



**Universitat Autònoma
de Barcelona**

Fonaments de Xarxes (MO34702)
Pràctica 3

Dr. Pere Tuset-Peiró
(pere.tuset@uab.cat)

Departament de Telecomunicacions i Enginyeria de Sistemes
Universitat Autònoma de Barcelona

1r semestre
Curs 2020-2021

Enunciat

En la tercera pràctica de l'assignatura Fonaments de Xarxes en centrarem en la simulació dels protocols de control d'enllaç de dades que es treballen al Capítol 7 del llibre “Comunicacions i Xarxes d'Ordinadors” de William Stallings.

Per fer-ho s'ha implementat un programa amb Python, anomenat `channel.py`, que permet simular el retràs de propagació i la pèrdua de paquets en un enllaç de comunicacions que connecta dues estacions, anomenades **Sender** i **Receiver**, tal com es mostra a la Figura 1. Per fer-ho el programa `channel.py` utilitza *sockets* UDP per rebre i reenviar els paquets del canal de pujada i de baixada cap a ambdues estacions. En concret, el programa `channel.py` utilitza el port UDP/9001 per rebre els paquets de dades de l'estació **Sender** i les re-envia cap al port UDP/9002 de l'estació **Receiver** després d'introduir el retràs i la pèrdua de paquets corresponent. De la mateixa manera, el programa utilitza el port UDP/9003 per rebre els paquets de confirmació (*acknowledgement*) de l'estació **Receiver** i les re-envia cap a l'estació **Sender** utilitzant el port UDP/9004 després d'introduir el retràs i la pèrdua de paquets corresponent.

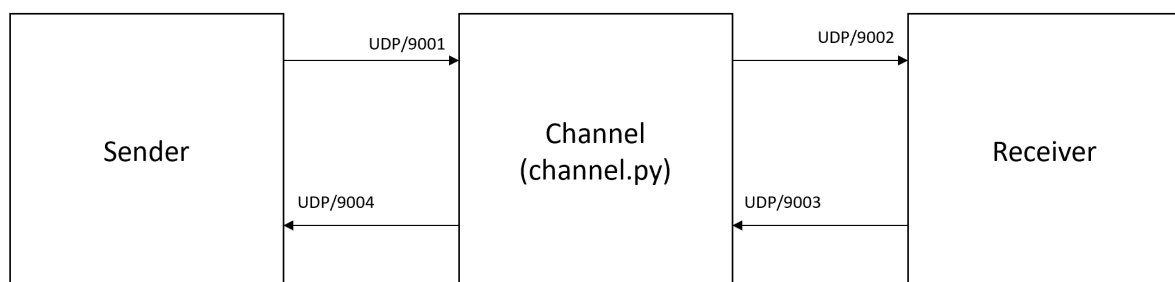


Figura 1

Tenint en compte que el programa `channel.py` ja està implementat i s'adjunta amb l'enunciat de la practica, es demana realitzar les activitats que es presenten a continuació. Tingueu en compte que la implementació dels diferents programes es pot realitzar amb qualsevol llenguatge de programació que estigui suportat per el sistema operatiu

GNU/Linux, tot i que es recomana fer-ho amb C/C++ o Python. De la mateixa manera, els gràfics i taules de l'informe es poden realitzar mitjançant qualsevol eina sempre que tinguin la qualitat adequada.

Activitat 0 (Treball previ)

Abans de realitzar l'activitat es demana que feu la instal·lació del sistema operatiu GNU/Linux en un entorn virtualitzat (VMWare o similar) i que en valideu el funcionament, tal com es descriu a l'Annex d'aquest document.

Activitat 1 (4 punts)

Es demana desenvolupar dos programes, anomenats **sender1** i **receiver1**, que implementin l'enviament de dades mitjançant un protocol de parada i espera (en anglès, *stop&wait*).

El programa **sender1**, que s'encarrega d'enviar les dades cap al port UDP/9001 i de rebre els paquets de confirmació a través del port UDP/9004¹, ha de tenir com a paràmetres d'entrada els següents paràmetres:

- la taxa d'enviament de dades en bits per segon,
- la mida dels paquets de dades en bits, amb un màxim de 1024 bytes i tenint en compte que el contingut és irrellevant (es pot omplir de manera aleatòria o bé amb dades conegudes),
- el nombre total de paquets de dades enviar abans de finalitzar l'execució.

