EJERCICIO 1

### Enlaces útiles

* [Java 8 API](http://docs.oracle.com/javase/8/docs/api/) - Documentación (JavaDoc) de las clases estándar de Java
* [JavaFX 8 API](http://docs.oracle.com/javase/8/javafx/api/) - Documentación de las clases JavaFX
* [ControlsFX API](http://controlsfx.bitbucket.org/) - Documentación para el proyecto [ControlsFX](http://fxexperience.com/controlsfx/), el cual ofrece controles JavaFX adicionales

**Crea el archivo FXML de diseño**

Especifica el nombre del proyecto (ej. **Agenda**) y aprieta *Finish*.

Borra el paquete ***javafxapplication*** y su contenido si ha sido automáticamente creado.

En el ratón hacemos clic derecho en la carpeta *src*, *New | Package*:

* ch.makery.controlador- contendrá la *mayoría* de clases de control (C)

**La clase principal en JavaFX**

Ahora necesitamos crear la **clase javaFX principal**, la cual iniciará nuestra aplicación mediante **Principal** .

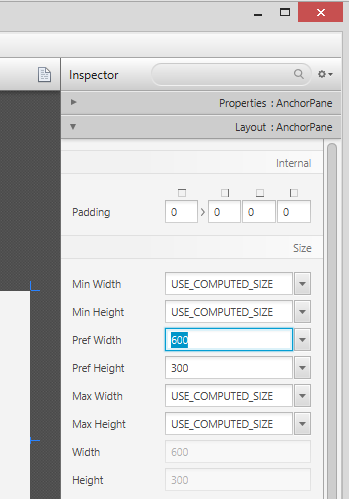
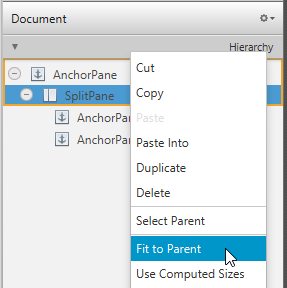
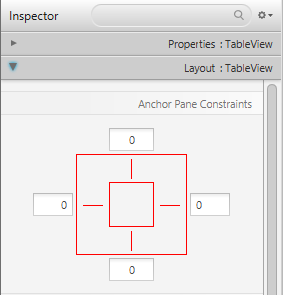
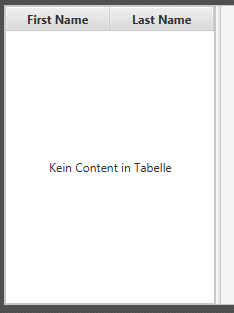
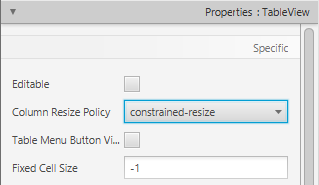
En el ratón hacemos clic derecho en la carpeta *src*, *New | Package*:

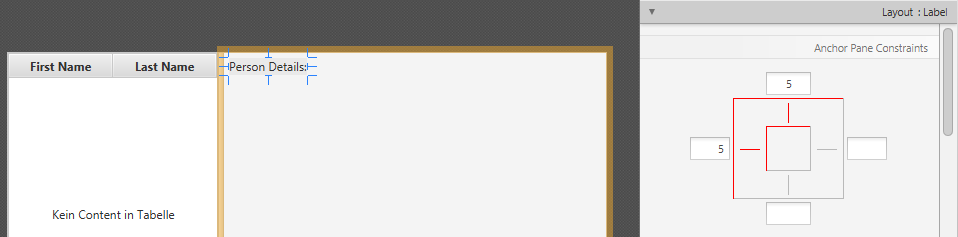
* ch.makery.controlador.Modelo - contendrá las clases del modelo (M)
* ch.makery.controlador.Vista - contendrá las vistas (V)

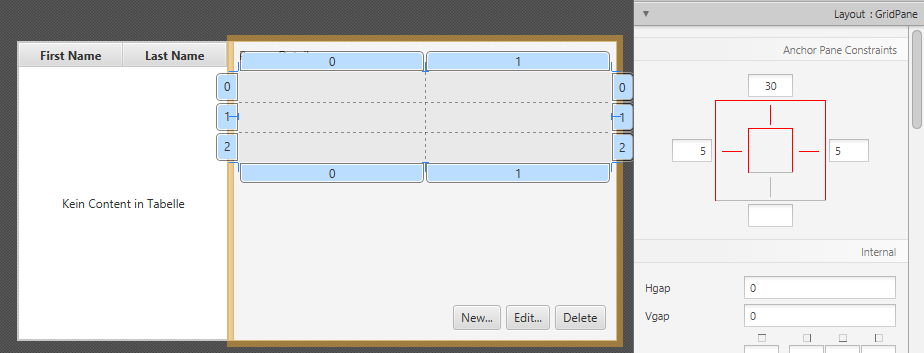
**Nota:** Nuestro paquete dedicado a las vistas contendrá también algunos controladores dedicados exclusivamente a una vista. Les llamaremos **controladores-vista**.

