Requisitos de un proyecto

Índice

1. ¿Qué es un requisito? 1
2. Requisitos funcionales 2
3. Requisitos no funcionales 2
4. Requisitos funcionales vs no funcionales 3
5. ¿De dónde obtenemos los requisitos? 4
6. Ejemplo 4

1. ¿Qué es un requisito?

Un **requisito** es una necesidad, condición o característica que debe cumplir una solución o proyecto. Es decir, describe claramente qué necesita o espera el usuario del producto final que se va a desarrollar.

Los requisitos son muy importantes ya que nos sirven para entender y concretar qué es lo que necesita el cliente. Nos permite definir el alcance, orientar el diseño, planificar las tareas, estimar los recursos, realizar pruebas y comprobar si el resultado final cumple con su objetivo.

Una gran parte de los posibles problemas que pueden surgir en el desarrollo de un proyecto pueden tener su origen en los requisitos si estos están incompletos, son ambiguos, contradictorios, cambiantes o se han entendido de forma incorrecta.

Por ejemplo, si una empresa quiere quiere un sistema de copias de seguridad, una petición poco concreta sería: "Queremos un sistema de backups bueno".

Para poder desarrollar el proyecto correctamente, necesitaremos transformar eso en unos requisitos más concretos como pueden ser:

- El sistema deberá realizar una copia diaria.
- Deberá conservar las copias durante al menos 30 días.
- Deberá permitir restaurar archivos individuales.
- Las copias deberán almacenarse en un dispositivo diferente al servidor.

Con estos requisitos ya sí podremos diseñar, desarrollar y comprobar el correcto funcionamiento del proyecto.

Los requisitos son importantes porque nos sirven para:

- Entender qué necesita el cliente
- Definir el alcance del proyecto
- Planificar las tareas
- Estimar los recursos
- Realizar pruebas
- Comprobar si el proyecto cumple su objetivo

Una gran parte de los problemas de un proyecto pueden tener su origen en requisitos incompletos, ambiguos, contradictorios, mal entendidos...

Un buen requisito debería cumplir las siguientes **características**:

- Ser claro: debe entenderse fácilmente.
- Ser específico: debe evitar expresiones ambiguas.
- Ser medible o verificable: debemos poder comprobar si se cumple.
- No ambiguo: no debería tener diferentes interpretaciones.

2. Requisitos funcionales

Los **requisitos funcionales** describen qué debe hacer el sistema. Son las funcionalidades, acciones, servicios, etc. que el sistema debe proporcionar.

A continuación, se muestran diferentes ejemplos de requisitos funcionales, referentes a diferentes tipos de sistemas.

- Sistema de usuarios
    - El sistema permitirá crear usuarios.
    - El sistema permitirá eliminar usuarios.
    - El sistema permitirá modificar usuarios.
    - El sistema permitirá iniciar sesión.
- Sistema de copias de seguridad
    - El sistema realizará las copias automáticamente al finalizar la semana.
    - El sistema permitirá restaurar archivos.
    - El sistema generará alertas cuando una copia de falle en su creación.
- Sistema de monitorización de recursos
    - El sistema monitorizará el uso de CPU.

    - El sistema monitorizará la memoria RAM.
    - El sistema enviará una alerta cuando el uso de CPU supere un determinado valor.
- Infraestructura
    - El servidor deberá proporcionar DNS.
    - El servidor deberá proporcionar DHCP.
    - El sistema deberá permitir realizar copias de seguridad.

3. Requisitos no funcionales

Los **requisitos no funcionales** describen características, restricciones o condiciones que debe cumplir el sistema. No se centran en qué tiene que hacer el sistema, si no en cómo debe hacerlo, o qué condiciones debe cumplir.

Los requisitos no funcionales **pueden ser de muchos tipos**. A continuación, se enumeran diferentes tipos de requisitos no funcionales junto a un ejemplo de cada uno de ellos.

- Rendimiento. Ej: El sistema deberá responder a las peticiones en menos de 1 segundo bajo condiciones normales.
- Seguridad. Ej. Las contraseñas deberán almacenarse utilizando un mecanismo de hash seguro.
- Disponibilidad. Ej: El servicio deberá tener una disponibilidad mínima del 99%.
- Capacidad. Ej: El sistema deberá soportar al menos 100 usuarios simultáneos.
- Compatibilidad. Ej: La aplicación deberá funcionar en Windows y Linux.
- Escalabilidad. Ej: La infraestructura deberá poder ampliarse hasta 500 usuarios sin sustituir los servidores principales.
- Mantenibilidad. Ej: la configuración deberá estar documentada y permitir modificar los parámetros principales sin alterar el código.

4. Requisitos funcionales vs no funcionales

Para que no haya lugar a dudas, vamos a ver diferentes ejemplos en los que se van a relacionar requisitos funcionales con no funcionales.

Ejemplo 1: sistema de copias de seguridad

- **RF**: El sistema realizará una copia de seguridad diaria.
- **RNF**: La copia de seguridad deberá realizarse en menos de 30 minutos.

Ejemplo 2: sistema de inicio de sesión

- **RF**: El usuario podrá iniciar sesión
- **RNF**: El proceso de autenticación deberá completarse en menos de 2 segundos.

Ejemplo 3: aviso ante fallos

- **RF:** El sistema enviará un correo cuando falle una copia de seguridad.
- **RNF**: El aviso ante un fallo de copia de seguridad deberá enviarse en menos de 5 minutos desde que se detecte el fallo.

5. ¿De dónde obtenemos los requisitos?

Los requisitos generalmente no se los inventa el equipo de desarrollo, si no que se obtienen por diferentes métodos, generalmente involucrando al cliente.

Algunos **métodos para obtener requisitos** son:

- Entrevistas: hablar con usuarios o clientes. Las entrevistas deben ser guiadas y con preguntas concretas.
- Cuestionarios: recoger información de muchas personas.
- Observación: ver cómo se desenvuelven los potenciales usuarios del sistema.
- Documentación: análisis de la normativa, documentación técnica, procedimientos y demás información relevante.
- Reuniones: analizar las necesidades con las personas implicadas, ya sean los clientes o potenciales usuarios.
- Prototipos: crear una versión inicial para obtener feedback.
- Análisis de un sistema ya existente: estudiar cómo funciona actualmente un sistema o infraestructura para analizar cómo lo podemos mejorar.

Como hemos visto, generalmente los requisitos nos vienen dados de diferentes personas que pueden ser:

- Cliente
- Usuarios
- Responsables de una organización o empresa
- Administradores
- Responsables de seguridad

No todas las personas o todos los roles dentro de una misma organización tendrán las mismas necesidades ni utilizarán las mismas partes de un sistema. Es importante recoger los requisitos que nos puedan dar personas con diferentes funciones y puntos de vista.

6. Ejemplo

Supongamos que tenemos el proyecto: "Implantación de un sistema de monitorización para una pequeña empresa."

Objetivo:

Implantar un sistema que permita supervisar los principales servidores de la empresa.

Requisitos funcionales:

- RF01. El sistema permitirá registrar servidores.
- RF02. El sistema monitorizará el uso de CPU.
- RF03. El sistema monitorizará la memoria RAM.
- RF04. El sistema monitorizará el espacio disponible en disco.
- RF05. El sistema generará alertas cuando se superen determinados límites.

Requisitos no funcionales:

- RNF01. El sistema deberá soportar al menos 20 servidores.
- RNF02. La información deberá actualizarse como máximo cada 60 segundos.
- RNF03. El acceso a la administración deberá requerir autenticación.
- RNF04. La información histórica deberá conservarse durante al menos 30 días.