparece como un cuadro con una marca de verificación seleccionada o no.	
<i>RadioButton</i>	Serie de elementos donde solo se puede seleccionar un elemento.	
<i>ChoiceBox</i>	Permite a los usuarios elegir una opción de una lista predefinida de opciones.	
<i>ListView</i>	Muestra una lista horizontal o vertical de elementos que el usuario puede seleccionar.	
<i>ScrollBar</i>	Una barra horizontal o vertical con botones de desplazamiento.	
<i>Spinner</i>	Un campo de texto de una sola línea que permite seleccionar un número en una secuencia ordenada.	
<i>TextArea</i>	Componente de entrada de texto que permite al usuario ingresar varias líneas de texto sin formato.	

<b><i>TextField</i></b>	Componente de entrada de texto que permite al usuario ingresar una sola línea de texto sin formato.	
-------------------------	---	--

Algunos layouts:

Elementos	Descripción	Figura
<b><i>GridPane</i></b>	Presenta sus elementos dentro de una cuadrícula flexible de filas y columnas.	
<b><i>HBox</i></b>	Presenta sus elementos secundarios en una sola fila horizontal.	
<b><i>VBox</i></b>	Presenta sus elementos secundarios en una sola columna vertical.	

## Objetivos de aprendizaje

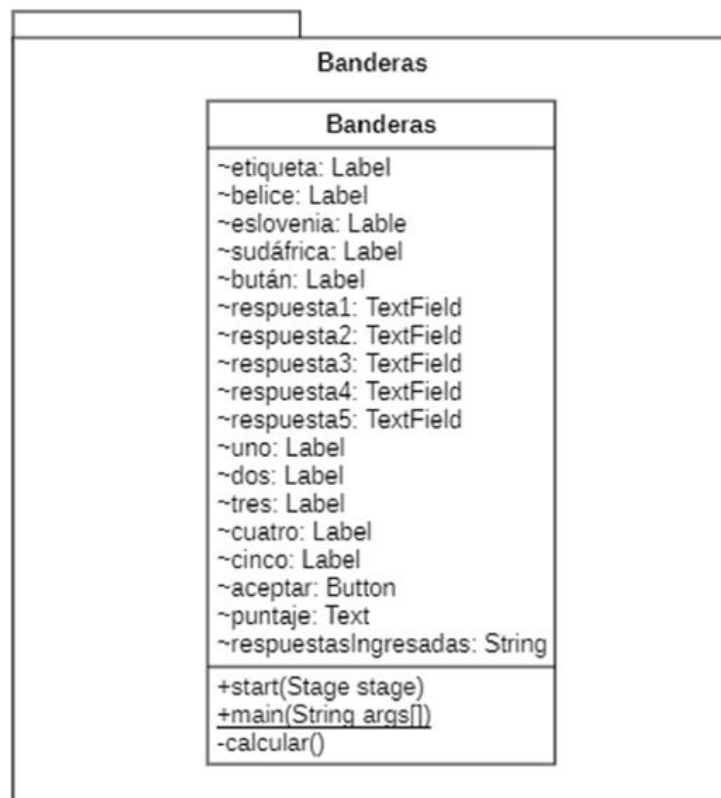
Al finalizar este ejercicio, el lector tendrá la capacidad para:

- Utilizar *GridPane* para distribuir componentes en una ventana.
- Insertar imágenes en ventanas gráficas.
- Insertar diferentes componentes gráficos en un escenario gráfico.

## Enunciado: Banderas

Se requiere desarrollar un programa que muestre en una ventana y en una primera columna el texto de cinco países. En otra columna habrá imágenes de cinco banderas identificadas con un número. El usuario debe ingresar en un campo de texto al lado del país, el número de la bandera que corresponde. El programa debe calcular, en la parte inferior de la ventana, la cantidad de aciertos que obtuvo el usuario.

