



Guía para la Evaluación y Selección de Soluciones de Software

Guía para la especificación de requerimientos no funcionales

Contenido

GUÍA PARA LA EVALUACIÓN Y SELECCIÓN DE SOLUCIONES DE SOFTWARE	3
GUÍA PARA LA ESPECIFICACIÓN DE REQUERIMIENTOS NO FUNCIONALES	3
1. Seguridad	3
1.1. Seguridad de Acceso	3
1.2. Integridad	3
1.3. No repudio.....	3
1.4. Autenticidad	3
1.5. Transferencia de información	3
1.6. Almacenamiento de información.....	3
1.7. Rastreo de transacciones.....	4
1.8. Seguridad de la aplicación.....	4
2. Eficiencia	4
2.1. Usuarios	4
2.2. Volúmenes de información esperados.....	5
2.3. Transacciones estimadas.....	5
2.4. Volúmenes de información por transacción	5
3. Compatibilidad.....	5
4. Usabilidad	6
5. Mantenimiento.....	6
6. Interoperabilidad.....	7
7. Confiabilidad.....	7
8. Portabilidad	7
9. Restricciones de Diseño y Construcción.....	7
10. Documentación y capacitación necesaria	8
11. Legales y reglamentarios.....	8
12. Soporte en la instalación	8
13. Soporte posterior a la instalación	8
14. Corrección de errores	9
15. Modificaciones, en caso de requerirse	9
16. Garantía del software	9
17. Nuevas liberaciones (releases) del software.....	10

GUÍA PARA LA EVALUACIÓN Y SELECCIÓN DE SOLUCIONES DE SOFTWARE

Guía para la especificación de requerimientos no funcionales

1. Seguridad

Garantía de que la información y los procesos del sistema sólo sean accedidos por los usuarios o sistemas autorizados y para las funciones (lectura o modificación) determinados, se debe considerar:

1.1. Seguridad de Acceso

Indicar cómo se controlará el acceso y privilegios de los usuarios, por ejemplo, por medio de:

- ◆ Usuario y contraseña.
- ◆ Dispositivos físicos como: el código de barras o tarjetas.
- ◆ Certificado digital, contraseña.
- ◆ Elementos biométricos, entre otros.

1.2. Integridad

Revisar que tanto el código como las bases de datos, módulos o componentes del software no permitan accesos o modificaciones no autorizados.

1.3. No repudio

Mantener un registro de las operaciones del software (logs del Sistema Operativo, Bases de Datos, Funciones del software, etc.), para poder identificar las acciones o eventos que sufrieron repudio o denegación del servicio y evitar que no se repita.

1.4. Autenticidad

Capacidad de demostrar la identidad de una persona o un recurso.

1.5. Transferencia de información

En su caso indicar si por la criticidad de la información, es necesario considerar algún mecanismo de seguridad o protocolo especial, por ejemplo: Transferencia en claro, SSL, HTTPS, SFTP, FTP.

1.6. Almacenamiento de información

Indicar la forma en que la información será almacenada, por ejemplo: En claro o cifrada. Además, identificar si se requiere algún mecanismo de protección para la base de datos.

1.7. Rastreo de transacciones

Indicar si se requerirá un registro y seguimiento de las transacciones en el sistema, por ejemplo, una bitácora (esto tendrá que ser especificado también como un requerimiento funcional).

1.8. Seguridad de la aplicación

Aspectos en general de protección de las aplicaciones como, por ejemplo: evitar la inyección de código malicioso.

