

Programación para Física y Astronomía

Departamento de Física.

Profesores Claudia Loyola / Alejandro Llanquihuen / Joaquín Peralta

Primer Semestre / 2022

Universidad Andrés Bello



Computadores y Programación

Introducción

El ambiente GNU/Linux, editores y manejo de archivos.

GNU/Linux

Utilidades GNU/Linux

La shell de linux

Editores de texto

Instalando GNU/Linux

Usando VMWARE Workstation Player

Google Colab

Computadores y Programación

- Casi todos han usado un computador en alguna ocasión, para jugar, escribir, administrar el dinero, etc.
- Los computadores se usan para predecir clima, diseñar aviones, crear películas, hacer negocios, entre muchas otras cosas.
- ¿Pero qué es un computador? ¿Por qué puede realizar distintas tareas?.
- Un computador moderno podría definirse como una máquina que almacena y manipula información bajo el control de un programa cambiante.

- Almacena y manipula:
 - Transforma información en otra información útil.
 - Como una calculadora, operando con cientos o miles de números.
 - Como una máquina para surtir gasolina a un vehículo (tarea específica).
- Operan bajo el control de un programa cambiabile. ¿?
 - Un programa, es un set de instrucciones que le dice al computador exactamente lo que debe hacer.
 - Si el programa cambia, entonces el computador realiza acciones distintas, por lo que la tarea también es distinta.
 - Esta flexibilidad es lo que permite que un computador sea en un momento un procesador de texto, y luego un juego de *arcade*.

Un computador es una máquina que lleva a cabo (ejecuta) programas. Es posible que estén familiarizados con Mac o PC, pero hay muchos tipos de computadores, tanto reales como teóricos.

- Así, es el *Software* el que controla el *Hardware*
- Es el *Software* el que determina lo qué puede hacer un programa.
- Programación es una actividad desafiante. Si bien no todos tienen el talento para ser programadores de primera línea (así como no todos tienen el talento de ser atletas), virtualmente cualquiera puede aprender cómo programar en computadores.
- Con paciencia y esfuerzo de su parte, este curso les entregará las herramientas necesarias para comenzar a programar.

- Se podría sorprender que aprender *Computer Science* (Ciencia de Computación) **no es** el estudio de computadores.
- Un famoso científico de *Computer Science*, Edgser Dijkstra, bromeó con que los computadores son a *Computer Science* lo que los telescopios son a astronomía.
- ¿Pero qué podemos calcular? Las principales técnicas de investigación para responder esto son **diseño, análisis, y experimentación**.
- **Diseño**, es el proceso paso a paso para obtener un resultado esperado. Lo que nos llevará al algoritmo. El tener un diseño, no siempre implica que la solución sea alcanzable.
- **Análisis** es examinar el algoritmo y el problema desde un punto de vista matemático.
- **Experimentación** es el estudio de los resultados, y se utiliza principalmente cuando el análisis es demasiado complejo para poder guiar la verdadera solución del problema.

- Similar a manejar un auto. Saber qué gasolina utilizar, cómo iniciar el motor, y utilizar distintos *ajustes*, son parámetros importantes en la conducción.
- Es decir, no necesitamos conocer el detalle fino, pero tener una idea es de gran ayuda para una conducción correcta. Ocurre lo mismo con los computadores.
- Si bien los computadores pueden variar, todos presentan ciertas cosas en común.
 - CPU: Central Processing Unit. Es el **cerebro** del computador. Realiza operaciones básicas, como aritmética simple y operaciones lógicas.
 - RAM: Random Access Memory. Acá se almacenan los programas, y los datos. La CPU sólo puede acceder a lo almacenado en esta memoria. Esta memoria es rápida, pero volátil, si el equipo se apaga, todo desaparece.
 - Permanent Storage: Esto corresponde a discos externos o CD/DVD o memorias flash, entre otros.

¿Qué ocurre cuando ejecutamos nuestro juego favorito, o editor de textos?

1. Las instrucciones que constituyen el programa son copiadas desde la memoria permanente (disco duro), a la memoria principal (RAM).
2. Una vez que las instrucciones fueron cargadas, la CPU comienza la ejecución de las instrucciones.
3. Técnicamente, la CPU sigue un proceso llamado *fetch-execute cycle*. Esto es tomar una instrucción, decodificarla, y ejecutarla. Si bien no parece tan interesante el proceso, las CPU son capaces de llevar a cabo este proceso de una manera muy rápida y eficiente.

