udp Escuela de Informática y Telecomunicaciones
FACULTAD DE INGENIERÍA Y CIENCIAS

Tutoría para Examen de Grado

Sesión 1: Historias de Usuario e Ingeniería de Requerimientos

Docente: Mauricio Hidalgo Barrientos

Temario de la sesión

Recordar las Historias de Usuario

Recordar la Ingeniería de Requerimientos de Software

HISTORIAS DE USUARIO

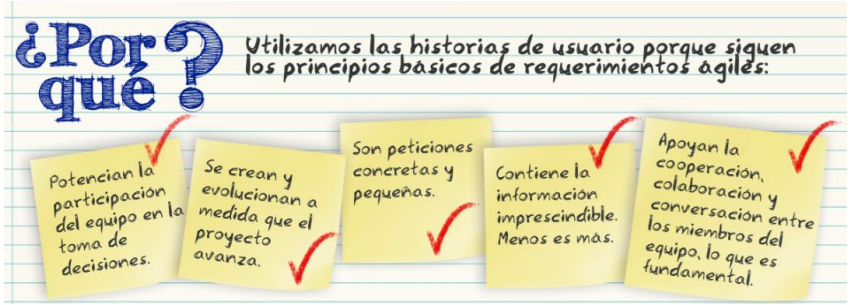
Historias de Usuario

¿Qué es una Historia de Usuario?

“Una historia de usuario es una explicación general e informal de una función de software escrita desde la perspectiva del usuario final. Su propósito es articular cómo proporcionará una función de software valor al cliente”.

Por lo general, una historia de usuario seguirá el siguiente formato:

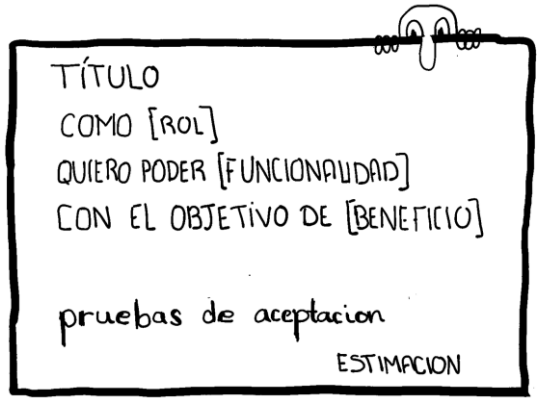
Como [perfil], quiero [objetivo del software], para lograr [resultado]”



Redacción de Historias de Usuario

Si bien es cierto que no existe una “receta de cocina”, se sugiere identificar los siguientes componentes:

- Perfil: Se refiere al rol del usuario final
- Necesidad: Se asocia al objetivo que tiene la función del Software para el usuario final
- Propósito: Hace referencia a la propuesta de valor que tendrá el Software para el usuario final



Identificación del Perfil

Para identificar el perfil del usuario final es necesario al público objetivo y tomar en cuenta a quién o quienes impactará la función de software.

Sugerencia de preguntas para identificar perfil del usuario:

- ¿Para quién estamos creando esta función de software?
- ¿Qué tipo de características del producto quiere el usuario final?
- ¿Cuáles son los datos vinculados al entorno del usuario final? (demográficos, poblacionales, etc.).

Importante

Puede haber varios perfiles en una historia de usuario determinada dependiendo del tamaño del público objetivo.

Ejemplo de perfil

Mauricio Hidalgo, un Jefe de Proyecto que tiene a cargo un equipo de desarrollo de cinco personas.

Descripción de la necesidad

Responde a cómo el usuario final utilizará una funcionalidad del software y por qué. Esto es fundamental para que tu equipo entienda por qué el público objetivo elegiría usar tu función.

Sugerencia de preguntas para describir la necesidad:

- ¿Qué está tratando de lograr el usuario final?
- ¿Cómo ayudará esta funcionalidad al usuario final para que pueda lograr sus objetivos?

Importante

Se debe considerar lo que el usuario final está buscando y la manera en que tu software lo ayudará a alcanzar sus objetivos por sobre las características específicas del software.

Ejemplo de necesidad

Ayudar a los desarrolladores a comprender como cumplir con sus entregas parciales contribuye al desarrollo de una solución global.

