

# Introducción al mundo IT

Saturday, August 17, 2024 6:34 PM

## Historia de la informática

- Ábaco en mesopotamia y china.
- Blaise Pascal invento la primera calculadora mecánica.
- Charles Babbage invento la Máquina Diferencial y la **Máquina Analítica (precursora de la computadora)**
- Ada Lovelace, fue la primera programadora conocida, ayudo a Babbage
- El término inteligencia artificial se acuñó en 1956 en la conferencia de Dartmouth, ahí comenzó su investigación formal.
- John McCarthy (no Oconor) es considerado el padre de la IA.
- En 1990 surgieron las primeras computadoras personales como la Altair 8800, la Apple 1 y el Apple 2.
- En 1960, ARPANET (Advance Research Project Agency Network), una red de comunicación del gobierno de USA sentó las bases de lo que hoy es la internet, usaba el protocolo TCP/IP.
- En 1989 Tim Berners-Lee creó la Wold Wide Web.
- Microsoft fue fundada en 1975 por Bill Gates y Paul Allen.
- Google fue fundada por Larry Page y Sergey Brin en 1998.

## Hardware

### Componentes esenciales

- Procesador (CPU): es el cerebro de la computadora, realiza las operaciones.
- Memoria RAM: Random Access Memory, almacenamiento rapido y temporal.
- Unidad de almacenamiento (DD, SSD): almacenamiento lento y permanente.
- Placa Base, Tarjeta gráfica.

### Servidores

- Estan diseñados para proporcionar servicios, recursos o funcionalidades a otros dispositivos o usuarios en una red.
- Están optimizados para tareas de almacenamiento, procesamiento y gestion de grandes cantidades de datos y tráfico de red.
- Pueden ser **servidores web, servidores de bases de datos, servidores de correo electrónico, servidores de archivos, etc.**
- Suelen operar de forma continua y están configurados para ser seguros y confiables.

### Sistemas embebidos

- Son sistemas informaticos diseñados para realizar funciones específicas dentro de dispositivos más grandes o sistemas integrados.
- Tienen recursos limitados en comparación con las computadoras de escritorio o servidores, pero están optimizados para tareas específicas.
- Se utilizan en una amplia gama de aplicaciones, como dispositivos médicos, electrodomésticos inteligentes, sistemas de control industrial, automóviles, etc.
- Suelen tener un diseño compacto, consumo de energía eficiente y pueden operar en condiciones ambientales adversas.

## Software

La parte intangible de un sistema informatico, es el conjunto de programas, instrucciones y datos

necesarios para realizar diversas tareas en una computadora.

### **Lenguaje de Máquina**

Es el conjunto de instrucciones directamente interpretable por el hardware de una computadora. Consiste en una secuencia de códigos binarios, representados por 0 y 1, que le indican al hardware operaciones elementales como sumas, restar, mover datos, entre otros.

Cada procesador tiene su propio conjunto de instrucciones en lenguaje máquina, conocido como conjunto de instrucciones de arquitectura (ISA). Estas instrucciones se ejecutan en la CPU y controlan el flujo de datos y las operaciones realizadas por la computadora.

Aunque es el lenguaje más básico y difícil de entender para los humanos, todos los programas de software eventualmente se traducen a lenguaje de máquina para poder ser ejecutados por el procesador.

### **Lenguaje de Ensamblador**

- También conocido como Assembly.
- Proporciona una representación simbólica de las instrucciones de máquina.
- Es más legible que el lenguaje máquina.
- Es como un traductor entre lenguaje máquina y lenguaje humano
- Cada instrucción en lenguaje ensamblador representa una instrucción del lenguaje máquina correspondiente.
- Ofrece mayor control sobre el hardware y la optimización de programas para tareas específicas.

### **Sistemas operativos**

- Corazón de cualquier dispositivo informático.
- Intermediario entre Hardware y software.
- Provee una interfaz usuario-máquina, y servicios como gestión de archivos, seguridad y multitarea.

## **Servidores**

- Actúan como el corazón de las redes informáticas, proporcionando servicios, almacenamiento y recursos a otras computadoras en la red.
- Un servidor es una computadora dedicada a proporcionar servicios, recursos y datos a otras computadoras en una red (cliente).
- Diseñados para administrar múltiples solicitudes de manera simultánea.

### **Servidores Web**

- Alojan y entregan sitios web y aplicaciones web a través de Internet.
- Utilizan protocolos como HTTP y HTTPS para comunicarse con navegadores web.

### **Servidores de correo electrónico**

- Gestionan la entrega y recepción de emails.
- Utilizan protocolos como SMTP, POP3 e IMAP para enviar, recibir y almacenar emails.

