

# TALLER HILOS PROGRAMACION III

Orlando Vianchá Tristancho, Johann Sebastián Joya.

UNIVERSIDAD PEDAGOGICA Y TECNOLOGICA DE COLOMBIA

[orlando.viancha@uptc.edu.co](mailto:orlando.viancha@uptc.edu.co), [johann.joya@uptc.edu.co](mailto:johann.joya@uptc.edu.co)

## I. DESCRIPCION.

Este documento se basa el desarrollo del aplicativo que permite visualizar los resultados de la ruta más óptima con base en el número de escalas, distancia y costo, Primero empezamos con la creación del repositorio en el cual nos permitió tener un trabajo más colaborativo y ya con la ayuda de este repositorio cada uno de los integrantes procedió con la implementación de cada una de las clases a utilizar. Se puede evidenciar el procesamiento múltiple de cada Thread que estará conectado para ir procesando los datos que contiene cada uno de los formatos JSON, en este caso 5 formatos JSON, En dicho formato se especifica las rutas aéreas entre el origen y el final, gracias a todas estas herramientas nombradas porque fueron necesarias para la construcción y la funcionalidad del sistema.

## II. LINK REPOSITORIO.

<https://github.com/OrlandoViancha/Taller-Hilos>

## III. DISEÑO DE LA INTERFAZ GRAFICA EN MOCKUPS.

### A. VENTANA PRINCIPAL.

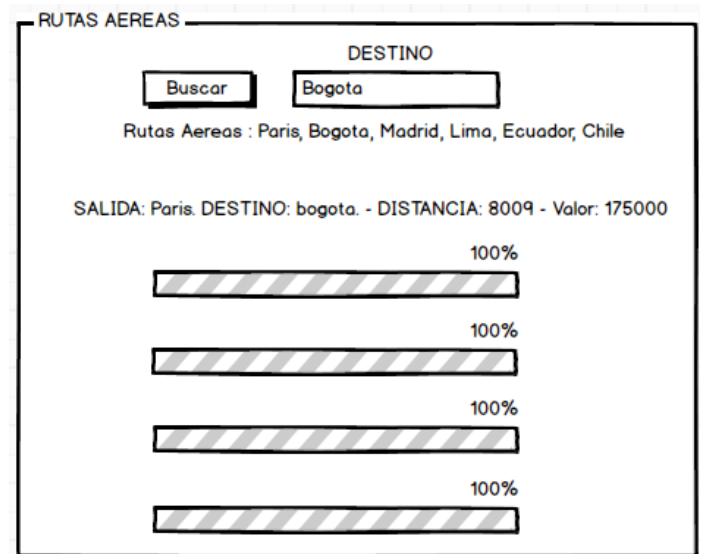
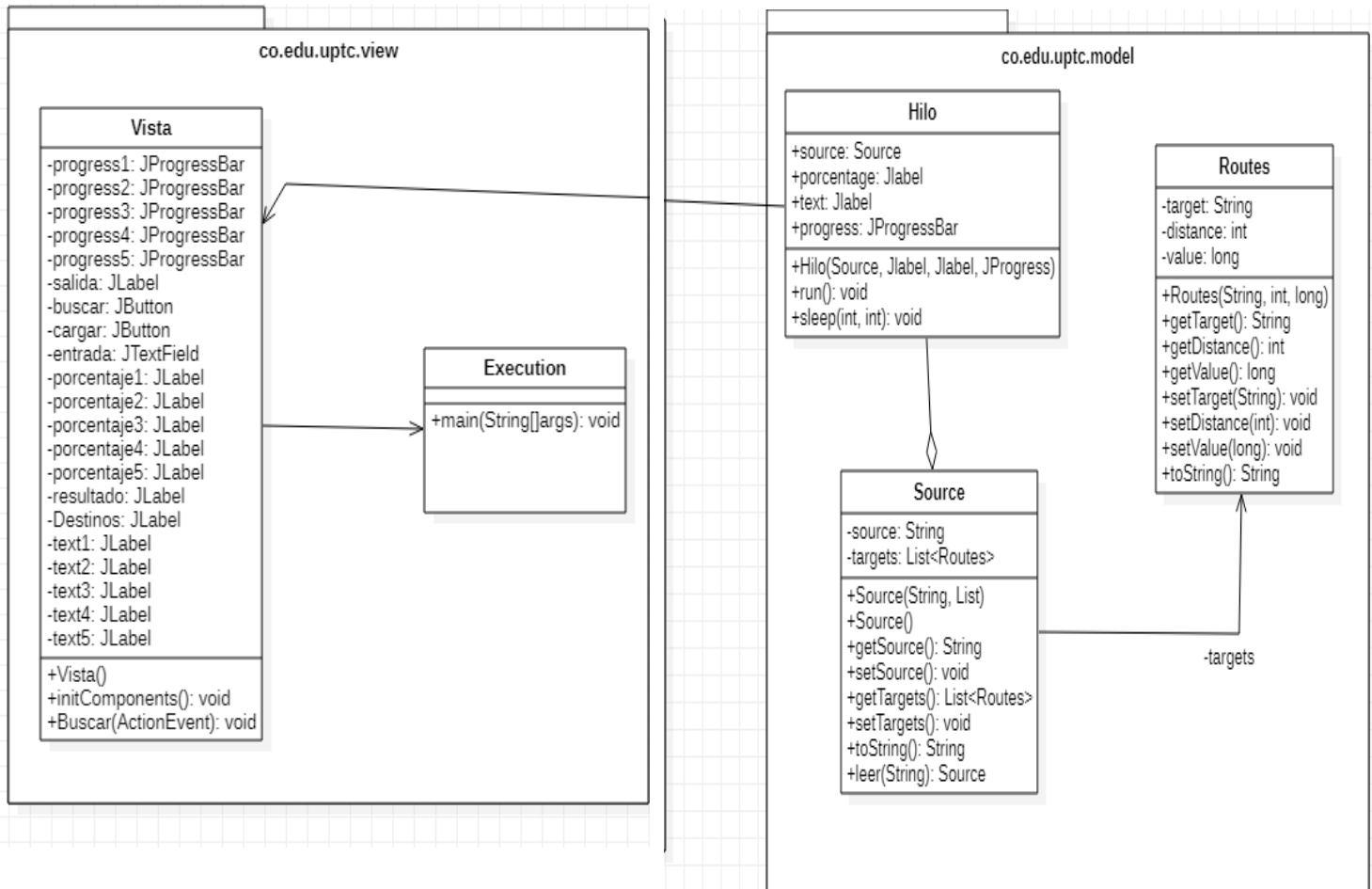