Definición del propósito

Para definir el propósito es importante analizar el panorama general de operación del software considerando cómo la función de software se ajusta a tus objetivos internos.

Sugerencia de preguntas para definir el propósito:

- ¿Cuál es el beneficio de la función de software?
- ¿Cuál es el problema que se está resolviendo?
- ¿Cómo se puede acoplar esta funcionalidad en metas más amplias?

Importante

El propósito es definir el valor de tu función del software en relación con los objetivos generales.

Ejemplo de propósito

Disminuir los tiempos de desarrollo a través del control de Sprints.

Ejemplo de Historias de Usuario

Ejemplo 1

Como usuario de una cuenta de correo electrónico, espero que mis datos se puedan guardar en las configuraciones de mi equipo personal para lograr un acceso más rápido a mi bandeja de entrada.

Ejemplo 2

Como comprador de una tienda online quisiera ver las críticas de un producto en particular para decidir si lo compro.

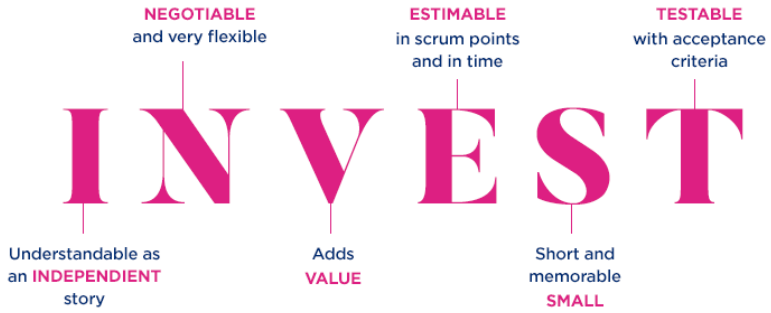
Ejemplo 3

Como docente de una asignatura quiero registrar las notas para obtener los promedios.

Consideración importante: INVEST

INVEST es un acrónimo asociado a criterios para ayudar a la redacción de Historias de Usuario.

- Independiente: Una Historia de usuario no debe depender de otras tareas (en lo ideal).
- Negociable: Debe dejar espacio para la discusión.
- Valiosa: Debe transmitir valor a través de ayudar a objetivos de largo plazo.
- Estimable: Se debe poder aproximar su ajuste a un sprint.
- Pequeña (small): El trabajo se debe poder realizar en poco tiempo.
- Comprobable (testable): Debe cumplir criterios de aceptación y pasar pruebas para verificar su calidad.



INGENIERÍA DE REQUERIMIENTOS

Dominio de Aplicación

Un dominio de aplicación constituye un límite de aislamiento para la seguridad, el control de versiones, la confiabilidad y la codificación.

En simple, son las actuales categorías de SW que se pueden desarrollar:

- Software de sistemas (compiladores, editores, etc) → Alta interacción con el HW
- Software de aplicación (puntos de venta, exploradores) → Hacen funciones específicas
- Software de ingeniería y ciencias (Matlab, Stellarium) → Uso diverso en investigación
- Software incrustado (termostatos, sensores) → Ejecutan funciones limitadas
- Software de línea de productos (Netflix, PS Store, Excel) → Capacidad específica para muchos consumidores diferentes
- Aplicaciones web (SAP Business One, Mercadolibre) → Webapps que grupan muchas aplicaciones
- Software de inteligencia artificial (Tensorflow y Keras) → Requieren gran capacidad de HW

Requerimientos de Software

Un requerimiento es la definición de las funciones, capacidades o atributos intrínsecos de un sistema de software.

