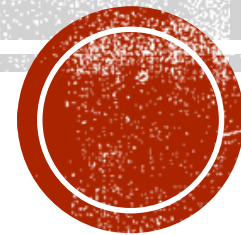


PROGRAMACIÓN ORIENTADA A OBJETOS



Ing. Sara Osorio
Especialista en Ingeniería de Software
UNINPAHU
Facultad de Ingeniería
2019

QUE ES SOFTWARE?

Es el conjunto de los programas de cómputo, procedimientos, reglas, documentación y datos asociados, que forman parte de las operaciones de un sistema de computación.

Considerando esta definición, el concepto de *software* va más allá de los programas de computación en sus distintos estados: código fuente, binario o ejecutable; también su documentación, los datos a procesar e incluso la información de usuario forman parte del *software*: es decir, *abarca todo lo intangible*, todo lo «no físico» relacionado.



CLASIFICACION DEL SOFTWARE?

Software de sistema

Su objetivo es desvincular adecuadamente al usuario y al programador de los detalles del sistema informático en particular que se use, aislándolo especialmente del procesamiento referido a las características internas de: memoria, discos, puertos y dispositivos de comunicaciones, impresoras, pantallas, teclados, etc.

El *software* de sistema le procura al usuario y programador adecuadas interfaces de alto nivel, controladores, herramientas y utilidades de apoyo que permiten el mantenimiento del sistema global.

Incluye entre otros:

- Sistemas operativos
- Controladores de dispositivos
- Herramientas de diagnóstico
- Herramientas de corrección y optimización
- Servidores
- Utilidades



CLASIFICACION DEL SOFTWARE?

Software de programación

Es el conjunto de herramientas que permiten al programador desarrollar programas de informática, usando diferentes alternativas y lenguajes de programación, de una manera práctica.

Incluye entre otros:

- Editores de texto
- Compiladores
- Intérpretes
- Enlazadores
- Depuradores
- Entornos de desarrollo integrados (IDE): Agrupan las anteriores herramientas, usualmente en un entorno visual, de forma tal que el programador no necesite introducir múltiples comandos para compilar, interpretar, depurar, etc. Habitualmente cuentan con una avanzada interfaz gráfica de usuario (GUI).



CLASIFICACION DEL SOFTWARE?

Software de programación

Es aquel que permite a los usuarios llevar a cabo una o varias tareas específicas, en cualquier campo de actividad susceptible de ser automatizado o asistido, con especial énfasis en los negocios.

Incluye entre otros:

- Aplicaciones para Control de sistemas y automatización industrial
- Aplicaciones ofimáticas
- Software educativo
- Software empresarial
- Bases de datos
- Telecomunicaciones (por ejemplo Internet y toda su estructura lógica)
- Videojuegos
- Software médico
- Software de cálculo numérico y simbólico.
- Software de diseño asistido (CAD)
- Software de control numérico (CAM)



EVOLUCION DEL SOFTWARE?

Primera Generación

1946 (1950) - 1965

Se trabajaba con la idea de “Codificar y Corregir”

No existía un planteamiento previo

No existía documentación de ningún tipo

Existencia de pocos métodos formales y pocos
creyentes de ellos

Desarrollo a base de prueba y error

Nace el lenguaje Fortran (1957); cobol (1960)

Eniac (1946)



<https://www.youtube.com/watch?v=TXMXKTvu5Bs>

<https://www.youtube.com/watch?v=oSssWHD1oSI>



EVOLUCION DEL SOFTWARE?

Segunda Generación

1965 - 1972

Se busca simplificar código

Aparición de multiprogramación y sistemas multiusuarios

Sistemas de tiempo Real apoyan la toma de decisiones

Aparición de software como producto (Casas de software)

Inicio a la crisis del software

Incremento en la demanda, en la complejidad y en los retos

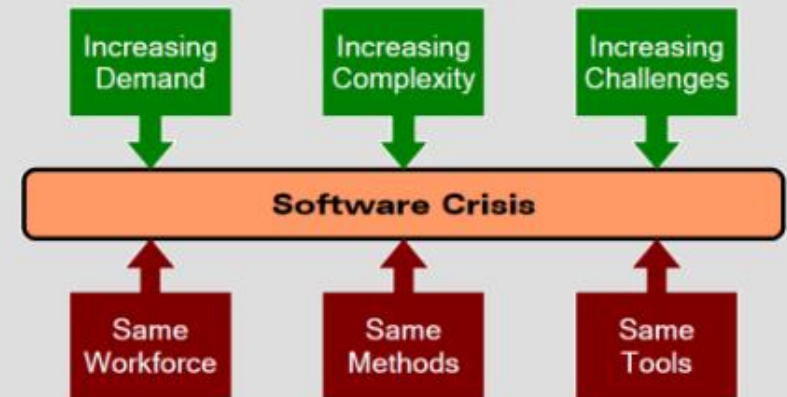
La misma fuerza de trabajo, métodos y herramientas

Se busca procedimientos para el desarrollo del software

Nace la ingeniería de software en 1968

Nace el lenguaje de programación C (1972)

Crisis de software (1968)



EVOLUCION DEL SOFTWARE?

Tercera Generación

1972 - 1985

Nuevo concepto: Sistemas Distribuidos

Complejidad en los sistemas de información

Aparecen: redes de área local y global, y comunicadores digitales

Amplio Uso de Microprocesadores

Nace el lenguaje de programación Basic (1975)



EVOLUCION DEL SOFTWARE?

Cuarta Generación

1985 – 1995 aprox.

Impacto colectivo del software

Aparecen: redes de información, Tecnologías orientadas a objetos

Aparecen: redes neuronales, sistemas expertos y SW de inteligencia artificial

La información como valor preponderante dentro de las organizaciones

Nace el lenguaje Java (1990)

Una computadora gana el campeonato de ajedrez (1997)



EVOLUCION DEL SOFTWARE?

