

## CLASE 1

# Ingeniería del Software

Página  
1

Nuestra materia no pretende profundizar en el estudio de la Ingeniería del Software, más bien introducimos el concepto como punto de partida y como marco referencial para todo lo que vendrá. Hablaremos de los pasos iniciales para la construcción de software de calidad, a partir de un enfoque científico, de ahí la importancia de la concepción del desarrollo del software desde la perspectiva de la ingeniería del software.

### Concepto



**“La ingeniería del software es el establecimiento y uso de principios sólidos de la ingeniería para obtener económicamente un software confiable y que funcione de modo eficiente en máquinas reales”.**<sup>1</sup>

Esta definición la propuso Fritz Bauer en una conferencia fundamental sobre Ingeniería del Software.

*“La ingeniería del software es una disciplina de la ingeniería que comprende todos los aspectos de la producción de software desde las etapas iniciales de la especificación del sistema, hasta el mantenimiento de éste después de que se utiliza. En esta definición, existen dos frases clave:*

1. **Disciplina de la ingeniería.** Los ingenieros hacen que las cosas funcionen. Aplican teorías, métodos y herramientas donde sean convenientes, pero las utilizan de forma selectiva y siempre tratando de descubrir soluciones a los problemas, aun cuando no existan teorías y métodos aplicables para resolverlos. Los ingenieros también saben que deben trabajar con restricciones financieras y organizacionales, por lo que buscan soluciones tomando en cuenta estas restricciones.
2. **Todos los aspectos de producción de software.** La ingeniería del software no sólo comprende los procesos técnicos del desarrollo de software, sino también con actividades tales como la gestión de proyectos de software y el desarrollo de herramientas, métodos y teorías de apoyo a la producción de software.

<sup>1</sup> Roger S. Pressman. (2005). 6<sup>ta</sup> Edición. Ingeniería del Software. Un Enfoque Práctico. McGraw-Hill Interamericana. México. Pág. 23.

*En general, los ingenieros de software adoptan un enfoque sistemático y organizado en su trabajo, ya que es la forma más efectiva de producir software de alta calidad. Sin embargo, aunque la ingeniería consiste en seleccionar el método más apropiado para un conjunto de circunstancias, un enfoque más informal y creativo de desarrollo podría ser efectivo en algunas circunstancias. El desarrollo informal es apropiado para el desarrollo de sistemas basados en Web, los cuales requieren una mezcla de técnicas de software y de diseño gráfico”.*<sup>2</sup>

**Página**

**2**



El uso de un proceso, de métodos de desarrollo y el apoyo de herramientas a lo largo de todo el ciclo de vida del software, es una forma racional de afrontar los proyectos de software que buscan automatizar los requerimientos de los usuarios, sin volver a caer en los errores de un proceso de desarrollo inmaduro, centrado en la fase de implementación, que da como resultado una base de software de difícil mantenimiento.

Es importante no confundir la definición de Ingeniería de Software con Ingeniería de Sistemas. La Ingeniería de Sistemas concierne a todos los aspectos del desarrollo de sistemas basados en el cómputo, es decir, hardware, software y el proceso de ingeniería. La Ingeniería de Software es parte de este proceso.

## Importancia y objetivos de la Ingeniería del Software

La revolución en la creación del hardware a partir de la década de los 60 trajo consigo la necesidad de crear software que acompañe esa magnífica y próspera evolución. Diseñar software bajo criterios que garanticen previsibilidad, empezó a ser más que un objetivo, una necesidad. No era posible establecer plazos correctos para la finalización de los proyectos de construcción de software, ni costos estimativos necesarios para su creación. Se sumaba la baja calidad de los programas y la casi nula posibilidad de mantenerlos, por tener un diseño poco estructurado.

En la construcción y desarrollo de proyectos se aplican métodos y técnicas para resolver los problemas, la informática aporta herramientas y procedimientos sobre los que se apoya la ingeniería del software, de ahí su importancia. Estos permiten cumplir con los objetivos de mejorar la calidad de los productos de software, aumentando la productividad, facilitando el control del proceso de desarrollo de software, es decir, suministrando a los desarrolladores las bases para construir software de alta calidad en una forma eficiente.

