

*Arquitectura del Software*  
*Ingeniería en Informática*  
*Profa. Darjeling Silva*  
*Modulo II: Marcos de Trabajo*  
*Frameworks*

***Introducción***

Luego de establecer conceptualmente hacia donde la arquitectura del software (AS) nos lleva a analizar en los sistemas, es importante concretar que la AS es un proceso de creación de sistemas, donde se diseña bajo un nivel más abstracto a nivel algorítmico y de estructura de dato. Es decir, es el diseño estructural a grandes rasgos del control, los protocolos de comunicación, la sincronización, el acceso a los datos, la funcionalidad, la distribución física, su diseño, la composición del diseño, el rendimiento, la escalabilidad y la selección de nuevas alternativas de diseño. Esto nos lleva a que AS es una aplicación sistémica para el desarrollo, aplicación o mantenimiento de un sistema, clasificándolos por modelos, donde estudiaremos los modelos estructurales (ADLs), los modelos Framework, los modelos dinámicos, los modelos de procesos y los modelos funcionales.

***Objetivo:*** Conocer los conceptos básicos relacionados con la arquitectura de software.

***Contenido:***

1. Definiciones.
2. Lenguajes de Descripción Arquitectónica (ADLs).
3. Frameworks y Vistas, Procesos y Metodologías.
4. Abstracción, Escenarios, Campos de la Arquitectura de Software

**1. Definiciones.**

***Arquitectura del software:*** Paul Clements, 1996: “La AS es, a grandes rasgos, una vista del sistema que incluye los componentes principales del mismo, la conducta de esos componentes según se la percibe desde el resto del sistema y las formas en que los componentes interactúan y se coordinan para alcanzar la misión del sistema. La vista arquitectónica es una vista abstracta, aportando el más alto nivel de comprensión y la supresión o diferimiento del detalle inherente a la mayor parte de las abstracciones”.

***Componente:*** Un componente es una cosa, una entidad.

***Modelo estructural:*** está compuesta por componentes, conexiones entre ellos y (usualmente) otros aspectos tales como configuración, estilo, restricciones, semántica, análisis, propiedades, racionalizaciones, requerimientos, necesidades de los participantes. El trabajo en esta área está caracterizado por el desarrollo de Lenguajes de Descripción Arquitectónica (ADLs).

***Modelo Framework:*** Son similares a la vista estructural, pero su énfasis primario radica en la (usualmente una sola) estructura coherente del sistema completo, en vez de concentrarse en su composición. Los modelos de framework a menudo se refieren a dominios o clases de problemas específicos. El trabajo que ejemplifica esta variante incluye arquitecturas de software específicas de dominios, como CORBA, o modelos basados en CORBA, o repositorios de componentes específicos, como PRISM.

**Modelos dinámicos:** Enfatizan la cualidad conductual de los sistemas. “Dinámico” puede referirse a los cambios en la configuración del sistema, o a la dinámica involucrada en el progreso de la computación, tales como valores cambiantes de datos.

**Modelos de proceso:** Se concentran en la construcción de la arquitectura, y en los pasos o procesos involucrados en esa construcción. En esta perspectiva, la arquitectura es el resultado de seguir un argumento (script) de proceso. Esta vista se ejemplifica con el actual trabajo sobre programación de procesos para derivar arquitecturas.

**Modelos funcionales:** Una minoría considera la arquitectura como un conjunto de componentes funcionales, organizados en capas que proporcionan servicios hacia arriba. Es tal vez útil pensar en esta visión como un framework particular.

