

# Laboratorio A.E.D. Ejercicio Individual 5

**Guillermo Román**

guillermo.roman@upm.es

**Lars-Åke Fredlund**

lfredlund@fi.upm.es

**Manuel Carro**

mcarro@fi.upm.es

**Marina Álvarez**

marina.alvarez@upm.es

**Julio García**

juliomanuel.garcia@upm.es

**Tonghong Li**

tonghong@fi.upm.es

**Sergio Paraiso**

sergio.paraiso@upm.es

**Juan José Moreno**

juanjose.moreno@upm.es

**Luis Miguel Danielsson**

lm.danielsson@alumnos.upm.es

# Normas.

- ▶ Fechas de entrega y nota máxima alcanzable:

Hasta el Miércoles 03 de noviembre, 12:00 horas	10
Hasta el Jueves 04 de noviembre, 12:00 horas	8
Hasta el Viernes 05 de noviembre, 12:00 horas	6
- ▶ Después la puntuación máxima será 0
- ▶ Se comprobará plagio y se actuará sobre los detectados
- ▶ Usad las horas de tutoría para preguntar sobre programación – son oportunidades excelentes para aprender

# Entrega

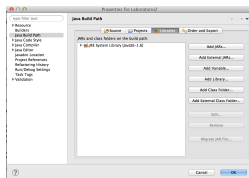
- ▶ Todos los ejercicios de laboratorio se deben entregar a través de `http://vps142.cesvima.upm.es`
- ▶ El fichero que hay que subir es `TempUtils.java`.

# Configuración previa

- ▶ Arrancad Eclipse
- ▶ Si trabajáis en portátil, podéis utilizar cualquier versión reciente de Eclipse. Es suficiente con que instaléis la *Eclipse IDE for Java Developers*.
- ▶ Cambiad a “Java Perspective”.
- ▶ Debéis tener instalado al menos Java JDK 8.
- ▶ Cread un proyecto Java llamado `aed`:
  - ▶ Seleccionad separación de directorios de fuentes y binarios.
  - ▶ **No debéis elegir la opción de crear el fichero `module-info.java`**
- ▶ Cread un *package* `aed.invididual5` en el proyecto `aed`, dentro de `src`
- ▶ Aula Virtual → AED → Laboratorios y Entregas Individuales → Individual 5 → Individual5.zip; descomprimidlo
- ▶ Contenido de Individual5.zip:
  - ▶ `TempData.java`, `TesterInd5.java`, `TempUtils.java`

# Configuración previa

- ▶ Importad al paquete `aed.individual15` los fuentes que habéis descargado (`TempData.java`, `TesterInd5.java`, `TempUtils.java`)
- ▶ Añadid al proyecto `aed` la librería `aedlib.jar` que tenéis en Moodle (en Laboratorios y Entregas Individuales).

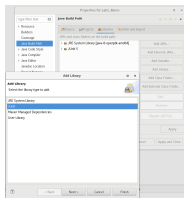


Para ello:

- ▶ Project → Properties → Java Build Path. Se abrirá una ventana como la de la izquierda
- ▶ Usad la opción “Add External JARs...”.
- ▶ Si vuestra instalación distingue `ModulePath` y `ClassPath`, instalad en `ClassPath`

# Configuración previa

- ▶ Añadid al proyecto aed la librería JUnit 5



- ▶ Project → Properties → Java Build Path. Se abrirá una ventana como la de la izquierda;
- ▶ Usad la opción “Add Library...” → Seleccionad “JUnit” → Seleccionad “JUnit 5”
- ▶ Si vuestra instalacion distingue ModulePath y ClassPath, instalad en ClassPath
- ▶ En la clase TesterInd5 tenéis las pruebas, para ejecutarlas, abrid el fichero TesterInd5, pulsando el botón derecho sobre el editor, seleccionar “Run as...” → “JUnit Test”
- ▶ NOTA: Si al ejecutar, no aparece la vista “JUnit”, podéis incluirla en “Window” → “Show View” → “Java” → “JUnit”

