

# ARQUITECTURA DE SOFTWARE

## Vistas Arquitectónicas

Eric Gustavo Coronel Castillo  
[www.desarrollasoftware.com](http://www.desarrollasoftware.com)  
[gcoronelc@gmail.com](mailto:gcoronelc@gmail.com)



# Temas

- Introducción
- Elementos
- Propuestas Arquitectónicas
- Modelo 4+1
- Recomendaciones
- Referencias



- Las vistas arquitectónicas representan un aspecto parcial de una arquitectura de software que muestran propiedades específicas del sistema.
- La representación única, de la arquitectura del sistema resultaría demasiada compleja, pues contendría mucha información.
- Descripción simplificada o abstracción de un sistema desde una perspectiva específica, que cubre intereses particulares y omite entidades no relevantes a esta perspectiva.
- Cada vista representa un comportamiento particular del sistema.

# Elementos

---

- Punto de vista de los involucrados e intereses de los mismos.
- Elementos que serán capturados y representados en la vista y las relaciones entre estos.
- Forma en que se relacionan los elementos de una vista con otras vistas.

# Propuestas Arquitectónicas

---

- Vista arquitectónica de John Zachman.
  - Matriz de 36 celdas, que incluye 6 punto de vista o perspectivas, iniciando desde un nivel alto de abstracción del negocio y llegando hasta el de implementación.
- Vista arquitectónica de Philippe Kruchten.
  - Propuso el modelo “4+1” , vinculado al Rational Unified Process (RUP), que define cuatro vistas diferentes de la arquitectura de software.

## Propuestas Arquitectónicas

---

- Vista arquitectónica de Grady Booch, James Rumbaugh e Ivar Jacobson
  - En la introducción a UML, formularon un esquema de cinco vistas interrelacionadas que conforman la arquitectura de software.
    - Vista de casos de uso
    - Vista de diseño
    - Vista de procesos
    - Vista de implementación
    - Vista de despliegue

## Modelo 4+1

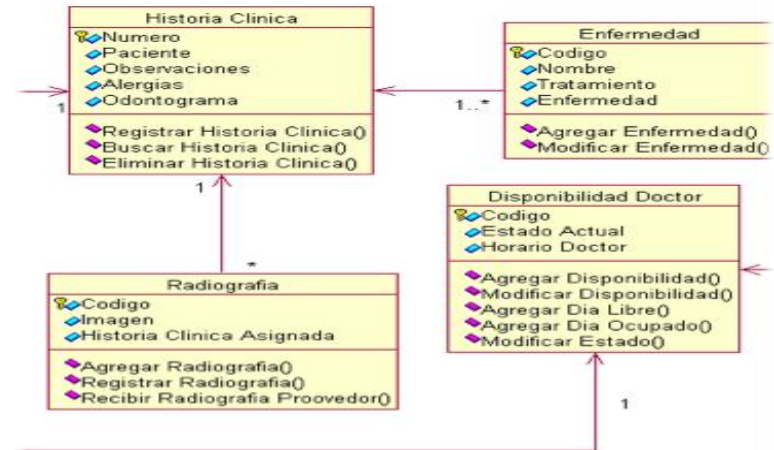
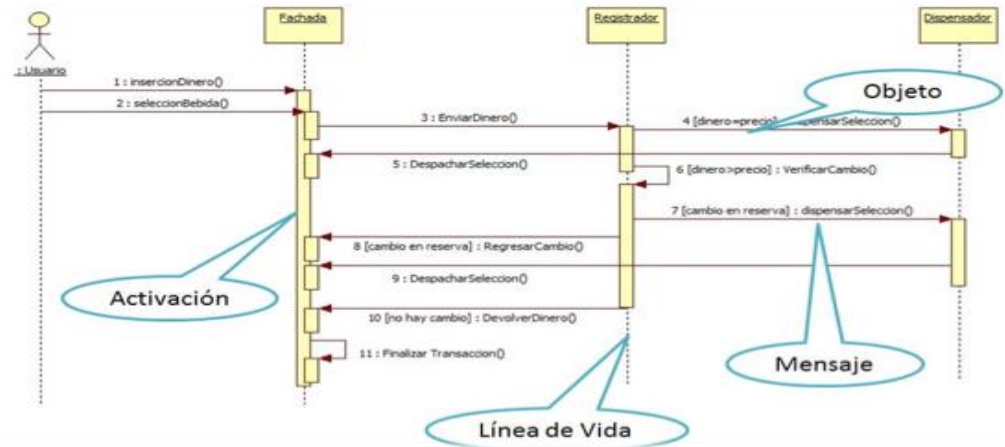
Propone 5 vistas, que ha sido el referente hasta la fecha. El nombre de este modelo describe el número de vistas que en él se proponen, 4 vistas son consideradas como ortogonales, y la otra es usada para vincular a las demás, en total son 5 vistas.



# Modelo 4+1

## Vista Lógica

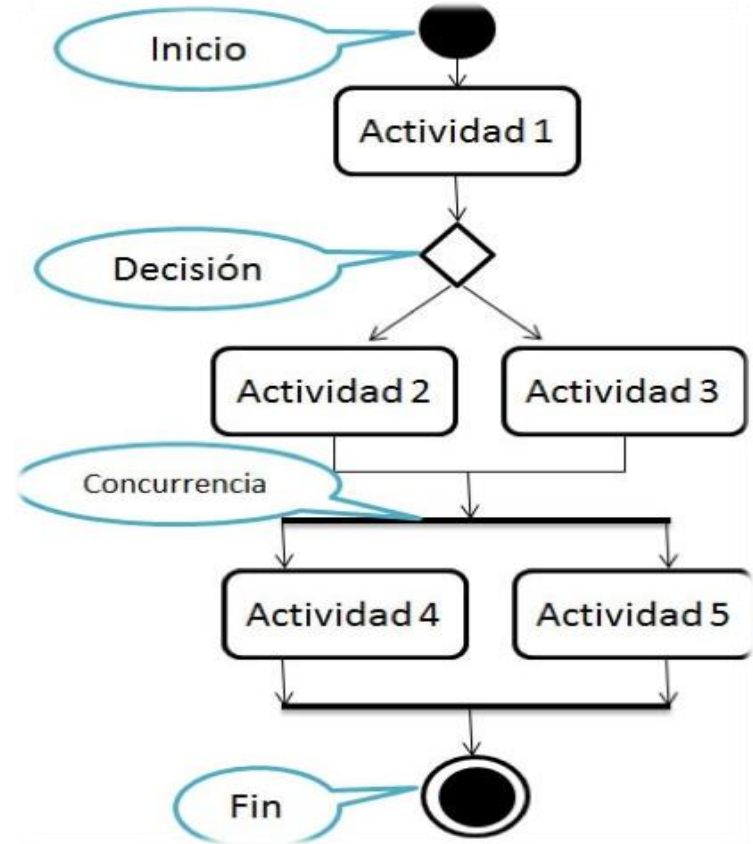
- Es una abstracción del modelo de diseño e identificación a gran escala del diseño de paquetes, subsistemas y clases
- Diagramas de Clases y Objetos
- Diagramas ER
- Se complementa con vistas del Área Dinámica
  - Diagramas de Actividad,
  - Diagramas de Interacción,
  - Diagramas de Estado.