Un clúster de computadores, corresponde a un arreglo de computadores que son utilizados en forma paralela para resolver un problema específico. Hoy en día muchos servicios son manejados por cluster de computadores

- Bancos
- Gobiernos
- Empresas de estudios de dato (Big Data)
- Desarrollo de AI
- Ciencia

La infraestructura de FENIX:

- 19 Nodos de cómputo de CPU (44 Core cada uno, un total de 836 Core).
- 4 Nodos de cómputo GPU (Tesla K80)
- Switch IB+(2xRJ45)
- Gestor de Acceso y Disco (3 Nodos)



The screenshot shows the MATBIO website with a navigation bar at the top containing links for Home, Investigación, Publicaciones, Proyectos, News, and Cluster. The main content area features a section for the 'Fenix Cluster' with tabs for 'Componentes' and 'Financiamiento'. The 'Componentes' tab is active, displaying a paragraph about the cluster's high-performance computing capabilities and its use in materials research. To the right of the text is a photograph of server racks. Below the text, there are two buttons: 'Accediendo a Fenix' and 'Software disponible'. On the right side of the page, there is a 'Solicitudes' section with a link to 'Realizar Solicitud'.

MATBIO Home Investigación Publicaciones Proyectos News Cluster

Fenix Cluster Componentes Financiamiento

Fenix es un cluster de cómputo de alto rendimiento. En este cluster se realiza investigación relacionada al estudio de materiales mediante técnicas computacionales que envuelven distintos tópicos dentro de la física, desde dinámica clásica, hasta técnicas de teoría funcional de la densidad tiempo dependiente. El equipo está compuesto de dos racks, ubicados en el campus casana de la Universidad Andrés Bello.

Información general para los usuarios

Accediendo a Fenix

Software disponible

Solicitudes

Solicitudes de cuentas, software o soporte.

Realizar Solicitud

Gran parte de los supercomputadores del mundo usan el sistema operativo **GNU/Linux**.

El ambiente GNU/Linux, editores
y manejo de archivos.

- GNU: es un sistema completo de software libre compatible con UNIX (Richard Stallman, 1983).
- GNU proviene de “GNU no es Unix”, en inglés: GNU’s Not Unix.
- El proyecto GNU: fue concebido en 1983 con la finalidad de retomar el espíritu cooperativo que prevalecía en la comunidad informática en sus comienzos y posibilitar la cooperación eliminando los obstáculos impuestos por los dueños de software privativo.
- La FSF (Free Software Foundation) se fundó en 1985 con el objetivo de recaudar fondos para el desarrollo de GNU.
- En 1991 Linus Torvals programó Linux, un núcleo (kernel) similar a Unix. La combinación de Linux con el sistema GNU formó un sistema operativo completo: el sistema GNU/Linux.

- En la actualidad se usan sistemas GNU/Linux a través de diversas distribuciones: CentOS, Debian, Fedora, Mandriva, OpenSUSE, Red Hat, Ubuntu, entre otras (algunas no son libres, ya que no respetan pautas de distribución de sistemas libres).
- Un sistema operativo GNU/Linux incluye: núcleo, compiladores, editores, procesadores de texto, software de correo, interfaces de línea de comandos (CLI), interfaces gráficas (GUI), bibliotecas, juegos, etc.
- CLI es una herencia de Unix y a pesar de su mala reputación, es la mejor forma de comunicación con el computador. Se dice que las GUI hacen fácil tareas fáciles, mientras que la CLI hace posible tareas difíciles.
- En GNU/Linux, la CLI que nos permite comunicarnos con el computador es la *shell*.

- La *shell* es un programa o intérprete que toma las órdenes introducidas a través del teclado y las entrega al sistema operativo para que sean ejecutadas.
- La *shell* más usada y por defecto en GNU/Linux es **bash**, que significa "Bourne Again Shell" y hace referencia a una versión mejorada de la shell original de Unix, *sh*, escrita por Steve Bourne (1978).
- Cuando usamos GUI, es necesario un emulador de terminal para poder interactuar con la shell. En KDE el emulador se llama *konsole* y en GNOME, *gnome-terminal*.

