



# Ingeniería de software

## ¿A qué llamamos ingeniería?

Es el estudio y aplicación de la tecnología (estructuras, herramientas, maquinas, sistemas, materiales y procesos). "Conjunto de **conocimientos** aplicados en **acciones** de **áreas** con un **fin**."

## 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.

- De sistema: pensados para dar servicios a otros programas.
- De aplicación: resuelven una necesidad específica.
- De ingeniería o de ciencias.
- Embebido.
- IA.

Y más...

## ¿Qué hace a un buen software?

- **Cumple su objetivo.**
- Puede ser utilizado.
- Tiene una buena performance.
- Es mantenible.
- Es confiable. (debe darle tranquilidad al usuario)
- Es seguro.

## ¿Qué es la Ingeniería de Software?

Consiste en aplicar ingeniería al proceso completo de creación de software. Desde el momento en que surge la necesidad/problemática hasta el despliegue y mantenimiento de la solución elaborada.

Incluye organizar el desarrollo, comprender la necesidad, diseñar el software, implementarlo, probarlo, validarlo, desplegarlo y darle mantenimiento. Hoy en día es iterativa e incremental.

## Etapas de la Ingeniería de Software

### Análisis de requerimientos

- Entender lo que el cliente necesita: ¿Cuál es el objetivo? ¿Cómo se adapta a las necesidades de los usuarios finales? ¿Cómo va a usarse?
- **Construir el producto adecuado  $\neq$  Construir adecuadamente el producto**
- Comprender el alcance para evitar desvíos, retrasos y costos extra.

¿Cómo? **Ingeniería de requerimientos**

Indagación, negociación, especificación (en esta no participa el cliente, se hace con tu equipo de desarrolladores), validación (importante, ya que el cliente nos orienta mejor sobre lo que quiere en base a lo que hicimos).

Muchas veces el cliente quiere lo mejor pero las ideas que trae no ayuda a eso.

- Describir lo que quiere el cliente
- Establecer base para el diseño de software
- Definir requerimientos que puedan ser validados

### Diseño

Si crees que una buena arquitectura es cara, intenta con una mala

En base a los requerimientos funcionales y no funcionales es muy importante escoger la arquitectura de software más adecuada.

### Implementación

Involucra los pasos necesarios para pasar de los requerimientos a un producto funcional. Tiene que ser un proyecto organizado.

## Testing y validación

El fracaso es, a veces, más fructífero que el éxito

¿No son lo mismo? No.

La validación es demostrarle al cliente que el software que hicimos cumple con lo pedido.

El testing va antes que validar, comprueba que mínimo funcione. A diferencia de la validación corre del lado del equipo de desarrollo.

### Tipos de test

- Pruebas unitarias
- Pruebas de integración
- UAT (User Acceptance Test)
- Pruebas de usabilidad

## Despliegue

### Entornos

- Development
- QA
- Preproducción -Staging
- Producción

## Mantenimiento

¿Qué pasa después del despliegue?

- Surgen nuevos requerimientos
- Hay cambios en el negocio
- El software tiene que actualizarse para mantener la utilidad
- Aparecen errores no detectados