

3. Diseño de Software

Ph.D Priscilla Jiménez P.

Diseño arquitectónico

Arquitectura de
software

Decisiones en el
diseño arquitectónico

Vistas arquitectónicas

Patrones
arquitectónicos

Arquitecturas de
aplicación

Arquitectura de Software

Refiere a la estructura general del software.

Provee una visión holística del sistema que se va a construir.

Es importante porque afecta el desempeño y la potencia, así como la capacidad de distribución y mantenimiento de un sistema.

Arquitectura de Software

Propiedades que deben especificarse como parte del diseño de la arquitectura.

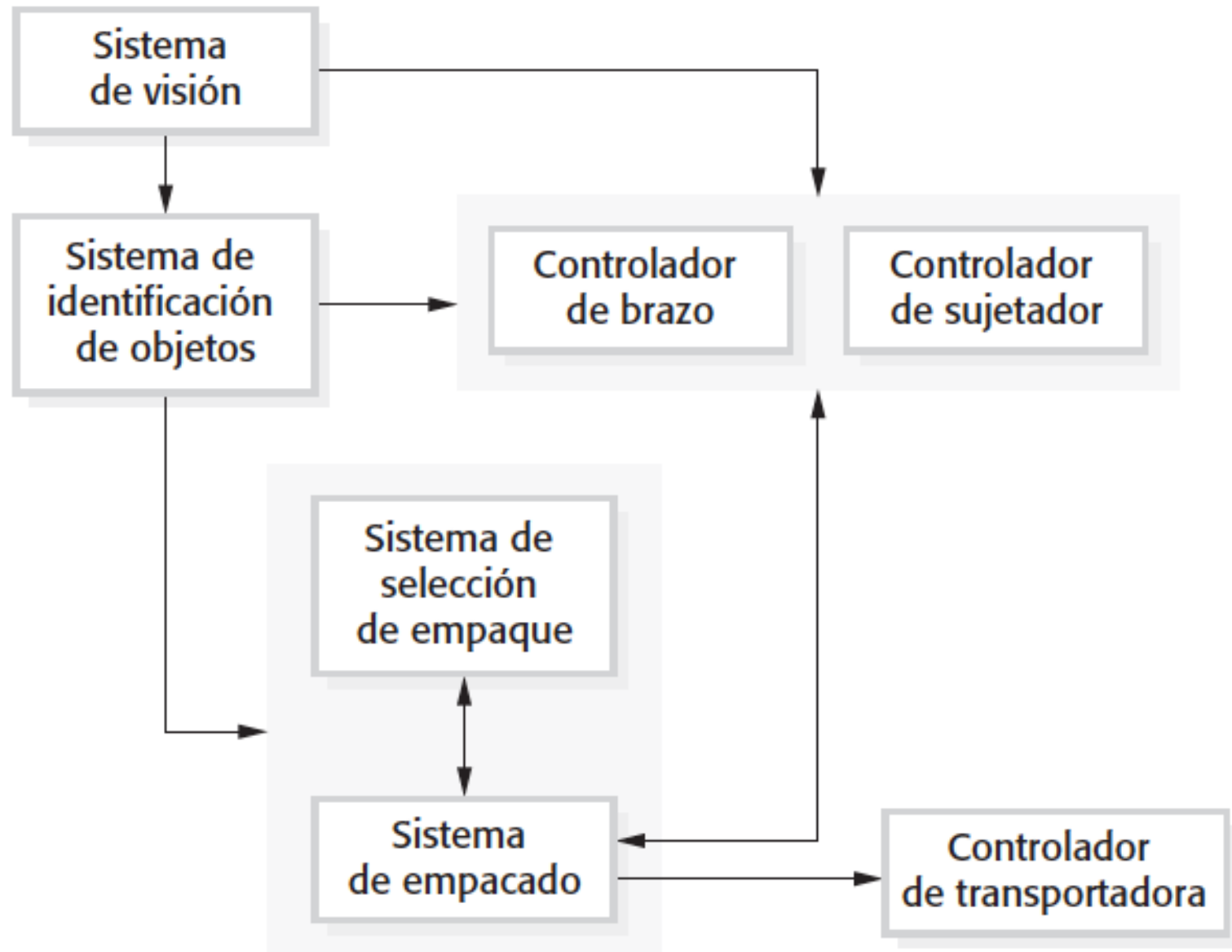
- Propiedades estructurales (módulos, objetos)
- Propiedades extra funcionales (desempeño, capacidad, confiabilidad, seguridad)
- Familias de sistemas relacionadas (patrones repetibles, reutilización)

Diseño arquitectónico

La salida del proceso de diseño arquitectónico consiste en un modelo que describe la forma en que se organiza el sistema como un conjunto de componentes en comunicación.

