

# HISTORIA Y EVOLUCIÓN DEL SOFTWARE

El contexto en el que se ha desarrollado el software está fuertemente ligado a las casi cinco décadas de evolución de los sistemas informáticos. Un mejor rendimiento del hardware, una reducción del tamaño y un costo más bajo, han dado lugar a sistemas informáticos más complejos. Hemos pasado de los procesadores con válvulas de vacío a los dispositivos microelectrónicos que son capaces de procesar 200 millones de instrucciones por segundo y más.

## **Primeros Años**

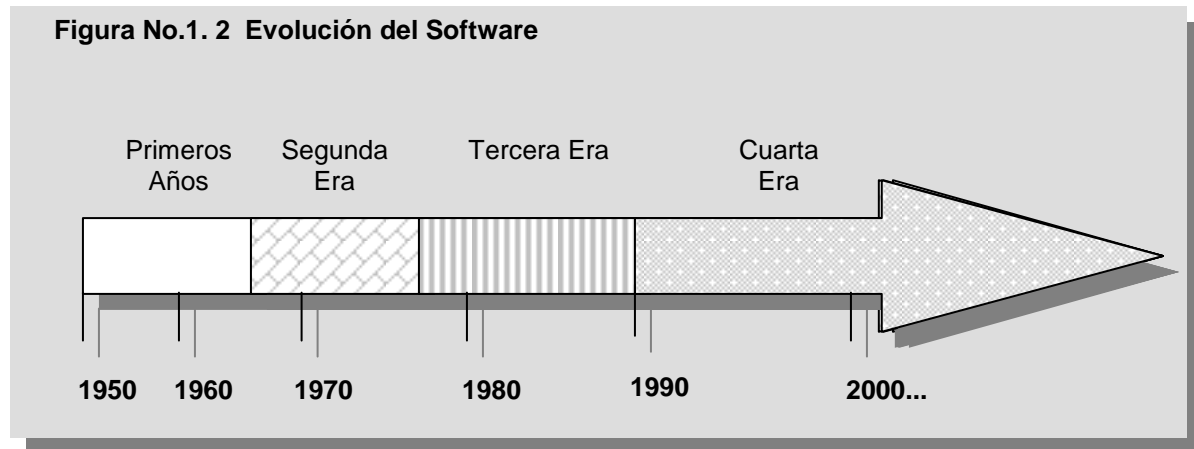
Durante los primeros años de desarrollo de las computadoras, el hardware sufrió continuos cambios, mientras que el software se contemplaba simplemente como un agregado.

La programación de computadoras era un arte para el que existían pocos métodos sistemáticos y el desarrollo del software se realizaba virtualmente sin ninguna planificación (los costos crecían y los planes eran un descalabro).

Durante este período, en la mayoría de los sistemas se utilizaba una orientación por lotes, siendo algunas excepciones notables varios sistemas interactivos tales como el sistema de reserva de pasajes de la American Airlines y los sistemas de tiempo real para la defensa. Sin embargo, la mayor parte del hardware se dedicaba a la ejecución de un único programa que, a su vez, se dedicaba a una aplicación específica. Lo normal era que el hardware fuera de propósito general y, por otra parte, el software se diseñaba a medida para cada aplicación y tenía una distribución relativamente pequeña.

La mayoría del software se desarrollaba y era utilizado por la misma persona u organización: la misma persona lo escribía, lo ejecutaba y, si fallaba, lo depuraba; debido a que la movilidad en el trabajo era baja, los ejecutivos estaban seguros de que esa persona estaría allí cuando se encontrara algún error. En este

entorno personalizado del software, el diseño era un proceso implícito, realizado en la mente de alguien, y la documentación normalmente no existía.



A lo largo de los primeros años se aprendió mucho sobre la implementación de sistemas informáticos, pero relativamente poco sobre la ingeniería de software. Sin embargo, es digno reconocer que durante esa era se desarrollaron muchos sistemas informáticos excepcionales, algunos de los cuales todavía se siguen usando hoy y, por sus características, siguen siendo admirados con toda justicia. (Ver figura 1.2)

### **Segunda Era**

La segunda era en la evolución de los sistemas computacionales se extiende desde la mitad de la década de los sesenta hasta finales de los setenta, como se puede observar en la figura 1.2. La multiprogramación y los sistemas multiusuario introdujeron nuevos conceptos de interacción hombre-máquina. Las técnicas interactivas abrieron un nuevo mundo de aplicaciones y nuevos niveles de complejidad del hardware y del software. Los sistemas de tiempo real podían recoger, analizar y transformar datos de múltiples fuentes, controlando así los procesos y produciendo salidas en milisegundos en vez de en minutos. Los

avances en los dispositivos de almacenamiento en línea condujeron a la primera generación de sistemas de gestión de bases de datos.

