



Financiado por
la Unión Europea
NextGenerationEU



MINISTERIO
DE EDUCACIÓN
Y FORMACIÓN PROFESIONAL



Plan de Recuperación,
Transformación
y Resiliencia



GENERALITAT
VALENCIANA
Conselleria d'Educació,
Cultura i Esport

GVANEXT
Fondos Next Generation
en la Comunitat Valenciana



Tema 12: Entorno Gráfico: 02 Controles

Programación

Desarrollo de Aplicaciones Multiplataforma.

Versión: 20240415.1534

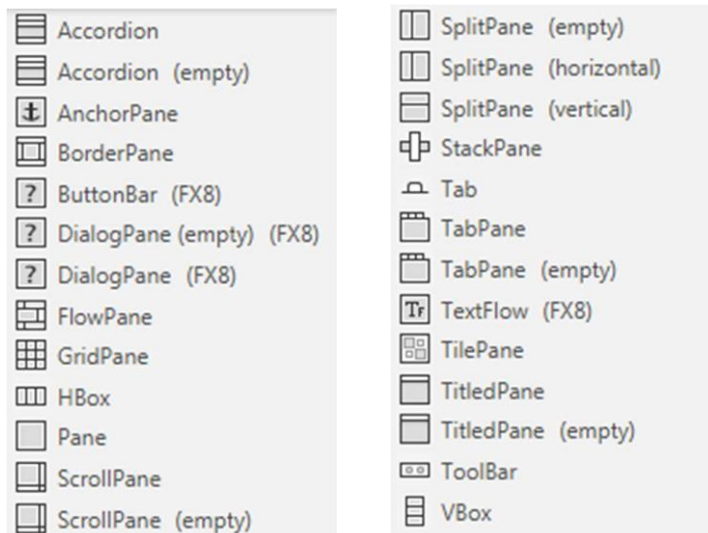
Índice de Contenidos

1.	Controles de JavaFX	3
1.1.	Contenedores	3
1.2.	Controles	3
1.3.	Otros Controles	4
1.3.1.	Gráficos 2D y 3D	4
1.3.2.	Menús	4
1.3.3.	Gráficas Otros.....	4
1.4.	Contenedores de JavaFX	4
1.4.1.	HBox.....	5
1.4.2.	VBox	5
1.4.3.	BorderPane	5
1.4.4.	StackPane.....	7
1.4.5.	GridPane	8
1.4.6.	FlowPane	8
1.4.7.	TilePane	10
1.4.8.	AnchorPane	10
1.5.	Tamaño de los nodos	11
1.6.	Igualando los tamaños de un grupo de nodos:	12
1.6.1.	Para evitar que un nodo crezca	12
1.6.2.	Para evitar que un nodo se haga más pequeño	12
1.6.3.	Para evitar que un nodo cambie de tamaño	12
1.7.	Ejercicios.....	13
1.7.1.	Ejercicio 01 JavaFX – FXML - Calculadora.	13
1.7.2.	Ejercicio 02 JavaFX – FXML - Tres en Raya.....	14

