



# CONTENEDORES

---

Interacción Persona Computador

Depto. Sistemas Informáticos y Computación

UPV

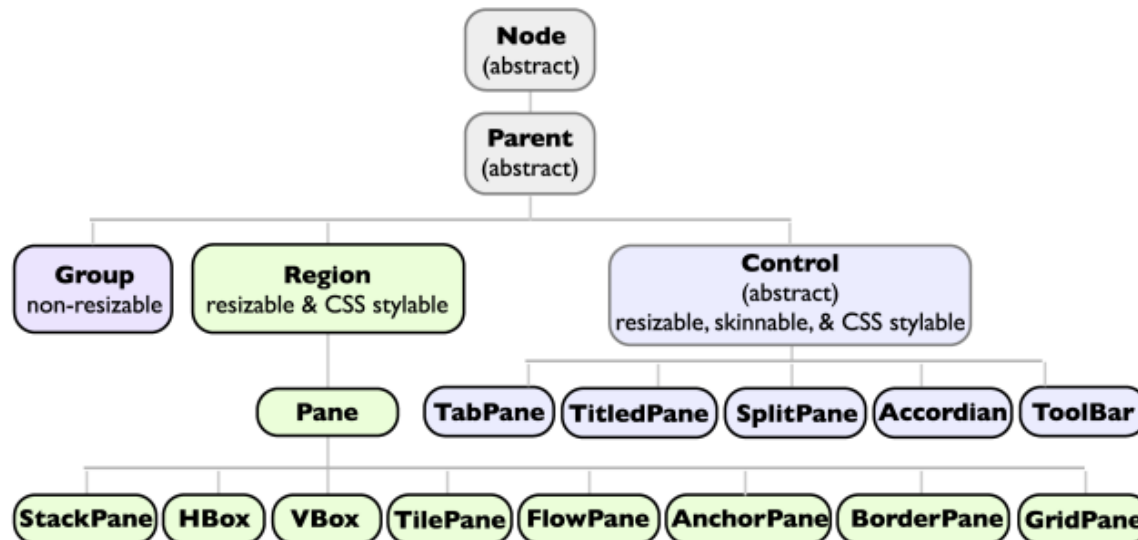
Práctica Anexo

Fuente: <https://o7planning.org/en/11009/javafx>

# Componentes para la IGU

- Clases de Contendores: un panel de layout es un nodo que contiene otros nodos (sus hijos)
- El panel de layout se encarga de reubicar/redimensionar a sus hijos de acuerdo a su **política de redimensionamiento**.
- Para ello calcula la posición (x,y) de los nodos hijos y la altura y anchura de los mismos.

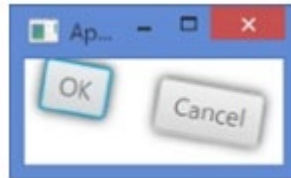
## JavaFX 2.0 Layout Classes



El resto de controles gráficos descenden directamente o indirectamente de Control

# Redimensionamiento de controles IGU

- El grupo (**Group**) no es redimensionable y no aplica layout a los controles contenidos en él.
- Los hijos se posicionan mediante código o con SceneBuilder.
- El tamaño del grupo depende del tamaño de los hijos.
- Útil para aplicar efectos como rotaciones



- El tamaño de un nodo **redimensionable** (Region, Control, etc) depende de dos cosas:
  - Sus **preferencias** de tamaño min/pref/max
  - La **política de redimensionamiento** del contenedor padre

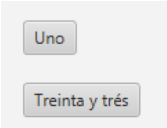
# Contenedores de JavaFX

## Controlando el tamaño de los nodos de JavaFX

- 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 a 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

# Contenedores de JavaFX

## Controlando el tamaño de los nodos de JavaFX

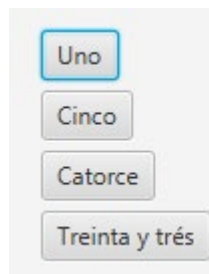
- Por defecto, los controles calculan su tamaño preferido basándose en su contenido
    - El botón se ajusta al texto que contiene
- 
- Los nodos definen tamaños mínimos y máximos
    - El máximo de un botón, por defecto está al tamaño preferido (por que normalmente no queremos que un botón crezca)
    - 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`)
      - Todos aceptan valores en píxeles, o `Double.MAX_VALUE`, `Control.USE_PREF_SIZE` o `Control.USE_COMPUTED_SIZE`

# Contenedores de JavaFX

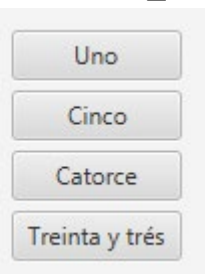
## Controlando el tamaño de los nodos de JavaFX

- Igualando los tamaños de un grupo de nodos:

Máximo por defecto

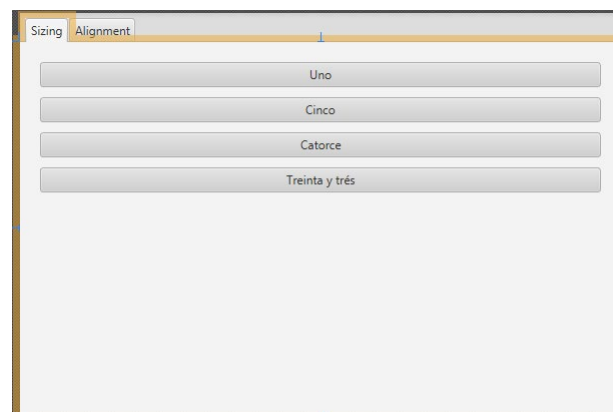
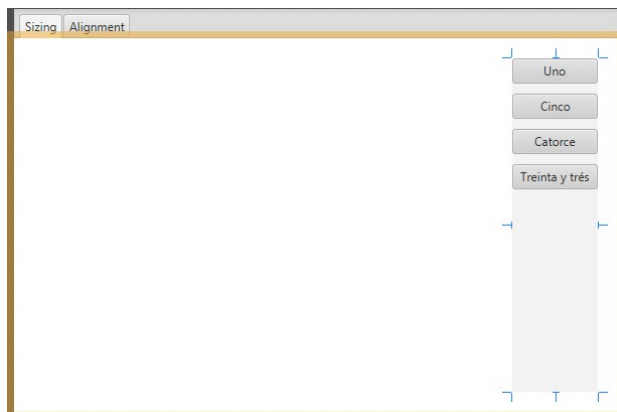


