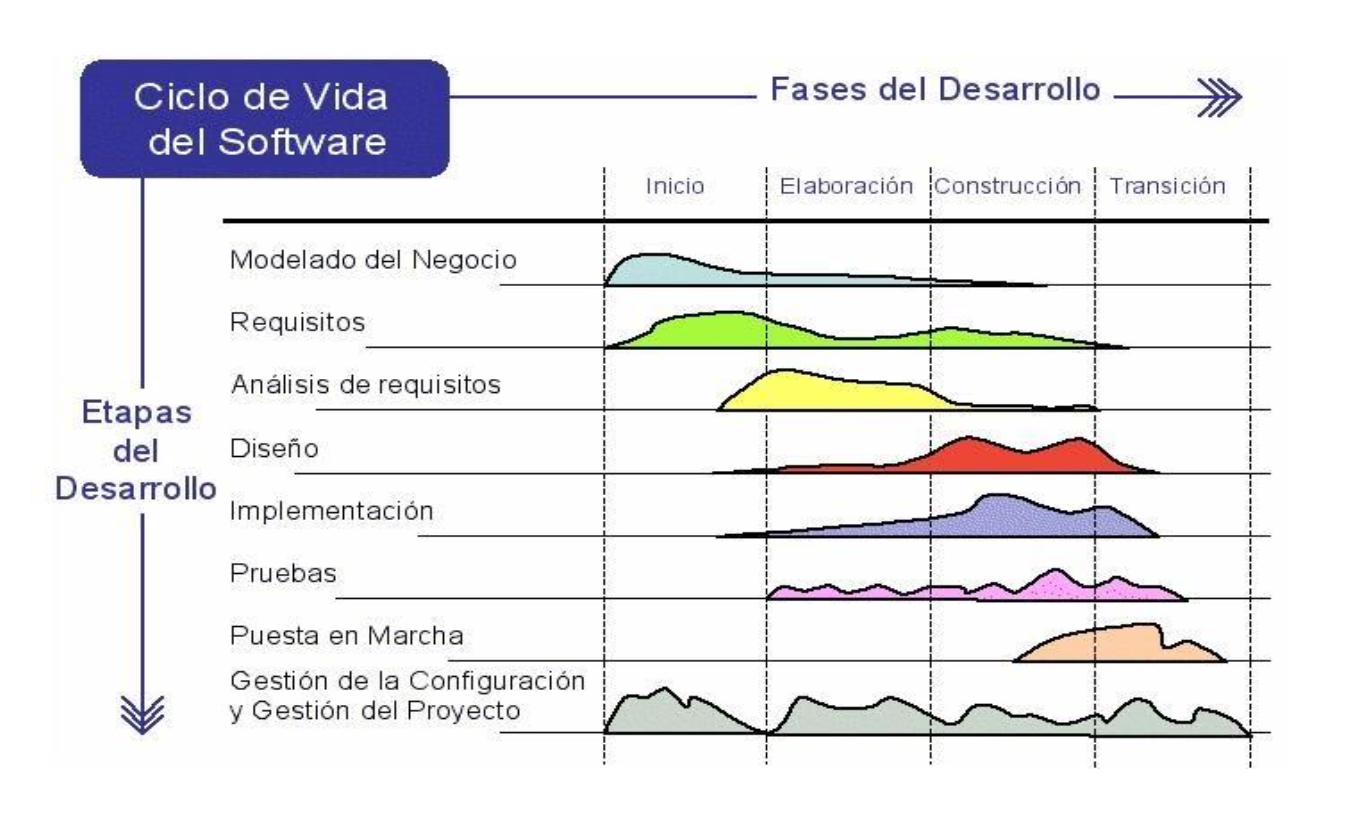
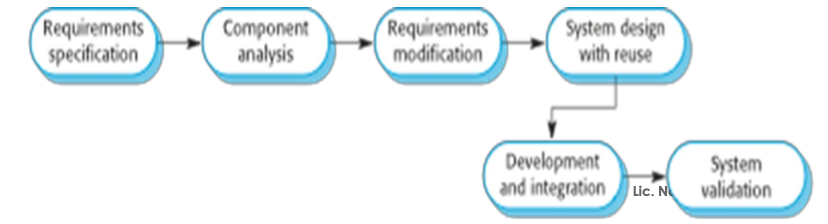
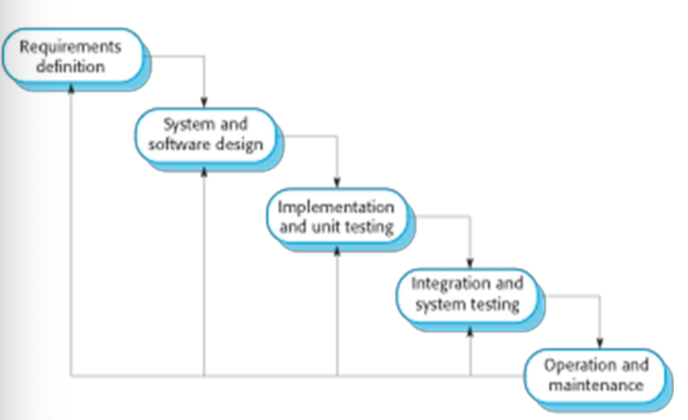
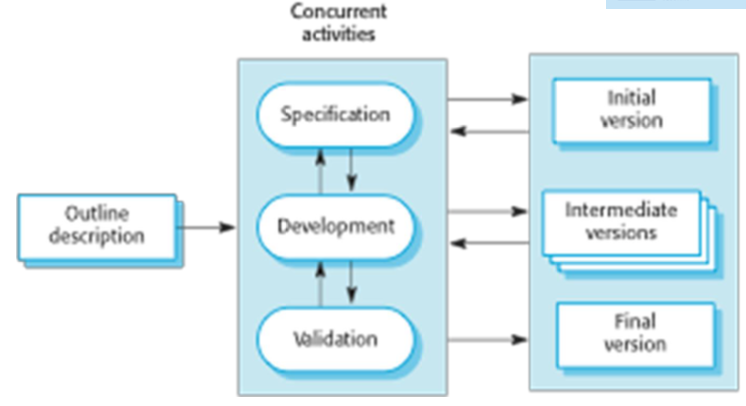
# Unidad 1



## Proceso de desarrollo





## Criterios a considerar

Entre los criterios a considerar podemos encontrar el nivel de flexibilidad de los requerimientos, la escala y el alcance del desarrollo, la conveniencia en cuanto a las exigencias del cliente, los usuarios finales y las características del equipo de desarrollo, entre otros.

## Modelado de negocio

* La perspectiva de cómo se mueven los procesos en el lugar en donde se construirá una solución de software (institución, empresa dominio)
* Los diagramas de proceso del negocio, son representaciones de los procesos que se dan en la organización y que son de interés para el sistema.
* Se trata de representar la relación existente entre un proceso y las unidades funcionales (como pueden ser los departamentos o personas) responsables del proceso.
* Es un diagrama de flujo extendido, en donde además de las operaciones, se detalla las personas que intervienen en un proceso

### Ejemplo de cómo debemos arrancar

* Objetos del sistema.
* Subsistemas que lo componen
* Requerimientos

### Casos de Uso

* Especifica requerimientos funcionales del Sistema.
* Se utiliza a nivel de Sistema o Negocio.
* El proceso de desarrollo está dirigido por los Casos de Uso.
* El diagrama trata de identificar:
* Actores:
  + Representa un rol que interactúa con el sistema y ejecuta las funcionalidades.
  + El nombre del actor debe describir el rol que desempeña.
  + Existen Actores Principales y Actores Secundarios.
* Casos de uso:
  + Describen la funcionalidad y requerimientos del sistema.
  + Deben ser descriptos con una acción en infinitivo al principio.
  + Deben ser equilibrados entre el tamaño y su descripción.
  + Evitar CU muy minimalista o abarcativos.
  + Un requerimiento o funcionalidad se puede descomponer y luego relacionar para formar un todo.
* Relaciones:
  + **Include:** debe ejecutarse siempre y ayudan a complementar la función de quién lo llama.
  + **Extend:** agrega funcionalidad a otro caso de uso, que puede o no ejecutarse siempre, es optativo.
  + **Generalización:** permite especificar comportamiento partiendo de la funcionalidad base.

## ¿Qué es un Modelo?

Un modelo es una simplificación (representación a bajo costo) de la realidad. Es una abstracción del sistema.

## ¿Qué lenguaje emplear para modelar software?

Lenguaje visual, fácil de interpretar y procesar.

## ¿Cuál es el objetivo de modelar?

* Nos ayudan a visualizar un sistema como es, o cómo queremos que sea.
* Nos permiten especificar la estructura o el comportamiento de un sistema.
* Nos dan una plantilla que nos guía en la construcción del sistema.
* Documentan las decisiones que hemos tomado.
* Nos facilitan la comunicación con el cliente.

## Principios de Modelado

* La elección de qué modelos crear tiene una profunda influencia en cómo atacamos un problema y cómo delineamos una solución.
  + Diferentes enfoques (analista estructurado, analista orientado a objeto o desarrollador de base de datos)
* Cada modelo debe poder expresarse a diferentes niveles de precisión.
* Los mejores modelos tienen una conexión con la realidad.
* Un único modelo no es suficiente. Cualquier sistema es enfocado mejor con un conjunto de modelos independientes, pero relacionados.

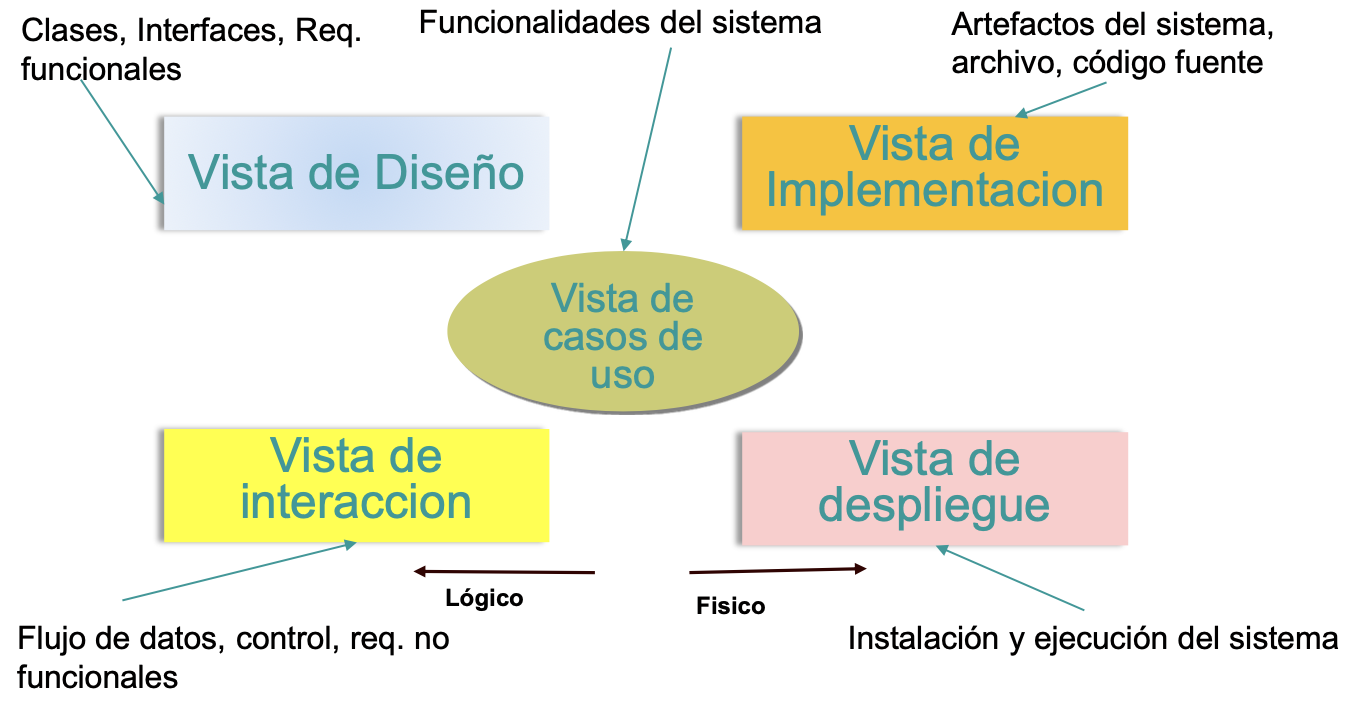
## Aportes del Modelado

* Se facilita la comunicación entre el equipo al existir un lenguaje común.
* Se dispone de documentación que trasciende al proyecto.
* Hay estructuras que trascienden lo representable en un lenguaje de programación.

## Conceptos del Modelado

* **Sistema software:** Descrito por un conjunto de modelos
* **Modelo:** Simplificación para comprender mejor un sistema
* **Diagrama:** Representación gráfica de un modelo (UML usa grafos)
* **Vista:** Subconjunto de diagramas de un modelo que analiza un aspecto concreto

## Vistas



## ¿Qué es UML?

Es un lenguaje de modelado, de propósito general, usado para la visualización, especificación, construcción y documentación de sistemas orientados a objetos.

* UML = Unified Modeling Language
* Un lenguaje de propósito general para el modelado orientado a objetos. Impulsado por el Object Management Group (OMG, Object Management Group, www.omg.org)
* Documento “OMG Unified Modeling Language Specification”
* UML combina notaciones provenientes desde:
  + Modelado Orientado a Objetos
  + Modelado de Datos
  + Modelado de Componentes
  + Modelado de Flujos de Trabajo (Workflows)

# Unidad 2

## El proceso unificado del desarrollo de software

Es un proceso **orientado a objetos**, guiado por **casos de uso**, centrado en la **arquitectura** y con un ciclo de vida **iterativo** **e** **incremental**.

* Guiado por **CASOS DE USO**
* Centrado en la **ARQUITECTURA**
* Ciclo de vida **ITERATIVO E INCREMENTAL**

Debe ofrecer un marco de trabajo genérico que unifique la **parte dinámica**, la **parte estática**, el **ciclo de vida** y la **interfaz**.

Utiliza **UML** en su notación, herramientas como "**visio**" o "**rational** **rose**" y sigue el "**proceso** **unificado** **de** **desarrollo** **de** **Rational**"

### Guiado por casos de uso

* Los sistemas se crean para dar servicio a los usuarios.
* Qué REQUISITOS se necesitan
* Un CASO de USO es una pieza de FUNCIONALIDAD de un sistema que le proporciona a algún USUARIO un RESULTADO o VALOR.
* Todos juntos constituyen el modelo de casos de uso (MCU)
* FUNCIONALIDAD COMPLETA PARA TODOS LOS USUARIOS

Los casos de uso:

* Capturan requisitos
* Se especifican (analizan)
* Se diseñan
* Se implementan
* Se prueban

### Centrado en la arquitectura

La arquitectura de un sistema de software es un extracto de los modelos del sistema que da una idea de qué forma tiene el sistema completo

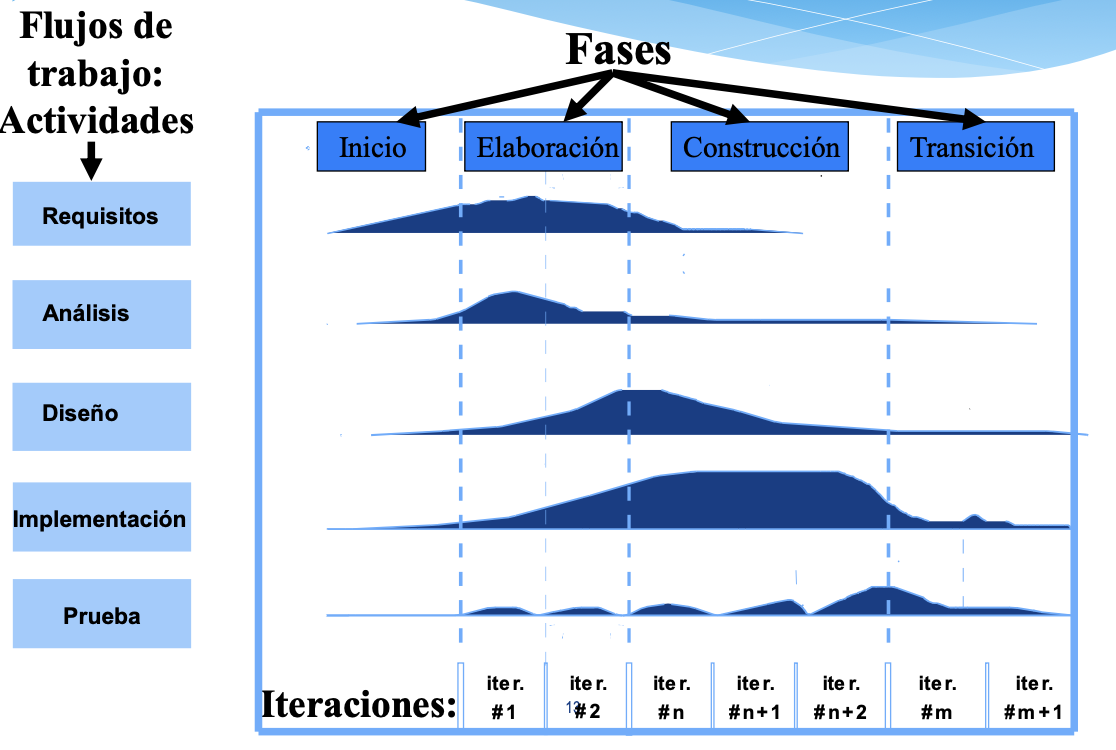
### El ciclo de vida del proceso unificado

Un ciclo de vida se repite a lo largo del tiempo. Tras cada ciclo de vida se produce una nueva versión del producto.

Un ciclo de vida se divide en fases, cada una de estas se divide en iteraciones, en cada iteración se realizan flujos de trabajo.

Las fases de un proyecto son:

* Inicio: definir el alcance del proyecto.
* Elaboración: planificar el proyecto y elaborar una arquitectura base.
* Construcción: construir el sistema.
* Transición: transición a los usuarios.



### Iniciación

* ¿Qué va a hacer el sistema para cada uno de sus usuarios principales?
  + Un MCU simplificado con los CU más críticos
* ¿Cómo sería la arquitectura para un sistema como ese?
  + Borrador con los subsistemas principales
* ¿Cuál es el plan y cuánto va a costar desarrollar el producto?
  + Identificar los riesgos principales y priorizarlos, planificar elaboración y presupuesto aproximado

### Elaboración

* Especificar en detalle los CU + críticos
* Diseñar la arquitectura
  + Mediante vistas de todos los modelos del SI
  + Vista arquitectónica de MCU, M. Análisis, M. Diseño, M.
    - Implementación (con los componentes que demuestran que la arquitectura es ejecutable) y M. Distribución.
* Al final de esta fase se debe poder planificar las actividades y estimar los recursos para poder completar el proyecto.
  + ¿Son los CU, arquitectura y planes lo suficientemente estables y los riesgos bajo control suficiente para firmar un contrato para terminar el trabajo de desarrollo?

### Construcción

* Se construye el producto. En esta fase:
  + La arquitectura se completa para construir un sistema bien cimentado
  + La visión evoluciona hasta convertirse en un producto preparado para los usuarios
  + Es donde se gastan la mayoría de los recursos
  + La arquitectura del sistema es estable. Sin embargo, se pueden realizar cambios mínimos a la misma.
  + ¿El producto se ajusta suficientemente a las necesidades de los usuarios como para enviarselo ya?

### Transición

* El producto se encuentra en fase beta
  + Un grupo reducido de usuarios experimentados prueba el producto e informa de los defectos y deficiencias y sugieren mejoras.
  + Los desarrolladores corrigen las deficiencias e incorporan algunas de las mejoras propuestas en una versión para un grupo de usuarios mayor.
  + En esta fase se encuentran actividades como la venta, formación de los usuarios, ofrecimiento de ayuda en línea y corrección de defectos descubiertos tras la implantación. Los defectos: (1) los que justifican la aparición de una nueva versión del sistema, (2) los que se pueden dejar para la siguiente versión que se cree.

# Unidad 3

## ¿Qué es un Requerimiento?

Un requerimiento de software define las funciones, capacidades o atributos de cualquier sistema de software.

Un requerimiento de software define tanto los servicios que un sistema de software debe ofrecer como sus limitaciones.

### Representan

* Factores de calidad del sistema.
* Lo que se requiere implementar.
* Una descripción de cómo el sistema deberá comportarse (dominio, restricciones, operaciones, atributos y propiedades)
* Un problema por resolver.

## Características de los requerimientos

* Estar por escrito
* Descritos como una característica del sistema a entregar.
* Necesarios
* No ambiguos
* Completos
* Correctos
* Consistentes
* Verificables
* Viables
* Priorizables
* Precisos

## Tipos de requerimientos

* Funcionales: describen servicios o funciones.
* No-funcionales: son un límite en el sistema o en el proceso de desarrollo.
* De Dominio: se obtienen de el dominio de la aplicación del sistema y que reflejan sus características.

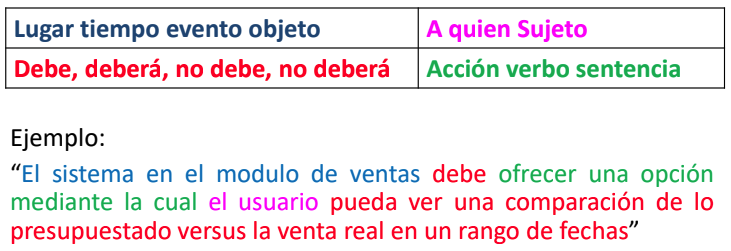
## Identificación de requerimientos

Tareas que facilitan la identificación de requerimientos:

* Analice los documentos “fuentes” (resultados de las reuniones con los usuarios).
* Extraiga las sentencias que contienen las palabras: DEBERA, DEBE.
* Identifique sentencias que impliquen requerimientos.
* Identifique sentencias de tipo VERBOS, ACCIONES y un RESULTADO.
* Haga una lista de FUNCIONALES y NO FUNCIONALES.

## Redacción de un requerimiento

***Todos los requerimientos deben tener el mismo estilo de redacción.***



## Problemas con los Requerimientos

* El proceso de requerimientos no es fácil.
* No es simplemente “tomar nota” de las necesidades del cliente.
* Es un proceso de comunicación.
* Es un proceso de negociación.
* No hay acuerdos en los requerimientos.
* Los requerimientos no se han priorizado.
* Requerimientos incompletos.
* Requerimientos contradictorios.
* Requerimientos ambiguos.
* Requerimientos de desarrollador. (Presunciones de diseño)

***¿Cómo solucionar estos problemas?***

*Llevar un proceso disciplinado que busque determinar y solucionar los diferentes problemas de los requerimientos, y un sistema de especificaciones que posibilite comunicar y negociar los requerimientos eficientemente con los usuarios.*

## Aplicamos el proceso

* *Definir el objetivo del sistema*
* *Definir subsistemas del sistema*
* *Identificar requerimientos de usuario*
* *Redactar requerimientos del sistema*

## Diagramas de casos de uso

**Nombre del caso de uso:** los mismos deben estar redactados desde la perspectiva del sistema. El caso de uso agrupa acciones que genera el sistema. El actor asociado al caso de uso activa estas acciones.

**Actores y nombre de los actores**: recuerden que el actor representa un rol dentro del dominio

**Asociaciones entre casos de uso:** relaciones de include (obligatorias) y extend (opcionales)

**Nivel de abstracción aplicado**

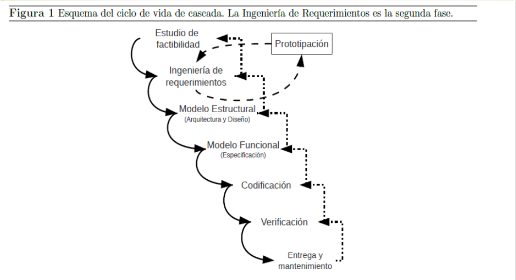
Especificaciones de caso de uso trazo fino, permiten validar :

* Si se muestra la interacción entre usuario y sistema.
* Si las actividades son las indicadas para cumplir el objetivo del caso de uso, y no incluyen actividades extras que quedan fuera del alcance del caso de uso.
* Si se describe el curso normal y los cursos alternativos asociados.
* Si se mantiene la trazabilidad con el modelo de caso de uso.

# Unidad 4

## Ingeniería de Requerimientos

Es el proceso para establecer los servicios que el sistema deberá proveer y las restricciones bajo las cuales deberá operar y ser desarrollado.



## Requerimiento

Un requerimiento es una característica del sistema o una descripción de algo que el sistema es capaz de hacer con el objeto de satisfacer el propósito del sistema.

## Requerimiento Funcional

Un requerimiento funcional describe una interacción entre el sistema y su ambiente. Los requerimientos funcionales describen cómo debe comportarse el sistema ante un estímulo.

**Los requerimientos funcionales incluyen:**

1. Chequeos de validación de la entrada

2. Secuencia exacta de las operaciones

3. Respuestas a situaciones anormales, incluyendo:

a) Desborde

b) Comunicaciones

c) Manejo de errores y recuperación

4. Efecto de los parámetros del sistema

5. Relación entre las salidas y las entradas, incluyendo:

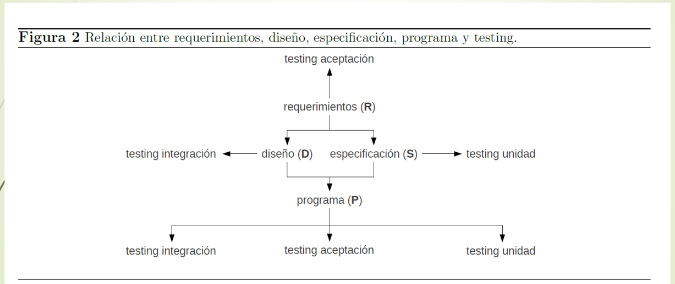
a) Secuencias de entrada/salida

b) Fórmulas para convertir entrada en salida

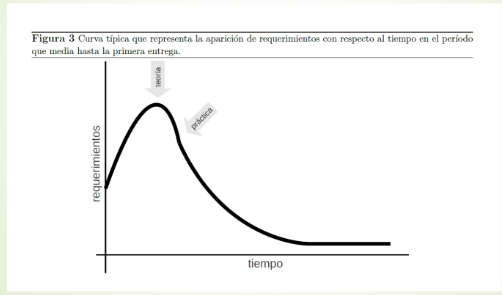
## Requerimientos no funcionales

Un requerimiento no funcional es una restricción sobre el sistema o su proceso de producción.

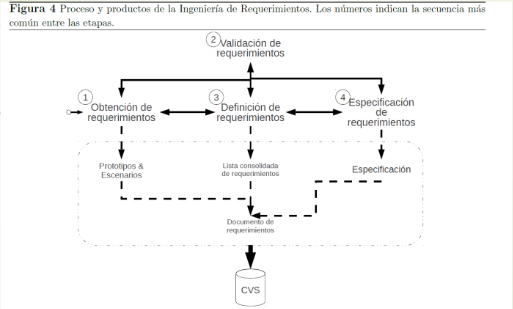
## Relación entre los requerimientos y el resto de las etapas del proceso



* El diseño y la especificación del sistema se obtienen a partir de los requerimientos.
* El diseño y la especificación se utilizan para implementar el programa.
* Finalmente, los requerimientos son el documento esencial para llevar adelante el testing de aceptación.
* Cuanto mayor calidad tengan los requerimientos mayores son las posibilidades de que el proyecto sea un éxito.
* Se podrá iniciar la fase de Arquitectura y Diseño (o la etapa de Definición De requerimientos) cuando se desacelera la cantidad de requerimientos validados que pide el cliente.
* Esto ocurre cuando cuando el cociente entre la aparición de requerimientos y el tiempo es bajo durante un periodo de tiempo razonable.



## Proceso de la Ingeniería de requerimientos



## Obtención de requerimientos

En esta etapa se obtienen los requerimientos del sistema a través de la observación de sistemas existentes y del entorno donde se instalará el sistema.

* Reuniones con los interesados.
* Generación de prototipos.
* Definición de escenarios.

## Validación de requerimientos

Validación de requerimientos. Los requerimientos solo pueden ser validados por el cliente. Por lo tanto, una vez que se ha obtenido un requerimiento (preguntando al cliente) se le debe preguntar si ese es el requerimiento que expresó.

## Definición de requerimientos

En esta etapa la información recolectada en la etapa anterior se vuelca en un documento consolidado, organizado y estructurado.

Priorización de requerimiento:

* Requerimientos que deben ser absolutamente satisfechos.
* Requerimientos que son muy deseables pero no indispensables.
* Requerimientos que son posibles, pero que podrán eliminarse.

## Calidad de los requerimientos

* La calidad de un requerimiento se refiere en general a que tan perdurable en el tiempo será tal y como está descrito en este momento.
* Un requerimiento es de mala calidad si el cliente no puede expresarlo claramente o directamente duda o desconoce que es lo que realmente necesita o quiere.
* La calidad de un requerimiento está directamente relacionada con el grado de validación que aún necesita.

Un requerimiento debe ser **cohesivo**, **completo**, **consistente**, **correcto**, **actualizado**, **observable** **externamente**, **factible**, **no** **ambiguo** y **validable**

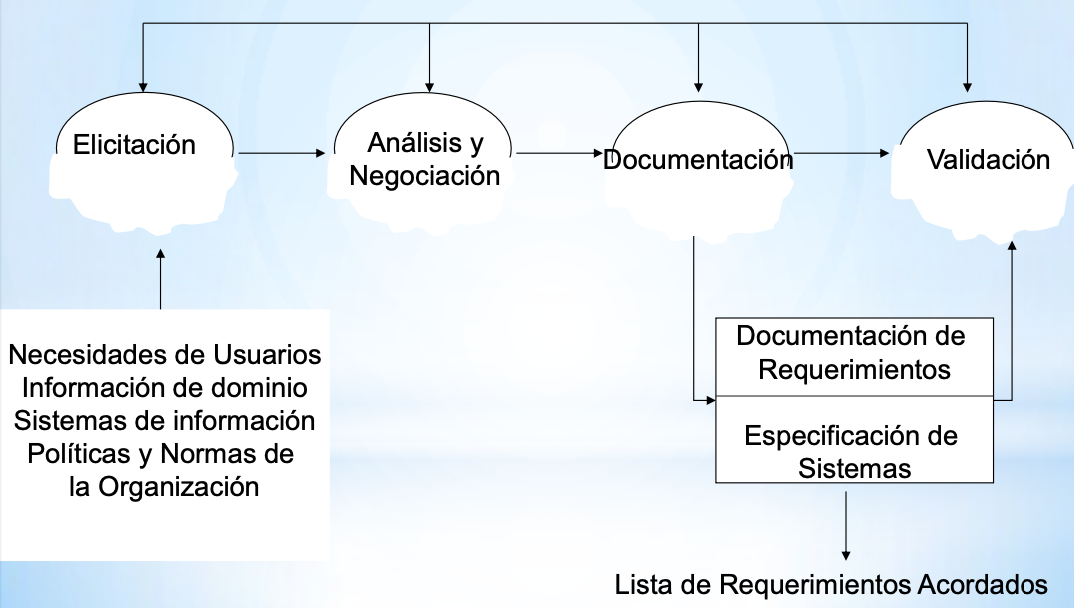
## Especificación de requerimientos

En pocas palabras en esta etapa la Lista consolidada de requerimientos se escribe desde la perspectiva del desarrollador. Esta etapa puede incluirse en la fases denominadas Modelo Estructural o Modelo Funcional en el modelo de ciclo de vida de cascada

## Productos del proceso de la ingeniería de requerimientos

* **Prototipos:** Debe ser la forma de interacción casi excluyente entre los ingenieros de requerimientos y el cliente, durante esta fase.
  + Nos referimos concretamente a prototipos desechables. Un prototipo desechable es un programa que imita al menos algunos de los requerimientos funcionales pero que luego de que los requerimientos se han validado, el programa se descarta.
* **Escenarios:** Es muy complejo o costoso, y muchas veces imposible, prototipar los atributos de calidad de un sistema (tales como disponibilidad, seguridad, testeabilidad, etc.). En consecuencia, en lugar de prototipos, se utilizan escenarios de atributos de calidad [BCK03] como técnica efectiva para determinar con suficiente precisión cual es el significado que el usuario le asigna a las cualidades del sistema.
* **Lista consolidada de requerimientos:** Es un documento que consolida, organiza y estructura la información recolectada durante la etapa de obtención de requerimientos.
  + El vocabulario y las notaciones que se utilicen para escribir esta lista deben ser estrictamente las del cliente o las del dominio de aplicación
  + No deberá incluir términos tales como: tabla, registro, campo, lista enlazada, función, proceso o procedimiento (en su acepción informática), variable, base de datos, archivo, directorio, sistema operativo, módulo, objeto, clase, hilo de control, servidor, HTTP, web service, etc.
  + Frases tales como el “sistema deberá implementar un módulo..." o un webservice tomará los datos de la tabla personas y los enviará vía SOAP al servidor Web..." están totalmente fuera del alcance de este documento.
  + Frases tales como “el sistema deberá implementarse en C++..." o “el sistema deberá procesar 10.000 transacciones por minuto..." pueden estar dentro del alcance de este documento, como requerimientos no funcionales.

## Modelos del proceso de IR



### Elicitación de Requerimientos

Significa descubrir requerimientos del sistema utilizando:

* Entrevistas con los stakeholders
* Documentación del sistema
* Conocimiento del dominio
* Estudio de mercado

### Análisis y Negociación

Los requerimientos son analizados en detalle y los stakeholders se ponen de acuerdo para determinar los requerimientos válidos.

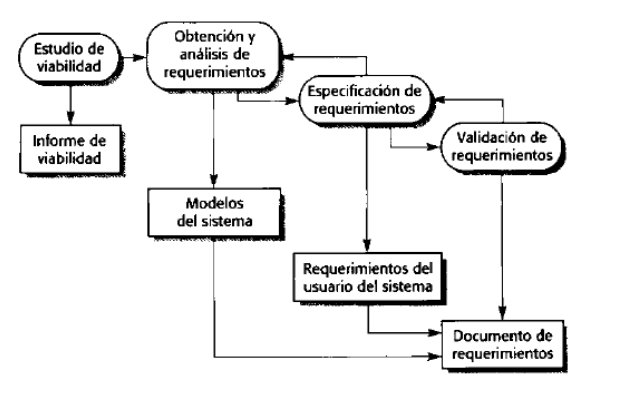
### Documentación de Requerimientos

Se documentan los requerimientos acordados utilizando lenguaje natural y diagramas que faciliten la comprensión de todos los stakeholders.

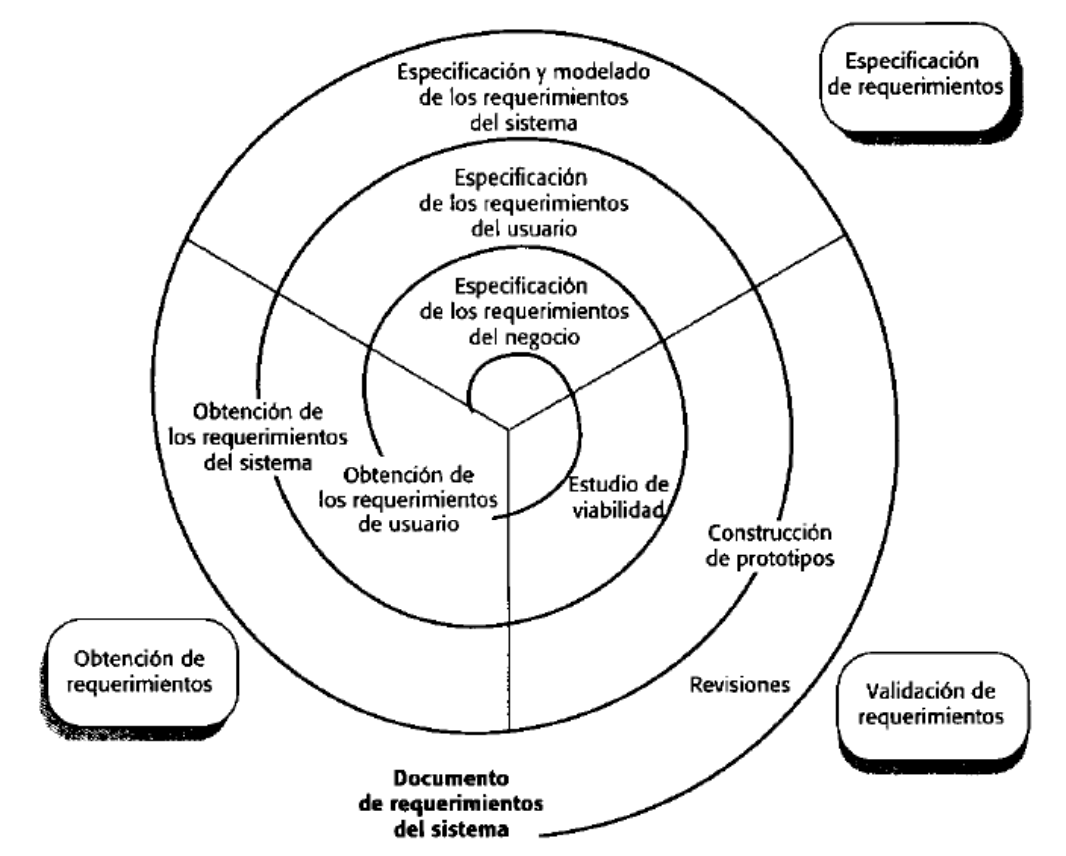
### Validación de Requerimientos

Se realiza una cuidadosa revisión de los requerimientos evaluando consistencia y completitud. Este proceso tiene la intención de detectar errores antes de que los requerimientos se tomen como base para el desarrollo

## El proceso de IR



## Modelo en espiral para los procesos de IR



## Estudio de Viabilidad