1. Controles de JavaFX

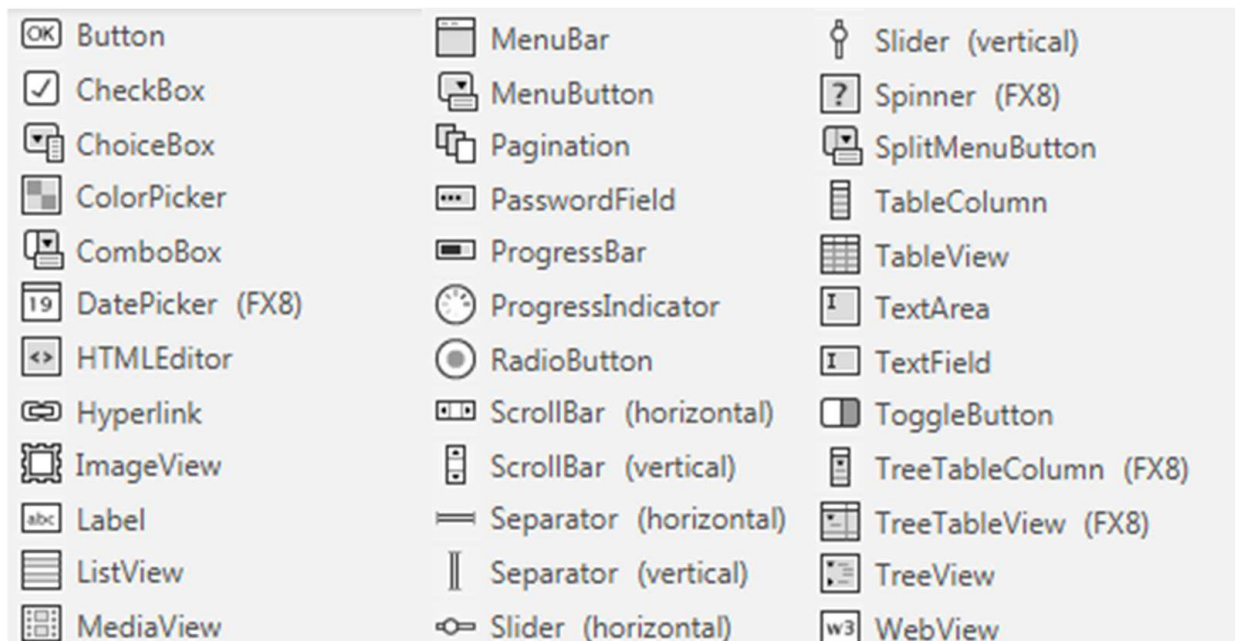
1.1. Contenedores

Contienen otros controles (y contenedores), organizados automáticamente



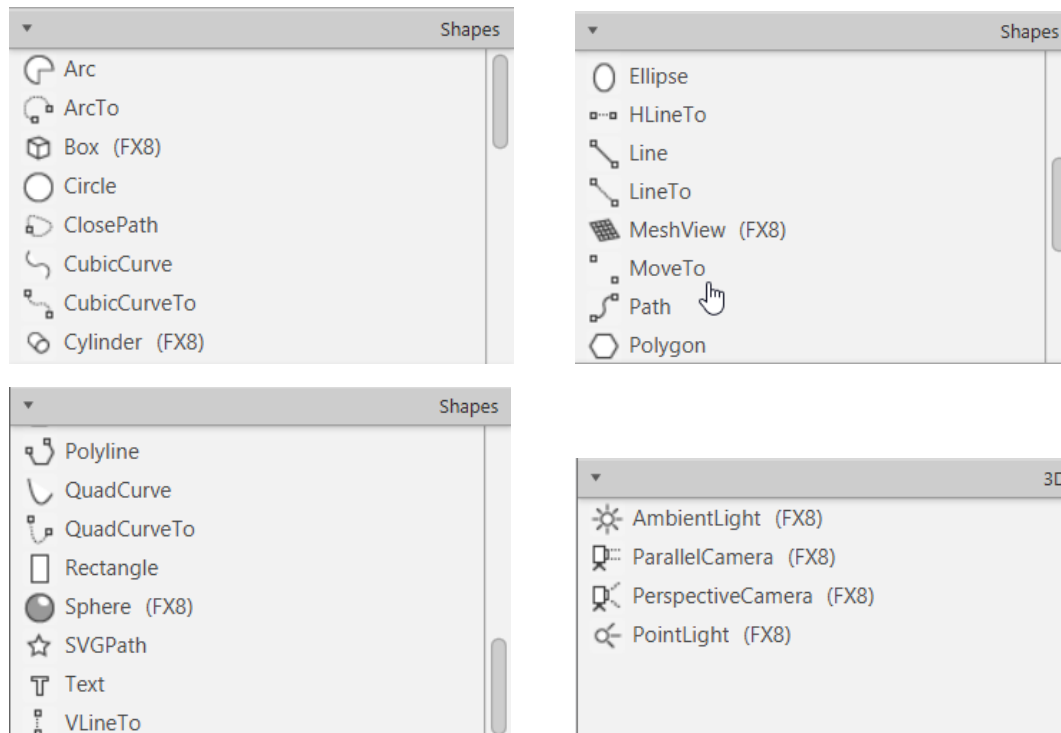
1.2. Controles

Implementan la interacción con el usuario

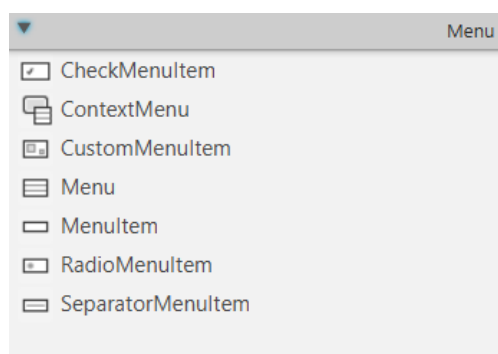


1.3. Otros Controles

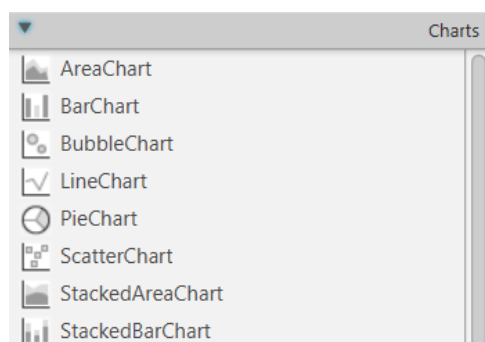
1.3.1. Gráficos 2D y 3D



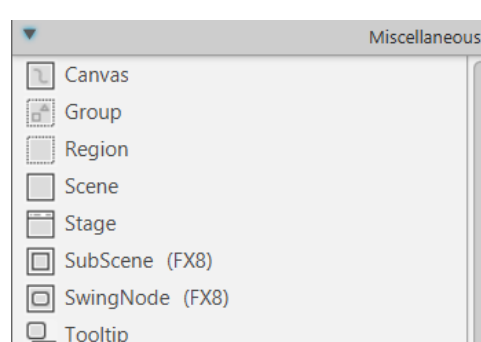
1.3.2. Menús



1.3.3. Gráficas



Otros



1.4. Contenedores de JavaFX

El programador puede establecer por código el tamaño y posición de los widgets en pantalla, pero es difícil de adaptarse a cambios de tamaños de la ventana

Es más fácil (y obligatorio) usar contenedores para hacer que los controles se ajusten correctamente a cualquier tamaño de ventana

Hay contenedores para organizar los controles por filas, columnas, apilamientos, en cuadrícula, etc.

