



PRÁCTICA 6. DIAGRAMAS EN JAVAFX 8

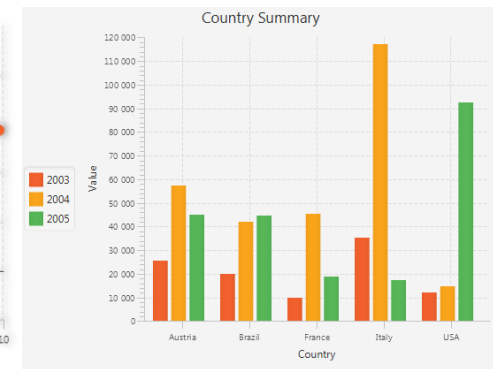
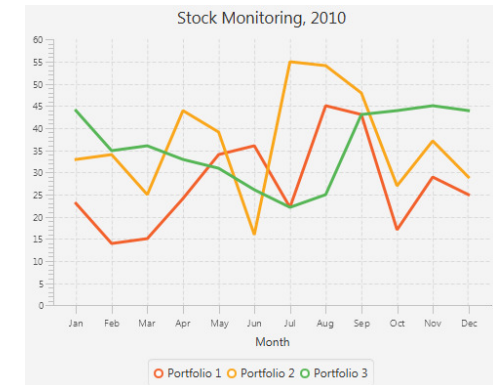
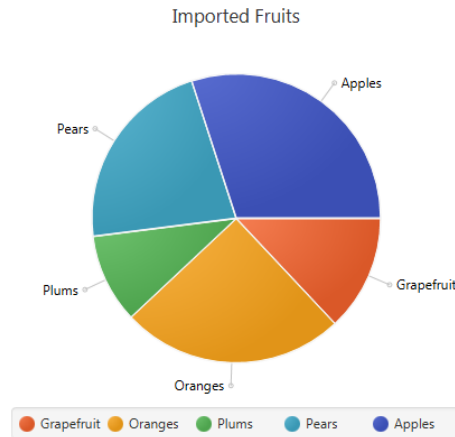
Interfaces Persona Computador

Depto. Sistemas Informáticos y Computación

UPV

Índice

- Diagramas en JavaFX 8
 - Diagrama de tarta
 - Diagrama de líneas
 - Diagrama de áreas
 - Diagrama de burbujas
 - Diagrama de puntos
 - Diagrama de barras
 - Operaciones con diagramas
- Ejemplo de diagrama
- Ejercicio



Introducción

- Los diagramas se pueden programar por código o utilizando la herramienta Scene Builder
- En la primera parte vemos la programación por código
- Luego veremos un ejemplo creado mediante Scene Builder
- Se recomienda utilizar el Scene Builder
- Todos los diagramas visualizan datos 2D, excepto el primero, el diagrama de tarta

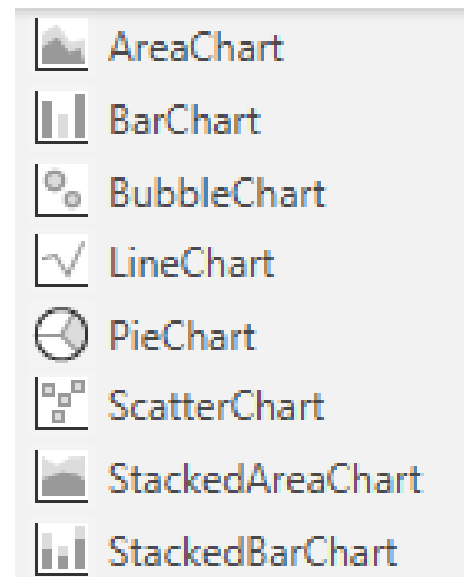


Diagrama de tarta

- Visualiza datos como un círculo dividido en trozos que representan un porcentaje correspondiente a un valor

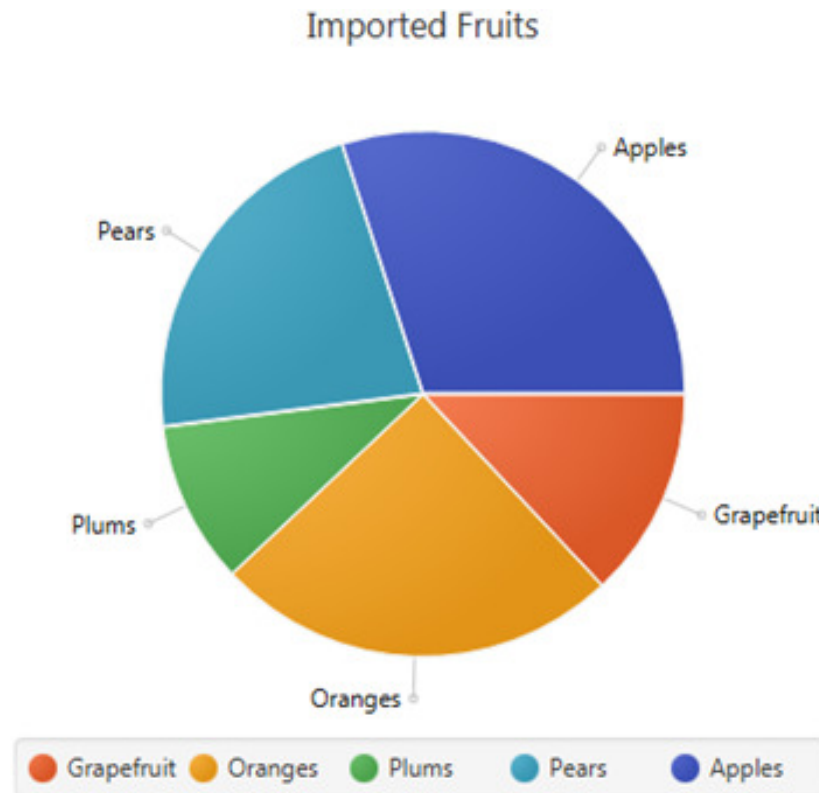


Diagrama de tarta

- Para visualizar el diagrama:
 - Crear un *PieChart*
 - Crear una lista observable para guardar los datos del diagrama
 - Los datos son del tipo *PieChart.Data* y cada uno contiene un *String* y el valor numérico asociado a ese *String*

```
PieChart chart = new PieChart();
ObservableList<PieChart.Data> pieChartData =
    FXCollections.observableArrayList(
        new PieChart.Data("Grapefruit", 13),
        new PieChart.Data("Oranges", 25),
        new PieChart.Data("Plums", 10),
        new PieChart.Data("Pears", 22),
        new PieChart.Data("Apples", 30));
chart.setData(pieChartData);
chart.setTitle("Imported Fruits");
```

