

Requisitos Avanzados - Clase 1

Parte 1: Introducción y Organización

Docentes:

- Bertin Maria Inés: Egresada de la universidad, 20 años de experiencia como Product Manager. Actualmente en software de Comercio Exterior.
- Gerardo Agustin Riera: Ingeniero en Informática, trabaja en desarrollo de apps para eficiencia operativa en tiendas.

Metodología de Evaluación

- 4 Trabajos prácticos: LEL, Escenarios Actuales, Escenarios Futuros, Especificación de Requisitos.
- Video final de hasta 10 minutos explicando mejoras a procesos.
- Entregas por MIEL. Clases por Teams con espacios de consulta.

Grupos

- 3 o 4 integrantes. Se recomienda elegir compañeros conocidos.
- Nota individual, trabajo grupal con participación activa.

Caso de Estudio

- Empresa mediana, con procesos accesibles.
- No debe ser unipersonal ni demasiado grande.
- Requiere entrevistas (1 o 2 mínimas).
- Debe ser "pseudo-real": no necesita requerir software realmente.

Cronograma

- Primeras clases: teoría + elección del caso.
- Luego: trabajo en modelos.
- **Fechas importantes:**
 - Entrega de TPs y video: *24 de junio*.
 - Presentación de videos seleccionados: *15 de julio*.

Parte 2: Teoría

1. Crisis del Software

- Problemas desde los años 60: sistemas no cumplen necesidades, tiempos y costos excedidos.
- Ejemplo gráfico: cliente pide hamaca → devs entregan algo complejo e inútil.

- Causas: mala comunicación, poca documentación, cliente no sabe lo que necesita.
- Solución: Ingeniería de Requisitos.

2. Ingeniería de Requisitos

- Objetivo: traducir necesidades del cliente en requisitos claros y viables.
- **Diferencias:**
 - Requerimiento: lo que el cliente quiere.
 - Requisito: lo que se va a entregar.
 - Especificación: documento formal, "contrato".
- Ejemplo: Denver Airport → mal manejo de requisitos = fracaso.
- Corregir errores tarde es costoso.

3. Modelos de Proceso

- Cascada (Royce, 1970): secuencial, poco flexible.
- Ágil: iterativo, adaptable. No siempre sirve para proyectos grandes.
- Conclusión: usar el modelo adecuado según proyecto.

4. Casos Reales Fallidos

- Denver Airport, apagón en EE.UU./Canadá 2003 → errores de requisitos = fracaso.

5. Tips prácticos

- Entrevistas: no proponer soluciones, solo relevar.
- Grabar y transcribir mínimo 30 minutos.
- No incluir datos confidenciales.
- Foco en procesos, no implementación técnica.

Próxima clase

- Buscar 2 opciones de PYMEs.
- Preparar preguntas para entrevistas.

Clase 2 - 8 abril 2025

1. Importancia de los Requisitos

- Son la base del desarrollo.
- Error en requisitos = falla cara.
- Ingeniería de Requisitos estructura el proceso desde la base.

2. Costo de corrección de errores

- Cuanto más tarde se detecta un error, más caro es corregirlo.

3. Rol del Analista

- Interpretar, distinguir deseos vs necesidades, documentar.
- Habilidades: comunicación, análisis, cuestionamiento.

4. Contexto Observable vs Contexto de Uso

- Observable: entorno actual sin sistema.
- Uso: entorno futuro con sistema.
- El software debe transformar, no solo automatizar.

5. Tipos de Requisitos

Tipo	Descripción	Ejemplo
Funcionales	Acciones que debe ejecutar el sistema	Registrar una venta
No funcionales	Atributos (rendimiento, seguridad)	El sistema debe responder en <2 segundos
De negocio	Objetivos estratégicos	Aumentar ventas un 20% en 6 meses

6. Stakeholders

- Involucrados directos e indirectos.
- Ej: cajeros, contadores, logística.

7. Técnicas de Elicitación

- Observación directa, workshops, prototipos.

8. Validación de Requisitos

- Casos de prueba, revisiones con stakeholders.
- Riesgo: ambigüedad.

9. Gestión de Cambios

- Matrices de trazabilidad.

10. Ejemplo práctico

- Problema: desabastecimiento → Requisito: alerta de stock bajo.

11. Documentación y Herramientas

- Historias de usuario, matrices, JIRA, Trello, EA.

12. Errores comunes

- Suponer que el cliente sabe lo que quiere.
- Ignorar requisitos no funcionales.

13. Consideraciones finales

- Meta: software que transforme.
- Proceso: necesidades → análisis → especificación.

Checklist de validación

- ¿Es medible? ¿Vinculado a objetivo de negocio? ¿Aprobado?