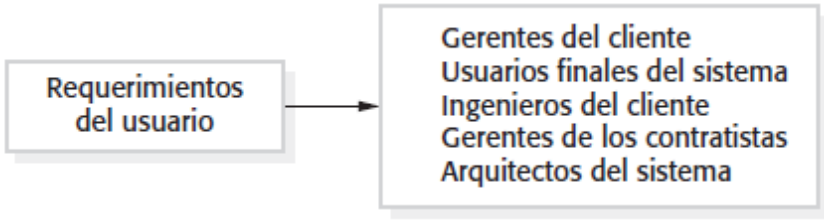
Características de los Requerimientos

Caracteristica	Descripción
Completo.	Todos los items necesarios para la especificación de la solución están incluidos.
Correcto.	Cada item es libre de errores.
Preciso,	no ambigüo y claro. Cada item es exacto y no vago; hay una sola interpretación; el significado de cada item es entendido; la especificación es fácil de entender.
Consistente.	Ningún item entra en conflicto con otro de la especificación.
Relevante.	Cada item es pertinente al problema y su solución.
Probable (testable).	Durante el desarrollo del programa y las pruebas de aceptación, es posible determinar si el item ha quedado satisfecho.
Factible.	Cada item puede ser implementado con las técnicas, herramientas, recursos y personal disponible y dentro de las limitaciones de costo y calendarización.
Registrabilidad (Traceable).	Cada item puede ser seguido durante cada etapa.
Libre de detalle de diseño.	La especificación de requerimientos son declaraciones de los requerimientos que se deben satisfacer por la solución del problema y no se deben ocultar por soluciones propuestas al problema.
Manejable.	Los requerimientos son expresados de tal manera que cada item puede ser cambiado sin causar un gran impacto a los demás items.

Requerimientos de Usuario

Son descripciones de los requerimientos funcionales y no funcionales de tal forma que sean compresibles por los usuarios del sistema sin conocimiento técnico detallado.

Tema:	Organizar Torneo			
Grupo:	Ronald Cárdenas, Israel Rey, Carlos Ojeda			
id	**Descripción**	**Prioridad**	**Tipo**	**Riesgo**
RU001	El sistema debe permitir el registro de los equipos de futbol.	Alta	F	ALTA
RU002	El sistema deberá ser capaz de poder planificar los encuentros de los equipos (partidos de futbol).	Media	F	Alta
RU003	Se podrá generar las tablas de posiciones de forma automática.	Alta	F	Alto
RU004	El sistema deberá identificara al mejor goleador	Media	F	Media
RU005	El sistema podrá publicar información acerca del torneo en curso en la web.	Media	F	Alta
RU006	Facilidad de uso: El sistema se diseñara de tal forma que permita al usuario una navegación intuitiva.		NF	
RU007	Seguridad: Las actualizaciones solo podrán ser efectuadas por el organizador del torneo.		NF	
RU008	Portabilidad: Se deberá tener acceso al sistema desde cualquier terminal con internet.		NF	



****Ejemplo de Israel Rey, QA test Engineer at Universidad Técnica Particular de Loja. extraído para fines académicos*

Requerimientos de Sistema

Son versiones extendidas de los requerimientos del usuario dado que agregan detalle y explican como el sistema debe proporcionar los requerimientos del usuario.

id	Descripción	Prioridad	Tipo	Riesgo
RU001	El sistema debe permitir el registro de los equipos de futbol.	Alta	F	ALTA
RU002	El sistema deberá ser capaz de poder planificar los encuentros de los equipos (partidos de futbol).	Media	F	Alta

Requerimientos del sistema → Usuarios finales del sistema
Ingenieros del cliente
Arquitectos del sistema
Desarrolladores de software

Deriva: RU001

- Al registrar un equipo de futbol el sistema presentara al usuario un formulario solicitando los siguientes campos :
  - Nombre del equipo
  - Ciudad
  - Nombre del Presidente del equipo
  - Nombre del Director Técnico
- Para el registro de los jugadores el sistema solicitara: el nombre, nacionalidad, fecha de nacimiento del futbolista, número de camiseta y rol que desempeña el jugador en la cancha (defensa, portero, etc.).
- El sistema debe almacenar toda esta información en una base de datos relacional.

