



El futuro digital  
es de todos

MinTIC

Mayo 3 de 2021

Rogelio Orlando Beltrán Castro



# Presentación

- ❑ Rogerio Orlando Beltrán Castro.
- ❑ Ingeniero de Sistemas
- ❑ Especialista en Tecnología Avanzadas de Software
- ❑ Candidato a Magister en Gestión, Aplicación y Desarrollo de Software.
- ❑ Candidato a Magister en Dirección de Centros Educativos.
  
- ❑ Ingeniero de desarrollo de software desde el año 1996.
- ❑ Docente Universitario desde el año 2004

# INTRODUCCIÓN A LOS CONCEPTOS BASICOS DE PROGRAMACIÓN DE COMPUTADORES

Se pueden utilizar muchos lenguajes para programar una computadora. El más básico es el lenguaje de maquina, una colección de instrucciones muy detallada que controla la circuiteria interna de la maquina. Este es el dialecto natural de la maquina. Muy pocos programas se escriben actualmente en lenguaje de maquina por dos razones importantes: primero, porque el lenguaje de maquina es muy incomodo para trabajar y segundo por que la mayoría de las maquinas se pide programar en diversos tipos de lenguajes, que son lenguajes de alto nivel, cuyas instrucciones son más compatibles con los lenguajes y la forma de pensar humanos como lo es el lenguaje c que además es de propósito general.

Debido a que los programas diseñados en este lenguaje se pueden ejecutar en cualquier maquina , casi sin modificaciones. Por tanto el uso del lenguaje de alto nivel ofrece tres ventajas importantes, sencillez, uniformidad y portabilidad.

# QUE ES UN LENGUAJE DE PROGRAMACIÓN

Sistema de símbolos y reglas que permite la construcción de programas con los que la computadora puede operar así como resolver problemas de manera eficaz.

Estos contienen un conjunto de instrucciones que nos permiten realizar operaciones de entrada / salida, calculo, manipulación de textos, lógica / comparación y almacenamiento / recuperación.

Los lenguajes de programación se clasifican en:

- Lenguaje Máquina: Son aquellos cuyas instrucciones son directamente entendibles por la computadora y no necesitan traducción posterior para que la CPU pueda comprender y ejecutar el programa. Las instrucciones en lenguaje maquina se expresan en términos de la unidad de memoria más pequeña el bit (dígito binario 0 ó 1).
- Lenguaje de Bajo Nivel (Ensamblador): En este lenguaje las instrucciones se escriben en códigos alfabéticos conocidos como mnemotécnicos para las operaciones y direcciones simbólicas.
- Lenguaje de Alto Nivel: Los lenguajes de programación de alto nivel (BASIC, pascal, cobol, fortran, etc.) son aquellos en los que las instrucciones o sentencias a la computadora son escritas con palabras similares a los lenguajes humanos (en general en ingles), lo que facilita la escritura y comprensión del programa.



# ALGORITMOS

La palabra algoritmo se deriva de la traducción al latín de la palabra árabe alkhwarizmi, nombre de un matemático y astrónomo árabe que escribió un tratado sobre manipulación de números y ecuaciones en el siglo IX.

La solución a cualquier problema de cómputo involucra la ejecución de una serie de acciones en orden específico. Un procedimiento para resolver un problema en términos de: a) Las acciones a ejecutarse y b) el orden en el cual estas acciones deben ejecutarse se llama algoritmo.

