Universidad Nacional Autónoma de México
Facultad de Contaduría y Administración
Sistema Universidad Abierta y Educación a Distancia

Licenciatura en Informática

Informatica II (Admón. de requerimientos)

Apunte electrónico

SUAyED

COLABORADORES

DIRECTOR DE LA FCA
Dr. Juan Alberto Adam Siade

SECRETARIO GENERAL
L.C. y E.F. Leonel Sebastián Chavarría

– – – –

COORDINACIÓN GENERAL
Mtra. Gabriela Montero Montiel
Jefe de la División SUAyED-FCA-UNAM

COORDINACIÓN ACADÉMICA
Mtro. Francisco Hernández Mendoza
FCA-UNAM

– – –

COAUTOR
Ing. René Montesano Brand

DISEÑO INSTRUCCIONAL
Mayra Lilia Velasco Chacón

CORRECCIÓN DE ESTILO
L.F. Francisco Vladimir Aceves Gaytán

DISEÑO DE PORTADAS
L.CG. Ricardo Alberto Báez Caballero
Mtra. Marlene Olga Ramírez Chavero
L.DP. Ethel Alejandra Butrón Gutiérrez

DISEÑO EDITORIAL
Mtra. Marlene Olga Ramírez Chavero

OBJETIVO GENERAL

Al finalizar el curso, el alumno será capaz de identificar y especificar los requerimientos de los involucrados en el desarrollo de un sistema de información a fin de orientar las actividades de análisis y diseño de sistemas.

TERMARIO OFICIAL

(96 HORAS)

	Horas
1. Introducción	16
2. Identificación de requerimientos	24
3. Especificación de requerimientos	28
4. Validación de requerimientos	28
TOTAL	96

INTRODUCCIÓN GENERAL

Dentro del ciclo de vida de los sistemas de información, las fases de compilación de información y análisis de requerimientos son de las más importantes.

En la fase de compilación de información, el analista podrá generar una vista general del sistema por desarrollar. Aquí es importante que el desarrollador tenga contacto con los que serán los usuarios finales del sistema, ya que a partir de sus necesidades, aportaciones e información ofrecida, será posible desarrollar un sistema robusto y amigable para ellos.

Una vez recopilada, ordenada y analizada la información, se procede a la fase de análisis de requerimientos, los que proporcionarán a los desarrolladores del sistema un panorama general de los requisitos mínimos indispensables para iniciar la construcción del sistema. De esta fase se obtiene información como las características de los equipos que operarán al sistema, el tipo de estrategia de desarrollo por seguir (selección del modelo de desarrollo), el tipo de información por manejar, estructura general, etcétera.

A lo largo de la presente asignatura, comprenderemos a profundidad cada una de las fases mencionadas en cuatro partes principalmente.

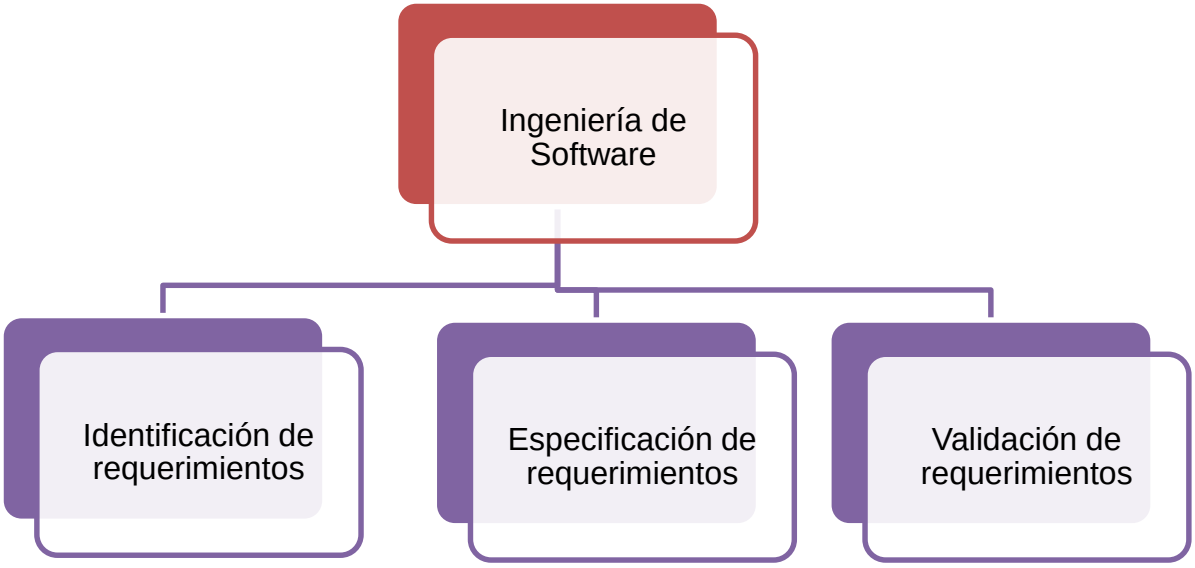
La unidad 1, Introducción, nos dará una visión general de la ingeniería de software, proporcionándonos los principios fundamentales para el desarrollo de sistemas de información robustos y completos.

En la unidad 2, Identificación de requerimientos, se profundizará en las características de aquellas partes que son esenciales para comenzar a analizar y desarrollar un sistema de información. Se revisará qué requisitos son necesarios y cómo los podemos clasificar para poder iniciar nuestra fase de diseño y posterior construcción del sistema.

En la unidad 3, Especificación de requerimientos, se analizarán formas para recabar información de los usuarios empleando diferentes herramientas como la entrevista, la encuesta, la observación participativa y no participativa, entre otras, con el objetivo de establecer requerimientos base para comenzar el desarrollo del sistema de información

En la unidad 4, Validación de requerimientos, veremos que, una vez recabada y analizada la información, se procederá a verificar si los requerimientos establecidos son funcionales, a través de la validación de cada uno de ellos, a fin de ser aceptados y se pueda proceder al desarrollo del proyecto de software.

ESTRUCTURA CONCEPTUAL



Unidad 1

Introducción

OBJETIVO PARTICULAR

Desarrollar un plan para la administración de requerimientos tomando como base los conceptos y clasificación de los requerimientos.

TEMARIO DETALLADO

(16 HORAS)

1. Introducción

1.1. Requerimientos del sistema
1.2. Requerimientos del usuario
1.3. Requerimientos de software

INTRODUCCIÓN

La ingeniería del software, en palabras de Fritz Bauer (Pressman, 2002, pp.17-19), “es el establecimiento y uso de principios robustos de la ingeniería a fin de obtener económicamente software que sea fiable y que funcione eficientemente sobre máquinas reales.” Esta definición puede ser aceptable a primera vista, pero no dice mucho sobre los aspectos técnicos de la calidad del software, no se enfrenta directamente con la necesidad de la satisfacción del cliente o de la entrega oportuna del producto, omite la mención de la importancia de mediciones y métricas, así como tampoco expresa la importancia de un proceso avanzado. Sin embargo, ésta nos proporciona una línea base para preguntarnos ¿cuáles son los principios robustos de la ingeniería aplicables al desarrollo de software para computadora? ¿Cómo construimos software económico que sea fiable? Estas cuestiones representan un reto para los ingenieros del software.

La ingeniería de software es una tecnología multicapas (figura 1). Por lo que no debemos olvidar que cualquier enfoque de la ingeniería (incluida la ingeniería del software), debe estar basado en un empeño de organización de calidad.

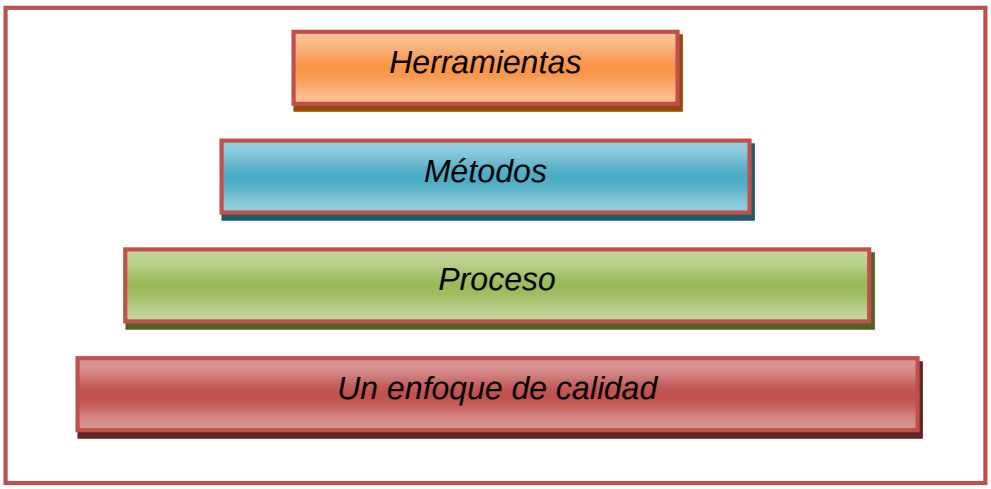


Figura 1. Capas de Ingeniería de Software

De lo expuesto se advierte que, el fundamento de la ingeniería de software es la capa proceso y el proceso de la ingeniería del software es la unión que mantiene juntas las capas de tecnología y que permite un desarrollo racional y oportuno de la ingeniería. Por lo que el proceso define un marco de trabajo, para un conjunto de áreas claves de proceso, que se debe establecer para la entrega efectiva de la tecnología de la ingeniería de software.

Los métodos de la ingeniería de software indican cómo construir técnicamente el software. Estos métodos abarcan una gran gama de tareas, que incluyen el análisis de requisitos, como el diseño, la construcción de programas, pruebas y mantenimiento.

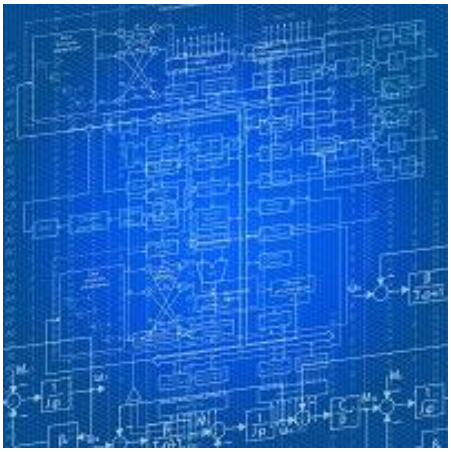
Las herramientas de la ingeniería del software proporcionan un soporte automático o semi-automático para el proceso y para los métodos. Cuando se integran herramientas para que la información creada por una de estas herramientas pueda ser utilizada por otra, se establece un sistema de soporte para el desarrollo del software llamado Ingeniería del Software Asistida por Computadora (*Computer-Aided Software Engineering CASE*). Se pueden ubicar las que combinan software, hardware y una base de datos de ingeniería de software (un depósito que contiene información importante sobre el análisis, el diseño, y la construcción de programas y pruebas), y las que crean un entorno de ingeniería del software que sea análogo a CAD/CAE para el hardware. (*Computer-Aide Desing/Engineering* / Diseño / Ingeniería Asistida por Computadora).

Para construir la ingeniería del software adecuadamente se debe definir un proceso o método de desarrollo. Dicho proceso, se puede dividir en tres fases genéricas, con independencia del área de la aplicación, tamaño o complejidad del proyecto.

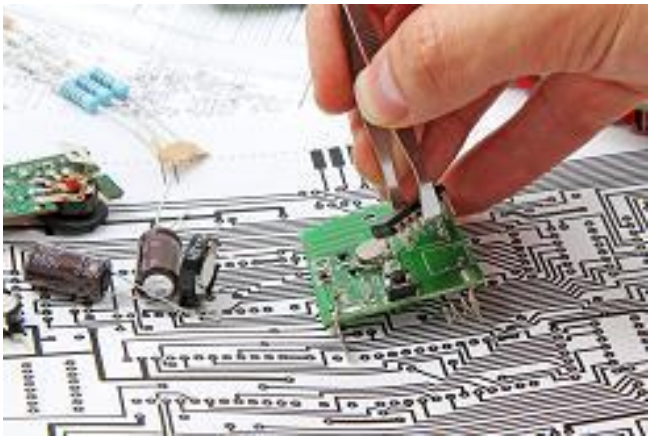
La primera es la fase de definición, la cual se centra sobre el ¿qué? Es decir, durante la definición, se intenta identificar ¿qué información ha de ser procesada?, ¿qué función y rendimiento se desea?, ¿qué comportamiento del sistema se quiere?, ¿qué interfaces van a ser establecidas?, ¿qué restricciones de diseño existen? y ¿qué criterios de validación se necesitan para definir un sistema correcto?

Por lo tanto, deben identificarse los requisitos clave del sistema y del software. Aunque los métodos aplicados durante la fase de definición varían, dependiendo de los paradigmas de ingeniería de software que se apliquen, éstos darán lugar a tres tareas principales: ingeniería de sistemas o de información; planificación del proyecto del software y análisis de los requisitos.

La siguiente es la fase de desarrollo, que se centra en el ¿cómo? Es decir, durante su desarrollo, un ingeniero del software intenta definir ¿cómo han de diseñarse las estructuras de datos?, ¿cómo ha de construirse la función como una arquitectura del software?, ¿cómo han de implantarse detalles sobre los procedimientos?, ¿cómo han de caracterizarse las interfaces?, ¿cómo ha de traducirse el diseño en el lenguaje de programación? y ¿cómo deben de realizarse las pruebas? Estos métodos aplicados durante la fase de desarrollo, variarán, no obstante las siguientes tareas ocurrirán siempre; en el diseño del software, la generación de código y la prueba del software.



Finalmente, la fase de mantenimiento se centra en el cambio que va asociado a la corrección de errores, a las adaptaciones requeridas a medida que evoluciona el entorno del software y a cambios debidos a las mejoras producidas por los requisitos cambiantes de las necesidades del cliente. Dentro de la fase de mantenimiento, se vuelven a aplicar los pasos de las fases de definición y desarrollo, pero en el contexto del software estas ya existen. Durante la fase de mantenimiento se encuentran cuatro tipos de cambios:



1. **Corrección**

El mantenimiento correctivo modifica el software para corregir los defectos.

2. Adaptación

El mantenimiento adaptativo produce modificación en el software para acomodarlo a los cambios de su entorno externo.

3. Mejora

Se lleva el software más allá de los requisitos funcionales originales.

4. **Prevención**

También llamado reingeniería del software, es hacer cambios en programas de computadora a fin de que se pueda corregir, adaptar y mejorar más fácilmente.

1.1. Requerimientos del sistema

Parte fundamental dentro del desarrollo y construcción de sistemas informáticos es la recopilación de información para su análisis y posterior uso.

Dentro de las principales consideraciones por tomar (al revisar los requerimientos iniciales de un sistema) está la de que toda organización tiene necesidades diferentes y, por tanto, la solución a implementar tanto de software como de hardware deberá ajustarse a fin de satisfacer dichas necesidades.

Al recopilar información, es necesario trabajar tanto con los empleados como con los administradores, es decir, no solamente con quienes desean el sistema, sino también con los usuarios finales; lo anterior implica un estudio detallado de todos los procesos de una organización que nos ayuden a responder una serie de preguntas que son clave para determinar los requerimientos iniciales, las cuales son:

1. ¿Qué es lo que se hace?
2. ¿Cómo se hace?
3. ¿Con qué frecuencia se presenta?
4. ¿Qué tan grande es el volumen de transacciones o de decisiones?
5. ¿Cuál es el grado de eficiencia con el que se efectúan las tareas?
6. ¿Existe algún problema?
7. 7.- Si existe un problema, ¿Qué tan serio es?
8. Si existe un problema, ¿Cuál es la causa que lo origina? (Flores, “Determinación de requerimientos”)

Las preguntas anteriores servirán como base para poder determinar las características iniciales que el sistema a desarrollar deberá tener, la forma de obtener dicha información puede variar, utilizando elementos como encuestas, cuestionarios, entrevistas, etc.

1.2. Requerimientos del usuario

Antes de definir cuáles son los requerimientos de un usuario, primero debemos identificar qué es un usuario, y de acuerdo con el profesor López Gormaz (n.d.): “el usuario es la persona que hace uso de un producto tecnológico” (p. 2), es decir, son las personas que van a tener contacto directo con el sistema.

Dentro del ámbito informático, es posible dividir a los usuarios en 5 categorías (Flores, n.d., Requerimientos de los usuarios):

1. Dueños del Sistema
2. Usuarios del sistema
3. Diseñadores del sistema
4. Constructores del sistena
5. Analistas del sisrtema

Los **dueños del sistema**, como el nombre lo indica, son aquellos que tendrán la propiedad del sistema final, en otras palabras, son los responsables de ordenar su construcción, cubrir el costo del mismo y de las decisiones que involucran su uso y funcionalidad.

Los **usuarios del sistema** son aquellos que tendrán contacto directo con el producto terminado, los que emplearán las diversas interfaces para procesar, capturar y consultar la información contenida en el sistema final.

Los **diseñadores del sistema** son los que se involucran en la creación del sistema desde sus etapas más tempranas; son los encargados de analizar todos los requerimientos iniciales y de determinar las etapas y métodos de construcción del mismo.

Los **constructores del sistema** son especialistas que se encargan de transformar el análisis previo de requerimientos en un sistema informático funcional, aquí podemos encontrar a los programadores, capturitas, analistas, entre otros.

El **analista de sistemas** evalúa, de manera sistemática, el funcionamiento de un negocio mediante el examen de la entrada, el procesamiento de datos y su consiguiente producción de información, con el propósito de mejorar los procesos de una organización. Muchas mejoras incluyen un mayor apoyo a las funciones de negocio a través del uso de sistemas de información computarizados (Flores, n.d., “Requerimientos de los usuarios”).

Para poder determinar las necesidades del usuario potencial de un sistema, se deben plantear preguntas que apunten a determinar cuáles son las características y funciones que debe cumplir el objeto para satisfacer las necesidades y las expectativas de éste. Para ello, existen algunas variables que es preciso tener en cuenta y que orientan a definir la información que se va a recoger:

Variables geográficas	Es importante que se defina en qué área geográfica se encuentra el usuario al cual está destinado el sistema por diseñar.
Variables preferenciales	Las distintas preferencias determinan las características de varias de las interfaces que puede contener el sistema por diseñar.
Variables de género y estilo	Aspectos como la edad, el sexo y estilos de vida, son importantes de considerar al momento de determinar las características y funciones del sistema.

1.3. Requerimientos de software

Dentro del ámbito de los requerimientos se manejan dos enfoques:

→ Desde el **punto de vista de la ingeniería del software**, los requerimientos parten del mismo análisis inicial del sistema. Es una etapa temprana dentro del desarrollo del sistema que se enfoca en la obtención, análisis, especificación y validación de los requerimientos para el software.

Dentro de la ingeniería de software, entendemos como requerimientos a “las declaraciones que identifican atributos, capacidades, características y/o cualidades que necesita cumplir un sistema para que tenga valor y utilidad para el usuario” (Diccionario de informática, 2011). Es decir, un requerimiento muestra todos los elementos que son necesarios para la construcción del sistema.

Desde el **punto de vista del software**, un requerimiento es una serie de elementos básicos necesarios para que las aplicaciones funcionen de manera correcta, estos pueden ser, cantidad de memoria del ordenador, sistema operativo, tipo de procesador, entre otros.

RESUMEN DE LA UNIDAD

La ingeniería de software se enfoca al desarrollo de software para computadora. Este desarrollo se hace con base en una planeación previa y un proceso definido, y está orientado a la atención de necesidades específicas. Parte de una investigación previa del entorno donde el software actuará y las funciones que realizará; recopila información para determinar las características y requerimientos del software, así como las necesidades del usuario final. La ingeniería de software contempla, además de la construcción del software, aspectos como calidad, pruebas de funcionamiento, mantenimiento y viabilidad del mismo, en un futuro.

BIBLIOGRAFÍA DE LA UNIDAD

SUGERIDA

Autor	Capítulo	Páginas
Bruegge (2001)	4	120-131
Pressman (2002)	1	17-25
Sommerville (2001)	1	24-35

Unidad 2

Identificación de requerimientos

OBJETIVO ESPECÍFICO

Al finalizar la unidad, el alumno podrá seleccionar y aplicar los métodos y las técnicas más apropiadas para identificar los requerimientos para la construcción de un sistema.

TEMARIO DETALLADO

(24 HORAS)

2. Identificación de los requerimientos

2.1. Concepto
2.2. Identificación de necesidades
2.3. Clasificación de los requerimientos
2.4 Análisis de requerimientos
2.5 Anticipación de requerimientos

INTRODUCCIÓN

A lo largo de la presente unidad, se entenderá el concepto general de requerimiento o requisito de un sistema, como la parte fundamental del diseño, y se genera en la etapa de análisis del sistema.

La etapa de análisis es fundamental para el buen desarrollo de cualquier sistema de software. En esta etapa se recaba toda la información necesaria para la construcción del sistema, partiendo de los requisitos mínimos para su funcionamiento, hasta la determinación de todas las variables que pueden afectar su funcionalidad y limitar su rendimiento.

Es conveniente contar con personal de experiencia en el análisis y selección de requerimientos, ya que gracias a ello, será posible evitar omisiones en la recopilación de datos y asegurar mayor fiabilidad de los mismos.

2.1. Concepto

Nos referiremos a los requerimientos como “las declaraciones que identifican atributos, capacidades, características y/o cualidades que necesita cumplir un sistema para que tenga valor y utilidad para el usuario” (Diccionario en informática, 2011). Es decir, un requerimiento muestra todos los elementos que son necesarios para la construcción de un sistema.

Los requerimientos de sistemas tienen que cumplir con una serie de características básicas para que se consideren válidos, como son:

1. Actual.	• Un requerimiento debe ser vigente, desde el inicio del desarrollo del sistema, hasta su finalización y liberación.
2. Cohesión.	• El requerimiento debe hacer referencia a un solo tema u objeto.
3. Completo.	• Al momento de generar un requerimiento, este debe estar perfectamente definido, no debe contener omisiones en su formulación.
4. Consistente.	• Un requerimiento no debe entrar en contradicción con otro, además de ser coherente y complementario con el sistema y su documentación.
5. Necesario.	• Un requerimiento debe ser útil en el proceso de creación del sistema, nunca debe estar de más.
6. Viable.	• Los requerimientos deben poder ser suministrables en el proceso de creación del sistema, por ello al ponderarse y definirse, éstos deben ser factibles de conseguir o de implementar.

7. Objetivo.	• Un requerimiento debe ser declarado de forma concisa, debe estar perfectamente definido sin información adicional, innecesaria o ambigua.
8. Obligatorio.	• Al presentar un requerimiento, este debe tener carácter de necesario e insustituible, por ello debe ser declarado y formulado de manera consciente y realista.
9. Cuantificable u observable.	• Los requerimientos deben poder mostrar su uso en el funcionamiento de un sistema, a fin de validar su existencia.
10. Verificable.	• Un requisito debe poder ser observado en alguna de las fases de desarrollo del sistema.

Los requerimientos de sistemas pueden ser divididos en 3 clasificaciones básicas, que son:

Requerimientos funcionales	Son aquellos que definen la forma en que el sistema deberá interactuar con su entorno, incluyendo a los usuarios y también la forma deseable de funcionamiento del mismo.
Requerimientos no funcionales	Describen aquellos aspectos que generalmente son percibidos por los usuarios del sistema y que no tienen relación directa con el funcionamiento del mismo, es decir, son aquellos aspectos que pueden impactar al sistema de forma indirecta como son, por ejemplo, el espacio de disco, el tiempo de respuesta, el consumo de recursos, etcétera.
Requerimientos externos	Son factores externos del sistema y de su proceso de creación. Entre ellos se incluyen aquellos aspectos que tienen que ver con la interacción con otros sistemas existentes, los aspectos legales, como el caso de licencias y propiedad intelectual. Este tipo de requerimientos son generalmente impuestos por los diseñadores y/o dueños del sistema, con el fin de que éste tenga aceptación entre sus usuarios directos y el público en general.

2.2. Identificación de necesidades

Para poder identificar las necesidades de un sistema, debemos partir del levantamiento de los requerimientos. Al analizar la información necesaria para la creación de un sistema será posible identificar las necesidades del mismo, lo que se desea de él, de su funcionalidad, operatividad, visión a futuro y los alcances del mismo.

Las necesidades son una serie de elementos indispensables para el correcto funcionamiento de un sistema, en el caso de las personas, por ejemplo, se tienen las necesidades básicas de alimento, resguardo y vestido, que son fundamentales para su existencia.

En los sistemas informáticos, las necesidades serán identificadas a partir del análisis de los requerimientos, es decir, se identificarán aquellos elementos que sean necesarios para cumplir con cada requerimiento dependiendo de su clasificación, pudiendo ser funcionales, no funcionales y externos.

La identificación de las necesidades de un sistema se realiza dentro de la fase de análisis del sistema, durante este proceso, el desarrollador se reúne con el cliente y/o los usuarios finales del sistema para determinar los objetivos generales del sistema, las metas globales, las perspectivas y necesidades del cliente.

Como parte de las necesidades del sistema, se definen los tiempos para el desarrollo y el presupuesto para su construcción.

La fase de identificación de necesidades suele dividirse en 5 partes:

1. *Reconocimiento del problema.*

Durante esta primera etapa, el desarrollador y el cliente buscan identificar la problemática por resolver mediante el sistema de información.

2. *Evaluación y síntesis*.

Dentro de esta etapa se define si es posible solucionar la problemática mediante el uso del sistema y se elabora un documento que describe, de forma general, la problemática y sus características.

3. *Modelado.*

En esta etapa el desarrollador realiza una serie de diagramas que identifican la forma general de operación del sistema y su interacción con los usuarios.

4. *Especificación.*

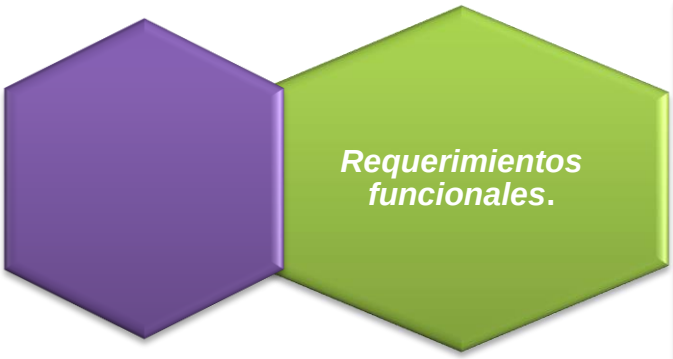
Una vez completadas las etapas anteriores, se procede a establecer los requisitos iniciales del sistema y las necesidades básicas de los mismos.

5. *Revisión*.

Establecidos los requisitos iníciales, el desarrollador se reúne con el cliente para su revisión y aprobación.

2.3. Clasificación de los requerimientos

Los requerimientos de los sistemas pueden ser divididos en 3 clasificaciones básicas:



Definen las capacidades que deberá tener el sistema por desarrollar, describiendo los procesos que llevan a la transformación de las entradas del sistema para obtener las salidas deseadas (lo que el software debe o no hacer).

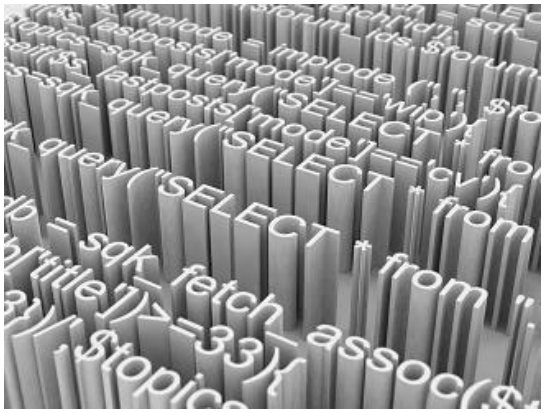
Los requerimientos funcionales se clasifican en:

- **Requerimientos de datos o información.** Estos requerimientos hacen referencia al contenido o tipo de información que manejará el sistema, el objetivo de los requerimientos de datos es responder a la pregunta ¿qué información debe almacenar y administrar el sistema?

- **Requerimientos de interfaz (con el usuario).** Son los requerimientos que buscan determinar la forma como los usuarios van a interactuar con el sistema, el objetivo de definir los requerimientos de interfaz es responder a la pregunta ¿cómo va a interactuar el usuario con el sistema?

- **Requerimientos de navegación.** Estos requerimientos están enfocados a determinar la forma en que los usuarios van a navegar o a desplazarse dentro del sistema entre las diferentes secciones que lo componen.
- **Requerimientos de personalización.** A través de los requerimientos de personalización se busca describir la forma en que el sistema deberá adaptarse a los diferentes tipos de usuario que interactuarán con él.
- **Requerimientos transaccionales o funcionales** internos. Son los requerimientos que buscan definir la funcionalidad interna del sistema, excluyendo los aspectos de interacción.
- **Requerimientos no funcionales.** Definen las posibles causas o características que son limitantes del sistema, como por ejemplo, el rendimiento del sistema, disponibilidad de equipos, etc.

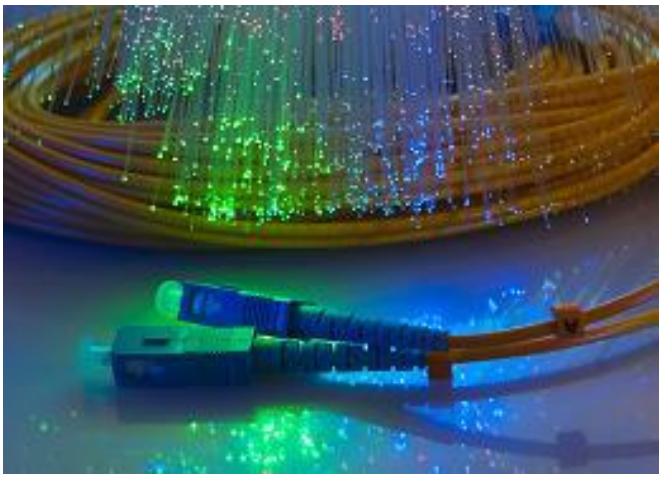
Para definir los requerimientos no funcionales los desarrolladores se basan en el estándar ISO/IEC 9126: que define un modelo independiente de la tecnología para caracterizar la calidad de software y considera las siguientes características:



- **Funcionalidad.** Define las diferentes características que son necesarias para alcanzar el funcionamiento deseado del sistema, entre estas características podemos encontrar la interoperabilidad con los usuarios y la seguridad del sistema.
- **Confiabilidad.** Se refiere a la capacidad del sistema para mantener sus niveles de rendimiento bajo ciertas condiciones específicas y en un tiempo de respuesta dado. Entre los requerimientos de confiabilidad encontramos la tolerancia a las fallas y la capacidad de recuperación del sistema.

- **Amabilidad.** Describe el nivel de complejidad que puede encontrar el usuario final para poder utilizar el sistema, entre estos requerimientos encontramos la curva de aprendizaje, la eficacia y la eficiencia de los usuarios. Para algunos autores se trata de la “usabilidad”.

- **Eficiencia.** Son las características que definen la diferencia entre el rendimiento del sistema y la cantidad de recursos que éste consume durante su funcionamiento, entre estos se encuentran la cantidad de memoria RAM empleada, el tiempo de respuesta, la cantidad de procesos en el sistema operativo y los recursos de red empelados.
- **Capacidad de mantenimiento.** Se refiere a la capacidad del sistema para aceptar cambios dentro de su estructura sin perder funcionalidad, dentro de estos aspectos se encuentran la estabilidad y las validaciones de los diferentes módulos del sistema.



- **Compatibilidad.** Es la capacidad del sistema de ser instalado en diferentes plataformas operativas sin presentar cambios en su funcionamiento como por ejemplo: adaptabilidad, capacidad de instalación.

- ***Requerimientos externos*.** Definen el medio ambiente al que se va a exponer el sistema, como por ejemplo políticas de uso, características legales, etcétera.

Los requerimientos externos dependen en gran medida de las condiciones establecidas entre los diseñadores y los clientes.

2.4. Análisis de requerimientos

El análisis de requerimientos es la tarea que facilita al desarrollador de sistemas especificar las funciones y el comportamiento de cada uno de los procesos que integran al sistema. Adicionalmente, ayuda a verificar la compatibilidad entre plataformas y con otros sistemas.

En la etapa de análisis de requerimientos es posible identificar la secuencia de la información, su representación y flujo dentro del sistema, así como es posible definir la arquitectura del mismo y el método de desarrollo a ser empleado.

Los pasos por seguir para el análisis de requerimientos son:

- Diseño y detalle de casos de uso del sistema. En este paso se muestran las distintas operaciones que se esperan de una aplicación o sistema y cómo se relaciona con su entorno (Figueroa, *Definición,* 2011).
- Definición de la interfaz inicial del sistema. Diseño preliminar en papel de las pantallas del sistema.
- Desarrollo de diagramas funcionales del sistema. Aquí se enlistan elementos y sus relaciones, como:

→ Elementos físicos y lógicos dentro del sistema por modelar.

→ Top-down: comenzar por la clase del objeto más general (el mundo). Encontrar sus componentes hasta llegar a clases de tipos básicos.

→ Identificar los sustantivos del enunciado del problema y determinar si son clases del modelo del mundo.

→ Identificar clases desde el punto de vista de la información, como:

- Elementos del espacio del problema.
- Otros sistemas relacionados como objetos externos.
- Dispositivos relacionados.
- Eventos que el sistema debe recordar y manipular.
- Roles de los elementos del mundo.
- Sitios.
- Unidades organizacionales importantes en el problema.

→ Identificar clases desde el punto de vista funcional (casos de uso).

- Objetos que participan en un caso de uso particular.

- Continuar con los mensajes de cada objeto, dejando para el final los atributos.

→ Identificar clases desde el punto de vista de sus estados.

- ¿En qué estados está el sistema? ¿Cuáles objetos determinan estos estados?
- ¿Cómo es el ciclo de vida de estos objetos? (véase, Figueroa, *Análisis*, 2011)

Derivado de lo anterior, se producen los siguientes documentos entregables:

- Casos de uso iniciales. El diagrama de casos de uso representa la forma en cómo un cliente (Actor) opera con el sistema en desarrollo, además de la forma, tipo y orden en cómo los elementos interactúan (operaciones o casos de uso).

