

ESPECIFICACION DE REQUERIMIENTOS DE SOFTWARE

INGENIERIA DE REQUERIMIENTOS

Mg. Marcela Daniele

Departamento de Computación

Facultad de Ciencias Exactas, Físico-Químicas y Naturales

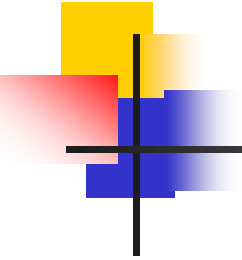
Universidad Nacional de Río Cuarto





Requerimientos de Software

- **Introducción, definición y conceptos**
- **Clasificación**
- **Ingeniería de Requerimientos**
- **Proceso de Requerimientos**
 - * **Análisis del Problema**
 - * **Descripción del Producto**
 - * **Validación**
- **Notación**
- **Documentación de la Especificación de Requerimientos de Software (SRS)**



**“Fallamos más a menudo por
resolver el problema incorrecto,
que por obtener una solución inadecuada
del problema correcto”**

Ackoff, R., Redesigning the Future. John Wiley & Sons (1974).

**“No tiene sentido ser preciso si no se
sabe de qué se está hablando”**
(Von Neumann)



Decidir **Qué** construir

Decidir precisamente **QUE CONSTRUIR**
es una de las tareas más importante en la
construcción de sistemas de software.

- La complejidad y el tamaño de los sistemas de software aumenta o se modifica continuamente

Surgen constantemente cambios, y ello conlleva a que aparezcan nuevos problemas que no existían cuando el sistema era más pequeño.

- Se necesita ser más rigurosos para el análisis de los requerimientos del software

*Se establecen los **requerimientos técnicos detallados**, incluyendo interfaces con humanos, máquinas y otros sistemas software.*

Descubrir y Analizar los Requerimientos



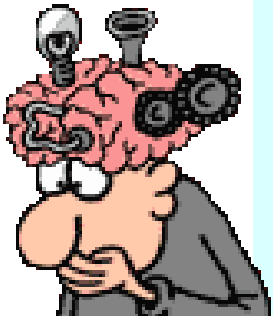
Descubrir y Analizar los Requerimientos del software es una actividad muy difícil y propensa a errores.

En la **Especificación de Requerimientos de Software** se traducen las ideas de la mente del cliente (**entrada**) a un documento formal (**salida**).

- ✓ La **entrada** es informal e imprecisa, y generalmente incompleta.
- ✓ La **salida** es un conjunto de requisitos bien definido, y se espera sea completa y precisa.

Descubrir y Analizar los Requerimientos

- La etapa de Especificación de Requisitos de Software no puede ser totalmente automatizada.
- Los métodos usados para la identificación de requisitos representan un conjunto de directrices o normas a seguir. Las metodologías o procesos de desarrollo de software incluyen y proponen técnicas para identificar y definir los requerimientos que deberá incluir el sistema.



La especificación de requerimientos **realizada en forma errónea, perjudica** significativamente el resultado final. Esta etapa del desarrollo es la que presenta mayores **dificultades para rectificar en etapas posteriores.**



Descubrir y Analizar los Requerimientos

Causas que podrían producir una **errónea conceptualización del problema** pueden ser:

- ✓ Dominio de problema demasiado complejo y/o escaso conocimiento del mismo
- ✓ Insuficiente formación del equipo asignado a ésta tarea
- ✓ Carencia de experiencia para identificar requerimientos y conceptualizarlos
- ✓ Escasa analogía con sistemas anteriores
- ✓ Un incorrecta planificación de los tiempos dedicados a esta tarea

Requerimientos del Software

Los **requerimientos de un sistema de software** son una descripción detallada de los servicios que el sistema provee bajo ciertas restricciones operativas.

IEEE define un requisito como:

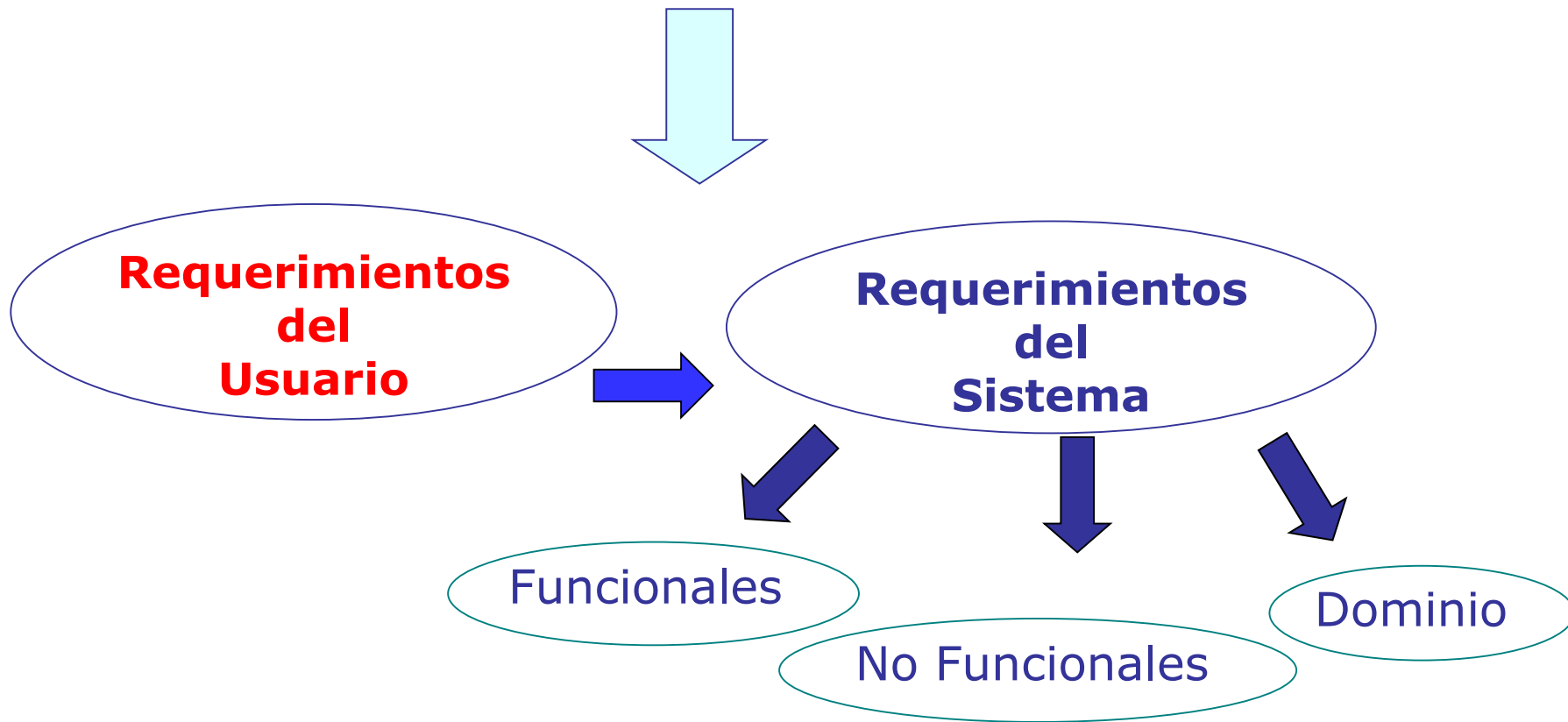
“Condición o capacidad que debe cumplir o poseer un sistema para satisfacer un contrato, un estándar, una especificación, o cualquier otro documento oficial establecido”.



Teniendo en cuenta que los requisitos de software que se están definiendo, son los que deberá satisfacer el sistema que se comienza a desarrollar.....

Requerimientos del Software

Clasificación según su nivel de abstracción





Requerimientos del Usuario

- El solicitante del sistema presenta una **descripción detallada**, generalmente **en lenguaje natural**, de los servicios que espera que le provea el sistema.
- El objetivo del solicitante es **mejorar u optimizar los procesos de negocio existentes**.

Problemas con el lenguaje natural: falta de claridad, confusión de requerimientos y conjunción de requerimientos.

Recomendaciones:

- Usar algún formato estándar y asegurar que todos los req. se adecuan.
- Distinguir entre *requerimientos* **obligatorios** y **deseables**.
- Resaltar el texto para distinguir las partes clave del requerimiento.
- Evitar el uso de términos muy técnicos o jerga informática.



Requerimientos del Sistema

Se describen de manera detallada, las **funciones** y **servicios** que brindará el sistema, con las correspondientes restricciones operativas.

Requerimientos Funcionales

- Declaran los **servicios** que proporciona el sistema y su **comportamiento** en los diferentes escenarios.
- Pueden estar descriptos en **diferentes niveles de abstracción**.

Todos los servicios solicitados por el usuario deben ser contemplados por el sistema

Los requerimientos no deben tener definiciones contradictorias

requerimientos funcionales sean planteados de manera **completa y consistente**.



Requerimientos del Sistema

Requerimientos No Funcionales

- Definen **restricciones** sobre los requerimientos funcionales o sobre el sistema total.
(Ej: tiempo de respuesta, seguridad, fiabilidad y capacidad de almacenamiento)
- Existen requerimientos No Funcionales **aplicables al proceso de desarrollo** del software.
(Ej: especificar los estándares de calidad a seguir por el proceso)
- Suelen ser requerimientos **críticos**.
(Ej: En un sistema de control de vuelos es indispensable que el sistema cumpla con los requerimientos de confiabilidad)
- **Pueden surgir** a partir de necesidades del usuario, restricciones de presupuesto, políticas de organización, necesidad de interoperabilidad, otros factores externos.

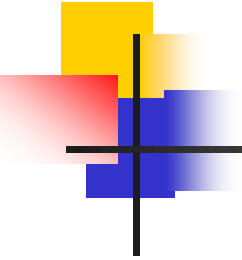


Requerimientos del Sistema

Requerimientos de Dominio

- Derivan del **dominio específico** de aplicación del sistema.
- Pueden representar **nuevos requerimientos funcionales** o **restringir los existentes**.