Quinta Generación

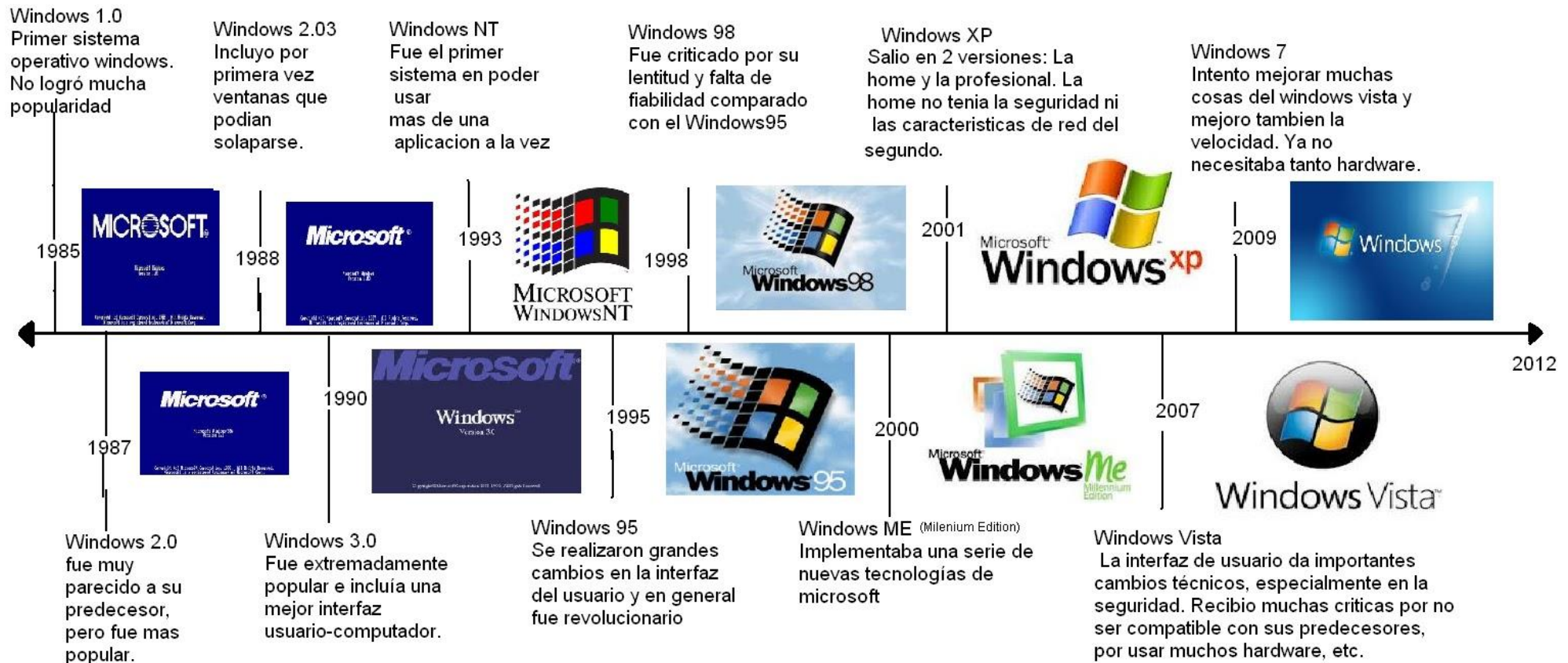
2000 hasta hoy.

Utiliza algunos requisitos de las eras anteriores solo que aumenta la omnipresencia de la web, la reutilización de la información y componentes de software



EVOLUCION DEL SOFTWARE?

Evolución de Windows



CARACTERISTICAS DEL SOFTWARE

Características Operativas

Son factores de funcionalidad, es como se presenta el software, es la “parte exterior” del mismo. Incluye aspectos como:

- Corrección: El software que estamos haciendo debe satisfacer todas las especificaciones establecidas por el cliente.
- Usabilidad / Facilidad de aprendizaje: Debe ser sencillo de aprender.
- Integridad: Un software de calidad no debe tener efectos secundarios.
- Fiabilidad: El producto de software no debería tener ningún defecto. No sólo esto, no debe fallar mientras la ejecución.
- Eficiencia: Forma en que el software utiliza los recursos disponibles. El software debe hacer un uso eficaz del espacio de almacenamiento y el comando ejecutar según los requisitos de tiempo deseados.
- Seguridad: Se deben tomar medidas apropiadas para mantener los datos a salvo de las amenazas externas



CARACTERISTICAS DEL SOFTWARE

Características de transición

- Interoperabilidad: Es la capacidad para el intercambio de información con otras aplicaciones.
- Reutilización: Es poder utilizar el código de software con algunas modificaciones para diferentes propósitos.
- Portabilidad: Capacidad para llevar a cabo las mismas funciones en todos los entornos y plataformas.



CARACTERISTICAS DEL SOFTWARE

Características de transición

Son los factores de ingeniería, la "calidad interior" del software como la eficiencia, la documentación y la estructura. Incluye aspectos como:

- Capacidad de mantenimiento: El mantenimiento del software debe ser fácil para cualquier tipo de usuario.
- Flexibilidad: Los cambios en el software debe ser fácil de hacer.
- Extensibilidad: Debe ser fácil de aumentar nuevas funciones.
- Escalabilidad: Debe ser muy fácil de actualizar para más trabajo.
- Capacidad de prueba: Prueba del software debe ser fácil.
- Modularidad: Debe estar compuesto por unidades y módulos independientes entre sí.

