



## EJERCICIO 1 – TIPO BASE (1 pto)

Se quiere realizar un estudio sobre los terremotos que se han producido en Turquía en los últimos años. Para ello se ha accedido a los datos registrados y se han almacenado en el fichero earthquake-utf8.csv, situado en la carpeta data del proyecto que se le proporciona. El fichero tiene la siguiente estructura:

```
fecha,lat,long,ciudad,richter
2017.07.31,37.42,31.37,konya,3.5
2017.07.31,36.9,27.56,mugla,3.3
2017.07.31,37,27.62,mugla,3.8
2017.07.30,37.39,31.37,konya,3.5
2017.07.30,37.01,27.6,mugla,4.5
...
```

Implemente la clase **Registro** con la siguiente descripción (puede ayudarse del diagrama de clases de la Figura 1).

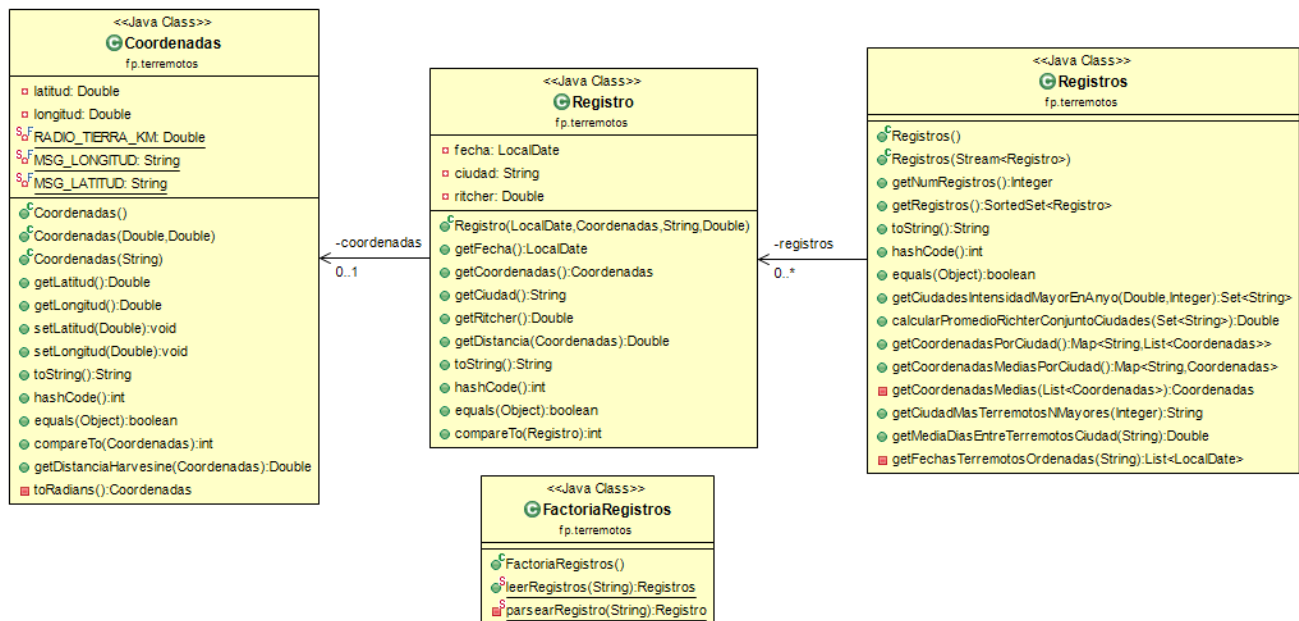


Figura 1. Diagrama de clases

### Propiedades:

- fecha, de tipo LocalDate. Consultable. Fecha en la que se produce el terremoto.
- coordenadas, de tipo Coordenadas. Consultable. Coordenadas terrestres del epicentro del terremoto.
- ciudad, de tipo String. Consultable. Nombre de la ciudad más cercana al epicentro del terremoto.
- richter, de tipo Double. Consultable. Magnitud en la escala de Richter del terremoto.

**Constructores:** un solo constructor que tiene como parámetros las propiedades básicas del tipo, y en el mismo orden en el que se han definido en la descripción de las propiedades anteriores.

**Criterio de igualdad:** dos registros son iguales si la fecha y las coordenadas del epicentro son iguales.

**Criterio de ordenación:** por fecha, y a igualdad de fecha, por coordenadas.

Representación como cadena: el valor de todas las propiedades, separados por coma.

