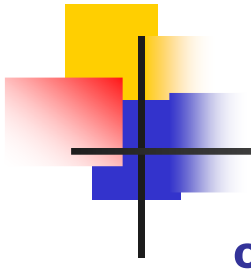


Requerimientos del Software

- 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 qué construir es la tarea más difícil en la construcción de sistemas de software.

- La complejidad y el tamaño de los sistemas de software aumenta continuamente.

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

- Surge la necesidad de 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 dificultosa y propensa a errores.

- ✓ La fase de especificación de requerimientos de software traduce las ideas de las mentes de los clientes (entrada), en un documento formal (salida).
  - ✓ La entrada es informal e imprecisa, y generalmente incompleta.
  - ✓ La salida es un conjunto de requisitos bien determinado, que se espera sea completa y precisa.

Descubrir y Analizar los Requerimientos

- La actividad de especificación de los 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.

(Cualquier proceso de traducción formal necesita una entrada precisa y sin ambigüedades para producir una salida formal)

La etapa de especificación de requerimientos perjudica enormemente el resultado final, si es realizada en forma errónea. Ninguna otra etapa del desarrollo es tan dificultosa de rectificar posteriormente.

Descubrir y Analizar los Requerimientos

Algunas causas que producen una errónea conceptualización del problema pueden ser:

- ✓ Carencia o escasez de conocimientos sobre el dominio
- ✓ Dominio de aplicación demasiado complejo
- ✓ Carencia de experiencia en:
  - detección de requerimientos,
  - conceptualización de los mismos,
  - analogía con sistemas anteriores, etc.

Requerimientos del Software

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

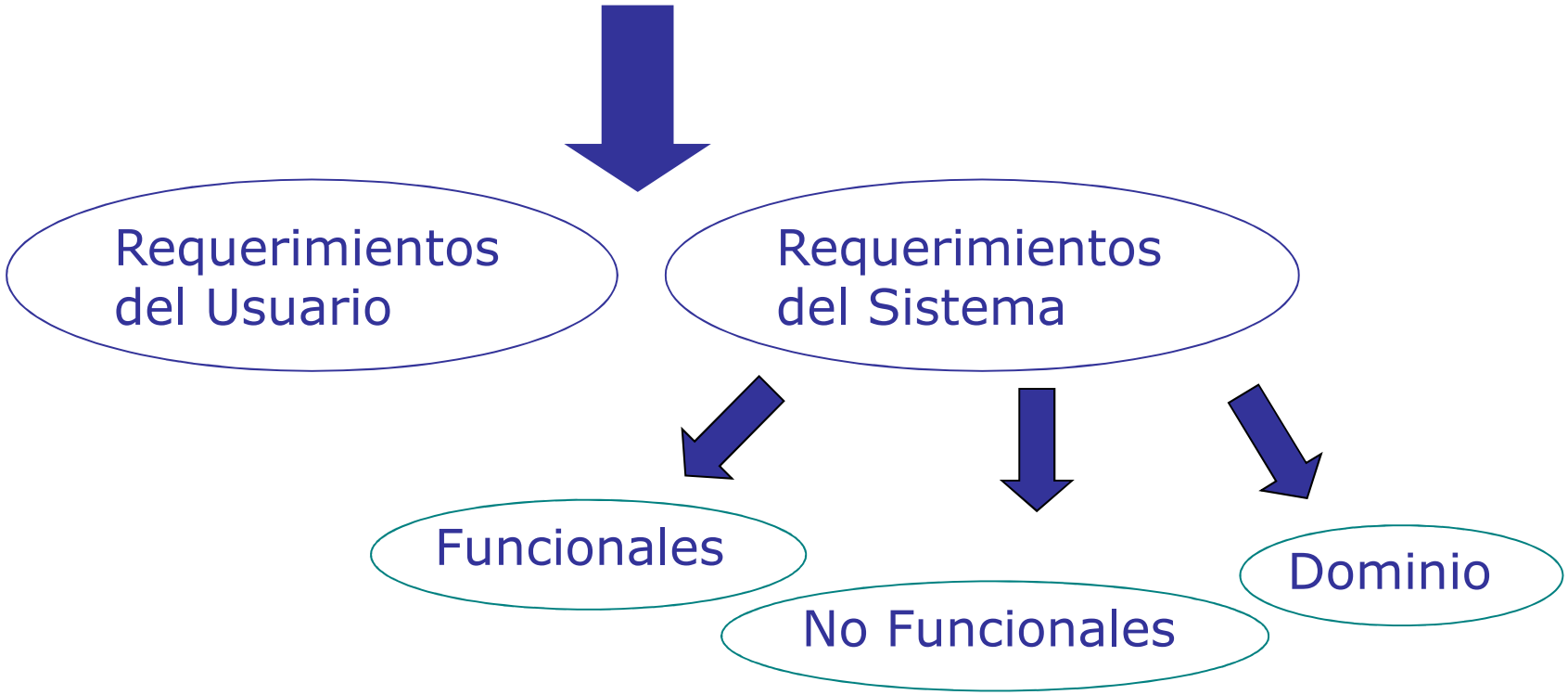
IEEE define un requisito como:

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

…….considerando que los requisitos de software que se están definiendo, son los que deberá satisfacer el sistema que aún no se está desarrollando.

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 más formal y 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 o su reacción ante diferentes escenarios.
- Pueden estar descriptos en diferentes niveles de abstracción.

Todos los servicios solicitados por el usu deben ser contempla por el sistema

Los requerimientos no deben tener definiciones contradictorias

nera muy

s.

que los

requ mientos fun nales 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)

- Hay 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)

- Surgen de necesidades del usuario, restricciones de presupuesto, políticas de organización, necesidad de interoperabilidad, o de 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.

Por ejemplo: en un sistema de control de vuelos podría ser:

"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 es la rama de la ingeniería de software que trata la identificación del propósito de un sistema de software y el contexto en el cual será usado.

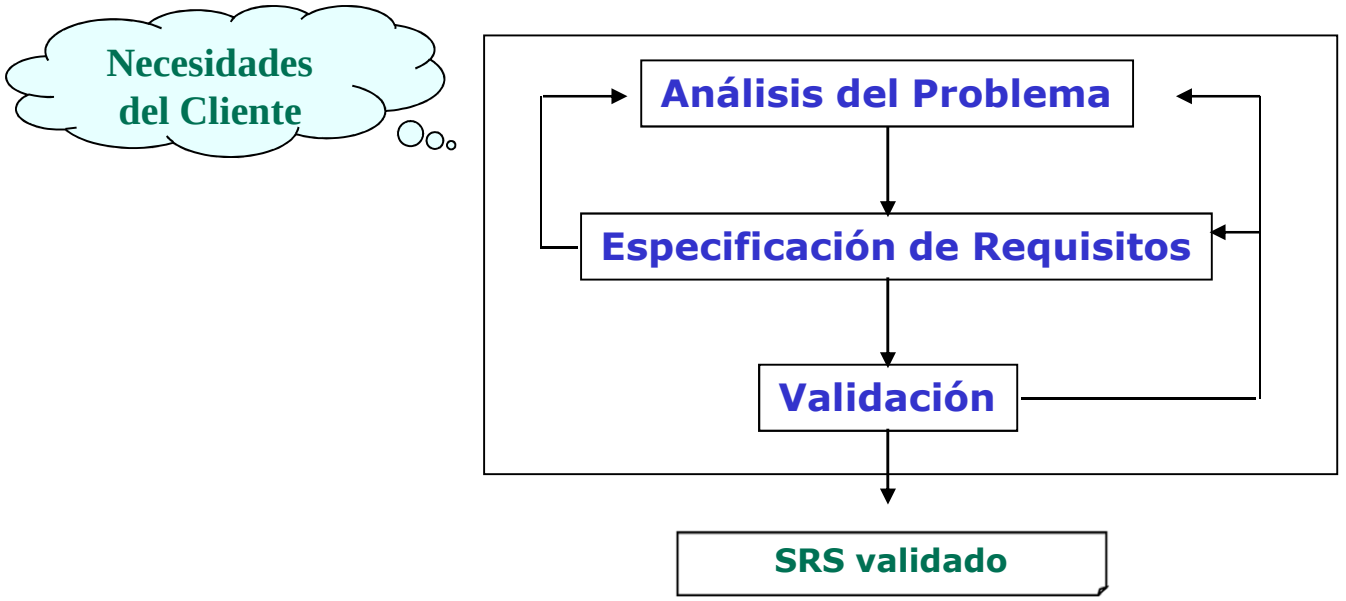
Impacto de los errores en la etapa de requisitos

- El software resultante puede no satisfacer a los usuarios.
- Las interpretaciones múltiples de los requisitos pueden causar desacuerdos entre clientes y desarrolladores.
- Puede insumirse demasiado tiempo y dinero construyendo el sistema erróneo.

Proceso de la Ing. Requerimientos

El proceso de la 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).

Consta de tres tareas básicas:



Proceso de la Ing. 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.

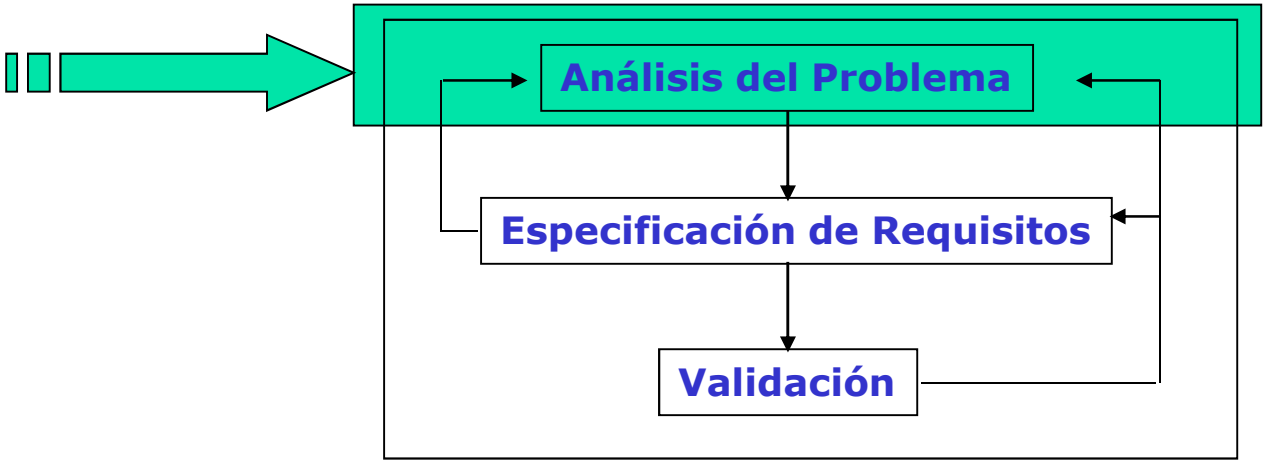
ESPECIFICACION DE REQUISITOS (DESCRIPCION DEL PRODUCTO)

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 la Ing. 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 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**

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.
- El analista tendrá reuniones con los clientes y usuarios finales:
  - En las primeras, los clientes y los 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.
  - El analista explica al cliente lo que hará el sistema, y utiliza las reuniones para verificar si el sistema que propone es coherente con los objetivos del cliente.