En executar-se el programa ha de realitzar l'enviament de les dades segons els paràmetres d'entrada i, en finalitzar l'enviament, el programa **sender1** ha d'imprimir per pantalla la taxa efectiva d'enviament de dades (en bits per segon) i la utilització de l'enllaç (tant per-cent). Tingueu en compte que l'ample de banda del canal de comunicació és molt elevat, degut a que només implica la copia del paquet a través de la memòria del l'ordinador, de manera que heu de generar el nombre de paquets per segon que correspongui a partir de la taxa d'enviament de dades i la mida del paquet, i afegint esperes de temps en l'enviament dels paquets consecutius. Aquest cas és especialment important, ja que de no implementar-se correctament els resultats obtinguts no coincidiran amb els esperats.

¹En tots els casos l'adreça d'enviament i recepció dels paquets de dades i dels paquets de confirmació és l'adreça de `localhost` (127.0.0.1).

D'altra banda, el programa `receiver1` ha rebre els paquets de dades a través del port UDP/9002 i ha d'enviar els paquets de confirmació al port UDP/9003. En aquest cas, el programa `receiver1` no cal que tingui paràmetres d'entrada, doncs envia el mateix paquet de dades que ha rebut per tal de fer la confirmació de recepció.

Tal com s'ha descrit anteriorment, entre el programa `sender1` i el programa `receiver1` hi haurà el programa anomenat `channel.py` que s'encarrega de fer el reenviament dels paquets de dades i de confirmació, i que simula el retràs en la transmissió i la pèrdua de paquets. En aquest cas, al no haver-hi errors en la transmissió dels paquets de dades ni de confirmació, no cal que s'implementi cap mecanisme de reenviament passat un temps màxim (*timeout*).

Activitat 2 (3 punts)

A partir de la implementació dels programes `sender1` i `receiver1` desenvolupats a l'activitat anterior, es demana realitzar la seva execució amb els paràmetres d'entrada que s'especifiquen a la Taula 1. En concret, per cada parella de paràmetres ample de banda i retràs cal obtenir la taxa efectiva d'enviament de dades (en bits per segon) i la utilització de l'enllaç (tant per-cent).

Ample de banda (kbits/segon)	Retràs (ms)			
	1	10	100	1000
1				
1000				
1000000				

Taula 1: Valors d'ample de banda (kbits/segon) i retràs (ms) pel calcul de la taxa efectiva d'enviament i la utilització de l'enllaç.

De cara l'execució dels diferents programes es recomana obrir tres consoles independents i executar els programes en el següent ordre: `channel.py`, `receiver` i `sender`. Per tal d'executar el programa `channel.py` heu d'utilitzar la següent sintaxi:

```
1 python3 channel.py --up_delay X --down_delay Y
```

on les variables `X` i `Y` son els valors en mili-segons de canal de pujada (`sender` cap a `receiver`) i baixada (`receiver` cap a `sender`) respectivament. Tingueu en compte que els valors per defecte de `X` i `Y` son 10 mili-segons. Tingueu en compte també que cal

realitzar l'enviament d'un nombre suficient de paquets (per exemple, 100 paquets) per tal que el resultat sigui acurat.

A partir dels resultats obtinguts segons els paràmetres d'entrada es demana crear:

- una taula on es mostrin els valors obtinguts de la taxa efectiva d'enviament i la utilització de l'enllaç,
- un gràfic per cada paràmetre d'entrada on es mostri la informació de forma visual.

Activitat 3 (2 punts)

A partir de la implementació del programa `sender1` realitzada a l'Activitat 1, es demana modificar-ne el funcionament per tal que implementi un protocol de finestra lliscant. El nou programa s'ha d'anomenar `sender2` i ha d'acceptar com a paràmetre d'entrada la mida de la finestra lliscant. En aquest cas tampoc cal que s'implementi cap mecanisme de reenviament passat un temps màxim (*timeout*), ja que assumim que el canal no introdueix errors en la transmissió.

Activitat 4 (1 punts)

Es demana refer l'execució de l'Activitat 2 però utilitzant els programes `sender2` i `receiver1`, i crear les mateixes taules i gràfics on es mostri el rendiment del protocol de finestra lliscant segons la mida de la finestra. Concretament, utilitzeu els valors de mida de finestra 10, 100 i 1000, i discutiu com afecta la seva mida als valors de la taxa efectiva d'enviament i la utilització de l'enllaç.

