

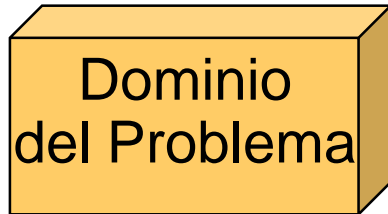


Fase de Requisitos

Índice

- ❑ 1. Motivación
- ❑ 2. ¿Qué son los requisitos?
- ❑ 3. Actividades de la Fase de Requisitos
 - ❖ 3.1. Obtención de requisitos
 - ❖ 3.2. Representación de requisitos
 - ❖ 3.3. Validación de requisitos

1. Motivación



Análisis del problema
¿Cuál es el problema?

Análisis de la solución
¿Cuál es la solución?

Diseño de la solución
¿Cuál es el mecanismo que mejor implementa la solución?



Implementación de la solución
¿Cómo está el problema resuelto?

Prueba del producto
¿Está el problema resuelto?

Transferencia
¿Puede el usuario utilizar la solución?

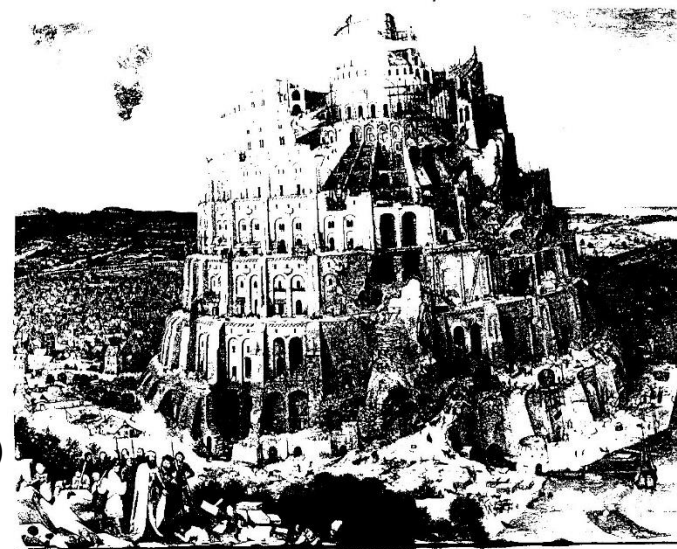
Mantenimiento
¿Son necesarias alguna mejoras?

Motivación

- Parece sencillo preguntar (al cliente, a los usuarios del software, a los expertos) qué software necesitan y cómo lo van a utilizar
- Problemas:
 - ❖ Comunicación y comprensión: los analistas no entienden totalmente el dominio del problema
 - ❖ Alcance del sistema
 - ❖ Volatilidad: las necesidades o requisitos evolucionan con el tiempo

El problema de la comunicación

- ❑ El ejemplo de la Torre de Babel: según el Génesis, fue el segundo gran proyecto de ingeniería después del Arca de Noé, pero fue el primer fiasco
- ❑ El proyecto contaba con todos los factores de éxito ...
 - ❖ Misión clara; mano de obra; materiales; tiempo suficiente ...
 - ❖ ¿Qué falló? La comunicación
- ❑ La comunicación cliente-analista es clave para el éxito en el proyecto: ¿Cómo mejorarla?
 - ❖ Experiencia



“The mythical man-month” , F.P.Brooks (1995)

El problema de la definición del alcance del sistema

- ❑ La información que nos aporta el cliente puede confundir más que clarificar los objetivos del sistema
- ❑ Definir claramente los límites del Dominio del Problema
 - ❖ ¿Qué hay dentro del sistema y qué hay fuera?
- ❑ Fase de Requisitos:
 - ❖ Definición del sistema en términos entendidos por el cliente (“Descripción del problema”)
- ❑ Fase de Análisis:
 - ❖ Especificación técnica del sistema en términos entendidos por el desarrollador (“Especificación del problema”)

2. ¿Qué son los requisitos? (I)

- Un requisito es una característica que debe incluirse en el sistema.
 - ❖ Ejemplos: una forma de capturar o procesar datos, producir información, etc.
- Distinción entre requisitos de usuario y requisitos de sistema según el punto de vista
 - ❖ Requisitos de usuario (desde el punto de vista del usuario):
 - Son las condiciones o capacidades necesarias para que el usuario pueda resolver un problema o alcanzar un objetivo
 - Se expresan normalmente en lenguaje natural y con poco nivel de detalle. Se conocen también como *(user) needs*:
 - *[user] needs to [do something] to [user goal]*
 - Ejemplo: El usuario necesita entender bajo que condiciones puede utilizar un conjunto de datos.