Un ejemplo de un algoritmo para llegar a la escuela

1. Salir de la cama
2. Quitarse la pijamas
3. Darse un baño
4. Vestirse
5. Desayunar

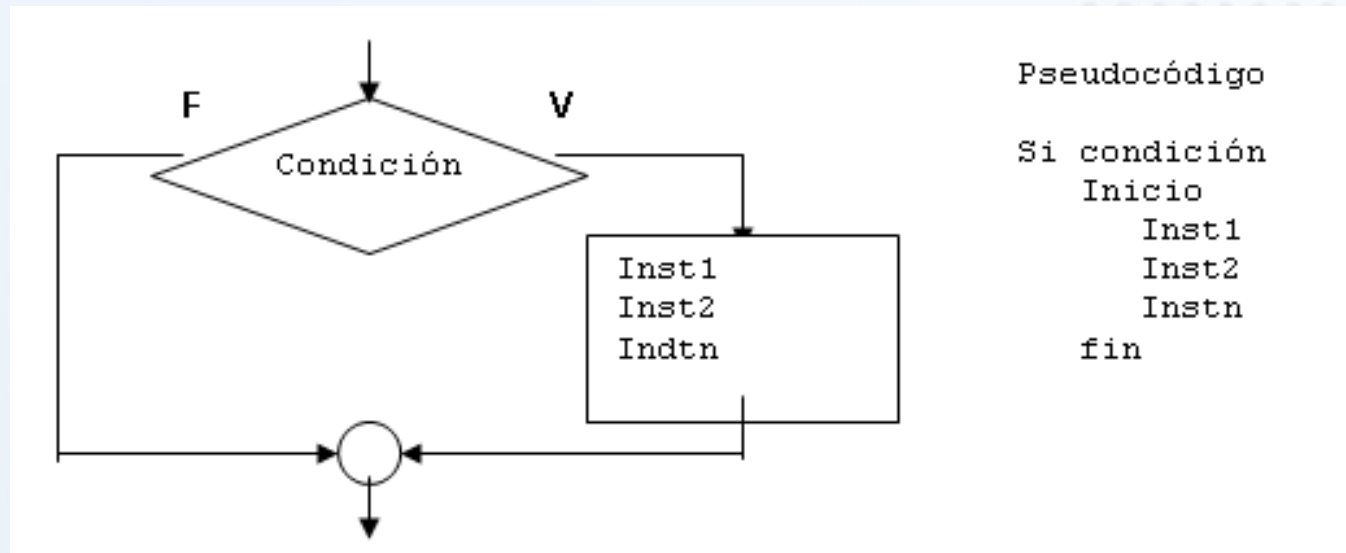
# ESTRUCTURAS CONDICIONALES

Existen tres tipos de estructuras de control selectivas, estas se basan en una condición o en una opción para decidir la parte del programa por el que pasará.

- a) Simple
- b) Doble o compuesta
- c) Múltiple

# ESTRUCTURAS CONDICIONALES

Selectiva simple.- evalúa una condición, si esta es verdadera ejecuta la acción o acciones especificadas, si es falsa no realiza ninguna acción.

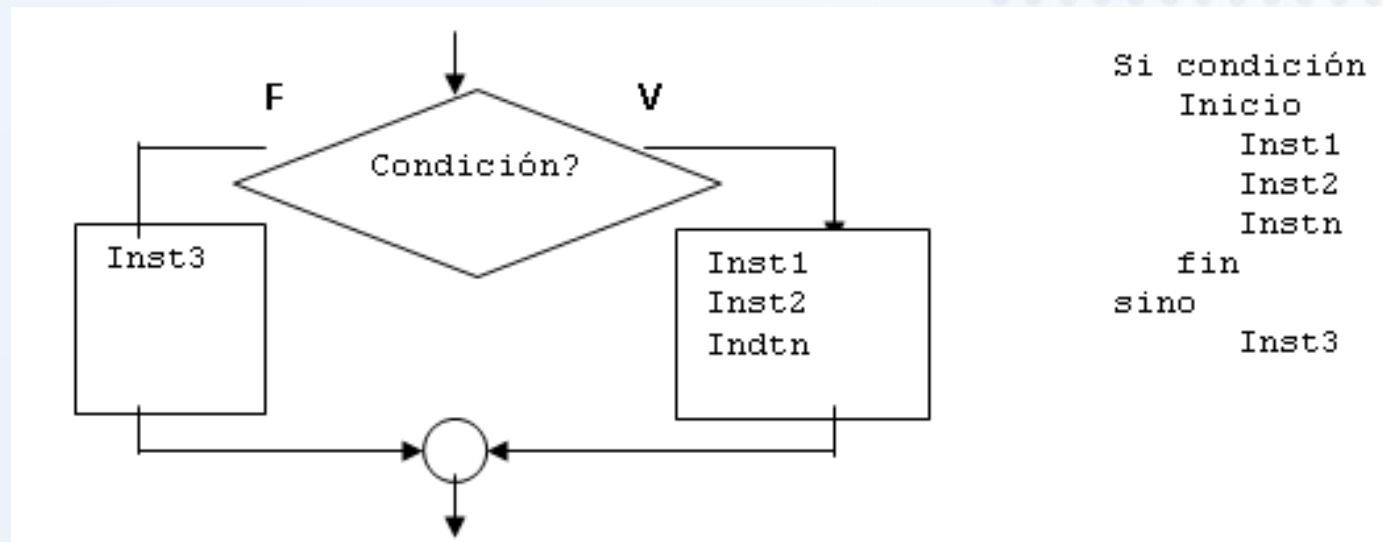


Pseudocódigo

```
Si condición  
  Inicio  
    Inst1  
    Inst2  
    Instn  
  fin
```

