



# Ingeniería de Software I

Análisis Estructurado

Modelos de Proceso

# Técnicas de Especificación de Requerimientos

---

Análisis Estructurado

# Análisis estructurado

---

- » Para entender los requerimientos, se debe poder reconocer además como se mueven los datos, los procesos o transformaciones que sufren dichos datos y sus resultados.
- » La elicitación proporciona una descripción verbal del sistema, una descripción visual puede consolidar la información.



# Análisis Estructurado

---

- La técnica de análisis estructurado permite lograr una representación gráfica que permite lograr una comprensión mas profunda del sistema a construir y comunicar a los usuarios lo comprendido.
- La notación no especifica aspectos físicos de implementación.
- Hace énfasis en el procesamiento o la transformación de datos conforme estos pasan por distintos procesos.

# Análisis estructurado – Modelado funcional y flujo de la información

---

## » Modelado de funciones del sistema

### **Diagrama de Flujo de Datos (DFD)**

- Es una herramienta que permite visualizar un sistema como una red de procesos funcionales, conectados entre sí por “conductos” y almacenamientos de datos.
- Representa la transformación de entradas a salidas y es también llamado diagrama de burbujas.
- Es una herramienta comúnmente utilizada por sistemas operacionales en los cuales las funciones del sistema son de gran importancia y son más complejas que los datos que éste maneja.

# Análisis estructurado – Modelado Funcional y Flujo de la información

---

Se utiliza un rectángulo para representar una **entidad externa**, esto es, un elemento del sistema (por ejemplo, un elemento hardware, una persona, otro programa) u otro sistema que produce información para ser transformada por el software, o recibe información producida por el software.

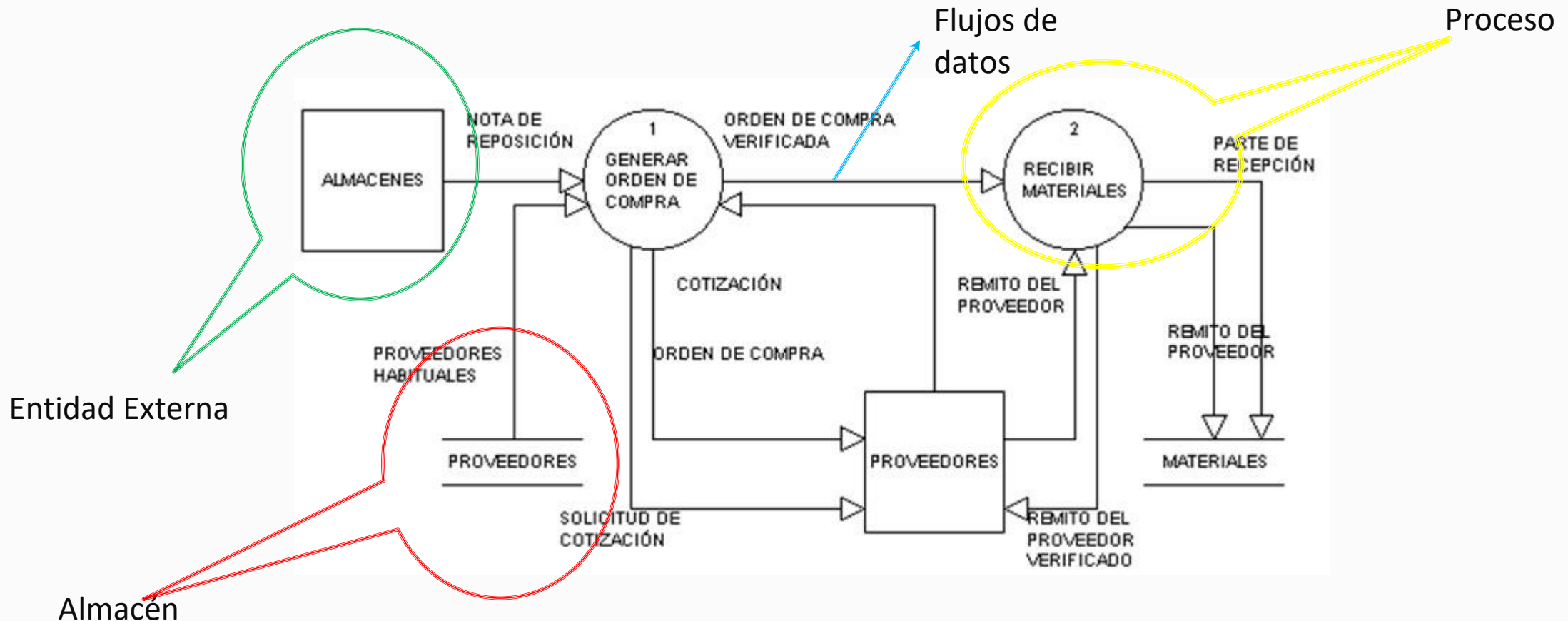
Un círculo (también llamado burbuja) representa un **proceso** o *transformación* que es aplicado a los datos (o al control) y los modifica.

Una flecha representa uno o más **elementos de datos** (objetos de dato).

Un rectángulo abierto (lado izquierdo y derecho) que representa un **almacén de datos**



# Análisis estructurado – Modelado funcional y flujo de la información



# Desarrollo de DFDs

---

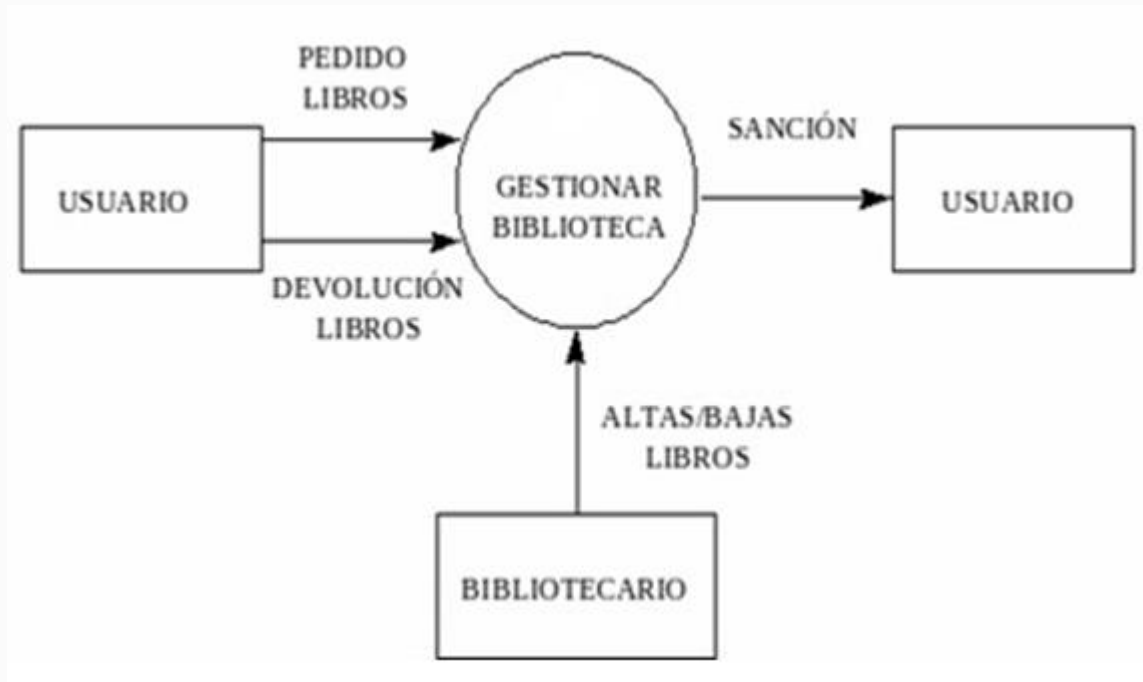
Se debe visualizar desde una perspectiva jerárquica de arriba hacia abajo.

