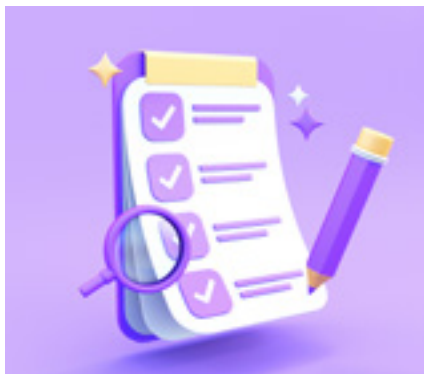




INGENIERÍA DE REQUISITOS

TÉCNICAS PARA LA IDENTIFICACIÓN DE REQUISITOS

TÉCNICAS PARA LA IDENTIFICACIÓN DE REQUISITOS



La identificación de requisitos es una de las etapas más críticas dentro del proceso de ingeniería de software, dado que permite establecer qué debe hacer un sistema para satisfacer las necesidades de sus usuarios. En este contexto, se reconocen diversas técnicas que facilitan la obtención, clarificación y documentación de los requerimientos tanto funcionales como no funcionales. Estas técnicas permiten comprender en profundidad, el dominio del problema y construir un producto alineado con los objetivos del negocio.

1. Uso de entrevistas para diferenciar entre requisitos funcionales y no funcionales.

Las entrevistas representan una técnica eficaz para explorar las necesidades del usuario y comprender a fondo cómo debe comportarse el sistema (Sommerville, 2011). En el análisis de requisitos, permiten distinguir entre acciones esperadas del sistema (funcionales) y condiciones o restricciones bajo las cuales debe operar (no funcionales).

- ✓ **Ejemplo.** Durante una entrevista, un cliente de una aplicación de ventas menciona: “Necesito que el sistema me permita generar facturas en PDF y que esté disponible todos los días sin fallas”. De esta declaración se extrae un requisito funcional (“generar facturas en PDF”) y uno no funcional (“alta disponibilidad del sistema”).

2. Aplicación de prototipos para validar requisitos funcionales.

El uso de prototipos facilita la visualización de funcionalidades antes de ser implementadas, lo que contribuye a la validación temprana de los requisitos funcionales. Los usuarios pueden interactuar con versiones simplificadas del sistema y confirmar si lo planteado satisface sus necesidades reales.

- ✓ **Ejemplo.** En el desarrollo de un sistema de reservas para un gimnasio, se presenta un prototipo que simula el proceso de selección de clases y horarios. Gracias a la retroalimentación del usuario, se ajustan los flujos de navegación y se corrigen funcionalidades mal interpretadas, asegurando una correcta definición de los requisitos funcionales.

3. Observación para detectar limitaciones del entorno (requisitos no funcionales).

La observación directa del entorno de trabajo permite identificar aspectos no explícitamente mencionados por los usuarios, pero que influyen en los requisitos no funcionales, como el tiempo de respuesta, la facilidad de uso o restricciones técnicas.

- ✓ **Ejemplo.** En un laboratorio clínico, se observa que los técnicos deben ingresar resultados de pruebas en menos de 5 segundos por paciente. Esta necesidad se traduce en el siguiente requisito no funcional: “El sistema debe registrar datos en menos de 5 segundos por ingreso”.

4. Análisis de casos reales: identificación de requisitos funcionales y no funcionales.

Estudiar casos reales permite aplicar técnicas de identificación de requisitos en contextos concretos, extrayendo y clasificando tanto requisitos funcionales como no funcionales.

Ejemplo práctico.

Funcionales:

- ✓ El sistema debe permitir agregar nuevos productos al inventario.
- ✓ El sistema debe generar reportes mensuales de stock.

No funcionales:

- ✓ El sistema debe tener una interfaz intuitiva para usuarios sin experiencia técnica.
- ✓ El sistema debe funcionar en navegadores web modernos y dispositivos móviles.