# ESTRUCTURAS CONDICIONALES

**Selectiva doble o compuesta.-** evalúa una condición, si esta es verdadera ejecuta la acción o acciones especificadas, si es falsa ejecuta otra acción o acciones.





# ESTRUCTURAS DE BUCLE

Los bucles son estructuras que permiten ejecutar partes del código de forma repetida mientras se cumpla una condición.

Esta condición puede ser simple o compuesta de otras condiciones unidas por operadores lógicos.

# ESTRUCTURAS DE BUCLE

Sentencia / Bucle While

Su sintaxis es:

```
while (condición) sentencia;
```

Con esta sentencia se controla la condición antes de entrar en el bucle. Si ésta no se cumple, el programa no entrará en el bucle.

- + Naturalmente, si en el interior del bucle hay más de una sentencia, éstas deberán ir entre llaves para que se ejecuten como un bloque.

# ESTRUCTURAS DE BUCLE

Sentencia / Bucle DO...WHILE

Su sintaxis es:

```
do {  
sentencia1;  
sentencia2;  
} while (condición);
```

Con esta sentencia se controla la condición al final del bucle. Si ésta se cumple, el programa vuelve a ejecutar las sentencias del bucle.

La única diferencia entre las sentencias while y do...while es que con la segunda el cuerpo del bucle se ejecutará por lo menos una vez.

# ESTRUCTURAS DE BUCLE

Sentencia / Bucle For

Su sintaxis es:

```
for (inicialización;condición;incremento) {  
sentencia1;  
sentencia2;  
}
```

La inicialización indica una variable (variable de control) que condiciona la repetición del bucle. Si hay más, van separadas por comas:

```
for(a=1,b=100;a!=b;a++,b--){ ... }
```

# CARACTERISTICAS DE LOS LENGUAJES

La popularidad de un lenguaje de programación depende de las funcionalidades y utilidades que proporcione a los programadores. Las características que debe tener un lenguaje de programación para destacar son las siguientes:

- Simplicidad: el lenguaje debe ofrecer conceptos claros y simples que faciliten su aprendizaje y aplicación, de manera que sea sencillo de comprender y mantener. La simplicidad no significa que se le pueda restar el poder óptimo de funcionamiento.
- Naturalidad: se refiere a que su aplicación en el área para la que fue diseñado debe hacerse de forma natural, proporcionando operadores, estructuras y sintaxis para que los operadores trabajen eficientemente.
- Abstracción: consiste en la capacidad de definir y utilizar estructuras u operaciones complicadas ignorando algunos detalles. Esto influye en la capacidad de escritura.
- Eficiencia: los lenguajes de programación deben traducirse y ejecutarse eficientemente para no ocupar demasiado espacio en la memoria ni gastar mucho tiempo.
- Estructuración: permite que los programadores escriban sus códigos de acuerdo con los conceptos de programación estructurada, para evitar la creación de errores.
- Compacidad: con esta característica es posible expresar las operaciones con concisión, sin tener que escribir demasiados detalles.
- Localidad: se refiere a que los códigos se concentran en la parte del programa con la cual se está trabajando en un momento determinado



# CARACTERISTICAS DE LOS LENGUAJES

- Java: lenguaje multiuso que se ajusta eficientemente al desarrollo web.
- Go: es de propósito general y ofrece un lenguaje flexible que facilita la creación de aplicaciones.
- Ruby on Rails: permite diseñar aplicaciones web rápidamente.
- Python: funciona en una amplia variedad de contextos y en la web tiene ventajas técnicas.
- JavaScript: se ubica del lado del cliente y se puede extender al servidor para distintas funciones.

# Python

Python es uno de los lenguajes de programación más usados actualmente y la tendencia sigue al alza. Lo tiene todo: es de código abierto, una sintaxis sencilla y fácil de entender, por lo que ahorra tiempo y recursos. Es uno de los mejores para iniciarse en el mundo de la programación.