Diagrama de líneas

- Representa una serie de puntos conectados por rectas
- Tiene dos ejes, los puntos y las rectas, una leyenda y, opcionalmente, un título
- Puede tener una o más series de datos

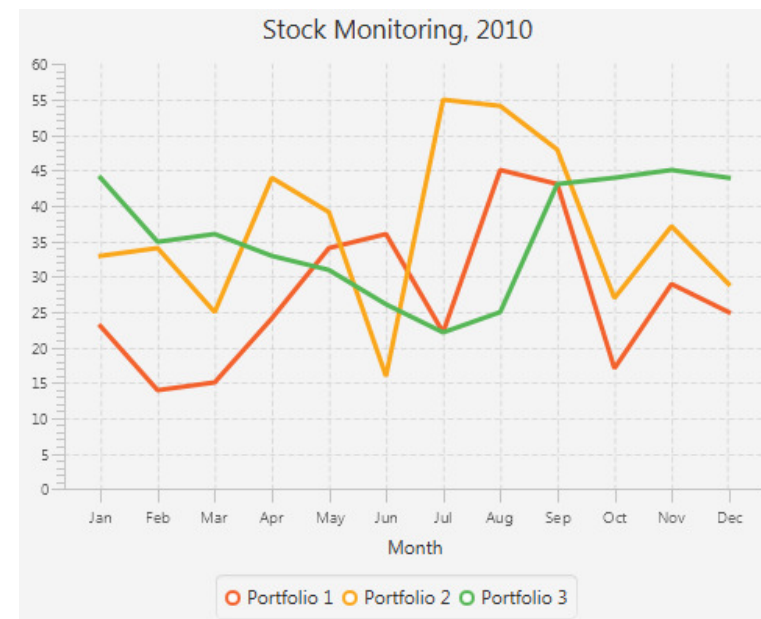
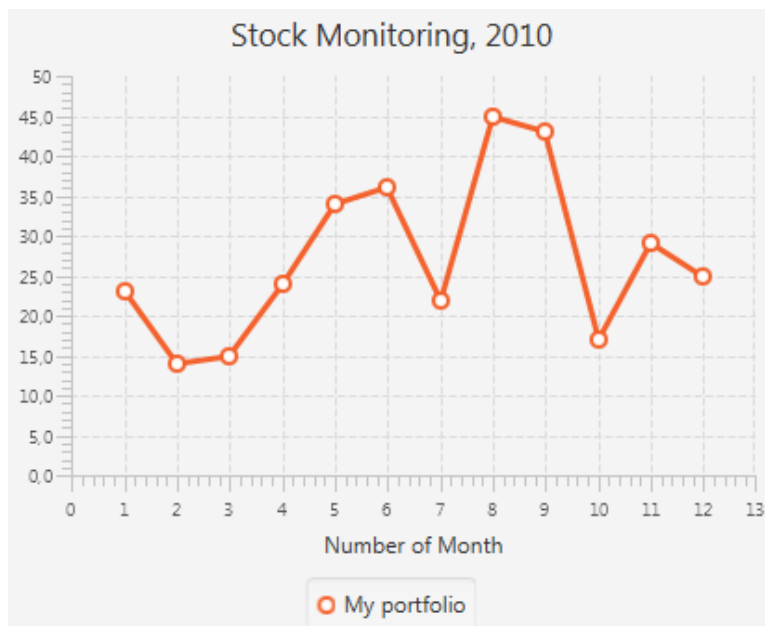


Diagrama de líneas

- Con una sola serie de datos:

```
CategoryAxis xAxis = new CategoryAxis();  
NumberAxis yAxis = new NumberAxis();  
xAxis.setLabel("Month");
```

```
LineChart<String,Number> lineChart =  
    new LineChart<String,Number>(xAxis,yAxis);
```

```
lineChart.setTitle("Stock Monitoring, 2010");
```

```
XYChart.Series series = new XYChart.Series();  
series.setName("My portfolio");
```

```
series.getData().add(new XYChart.Data("Jan", 23));  
series.getData().add(new XYChart.Data("Feb", 14));  
...  
series.getData().add(new XYChart.Data("Nov", 29));  
series.getData().add(new XYChart.Data("Dec", 25));
```

```
Scene scene = new Scene(lineChart,800,600);  
lineChart.getData().add(series);
```



Diagrama de líneas

- Con tres series de datos:

```
XYChart.Series series1 = new XYChart.Series();
series1.setName("Portfolio 1");
series1.getData().add(new XYChart.Data("Jan", 23));
...
series1.getData().add(new XYChart.Data("Dec", 25));
```

```
XYChart.Series series2 = new XYChart.Series();
series2.setName("Portfolio 2");
series2.getData().add(new XYChart.Data("Jan", 33));
...
series2.getData().add(new XYChart.Data("Dec", 29));
```

```
XYChart.Series series3 = new XYChart.Series();
series3.setName("Portfolio 3");
series3.getData().add(new XYChart.Data("Jan", 44));
...
series3.getData().add(new XYChart.Data("Dec", 44));
```

```
Scene scene = new Scene(lineChart, 800, 600);
lineChart.getData().addAll(series1, series2, series3);
```

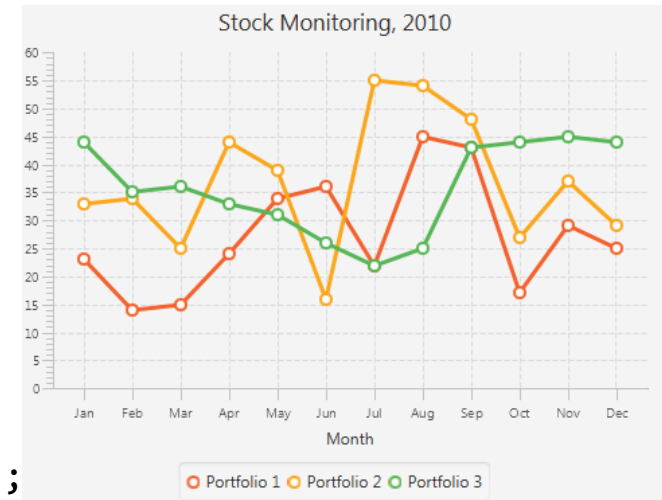


Diagrama de líneas

- Opciones:

- Poner el eje X en la parte superior

```
xAxis.setSide(Side.TOP);
```

- Eliminar las marcas de los puntos de la gráfica

```
lineChart.setCreateSymbols(false);
```

- Utilizar números en el eje X

```
NumberAxis xAxis = new NumberAxis();  
NumberAxis yAxis = new NumberAxis();  
//creating the chart  
LineChart<Number,Number> lineChart =  
    new LineChart<Number,Number>(xAxis,yAxis);
```

```
//defining a series  
XYChart.Series series = new XYChart.Series();  
//populating the series with data  
series.getData().add(new XYChart.Data(1, 23));  
...
```

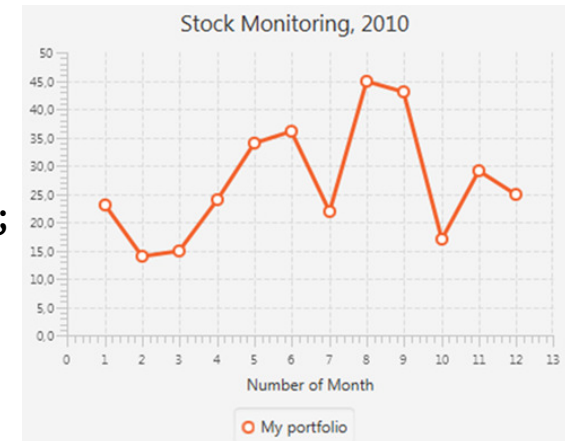


Diagrama de áreas

- Similar al diagrama de líneas pero con el área bajo las líneas rellenada
- Permiten también una o varias series de datos

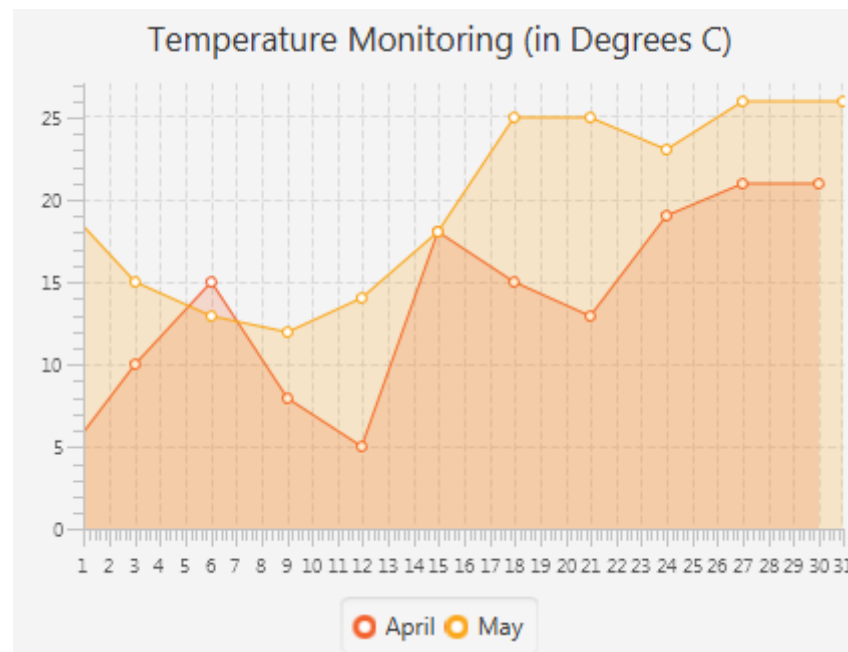


Diagrama de áreas

- Se crea como el diagrama de líneas

```
stage.setTitle("Area Chart Sample");  
NumberAxis xAxis = new NumberAxis(1, 31, 1);  
NumberAxis yAxis = new NumberAxis();  
AreaChart<Number,Number> ac =  
    new AreaChart<>(xAxis,yAxis);  
ac.setTitle("Temperature Monitoring (in Degrees C)");
```

```
XYChart.Series seriesApril= new XYChart.Series();  
seriesApril.setName("April");  
seriesApril.getData().add(new XYChart.Data(1, 4));  
...  
seriesApril.getData().add(new XYChart.Data(30, 21));
```

```
XYChart.Series seriesMay = new XYChart.Series();  
seriesMay.setName("May");  
seriesMay.getData().add(new XYChart.Data(1, 20));  
...  
seriesMay.getData().add(new XYChart.Data(31, 26));
```

```
ac.getData().addAll(seriesApril, seriesMay);
```

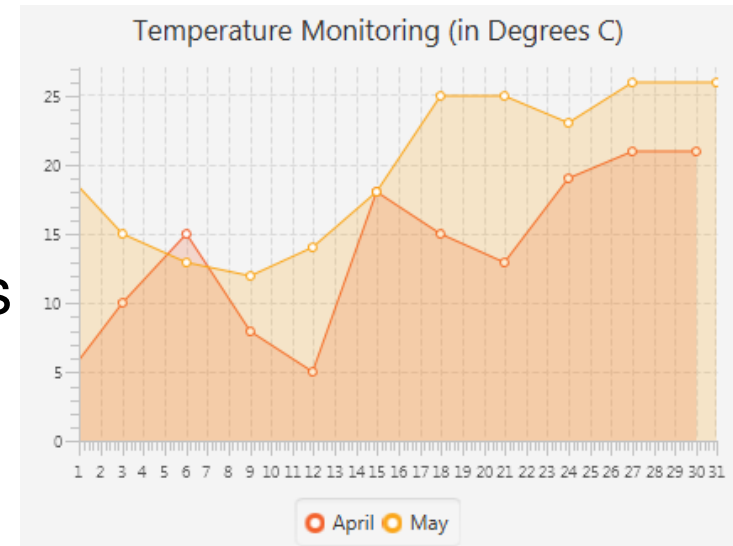


Diagrama de áreas

- Opciones

- En la declaración del eje X

```
NumberAxis xAxis = new NumberAxis(1, 31, 1);
```

se indican el mínimo (1), el máximo (31) y la distancia entre marcas (1)

- También se puede hacer explícitamente

```
xAxis.setLowerBound(1);  
xAxis.setUpperBound(31);  
xAxis.setTickUnit(1);
```

- Para eliminar las marcas pequeñas

```
xAxis.setMinorTickCount(0);
```

- Para eliminar marcas y etiquetas, respectivamente:

```
xAxis.setTickMarkVisible(false);  
xAxis.setTickLabelsVisible(false);
```