Format de l'entrega

La data d'entrega de la Practica 3 és el **29 de desembre de 2020 a les 23:59h** per tots els grups de pràctiques. Cal realitzar una entrega per grup a través del Campus Virtual mitjançant un fitxer comprimit en format `zip` que contingui:

- L'informe de la practica en format `pdf` on a la primera pagina s'indiqui clarament els membres del grup
- Una carpeta anomenada `activitat1` que contingut el codi font dels programes `sender1` i `receiver1`

- Una carpeta anomenada **activitat2** que contingut el codi font del programa **sender2**

De cara l'avaluació de la practica es tindrà en compte:

- La claredat de la implementació dels diferents programes (**sender1**, **receiver1** i **sender2**), especialment els noms de les variables i la incorporació de comentaris que ajudin a comprendre'n el funcionament
- L'explicació que es faci dels diferents resultats obtinguts, indicant com es comparen respecte els resultats teòrics esperats, i la qualitat de les taules i els gràfics de l'informe, especialment els títols dels eixos, les unitats de les variables, etc.

Comentaris

De cara la implementació dels programes **sender** i **receiver** és convenient tenir en compte els següents aspectes:

- El pas de paràmetres per l'execució d'un programa en el cas del llenguatge de programació C es fa a través de les variables **argc** i **argv** de la funció **main**. En el cas d'utilitzar el llenguatge de programació Python podeu utilitzar el modul **argparse**, tal com s'utilitza en el programa **channel.py**.
- Per tal d'implementar el retràs entre l'enviament de paquets consecutius utilitzant C podeu utilitzar la funció **gettimeofday** de la llibreria **sys/time.h** i la funció **usleep** de la llibreria **unistd.h**. En el cas del llenguatge de programació Python podeu utilitzar les funcions **time()** i **sleep()** del modul **time**, tal com s'utilitza al programa **channel.py**.

Treball previ

El treball previ del Laboratori 3, que cal realitzar abans de la sessió de pràctiques, inclou la instal·lació del sistema operatiu GNU/Linux sobre l'entorn VMWare Workstation Player i les eines necessaries, tal com es descriu a continuació.

Instal·lació de GNU/Linux sobre VMWare

El primer pas és realitzar la instal·lació del sistema operatiu GNU/Linux.

Per fer-ho utilitzarem l'entorn de virtualització VMWare Workstation Player (versió 15.1.0), que es pot descarregar d'aquesta [web](#)². El procés d'instal·lació de l'entorn de virtualització VMWare Workstation Player és directe i no es detalla en aquesta guia.

A continuació cal descarregar la imatge del sistema operatiu GNU/Linux. En aquest cas utilitzarem la versió 20.04 LTS (Focal Fosa) de la distribució Ubuntu, que té suport per a 5 anys i es pot descarregar de la següent [web](#)³ (atenció, cal descarregar la versió anomenada "*Desktop image*").

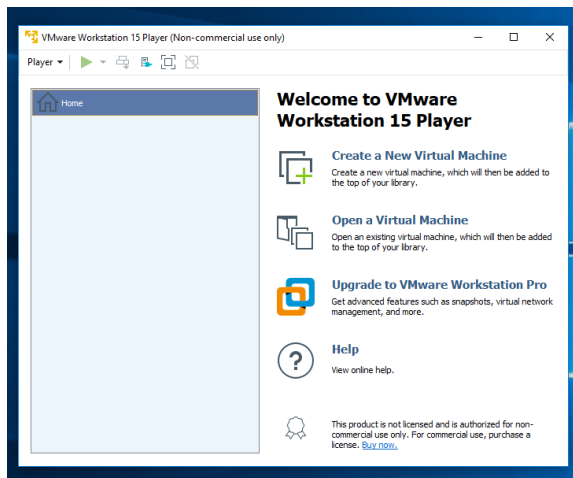
Un cop instal·lat l'entorn de virtualització i descarregada la imatge del sistema operatiu procedirem a la seva instal·lació.

El primer pas és obrir VMWare Player i crear una màquina virtual nova. Per a això cal escollir l'opció "*Create a New Virtual Machine*", tal com es mostra a la Figura 2a. A continuació cal triar l'opció "*Typical (recommended)*" (Figura 2b) i triar el fitxer `.iso` que conté la imatge del sistema operatiu (Figura 2c).