¿Qué son los requisitos? (II)

- ❖ Requisitos de sistema (desde el punto de vista del desarrollador):
 - Son las condiciones o capacidades que debe reunir un sistema para satisfacer un contrato, estándar o cualquier otro documento formalmente impuesto
 - Son una extensión de los requisitos de usuario expresados con mayor nivel de detalle y formalismo
 - *[entity/system] should [do something] to enable users to [do something]*
 - Ejemplo: El sistema deberá publicar el fichero con la licencia de uso del conjunto de datos en la página de descarga de los conjuntos de datos para que los usuarios puedan consultar bajo que condiciones se pueden utilizar.

❑ De ahora en adelante nos centraremos en requisitos del sistema

Tipos de requisitos de sistema

□ Funcionales

- ❖ Establecen la funcionalidad o comportamiento del sistema: ¿Qué hace el sistema?
- ❖ Describen las interacciones entre el sistema y el entorno
- ❖ Son independientes de la implementación

□ No funcionales

- ❖ Criterios para juzgar cómo opera el sistema
- ❖ Atributos de calidad
- ❖ Ejemplos: tiempo de respuesta menor a 1 sg., disponibilidad del servidor 24x7

Requisitos Funcionales

- ❑ Describen el comportamiento del software en cuanto a:
 - ❖ Cálculos
 - ❖ Funciones de consulta, actualización de datos
 - ❖ Informes proporcionados
 - ❖ Interacción con otros sistemas
- ❑ Son las acciones fundamentales que realiza el software. Los servicios que proporciona
- ❑ Describen las interacciones entre el sistema y el entorno

Requisitos No Funcionales

□ Describen las facilidades que debe proporcionar el sistema en cuanto a

- ❖ Rendimiento

- Tiempo de respuesta, nº de usuarios, etc.

- ❖ Seguridad

- Niveles de acceso, protección, claves, integridad, etc.

- ❖ Operación

- Modos de operación, copias de respaldo, etc.

- ❖ Recursos

- Limitaciones de memoria, almacenamiento, etc.

Requisitos No Funcionales

- ❖ Fiabilidad
 - Qué hacer en situaciones de error
- ❖ Documentación
 - Idiomas, ayuda on-line, tutoriales, etc.
- ❖ Mantenibilidad y portabilidad
 - Cómo cambiar o portar el software
- ❖ Interfaz y usabilidad
 - Menús, ventanas, mensajes de error, formatos de pantalla, etc.
- ❖ Comunicaciones
- ❖ Etc.

¿Qué no son los requisitos?

- ❑ Todo aquello que no indica qué hace el software o cómo opera.
- ❑ Todo aquello que pertenece al Dominio de la Solución, es decir, a cómo se soluciona el problema:
 - ❖ Tecnología de implementación. Entorno de desarrollo
 - ❖ Metodología de desarrollo
 - ❖ Arquitectura del software/sistema
- ❑ **Se conocen como RESTRICCIONES y NO deberían ser establecidas por el cliente**

3. Actividades de la Fase de Requisitos

- ☐ Obtención de requisitos
- ☐ Representación de requisitos
- ☐ Validación de requisitos

3.1. Obtención de requisitos

□ ¿En qué consiste?