5. Consejos y errores comunes al aplicar estas técnicas.

Consejos:

- ✓ Formular preguntas específicas en entrevistas para distinguir claramente, ambos tipos de requisitos.
- ✓ Usar prototipos centrados en flujos claves durante sesiones breves.
- ✓ Complementar la observación con anotaciones sobre el entorno físico y digital.

Errores comunes:

- ✓ Asumir que todo lo expresado por el cliente es funcional (muchos requisitos no funcionales son implícitos).
- ✓ No actualizar los requisitos tras validar prototipos.
- ✓ Ignorar las condiciones técnicas del entorno durante la observación.

Validación y verificación de requisitos

En el desarrollo de software, interpretar y cumplir correctamente los requisitos es decisivo para el éxito del proyecto. Para garantizar que el producto se ajuste a las necesidades planteadas, se realizan dos actividades claves: la verificación y la validación de requisitos (Sommerville, 2011). Aunque suelen confundirse, ambas tienen objetivos y momentos distintos en el ciclo de vida del desarrollo.

Verificación de requisitos

La verificación consiste en comprobar que los requisitos estén correctamente documentados, completos, coherentes y sin ambigüedades. Responde a la pregunta: ¿Estamos construyendo correctamente los requisitos? Se utilizan técnicas como revisiones, inspecciones y análisis de trazabilidad.

- ✓ **Ejemplo.** En un sistema de gestión académica, se encuentra el requisito “El sistema debe generar reportes rápidamente”. Esta redacción es ambigua, por lo que se corrige a: “El sistema debe generar reportes de rendimiento académico en un tiempo menor a 5 segundos”.

Validación de requisitos

La validación busca garantizar que los requisitos reflejen las necesidades, expectativas y objetivos del usuario. Responde a la pregunta: ¿Estamos construyendo el sistema correcto? Suele realizarse con los stakeholders mediante reuniones, simulaciones o prototipos (Águila Cano, 2022).

- ✓ **Ejemplo.** En el mismo sistema académico, se presenta un prototipo a los docentes. Uno de ellos sugiere que el sistema también permita ingresar observaciones por evaluación. Este requerimiento se incorpora al documento de requisitos funcionales tras la validación.

Importancia conjunta

Ambas actividades son complementarias. Verificar sin validar puede llevar a un sistema técnicamente correcto pero inútil para el usuario; validar sin verificar puede resultar en incoherencias que afecten el desarrollo. Aplicarlas sistemáticamente desde fases tempranas reduce errores, retrabajos y aumenta la satisfacción del cliente.

Técnicas y herramientas para la verificación de requisitos

1. **Revisiones e inspecciones técnicas.** Evaluaciones sistemáticas por parte de expertos, para detectar ambigüedades o errores formales.
2. **Listas de chequeo (checklists).** Conjuntos de criterios para verificar claridad, viabilidad y trazabilidad de los requisitos.
3. **Matrices de trazabilidad.** Permiten rastrear cada requisito desde su origen hasta su implementación y prueba.
4. **Herramientas CASE.** Software como IBM DOORS, Enterprise Architect o Visual Paradigm, para modelar y verificar requisitos automáticamente.

Técnicas y herramientas para la validación de requisitos

1. **Reuniones de revisión con usuarios.** Talleres con stakeholders para confirmar que los requisitos reflejan sus expectativas.
2. **Prototipos.** Versiones preliminares del sistema que permiten validar funcionalidades antes del desarrollo completo.

3. **Simulaciones y escenarios.** Representaciones de comportamiento del sistema en contextos reales.
4. **Casos de uso y diagramas UML.** Descripciones de interacciones entre el sistema y los actores para validar comportamientos funcionales.

Bibliografía

- ✍️ Águila Cano, I. M. D. (2022). Fundamentos de ingeniería de los requisitos. Editorial Universidad de Almería. <https://elibro.net/es/lc/tecnologicadeloriente/titulos/280517>
- ✍️ Sommerville, I. (2011). Ingeniería de software (9ª ed.). Pearson Educación. https://gc.scalahed.com/recursos/files/r161r/w25469w/ingdelsoftwarelibro9_compressed.pdf