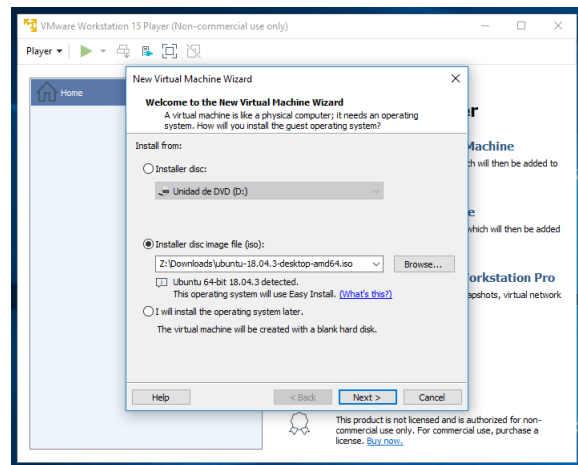
Un cop seleccionada la imatge del sistema operatiu procedirem a escollir un nom d'usuari i contrasenya (en aquest cas: `ubuntu` / `ubuntu`) i el nom de la màquina virtual (Ubuntu 64 bits), tal com es mostra a la Figura 3a. Finalment escollirem el nom de la màquina virtual (Ubuntu 64 bits) i la mida del disc virtual (32 GBytes), tal com es mostra a la Figura 3b i Figura 3c, respectivament.

²<https://www.vmware.com/go/downloadworkstationplayer>

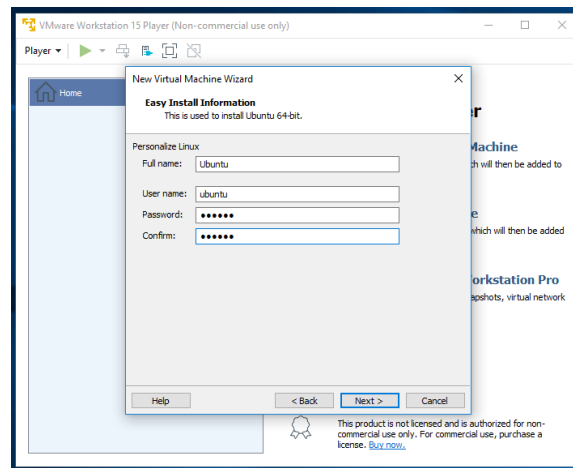
³<http://releases.ubuntu.com/20.04/>



(a)



(b)



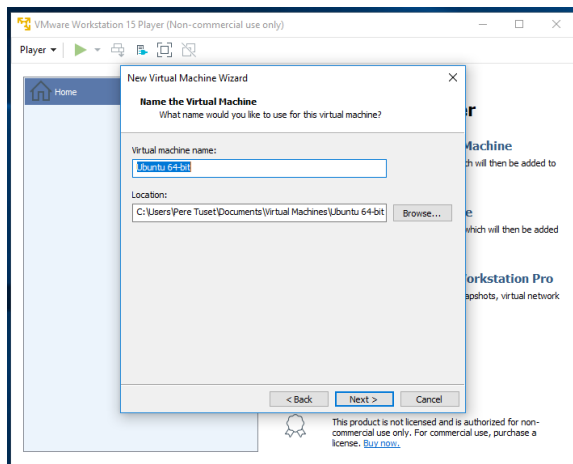
(c)

Figura 2: Creació de la màquina virtual Ubuntu 20.04 LTS amb VMWare Player.

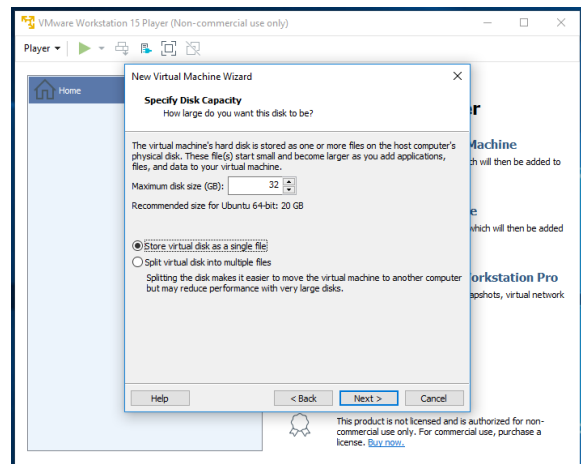
VMWare disposa d'una manera d'instal·lació desatesa, de manera que un cop s'arrenca la màquina virtual per primera vegada es procedeix a la instal·lació automàtica del sistema operatiu. Un cop completat el procés d'instal·lació la màquina virtual es reiniciarà i se'ns presentarà amb la pantalla d'inici de sessió, tal com es mostra a la Figura 4a, on haurem d'utilitzar l'usuari i la contrasenya elegits anteriorment.