- En Linux todo es un fichero: los directorios son ficheros, los ficheros son ficheros, los dispositivos son ficheros, etc.
- Los sistemas de ficheros se organizan en una estructura jerárquica de tipo árbol. El nivel más alto del sistema de ficheros es / o directorio raíz.

Example (Linux)

```
/home/pepito/archivo.dat
```

Example (Windows)

```
C:\Users\pepito\archivo.dat
```


A tener en cuenta en GNU/Linux

- Existe un y sólo un ser supremo, que gobierna el kernel de Linux, ese es “**root**”.
- Cada instalación de GNU/Linux tiene un usuario **root**, o algún usuario adicional con permisos de acceso.
- El usuario **root** puede:
 - Destruir los discos.
 - Instalar/Eliminar programas.
 - Añadir/Eliminar usuarios.
 - Borrar lo que nadie más puede borrar.
 - Husmear la máquina, sin embargo mucha información encriptada no es directamente ‘observable’.
 - Casi todo lo que se desee hacer.
- Actualmente, muchas distribuciones GNU/Linux utilizan **sudo**, para que un usuario pueda **ejecutar instrucciones**, como si fuera root.

- Instalación de programas: para poder instalar programas el usuario deber tener acceso como **root** al sistema.

Example (comando sudo / Dist. basada en Debian)

```
sudo 'comando de root'  
sudo ls  
sudo rm algo  
sudo apt-get install algo
```

- El comando **sudo** ejecuta el comando deseado pero con la autorización de **root** (como si **root** mismo lo ejecutara en vez de un simple usuario).
- En este curso utilizaremos distribuciones basadas en Debian GNU/Linux, por lo que **apt** será la herramienta preferida de instalación.

- El comando ***apt-get*** permite instalar paquetes de software en su máquina, desde un repositorio central.
- Este sistema centralizado evita tener que “salir a buscar” un programa en la web, descargarlo, revisar si es seguro e instalarlo manualmente. ***apt-get*** lo hace todo automático.
- Distribuciones como Ubuntu o Debian tienen fácilmente más de 8000 paquetes a su disposición.

Modo de uso utilidad ***apt-get***

```
apt-get [options] command
```

Example

```
>$ sudo apt-get install vim
```

Comandos básicos

| Comando | Descripción |
|--------------|---|
| <i>ls</i> | Listar directorios (-a, -R, -t, -r) |
| <i>pwd</i> | Imprime el directorio de trabajo actual |
| <i>cd</i> | Cambia/Ingresa a un directorio |
| <i>mv</i> | Mueve un archivo o directorio a otro lugar o cambia su nombre |
| <i>cp</i> | Copia un archivo o carpeta (-r) |
| <i>rm</i> | Elimina archivos o directorios (-r) |
| <i>mkdir</i> | Crea un directorio |
| <i>rmdir</i> | Elimina un directorio vacío (rm -rf: para borrar recursivamente un directorio no vacío) |

La mayoría de los comandos posee un manual de ayuda. Para acceder a dicho manual se ejecuta: *man comando*

Example (comando man)

```
>$ man ls  
>$ man cd  
>$ man pwd
```

Más comandos básicos

| Comando | Descripción |
|--------------|---|
| <i>cat</i> | Muestra el contenido ASCII de un archivo. |
| <i>more</i> | Muestra el contenido ASCII de un archivo con paginación. |
| <i>less</i> | Mejora de <i>more</i> . Movimiento en ambas direcciones y más rápido. |
| <i>head</i> | Muestra las primeras líneas de un archivo. |
| <i>tail</i> | Muestra las últimas líneas de un archivo. |
| <i>find</i> | Busca archivos. |
| <i>chmod</i> | Permite cambiar los permisos de acceso de un fichero o directorio. |
| <i>grep</i> | Filtro de contenido. |
| <i>awk</i> | Manipulador de columnas. |
| <i>sed</i> | Manipulador de filas. |

- Existen muchos: los más populares son nano, pico, vim y emacs, entre otros.
- Vim o Vi mejorado es un editor de texto que funciona en la terminal, entre sus características destacan:
 1. Gran flexibilidad en la asignación de comandos, casi completamente programable.
 2. Ambientes de edición y de ejecución separados (escape).
 3. Compatibles con muchos lenguajes y aplicaciones.
- ¿Por qué es necesario aprender VIM?
 1. La mayoría de las investigaciones científicas, necesitan acceso remoto a computadores.
 2. El acceso gráfico, pese a ser posible, es el más lento que existe. Por lo que las utilidades de terminal son una fuente inagotable de mejoras.
 3. Si saben algo de vim, entonces conocen lo básico de vi, que se encuentra en todas las distribuciones GNU/Linux.
 4. Entre otras.

