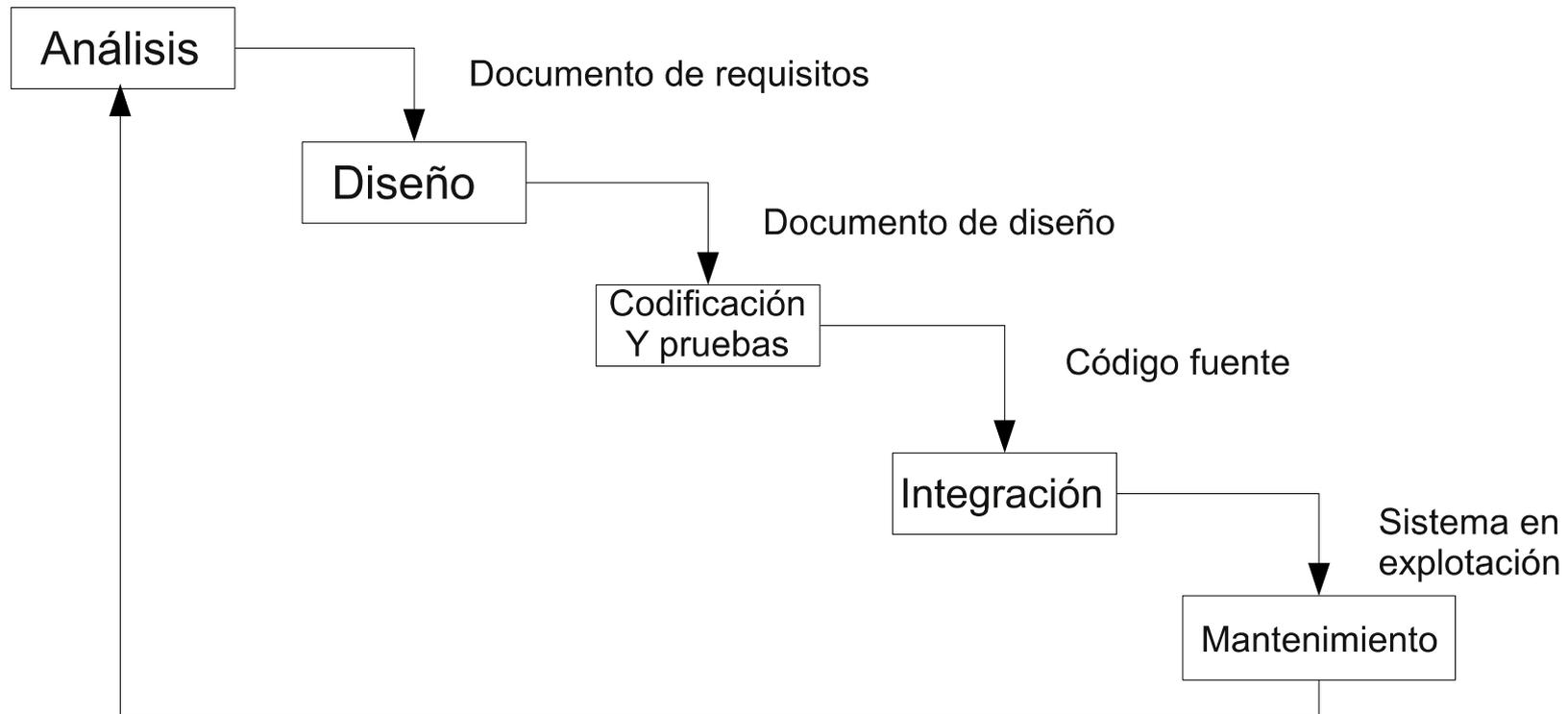


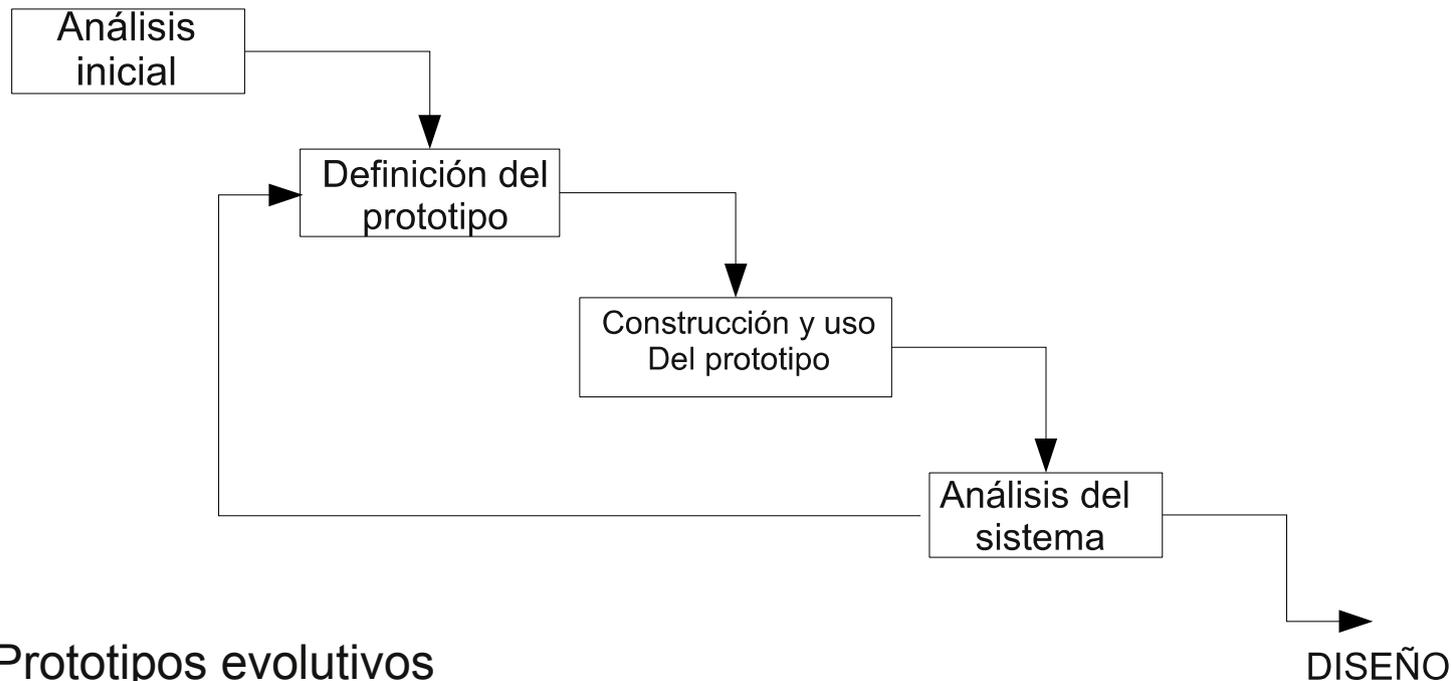
# Proceso de desarrollo del software modelo en cascada



- Análisis: Necesidades del usuario → especificaciones
- Diseño: Descomposición en elementos que puedan desarrollarse por separado → especificaciones de cada elemento
- Codificación: Programación de cada elemento por separado (+pruebas aisladas)
- Integración: Se juntan los elementos y se prueba el sistema completo
- Mantenimiento: Cambios ocasionales (errores o mejoras)

# Prototipos

- Prototipos rápidos
  - Sólo para adquirir experiencia
  - El código no se reusa
  - Se usan para las fases de análisis diseño



- Prototipos evolutivos
  - El código se reusa
  - Proceso cíclico del modelo en cascada
  - En cada vuelta se va mejorando el prototipo hasta llegar a un sistema completo