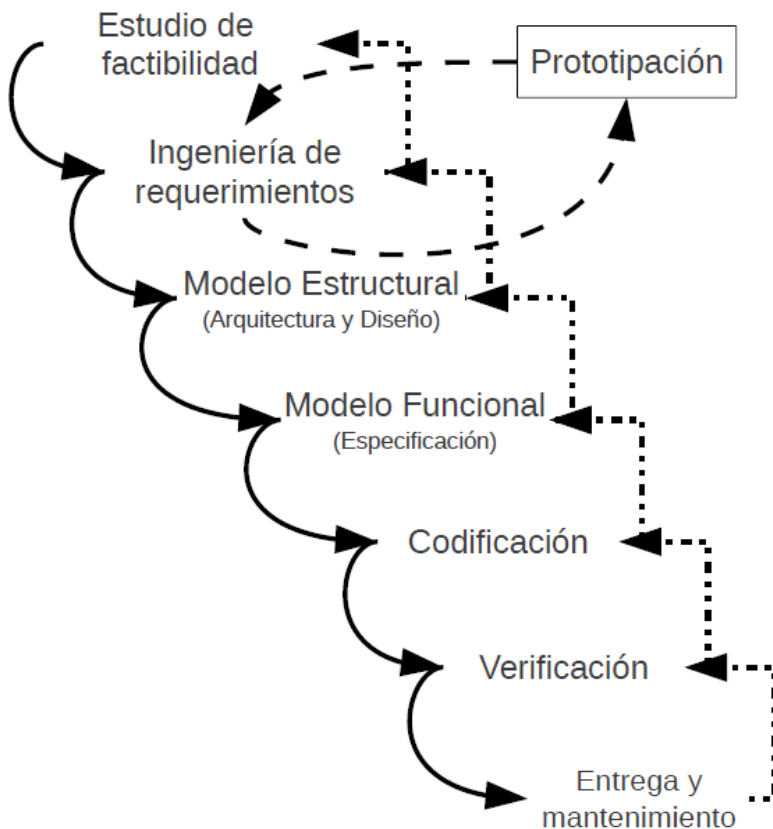
Ejemplo de un requerimiento de dominio en un sistema de control de vuelos: “Cuando el avión llega a una cierta altura el sistema informa automáticamente a la central de control”.



Ingeniería de Requerimientos

Ingeniería de Requerimientos

La Ingeniería de Requerimientos se encarga de la identificación del propósito de un sistema de software y el contexto en el cual será usado.

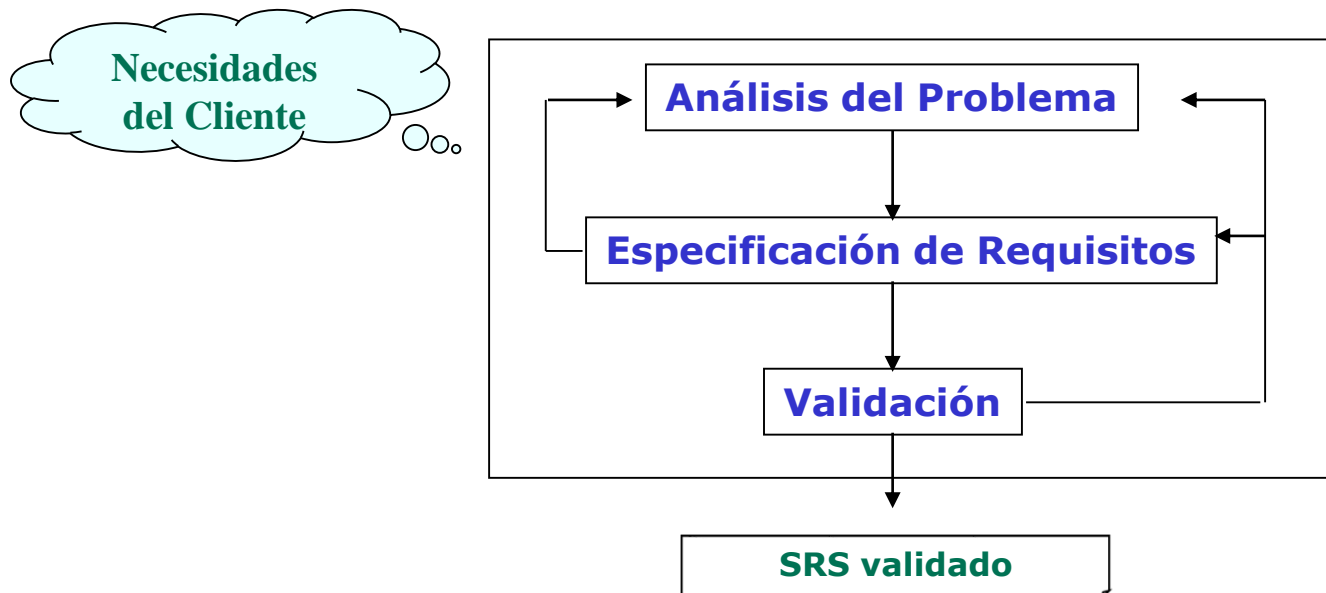


La Ingeniería de Requerimientos es la etapa que le sigue, al estudio de factibilidad de llevar adelante el desarrollo del sistema de software.