Python es un lenguaje versátil que puede tener múltiples aplicaciones. Una de ellas, la Inteligencia Artificial, gracias a bibliotecas como Keras o TensorFlow. También puede ser de utilidad para aplicaciones de Big Data, gracias a bibliotecas de procesamiento de datos.

Este lenguaje de programación también se emplea en el desarrollo web, sobre todo, gracias a sus frameworks Django o Flask. Por poner algunos ejemplos, los sitios web de SemRush o Reddit están desarrollados con Python.

# Java

Java es un lenguaje de programación de propósito general, orientado a objetos y diseñado para tener las dependencias de implementación mínimas posibles. Con este lenguaje de programación se pueden crear aplicaciones y procesos en múltiples dispositivos.

Su ámbito de aplicación es amplísimo, por lo que permite crear software para dispositivos móviles, terminales de venta, cajeros automáticos, IoT (internet de las cosas), además de páginas web.

Java fue el lenguaje por defecto para desarrollar aplicaciones para Android y, aunque Kotlin está ganando enteros día a día, Java sigue siendo muy necesario.

Respecto a los juegos móviles, los primeros estaban desarrollados con Java. Hoy en día, Java sigue teniendo un papel importante en el mundo de los videojuegos, gracias a frameworks como JMonkeyEngine o Ardor 3D.

# JavaScript

JavaScript es un lenguaje de programación interpretado, orientado a objetos y que se utiliza principalmente en la forma del lado del cliente.

Este lenguaje de programación sirve para todo: aplicaciones web, servidores, aplicaciones móviles... Su peculiaridad es que no necesita compilación ninguna, ya que es el propio navegador quién lee el código y realiza las acciones que le indica. Por este motivo, es uno de los lenguajes para crear páginas web cuando se quieren plantear elementos interactivos o más visuales.

Sin embargo, gracias a las prestaciones de HTML5 y las librerías de gráficos 2D y 3D, Javascript tiene también un papel relevante para el desarrollo de videojuegos, especialmente, si éste se ejecuta desde el navegador.



# C#

C# es un lenguaje que sigue apareciendo en los listados de lenguajes de programación más usados. Creado por Microsoft, está presente en entornos empresariales, como instituciones gubernamentales, entidades bancarias o médicas.

Aunque más allá de este tipo de aplicaciones, tiene también usos muy versátiles: internet de las cosas, desarrollo de videojuegos, web o aplicaciones móviles. Para el desarrollo web, puedes utilizar el framework ASP.NET o la herramienta Xamarin para desarrollar aplicaciones nativas para Android e iOS.

Pero si lo tuyo son los videojuegos, C# es el lenguaje que debes dominar. Y es que una de las herramientas más potentes hoy en día como es Unity soporta varios lenguajes de forma nativa. Uno de ellos es C#.