- VIM = Vi IMproved
- La principal característica tanto de VIM como de Vi, consiste en que disponen de **diferentes modos** entre los que se alterna para realizar ciertas operaciones. Otros editores tienen un solo modo en el que se introducen órdenes mediante combinaciones de teclas o interfaces gráficas.
- VIM es un editor hecho por programadores para programadores. Con el fin de facilitar la programación, VIM dispone de un modo "editar, compilar, corregir".
- Existe una larga variedad de comandos en VIM. Observemos VIM en acción.

- Durante el desarrollo del curso, utilizaremos *vim* para las ediciones principales de código, tanto como para las evaluaciones.
- Existen muchos manuales de *vim*, acá dejamos algunos links de interés.
 - Instalar y ejecutar *vimtutor*
 - <https://www.linux.com/learn/vim-101-beginners-guide-vim>
 - <http://www.openvim.com>

Instalando GNU/Linux

El uso de máquinas virtuales se ha expandido mucho en la última década, principalmente por la calidad de los equipos computacionales con los que se trabaja actualmente, en los que es posible correr muchos OS (Operative System) de forma Simultánea, como GNU/Linux, Windows, OS/X, etc.

VMWARE

VMWARE es un software que nos permite correr un sistema operativo dentro de otro sistema operativo. Por ejemplo, nos permite correr GNU/Linux dentro de nuestro computador con Windows.

- Descargue VMWARE Workstation Player.
- Proceda con la instalación en su OS (ej: Windows).
- Descargue la ISO del OS que desea instalar (ej:GNU/Linux Ubuntu).
- Proceda con la instalación.

Un video


<https://www.youtube.com/watch?v=2ykto1mD4pc>


También es posible hoy en día instalar directamente ubuntu dentro de windows 10. Sin embargo en esta versión de Windows también necesita habilitar un ambiente gráfico instalando GWSL (disponible en Microsoft Store).

Es un GNU/Linux más que suficiente para los desafíos de este curso.


Microsoft Store

← Home Gaming Entertainment Productivity Deals

Search  ...


 This product is installed.

Launch ...




Ubuntu 18.04 LTS

Canonical Group Limited • [Developer tools > Servers](#)

★★★★★ 9  Share

Ubuntu 18.04 on Windows allows one to use Ubuntu Terminal and run Ubuntu command line utilities including bash, ssh, git, apt and many more.

[More](#)

 ALL AGES

Wish list

Tanto en windows 10, como en 11 es posible instalar GNU/Linux, una de las formas más simples de instalar WSL en windows es utilizando powershell de microsoft (como administrador), y ejecutar el comando :

Instalando WSL

```
wsl --install
```

Esto instalará todo lo básico, que podremos ir complementando.

A terminal window with a white title bar and standard window controls. The title bar text is 'jperaltac@DESKTOP-JJBP67L: ~'. The terminal area has a black background with a green prompt 'jperaltac@DESKTOP-JJBP67L:~\$' at the top left. A vertical scrollbar is on the right side of the terminal area.

```
jperaltac@DESKTOP-JJBP67L: ~  
jperaltac@DESKTOP-JJBP67L:~$
```

Google Colab



- Esta plataforma es para programar en Python.
- No tiene relación con los terminales de GNU/Linux ni *bash*.
- Pese a lo anterior es bueno ir habilitando una cuenta para que más adelante pueda programar en este sistema.
- Utiliza Jupyter, y enre sus ventajas es permitir de forma gratuita hacer uso de cómputo en GPU.

Servicio de colaboración en programación brindado por google. La web oficial ese encuentra en:

<https://colab.research.google.com/>

Welcome To Colaboratory

File Edit View Insert Runtime Tools Help

Share Settings Profile

Table of contents

- Getting started
- Data science
- Machine learning
- More Resources
- Machine Learning Examples
- Section

+ Code + Text Copy to Drive

Connect Editing

What is Colaboratory?

