

Benvinguts!

Aquest document conte els fonaments que necessitareu per les pràctiques dels dos primers terços de l'assignatura. Si us plau llegiu-lo atentament.

Les pràctiques

En aquesta primera part del curs, a les pràctiques, treballarem els següents punts:

- **Pràctica 0**
El que esteu llegint ara mateix. Us familiaritzareu amb l'entorn de pràctiques, us assegurareu que teniu tot el necessari, i fareu els vostres primers tests. No s'avalua, però us dona el que necessiteu per les següents dues pràctiques.
- **Pràctica 1**
Implementareu solucions per alguns problemes relacionats amb exploració de grafs i cerca de camins òptims.
- **Pràctica 2**
Resoldreu problemes més interessants relacionats amb l'assignatura

L'entorn de pràctiques

Aquesta secció us ajudarà a muntar els vostres entorns de test i de pràctiques.

Dependències de software

Heu d'assegurar-vos que teniu instal·lats:

- Python, versió 3.6 o superior (<https://python.org>)
- pip3, que utilitzareu per instal·lar paquets de software per Python

Després d'instal·lar pip3, fareu

```
pip3 install --user networkx matplotlib numpy scipy pandas
```

per instal·lar

- networkx (per treballar amb grafs)
- matplotlib (per fer gràfiques)
- numpy i scipy (funcions científiques)
- pandas (taules)

Podeu fer servir Windows, macOS o Linux. A Windows, recomaneu instal·lar el subsistema de Windows per Linux (<https://www.microsoft.com/store/productId/9P9TQF7MRM4R>), però hi ha múltiples altres opcions. Per assegurar-vos que tot està OK, obriu una consola Python (en general, amb la comanda python3) i executeu les següents línies:

```
import networkx

import matplotlib as mpl

from matplotlib import pyplot as plt

import numpy as np

import scipy
```

```
import pandas as pd

printf("Installation seems ok? {1+1==2}")
```

L'entorn de pràctiques

L'entorn de pràctiques és una carpeta que conté

- una mica de codi (com `config.py` o `renfe.py`)
- diverses carpetes, amb mostres de dades (`/data`), tests unitaris (`/test`), etc.

Instruccions generals

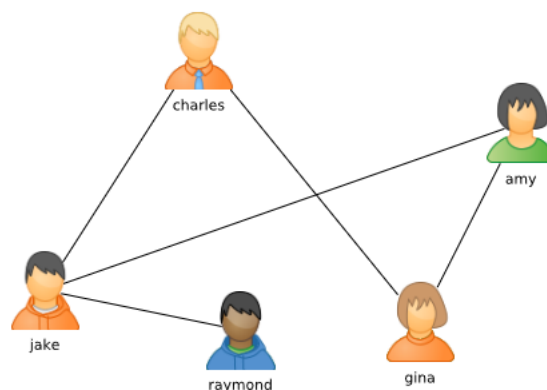
1. Baixeu la versió més recent del laboratori del Campus Virtual.
2. Dupliqueu la carpeta `lab0_template` amb un altre nom.
3. Feu la feina que se us demana a continuació en aquesta carpeta.

Tasques de la pràctica 0

La biblioteca `networkx` per Python és una eina molt potent per definir i analitzar grafs. Hi ha documentació molt completa a <https://networkx.github.io/documentation/latest/> i un tutorial introductori a <https://networkx.github.io/documentation/stable/tutorial.html>. Feu-los servir per fer les següents tasques:

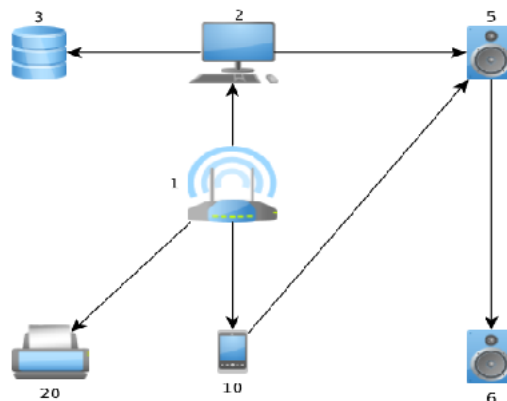
Funció `create_graph_1`

Implementeu la funció `create_graph_1` que retorna un graf de `networkx` representant la següent figura:



Funció create_graph_2

Implementeu la funció create_graph_2 que retorna un graf de networkx que representa la següent figura:



Funció create_station_graph

El mòdul renfe.py que se us dona permet accedir a un data set real. S'hi defineixen unes quantes classes que descriuen Station (una estació de tren), Advance (trajecte d'un tren d'una estació a una altra) i Journey (compost de un o més Advance). La classe RenfeReader implementa els mètodes get_stations_by_id() i get_journeys(). El primer és un diccionari que vincula identificadors d'estació amb instàncies de Station. El segon retorna una llista de tots els Journeys d'un determinat dia. Per exemple, per obtenir les primeres IDs d'estacions de Renfe en ordre alfabètic, podríem fer:

```
import renfe
```

```
reader = renfe.RenfeReader()
```

```
stations_by_id = reader.get_stations_by_id()
```

```
print(sorted(stations_by_id.keys())[:5])
```

i ens hauria de donar una llista de cadenes de text com

```
['04040', '06001', '06002', '06003', '06004']
```

Per iterar sobre tots els journeys, podem fer

```
journeys = reader.get_journeys()
```

```
for i in range(5):
```

```
    print(journeys[i])
```

i ens donarà uan sortida com

```
[Journey:1038L17102C3:C3: Chamartin - Aranjuez:(12 advances)]
```

```
[Journey:1038L17104C3:C3: Chamartin - Aranjuez:(12 advances)]
```

```
[Journey:1038L20000C3:C3: Chamartin - Aranjuez:(12 advances)]
```

```
[Journey:1038L20002C3:C3: Chamartin - Aranjuez:(12 advances)]
```

```
[Journey:1038L20004C3:C3: Chamartin - Aranjuez:(12 advances)]
```

La darrera tasca és crear una funció `create_station_graph` que ens retorni un graf `G` tal que

- Cada node és una instància de `Station`.
- Hi ha exactament una instància per a cada estació retornada per `RenfeReader`.
- Hi ha exactament una aresta entre cada parell d'estacions connectades directament per tren.
- Les arestes tenen una propietat `count` que és el nombre de `journeys` (trens) que passeu per les estacions de l'aresta, en una direcció o altra. Per exemple, ha de ser 67 per l'aresta que connecta l'Autònoma amb Cerdanyola del Vallès. Ho podeu comprovar amb una comanda de l'estil de `G.edges[u,v]['count']`.