# PHP

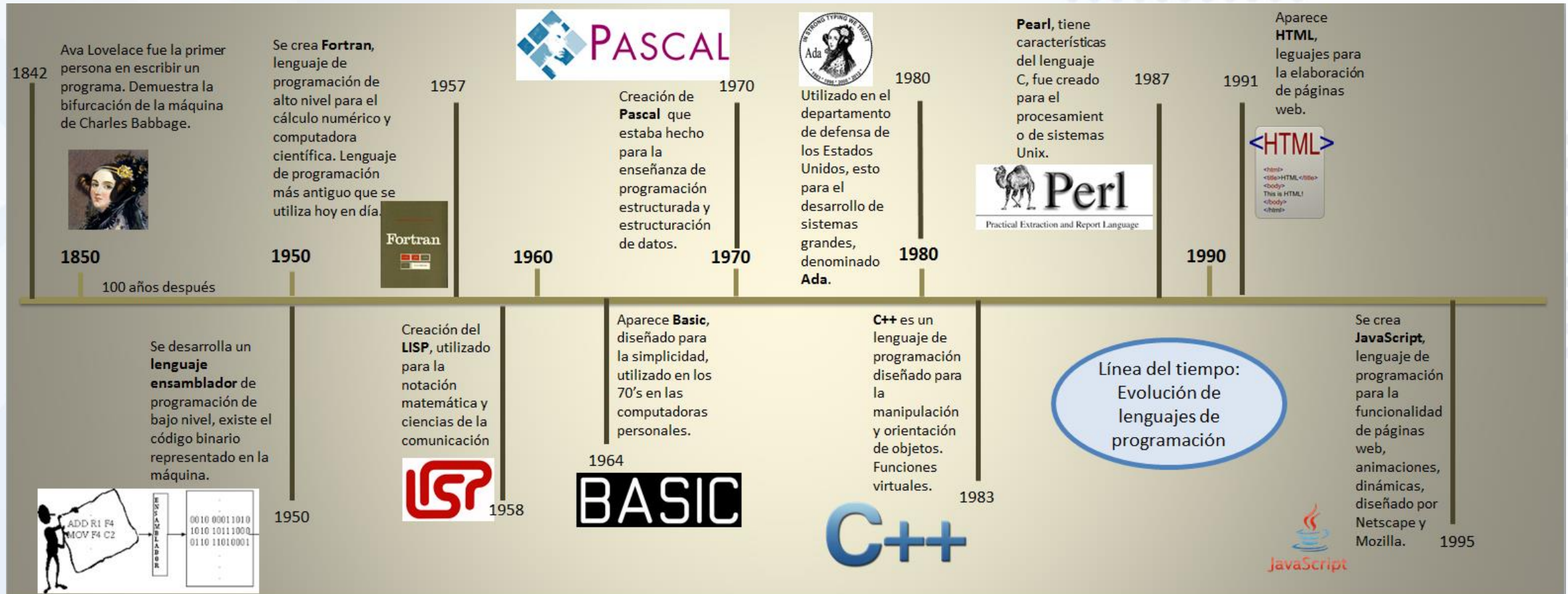
PHP es un lenguaje de programación de propósito general de código del lado del servidor. Este lenguaje garantiza una buena comunicación entre web y servidor, por lo que las páginas web desarrolladas con este lenguaje son estables y con buen rendimiento.

Actualmente, muchas páginas web están diseñadas con WordPress, que trabaja con PHP. Por ello, este lenguaje te permitirá desarrollar tanto proyectos como plugins para esta plataforma.

# C/C++

C++ es un lenguaje de programación surgido como extensión del lenguaje C para que pudiese manipular objetos. Ambos lenguajes de programación siguen siendo utilizados en la actualidad en bases de datos, navegadores web, compiladores o videojuegos.

# EVOLUCIÓN DE LOS LENGUAJES





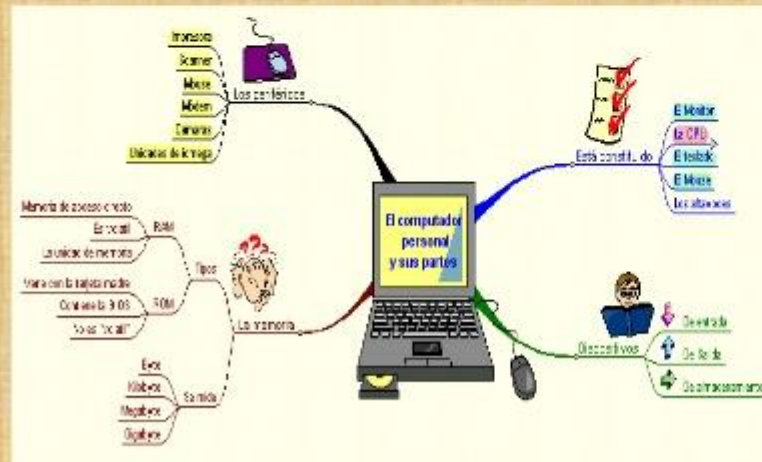
# COMPOSICIÓN DE UN SISTEMA COMPUTACIONAL

## Componentes Sistema Computacional

Tener una visión en perspectiva de un sistema computacional dividido en los respectivos subsistemas.

- Componentes:

- Hardware
- Software
- Datos
- Usuario



# HARDWARE





# SISTEMAS OPERATIVOS

El sistema operativo es el software que coordina y dirige todos los servicios y aplicaciones que utiliza el usuario en una computadora, por eso es el más importante y fundamental. Se trata de programas que permiten y regulan los aspectos más básicos del sistema. Los sistemas operativos más utilizados son Windows, Linux, OS/2 y DOS.

Los sistemas operativos permiten que otros programas puedan utilizarlos de apoyo para poder funcionar. Por eso, a partir del sistema utilizado pueden ser instalados ciertos programas y otros no.

