**Informe Trabajo Práctico**

**Programación III**

**MapViewer**

<https://gitlab.com/ElMaxe10/ungsp320162c_tp2.git>



Integrantes:

* Lucas Vargas ([lucasjv92@gmail.com](mailto:lucasjv92@gmail.com) 37369589/2014)
* Maximiliano Lucero ([mlucerocorrea@outlook.com](mailto:mlucerocorrea@outlook.com) 38842392/2014)

Profesores:

* Patricia Bagnes - *pbagnes@ungs.edu.ar*
* Javier Marenco - *jmarenco@ungs.edu.ar*
* Daniel Rondelli

## Explicación de Paquetes y Clases

Para el desarrollo del trabajo práctico se decidió crear 5 paquetes en el proyecto de java: **controller, dataAccess, domain, util y view.**

### Paquete controller

En este paquete, se encuentran las clases encargadas de manejar la lógica del mapa.

**controller.Algoritmos.java:** clase encargada de el algoritmo AGM.

**controller.AlgoritmosTest.java:** clase Test de Algoritmos.java

**controller.Arista.java:** crea una arista a partir de 2 coordenadas.

**controller.Cluster.java:** clase encargada de generar la cantidad de cluster que se solicita..

**controller.Coordinates.java:** clase encargada de hacer una lista de coordenadas y cargarla desde un archivo .json.

**controller.CoordinatesTest.java:** clase Test de Coordinates.java

**controller.FicheroCoordenadas.java:** Se encarga de tener la referencia del archivo cargado y sus coordenadas en memoria

**controller.GrafoCoordinates.java:** Clase que implementa un grafo, usando las coordenadas como vertices.

**controller.GrafoCoordinatesTest.java:** clase Test de GrafoCoordinates.java

**controller.MapController.java:** Clase encargada de relacionar las llamadas de la vista, con la logica del programa.

**controller.NeighborsCoordinate.java:** Encargada de contener todos los vecinos de una corrdenada especifica.

**controller.NeighborsCoordinateTest.java:** clase Test de NeigthCoordinate.java

### 

### Paquete dataAccess

En este paquete, se encuentran las clases encargadas de acceder a los datos.

**dataAcces.AristaDao.java:** accede a la lista de aristas y grabarlas.

**dataAcces.CoordinateDao.java:** accede a la lista de coordenadas de las instancias.

### Paquete domain

En este paquete, se encuentran las clases encargadas de relacionar la vista con los datos almacenados.

**domain.AristaPojo.java:** Clase que relaciona la persistencia y lectura de datos de las aristas.

**domain.Coordinate.java:** Clase que re implementa la clase Coordinate de JmapViewer, y que permite relacionar la logica, la vista y la persistencia

### Paquete util

En este paquete, se encuentran las clases encargadas de usar funciones utiles para el proyecto

**util.DistanceCalculator.java:** Calcula la distancia que hay entre 2 coordenadas, en especial en Kilometros.

**util.FileChecker.java:** Permite saber si un archivo existe en disco o no.

**util.OfflineOsmTileSource.java:** Implementa la posiblidad de ver los mapas de forma offline.

**util.Proxy.java:** Permite conectar la aplicacion a internet mediante un proxy.

**util.TileDownloader.java:** Permite descargar las cuadriculas del mapa para luego usarlo de forma offline.

### 

### 

### Paquete View

En este paquete, se encuentran las clases encargadas de la interfaz de la aplicación.

**view.Console.java:** es la interfaz por consola para probar la aplicación.

**view.MainForm.java:** clase encargada de iniciar la aplicación grafica.

**view.OpenFilesForm.java:** Clase que se encarga de mostrar por pantalla un selector de archivos..

## Diagrama de Clases