Máximo a MAX\_VALUE



```
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:

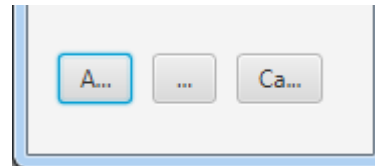
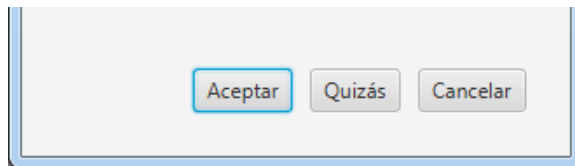


Mismo VBox que arriba, en el área derecha y en el central de un BorderPane

# Contenedores de JavaFX

## Controlando el tamaño de los nodos de JavaFX

- Para evitar que un nodo crezca:
  - Establecer el tamaño máximo a `Control.USE_PREF_SIZE`, o a un tamaño máximo en píxeles
- Para evitar que un nodo se haga más pequeño
  - Establecer su tamaño mínimo a `Control.USE_PREF_SIZE`

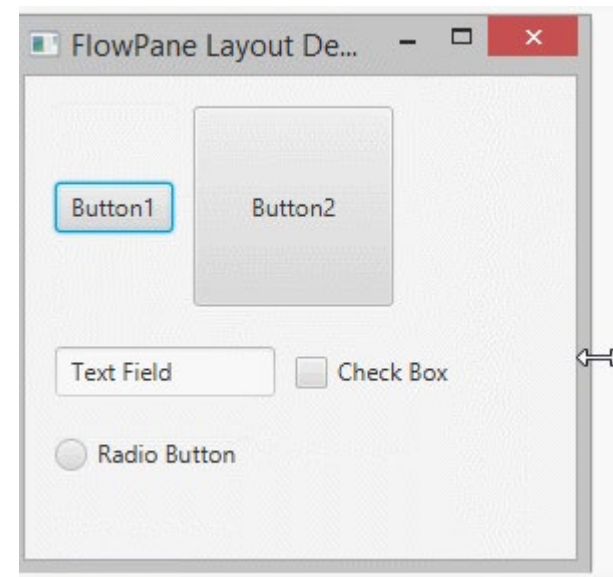


Por defecto, los botones encogen

- Para evitar que un nodo cambie de tamaño