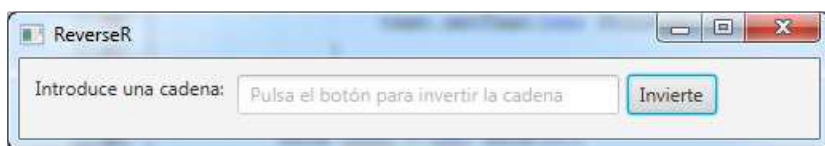
- El contenedor recalcula automáticamente la posición y tamaño de los controles que contiene al cambiar el tamaño de la ventana

1.4.1. HBox

Organiza a sus hijos en fila

Podemos establecer un área en blanco en cada lado del HBox (padding), y el espacio entre los hijos (spacing)

```
Label lbl = new Label("Introduce una cadena:");
TextField text = new TextField();
text.setPrefColumnCount(20);
text.setPromptText("Pulsa el botón para invertir la cadena");
Button btn = new Button("Invierte");
btn.setOnAction(new EventHandler<ActionEvent>() {
    @Override public void handle(ActionEvent event) {
        String tmp = text.getText();
        text.setText(new StringBuffer(tmp).reverse().toString());
    }
});
HBox root = new HBox(5);           // 5 píxeles entre hijos
root.setPadding(new Insets(10));   // 10 píxeles alrededor
root.getChildren().addAll(lbl, text, btn);
```

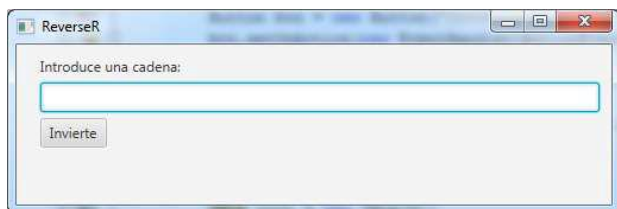


1.4.2. VBox

Similar a HBox, para organizar nodos en vertical

También define padding y spacing

```
// El mismo código de antes
VBox root = new VBox(5); // 5 píxeles entre hijos
root.setPadding(new Insets(10, 10, 10, 20)); // arriba, derecha, abajo, izq.
root.getChildren().addAll(lbl, text, btn);
```



1.4.3. BorderPane

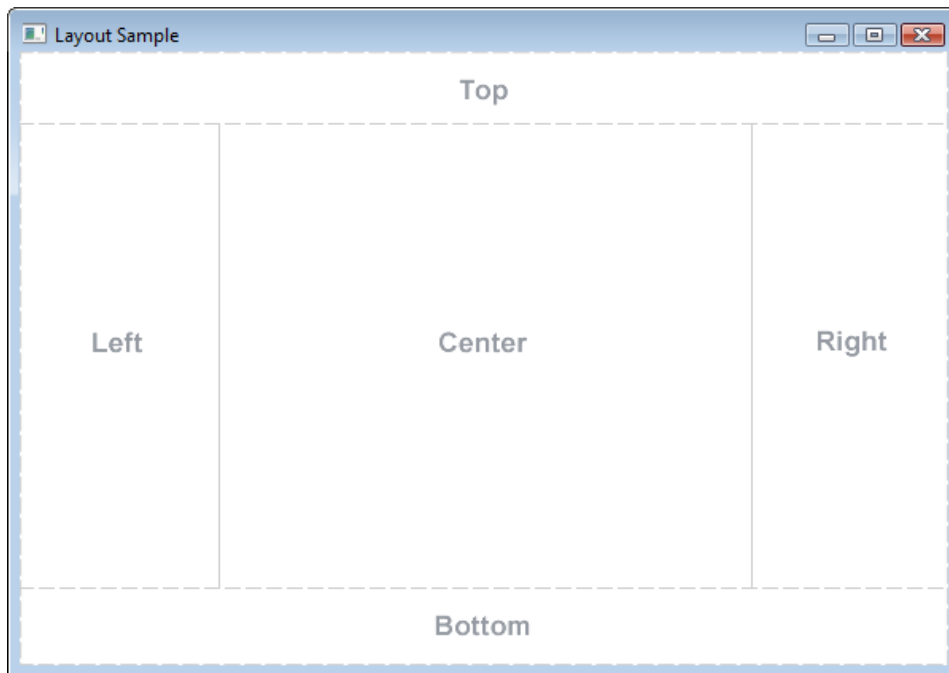
Define 5 regiones: Arriba, izquierda, centro, derecha y abajo

Se usa para construir ventanas principales (con barra de herramientas, barra de estado, panel de navegación a la izquierda y área de trabajo en el centro)

Si sobra espacio, crece la zona central. Si falta, se pueden solapar

Puede haber regiones vacías

```
BorderPane root = new BorderPane();  
root.setTop(new Button("Arriba"));  
HBox group = new HBox();  
group.getChildren().addAll(  
    new Button("Abajo 1"),  
    new Button("Abajo 2"));  
root.setBottom(group);  
// y setLeft, setCenter y setRight
```

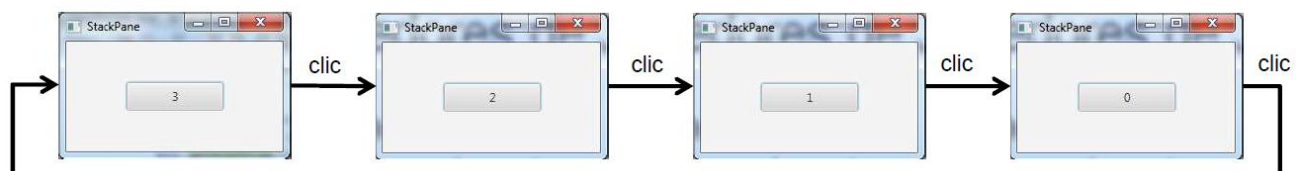
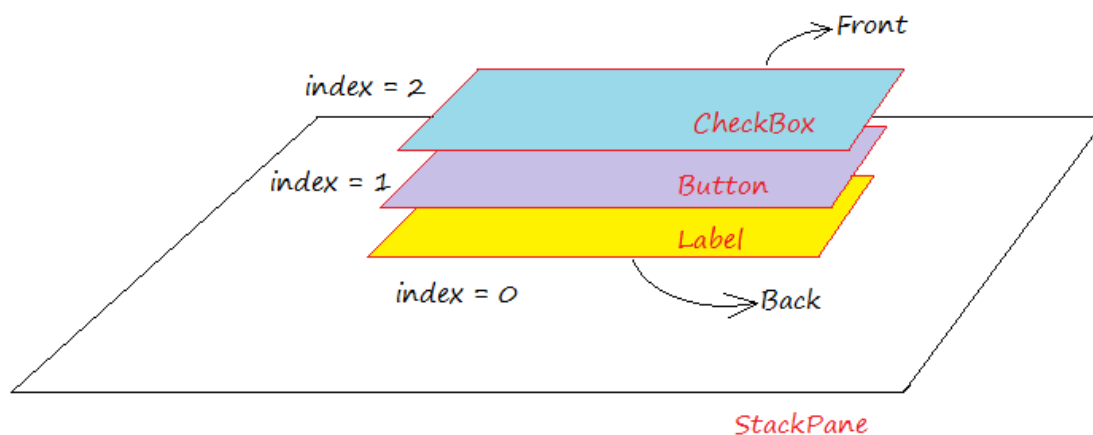


1.4.4. StackPane

Organiza sus elementos uno encima del otro

Esto permite superponer texto sobre otros elementos (formas, imágenes, etc) o mostrar sólo un elemento de una pila

```
StackPane root = new StackPane();
for (int i = 0; i < 4; i++) {
    Button btn = new Button(Integer.toString(i));
    btn.setPrefSize(100, 30);
    btn.setOnAction(new EventHandler<ActionEvent>() {
        @Override public void handle(ActionEvent event) {
            ((Node)event.getSource()).toBack(); // getSource devuelve la fuente del
            evento
        } // toBack envía al nodo al final de
    }); // la lista de hijos del contenedor
    root.getChildren().add(btn);
}
```



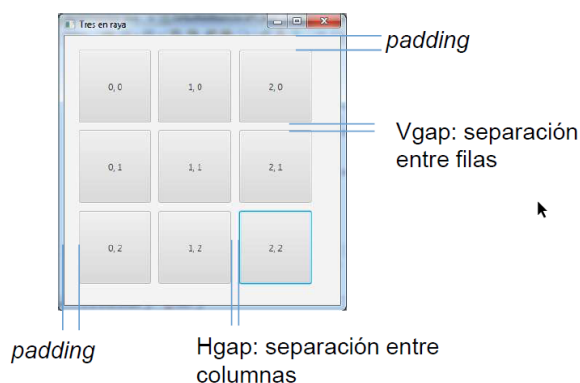
1.4.5. GridPane

Organiza sus elementos en una matriz de filas y columnas

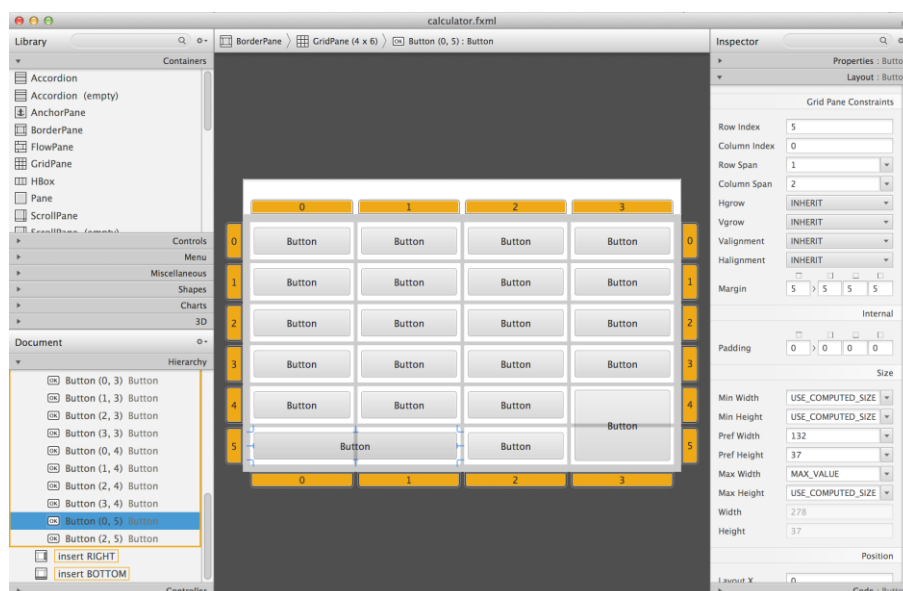
Un nodo puede expandirse a varias celdas adyacentes

Se utiliza para crear formularios

```
GridPane root = new GridPane();
for (int col = 0; col < 3; col++) {
    for (int row = 0; row < 3; row++) {
        Button btn = new Button(
            Integer.toString(col) + ", " + row);
        btn.setPrefSize(100, 100);
        root.getChildren().add(btn);
        GridPane.setConstraints(btn, col, row);
    }
}
root.setVgap(10);
root.setHgap(10);
root.setPadding(new Insets(20));
```



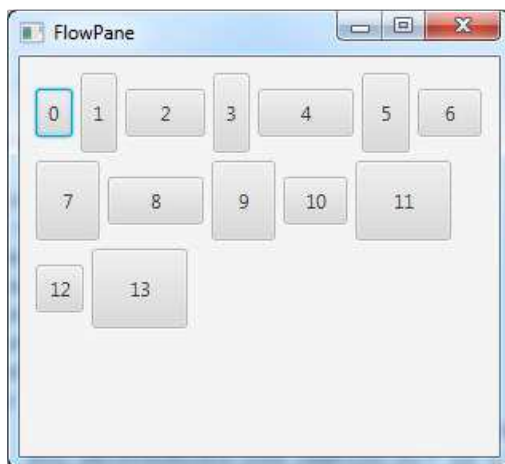
Tenemos la posibilidad de realizar expansión a celdas adyacentes



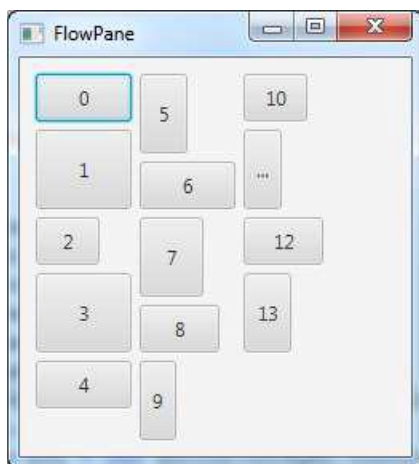
1.4.6. FlowPane

Organiza los elementos secuencialmente, de izquierda a derecha o de arriba abajo, de acuerdo con el tamaño del contenedor

Orientación Horizontal (por defecto)

