

Gestión de Requisitos

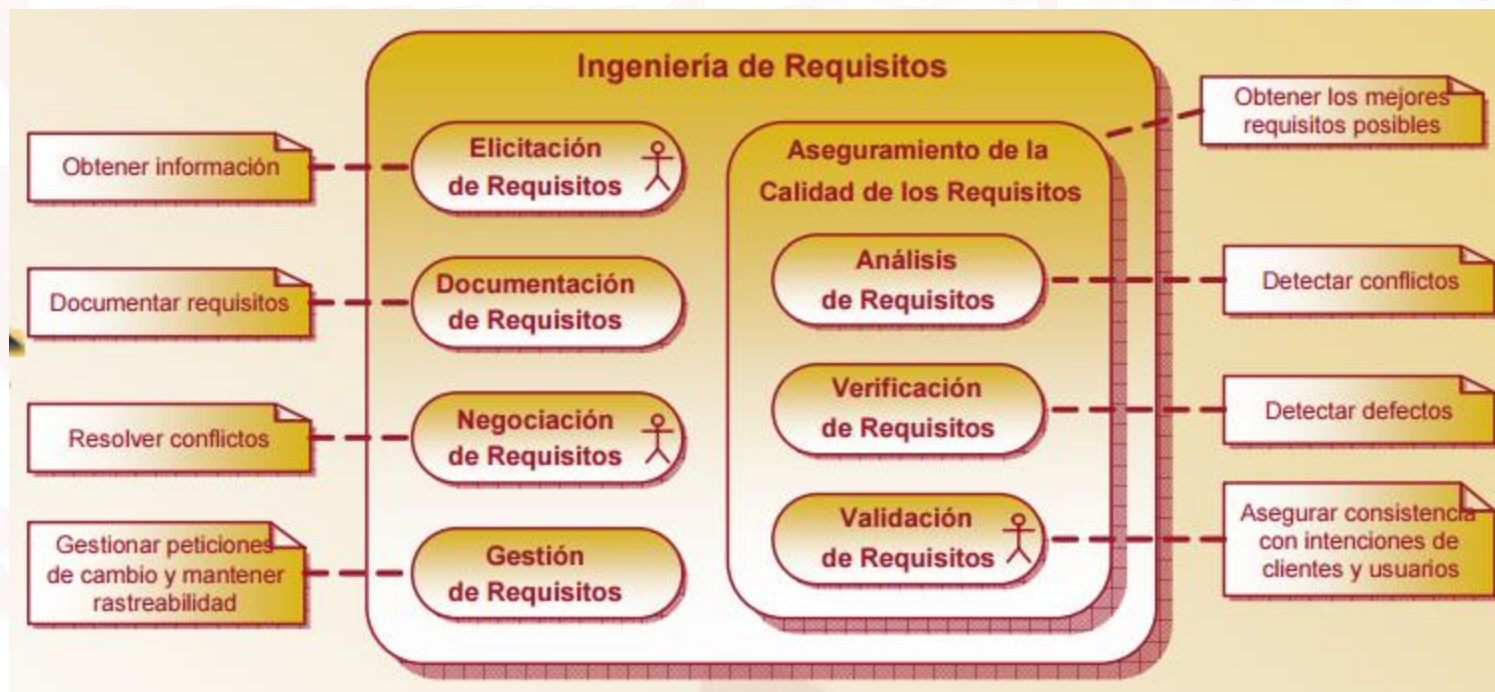
Definición y clasificación



Ingeniería de Requisitos

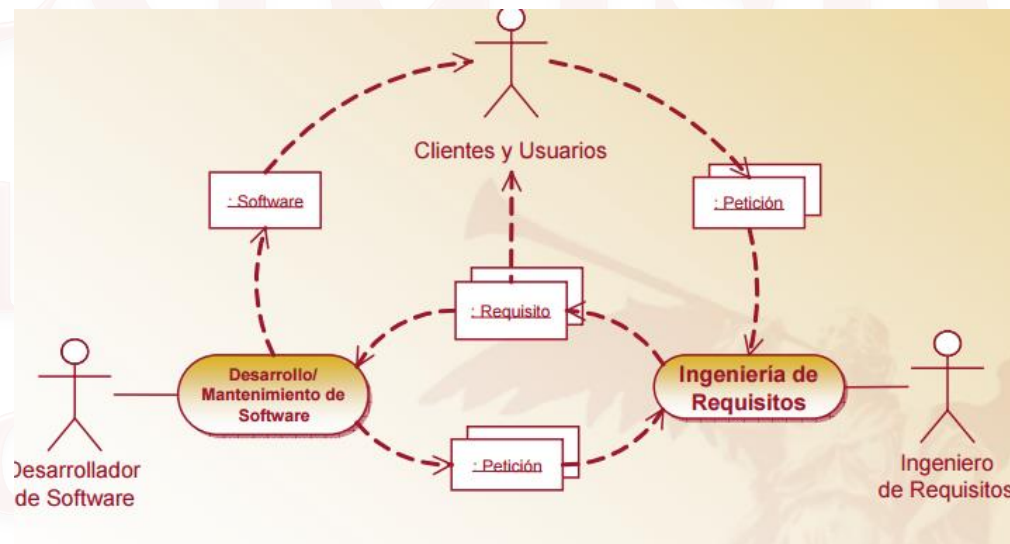
- Rama de la Ingeniería del Software que se ocupa de la 1ª etapa en el proceso de desarrollo del software.
- Las actividades de la Ingeniería de sistemas/software relacionadas con la Ingeniería de Requisitos son:
 - Identificación y documentación de necesidades de clientes y usuarios
 - Creación de un documento que describe la conducta externa y las restricciones asociadas (de un sistema) que satisfará dichas necesidades
 - Análisis y validación del documento de requisitos para asegurar la consistencia, completión y viabilidad
 - Evolución de las necesidades
- Por tanto, la IR trata de los principios, métodos, técnicas y herramientas que permiten **descubrir, documentar y mantener** los requisitos para sistemas basados en computadora, de forma **sistemática** y **repetible**.