  - Establecer los tamaños mínimo, máximo y preferido al mismo valor

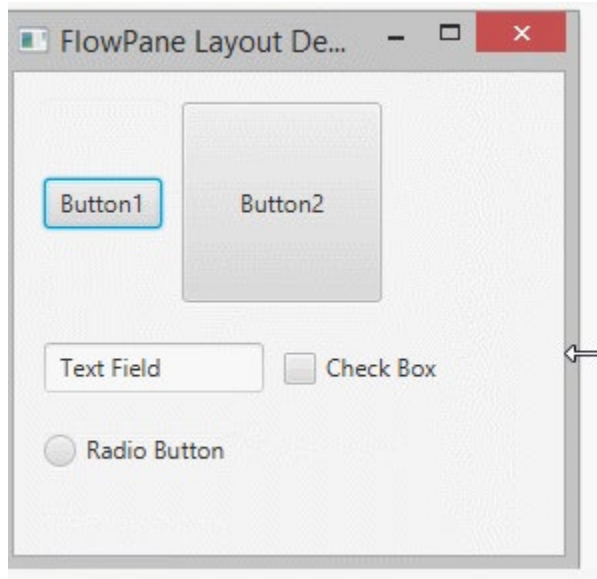
# FlowPane

- **Redimensiona** sus nodos al tamaño preferido de los mismos.
- Por defecto los sitúa horizontalmente, en el mismo orden en el que han sido añadidos.
- Al decrecer en anchura desplaza los nodos que no caben a las siguientes filas.

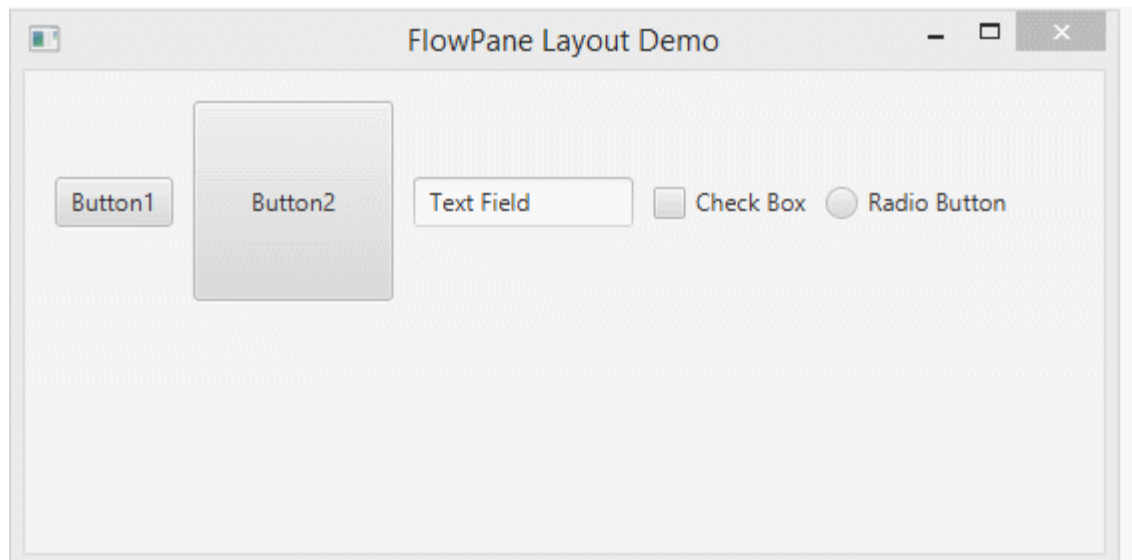
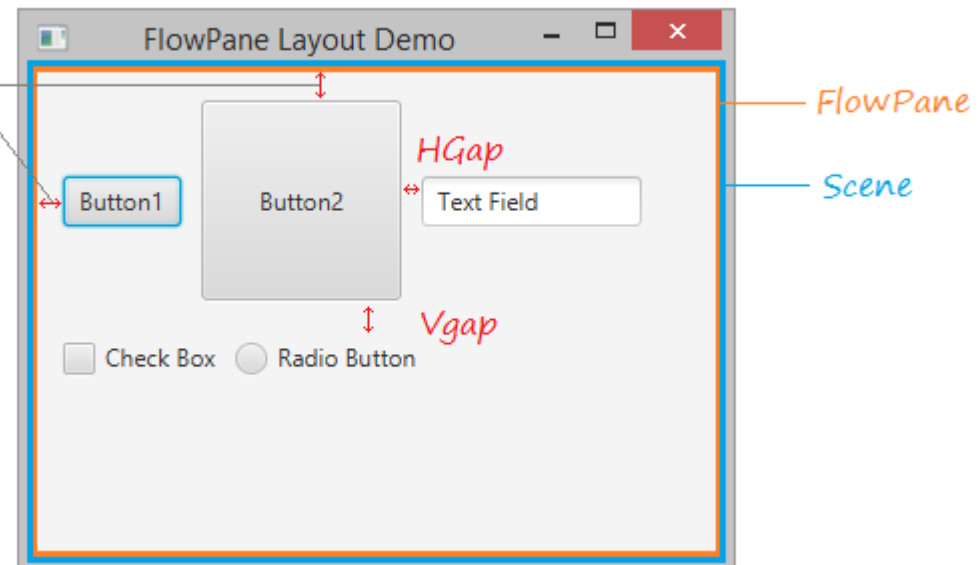




# FlowPane

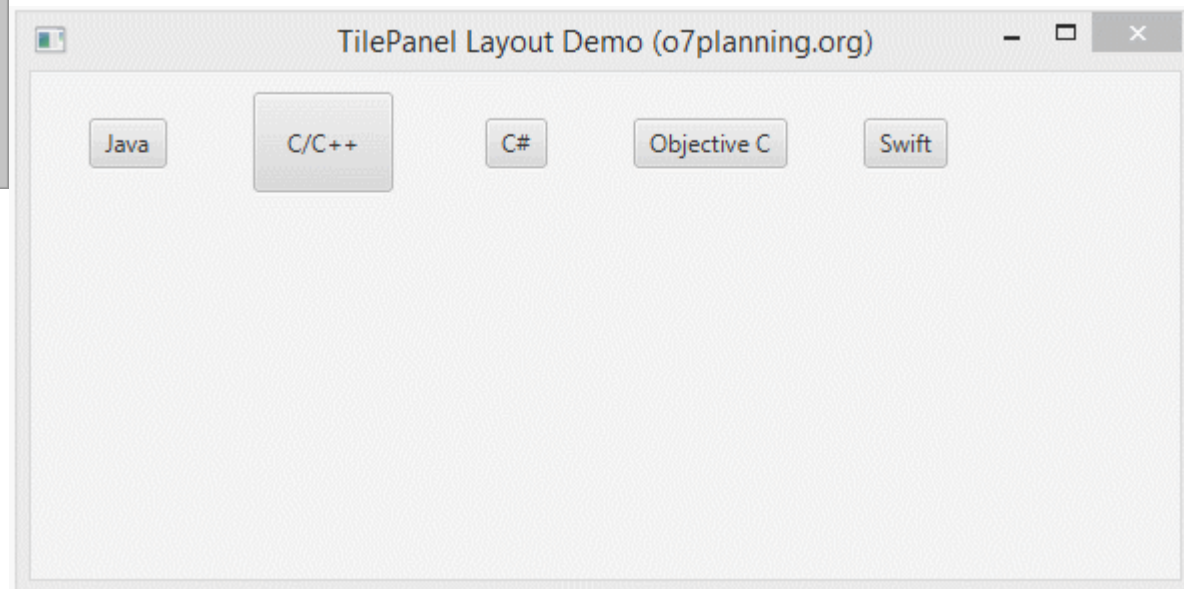
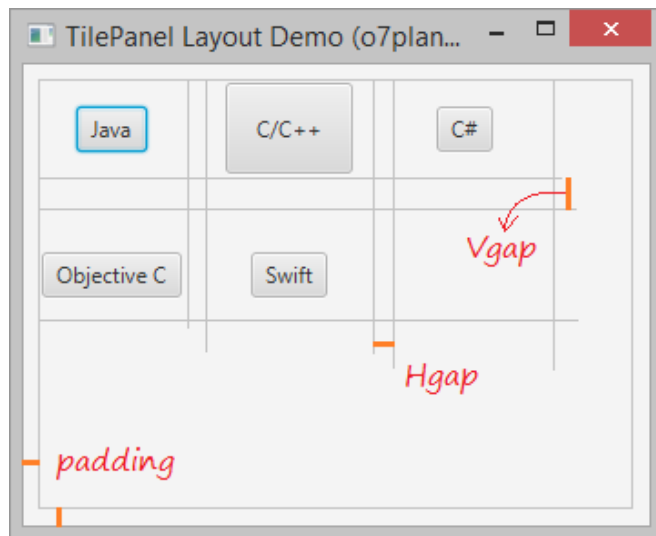


*padding  
(top, left)*



# TilePane

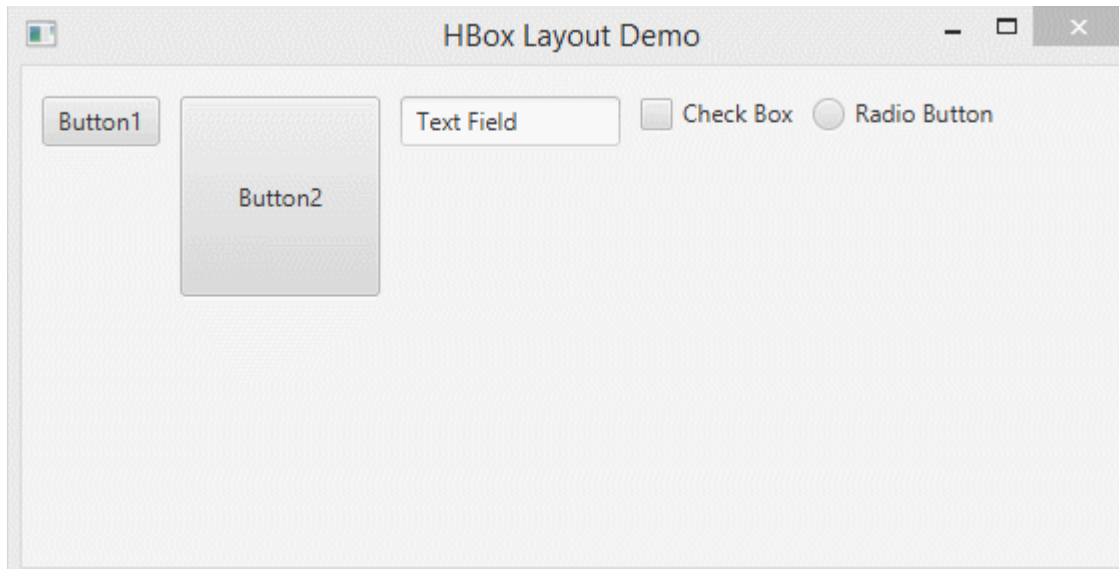
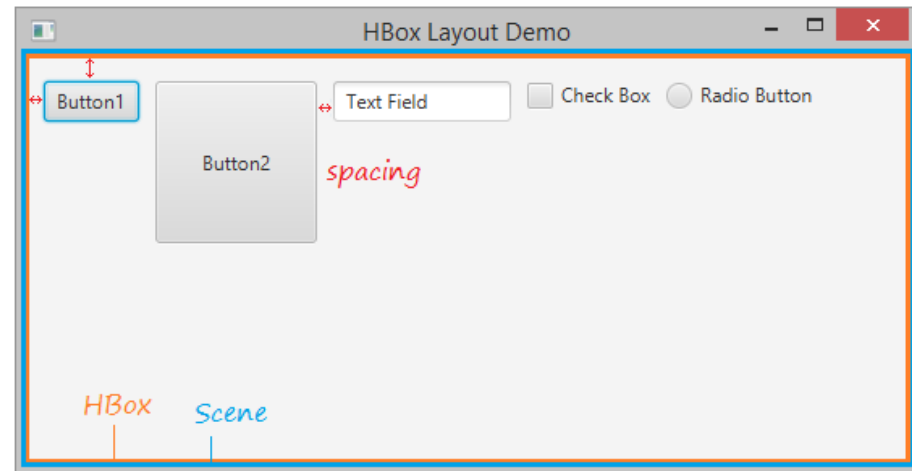