Esta era se caracterizó también por el establecimiento del software como producto y la llegada de las "casas de software", donde el software ya se desarrollaba para tener una amplia distribución en un mercado multidisciplinario: los programas se distribuían para computadoras grandes y para minicomputadoras, a cientos e incluso a miles de usuarios; la industria, el gobierno y la universidad se aprestaban a "desarrollar el mejor paquete de software" y ganar así mucho dinero.

Conforme crecía el número de sistemas informáticos, comenzaron a extenderse las bibliotecas de software, las casas desarrollaban proyectos en los que se producían programas de decenas de miles de sentencias fuente y los productos de software comprados al exterior incorporaban cientos de miles de nuevas sentencias. Todos esos programas (todas esas sentencias) tenían que ser corregidos cuando se detectaban fallos, modificados cuando cambiaban los requisitos de los usuarios o adaptados a nuevos dispositivos de hardware que se hubieran adquirido; estas actividades se llamaron colectivamente mantenimiento del software.

El esfuerzo gastado en el mantenimiento del software comenzó a absorber recursos en una medida alarmante, aún peor, la naturaleza personalizada de muchos programas los hacía virtualmente imposibles de mantener. Había comenzado una "crisis del software".

### **Tercera Era**

La tercera era en la evolución de los sistemas computacionales, comenzó a mediados de los setenta (Ver figura 1.2). El procesamiento distribuido (múltiples computadoras, cada una ejecutando funciones concurrentemente y comunicándose con alguna otra) incrementó notablemente la complejidad de los

sistemas informáticos. Las redes de área local y de área global, las comunicaciones digitales de gran ancho de banda y la creciente demanda de acceso "instantáneo" a los datos, supusieron una fuerte presión sobre los desarrolladores del software.

Se produce la llegada y el amplio uso de los microprocesadores y las computadoras personales. El microprocesador es una parte integral de un amplio espectro de productos "inteligentes" que incluyen automóviles, hornos microondas, robots industriales y equipos de diagnóstico médico.

En muchos casos, la tecnología del software es integrada en esos productos por equipos técnicos que conocen el hardware, pero que a menudo no tienen experiencia en desarrollo de software. Las computadoras personales han sido el catalizador del gran crecimiento de muchas compañías de software. Mientras que las compañías de software de la segunda era vendían cientos o miles de copias de sus programas, las compañías de software de la tercera era venden decenas e incluso centenares de miles de copias. El hardware de las computadoras personales se ha convertido rápidamente en un producto estándar, mientras que el software que se suministre con ese hardware, es lo que marca la diferencia. De hecho, mientras que las ventas de computadoras personales se estabilizaron hacia la mitad de los 80, las ventas de productos de software han continuado creciendo. Mucha gente en el campo industrial y muchos particulares han gastado más dinero en software que lo que se gastaron en la computadora sobre la que se ejecuta el software.

### **Cuarta Era**

La cuarta era del software se aleja de las computadoras individuales y de los programas de computadoras, dirigiéndose al impacto colectivo de las computadoras y del software. Potentes máquinas personales controladas por sistemas operativos sofisticados, en redes globales y locales, acompañadas por aplicaciones de software avanzadas se han convertido en la norma. Cambia la

arquitectura informática de entornos centralizado de grandes computadores a entornos descentralizados cliente / servidor. De hecho Internet se puede observar como un *software* al que pueden acceder usuarios individuales.

La cuarta era del software está enfocada a las tecnologías orientadas a los objetos y están desplazando rápidamente a enfoques de desarrollo de software más convencionales en muchas áreas de aplicación. Las técnicas de cuarta generación para el desarrollo de software ya están cambiando la forma en que algunos segmentos de la comunidad informática construyen los programas computacionales.

