

# Arquitectura de Software

Puntos de Vista

Departamento de Ingeniería de Sistemas y Computación



# Agenda del día



1. El proceso de definición de arquitectura

2. Viewpoints / Views

3. Ejercicio

# 1. El proceso de Definición de la Arquitectura

- Normalmente comienza
  - Con un conocimiento inicial del problema y su alcance
  - Sin conocer el tamaño total del producto
  - Sin conocer todos los riesgos
  - Sin entender los conflictos internos entre los Stakeholders

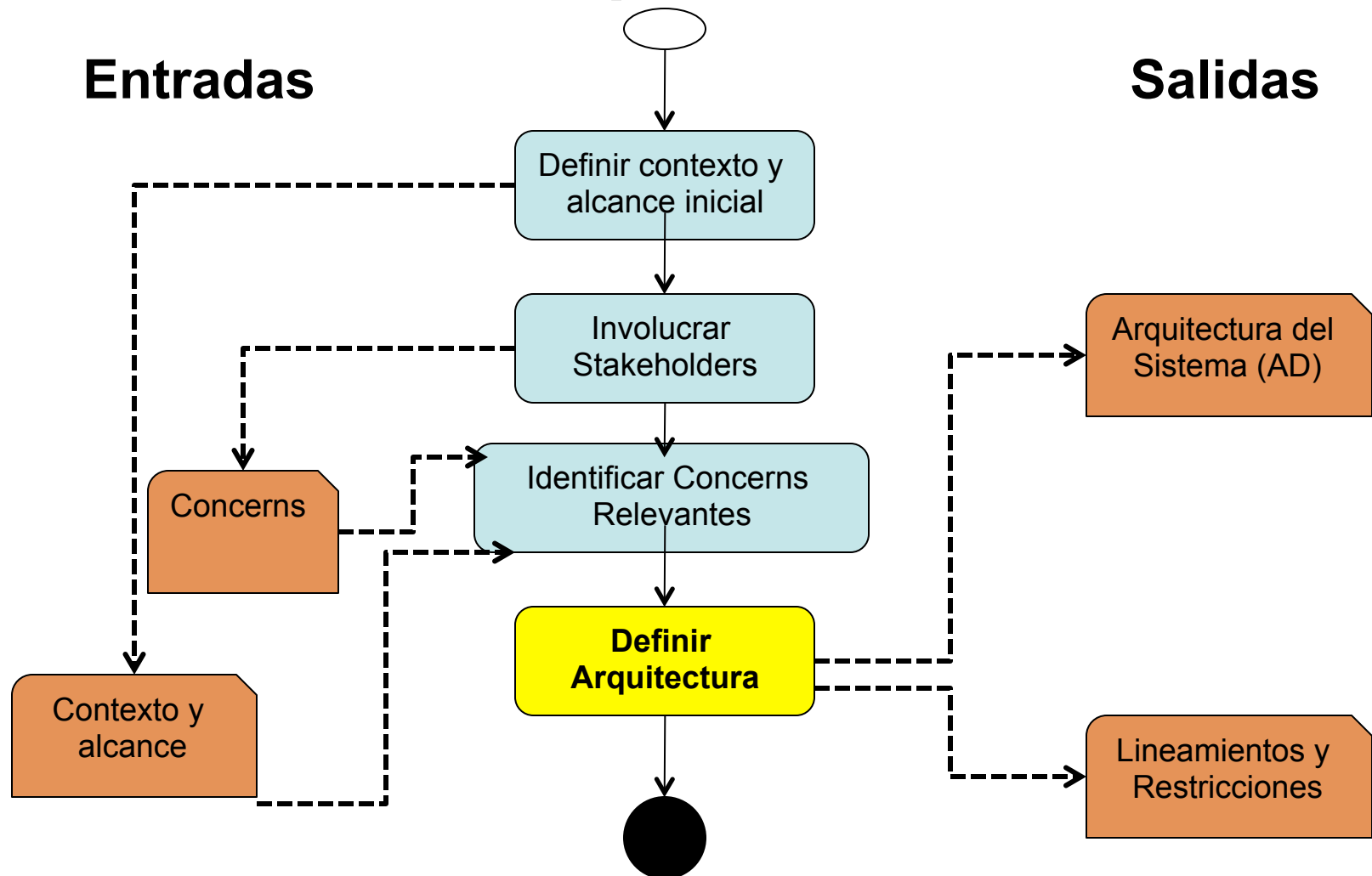
# 1. El proceso de Definición de la Arquitectura

