

Unidad 4: Compresión de Requerimientos

¿Qué es la Ingeniería de Requerimientos?

Es el proceso para:

- Entender **qué necesita el cliente**
- Analizar y evaluar **qué es factible**
- **Negociar** la mejor solución
- **Especificar** clara y correctamente lo que se va a construir
- **Validar** que la especificación refleje lo deseado

Ejemplo real: Si un cliente pide “un sistema para registrar ventas”, hay que descubrir si se refiere a:

- ¿Un POS?
- ¿Una app móvil?
- ¿Con facturación?
- ¿Usuarios con permisos?

Fundamentos iniciales

Para entender bien el problema:

- Identificar a **todos los participantes**
- Aceptar que habrá **múltiples puntos de vista**
- Fomentar la **colaboración**
- Hacer las preguntas correctas, por ejemplo:
 - ¿Quién lo va a usar?
 - ¿Cuál es el beneficio económico?
 - ¿Qué entorno de negocio tiene?
 - ¿Hay restricciones técnicas?

Modelado de Requerimientos

Objetivos principales

1. Describir lo que el cliente **realmente necesita**
2. Servir como **base para el diseño** del software
3. Generar requerimientos que se puedan **verificar/validar**

En esta etapa no se programa, solo se documenta el qué, no el cómo.

¿Qué se modela?

- Interacciones del usuario
 - Objetos manipulados
 - Funciones requeridas
 - Comportamiento del sistema
 - Interfaces esperadas
 - Restricciones (seguridad, tiempo, costos, etc.)
-

Buenas prácticas (según Pressman)

- Mantenerlo simple
 - No enfocarse aún en la infraestructura
 - Minimizar el acoplamiento
 - Asegurar que el modelo sea útil y entendible para todos
 - Documentar lo esencial: información, funciones y comportamiento
-

Herramientas para Modelar Requerimientos

1. Diagramas de Actividad (UML)

- Representan **flujos de tareas**
- Rectángulos redondeados: actividades
- Flechas: flujo

- Rombos: decisiones (condiciones)
 - Líneas paralelas: procesos en paralelo
-

2. Diagramas de Clases (UML)

- Muestran la **estructura estática del sistema**
 - Incluyen: nombre de clase, atributos y operaciones
 - También revelan relaciones entre clases (herencia, asociación, etc.)
-

3. Diagramas de Estado

- Representan los **estados de un objeto** y cómo cambia con eventos
 - Útiles para modelar comportamiento reactivo y de ciclo de vida
-

4. Diagramas de Canal (Swimlanes)

- Variante de los diagramas de actividad
 - Dividen las tareas **por actor o responsable**
 - Ayudan a ver quién hace qué en un proceso
-

5. DER (Diagrama Entidad-Relación)

- Muestra **datos, relaciones y atributos**
 - Útil para sistemas basados en bases de datos
 - Cada entidad tiene atributos y se vincula con otras
-

6. Casos de Uso e Historias de Usuario

- Casos de uso: Escenarios en UML que explican cómo interactúan los actores con el sistema
 - Historias de usuario: Breves descripciones desde el punto de vista del usuario ("Como cliente, quiero ver mis pedidos para...")
-

Complemento desde Pressman (Capítulo 6)

- Pressman insiste en **modelar desde 3 perspectivas:**

1. **Escenarios (casos de uso):** qué pasa en situaciones concretas
2. **Información:** datos relevantes y su estructura
3. **Clases de análisis:** objetos clave y relaciones

El objetivo es crear una base sólida para el diseño, entendiendo bien el problema antes de construir la solución.

Resumen Visual

Tema	Qué aporta
Ingeniería de Reqs.	Entender, negociar, especificar
Modelado	Qué hace el sistema, no cómo lo hace
Actividades	Flujo de tareas
Clases	Estructura del sistema
Estado	Comportamiento del sistema
Canal	Quién hace qué
DER	Datos y relaciones
Casos de uso	Escenarios de uso típicos