Debe modelar la estructura de un sistema y la manera en la que los datos y componentes del procedimiento colaboran entre sí.

Los requerimientos no funcionales dependen de la arquitectura del sistema. (comunicación y organización)



Arquitectura de un sistema de control para un robot empacador – Vista conceptual

Diseño arquitectónico

Las arquitecturas de software se diseñan en dos niveles de abstracción:

1. Arquitecturas de programas
2. Arquitecturas de sistemas complejos y distribuidos

Ventajas de diseñar y documentar la arquitectura de software (Bass et al., 2003)

1. Comunicación con los participantes
2. Análisis del sistema
3. Reutilización a gran escala

Decisiones en el diseño arquitectónico

- ¿Existe alguna arquitectura de aplicación genérica que actúe como plantilla para el sistema que se está diseñando?
- ¿Cómo se distribuirá el sistema a través de algunos núcleos o procesadores?
- ¿Cómo los componentes estructurales en el sistema se separarán en subcomponentes?
- ¿Cuál organización arquitectónica es mejor para entregar los requerimientos no funcionales del sistema?

Los sistemas son únicos, pero si están en el mismo dominio de aplicación tienen normalmente arquitecturas similares.

Decisiones en la arquitectura de software

1. Rendimiento
2. Seguridad
3. Protección
4. Disponibilidad
5. Mantenibilidad

Pensar en cada una de estas áreas como requerimiento crítico.

Vistas arquitectónicas

1. ¿Qué vistas o perspectivas son útiles al diseñar y documentar una arquitectura del sistema?
2. ¿Qué notaciones deben usarse para describir modelos arquitectónicos?

Vistas arquitectónicas (Krutchen, 1995)

1. Vista lógica
2. Vista de proceso
3. Vista de desarrollo
4. Vista física

Hofmeister et al. (2000) + Vista conceptual.

Los requerimientos de alto nivel se descomponen en especificaciones más detalladas, ayudan a tomar decisiones sobre componentes que pueden reutilizarse.

Patrones Arquitectónicos

Patrón es una forma de presentar, compartir y reutilizar el conocimiento sobre los sistemas.

El patrón arquitectónico se puede considerar como una descripción abstracta de buena práctica, que se puso a prueba en diferentes sistemas y entornos.

Debe incluir información sobre cuándo es y cuando no es adecuado usar dicho patrón, sus fortalezas, y debilidades.

Patrones Arquitectónicos

Arquitectura
Modelo Vista
Controlador (MVC)

