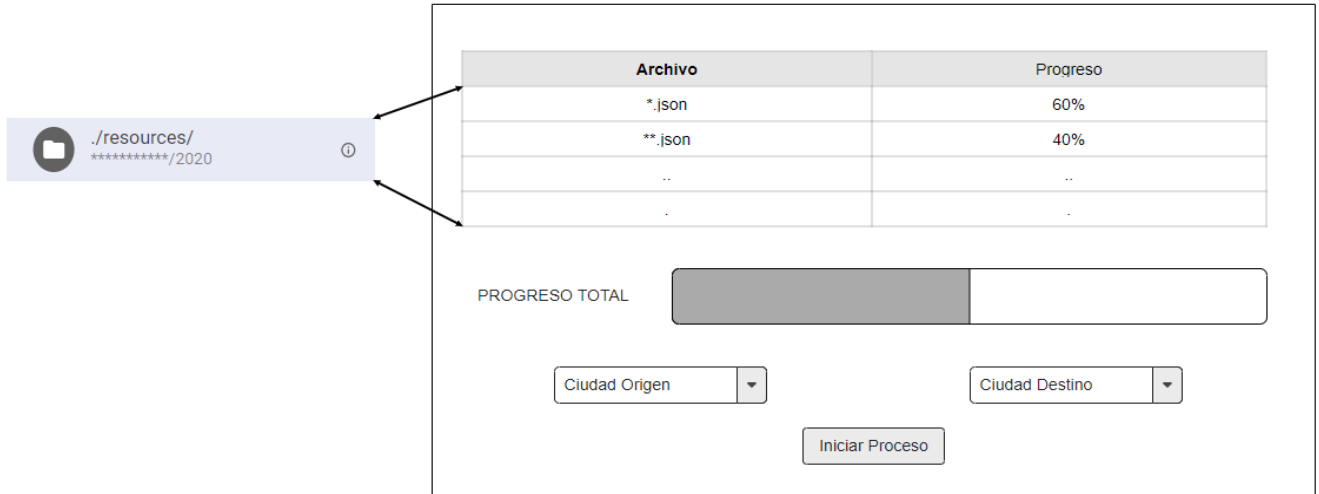


1. MOCKUPS



Archivo	Progreso
*.json	60%
**json	40%
..	..
.	.

PROGRESO TOTAL

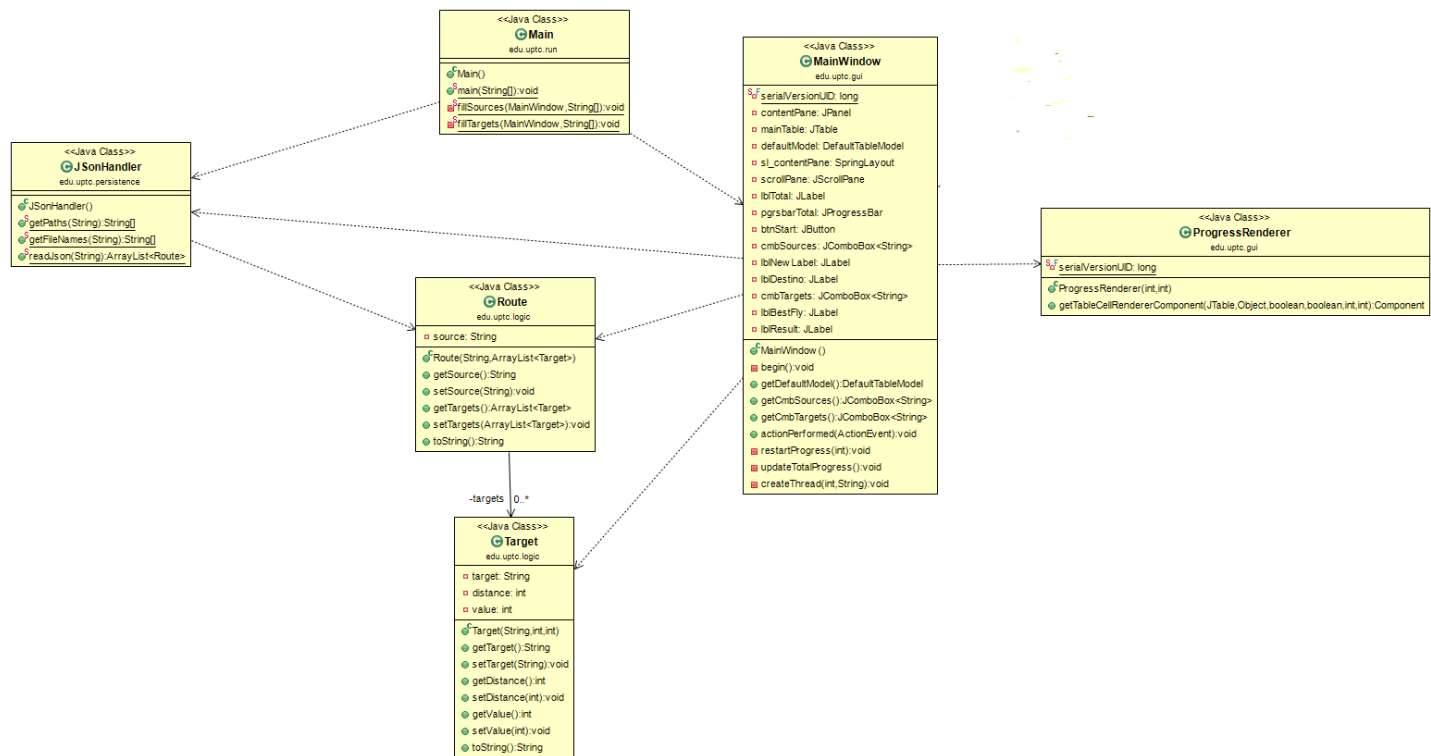
Ciudad Origen

Ciudad Destino

Iniciar Proceso

Por el medio del cual se define la ruta en la que están almacenados los archivos json que serán identificados por el software.

2. UML



- Main: Clase de ejecución la cual se encarga de un llenado preliminar de los posibles orígenes y destinos del software
- JSonHandler: Clase que contiene métodos estáticos capaces de leer los nombres, rutas, y contenido de los archivos JSON de la ruta especificada.
- Route: Clase de definición para el objeto de rutas el cual contiene como atributos el nombre del origen de la misma y los destinos posibles en una colección.
- Target: Clase que contiene los atributos de un destino dentro de una ruta: nombre del destino, distancia y valor.
- ProgressRenderer: Clase que hereda de JProgressBar, con la cual se dará el renderizado a las celdas de la tabla de MainWindow
- MainWindow: Clase principal que contiene toda la interfaz del programa y hace un llamado a los métodos de administración de archivos JSON.
 - CreateThread: El método utiliza una aproximación que crea un hilo anónimo no sincronizado por cada archivo almacenado en la carpeta de recursos, una vez iniciado el proceso, aleatoriamente generará retrasos para evitar que la

ejecución termine demasiado rápido. Actualizando en pantalla el vuelo con el menor valor encontrado entre la ruta establecida por el usuario.