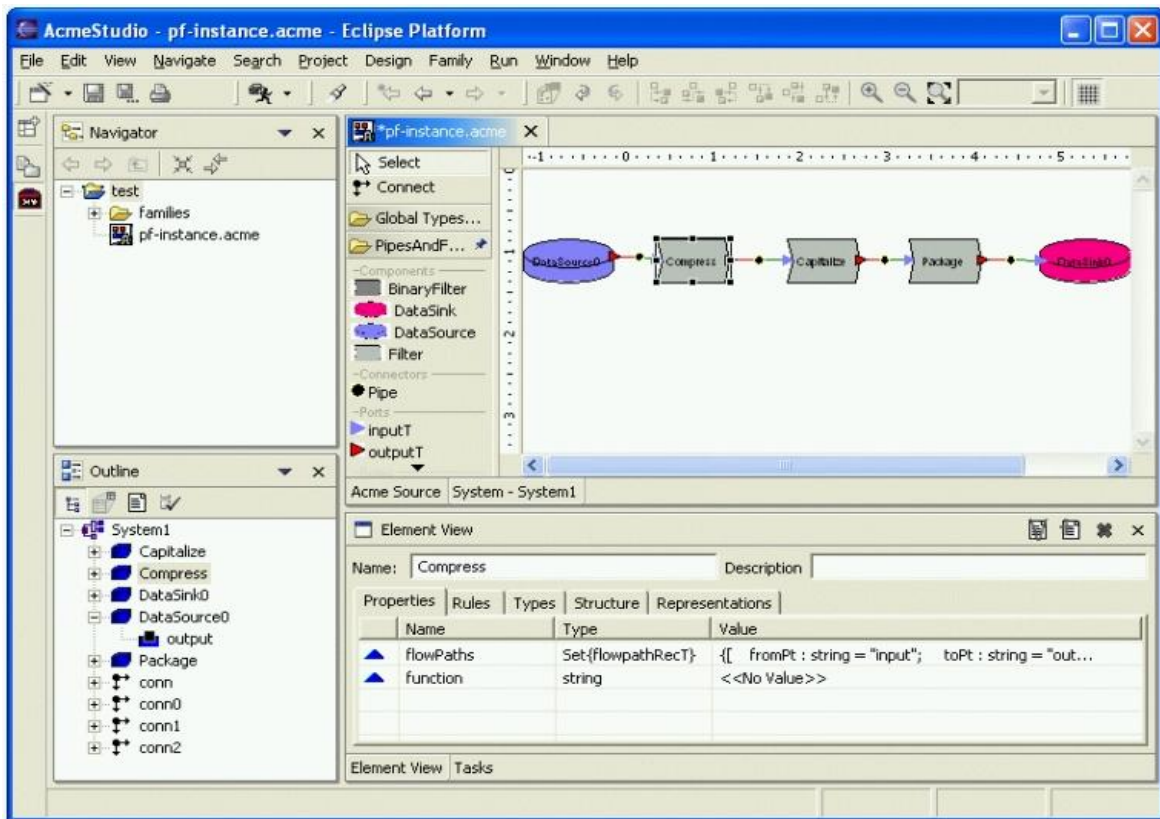
**Estilo:** una forma de articulación u organización arquitectónica de un sistema. Los estilos conjugan elementos (o “componentes”), conectores, configuraciones y restricciones.

**Patrón:** es una solución a un problema en un contexto donde se codifica y se estructura un sistema, es decir un patrón está lleno de patrones, que nos van describiendo el ambiente y el núcleo de un sistema.

## 2. Lenguajes de Descripción Arquitectónica (ADLs).

Los lenguajes de descripción de arquitecturas, o ADLs, es un lenguaje para el modelado, la descripción, la evaluación y prueba de una arquitectura de un sistema, representada en componentes, conectores, configuraciones y restricciones. Podemos señalar que sus comienzos parten de 1990 y va acompañado con los proyectos de: unificación de los lenguajes de modelado (UML); Lenguajes de especificación (LARCH, Z); Lenguajes de prototipado (Modechart, PSDL); Lenguajes de programación (CODE, Ada) y Herramientas para definición de ciclo de vida (UNAS/SALE).

En si los ADLs nacieron de estudios universitarios, debido a que permiten modelar una arquitectura mucho antes que se lleve acabo la programación de las aplicaciones que lo van a componer, en el se analiza y se determinan los puntos criticos y poder simular su comportamiento. Existen muchos ADLs, como los son Acme, Aesop, ArTek, Armani, C2 SADL, CHAM, Darwin, Jacal, LILEANNA, MetaH, Rapide, SADL, UML, UniCon, Wright, xADL. Ahora bien observemos un ejemplo de ellos tenemos Acme/ Armani:



### 3. Frameworks y Vistas, Procesos y Metodologías.

Los frameworks dentro de AS es simplemente las vistas de los sistemas o niveles de abstracción. No obstante existen organismos de estandares que nos homogenizan la AS (ISO, CEN, IEEE, OMG) y nos los presentan en una gran variedad de proyectos y modelos como los son: RM-ODP, RUP, RDS, MDA, MOF, MEMO, XMI o IEEE 1471-2000. Ahora bien, para Rich Hilliard en 1999 observa los frameworks o marcos de referencia, y las resume en la siguiente tabla:

Zachman (Niveles)	TOGAF (Arquitecturas)	4+1 (Vistas)	[BRJ99] (Vistas)	POSA (Vistas)	Microsoft (Vistas)
Alcance	Negocios	Lógica	Diseño	Lógica	Lógica
Empresa	Datos	Proceso	Proceso	Proceso	Conceptual
Sistema lógico	Aplicación	Física	Implementación	Física	Física
Tecnología	Tecnología	Desarrollo	Despliegue	Desarrollo	
Representación		Casos de uso	Casos de uso		
Funcionamiento					

Tabla 1 - Vistas en los marcos de referencia

Estas vistas las podemos resumir:

- John Zachman, identifica 36 vistas en la arquitectura (“celdas”) basadas en seis niveles (scope, empresa,sistema lógico, tecnología, representación detallada y funcionamiento empresarial) y seis aspectos (datos, función, red, gente, tiempo, motivación).

- TOGAF. completada en diciembre de 1997, es de las vistas arquitectónicas homologadas son la Operacional (que identifica relaciones y necesidades de información), la de Sistemas (que vincula capacidades y características a requerimientos operacionales) y la Técnica (que prescribe estándares y convenciones). The Open Group (TOGAF) reconoce cuatro componentes principales, uno de los cuales es un framework de alto nivel que a su vez define cuatro vistas: Arquitectura de Negocios, Arquitectura de Datos/Información, Arquitectura de Aplicación y Arquitectura Tecnológica. The Open Group propone un modelo de descripción arquitectónica, Architecture Description Method (ADM) que se supone independiente de las técnicas de modelado, aunque en la versión 7 se propone Metis como herramienta.
- El modelo “4+1”, sus comienzos para 1995 y propuesto por Philippe Kruchten, no obstante esta también vinculado con el proyecto Rational Unified Process (RUP), el cual define cuatro vistas diferentes de la arquitectura de software: (1) La vista lógica, que comprende las abstracciones fundamentales del sistema a partir del dominio de problemas. (2) La vista de proceso: el conjunto de procesos de ejecución independiente a partir de las abstracciones anteriores. (3) La vista física: un mapeado del software sobre el hardware. (4) La vista de desarrollo: la organización estática de módulos en el entorno de desarrollo. El quinto elemento considera todos los anteriores en el contexto de casos de uso.
- El modelo (BRJ99) es el propuesto por Grady Booch, James Rumbaugh e Ivar Jacobson en 1999 lo denominaron UML, es una arquitectura que se caracteriza por cinco vistas (1) la organización de un sistema de software; (2) la selección de elementos estructurales y sus interfaces a través de los cuales se constituye el sistema; (3) su comportamiento, según resulta de las colaboraciones entre esos elementos; (4) la composición de esos elementos estructurales y de comportamiento en subsistemas progresivamente mayores; (5) el estilo arquitectónico que guía esta organización: los elementos estáticos y dinámicos y sus interfaces, sus colaboraciones y su composición. Los autores proporcionan luego un esquema de cinco vistas posibles de la arquitectura de un sistema: (1) La vista de casos de uso, como la perciben los usuarios, analistas y encargados de las pruebas; (2) la vista de diseño que comprende las clases, interfaces y colaboraciones que forman el vocabulario del problema y su solución; (3) la vista de procesos que conforman los hilos y procesos que forman los mecanismos de sincronización y concurrencia; (4) la vista de implementación que incluye los componentes y archivos sobre el sistema físico; (5) la vista de despliegue que comprende los nodos que forma la topología de hardware sobre la que se ejecuta el sistema.

