

# Informática I (2023-24)

Grado en Ingeniería de Sistemas Audiovisuales y Multimedia (URJC)

Jesús M. González Barahona, Ángel Madridano Carrasco, Sergio  
Montes León

<http://cursoprogram.gitlab.io>

Universidad Rey Juan Carlos

September 19, 2023



Universidad  
Rey Juan Carlos





# Guión de la sesión

A lo largo de esta sesión vamos a familiarizarnos con el sistema operativo Linux y con el entorno de desarrollo integrado PyCharm:

- Creación de cuenta de laboratorio Linux.
- Presentación de Linux-Ubuntu y del entorno gráfico.
- Introducción a Shell de Linux: comandos básicos.
- Intérprete de Python en Linux: python3.
- IDE PyCharm: instalación, creación de proyectos, ejecución de código.

# Pero...¿Por qué Python?

## Utilidades de python: Servidor WEB

- Permite crear un servidor HTTP con una única línea de código.

```
python3 -m http.server
```

- Probar localmente sitios webs y aplicaciones web en navegador local.
- Compartir archivos localmente.
- Ejecutar páginas webs simples sin necesidad de configurar un servidor.

# Creación de cuentas de usuarios.

- Es necesario disponer de una cuenta de usuario en los laboratorios Linux de la URJC.

<https://labs.eif.urjc.es>



- Los alumnos nuevos deberán crear una cuenta de usuario:
  - 1 CUENTAS DE LINUX.**
  - 2 FAQ Laboratorios Docentes Linux.**
  - 3 Crear Cuenta de Linux.**

## Renovación de cuentas de usuarios.

- En el caso de los repetidores, es necesario que renueven su cuenta de usuario.

<https://labs.eif.urjc.es>

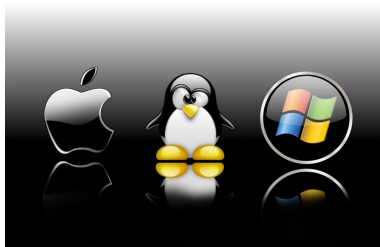


- Los alumnos nuevos deberán crear una cuenta de usuario:
  - ① **CUENTAS DE LINUX.**
  - ② **Renovar Cuenta de Linux.**

# Introducción a Linux

## ¿Por qué LINUX?

- Linux es un sistema operativo de código abierto diseñado para ser eficiente, flexible y versátil.
- Estable y seguro.
- Emplea Shell, interfaz de línea de comandos. Útil para tareas de programación.
- Ofrece herramientas de desarrollo potentes y gratuitas (compiladores, IDEs, etc).





# Introducción a Linux

Está formado por:

- **Núcleo(Kernel):**

- Corazón del Sistema Operativo.
- Gestiona los recursos hardware (CPU, memoria, discos, etc).
- Proporciona interfaz entre el HW y las aplicaciones.

- **Shell:**

- Interfaz para interactuar con el sistema a través de comandos.
- Los más comunes son Bash y Zsh.

- **Sistema de Archivos:**

- Estructura jerárquica de directorios del sistema de archivos Unix.
- El sistema de archivos organiza los archivos y directorios en una estructura de árbol.
- Los directorios importantes incluyen `"/home"` para usuarios, `"/bin"` para programas esenciales, `"/etc"` para configuraciones y más.

- **Utilidades y Aplicaciones.**

# Introducción a Linux

A la hora de trabajar con Linux, es conveniente familiarizarse con algunos conceptos básicos:

- Ubuntu como distribución de Linux.
- Interfaz gráfica de usuario(GNOME).
- Escritorio Ubuntu: Ventanas, lanzamiento de aplicaciones, entrada y salida.



Guía Escritorio Ubuntu

# Introducción a Linux: Ubuntu y GNOME

Ubuntu es una distribución de Linux basada en Debian, diseñada para ser fácil de usar. Algunos puntos clave:

- 1 Utiliza el entorno de escritorio GNOME de forma predeterminada, lo que proporciona una GUI intuitiva.
- 2 Estabilidad y seguridad a largo plazo gracias a las actualizaciones automáticas durante 5 años (para versiones de escritorio).
- 3 Amplia gama de software libre incluido por defecto.

