



Pràctica 6: Tractament de dades de sensors

Tecnologia de la Programació — iTIC

Aleix Llusà-Serra Sebastià Vila-Marta Marta I. Tarrés-Puertas

February 5, 2021

Contents

1	Introducció	1
1.1	Objectius	1
1.2	Condicions	2
1.3	Lliurament	2
1.4	Context	2
2	Descripció del sistema	2
2.1	Arquitectura	2
2.2	Format del fitxer	3
2.3	Interpretació de les dades	3
2.4	Estructura de mòduls i classes	4
3	Mòdul dataset	4
4	Mòdul datafetcher	6
5	Mòdul dsplot	7
6	Mòdul sensortransf	11
7	Mòdul explora	11
8	Bibliografia	12

1 Introducció

1.1 Objectius

Els objectius d'aquesta pràctica són:

- Tenir un primer contacte amb alguns serveis de la extensa llibreria de Python.
- Consolidar el treball amb classes d'objectes.
- Consolidar l'ús de les eines de test i el disseny basat en tests.

- Consolidar l'ús de les eines de documentació de programari i de les eines de gestió de versions.

La pràctica té com a context la captura, emmagatzemat, tractament i visualització de dades provinents d'un sistema de sensors que capturen les temperatures de diferents punts del laboratori de recerca de telecomunicacions.

La pràctica és molt curta pel que fa al codi que cal escriure. La solució no s'allarga massa més de les 200 línies de codi. La principal dificultat està en entendre i saber aprofitar els mòduls de la llibreria que cal utilitzar. És doncs una pràctica més d'estudi i exploració que no pas de desenvolupament.

1.2 Condicions

- La pràctica té una durada de dues sessions de laboratori.
- Cal fer la pràctica amb l'equip de treball.
- Cal fer el desenvolupament usant control de versions amb subversion sobre el dipòsit **escriu**³. L'ús de l'eina serà part de l'avaluació de la pràctica.

1.3 Lliurament

Caldrà lliurar el resultat de la pràctica a través de l'activitat escaient d'Atenea. El lliurament haurà d'incloure:

- El codi font del projecte.
- Els doctests associats al codi font.
- La documentació del projecte escrita amb **Sphinx** que ha d'incloure una taula del temps de dedicació de cada persona de l'equip.
- Un script amb les càlculs automatitzats corresponents a la tasca 1, i els resultats obtinguts.
- El fitxer de dades utilitzat.
- Es valorarà positivament la implementació de **mètodes recursius**.

1.4 Context

En el laboratori de recerca de telecomunicacions hi ha instal·lats 6 sensors numerats de 0 a 5 en diferents punts de l'habitable. Aquests sensors estan connectats a petit sistema de control basat en un microcontrolador que interroga tots els sensor cada pocs segons i enregistra la lectura que cadascun fa de la temperatura. Cada dispositiu sensor està implementat de forma diferent i, per tant, té unes característiques de treball diferents. El sistema de control va acumulant les dades de cada dia en un fitxer de text. Un computador que actua de servidor recupera diàriament aquest fitxer, l'emmagatzema conjuntament amb els de la resta de dies i els serveix públicament a través d'un servidor web.

L'objectiu d'aquesta pràctica és implementar un programa que es connecta al servidor web, obté les dades d'un o més sensors per a un període de dies determinat i, finalment, les visualitza mitjançant una gràfica en la pantalla del vostre computador.

2 Descripció del sistema

2.1 Arquitectura

L'esquema general de l'aplicació es pot veure a la figura 1. Tot i que en realitat hi ha 6 sensors numerats de 0 a 5, el diagrama només en mostra els tres primers. En aquest diagrama s'observa com cada sensor de temperatura està connectat a un petit sistema electrònic que tracta el senyal del sensor per tal que sigui utilitzable. Cada sensor té un sistema de tractament diferent. Un petit computador fa la feina de recollir les dades de cada sensor periòdicament. Aquest computador va emmagatzemant cada dada en un fitxer de text associat a la data. Cada dia crea un nou fitxer de text. Aquest mateix computador serveix la col·lecció de tots els fitxers de text a través d'internet usant el protocol HTTP. D'aquesta forma podem accedir als fitxers usant un navegador web.