Fig. 1. Diseño en mockup de la ventana principal del aplicativo.

En la fig. 1. nos muestra un diseño sencillo de la ventana principal del aplicativo esta tiene la función agregar el destino y permite validar los datos de las rutas aéreas que hay dentro del formato JSON e identifica si esos datos son valido o no.

### III. DIAGRAMA DE CLASES



### IV. DESCRIPCION DEL PAQUETE MODELO CON SUS RESPECTIVAS CLASES.

- Implementación de la clase **Routes** con sus respectivos atributos y métodos a utilizar.

#### **Routes (string, int, long):**

Método (Constructor) que recibe como parámetro los valores de las variables que están en la clase Routes que son TARGET de tipo string , DISTANCIA de tipo int Y VALOR de tipo long y se procede a inicializar los valores de cada variable asignando el valor que tienen los parámetros según corresponda.

#### **Métodos set ();**

**Void setTarget (string);**

**Void setDistancia (int);**

**Void setValue (long):**

Estos métodos set cambian el valor de la variable por el valor que está en el parámetro que recibe.

#### **Métodos get ();**

**string getTarget ();**

**int getDistancia();**

**long getValue();**

Estos métodos get retornan un valor de tipo (string, int, double, long etc.) correspondiente al del objeto.

### **string toString();**

Este método me devuelve en cadena la información de un objeto (Routes) con cada una de sus variables (target, distance, value).

- Implementación de la clase **Source** con sus respectivos atributos y métodos a utilizar.

En esta clase Source implementamos dos atributos, source de tipo de string y una colección llamada targets donde almacenara la clase Routes.

### **Métodos set ();**

**Void setSource (string);**  
**Void setTargets (string);**

Estos métodos set cambian el valor de la variable por el valor que está en el parámetro que recibe.

### **Método get ();** **string getSource();**

**List<Routes> getTargets ();**  
Método que te retorna la colección.

**string toString();**  
Este método me devuelve en cadena la información de un objeto (Source) con cada una de sus variables.

**Source leer(String);**  
Este método de tipo Source te permite interactuar con todos los archivos json presentes en el proyecto.

- Implementación de la clase **Hilo** con sus respectivos atributos y métodos a utilizar.

### **Hilo (Source, Jlabel, Jlabel, Jprogress):**

Constructor de la clase hilo que recibe como parámetro los valores de las variables que están en la clase Hilo, procede a inicializar los valores de cada variable asignando el valor que tienen los parámetros según corresponda.

### **Void run ():**

Método run de tipo void donde tiene como fin hacer la ejecución de los hilos.

### **Void sleep (int, int):**

Este método se encarga de decirle al hilo que duerma unos milisegundos.