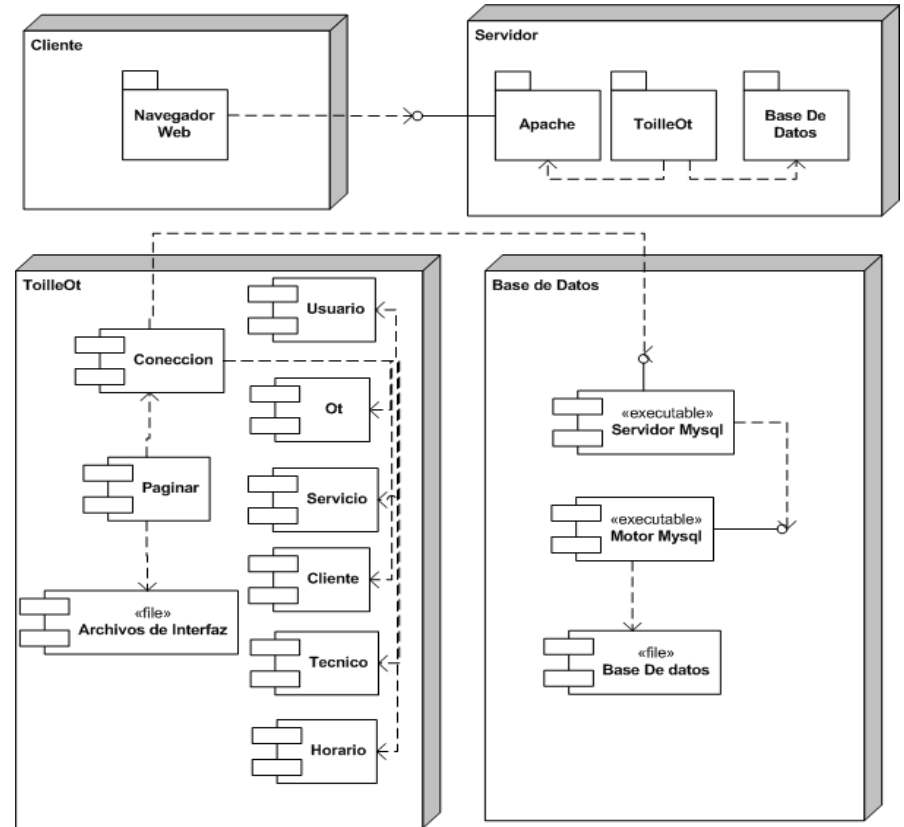
## Vista de Procesos

- Toma en cuenta algunos requerimientos no-funcionales: Rendimiento, disponibilidad, integridad del sistema, tolerancia a fallas.
- Captura aspectos de Sincronización y Concurrencia del diseño.
- Control de los procesos concurrentes.



## Vista de Despliegue

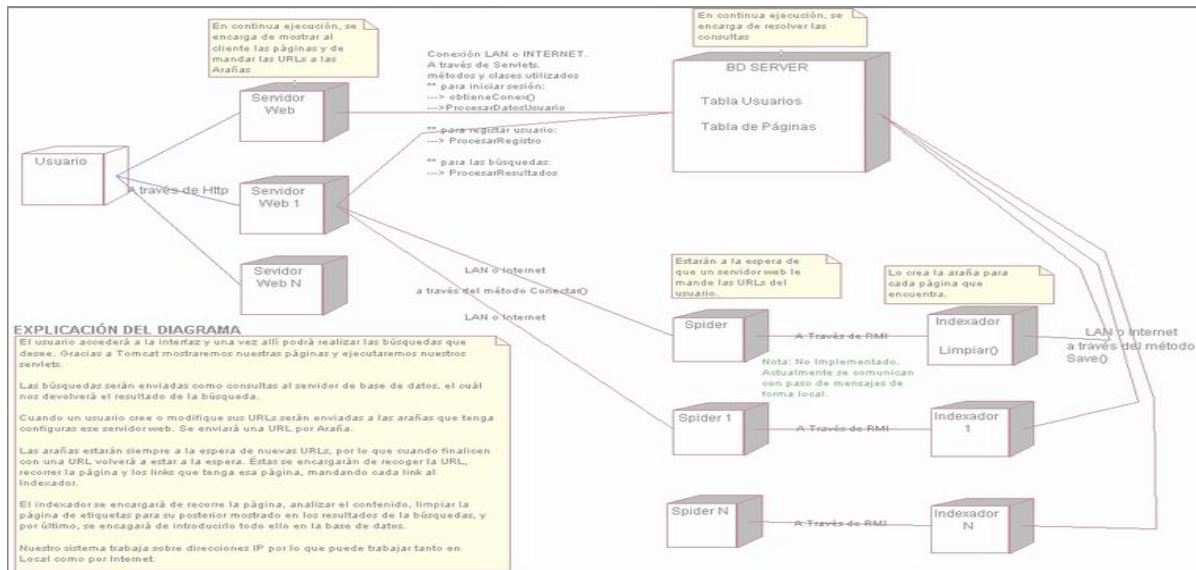
- La vista de despliegue se enfoca en la organización de los módulos del software actual en el ambiente de desarrollo de software.
- Diagramas de Componentes
- Se complementa con vistas del Área Dinámica
  - Diagramas de Actividad,
  - Diagramas de Interacción,
  - Diagramas de Estado.