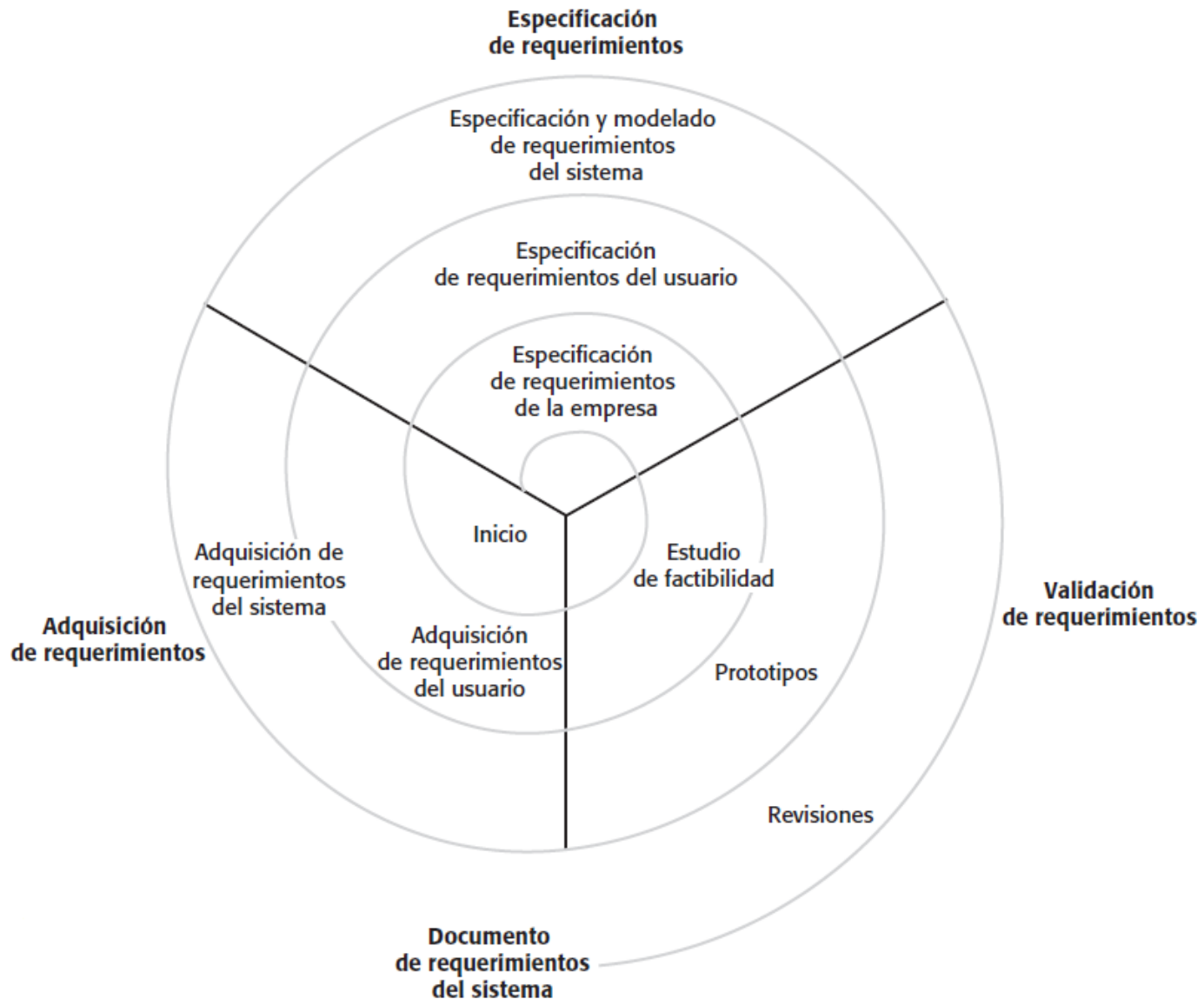
Proceso de Ingeniería de Requerimientos

El proceso de ingeniería de requerimientos es la secuencia de actividades que se desarrollan en la fase de requisitos y que culminan en la producción de un documento de alta calidad que contiene la especificación completa de los requerimientos de software (SRS).

El Proceso de IR está compuesto por tres tareas básicas:



Proceso de Ingeniería de Requerimientos





Proceso de Ingeniería de Requerimientos

ANALISIS DEL PROBLEMA

Se modela el dominio del problema para comprender el comportamiento del sistema, limitaciones, entradas-salidas. El objetivo principal es obtener un conocimiento profundo del problema a resolver.

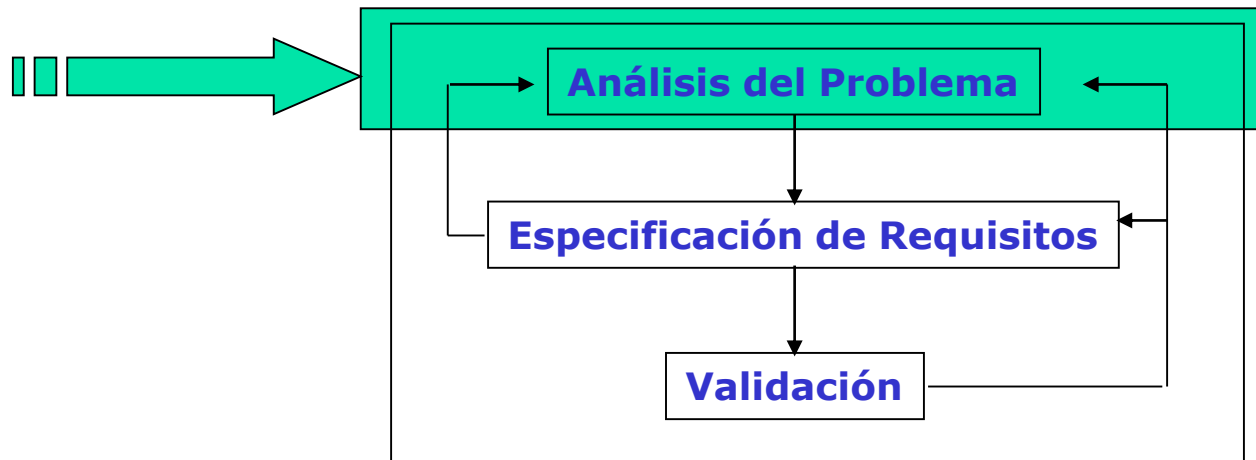
ESPECIFICACION DE REQUISITOS (DESCRIPCION DEL PRODUCTO)

El análisis produce mucha información y con posibles redundancias, es necesario organizar la descripción de los requisitos.