Un cop iniciada la sessió ens trobarem l'escriptori (GNOME), on podrem començar a treballar tal com es mostra a la Figura 4b.

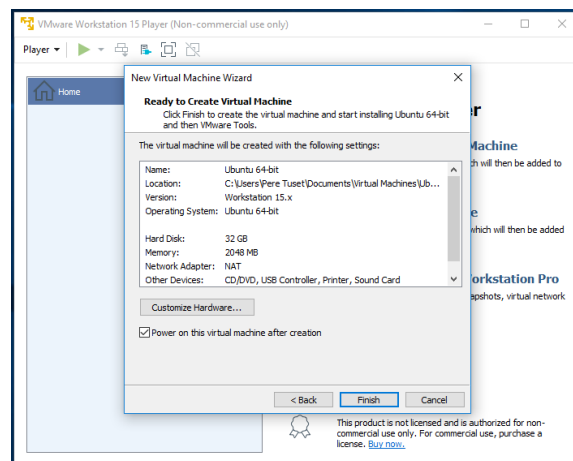
El primer pas per a configurar el sistema serà canviar la localització del teclat, ja que per defecte està en anglès i presentarà problemes a l'hora d'introduir caràcters especials. Per a això premerem la combinació de tecles CTRL + ALT + T per obrir un terminal i tot



(a)

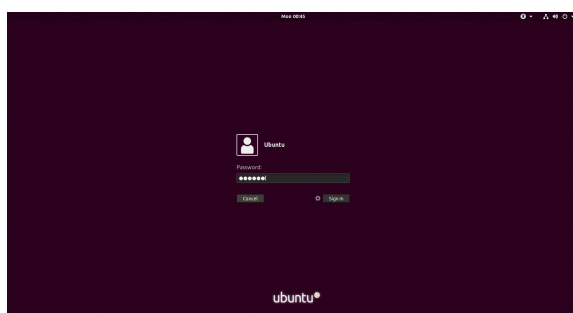


(b)



(c)

Figura 3: Creació de la màquina virtual Ubuntu 20.04 LTS amb VMWare Player.



(a)



(b)

Figura 4: Inici de sessió de Ubuntu 20.04 LTS amb VMWare Player.

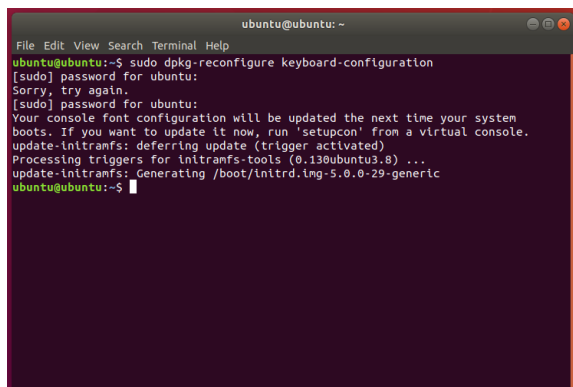
seguit executarem les següents comandes (una comanda per línia) per tal de canviar la localització del teclat a espanyol de manera permanent:

```
1 gsettings set org.gnome.desktop.input-sources sources "[('xkb', 'es' )]"
2 gsettings set org.gnome.desktop.input-sources current 1
```

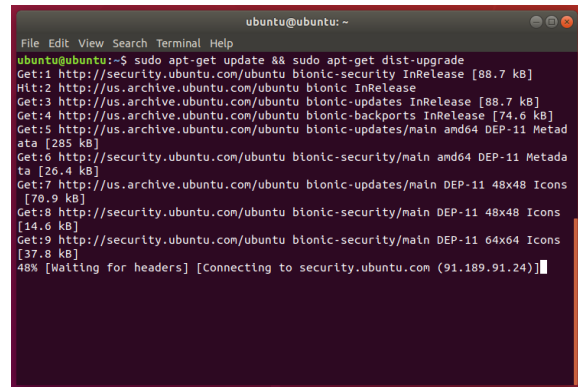
Una vegada modificada la localització del teclat actualitzarem el sistema operatiu executant les següents comandes en el terminal, tal com es mostra a la Figura 5b.

```
1 sudo apt-get update && sudo apt-get dist-upgrade
```

El sistema ens demanarà la contrasenya de *root* (ubuntu) i procedirà a actualitzar-se.