2. Eficiencia

Mide el desempeño (comportamiento) del software bajo determinadas condiciones de carga (usuarios conectados simultáneamente, recursos usados, velocidad de la red o transmisión de datos, etc.), se debe considerar:

- A) Comportamiento temporal.** Los tiempos de respuesta y procesamiento y los ratios de throughput de un sistema cuando lleva a cabo sus funciones bajo condiciones determinadas en relación con un banco de pruebas establecido (benchmark).
- B) Utilización de recursos.** Las cantidades y tipos de recursos utilizados cuando el software lleva a cabo su función bajo condiciones determinadas:
 - ◆ Cantidad de memoria RAM usada
 - ◆ Tiempo de respuesta del sistema (latencia de red, respuesta del servidor, respuesta de la base de datos y respuesta del código)
 - ◆ Cantidad de procesos que corre (uso de procesador)
 - ◆ Recursos de red usados (llamadas a la base de datos, paquetes generados en la red por cada transacción)
- C) Capacidad.** Grado en que los límites máximos de un parámetro de un producto o sistema software cumplen con los requisitos.
- D) Volúmenes a soportar.** Indicar con la mayor precisión posible los volúmenes esperados en cuanto a las siguientes cuatro dimensiones: Usuarios, Información, Transacciones y Volúmenes de información por transacción.

2.1. Usuarios

Si es necesario, clasificar usuarios por perfil.

TOTAL DE USUARIOS	USUARIOS CONCURRENTES	INCREMENTO/PERIODO (ANUAL MENSUAL)
Cuantificar el universo de usuarios que podrán acceder al sistema a determinada fecha.	Cuantificar el total de usuarios que podrán acceder al sistema al mismo tiempo.	En su caso, indicar la cantidad o porcentaje de crecimiento de usuarios en un periodo determinado.

2.2. Volúmenes de información esperados

INFORMACIÓN	TOTAL ESTIMADO AL [FECHA]	INCREMENTO (ANUAL MENSUAL)
Nombre del universo de información, por ejemplo: Expedientes, Pacientes, Solicitudes, etc.	Cantidad de registros estimados que se incorporarán al sistema a una fecha determinada.	Cantidad o porcentaje de crecimiento de los registros por periodo.

2.3. Transacciones estimadas

TRANSACCIÓN	R/W/RW	TRANSACCIONES EN PERÍODO CRÍTICO	TRANSACCIONES EN PERÍODO NORMAL	ESCALA	PERIODO(S) CRÍTICO(S)	HORA(S)
Descripción de un tipo de transacción; puede ser un Caso de Uso en particular o alguno de sus flujos alternos.	Indica si la transacción implica lectura/escritura/lectura-escritura.	Total de transacciones esperadas en un periodo crítico. Se entiende por total de transacciones cuántas veces se realiza la "petición" de dicha transacción.	Total de transacciones esperadas en un periodo no crítico.	Escala de tiempo utilizada para expresar el número de transacciones, por ejemplo: Minuto / Día / Semana / Mes / Año	Precisión del periodo crítico de las transacciones. Indicar como: fecha inicio – fecha fin, o bien "Última semana de cada mes", por ejemplo.	En su caso, el rango de horas a considerar como periodo crítico de transacciones.

2.4. Volúmenes de información por transacción

(Aplicable cuando una transacción en particular lleva asociado un volumen de información significativo).

TRANSACCIÓN	VOLUMEN MÁXIMO	VOLUMEN MÍNIMO	MODA
Descripción de la instancia de una transacción.	Cantidad máxima de registros que pueden tener asociada la transacción.	Cantidad mínima de registros que pueden tener asociada la transacción.	Cantidad frecuente de registros que conlleva la transacción. Puede ser un número o bien rango(s) modal(es).

3. Compatibilidad

Capacidad de dos o más sistemas o componentes para intercambiar información y/o llevar a cabo sus funciones requeridas cuando comparten el mismo entorno hardware o software. Considerar:

A) Coexistencia. Capacidad del producto para coexistir con otro software independiente, en un entorno común, compartiendo recursos comunes sin detrimento.

- B) Interoperabilidad.** Capacidad de dos o más sistemas o componentes para intercambiar información y utilizar la información intercambiada.