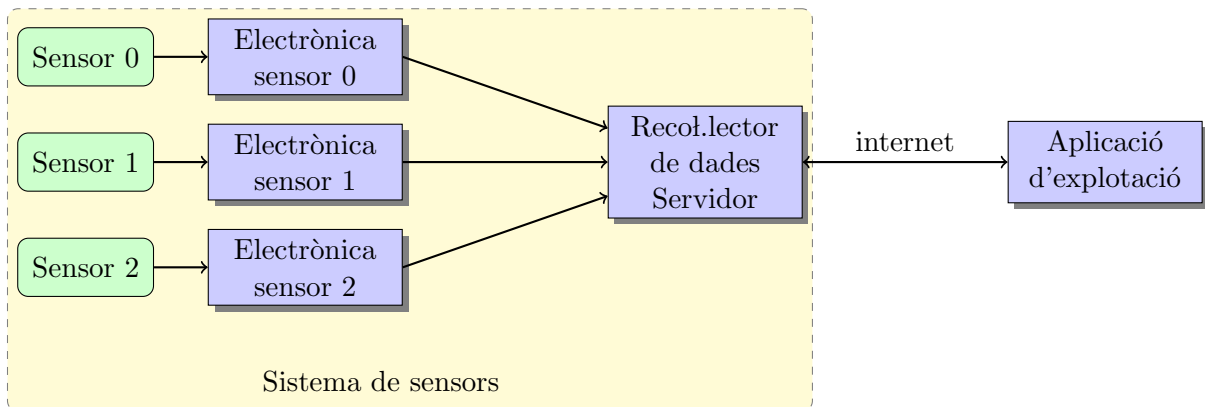


Figure 1: Arquitectura del sistema. Només es mostren tres dels sis sensors.

2.2 Format del fitxer

Els fitxers que emmagatzemen les dades dels sensors i que serveix el sistema de sensors són fitxers de text. El nom del fitxer té relació amb la data a la qual pertanyen les dades. Així el fitxer `dades_11_08_01`, per exemple, conté les dades dels sis sensors corresponents al dia 1 d'agost del 2011.

Un fitxer de dades s'organitza en línies. Cada línia correspon a una dada d'un sensor en un instant de temps. El format d'una línia és el que segueix:

```
00:00:04,5,526.000000
```

Noteu que hi ha tres camps separats per comes. El primer és l'instant de temps a que correspon la dada. Fixeu-vos que la sintaxi és la clàssica de `HH:MM:SS`. El segon és el número de sensor. Finalment el tercer és el valor de la dada i és de tipus real. El fitxer conté molts centenars de dades i per tant no és convenient emmagatzemar-lo completament a la memòria del computador.

Aquest tipus de formats en que les dades s'organitzen en línies i a la vegada cada línia s'organitza en camps separats per comes es coneix com a *comma separated values* o format CSV, i és un format molt freqüent per intercanviar dades. Molts fulls de càlcul i aplicacions estadístiques llegeixen aquest format de manera nativa.

2.3 Interpretació de les dades

Les dades que s'emmagatzemen en els fitxers no són temperatures: són les dades que subministren els sensors. Per obtenir la temperatura a partir d'aquestes dades cal aplicar-hi un procés de transformació. Com cada sensor és diferent, la transformació és específica per a cada sensor.

Assumim que tenim un valor que prové del sensor S_0 . Aleshores la temperatura T_0 que aquest sensor indica és realment $T_0 = 100V_0 - 50 + dT_0$ on dT_0 és una constant de calibrat empírica que depèn de cada sensor. Per la seva banda, $V_0 = \frac{Va_0 + dc_0}{k_0}$ on dc_0 i k_0 són constants que depenen del sensor. Finalment $Va_0 = \frac{Vcc}{1024.0} D_0$ on Vcc és una constant única per a tots els sensor i D_0 és el valor que llegim del fitxer.

La constant Vcc és 3.3040. La resta de constants que depenen del sensor són les que es mostren a la taula 1.

i	dc_i	k_i	dT_i
0	3.1418	5.6500	0.24
1	0.0000	1.0000	1.37
2	2.1850	5.3972	0.95
3	2.1313	5.2959	0.42
4	10.5953	16.8243	0.29
5	10.5842	16.5574	0.79

Table 1: Constants que caracteritzen els sensors.

TASCA PRÈVIA 1 Com les transformacions que s'apliquen a cada sensor són lineals es poden expressar en el seu conjunt com $T_i = A_i D_i + B_i$. Determineu per a cada sensor i el valor dels coeficients A_i i B_i . Penseu l'estratègia de resolució i féu un petit programa que faci els càlculs d'A i B.

2.4 Estructura de mòduls i classes

L'aplicació la constitueixen els mòduls:

dataset Conté la classe `DataSet`. Un objecte de la classe `DataSet` és un contenidor de valors cadascun d'ells associat a un temps.

datafetcher Conté la classe `DataSetFetcher`. Un objecte d'aquesta classe és una eina que permet obtenir les dades d'un sensor determinat i per a una data concreta.

dsplot Conté la classe `DataSetPlot`. Un objecte d'aquesta classe és una entitat que permet visualitzar una sèrie de dades en el temps sobre una gràfica en la pantalla.

sensortransf Conté eines per normalitzar la informació que prové de cada sensor.

explora Conté el programa principal.

La relació d'ús entre els diversos mòduls és la que es mostra en la figura 2.

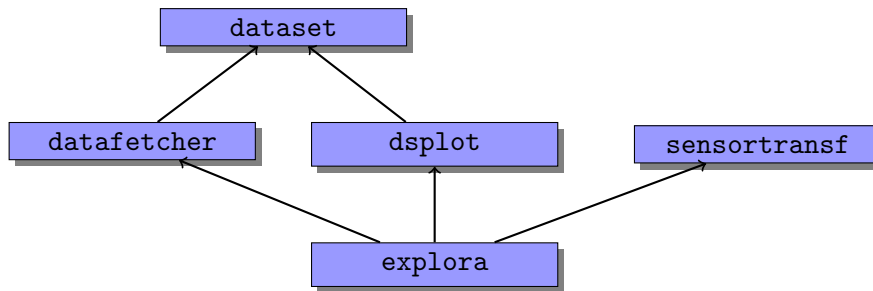


Figure 2: Diagrama d'ús entre els mòduls de la pràctica.

TASCA 2 Prepareu l'entorn de treball per a la practica: doneu d'alta un nou directori de nom p6 en el vostre dipòsit de versions i seguïu la mateixa estratègia de treball que en la pràctica anterior.

3 Mòdul dataset

TASCA PRÈVIA 3 Estudieu la classe `date` i `datetime`, que forma part del mòdul `datetime` de la llibreria de Python. Proveu els seus mètodes i feu-vos a la idea de com s'utilitza. Serà important per a tot el projecte.

El mòdul `dataset` conté la classe `DataSet`. Un objecte d'aquesta classe és un contenidor d'observacions que provenen d'un únic sensor. A més el mòdul també defineix la classe d'excepcions `OutOfOrderException`.

La classe `DataSet` compta amb el següents atributs:

name [`str`, Privat] És el nom del dataset. Cada objecte té un nom que l'identifica.

ds [`list`, Privat] És una llista de tuples (t,v) on t és un temps i v el valor d'un sensor en aquest temps. t és un objecte de tipus `datetime`, un tipus de la llibreria de Python que heu d'usar obligatòriament.

També implementarà els mètodes següents:

- `__init__(self, name='')`

Inicialitza els objecte de la classe.

- `__len__(self)`

Retorna el nombre d'elements del dataset.

- `__str__(self)`

Retorna una representació en forma de cadena de caràcters del dataset.

- `add(self, t, v)`

Afegeix una nova observació al dataset. Cal que aquesta observació sigui feta en un temps posterior a la darrera observació que contenia el dataset. En cas contrari, el llença l'excepció `OutOfOrderException`.

- `time_vector(self)`
Retorna una llista amb els temps corresponents a totes les observacions del dataset ordenats de menor a major.
- `value_vector(self)`
Retorna una llista amb els valors corresponents a totes les observacions del dataset ordenats per temps creixents.
- `decimate(self,k=10)`
Calcula un nou `DataSet` on cada element resulta d'agrupar els elements originals de `self` agafats de `k` en `k` i representar-los per un nou element que té per valor la mitjana dels originals i per temps el més gran dels originals.
- `moving_average(self,k=50)`
Calcula un nou `DataSet` on cada element resulta de calcular la mitjana mòbil. És a dir substituïm cada element per la mitjana dels `k` elements anteriors.
- `concat(self, ds2)`
Afegeix a `self` el `DataSet` `ds2`. La primera observació de `ds2` ha de ser posterior a la darrera de `self`. Altrament s'aixeca l'excepció `OutOfOrderException`.
- `transform(self, a=1.0, b=0.0)`
Modifica `self` tot aplicant al valor `v` de cada observació la transformació lineal $a*v + b$.

TASCA 4 Implementeu les classes `OutOfOrderException` i `DataSet`. A continuació definiu els doctests corresponents, assegureu-vos que són complets i funcionen correctament. Finalment documenteu el codi correctament i annexeu aquesta documentació a la documentació del projecte usant `Sphinx`.

4 Mòdul `datafetcher`

TASCA PRÈVIA 5 Estudieu i familiaritzeu-vos amb el mòdul `urllib2` de la llibreria de `Python`. Aquest mòdul l'usarem per a obtenir els fitxers del servidor de dades. Usant aquest mòdul la feina es molt senzilla i no és massa diferent al treball amb fitxers de text corrents. En particular és especialment interessant pel projecte el mètode `urlopen`.

TASCA PRÈVIA 6 Estudieu i familiaritzeu-vos amb el mòdul `csv` de la llibreria de `Python`. Aquest mòdul s'usa per facilitar la lectura de fitxers en format CSV. S'usarà en aquesta pràctica per a descodificar amb facilitat les dades que es llegeixen del servidor.

TASCA PRÈVIA 7 Dissenyeu un petit script que permeti la lectura de dades remota, utilitzant les llibreries de `python` `urllib2` i `csv`.

Aquest mòdul implementa la classe `DataSetFetcher` i també l'excepció `UnknownDataSetException`. Els objectes de la classe `DataSetFetcher` són utilitats que saben importar dades d'un sensor corresponents a un dia o un rang de dies des d'un servidor de dades.

Els objectes d'aquesta classe tenen un únic atribut que és l'URL del servidor d'on s'han d'obtenir les dades. Pel que fa als mètodes són els següents:

- `__init__(self, url='http://localhost:8000')`
Inicialitza els objectes de la classe. Cal indicar el servidor d'on cal obtenir els fitxers de dades.
- `fetch(self, dia, sensor=0)`
El mètode retorna un `DataSet` de nom '`Sensor X`', essent `X` el número de sensor, que conté les observacions del sensor `sensor` corresponents al dia `dia`. `dia` és un objecte de classe `date`. Si no es poden obtenir les dades aixeca l'excepció `UnknownDataSetException`.
- `fetch_interval(self, from_day, to_day, sensor=0)`
El mètode retorna un `DataSet` de nom '`Sensor X`', essent `X` el número de sensor, que conté les observacions del sensor `sensor` corresponents als dies que van de `from_day` a `to_day` incloent el primer i excloent el darrer. `from_day` i `to_day` són objectes de la classe `date`. Si no es poden obtenir les dades aixeca l'excepció `UnknownDataSetException`.

De cara a provar aquest mòdul és necessari tenir un servidor d'on obtenir les dades. Per fer proves és molt còmode tenir el servidor en el mateix computador on esteu desenvolupant. Això és molt senzill. Del mateix OCW iTIC us podeu descarregar un fitxer de dades que correspon a un dia concret. Deixeu aquest fitxer en el mateix directori on esteu desenvolupant el projecte (no cal que el pugueu al sistema de versions!!). Si des d'una terminal i prenent en aquest directori com a directori de treball executeu l'ordre

```
$ python -m SimpleHTTPServer
```

tindreu un servidor funcionant que podeu usar tranquilament. La seva adreça és `http://localhost:8000`. Proveu a visitar-la des del navegador, per exemple.

Aquest mecanisme és molt pràctic per altres tasques com, per exemple, intercanviar-se fitxers. Cal anar, però, amb compte ja que fa accessibles a tot el món els continguts del directori en qüestió!

TASCA 8 Implementeu les classes `UnknownDataException` i `DataSetFetcher`. A continuació proveu-les i assegureu-vos que el mòdul funciona correctament. Finalment documenteu el codi i annexeu aquesta documentació a la documentació del projecte usant `Sphinx`.

5 Mòdul `dsplot`

TASCA PRÈVIA 9 Aquest mòdul fa un ús important de `matplotlib`, una llibreria de `Python` especialitzada en visualitzar grafiques. És doncs molt important familiaritzar-se abans amb aquestes eines. A tal efecte és important llegir i provar el tutorial que trobareu en la documentació de la llibreria `matplotlib`. Si treballeu en el vostre computador us caldrà prèviament instal·lar la llibreria executant l'ordre:

```
$ sudo apt-get install python-matplotlib
```

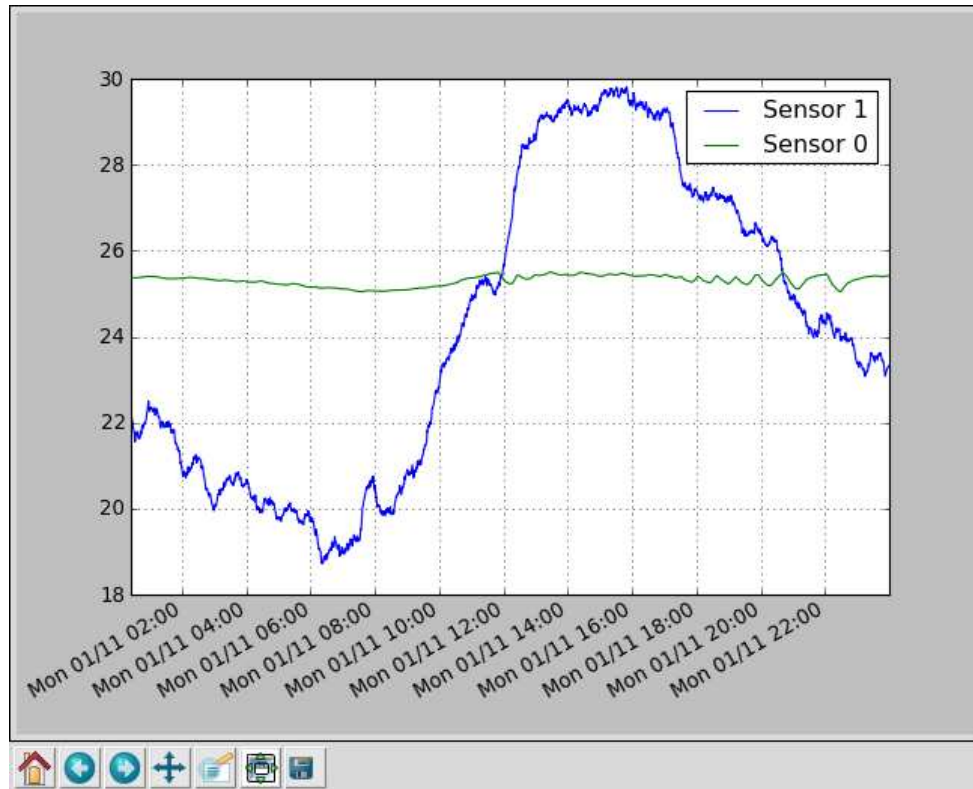


Figure 3: Aspecte de les gràfiques que cal generar.

Aquest mòdul implementa la classe `DataSetPlot`. Un objecte d'aquesta classe representa un gràfic de un o més objectes `DataSet`. La implementació és molt senzilla si s'entén correctament com ha d'usar-se la llibreria `matplotlib`.

TASCA PRÈVIA 10 Remeneu la llibreria `matplotlib` fins dominar la creació d'una gràfica a partir de dades.

Els mètodes de la classe `DataSetPlot` a implementar són els següents:

- `__init__(self)`
Inicialitza un objecte de la classe. En essència la inicialització s'encarrega de definir tota la gràfica excepció feta de les dades a visualitzar.
- `plot(self, d)`
Afegeix a la gràfica un conjunt de dades determinat per `DataSet` `d`. Si s'afegeix més d'un `DataSet` es representa cadascun d'un color diferent. L'etiqueta de la llegenda corresponent a aquest `DataSet` és el nom del mateix.
- `show(self)`
Mostra la gràfica per la pantalla. El mètode no retorna fins que l'usuari tanca la finestra de la gràfica.

Les gràfiques que cal generar han de tenir graella i llegenda. En l'eix de les abscisses hi ha el temps de les observacions i en les ordenades la temperatura en graus Cèlsius. Les Figures 3, 4, 5

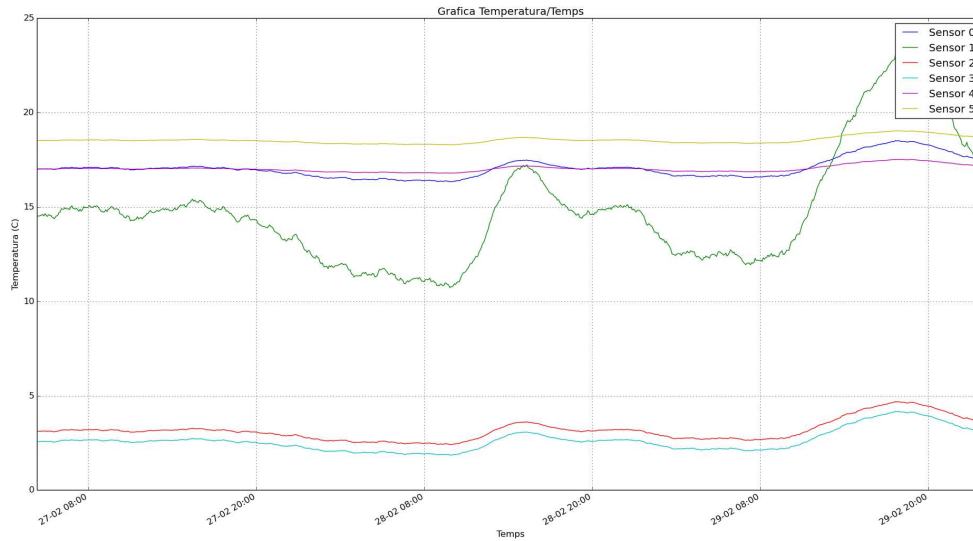


Figure 4: Aspecte de les gràfiques que cal generar.

mostren exemples d'aquestes gràfiques. La Figura 6 n'és un altre exemple més complex, que seria interessant que implementéssiu.

TASCA 11 Implementeu la classe `DataSetPlot`. A continuació proveu-la i assegureu-vos que el mòdul funciona correctament. Finalment documenteu el codi i annexeu aquesta documentació a la documentació del projecte usant `Sphinx`.

6 Mòdul `sensortransf`

Aquest mòdul solament exporta una funció capaç d'aplicar la transformació escaient a un `DataSet` per tal d'obtenir els valors de les temperatures.

Tal i com s'explica a l'apartat 2.3, les dades que es llegeixen dels sensors no són pròpiament temperatures. Cal aplicar les transformacions lineals que allí s'expliquen, diferents per a cada sensor, per tal d'obtenir-ne les temperatures.

La funció `normalize(d,s=0)` aplica la transformació escaient per al sensor `s` sobre el `DataSet` `d`. Una vegada aplicada els valors de l'objecte són temperatures en graus Cèlsius.

TASCA 12 Implementeu la funció `normalize`. Emprant els doctests assegureu-vos que el mòdul funciona correctament. Finalment documenteu el codi i annexeu aquesta documentació a la documentació del projecte usant `Sphinx`.