- Parecido al FlowPane, pero cada componente hijo ocupa una celda del mismo tamaño



# HBox

- Es un contenedor que sitúa sus componentes en una única fila

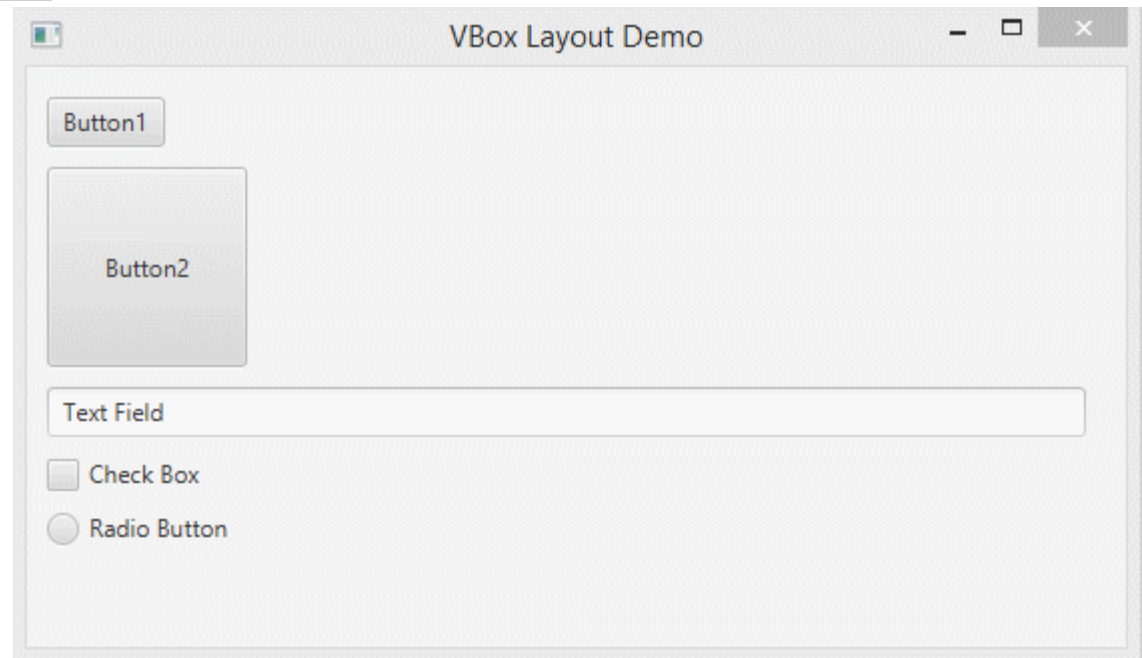
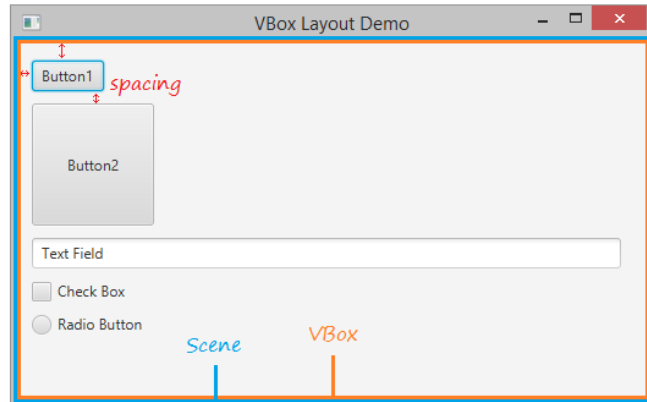
*padding  
(top, left)*



# VBox

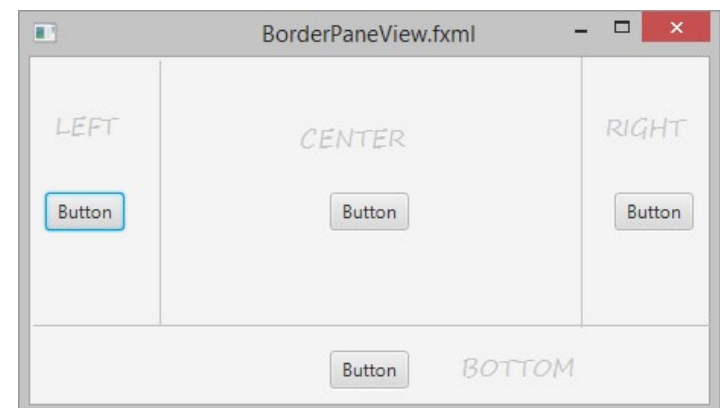
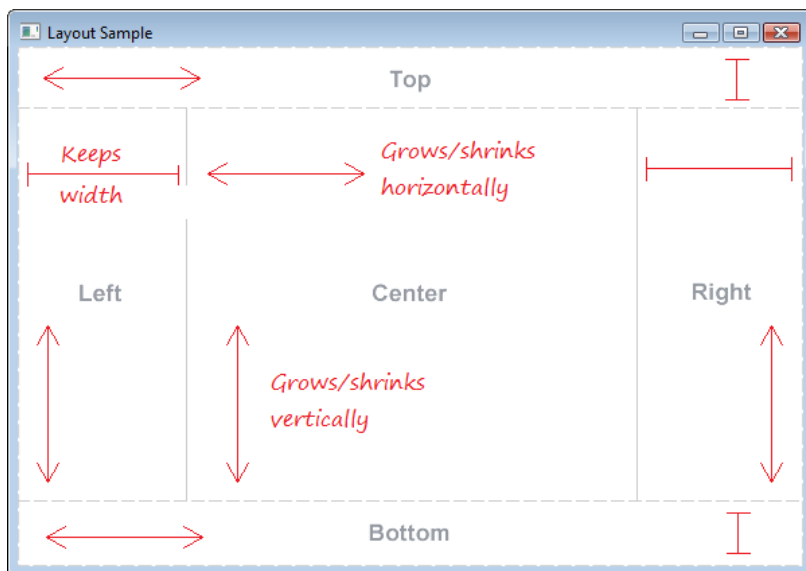
- Utiliza una columna para situar los componentes

*padding  
(top, left)*



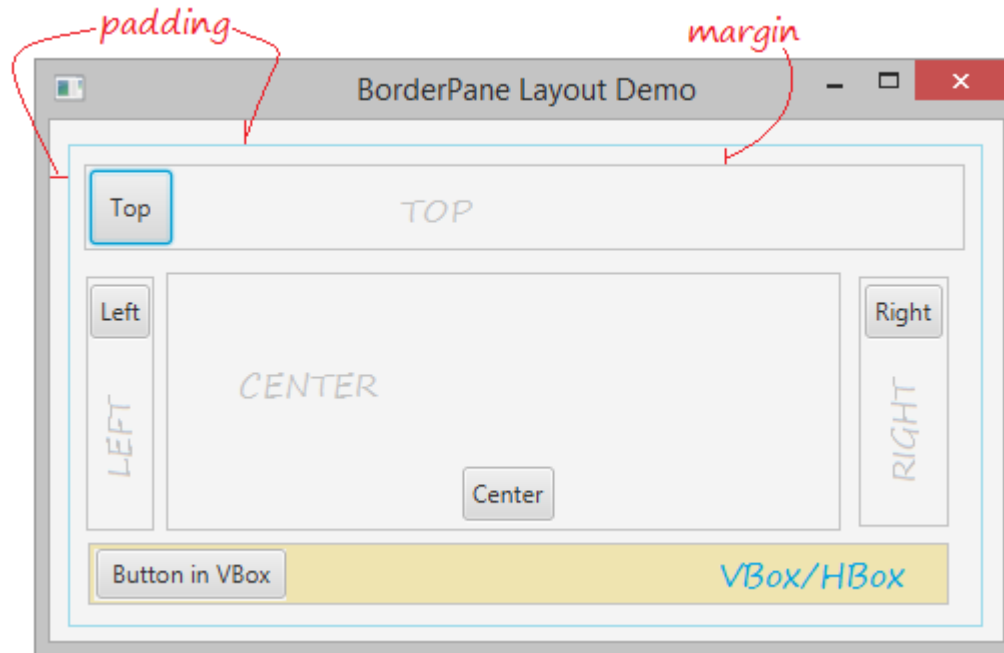
# BorderPane

- Dividido en 5 zonas: Top, Bottom, Left, Center y Right
  - La zona superior/inferior puede crecer/decrecer horizontalmente manteniendo la altura.
  - La zona derecha/izquierda puede crecer/decrecer verticalmente manteniendo la anchura.
  - Si una zona no está presente (Top por ejemplo) el resto de las zonas ocupan su espacio.



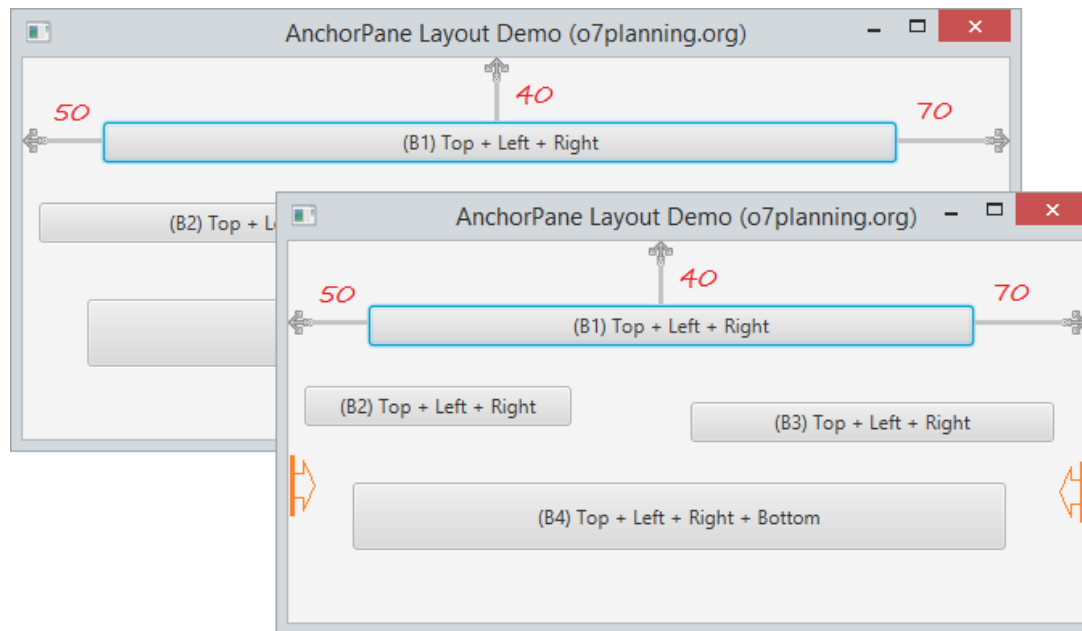
# BorderPane

- Los componentes que se depositan en una de las áreas puede que no la llenen, esto pasa con los botones.
- Si se sitúa un Hbox o VBox estos llenan la zona



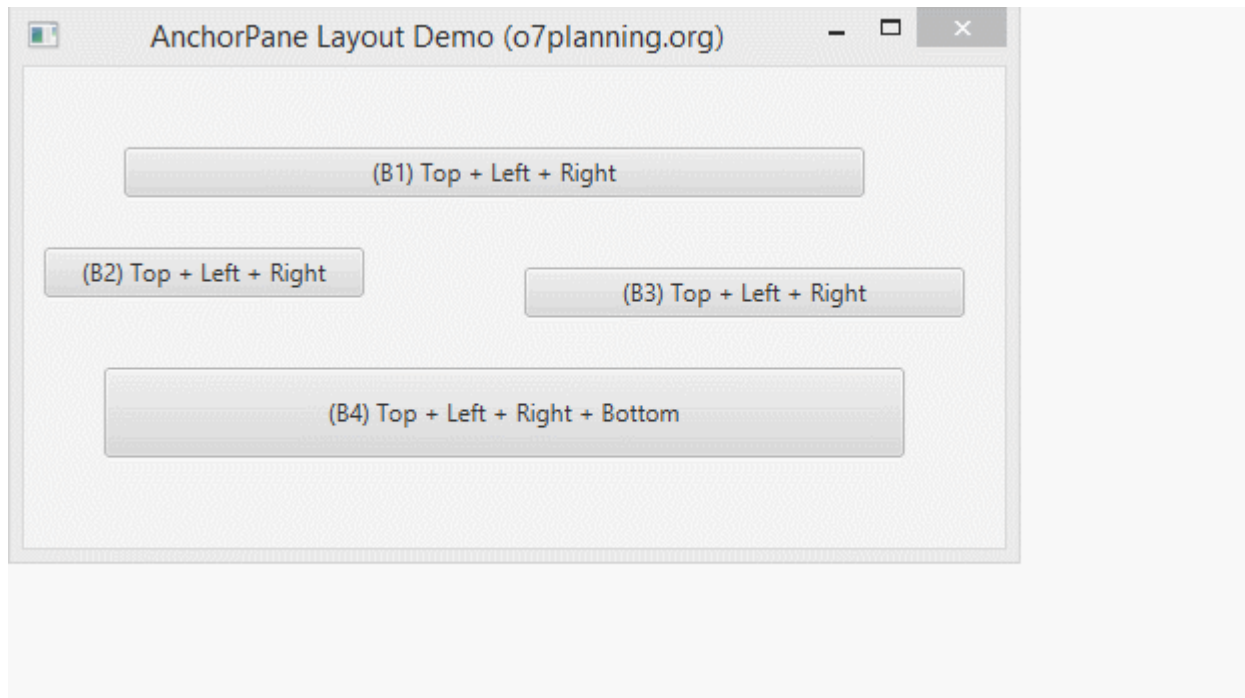
# AnchorPane

- Permite anclar los componentes a una o más zonas: superior, inferior, derecha e izquierda



# AnchorPane

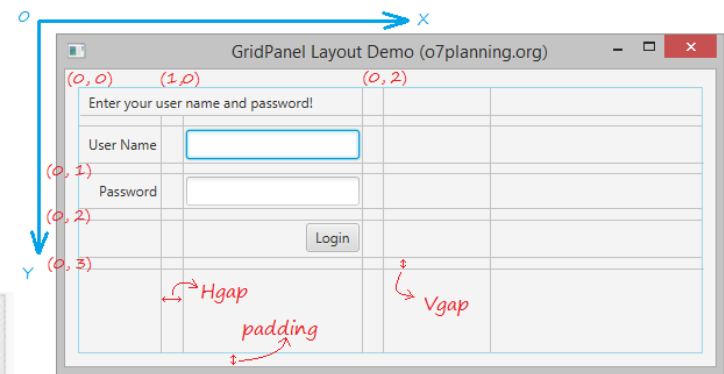
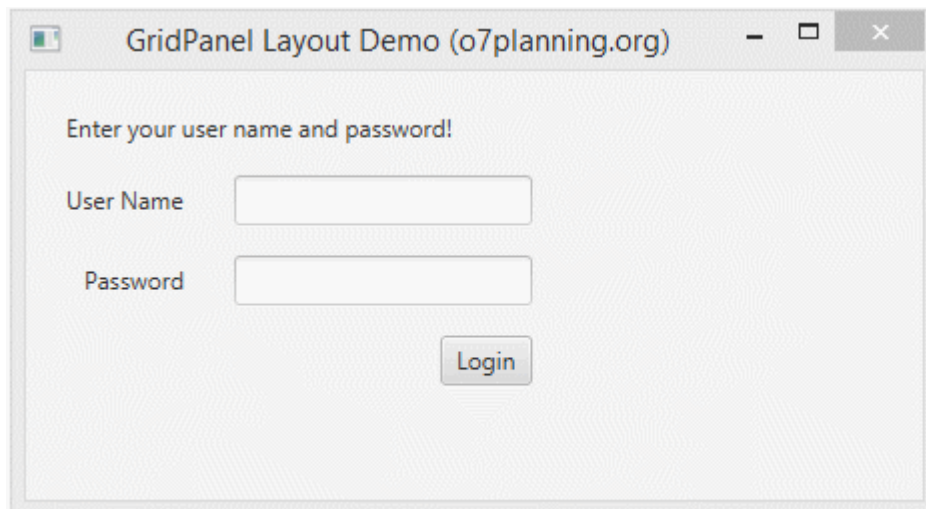
- Al redimensionar se mantiene la distancia a los puntos de anclaje