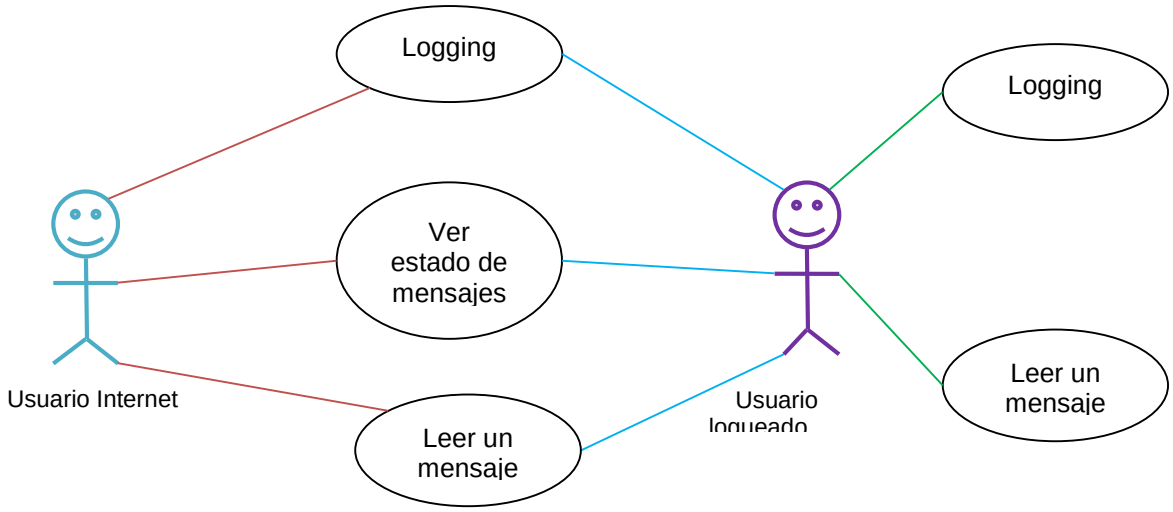
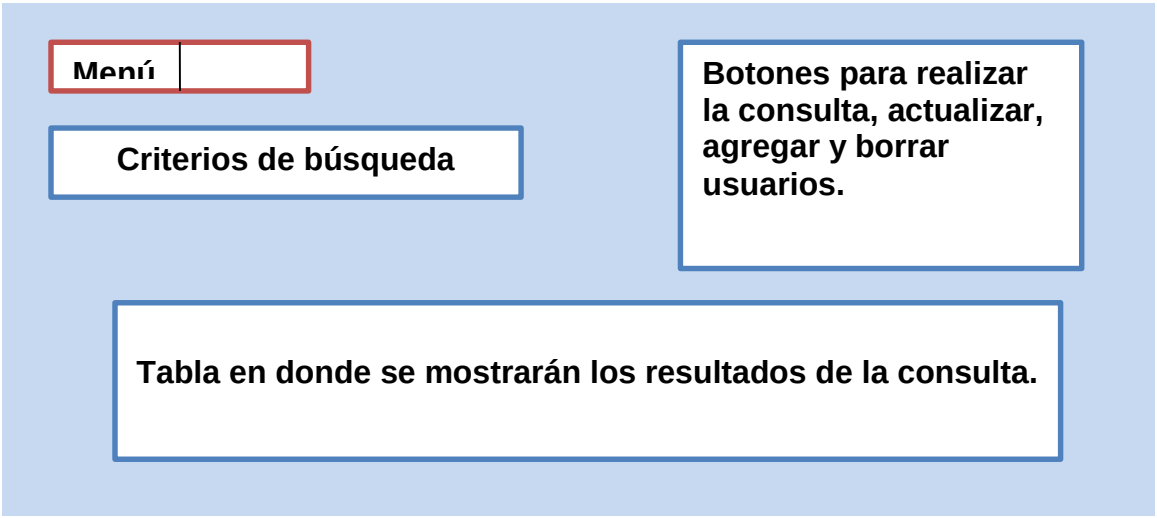


Diagrama de caso de uso[1]

- Borradores de interfaz. Los borradores de interfaz son el diseño preliminar de la forma de cómo será el aspecto de las pantallas del sistema con las que el usuario va a tener contacto.

[1] Fuente: J. Gracia (2003) UML Casos de uso, disponible en línea: http://www.ingenierosoftware.com/analisisydiseno/casosdeuso.php, recuperado el 20/11/11.



Bosquejo de una pantalla de sistema

- Modelo del mundo inicial. Es el conjunto total de diagramas generados en la fase de análisis del sistema, que muestra de forma gráfica la interacción del sistema con su entorno.

2.5. Anticipación de requerimientos

Dentro de la etapa de determinación de requerimientos de un sistema, la anticipación de requerimientos se basa en “la experiencia de los analistas, [esto] les permite anticipar ciertos problemas o características y requerimientos para un nuevo sistema” (Senn).

La experiencia en el desarrollo de sistemas es importante para poder conducir la creación de nuevos sistemas de manera sólida, eliminando posibles omisiones en el análisis de la información, además de ser base para poder determinar las preguntas o los aspectos necesarios para la funcionalidad del sistema.

Es importante no hacer omisiones o, en su caso, dar por sentado detalles en la fase de análisis, lo que puede conducir a faltantes de información que convertirían la etapa de análisis en un problema significativo.

RESUMEN DE LA UNIDAD

Es esencial contar con la lista de requerimientos, previo al diseño y construcción de un sistema. Al definir los elementos necesarios para su elaboración, permitirá contar no solo con un producto funcional que responda a necesidades particulares, sino que también ayudará a prever situaciones que limiten su rendimiento.

Para recabar esta lista de particularidades, debemos regirnos por un índice de calidad en la información que obtengamos, por lo que el análisis de ella no debe pasar por alto para verificar su eficacia y operatividad en el producto final, tanto en el ambiente interno como en el externo.

BIBLIOGRAFÍA DE LA UNIDAD

SUGERIDA

Autor	Capítulo	Páginas
Bruegge (2001)	4	120-131
Pressman (2002)	1	17-25
Sommerville (2001)	1	24-35

Unidad 3

Especificación de requerimientos

OBJETIVO ESPECÍFICO

Al finalizar la unidad el alumno registrará el detalle de los requerimientos funcionales y no funcionales.

TEMARIO DETALLADO

(28 HORAS)

3. Especificación de requerimientos

3.1. Planificación de la gestión de requerimientos
3.2. Métodos de recopilación de la información
3.2.1. Métodos estructurados
- La entrevista
- La encuesta
- El cuestionario
3.2.2. Métodos no estructurados
- La observación
- La observación participativa
3.3. El análisis documental

INTRODUCCIÓN

Entre más detallada sea la información recopilada para el desarrollo de sistemas, mejor será su diseño e implementación, para ello, es necesario contar con estrategias y formas adecuadas para recopilar dicha información.

Existen muchas herramientas que son útiles para los desarrolladores de software en el auxilio de la recopilación de datos, entre ellas encontramos formas estructuradas bien delineadas como las encuestas, cuestionarios y entrevistas, o formas sin una estructura determinada, como la observación.

A lo largo de esta unidad estudiaremos a profundidad cada una de ellas.

3.1. Planificación de la gestión de requerimientos

La gestión de requerimientos es el proceso de documentación, análisis, seguimiento, priorización y consenso de las necesidades, control del cambio y de la comunicación, con las partes interesadas. La gestión de requerimientos es un proceso continuo a lo largo de un proyecto.

La gestión de requerimientos tiene por objetivo documentar el proceso de desarrollo del sistema, sin perder de vista la satisfacción de las necesidades y expectativas de los clientes.

Dentro de las tareas de la gestión de requerimientos se producen diversos documentos que nos permiten conocer al detalle cada uno de los requerimientos funcionales o no funcionales en el desarrollo del sistema.

Algunos de los documentos obtenidos dentro del proceso de gestión de requerimientos son:

1. Diagramas de casos de uso.
2. Diagramas entidad-relación.
3. Diagramas de flujos de datos.
4. Diccionario de datos.

Diagramas de casos de uso

Los diagramas de caso de uso son documentos que describen la forma como el sistema deberá comportarse desde el punto de vista de los usuarios. Estos diagramas son la fuente principal de información que ayuda a los diseñadores de sistemas a determinar los requerimientos funcionales del sistema, en otras palabras, los diagramas de caso de uso representan las funciones que el sistema puede ejecutar.

Una de las principales ventajas de usar diagramas de casos de uso es la facilidad para ser interpretados, lo que facilita a los desarrolladores la comunicación con los clientes.

Los elementos principales que integran un diagrama de caso de uso son los siguientes:

Actores

Los actores son la representación gráfica de los diversos tipos de usuarios del sistema, concibiendo que un usuario sea toda aquella entidad externa que tiene interacción con el sistema, sean estas personas, departamentos de una empresa, otros sistemas de información, etc.

Para desarrollar de forma efectiva un diagrama de caso de uso, es recomendable independizar a los actores de acuerdo con la forma en que actúan con el sistema, por ejemplo, en un sistema de consulta, el usuario que solamente consulta la información del sistema no realiza las mismas tareas que el usuario encargado de alimentar dicha información dentro del sistema.

Los actores en un diagrama de casos de usos representan papeles que un entidad puede jugar al interactuar con el sistema, estos papeles o roles como también se les denomina,

pueden representar a más de un usuario. Volviendo al ejemplo del sistema de consulta, la forma de interactuar de dicho usuario con el sistema puede representar a más de una persona que realiza la misma acción, por ejemplo, en el caso de las páginas de Internet del gobierno federal, como la de Hacienda, existen diferentes usuarios que utilizan su portal para enviar sus declaraciones anuales, en un diagrama de caso de uso no vamos a representar a cada individuo, más bien representamos a ese conjunto de usuarios como una sola entidad que realiza una tarea genérica.

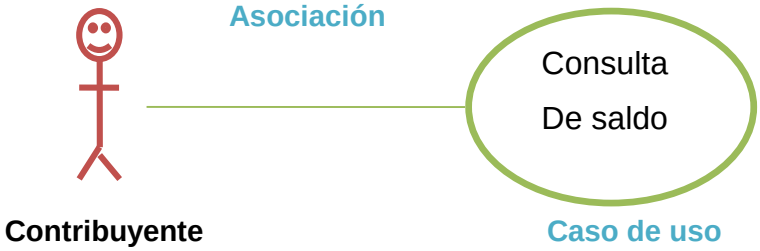
El símbolo para representar a los actores es el siguiente:

Caso de uso

El caso de uso representa de forma gráfica una tarea o función que el usuario puede realizar apoyándose del sistema por desarrollar. Los casos de uso se representan habitualmente mediante un texto que describe la actividad, encerrado en un óvalo.

Asociaciones

Las asociaciones son líneas rectas que relacionan a cada caso de uso con los actores que realizan dicha tarea. Entonces, para cada usuario existirá al menos una asociación con una acción o caso de uso, todo dependiendo de la forma como se interactúe con el sistema.



Escenarios

Los escenarios son la representación de una interacción entre un usuario y el sistema, dentro de la fase de diseño del sistema pueden surgir múltiples escenarios que describan la forma en que diferentes usuarios interactúen con el sistema.

Ejemplos

Escenario 1	**Escenario 2**
Cliente consultando su saldo en cajero automático.	Capturista ingresando datos personales para su registro en base de datos.

Durante el proceso de documentación del sistema, todos los escenarios que sean generados deben registrarse en un *diagrama* denominado *de secuencia*, que tiene por objetivo numerar e indicar la forma en que interactúan los usuarios en todas las formas posibles con el sistema.

Diagramas entidad-relación

Se basan en la percepción del mundo real, que consiste en un conjunto de objetos llamados *entidades* y las relaciones entre estas; representan la estructura lógica general de una base de datos. A continuación se describen los objetos que se utilizan en un diagrama entidad-relación (DER) así como sus tipos.

Las entidades son objetos concretos como un libro, o abstractos como un día festivo. Un ejemplo sería: Juan López.

Una entidad está representada por un conjunto de características propias o "atributos". Para cada atributo existe un rango de valores permitidos llamados *dominio del atributo;* por ejemplo, el dominio del atributo "*número de cuenta*" podría ser el conjunto de todos los enteros positivos. Así un conjunto de entidades es un grupo de entidades del mismo tipo.

