



Alumno : Jesus Uriel Santana Oliva

Profesor : **PROF.-DANIEL GUILLERMO CONRADO MOGUEL**

Materia: **215 INGENIERÍA DE REQUISITOS DE SOFTWARE 25-2**

Grupo : 1 – A B

¿Qué es la Ingeniería de Software?

La ingeniería de software es una disciplina técnica que combina principios científicos y prácticos para diseñar, construir, mantener y mejorar sistemas de software. Este campo busca garantizar la calidad, la eficiencia y la sostenibilidad de soluciones tecnológicas en un mundo donde el software se ha vuelto esencial en diversos sectores. Adoptar un enfoque sistemático y bien estructurado es crucial para abordar la creciente complejidad de los proyectos modernos. Este documento explora los principios fundamentales, los tipos de requisitos y las estrategias clave en la ingeniería de software.

Estrategias de Planificación

Una planificación adecuada es el pilar de cualquier proyecto exitoso en ingeniería de software. Esta involucra:

- **Definición de objetivos claros:** Identificar lo que el software debe lograr y los indicadores de éxito.

- **Asignación de recursos:** Determinar los equipos, herramientas y tecnologías necesarias para alcanzar los objetivos.
- **Gestín del tiempo:** Establecer cronogramas realistas que permitan cumplir con los plazos establecidos.
- **Coordinación de equipos:** Asegurar que todas las partes trabajen de manera sincronizada, reduciendo riesgos y maximizando la eficiencia.

Aspectos Críticos a Considerar

Un liderazgo efectivo es esencial en la ingeniería de software. Los líderes deben:

- **Guiar equipos hacia objetivos comunes:** Fomentar una comunicación clara y un entendimiento compartido de las metas del proyecto.
- **Monitorear el progreso:** Detectar y corregir desviaciones en etapas tempranas para evitar problemas mayores.
- **Facilitar la resolución de problemas:** Proporcionar soluciones rápidas y eficientes ante cualquier obstáculo.

Requisitos Funcionales y No Funcionales

Los requisitos del software se clasifican en dos categorías principales:

1. Requisitos Funcionales

Definen las capacidades específicas del sistema, como:

- **Gestín de bases de datos:** Permitir el almacenamiento y recuperación eficiente de información.
- **Generación de informes:** Proveer datos claros y comprensibles para los usuarios.
- **Integración de sistemas:** Asegurar la interoperabilidad con otras aplicaciones.

Además, deben cumplir criterios como:

- **Confiabilidad:** Garantizar estabilidad y minimizar fallos.
- **Desempeño:** Ofrecer respuestas rápidas y eficientes.
- **Seguridad:** Proteger contra amenazas externas y garantizar la privacidad de los datos.

2. Requisitos No Funcionales

Definen las características generales del sistema, incluyendo:

- **Rendimiento:** Por ejemplo, asegurar tiempos de respuesta menores a un segundo.
- **Usabilidad:** Diseñar interfaces intuitivas y amigables para los usuarios.
- **Escalabilidad:** Facilitar la adaptación a un mayor volumen de usuarios o datos.
- **Compatibilidad:** Garantizar el funcionamiento en diversas plataformas y dispositivos.

Elementos Fundamentales en el Desarrollo de Software

El desarrollo de software exitoso se basa en componentes clave como:

1. **Algoritmos y Código Fuente:** Determinan el funcionamiento y comportamiento del sistema.
2. **Estructuras de Datos:** Permiten organizar y manipular la información de manera eficiente.
3. **Documentación Técnica y de Usuario:** Incluye especificaciones detalladas y manuales para facilitar el uso y mantenimiento.

Principios Fundamentales de la Ingeniería de Software

La práctica de la ingeniería de software está guiada por principios esenciales:

- **Enfoque en el usuario:** Garantizar que el sistema satisfaga las necesidades y expectativas de los usuarios finales.
- **Simplicidad:** Diseñar soluciones claras y evitar estructuras innecesariamente complejas.
- **Visión estratégica:** Mantener una perspectiva clara de los objetivos del proyecto y del producto.
- **Adaptabilidad:** Diseñar sistemas que puedan ajustarse a nuevas demandas y cambios futuros.
- **Reutilización de componentes:** Crear módulos modulares para optimizar recursos y tiempos.
- **Evaluación constante:** Aplicar un enfoque crítico y proactivo para anticipar problemas y mejorar soluciones.

Optimización de Recursos en el Desarrollo de Software

El uso eficiente de recursos es esencial para garantizar el éxito de un proyecto. Esto incluye:

- **Planificación rigurosa:** Identificar posibles cuellos de botella y evitarlos.

- **Uso adecuado de tecnologías:** Seleccionar herramientas que potencien el desempeño del equipo y del sistema.
- **Monitoreo continuo:** Evaluar el progreso para asegurar que los recursos se utilicen de manera óptima.