Por fin, los sistemas expertos y el software de inteligencia artificial se han trasladado del laboratorio a las aplicaciones prácticas, para un amplio rango de problemas del mundo real. El software de redes neuronales artificiales ha abierto excitantes posibilidades para el reconocimiento de formas y habilidades de procesamiento de información al estilo de como lo hacen los humanos. Pero aún en la cuarta era, continúan intensificándose los problemas asociados con el software:

- La tecnología del hardware ha dejado desfasada a la capacidad de construir software que pueda explotar el potencial del hardware.
- La capacidad de construir nuevos programas no puede dar abasto a la demanda de nuevos programas.
- La capacidad de mantener los programas existentes está amenazada por el mal diseño y el uso de recursos inadecuados.
- Sin una buena ingeniería de software, es prácticamente imposible lograr un producto con calidad.

## **HISTORIA DE LA INGENIERÍA DE SOFTWARE**

El término “ingeniería del software” se introdujo por primera vez a finales de la década de 1960 en una conferencia celebrada para analizar la llamada “crisis del software”. Esta crisis fue el resultado directo de la aparición del hardware de computadores de la tercera generación. Estas máquinas eran de una capacidad muy superior a la de las máquinas más potentes de la segunda generación, y su potencia hizo posibles las aplicaciones que hasta ese momento eran irrealizables. El desarrollo de esas aplicaciones requirió la construcción de grandes sistemas de software.

Las primeras experiencias en la construcción de grandes sistemas de software mostraron que las metodologías de desarrollo de software hasta entonces existentes eran inadecuadas. No podía hacerse una simple ampliación a escala de las técnicas aplicables a los sistemas pequeños. Varios proyectos importantes se retrasaron (algunas veces años), costaron mucho más de lo previsto en principio y resultaron poco confiables, difíciles de mantener y de rendimiento pobre.

El desarrollo del software estaba en una situación de crisis. Los costos del hardware caían mientras que los del software aumentaban con rapidez. Había una urgente necesidad de nuevas técnicas y metodologías que permitieran controlar la complejidad inherente a los grandes sistemas de software.

Además, el término “software” no sólo abarca los programas de computación asociados con alguna aplicación o producto. Junto con los programas, el software incluye toda la documentación necesaria para instalar, usar, desarrollar y mantener esos programas.

## **EVOLUCIÓN DE LA INGENIERÍA DE SOFTWARE**

**1968:** Surge la **crisis del software** pues el proceso de desarrollo de software sobrepasa los costos y el tiempo estimados y tiene demasiadas fallas. Se acuña el nombre de **Ingeniería de software**.

**1969 - 19771:** Se establecen los primeros principios como resultado de la investigación en buenas prácticas de programación. Tales como: Diseño descendente (top-down), Refinamiento sucesivo, Modularidad, Surgen los lenguajes estructurados y la noción de desarrolladores de software dirigidos por un líder.

**1972 - 1973:** Surge la programación estructurada y el concepto de estilo de programación. Aparece el concepto de ciclo de vida en el desarrollo de software. Se proponen ayudas para la administración y de apoyo en el desarrollo.

**1974 - 1975:** Se inicia el manejo de la noción de confiabilidad y calidad del software. Se inician las pruebas sistemáticas, la noción de corrección formal, modelos de tolerancia a fallas y confiabilidad total.

**1976 - 1977:** Se pone atención a las fases anteriores a la codificación como análisis de requerimientos, especificación y diseño. Surgen las técnicas como abstracción y descomposición modular. Se hacen esfuerzos para integrar y validar las fases sucesivas del desarrollo.

**1978 -1980:** Incremento en el uso de herramientas automatizadas de desarrollo de software. Se llevan a cabo cursos de ingeniería de software. Los principios desarrollados en 69 a 71 se usan masivamente en la industria del software.

**1980 - 1985:** Se inician las herramientas automatizadas para cada fase del ciclo de vida siguiendo la programación estructurada.

**1986 - 1995:** Aparece el paradigma orientado a objetos para el desarrollo de software grande. Promueve conceptos como la abstracción, la herencia el

reuso. Se inicia el uso masivo de técnicas automatizadas de desarrollo de software.

**1995-2003:** La evaluación de los procesos evita especificaciones incompletas o anómalas, la aplicación incorrecta de metodologías, etc. Para ello se utilizan distintos modelos de madurez de procesos que tienen como objetivo apoyar distintas estrategias de desarrollo y evaluación para así lograr una mejora continua en los productos. Cabe resaltar que no se debe aplicar alguno de estos modelos de madurez bajo el supuesto de mejorar en su calidad sin antes establecer y definir los procesos correspondientes. En particular, la calidad de un sistema de software está gobernada por la calidad del proceso utilizado para desarrollarlo y mantenerlo.