## Diagrama de clases

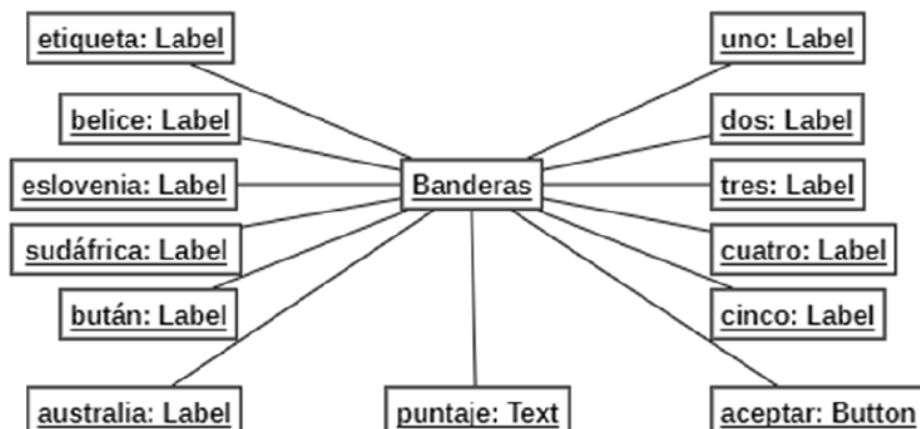


## Explicación del diagrama de clases

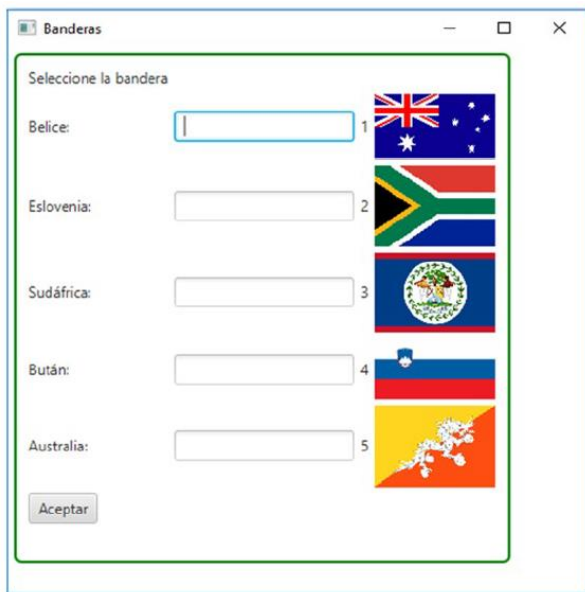
El diagrama de clases muestra un solo paquete denominado "Banderas", el cual contiene una sola clase denominada también "Banderas". La clase cuenta con atributos que indican los diferentes componentes gráficos de una ventana gráfica: etiquetas (*Label*), campos de ingreso de texto (*TextField*), un botón (*Button*), un texto (*Text*) y un atributo que almacena las respuestas ingresadas por el usuario como un *array* de *String*.

La clase *Banderas* también cuenta con los métodos: *start* que inicia la aplicación gráfica; el método *main* (punto de entrada a la aplicación); y el método *calcular* que obtiene el puntaje de respuestas válidas ingresadas por el usuario. El método *main* es un método estático, por lo cual se representa con su texto subrayado. Los métodos *start* y *main* son públicos, esto se indica con el símbolo +. Mientras que el método *calcular* es privado que se indica con el símbolo -.