- ❖ Recopilar información y hechos para representar los requisitos
- ❖ Identificar usuarios potenciales
- ❖ Identificar fuentes relevantes de conocimiento e información
- ❖ Preparar y preguntar cuestiones concisas y directas

□ Técnicas

- ❖ Entrevistas, Reuniones, Cuestionarios

□ Fuentes

- ❖ Usuarios del sistema
- ❖ Formularios y documentos
- ❖ Manual de procedimiento
- ❖ Informes
- ❖ Programas

Entrevistas

❏ Objetivo

- ❖ Maximizar la cantidad de información relevante a obtener del entrevistado (cliente, usuario, experto, etc.)
- ❖ Esa información será la fuente para confeccionar los requisitos

❏ Consejos

- ❖ Preparación: definir objetivo y preparar preguntas directas
- ❖ Elegir a la persona clave
- ❖ Poner en conocimiento del entrevistado la información que deseamos obtener
- ❖ Realizar la entrevista de manera educada y amistosa
- ❖ No utilizar jerga informática para impresionar al entrevistado
- ❖ Explicar las limitaciones del ordenador en términos cotidianos y describir al usuario cómo le puede ayudar en su trabajo
- ❖ Preguntar al entrevistado por nuevas fuentes de información

Reuniones (I)

- ❑ Continuación de las entrevistas. Varios entrevistadores y varios entrevistados
- ❑ Se puede aplicar en diferentes fases del proceso de desarrollo: Análisis, Diseño, Seguimiento del proyecto, ...
- ❑ Agenda formal para cubrir los puntos importantes
- ❑ Agenda informal para estimular el *brain-storming* (tormenta de ideas)

Reuniones (II) - Decálogo

1. Pensar si la reunión es necesaria y quién es imprescindible que asista.
2. Enviar convocatoria, lo más documentada posible, fijando la hora de comienzo y de terminación.
3. Todos los participantes prepararán adecuadamente la reunión antes de su comienzo.
4. Ser puntuales. Comenzar a la hora prevista.
5. El convocante, o quien él designe, actuará como moderador.
6. Se evitarán todo tipo de interrupciones. Dejarlo advertido previamente.
7. No tratar otros temas que los propios de la reunión.
8. Respetar las intervenciones de los demás. Practicar la escucha activa. Ser concretos.
9. Terminar a la hora indicada en la convocatoria.
10. Hacer un acta de la reunión, a no ser que se indique lo contrario.

Cuestionarios

- ☐ Un cuestionario contiene todas las preguntas que el usuario debe responder para proporcionar la información que busca el analista
- ☐ Son útiles cuando se busca la misma información en múltiples usuarios
- ☐ Un cuestionario es fácil enviarlo a todos los usuarios de la organización
- ☐ Son útiles para recoger datos numéricos u obtener opiniones relativamente simples de un número de personas
- ☐ No son útiles para obtener información detallada

3.2. Representación de requisitos

- ❑ La información recogida es útil para elaborar un catálogo de requisitos que el software deberá satisfacer
- ❑ Deben facilitar la construcción de los productos de la Fase de Análisis
- ❑ Técnicas para representar requisitos
 - ❖ Tablas de requisitos
 - Si es necesario, se incorporan también arboles y tablas de decisión
 - ❖ Prototipado ([vistos en asignatura IPO](#))
 - ❖ Casos de uso ([ver siguiente tema](#))
 - Incorporan descripciones textuales y ocasionalmente diagramas de modelado dinámico (principalmente diagramas de secuencia)

Tablas de requisitos (I)

- ❑ Es una técnica muy sencilla donde los requisitos se representan en una tabla
- ❑ Un requisito se representa con un código y un texto que lo describe
- ❑ La estructura del código es fijada por la organización

Código	Descripción

Ejemplo de tabla de requisitos

Administrador:

Código	Descripción
RF-1	La aplicación permitirá dar de alta, baja y modificar los datos de los alumnos
RF-2	...

Secretaría:

Código	Descripción
RF-1	La aplicación permitirá generar listados de los horarios del profesorado y los horarios de los grupos
RF-2	...

Código	Descripción
RNF-1	La aplicación utiliza la imagen institucional de la Consejería de Educación. La interfaz de usuario usa los colores institucionales, logos y plantillas definidas para ello

Tabla de requisitos (II)

- ❑ En proyectos donde se aplican metodologías ágiles (varias iteraciones guiadas por historias de usuarios) suele ser habitual añadir un atributo adicional a la tabla de requisitos indicando la prioridad de implementación
- ❑ Codificación MoSCoW para indicar la prioridad:
 - ❖ M – MUST: Requisitos totalmente imprescindibles que tienen que estar incluidos ya que si no se llevan a cabo el proyecto no puede salir adelante
 - ❖ S – SHOULD: Requisitos que deberían de llevarse a cabo si es posible, es decir, requisitos importantes y de gran valor para el producto que se está construyendo
 - ❖ C – COULD: Requisitos que podrían incluirse si no afecta a nada más, es decir, requisitos que sería bueno tener y podrían incluirse porque no cuesta demasiado implementarlos
 - ❖ W – WON'T: Requisitos que no se implementarán en la iteración actual, pero que serán tenidos en cuenta en el futuro

Ejercicio: Construir tabla de requisitos para la aplicación "Gestión de aula informática"

- ❑ Un aula informática de una escuela universitaria tiene la siguiente política de reservas:
 - ❖ Los alumnos pueden reservar un máximo de 4 horas a la semana.
 - ❖ Las reservas pueden realizarse hasta el día anterior al uso.
 - ❖ Cuando un alumno hace una reserva A, se le entrega un justificante de la misma. El formato del justificante es el siguiente.

Reserva Aula Informática

Equipo: 014

Día reservado: 28-06-93

Hora reservada: 16:00

Número de Expediente: 54.321

Nombre: Alicia Kensington Carroll

Ejercicio: Gestión de aula informática (II)

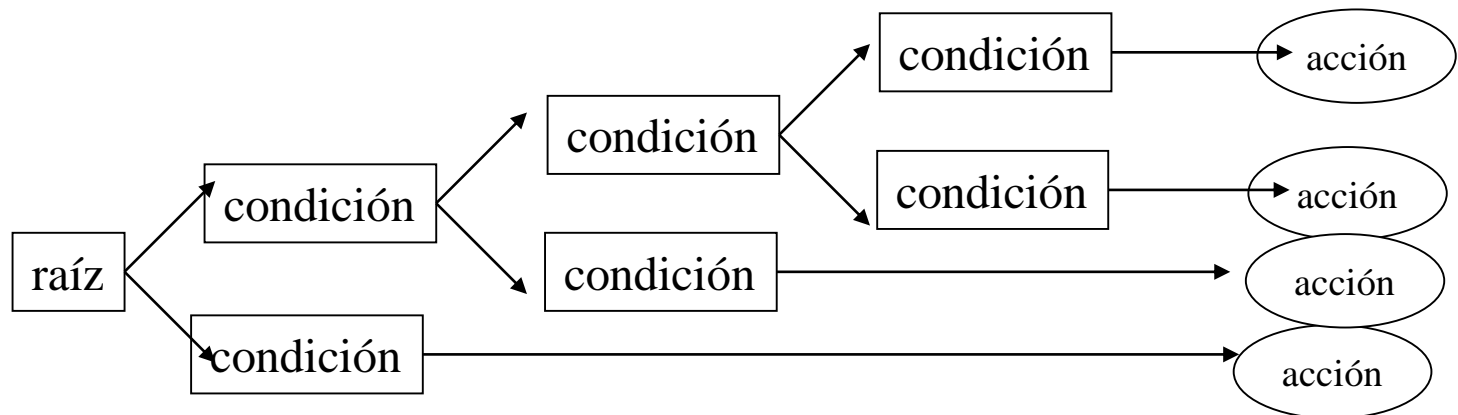
- Cuando se abre el Aula hay un listado de las reservas realizadas sobre cada equipo, de modo que la gente puede usar los equipos no reservados. El formato del listado de reservas es el siguiente:

Reservas del aula informática		Día: 28-06-93
Equipo	Hora	Alumno
001	10:00	Peter Pan
014	10:00	María Montesa
014	16:00	Alicia Pérez
...

- El horario del aula es de 10:00 a 18:00 todos los días lectivos y las reservas comienzan siempre a una hora en punto

Arboles y tablas de decisión (I)

- ❑ Si un requisito describe una toma de decisión, se puede especificar mediante un árbol o una tabla de decisión para evitar malas interpretaciones
- ❑ Árbol de decisión
 - ❖ En la raíz del árbol comienza la secuencia de decisión, la rama a seguir depende de las condiciones. La parte final es la acción

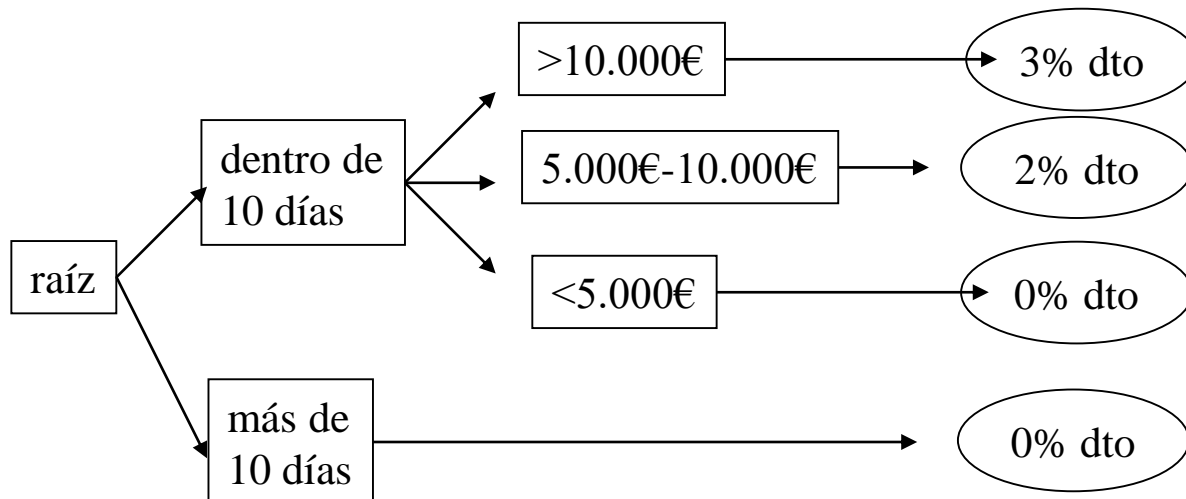


Arboles y tablas de decisión (II)

❖ Ejemplo de árbol de decisión

➤ Requisito: "Aplicación de descuento por pronto pago:

- Si el cliente paga dentro de los 10 primeros días y el importe es superior a 10.000€, se le aplica un 3% de dto.
- Si el cliente paga dentro de los 10 primeros días y el importe es entre 5.000€ y 10.000€, se le aplica un 2% de dto.
- En el resto de los casos no se autorizan dtos."



Arboles y tablas de decisión (III)

□ Tabla de decisión

- ❖ Se dividen en dos partes (condiciones, acciones) y está formada por 4 secciones

Identificación de condiciones	Entrada de condiciones (cada columna representa una combinación de condiciones)
Identificación de acciones	Entrada de acciones (cada cruz representa una regla de decisión)

Arboles y tablas de decisión (IV)

□ Ejemplo de tabla de decisión

- ❖ Requisito: "La atención sanitaria en un hospital es de carácter obligatorio
 - Si el paciente dispone de seguridad social, su asistencia estará exenta de pago, si no es así pero dispone de un seguro médico sólo hará frente al pago de la consulta.
 - Sólo en el caso de no disponer el paciente ni de seguridad social, ni de seguro médico pagará todos los servicios."

C1: El paciente tiene seguro médico	S	S	N	N
C2: El paciente tiene seguridad social	S	N	S	N
A1: Pagar la consulta		X		
A2: Exento de pago	X		X	
A3: Pagar los servicios				X

Ejercicios: representar mediante árboles o tablas de decisión

□ Ejercicio 1:

- ❖ "A un cliente se le dará trato preferencial si cumple una de estas tres condiciones:
 - compra más de 10.000 € por año y tiene una buena historia de pagos
 - ; compra más de 10.000 € por año y ha comerciado con nosotros por más de 20 años
 - ; compran 10.000 € o menos por año, pero tiene una buena historia de pagos."

□ Ejercicio 2:

- ❖ "Si la cuenta del cliente se factura usando un método de tarificación fijo,
 - se establece una carga mensual mínima para consumos menores de 100 Kwh.
 - En los demás casos, la facturación aplica la tarifa A.
- ❖ Sin embargo, si la cuenta se factura usando un método de facturación variable,
 - se aplicará la tarifa A a los consumos menores de 100Kwh.
 - En otro caso, se factura de acuerdo a la tarifa B."

3.3. Validación de requisitos (I)

□ Objetivo

- ❖ Demostrar que los requisitos definen el sistema que quiere el usuario
- ❖ El coste de un cambio debido a requisitos erróneos puede ser 100 veces mayor que el coste de un error de programación → Un cambio implica a todas las Fases → Los errores se propagan a todas las Fases

Validación de requisitos (II)

□ La revisión de requisitos es un *proceso manual*

- ❖ Se involucran clientes, usuarios, analistas, ...
- ❖ Los analistas explican al cliente las implicaciones de cada requisito

□ Chequear:

- ❖ Precisión, no ambigüedad y claridad
 - El requisito no resulta vago, tiene una interpretación única y es fácil de leer
- ❖ Validez (corrección)
 - El requisito identifica una característica adicional o diferente solicitada por el usuario
- ❖ Consistencia
 - Los requisitos no se contradicen

Validación de requisitos (III)

- ❖ Completitud

- Los requisitos cubren todas las necesidades

- ❖ Realismo

- El requisito es realizable

- ❖ Verificable

- Será posible verificar que se ha satisfecho una vez finalizado el sistema

- ❖ Trazable

- Se puede seguir la pista del requisito a lo largo del desarrollo

- ❖ Manejable (adaptable)

- Se puede cambiar sin afectar excesivamente al resto de requisitos

- ❖ Libre de detalles indeseables de diseño

- Los requisitos deben ser un modelo de conocimiento, no de diseño ni de implementación

Ejercicio de validación de requisitos (I)

- ❑ El Centro de Vacaciones del Alto Pirineo, dependiente de la DGA, tiene un conjunto de plazas reservadas en una serie de zonas de acampada para disfrute exclusivo de los niños descendientes de habitantes del alto Pirineo Aragonés
- ❑ Se desea construir un sistema de reservas de plazas que pueda operarse en un PC con la siguiente política de reservas:
 1. Hay tres zonas de acampada con las siguientes capacidades:
 - Campo principal: 50 campistas máximo
 - Campo secundario: 30 campistas máximo
 - Campo terciario: 30 campistas máximo

Ejercicio de validación de requisitos (II)

2. Cada zona de acampada esta dividida en el siguiente número de áreas:
 - Campo principal: 5 áreas
 - Campo secundario: 3 áreas
 - Campo terciario: 3 áreas
3. Algunas áreas pueden ser cerradas al público cuando se necesitan para eventos especiales de Agencia.
4. Máximo 10 personas por reserva.
5. Las reservas se realizan con no más de un mes de adelanto. Un periodo de espera de un mes debe transcurrir entre reservas, comenzando a contar desde el día que la llave de acceso es devuelta.
6. Una reserva consiste en una lista de nombres de adultos y niños. Por lo menos el 30% deben ser niños.
7. Las reservas están limitadas a:
 - 7 días consecutivos durante el periodo escolar
 - 4 días / 3 noches durante vacaciones escolares

Ejercicio de validación de requisitos (III)

8. Las reservas pueden ser canceladas llamando con al menos dos días de adelanto. Una *no presentación* es un aviso. El segundo aviso implica una suspensión de tres meses de derecho de reserva. El tercer aviso son seis meses de suspensión.
9. Se solicita un depósito de 9 € a la entrega de la llave. Este depósito es devuelto cuando se devuelve la llave. Las llaves deben ser recogidas en día laborable y no antes de una semana antes de la reserva. Las llaves deben ser devueltas el primer día laborable después de la estancia. El retraso en la devolución de la llave implica la pérdida del depósito. La pérdida de una llave implica tres meses de suspensión y pérdida del depósito.

□ Ejercicio: Chequear posibles problemas en la especificación de requisitos anterior