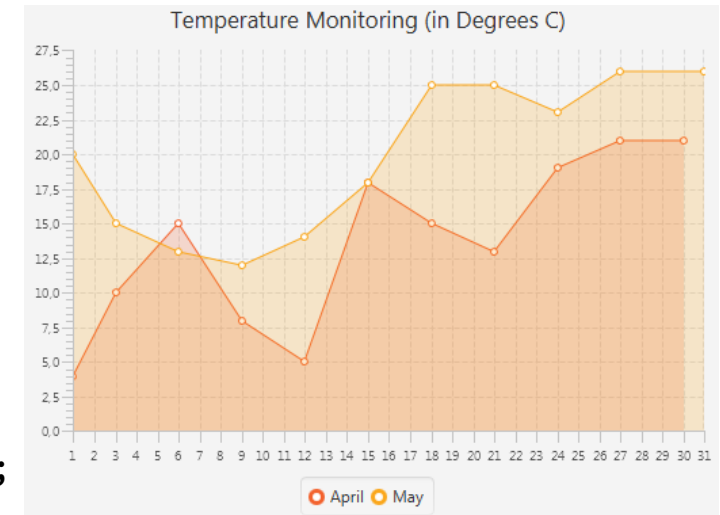


Diagrama de áreas apiladas

- Es un diagrama de áreas donde la segunda serie y sucesivas se visualizan acumulando los valores de las series anteriores
- Utiliza la clase *StackedAreaChart* en lugar de *AreaChart*

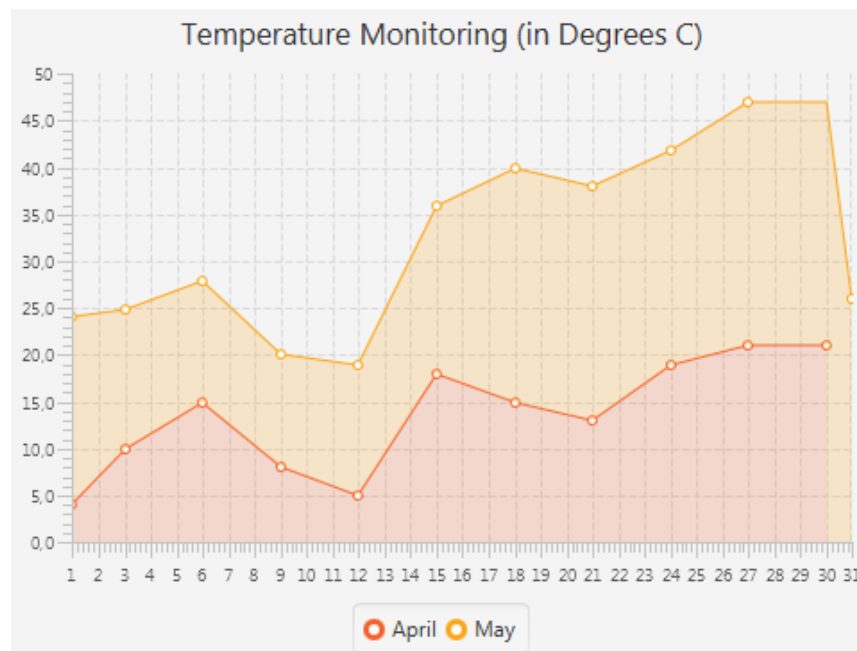


Diagrama de burbujas

- Diagrama bidimensional que dibuja burbujas para los puntos de una serie
- Las burbujas pueden tener un radio distinto

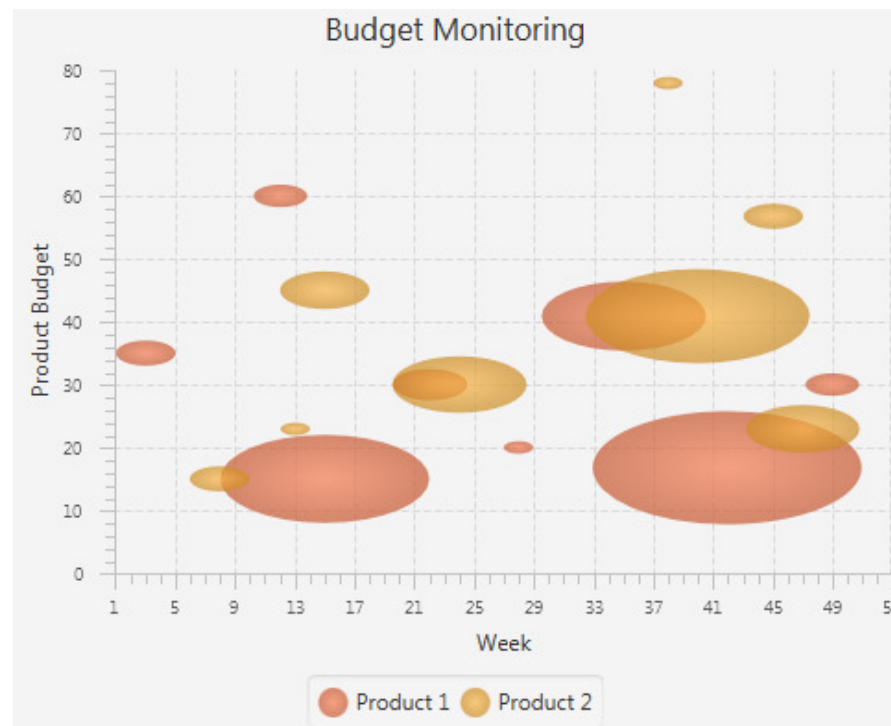


Diagrama de burbujas

- El radio de cada burbuja se especifica como un parámetro más en la definición de los *XYChart.Data()*
- Debe ser de tipo Number

```
NumberAxis xAxis = new NumberAxis(1, 53, 4);
NumberAxis yAxis = new NumberAxis(0, 80, 10);
BubbleChart<Number,Number> blc = new
    BubbleChart<>(xAxis,yAxis);

XYChart.Series series1 = new XYChart.Series();
series1.setName("Product 1");
series1.getData().add(new XYChart.Data(3, 35, 2));
...

XYChart.Series series2 = new XYChart.Series();
series2.setName("Product 2");
series2.getData().add(new XYChart.Data(8, 15, 2));
...

blc.getData().addAll(series1, series2);
```

Diagrama de burbujas

- Opciones:

- Formatear las etiquetas de un eje

```
yAxis.setTickLabelFormatter(new  
    NumberAxis.DefaultFormatter(yAxis,"$ ",null));
```

- Eliminar las líneas del fondo del diagrama

```
blc.setHorizontalGridLinesVisible(false);  
blc.setVerticalGridLinesVisible(false);
```

- Cambiar el color de las etiquetas de los ejes

```
xAxis.setTickLabelFill(Color.CHOCOLATE);  
yAxis.setTickLabelFill(Color.CHOCOLATE);
```

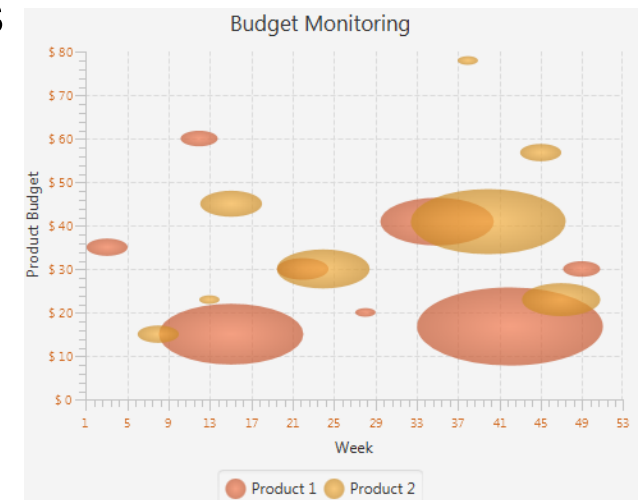
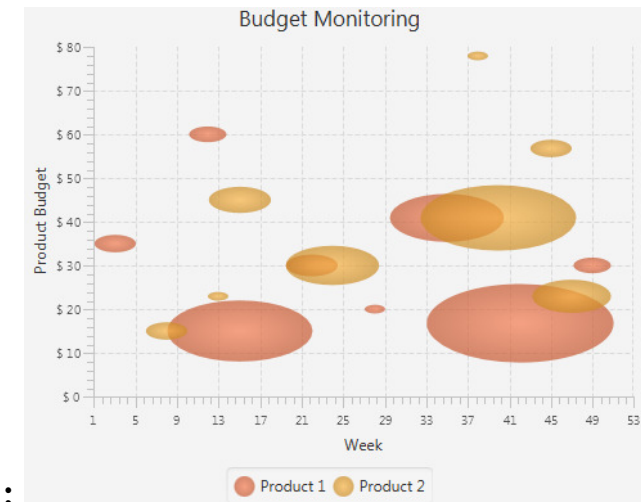


Diagrama de puntos

- Diagrama bidimensional formado por puntos definidos por un par de valores X e Y
- Se crean igual que el resto de diagramas bidimensionales

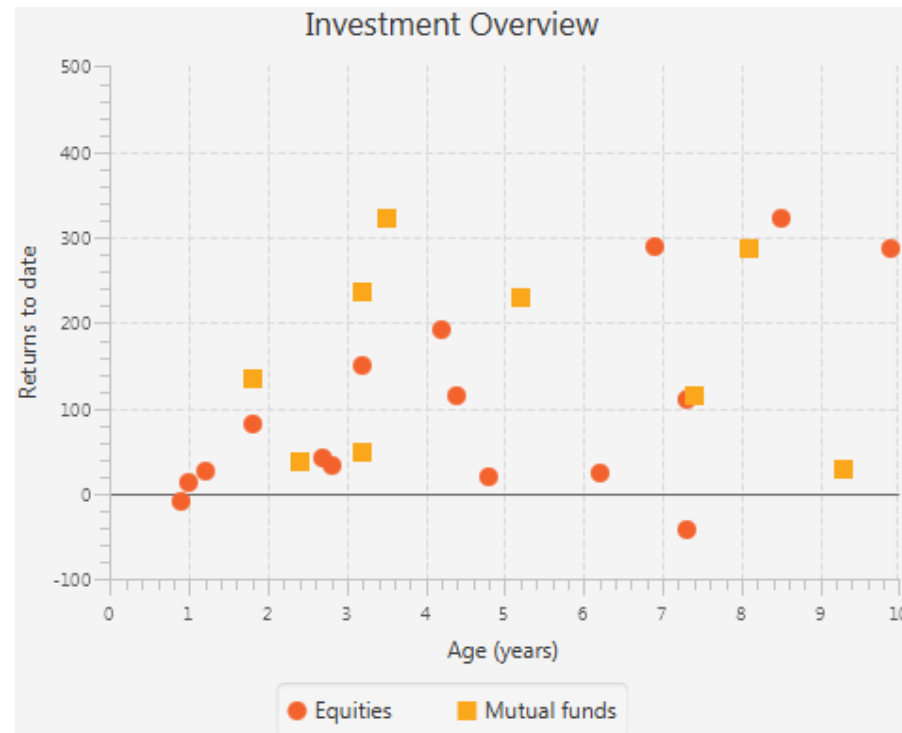


Diagrama de puntos

- Se crea igual que el resto de diagramas bidimensionales

```
NumberAxis xAxis = new NumberAxis(0, 10, 1);  
NumberAxis yAxis = new NumberAxis(-100, 500, 100);  
ScatterChart<Number,Number> sc = new  
    ScatterChart<>(xAxis,yAxis);  
xAxis.setLabel("Age (years)");  
yAxis.setLabel("Returns to date");  
sc.setTitle("Investment Overview");
```

```
XYChart.Series series1 = new XYChart.Series();  
series1.setName("Equities");  
series1.getData().add(new XYChart.Data(4.2, 193.2));  
...
```

```
XYChart.Series series2 = new XYChart.Series();  
series2.setName("Mutual funds");  
series2.getData().add(new XYChart.Data(5.2, 229.2));  
...
```

```
sc.getData().addAll(series1, series2);
```

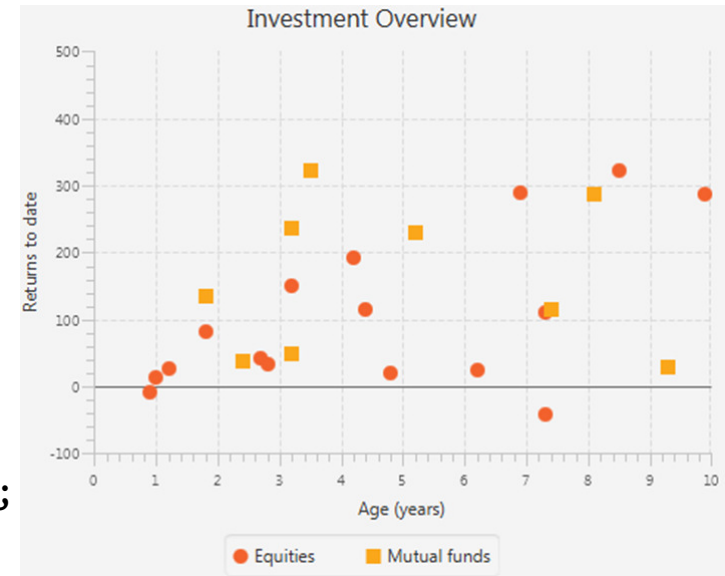


Diagrama de barras

- Es un diagrama bidimensional donde los datos se representan como barras
- Soportan una o varias series

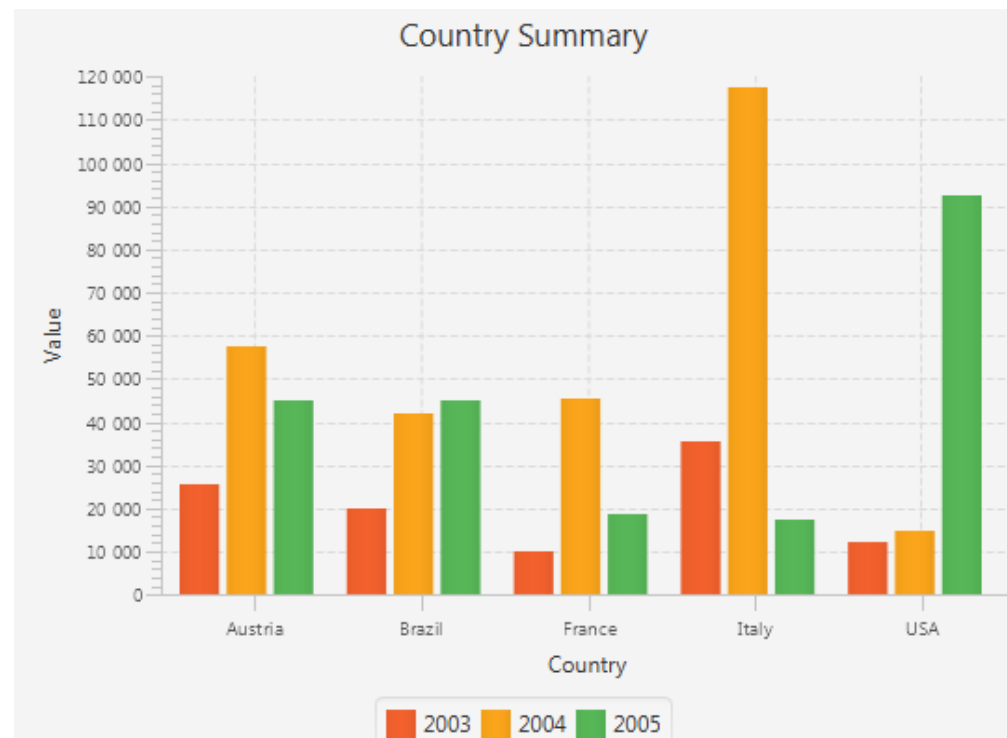


Diagrama de barras

- Código del ejemplo

```
CategoryAxis xAxis = new CategoryAxis();
NumberAxis yAxis = new NumberAxis();
BarChart<String,Number> bc =
    new BarChart<>(xAxis,yAxis);
bc.setTitle("Country Summary");
xAxis.setLabel("Country");
yAxis.setLabel("Value");
```

```
XYChart.Series series1 = new XYChart.Series();
series1.setName("2003");
series1.getData().add(new XYChart.Data("Austria", 25601.34));
...
XYChart.Series series2 = new XYChart.Series();
series2.setName("2004");
series2.getData().add(new XYChart.Data("Austria", 57401.85));
...
XYChart.Series series3 = new XYChart.Series();
series3.setName("2005");
series3.getData().add(new XYChart.Data("Austria", 45000.65));
...
bc.getData().addAll(series1, series2, series3);
```