Arquitectura de
Repositorio

Arquitectura
cliente-servidor

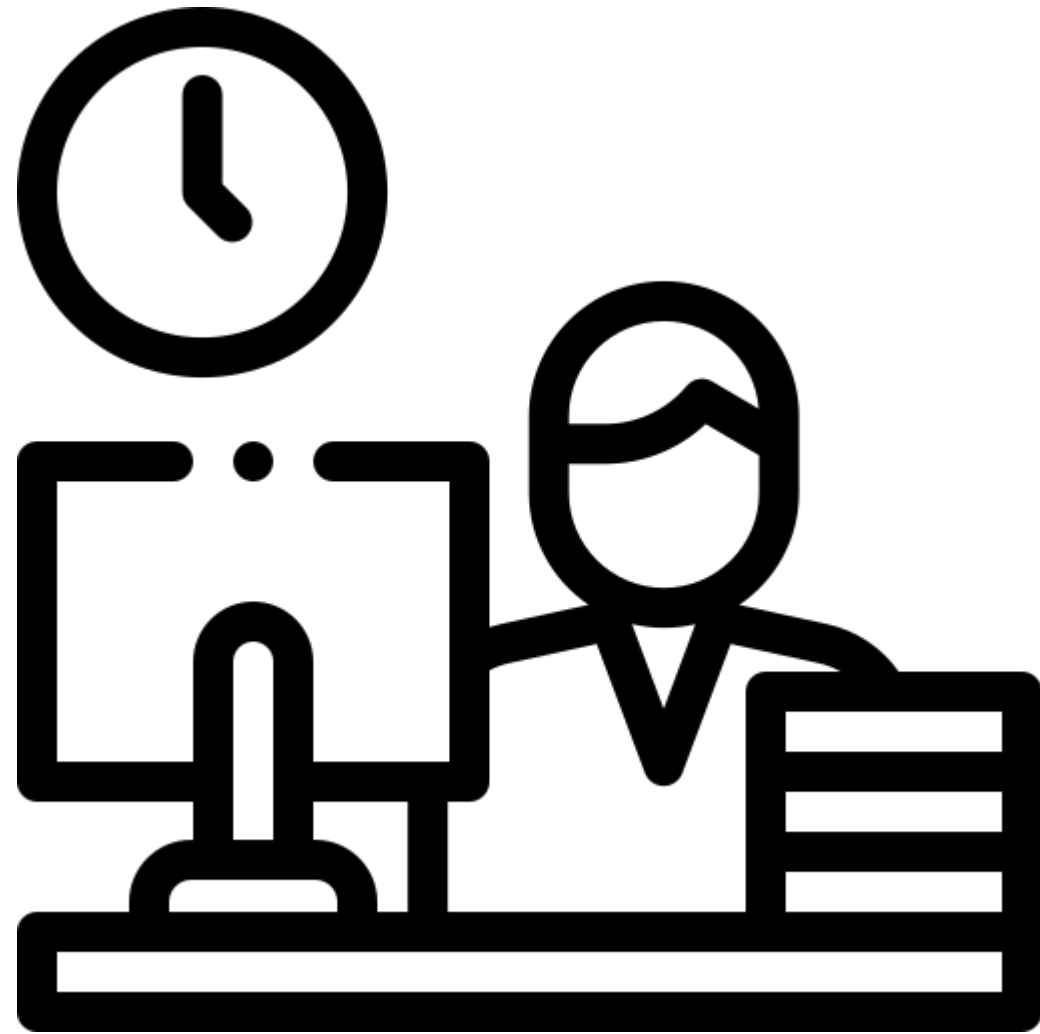
Arquitectura de
tubería y filtro

Arquitectura de
microservicios

Arquitectura en
capas

Práctica 9

Patrones arquitectónicos



Arquitectura en capas

Logra separación e independencia.

La funcionalidad del sistema está separada en capas separadas.

Es cambiable y portátil.

Cada capa se apoya sólo en las facilidades y servicios ofrecidos por la capa inmediatamente debajo de ella.

Este enfoque soporta el desarrollo incremental de sistemas.