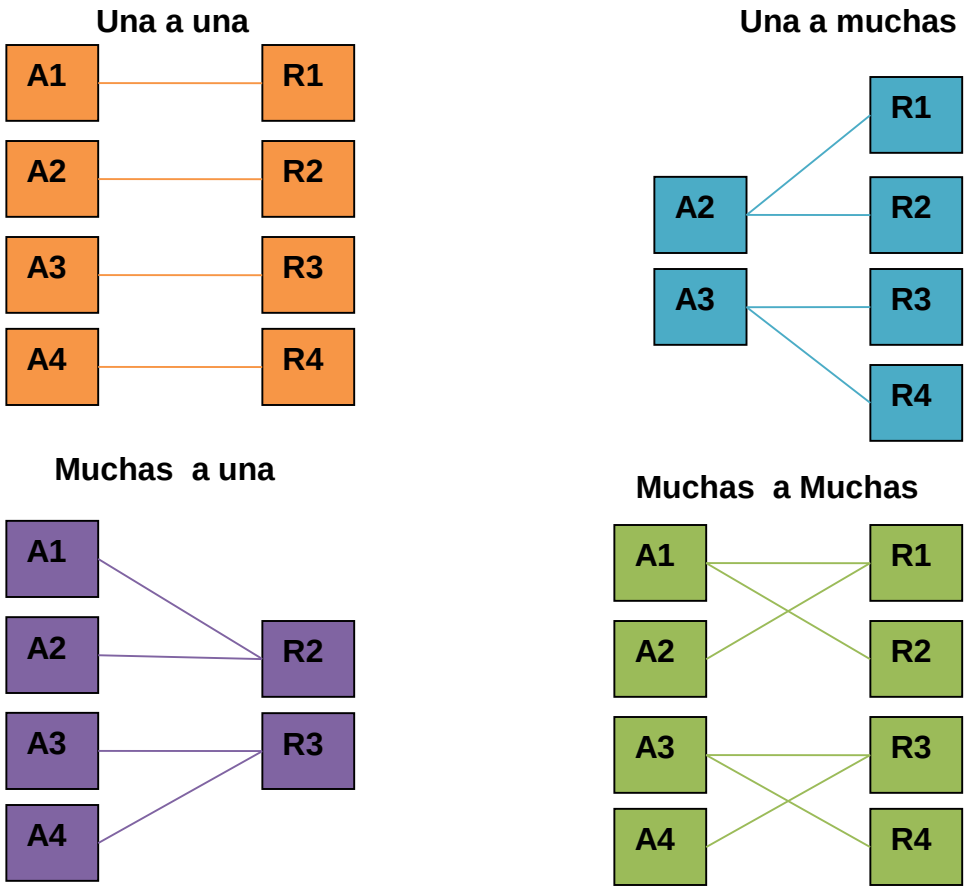
Ejemplo. En el caso de las personas que tienen cuenta en un banco:

```
Sucursal (Nombre_Sucursal, Ciudad_Sucursal, Activo)
Cliente (Nombre_Cliente, Seguro_Social, Calle)
```

Una relación es la asociación entre entidades. Por ejemplo, la asociación del cliente Juan López con número de cuenta 610.

Un conjunto de relaciones es un grupo de relaciones del mismo tipo. Por ejemplo: las relaciones cliente-cuenta, para denotar la relación cliente y cuenta.

Las *relaciones entre entidades* pueden ser de los siguientes tipos:



Tipos de relaciones

A través del diagrama entidad-relación, es posible determinar los requerimientos funcionales necesarios para la creación de la base de datos que acompañará al sistema de información en desarrollo.

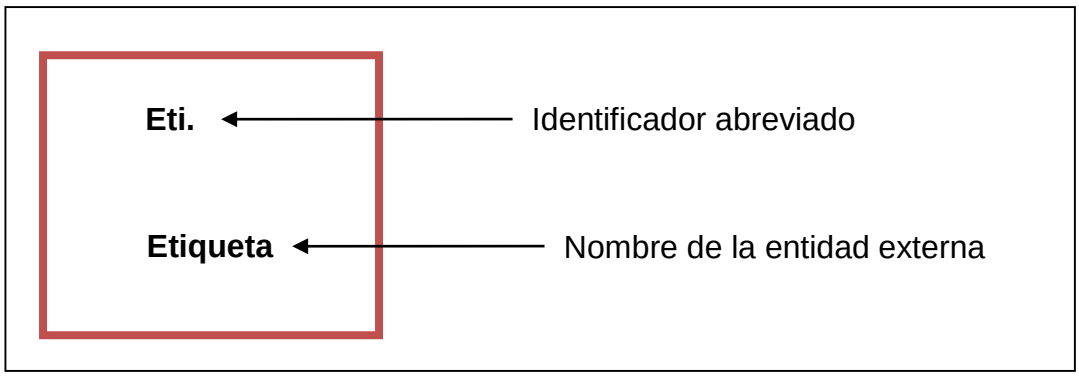
Diagramas de flujos de datos (DFD)

El diagrama de flujo de datos es la representación gráfica de los flujos de información dentro de un sistema, en otras palabras es el seguimiento del origen y destino de la información en cada proceso que integra al sistema.

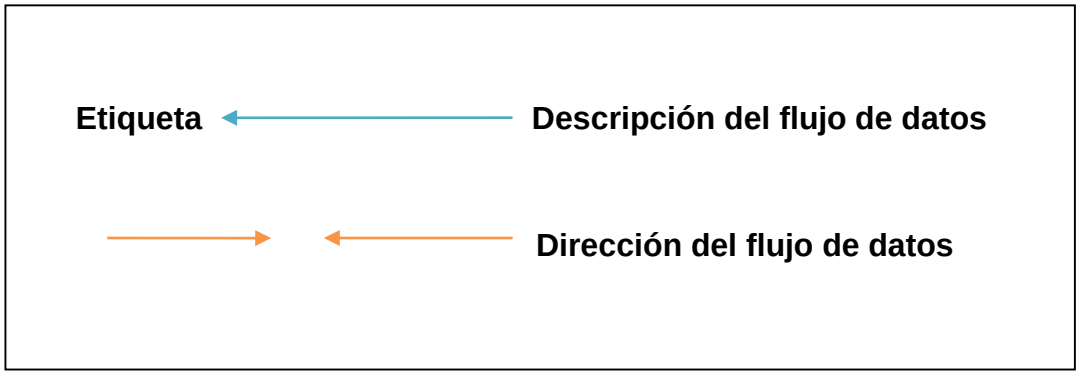
Dentro de un DFD, es posible identificar aquellos requerimientos funcionales que precisará el sistema para poder realizar de forma eficiente el manejo de la información almacenada y procesada en él.

Definamos la simbología utilizada dentro de un diagrama de flujo de datos:

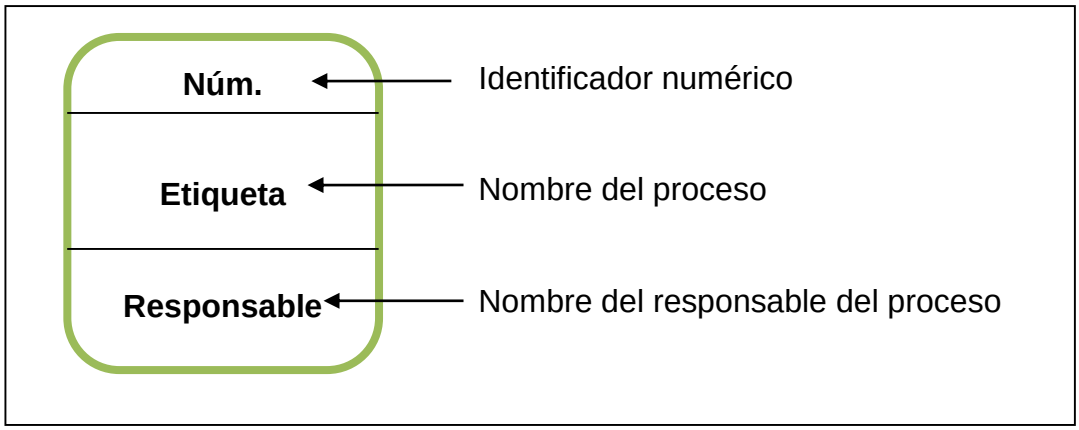
- ***Entidad externa.*** Representa los orígenes y destinos de los datos del sistema de información.



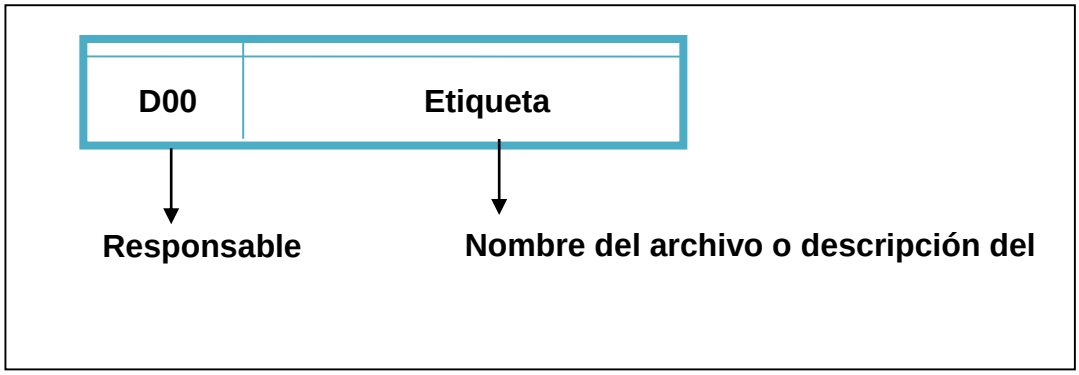
- ***Flujo de datos.*** Representa el movimiento de los datos dentro del sistema.

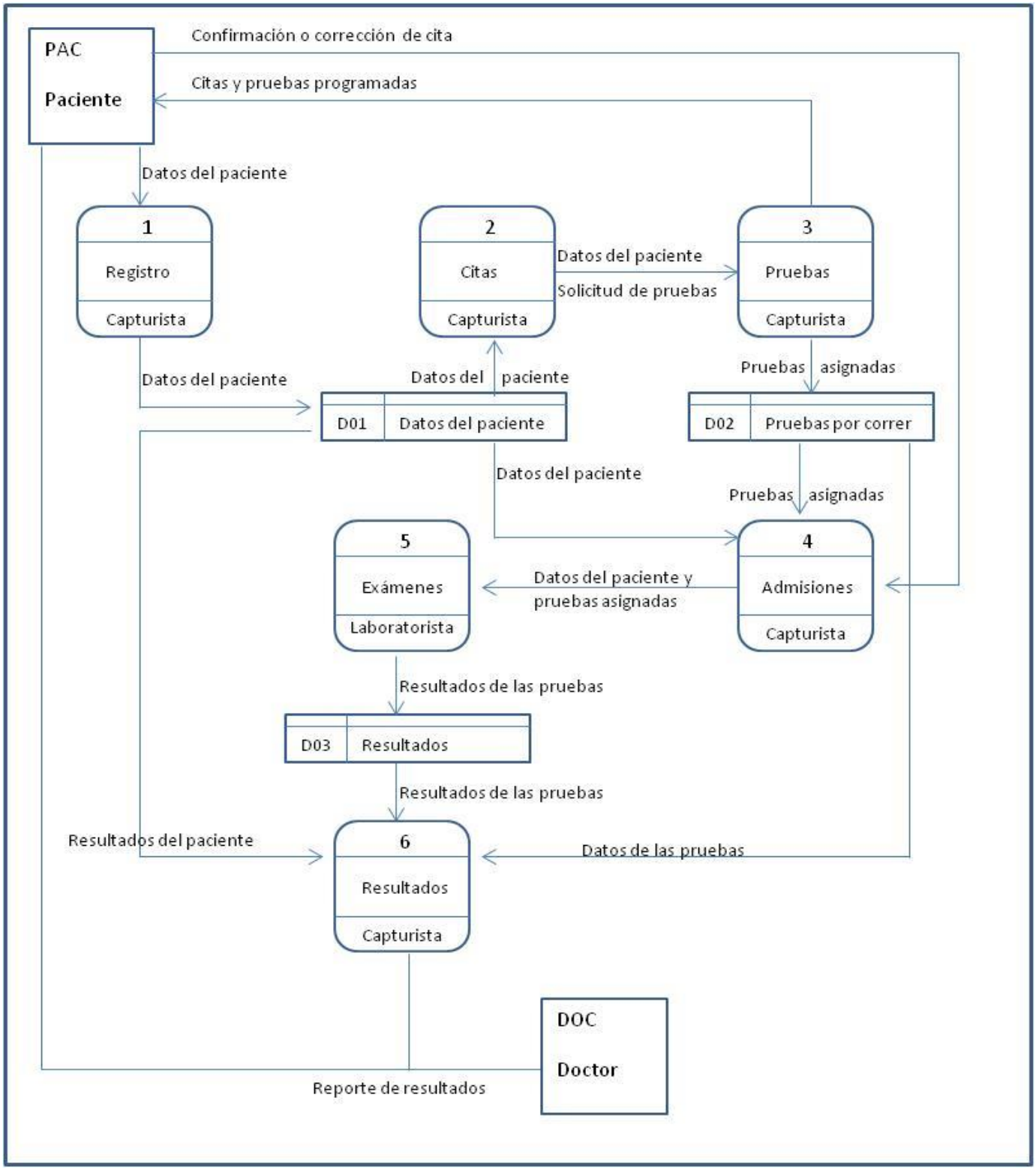


- ***Procesos.*** Representa los puntos dentro del sistema donde la información sufre modificaciones.