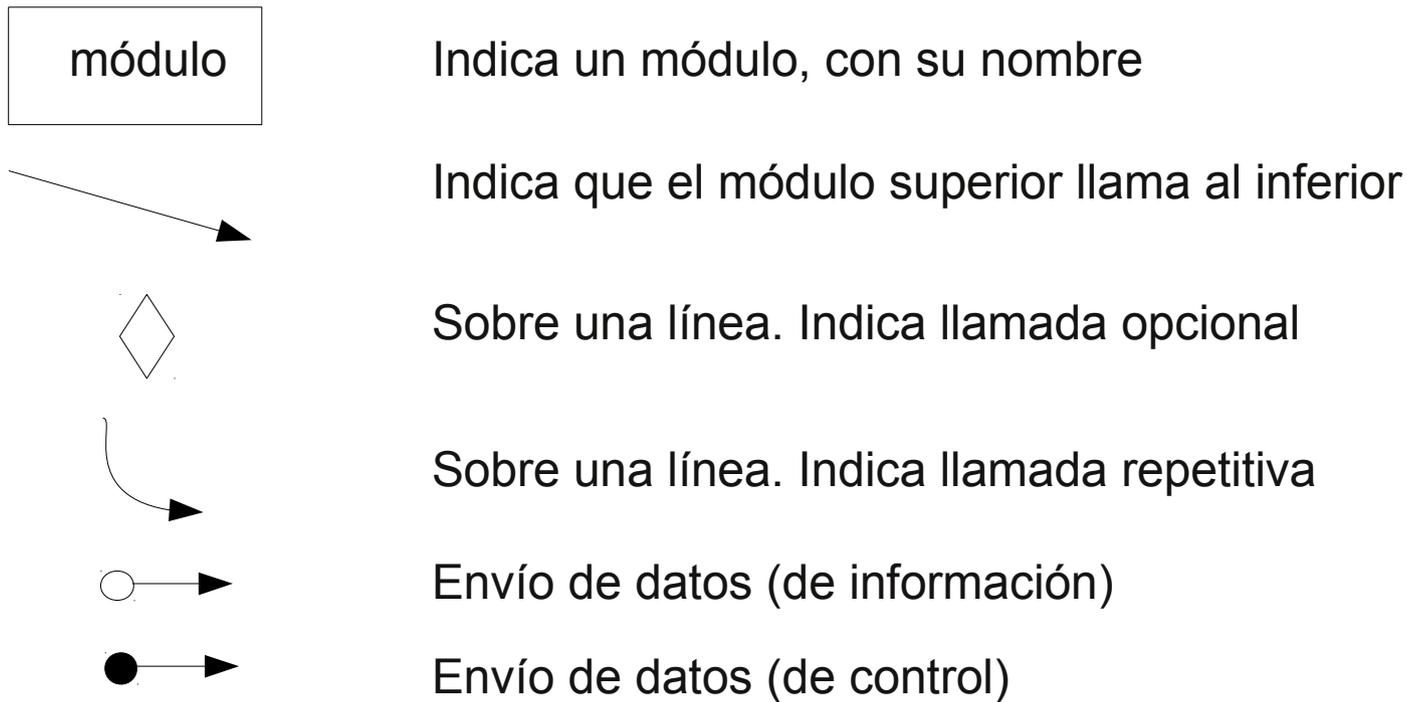
# Especificación de software

- Concepto de modelo del sistema
  - El modelo especifica el QUÉ hace el sistema sin especificar el CÓMO lo hace
  - Se pueden usar distintas técnicas
    - ▶ Descomposición en subsistemas
    - ▶ Modificación de un modelo existente
    - ▶ Análisis del dominio → estudiar entorno, terminología, sistemas similares....
- Análisis de requisitos
  - Objetivo → obtener las especificaciones del software (construir el modelo)
  - Fases
    - ▶ Estudio del sistema en su contexto: sistema SW es parte de un sistema complejo (SW+HW+mecánica+....) → estudio de todos los demás sistemas + estudio del dominio
    - ▶ Identificación de necesidades: interacción con el cliente → necesidades reales
    - ▶ Establecimiento del modelo del sistema
      - Desarrollo jerárquico → división en subsistemas + desarrollo de cada subsistema
      - Finaliza con un **documento de especificación de requisitos**
  - Distintas notaciones posibles para la especificación
    - ▶ Lenguaje natural → para sistemas muy sencillos o como complemento de otros
    - ▶ Diagramas de flujo de datos (DFD) → modelan el procesamiento de los datos en el sistema
    - ▶ Diagramas de transición de estado (DTE) → modelan la dinámica del sistema
    - ▶ Diccionario de datos → modela los datos
    - ▶ .....

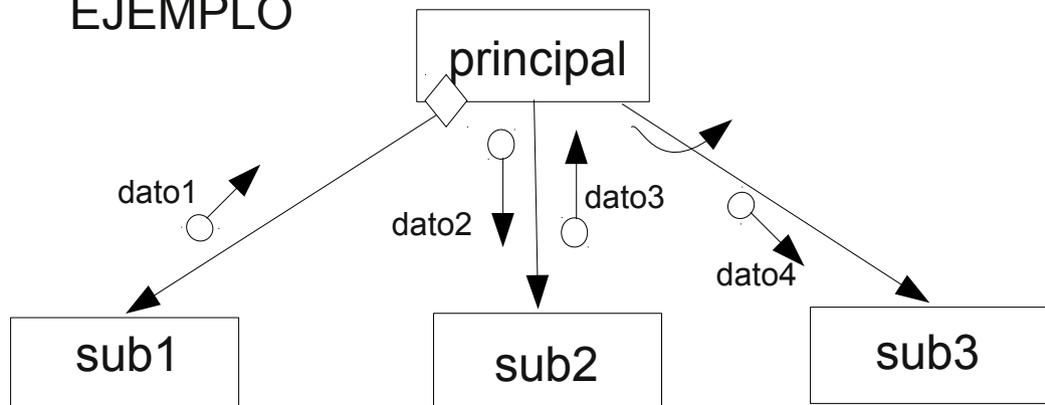
# Diseño de software

- Diseño
  - Decir CÓMO va a hacer el sistema lo que tiene que hacer
  - Finaliza con un **documento de diseño arquitectónico** y un **documento de diseño detallado**
- Fases
  - Diseño arquitectónico
    - Estructura y organización del sistema
    - División en subsistemas o módulos + interfaces entre ellos
  - Diseño detallado → desarrollo de cada módulo
    - Aparecen nuevos módulos, se agrupan o desaparecen otros
    - Definir la estructura de cada módulo, con sus datos y servicios asociados
    - Diseñar los algoritmos para el desarrollo de cada módulo → se detalla en pseudocódigo sin llegar a un nivel muy detallado (sería casi codificación)
  - Diseño de datos → diseño de las bases de datos asociadas al sistema (si es necesario)
- Diagramas de estructura
  - Es uno de las muchas herramientas para el diseño
  - Propuesta por E. Yourdon como herramienta para el diseño estructurado
  - Describen la jerarquía de módulos y submódulos (diseño arquitectónico)
  - El concepto de módulo de Yourdon encaja en lo que es una función de C

# Simbología de los diagramas de estructura



## EJEMPLO



# Características que debe cumplir un módulo

- Acoplamiento (debe ser débil) → es la interrelación que tiene con otros módulos
  - (muy fuerte) Por contenido → acceso a datos locales y código (entre módulos)
  - (fuerte) Común → zona de datos comunes a varios módulos
  - (medio) De control → los módulos se pasan señales de control
  - (débil) Por referencia → los módulos se pasan datos por referencia (p.e.: struct de C)
  - (muy débil) Por valor → paso de datos de un módulo a otro (sólo los que necesita)
- Cohesión (debe ser media/alta) → agrupar en un módulo elementos afines
  - (muy baja) casual → no hay relación (p.e.: cojo un programa de 1000 líneas de código, lo parto en bloques de 100 líneas y hago un módulo con cada bloque)
  - (baja) Lógica → el módulo contiene operaciones cuya ejecución depende de un parámetro (p.e.: una función calcular(operacion,datos) que puede hacer sumas o productos)
  - (media-baja) temporal → el módulo contiene operaciones que se ejecutan en el mismo momento (p.e.: rutinas de inicialización del sistema)
  - (media) comunicación → el módulo realiza distintas operaciones que se ejecutan en “paralelo” y que operan todos sobre el mismo conjunto de datos
  - (media-alta) secuencial → el módulo realiza distintas operaciones que se realizan de forma secuencial sobre los datos, de forma que los datos de salida de una operación son datos de entrada para la siguiente
  - (alta) funcional → el módulo realiza sólo una función
- Comprensibilidad → simple y con funcionamiento comprensible (por quien no lo ha diseñado)
- Adaptabilidad (muy difícil) → posibilidad de cambiarlo con facilidad

# Documento de diseño (modelo de la Agencia Espacial Europea)

- 1. **Introducción** → visión general del documento
  - 1.1. Objetivo
  - 1.2. Ámbito
  - 1.3. Definiciones, siglas y abreviaturas
  - 1.4. Referencias
- 2. **Panorámica del sistema** → visión general de los requisitos + referencia al *documento de especificación de requisitos*
- 3. Contexto del sistema → conexiones con otros sistemas
  - 3.n. Definición de interfaz externa
- 4. **Diseño del sistema** → descripción del nivel superior de diseño (diseño arquitectónico)
  - 4.1. **Metodología de diseño de alto nivel** → descripción de la metodología usada
  - 4.2. **Descomposición del sistema** → componentes del sistema (módulos<sup>1</sup>) y la relación entre ellos
- 5. **Diseño de los componentes** → diseño de cada módulo<sup>1</sup>
  - 5.n.0. **Identificador** del componente
  - 5.n.1. Tipo → módulo<sup>1</sup>
  - 5.n.2. **Objetivo** → justificación de la necesidad de que exista
  - 5.n.3. **Función** → ¿qué hace?
  - 5.n.4. **Subordinados** → componentes (módulos<sup>1</sup>) que usa
  - 5.n.5. **Dependencias** → componentes (módulos<sup>1</sup>) por los que es usado
  - 5.n.6. **Interfases** → reglas de interacción con otros elementos (módulos<sup>1</sup>)
  - 5.n.7. Recursos
  - 5.n.8. Referencias
  - 5.n.9. **Proceso** → algoritmos (se definen con pseudocódigo)
  - 5.n.10. **Datos** → datos internos que usa el componente (módulo<sup>1</sup>)
- 6. Viabilidad y recursos estimados → para llevar a cabo el sistema
- 7. Matriz requisitos/componentes

1.- En el caso de diseño modular

# Propuesta de desarrollo para sistemas pequeños

- Especificación (Análisis)
  - Muy brevemente decir qué hace el sistema sin decir cómo
  - En lenguaje natural o bien lenguaje natural estructurado
  - Sin documento de especificación de software → se incluye en el documento de diseño
- Diseño
  - Diseño arquitectónico
    - ▶ División en módulos y los interfaces entre ellos
    - ▶ Reflejado en un diagrama de estructura
  - Diseño detallado
    - ▶ Diseño de cada uno de los módulos
    - ▶ Se especificará como pseudocódigo (mejor) o diagrama de flujo
    - ▶ Se plasma en el **documento de diseño**
  - Codificación
  - Pruebas

} Se realizarán ambas a la vez y por módulos  
(ojo, no empezar hasta que no esté terminado el diseño detallado)
- Documento de diseño
  - Breve introducción y panorámica del sistema
  - Desarrollo detallado de diseño del sistema y de los componentes