Actividades básicas de la Ingeniería de Requisitos

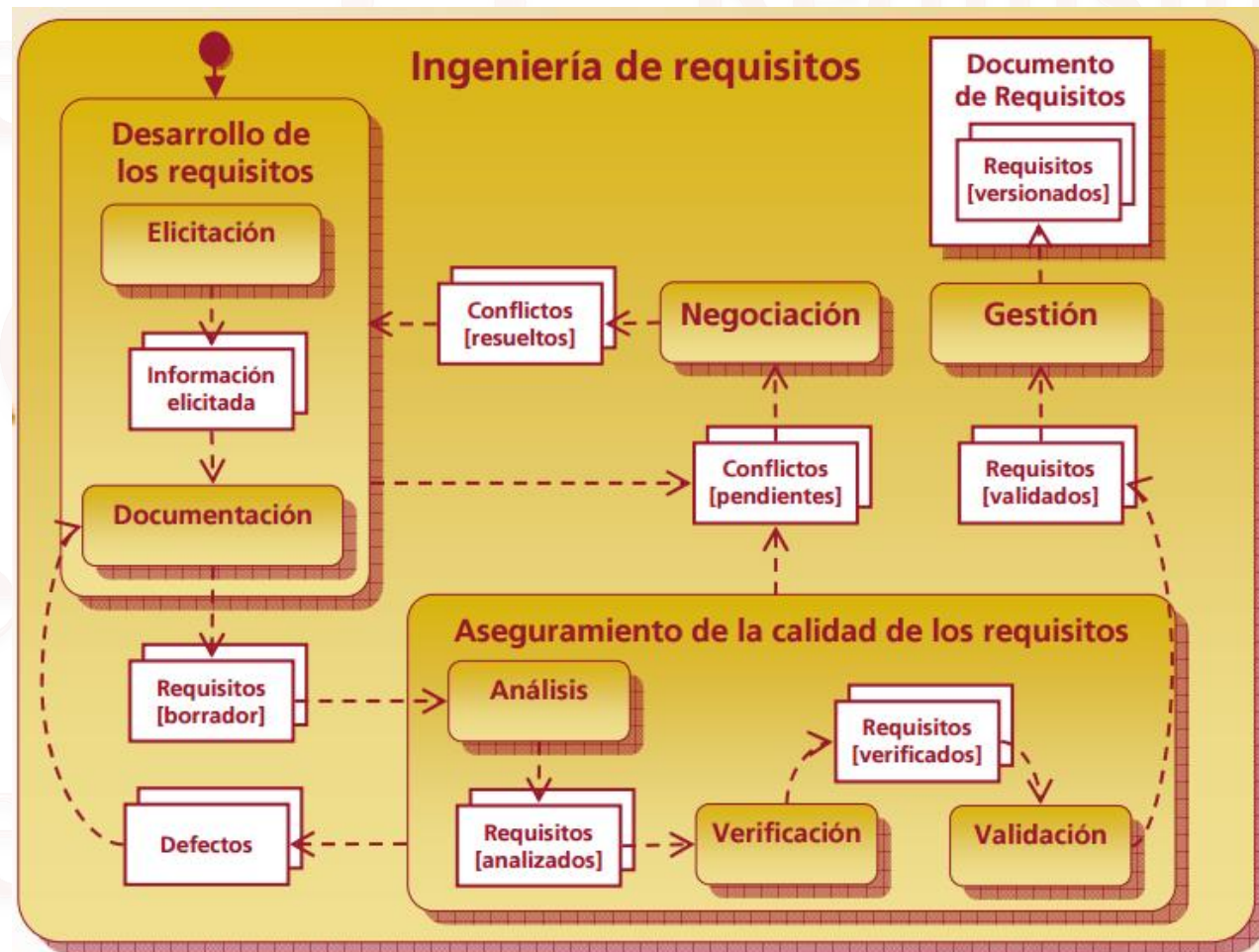


La Ingeniería de Requisitos en el ciclo de vida del sistema

- La IR comienza con el proyecto y continúa durante toda la vida del software
- El esfuerzo principal debe realizarse al comienzo del proyecto (fase de inicio del proceso unificado)
- Sus resultados marcarán el futuro del proyecto.