- ***Almacenamiento de datos.*** Es el lugar físico ~~dentro del sistema d~~onde se guarda la información (en nuestro caso la base de datos).





Ejemplo de diagrama de flujo de datos

Diccionario de datos

El diccionario de datos es un listado organizado de todos los elementos de datos que son pertinentes para el sistema con definiciones precisas y rigurosas que permiten que el usuario y el analista del sistema tengan una misma comprensión de las entradas, salidas, y de los cálculos intermedios que son realizados por el sistema.

Aunque el formato del diccionario de datos varía de autor en autor, la mayoría contiene la siguiente información:

1. *Nombre:* el nombre principal del elemento de datos o de control, del almacén de datos o de una entidad externa.
2. *Alias:* otros nombres usados para el nombre.
3. *Dónde se usa / cómo se usa:* un listado de los procesos que usan el elemento de datos o de control y cómo lo usan.
4. *Descripción del contenido:* el contenido representado mediante una notación.
5. *Información adicional:* otra información sobre los tipos de datos, los valores implícitos, las limitaciones y las restricciones, etc.

A continuación se muestra un ejemplo de la forma en que se registra una tabla en un diccionario de datos.

Tabla de Valores de referencia: Esta tabla hace referencia a las claves de los diversos perfiles o pruebas por realizarse en el laboratorio clínico.

CAMPO	LLAVE	NOMBRE	ABREVIATURA	TIPO	LONGITUD	NULO
1	Si	Clave de sección	Cve_Seccion	Char	3	N
2	Si	Clave de perfil	Cve_Perfil	Char	3	N
3	Si	Clave de la prueba	Cve_Prueba	Char	3	N
4	Si	Clave de la fase	Cve_Fase	smallint	Variable	N
5		Sexo del paciente	Sexo	Char	1	S
6		Fase de la prueba	Fase	Char	1	N
7		Rango de valores de la prueba	Rango	Varchar	25	S
8		Tipo de rango de la prueba	Tipo_Rango	Varchar	25	S
9		Descripción del tipo de rango	Descripción	Varchar	75	S
10		Valor cuantitativo 1	Vcuant1	Varchar	10	S
11		Valor cuantitativo 2	Vcuant2	Varchar	10	S

Ejemplo de una tabla definida en el diccionario de datos

El diccionario de datos ayuda a definir los requerimientos funcionales necesarios para el correcto almacenamiento de la información contenida en el sistema por desarrollar.

3.2. Métodos de recopilación de información

Una forma de poder identificar los requerimientos funcionales y no funcionales de un sistema es mediante el empleo de métodos de recopilación de información que pueden ser aplicados a los usuarios del sistema, existen diferentes métodos de recopilación de información que permiten obtener información amplia y exacta para la construcción de los sistemas. Entre estos métodos ubicamos a los estructurados y no estructurados, que incluyen la entrevista, los cuestionarios, las encuestas y la observación, principalmente.

3.2.1. Métodos estructurados

La entrevista

Las entrevistas son una forma sencilla de recabar información directa de personas o grupos, donde por lo regular, los entrevistados forman parte del grupo de usuarios del sistema por desarrollar en alguna de las 5 categorías ya comentadas.

Debido a que las entrevistas son laboriosas y lleva tiempo su aplicación, no son siempre el método más empleado entre los diseñadores, pero son una forma muy confiable de recopilación de información, ya que tiene la ventaja de ser de individuo a individuo, dando pauta para observar reacciones corporales y tonos de voz.

Las entrevistas pueden clasificarse como estructuradas o no estructuradas. Las entrevistas no estructuradas utilizan un formato pregunta-respuesta y son apropiadas cuando el analista desea adquirir información general del sistema. Este formato anima a los entrevistados a compartir sus sentimientos, ideas y creencias. Por otro lado, las entrevistas estructuradas utilizan preguntas estándar en un formato de respuesta abierta o cerrada. El primero permite que el entrevistado dé respuesta a las preguntas con sus propias palabras; el segundo utiliza un conjunto anticipado de respuestas. Cada enfoque tiene sus ventajas y desventajas (Senn).

La encuesta

Podemos definir a la encuesta como: “Acopio de datos obtenidos mediante consulta o interrogatorio, referentes a estados de opinión, costumbres, nivel económico o cualquier otro aspecto de la actividad humana” (Senn), en otras palabras, a través de las encuestas podemos ver las tendencias de opinión sobre un cierto tema en particular.

A continuación se presenta una serie de pasos por seguir para la elaboración de un buen cuestionario para encuesta.

1. Lenguaje claro y adaptado a la población que debe responder.
2. Eliminación de ambigüedades en la formulación de las preguntas, para evitar generar confusión.
3. Planteamiento de una única cuestión por pregunta.
4. Ausencia de suposiciones implícitas en los textos de las preguntas.
5. Recogida, en las preguntas, de la información más relevante sobre un tema, para evitar manipulaciones interesadas.

6. Anonimato de las respuestas, para garantizar tanto la confidencialidad de las mismas como su veracidad.
7. Formato sencillo, que facilite su complementación: diferenciación clara de cada pregunta, espacio suficiente para responder, redacción correcta, letra fácilmente legible, etc. (Senn)

El cuestionario

El uso de los cuestionarios permite a los analistas reunir información proveniente y relacionada con varios aspectos de un sistema de un grupo grande de personas.

El empleo de formatos estandarizados para las preguntas puede proporcionar datos más confiables que otras técnicas; por otra parte, su amplia distribución asegura el anonimato de los encuestados, situación que puede conducir a respuestas más honestas. Sin embargo, este método permite al analista observar las excepciones o reacciones de los encuestados. Asimismo, la respuesta puede ser limitada ya que es posible que no tenga mucha importancia para los encuestados llenar el cuestionario (Senn).

3.2.2. Métodos no estructurados

Los métodos no estructurados son métodos de recopilación de información sin una estructura previamente elaborada, sin un guión por así decirlo, como una encuesta o un cuestionario, entre los más destacados está la observación y la observación participativa.

La observación

La observación permite al analista ganar información que no se puede obtener por otras técnicas. Por medio de la observación el analista obtiene información de primera mano sobre la forma en que se efectúan las actividades. El método es más útil cuando el analista necesita observar, por un lado, la forma en que se manejan los documentos y se llevan a cabo los procesos y, por otro, si se siguen todos los pasos especificados. Los observadores experimentados saben qué buscar y cómo evaluar la significancia de lo que observan (Senn).

La observación participativa

Se caracteriza por la existencia de un conocimiento previo entre observador y observado y una permisividad en el intercambio, lo cual da lugar a una iniciativa por parte de cada uno de ellos en su interrelación con el otro. El observado puede dirigirse al observador, y el observador al observado, en una posición de mayor cercanía psicológica pero con un nivel de participación bajo o nulo.

La observación participante se refiere a una práctica que consiste en vivir entre la gente que uno estudia, llegar a conocerlos, a conocer su lenguaje y sus formas de vida a través de una intrusa y continuada interacción con ellos en la vida diaria.

El mecanismo de la observación consiste en buscar siempre una regularidad en las interacciones y una amplitud de forma continuada, manteniendo y creando relaciones.

Las normas de la observación participante son:

1. No bajar la guardia dando las cosas por supuesta.
2. Prestar atención a los aspectos culturales de la situación.
3. Tener experiencias desde dentro y desde fuera.
4. Realizar un registro sistemático de la observación. (Méndez)

3.3. El análisis documental

El análisis documental es una forma de investigación técnica, un conjunto de operaciones intelectuales, que buscan describir y representar los documentos de forma unificada sistemática para facilitar su recuperación. Comprende el procesamiento analítico - sintético que, a su vez, incluye la descripción bibliográfica y general de la fuente, la clasificación, indización, anotación, extracción, traducción y la confección de reseñas (Dulzaides y Molina, 2004).

En el caso de la recopilación de información para los sistemas de información, este tipo de análisis nos ayuda a recabar información a través de documentos ya existentes que nos ayuden en la construcción del sistema, algunos de los documentos que son susceptibles de análisis son: documentación sobre sistemas similares, manuales de la organización, manuales de procedimientos, etc.

El análisis documental se realiza en dos fases principales: el **análisis formal** y el **análisis de contenido**.

El **análisis formal** recolecta toda la información objetiva del documento que permite conocer aquellos datos que permiten diferenciar a un documento de otro, como es el caso de tipo de documento, autores, título de la obra, editoriales, fechas, idiomas, número de páginas, entre otros.

El **análisis documental** permite realizar un control e identificación de los diversos documentos en una colección de los mismos realizando dos operaciones básicas, *la catalogación* y *descripción documental*.

La *catalogación* tiene como objetivo principal el establecer un listado de todos aquellos documentos que integran a una colección, separando perfectamente los temas, autores y tipos de documentos existentes, el resultado final de la fase de catalogación es el catálogo de documentos, un instrumento que sirve de vínculo entre los usuarios de la colección y los documentos contenidos en ella.

La *descripción documental* es el proceso por el cual se describen los documentos de acuerdo con sus características internas y externas, en éste proceso se consideran atributos como el autor, el título, el lugar de edición, el editor, el año de publicación, entre otras.

La descripción documental requiere estar sujeta a normas concretas que nos permitan manejar la información de forma precisa y que facilite la clasificación y posterior búsqueda de la información dentro de la colección de documentos, las normas de catalogación más extendidas son las ISBD (*International Standard Bibliographic Description*) y las Normas de Catalogación Angloamericanas.

En el análisis de contenido, se realizan acciones que permiten llegar a la descripción del contenido o temática de un documento generando 3 resultados que permiten un mejor manejo de la información que son, **la clasificación**, la **indización** y el **resumen**. (Solis, n.d.).

"Indizar es extraer una serie de conceptos que responden a los temas tratados en el documento, y que servirán como puntos de acceso para su recuperación" (Rubio Liniers, 2004, p.5).

La clasificación es un conjunto ordenado de conceptos que se presentan distribuidos sistemáticamente en clases conformando una estructura. La creación de catálogos e índices ha sido de mucha ayuda a lo largo de la historia de la humanidad para buscar y recuperar información. Las clasificaciones más extendidas en el mundo son la CDU (Clasificación Decimal Universal), la Clasificación Decimal Dewey y la LCC (Clasificación de la Biblioteca del Congreso de Washington). (Solís)

El resumen es una representación abreviada y precisa del contenido de un documento, sin interpretación crítica y sin mención del autor del documento (ISO) (Rubio Liniers, 2004, p. 9). En otras palabras, es una visión general sintetizada del contenido completo de un documento elaborado en un lenguaje natural.

Los resúmenes ayudan a los usuarios a determinar si la información que buscan se encuentra contenida en el documento, de esta forma es posible seleccionar y descartar aquellos documentos que son útiles para el usuario.
El análisis documental implica el conocimiento del documento, el análisis, la síntesis, la representación y la recuperación.

El principio de la construcción de sistemas y de bases de datos, tiene mucho en común con el análisis documental, en los sistemas es necesario seleccionar, sintetizar, catalogar y crear índices de la información que va a ser manejada y procesada.

RESUMEN DE LA UNIDAD

Para especificar los requisitos de un sistema, necesitamos contar con determinada información que nos dará la pauta para establecer los puntos clave en la elaboración del mismo. La información que se recaba proviene principalmente de los usuarios potenciales que lo emplearán, y se obtiene a través de distintos instrumentos de recopilación de información, como lo es el cuestionario, la entrevista, las encuestas y la observación. También se puede obtener mediante la revisión y análisis de información contenida en documentos ya existentes, como lo pueden ser los manuales y la documentación generada a lo largo del proceso de desarrollo del sistema.

Es de destacar, que todo sistema de información bien diseñado debe contar con su documentación técnica, misma que puede ayudar a los analistas y desarrolladores a recabar información para definir los requerimientos funcionales y no funcionales del sistema en proceso.