Área	Vista	Diagramas	Conceptos principales
Estructural	Vista estática	Diagrama de clases	Clase, asociación, generalización, dependencia, realización, interfaz
	Vista de casos de uso	Diagramas de casos de uso	Caso de uso, actor, asociación, extensión, inclusión, generalización de casos de uso
	Vista de implementación	Diagrama de componentes	Componente, interfaz, dependencia, realización
	Vista de despliegue	Diagrama de despliegue	Nodo, componente, dependencia, localización
Dinámica	Vista de máquinas de estados	Diagrama de estados	Estado, evento, transición, acción
	Vista de actividad	Diagrama de actividad	Estado, actividad, transición de terminación, división, unión
	Vista de interacción	Diagrama de secuencia	Interacción, objeto, mensaje, activación
		Diagrama de colaboración	Colaboración, interacción, rol de colaboración, mensaje
Gestión del modelo	Vista de gestión del modelo	Diagrama de clases	Paquete, subsistema, modelo

Tabla 2 - Vistas y diagramas de UML, basado en [RJB00: 22]

- POSA propuesto por Frank Buschmann, Regine Meunier, Hans Rohnert, Peter Sommerlad y Michael Stal., en 1996, el mismo se basa en la práctica de los patrones, sus vistas las agrupan en listas; la primera lista son: (1) Arquitectura conceptual: componentes, conectores; (2) Arquitectura de módulos: subsistemas, módulos, exportaciones, importaciones; (3) Arquitectura de código: archivos, directorios, bibliotecas, inclusiones; (4) Arquitectura de ejecución: tareas, hilos, procesos. La segunda lista de vistas, incluye: (1) Vista lógica: el modelo de objetos del diseño, o un modelo correspondiente tal como un diagrama de relación; (2) Vista de proceso: aspectos de concurrencia y sincronización; (3) Vista física: el mapeo del software en el hardware y sus aspectos distribuidos; (4) Vista de desarrollo: la organización estática del software en su entorno de desarrollo.
- La estrategia de arquitectura de Microsoft define, en consonancia con las conceptualizaciones más generalizadas, cuatro vistas, ocasionalmente llamadas también arquitecturas: Negocios, Aplicación, Información y Tecnología [Platt02]. La vista que aquí interesa es la de la aplicación, que incluye, entre otras cosas: (1) Descripciones de servicios automatizados que dan soporte a los procesos de negocios; (2) descripciones de las interacciones e interdependencias (interfaces) de los sistemas aplicativos de la organización, y (3) planes para el desarrollo de nuevas aplicaciones y la revisión de las antiguas, basados en los objetivos de la empresa y la evolución de las plataformas tecnológicas. Cada arquitectura, a su vez, se articula en vistas también familiares desde los días de OMT que son (1) la Vista Conceptual, cercana a la semántica de negocios y a la percepción de los usuarios no técnicos; (2) la Vista Lógica, que define los componentes funcionales y su relación en el interior de un sistema, en base a la cual los arquitectos construyen modelos de aplicación que representan la perspectiva lógica de la

arquitectura de una aplicación; (3) la Vista Física, que es la menos abstracta y que ilustra los componentes específicos de una implementación y sus relaciones.

## Actividades del Alumno

Actividad en Clases:

Se establecerán unas preguntas de control para la elaboración de un taller.

Actividad a Realizar.

1. Desarrolle un artículo argumentativo sobre el documento IEEE Std 1471-2000, y contrástelo con lo establecido por la Microsoft y el estándar IEEE 610.12.1990.
2. Se establecerá por grupo el desarrollo de un ADLs: Acme, Aesop, ArTek, Armani, C2 SADL, CHAM, Darwin, Jacal, LILEANNA, MetaH, Rapide, SADL, UML, UniCon, Wright, xADL. Con la finalidad de ampliar como estos se representan.
3. Luego de analizado las Vistas de los marcos de referencia, seleccione un modelo y profundice en el tema a fin de presentar un sinóptico ejemplificado de sus proceso y metodología.
4. Desarrolle las vistas y contrapóngalo con los niveles de abstracción.
5. Desarrolle el tema de Escenarios a partir de las arquitecturas ALMA, SAAM y ATAM.
6. A partir de los siguientes campos de la AS en cual se identifica y por que:
  - a. Lenguajes de descripción de arquitecturas.
  - b. Fundamentos formales de la AS.
  - c. Técnicas de análisis arquitectónicas.
  - d. Métodos de desarrollo basados en arquitectura.
  - e. Recuperación y reutilización de arquitectura.
  - f. Codificación y guía arquitectónica
  - g. Herramientas y ambientes de diseño arquitectónico
  - h. Estudios de casos.
  - i. Diseño o selección de la arquitectura.
  - j. Representación de la arquitectura.
  - k. Evaluación y análisis de la arquitectura.
  - l. Desarrollo y evolución basados en arquitectura.
  - m. □ Recuperación de la arquitectura.

Éxitos en su nuevo aprendizaje....