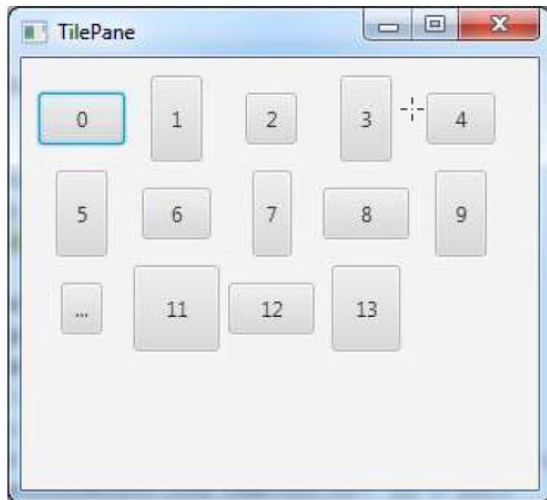


Orientación Vertical



1.4.7. TilePane

Es similar al FlowPane, pero organiza los elementos secuencialmente en una matriz de filas y columnas donde cada celda es del mismo tamaño



Orientation: HORIZONTAL
(por defecto)



Orientation: VERTICAL

1.4.8. AnchorPane

Permite anclar sus hijos a una distancia de uno de los bordes del contenedor, o al centro

Al cambiar el tamaño de la ventana, los nodos mantendrán su posición relativa al punto de anclaje

Un nodo puede estar anclado a más de un punto de anclaje

1.5. Tamaño de los nodos

La ventaja de utilizar los contenedores de JavaFX es que recalculan la posición/tamaño de los nodos al cambiar el tamaño de la ventana

Los nodos cambian su tamaño de acuerdo con varios parámetros, que veremos a continuación

- Hay nodos que no cambian su tamaño: formas, texto o grupos.

En general, podemos especificar:

- el tamaño de un nodo mediante su tamaño preferido (Pref Width y Pref Height)
- la posición de un nodo mediante las propiedades de alineamiento del contenedor

Por defecto, los controles calculan su tamaño preferido basándose en su contenido

- El botón se ajusta al texto que contiene

Los nodos definen tamaños mínimos y máximos

- El máximo de un botón, por defecto está al tamaño preferido (por que normalmente no queremos que un botón crezca)
- Otros nodos, como por ejemplo un ListView, sí que queremos que crezcan hasta ocupar todo el espacio disponible.

Podemos cambiar los tamaños:

- Preferidos (setPrefHeight, setPrefWidth o setPrefSize)
- Máximo (setMaxHeight, setMaxWidth, o setMaxSize)
- Mínimo (setMinHeight, setMinWidth, o setMinSize)

Aceptan valores en píxeles, o Double.MAX_VALUE, Control.USE_PREF_SIZE o Control.USE_COMPUTED_SIZE

1.6. Igualando los tamaños de un grupo de nodos:

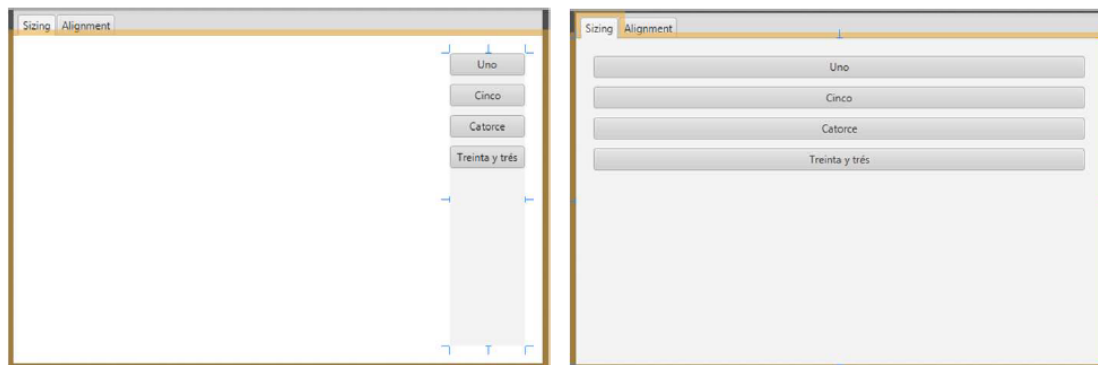
Máximo por defecto Máximo a MAX_VALUE



```
VBox root = new VBox();
Button b1 = new Button("Uno");
Button b5 = new Button("Cinco");
[...]
b1.setMaxWidth(Double.MAX_VALUE);
b5.setMaxWidth(Double.MAX_VALUE);
b14.setMaxWidth(Double.MAX_VALUE);
b33.setMaxWidth(Double.MAX_VALUE);
```