4. Usabilidad

Característica de la aplicación para facilitar la operación al usuario final, incluyendo aspectos estéticos de la misma, cuando se usa bajo determinadas condiciones. Considerar lo siguiente:

- A) Capacidad de reconocimiento.** Los módulos, flujos, acciones y funciones deben permitir al usuario reconocer que el producto es adecuado para satisfacer sus necesidades.
- B) Facilidad de uso.** Mediante una navegación clara, ayuda en línea o consulta de manuales entre otros, permite al usuario aprender de forma casi intuitiva el uso del software.
- C) Protección contra errores de usuario.** Capacidad para alertar al usuario sobre posibles errores de operación del software.
- D) Apariencia y estilo.** El diseño de la interfaz debe permitir una navegación sencilla, agradable visualmente y satisfacer la interacción con el usuario.
- E) Accesibilidad.** Capacidad del software para permitir el uso por usuarios con determinadas características y capacidades diferentes.

5. Mantenimiento

Es opcional. Descripción de los elementos que facilitan la comprensión y la realización de las modificaciones futuras del software, incluye características relacionadas con la evolución, corrección y perfección del mismo, se debe considerar:

- A) Modularidad.** Característica del diseño del software para funcionar por módulos o componentes con suficiente independencia para minimizar el impacto del cambio en los demás módulos o componentes.
- B) Reusabilidad.** Eficiencia o capacidad del software para reutilizar código para funciones similares.
- C) Auditabilidad.** Facilidad para evaluar el impacto de los cambios en el código, estructura de datos, ambiente, interfaz, etc. y su alcance sobre el resto del software.
- D) Parametrización.** Facilidad de configurar los parámetros de operación del sistema, sin modificar el código, introducir defectos o degradar el desempeño.
- E) Capacidad para ser probado.** Facilidad con la que se pueden establecer criterios de prueba para un sistema o componente y con la que se pueden llevar a cabo las pruebas para determinar si se cumplen dichos criterios.

6. Interoperabilidad

Es opcional. Característica que describe si la aplicación debe compartir información o ser compatible con otras aplicaciones. Si se pretende interactuar con otro sistema indicar las formas de interoperabilidad, que en cuanto a intercambio de información pueden ser, por ejemplo:

- ◆ En línea
- ◆ En batch
- ◆ Por archivos de texto

7. Confiabilidad

Es opcional. Especificación del nivel de desempeño del software con respecto a la madurez, tolerancia a fallas y recuperación. Capacidad de un sistema o componente para desempeñar las funciones especificadas, cuando se usa bajo unas condiciones y periodo de tiempo determinados. Esta característica se subdivide a su vez en las siguientes subcaracterísticas:

- A) Madurez.** Capacidad del sistema para satisfacer las necesidades de fiabilidad en condiciones normales.
- B) Disponibilidad.** Capacidad del sistema o componente de estar operativo y accesible para su uso cuando se requiere.
- C) Tolerancia a fallos.** Capacidad del sistema o componente para operar según lo previsto en presencia de fallos hardware o software.
- D) Capacidad de recuperación.** Capacidad del producto software para recuperar los datos directamente afectados y restablecer el estado deseado del sistema en caso de interrupción o fallo.

8. Portabilidad

Es opcional. Capacidad del código, estructura de datos, interfaz, modelo o entorno de operación (Framework) de los módulos o componentes del software para ser transferidos de forma efectiva y eficiente a otro entorno de hardware o software. Considerar lo siguiente:

- A) Capacidad de instalación.** Facilidad del software para instalarse y desinstalarse en diferentes ambientes o entornos de hardware, software y telecomunicaciones.
- B) Reemplazabilidad.** Capacidad del software para ser sustituido por otro producto de software con funciones y propósitos similares bajo condiciones y entornos específicos.

9. Restricciones de Diseño y Construcción

Es opcional. Son requisitos que nos impone la naturaleza del dominio del problema. Estos pueden ser:

- ◆ Ajuste a estándares o lineamientos (p.e. una determinada manera de codificar un dato)
- ◆ Limitaciones hardware (por los equipos disponibles).
- ◆ Seguridad (por los distintos niveles de acceso a la información que deben tener los usuarios).
- ◆ Mantenimiento (se debe tener en cuenta la ampliación del sistema).
- ◆ Adaptación al entorno.
- ◆ Tecnologías a utilizar.
- ◆ Políticas de borrado.