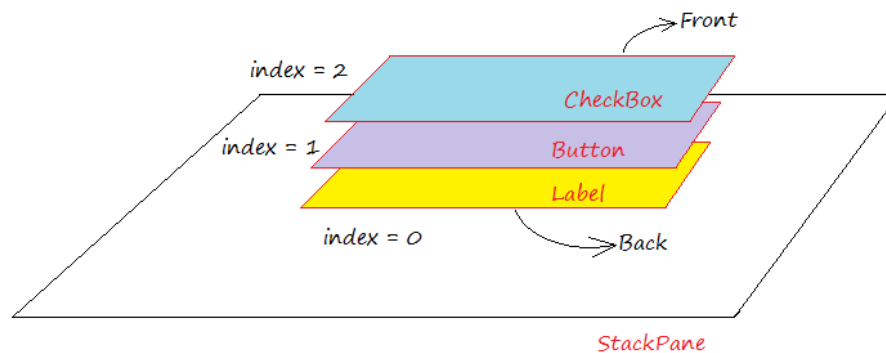
# GridPane

- Es un contenedor que divide su superficie en filas y columnas.
- Los componentes situados en sus celdas pueden ocupar más de una fila y/o columna

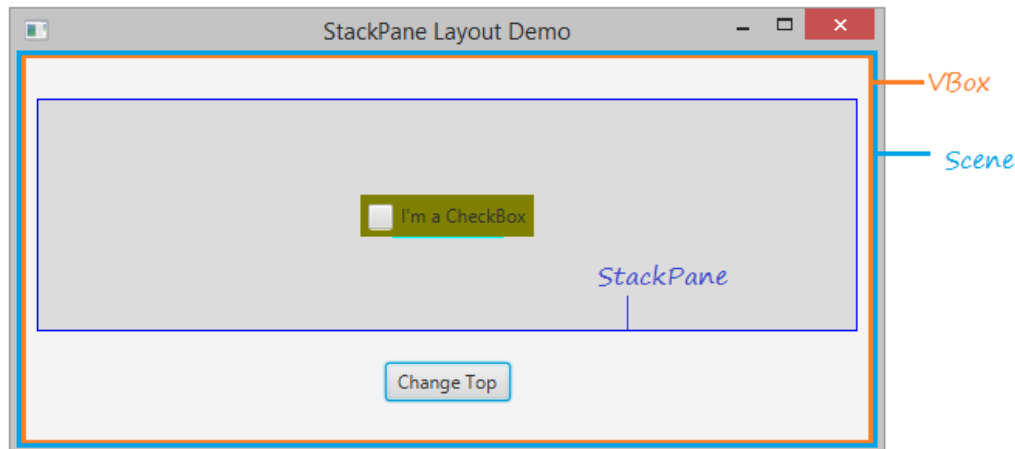


# StackPane

- Permite apilar distintos componentes de interfaz de usuario de modo que únicamente puede verse el que está en la cima de la pila

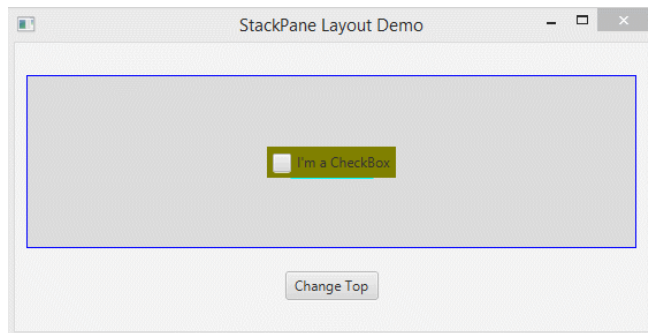


- Puede cambiarse el orden de un elemento en la pila:
  - `childNodes.toFront()`
  - `childNodes.toBack()`

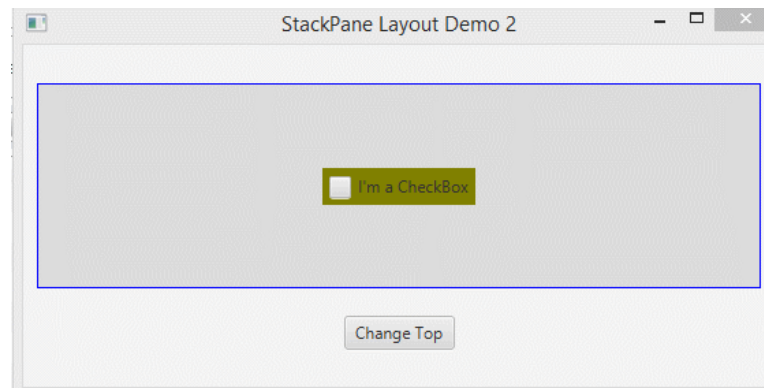


# StackPane

- Por defecto los muestra superpuestos y dependiendo del tamaño podría verse alguno de los inferiores



- Usando `Node.setVisible(false)` se ocultan los que sobresalen



# Tamaño mínimo de un Stage

- Para limitar la mínima anchura y altura de una ventana (Stage) deben usarse las siguientes instrucciones:

```
@Override
public void start(Stage stage) throws Exception {
    Parent root = FXMLLoader.load(getClass().getResource("FXMLDocument.fxml"));

    Scene scene = new Scene(root);
    stage.setScene(scene);

    stage.setMinWidth(root.minWidth(-1)); //anchura mínima del contenedor raíz
    stage.setMinHeight(root.minHeight(-1)); //altura mínima del contenedor raíz
    stage.show();
}
```