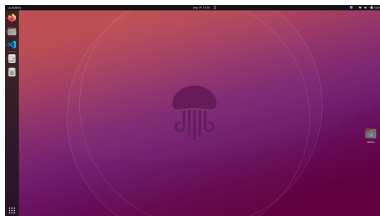
GNOME es el entorno de escritorio predeterminado en Ubuntu y la mayoría de distribuciones GNU/Linux. Algunas características principales:

- Interfaz gráfica intuitiva basada en ventanas, iconos y menús.
- Barra con acceso rápido a aplicaciones, configuración y notificaciones.

# Escritorio Ubuntu

El escritorio de Ubuntu proporciona una interfaz gráfica intuitiva basada en ventanas, iconos y menú similares a otros sistemas operativos populares. Los elementos principales del escritorio incluyen:

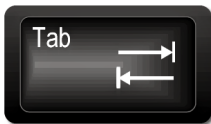
- **Barra superior:** Contiene accesos a aplicaciones, configuración del sistema y notificaciones.
- **Barra lateral:** Acceso rápido a aplicaciones y archivos más utilizados.
- **Menú Actividades:** Acceso a todas las aplicaciones abiertas y accesibles con un clic.
- **Menú Energía:** Permite apagar, reiniciar o bloquear el sistema.



# Shell: Línea de comandos de Linux

La shell o terminal es la interfaz de línea de comandos de Linux que permite interactuar con el sistema a través de comandos textuales.

- Ideal para administración y desarrollo.
- Permite un control preciso y rápido.
- **Sintaxis:** "Comando -opciones -argumento". Ejemplo: "ls -l".
- **Ayuda:** "Comando -help" o "man comando" muestra ayuda. Ejemplo: "man ls". Ejemplo: "ls -h".
- **Historial y Autocompletado:** Las flechas arriba/abajo permiten acceder al historial de comandos. Tabulador completa nombres de archivos y directorios.



gitlab.com

# Shell: Comandos básicos

## Comandos básicos de Shell

- **pwd**: Muestra el directorio de trabajo actual.
- **ls**: Lista los archivos y directorios del directorio actual.
  - **ls -l**: Muestra detalles adicionales como tamaño, permisos, fecha de modificación, etc.
  - **ls -a**: Muestra todos los archivos incluyendo los ocultos.
- **cd**: Cambia el directorio de trabajo actual.
  - **cd [Directorio]**: Cambia al directorio especificado.
  - **cd ..**: Retrocede un nivel en la estructura de directorios.
- **mkdir**: Crea un nuevo directorio.
  - **mkdir -p**: Crea directorios de forma recursiva, incluyendo directorios padres si no existen.

# Shell: Continuación Comandos básico

- **rm**: Elimina archivos.
  - **rm -r**: Elimina directorios de forma recursiva, incluyendo su contenido.
  - **rm -i**: Pregunta antes de eliminar cada archivo para confirmar.
  - **rm -f**: Elimina archivos sin confirmar.
- **cp**: Copia archivos y directorios.
  - **cp -r**: Copia directorios de forma recursiva, incluyendo su contenido.
- **mv**: Mueve o renombra archivos y directorios.
  - **mv -i**: Pregunta antes de sobrescribir.
- **cat**: Muestra el contenido de un archivo.
- **nano o vi**: Abre un editor de texto en la terminal.
- **grep**: Busca patrones de texto en archivos.
  - **grep -i**: Realiza una búsqueda sin distinción entre mayúsculas y minúsculas.

# Shell: Continuación Comandos básico

- **sudo**: Ejecuta comandos con privilegios de superusuario.
- **chmod**: Cambia los permisos de archivos y directorios.
- **ps**: Muestra procesos en ejecución.
  - **-e**: Muestra todos los procesos.
- **history**: Muestra el historial de comandos.
- **df** : Muestra el espacio en disco disponible.
- **du**: Muestra el espacio utilizado por archivos y directorios.
- **wget o curl**: Descarga archivos desde Internet.