VALIDACION

La validación debe asegurar que el documento SRS es de alta calidad y especifica **todos** los requisitos del software.

Proceso de Ingeniería de Requerimientos





Análisis del Problema

El principio básico del análisis es “*Divide y Vencerás*”

Particionar el problema en subproblemas, entender cada subproblema y su relación con otros, y de esa manera entender el problema total.

**El objetivo central del Análisis es
comprender el dominio del problema**

El analista debe estar familiarizado con los diferentes métodos de análisis y selección del que mejor se ajuste al problema a resolver.

- **Enfoque Informal**
- **Modelado de Flujo de Datos -Método de Análisis Estructurado**
- **Modelado Orientado a Objetos. Modelado Dirigido al Modelo**



Análisis del Problema: Enfoque Informal

- No posee una metodología definida.
- El problema y los modelos del sistema están esencialmente **en la mente de los analistas del sistema**.
- El analista tendrá reuniones con los clientes y usuarios finales
 - Inicialmente, clientes y usuarios finales explicarán al **analista** sobre su trabajo, su ambiente y sus necesidades.
 - El **analista** documenta el resultado de las reuniones y elabora una propuesta para el sistema que brinda una posible solución al problema planteado.
 - El **analista** explica al cliente las funcionalidades del sistema, y utiliza las reuniones para verificar si el sistema que propone es coherente con los objetivos del cliente.



Análisis del Problema: Método de Análisis Estructurado

El sistema es visto como funciones que operan en un entorno y transforman entradas para producir salidas.

- Cuando la función de transformación global del sistema es demasiado compleja, se divide en subfunciones.
 - La subdivisión se produce hasta que cada función sea comprendida fácilmente.
 - Se realiza un seguimiento de los datos que fluyen desde sus entradas hasta sus salidas.
- Los Diagramas de Flujo de Datos (DFDs) son la notación usada en este enfoque
 - Se comienza con un DFD a un alto nivel de abstracción llamado *diagrama de contexto*, que contiene las principales entradas y salidas de los procesos actuales.
 - Luego, este DFD es refinado con una descripción de sus diferentes partes, mostrando más detalles.



Diagrama de Flujo de Datos (DFD)

- Un DFD es una descripción abstracta de un sistema, independientemente de si el sistema está automatizado o es manual.
- Si se desean más detalles, el DFD puede ser refinado.
- Un DFD representa un flujo de datos, y no un flujo de control.
- Al elaborar el DFD, el diseñador tiene que especificar las principales transformaciones en el flujo de los datos derivados de la entrada a la salida.
- Muchos sistemas son demasiado grandes para un único DFD. Es necesario usar algunos mecanismos de descomposición y abstracción.
- Los DFD son muy utilizados en las metodologías de Análisis Estructurado, que ven a cada sistema como funciones que operan en un entorno y transforman entradas del mismo para producir salidas al mismo.

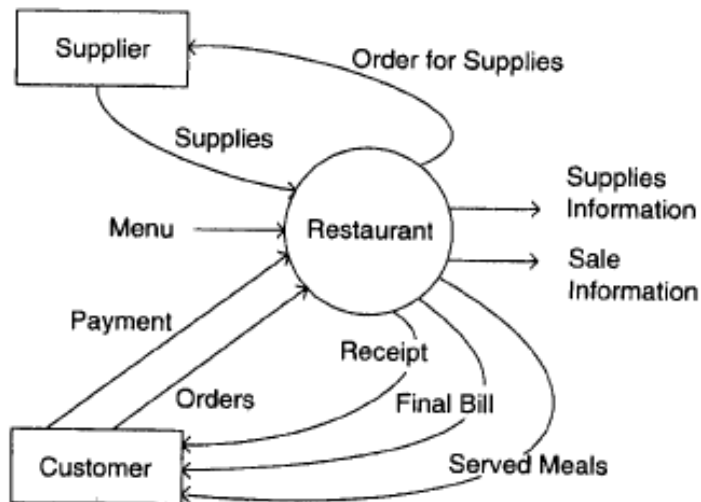
Análisis del Problema: Método de Análisis Estructurado

EJEMPLO DE UN DFD:

El propietario de un restaurante quiere automatizar algunas tareas para hacer su negocio más eficiente.

Cliente del sistema: Propietario. **Usuarios potenciales:** Propietario. Camareros, Cajero.

DFD: Diagrama de Contexto



DFD de Sistema





Análisis del Problema: Modelado Orientado a Objetos

El sistema es visto como un conjunto de objetos.
Los objetos interactúan unos con otros a través de los servicios que prestan.

El objetivo del modelado OO es construir un modelo global que soporte los servicios deseados por el cliente:

- identificando los objetos que existen en el dominio del problema,
- especificando la información que encapsulan y los servicios que prestan,
- identificando las relaciones que existen entre dichos objetos.

Diagramas de UML → notación más usada en este enfoque.