****Ejemplo de Israel Rey, QA test Engineer at Universidad Técnica Particular de Loja. extraído para fines académicos*

Requerimientos de Sistema

Importante

- No deben tratar de cómo se debe diseñar o implementar.
- Deben describir el comportamiento externo del sistema y sus restricciones operativas.
- Son requisitos funcionales pues reflejan las operaciones que el sistema llevará a cabo.

id	Descripción	Prioridad	Tipo	Riesgo
RU001	El sistema debe permitir el registro de los equipos de futbol.	Alta	F	ALTA
RU002	El sistema deberá ser capaz de poder planificar los encuentros de los equipos (partidos de futbol).	Media	F	Alta

Planificar encuentros SRS002.
Tipo funcional
Deriva: RU002

- El sistema deberá contar con formularios que permitan al usuario especificar el número de equipos que tendrá cada grupo o división. El sistema contara con dos opciones para la organización de los encuentros:
  - Planificación de los partidos de forma aleatoria
  - Planificación de los partidos de forma manual.
- Para la creación del calendario de futbol el sistema debe presentar la siguiente información:
  - Grupos / divisiones.
  - Nombres de los equipos.
  - Fecha y hora del encuentro.
  - Estadio.
  - Ciudad.

****Ejemplo de Israel Rey, QA test Engineer at Universidad Técnica Particular de Loja. extraído para fines académicos*

Requerimientos de Dominio

Son aquellos que se derivan del dominio de aplicación del sistema más que de las necesidades específicas de los usuarios. Se pueden considerar “un complemento” para los Requerimientos de Sistema.

Importante

- Se deben satisfacer para que el sistema funcione correctamente.
- Incluyen terminología especializada del dominio o referencia a conceptos del dominio.
- Pueden restringir los requerimientos existentes o establecer cómo se deben ejecutar dichos requerimientos.

Ejemplo *(sistema para créditos bancarios)*

Regla de Negocio

El cálculo del monto total del crédito debe ser calculado utilizando la fórmula:

Monto Total = cuota mensual x cantidad de meses = (Monto solicitado)/(cantidad de meses)x(tasa del periodo)

El cálculo de la cuota debe ser calculado utilizando la fórmula:

Cuota Mensual = (Monto solicitado)/(cantidad de meses) + Interés mensual

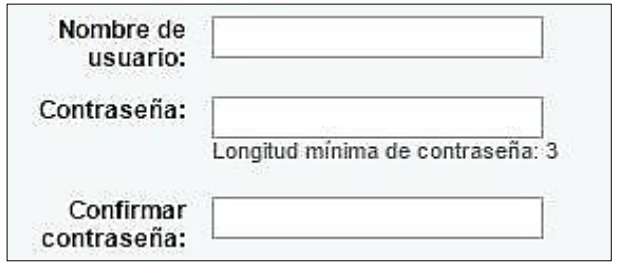
Requerimientos Funcionales

Definiciones posibles

- Se refiere a aquellos requisitos que definen una función del software o de sus componentes.
- Son aquellos que establecen los comportamientos del Software ante input determinados.

¿Qué establecen los Requerimientos Funcionales?

- Servicios o funciones que proveerá el sistema
- Interacción entre el sistema y su entorno



Un requerimiento funcional típico contiene:

- Identificador y/o Nombre
- Resumen y Descripción
- Prioridad

Ejemplo de Requerimientos Funcionales

Número	Requerimiento	Descripción	Prioridad
RF1	LA ADMINISTRACION LLEVARA UN CONTROL DE NOTAS DE LOS ALUMNOS	El sistema tendrá las notas de los alumnos en cada curso que ha llevado y que esta cursando actualmente,	5
RF2	EL ADMINISTRADOR PODRA MODIFICAR LAS NOTAS	El sistema deberá permitir la modificación de las notas del curso del ciclo vigente	5
RF3	VIZUALIZAR LA RELACION DE CURSOS Y NOTAS.	El sistema proporcionara al Profesor y al Alumno la información de notas de los cursos del ciclo vigente	4
RF4	TENER UN CONTROL DE LA INFORMACION DE LOS PAGOS	El sistema proporcionara los datos de los pagos realizados por el alumno.	4
RF5	GENERACION DE INFORMES Y REPORTES	Con los informes se podrá obtener resultados detallados sobre las notas del curso, notas por evaluación, además del promedio final, grado académico, clasificarlos por alumno, área, fecha, etc. Estos podrán ser impresos.	4
RF6	GENERACION DE USUARIOS CURSOS Y CICLOS PARA LA ADMINISTRACIOND EL SISTEMA	El sistema permitirá el ingreso de nuevos usuarios en cualquiera de las tres categorías, ciclos academicos y cursos para cada programa.	4

Ejemplo obtenido de https://es.slideshare.net/unimauro/sistema-de-gestion-de-notas

Requerimientos no Funcionales

Definiciones posibles

- Se refiere a criterios que se pueden utilizar para evaluar la operación de un sistema en lugar de sus comportamientos específicos
- Son una declaración de propiedades emergentes del sistema. Por ejemplo: Disponibilidad, Fiabilidad, Tiempos de respuesta, entre otras.

