

## ESTRUCTURA Y DISEÑO

Al final he decidido utilizar un dos clases, una main que inicializa el programa general (UT12Problema1.java) y otra que recoge los datos meteorológicos del csv (DadesMetereologiques.java)

Para recoger los datos, primero le pido al usuario que escriba la fecha en el formato adecuado y luego lo adapta al nombre del fichero (Aemetxxxxxxxx.csv) y lo guarda en un String ruta. Luego leo línea por línea el csv dado utilizando BufferedReader. Recojo el dato previo a el primer ‘;’, es decir, la ciudad, y la asigno como clave del map que guarda los datos (la línea en formato String)

Después, en el programa principal hago un menú al uso. (Lo primero que hago es guardar un array de string de todas las claves de mi map de datos meteorológicos)

Para sacar por consola los datos de la ciudad, pido la ciudad y le cambio la primera letra a mayúscula. A continuación itero en el array de claves de ciudad, y, si la iteración actual contiene la ciudad puesta por el usuario, obtengo los datos de esa clave del map.

Separo los datos cada vez que haya un ‘;’ en el string y los coloco en un array string. Ahora simplemente hago una interfaz que muestre los datos del array.

Los otros casos del switch simplemente cambia el hecho de que busco un dato máximo o mínimo entre todos los datos y guardo sus registros para al final del array imprimir la interfaz con esos datos.

## RETOS Y PROBLEMAS

El mayor problema que he tenido es conseguir separar los registros individualmente de cada linea de datos, al final he encontrado lo que necesitaba utilizar (string.split(“;”))

## VENTAJAS ALMACENAMIENTO DATOS CSV

Al almacenar estos datos en formato csv, podemos tener ficheros de todos los días de todos los años, y con el mismo programa podemos ejecutarlos de forma sencilla, sin necesidad de una base de datos normal.