7 Mòdul `explora`

[TASCA PRÈVIA 13](#) [Comproveu com cal configurar un script Python per tal que sigui autoexecutable.](#)

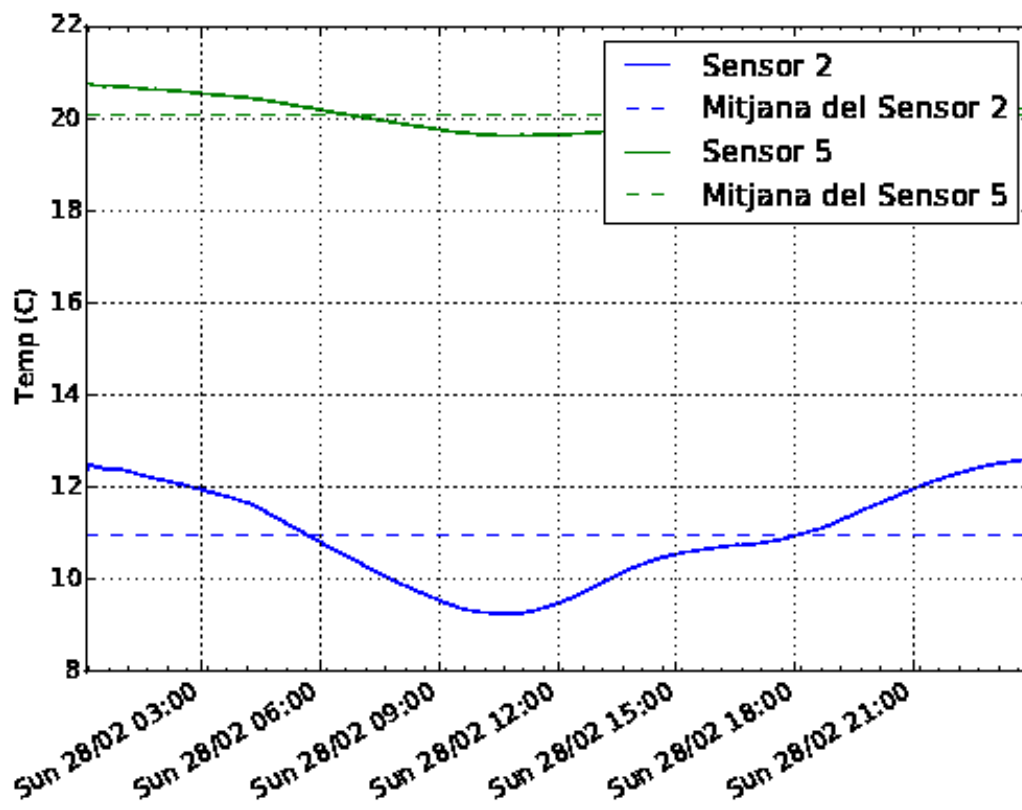


Figure 5: Aspecte de les gràfiques que cal generar.

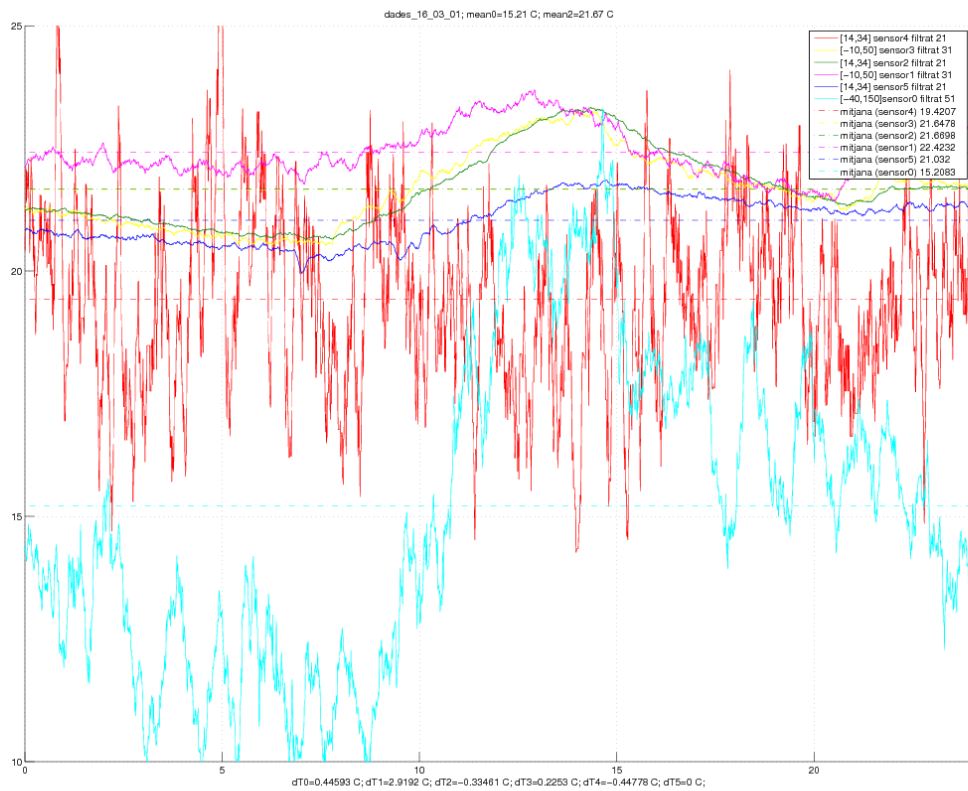


Figure 6: Aspecte de les gràfiques que cal generar.

Aquest mòdul és el programa principal. L'usuari l'ha de poder usar com una ordre més del sistema a través del terminal. En l'ús més senzill s'ha de poder dir:

```
$ explora -v http://localhost:8000 -s 0 -d 12/2/2016
```

i ha de mostrar-nos la gràfica corresponent a les dades del sensor 0 pel dia 1/8/2011 i extreus del servidor `http://localhost:8000`.

Si l'ordre és:

```
$ explora -v http://localhost:8000 -s 0,2,3 -d 12/2/2016
```

hauria de mostrar les dades dels sensors 0, 2 i 3 en el mateix gràfic.

Per últim, si l'ordre fos:

```
$ explora -v http://localhost:8000 -s 0,2 -d 12/2/2016 -f 15/2/2016
```

Hauria de mostrar les dades dels sensors 0 i 2 que van del dia 12/2/2016 fins al 15/2/2016 ambdós inclosos.

En cas que les dades no estiguin disponibles, l'ordre ha d'escriure un missatge d'error amable i acabar la seva execució.

De cara a mostrar les dades cal seguir el següent procés:

1. Obteniu-les del servidor corresponent.
2. Transformeu les dades del sensor a temperatura.
3. Apliqueu una reducció de la mida mitjançant una decimació que elimina 9 de cada 10 elements.
4. Posteriorment, calculeu una mitjana mòvil amb una finestra de 50 elements i visualitzeu aquest resultat.

TASCA 14 Implementeu el mòdul `explora`. A continuació proveu-lo i assegureu-vos que el mòdul funciona correctament. Finalment documenteu el codi i annexeu aquesta documentació a la documentació del projecte usant `Sphinx`. Diferencieu correctament la documentació d'usuari de la documentació tècnica.

TASCA 15 Adjunteu la gràfica en format `.png`, `.eps` o `.pdf` que heu generat, i afegiu una funcionalitat addicional.

8 Bibliografia

[1] John Hunter i Matplotlib contributors. Matplotlib. English. 2016. <http://matplotlib.org/> (visitat el 15 d'abril de 2016).

[2] Wikipedia contributors. Comma-Separated Values. English. Wikipedia, The Free Encyclopedia. 2016. url: https://en.wikipedia.org/wiki/Comma-separated_values (visitat el 15 d'abril de 2016).

[3] Wikipedia contributors. Matplotlib. English. Wikipedia, The Free Encyclopedia. 2016. <https://en.wikipedia.org/wiki/Matplotlib> (visitat el 15 d'abril de 2016).