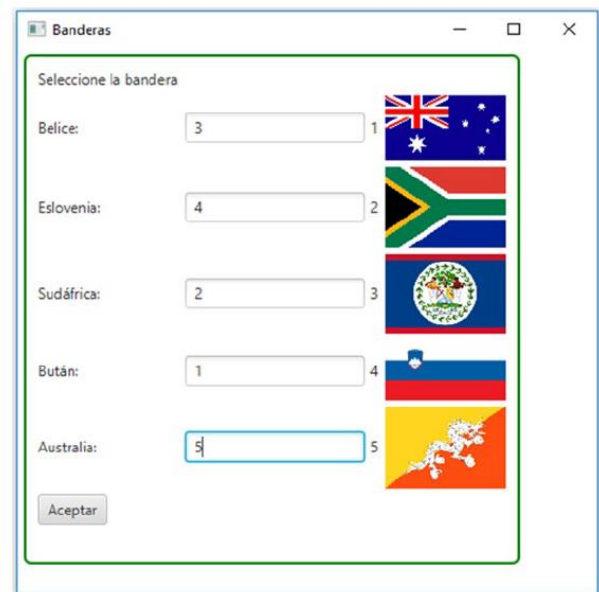
## Diagrama de objetos



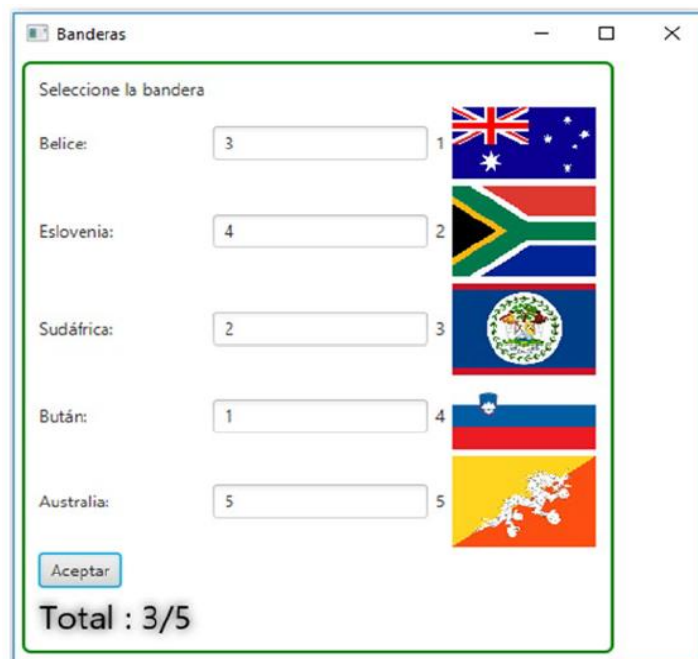
## Ejecución del programa



a) Ventana inicial



b) Ingreso de respuestas



c) Presentación de resultados

## Ejercicios propuestos

- Modificar el programa anterior para que se muestre al usuario cuáles fueron las respuestas correctas e incorrectas.
- Modificar el programa anterior para que se amplíe el juego a diez países con sus banderas.
- Modificar el programa anterior para que de un conjunto de veinte países con sus banderas, el programa seleccione y muestre en pantalla en forma aleatoria solo cinco países.

Instrucciones java del ejercicio

Clase	Método	Descripción
<b>Label</b>	<i>Label()</i>	Constructor de la clase <i>Label</i> .
	<i>void setText(String valor)</i>	Establece el texto de la etiqueta.
	<i>void setFont(Font valor)</i>	Establece la fuente de la etiqueta.
	<i>void setEffect(Effect valor)</i>	Establece el efecto de la etiqueta.
<b>DropShadow</b>	<i>DropShadow()</i>	Constructor de un efecto de alto nivel que genera una sombra del contenido dado.
<b>TextField</b>	<i>TextField()</i>	Constructor de la clase <i>TextField</i> .
<b>Button</b>	<i>Button()</i>	Constructor de la clase <i>Button</i> .
<b>GridPane</b>	<i>GridPane()</i>	Constructor de la clase <i>GridPane</i> .
	<i>void add(Component componente, fila, columna)</i>	Añade un componente en la fila y columna especificada.
	<i>void setStyle(String valor)</i>	Establece un estilo CSS para el panel.
<b>Image</b>	<i>Image()</i>	Constructor de elemento que representa imágenes gráficas.
<b>ImageView</b>	<i>ImageView()</i>	Constructor de un nodo utilizado para pintar imágenes cargadas con la clase <i>Image</i> .
	<i>void setX(double valor)</i>	Establece la posición en el eje x.
	<i>void setY(double valor)</i>	Establece la posición en el eje y.
	<i>void setFitHeight(double valor)</i>	Establece la altura de la imagen.
	<i>void setFitWidth(double valor)</i>	Establece la anchura de la imagen.
	<i>void setPreserveRatio(boolean valor)</i>	Preserva el tamaño de la imagen.
<b>Group</b>	<i>Group()</i>	Constructor de un contenedor con una lista de elementos que se representan en orden.
<b>Stage</b>	<i>Stage(Parent root, double anchura, double altura)</i>	Crea un objeto <i>Scene</i> para un nodo específico con un tamaño específico.
	<i>void setScene(Scene valor)</i>	Especifica la escena utilizada en este escenario.
	<i>void setTitle(String valor)</i>	Establece el título del escenario.
	<i>void sizeToScene()</i>	Establece el tamaño del escenario.
	<i>void show()</i>	Muestra la ventana.
<b>Component</b>	<i>setOnAction()</i>	Gestiona eventos de un componente.