**Servidores de Base de Datos:** Almacenan y gestionan datos estructurados, proporcionan acceso concurrente a múltiples usuarios y apps.

**Servidores de archivos:** almacenan y comparten archivos en una red. Permiten a los usuarios acceder y compartir archivos de forma centralizada.

**Servidores de aplicaciones:** Los servidores de aplicaciones ejecutan y gestionan

aplicaciones empresariales y servicios en una red. Proporcionan entornos de ejecución para aplicaciones web y empresariales, facilitando el desarrollo, implementación y administración de software.

Los servidores desempeñan funciones en una red como el almacenamiento y distribución de archivos, la gestión de bases de datos, la provisión de servicios de email y hospedaje de sitios web. También pueden **proporcionar seguridad**, control de acceso, monitoreo y administración de la red.

CLI: Command Line Interface

GUI: Gráfico User Interface (iconos, ventanas, menús)

NUI: Native User Interface (gestos, voz y otros)

## Terminal

### Consola, terminal y CLI

**Consola:** Es el conjunto de teclado y pantalla, físico o virtual (ventana de consola)

**Terminal:** Es el software o la aplicación que proporciona una interfaz de línea de comandos para interactuar con el OS. En sistemas Linux, la terminal es la principal forma de acceder a la CLI.

**CLI:** Interfaz que permite al usuario interactuar con un programa o sistema, emitiendo comandos de texto simples. La CLI se utiliza a menudo a través de la terminal.

Ventajas de la terminal

- Eficiencia
- Automatización
- Acceso Remoto
- Potencia y flexibilidad

## Bash

- **directorio-interno/** ---> Esto es un directorio, porque tiene el / al final
- **mkdir:** para crear directorio
- **touch:** para crear un archivo
- **rm:** para borrar un archivo
- **mv:** mover archivo -> Ejemplo: **mv mi.txt directorio-interno/**
- **cp:** copiar archivo
- **pwd:** path absoluto de donde estoy
- **ls | grep "no":** (grep: patrón) lista todos los archivos que incluyen "no"
- **ls > listado.txt:** crea un archivo de texto con el nombre de todos los archivos que listaría el comando ls.
- **cat archivo.txt:** permite leer el texto que hay dentro del archivo.
- **chmod:** cambia los permisos de acceso de archivos y directorios
- **chown:** cambia al propietario y al grupo de archivos y subdirectorios.
- **find:** busca archivos y directorios

## Bash VS Linux

- mkdir - mkdir
- clear - cls
- touch archivo.txt - New-Item archivo.txt -ItemType File
- rm archivo.txt - del archivo.txt
- rm folder - rmdir folder
- ls - dir
- \_ - ren renombrar.txt renombrado.txt
- cp - copy original.txt copia.txt

## Git

Git es el VCS (version control manager) mas conocido.

### Configuración inicial

git config --global user.name

git config --global user.email

Las credenciales seran usadas de manera global, en el caso de trabajar con diferentes empresas, debes ingresar las credenciales correspondientes.

### Estados de archivos

1. Untracked: Existen en el directorio de trabajo pero no han sido agregados al staging area. Git no rastrea los cambios en estos archivos.
2. Unmodified: Archivos que no han sido modificados desde el ultimo commit.
3. Modified: Archivos que se han modificado pero aun no se agregan al area de staging.
4. Staged: Archivos añadidos al area de preparación (staging) mediante el comando git add.

### Comandos Basicos

1. # git init: inicia git
2. # git add: agrega repositorios al stage
3. # git commit: confirma los cambios realizados en el repositorio creando un nuevo punto en la historia del proyecto
4. # git config: se utiliza para configurar variables de entorno especificas de Git como usuario y email. Es util para personalizar la configuracion de git en un proyecto especifico.
5. # git rm --cached: elimina el archivo del staging.
6. # git restore: elimina los cambios realizados restaurando el archivo a la version del ultimo commit.
7. # git log --oneline
8. # git status -s
9. # git diff: funciona para ver las diferencias entre archivos que aun no estan en el stage.

### Ignorar y borrar archivos del repositorio

- # git ignore: ignora los archivos que no deseamos incluir en el repositorio.
- # git rm archivo.txt: elimina archivos que han sido agregados al repositorio, pero que no queremos incluir anymore.

### Otros comandos

- # git config --global core.editor "code --wait"
- # git config --global core.autocrlf
  - Input (mac/linux): configuracion para que los saltos de linea sean como linux
  - true (windows): configuracion para que los saltos de linea sean como windows
  - false (windows): configuracion para que los saltos de linea sean como linux

(recomendado)

### **Ramas Git**

- Rama principal (master/main)
- Ramas de características (feature branches)
- Ramas de corrección (bugfix branches)
- Ramas de version (release branches)
- Ramas de desarrollo (development branches)