

## DEFINICIÓN DE REQUERIMIENTO

Requerimiento es una necesidad o solicitud cuyo objetivo es resolver un problema.  
*Ejemplo: requiero un nuevo integrante en mi equipo.*

También se puede encontrar el término Requisito como sinónimo de Requerimiento. **Un Requisito es una condición o característica necesaria para algo.** *Ejemplo: para entrar al equipo debe ser estudiante de sistemas, con experiencia en Java, entre 20 y 25 años.*

Por lo tanto, los Requisitos son condiciones o capacidades que debe tener un sistema (Requerimientos del Sistema) para satisfacer las necesidades del usuario (Requerimientos del Usuario).

## REQUERIMIENTOS EN LA METODOLOGÍA

Si nos situamos en los pasos de la Metodología ya estudiada, definimos que:

- En la etapa de **RELEVAMIENTO** se relevan, capturan, escuchan, observan, detectan y documentan las necesidades del usuario (requerimientos del usuario).
- En la etapa de **ANÁLISIS DE REQUISITOS** se definirán los Requerimientos (requerimientos del sistema) que deberá tener el sistema para cumplir con todas esas necesidades (requerimientos del usuario).
- En la etapa de **GESTION DEL PROYECTO** se administrarán los requerimientos, incorporando los cambios de manera controlada para poder seguir cumpliendo con los tiempos planificados.

## IMPORTANCIA DE LOS REQUERIMIENTOS PARA EL ÉXITO DE UN PROYECTO

Una empresa dedicada a las estadísticas, que cada dos años realiza un estudio acerca del estado del desarrollo del software realizó un estudio en más de 350 empresas acerca del resultado de más de 8000 proyectos de software:

- El 31% fueron cancelados antes de terminar.
- El 25% cumplieron con el proyecto original en tiempos y costos.
- El 44% fallaron.

Los que fallaron, ¿Por qué fallaron?

- 14% Requerimientos incompletos
- 13% Poco involucramiento del usuario final
- 11% Falta de recurso
- 10% Expectativas no realistas
- 10% Sin soporte del área gerencial
- 9% Sin control de cambios en los Requerimientos
- 9% Falta de planificación
- 8% El sistema quedó obsoleto
- Otros

Es decir que aproximadamente la mitad de los **proyectos fallaron por causas que directamente o indirectamente están motivadas por malos procesos de ingeniería de**

**requerimientos:** no reflejan las necesidades reales, son ambiguos, son inconsistentes o incompletos, son difíciles o caros de cambiar.

La ingeniería de requerimientos es una cuestión cultural de las organizaciones. En la mayoría de los proyectos de software, se **comienza a programar demasiado pronto**, concentrando el esfuerzo en producir código para obtener rápidos resultados.

¿Pero cómo se genera una rápida solución sin tener conocimientos específicos del problema?

Lo más probable es que se genere una solución rápida pero no eficaz y que ya no cumple con los requisitos del sistema.

Por lo tanto, la principal consecuencia de los problemas asociados con los requerimientos es el rehacer algo que está hecho -> **RETRABAJO**.

#### CARACTERÍSTICAS DE UN REQUERIMIENTO CORRECTAMENTE DEFINIDO

- Completo: contiene toda la información necesaria y no necesita ser expandido en otro ni dividido. *Ejemplo: Cuando definen cambiar algo en una pantalla deben definir los 3 idiomas.*
- No ambiguo: debe tener una y solo una interpretación. *Ejemplo: Alta disponibilidad.*
- Verificable: debe ser demostrado o probado su cumplimiento. *Ejemplo: Tiempo de transferencia desde el sistema a Internet. ¿Cómo se verifica su cumplimiento?*
- Consistente: no deberá contradecir otros requerimientos. *Ejemplo: dos requerimientos que se llamen diferente y digan lo mismo.*

#### VENTAJAS DE REQUERIMIENTOS CORRECTOS

- Identificación de riesgos de manera temprana. Atacar más rápido al problema
- Análisis y Desarrollo más claro y rápido (menos malos entendidos, reducción del Retrabajo).
- Bases para un buen diseño.
- Definición clara de casos de prueba para realizar un Testing claro, completo y rápido.

## TIPOS DE REQUERIMIENTOS

- **Funcionales:** describen las funcionalidades del sistema, es decir lo que el sistema debe hacer, su comportamiento específico. Describen las transformaciones que el sistema realiza sobre las entradas para producir salidas. Deben estar redactados de tal forma que sean comprensibles para usuarios sin conocimientos técnicos avanzados (de Informática, se entiende) *NOTA: A veces, también es conveniente indicar lo que no hará el sistema. Los requisitos funcionales definen qué debe hacer un sistema.*

- **No funcionales:** son los atributos o características que definen el cómo el sistema realizará el trabajo. Pueden considerarse como las restricciones planteadas al sistema respecto a cómo los requerimientos funcionales son implementados. *Los requisitos no funcionales definen cómo debe ser el sistema.* Se clasifican en:

- **De Producto o Calidad:**

Limites o restricciones sobre el comportamiento del sistema

- **Mantenibilidad:** significa que puede cambiar sin generar un gran impacto.
- **Usabilidad:** Amigable para el usuario, respecto a la navegabilidad de las pantallas.
- **Performance:** eficiencia esperada, es decir buen desempeño del frente a la demanda esperada.
- **Disponibilidad:** La disponibilidad del sistema debe ser continua con un nivel de servicio para los usuarios de 7 días X 24 horas.
- **Seguridad:** La solución debe reflejar patrones de seguridad teniendo en cuenta la alta sensibilidad de la información que maneja de acuerdo a las especificaciones funcionales dadas y a las políticas, normas y estándares de seguridad requeridas

- **Organizacionales:**

Se derivan de las políticas y procedimientos de la organización como por ejemplo estándares de procesos o requerimientos de implementación.

- **De entorno u organizacionales:** Procedimientos operativos que describen como será usado el sistema dentro del contexto de la organización.
- **De diseño, desarrollo e implementación:** Lenguaje de programación a usar, estándares de codificación, patrones (y antipatrones) de diseño y programación, herramientas para gestionar el desarrollo de software, entorno de desarrollo de software (ambiente de desarrollo), entorno de pruebas de software (ambiente de pruebas), entre otros aspectos.

- **Externos:**

- **Regulatorios:** Leyes y reglamentos que establecen que debe hacer el sistema y como debe hacerlo para cumplirlas

- **Éticos:** Requerimientos que aseguran que el sistema será aceptable para el usuario, público en general y se adapta a las costumbres de la sociedad en la que se desenvuelve o a la que presta servicios.
- **Legislativos:** Características que debe cumplir el sistema para cumplir con la ley, por ejemplo en el área de contabilidad (normas contables y estándares financieros), requerimientos de seguridad industrial (para sistemas críticos), entre otros aspectos.

La distinción entre requerimientos funcionales y no funcionales no siempre resulta evidente...

*Ejemplo: La seguridad puede interpretarse inicialmente como un requerimiento no funcional al principio. No obstante, su elaboración puede conducir a nuevos requerimientos funcionales, como la necesidad de autenticar a los usuarios del sistema.*

¿Estos requerimientos están bien o mal definidos?

El sistema debe ser seguro.

MAL

¿Por qué?

Objetivo general, vago. ¿Qué tan seguro es “seguro”? ¿En qué situaciones? ¿Existe una norma a cumplir? ¿En qué secciones? ¿Qué debe ocurrir si el sistema no puede funcionar tan rápido como se requiere?

Todas las comunicaciones externas entre los servidores de datos, la aplicación y el cliente del sistema deben estar cifradas utilizando el algoritmo RSA.

BIEN

Se especifica qué tipo de comunicaciones necesitan ser encriptadas. Se sabe qué algoritmo usar y validar.

- El sistema será lo más fácil de utilizar posible.
- El sistema proporcionará una respuesta rápida al usuario.
- El sistema se recuperará automáticamente tras producirse un fallo.

MAL

¿Por qué?

Objetivos generales, vagos y abiertos a distintas interpretaciones

- Un usuario experimentado debe ser capaz de utilizar todas las funciones del sistema tras un entrenamiento de 2 horas, tras el cual no cometerá más de 3 errores diarios en media.
- Cuando haya hasta 100 usuarios accediendo simultáneamente al sistema, su tiempo de respuesta no será en ningún momento superior a 2 segundos.
- Ante un fallo en el software del sistema, no se tardará más de 5 minutos en restaurar los datos del sistema (en un estado válido) y volver a poner en marcha el sistema.

BIEN

¿Por qué?

Requisitos verificables

Para facilitar el uso del editor gráfico, se podrá activar y desactivar una rejilla que permitirá alinear las figuras del diagrama. Cuando se ajuste la figura al tamaño de la pantalla, se reducirá el número de líneas de la rejilla para que no se dificulte la visualización del diagrama.

MAL

¿Por qué?

Amalgama de varios requisitos.

El editor permitirá el uso de una rejilla de líneas horizontales y verticales que aparecerán dibujadas tras el diagrama.

Justificación: La rejilla facilita la creación de diagramas cuidados en los que las figuras se puedan alinear con facilidad

BIEN

¿Por qué?

Preciso, conciso y justificado correctamente.

El usuario del Call Center deberá poder cargar en el Sistema un “Pedido a Sistemas”, en el cual ingresará la fecha de solicitud, la prioridad y una descripción del mismo. El sistema deberá validar si el usuario está autorizado para realizar dichos pedidos. Para esto deberá consultar una tabla SQL de permisos de Usuario, que es administrada por el área de seguridad. En caso de ser un Usuario no autorizado, aparecerá un pedido nuevo y se le devolverá al usuario un código único de identificación. A la vez, el sistema deberá tener conexión con el Outlook y le deberá llegar al usuario un correo, con la información del pedido realizado, la fecha y el estado de pedido “Ingresado”. Esta pantalla de alta de pedido deberá ser amigable y fácil de usar, deberá poder generar reportes en pdf con pedidos pendientes y pedidos realizados por usuarios.