## Principios de la Ingeniería del Software

Cualquiera sea el proceso elegido para la construcción del software, requiere la elección de principios adecuados que conlleven a la creación de software confiable. Para ello es importante partir con éstas preguntas:

¿Cómo se logra construir software de calidad?

¿Existe algún principio o práctica general que asegure que el producto de software creado tendrá ciertas cualidades?

<sup>2</sup> Ian Sommerville. (2006). 7<sup>ma</sup> Edición. Ingeniería del Software. Pearson Addison Wesley. Madrid España. Pág. 6.

Necesitamos ofrecer estabilidad, control y organización a una actividad que puede volverse incontrolable. Cualquier proceso de construcción que se adopte, dependerá del tipo de software que se está construyendo.

Página  
3

Algunos principios de la Ingeniería del Software son:

- **Rigor y formalidad:** El rigor es una cualidad intuitiva que no puede ser definida en forma rigurosa, pero sí pueden alcanzarse varios niveles de rigurosidad siendo el más alto la formalidad.
- **Separación de intereses:** Permite enfrentarse a los distintos aspectos individuales de un problema de forma de concentrarse en cada uno por separado.
- **Modularidad:** Un sistema complejo puede dividirse en partes más simples llamadas módulos, un sistema compuesto de módulos es llamado modular.
- **Abstracción:** La abstracción es un proceso mediante el cual se identifican los aspectos relevantes de un problema ignorando los detalles.
- **Anticipación al cambio:** Estos cambios pueden surgir por la necesidad de apoyar la evolución de la aplicación debido a nuevos requerimientos o cambios en los requerimientos existentes.
- **Generalidad:** El principio de generalidad establece que al tener que resolver un problema se debe buscar un problema más general que posiblemente esté oculto tras el problema original.
- **Incrementalidad:** Caracteriza un proceso que se desarrolla en forma de pasos, en incrementos, alcanzando el objetivo deseado mediante aproximaciones sucesivas al mismo, donde cada aproximación es alcanzada a través de un incremento de la previa.

## Metodología del desarrollo del software



**A**l elegir una estrategia para la construcción de software es importante adoptar una metodología que nos permita una práctica adaptativa en vez de predictiva. Podremos decidir por una estándar o construir nuestra propia metodología.

Se deben considerar algunos aspectos que la metodología debe satisfacer:

- Permitir cubrir el desarrollo completo del software.
- Facilitar la integración de las distintas fases del ciclo de vida del software.
- Poder facilitar una comunicación efectiva.
- Incluir la realización de validaciones.
- Funcionar en un entorno dinámico orientado al usuario.
- Debe poderse enseñar.
- Soportar la evolución del sistema.

Existen 2 enfoques para la elección de las metodologías, uno clásico y el otro que propone una metodología “ágil”.

Este cambio radical de enfoque en las metodologías se produjo en respuesta a lo que vienen siendo los problemas endémicos en el desarrollo de software, en donde:

- ✓ Los requerimientos nunca están totalmente definidos y claros antes de comenzar el proyecto.
- ✓ Los usuarios comprenden lo que realmente quieren tan solo después de ver una versión inicial del software.
- ✓ Los requerimientos suelen sufrir cambios frecuentes durante la fase de desarrollo del proyecto.
- ✓ El uso de nuevas tecnologías y herramientas, nunca usadas anteriormente, hacen muy difícil el definir a priori las mejores estrategias de trabajo.

Las metodologías ágiles se centran en desarrollar métodos de trabajo flexibles que permitan adaptarse al cambio, a “ir trabajando sobre la marcha”, definiendo el camino según va avanzando el proyecto. Pero siempre sin perder el rumbo.

La principal diferencia entre ambas corrientes metodológicas:

- Las metodologías tradicionales ponen mucho énfasis sobre el plan de proyecto. En tenerlo todo bien especificado antes de comenzar, en seguir fielmente el camino planificado y en documentar exhaustivamente todo lo realizado.
- Las metodologías ágiles por el contrario, ponen el énfasis en entregar buen código al usuario, en obtener resultados que satisfagan, adaptándose a sus siempre cambiantes necesidades.