Arquitectura N-Capas

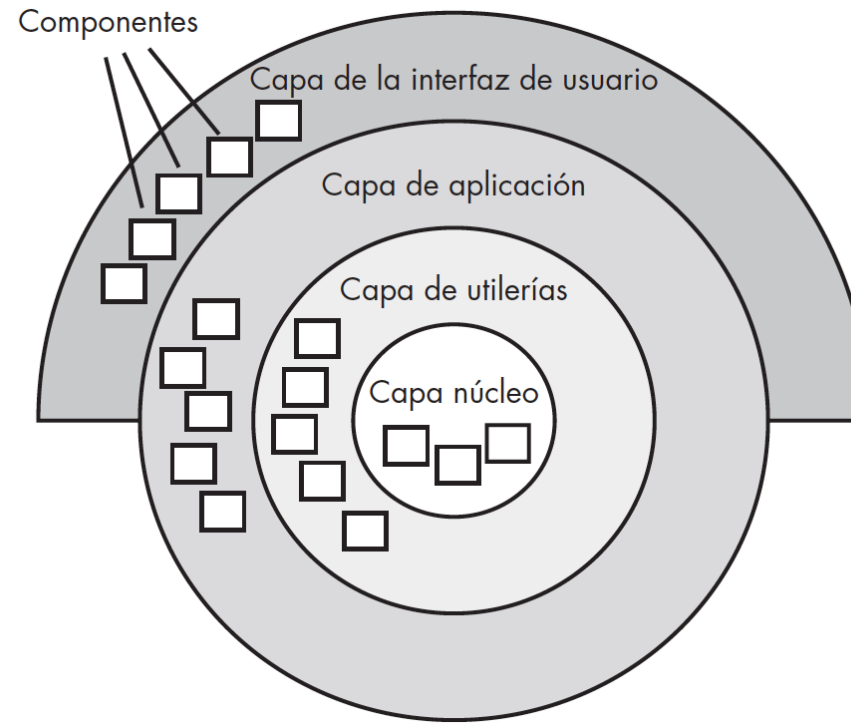
Capa N

Capa N-1

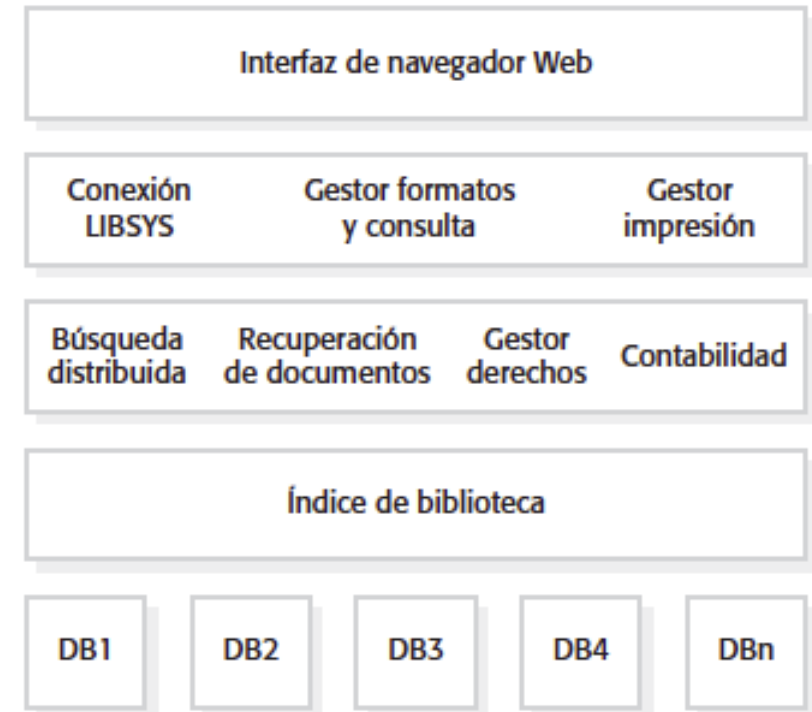
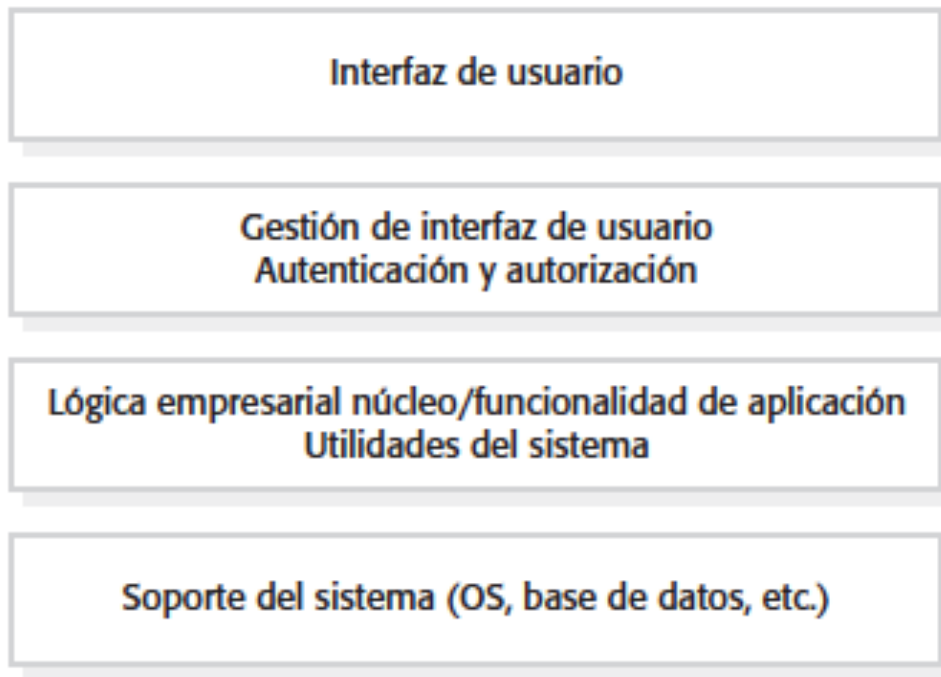
...