Análisis del Problema: Modelado Orientado a Objetos

Las principales consideraciones a tener en cuenta para realizar el Análisis del Problema son:

1. Identificación de objetos y clases

Considerar como **objetos** aquellos que representan entidades en el espacio del problema

2. Identificación de estructuras

Jerarquías entre clases. Capturar relaciones de generalización-especialización, todo-parte

3. Identificación de atributos

Los atributos agregan propiedades de las clases, dependiendo del problema que se está modelando. Establecer los atributos de cada clase que defina el dominio del problema

4. Identificación de relaciones/asociaciones

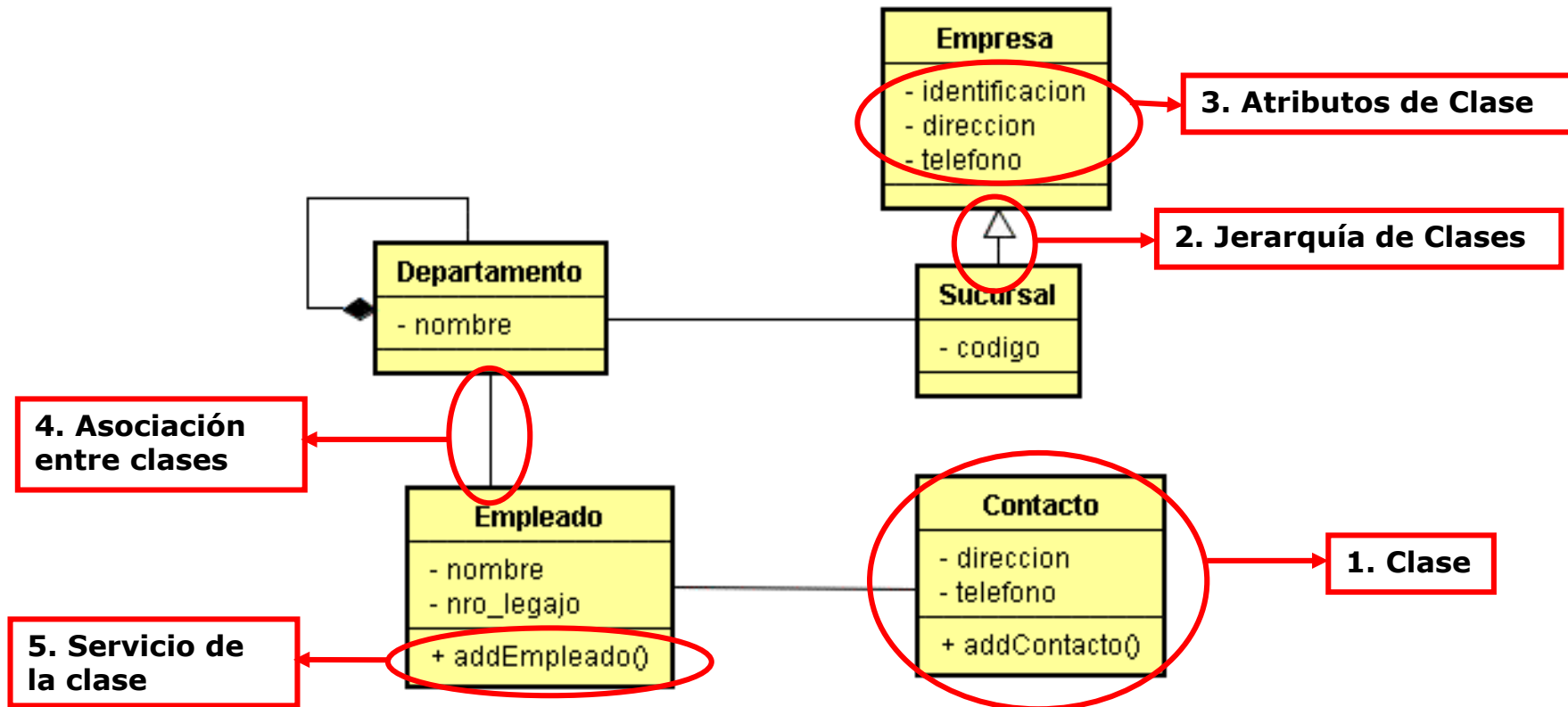
Las asociaciones capturan relaciones entre instancias de clases. Estas conexiones entre objetos se derivan del dominio del problema

5. Definición de servicios

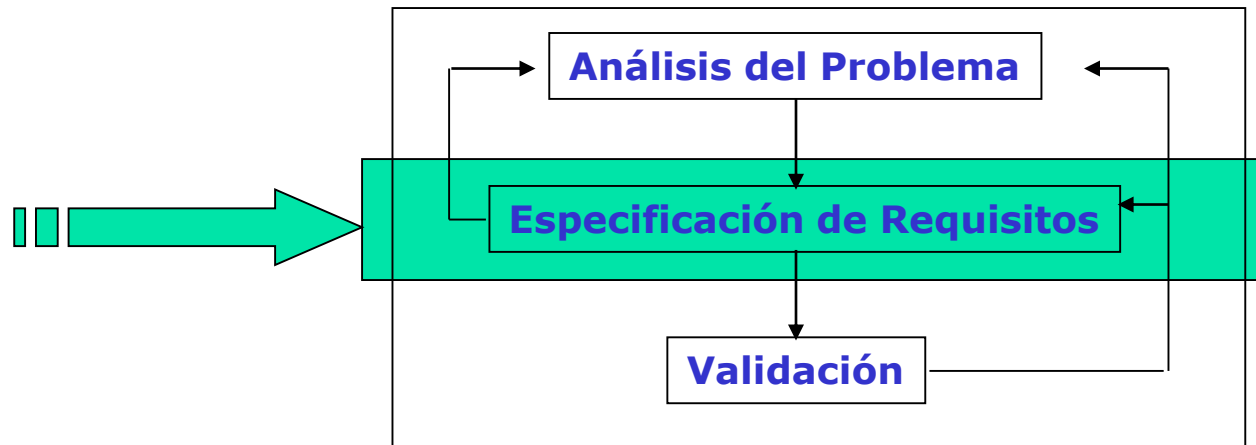
Un objeto presta y realiza un conjunto de servicios predefinidos. Un método para identificar los servicios de un objeto es, definir los posibles estados del sistema, eventos y respuestas requeridas