# Sistemas de Información

## Conceptos de Sistemas



**E**s posible definir como sistema a un conjunto de partes o elementos organizados y relacionados que interactúan entre sí para lograr un objetivo común. Los sistemas esperan de su entorno o ambiente entrada de datos, energía o materia y facilitan una salida de información, energía o materia.

Un sistema puede ser físico o concreto (un lavarropas, una radio, un humano) o puede ser abstracto o conceptual (un software). Nosotros nos abocaremos principalmente al diseño de sistemas que permitan automatizar los procesos en las empresas, lo que se conoce como sistemas transaccionales.

Un sistema existe dentro de otro más grande, por lo tanto cada sistema puede estar formado por subsistemas y partes, y a la vez puede ser parte de un supersistema. Esto servirá para que el analista de sistemas pueda “negociar su frontera”, es decir,

Cada sistema tiene su límite o frontera, que lo diferencia del ambiente. Ese límite puede ser físico o conceptual. Si hay algún intercambio entre el sistema y el ambiente a través de ese límite, el sistema es abierto, de lo contrario, el sistema es cerrado.

Un grupo de elementos no constituye un sistema si no hay una relación e interacción, que de la idea de un "todo" con un propósito u objetivo común.

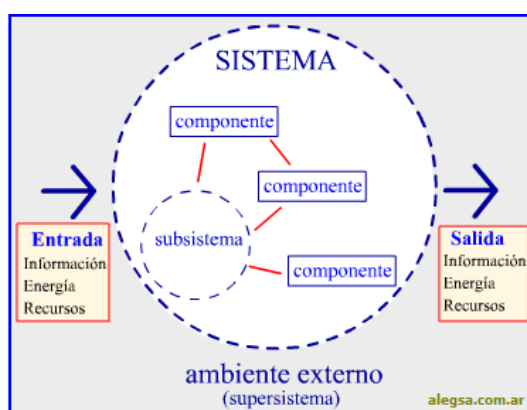


Gráfico de Sistema de Información

## Tipos de Sistemas

**H**ay distintos tipos de sistemas, que se agrupan de acuerdo a sus características:

- **Sistemas Transaccionales:** Se automatizan tareas operativas de la organización. Son intensivos en entrada, como recolectores de información, y salidas, para su explotación posterior; sus cálculos y procesos suelen ser simples y poco sofisticados. Son fáciles de justificar ante la dirección general, ya que sus beneficios son visibles y palpables.
- **Sistemas de Apoyo de las Decisiones:** Se introducen después de haber implantado los Sistemas Transaccionales más relevantes de la empresa, ya que estos últimos constituyen su plataforma de información. La información que generan sirve de apoyo a los mandos intermedios y a la alta administración en el proceso de toma de decisiones. Intensivos en cálculos y escasos en entradas y salidas de información. Son interactivos y amigables, con altos estándares de diseño gráfico y visual, ya que están dirigidos al usuario final.
- **Sistemas Estratégicos:** Se desarrollan dentro de la organización, por lo tanto no pueden adaptarse fácilmente a paquetes disponibles en el mercado. Su forma de desarrollo es a base de incrementos y a través de su evolución dentro de la organización. Se inicia con un proceso o función en particular y a partir de ahí se van agregando nuevas funciones o procesos. Su función es lograr ventajas que los competidores no posean.
- Por último, es importante aclarar que algunos autores consideran un cuarto tipo de sistemas de información denominado **Sistemas Personales** de Información, el cual está enfocado a incrementar la productividad de sus usuarios.



**Para pensar**

- A partir de ahora, para la construcción de un sistema informático, ¿Cualquier proyecto de software debe encararse bajo la disciplina de la Ingeniería del Software?
- Todos los sistemas de información, ¿necesariamente deben estar informatizados, funcionando desde una red de computadoras?
- Como se muestra en el gráfico de sistemas de información, se puede observar que hay otro subsistema dentro del sistema, ¿habrá otro sistema por encima del nuevo sistema que queramos construir, y así infinitamente?