10. Documentación y capacitación necesaria

Tomar en cuenta la generación o actualización de los siguientes documentos técnicos y para el usuario con la finalidad de contar con elementos que permitan la comprensión en el uso del sistema y su actualización, así como la capacitación de usuarios:

- ◆ Manuales de usuario
- ◆ Manuales técnicos
- ◆ Diagrama entidad relación
- ◆ Infografías
- ◆ Material audiovisual
- ◆ ¿Se planea capacitar a los usuarios del sistema para que lo usen o capaciten a otros?
- ◆ ¿Se planea realizar una prueba piloto de manera controlada?

11. Legales y reglamentarios

Es opcional. Necesidades impuestas por leyes, reglamentos, entre otros.

12. Soporte en la instalación

Es opcional. Describe el esquema de apoyo para el proceso de instalación y puesta en marcha del software, tomar en cuenta si:

- ◆ Existe un equipo técnico que apoye durante la instalación
- ◆ Se cuenta con manuales o tutoriales que documente los posibles problemas y soluciones que pueden presentarse
- ◆ Existen documentos que detallen la infraestructura o software complementarios para llevar a cabo la instalación.

13. Soporte posterior a la instalación

Describe el tipo y esquemas de apoyo que pueden darse una vez que el software se encuentra en operación, como son:

- ◆ Se cuenta con soporte técnico y cuál es el procedimiento para solicitar el soporte.
- ◆ Alcance del soporte (asesoría, configuración directa en la aplicación).

- ◆ Medios de contacto (Telefónico, en sitio, correo electrónico, acceso remoto).

14. Corrección de errores

Para detectar errores y su corrección se realizan pruebas que también ayudan a realizar mejoras, determinar la calidad del software, durabilidad y garantía, analizar el rendimiento, estabilidad, funcionalidad y eficacia. Se pueden realizar tres tipos de prueba:

- A) Positivas.** Son las pruebas para validar la funcionalidad total esperada.
- B) Negativas.** Son pruebas cuyo objetivo es romper la funcionalidad del software.
- C) "Happy path".** Son las pruebas para revisar si la funcionalidad es la esperada, no exhaustivas y sobre el flujo ideal del sistema. Para la realización de pruebas tomar en cuenta los siguiente:
 - ◆ Objetivo de las pruebas a realizar.
 - ◆ ¿Cómo se va a probar?; esquema de pruebas.
 - ◆ Duración de las pruebas.
 - ◆ Planes de acción para:
 - Corrección de errores detectados
 - Mejoras detectadas

15. Modificaciones, en caso de requerirse

Identificar las vías para realizar las modificaciones (ambiente de desarrollo y posterior integración o correcciones en producción con ventanas de mantenimiento alto riesgo). Dos de las principales razones por las cuales pueden existir modificaciones de software:

- ◆ **Correcciones.** Realizar un plan de acción que no afecte la operación del software en producción.
- ◆ **Mejoras.** Se recomienda aplicar cuando existe tiempo y no afecte en el cronograma y lanzamiento del software a producción.

16. Garantía del software

Nivel de certeza de que el software está libre de vulnerabilidades y funcione como se tiene previsto. Se debe observar lo siguiente:

- ◆ **Confiability.** No existen vulnerabilidades explotables, maliciosas o insertadas no intencionalmente.
- ◆ **Ejecución predecible.** Confianza de que el software, cuando se ejecute, funcione como debe de hacerlo.
- ◆ **Conformidad.** Conjunto planeado y sistemático de funciones interrelacionadas que garanticen los procesos de software y cumplimiento de los requisitos, normas y procedimientos.

17. Nuevas liberaciones (releases) del software

Las nuevas versiones se refieren a todas aquellas mejoras o nuevos alcances (requerimientos) identificados a lo largo de la operación del Software. En las nuevas versiones no se deberían incluir corrección de errores. Se deben considerar:

- ◆ Periodicidad programada de la liberación de versiones.
- ◆ Identificar previamente el contenido de cada versión.
- ◆ Definir un esquema de implementación (liberación) de cada versión (pilotaje)