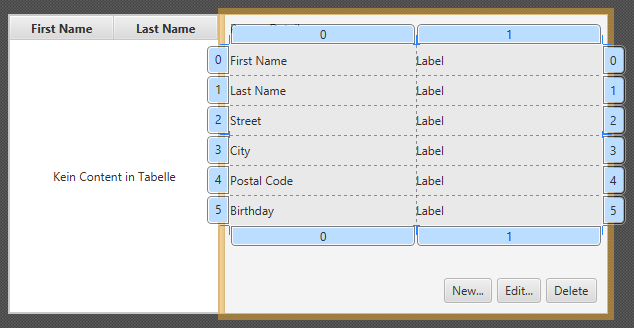
Haz clic derecho el paquete **“vista”** y crea un nuevo archivo FXML (New | Other | FXML | New FXML Document) llamado **VistaPersona**.

Haz clic derecho sobre  **VistaPersona.fxml**  y elige *Open with Scene Builder*. Ahora deberías ver el Scene Builder con un *AnchorPane* (visible en la jerarquía de componentes (herramienta Hierarchy) situada a la izquierda).

1. Selecciona el *AnchorPane* en tu jerarquía y ajusta el tamaño en el apartado Layout (a la derecha):  
   
2. Añade un *SplitPane (Horizontal Flow)* arrastrándolo desde la librería (Library) al área principal de edición, ajústalo al tamaño del AnchorPane. Haz clic derecho sobre el *SplitPane* en la jerarquía y elige*Fit to Parent*.  
   
3. Arrastra un *TableView* (bajo *Controls*) al lado izquierdo del *SplitPane*. Selecciona la TableView (no una sola columna) y establece las siguientes restricciones de apariencia (Layout) para la TableView. Ajustar tabla y columnas y después anclar la tabla. Dentro de un *AnchorPane* siempre se pueden establecer anclajes (anchors) para los cuatro bordes.  
   
4. Ve al menú *Preview | Show Preview in Window* para comprobar si se visualiza correctamente. Intenta cambiar el tamaño de la ventana. La TableView debería ajustar su tamaño al tamaño de la ventana, pues está "anclada" a sus bordes.
5. Cambia el texto de las columnas (bajo Properties) a "First Name" y "Last Name".  
   
6. Selecciona la *TableView* y elige *constrained-resize* para la *Column Resize Policy* (en Properties). Esto asegura que las columnas usarán siempre todo el espacio que tengan disponible.  
   
7. Añade una *Label* al lado derecho del *SplitPane* con el texto "Detalles Personales".

Ajusta su apariencia usando anclajes.  


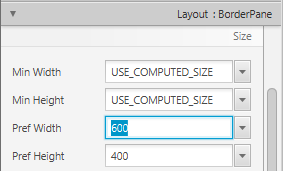
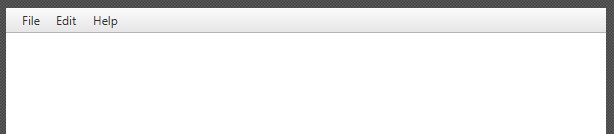
1. Añade un GridPane\* al lado derecho, selecciónalo y ajusta su apariencia usando anclajes (superior, derecho e izquierdo).  
   
2. Añade las siguientes etiquetas (Label) a las celdas del *GridPane* .  
   *Nota: Para añadir una fila al GridPane selecciona un número de fila existente (se volverá de color amarillo), haz clic derecho sobre el número de fila y elige "Add Row".*



1. Añade 3 botones a la parte inferior. Truco: Selecciónalos todos, haz click derecho e invoca *Wrap In | HBox*. Esto los pondrá a los 3 juntos en un HBox. Podrías necesitar establecer un espaciado *spacing* dentro del HBox. Después, establece también anclajes (derecho e inferior) para que se mantengan en el lugar correcto.  
   Button Group
2. Ahora deberías ver algo parecido a lo siguiente. Usa el menú *Preview* para comprobar su comportamiento al cambiar el tamaño de la ventana.  
   

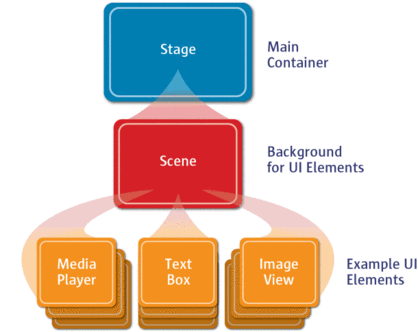
**Crea la aplicación principal**

Necesitamos otro archivo FXML para nuestra vista raíz, la cual contendrá una barra de menús y encapsulará la vista recién creada **VistaPersona.fxml**.

1. Crea otro archivo FXML dentro del paquete **vista** llamado **VistaPrincipal.** Esta vez, elige *BorderPane* como elemento raíz (elimina el AnchorPane).
2. Abre RootLayout.fxml en el Scene Builder.
3. Cambia el tamaño del *BorderPane* con la propiedad *Pref Width* establecida en 600 y*Pref Height* en 400.  
   
4. Añade una *MenuBar* en la ranura superior del *BorderPane*. De momento no vamos a implementar la funcionalidad del menú.  
   