Capa 1





Arquitectura en capas – 4 capas



Arquitectura en Capas – 4 capas

| Nombre | Arquitectura en capas |
|---------------|---|
| Descripción | Organiza el sistema en capas con funcionalidad relacionada con cada capa. Una capa da servicios a la capa de encima, de modo que las capas de nivel inferior representan servicios núcleo que es probable se utilicen a lo largo de todo el sistema. |
| Ejemplo | Un modelo en capas de un sistema para compartir documentos con derechos de autor se tiene en diferentes bibliotecas, |
| Cuándo se usa | Se usa al construirse nuevas facilidades encima de los sistemas existentes; cuando el desarrollo se dispersa a través de varios equipos de trabajo, y cada uno es responsable de una capa de funcionalidad; cuando exista un requerimiento para seguridad multinivel. |
| Ventajas | Permite la sustitución de capas completas en tanto se conserve la interfaz. Para aumentar la confiabilidad del sistema, en cada capa pueden incluirse facilidades redundantes (por ejemplo, autenticación). |
| Desventajas | En la práctica, suele ser difícil ofrecer una separación limpia entre capas, y es posible que una capa de nivel superior deba interactuar directamente con capas de nivel inferior, en vez de que sea a través de la capa inmediatamente abajo de ella. El rendimiento suele ser un problema, debido a múltiples niveles de interpretación de una solicitud de servicio mientras se procesa en cada capa. |

Arquitectura en Capas

- Los estilos se suelen mezclar. Se puede usar un estilo diferente en cada capa.
 - Ultima (nivel bajo): arquitectura centrada en datos
 - Otra capa se puede implementar como un flujo de datos o componentes independientes.

Arquitecturas de aplicación

Los sistemas de aplicación tienen la intención de cubrir las necesidades de una empresa u organización.

Las empresas que operan en el mismo sector tienen similares necesidades de sistemas de aplicación.

Las arquitecturas de software describen la estructura y la organización de tipos particulares de sistemas de software.

Encapsulan las principales características de una clase de sistemas.

Arquitecturas de aplicación

Formas de usar modelos de arquitecturas de aplicación:

- Como punto de partida para el proceso de diseño arquitectónico
- Como lista de verificación del diseño
- Como una forma de organizar el trabajo del equipo de desarrollo
- Como un medio para valorar los componentes a reutilizar
- Como un vocabulario para hablar acerca de los tipos de aplicaciones

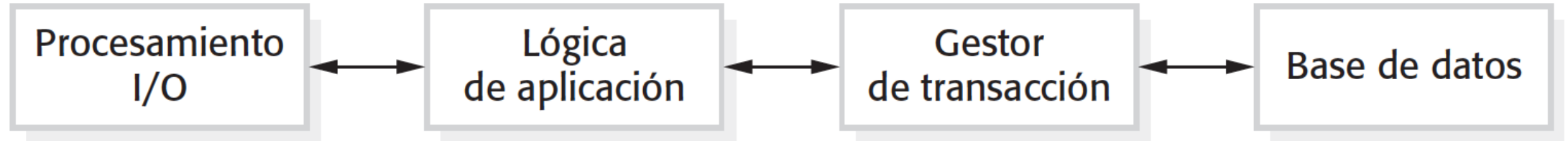
Sistemas de procesamiento de transacción

Aplicaciones centradas en bases de datos

- Sistemas bancarios interactivos
- Sistemas de comercio electrónico
- Sistemas de información
- Sistemas de reservaciones.

Un usuario hace una petición

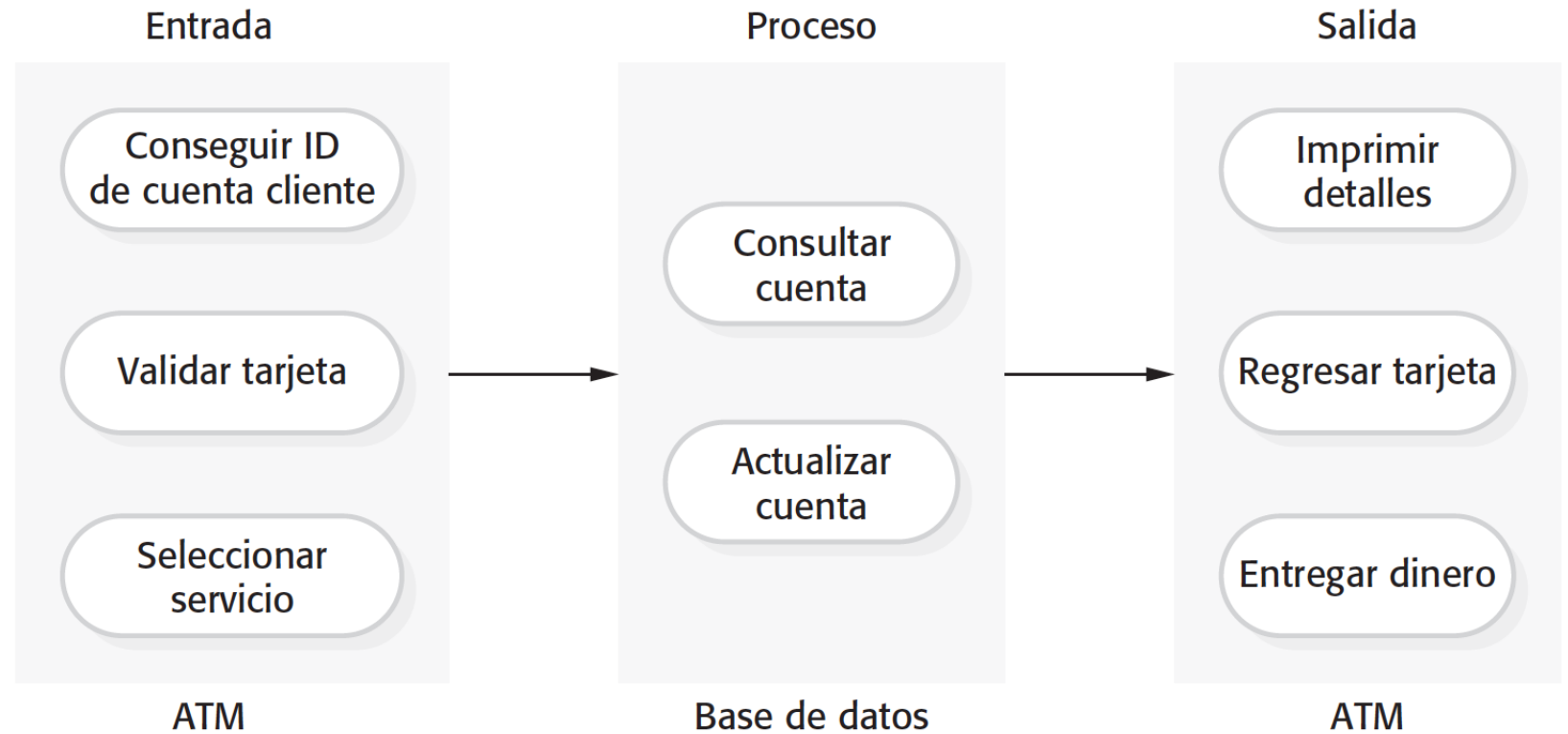
Luego de que el gestor de aplicación asegura que la transacción se ha completado, señala a la aplicación que terminó el procesamiento



La petición se procesa mediante la lógica de la aplicación
Se crea una transacción y pasa al gestor de transacción
(embebido en el sistema de manejo de la BD)

Sistemas de procesamiento de transacción

Sistemas de procesamiento de transacción – Sistema ATM



Arquitectura de Sistema ATM, ilustra las funciones de los componentes de entrada, proceso y salida.

Sistemas de Información

Permite acceso controlado a una base de información.

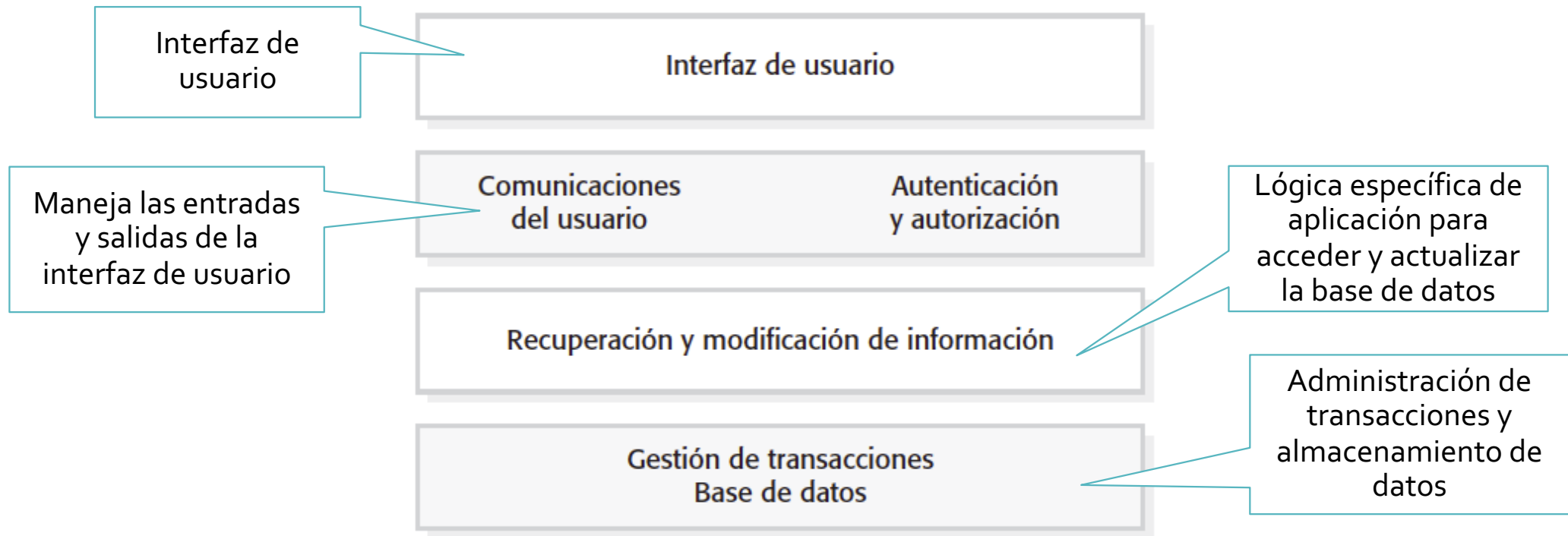
- Catálogos de biblioteca
- Horario de vuelos
- Registros de pacientes de un hospital

Arquitectura 3-Capas

Capa Presentación

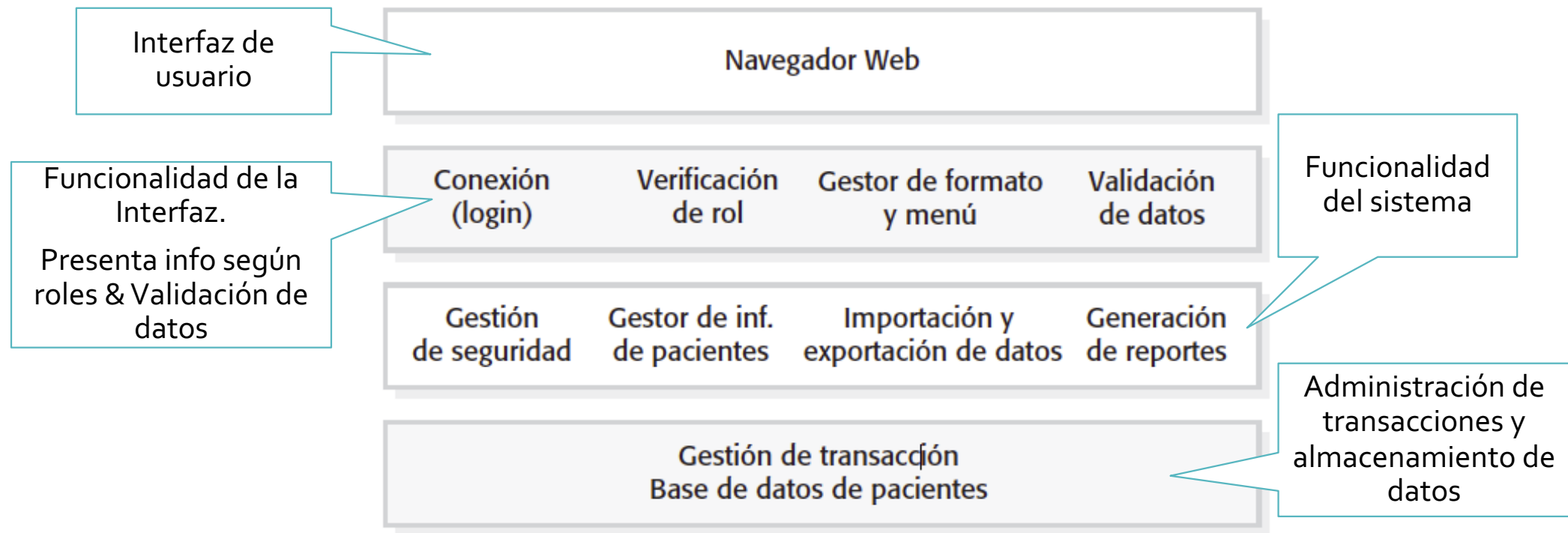
Capa Lógica / Negocio

Capa Acceso a Datos



Modelo general de un sistema de información

Enfoque por capas



Ejemplo de un modelo en capas: Sistema de Registro de Pacientes

Puntos claves

La arquitectura de software es una descripción de cómo se organiza un sistema de software.