Son parte esencial del funcionamiento de los sistemas informáticos y la pieza de software central en la cadena de procesos, ya que establecen las condiciones mínimas para que todo funcione: la administración de los recursos, el método de comunicación con el usuario y con otros sistemas, las aplicaciones adicionales.

# SISTEMAS OPERATIVOS

- Microsoft Windows. De los más populares que existen, inicialmente se trató de un conjunto de distribuciones o entornos operativos gráficos, cuyo rol era brindar a otros sistemas operativos más antiguos como el MS-DOS una representación visual de soporte y de otras herramientas de software. Se publicó por primera vez en 1985 y desde entonces se ha actualizado a nuevas versiones.
- UNIX. Este sistema operativo fue desarrollado en 1969 para ser portable, multitarea y multiusuario. Se trata realmente de una familia entera de SO similares, algunas de cuyas distribuciones se han ofrecido comercialmente y otros en formato libre, siempre a partir del núcleo llamado Linux.
- MacOS. Es el sistema operativo de los computadores Macintosh de Apple, también conocido como OSX o Mac OSX. Basado en Unix y desarrollado y vendido en computadores Apple desde 2002, se trata de la competencia más acérrima del popular Windows.
- Lunux. Este sistema operativo es libre y de código abierto, o sea, todo el mundo puede modificarlo sin violar derechos autorales. Toma su nombre de cierta filosofía surafricana ancestral, enfocada en la lealtad del hombre hacia su propia especie por encima de todo. Basado en GNU/Linux, Ubuntu se orienta hacia la facilidad de uso y la libertad total. La empresa británica que lo distribuye, Canonical, subsiste brindando servicio técnico.
- Android. Este sistema operativo basado en el núcleo Linux opera en teléfonos celulares y tablets y otros artefactos dotados de pantalla táctil. Fue desarrollado por Android Inc. y comprado posteriormente por Google, gracias a lo cual es tan popular que las ventas de sistemas informáticos Android superan a las de IOS (para teléfonos celulares Macintosh) y a las de Windows Phone (para teléfonos celulares MicroSoft).

# LAS APLICACIONES

Una aplicación es un programa informático diseñado como una herramienta para realizar operaciones o funciones específicas. Generalmente, son diseñadas para facilitar ciertas tareas complejas y hacer más sencilla la experiencia informática de las personas.

# LAS APLICACIONES

## 2. Tipos de aplicaciones

- A continuación se muestran las más utilizadas pero existen muchas más:
  - Imagen y diseño: OpenOffice Draw, Microsoft Paint
  - Gestión contable y finanzas: ContaPlus, Microsoft Money
  - Seguridad: Microsoft Security Essentials
  - Multimedia: Windows Media Player, Winamp, Audicity
  - Presentaciones: Powerpoint, OpenOffice Impress
  - Navegadores: Google Chrome, Mozilla Firefox, Internet Explorer
  - Hojas de cálculo: Excel, OpenOffice Calc
  - Bases de datos: Access, OpenOffice Base
  - Procesadores de texto: Word, OpenOffice Write
  - Correo electrónica: Mozilla Thunderbird, Microsoft Outlook



# USUARIOS

En informática y la cultura Web, se entiende por usuario a un conjunto de permisos y de recursos asignados a un operador como parte de una red informática, y que bien puede ser una persona, un programa informático o un computador.



# USUARIOS

- Usuarios registrados. Se trata de aquellos usuarios dotados de una cuenta de usuario y que hacen vida regularmente en la red, ya sea como consumidores o productores de información, o ambas figuras intermitentemente.
- Usuarios anónimos. Aquellos que navegan en Internet sin hacer manifiesta su presencia a través de registros, formalizaciones o cuentas asignadas a un usuario, sino que se mantiene sin identificar. Normalmente posee menos privilegios que el usuario registrado.
- Trolls. En esta categoría entran los usuarios de foros y redes sociales cuya presencia en estos ámbitos sociales es problemática o abusiva: incitan al odio, agreden verbalmente a los demás y hacen la experiencia menos placentera.
- Beta-testers. Se trata de usuarios de prueba, es decir, que emplean un software de manera experimental o en desarrollo, para así tomar nota de sus debilidades o evaluar su funcionamiento, etc.
- Hackers. Aquellos usuarios dotados de conocimientos profundos de informática y que son capaces de sabotear o alterar segmentos de código de los programas Web para beneficiarse de ello de distintas maneras.