Análisis del Problema: Modelado Orientado a Objetos

Ejemplo de Diagrama de Clases



Proceso de Ingeniería de Requerimientos





Software Requirement Specification (**SRS**)

El objetivo central de la **Especificación** es
producir el documento SRS.

- Se obtiene como resultado un ***Documento de Especificación de Requerimientos de Software (SRS)***.
- El **SRS** está escrito en base a los conocimientos adquiridos durante el **análisis del problema**.
- La transición del **análisis del problema** a la **especificación** no es una tarea sencilla.



Documento **SRS**

El documento **SRS** es un escrito oficial que describe los requisitos del sistema para clientes, usuarios y desarrolladores.

El documento **SRS** describe:

- Los ***servicios*** y ***funciones*** que el sistema deberá proveer.
- Las ***restricciones*** bajo las cuales el sistema debe operar.
- Las ***propiedades generales del sistema***.
- Definiciones de ***otros sistemas*** con los cuales el sistema se debe integrar.
- Información acerca del ***dominio de aplicación*** del sistema.
- ***Restricciones sobre el proceso*** usado para desarrollar el sistema.
- ***Glosario*** de Términos.



Necesidad de un Documento **SRS**

- El **SRS** establece la base para un acuerdo entre el cliente y el desarrollador de lo que el sistema tendrá que hacer.

Se formaliza con un contrato legal entre el cliente y el desarrollador. El cliente describe claramente lo que espera del sistema y el desarrollador entiende claramente qué software debe construir.

- El **SRS** es una referencia para la validación del producto final.

Ayuda al cliente a determinar si el software cumple con los requerimientos solicitados, y le permite al desarrollador demostrar que los requerimientos acordados fueron cumplidos.

- El **SRS** de calidad es el requisito previo para obtener un software alta calidad.

La calidad de la SRS tiene un alto impacto en el costo y planificación del proyecto. El costo de corregir un error aumenta casi exponencialmente a medida que se avanza en el desarrollo del producto de software.



Notación para el documento **SRS**

- **Lenguaje Natural Estructurado**: plantillas estándares.
- **Lenguajes de Descripción de Diseño**: similar a lenguajes de programación, pero más abstractos.
- **Notaciones gráficas**: lenguajes gráficos, complementados con notas de texto (DFD, **UML**).
- **Especificaciones Matemáticas**: carecen de ambigüedad, sin embargo muchos clientes pueden resistirse a firmar un contrato del sistema porque no comprende las especificaciones formales.

En general, es preciso utilizar más de una notación.

Características del Documento SRS

Correcto: si todos los requisitos del SRS representan las necesidades reales de los usuarios.

Completo: si especifica todas las respuestas a todas las entradas.

No Ambiguo: si c/requisito tiene una única interpretación. Minimiza ambigüedades el uso de un lenguaje formal de especificación.

Verificable: si para cada requisito existe algún proceso que pueda comprobar si el sistema construido cumple con ese requisito.

Consistente: si no hay conflictos entre requisitos. Ej: diferentes requisitos utilizan diferentes términos para referirse al mismo objeto, conflictos lógicos o temporales.

Clasificado por importancia y/o estabilidad de los requisitos: **Críticos, importantes** pero no críticos, y **deseables**. Estabilidad de un requisito refleja las posibilidades de cambio a futuro del mismo.

Modificable: cualquier cambio puede hacerse fácilmente preservando la integridad y la consistencia. La redundancia es un importante obstáculo.

Trazable: si es claro el origen de cada requisito y su referencia en el desarrollo futuro. Ayuda a la verificación y validación.

Componentes del documento SRS

Requisitos funcionales: operaciones que se realizan sobre los datos de entrada para obtener la salida (funciones del sistema). Para c/u, especificar los datos de entrada y su origen, unidades de medida, rango de validez. También el comportamiento del sistema en situaciones anormales (ej. entrada inválida, sucesos no esperados).

Requisitos de Rendimiento: limitaciones de rendimiento del sistema. Define los requisitos estáticos (que no imponen restricciones sobre la ejecución del sistema) y requisitos dinámicos (especifican restricciones sobre el comportamiento de la ejecución del sistema, como tpo. de respuesta).

Restricciones de Diseño: factores en el entorno del cliente que pueden restringir al diseñador (normas a seguir, límites de recursos, el ambiente de funcionamiento, y las políticas que puedan tener un impacto en el diseño del sistema). Incluye: cumplimiento de los estándares, limitaciones del hardware, fiabilidad y tolerancia a fallos, y seguridad.

Requerimientos de Interfaces externas: interacción sistema-persona, hardware y con otros sistemas de software.

- **Interfaces de usuario:** manual de usuario, formatos de pantallas, comentarios, mensajes de error.
- **Hardware:** características lógicas de cada interfaz entre el producto de software y los componentes de hardware, si el software ejecutará en el hardware existente, restricciones de memoria, etc.
- **Interfaz con otros Sistemas** que el sistema utilizará o que usarán a él.