Próxima clase

- Llevar primera entrevista.

Clase 3 - 15 abril 2025

1. ¿Qué es la Ingeniería de Requisitos?


Es el proceso mediante el cual se identifica, analiza, documenta y gestiona lo que se espera que haga un sistema. A lo largo del proceso se delimita el **contexto** o **universo del discurso** (también llamado dominio de aplicación).

Ejemplo: si el sistema es de stock, el universo de discurso será el área de stock y su entorno.

2. Fuentes del contexto

Son todos los elementos que definen el dominio:

- Manuales de procedimientos
- Documentación de procesos de negocio
- Normas de calidad
- Sistemas informáticos preexistentes
- Normativas legales y estándares nacionales o internacionales

 *Estos documentos o fuentes pueden no estar disponibles fácilmente. Algunos requisitos requieren que el cliente proporcione información que puede no tener o no contar con las herramientas adecuadas para obtenerla.*

3. Definición y clasificación de requisitos

¿Qué es un requisito?

Una característica o condición que debe tener el sistema.

Clasificación

1. Por estabilidad

- *Estables*: cambian poco (ej. universidad)
- *Volátiles*: cambian frecuentemente (ej. Google)

2. Por nivel de detalle

- *Generales*
- *Detallados*

3. Por obligatoriedad

- *No negociables*
- *Negociables*

4. Por tipo

- *Funcionales*: comportamiento ante entradas
- *No funcionales*: restricciones o atributos

4. Propiedades de los requisitos (deben ser)

- Consistentes
- Completos
- Correctos
- No ambiguos
- Entendibles
- Realizables
- Rastreables
- Verificables

La mayoría de las especificaciones reales incumplen varias de estas propiedades.

5. Verificabilidad de los requisitos

Especialmente difícil en los **no funcionales**, ya que dependen del contexto real de uso (hardware, red, entorno distribuido, etc.). Requieren definir condiciones claras de prueba.

6. Tipos de requisitos según el foco

Tipo	Descripción
Negocio	Reglas del proceso (ej. el vendedor verifica stock).
Proyecto	Condiciones del desarrollo (ej. test sin errores por 1 día).
Software	Se implementan en código (ej. procesar pedidos desde archivo).

🔗 *Ejemplo agregado: Si precisamos, por ejemplo, una licencia de Oracle, eso es un **requisito del proyecto**, no del software. Debe figurar en la especificación correctamente titulado como tal, sin confundirse.*

7. Trazabilidad entre requisitos y código

- Un requisito puede traducirse en muchas líneas de código.
- Puede haber funcionalidades sin requerimiento asociado (*peligroso*).
- Puede faltar la implementación de un requisito (*grave*).

⚠️ *Las líneas de código que no tienen requerimiento asociado suelen ser "cosas que el desarrollador imaginó que podrían necesitarse". Estas decisiones no justificadas pueden derivar en funcionalidad innecesaria y difícil de mantener.*

8. Importancia de los requisitos bien definidos

- **45–56%** de los errores tienen origen en esta fase.
- Detectarlos tarde incrementa el costo.
- **Fuente:** Misuno, Tom De Marco, David (1993)

9. Tipología de errores frecuentes

- **Hechos incorrectos:** se entendió mal el requerimiento (49%)
- **Omisiones:** requisitos incompletos o ausentes
- **Inconsistencias y ambigüedades:** uso del lenguaje natural

10. Requisitos No Funcionales

Características

- Abstractos e intangibles
- Difíciles de detectar y especificar
- Su omisión puede inutilizar el sistema
- Suelen estar "escondidos": si no se indagan, no aparecen solos

Ejemplos

- Rendimiento
- Seguridad

- Portabilidad
- Usabilidad

Clasificación según Somerville

- **Producto:** eficiencia, portabilidad, usabilidad
- **Organización:** implementación, procesos internos
- **Externos:** legales, interoperabilidad

En esta materia se usa la clasificación **funcionales vs no funcionales**, según si es un servicio o una restricción/atributo.

Puntos clave para el examen

- Clasificación y propiedades de requisitos
- Diferencias entre funcionales y no funcionales
- Ejemplos de cada tipo
- Trazabilidad entre requisitos y código
- Errores frecuentes
- Verificación y validación de requisitos

Trabajo Práctico en clase

Actividad individual (post-receso):

- Escribir:
 - 2 requisitos funcionales
 - 2 requisitos no funcionales
 - Contexto: sistema de biblioteca de la UNLaM
 - Usar formato: **"El sistema debe..."**
 - Opcional: requisitos del negocio o del proyecto (bien diferenciados)
-