Las decisiones de diseño arquitectónico incluyen decisiones sobre el tipo de aplicación, la distribución del sistema, los estilos arquitectónicos a usar y la forma de documentarse.

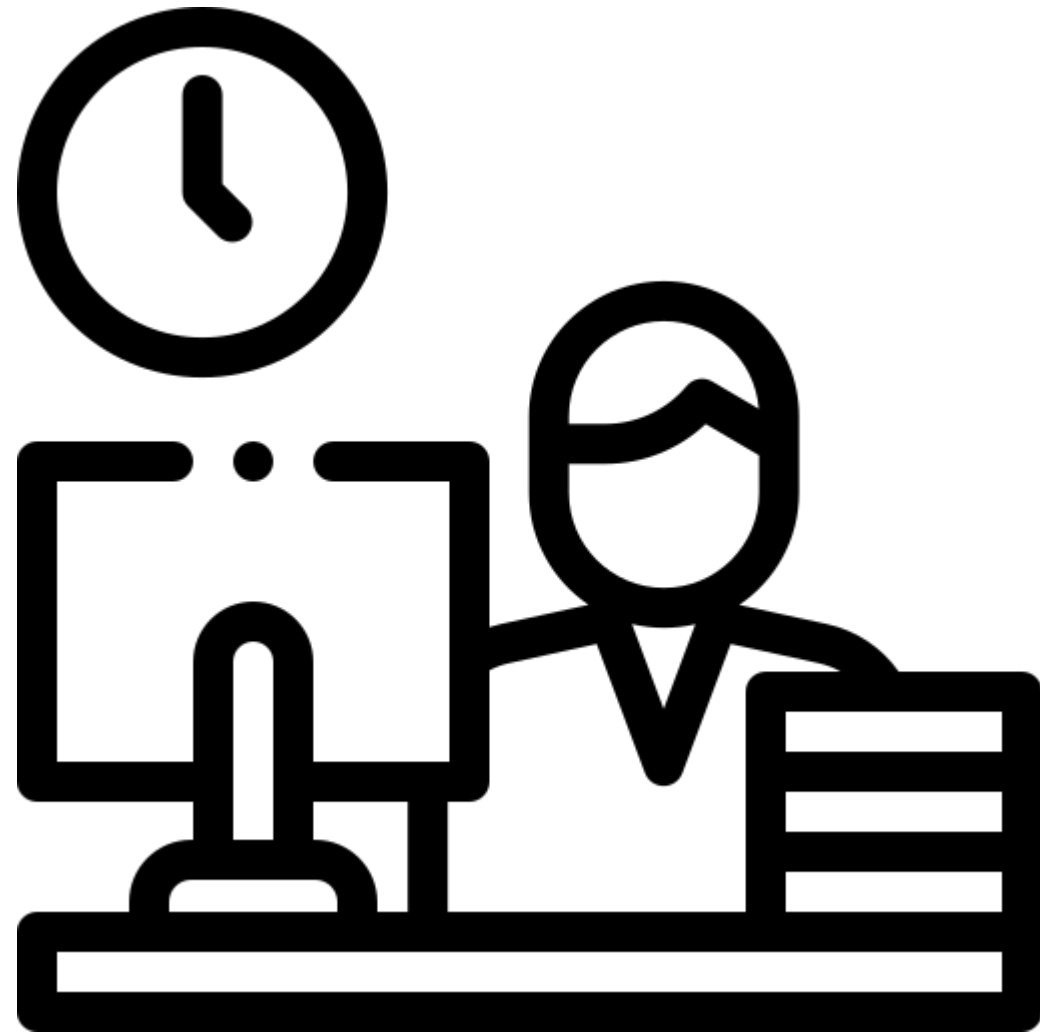
Puntos claves

Las arquitecturas pueden documentarse desde varias perspectivas: conceptual, lógica, proceso, desarrollo, y física.

Los patrones arquitectónicos son medios para reutilizar el conocimiento sobre las arquitecturas de sistemas genéricos.

Práctica 10

Arquitectura del sistema





Bibliografía y Recursos útiles

- Sommerville, I. (2011). Ingeniería de Software. 9^{na} Edición. (Cap. 6, 18).
- Pressman, R. (2010). Ingeniería del Software. Un Enfoque Práctico. 7^{ma} Edición. (Cap. 9).
- <https://www.youtube.com/watch?v=uFDT-NHPe-s>
- <https://www.youtube.com/watch?v=QxqkmOC-eQY>