# Shell: Atajos de teclado

## Atajos de Teclado

- **Ctrl + Shift + C / Ctrl + Shift + V**: Copia y pega texto en terminales que no admiten los atajos estándar Ctrl + C y Ctrl + V.
- **Ctrl + C**: Interrumpe la ejecución de un comando en curso o un programa.
- **Ctrl + D**: Cierra la terminal o, si no hay entrada de texto, indica el final de la entrada estándar.
- **Ctrl + Z**: Detiene un proceso en segundo plano y lo coloca en estado suspendido.
- **Ctrl + L o clear**: Limpia la pantalla, lo que es útil para mantener tu terminal ordenada.
- **Ctrl + A**: Mueve el cursor al principio de la línea de comando.
- **Ctrl + E**: Mueve el cursor al final de la línea de comando.
- **Ctrl + R**: Inicia una búsqueda inversa en el historial de comandos.
- **Tab**: Autocompleta comandos, nombres de archivos y directorios.
- **Flecha Arriba/Flecha Abajo**: Navega por el historial de comandos previos.

# Shell: Ejercicio

- ➊ Crea un nuevo directorio llamado "MiDirectorio" en tu directorio de inicio.
- ➋ Entra en ese directorio.
- ➌ Crea un archivo llamado "MiArchivoenLinux.txt" dentro de "DirectorioOrigen".
- ➍ Escribe un mensaje en él.
- ➎ Visualiza el contenido dle archivo en la terminal.
- ➏ Crea un segundo directorio llamado "DirectorioCopiasSeguridad" en tu directorio de inicio.
- ➐ Copia "MiArchivoenLinux.txt" de "MiDirectorio" a "CopiasSeguridad".
- ➑ Crea un nuevo archivo llamado "NuevoArchivo.txt" en "MiDirectorio".
- ➒ Actualiza el archivo introduciendo algún dato abriéndolo con la aplicación *gedit*.
- ➓ Visualiza en terminal este nuevo fichero.
- ➔ Mueve "NuevoArchivo.txt" de "MiDirectorio" a "CopiasSeguridad".
- ➕ Elimina "MiArchivoenLinux.txt" de "MiDirectorio".
- ➖ Verifica que "MiDirectorio" esté vacío.
- ➗ Muestra el contenido de "CopiasSeguridad" para confirmar que contiene "MiArchivoenLinux.txt" y "NuevoArchivo.txt".

# Python desde Shell

- Python puede ser ejecutado desde terminal:

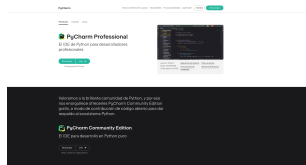
`python3`

- Es necesario importar librerías para poder usar funciones en intérprete. Por ejemplo, `import math`, `math.sqrt()` para el cálculo de la raíz cuadrada de un número.
- Es posible salir del interprete pulsando **ctrl + D** o ejecutando el comando **exit()**
- Ejercicios sencillos para explorar Python en terminal:
  - 1 Imprime un mensaje de saludo en la pantalla.
  - 2 Realizar operaciones de suma, resta, multiplicación y división. Usa Python como Calculadora.
  - 3 Calcula la longitud de una cadena de texto.
  - 4 Calcula la raíz cuadrada de 125.

# PyCharm nuestro Entorno de Desarrollo Integrado

PyCharm es un IDE (Entorno de Desarrollo Integrado) altamente potente y amigable, que nos va a permitir programar en python, depurar el código y ejecutar nuestros programas y aplicaciones en una misma ventana.

- PyCharm es gratuito y está disponible tanto para Windows, como para macOS, como para Linux.
- Se puede descargar en: [Descargar PyCharm](#)
- **Importante:** Instalar el **PyCharm Community Edition** el Professional no es gratuito (Parte inferior).



Guía de Inicio de PyCharm  
Mis primeros pasos

# Ejercicios

Usando el IDE de PyCharm, crea un fichero para cada uno de los problemas siguientes:

- 1 En *operacionesMatematicas.py*: Realiza operaciones matemáticas con 2 números que elijas (Suma, resta, multiplicación, división, exponencial y raíz cuadrada de cada uno de los números).
- 2 En *conversionTipos.py*: Muestra el valor ordinal de un caracter como entero y como coma flotante.
- 3 En *notaMedia.py*: Calcula el promedio de tres calificaciones dadas. Muestra el promedio redondeado a dos decimales.
- 4 Modifica el script anterior, para que las notas las introduzca el usuario por teclado.
- 5 En *areaPoligono.py*: Calcula el área de un polígono regular, a partir del número de lados y la longitud del lado que elijas. Muestra el perímetro, apotema y área del polígono.