BIBLIOGRAFÍA DE LA UNIDAD

SUGERIDA

Autor	Capítulo	Páginas
Bruegge (2001)	4	120-131
Pressman (2002)	1	17-25
Sommerville (2001)	1	24-35
Weitzenfield (2003)	4	67-127
Carratalà (2002)	NA	aquí
Dulzaides (2004)	NA	aquí

Unidad 4

Validación de requerimientos

OBJETIVO ESPECÍFICO

Al terminar la unidad el alumno podrá seleccionar los requerimientos que están alineados con las necesidades del negocio.

TEMARIO DETALLADO

(28 HORAS)

4. Validación de requerimientos

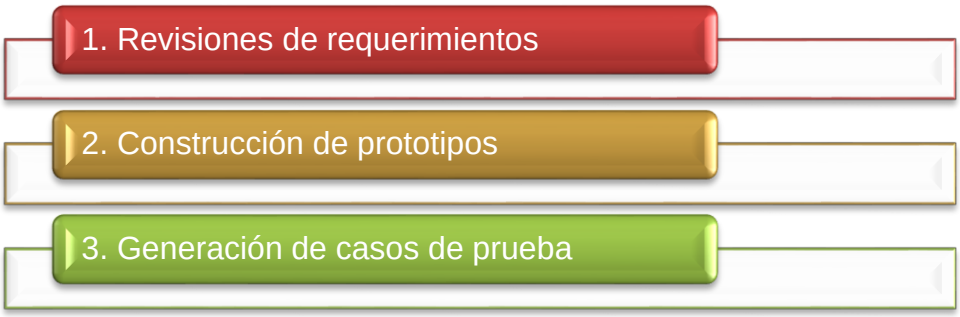
4.1. Revisión de requisitos
4.2. Prototipos
4.3. Casos de prueba

INTRODUCCIÓN

La validación de requerimientos es un proceso muy similar al de análisis, con la diferencia principal de que el objetivo de la validación es revisar y seleccionar aquellos requerimientos completos que son útiles para el desarrollo del sistema.

El proceso de validación es muy importante debido a que si existe algún error en los requerimientos, estos pueden ocasionar costos excesivos que impacten en la construcción del sistema.

Aunque puede resultar difícil demostrar que un conjunto de requerimientos es de utilidad para el usuario, sin éstos el sistema no podrá ponerse en marcha. Para poder hacer dicha demostración, será necesario validar los requerimientos, las tres formas más comunes de validación son las siguientes:



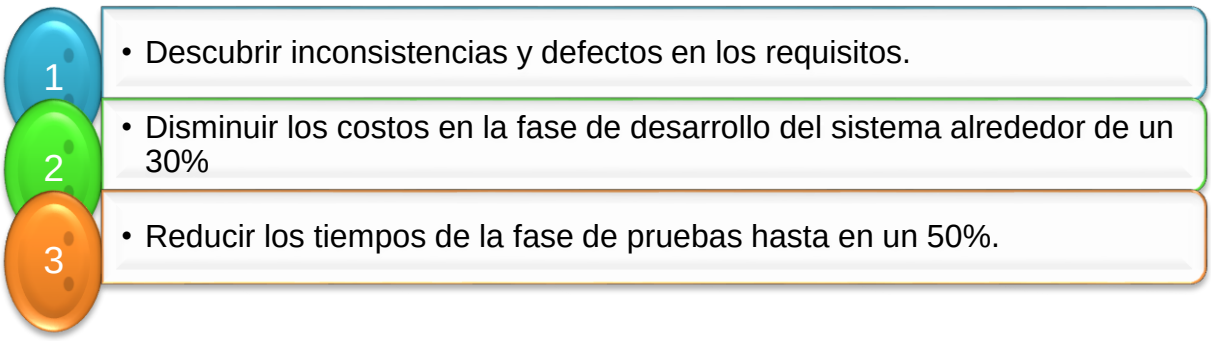
A lo largo de la unidad veremos con detalle las formas y procesos que nos permitirán validar y, con ello, seleccionar los requerimientos adecuados para la construcción de los sistemas de información.

4.1. Revisión de requisitos

Antes de poder continuar con el desarrollo de nuestro sistema de información a partir de los requerimientos adquiridos, es necesario realizar una comprobación de su validez, haciendo una comparación con las descripciones iniciales y verificando si el modelo seleccionado corresponde a lo que se ha solicitado.

La fase de validación de requerimientos suele realizarse haciendo una simulación que nos permita comprobar que el modelo de sistema seleccionado responde a la forma en que lo ha solicitado el cliente. Otra forma de validar los requerimientos, es verificar con el cliente si el modelo seleccionado es de su conveniencia. En algunos casos, dependiendo del modelo de solución empleado, será necesario construir prototipos con una funcionalidad similar que se aproxime al resultado final.

La revisión de requisitos es uno de los métodos de validación más eficiente; por medio de la técnica de revisión es posible:



El proceso de revisión de requisitos consiste principalmente en realizar reuniones planificadas entre los clientes y desarrolladores, dichas reuniones tienen el objetivo de corroborar que los requerimientos seleccionados poseen los atributos necesarios para garantizar la calidad del sistema.

Como parte de los integrantes de las reuniones, adicionalmente al cliente y el desarrollador, es deseable que se incluya a personas con experiencia en el manejo de requerimientos que sean externos al proyecto, lo anterior con la finalidad de que la selección de los mismos sea de forma imparcial.

Las reuniones de revisión siguen seis etapas principales:

1. Establecimiento del plan de revisión.

- Este paso se enfoca a establecer las tareas por realizar en la reunión, los participantes de la misma y la planificación temporal.

2. Distribución de documentos para revisión.

- El documento generado en las especificaciones de requerimientos es el centro de la revisión, pero es necesario complementar a los participantes con aquellos documentos que faciliten su comprensión y análisis.

3. Preparación de la reunión.

- Dentro de este paso se define la logística de la reunión, los tiempos y formas de realizarla, adicionalmente cada participante debe revisar de forma individual la documentación recibida para exponer las diferencias que encuentren en los diversos requerimientos.

4. Realización de la reunión.

- Esta parte es la ejecución propiamente hablando de lo planificado en el paso 3.

5. Identificación de defectos.

- Una vez realizada la reunión, antes de finalizarla, es deseable listar y especificar aquellos defectos en los requerimientos identificados en la revisión, la lista puede ser desarrollada en forma de tabla como se muestra en la siguiente tabla:

N° de requisito	Defectos detectados	Acciones recomendaciones
1	Error de estilo, que lleva a ambigüedad	Modificar el texto del requerimiento de tal forma que diga algo como “El sistema deberá permitir el registro de los fondos bibliográficos”.
2	Idem	Idem
11	Ambigüedad	Precisar la duración de las reservas
12	Concisión	No se han identificado diferencias entre profesores y alumnos a todo lo largo de la lista de requisitos. Este requisito se deberá eliminar, ya que proporciona ningún tipo de información relevante.
15	Realizabilidad	El sistema no puede realizar automáticamente los préstamos. Quizás se refiere a que debe proporcionar el máximo de automatización. Precisar, en este caso, o eliminar por irreal.
17	Concisión	Separar lo referido a libros prestados de lo referido a libros reservados.

Formato recomendado para listar defectos de requerimientos (Validación de requerimientos, UPM, 2011)

6. Correcciones de documentos.

- Durante esta fase, el desarrollador debe de evaluar y realizar los cambios que sean convenientes, resultado del análisis obtenido en la reunión y notificar de los cambios a los participantes para su aprobación.

Durante la preparación y realización de la reunión se aconseja utilizar un listado de revisión o "*requirements review checklist*", este documento es un estándar que se debe de generar a partir de la obtención de los requerimientos para revisarlos y aprobarlos en la fase de validación.

Un ejemplo del listado de revisión de requerimientos se puede descargar de la siguiente dirección electrónica:

http://www.google.com.mx/url?sa=t&rct=j&q=requirements%20review%20checklist%20&source=web&cd=1&ved=0CCMQFjAA&url=http%3A%2F%2Fvast.uccs.edu%2F~tboult%2FCS330%2FDOCS%2FRequirements%2520Review%2520Checklist.doc&ei=bv_DTt98svmxAq2Ppe4K&usg=AFQjCNEjfnkEwZ9y_ne4Nyfv3L1QVQuh5A&cad=rja [opción 2: http://bit.ly/yRWxwH]

4.2. Prototipos

Cuando se emplean prototipos como el enfoque de validación de requerimientos, es necesario construir un modelo del sistema ejecutable para ser mostrado a los clientes y los usuarios finales del mismo.

El objetivo de la validación mediante prototipos es que el usuario pueda experimentar el sistema propuesto y ver si efectivamente cumple con sus necesidades.

De acuerdo con el autor Ian Somerville (2001, p. 374), “un prototipo es una versión inicial del sistema que se utiliza para demostrar conceptos, probar opciones de diseño y, en general, informarse más del problema y sus posibles soluciones”.

El uso de prototipos es una práctica ampliamente empleada por los desarrolladores de software, a través de ellos se identifican problemas y permite implementar las soluciones hasta llegar a un producto final libre de errores.

El modelo de desarrollo de prototipos conlleva 4 etapas principales:

I. Establecimiento de los objetivos del prototipo. El establecimiento de los objetivos del prototipo por construir es muy importante, dichos objetivos deben ser perfectamente definidos desde el principio del proceso, estos objetivos pueden ser la validación de requerimientos, la creación de la interfaz de usuario para su prueba y aprobación o la viabilidad del sistema en sí.

II. Definición de la funcionalidad del prototipo. Durante ésta etapa se deben definir los aspectos por incluir y excluir, de acuerdo con los solicitados en el sistema final, lo anterior se realiza con la finalidad de reducir costos y afinar detalles de funcionalidad del sistema final.

III. Desarrollo del prototipo. Una vez seleccionados los aspectos por incluir en el prototipo, se procede a su programación para su presentación a los usuarios.

IV. Evaluación del prototipo. Una vez presentado el prototipo a los usuarios para su prueba, los desarrolladores deben enfocarse en observar si el prototipo cumple con los objetivos planteados en la primera etapa. Los objetivos planteados deben ayudar al desarrollador a establecer un plan de evaluación. En dicho plan hay que considerar el acercamiento constante de los usuarios con el sistema prototipo, a fin de que se acostumbren a él e interactúen libremente, dicha interacción tendrá la finalidad de descubrir errores de funcionalidad que nos llevarán a descubrir errores u omisiones en los requerimientos del sistema.

Es necesario mencionar que no todos los prototipos de sistemas son necesariamente programables, algunos tipos de prototipos que pueden ser empleados, son:

Mock-Ups

Son pantallas dibujadas, generalmente a mano en papel, que representan un aspecto en concreto del sistema. Estos dibujos son empleados principalmente en la definición de las características que tendrá la interfaz del sistema.

Storyboards

Se pueden considerar como el paso siguiente de los “Mock-Ups”, los storyboards presentan, adicionalmente de la parte visible del sistema, una secuencia de acciones o escenarios que el sistema debe contener. Lo recomendado en el uso de storyboards es diseñar pantallas individuales que contemplen la mayor parte posible de las acciones que los usuarios pueden realizar en ellas, además de las posibles respuestas del sistema.

Maquetas

Consisten en una forma sintetizada del sistema deseado con algo de interacción básica, generalmente las maquetas representan a las pantallas que serán empleadas como la interfaz de usuario y una funcionalidad limitada, como el paso de una pantalla a través de botones y datos generalmente fijos que representan los resultados de las diversas acciones que pueden ser realizadas en dichas pantallas. Si se desea agregar un poco de funcionalidad a la maqueta es posible realizar un poco de programación a la misma para un resultado más realista.

Las maquetas son generalmente desarrolladas bajo algún lenguaje de programación gráfico como Visual Basic o Power Builder.

Al igual que en la fase de revisión de requerimientos, es deseable emplear el documento de listado para registrar los problemas y observaciones encontrados en los requerimientos y realizar sus cambios posteriormente.