La clase generada (**AplicaciónPrincipal.java**) extiende a la clase Application y contiene dos métodos. Esta es la estructura básica que necesitamos para ejecutar una Aplicación JavaFX. La parte más importante para nosotros es el método start(Stage primaryStage). Este método es invocado automáticamente cuando la aplicación es lanzada desde el método main.

Como puedes ver, el método start(...) tomo un Stage como parámetro. El gráfico siguiente muestra la estructura de cualquier aplicación JavaFX:



**Es como una obra de teatro**: El Stage (escenario) es el contenedor principal, normalmente una ventana con borde y los típicos botones para maximizar, minimizar o cerrar la ventana. Dentro del Stage se puede añadir una Scene (escena), la cual puede cambiarse dinámicamente por otra Scene. Dentro de un Scene se añaden los nodos JavaFX, tales como AnchorPane, TextBox, etc.

AddressApp = Agenda

MainApp = Principal

Address = Controlador

View = Vista

Model = Modelo

**VistaPrincipal.fxml** = RootLayout

**VistaPersona.fxml =** PersonOverview

Abre el archivo **AplicaciónPrincipal.java** y reemplaza todo su código con el código siguiente:

package ch.makery.address;

import java.io.IOException;

import javafx.application.Application;

import javafx.fxml.FXMLLoader;

import javafx.scene.Scene;

import javafx.scene.layout.AnchorPane;

import javafx.scene.layout.BorderPane;

import javafx.stage.Stage;

public class MainApp extends Application {

private Stage primaryStage;

private BorderPane rootLayout;

@Override

public void start(Stage primaryStage) {

this.primaryStage = primaryStage;

this.primaryStage.setTitle("AddressApp");

initRootLayout();

showPersonOverview();

}

/\*\*

\* Initializes the root layout.

\*/

public void initRootLayout() {

try {

// Load root layout from fxml file.

FXMLLoader loader = new FXMLLoader();

loader.setLocation(MainApp.class.getResource("view/RootLayout.fxml"));

rootLayout = (BorderPane) loader.load();

// Show the scene containing the root layout.

Scene scene = new Scene(rootLayout);

primaryStage.setScene(scene);

primaryStage.show();

} catch (IOException e) {

e.printStackTrace();

}

}

/\*\*

\* Shows the person overview inside the root layout.

\*/

public void showPersonOverview() {

try {

// Load person overview.

FXMLLoader loader = new FXMLLoader();

loader.setLocation(MainApp.class.getResource("view/PersonOverview.fxml"));

AnchorPane personOverview = (AnchorPane) loader.load();

// Set person overview into the center of root layout.

rootLayout.setCenter(personOverview);

} catch (IOException e) {

e.printStackTrace();

}

}

/\*\*

\* Returns the main stage.

\* @return

\*/

public Stage getPrimaryStage() {

return primaryStage;

}

public static void main(String[] args) {

launch(args);

}

}

Los diferentes comentarios deben darte pistas sobre lo que hace cada parte del código.

Si ejecutas la aplicación ahora, verás algo parecido a la captura de pantalla incluida al principio de este artículo.

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package Controlador;

import java.io.IOException;

import javafx.application.Application;

// import javafx.event.ActionEvent;

// import javafx.event.EventHandler;

// import javafx.scene.control.Button;

// import javafx.scene.layout.StackPane;

import javafx.fxml.FXMLLoader;

import javafx.scene.Scene;

import javafx.scene.layout.AnchorPane;

import javafx.scene.layout.BorderPane;

import javafx.stage.Stage;

/\*\*

\*

\* @author Joseluis

\*/

public class AplicacionPrincipal extends Application {

private Stage primaryStage;

private BorderPane rootLayout;

@Override

public void start(Stage primaryStage) {

this.primaryStage = primaryStage;

this.primaryStage.setTitle("Agenda");

initVistaPrincipal();

showVistaPersona();

}

/\*\*

\* Initializes the root layout.

\*/

public void initVistaPrincipal() {

try {

// Load root layout from fxml file.

FXMLLoader loader = new FXMLLoader();

loader.setLocation(AplicacionPrincipal.class.getResource("Vista/VistaPrincipal.fxml");

rootLayout = (BorderPane) loader.load();

// Mostrar el escenacio con la VistaPrincipalt.

Scene scene = new Scene(rootLayout);

primaryStage.setScene(scene);

primaryStage.show();

} catch (IOException e) {

e.printStackTrace();

}

}

/\*\*

\* Mostrar la VistaPersona dentro de la vistaPrincipal.

\*/

public void showVistaPersona() {

try {

// Cargar la VistaPersona.

FXMLLoader loader = new FXMLLoader();

loader.setLocation(AplicacionPrincipal.class.getResource("Vista/VistaPersona.fxml"));

AnchorPane personOverview = (AnchorPane) loader.load();

// Meter VistaPersona dentro de VistaPrincipal

rootLayout.setCenter(personOverview);

} catch (IOException e) {

e.printStackTrace();

}

}

/\*\*

\* Returns the main stage.

\* @return

\*/

public Stage getPrimaryStage() {

return primaryStage;

}

public static void main(String[] args) {

launch(args);

}

}