

Fundamentos de Diseño

■ Conceptos fundamentales de Diseño:

1. Abstracción
2. Refinamiento
3. Modularidad
4. Arquitectura del Software
5. Jerarquía de control
6. Estructura de Datos
7. Procedimientos del Software
8. Ocultamiento de Información

Fundamentos de Diseño

➤ 3. Modularidad

Software Modular:

- Es un concepto considerado hace mas de 4 décadas.
- La arquitectura implica modularidad:

La modularidad es un atributo individual del software que permite que un programa sea **intelectualmente manejable**.

Fundamentos de Diseño

➤ 3. Modularidad

Software Monolítico:

- Es un gran programa compuesto por un único módulo.
- El número de variables, las múltiples referencias y la complejidad podría hacer imposible una correcta comprensión.
- Hoy en día su implementación **NO ES** una buena practica.

Fundamentos de Diseño

➤ 3. Modularidad

Datos empíricos observados sobre la resolución de problemas:

Siendo:

- **C(x)**: complejidad de un problema x.
- **E(x)**: esfuerzo en tiempo requerido para resolver un problema x.

...concluimos facilmente que:

- Si **C(p1) > C(p2) \Rightarrow E(p1) > E(p2)**

...y también que:

- **C(p1 + p2) > C(p1) + C(p2)**

...y se concluye que:

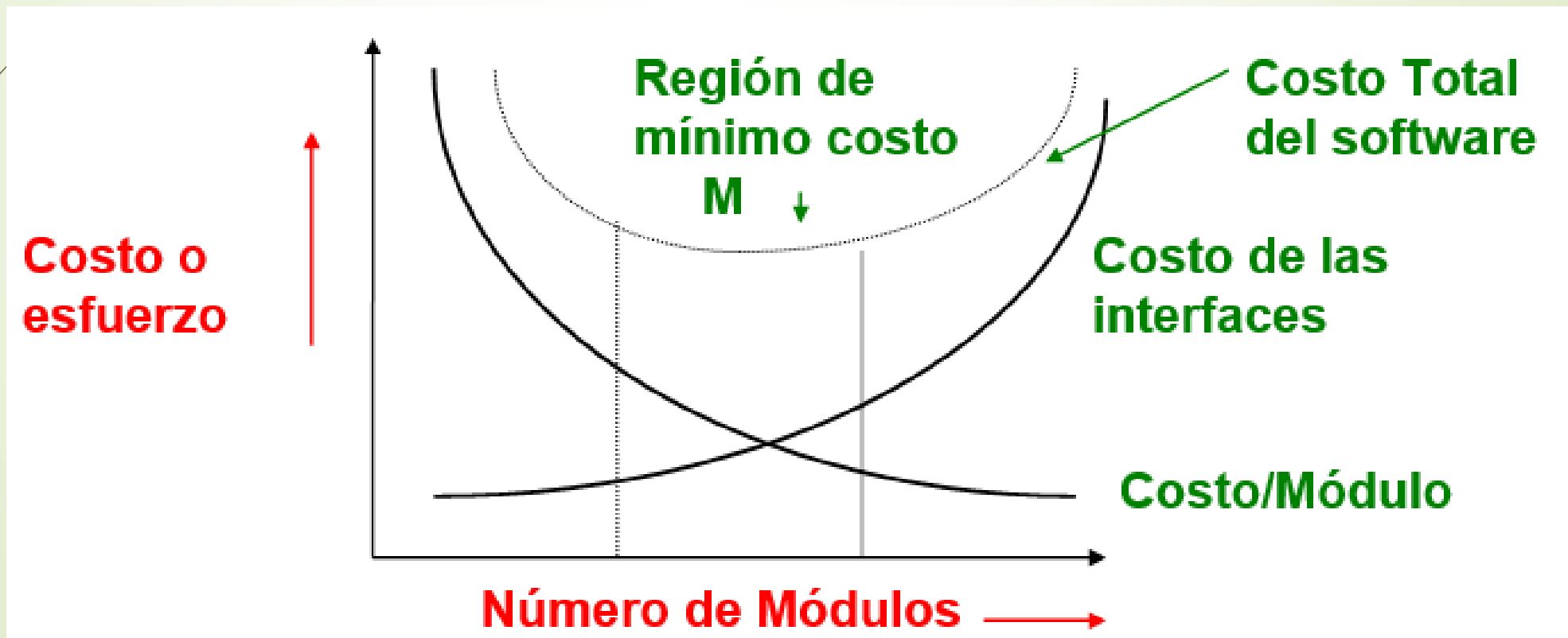
- **E(p1 + p2) > E(p1) + E(p2)**

\therefore Es mas fácil resolver un problema complejo cuando se divide en trozos mas manejables.

Fundamentos de Diseño

➤ 3. Modularidad

Relación entre el esfuerzo y el numero de módulos (Punto M)

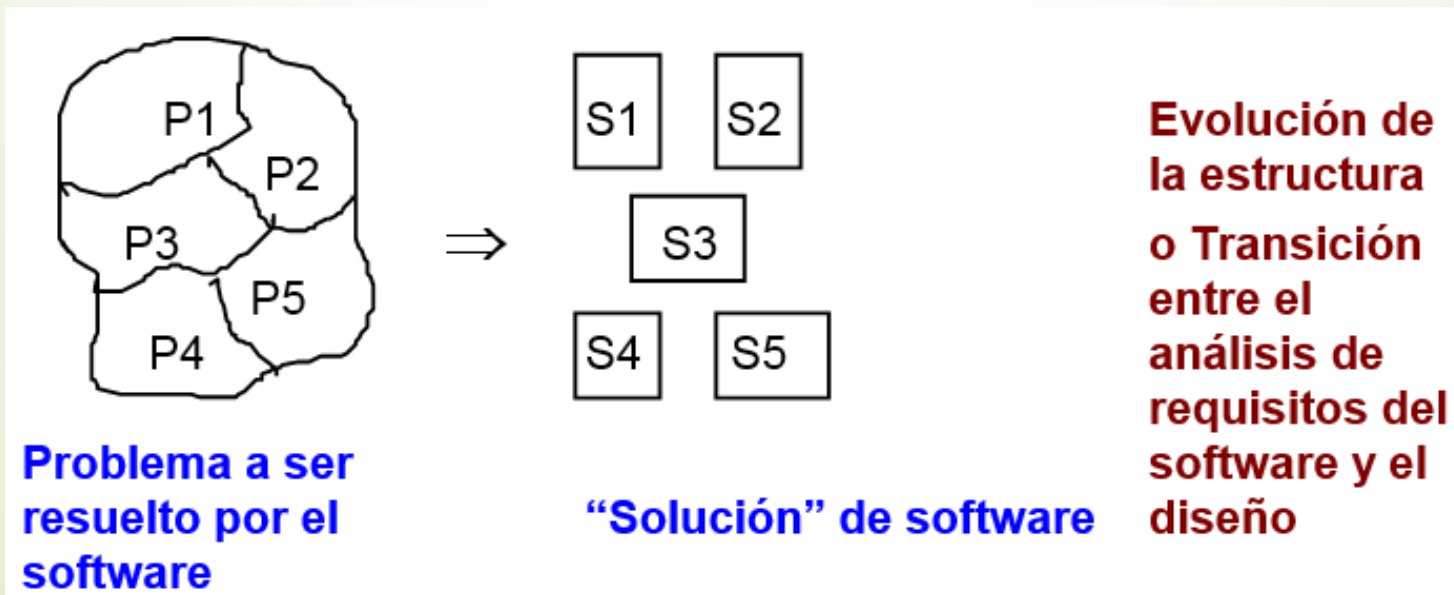


Fundamentos de Diseño

➤ 4. Arquitectura de Software

Existen 2 características importantes:

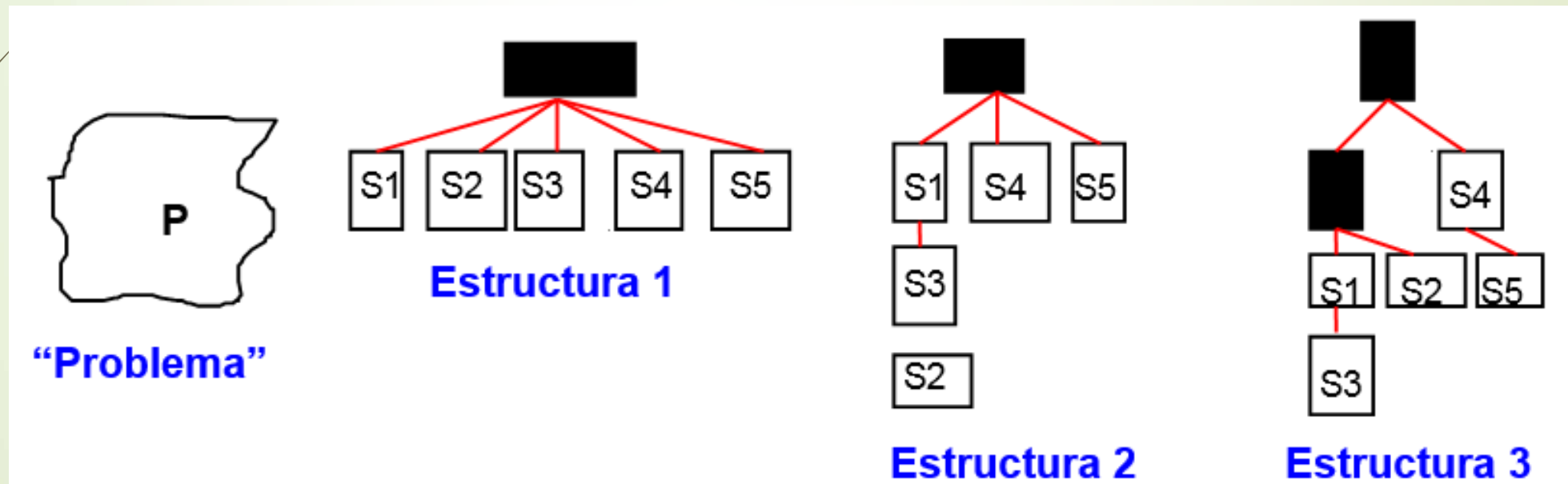
- La estructura jerárquica de los componentes procedimentales (módulos).
- La estructura de datos.



Fundamentos de Diseño

➤ 4. Arquitectura de Software

Las soluciones particulares deben ser relacionadas mediante alguna estructura.



Fundamentos de Diseño

➤ 5. Jerarquía de Control

La jerarquía de control (o estructura del programa) representa la organización (frecuentemente jerárquica) de los módulos del programa e implica una jerarquía de control.

- No representa aspectos procedimentales tales como:
 - La secuencia de procesos
 - La ocurrencia u orden de decisiones
 - La repetición de operaciones.
- **Implica la dependencia entre módulos.**

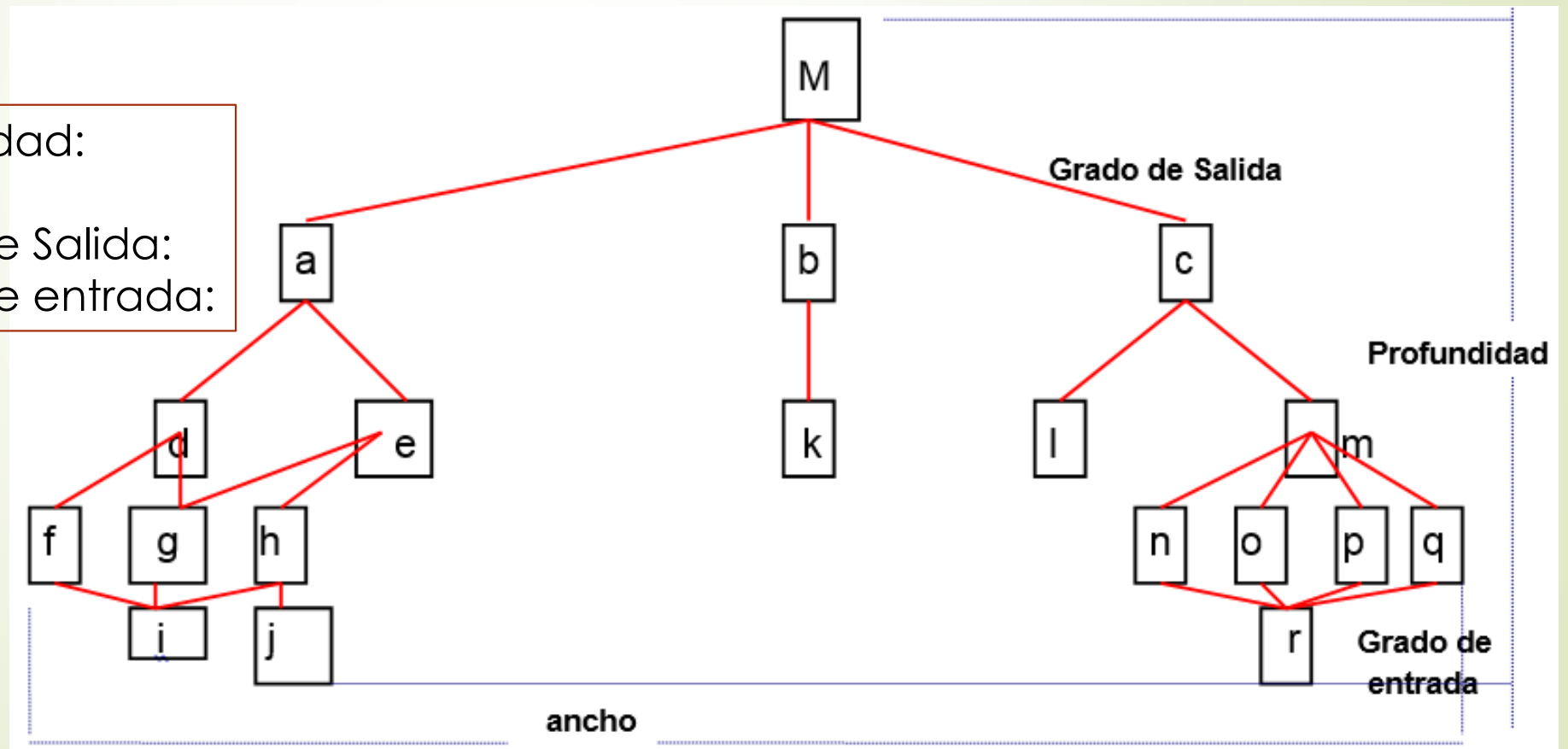
Fundamentos de Diseño

► 5. Jerarquía de Control

La jerarquía de control se representa con diferentes notaciones.

Diagrama de Árbol:

Profundidad:
Ancho:
Grado de Salida:
Grado de entrada:



Fundamentos de Diseño

➤ 5. Jerarquía de Control

Representa otras 2 características diferentes de la arquitectura:

- **Visibilidad**

Indica el conjunto de componentes del programa que pueden ser invocados por un componente dado o cuyos datos pueden ser usados por un componente dado, aún cuando se haga indirectamente.

- **Conectividad**

Indica el conjunto de componentes a los que directamente se invoca o se utilizan sus datos en un determinado módulo. Por ejemplo, un módulo que directamente puede provocar la ejecución de otro módulo, está conectado a él.

Fundamentos de Diseño

➤ 6. Estructura de Datos

- La estructura de datos es una representación de la relación lógica existente entre los elementos individuales de datos.
- En la representación de la arquitectura del software es tan importante como la estructura del programa, debido a que la **estructura de la información afectará invariablemente el diseño procedimental final**.
- La estructura de datos define:
 - la organización
 - los métodos de acceso
 - el grado de asociatividad
 - las alternativas de procesamiento para la información.

Fundamentos de Diseño

➤ 7. Procedimientos del Software

El procedimiento del software se centra sobre los detalles de cada módulo individual.

- Debe proporcionar una especificación precisa del procesamiento incluyendo:
 - la secuencia de sucesos
 - los puntos concretos de decisiones
 - la repetición de operaciones
 - e incluso la organización/estructura de los datos.

Fundamentos de Diseño

■ 8. Ocultamiento de la Información

- Este principio sugiere que los módulos deben especificarse y diseñarse de forma que la información (procedimientos y datos) contenida dentro de un módulo sea inaccesible a otros módulos que no necesiten tal información.
- El ocultamiento establece y refuerza las restricciones de acceso a los detalles procedimentales internos de un módulo y a cualquier estructura de datos localmente usada en el módulo.
- Este concepto es muy útil cuando se modifica, prueba o mantiene el software porque evita o restringe la propagación de errores.