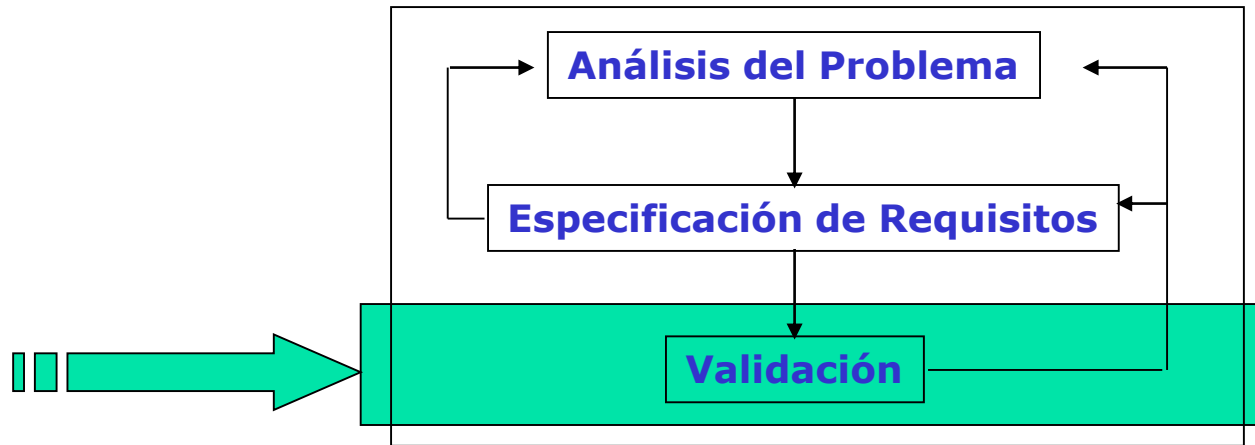
Estructura de Documento SRS

Un método para organizar la especificación de requerimientos es especificar primero las interfaces externas, luego los requerimientos funcionales, requerimientos de rendimiento, restricciones de diseño, y atributos del sistema.

IEEE/ANSI 830-1998:
Standard for Software
Requirements Specification

1. Introduction
 - 1.1 Purpose
 - 1.2 Scope
 - 1.3 Definitions, Acronyms, and Abbreviations
 - 1.4 References
 - 1.5 Overview
2. Overall Description
 - 2.1 Product Perspective
 - 2.2 Product Functions
 - 2.3 User Characteristics
 - 2.4 General Constraints
 - 2.5 Assumptions and Dependencies
3. Specific Requirements
 - 3.1 External Interface Requirements
 - 3.1.1 User Interfaces
 - 3.1.2 Hardware Interfaces
 - 3.1.3 Software Interfaces
 - 3.1.4 Communication Interfaces
 - 3.2. Functional Requirements
 - 3.2.1 Mode 1
 - 3.2.1.1 Functional Requirement 1.1
 - :
 - 3.2.1.*n* Functional Requirement 1.*n*
 - :
 - 3.2.*m* Mode *m*
 - 3.2.*m*.1 Functional Requirement *m*.1
 - :
 - 3.2.*m*.*n* Functional Requirement *m*.*n*
 - 3.3 Performance Requirements
 - 3.4 Design Constraints
 - 3.5 Attributes
 - 3.6 Other Requirements

Proceso de Ingeniería de Requerimientos





Validación

El objetivo básico de la actividad de **validación** es **asegurar que el documento SRS refleja las necesidades reales, de forma clara y precisa**, y comprobar que dicho documento es de "buena calidad".

- Los tipos errores que normalmente se producen son: omisión, inconsistencia, hecho incorrecto, ambigüedad.
- La **validación** debe centrarse en el descubrimiento de estos errores que ocurren en cualquier proyecto.
- La inspección del documento SRS, es el método más común de **validación**.
- La **validación** es realizada por un equipo conformado tanto por el cliente, como por usuarios y desarrolladores.



Validación: Revisión de Requerimientos

- La revisión de requerimientos es realizada por un grupo de personas para encontrar errores en la especificación de los requerimientos de un sistema.
- El grupo de revisión debe incluir: el **autor de los requisitos**, que entendió las necesidades del cliente, algún **integrante del equipo de diseño**, y el **responsable** de mantener el documento **SRS**. También, es buena práctica incluir personas como un ingeniero calidad de software.
- El proceso de revisión es también usado para estudiar factores que afectan la calidad, tales como la prueba.



Validación: Revisión de Requerimientos

Se pueden utilizar listas de verificación (Checklist)

- ¿Están todos los recursos de hardware definidos?
- ¿Están bien especificados los tiempos de respuesta de las funciones?
- ¿Todo el hardware, el software externo, y las interfaces de usuario están definidos?
- ¿Se han especificado todas las funciones requeridas por el cliente?
- ¿Cada requisito es comprobable?
- ¿Está definido el estado inicial del sistema?
- ¿Están especificadas las respuestas a las condiciones anormales?
- ¿Son posibles las modificaciones futuras especificadas?



Bibliografía Consultada

- Pankaj Jalote. An Integrated Approach to Software Engineering, Third Edition, 2005.

Capítulo 3: Software Requirements Analysis and Specification

- Ian Sommerville. Ingeniería del Software, 9th edition, Addison Wesley. Pearson Education, 2011.

Capítulo 4: Ingeniería de Requerimientos