Pasos :

1. Redactar la lista de actividades de la organización para determinar:
  - Entidades externas
  - Flujos de datos
  - Procesos
  - Almacenes de datos
2. Crear un diagrama de contexto que muestre las entidades externas y los flujos de datos desde y hacia el sistema.
3. Dibujar el Diagrama 0 (siguiente nivel), con procesos generales y los almacenes correspondientes
4. Dibujar un diagrama hijo por cada uno de los procesos del Diagrama 0



# Diagrama de contexto

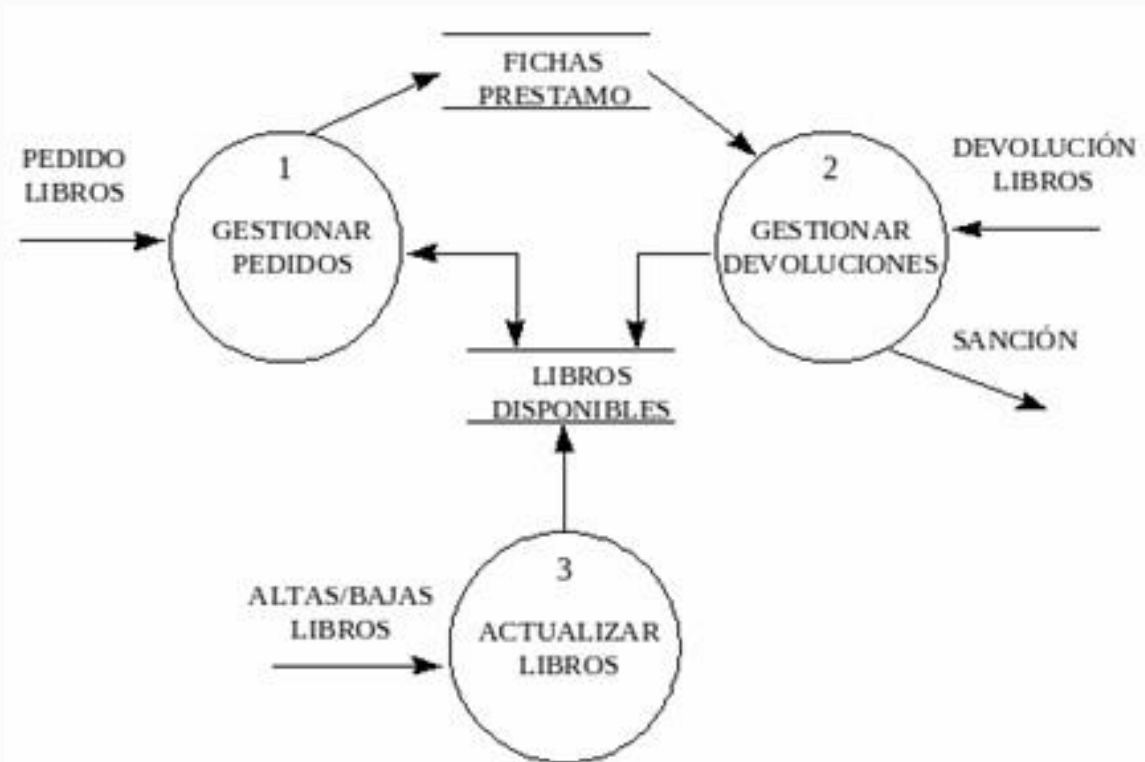


Ambiente

Se muestra un panorama global que muestre las entradas básicas y las salidas

Es el nivel más alto en un DFD y contiene un solo proceso que representa a todo el sistema

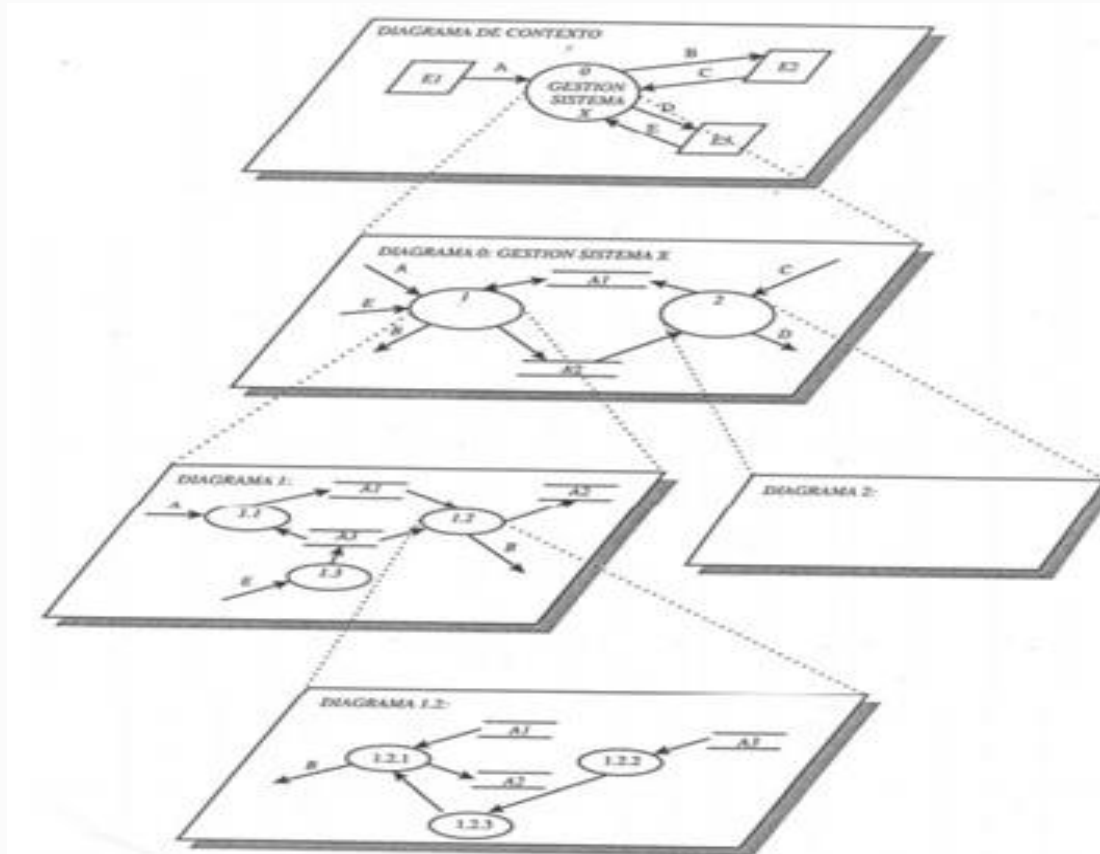
# Nivel 0



Es la ampliación del Diagrama de contexto.

Las entradas y salidas del Diagrama de contexto permanecen, sin embargo, se amplía para incluir hasta 9 procesos (como máximo) y mostrar los almacenes de datos y nuevos flujos.

# Nivelación de un DFD



Cada proceso se puede a su vez ampliar para crear un diagrama hijo más detallado.

Las entradas y salidas del proceso padre permanecen, sin embargo, pueden aparecer nuevos almacenes de datos y nuevos flujos.

# Modelos de Proceso

---

# Proceso

---

- ❖ Cuando proveemos un servicio o creamos un producto, siempre seguimos una secuencia de pasos para realizar un conjunto de tareas.
- ❖ Las tareas son realizadas usualmente en el mismo orden.
- ❖ Por ejemplo, no se puede cocinar una torta antes de que todos los ingredientes sean mezclados.
- ❖ Se puede pensar que un “conjunto ordenado de tareas” como un proceso.



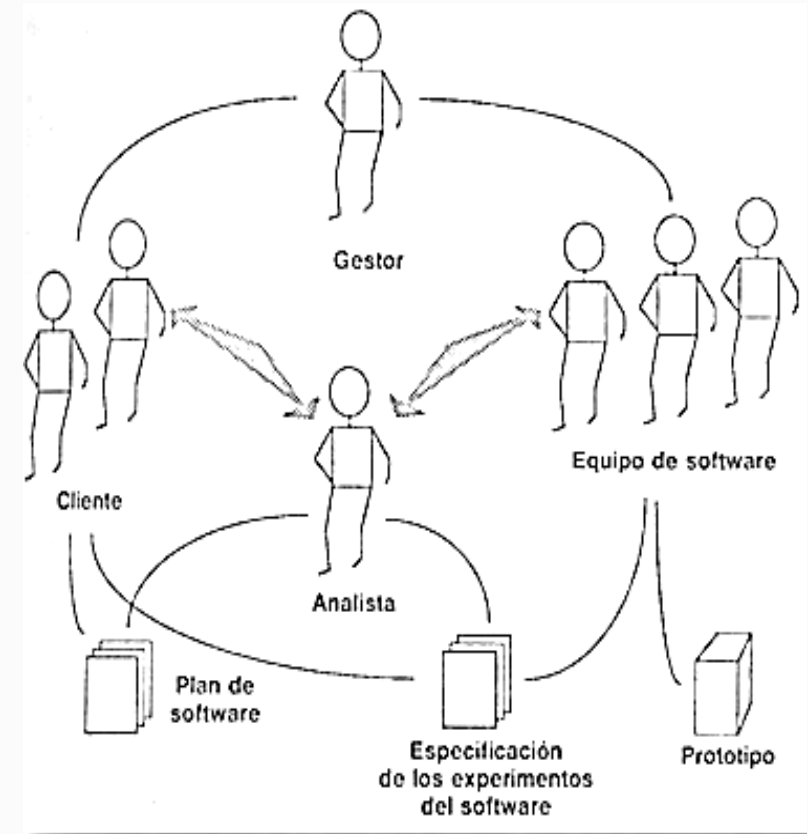
# ¿Qué es un proceso de software?

- ❖ Es un conjunto de actividades y resultados asociados que producen un producto de software.



# ¿Qué es un modelo de proceso de software?

Marco de referencia que contiene los procesos, las actividades y las tareas involucradas en el desarrollo, la explotación y el mantenimiento de un producto de software, abarcando la vida del sistema desde la definición de los requisitos hasta la finalización de su uso (*norma ISO 12207-1*) [ISO/IEC, 1995]



# Modelos de Proceso

---

## Características

- ❖ Establece todas las actividades.
- ❖ Utiliza recursos, está sujeto a restricciones y genera productos intermedios y finales.
- ❖ Puede estar compuesto por subprocesos.
- ❖ Cada actividad tiene entradas y salidas definidas.
- ❖ Las actividades se organizan en una secuencia.
- ❖ Existen principios que orientan sobre las metas de cada actividad.
- ❖ Las restricciones pueden aplicarse a una actividad, recurso o producto.



# Modelos de Proceso

---

## **Ciclo de vida**

Proceso que implica la construcción de un producto

## **Ciclo de vida del Software**

Describe la vida del producto de software desde su concepción hasta su implementación, entrega, utilización y mantenimiento

## **Modelos de proceso de software**

Es una representación abstracta de un proceso del software.

Modelo de proceso

Paradigma de software

Ciclo de vida del software



Términos Equivalentes

# Modelos de Proceso

---

## Modelos **prescriptivos**

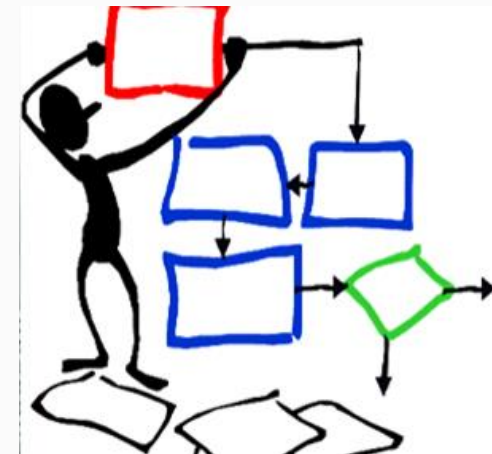
Prescriben un conjunto de elementos del proceso: actividades del marco de trabajo, acciones de la ingeniería del software, tareas, aseguramiento de la calidad y mecanismos de control.

Cada modelo de proceso prescribe también un “flujo de trabajo”, es decir de qué forma los elementos del proceso se interrelacionan entre sí.

## Modelos **descriptivos**

Descripción en la forma en que se realizan en la realidad

Ambos modelos deberían ser iguales



# Modelos de Proceso

---

## Modelos tradicionales

Formados por un conjunto de fases o actividades en las que e no tienen en cuenta la naturaleza evolutiva del software

- ❖ Clásico, lineal o en cascada
- ❖ Modelo en V
- ❖ Basado en prototipos

## Modelos evolutivos

Son modelos que se adaptan a la evolución que sufren los requisitos del sistema en función del tiempo

- ❖ En espiral
  - ❖ Evolutivo
  - ❖ Incremental
- Desarrollo por fases

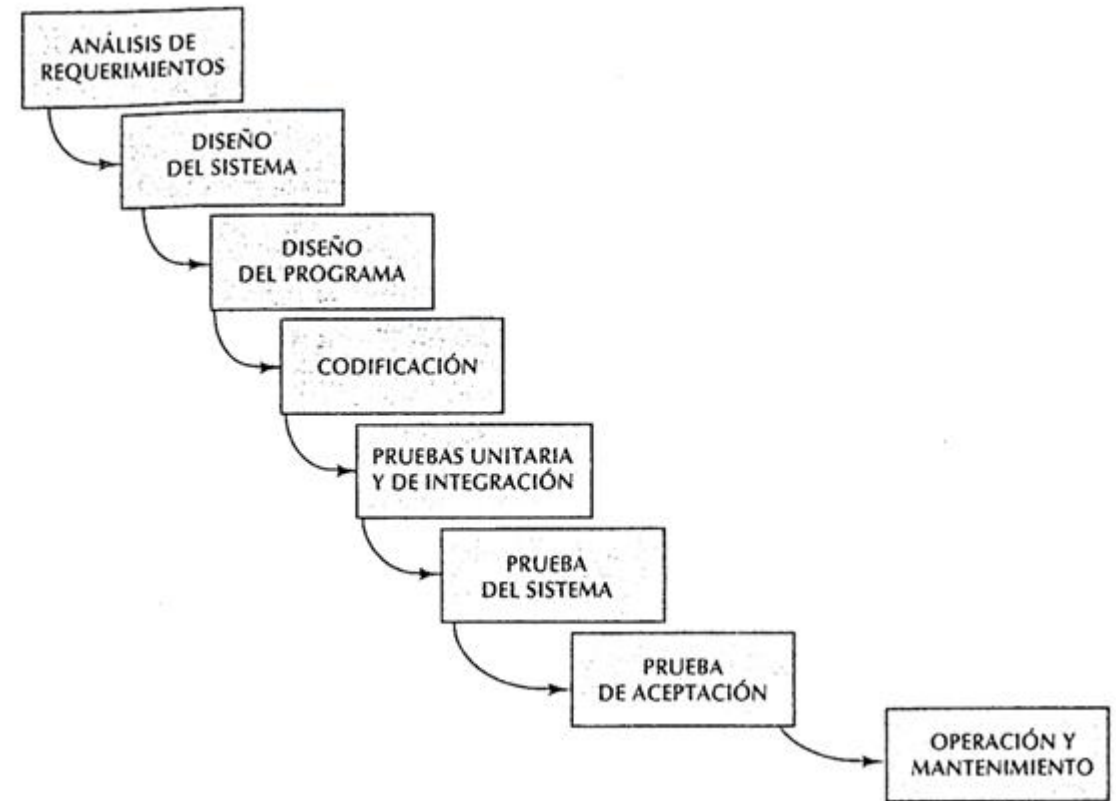
## Procesos ágiles

...



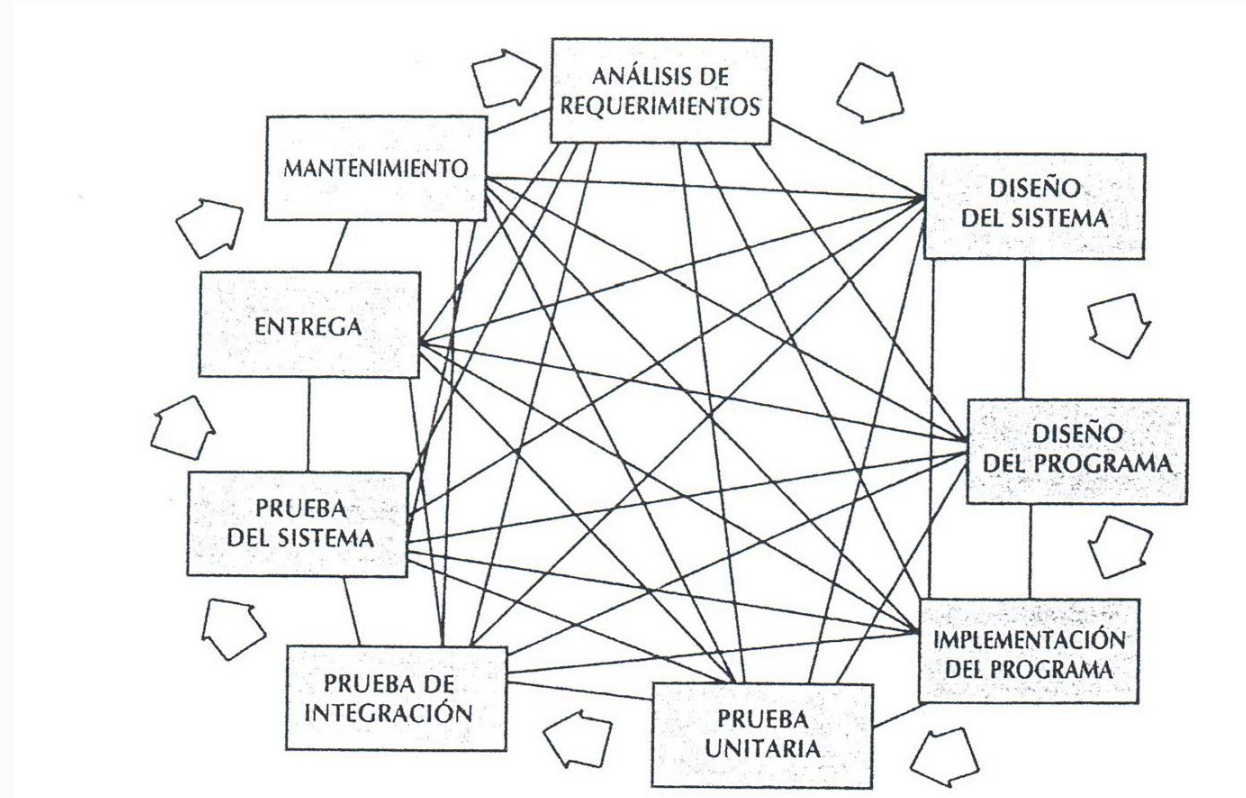
# Modelo en cascada

- ❖ Las etapas se representan cayendo en cascada
- ❖ Cada etapa de desarrollo se debe completar antes que comience la siguiente
- ❖ Útil para diagramar lo que se necesita hacer
- ❖ Su simplicidad hace que sea fácil explicarlo a los clientes



# Modelo de la realidad en comparación con cascada

❖ Modelo de la realidad (sin control entre las etapas)



# Modelo en cascada

---

## **Dificultades:**

No existen resultados concretos hasta que todo este terminado.

Las fallas más triviales se encuentran al comienzo del período de prueba y las más graves al final.

La eliminación de fallas suele ser extremadamente difícil durante las últimas etapas de prueba del sistema.

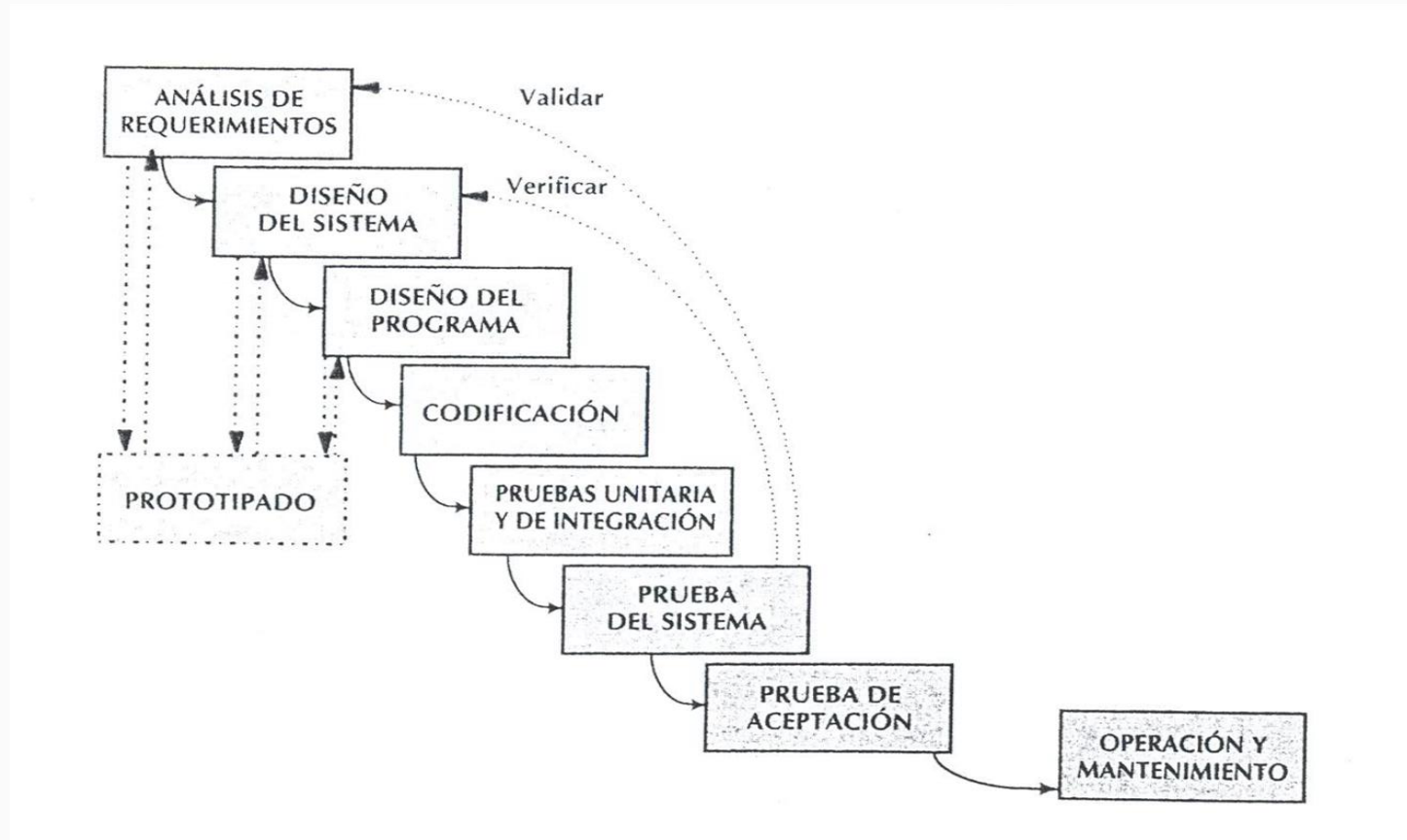
Deriva del mundo del hardware y presenta una visión de manufactura sobre el desarrollo de software.

La necesidad de pruebas aumenta exponencialmente durante las etapas finales.

"CONGELAR" una fase es poco realista.

Existen errores, cambios de parecer, cambios en el ambiente.

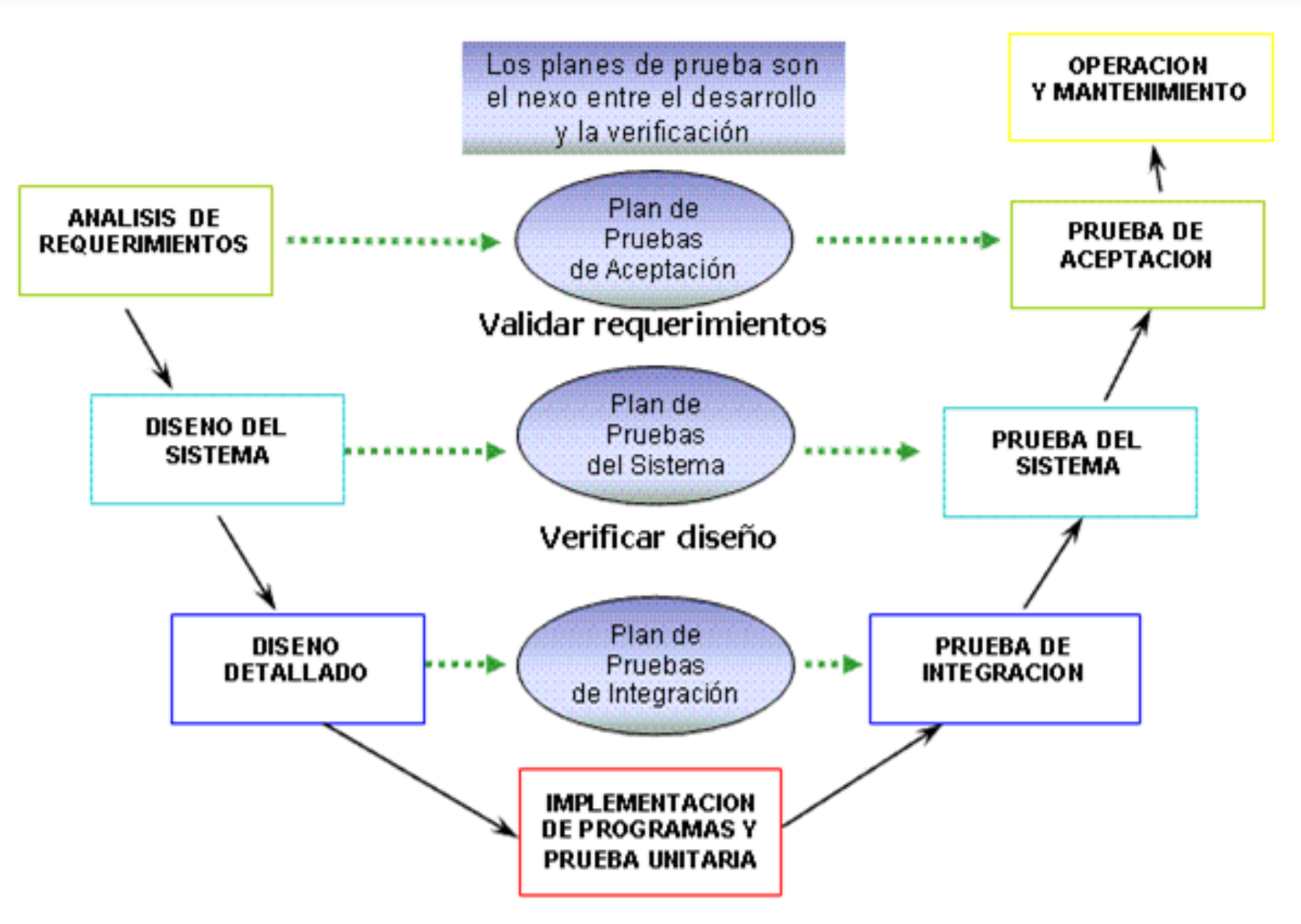
# Modelo en cascada con prototipo



»Pfleeger

# Modelo en V

- Demuestra cómo se relacionan los diferentes componentes del desarrollo de software.
- Sugiere que la prueba unitaria y programa
- La vinculación entre los lados de la verificación y validación, entorpeciendo la solución del problema.





# Modelo de prototipos

---

Un prototipo es un producto parcialmente desarrollado que permite que clientes y desarrolladores examinen algunos aspectos del sistema propuesto, y decidan si éste es adecuado o correcto para el producto terminado.

Esta es una alternativa de especificación para tratar mejor la incertidumbre, la ambigüedad y la volubilidad de los proyectos reales.



# Modelo de prototipos

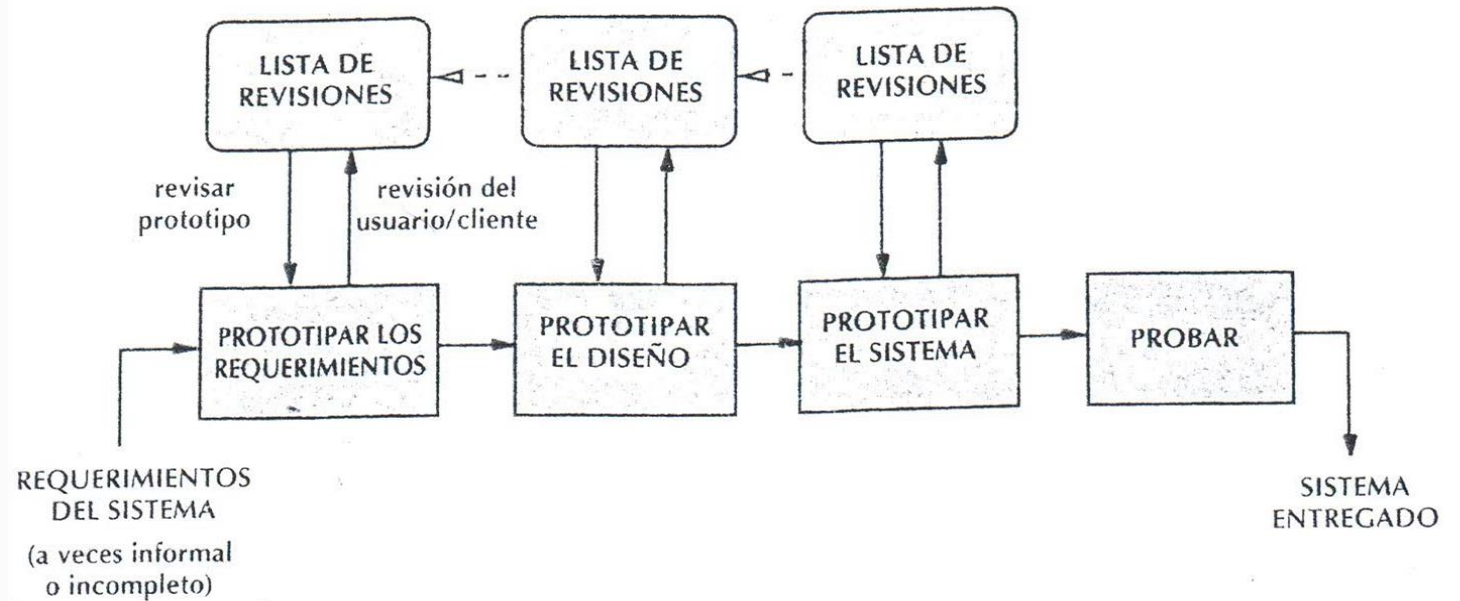
## Tipos Evolutivos

El objetivo es obtener el sistema a entregar.

Permite que todo el sistema o alguna de sus partes se construyan rápidamente para comprender o aclarar aspectos y asegurar que el desarrollador, el usuario y el cliente tengan una comprensión unificada tanto de lo que se necesita como de lo que se propone como solución

## Descartables

No tiene funcionalidad  
Se utilizan herramientas de modelado



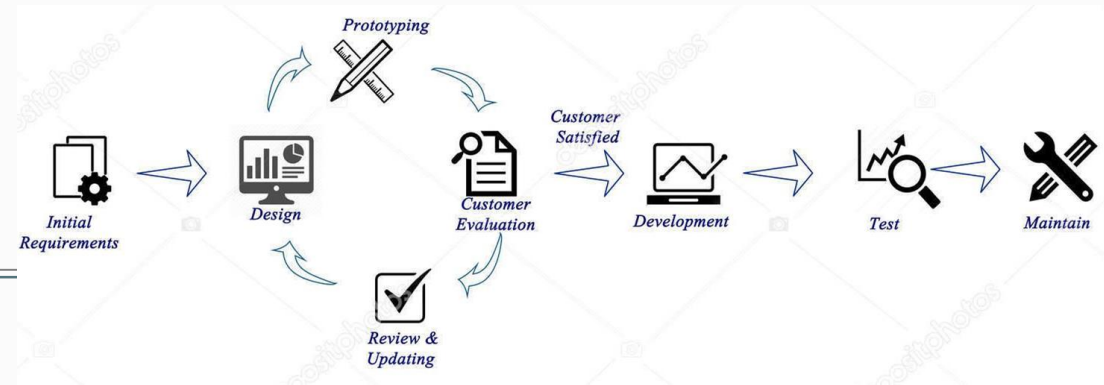
# Modelo de prototipos

---

	Descartable	Evolutivo
Enfoque de desarrollo	Rápido y sin rigor	Riguroso
Que construir	Solo las partes problemáticas	Primero las partes bien entendidas. Sobre una base sólida
Objetivo ultimo	Desecharlo	Lograr el sistema

# Modelo de prototipos

## Proyectos candidatos



- ❖ Usuarios que no examinarán los modelos abstractos
- ❖ Usuarios que no determinarán sus requerimientos inicialmente
- ❖ Sistemas con énfasis en los formatos de E/S más que en los detalles algorítmicos
- ❖ Sistemas en los que haya que explorar aspectos técnicos
- ❖ Si el usuario tiene dificultad al tratar con los modelos gráficos para modelar los requerimientos y el comportamiento
- ❖ Si se enfatiza el aspecto de la interfaz humana

# Modelo de prototipos

---

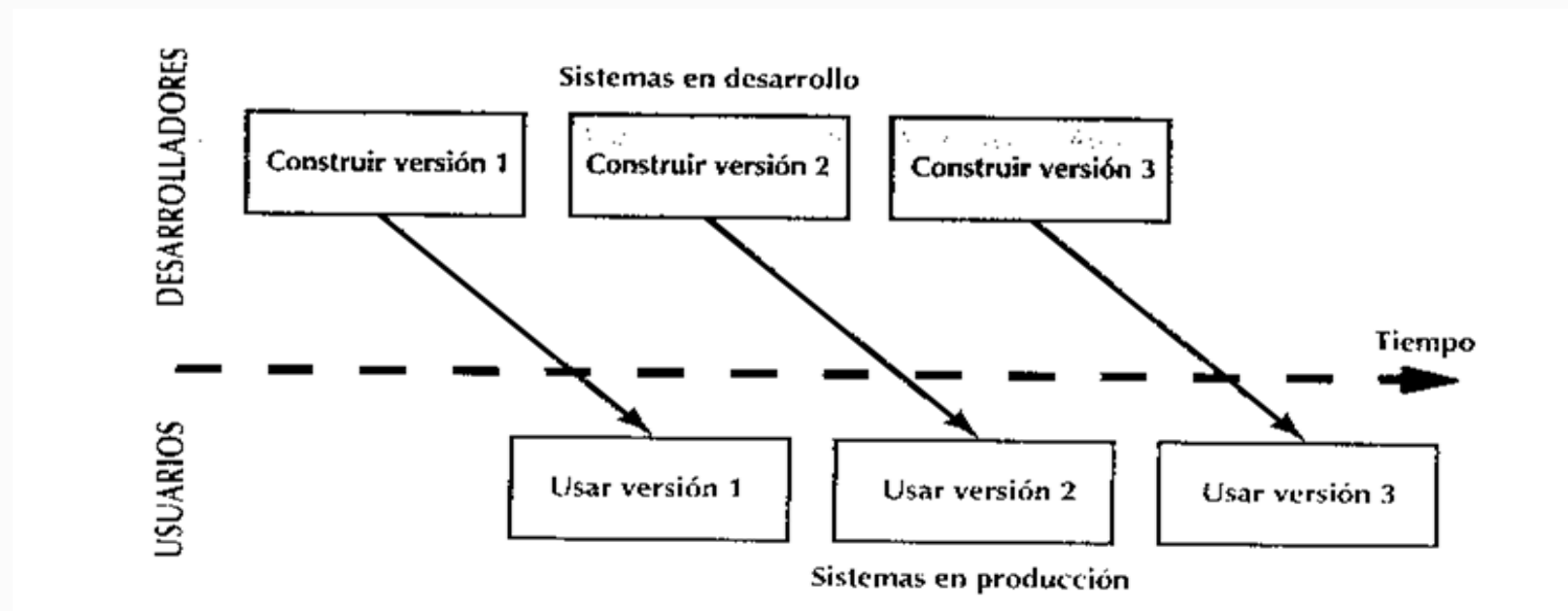
Para asegurar el éxito:

- ❖ Debe ser un sistema con el que se pueda experimentar
- ❖ Debe ser comparativamente barato ( $< 10\%$ )
- ❖ Debe desarrollarse rápidamente
- ❖ Énfasis en la interfaz de usuario
- ❖ Equipo de desarrollo reducido
- ❖ Herramientas y lenguajes adecuados



# Modelo de desarrollo por fases

Se desarrolla el sistema de tal manera que puede ser entregado en piezas. Esto implica que existen dos sistemas funcionando en paralelo: el sistema operacional y el sistema en desarrollo.



# Modelo de desarrollo por fases

---

## Tipos de modelos de desarrollo por fases

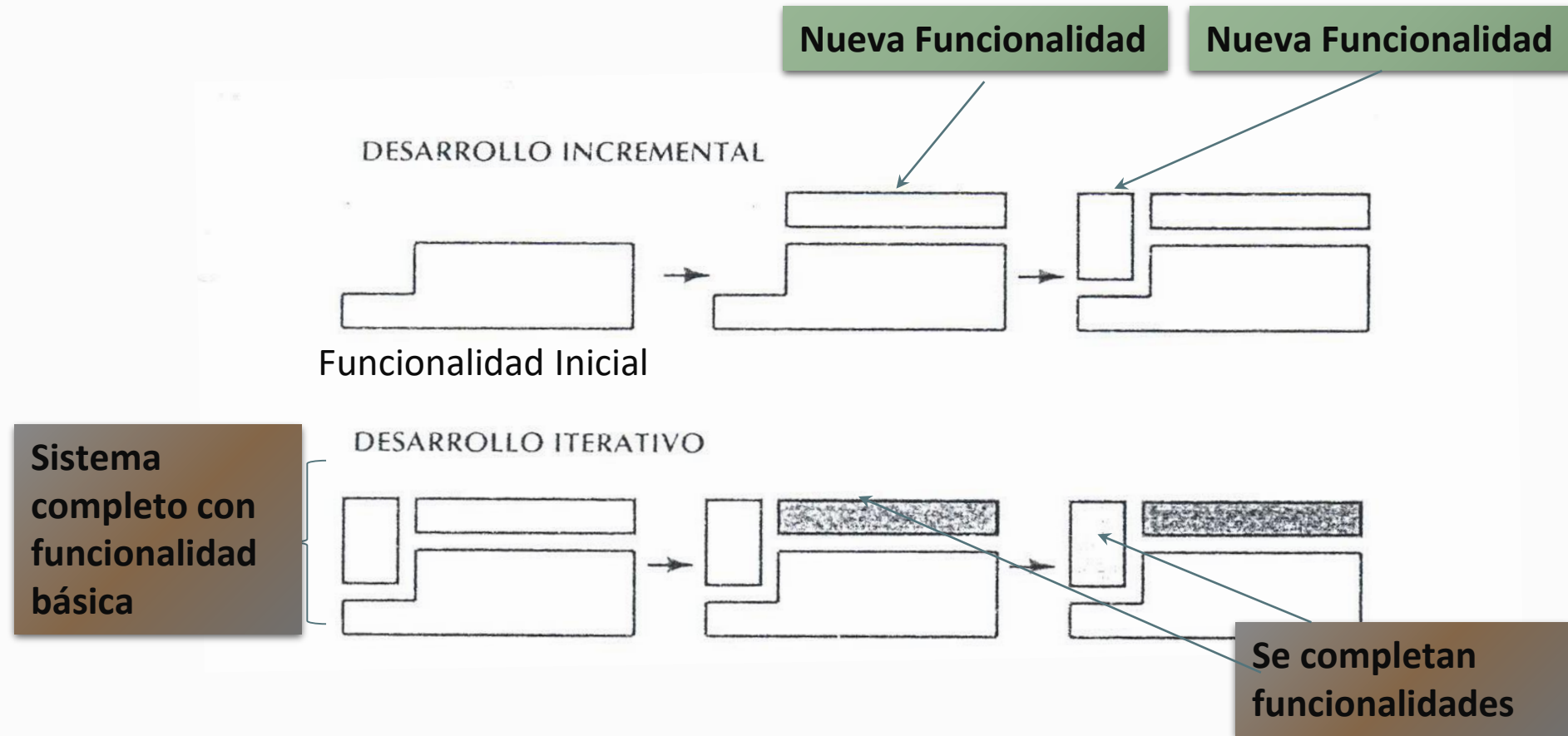
### **Incremental**

El sistema es particionado en subsistemas de acuerdo con su funcionalidad. Cada entrega agrega un subsistema.

### **Iterativo**

Entrega un sistema completo desde el principio y luego aumenta la funcionalidad de cada subsistema con las nuevas versiones.

# Modelos de desarrollo por fases



Pfleeger

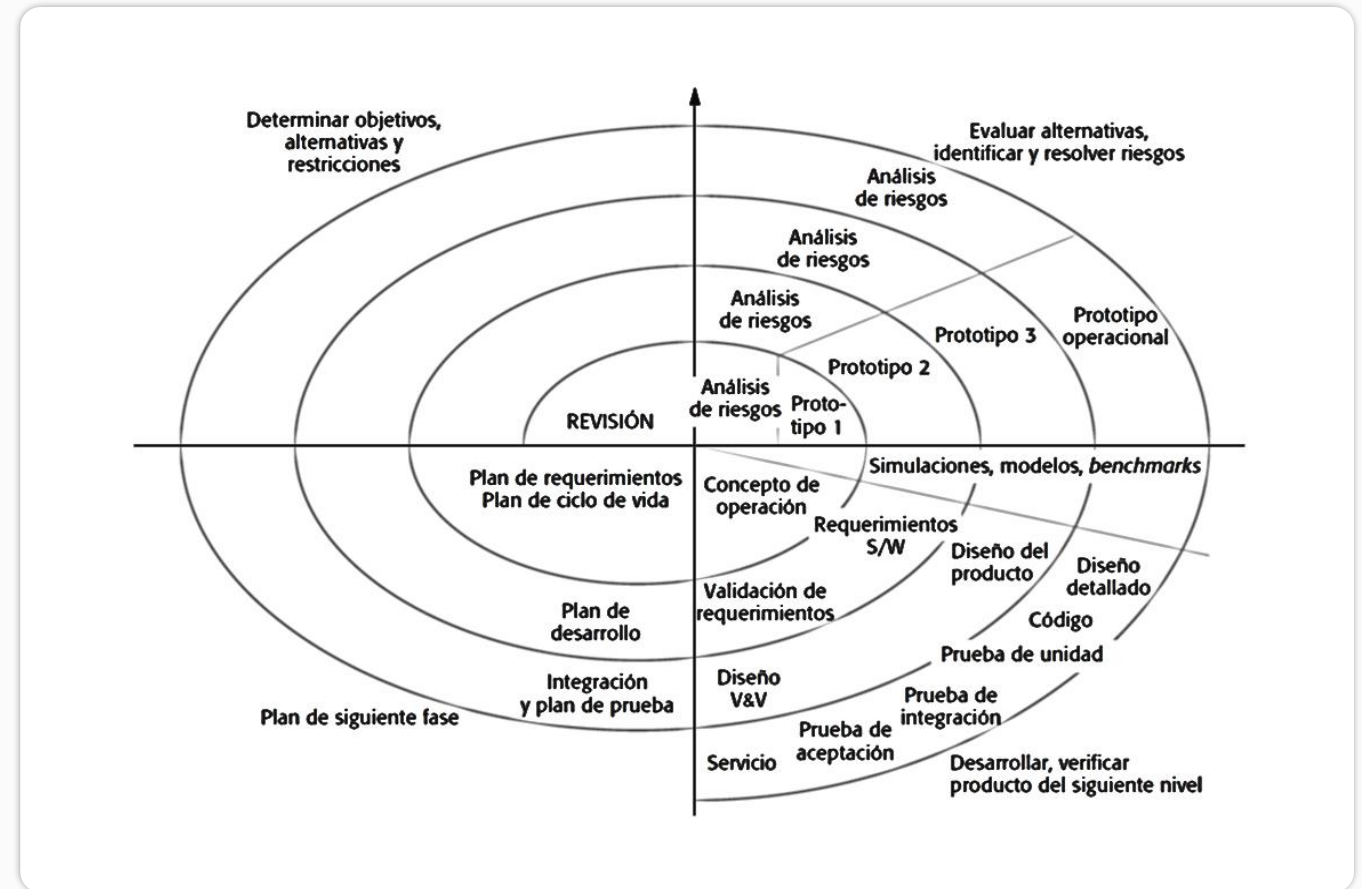


# Modelo en espiral (Boehm)

- ❖ Combina las actividades de desarrollo con la gestión del riesgo
- ❖ Trata de mejorar los ciclos de vida clásicos y prototipos.
- ❖ Incorpora objetivos de calidad
- ❖ Elimina errores y alternativas no atractivas al comienzo
- ❖ Permite iteraciones, vuelta atrás y finalizaciones rápidas
- ❖ Cada ciclo empieza identificando:
  - ❖ Los objetivos de la porción correspondiente
  - ❖ Las alternativas

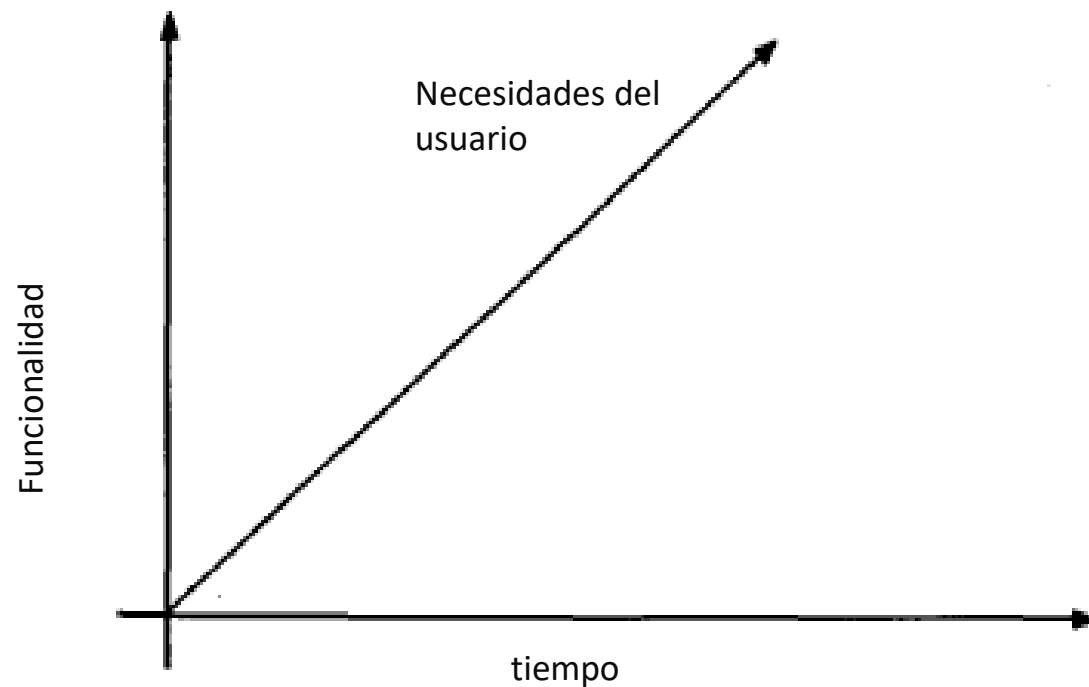
## Restricciones

- ❖ Cada ciclo se completa con una revisión que incluye todo el ciclo anterior y el plan para el siguiente



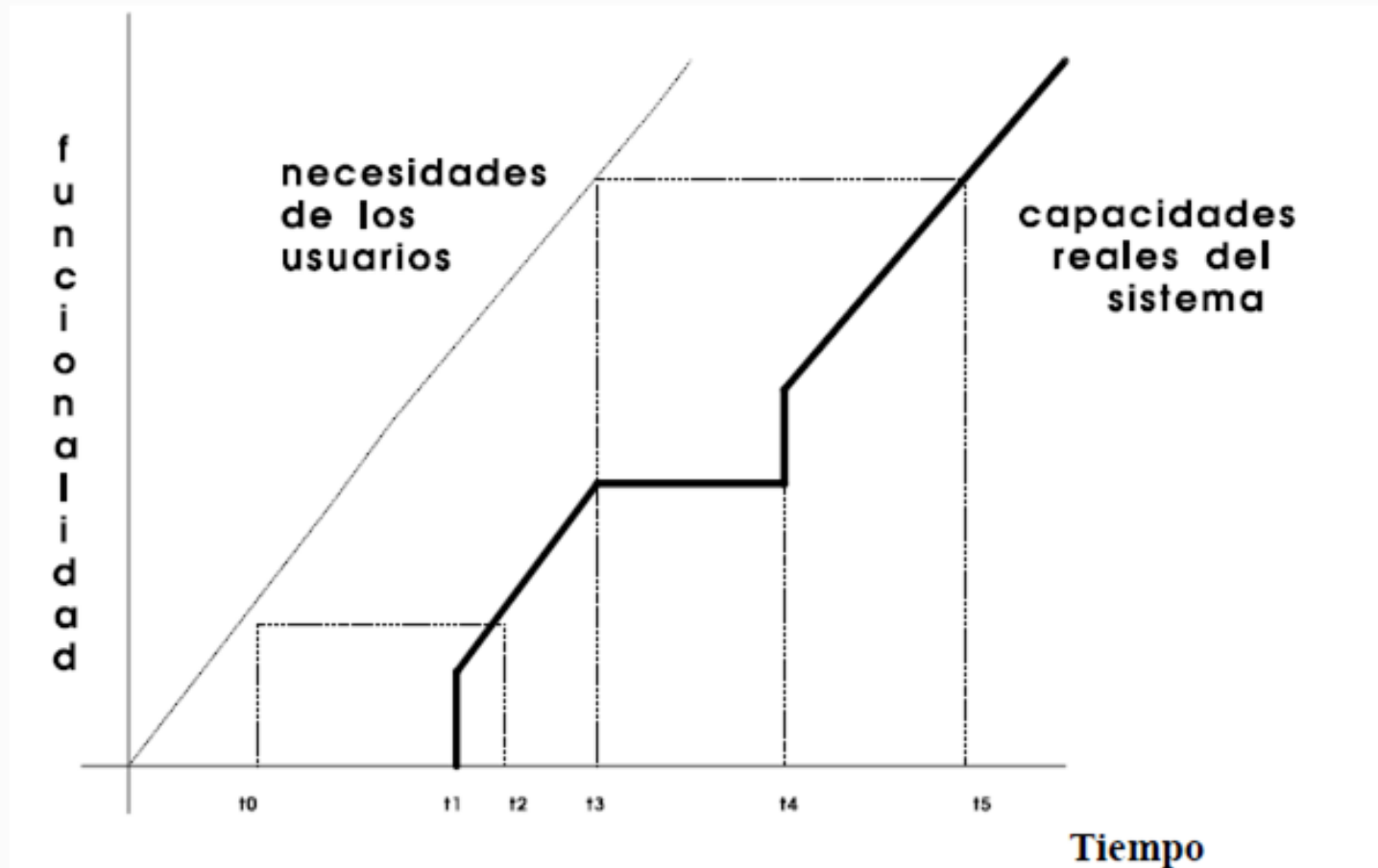
# Modelos de proceso – Análisis comparativo (Davis)

---



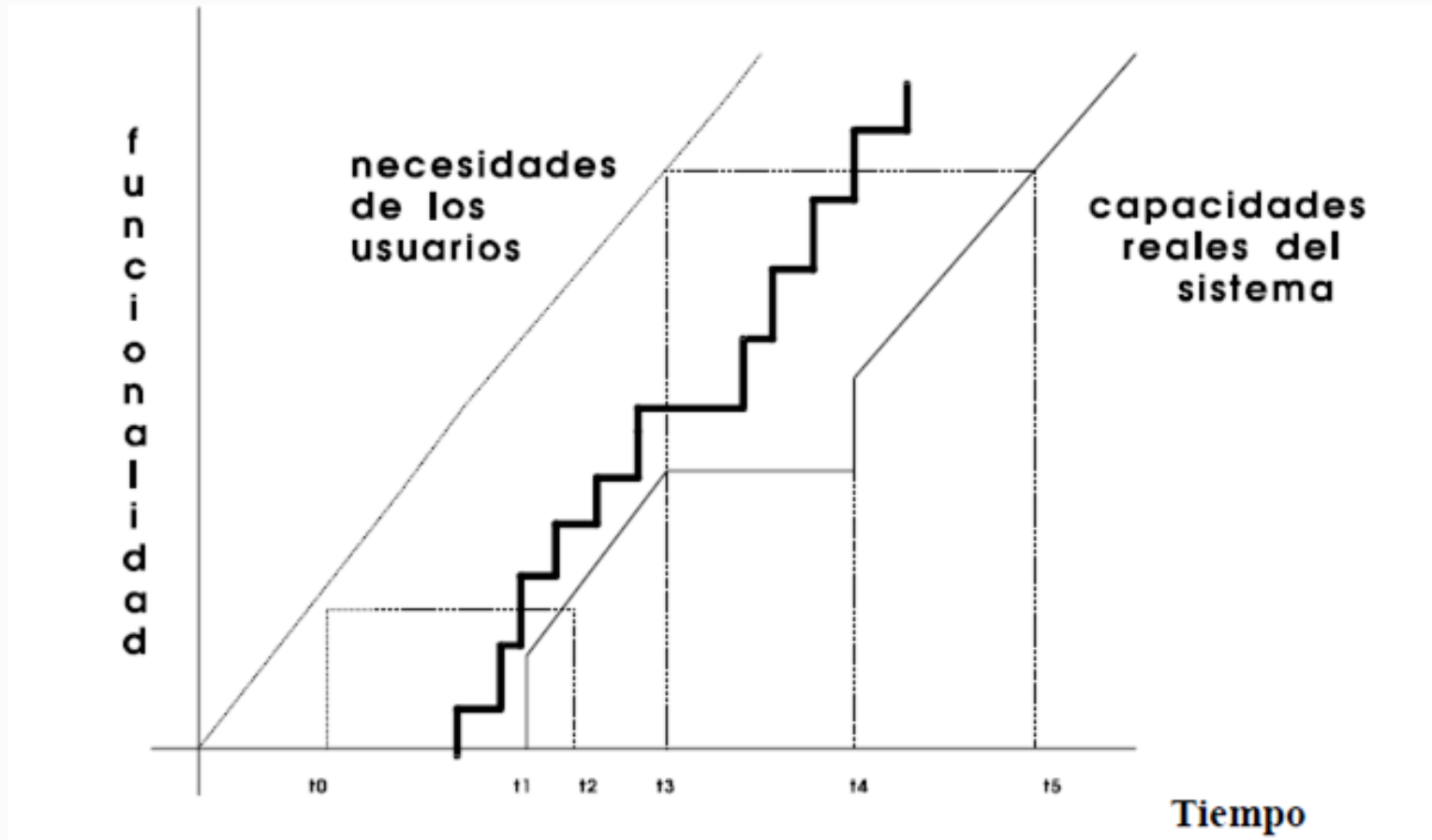
**Las necesidades de los usuarios evolucionan de manera constante**

# Modelos de Proceso-Análisis comparativo (Davis)



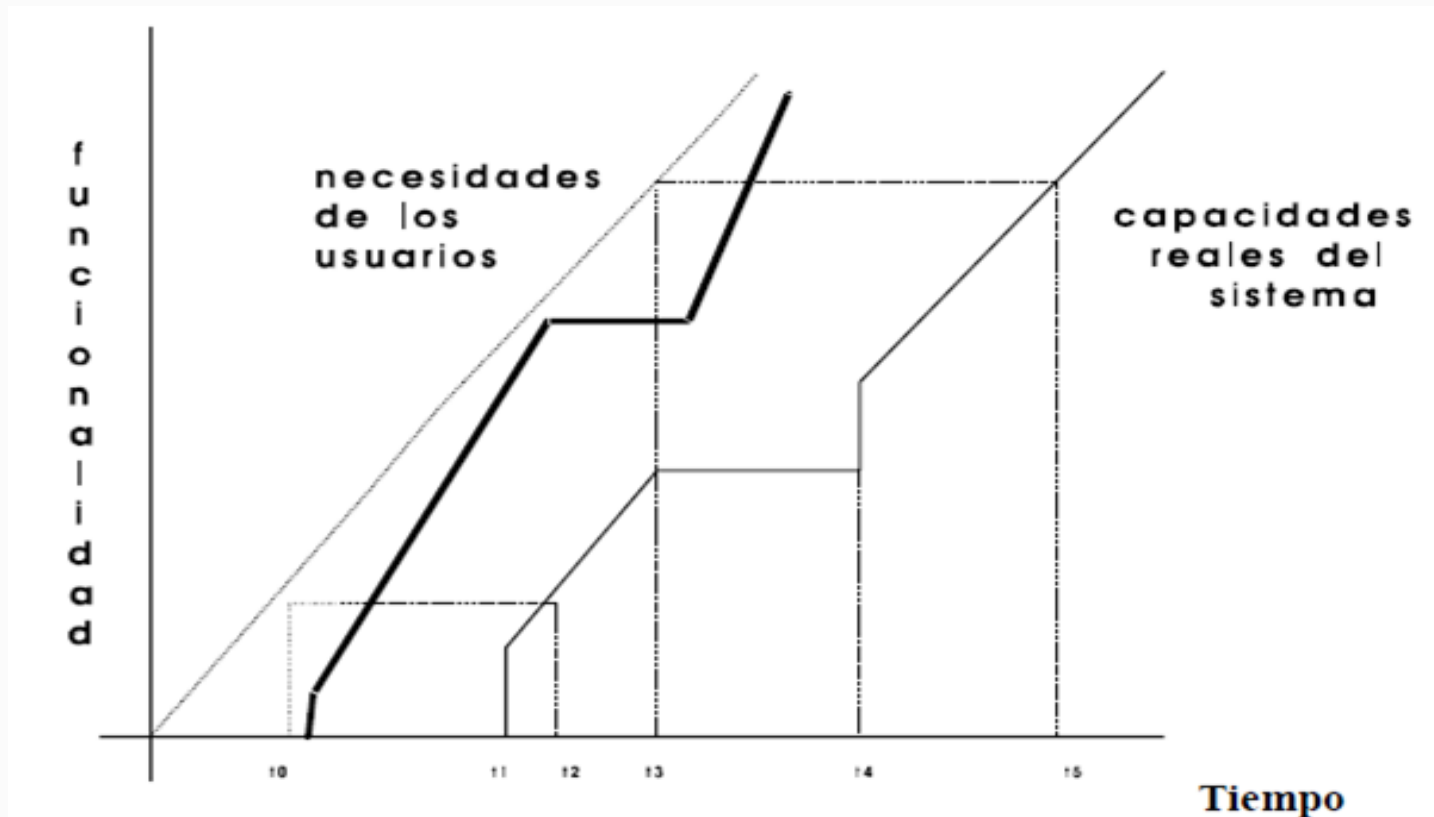
Clásico

# Modelos de Proceso



Incremental

# Modelos de Proceso – Análisis comparativo



Prototipo  
Evolutivo

# Bibliografía

---

## » Libros Utilizados en la Teoría

Pressman, Capítulo 2, Ingeniería de Software Un enfoque práctico, Mc Graw Hill 2010

## » Artículo

Alan Davis et al, “A strategy for comparing alternative software development life cycle models”, IEEE