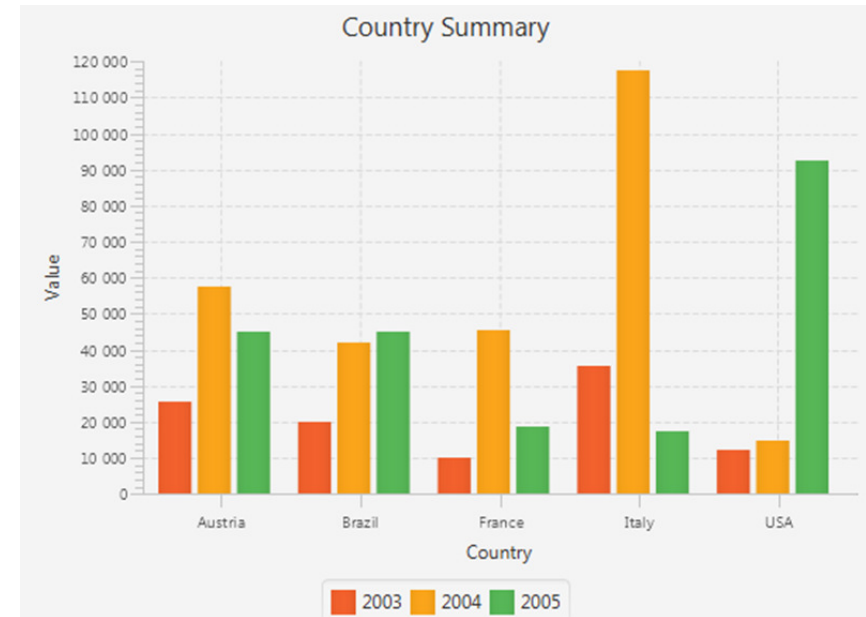
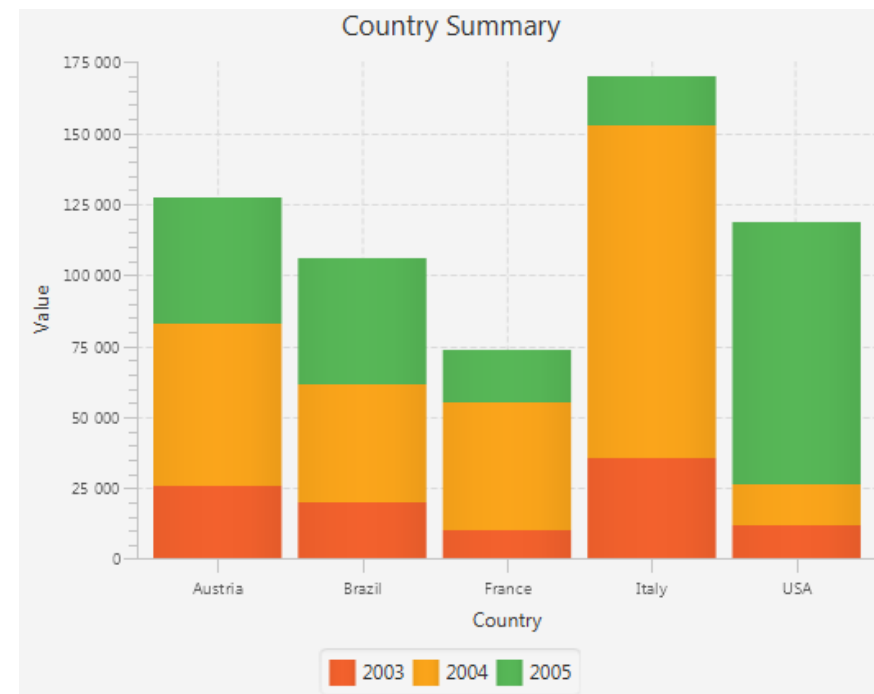
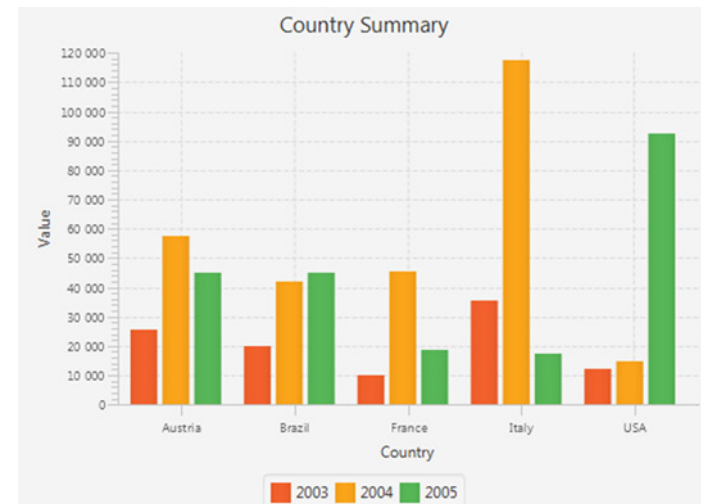


Diagrama de barras

- Opciones:
 - Fijar separación entre barras
`barChart.setBarGap(3);`
 - Fijar separación entre categorías
`barChart.setCategoryGap(20);`
- Diagrama de barras apiladas:
 - En el eje vertical las áreas de las barras muestran valores acumulados
 - En el ejemplo: el valor de 125000 para Austria es el acumulado de 2003, 2004 y 2005
 - Utilizar *StackedBarChart*

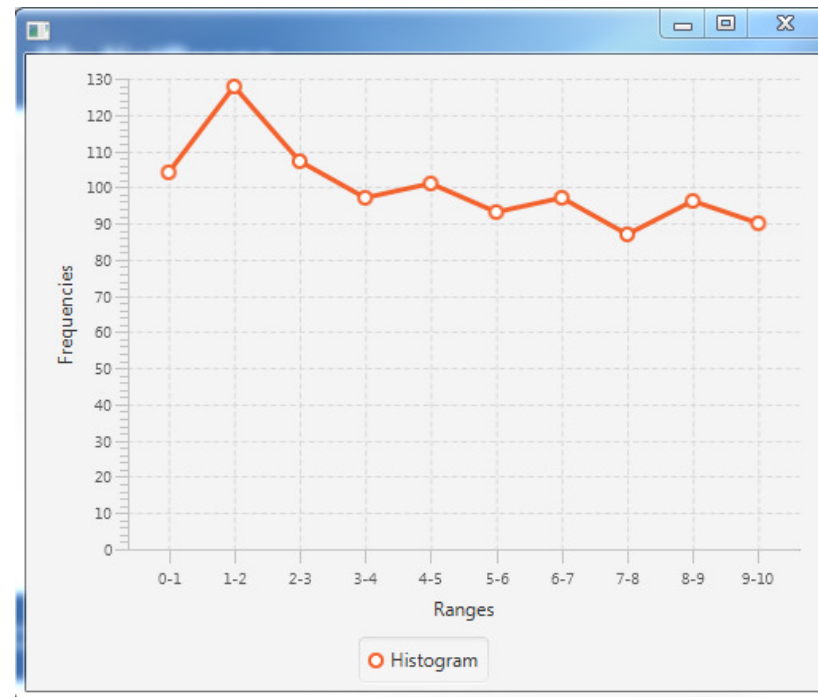


Diagramas en JavaFX

- Se recomienda crearlos desde Scene Builder
- En Scene Builder en los diagramas bidimensionales:
 - El eje X es siempre un *CategoryAxis*
 - El eje Y es un *NumberAxis*
 - Por lo tanto, los datos son *XYChart.Data(String, Number)*
 - Excepto para el *BubbleChart* y el *StackedAreaChart* donde los dos ejes son *NumberAxis*
- Más información sobre diagramas, así como los ejemplos completos, en:
 - <http://docs.oracle.com/javase/8/javafx/user-interface-tutorial/charts.htm>

Ejemplo de diagrama

- Vamos a crear un diagrama con el histograma de un conjunto de números aleatorios
- Como los números son aleatorios el histograma revelará una distribución casi uniforme



Ejemplo de diagrama

- Para crear el diagrama desde Scene Builder basta con elegirlo entre los *Charts* y arrastrarlo
- Para asignar los *fx:id*'s a los ejes seleccionarlos en el área central

@FXML

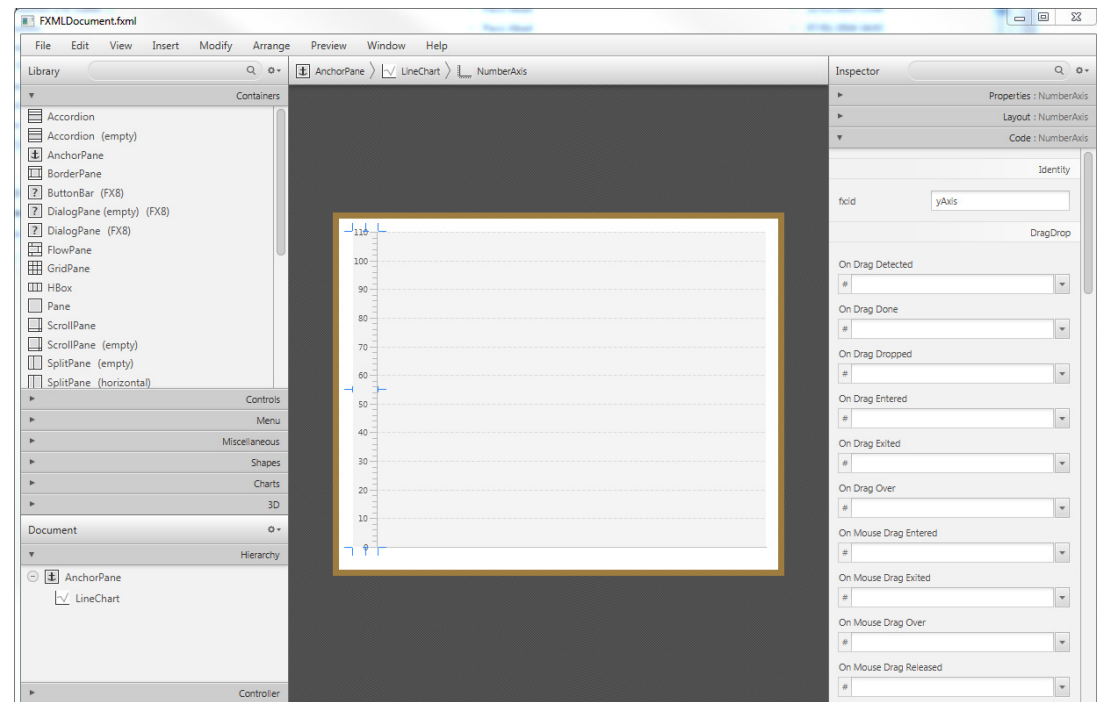
```
private LineChart<String,Number>  
    chart;
```

@FXML

```
private CategoryAxis xAxis;
```

@FXML

```
private NumberAxis yAxis;
```



Ejemplo de diagrama

- Después generar los números aleatorios y obtener el histograma

```
final int NBRACKETS = 10;
int hist[] = new int[NBRACKETS];

for (int i = 0; i < hist.length; i++) {
    hist[i] = 0;
}

for (int j = 0; j < 1000; j++) {
    double value = Math.random() * NBRACKETS;
    hist[(int) value]++;
}
```

Ejemplo de diagrama

- Finalmente generar y añadir los puntos de datos al diagrama

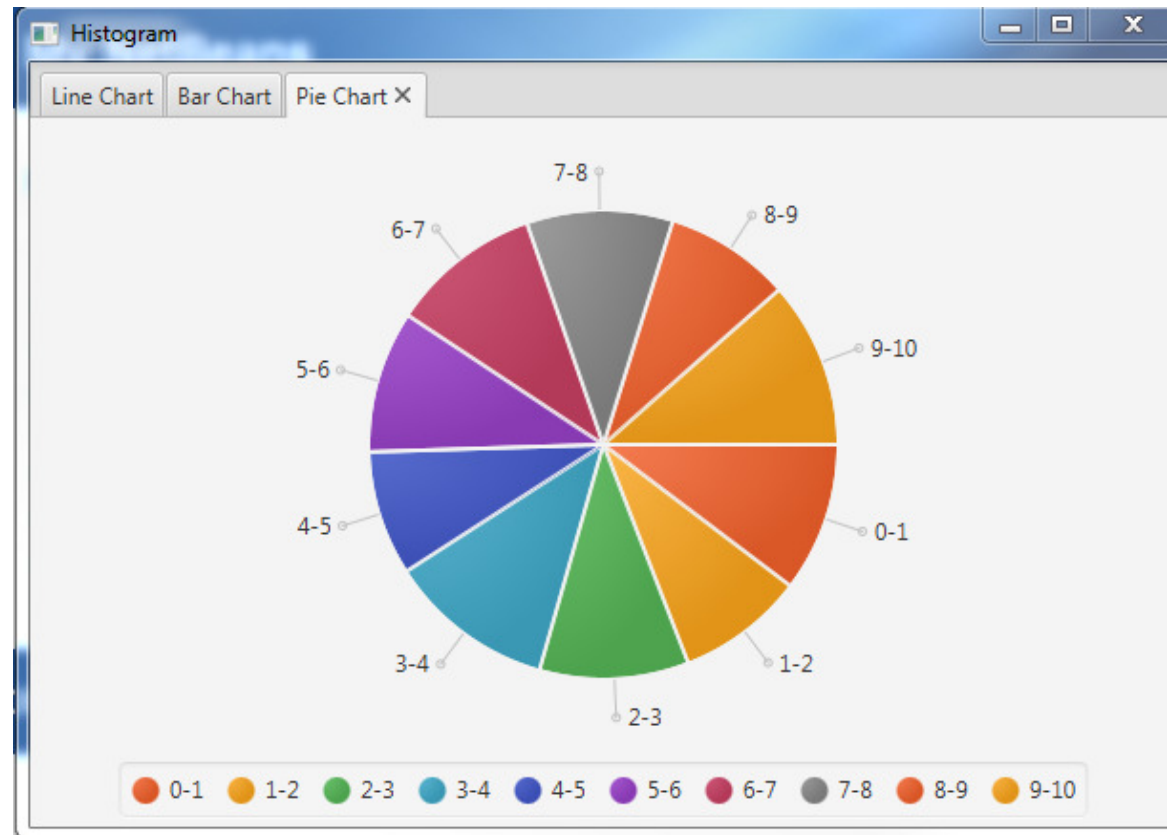
```
xAxis.setLabel("Ranges");
yAxis.setLabel("Frequencies");

XYChart.Series<String,Number> series = new XYChart.Series();
for (int i = 0; i < hist.length; i++)
    series.getData().add(new XYChart.Data<>(i + "-" + (i+1), hist[i]));

series.setName("Histogram");
chart.getData().add(series);
```

Ejercicio

- Dado el código del ejemplo anterior, implementar otros diagramas diferentes



Bibliografía

- <http://docs.oracle.com/javase/8/javafx/user-interface-tutorial/charts.htm>