

Modelos de sistemas

Los modelos se utilizan para comprender el sistema existente, o para especificar el sistema requerido.

Los métodos de análisis estructurado tienen varias debilidades:

1. No proveen apoyo efectivo para comprender o modelar requerimientos del sistema no funcionales
2. Son discriminados ya que no incluyen lineamientos para decidir cuando un método es apropiado para un problema particular.
3. A menudo producen demasiada documentación

Ejemplos de diferentes tipos de modelos del sistema:

1. Modelo de procesamiento de datos. Diagrama de flujo de datos
2. Modelo de composición. Diagramas Entidad-Relación
3. Modelo arquitectónico. Muestra los subsistemas principales que componen a un sistema
4. Modelos de clasificación. Diagramas de clases de objetos, herencia,
5. Modelo estímulo-respuesta. Diagramas de transición de estados.

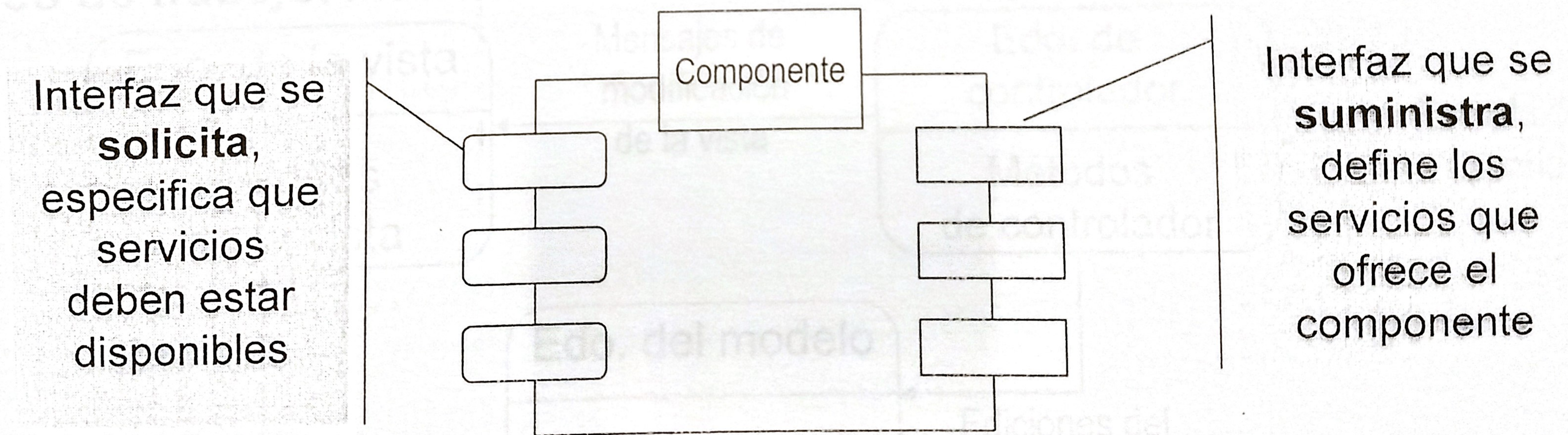
Una vez que se han tomado las decisiones sobre los límites del sistema, lo que sigue es definición ese contexto y las dependencias con su entorno. Por lo general, el primer paso es la producción de un modelo arquitectónico sencillo

Diseño con reutilización

Es un enfoque de desarrollo que trata de maximizar la reutilización de software existente. Reutilización de sistemas, componentes o funciones

Ventajas. Mínimos costos de desarrollo, fiabilidad, mínimo riesgo, desarrollador especialistas, conformidad con los estándares, desarrollo acelerado

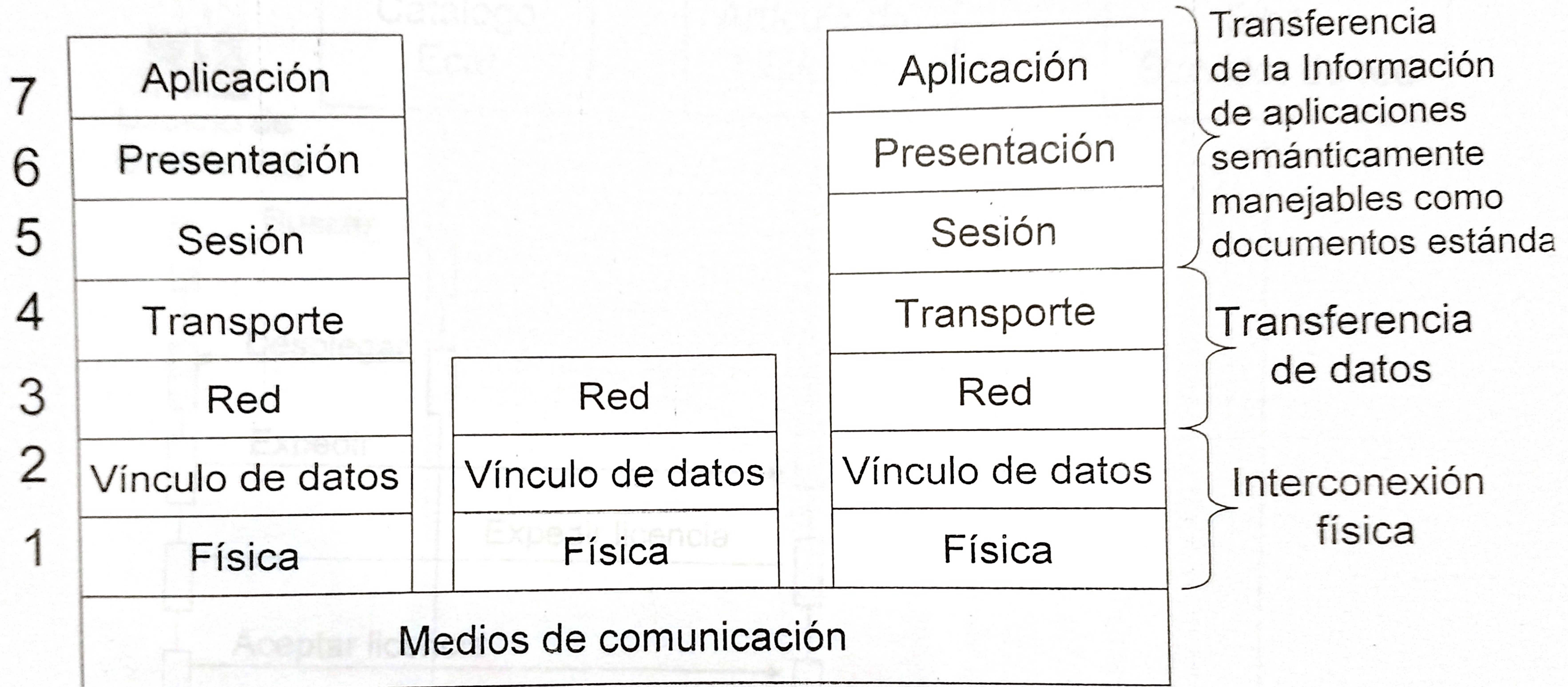
Desventajas. Incremento costos mantenimiento= código fuente no disponible, no apoyo de herramientas case, síndrome de aquí no se ha inventado.



Diseño con reutilización. Desarrollo basado en componentes

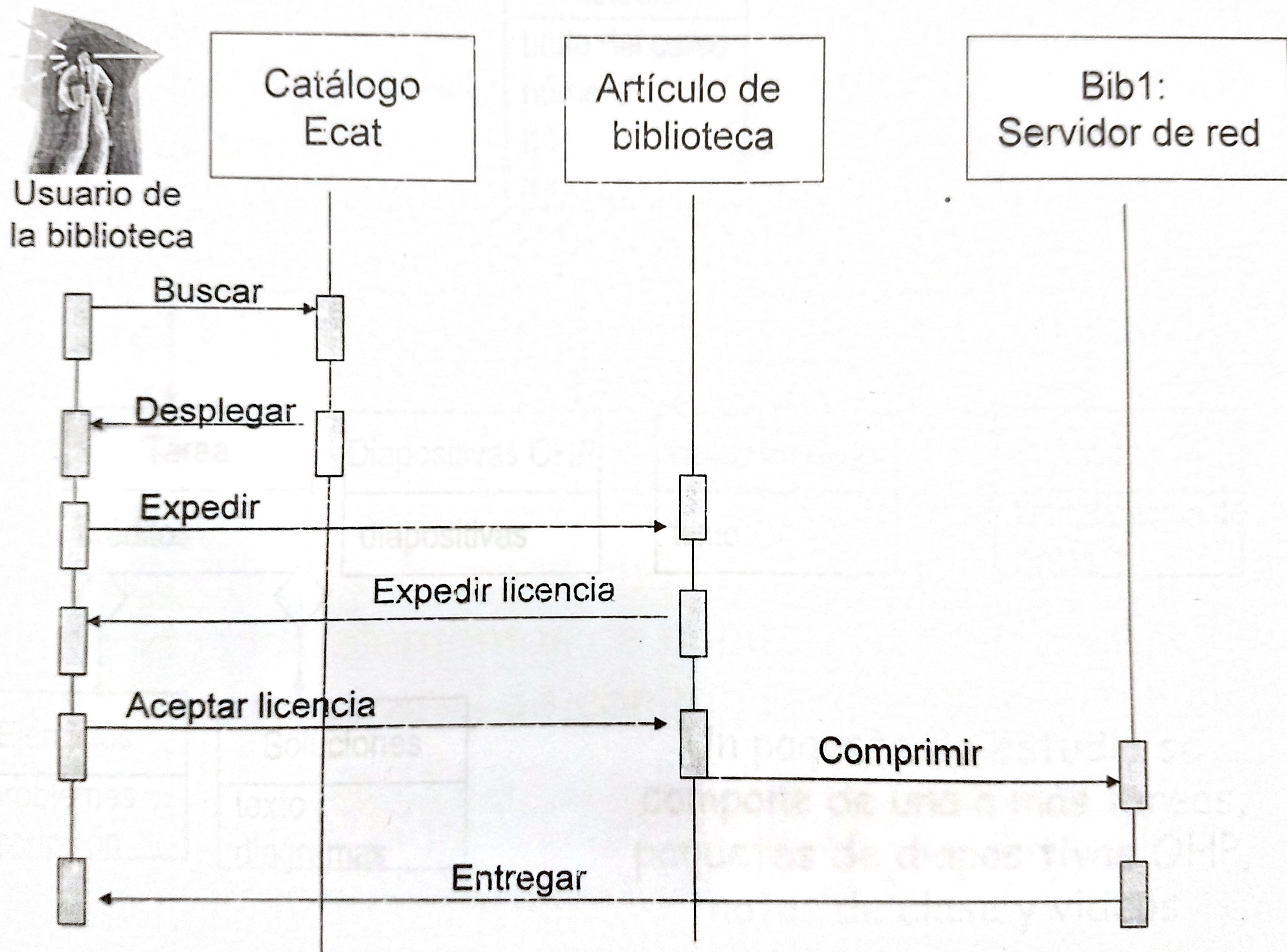
Arquitectura de referencia.

Modelo OSI (Interconexión de sistemas abiertos)



La arquitectura de referencia se utiliza como base para la implementación de sistemas. Ésa fue la intención original de los modelos de referencia OSI (Interconexión de sistemas abiertos)

Modelo de comportamiento de objetos



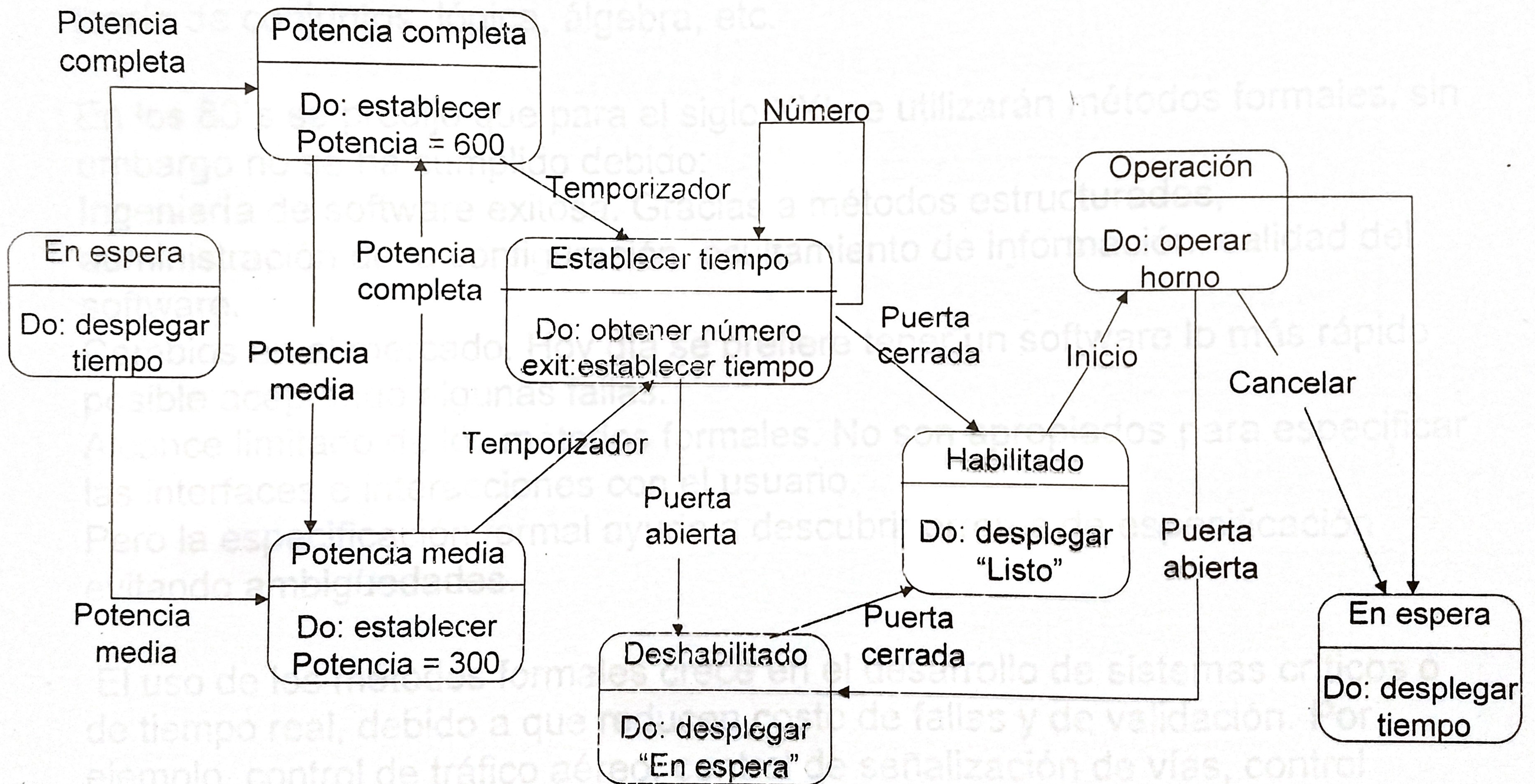
Modelos de máquina de estados (Estimulo-respuesta)

- Se utilizan para modelar el comportamiento de un sistema en respuesta a eventos internos y externos
- No muestra flujo de datos, sino flujo de control
- Son útiles para modelar sistemas de tiempo real, debido a que están dirigidos por estímulos provenientes del entorno del sistema

Considerar:

- Identificar todos los sucesos que involucren al sistema
- Revisar todos los elementos de control, como posibles entradas o salidas, Panel de control, sensores, alarma, línea telefónica
- Identificar como se alcanza cada estado
- Definir transacciones entre los estados

Modelo máquina de estados (estímulo-respuesta)



Especificación formal

Los métodos formales, no son muy utilizados, por el costo que representan. Su especificación se basa en conceptos matemáticos: matemáticas discretas, teoría de conjuntos, lógica, álgebra, etc.

En los 80`s se predijo que para el siglo XXI se utilizarán métodos formales, sin embargo no se ha cumplido debido:

Ingeniería de software exitosa. Gracias a métodos estructurados, administración de la configuración, ocultamiento de información, calidad del software.

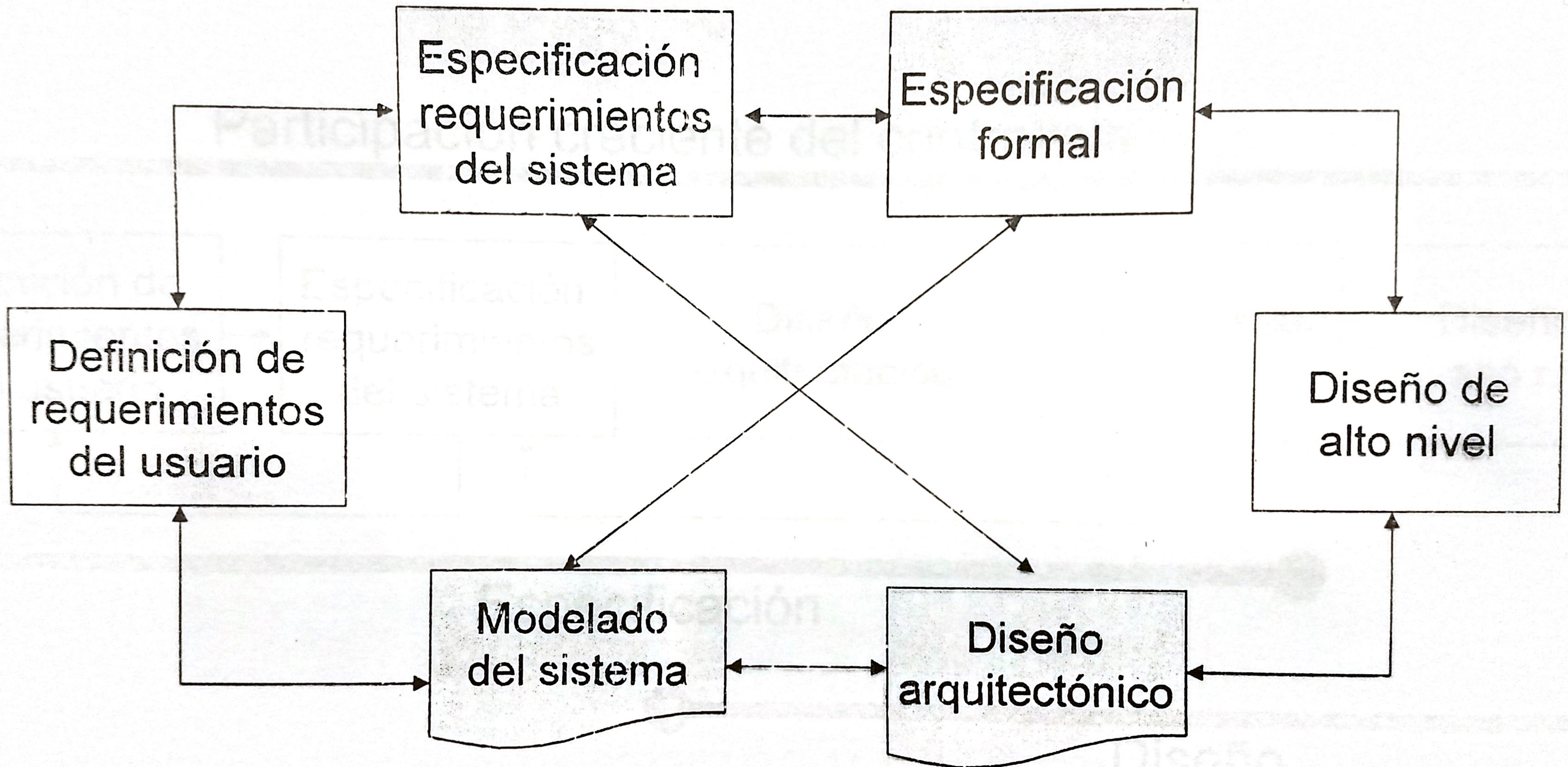
Cambios en el mercado. Hoy día se prefiere tener un software lo más rápido posible aceptando algunas fallas.

Alcance limitado de los métodos formales. No son apropiados para especificar las interfaces e interacciones con el usuario.

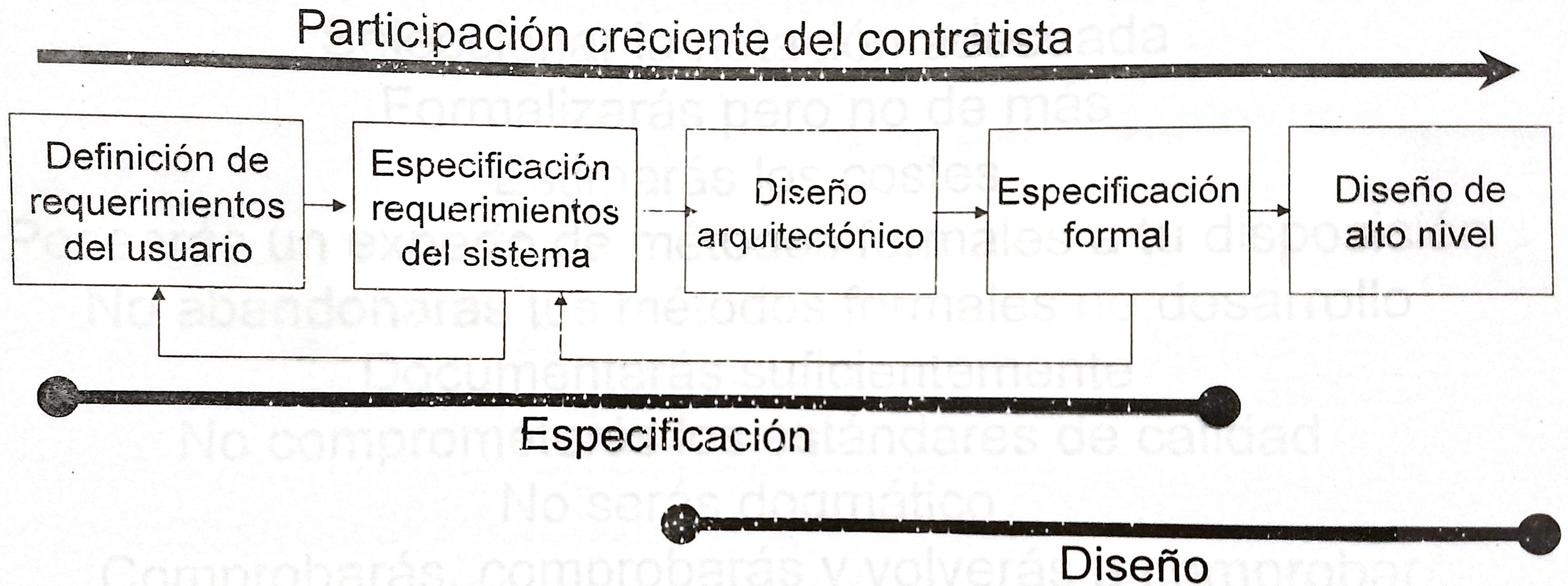
Pero la especificación formal ayuda a descubrir errores de especificación evitando ambigüedades.

El uso de los métodos formales crece en el desarrollo de sistemas críticos o de tiempo real, debido a que reducen costo de fallas y de validación. Por ejemplo; control de tráfico aéreo, control de señalización de vías, control médico, etc.

Especificación formal en el proceso de software



Especificación formal en el proceso de software



Los diez mandamientos de los métodos formales

Seleccionar la notación adecuada

Formalizarás pero no de más

Estimarás los costes

Poseerás un experto de métodos formales a tu disposición

No abandonarás tus métodos formales de desarrollo

Documentarás suficientemente

No comprometerás los estándares de calidad

No serás dogmático

Comprobarás, comprobarás y volverás a comprobar

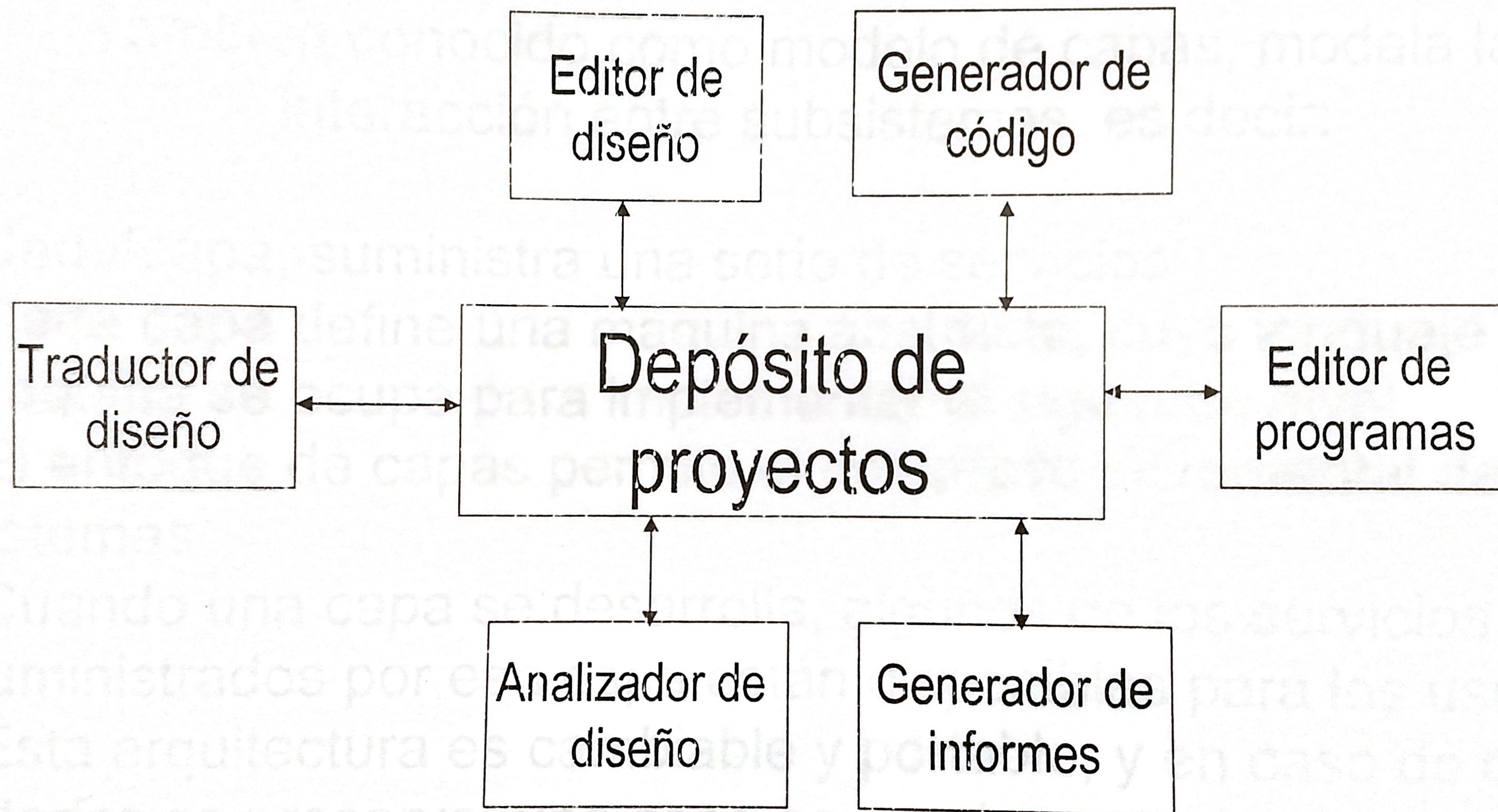
Reutilizarás cuanto puedas

Modelo de depósito

- Es una forma eficiente de compartir datos, en grandes cantidades
- Los subsistemas deben estar acordes al modelo de depósito de datos. De forma inevitable, esto es un compromiso entre las necesidades específicas de cada herramienta.
- Los subsistemas que producen datos no necesitan saber como son utilizados en otros subsistemas
- Sin embargo si se genera un gran volumen de información será difícil migrar a un nuevo modelo
- Las actividades como respaldo, seguridad, control de acceso y recuperación de errores están centralizadas

Modelo de depósito

(Arquitectura de un conjunto integrado de herramientas CASE)



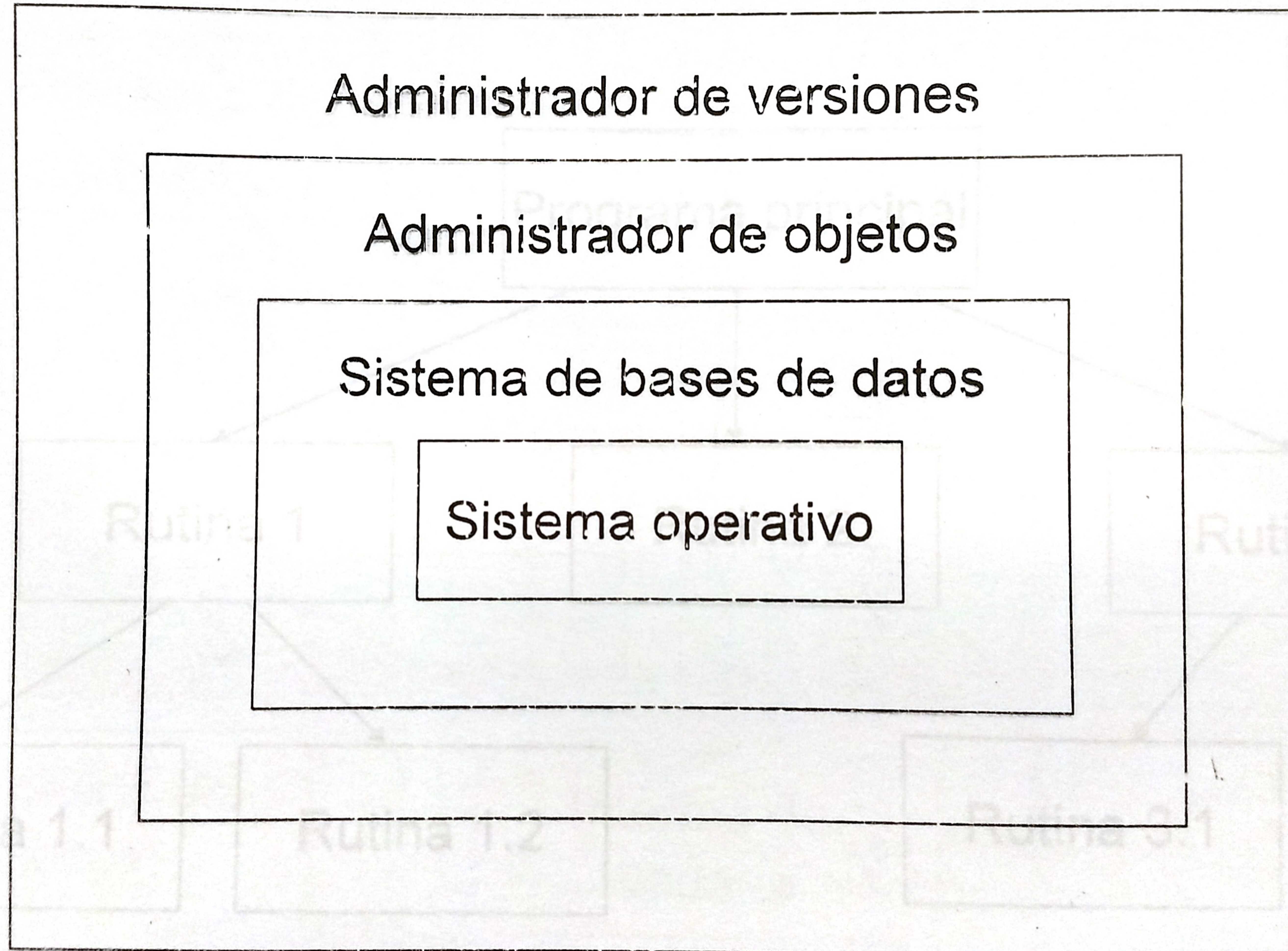
Todos los datos compartidos se ubican en una base de datos central
Cada subsistema tiene su propia base de datos, los datos se intercambian con otros subsistemas

Modelo de máquina abstracta

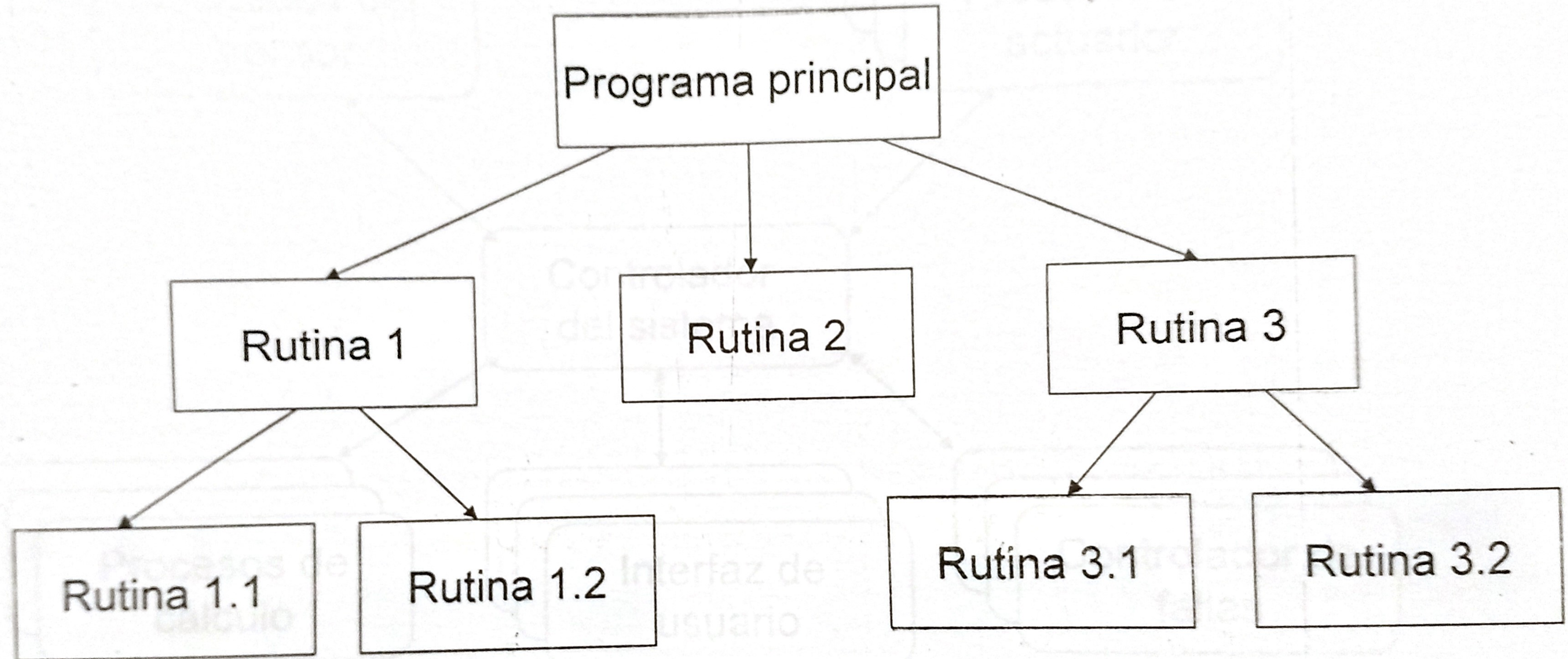
También conocido como modelo de capas, modela la interacción entre subsistemas, es decir:

- Cada capa, suministra una serie de servicios
- Cada capa define una máquina abstracta, cuyo lenguaje de máquina se ocupa para implementar el siguiente nivel
- El enfoque de capas permite el desarrollo incremental de sistemas
- Cuando una capa se desarrolla, algunos de los servicios suministrados por esa capa están disponibles para los usuarios
- Esta arquitectura es cambiante y portable, y en caso de que la interfaz se preserve, una capa se puede reemplazar a otra
- Cuando las interfaces de las capas cambian solo afecta a la capa adyacente
- La estructura en esta forma es difícil de elaborarla

Sistema de administrador de versiones



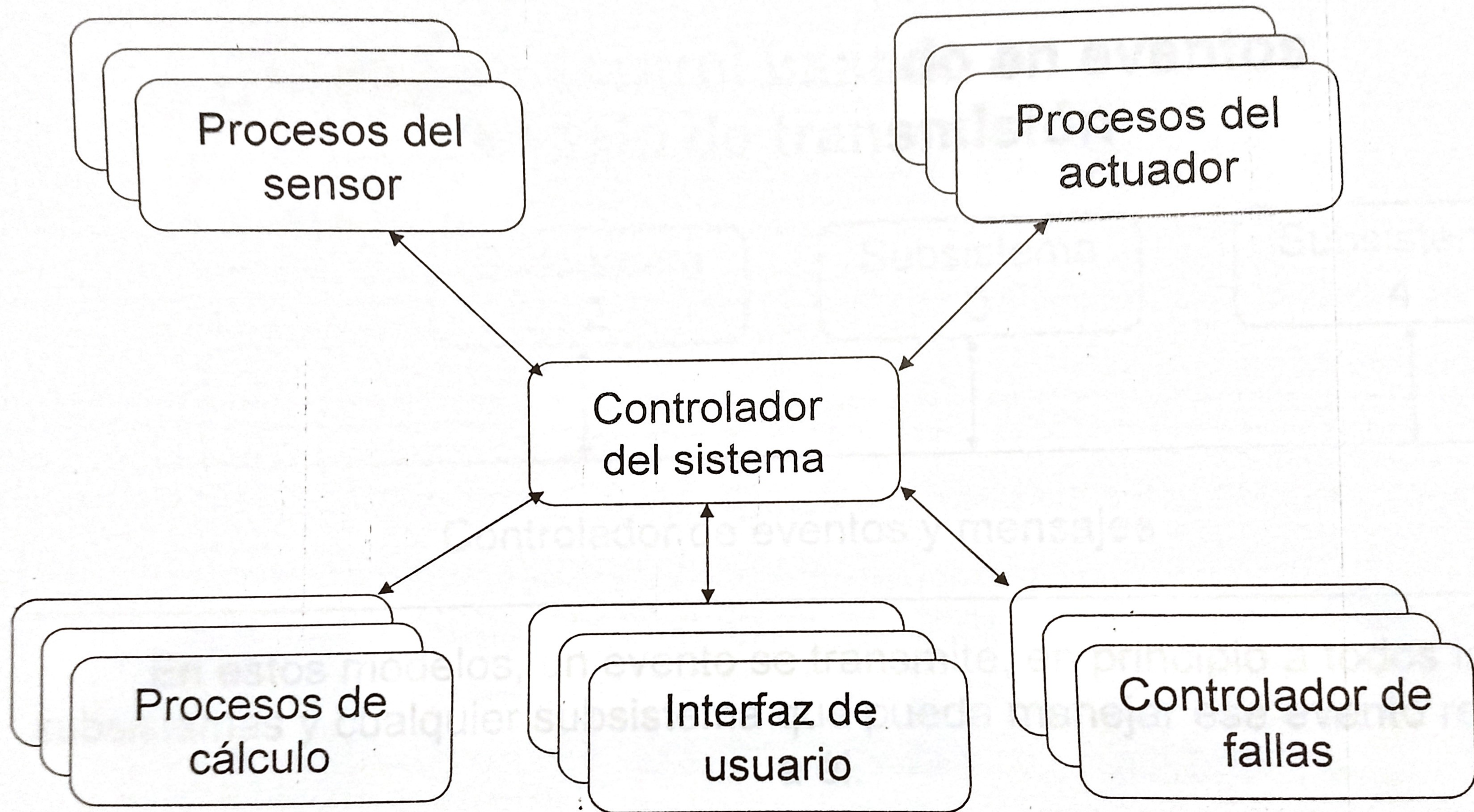
Modelo de control centralizado. (secuencial) Modelo Llamada-Retorno



Su debilidad se da debido a que las excepciones (clase o un evento inesperado durante la ejecución de un programa) a las operaciones, son tediosas de manejar y se tiene que agregar mucho código para que el programa libere el error

Modelo de control centralizado. (paralelo)

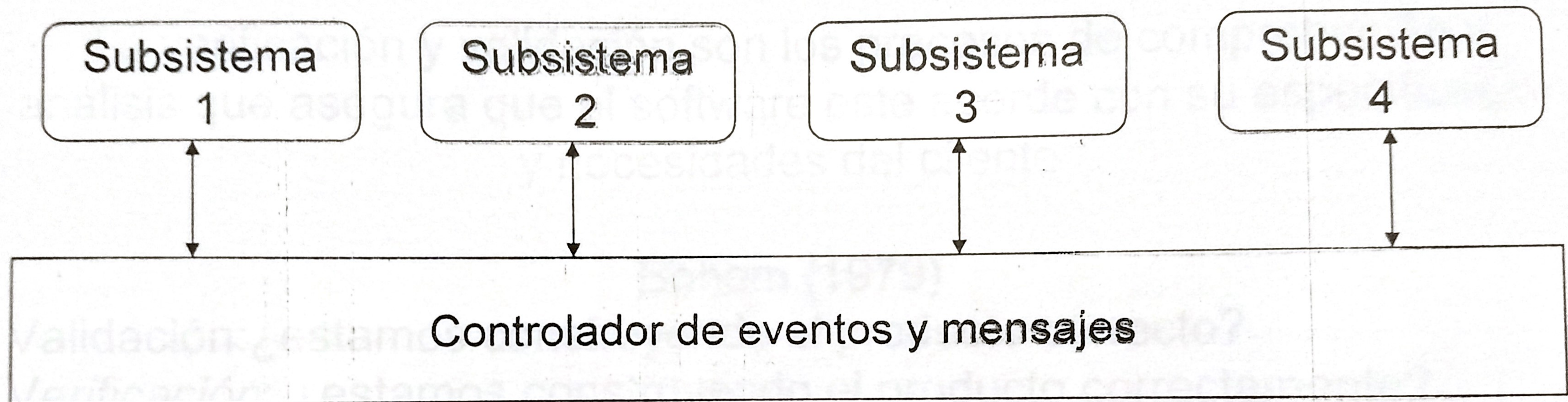
Modelo del administrador



- Se aplica a modelos concurrentes
- Un componente del sistema se designa como sistema administrador y controla el inicio, detención y la coordinación de otros procesos

Modelo de control basado en eventos

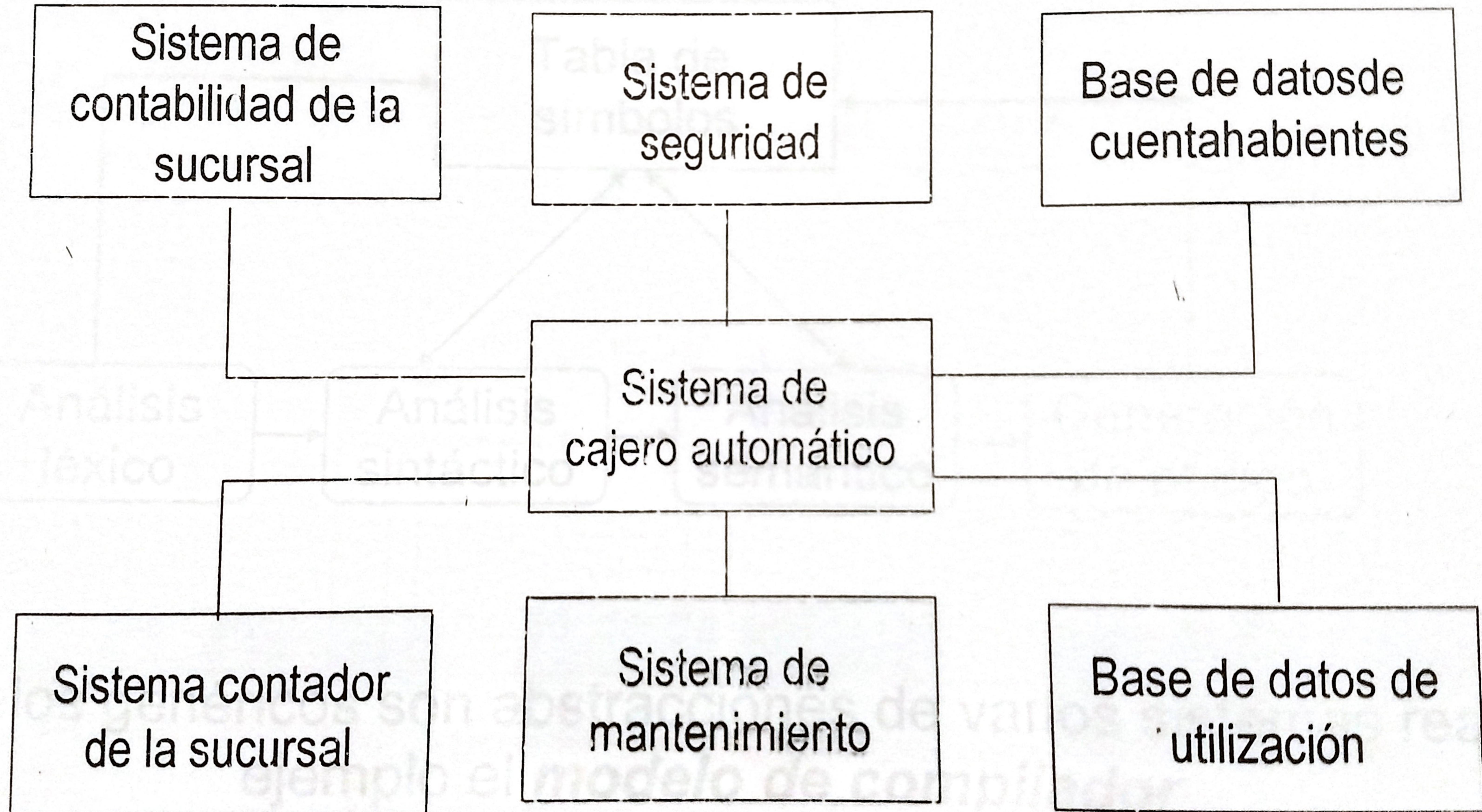
Modelo de transmisión



En estos modelos, un evento se transmite, en principio a todos los subsistemas y cualquier subsistema que pueda manejar ese evento responde a él.

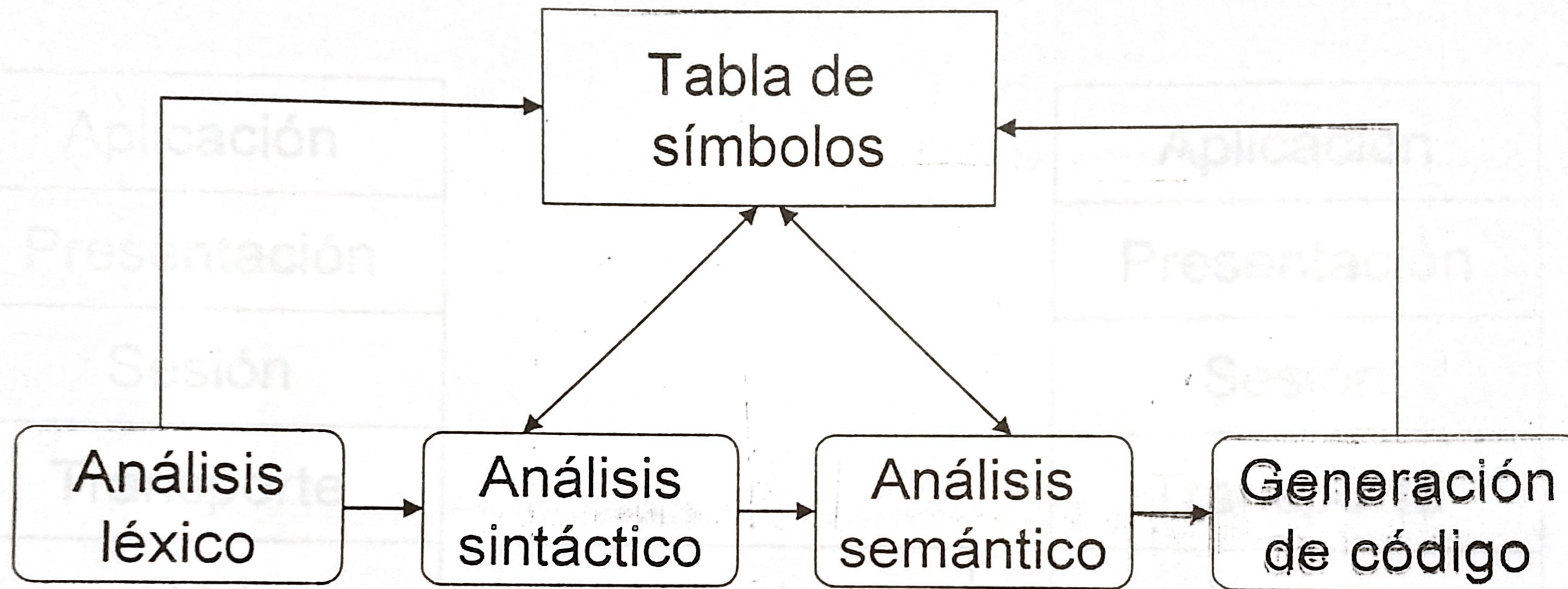
- Son efectivos para integrar subsistemas distribuidos a lo largo de diferentes computadoras en red.
- Los subsistemas deciden qué eventos requieren, y el controlador de eventos y mensajes asegura que éstos eventos sean enviados a dichos subsistemas.
- Todos los eventos se pueden transmitir a todos los subsistemas, pero esto impone un compromiso más alto.

Modelo Arquitectónico Sencillo



Arquitectura de dominio específico

Modelo compilador

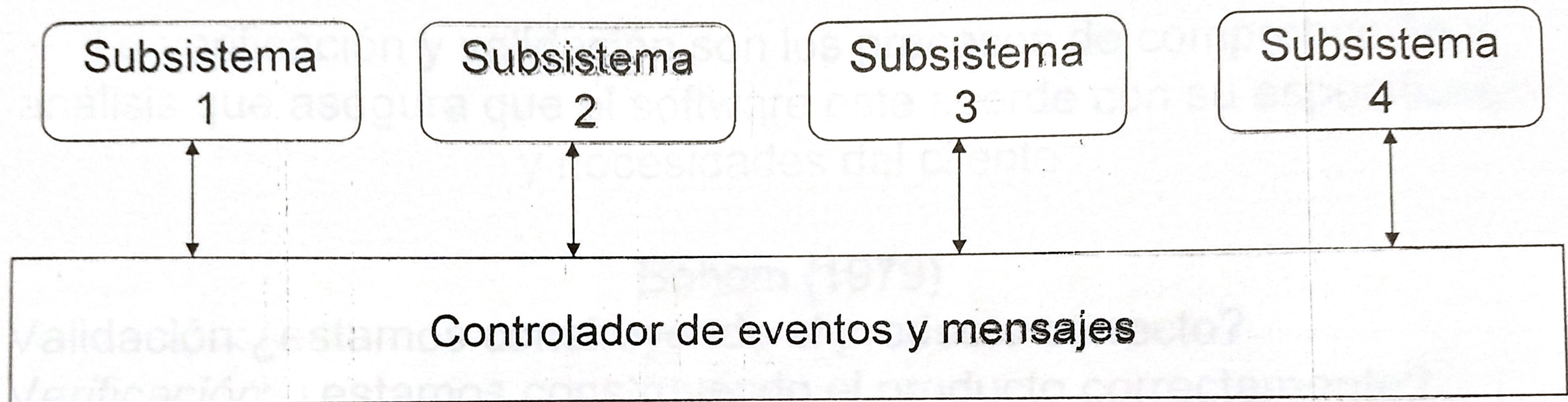


Los modelos genéricos son abstracciones de varios sistemas reales, por ejemplo el **modelo de compilador**

A menudo representa la arquitectura de una línea de productos, donde éstos se implementan utilizando la misma arquitectura básica, por ejemplo controlador de impresión con diferentes capacidades (velocidad, opciones de papel, etc.)

Modelo de control basado en eventos

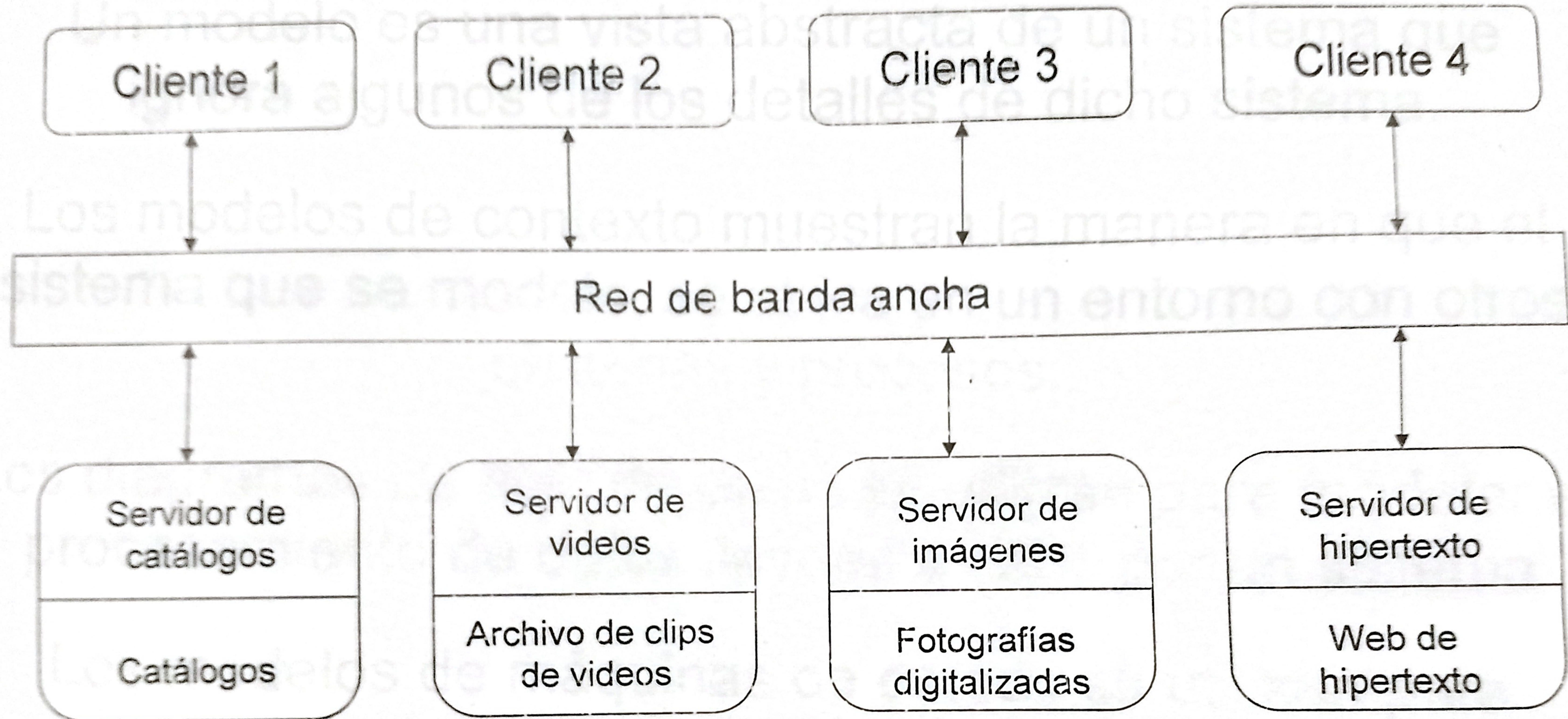
Modelo de transmisión



En estos modelos, un evento se transmite, en principio a todos los subsistemas y cualquier subsistema que pueda manejar ese evento responde a él.

- Son efectivos para integrar subsistemas distribuidos a lo largo de diferentes computadoras en red.
- Los subsistemas deciden qué eventos requieren, y el controlador de eventos y mensajes asegura que éstos eventos sean enviados a dichos subsistemas.
- Todos los eventos se pueden transmitir a todos los subsistemas, pero esto impone un compromiso más alto.

Modelo Cliente Servidor. Sistemas distribuidos



Muestran como los datos y el procesamiento se distribuyen a lo largo de varios procesadores

Los componentes de este modelo son: servidores independientes que ofrecen servicios a otros subsistemas, clientes que llaman a los servicios ofrecidos por los servidores y una red que permite a los clientes acceder a estos servicios

RESUMEN

Un modelo es una vista abstracta de un sistema que ignora algunos de los detalles de dicho sistema.

Los modelos de contexto muestran la manera en que el sistema que se modela, se ubica en un entorno con otros sistemas y procesos.

Los diagramas de flujo de datos se utilizan para modelar el procesamiento de datos llevado a cabo por un sistema

Los modelos de máquinas de estado se utilizan para modelar un comportamiento del sistema en respuesta a los eventos internos o externos

Los modelos semánticos de datos describen la estructura lógica de los datos que se importan o que son exportados por el sistema. (entidades del sistema, sus atributos y relaciones) se complementan con diccionarios de datos