- Principios
  - Basado en las necesidades de los Stakeholders
  - Comunicación de las decisiones
  - Debe ser un proceso estructurado
  - Debe ser pragmatico (tiempo, dinero, etc.)
  - Debe ser flexible
  - Debe ser independiente de tecnología
  - Debe ser integrado al proceso de desarrollo
  - Estar alineado con las políticas de calidad

# 1. El proceso de Definición de la Arquitectura

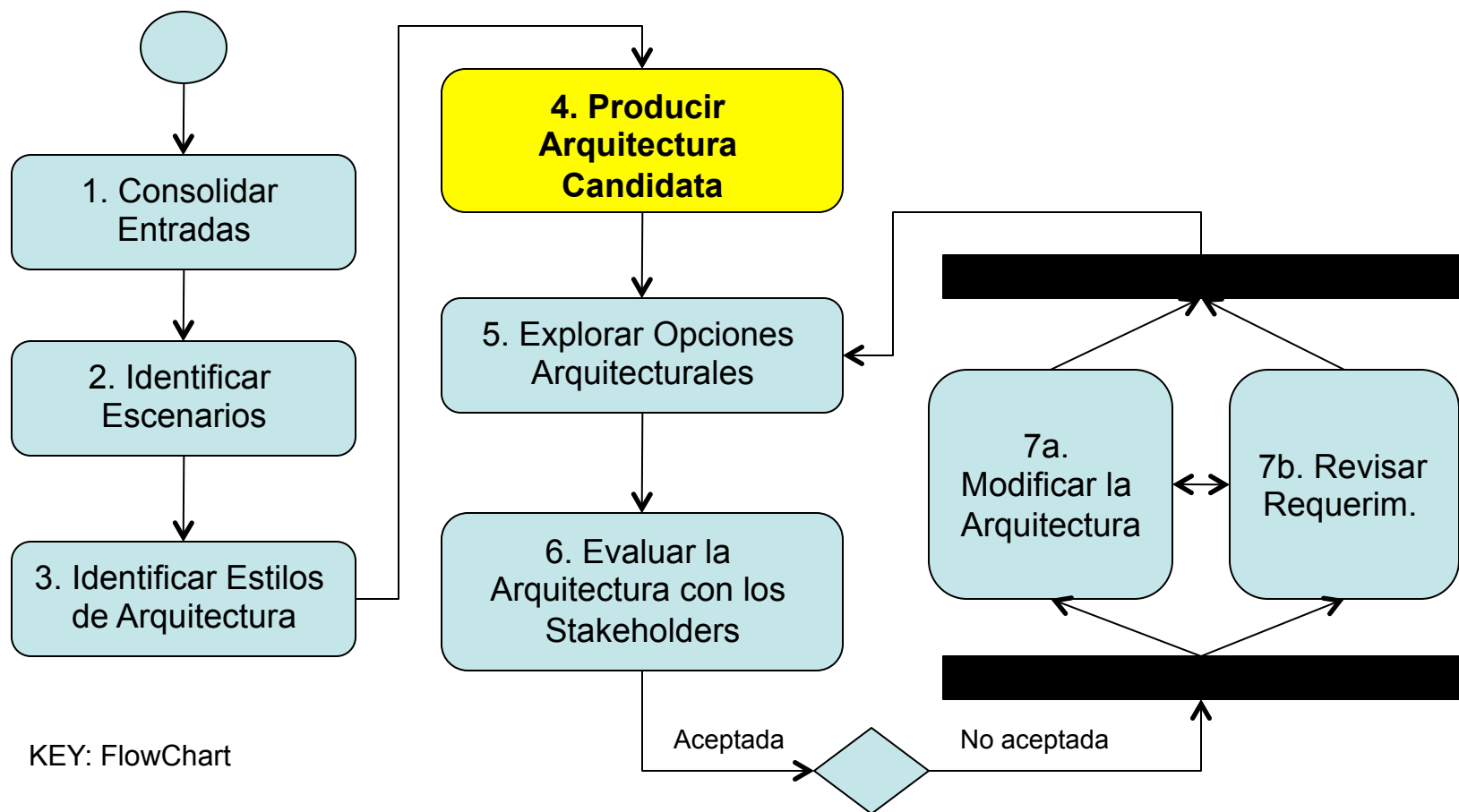
- Productos Esperados
  - Principal
    - Documento de Arquitectura
  - Secundarios
    - Claridad en los requerimientos
    - Expectativas de los Stakeholders
    - Identificación y evaluación de posibles arquitecturas
    - Criterios de aceptación de la arquitectura
    - Un conjunto de artefactos para iniciar el diseño detallado

# 1. El proceso de Definición de la Arquitectura



# 1. El proceso de Definición de la Arquitectura

- Definición de la arquitectura del sistema



KEY: FlowChart

# 1. El proceso de Definición de la Arquitectura

- Producir Arquitectura Candidata
  - Objetivos
    - Producir una primera versión de la arquitectura
  - Entradas
    - Concerns
    - Architectural Styles
    - **Viewpoints**
    - **Perspectives**
  - Salidas
    - Architectural Views (Draft)



# Agenda



1. El proceso de definición de arquitectura



2. Viewpoints / Views

3. Ejercicio

## 2. Viewpoints / Views

- Algunas preguntas relevantes durante la definición de la arquitectura
  - Cuáles son los principales elementos funcionales?
  - Cómo interactúan entre ellos?
  - Qué información almacenar y presentar?
  - Qué hardware y software se va a necesitar?
  - Cómo serán los ambientes de desarrollo, pruebas y producción ?

## 2. Viewpoints / Views

- Opciones para responder estas y otras preguntas
  - Hacer un único modelo 
  - Utilizar las vistas arquitecturales 

## 2. Viewpoints / Views

### Estrategia

Describir un sistema complejo, utilizando un conjunto de **vistas** interrelacionadas para ilustrar sus características funcionales y propiedades de calidad.[1]

## 2. Viewpoints / Views

“ A **view** is a representation of one or more structural aspects of an architecture that illustrates how the architecture addresses one or more **concerns** held by one or more **stakeholders**” [1]

## 2. Viewpoints / Views

- Qué información incluir en una vista?
  - A cuál stakeholder esta orientada?
  - Qué tanto entiende de tecnología el stakeholder?
  - Cuáles concerns se pretenden resolver en dicha vista?

## 2. Viewpoints / Views

“ A **viewpoint** is a collection of patterns, templates, and conventions for constructing one type of **view**. It defines the **stakeholders** whose **concerns** are reflected in the viewpoint and the guidelines, principles, and template models for constructing its views” [1]

## 2. Viewpoints / Views

- Beneficios al utilizar vistas
  - Separación de Concerns
  - Comunicación con los Stakeholders
  - Reducción de complejidad
  - Facilidad en el diseño y desarrollo del sistema

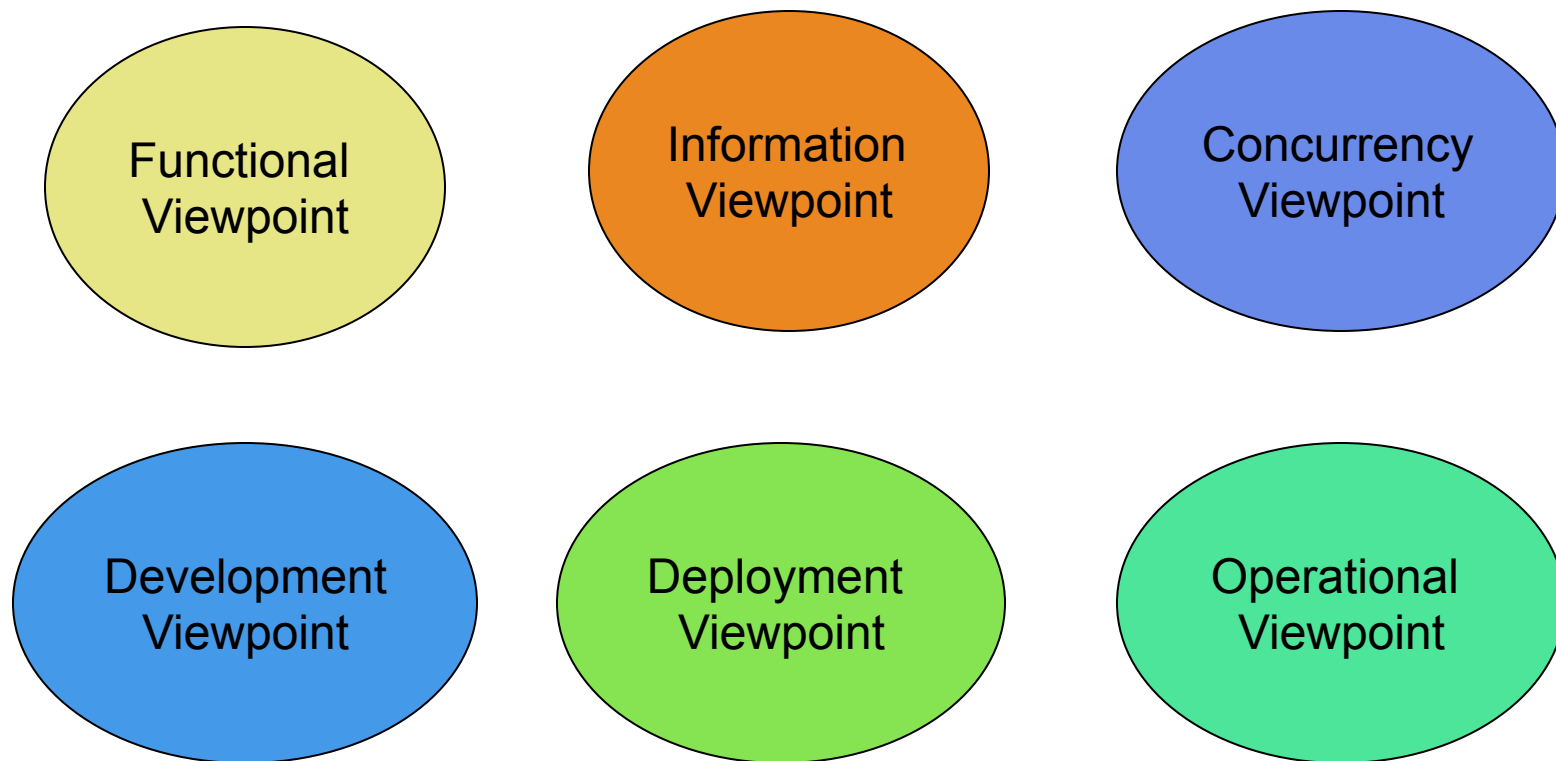


## 2. Viewpoints / Views

- Problemas al utilizar vistas
  - Inconsistencia entre las vistas
  - Error en la selección de las vistas
  - Fragmentación de la información

## 2. Viewpoints / Views

- El catálogo de Puntos de vista <sup>[1]</sup>



## 2. Viewpoints / Views

Viewpoint	Propósito
<b>Functional</b>	Describir los elementos funcionales del sistema, sus responsabilidades, interfaces e interacciones.
<b>Information</b>	Describe la manera como la arquitectura guarda, manipula, maneja y distribuye información.
<b>Concurrency</b>	Describe la estructura de concurrencia del sistema y relaciona elementos funcionales con unidades de concurrencia y como serán coordinadas y controladas.
<b>Development</b>	Describe la arquitectura que soporta el proceso de desarrollo del sistema.
<b>Deployment</b>	Describe el ambiente dentro del cual el sistema será instalado, incluyendo las dependencias con el ambiente de ejecución, incluyendo una correspondencia de los elementos de software con el ambiente de su ejecución.
<b>Operational</b>	Describe como el sistema será operado, administrado y soportado cuando este ejecutandose en su ambiente de

# Agenda

1. El proceso de definición de arquitectura

2. Viewpoints / Views



3. Ejercicio

# Caso de Estudio WWW

- Para cada una de las vistas identifique:
  - Que *concerns* abordan?
  - Si las pudiera clasificar en alguna de los 6 tipos propuestos, cuál sería, y porqué?
  - A cuál (es) stakeholders les puede interesar cada vista?
  - Qué tipo de elementos arquitecturales presenta?

# Caso de Estudio WWW

- Para cada una de las vistas identifique:
  - Cuáles son sus responsabilidades?
  - Qué conjunto de interfaces deben ofrecer?
  - Qué relaciones existen entre los elementos?
  - Corresponden a estructuras estáticas o dinámicas?

# Caso de Estudio WWW

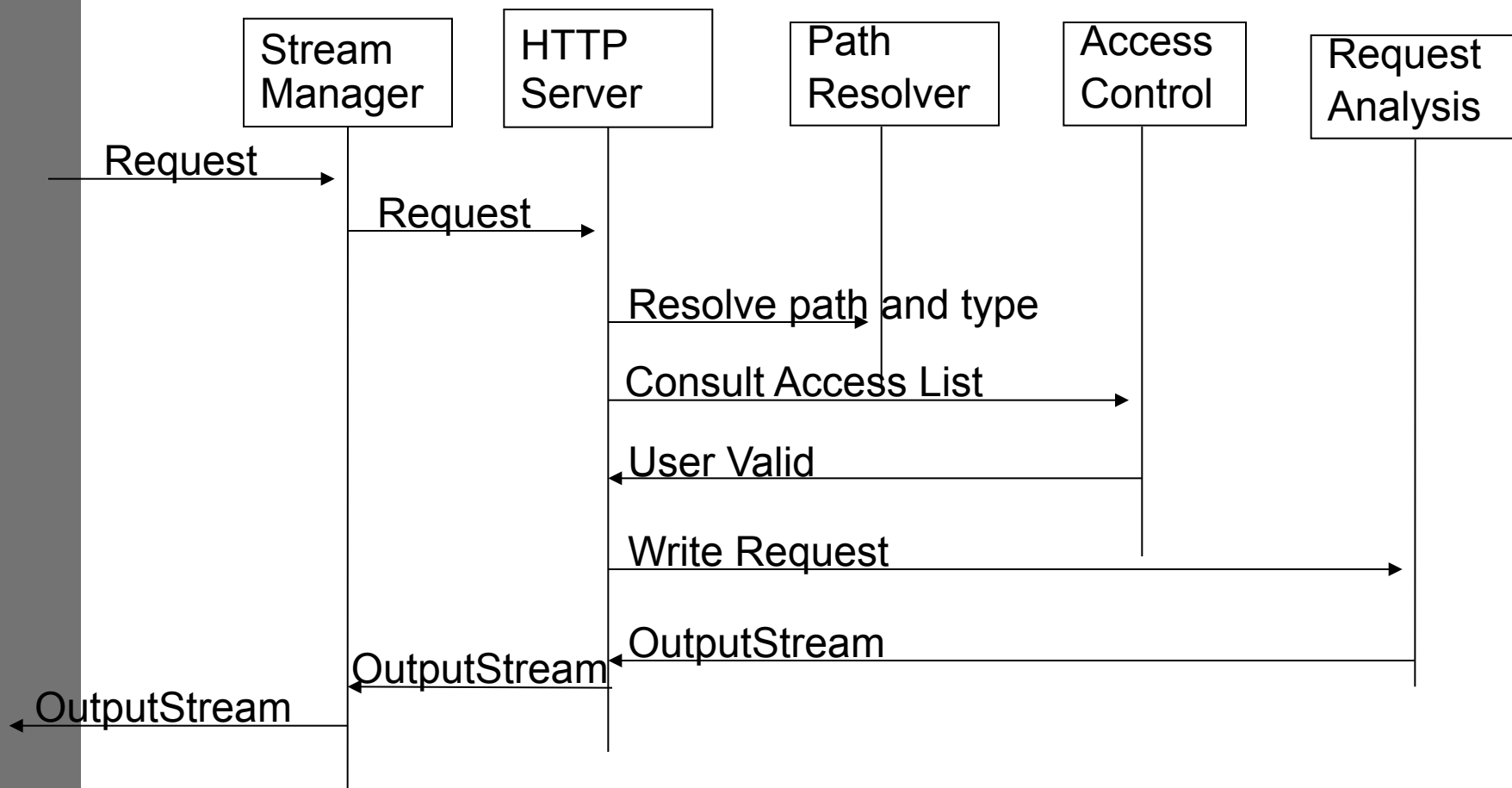
- El primer párrafo de la pagina 339 describe el proceso de manejo de una solicitud de un CGI por parte del servidor.
  - Qué relación puede encontrar entre la descripción del proceso y los elementos los elementos de la figura 13.5?

# Caso de Estudio WWW

- Construya una diagrama de secuencia del flujo de datos y de control entre los varios componentes del servidor HTTP para cumplir la solicitud (vista dinámica)
  - Comience el diagrama con un evento del HTTP Client llamado “request”