# Modelo 4+1

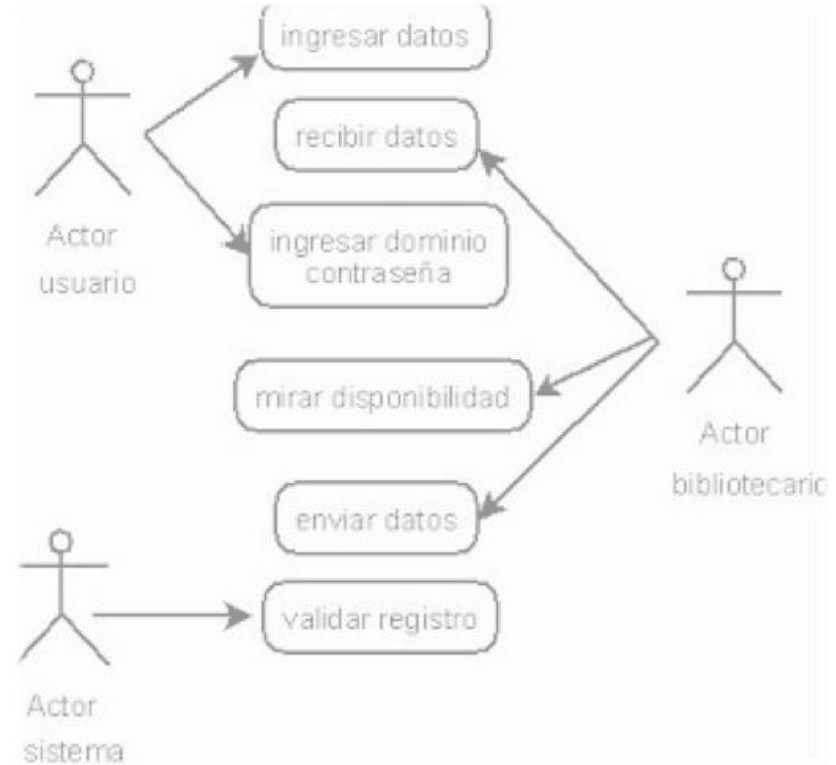
## Vista Física

- La vista física se centra en los requisitos no funcionales, tales como la disponibilidad del sistema, la fiabilidad (tolerancia a fallos), ejecución y escalabilidad.
- Diagrama de despliegue.
- Básicamente este tipo de diagrama se utiliza para modelar el Hardware utilizado en la implementación del sistema y la relaciones entre sus componentes.
- Los elementos usados por este tipo de diagrama son nodos, componentes y asociaciones.



## Vista de Escenarios

- Esta vista contiene los escenarios o casos de uso claves, para cada uno de los cuales se describen las secuencias de interacción entre objetos y procesos.
- Diagramas de Casos de Uso



## Recomendaciones

- Documentar las vistas que sean de mayor utilidad
- Documentar tomando en cuenta los intereses de los involucrados.
- En sistemas muy grandes utilizar paquetes de vistas.

## Para seleccionar las vistas:

- Elaborar una lista de las vistas candidatas.
- Combinar las vistas
- Priorizar las vistas

# Referencias

---

- Ana Orellana, Verónica Velasteguj; EVALUACION DE LA ARQUITECTURA DE SOFTWARE DE APLICACIONES DE PRODUCCION; Mayo 2007; 1.4 Vistas Arquitectónicas pág 11 - 18; enlace: <http://bibdigital.epn.edu.ec/bitstream/15000/414/1/CD-0818.pdf>
- Anónimo; Arquitectura de Software; Usar Arquitecturas basadas en Componentes; Octubre 2007; enlace <http://ldc.usb.ve/~mgoncalves/IS2/sd07/clase7.pdf>
- Anomino; Arquitectura del Software; Diseño de la Arquitectura; Octubre 2013; enlace: [http://www.slideshare.net/landeta\\_p/2-1-vistas-arquitectonicas#](http://www.slideshare.net/landeta_p/2-1-vistas-arquitectonicas#)
- Kruchten P. Architectural Blueprints—The “4+1” View Model of Software Architecture. IEEE Software, November 1995, 12 (6), pp.42-50.
- Erika Camacho, Fabio Cardeso, Gabriel Nuñez; Arquitecturas de Software; 4.1 Estilo Arquitectónico 19 - 21; Abril 2004; enlace: <http://prof.usb.ve/lmendoza/Documentos/PS-6116/Guia%20Arquitectura%20v.2.pdf>
- Ian Sommerville; Ingeniería de Software; Diseño Arquitectónico; 6.2 Vistas arquitectónicas pág 153 – 155
- Javier Garzas; Modelo 4+1; enlace: <https://sites.google.com/site/jgarzas/4mas1>



# ARQUITECTURA DE SOFTWARE

**Gracias**

**Eric Gustavo Coronel Castillo**

[www.desarrollasoftware.com](http://www.desarrollasoftware.com)

[gcoronelc@gmail.com](mailto:gcoronelc@gmail.com)