Restricciones:

- La magnitud del terremoto debe estar en el intervalo [1.5, 12].
- La fecha debe ser anterior o igual al día de hoy.

Otras operaciones:

- **Double getDistancia(Coordenadas c)**. Devuelve la distancia terrestre entre las coordenadas dadas como parámetro y las coordenadas del epicentro del terremoto.

## **EJERCICIO 2 – TIPO CONTENEDOR (8 puntos, de los cuales 7,5 son las operaciones)**

Implemente el tipo contenedor **Registros**, con la siguiente descripción:

Propiedades:

- registros, de tipo List<Registro>. Consultable. Lista con registros de terremotos.
- número de registros, de tipo Integer. Consultable. Número de registros de terremotos disponibles en la clase contenedora.

Constructores:

- Un constructor sin parámetros, que crea un objeto de tipo Registros, sin datos de terremotos.
- Un constructor que tiene como parámetro un Stream<Registro> y crea un objeto de tipo Registros a partir del Stream.

Criterio de igualdad: dos objetos de tipo Registros son iguales si sus listas de registros son iguales.

Representación como cadena: la representación como cadena de la lista de registros.

Otras operaciones:

- a) **Set<String> getCiudadesIntensidadMayorEnAnyo(Double umbralRichter, Integer anyo)**. Devuelve un conjunto con el nombre de las ciudades en las que se han producido terremotos en el año dado como parámetro y con una intensidad mayor al umbral dado como parámetro. **(1 Pto)**
- b) **Double calcularPromedioRichterConjuntoCiudades(Set<String> ciudades)**. Devuelve el promedio de la magnitud en la escala Richter de los terremotos producidos en las ciudades incluidas en el conjunto que se pasa como parámetro. Si el promedio no se puede calcular, se elevará *NoSuchElementException*. **(1 Pto)**
- c) **Map<String, List<Coordenadas>> getCoordenadasPorCiudad()**. Devuelve un Map en el que las claves son los nombres de ciudades y los valores son listas de las coordenadas de los epicentros de los terremotos que se han producido en esa ciudad. **(1 Pto)**
- d) **Map<String, Coordenadas> getCoordenadasMediasPorCiudad()**. Devuelve un Map en el que las claves son los nombres de las ciudades y los valores son objetos de tipo Coordenadas que tienen la latitud y longitud media de los terremotos de esa ciudad. Tenga en cuenta que ya que las coordenadas del epicentro del terremoto se anotan a la ciudad más cercana, una aproximación a las coordenadas de una ciudad es calcular el punto medio de todos los terremotos asignados a esa ciudad. Para resolver el ejercicio use el método del apartado anterior y defina un método auxiliar getCoordenadasMedias que dada una lista de coordenadas, devuelva un objeto de tipo Coordenadas cuya latitud es la media de las latitudes de la lista y cuya longitud es la media de las longitudes. **(1,5 Ptos)**
- e) **String getCiudadMasTerremotosNMayores(Integer n)**. Dados los n mayores terremotos (atendiendo a la escala de Richter), devuelve el nombre de la ciudad en la que se han producido más terremotos entre los n de mayor magnitud. **(1,5 Ptos)**

- f) **Double getMediaDiasEntreTerremotosCiudad(String ciudad).** Devuelve el número medio de días transcurridos entre los terremotos producidos en la ciudad dada como parámetro. Esto es, si en la ciudad ha habido 4 terremotos, de forma que entre el 1º y 2º han pasado 50 días, entre el 2º y el 3º 100 días y entre el 3º y el 4º 60 días, la media es  $210/3=70$  días. Use el método *until* de *LocalDate* con el argumento *ChronoUnit.DAYS* para calcular el número de días transcurridos entre dos fechas. Cree un método auxiliar para obtener una lista ordenada de las fechas en las que se han producido terremotos en la ciudad dada como parámetro. Si no se puede calcular la media, devuelve cero. **(1,5 Ptos)**

### **EJERCICIO 3 – FACTORÍA (1 pto)**

Implemente el tipo **FactoriaRegistros** para poder cargar la información del fichero de datos proporcionado y añádale el siguiente método:

- **Registro parsearRegistro(String lineaCSV).** Crea un objeto de tipo *Registro* a partir de una cadena de caracteres. La cadena de caracteres debe tener el mismo formato que una de las líneas del fichero especificado anteriormente. Si la línea no tiene el formato esperado, se elevará *IllegalArgumentException*. Use el formato *yyyy.M.d* para las fechas.