



# Instituto Politécnico Nacional

## Escuela Superior de Cómputo



# DIAGRAMAS ARQUITECTÓNICOS

Investigación

*Análisis y Diseño Orientado a Objetos*

Profesor: Chadwick Carreto Arellano

Grupo: 2CM7

Fecha: 15 / Mayo / 2018

Calva Hernández José Manuel

Alumno:  
2017630201

## Contenido

<b>Introducción .....</b>	<b>2</b>
¿Qué es la arquitectura de software? .....	2
Descripciones arquitectónicas .....	2
¿Qué es el diseño arquitectónico? .....	2
¿Cuál es el producto final? .....	2
<b>Clasificación .....</b>	<b>3</b>
Arquitecturas centradas en los datos .....	3
Arquitecturas de flujo de datos.....	3
Arquitecturas de llamar y regresar.....	3
Arquitecturas orientadas a objetos .....	4
Arquitecturas en capas .....	4
<b>Patrones Arquitectónicos .....</b>	<b>5</b>
Interacción entre componentes.....	5
Interacción con el usuario .....	5
Interacción con la capa de datos .....	6
<b>Diagrama de Arquitectura de Software .....</b>	<b>7</b>
<b>Bibliografía.....</b>	<b>8</b>
<b>Webgrafía .....</b>	<b>8</b>

# Introducción

## ¿Qué es la arquitectura de software?

La arquitectura del software de un programa o sistema de cómputo es la estructura o estructuras del sistema, lo que comprende a los componentes del software, sus propiedades externas visibles y las relaciones entre ellos.

La arquitectura no es el software operativo. Es una representación que permite:

- 1) analizar la efectividad del diseño para cumplir los requerimientos establecidos,
- 2) considerar alternativas arquitectónicas en una etapa en la que hacer cambios al diseño todavía es relativamente fácil y
- 3) reducir los riesgos asociados con la construcción del software.

## Descripciones arquitectónicas

El estándar IEEE define una descripción arquitectónica (DA) como “un conjunto de productos para documentar una arquitectura”. La descripción en sí se representa con el uso de perspectivas múltiples, donde cada perspectiva es “una representación del sistema completo desde el punto de vista de un conjunto de preocupaciones relacionadas [de un participante]”. Una perspectiva se crea de acuerdo con reglas y convenciones definidas en un punto de vista: “especificación de las convenciones para construir y usar una perspectiva”.

## ¿Qué es el diseño arquitectónico?

El diseño arquitectónico representa la estructura de los datos y de los componentes del programa que se requieren para construir un sistema basado en computadora. Considera el estilo de arquitectura que adoptará el sistema, la estructura y las propiedades de los componentes que lo constituyen y las interrelaciones que ocurren entre sus componentes arquitectónicos.

## ¿Cuál es el producto final?

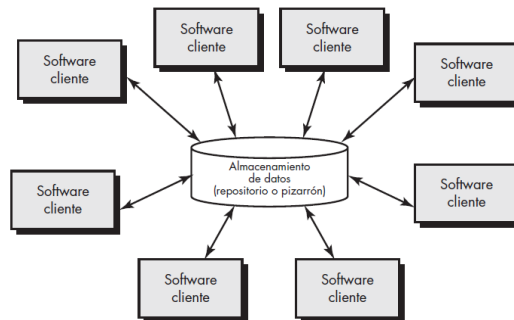
El diseño arquitectónico, da el panorama y asegura que sea el correcto. Durante el diseño arquitectónico se crea un modelo de arquitectura que incluye la arquitectura de los datos y la estructura del programa.

Además, se describen las propiedades y relaciones (interacciones) que hay entre los componentes.

### Arquitecturas centradas en los datos

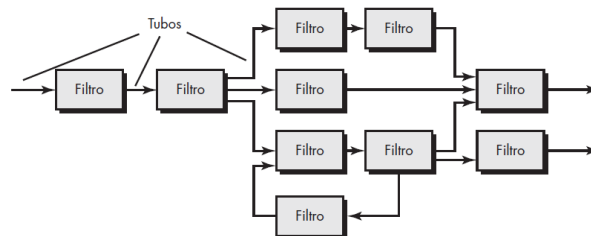
En el centro de esta arquitectura se halla un almacenamiento de datos (como un archivo o base de datos) al que acceden con frecuencia otros componentes que actualizan, agregan, eliminan o modifican los datos de cierto modo dentro del almacenamiento.

Las arquitecturas centradas en datos promueven la integrabilidad. Además, pueden pasarse datos entre clientes con el uso de un mecanismo de pizarrón (el componente pizarrón sirve para coordinar la transferencia de información entre clientes). Los componentes del cliente ejecutan los procesos con independencia.



### Arquitecturas de flujo de datos

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 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.



### Arquitecturas de llamar y regresar

Este estilo arquitectónico permite obtener una estructura de programa que es relativamente fácil de modificar y escalar. Dentro de esta categoría existen varios subestilos:

- Arquitecturas de programa principal/subprograma. Esta estructura clásica de programa 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.

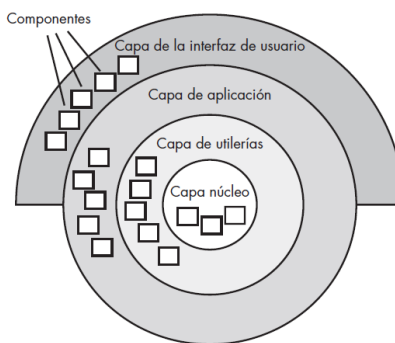
## Arquitecturas orientadas a objetos

Los componentes de un sistema incluyen datos y las operaciones que deben aplicarse para manipularlos. La comunicación y coordinación entre los componentes se consigue mediante la transmisión de mensajes.

## Arquitecturas en capas

Se define un número de capas diferentes; cada una ejecuta operaciones que se aproximan progresivamente al conjunto de instrucciones de máquina. 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.

Estos estilos arquitectónicos tan sólo son un pequeño subconjunto de los que están disponibles. Una vez que la ingeniería de requerimientos revela las características y restricciones del sistema que se va a elaborar, se elige el estilo arquitectónico o la combinación de patrones que se ajusten mejor a esas características y restricciones. En muchos casos, más de un patrón es apropiado y es posible diseñar y evaluar estilos arquitectónicos alternativos.



Lo habitual es presentar el modelo arquitectónico como un modelo de tres capas en aplicaciones de gestión, las cuales se conforman de la siguiente manera:

- a) Capa de presentación: Es la interfaz de usuario. Hace la información accesible al usuario. Responsable de presentar información e interactuar con entidades externas.
- b) Capa lógica de aplicación: Coordina la aplicación, procesa los comandos, toma decisiones, realiza los cálculos y mueve los datos entre las dos capas. Responsable de implementar las operaciones solicitadas por los clientes a la capa de presentación.
- c) Capa de gestión de recursos: Es de donde se obtiene la información y los datos. Suele ser una base de datos, ficheros externos, recursos accesibles por la web. Responsable de: gestionar todos los elementos de información del SI; ficheros planos, XML, SGBD.

# Patrones Arquitectónicos

Los patrones arquitectónicos se abocan a un problema de aplicación específica dentro de un contexto dado y sujeto a limitaciones y restricciones. El patrón propone una solución arquitectónica que sirve como base para el diseño de la arquitectura. Tienen un alcance menor que los estilos arquitectónicos (se concentran en un solo aspecto).

## Interacción entre componentes

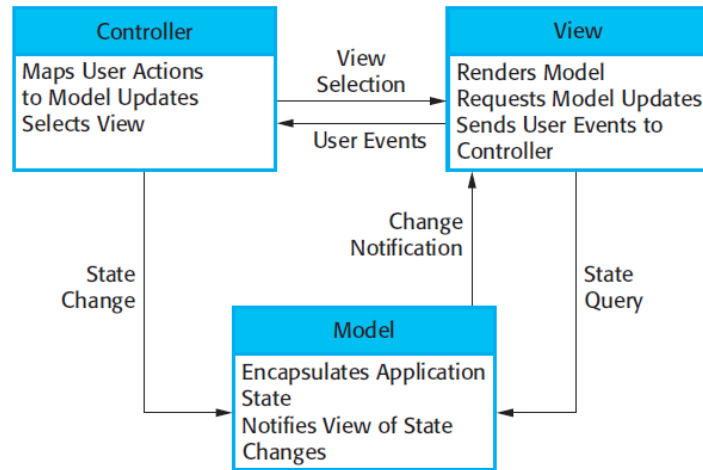
Arquitecturas x-tier: Durante el Diseño Arquitectónico la vista lógica de una arquitectura en capas (layer) conceptuales da lugar a una vista física que se materializa en una arquitectura en uno o más niveles (tier). Existen 4 tipos básicos de arquitecturas:

- **Arquitectura mononivel:** Por razones de rendimiento el resultado de implementar las tres capas se queda en un único aplicativo. Se despliega en un único host. No ofrecen acceso programático (API).  
Ventajas como eficiencia, coste casi nulo de despliegue y desarrollo en clientes.  
Inconvenientes como coste de mantenimiento de la aplicación y mainframes es una tendencia opuesta a la de clusters.
- **Arquitectura en dos niveles:** La popularización del PC hizo rentable pasar la responsabilidad de la capa de presentación al cliente. Se conoce como Cliente/Servidor.  
Ventajas como se puede aprovechar las capacidades de cómputo del cliente, permite personalizar la capa de presentación para distintos fines y portarla a distintos entornos (multiplataforma), y eficiencia en el lado del servidor.  
Inconvenientes como protocolos más complejos y gestión de sesiones complican la escalabilidad, arquitectura inadecuada cuando se necesita integrar más de un servidor.
- **Arquitectura en tres niveles:** Evolución natural de las dos capas para resolver el problema de la integración de varios servidores. La responsabilidad de integrar pasa al middleware, que también se encarga de (CORBA, X/OPEN, DCOM) transacciones, balanceo de carga, replicación, entre otros. Permiten desplegar lógica en otro host. La latencia aumenta.
- **Arquitecturas multinivel:** Es la arquitectura en n-niveles escalada tantas veces como sea necesario. La capa de recursos (datos) puede tener a su vez otra arquitectura n-nivel. Surge de manera natural cuando se desea integrar varios sistemas de información y/o se desea utilizar Internet como canal de comunicación.

## Interacción con el usuario

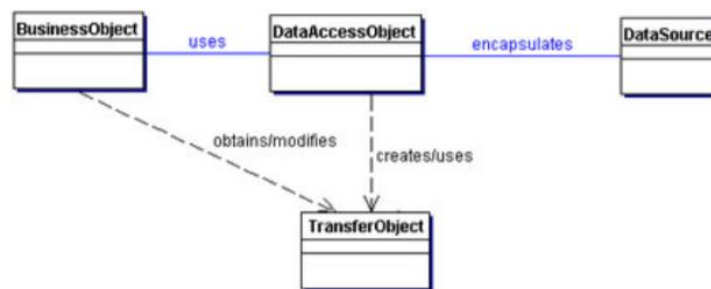
MVC (Modelo – Vista – Controlador):

- **Modelo** es la representación específica de la información con la que se opera. Incluye los datos y la lógica para operar con ellos.
- **Vista** es la presentación del modelo de forma adecuada para interactuar con ella, normalmente a través de una interfaz de usuario.
- **Controlador** responde a eventos de la interfaz de usuario e invoca cambios en el modelo y probablemente en la vista.

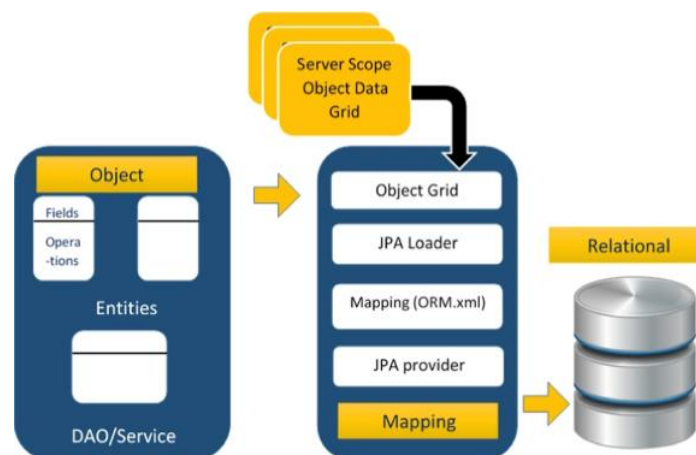


## Interacción con la capa de datos

- Data Access Object (DAO): Es un componente de software que suministra una interfaz común entre la aplicación y uno o más dispositivos de almacenamiento de datos, tales como una Base de datos o un archivo.



- Object Relational Mapping: Es una técnica de programación para convertir datos entre el sistema de tipos utilizado en un lenguaje de programación orientado a objetos y la utilización de una base de datos relacional como motor de persistencia. En la práctica esto crea una base de datos orientada a objetos virtual, sobre la base de datos relacional. Esto posibilita el uso de las características propias de la orientación a objetos (básicamente herencia y polimorfismo).



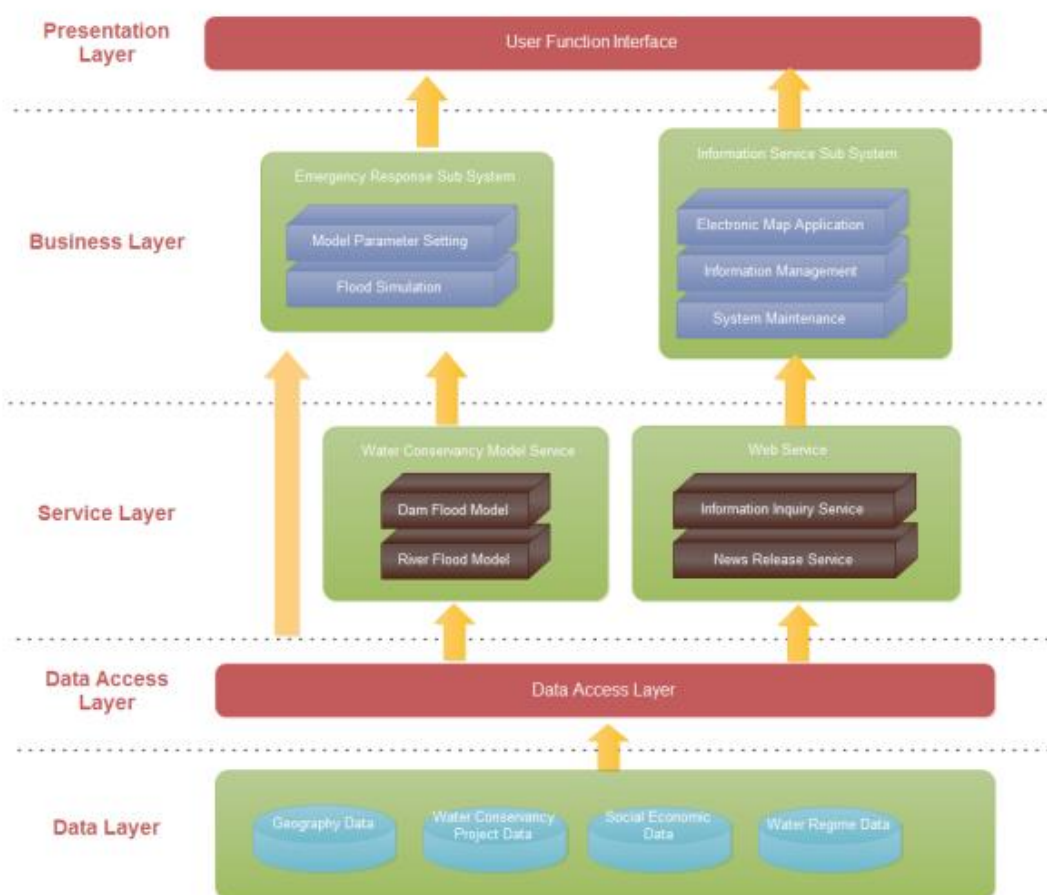
## Diagrama de Arquitectura de Software

Un diagrama de arquitectura es una colección de cuadros de colores llamativos y dibujos que están conectados por flechas. Es una representación de la más obvia y común estructura de los componentes del sistema.

El diseño de este diagrama es un paso crucial para el desarrollo de software y aplicaciones que describe la estructura básica del software dividiendo las áreas funcionales en capas. Esto representa cómo será la interacción típica del software con los usuarios, sistemas externos, datos y servicios.

Usualmente se describe en cuatro capas que son, de arriba hacia abajo, las siguientes:

- Capa de presentación: Contiene componentes para que los usuarios interactúen con la aplicación. Es la responsable de procesar las entradas de los usuarios y devolver la correcta respuesta.
- Capa de servicio: Actúa como un paso intermedio ya que contiene servicios tanto de infraestructura como de aplicación.
- Capa de negocio: Incluye el núcleo del modelo de funcionamiento de la aplicación.
- Capa de datos: Es la última capa de la aplicación, es la responsable de la comunicación con el almacenamiento de datos.





## Bibliografía

Software Architecture Diagramming Tool. (2018). Retrieved from <https://www.edrawsoft.com/software-architecture.php>

Sommerville, I. (2011). *Software engineering* (9th ed.). Boston: Pearson.

## Webgrafía

Escuela Técnica Superior de Ingeniería Informática. (2013). *Diseño arquitectónico*. Sevilla: Universidad de Sevilla.  
Retrieved from [https://scontent.fmex2-1.fna.fbcdn.net/v/t1.0-9/32675390\\_181402516017500\\_6286522602995318784\\_n.jpg?\\_nc\\_cat=0&oh=ae4eb812d923f6d5992b8bf9d42839a6&oe=5B86426F](https://scontent.fmex2-1.fna.fbcdn.net/v/t1.0-9/32675390_181402516017500_6286522602995318784_n.jpg?_nc_cat=0&oh=ae4eb812d923f6d5992b8bf9d42839a6&oe=5B86426F)

Pressman, R. (2010). *Ingeniería de Software. Un enfoque práctico* (7th ed.). Ciudad de México: Mc Graw Hill Educación.

Rogachevsky, A. (2016). *What is an architecture diagram?* Retrieved from <https://www.quora.com/What-is-an-architecture-diagram>

Ruíz, F., & López, P. (2010). *Arquitectura Lógica del Sistema (en desarrollo OO)*. Retrieved from <https://www.istr.unican.es/asignaturas/is1/is1-t11-trans.pdf>