Modelo de los procesos de la Ingeniería de Requisitos



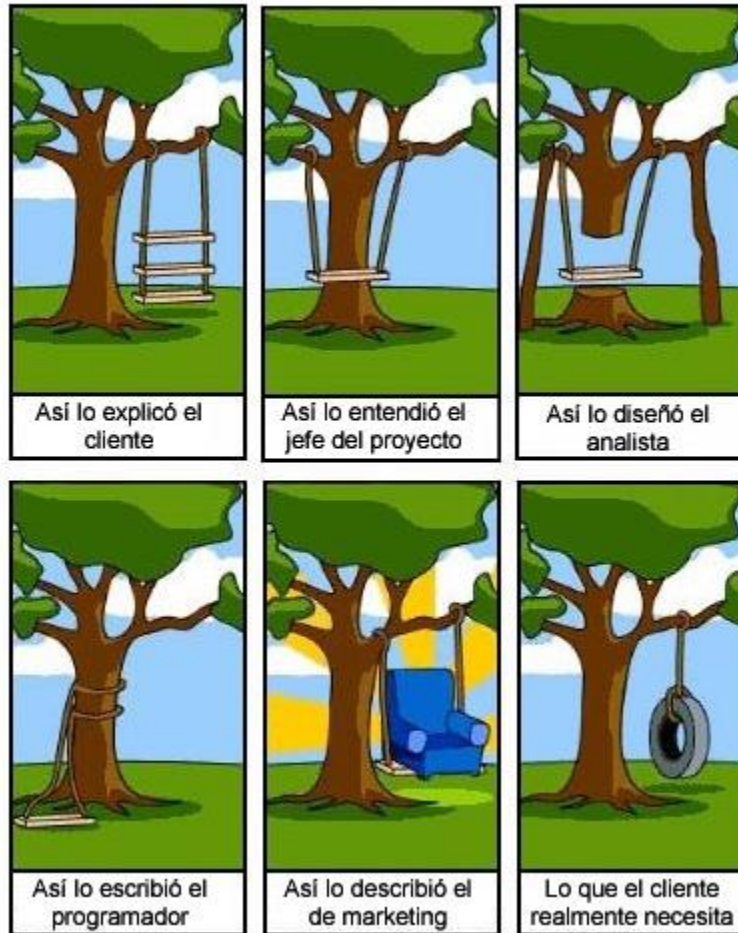
Definición de Requisito

- Es algo que el producto debe hacer o una característica que debe tener.
- Es una declaración abstracta de alto nivel de un servicio que debe proporcionar el sistema o una restricción de éste (Sommerville, 2005).
- Un requisito existe por el tipo de demanda que tiene el producto o porque el cliente quiere que el requisito sea parte del producto entregado.
- En la ingeniería clásica, los requisitos se utilizan como datos de entrada en la etapa de diseño del producto.
- Establecen **qué** debe hacer el sistema, pero no **cómo** hacerlo.

Por qué necesitamos los requisitos

- Sólo conociendo los requisitos correctos se podrá diseñar y construir el producto correcto.
- A pesar de todo esto, los requisitos no son siempre entendidos.
- El fundamento básico de cualquier software recae sobre su proceso de ingeniería de requisitos
 - La ingeniería de requisitos es la fase de la ingeniería del software donde se definen las propiedades y la estructura del software.
 - La ingeniería de requisitos debe tener un entendimiento adecuado del dominio para alcanzar los requisitos esenciales y documentarlos de forma apropiada.

Por qué necesitamos los requisitos



Fuente: <http://www.blogtrw.com>

Requisitos mal definidos generan

- Ambigüedad
 - Ejemplos: ‘Se llevará a cabo un control adecuado’
 - ‘La actualización del sistema se realizará frecuentemente’
- Definición incompleta
 - No se contemplan todas las posibilidades
- Definición con contradicciones
 - Ejemplo: se precisa un sistema que garantice el aprendizaje sin esfuerzo
- Confusión entre requisitos
 - Requisitos funcionales y no funcionales suelen mezclarse
 - Ejemplo: ‘La actualización del sistema será diaria y amigable (calidad deseable vs acción a realizar)’
- Conjunción entre requisitos
 - Definición de dos requisitos en uno
 - Ejemplo: ‘La actualización del sistema será segura y amigable con el usuario’

Calidad en la definición de requisitos

- Completos
 - todo lo que se supone que el software debe hacer está incluido
- Consistentes
 - No existen subconjuntos de requisitos contradictorios
- No ambiguos
 - todo requisito posee una sola interpretación
- Entendibles
 - Todo tipo de actores los entienden
- Factibles
 - Con los actuales recursos, es implementable
- Modificables
 - Los cambios son fáciles de introducir
- Rastreables
 - Cada requisito se puede referenciar de forma unívoca
- Verificables
 - Si cada requisito tiene un indicador que demuestra que el sistema lo satisface

Tipos de Requisitos

- Requisitos de negocio
 - Describen lo que el sistema debe hacer (al más alto nivel)
 - Proporcionan la dirección que ha de seguir el proyecto
- Requisitos de usuario
 - Declaraciones en lenguaje natural y en diversos diagramas de los servicios del sistema y de las restricciones bajo las que debe operar.
 - Describen las tareas que el sistema ha de ejecutar cuando el usuario opera con él.
- Requisitos de sistema
 - Documento estructurado que determina las descripciones detalladas de los servicios del sistema.
 - Definen las funcionalidades y características que debe tener el sistema para satisfacer tanto los requisitos de negocio como los de usuario. Serán los que servirán para realizar la arquitectura, diseño y planes de prueba del sistema.

Tipos de Requisitos

- Clasificación en función de la distinta naturaleza de los mismos.
 - Requisitos generales
 - Requisitos funcionales
 - Requisitos de Actores
 - Requisitos de Interfaz
 - Requisitos de Procesamiento
 - Requisitos de Persistencia
 - Requisitos de Gestión y administración
 - Requisitos no funcionales
 - Fiabilidad
 - Usabilidad
 - Eficiencia
 - Mantenibilidad
 - Portabilidad
 - Seguridad

Requisitos Generales

- Definición:
 - denominados también *características del sistema* (*system features*) u *objetivos del sistema*, describen a alto nivel las características básicas del sistema que luego se detallarán mediante los otros tipos de requisitos.
- Ejemplo:
 - El sistema deberá registrar los accesos al edificio mediante el uso de lectores de tarjetas magnéticas.
 - El sistema deberá respetar la siguiente regla de negocio: un socio del club deportivo podrá reservar un máximo de 5 pistas a la semana y una única pista el mismo día a la misma hora.

Tipos de Requisitos

- Requisitos funcionales
 - Servicios que debe proveer el sistema, cómo reacciona ante una entrada y cómo se comporta (función, entradas, salidas, excepciones). También pueden declarar lo que no se puede hacer.
 - Describen las acciones que el producto debe llevar a cabo. Pueden verse como requisitos independientes a cualquier tecnología. Son la esencia del trabajo
- Requisitos no funcionales
 - Son las cualidades que debe tener el producto. Estos requisitos hacen que el producto sea atractivo, útil, rápido, fiable o seguro. Estas propiedades no son requeridas porque no son actividades funcionales del producto, pero son deseables, ya que el cliente espera que las actividades sean ejecutadas de cierta manera y con un específico grado de calidad.
 - No alteran la funcionalidad esencial del producto pero pueden añadir más.
 - Propiedades emergentes del sistema y restricciones de los servicios ofrecidos por el sistema y del proceso a seguir (de implementación, de rendimiento, de usabilidad...).
- 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.

Requisitos funcionales (RF)

- En este apartado vamos a recoger los requisitos que afectan directamente a la funcionalidad principal del sistema. Normalmente esta funcionalidad describe los procesos de negocio a los que se destina el sistema
 - Requisitos de Actores
 - Requisitos de Interfaz
 - Requisitos de Procesamiento
 - Requisitos de Persistencia
 - Requisitos de Gestión y administración

RF. Requisitos de actores

- Son todos los requisitos que afectan a los diferentes actores del sistema, es decir aquellas personas o sistemas externos que pueden interactuar con el sistema, que van a proporcionarle la información de entrada y van a recibir la información de salida del sistema. Aquí se encuentran los requisitos por ejemplo de :
 - Usuarios. Son los responsables de interactuar con el sistema. Pueden ser usuarios físicos o sistemas externos.
 - Grupos. Permite realizar conjuntos de usuarios con características comunes, para simplificar las reglas de interacción entre los usuarios y el sistema.
 - Perfiles. Permite agrupar todas las características que distinguen a un usuario o grupo.
 - Papeles (roles). Permite asignar grupos de funcionalidades a los usuarios y grupos, permitiendo diferenciar el comportamiento de los usuarios con el sistema según la actividad o proceso a realizar en el mismo.

RF. Requisitos de interfaz

- En este apartado aparecen reflejados todos los requisitos que definen la forma de enviar la información a procesar por los usuarios al sistema, y la forma de recibir la respuesta del sistema por el usuario. Entre ellos podemos distinguir
 - Medios de interacción. (Aplicación de escritorio, páginas Web, Sistemas de Kiosko, etc)
 - Pantallas, formularios y demás elementos de la interfaz de usuario.
 - Mensajes intercambiados y protocolos de comunicación, fundamentales para describir las interacciones entre sistemas.
 - Movimiento por la interfaz de usuario del sistema.
 - CLAMB. Pantallas de gestión normalizadas: consultas, listados, altas, modificaciones y bajas.
 - Ayudas a los usuarios. Deberemos recoger el nivel de ayuda y formato de dicha ayuda que proporcionará el sistema a los diferentes tipos de usuarios.
 - Informes, documentos, archivos y datos en general generados por el sistema.
 - Internacionalización. Aquí hay que distinguir entre la internacionalización de la interfaz y la internacionalización de la información contenida en el sistema.
 - Accesibilidad. Define los requisitos de adaptación de un sistema para su uso por usuario con diferentes tipos y grados de discapacidad.
 - Libros de estilo y cosmética del sistema. Definen los diferentes estándares, estilos, formatos de pantallas, formatos de la información, formatos de salida de los documentos que serán tomados como guía para el desarrollo del sistema

RF. Requisitos de Procesamiento

- Son requisitos que permiten realizar las principales funciones del sistema y procesos internos.
- Indican qué hacer con los datos de entrada, cómo procesarlos y generar datos de salida.
- Hay que definir qué datos hay que tratar y mediante qué procesos se van a tratar.
- Normalmente para una aplicación de gestión se recogen los requisitos que definen la lógica de negocio.
- Por ejemplo: El sistema deberá imprimir, a petición de los usuarios, un listado de contribuyentes cuyo resultado "a devolver" del IRPF supere los 3.000 euros.

RF. Requisitos de Persistencia

- En este apartado se recogen los requisitos que afectan a la información que se debe persistir en el sistema, es decir la información que se debe guardar entre diferentes ejecuciones del sistema.
- Estos requisitos que nos permitirán construir el modelo de datos del sistema.
- Por ejemplo:
 - Para una aplicación de gestión se recogen los requisitos que van a definir las entidades, o información de negocio que se debe persistir.
 - El sistema deberá almacenar información sobre los perros de caza que componen una reala. En concreto, el código del chip de identificación, el nombre al que responde, la raza, la fecha de nacimiento y la ubicación actual (finca o similar)

RF. Requisitos de Gestión y Administración

- Son los requisitos que recogen todas las funciones que son necesarias para gestionar es el sistema, por ejemplo:
 - La gestión de usuarios,
 - La gestión de la configuración del sistema
 - Otras funciones del sistema que se apartan de la función principal del sistema.

Requisitos No Funcionales

- Son condiciones que se le imponen al sistema a desarrollar relacionadas con aspectos principalmente de calidad, algunos de los cuales influyen en la arquitectura del sistema. El modelo de calidad del software de la norma ISO-9126 organiza los requisitos no funcionales en los siguientes:
 - Fiabilidad
 - Usabilidad
 - Eficiencia
 - Mantenibilidad
 - Portabilidad
 - Seguridad
- Este conjunto de requisitos no funcionales puede ser ampliado si el proyecto de desarrollo así lo requiriera.
- Pueden ser más críticos que los funcionales
 - Si un RF no se cumple, el sistema se degrada
 - Si un RNF no se cumple, el sistema puede inutilizarse.

Requisitos No Funcionales

- **Fiabilidad:**

- Definen aspectos relacionados con la capacidad del software desarrollado para mantener su **nivel de prestación** bajo condiciones establecidas y durante un período de tiempo establecido. Las subcaracterísticas son: *madurez, recuperabilidad, tolerancia a fallos*.
- Por ejemplo: El sistema deberá tardar un máximo de 10 minutos para la recuperación de un fallo de caída total, en el 95% de las ocasiones.

- **Usabilidad:**

- Definen aspectos relacionados con las **dificultades** que pueden encontrar los usuarios al enfrentarse al uso del nuevo software. Las subcaracterísticas son: *aprendizaje, comprensión, operatividad, atraktividad*.
- Por ejemplo: El sistema deberá permitir en el 80% de las veces que con un máximo de 5 clicks sea suficiente para llegar a la información deseada.

Requisitos No Funcionales

- **Eficiencia:**

- Define aspectos que indican la proporción entre el nivel de cumplimiento del software y la cantidad de recursos necesitados bajo condiciones establecidas. Las subcaracterísticas son: *comportamiento en el tiempo, comportamiento según otros recursos.*
- Por ejemplo: El sistema deberá tener un tiempo máximo de respuesta de 5 segundos para cualquier operación de consulta.

- **Mantenibilidad:**

- define aspectos relacionados con la facilidad de ampliar la funcionalidad, modificar o corregir errores en un sistema software. Las subcaracterísticas son: *estabilidad, facilidad de análisis, facilidad de cambio, facilidad de pruebas.*
- Por ejemplo: El código fuente que se implemente en JAVA deberá cumplir las recomendaciones de Code Conventions for the Java Programming Language

Requisitos No Funcionales

- **Portabilidad:**

- Define aspectos relacionados con la capacidad de un sistema software para ser transferido desde una plataforma a otra. Las subcaracterísticas son: *capacidad de instalación, capacidad de sustitución, adaptabilidad, coexistencia, compatibilidad con hardware o software, etc.*
- Por ejemplo El sistema deberá evitar el uso de extensiones propietarias al estándar SQL-92 en el sistema de gestión de bases de datos que utilice.

- **Seguridad:**

- Define aspectos relativos a la políticas de privacidad del sistema a desarrollar. Las subcaracterísticas son: *accesos al sistema, identificación y autenticación, protección de datos y privacidad.*
- Por ejemplo: El sistema deberá ser capaz de evitar ataques de inyección de SQL sistemáticos.

Requisitos FURPS

- Desarrollado por Hewlett Packard (1987), es un modelo de calidad fijo que establece cinco características como factores de calidad para todas las actividades del proceso de desarrollo de un software, que son los que le dan nombre:
 - **FUNCIONALITY**
 - mide la **funcionalidad** del sistema: conjunto de características y capacidades que tiene el producto de software;
 - **USABILITY**
 - mide la **usabilidad** del sistema: se refiere más que todo a las operaciones que tiene que ver con los factores humanos, la estética coherencia y documentación del software
 - **RELIABILITE**
 - mide la **fiabilidad** del sistema: especifica sobre la precisión y el tiempo medio de fallos que puede presentar el software durante su desarrollo y después de haberse comercializado
 - **PERFORMANCY**
 - mide el **rendimiento** del sistema: hace mención sobre la velocidad, la eficiencia, el consumo de los recursos y el tiempo de respuesta que presenta el software en su funcionamiento e interacción con el usuario
 - **SUPPORTABILITY**
 - mide la capacidad de **mantenimiento** del sistema: implican la extensibilidad, adaptabilidad, capacidad de instalación , localización y portabilidad del producto final.

Requisitos FURPS +

- Todas estas referencias de métricas mencionadas constituyen los campos básicos para dar la calificación de “alta calidad” al producto de software, antes y después del desarrollo .
- El FURPS+ indica requisitos adicionales, tales cómo:
 - **Implementación:**
 - limitaciones de recursos, lenguajes y herramientas,....
 - **Interfaz:**
 - restricciones impuestas para la interacción con sistemas externos.
 - **Operaciones:**
 - gestión del sistema en su puesta en marcha empaquetamiento: forma de distribución.
 - **Legales:**
 - normativas, licencias, etc.

Ejemplo de Catálogo de Requisitos

- **Requisitos funcionales.** Deben estar organizados jerárquicamente y desglosados en requisitos individuales

Matriculación

- La matrícula será realizada de forma interactiva. Se le preguntará al alumno cuál es el plan de estudios en que desea matricularse (pueden ser varios).
- Se podrá generar una copia impresa de la matrícula (sin valor oficial) en el ordenador desde donde se realice el proceso de matriculación.
- Se podrá generar el impreso de pago debidamente cumplimentado.
- Para la matriculación se consultarán los datos del expediente y se realizarán las validaciones necesarias, descritas a continuación...
- Pago de matrícula:
 - La aplicación generará un impreso para que el alumno realice el pago correspondiente a la matrícula en 1 ó 2 plazos (según las fechas establecidas).
 - Si el alumno tiene matrículas de honor de cursos anteriores o disfruta de algún tipo de beca, la aplicación deberá calcular automáticamente los descuentos correspondientes...

Ejemplo de Catálogo de Requisitos

- **Requisitos No funcionales:** Deben ser requisitos claros, concretos, concisos y verificables
 - De rendimiento
 - Volumen de 500 socios
 - De frecuencia de tratamiento
 - Facturación mensual típica de 250 socios, con picos de hasta 5000
 - Los impagos suelen alcanzar el 2% del volumen total facturado al mes.
 - De seguridad
 - Control de accesos: nombre de usuario, contraseña
 - Copias de respaldo: Se realizarán 2 copias semanales

Buenas prácticas en el desarrollo de requisitos

- Documentar el alcance y visión del proyecto:
 - permitirá tener un mejor entendimiento de los requisitos y asegurará que todas las personas involucradas en el proyecto trabajen hacia la misma meta.
- Mantener un glosario del proyecto:
 - facilitará una comunicación efectiva asegurando un entendimiento unánime.
- Uso de técnicas de obtención de requisitos de usuario:
 - para facilitar esta tarea.
- Involucrar a toda la gente implicada:
 - asegura una validación temprana del entendimiento de los requisitos.
- Desarrollo incremental de requisitos:
 - puede minimizar la cantidad de re-trabajo del proyecto.
- Captura de requisitos usando casos de uso:
 - será más fácil gestionar los requisitos y hacer un seguimiento de los mismos. –
- Validar requisitos:
 - para mejorar el éxito de los proyectos es crítico que se validen los requisitos de forma adecuada. –
- Verificar requisitos:
 - para asegurar que los requisitos proporcionan una base adecuada para llevar a cabo el diseño, la construcción y las pruebas.

Buenas prácticas en la gestión de requisitos

- Priorizar requisitos:
 - para determinar aquellos que se deberían cumplir en la primera versión o producto y aquellos que pueden llevarse a cabo en sucesivas versiones.
- Establecer líneas base de los requisitos:
 - para asegurar que cualquier modificación en los requisitos que cambie la línea base se trata como cambios de alcance.
- Comunicación abierta:
 - para asegurar que la información relacionada con los requisitos se comunica de forma consistente. Una comunicación abierta también implica comunicar a la gente correcta y al conjunto mínimo de personas
- Gestión de cambios de los requisitos:
 - es esencial gestionar estos cambios de forma efectiva y eficiente.
- Uso de herramientas para la gestión de requisitos:
 - para facilitar la gestión de requisitos.
- Mantener trazabilidad de requisitos:
 - para llevar un seguimiento de la vida de un requisito.
- Establecer un plan de mejora de procesos para la ingeniería de requisitos:
 - para cumplir con las necesidades actuales y futuras de forma más eficiente y con mayor calidad.
- Formar a los analistas de requisitos:
 - para asegurar que los analistas de requisitos tienen el conocimiento, entre otros aspectos, de cómo escribir buenos requisitos, etc.

Fuentes consultadas

- Ingeniería del Software I de Miguel A. Laguna. Universidad de Valladolid
- Unidades temáticas de Ingeniería del software. Proceso de requisitos. ULPGC
- Guía avanzada de gestión de requisitos. Instituto Nacional de Tecnologías de la Comunicación (INTECO)
- Introducción a la Ingeniería de Requisitos. Universidad de Sevilla.
<http://www.lsi.us.es/docencia/get.php?id=2005>
- Especificación de requerimiento. DECSAI. Universidad de Granada.
<http://elvex.ugr.es/idbis/db/docs/design/2-requirements.pdf>
- Cristóbal González Almirón
 - Gestión de Requisitos
<http://www.adictosaltrabajo.com/tutoriales/pdfs/RM.pdf>
 - Captura de requisitos con REM
<http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=REM>