

# Principios de Arquitectura Empresarial

Fabio Castro Rozo



# Contenido

1. Atributos de Calidad
2. Conceptos de Diseño Arquitectónico
3. Estilos de Arquitectura



# Arquitectura de Software y Calidad

## ¿Qué es un Atributo de Calidad?

Un atributo de calidad (QA) es una propiedad medible o comprobable de un sistema que se utiliza para indicar qué tan bien el sistema satisface las necesidades de sus partes interesadas.



# Arquitectura de Software y Calidad

## Arquitectura y Requerimientos

No importa la fuente, todos los requisitos abarcan las siguientes categorías:



**Requerimientos funcionales.** Estos requisitos establecen qué debe hacer el sistema y cómo debe comportarse o reaccionar ante los estímulos de tiempo de ejecución.



**Requisitos de atributos de calidad.** Estos requisitos son calificaciones de los requisitos funcionales (p. ej. rapidez con la que se debe realizar la función o la resistencia que debe tener a una entrada errónea) o del producto general (p.ej el tiempo de implementación del producto o una limitación en los costos operativos.)



**Restricciones:** Una restricción es una decisión de diseño con cero grados de libertad. Es decir, es una decisión de diseño que ya se ha tomado. P. ej. el requisito de usar un cierto lenguaje de programación o reutilizar un cierto módulo existente,



# Arquitectura de Software y Calidad

## ¿Y Arquitectura como gestiona estos requerimientos?

No importa la fuente, todos los requisitos abarcan las siguientes categorías:



**Requerimientos funcionales.** Se satisfacen asignando una secuencia apropiada de responsabilidades a lo largo del diseño



**Requisitos de atributos de calidad.** Se satisfacen por las diversas estructuras diseñadas en la arquitectura y los comportamientos e interacciones de los elementos que pueblan esas estructuras



**Restricciones:** se satisfacen al aceptar la decisión de diseño y conciliarla con otras decisiones de diseño afectadas.



# Arquitectura de Software y Calidad

## Atributos de Calidad

- Disponibilidad
- Performance
- Flexibilidad
- Interoperabilidad
- Mantenibilidad
- Portabilidad
- Reusabilidad
- Robustez
- Testeabilidad
- Usabilidad
- Integridad
- Confiabilidad



# Arquitectura de Software y Calidad

## Atributos de Calidad

- Disponibilidad
- Performance
- Flexibilidad
- Interoperabilidad
- Mantenibilidad
- Portabilidad
- Reusabilidad
- Robustez
- Testeabilidad
- Usabilidad
- Integridad
- Confiabilidad



# Arquitectura de Software

## Lo que se necesita especificar sobre las arquitecturas

El diseño arquitectónico determina cómo componer sistemas a partir de piezas más pequeñas para que el resultado cumpla con los requisitos del sistema

La mayoría de los requisitos del sistema se extienden más allá de la funcionalidad a una variedad de otras propiedades importantes para el cliente

las especificaciones arquitectónicas deben abordar las propiedades **extrafuncionales** de los componentes (estructura, empaquetamiento, dependencias del entorno, representación y rendimiento), la naturaleza de las **interacciones** entre los componentes y las **características estructurales** de las configuraciones.



# Conceptos de Diseño Arquitectónico

¿Cómo se extraen los detalles de la función o la estructura de datos de la representación conceptual del software?

¿Cuáles son los criterios uniformes que definen la calidad técnica de un diseño de software?



# Arquitectura de Software

## Conceptos de Diseño Arquitectónico

### 1. Abstracción

*“La abstracción es uno de los modos fundamentales con los que los humanos luchamos con la complejidad.”*

Grady Booch

Aislar un elemento de su contexto o del resto de los elementos que lo acompañan. En **programación**, el término se refiere al énfasis en el "**¿qué hace?**" más que en el "**¿cómo lo hace?**"

Desde el punto de vista de la programación orientada a objetos expresa las características esenciales de un objeto, las cuales distinguen al objeto de los demás.

# Arquitectura de Software

## Conceptos de Diseño Arquitectónico

### 1. Abstracción

Objeto:

Unidad dentro de un programa de computadores que consta de un estado y de un comportamiento, que a su vez constan respectivamente de datos almacenados y de tareas realizables durante el tiempo de ejecución

**(POO):** un objeto es el resultado de la instanciación de una **clase**. Representa alguna entidad de la vida real, es decir, alguno de los objetos que pertenecen al negocio con que estamos trabajando o al problema con el que nos estamos enfrentando



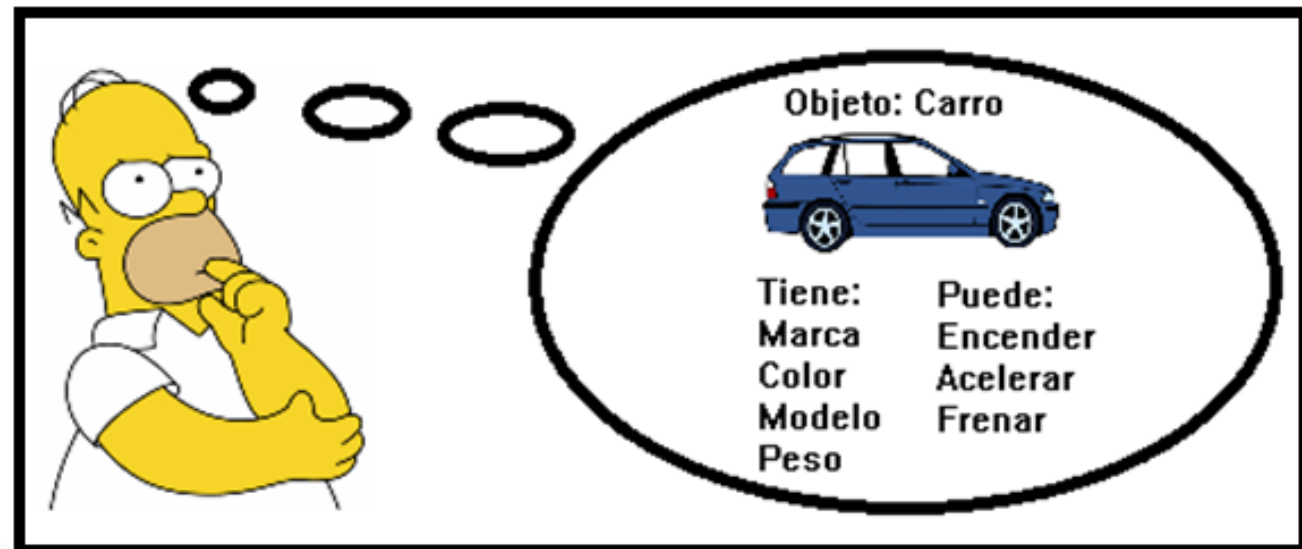
# Arquitectura de Software

## Conceptos de Diseño Arquitectónico

### 1. Abstracción

Características Objeto:

- Identidad
- Comportamiento
- Estado



# Arquitectura de Software

## Conceptos de Diseño Arquitectónico

### 2. Arquitectura

<http://www.cs.cmu.edu/~Compose/ProgCodif.pdf>

Formulations and Formalisms in Software Architecture

“la estructura general de éste (del software) y a las formas en las que ésta da integridad conceptual a un sistema”

MÁS SENCILLO:



La arquitectura es la estructura de organización de los componentes de un programa (módulos), la forma en la que éstos interactúan y la estructura de datos que utilizan.



# Arquitectura de Software

## Conceptos de Diseño Arquitectónico

### 2. Arquitectura

#### Propiedades Estructurales

Define los componentes de un sistema (módulos, objetos, filtros, etc.) y la manera en la que están agrupados e interactúan unos con otros. Por ejemplo, los objetos se agrupan para que encapsulen tanto datos como el procedimiento que los manipula e interactúen invocando métodos.

COMPONENT TYPE	COMMON TYPES OF INTERACTION
Module	Procedure call, data sharing
Object	Method invocation (dynamically bound procedure call)
Filter	Data flow
Process	Message passing, remote procedure call various communication protocols, synchronization
Data file	Read, write
Database	Schema, query language
Document	Shared representation assumptions

# Arquitectura de Software

## Conceptos de Diseño Arquitectónico

### 2. Arquitectura

#### Propiedades Extrafuncionales

La descripción del diseño arquitectónico debe abordar la forma en la que la arquitectura del diseño satisface los requerimientos de desempeño, capacidad, confiabilidad, seguridad y adaptabilidad, así como otras características del sistema.

requisitos de tiempo	facilidad de uso
variabilidad de tiempo	confiabilidad
respuesta en tiempo real	robustez
latencia	capacidad de servicio de p. ej., # de clientes / servidor)
rendimiento	ancho de banda
dependencia de bibliotecas, servicios específicos	requisitos de espacio
conformidad con una interfaz estándar	conformidad con el estándar de implementación
adaptabilidad	precisión y exactitud
mínima configuración de hardware	seguridad

# Arquitectura de Software

## Conceptos de Diseño Arquitectónico

### 2. Arquitectura

### Propiedades Extrafuncionales

For example, this product description specifies the interface between a software product and the operating system/hardware it requires [Com95].

Ejemplo:

1. IBM or 100% IBM-compatible microcomputer with Intel 80386 microprocessor or higher or 100%-compatible processor.
2. Minimum 4 MB RAM.
3. 3 MB of available space on a hard disk.
4. ISO 9660-compatible CD-ROM drive with 640+ MB read capacity and Microsoft CD-ROM extensions.
5. Microsoft Windows'-compatible printer (not plotter) recommended, with 1.5 MB printer memory for 300 dpi laser printing, 6 MB for 600 dpi.
6. Microsoft Windows'-compatible mouse (recommended).
7. Microsoft Windows'-compatible VGA card and monitor.
8. Microsoft Windows' version 3.1 and MS-DOS version 4.01 or later.



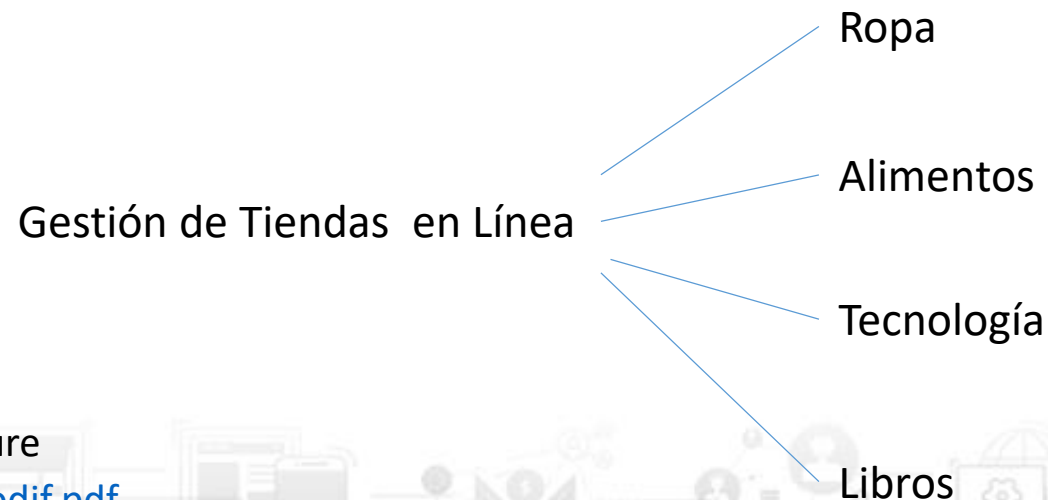
# Arquitectura de Software

## Conceptos de Diseño Arquitectónico

### 2. Arquitectura

#### Familias de sistemas relacionados.

El diseño arquitectónico debe basarse en patrones repetibles que es común encontrar en el diseño de familias de sistemas similares. En esencia, el diseño debe tener la capacidad de reutilizar bloques de construcción arquitectónica, haciendo reutilización al máximo, pero permitiendo adaptación y variabilidad en cada producto particular.



Formulations and Formalisms in Software Architecture

<http://www.cs.cmu.edu/~Compose/ProgCodif.pdf>

# Conceptos de Diseño Arquitectónico

Un patrón de diseño describe una estructura de diseño que resuelve un problema particular del diseño dentro de un contexto específico y entre “fuerzas” que afectan la manera en la que se aplica y en la que se utiliza dicho patrón.

- 1) si el patrón es aplicable al trabajo en cuestión,
- 2) si puede volverse a usar (con lo que se ahorra tiempo de diseño) y
- 3) si sirve como guía para desarrollar un patrón distinto en funciones o estructura.

# Arquitectura de Software

## Conceptos de Diseño Arquitectónico

### 4. División de Problemas

Cualquier problema complejo puede manejarse con más facilidad si se subdivide en elementos susceptibles de resolverse u optimizarse de manera independiente

#### ¿Qué es un problema?

Un *problema* es una característica o comportamiento que se especifica en el modelo de los requerimientos para el software. Al separar un problema en sus piezas más pequeñas y por ello más manejables, se requiere menos esfuerzo y tiempo para resolverlo.

cuando se combinan dos problemas, con frecuencia la **complejidad** percibida es mayor que la suma de la complejidad tomada por separado. Esto lleva a la estrategia de divide y vencerás, pues es más fácil resolver un problema complejo si se divide en elementos manejables.

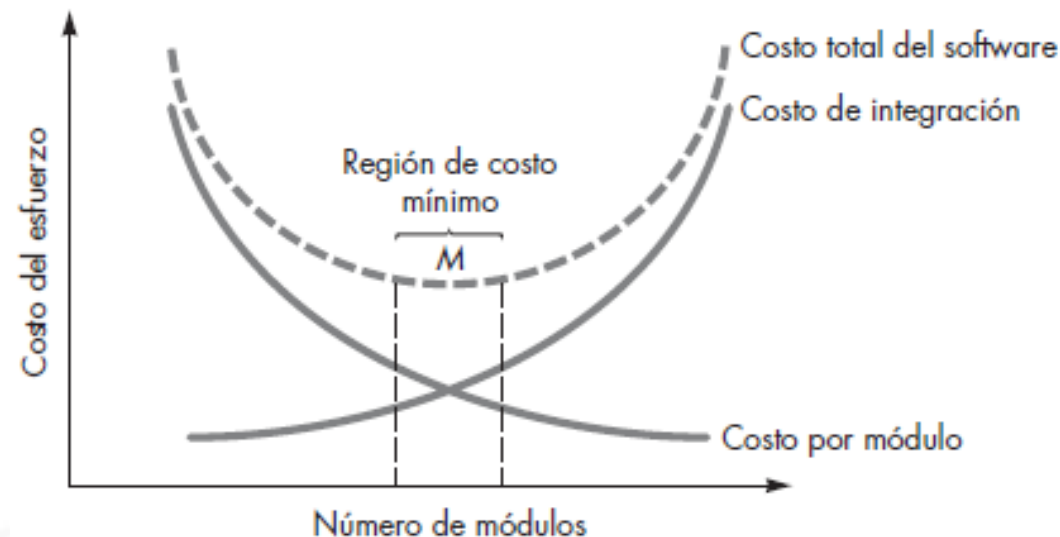
<http://www.utm.mx/~temas/temas-docs/nfnotas217.pdf>

# Arquitectura de Software

## Conceptos de Diseño Arquitectónico

### 5. Modularidad

Es la manifestación más común de la división de problemas. El software se divide en componentes con nombres distintos y abordables por separado, en ocasiones llamados **módulos**, que se integran para satisfacer los requerimientos del problema.



# Arquitectura de Software

## Conceptos de Diseño Arquitectónico

### 5. Ocultamiento de Información

El principio del ocultamiento de información sugiere que los módulos se “caractericen por decisiones de diseño que se oculten (cada una) de las demás”

Deben especificarse y diseñarse módulos, de forma que la información (algoritmos y datos) contenida en un módulo sea inaccesible para los que no necesiten de ella.

La modularidad efectiva se logra definiendo un conjunto de módulos independientes que intercambien sólo aquella información necesaria para lograr la función del software

# Arquitectura de Software

## Conceptos de Diseño Arquitectónico

### 6. Independencia Funcional

Resultado directo de la separación de problemas y de los conceptos de abstracción y ocultamiento de información

Debe diseñarse software de manera que cada módulo resuelva un subconjunto específico de requerimientos y tenga una interfaz sencilla cuando se vea desde otras partes de la estructura del programa

Los módulos independientes son más fáciles de mantener (y probar) debido a que los efectos secundarios causados por el diseño o por la modificación del código son limitados, se reduce la propagación del error y es posible obtener módulos reutilizables.



# Arquitectura de Software

## Conceptos de Diseño Arquitectónico

### 6. Independencia Funcional

#### Cohesión:

Un módulo cohesivo ejecuta una sola tarea, por lo que requiere interactuar poco con otros componentes en otras partes del programa. En pocas palabras, un módulo cohesivo debe (idealmente) hacer sólo una cosa.

#### Acoplamiento:

Medida en que los cambios de un componente tiende a necesitar cambios de otro componente. Los elementos de un componente sólo deben interactuar débilmente, y a través de interfaces claras y bien establecidas con otros componentes



# Arquitectura de Software

## Conceptos de Diseño Arquitectónico

### 6. Independencia Funcional

Debemos de buscar siempre:

**ALTA COHESIÓN y BAJO ACOPLAMIENTO**





# Arquitectura de Software

## Conceptos de Diseño Arquitectónico

### 6. Independencia Funcional

SOA (Service Oriented Architecture)

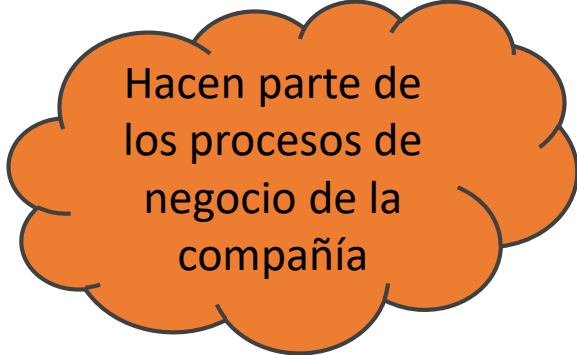


Filosofía de construcción de aplicaciones

¿Cómo? Combinando servicios

¿Y qué es un servicio?

- Los servicios son unidades autocontenidas de funcionalidad
- Reflejan las actividades del negocio en el mundo real
- No dependen del estado de otras funciones o procesos.
- Aceptan llamada(s) y devuelven respuesta(s) mediante una interfaz bien definida



Hacen parte de los procesos de negocio de la compañía

# Arquitectura de Software

## Conceptos de Diseño Arquitectónico

### 6. Refinamiento

Un programa se elabora por medio del refinamiento sucesivo de los detalles del procedimiento

Es un proceso de elaboración. Se comienza con un enunciado de la función (o descripción de la información), definida en un nivel de abstracción alto. el enunciado describe la función o información de manera conceptual, pero no dice nada sobre los trabajos internos de la función o de la estructura interna de la información.

Después se trabaja sobre el enunciado original, dando más y más detalles conforme tiene lugar el refinamiento (elaboración) sucesivo

El refinamiento ayuda a revelar los detalles ocultos por la abstracción a medida que avanza el diseño.



# Arquitectura de Software

## Conceptos de Diseño Arquitectónico

### 7. Aspectos

Relacionados con el comportamiento del software que no son del dominio del problema o aplicación, sino de dominios laterales que afectan de manera transversal a la funcionalidad del sistema.

### Concurrencia

Cómo repartir el software en procesos, tareas o hilos de ejecución y abordar los problemas de eficiencia, atomicidad, sincronización y planificación asociados



# Arquitectura de Software

## Conceptos de Diseño Arquitectónico

### 7. Aspectos

#### Control y Manejo de Eventos

- Cómo organizar datos y flujo de control.  $f$
- Cómo manejar eventos reactivos y temporales

Mediante mecanismos como invocación implícita o llamadas hacia atrás (callbacks).

# Arquitectura de Software

## Conceptos de Diseño Arquitectónico

### 7. Aspectos

#### Distribución de Componentes

- Cómo distribuir el software en el hardware.  $f$
- Cómo comunicar los componentes.  $f$
- Cómo utilizar el middleware para tratar con software heterogéneo.



# Arquitectura de Software

## Conceptos de Diseño Arquitectónico

### 7. Aspectos

#### Manejo de Errores y Excepciones y Tolerancia a Fallos

- Cómo prevenir y tolerar fallos y tratar condiciones excepcionales (no previstas).

#### Interacción y Presentación

- Cómo estructurar y organizar las interacciones con el usuario y la presentación de información. •

Ej.: separando la presentación de la lógica de negocio usando la aproximación MVC (Modelo-Vista-Controlador).



# Arquitectura de Software

## Conceptos de Diseño Arquitectónico

### 7. Aspectos

#### Persistencia de Datos

- Cómo se manejan los datos que tienen una vida superior e independiente a las ejecuciones del software.



# Arquitectura de Software

**Pero... ¿Cómo describo una Arquitectura de Software?**

**ADL**



Architecture Description Language





# Arquitectura de Software

## Architecture Description Language

Lenguaje descriptivo de modelado que se focaliza en la estructura de alto nivel de la aplicación antes que en los detalles de implementación de sus módulos concretos

### Características

- **Composición:** que permiten la representación del sistema como la composición de una serie de partes.
- **Configuración y Abstracción:** Mediante las cuales se describen los roles o papeles abstractos que juegan los componentes dentro de la arquitectura.
- **Flexibilidad:** Ya que permiten la definición de nuevas formas de interacción entre componentes.
- **Reutilización:** Pues permiten la reutilización tanto de los componentes como de la propia arquitectura, Heterogeneidad ya que pueden combinar descripciones heterogéneas.
- **Análisis:** Permiten diversas formas de análisis de la arquitectura y de los sistemas desarrollados a partir de ella.



# Arquitectura de Software

<http://carlosreynoso.com.ar/archivos/arquitectura/ADL.PDF>

## Architecture Description Language

ADL	Fecha	Investigador - Organismo	Observaciones
Acme	1995	Monroe & Garlan (CMU), Wile (USC)	Lenguaje de intercambio de ADLs
Aesop	1994	Garlan (CMU)	ADL de propósito general, énfasis en estilos
ArTek	1994	Terry, Hayes-Roth, Erman (Teknowledge, DSSA)	Lenguaje específico de dominio - No es ADL
Armani	1998	Monroe (CMU)	ADL asociado a Acme
C2 SADL	1996	Taylor/Medvidovic (UCI)	ADL específico de estilo
CHAM	1990	Berry / Boudol	Lenguaje de especificación
Darwin	1991	Magee, Dulay, Eisenbach, Kramer	ADL con énfasis en dinámica
Jacal	1997	Kicillof , Yankelevich (Universidad de Buenos Aires)	ADL - Notación de alto nivel para descripción y prototipado
LILEANNA	1993	Tracz (Loral Federal)	Lenguaje de conexión de módulos
MetaH	1993	Binns, Englehart (Honeywell)	ADL específico de dominio
Rapide	1990	Luckham (Stanford)	ADL & simulación
SADL	1995	Moriconi, Riemenschneider (SRI)	ADL con énfasis en mapeo de refinamiento
UML	1995	Rumbaugh, Jacobson, Booch (Rational)	Lenguaje genérico de modelado – No es ADL
UniCon	1995	Shaw (CMU)	ADL de propósito general, énfasis en conectores y estilos
Wright	1994	Garlan (CMU)	ADL de propósito general, énfasis en comunicación
xADL	2000	Medvidovic, Taylor (UCI, UCLA)	ADL basado en XML

# Arquitectura de Software

## Architecture Description Language

**Pero ... ¿Qué modelo?**



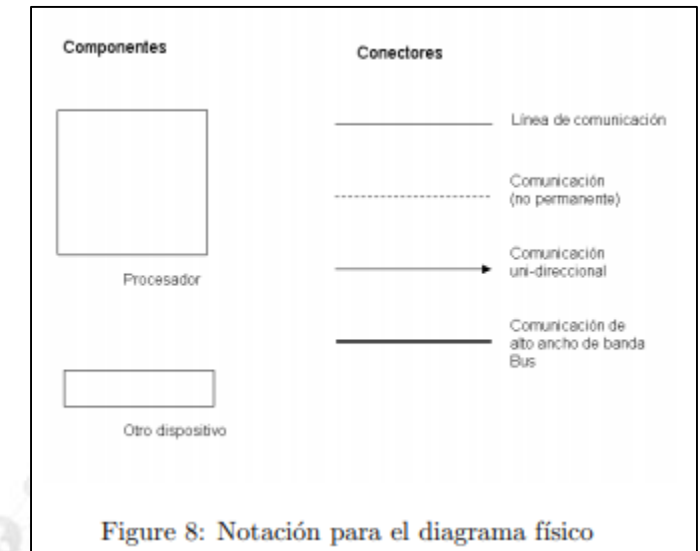
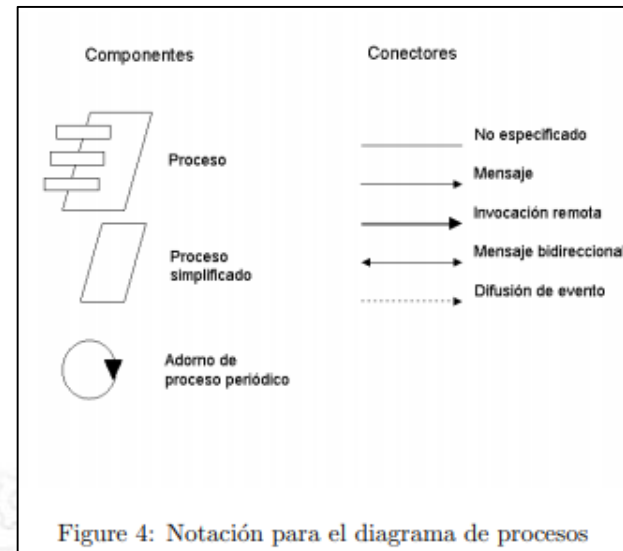
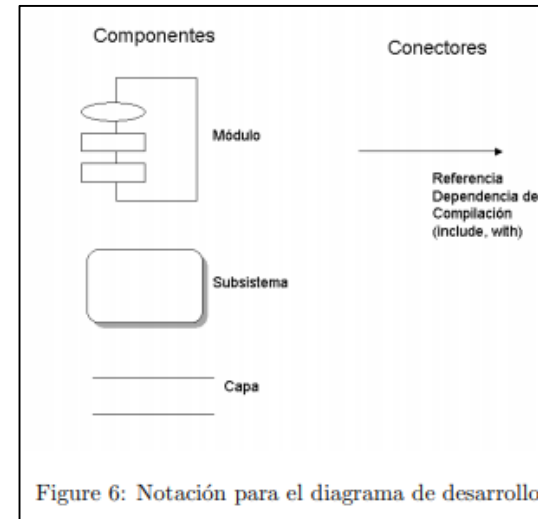
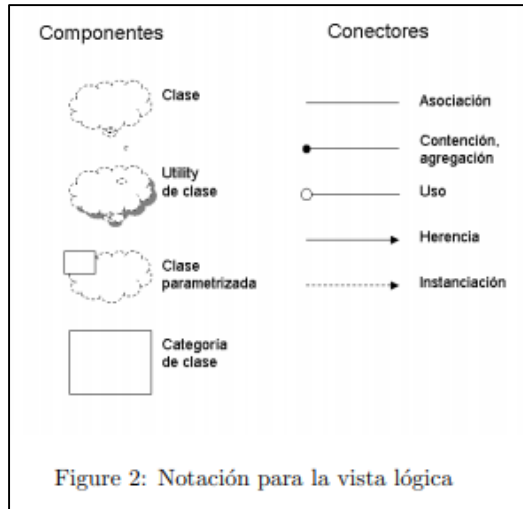
# Arquitectura de Software

## Vistas



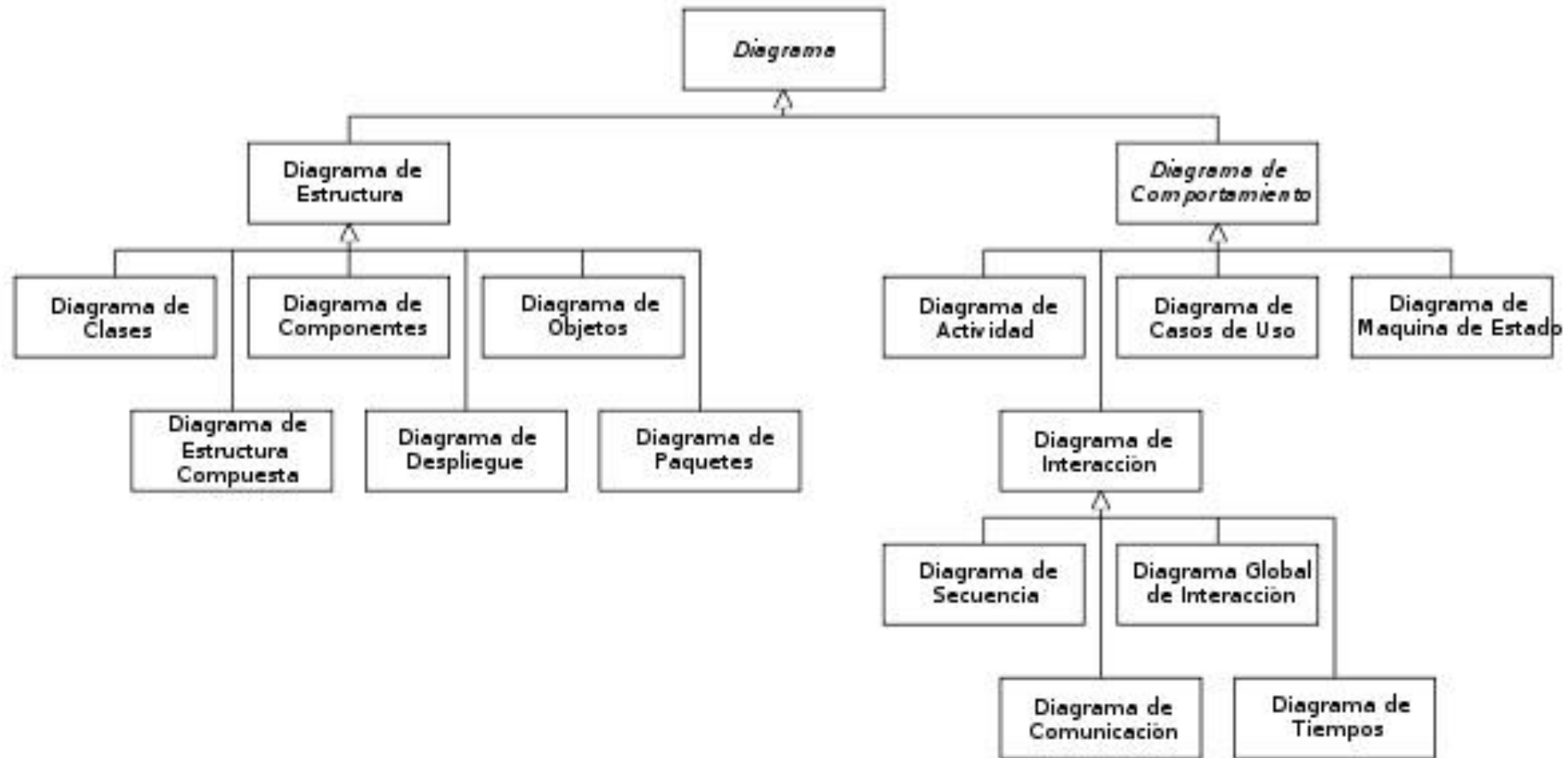
# Arquitectura de Software

## Modelo 4+1



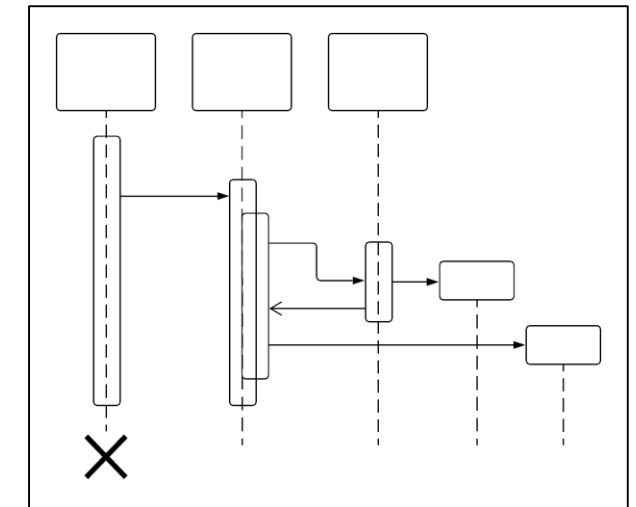
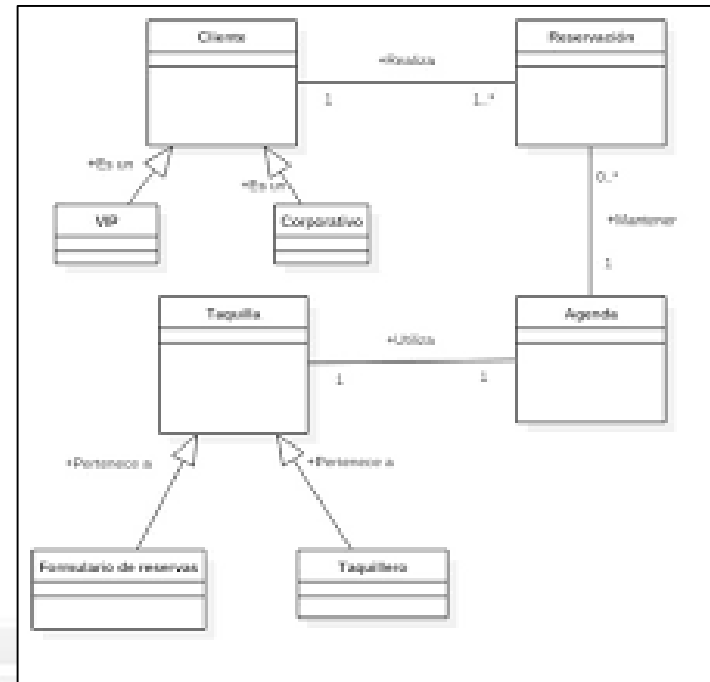
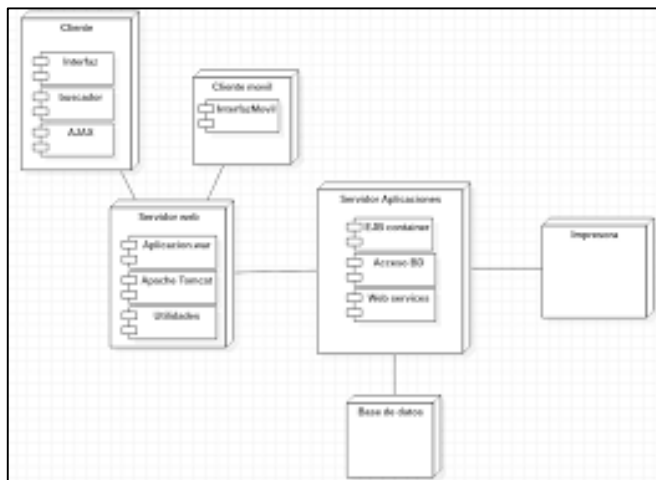
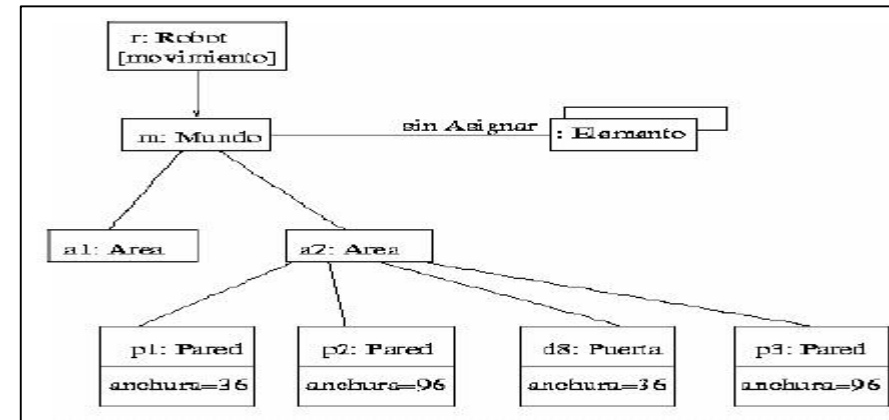
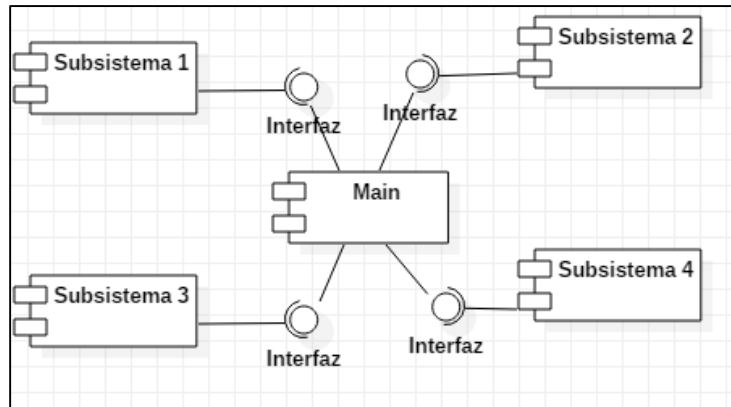
# Arquitectura de Software

## Modelado UML



# Arquitectura de Software

## Modelado UML





# Arquitectura de Software

## Estilos Arquitectónicos

*“En el fondo de la mente de todo artista hay un patrón o tipo de arquitectura.”*

**G. K. Chesterton**

Conjunto de restricciones que definen un conjunto o familia de arquitecturas afines, que satisfacen dichas restricciones.

### Pero ... ¿Qué más es?



El estilo arquitectónico también es una plantilla para la construcción. Deben definirse más detalles, especificar sus dimensiones finales, agregar características personalizadas, determinar los materiales de construcción, pero el estilo orienta al constructor en su trabajo.





# Arquitectura de Software

## Estilos Arquitectónicos

*“En el fondo de la mente de todo artista hay un patrón o tipo de arquitectura.”*

**G. K. Chesterton**

Conjunto de restricciones que definen un conjunto o familia de arquitecturas afines, que satisfacen dichas restricciones.

### Pero ... ¿Qué más es?



El estilo arquitectónico también es una plantilla para la construcción. Deben definirse más detalles, especificar sus dimensiones finales, agregar características personalizadas, determinar los materiales de construcción, pero el estilo orienta al constructor en su trabajo.



# Arquitectura de Software

## Estilos Arquitectónicos

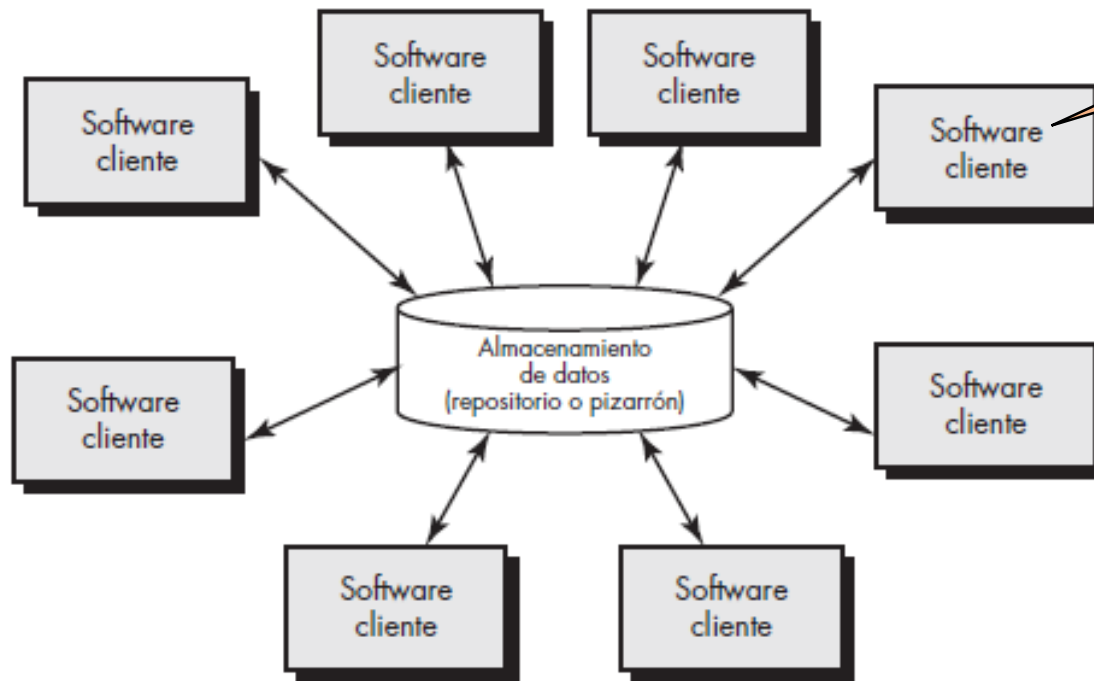
- Un estilo arquitectónico es una transformación que se impone al diseño de todo el sistema.
- El objetivo es establecer una estructura para todos los componentes del sistema

Puede ser visto como un modelo de modelos (metamodelo) que enmarca a muy alto nivel la organización del software (macro-arquitectura)

# Arquitectura de Software

## Estilos Arquitectónicos

### 1. Arquitecturas centradas en los datos



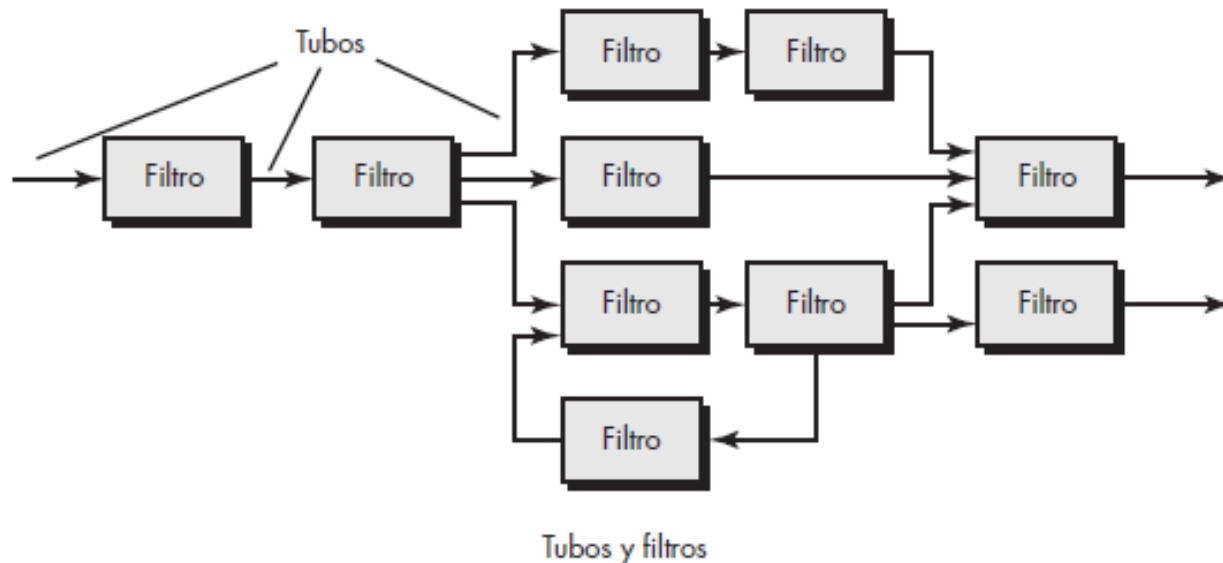
Actualizan, agregan, eliminan o modifican los datos de cierto modo dentro del almacenamiento

Promueve la capacidad de integración, es decir, que es posible cambiar componentes existentes y agregar nuevos componentes a la arquitectura sin preocuparse por otros clientes, además es posible pasar datos entre clientes empleando el mecanismo del pizarrón. Los componentes clientes ejecutan los procesos de manera independiente.

# Arquitectura de Software

## Estilos Arquitectónicos

### 2. Arquitecturas de flujo de datos (Tubos y Filtros)

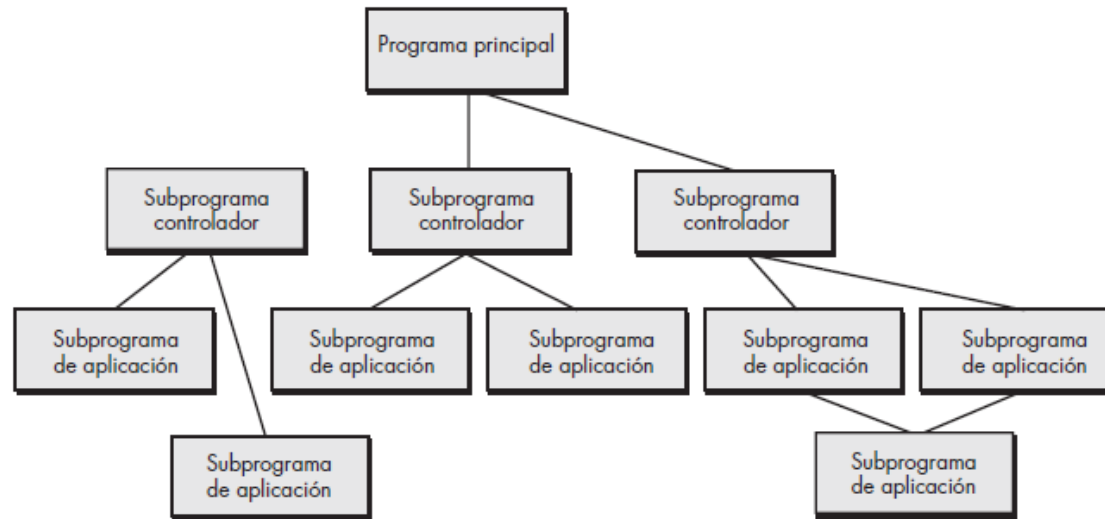


- Esta arquitectura se aplica cuando datos de entrada van a transformarse en datos de salida a través de una serie de componentes computacionales o manipuladores.
- Un patrón de tubo y filtro (véase la figura 9.2) tiene un conjunto de componentes, llamados *filtros*, conectados por *tubos* que transmiten datos de un componente al siguiente.
- Cada filtro trabaja en forma independiente de aquellos componentes que se localizan arriba o abajo del flujo; se diseña para esperar una entrada de datos de cierta forma y produce datos de salida (al filtro siguiente) en una forma especificada. Sin embargo, el filtro no requiere ningún conocimiento de los trabajos que realizan los filtros vecinos.

# Arquitectura de Software

## Estilos Arquitectónicos

### 3. Arquitecturas de llamar y regresar



*Arquitecturas de programa principal/subprograma.*

Descompone una función en una jerarquía de control en la que un programa “principal” invoca cierto número de componentes de programa que a su vez invocan a otros

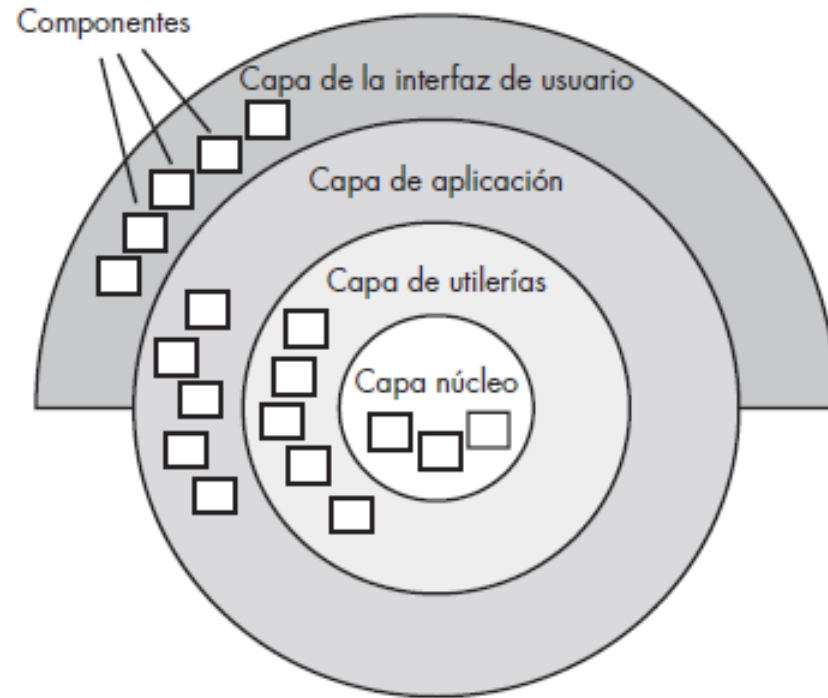
*Arquitecturas de llamada de procedimiento remoto.*

Los componentes de una arquitectura de programa principal/subprograma están distribuidos a través de computadoras múltiples en una red.

# Arquitectura de Software

## Estilos Arquitectónicos

### 4. Arquitecturas en Capas



Organiza el sistema en un conjunto de capas, cada una de las cuales provee una serie de servicios a las capas superiores usando los de las capas inferiores.

En la capa externa, los componentes atienden las operaciones de la interfaz de usuario. En la interna, los componentes realizan la interfaz con el sistema operativo. Las capas intermedias proveen servicios de utilerías y funciones de software de aplicación.