NOTA AL MARGEN

También son conocidos como “atributos de calidad” en el ámbito de la Arquitectura de Software.

Discutamos con un ejemplo

Supongamos que tenemos mucho presupuesto y vamos a

construir una casa: ¿Qué preguntas podríamos hacer a nuestro cliente/usuario?

Requerimientos no Funcionales

¿Qué establecen los Requerimientos no Funcionales?

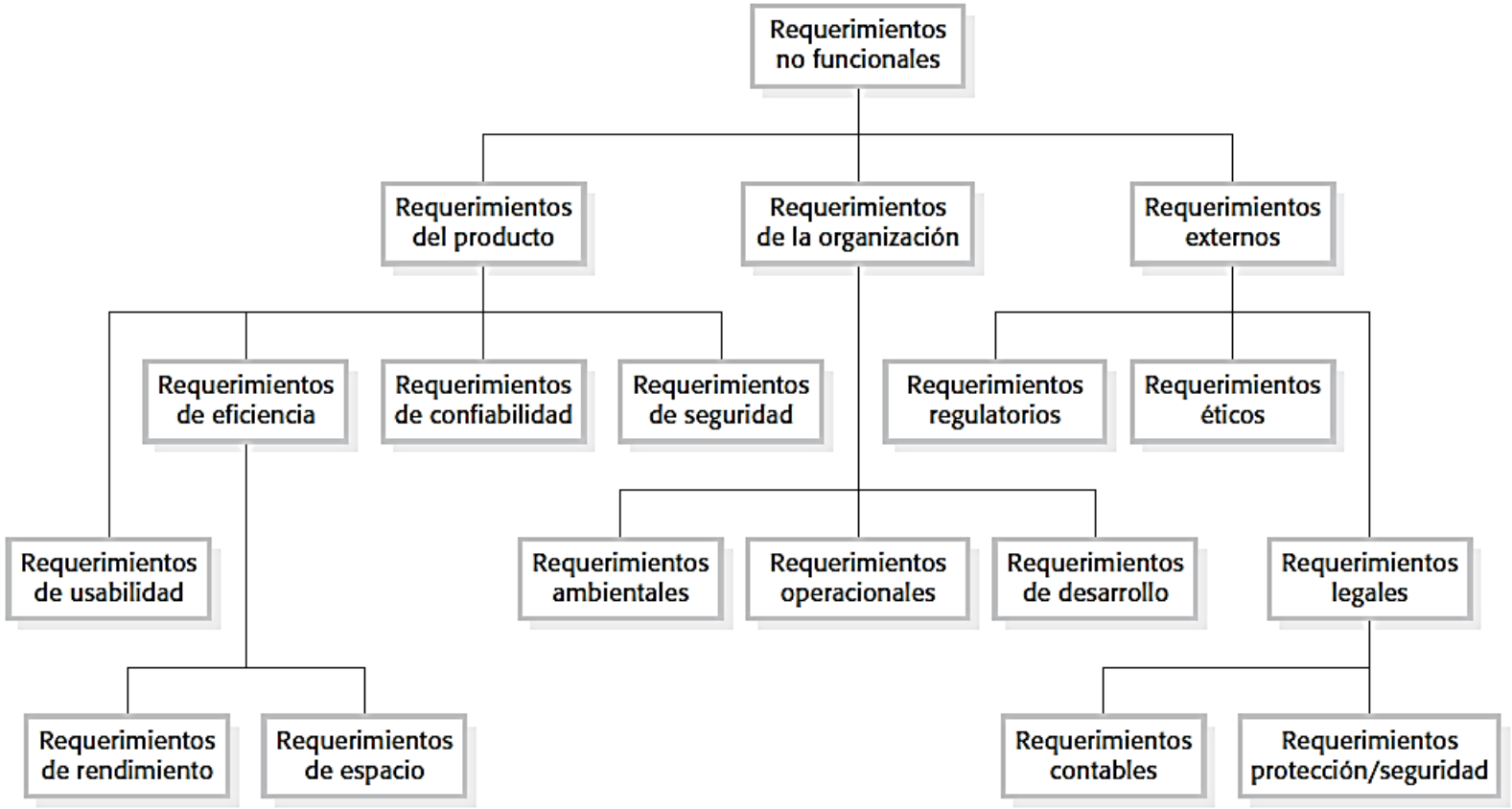
- Entornos definidos para los servicios o funciones ofrecidas por el sistema
- Restricciones para la elección en cuanto a la construcción del sistema

<u>Ejemplos simples</u>

- *La carga del formulario inicial debe tomar, en condiciones normales de conectividad, un máximo de 3 segundos.*
- *El sistema deberá estar programado en lenguaje Python 3.*

Tipos de Requerimientos no Funcionales

Según Sommerville, tenemos los siguientes tipos de RNF



Tipos de Requerimientos no Funcionales

SQuaRE: System and Software Quality Requirements and Evaluation

Es la descripción de como los modelos de calidad pueden ser usados para los requerimientos de calidad de software.

ISO/IEC 25010

La información de estas diapositivas fue adaptada de iso25000.com para fines exclusivamente académicos

ISO/IEC 25010 - Adecuación Funcional

Capacidad del producto para proporcionar funciones que satisfacen las necesidades declaradas e implícitas, cuando el producto se usa en las condiciones especificadas.

- ***Completitud funcional***

Grado en el cual el conjunto de funcionalidades cubre todas las tareas y los objetivos del usuario especificados.

- ***Corrección funcional***

Capacidad para proveer resultados correctos con el nivel de precisión requerido.

- ***Pertinencia funcional***

Capacidad para entregar un conjunto apropiado de funciones para tareas y objetivos de usuario especificados.

ISO/IEC 25010 - Eficiencia de desempeño

Representa el desempeño relativo a la cantidad de recursos utilizados bajo determinadas condiciones.

- ***Comportamiento temporal***

Los tiempos de respuesta, procesamiento y ratios de throughput de un sistema cuando lleva a cabo sus funciones bajo condiciones determinadas en relación con un banco de pruebas (benchmark) establecido.

- ***Utilización de recursos***

Las cantidades y tipos de recursos utilizados cuando el software lleva a cabo su función bajo condiciones determinadas.

- ***Capacidad***

Grado en que los límites máximos de un parámetro de un producto o sistema software cumplen con los requisitos.

ISO/IEC 25010 – 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.

- ***Coexistencia***

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

- ***Interoperabilidad***

Capacidad de dos o más sistemas o componentes para intercambiar información y utilizar la información intercambiada.

ISO/IEC 25010 – Usabilidad

Capacidad del producto software para ser entendido, aprendido, usado y resultar atractivo para el usuario, cuando se usa bajo determinadas condiciones.

- ***Capacidad para reconocer su adecuación***

Capacidad del producto que permite al usuario entender si el software es adecuado para sus necesidades.

- ***Capacidad de aprendizaje***

Capacidad del producto que permite al usuario aprender su aplicación.

- ***Capacidad para ser usado***

Capacidad del producto que permite al usuario operarlo y controlarlo con facilidad.

ISO/IEC 25010 – Usabilidad

- ***Protección contra errores de usuario***

Capacidad del sistema para proteger a los usuarios de hacer errores.

- ***Estética de la interfaz de usuario***

Capacidad de la interfaz de usuario de agradar y satisfacer la interacción con el usuario.

- ***Accesibilidad***

Capacidad del producto que permite que sea utilizado por usuarios con determinadas características y discapacidades.

ISO/IEC 25010 – Fiabilidad

Capacidad de un sistema o componente para desempeñar las funciones especificadas, cuando se usa bajo unas condiciones y periodo de tiempo determinados.

- ***Disponibilidad***

Capacidad del sistema o componente de estar operativo y accesible para su uso cuando se requiere.

- ***Capacidad de recuperación***

Capacidad del producto software para recuperar los datos directamente afectados y reestablecer el estado deseado del sistema en caso de interrupción o fallo.

- ***Tolerancia a fallos***

Capacidad del sistema o componente para operar según lo previsto en presencia de fallos hardware o software.

- ***Madurez***

Capacidad del sistema para satisfacer las necesidades de fiabilidad en condiciones normales.

ISO/IEC 25010 – Seguridad

Capacidad de protección de la información y los datos de manera que personas o sistemas no autorizados no los puedan leer y/o modificar.

- ***Confidencialidad***

Capacidad de protección contra el acceso de datos e información no autorizados, ya sea accidental o deliberadamente.

- ***Autenticidad***

Capacidad de demostrar la identidad de un sujeto o un recurso.

- ***Integridad***

Capacidad del sistema o componente para prevenir accesos o modificaciones no autorizados a datos o programas de ordenador.

ISO/IEC 25010 – Seguridad

- ***Responsabilidad***

Capacidad de rastrear de forma inequívoca las acciones de una entidad.

- ***No repudio***

Capacidad de demostrar las acciones o eventos que han tenido lugar, de manera que dichas acciones o eventos no puedan ser repudiados posteriormente.

ISO/IEC 25010 – Mantenibilidad

Esta característica representa la capacidad del producto software para ser modificado efectiva y eficientemente, debido a necesidades evolutivas, correctivas o perfectivas.

- ***Modularidad***

Capacidad de un sistema o programa de ordenador (compuesto de componentes discretos) que permite que un cambio en un componente tenga un impacto mínimo en los demás.

- ***Reusabilidad***

Capacidad de un activo que permite que sea utilizado en más de un sistema software o en la construcción de otros activos.

- ***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 para determinar si se cumplen.

ISO/IEC 25010 – Mantenibilidad

- ***Capacidad para ser modificado***

Capacidad del producto que permite que sea modificado de forma efectiva y eficiente sin introducir defectos o degradar el desempeño.

- ***Analizabilidad***

Facilidad con la que se puede evaluar el impacto de:

- ✓ Un determinado cambio sobre el resto del software,
- ✓ diagnosticar las deficiencias o causas de fallos en el software, o
- ✓ identificar las partes a modificar.

ISO/IEC 25010 – Portabilidad

Capacidad del producto o componente de ser transferido de forma efectiva y eficiente de un entorno hardware, software, operacional o de utilización a otro.

- ***Adaptabilidad***

Capacidad del producto que le permite ser adaptado de forma efectiva y eficiente a diferentes entornos determinados de hardware, software, operacionales o de uso.

- ***Capacidad para ser instalado***

Facilidad con la que el producto se puede instalar y/o desinstalar de forma exitosa en un determinado entorno.

- ***Capacidad para ser reemplazado***

Capacidad del producto para ser utilizado en lugar de otro producto software determinado con el mismo propósito y en el mismo entorno.

Ejemplo: Requerimientos no Funcionales

Número	Requerimiento	Descripción	Prioridad
RNF1	USABILIDAD	Debe ser fácil de usar. Con ayudas e interfaces intuitivas.	5
RNF2	SEGURIDAD	El ingreso al sistema estará restringido bajo contraseñas cifradas y usuarios definidos.	5
RNF3	PORTABILIDAD	El sistema debe brindar comodidad al usuario y a otras áreas que trabajan o necesitan del Área de personal. Por ejemplo El Sistema de Pago y Planillas no debe tener problemas en acceder al Sistema de Personal.	5
RNF4	MULTIPLATAFORMA	El sistema deberá funcionar en distintos tipos de sistemas operativos y plataformas de hardware.	3
RNF5	RENDIMIENTO	El sistema debe soportar el manejo de gran cantidad de información durante su proceso.	3
RNF6	DESEMPEÑO	El sistema no presentara problemas para su manejo e implementación.	1

Ejemplo obtenido de https://es.slideshare.net/unimauro/sistema-de-gestion-de-notas

Cierre de la sesión