Colaboratory, or "Colab" for short, allows you to write and execute Python in your browser, with

- Zero configuration required
- Free access to GPUs
- Easy sharing

Whether you're a **student**, a **data scientist** or an **AI researcher**, Colab can make your work easier. Watch [Introduction to Colab](#) to learn more, or just get started below!

Getting started

The document you are reading is not a static web page, but an interactive environment called a **Colab notebook** that lets you write and execute code.

For example, here is a **code cell** with a short Python script that computes a value, stores it in a variable, and prints the result:

```
[ ] 1 seconds_in_a_day = 24 * 60 * 60
    2 seconds_in_a_day
```

86400

To execute the code in the above cell, select it with a click and then either press the play button to the left of the code, or use the keyboard shortcut "Command/Ctrl+Enter". To edit the code, just click the cell and start editing.

The screenshot displays the Google Colaboratory web interface. A 'File' menu is open, showing options for creating, opening, and saving notebooks. The background notebook, titled 'What is Colaboratory?', contains introductory text and a code cell with a Python script for calculating seconds in a day.

Welcome To Colaboratory

File Edit View Insert Runtime Tools Help

- New notebook
- Open notebook... Ctrl+O
- Upload notebook...
- Rename...
- Move to trash
- Save a copy in Drive...
- Save a copy as a GitHub Gist...
- Save a copy in GitHub...
- Save Ctrl+S
- Save and pin revision Ctrl+M S
- Revision history
- Download .ipynb
- Download .py
- Update Drive preview
- Print Ctrl+P

Code + Text Copy to Drive

What is Colaboratory?

Colaboratory, or "Colab" for short, allows you to write and execute code in a Jupyter notebook in the cloud.

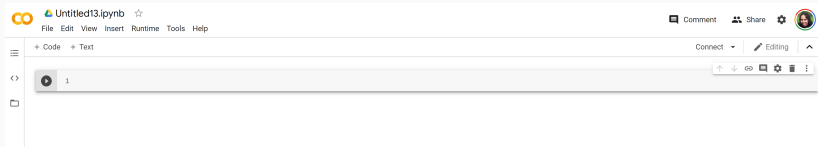
- Zero configuration required
- Free access to GPUs
- Easy sharing

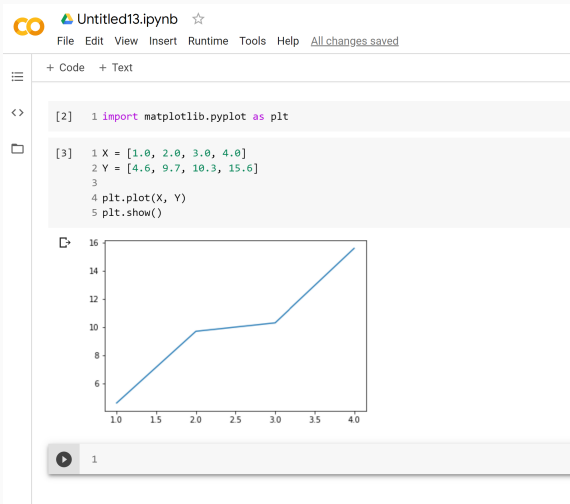
Whether you're a **student**, a **data scientist** or an **AI researcher**, you can get started below!

Getting started

The document you are reading is not a static web page, but a Jupyter notebook. For example, here is a **code cell** with a short Python script

```
[ ] 1 seconds_in_a_day = 24 * 60 * 60
     2 seconds_in_a_day
```





Algo de práctica **para hoy**

- Instale alguna versión de terminal de GNU/Linux. Por ejemplo
 - WSL en windows
 - Un OS completo con VMWARE
 - Instale un sistema operativo junto al existente (the hard way)

Algo de práctica **para esta semana**

- Cree los siguientes directorios
 - Archivos/experimento1/datos
 - Fotografias/
 - Documentos/personales/banco
 - Documentos/personales/otros
- Genere archivos de texto en cada uno de estos directorios, puede utilizar el editor que usted prefiera.
- Elimine y renombre archivos, a su gusto.
- Mueva archivos entre distintos directorios.

Algo de práctica **para hoy** en términos de programación.

- Tome su telefono celular
- Instale el juego Lightbot Hour
- Consigue llegar al nivel 3!, No es necesario que supere el nivel 3 :).

Fin