4.3. Casos de prueba

La generación de casos de pruebas es un enfoque de validación de requerimientos que pretende probar los requerimientos seleccionados y, a través de ellas, revelar problemas en los requerimientos que puedan impactar en el desarrollo del sistema final.

Para la validación de requerimientos por medio de casos de pruebas, se deben diseñar situaciones que permitan evaluar el desempeño del requerimiento, si en un caso dado, un requerimiento no puede ser puesto a prueba, esto significa que tampoco podrá ser implementado en el sistema, por lo que debe re-evaluarse.

Un caso de prueba es una acción perfectamente definida que el sistema debe de ser capaz de realizar (Validación de requisitos, UPM, 2011), en otras palabras, se puede decir que un caso de prueba es un escenario que debe contemplar con detalle todos aquellos datos de entrada que van a ser procesados, las tareas por realizar con dichos datos y los resultados que se desea obtener en dicho escenario. El enfoque de desarrollo de software que utiliza de forma consistente los casos de prueba es la "*programación extrema".*[2] Dentro de dicha metodología, las pruebas tienen las siguientes características:

☑ **Desarrollo previamente probado**. Este concepto se refiere a que en el momento de escribir una prueba, se definirá de forma implícita tanto la interfaz por utilizar como el requerimiento funcional por evaluar en el funcionamiento del sistema. La intención del desarrollo previamente probado es establecer con claridad las tareas por realizar con el requerimiento definido y, a través de dichas tareas, evaluar su validez.

☑ **Desarrollo de pruebas incrementales a partir de escenarios**. En este enfoque, el desarrollador, aparte de definir los recursos, indica una secuencia de acciones o historias por seguir. Dentro de esta metodología, el usuario puntualiza cuáles son prioritarias para él, y el desarrollador puede precisar tareas específicas que definen las características del sistema.

☑ **Participación del usuario en el desarrollo de pruebas y de la validación**. Este enfoque pretende involucrar a los usuarios finales en el desarrollo y planificación de la pruebas que serán implementadas en la siguiente entrega del sistema o el siguiente prototipo, de esta forma es posible verificar que el sistema cumple con sus necesidades reales y también nos ayuda a verificar y validar los requerimientos utilizados en cada escenario presentado.

[2] La Programación Extrema (PX), mejor conocida por su nombre en inglés *Extreme Programming* (XP), es una de las llamadas *Metodologías Ágiles* de desarrollo de software más exitosas de los tiempos recientes. Fuente: Programación extrema, www.programacionextrema.org

☑ **Banco de pruebas automatizado**. Consiste en diseñar un programa de pruebas que interactúe con el sistema antes de implementar una tarea en el mismo. Una vez diseñadas, definidas y planificadas las pruebas por realizar, el desarrollador las puede implementar en el programa de pruebas para su ejecución de forma automática; cabe recalcar que la aplicación donde se programen las pruebas debe ser totalmente independiente del sistema en desarrollo. Este programa debe ser capaz de simular el envió de los datos de entrada y de verificar que los datos de salida cumplan con las especificaciones deseadas.

Al igual que en los métodos de validación vistos con anterioridad, en el uso de casos de pruebas deben generarse listados de aquellos requerimientos que presenten problemas para su corrección o eliminación.

Prueba 4. Prueba de la validez de la tarjeta de crédito
Entrada: Una cadena que representa el número de tarjeta de crédito y dos enteros que representan el mes y el año de la caducidad de la tarjeta. Pruebas: • Comprobar que todos los bytes de la cadena son dígitos. • Comprobar que el mes se encuentra entre 1 y 12 y que el año es mayor o igual que el año actual. Utilizando los 4 primeros dígitos del número de tarjeta de crédito, comprobar que el emisor de la tarjeta es válido consultando la tabla de emisores de tarjetas. Comprobar la validez de la tarjeta de crédito enviando el número de tarjeta y la fecha que caduca el emisor de la tarjeta. Salida: OK o un mensaje de error indicando que la tarjeta no es válida.

Ejemplo de un caso de prueba (Somerville, 2001, p. 368)

RESUMEN DE LA UNIDAD

La validación de requerimientos es una fase crucial en el desarrollo de los sistemas de información. Desde el inicio del análisis del sistema se recaban requerimientos que, a la postre, deben probarse y validarse para poder determinar si son o no de utilidad.

El objetivo de la validación de requerimientos es la detección de aquellos requerimientos que son incorrectos, que presentan inconsistencias o errores, y que son de difícil implementación, lo anterior será registrado en un documento en forma de lista que permita a los desarrolladores identificar y corregir dichos requerimientos.

Las técnicas más comunes para la validación de requerimientos van desde la revisión detallada de la documentación asociada al diseño del sistema, hasta la implementación de prototipos o pruebas funcionales que permitan evaluar el desempeño de dichos requerimientos al ser utilizados en tareas específicas.

La validación de requerimientos es un proceso obligado en la ingeniería de software, que posteriormente se verá recompensada con la disminución de costos y tiempos en la construcción de los sistemas de información.

BIBLIOGRAFÍA DE LA UNIDAD

SUGERIDA

Autor	Capítulo	Páginas
Pressman (2002)	2	21
	5	79-80
	10	171-174
Sommerville (2001)	7	129-150
	17	355-373

REFERENCIA BIBLIOGRÁFICA

BÁSICA

JOYANES, L. (2003). *Fundamentos de programación Algoritmos Estructuras de datos y objetos*. (3ª ed.) Madrid: McGraw-Hill.

PFLEEGER, S.L. (2002). *Ingeniería de software, teoría y práctica.* México: Prentice Hall.

PIATTINI VELTHUS, Mario y García Rubio, Félix Oscar. (coords.) (2003) *Calidad en el desarrollo y mantenimiento de software.* México: Alfa omega-Ra-Ma. [Vista previa]

PIATTINI VELTHUS, Mario; Calvo Manzano Villalón, José; Cervera Bravo, Joaquín. (2003a). *Análisis y diseño de aplicaciones informáticas de gestión: Una perspectiva de ingeniería del software.* México: Alfa Omega-Rama. [Vista previa]

COMPLEMENTARIA

BROWN, David W. (1997). *Object-Oriented Analysis.* (Edición ilustrada) Hoboken, NJ: John Wiley & Sons.

DENNIS, Alan; Wixon, Barbara H. (2000). *Systems Analysis and Design: An applied approach.* (Edición abreviada) Hoboken, NJ: John Wiley & Sons.

FIGUEROA, Pablo. (1997). Definición de caso de uso. *Elementos notacionales de UML*, University of Alberta, disponible en línea: http://webdocs.cs.ualberta.ca/~pfiguero/soo/uml/casos_uso01.html, consultado el 20/03/11.

FLORES LARA, J. A. (n.d.) "Determinación de requerimientos del sistema", Mitecnológico, disponible en línea: http://www.mitecnologico.com/Main/DeterminacionRequerimientosSistema, recuperado el 19/03/11

----------. "Requerimientos de los usuarios", Mitecnológico, disponible en línea: http://www.mitecnologico.com/Main/RequerimientosDeLosUsuarios, recuperado el 19/03/11.

GARCÍA GUTIÉRREZ AL. Tratamiento y análisis de la documentación. En: Vizcaya Alonso, D. (comp). Selección de lecturas: Fundamentos de la organización de la información. La Habana: Universidad de La Habana, 2002. En Dulzaides Iglesias, Mª. Elinor y Molina Gómez, Ana Mª. (2004) *Análisis documental y de información: dos componentes del mismo proceso.* Centro de Información de ciencias médicas, Cienfuegos, Cuba. http://bvs.sld.cu/revistas/aci/vol12_2_04/aci11204.htm. 24/11/11

INCE, Darrel. (1993). *Ingeniería de Software*. México: Addison-Wesley. [Vista previa]

KENDALL, K. y Kendall J. (1990). *Análisis de diseño de sistemas*. México: Prentice Hall. [Vista previa]

LARMAN CRAIG. (1999) *UML y patrones: introducción al análisis y diseño orientado a objetos*. México: Prentice-Hall.

LÓPEZ GORZMAN, C. (2004) "¿Cómo determinar los requerimientos del usuario?", Liceo Industrial y de Minas Ignacio Domeyko, Chile, disponible en línea: http://www.corporacionminera.cl/html/plangral/1ros/como.pdf, consultado el 14/10/11

MÁRQUEZ VITE, J.M. (2002). *Sistemas de información por computadora, metodología de desarrollo*. México: Trillas.

MÉNDEZ, Reny. (s/f). La observación participativa partes 1 y 2, Monografías, disponible http://www.monografias.com/trabajos65/observacion-participante/observacion-participante.shtml recuperado el 04/11/11

MEYER, B. (1999). *Construcción de Software Orientado a Objetos*. Madrid: Prentice-

---------- (1998) *Análisis de requerimientos*. Versión 1.1., University of Alberta, disponible en línea: http://webdocs.cs.ualberta.ca/~pfiguero/soo/metod/requerimientos.html, consultado el 20/03/11.

RUBIO LINIERS, M. C. (2004) *El análisis documental: Indización y resumen en base de datos especializados*. CINDOC CSIC, Madrid, disponible en: http://www.iberius.net/es/AisManager?Action=ViewDoc&Location=getdocs:///DocMapCSDOCS.dPortal/2519, recuperado el 20/11/11

SENN, James A. (n.d.), Análisis en *Análisis y diseño de sistemas de información*, disponible en línea: http://une-senn.tripod.com/new_page_1.htm#Herramientas_para_determinar_requerimientos_de_sistemas, consultado el 20/03/11.

SOLÍS HERNÁNDEZ, Isabel A. (n.d.) "El análisis documental como eslabón fundamental para la eficiencia de servicios de información", Monografías, disponible en línea: http://www.monografias.com/trabajos14/analisisdocum/analisisdocum.shtml, recuperado el 4/11/2011.

BIBLIOGRAFÍA ELECTRÓNICA

(Nota: todos los enlaces, consultados o recuperados, funcionan al 30/05/14[dd/mm/aa])

Libros		
Fuente	**Capítulos (s) Unidad (es) que soporta**	**Liga**
Diccionario de informática: Definición de requerimiento. Consultado el 19 de marzo del 2011.	(U1, U2, U3 y U4)	http://www.alegsa.com.ar/Dic/requerimientos.php
Diagramas de casos de usos por Jesús Cáceres Tello. Universidad de Alcalá de Henares, España.	(U, 3)	http://www2.uah.es/jcaceres/uploaded/capsulas/DiagramaCasosDeUso.pdf
Documento de listado para la revisión de requerimientos.	(U, 4)	http://www.google.com.mx/url?sa=t&rct=j&q=requirements%20review%20checklist%20&source=web&cd=1&ved=0CCMQFjAA&url=http%3A%2F%2Fvast.uccs.edu%2F~tboult%2FCS330%2FDOCS%2FRequirements%2520Review%2520Checklist.doc&ei=bv_DTt98svmxAg2Ppe4K&usg=AFQjCNEjfnkEwZ9y_ne4Nyfv3L1QVQuh5A&cad=rja

El análisis documental como eslabón fundamental para la eficiencia de servicios de información.	(U,3)	http://www.monografias.com/trabajos14/analisisdocum/analisisdocum.shtml
Gracia, Joaquín. (2003) UML Casos de uso, (27/09/03), Ingeniero software.	(U,2)	http://www.ingenierosoftware.com/analisisydiseno/casosdeuso.php
Procesos de la ingeniería de requerimientos. Mena Mendoza, Gonzalo.	(U,2)	http://mena.com.mx/gonzalo/maestria/ingreq/presenta/procesos_ir/
Requerimientos del software.	(U,2)	http://requerimientos.galeon.com/
Tutorial de casos de uso. Universidad de Chile. Patricio Salinas C.	(U2 y U3)	http://www.dcc.uchile.cl/~psalinas/uml/casosuso.html
Validación de requisitos - Maestría en Ingeniería de Software. Universidad Politécnica de Madrid (UPM).	(U,4)	http://is.ls.fi.upm.es/docencia/masterTI/ARS/docs/Manual_M2C1U11.pdf

Facultad de Contaduría y Administración
Sistema Universidad Abierta y Educación a Distancia