# Documentación de la librería aedlib.jar

- ▶ La documentación de la API de aedlib.jar está disponible en <http://costa.ls.fi.upm.es/entrega/aed/docs/aedlib/>
- ▶ También se puede añadir la documentación de la librería a Eclipse (*no es obligatorio*):
  - ▶ En el “Package Explorer”: “Referenced Libraries” → aedlib.jar y elige la opción “Properties”. Se abre una ventana donde se puede elegir “Javadoc Location” y ahí se pone como “javadoc location path:”

<http://costa.ls.fi.upm.es/entrega/aed/docs/aedlib/>  
y presionar el botón “Apply and Close”

# Tarea para hoy: Entrega Individual 5

- ▶ Terminar la implementación de la clase `TempUtils`, es decir, completar la implementación de los métodos:  
`maxTemperatures(...)` y  
`maxTemperatureInComunidad(...)`



# Notas máximas

- ▶ Si los dos métodos se implementan correctamente, la nota máxima es 10. Si se implementan correctamente un método, la máxima nota será 6.

## La clase TempData

La clase TempData contiene datos sobre la temperatura en un lugar español en un hora dada, y tiene 3 métodos “getters”:

- ▶ `string getLocation()` – devuelve el lugar donde se ha medido la temperatura
- ▶ `long getTime()` – devuelve la hora a la que se ha medido la temperatura
- ▶ `int getTemperature()` – devuelve la temperatura medida

Por ejemplo, `new TempData("Madrid",300,25)` expresa que la temperatura en Madrid en la hora 300 ha sido de 25 grados.

# Implementación

## ► El método

```
public static Map<String,Integer>  
    maxTemperatures(long startTime,long endTime,  
                    TempData[] tempData)
```

tiene los parámetros `startTime` (hora inicial), `endTime` (hora final), y `tempData` (un array con las temperaturas medidas).

- El método debe devolver un map (usad la clase `HashMap` para crear el map) que para cada ciudad donde se ha medido la temperatura en el intervalo `[startTime...endTime]` contiene un `Entry<String,Integer>` con la ciudad como clave y como valor la temperatura **máxima** medida.

# Ejemplo

```
maxTemperatures(310,330,  
    [TempData("Madrid",300,40),  
      TempData("Madrid",310,20),  
      TempData("Madrid",320,30)])  
==> [("Madrid",30)]  
      // Notad que el primer dato es fuera del intervalo
```

```
maxTemperatures(310,330,  
    [TempData("Madrid",310,20),  
      TempData("Malaga",330,35),  
      TempData("Madrid",320,30)])  
==> [("Madrid",30),("Malaga",35)]
```

# Implementación

## ► El método

```
public static Pair<String,Integer>  
    maxTemperatureInComunidad(long startTime,long endTime,  
                               String comunidad,TempData[] tempData,  
                               Map<String,String> ciudadEnComunidad)
```

recibe una hora inicial, una hora final, una comunidad, un array con datos de temperaturas, y un mapa que asocia un ciudad (clave) a una comunidad (valor).

- El método debería devolver un `Pair<String,Integer>`, con la ciudad, y la temperature máxima, medido durante en el intervalo `[startTime...endTime]`, **en la comunidad especificado**.
- Si no hay ninguna medida el método debería devolver `null`.
- Si hay multiples ciudades con la misma temperature máxima se puede devolver cualquier ciudad.
- El parámetro `ciudadEnComunidad` asocia a cada ciudad una comunidad. Por ejemplo, “Malaga” (clave) esta asociada a “Andalucia” (valor).

# Ejemplo

```
maxTemperatureInComunidad(310,330,  
  
    "Madrid",  
  
    [TempData("Mostoles",300,40),  
      TempData("Malaga",330,35),  
      TempData("Leganes",310,20),  
      TempData("Getafe",320,30)],  
  
    [("Mostoles","Madrid"),  
      ("Getafe","Madrid"),  
      ("Leganes","Madrid"),  
      ("Malaga","Andalucia")])  
  
==>  [("Getafe",30)]  
  
// Notad que se pide datos sobre ciudades en la Comunidad  
// de Madrid, y en el intervalo [310..330]
```

# Comentarios generales

- ▶ Debe ejecutar `TesterInd5` correctamente sin mensajes de error
- ▶ Nota: una ejecución sin mensajes de error no significa que el método sea correcto (es decir, que funcione bien para cada posible entrada)
- ▶ Todos los ejercicios se comprueban manualmente antes de dar la nota final