(a)



(b)

Figura 5: Canvi de la localització del teclat i actualització del sistema.

Un cop finalitzada l'actualització també haurem d'instal·lar les eines de suport de virtualització del sistema operatiu. Per fer-ho heu d'executar la següent comanda:

```
1 sudo apt install open-vm-tools
```

Un cop finalitzat el procés d'instal·lació i actualització del sistema operatiu haurem de reiniciar la màquina i ja estarem llestos per acabar d'instal·lar l'entorn de treball.

Instal·lació i test de l'entorn de treball

Un cop tingueu el sistema operatiu GNU/Linux funcionant a través de la màquina virtual, podeu procedir amb la instal·lació de les eines, tal com es descriu a continuació.

El primer pas és instal·lar les dependències. Per fer-ho obriu una consola de comandes i executeu la comanda següent:

```
1 sudo apt-get install build-essential git python3 python3-numpy python3-matplotlib
```

Per tal de validar el correcte funcionament de Python al vostre entorn de treball cal que obriu un terminal de comandes i executeu la següent comanda:

```
1 python3
```

Si la instal·lació ha estat correcte hauríeu de veure un terminal amb la consola Python similar al que es mostra a la Figura 6.

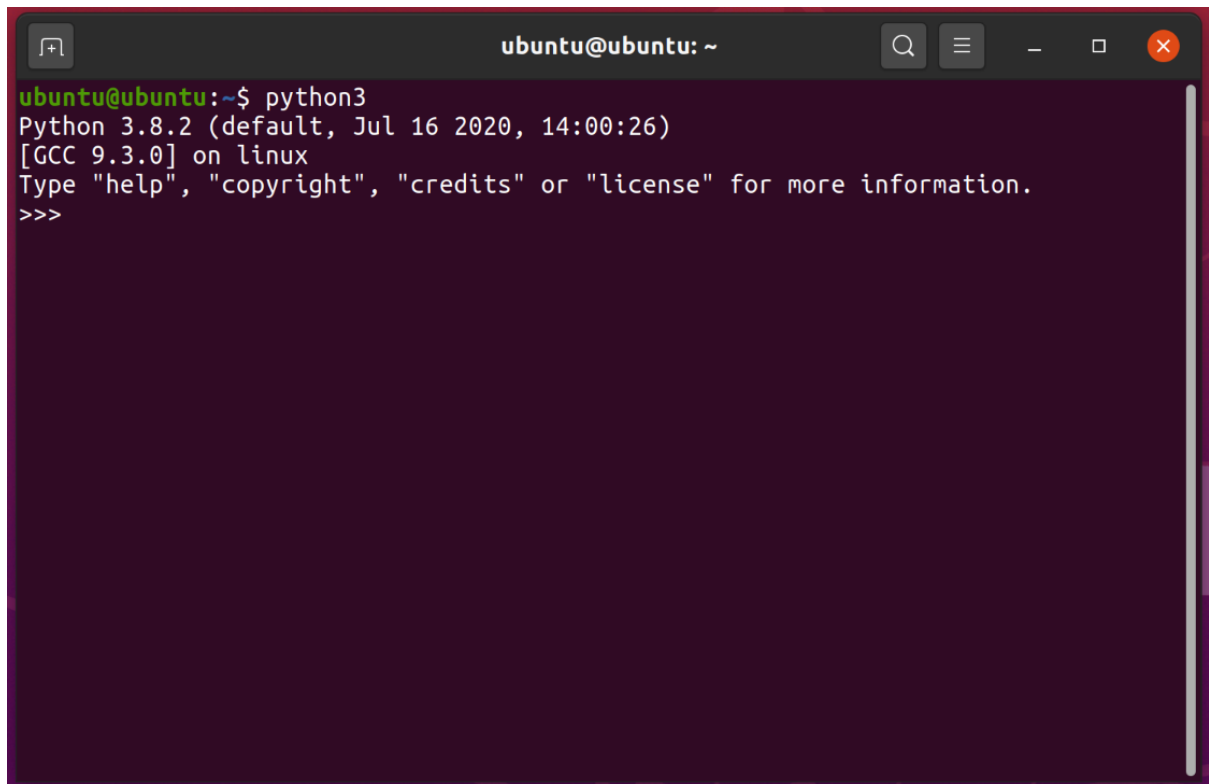


Figura 6: Terminal GNU/Linux amb consola Python.