

Mi Análisis de Requerimientos en el Desarrollo de Software

En mi experiencia dentro de la ingeniería de software, considero que la definición de requisitos es el cimiento de cualquier proyecto exitoso. He aprendido que no basta con saber qué botones debe presionar el usuario; necesito comprender profundamente el entorno en el que operará mi solución. Por ello, divido mi análisis en dos pilares fundamentales: los requerimientos **funcionales** y los **no funcionales**.

Característica	Mis Requerimientos Funcionales	Mis Requerimientos No Funcionales
Mi Definición	Describo las funciones, tareas o servicios específicos que mi sistema debe ejecutar obligatoriamente.	Describo los atributos de calidad y las restricciones que imponen límites al comportamiento del sistema.
Mi Enfoque	Me concentro en el " qué " debe hacer mi software para satisfacer al usuario.	Me centro en el " cómo " debe operar mi software bajo presión o condiciones reales.
Mi Origen	Los extraigo directamente de las necesidades del cliente y de los procesos de negocio que busco automatizar.	Los establezco basándome en estándares técnicos, normativas legales y criterios de rendimiento óptimo.
Mi Medición	Los valido mediante pruebas de aceptación (el sistema cumple o no	Los cuantifico mediante métricas específicas (latencia en

	cumple con la tarea).	milisegundos, capacidad de usuarios concurrentes).
Ejemplos Propios	Procesamiento de pagos, registro de usuarios, envío de notificaciones y generación de facturas.	Cifrado de datos (seguridad), tiempo de carga menor a 2 segundos, y diseño responsivo.
Consecuencia	Si fallo aquí, mi sistema no sirve para el propósito con el que fue diseñado.	Si fallo aquí, mi sistema puede ser funcional pero será frustrante, lento o vulnerable para el usuario.

Profundización del Análisis

Desde mi punto de vista, la relación entre ambos es simbiótica. He observado que, a menudo, los desarrolladores nos enfocamos excesivamente en lo funcional (que la app "haga cosas") y descuidamos lo no funcional. Sin embargo, un software que procesa datos correctamente pero tarda 30 segundos en cargar, es un software destinado al fracaso.

Por lo tanto, mi metodología de trabajo siempre busca un equilibrio: mientras diseño la lógica interna y las bases de datos para las funciones, establezco simultáneamente las metas de seguridad y escalabilidad que permitirán que mi producto crezca sin colapsar. En mi flujo de trabajo, un requerimiento no funcional es tan crítico como una función principal, ya que define la supervivencia del sistema a largo plazo.