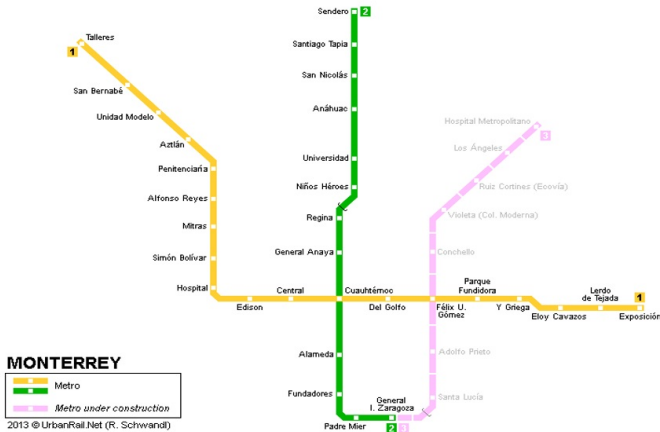


PRÁCTICA METRO MONTERREY

Genevieve CIRERA,
Jaime LABIAGA,
Juan Francisco SALAMANCA CARMONA

3 de junio 2014

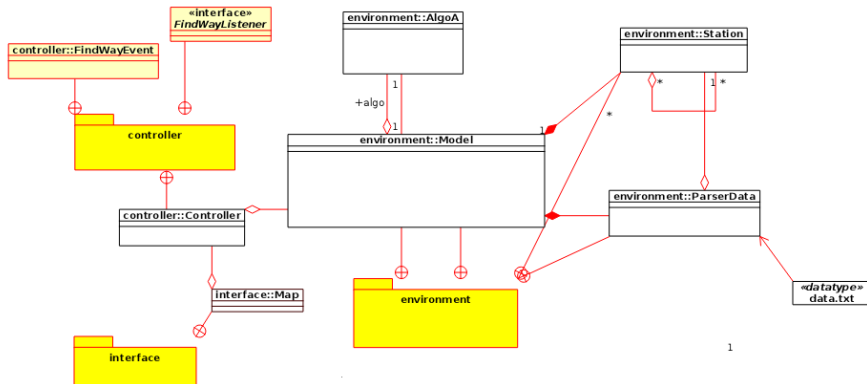
Objetivo



MEDIOS UTILIZADOS

- Algoritmo A*
- Java
- Interfaz : Swing
- Datos : .txt

Estructura : MVC



Recogida de datos

- Declaración de las estaciones

```
*Stations
```

```
Talleres:San Bernabé:Unidad Modelo:Aztlán:Penitenciaria
```

```
$
```

- Distancias

```
%Talleres
```

```
0:690:1680:3000:4080:4800:5810:6500:7200:8090:8490:8980:9660:10380:10990:
```

- Vínculos

```
;10:12:26:27
```

Environment

Dividida en cinco clases

- AlgoA
- Main
- Station
- ParserData
- Model

ENVIRONMENT.Model

```
public void ok(String origin, String destination) {  
    // Encontrar las estaciones que tienen estos nombres  
    clearAllStation();  
    Station si = null, sf = null;  
    for (int i = 0; i < stations.length; i++) {  
        if (stations[i].getName().equals(origin)) {  
            si = stations[i];  
        }  
        if (stations[i].getName().equals(destination)) {  
            sf = stations[i];  
        }  
        if (sf != null && si != null)  
            break;  
    }  
    if (sf == null || si == null) {  
        System.out.println("Una de las dos estaciones no existen");  
    }  
    this.way = algo.findWay(si, sf);  
    fireWayChanged();// Draw way  
}
```

ENVIRONMENT.Find

Llamado por el Model

```
public ArrayList<Station> findWay(Station si, Station sf)
```

Llamado recursivamente

```
private void find(Station padre)
```

Criterio stop

```
if (!padre1.equals(sf)) {  
    // Llamar de nuevo find  
    find(padre1);  
} else { // Si el último nodo de la lista cerrada es el destino  
    this.way.addAll(padre1.getWayToHere());  
    this.way.add(sf);  
}
```


ENVIRONMENT. AlgoA

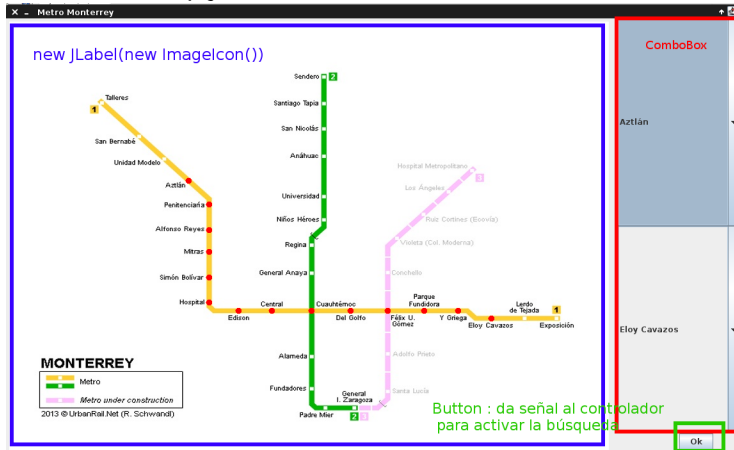
```
public void almacenar(ArrayList<Station> s) {
    for (int i = 0; i < s.size(); i++) {
        if (!this.openedList.contains(s.get(i)) && !this.closedList.contains(s.get(i))) {
            this.openedList.add(s.get(i));
        }
    }
}

public void reordenar(ArrayList<Station> stationClone) {
    if (stationClone.size() > 0) {
        for (int i = 0; i < stationClone.size(); i++) {
            int j = findMin(stationClone);
            this.openedList.add(stationClone.get(j));
            stationClone.remove(j);
            reordenar(stationClone);
        }
    }
}

public int findMin(ArrayList<Station> s) {
    int j = 0;
    int min = s.get(0).getF();
    for (int i = 1; i < s.size(); i++) {
        if (min > s.get(i).getF()) {
            min = s.get(i).getF();
            j = i;
        }
    }
    return j;
}
```

Interfaz gráfica

En la clase Map.java



Problemas encontrados

- Hacer
- Tomar distancias aéreas
- Las resources en la exportación del proyecto
- Organización en el grupo

Demo