Este enfoque puede ser muy útil y práctico al Análisis en algunas situaciones. Además si se realiza la validación, se minimizan los riesgos de trasladar errores a la etapa de especificación.

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)

- Los DFD se usaban mucho antes del comienzo de la disciplina de la ingeniería de software.
- Son muy útiles para comprender un sistema y pueden utilizarse eficazmente en el análisis.
- Un DFD muestra el flujo de datos a través de un sistema. Muestra una función que transforma las entradas en los productos deseados.
- En cualquier sistema complejo esta transformación no es simple y es necesario una serie de transformaciones antes de que se produzca la salida.
- El agente que realiza la transformación de los datos de un estado a otro se llama proceso (burbuja).
- Por lo tanto, un DFD muestra el movimiento de datos a través de las diferentes transformaciones o procesos del sistema.

Diagrama de Flujo de Datos (DFD)

Gráficamente:

- Los procesos se muestran con un círculo y un nombre.
- Los flujos de datos tienen nombre y flechas de entrada o salida de las burbujas.
- Un rectángulo representa una fuente y es un creador de red o de consumidores de los datos.

Ejemplo de un DFD de un Sistema de pago a trabajadores.

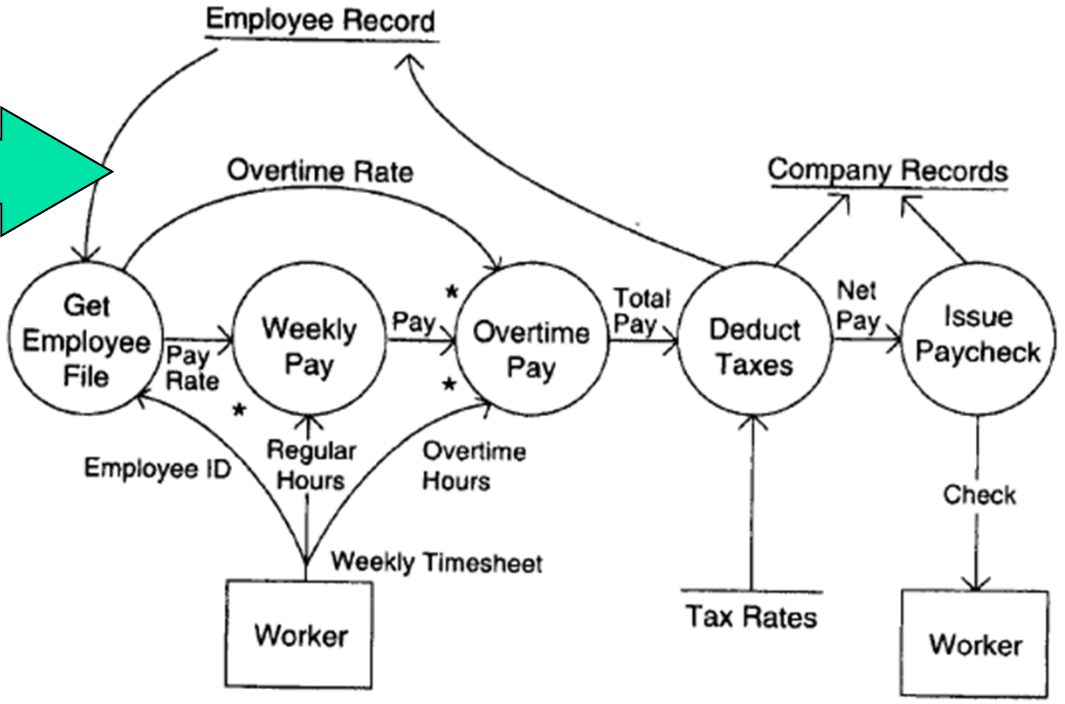


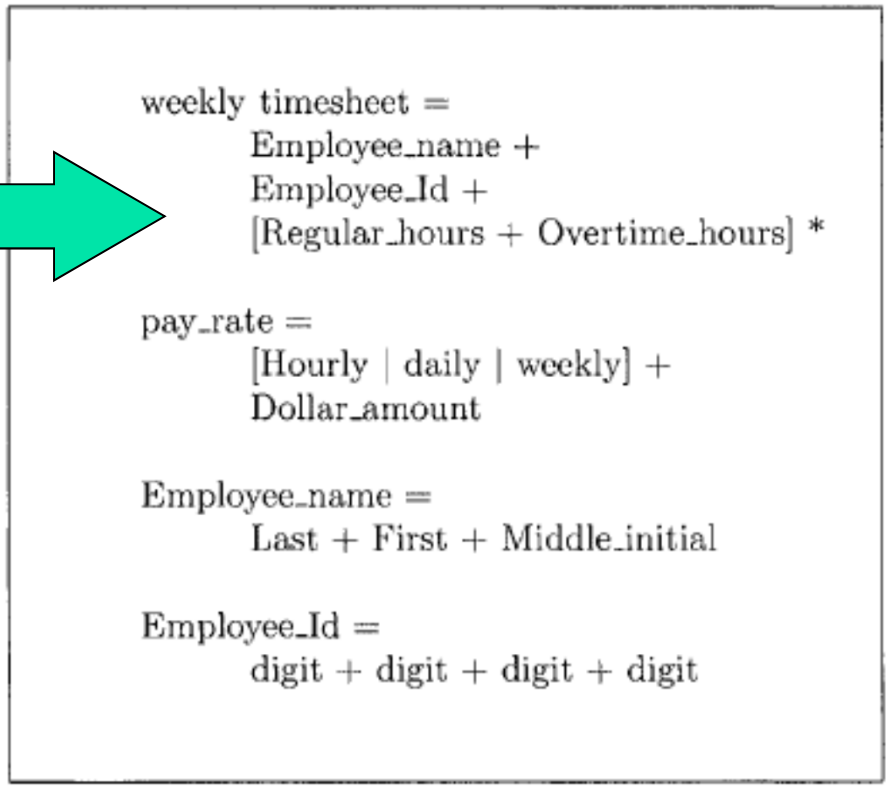
Diagrama de Flujo de Datos (DFD)

- 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.

Diagrama de Flujo de Datos (DFD)

El diccionario de datos es un repositorio de datos de diversos flujos de datos definidos en un DFD.

Ejemplo de un Diccionario de Datos



weekly timesheet =
Employee_name +
Employee_Id +
[Regular_hours + Overtime_hours] *

pay_rate =
[Hourly | daily | weekly] +
Dollar_amount

Employee_name =
Last + First + Middle_initial

Employee_Id =
digit + digit + digit + digit

Análisis del Problema: Método de Análisis Estructurado

EJEMPLO:

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

Cree que un sistema automatizado puede agregar un atractivo interesante para sus clientes.

1) El primer paso es identificar las diferentes partes implicadas:

Cliente del sistema:

El propietario del restaurante.

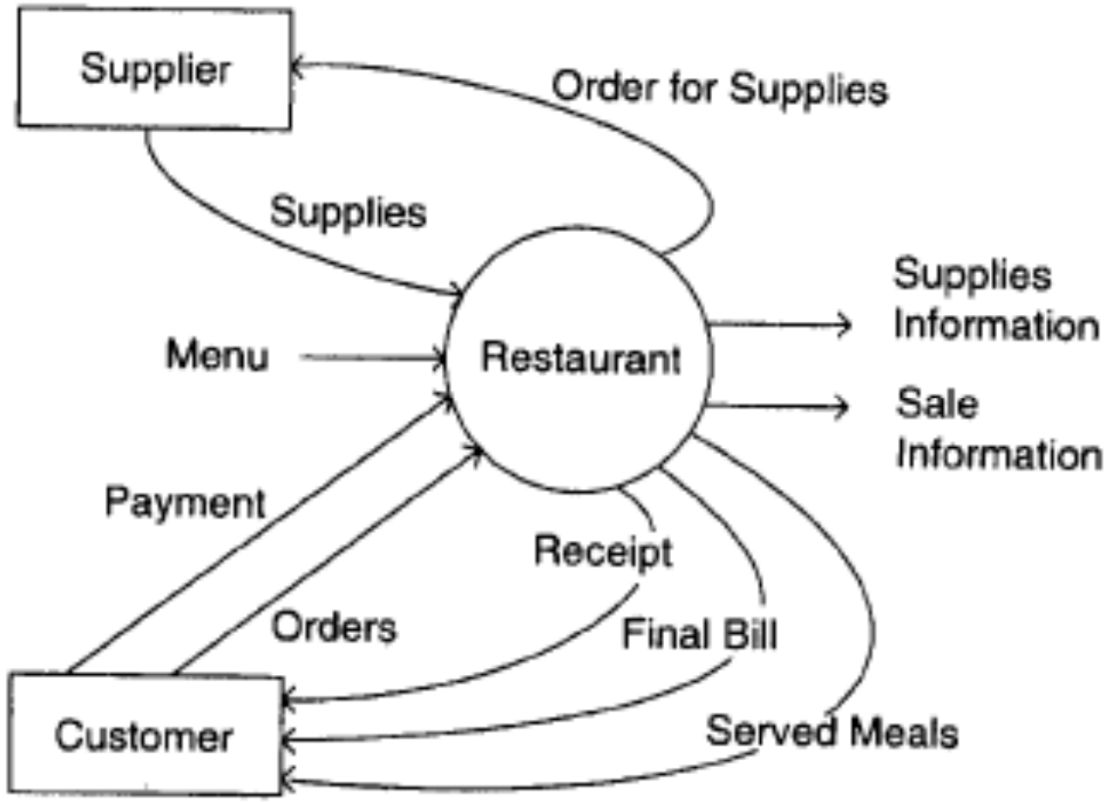
Usuarios potenciales:

Camareros, Operador de la caja registradora.

Método de Análisis Estructurado: Ejemplo

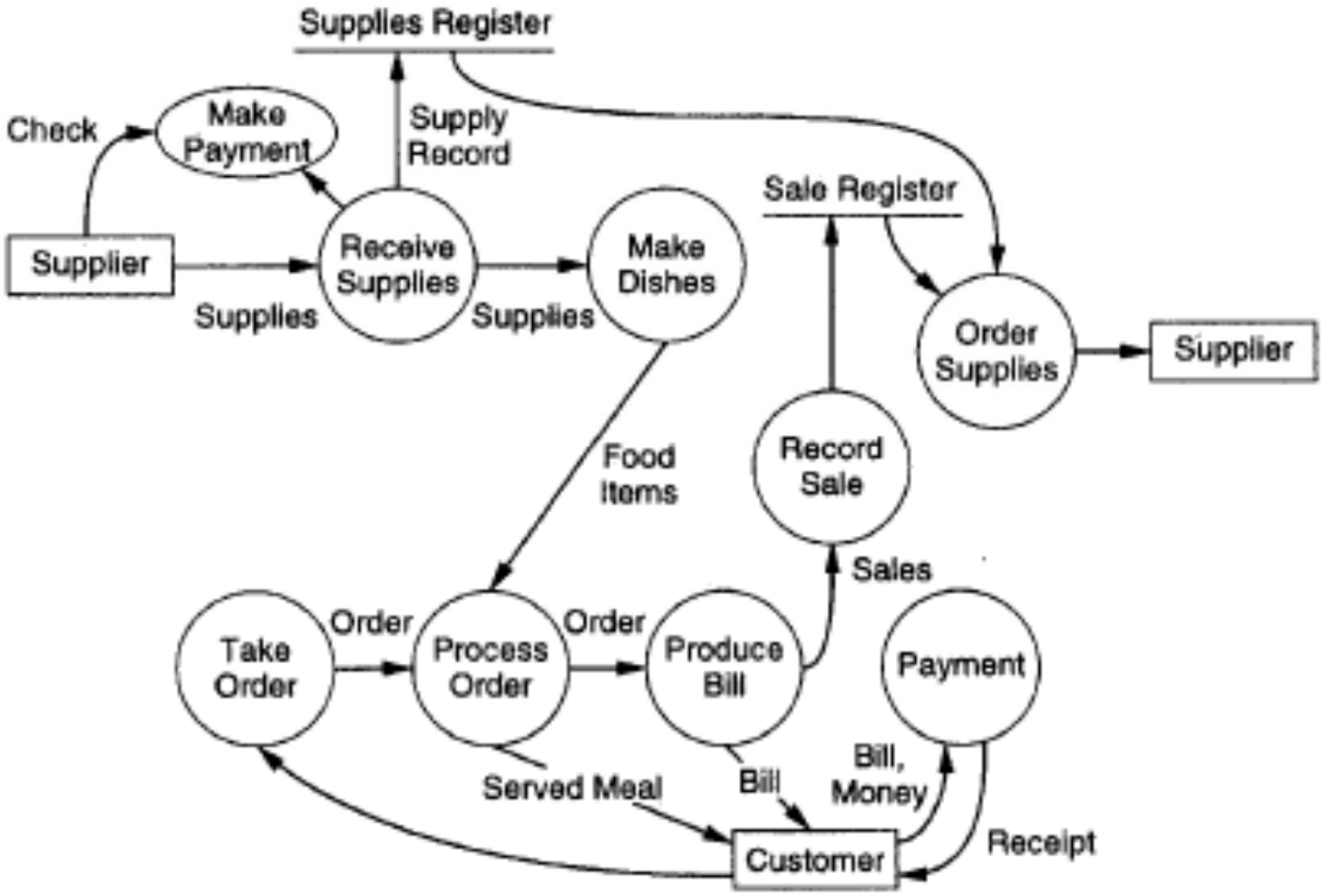
2) Diagrama de Contexto

Este DFD muestra entradas y salidas del sistema actual del restaurante, pero sin detalles sobre su funcionamiento.



Método de Análisis Estructurado: Ejemplo

3) Se refina el Diagrama de Contexto, se agregan detalles de funcionamiento y se construye un DFD lógico del sistema físico.



Método de Análisis Estructurado: Ejemplo

4) Se establecen los objetivos para el nuevo sistema:

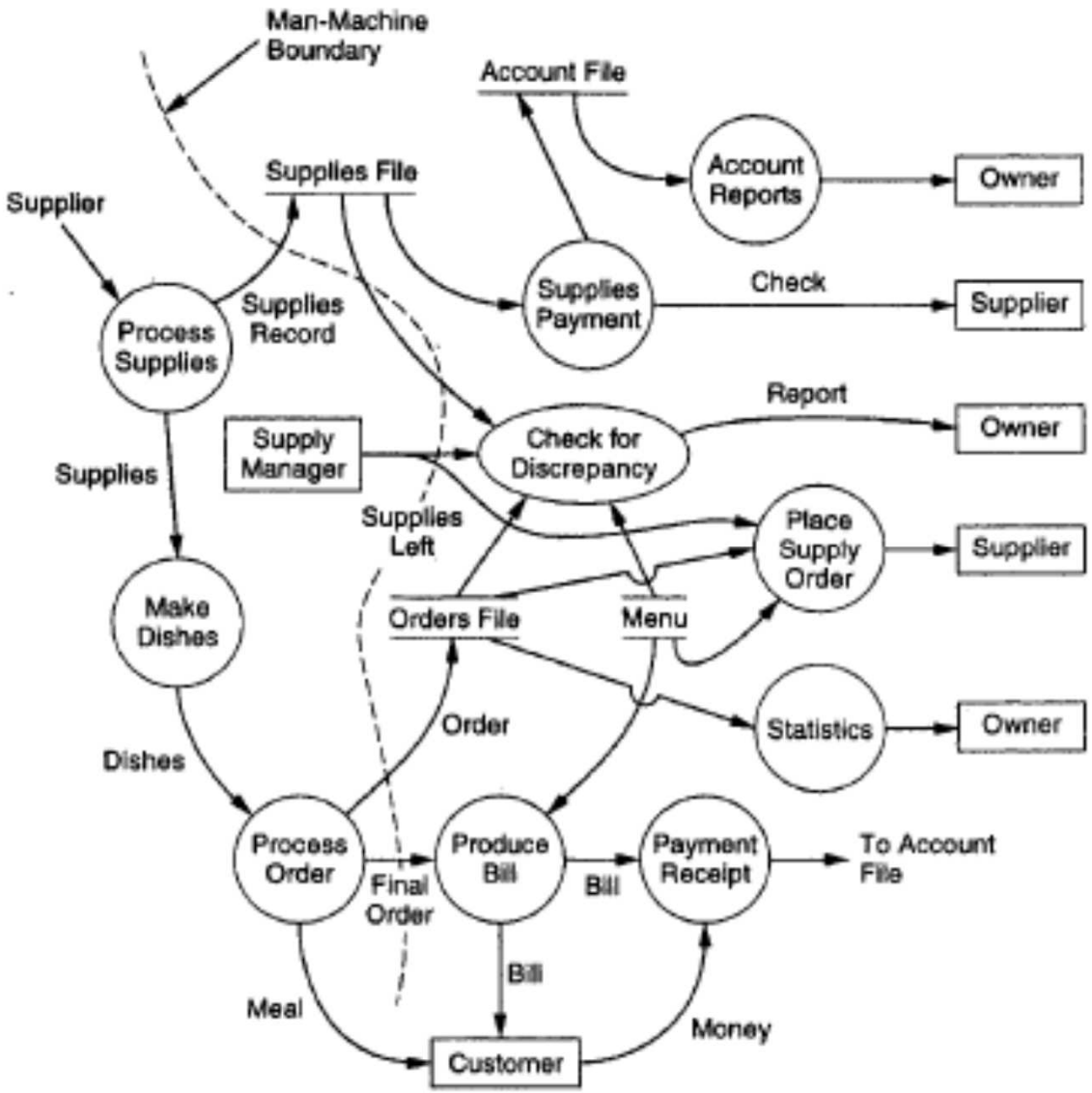
- Automatizar el procesamiento de pedidos y de facturación.
- Automatizar la contabilidad.
- Hacer pedidos de insumos más precisos para reducir las sobras al mínimo al final del día, y que se minimicen las órdenes no satisfechas debido a la falta de insumos.
- Detectar si el personal está sacando algunos alimentos.
- Contar con estadísticas sobre las ventas.

Con estos objetivos, se pueden definir los **límites** del sistema del software a construir, el cual afectará a la mayoría de los aspectos del sistema anterior, excepto a la preparación de platos.

Método de Análisis Estructurado: Ejemplo

5) DFD del nuevo sistema

- Algunos procesos siguen manuales, pero podrían cambiar por ser afectados por otros.
- Archivos del sistema: Insumos, Contabilidad, Ordenes y Menú.
- Nuevos procesos se incluyen en el sistema: Comprobar Diferencias, Informes de Contabilidad, Estadísticas.



Método de Análisis Estructurado: Ejemplo

6) Diccionario de Datos del nuevo Sistema

Supplies_file = [date + [item_no + quantity + cost]*]*
Orders_file = [date + [menu_item_no + quantity + status]*]*
status = satisfied | unsatisfied
order = [menu_item_no + quantity]*
menu = [menu_item_no + name + price + supplies_used]*
supplies_used = [supply_item_no + quantity]*
bill = [name + quantity + price]* +
total_price + sales_tax + service_charge + grand_total
discrepancy_report = [supply_item_no +
amt_ordered + amt_left + amt_consumed + descr]*

Cuando el DFD y el Diccionario de Datos han sido aprobados por el propietario del restaurante, la actividad de la análisis del problema está completa.

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 que se modela. Son objetos candidatos aquellos de los que el sistema debe recordar algo, si el sistema necesita algunos servicios del objeto para realizar sus propios servicios, y/o si el objeto tiene varios atributos.

2. Identificación de estructuras

Representan jerarquías entre clases de objetos. Se definen capturando relaciones de generalización-especialización, todo-parte.

3. Identificación de atributos

Los atributos agregan detalles acerca de las clases y son repositorios de datos de un objeto. Dependen del problema que se está modelando.
Para identificar atributos, considerar cada clase y determinar cuales atributos de la clase serán necesarios para el dominio del problema.

4. Identificación de asociaciones

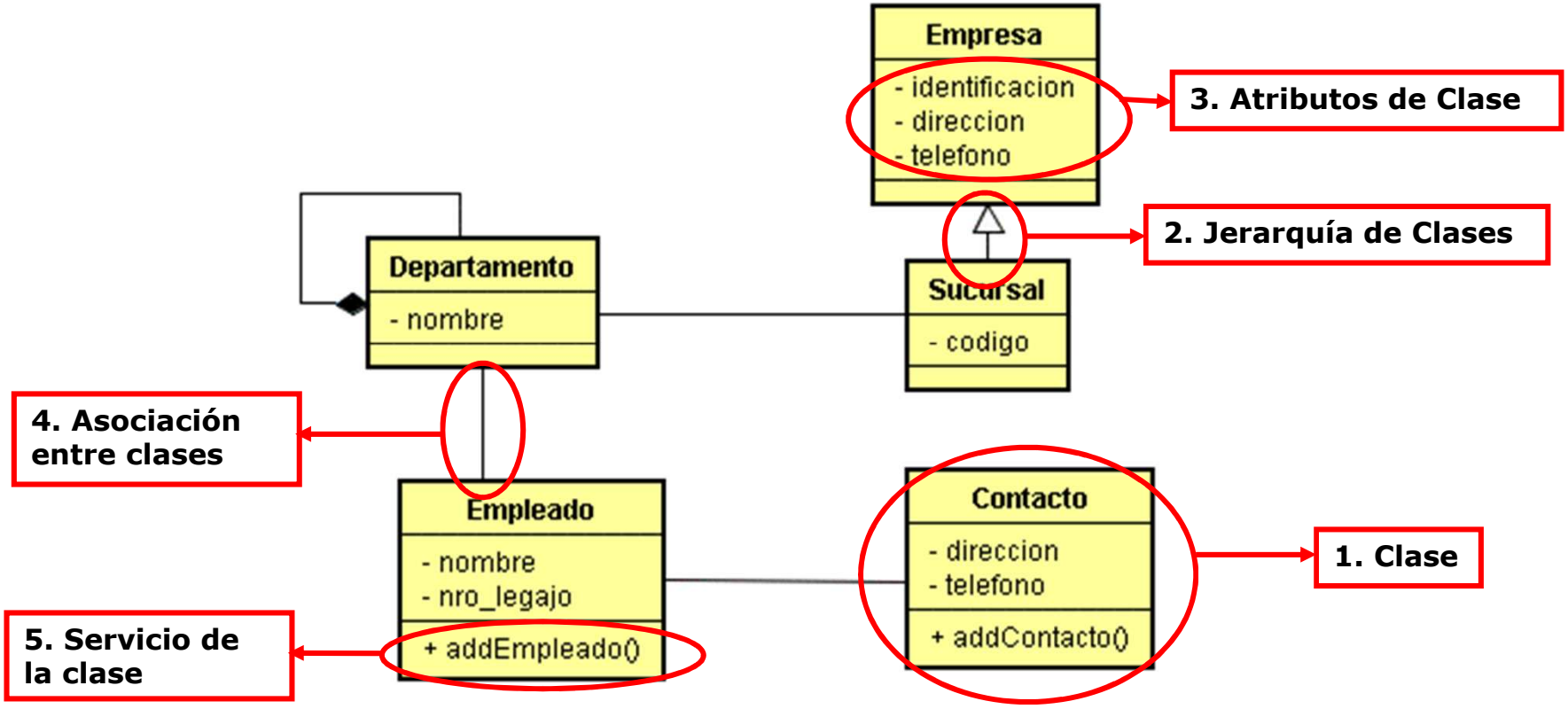
Las asociaciones capturan relaciones entre instancias de varias clases, y contienen una multiplicidad que puede ser 1:1, 1:N o N:N. Las asociaciones o conexiones entre objetos se derivan directamente del dominio del problema una vez que los objetos han sido identificados.

5. Definición de servicios

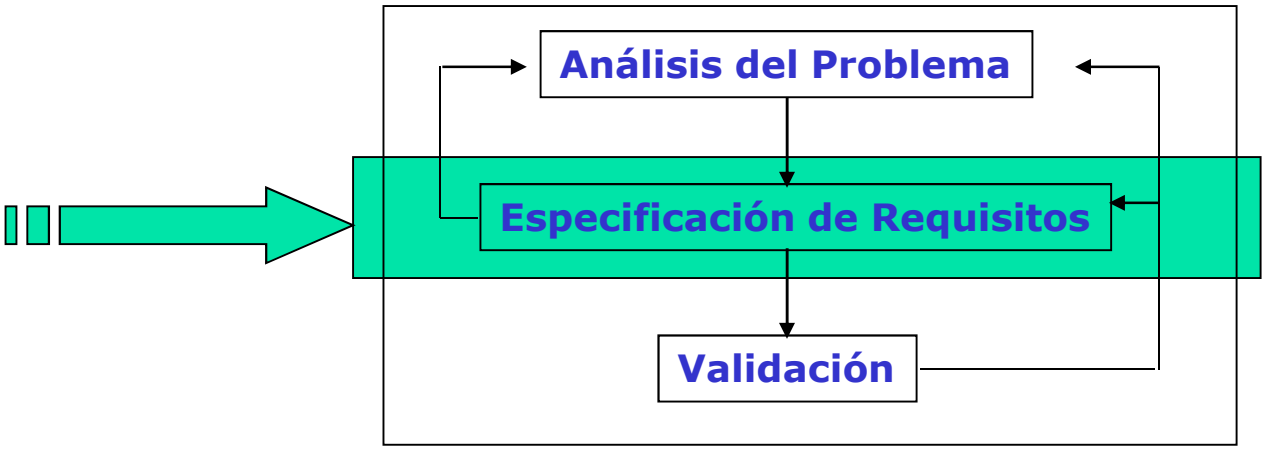
Un objeto realiza un conjunto de servicios predefinidos. Estos se ejecutan cuando el objeto recibe un mensaje solicitándole dicho servicio. Los servicios representan el elemento activo en el MOO, son un agente del cambio de estado o procesamiento. Se identifican los servicios que se necesitan para crear, destruir, y mantener las instancias de las clases.
Un método para identificar servicios es, definir los estados del sistema y luego en cada uno listar los acontecimientos o eventos y las respuestas requeridas. Para c/u, identificar los servicios que las clases deben poseer.

Análisis del Problema: Modelado Orientado a Objetos

Ejemplo de Diagrama de Clases



Proceso de la Ing. Requerimientos



Especificación de Requerimientos de Software

- El resultado es el ***Documento de Especificación de Requerimientos de Software*** (SRS).
- La transición del **análisis** a la **especificación** no es una tarea sencilla.
- El SRS está escrito en base a los conocimientos adquiridos durante el **análisis**.
- El objetivo primordial del **análisis** es la comprensión del problema, mientras que el objetivo básico de la **especificación** es producir el documento de la SRS.
- El SRS debe especificar: interfaces de usuario, errores de salida, limitaciones, restricciones de diseño, estándares de cumplimiento, recuperación, etc.

Documento de SRS

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

El documento de SRS describe:

- Los ***servicios*** y ***funciones*** que el sistema debería 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 de SRS

- La SRS establece la base para un acuerdo entre el cliente y el desarrollador de lo que el sistema hará.

  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.

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

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

- La SRS de calidad es un 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 la 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 de SRS

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

Completo: si el sistema hace todo lo que debe hacer y especifica todas las respuestas a todas las entradas.

No Ambiguo: si y sólo si cada requisito tiene una única interpretación. Una forma de evitar ambigüedades es utilizar algún lenguaje formal de especificación de requisitos.

Verificable: si y sólo si para cada requisito existe algún proceso que pueda comprobar si el software final cumple con ese requisito.

Consistente: si no hay conflictos entre requisitos. Por ej, diferentes requisitos pueden utilizar 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 coherencia. 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 de SRS

Requisitos funcionales: especifica todas las 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, las unidades de medida, y su rango de validez. También el comportamiento del sistema en situaciones anormales (entrada inválida, error durante una computación).

Requisitos de Rendimiento: especifica 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 las opciones del 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: especificar todas las interacciones del sistema con personas, hardware y otros software.

- Interfaces de usuario: manual de usuario con los comandos de usuario, los formatos de pantallas, comentarios y mjes 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 programas que el sistema utilizará o que usarán a él.

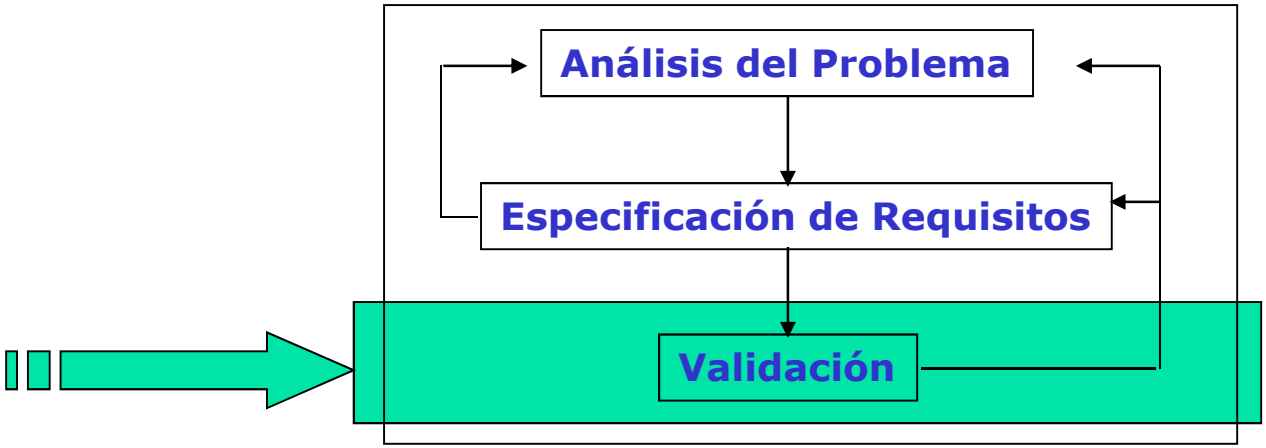
Estructura de Documento de SRS

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 la Ing. 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. Además, debe comprobar que dicho documento es de "buena calidad".

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

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 entiende las necesidades del cliente, alguien del equipo de diseño, y el responsable de mantener el documento de requisitos. También, es buena práctica incluir personas como un ingeniero calidad de software.
- El proceso de revisión es también usado para estudiar los factores que afectan la calidad, tales como la capacidad de prueba.

Validación: Revisión de Requerimientos

Se utilizan listas de verificación (Checklist) para realizar la revisión y asegurarse de que la principal fuente de errores no es pasada por alto por los revisores.

- **¿Están todos los recursos de hardware definidos?**
- **¿Están bien especificadas 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?**
- **¿Es cada requisito 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, 7th edition, Addisson Wesley. Pearson Education, 2005. ISBN: 978-84-7829-074-1.

Capítulo 6: Requerimientos del Software

Capítulo 7: Ingeniería de Requerimientos