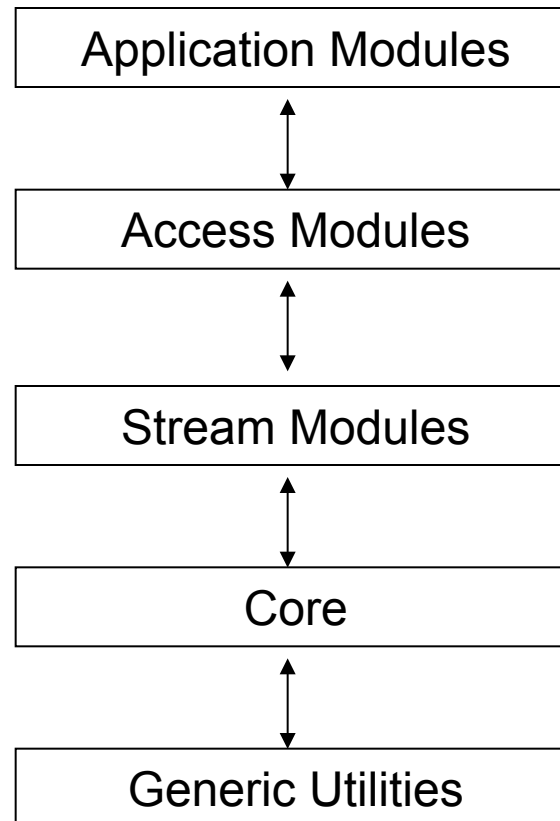
# Diagrama de Secuencia



# Caso WWW

- **Relacione los componentes** de la figura 13.5 con las capas de las que dependen para ejecutar sus servicios.
- Piense en el tipo de servicios que debe usar cada componente de la figura 13.5 y que capa debería ser la encargada de ofrecerlos.

# Vista por capas de libWWW



# Vista Despliegue

## HTTP Client

Access  
Control

Presentation  
Manager

Cache  
Manager

UI  
Manager

Stream  
Manager

Protocol  
Manager

## HTTP Server

Logging

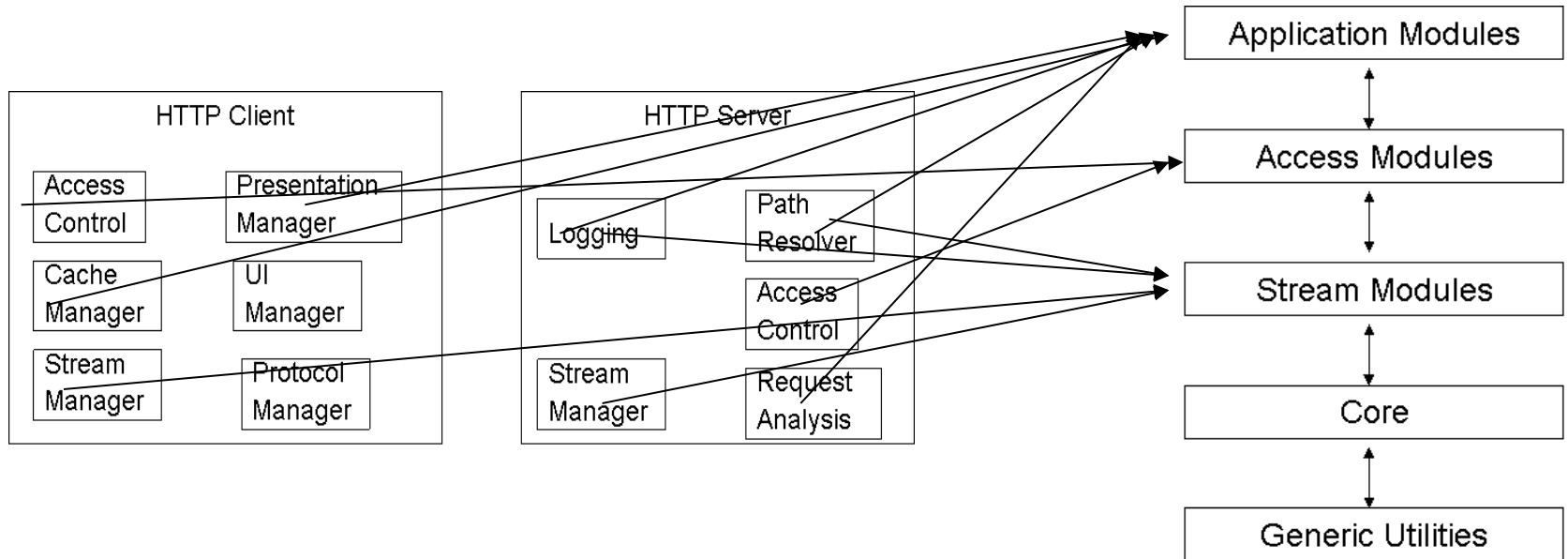
Path  
Resolver

Access  
Control

Stream  
Manager

Request  
Analysis

# Relaciones



- Material preparado por
  - Darío Correal
  - Nicolás López

# Bibliografía

- [1] Rozanski N, Woods E. “Software Systems Architecture” Addison-Wesley. 2005