En un BorderPane, el área central toma todo el espacio disponible; el resto sólo lo que necesitan:

Mismo VBox que arriba, en el área derecha y en el central de un BorderPane



1.6.1. Para evitar que un nodo crezca

Establecer el tamaño máximo a `Control.USE_PREF_SIZE`, o a un tamaño máximo en píxeles

1.6.2. Para evitar que un nodo se haga más pequeño

Establecer su tamaño mínimo a `Control.USE_PREF_SIZE`

1.6.3. Para evitar que un nodo cambie de tamaño

Establecer los tamaños mínimo, máximo y preferido al mismo valor

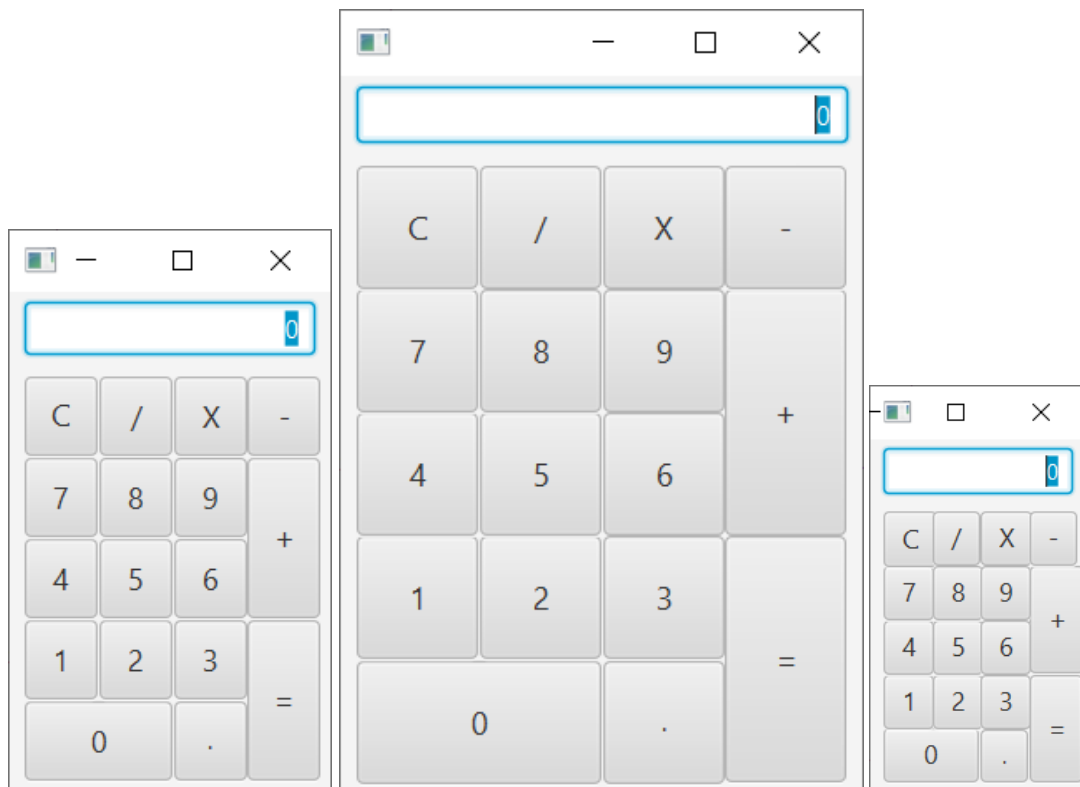
1.7. Ejercicios

1.7.1. Ejercicio 01 JavaFX – FXML - Calculadora.

1. Crear un proyecto JavaFX con la vista de una calculadora:



Ejemplos redimensionando la ventana:



1.7.2. Ejercicio 02 JavaFX – FXML - Tres en Raya

2. Crear un proyecto JavaFX con la vista siguiente:

