

ÍNDICE

Criterios de requisitos - Nivel de detalle	7
Desarrollo de requisitos - Etapas	4
Errores	2
Etapas de un proceso	3
Fuente de información (Educación y extracción)	10
Ingeniería de requisitos - Definición	6
Introducción	2
Modelo de dominio (Clair Larman)	16
Proceso de adquisición de conocimientos	9
Elicitación	9
Relevar	10
Proceso de Ingeniería de Requerimientos (Ian Sommerville)	5
Proceso de Ingeniería de Requerimientos (Karl Wiegers)	4
Requisitos (funcionales y no funcionales)	6
Requisitos no funcionales - Extensión	
Requisitos no funcionales - Sommerville	7
Roles (de desarrollo y de cliente)	3
Técnicas de Educación	
Análisis de protocolos	15
Brainstorming	13
Card sorting	15
Concepto Mapping	14
Cuestionario	11
Desarrollo conjunto de aplicaciones	13
Entrevista	11
Observación	15
Storyboard	14
Caso de Estudio	20

RESUMEN INTRODUCCIÓN A LA GESTIÓN DE REQUISITOS

Trato con el usuario: función del analista de sistemas (área de desarrollo), para resolver algún problema o iniciar algún proyecto e informatizarlo.

Trabajo del experto: entender/comprender lo que necesita el usuario (no necesariamente es igual a lo que quiere); analizar la viabilidad del proyecto.

Desarrollo de un producto software: Realizado por un equipo, difícil coordinación

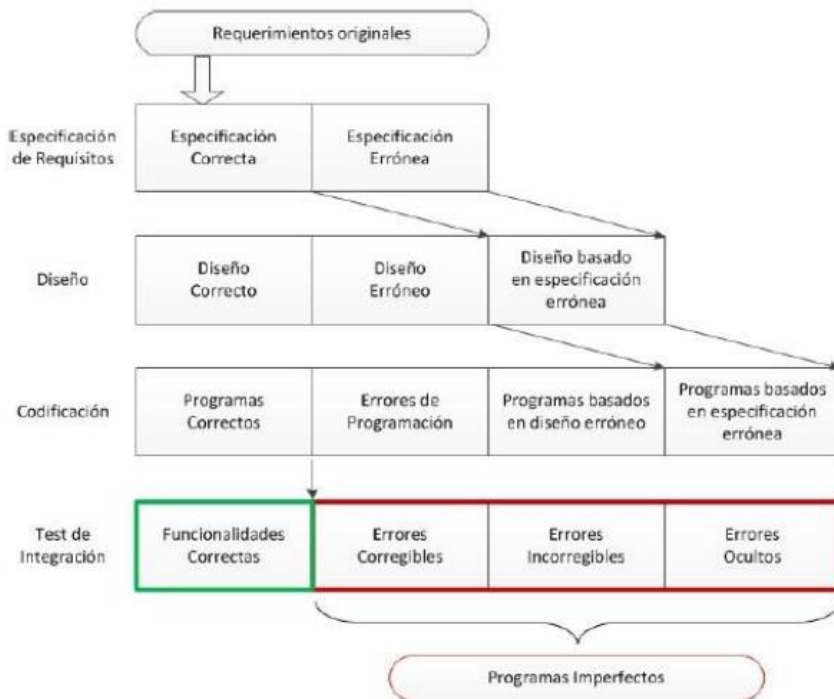
Experto → encargado de pensar la manera de que sea exitoso.

Etapas (ampliadas) de un proceso: (Analogía con arquitectura)

1. Se identifica lo que el cliente necesita.
2. Se recolecta información: ver contexto en que se desarrollará.
3. Se identifica que se tiene que hacer: boceto de lo que se entendió que el cliente precisaba, para corroborarlo con él.
4. Con alto grado de certeza → ver que solución se le propondrá al cliente: boceto.
5. El cliente termina de decidir ciertos aspectos: se negocia.
6. Se contrata un equipo de trabajo para empezar a realizarlo.

Importancia de boceto: ver si surgen otros problemas, garantizar si la idea es efectivamente la que se realizó, validar si están de acuerdo.

Mucho más económico → frenar en las primeras etapas.



Etapas de un proceso de desarrollo de SW:

1. Etapas de Adquisición y Análisis: Recopilamos y analizamos información. Luego se identifican los **requisitos**. (product owner, analista de sistemas).
2. Etapas de Diseño: Se busca una propuesta y/o una solución para aquello que se está necesitando. (lo que se identificó en la etapa 1). (diseñador, proyectista)
3. Etapas de Implementación: Se lleva a cabo la solución encontrada en etapa 2 (programador)

8 roles:

Stakeholder: todas las personas interesadas e involucradas en el proyecto.

Analistas Diseñadores Desarrolladores Gestores (managers)	Equipo de desarrollo
Cliente Usuario Experto Sponsor	Equipo de cliente

Cliente: el que lo pide y lo paga; el ente de la organización que solicita el desarrollo del proyecto en específico.

Usuario: el que usará el sistema.

El cliente no necesariamente es el usuario. Ej.: intraconsulta.

Experto: el que puede (o no) ser el que paga -cliente-, el que va a usar (o no) el sistema - usuario- pero tiene una característica en particular: es el que sabe, que tiene talento y/o es especialista en eso. Ej.: sistema para el buffet de la facu; el experto es el que cocina, etc.

Sponsor (facilitador): el que va a organizar e iniciar todos los mecanismos necesarios para que el desarrollo del sistema se agilice.

CRC: Conocer, Representar, Comunicar

Conocer	Entender qué es lo que <u>no</u> está transmitiendo el usuario, que precisa.
---------	--

Control de calidad \neq **Aseguramiento de calidad** → más barata

Una vez que tengo el producto terminado

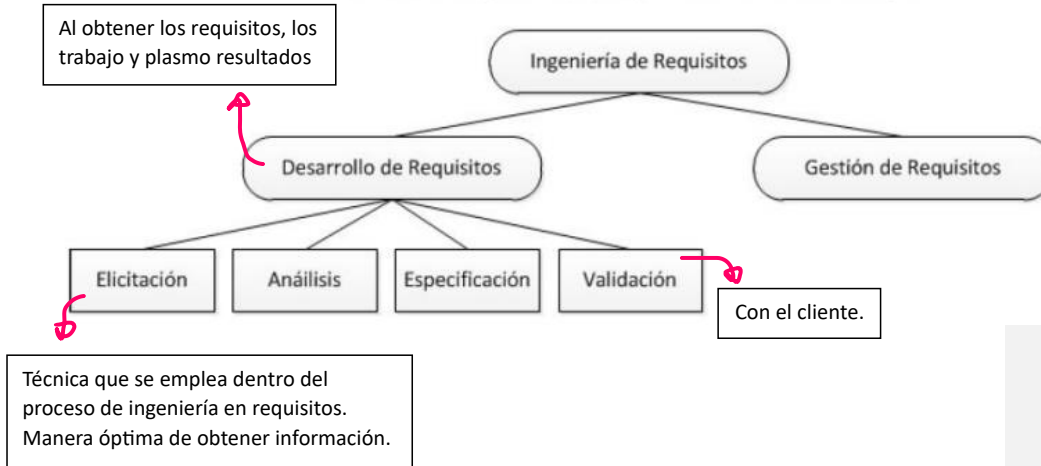
Mientras se está produciendo

Relevamiento: Situación pasiva que escucha lo que precisa el cliente.

Elicitación:

Más sencillo. Habla de funciones o procesos

Proceso de Ingeniería de Requerimientos – Karl E. Wieggers



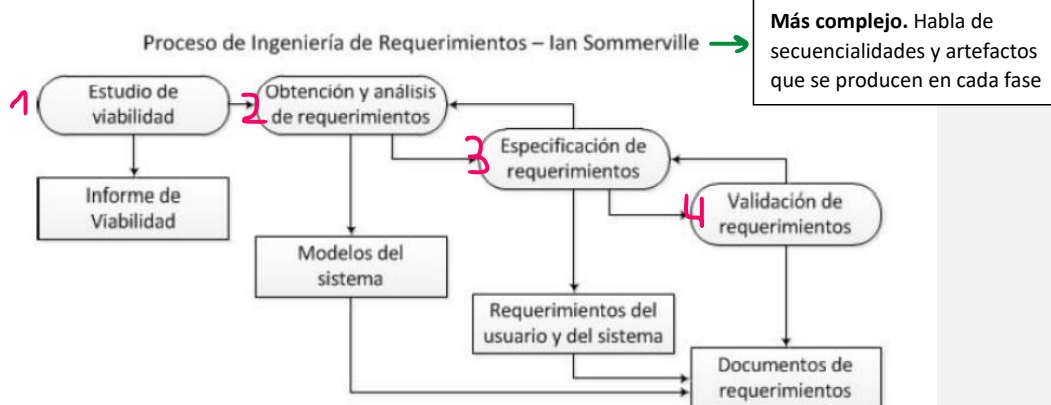
Requisitos del sistema: Conjunto de funcionalidades que el sistema debe cumplir. Tiene algún tipo de formato en particular. Lo expresamos de diferentes formas. → salida

Requisitos del cliente: lo que el cliente precisa → entrada.

Proceso Genérico de la Ingeniería de Requisitos			
1.Elicitación	2.Modelización	3.Análisis	4.Gestión
Identificación de fuentes de Información	Representación	Verificación	Identificación de cambios
Recolección de hechos	Organización	Validación	Análisis de cambios
Comunicación	Almacenamiento (registración)	Negociación	Realización de cambios

Características del proyecto: costo, tiempo y alcance

Historia del usuario: forma que tiene el cliente de expresar sus requisitos.



1. **Estudio de viabilidad:** Ver si el problema que debo resolver lo puedo hacer. Acá si o si se trabaja el requisito esencial del sistema: NO son todos los requisitos, sino el más importante. Después podés darte cuenta de que no era tan viable como pensabas.
 - **Informe de viabilidad:** primer documento que se hace y se entrega al cliente y se dice si es posible o no.
2. **Obtención y análisis y de los requerimientos:** del cliente. Juntamos la información de lo que el cliente quiere y la analizamos. Producimos **un modelo** (del sistema – los más importantes – y del usuario) como representación para explicar lo que entendimos que quiere el cliente.
3. **Especificación de requerimientos:** lo que entendimos el 2, lo volcamos a algún tipo de documento. El ingeniero d requisitos junta la info., la analiza y arma la especificación, y se puede dar cuenta de que falta más, y se tiene que volver a analizar (feedback); se genera entonces una nueva versión de la especificación de los requerimientos. Puede ir y volver las veces que se precise.
4. **Validación de requerimientos:** con el cliente. Puede haber feedback si tuve que hacer de vuelta el punto 3.

Documentos de requerimientos: lo conforma todo lo que se fue produciendo en las etapas.

ERS: Especificación de los Requisitos del Sistema

Requisitos de un sistema: dos tipos

- ➔ **FUNCIONALES:** lo que el sistema debe permitir hacer.
 - Alto nivel: poco detalle (*)
 - Bajo nivel: mucho detalle (*)
- ➔ **NO FUNCIONALES:** aquello que sobre lo funcional ya definido actúa como condicionante. Características, comportamiento.

Ejemplo: SUBE

“el sistema BICICLETA deberá permitirme transportarme de un lugar a otro”

➔ **alto nivel**

“..... a través del esfuerzo del propio pasajero, es un medio de transporte de una o dos plazas, y con la necesidad de tener equilibrio”, ➔ **bajo nivel**

FUNCIONALES

Ej.: “que al apoyar la sube no tarde +2 seg, que al fijarse el saldo en el sistema, este sea responsive”

NO
FUNCIONALES

Funcionalidades: cosas que el sistema provee o hace.

Requisito: describe funcionalidad que sistema cumple o debe cumplir.

Un requisito no funcional caracteriza y/o condiciona al funcional y al sistema

Ingeniería de requisitos: conjunto de técnicas para reunir información y así procesarla, analizarla y destilar los requisitos.

Según PRESSMAN: “es un conjunto de procesos, tareas y técnicas que permiten la definición y gestión de los requisitos de un producto de un modo sistemático”

Ingeniero de requisitos: escucha al cliente en todas sus manifestaciones y produce a través de distintas técnicas ese conjunto de funcionalidades que el sistema debe tener

- **Entrada:** del cliente ➔ manifestaciones de lo que desea ➔ aplicamos técnicas, herramientas, estrategias para producir salida
- **Salida:** requisitos del sistema ➔ lo que va a hacer el sistema, orientado a lo que el cliente realmente necesita ➔ a través de sistema informático

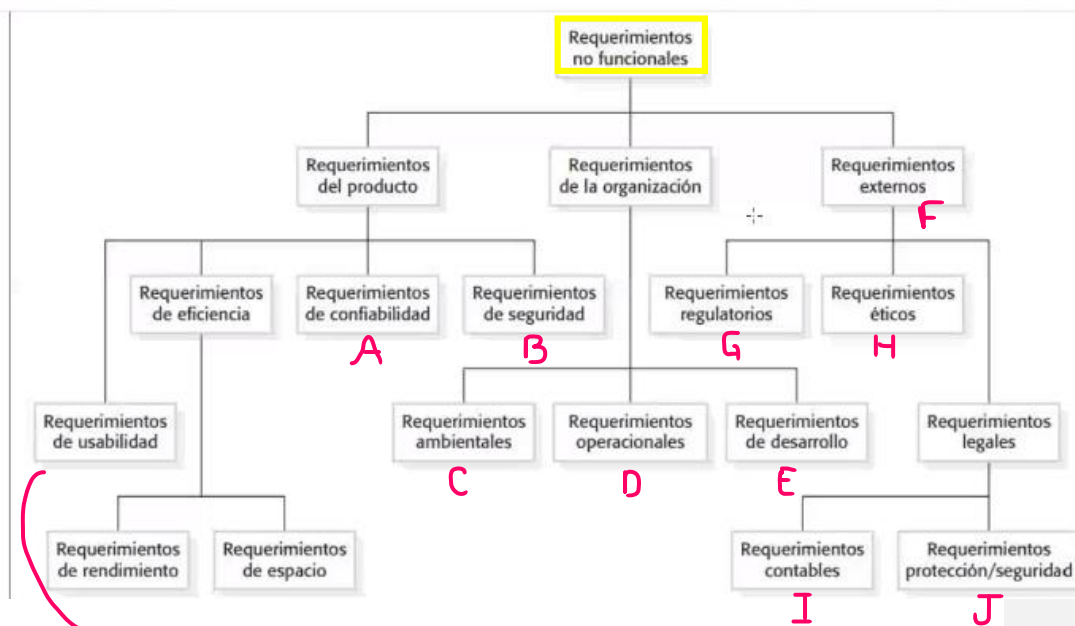
Criterios de requisitos:

1. **Requisitos del cliente** (en alto nivel por lo general) **y del sistema**
2. **Nivel de detalle: (*)**

Mucho detalle: requisito de **bajo nivel** → expresión en términos muy **completos, detallados, específicos de un requisito**. Extensión **mediana o grande**. Aborda condicionante o determinante en profundidad. Aborda tema de las alternativas y escenarios posibles. Sesgo u organización. Busca evitar cualquier ambigüedad o interpretación múltiple → un montón de info.

Poco detalle: requisito de **alto nivel** → expresión en **términos generales**; permite de forma rápida y sencilla introducirnos a la funcionalidad o restricción que expresa ese requisito → solo **da una idea**

Estrategia para identificar requisitos no funcionales a través de una caracterización de Sommerville:



Usabilidad: fácil de usar.

Según Sommerville: “A veces son más importantes los **no funcionales** que los **funcionales**, ya que muchas veces cuando un sistema no provee ni satisface una funcionalidad, a veces el usuario encuentra otro camino para satisfacerla. Sin embargo, otras veces hay **no funcionales** que **si no se cumplen**, por más que este la funcionalidad, el sistema es inútil.

Ejemplo: SUBE → si funciona pero no es rápido, no sirve.

- A. **Requerimientos de confiabilidad:** “tasa de fallos menor al 1% sobre una media de 5000 operaciones” “tiempo entre fallos no menor a los 3 meses de funcionamiento continuo”
- B. **Requerimientos de seguridad:** “El sistema SUBE debe ser inmune a la copia de sus datos a través de un dispositivo estándar de lectura en el mercado” “Debe garantizar que los datos personales del titular de la cuenta no sean accesibles o vulnerables para la empresa de transporte”
- C. **Requerimientos ambientales:** No por el medio ambiente, sino por el lugar en donde se va a utilizar (interacción con el entorno): “el sistema SUBE debe ser móvil y resistente a vibraciones”
- D. **Requerimientos operacionales:** Que sea cómodo: si el sistema SUBE tiene un teclado para que cada uno tipee su número de DNI no sería operacional; “el sistema SUBE debe ser offline y resolver sola si cobra o no”
- E. **Requerimientos de desarrollo:** De qué forma se va a desarrollar ese producto que yo quiero obtener. No es tarea del analista determinar (por ejemplo) en que lenguaje se va a desarrollar un sistema, sino del equipo de programadores.

Importante: “se deberá desarrollar en java” está mal. A veces hay algunas circunstancias particulares que el ingeniero de requisitos detecta: hay condicionantes de la organización que afectan a este desarrollo.

- F. **Requerimientos externos:** No afectan al producto ni surgen de la organización.
- G. **Requerimientos regulatorios:** El cumplimiento de una regla, ley o norma. “el sistema SUBE debe respetar la ley X de protección de datos personales”
- H. **Requerimientos éticos:** No necesariamente está determinado por la ley o norma, sino que tiene que ver con valores o principios de la organización o del dominio: “el sistema podría evitar la caracterización por género” “el docente no puede enterarse de si el alumno pago o no la cuota”
- I. **Requerimientos contables:** Lo que regula todo lo referido a la contabilidad → manejo de transacciones económicas y capitales. Si se hace un sistema de software contable, va a haber que fijarse todas las cuestiones legales contables
- J. **Requerimientos de protección y seguridad:** Ejemplo: ley de protección d datos personales.

Proceso de adquisición de conocimientos

Técnicas:

- **Elicitar:** actitud activa/escucha activa. No ser estático y aceptar solo lo que el cliente nos dice. Repregunto, analizo, parafraseo, etc. para descubrir nuevas necesidades. Se plantean distintas situaciones. Además, eso mismo que nos cuentan, podemos verificarlo con ellos, y si hay algo que vemos incoherente se puede revisar. En la jerga no lo aplicamos, pero a veces no se conoce. Vamos hacia las personas, utilizamos a las personas con distintos métodos.

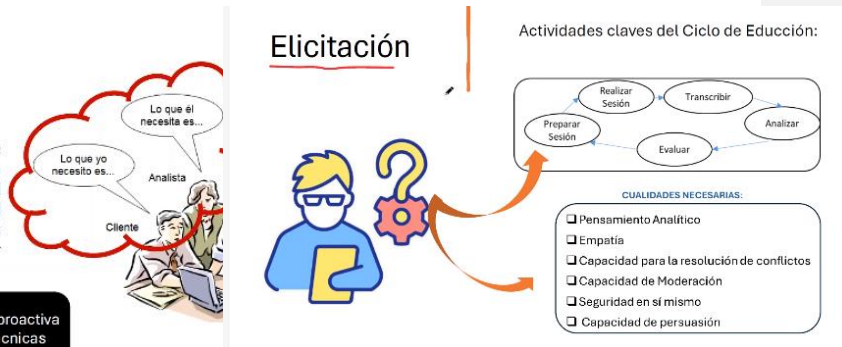


Figura 11 - Elementos que forman el proceso de elicitación

¿Qué es la Elicitación?

Es un proceso de adquisición de conocimiento donde se aplican técnicas para entender mejor el negocio que será impactado por el proyecto, para identificar a los interesados y para mejorar o definir los requisitos (RF, RNF y RN).

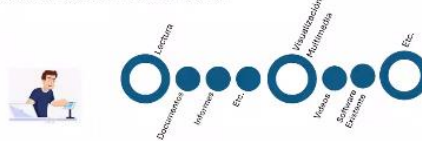
Objetivo: obtener información de manera proactiva junto a las partes interesadas utilizando técnicas seleccionadas en la preparación.



Técnicas de Elicitación

Se utilizan para recopilar información sobre el negocio del cliente o sobre los requisitos del sistema a desarrollar

Técnicas de Extracción de Información



Técnicas de Educación de Información



- **Relevar:** Concepto estático. Solo escuchar y tomar nota del cliente, nos conformamos con lo que nos dicen.

Importante: Se piensa que los analistas deben determinar el sistema que el cliente quiere. ERROR. Por el contrario, el objetivo es determinar qué software es el que el cliente necesita.

Fuente de información: Todo aquello que pueda ayudarnos a comprender tanto el dominio, como el problema en sí mismo.

Identificación de fuente:

Educir información: Obtenerla de una persona; de sus mentes, sus ideas.
→ si queremos implementar algo que no existe, y no tiene antecedentes de ningún tipo.

Extraer información: Obtenerla de un documento o artefacto; registros documentales. Múltiples fuentes; ninguna es humana.
→ Si tenemos antecedentes y queremos digitalizarlo, mejorarlo, etc

Importante: Priorizar fuentes de información (dependiendo el contexto del proyecto) según:

- costo de cada fuente de información
- cantidad de información que pueda obtenerse
- calidad de la información
- disponibilidad de la fuente de información

Recolección de hechos: conseguir todos los requerimientos del sistema.

Dos métodos de educación:

Técnicas de Educación

[A. Entrevistas \(Abiertas y Cerradas\)](#)
[B. Cuestionario / Encuesta \(Abiertos y Cerrados\)](#)
[C. Joint Application Development JAD](#)
[D. Brain Storming](#)
[E. Concept Mapping](#)
[F. Storyboard](#)
[G. Card sorting](#)
[H. Análisis de protocolos](#)
[I. Observación](#)

→ **Directos:** le preguntan directamente al experto lo que sabe; es decir, en ellos el experto reporta los conocimientos que él puede articular directamente. El experto es la única fuente de información.

- Entrevistas
- Cuestionarios

→ **Indirectos:** Se usan porque no siempre los expertos pueden acceder a los detalles de sus conocimientos o procesos mentales, y para confirmar lo adquirido mediante técnicas directas.

- Observación de Tareas Habituales
- Clasificación de Conceptos
- Análisis de Protocolos

A. Entrevista:

Entrevista

Es una forma de diálogo formal entre **dos o más personas**, donde el entrevistador **busca respuestas** a un conjunto de preguntas **planeadas** y el entrevistado les da respuesta.



“AQUEL QUE PREGUNTA
ES UN TONTO
POR CINCO MINUTOS,
PERO EL QUE NO PREGUNTA
PERMANECE TONTO
POR SIEMPRE”

Proverbio chino

PASOS PARA PREPARAR UNA ENTREVISTA:

1. Conocer antecedentes
2. Establecer objetivos de la entrevista
3. Decidir a quién entrevistar
4. Preparar al entrevistado
5. Decidir el **tipo y estructura de las preguntas**

Entrevista Abierta o No Estructurada

Vs

Entrevista Cerrada o Estructurada



- 1) De que empresa es, con quien voy a trabajar
- 2) Preciso entender el proceso paso a paso
- 3) Preparar al entrevistado: transmitirle quienes van a hablar

Tipos:

Abierta (no estructurada): puedo ir planteando distintas cosas sabiendo el objetivo final de la reunión; voy sacando temas consciente o inconscientemente que me permita ver a donde quiero llegar. El texto de la entrevista se puede extender todo lo que uno quiera. **Buen método para directivos**

Cerrada (estructurada): preguntas por si o por no, o por determinado dominio de valores; si hay documentación.

Ventajas:

- la posibilidad del contacto directo con los entrevistados, quienes tienen el conocimiento sobre los objetivos del sistema
- posibilidad de validación inmediata a través de procesos de comunicación.

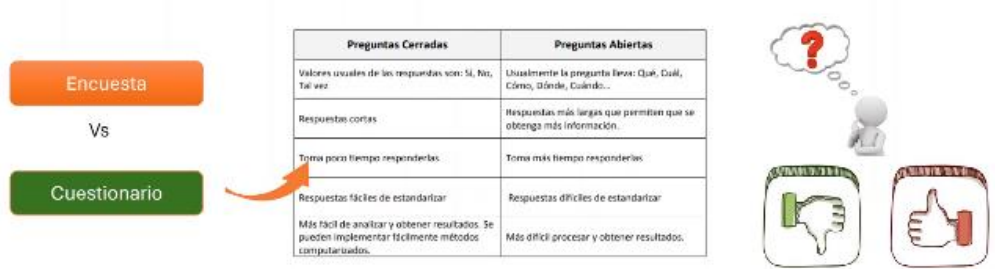
Desventajas:

- Asumir un conocimiento tácito tanto por parte del analista como del entrevistado
- Problemas por no detectar el vocabulario del entrevistado.
- Diferencias culturales entre entrevistador y entrevistado.

B. Cuestionario/Encuesta

Cuestionario / Encuesta

Consiste en plantear un conjunto de preguntas a individuos representativos, con el fin de obtener información cuantitativa o cualitativa sobre opiniones, actitudes o comportamientos.



Diferencia entre Encuesta y Cuestionario:

- La encuesta es un procedimiento
- El cuestionario es un instrumento.


C. Desarrollo conjunto de aplicaciones (JAD)

Técnica que permite definir y desarrollar software

Juntar distintos perfiles de una organización p desarrollar trabajo conjunto

Desarrollo Conjunto de Aplicaciones (JAD)

El JAD se basa en organizar reuniones integradas por **directivos, desarrolladores y miembros de la organización cliente** para la que se va a desarrollar el sistema software.



PRINCIPIOS EN LOS QUE SE BASA JAD :


- ✓ Dinámica de grupo
- ✓ El uso de ayudas visuales
- ✓ Mantener un proceso organizado y racional
- ✓ Documentación WYSIWYG (What You See Is What You Get)



D. Brainstorming

Lluvia de Ideas (Brainstorming)

Es una técnica de reuniones en grupo cuyo objetivo es la generación de ideas en un ambiente libre de críticas o juicios.



PASOS PARA PREPARAR UN BRAINSTORMING:

1. Definición del objetivo
2. Selección del grupo
3. Creación de un ambiente favorable
4. Generación de ideas
5. Analizar ideas
6. Selección de la idea

Idea / criterio	Tiempo de implementación	Originalidad	Presupuesto	Total
Idea 1	2	3	3	9
Idea 2	4	5	3	12
Idea 3	3	2	5	10



2) selección de grupo: los de adentro y los de afuera

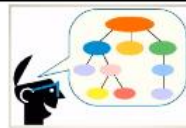
Los de adentro tienen todo el conocimiento de lo que debe hacerse. Pueden estar sesgados por lo que creen que es correcto según ellos. Condicionados por la historia.

Desventajas del brainstorming: precisas una buena compenetración de la gente.

E. Concept Mapping

Mapas de Concepto (Concept Mapping)

Son grafos en los que los vértices representan conceptos y las aristas representan posibles relaciones entre dichos conceptos. Permiten aclarar conceptos relacionados al sistema a desarrollar.



ETAPAS DEL PROCESO DE CREACIÓN DE UN CONCEPT MAPPING:

1. Extracción de los conceptos
2. Selección y primera organización
3. Creación del mapa conceptual a mano
4. Diseño digital



F. Storyboard

Maquetas de Interacción – Guión Gráfico (Storyboard)

El Storyboarding es una técnica de elicitación cuyo principal objetivo es lograr una reacción o feedback por parte de los usuarios acerca del sistema, en las primeras etapas del proceso de desarrollo



TIPS PARA LA CREACIÓN DE STORYBOARDS:

1. No invertir mucho tiempo en el StoryBoard
2. Simple de entender y fácil de modificar
3. No debe ser demasiado funcional
4. Buscar la interacción con el usuario



ETAPAS:

1. Planificación
2. División de escenas y secuencias
3. Dibujo de paneles
4. Anotaciones y detalles
5. Revisión y ajustes



Cada parte debe tener un objetivo: se dibuja en paneles, se hacen anotaciones y se van haciendo revisión y ajustes. Técnica donde represento en papel técnicas de usuario como un sketch

G. Card Sorting (ordenamiento/priorización de tarjetas)

Priorización de Tarjetas (Card Sorting)

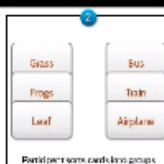
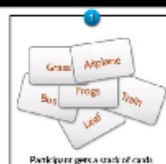
Técnica que se usa para categorizar contenidos. Se basa en la observación de cómo los usuarios agrupan y asocian entre sí un número predeterminado de tarjetas etiquetadas con las diferentes categorías temáticas



Abierto

Vs

Cerrado



Se definen diferentes áreas o partes temáticas, tipos.

H. Análisis de protocolos

Análisis de Protocolos

Esta estrategia consiste en analizar el trabajo del experto a través de sus relatos. Esto se hace para descubrir el proceso mental subyacente que realiza en forma natural el experto.



Se relata y se transcribe

I. Observación

Observación / Etnografía

Consiste en la observación directa de las prácticas profesionales que se realizan habitualmente en la organización para la que se va a desarrollar el software. Se debe intentar que los resultados de la práctica profesional sean observables en el entorno real de trabajo



Observación Participante

Vs

Observación No Participante



Trae personas de uno u otro grupo que puedan observar. Observación de un experto, de manera sistemática (mide).

Dos tipos:

Observación participante: escucha activa, embebido en la situación que se está desarrollando. Orientador de lo que se lleva a cabo.

Observación no participante: escucha pasiva. Observa únicamente lo que se va trabajando. Influye y puede ser empatizado.

Modelo conceptual/de dominio (Clair Larman): elemento o grafico que permite determinar distintos elementos conceptuales que intervienen en un espacio que se está analizando; permite modelar una parte de la realidad; permite ver y conocer cuáles son los datos requeridos, los elementos conceptuales que intervienen. No le interesa el proceso.

Reglas de dominio/negocio

Dominio: Pequeña porción de la realidad donde un sistema va a operar. No necesariamente es un requisito (no es algo que se pide del sistema, sino algo que rige en el lugar donde este se desarrolla). Posee reglas, pero estas quedan por fuera de los límites del sistema, a pesar de ser una regla del negocio.

Analogía dominio de funciones: valores que puede tomar esa función.

Ejemplo: SUBE opera en el pago de pasajes de pasajeros de Argentina.

Ejemplo: PedidosYa no tiene la obligación de asegurar que no se venda una bebida alcohólica a un menor, a pesar de ser una regla del dominio donde ocurre esta aplicación. No controla esa regla.

Definir dominio de sistema: Restringir esa porción de la realidad sobre la cual va a actuar y va a afectar. Determinar los límites de un sistema, el cual recibe información del mismo y tendrá una salida: información. Ingeniero debe realizar una buena especificación del dominio.

Modelo de dominio: usar instancias de lenguaje formal, o una imagen para explicar mejor. Pequeña porción de la realidad, no del sistema en sí, sino de en donde lo voy a meter.

Sistema ≠ Dominio

De un sistema de información se toman datos, que se procesan y se obtiene información para la toma de decisiones.

Fuera del dominio → no es relevante para el sistema

Cardinalidad: # de atributos de un concepto que pueden relacionarse con los de otro concepto. de 1 a muchos; de 1 a 0 a muchos (punto 1), de muchos a muchos, o de 1 a 1

El mínimo no es tan importante: 0 → opcional; 1 → obligatorio

De 1 a muchos: sentido → del que tiene menos al que tiene mas

De muchos a muchos → es un problema, y es altamente probable que en el medio falte otro concepto. Para qué lado vaya la flecha es trivial, o preguntarse: de los dos conceptos, cual aparece primero?

Elementos:

Concepto (sinónimo de elemento) o clase conceptual: una cosa que hay en ese dominio. Ej: dominio: ecosistema de laguna, hay peces, agua, plantas, etc. → conceptos de dominio “laguna”

Los conceptos tienen atributos, y además se relacionan entre sí

Atributos: datos que cada una de esas “cosas” (concepto) tienen.

Sirven para:

- diferenciar casos particulares de esos conceptos. Ej: concepto “pececito”, casos particulares: dory y nemo. Se diferencian por el atributo: color
- definir un concepto en base a un conjunto de atributos (definiciones: por comprensión, por descripción, o por atributos)

Para armar modelos de dominio: ponemos **conceptos**, que **atributos** tienen esos conceptos, y por ultimo las **relaciones** entre conceptos.

Técnicas: (rebeca wilf)

- Cuando parto de un texto que describe un dominio, de los SUSTANTIVOS puedo sacar los CONCEPTOS y los ATRIBUTOS (no me permite diferenciarlos)
- Los VERBOS van a estar relacionados con los VINCULOS de esos conceptos.
- Regla del cartógrafo: cuando hagas un mapa (diagrama) utilice los nombres del territorio (los términos)
- Arrancar por transacción o elemento principal.

Limitación de modelo: no puede haber relación $n \times n$ porque no trae claridad: es importante poder identificar si hay una relación que justifique una cardinalidad mayor. Si no es relevante ese n a n , no la ponemos, resaltamos el concepto mas fuerte. Ahora, si esa relación existe y es persistente en el tiempo, debería haber una relación conceptual que las vincule. Tiene que ser real el vínculo, no algo ficticio.

Ejemplo: un escritor escribe varios libros – un libro puede ser escrito por varios escritores → se busca atributo que vincule y explique esa relación

Comentado [JD1]: Consultas varias

Caso de Estudio:

TULM (Transportes Unidos de La Matanza) es una empresa de servicios de transporte público de pasajeros mediante micro ómnibus ("colectivos") que opera dentro del ámbito de nuestro municipio.

TULM administra su operatoria con un sistema denominado MasterBondi, que le permite automatizar varios procesos y tareas para los diferentes roles que interactúan con él.

Uno de los más destacados de estos roles es el del supervisor y (entre otras cosas), será el encargado de registrar mediante MasterBondi, la asignación de chofer y vehículo a los diferentes servicios oportunamente predefinidos.

Por ejemplo, para el servicio "San Justo-LaFerrere por Ruta 3", con hora de inicio definida para las 09:00 Hs., se le asigna la unidad 201 (Dominio "NPS907"- Modelo "Mercedes Benz 2084"- Año 2013) y al chofer "Juan Gómez" (de Legajo 2164 y Licencia Profesional de Conducir N° 23.749.106).

Cada vez que se registra la asignación de un servicio (al menos dos horas antes de la partida del mismo), Master Bondi enviará automáticamente un mensaje de WhatsApp al chofer designado, con el objeto de notificarle, y un aviso automático al sistema de contralor de la SNTP (Superintendencia Nacional de Transporte Público), a los efectos de que ésta contabilice dicha asignación.

Si por algún motivo de fuerza mayor, el servicio no se cumple (la unidad no sale), el supervisor deberá reportar mediante MasterBondi, a la SNTP el incumplimiento, seleccionando uno de los causales predeterminados para el mismo.

Responda:

- La expresión "Para MasterBondi, el tráfico de información con la SNTP deberá estar protegida por tecnologías de cifrado de extremo a extremo(E2EE)" por técnicas de intercambio de clave de Diffie-Hellman, según lo establecido en el protocolo de interconexión determinado para el Web Service de la SNTP, especificado en la circular 1585/22 de la misma, refiere a:
 - a) Un requisito no funcional;**
 - e) Una regla del negocio (o dominio);**
- La expresión "Automáticamente, todos los días a las 02:00 horas, MasterBondi deberá realizar una copia de seguridad automática de todos sus datos de las últimas 24 horas y dicho proceso no debe tardar más que(a lo sumo) 3 minutos":(única opción)
 - c)Expone un requisito funcional, y uno no funcional asociado a él**
- Según Sommerville: ¿Que artefactos no forman parte de la documentación de requisitos ("Requerimientos", según el autor)?
 - b) El informe de viabilidad;**
- Según Karl Wieggers¿Cuales son las cuatro fases del desarrollo de requisitos?
 - a) Análisis;**
 - f) Verificación;**
 - i) Especificación;**
 - j) Elicitación;**

- Cuando la Ingeniería de Requisitos falla, suele terminar produciéndose:
b) Un sistema incorrecto, que puede estar correcta o incorrectamente construido;
- En el "Juego del Desarrollo de Sistemas", son integrantes del equipo del cliente (debe señalar a todos ellos):
a) El sponsor (o facilitador);
f) Los usuarios actuales (o futuros) del sistema;
- Según lo estrictamente especificado en el enunciado, un modelo del dominio de aplicación para el sistema MasterBondi presentará, como conceptos infaltables:
c) La asignación de un chofer y una unidad a un servicio;
e) El servicio;
- En un modelo del dominio de aplicación para MasterBondi (señale las respuestas correctas):
a) La asociación entre las clases conceptuales "Unidad" y chofer sería directa y de cardinalidad "uno a muchos", de la primera hacia la segunda;
- Técnicas de extracción válida para la adquisición del conocimiento sobre MasterBondi podría ser:
- **b) Ninguna de las opciones listadas es una técnica de extracción;**
- Al momento de educir información de los niveles superiores de la organización del cliente (dueños, directores, gerentes, etc). ¿Cuales son técnicas óptimas de aplicación?
d) Una entrevista abierta;

Casos de estudio:

- 1) Rta
- 2) Rta
- 3) Rta
- 4) Rta
- 5) Rta
- 6) Rta
- 7) Rta
- 8) Rta
- 9) Rta
- 10) Rta
- 11) Rta
- 12) Rta
- 13) Rta

Comentado [JD2]: Pregunta 1:

Comentado [JD3]: Pregunta 2:

Comentado [JD4]: Pregunta 3:

Comentado [JD5]: Pregunta 4:

Comentado [JD6]: Pregunta 5:

Comentado [JD7]: Pregunta 6:

Comentado [JD8]: Pregunta 7:

Comentado [JD9]: Pregunta 8: