



**UT
SELVA**

Manual de Asignatura

MC. Armando Méndez Morales

**Ingeniería en
DESARROLLO Y GESTIÓN DE SOFTWARE**

Materia
ARQUITECTURAS DE SOFTWARE



Desarrollo del curso

Programa cuatrimestral iniciando el **1 de septiembre** y terminando el **13 de diciembre** (Para quienes PASEN EN TIEMPO NORMAL la materia) o hasta el **17 de diciembre** (Para quienes requieran tiempo extra)

Septiembre						
D	L	M	M	J	V	S
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30		

Octubre						
D	L	M	M	J	V	S
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
31						

S I M B O L O G Í A									
▲	INICIO DE CURSO				●	DÍAS NO LABORABLES			
▼	FIN DE CURSO				■	REINSCRIPCIONES Y BOLETAS			
■	CORTE DE INDICADORES				■	EXAMEN DE ADMISIÓN			
■	SEMANA DE ANIVERSARIO				■	RECESO LABORAL			
					■	VACACIONES			
					■	GLOBALES			
					■	NIVELACIONES			

Noviembre						
D	L	M	M	J	V	S
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30				

Diciembre						
D	L	M	M	J	V	S
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31



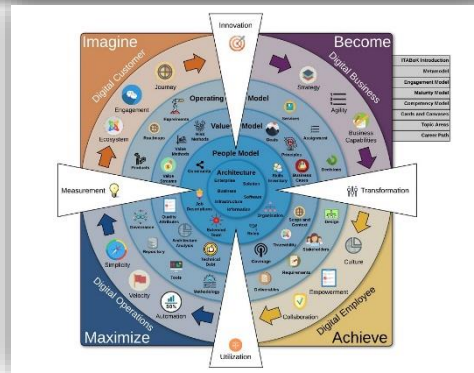


Competencias y Objetivo del curso

1. Competencias	Construir soluciones de software seguro y sistemas inteligentes mediante la dirección y el liderazgo en la gestión de proyectos , integrando metodologías y arquitecturas de desarrollo para la optimización de proyectos de investigación, innovación, desarrollo tecnológico y emprendimiento, bajo la normatividad aplicable.
2. Cuatrimestre	Séptimo
3. Horas Teóricas	21
4. Horas Prácticas	54
5. Horas Totales	75
6. Horas Totales por Semana Cuatrimestre	5
7. Objetivo de aprendizaje	El alumno establecerá arquitecturas de software con base en el análisis de requerimientos para satisfacer los atributos de calidad del software y servir como guía en el desarrollo

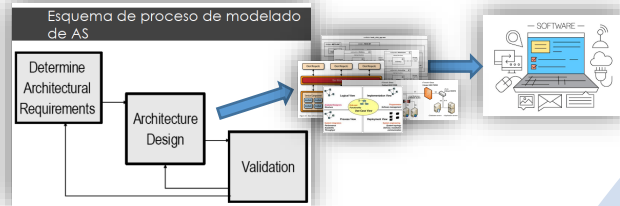


INDICE - Unidades temáticas

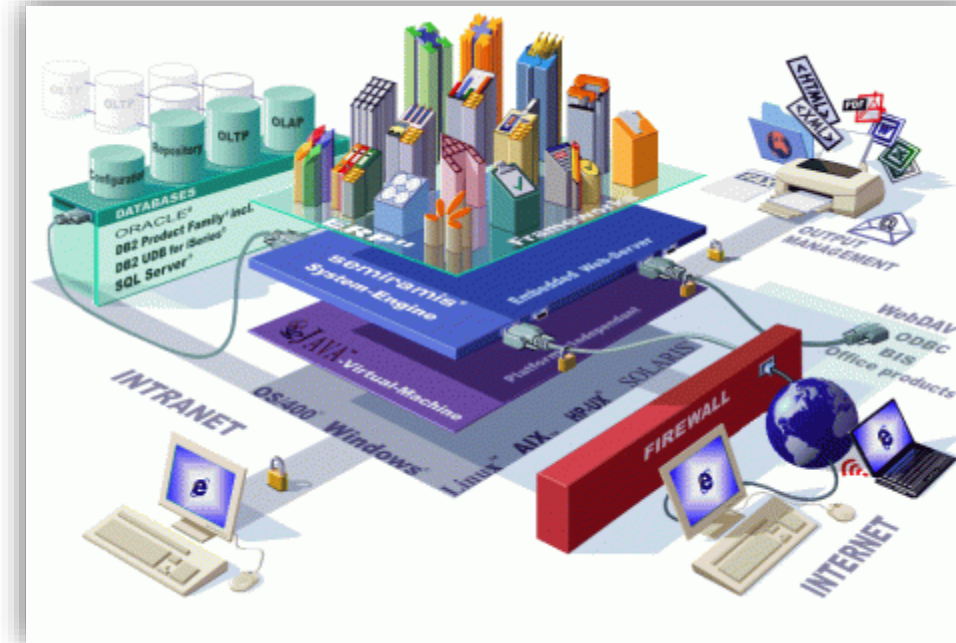
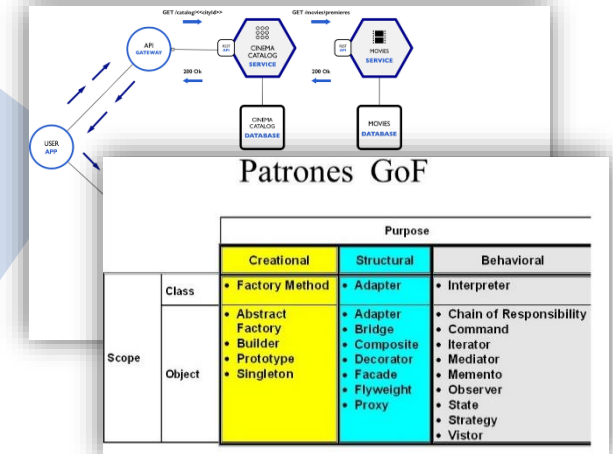


2. Modelado de arquitecturas de software (5 hrs. teóricas, 25 hrs. prácticas)

1. Fundamentos de arquitecturas de software (10 hrs. teóricas, 5 hrs. prácticas)



3. Patrones de diseño (6 hrs. teóricas, 24 hrs. prácticas)



Fuentes de las imágenes:

<https://www.danielprimo.io/files/2018-01/microservicios-aplicacion.png>

<https://itabok.iasaglobal.org/wp-content/uploads/2020/01/itabok-image.jpeg>

<https://slideplayer.es/slide/3847891/12/images/8/Patrones+GoF.jpg>

<https://tomasinsoftarchitecture.wordpress.com/2013/11/17/2/>



Evaluación del aprendizaje

- + Evaluación diagnóstica
- + Evaluaciones por unidades temáticas
 - + **(40%) Saber.** Examen Teórico, Participaciones, Exposiciones, Mapas conceptuales y mentales.
 - + **(50%) Saber Hacer.** Resultado de aprendizaje, **Trabajo en equipo (Auto y co-evaluación/evaluación pares)**, Prácticas, Ejercicios, Res. Problemas.
 - + **(10%) Saber Ser.** Puntualidad, Respeto, Ética, Asesoramiento, Apoyo.
- + Evaluación sumativa.

Beneficios como Arquitecto de software

🇲🇽 ¿Cuanto es el sueldo como Arquitecto de software?

glassdoor.com.mx/Sueldos/arquitecto-de-software-sueldo-SRCH_KO0,22.htm

pdf to office word

Empleos Evaluaciones de la empresa **Sueldos** Entrevistas

Q arquitecto de software

Sueldos

Ubicación



> **Sueldos para Arquitecto De Software** México

Sueldos

Entrevistas

Sueldos para Arquitecto De Software

12 Sueldos Actualizado el 1 sep 2020

Sueldo base promedio

MXN\$ 60 mil / mes

No hay suficientes informes como para mostrar la distribución de sueldos



¿Cuánto gana un Arquitecto De Software?

El sueldo nacional promedio de un Arquitecto De Software es de \$59,671 en México. Filtra por ubicación para ver los sueldos de Arquitecto De Software en tu área. Las estimaciones de los sueldos se basan en los 12 sueldos que los empleados con un cargo de Arquitecto De Software informaron a Glassdoor de manera anónima.

mx.talent.com/salary?job=De+Arquitecto+Software

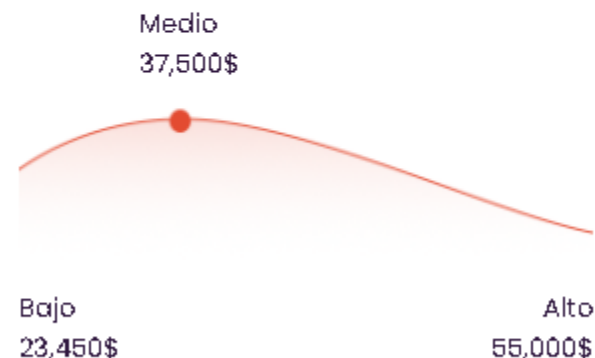
pdf to office word

De Arquitecto Software: Salario

37,500\$ / Mes

Basado en 132 salarios

El salario **De Arquitecto Software** promedio en **México** es de **450,000\$** al año o **231\$** por hora. Los cargos de nivel inicial comienzan con un ingreso de **281,400\$** al año, mientras que profesionales más experimentados perciben hasta **660,000\$** al año.



Beneficios como Arquitecto

🚦 ¿Qué requisitos solicitan para Arq. de software?

🔒 <https://www.indeed.com.mx/ver-empleo?cmp=AsTecI&t=Arquitecto+soluciones+ti&jk=b41977k>

Arquitecto de soluciones TI

AsTecI - Toluca, Méx.

\$25,000 al mes - Tiempo completo, Indefinido

Postularse

AsTecI, con 30 años de trayectoria en el mercado de las Tecnologías de la Información es un empresa 100% mexicana y te invita a formar parte de nuestro equipo de trabajo como:

Arquitecto de soluciones TI

Deberás cumplir los siguientes requisitos:

- Ingeniería en Informática, Sistemas Computación o afín.
- Experiencia mínima de 3 años en el Sector Bancario.
- Inglés Avanzado.
- TOGAF, Azure/AWS Solution Architect.

Conocimientos:

- Analisis y Diseño
- ITIL
- Proyectos de Infraestructura
- Seguridad de software
- Implementación de software
- Diseño de sistemas

Experiencia:

- Sector bancario: 3 años (Requisito deseable)

Idioma:

- Inglés avanzado (Requisito deseable)

Trabajar desde casa:

- Temporalmente debido al COVID-19

🔒 <https://www.indeed.com.mx/ver-empleo?cmp=Ancient-Tecnología-S.A.-de-C.V.8>

Sr. de Software

Ancient Tecnología S.A. de C.V. - Cuernavaca, Mor.

\$45,000 - \$55,000 al mes - Tiempo completo, Por contrato

Postularse

El especialista que cubra este rol debe de tener excelentes skills de comunicación y coordinación de equipos, así como dominio del idioma inglés.

Sr. de Software.

- Experiencia en Desarrollo de plataformas BaaS
- Experiencia en Open Banking
- Experiencia en liderazgo de células de desarrollo

Experiencia laboral:

- Al menos 5 Años Liderando Equipos de Desarrollo.
- Conocimiento de estándares basados en el NIST, enfocados en temas Biométricos.
- Experiencia en compañías Bancarias o FinTech
- Al menos 5 años como Arquitecto de Soluciones Tecnológicas.
- 5 Años de Experiencia en Ejecución de Programas Digitales con Filosofías Ágiles.

Conocimientos requeridos:

- Conocimiento de Bases de Datos NoSQL (Mongo/REDIS) y SQL.
- Conocimiento de Lenguaje de Programación Java, JWS, Spring.
- Arquitectura y Desarrollo de Microservicios.
- Conocimiento y Manejo de Linux y Shell Script.
- Conocimiento de Esquemas de Seguridad Informática, PCI Compliance.
- OAUTH 2.0
- Kubernetes,
- Jenkins y Sonar
- Conocimiento de Lenguaje de Programación Java.

- Java 8
- JWS
- POO
- Spring Core
- Spring boot

Reglas de operación del curso



- ✚ Tratar con respeto a todos los participantes del curso en los diferentes medios de comunicación.

- ✚ Diviértete **SIN OFENDER A NADIE... =D**

- ✚ Sé **buen** hijo(a), vecino(a), novio(a) **estudiante !!!**

- ✚ SÉ HONESTO evitando plagios de trabajos (**aprende ahora y no cuando estés buscando trabajo**)

- ✚ Contribuye en la elaboración y revisión de los trabajos en equipo.

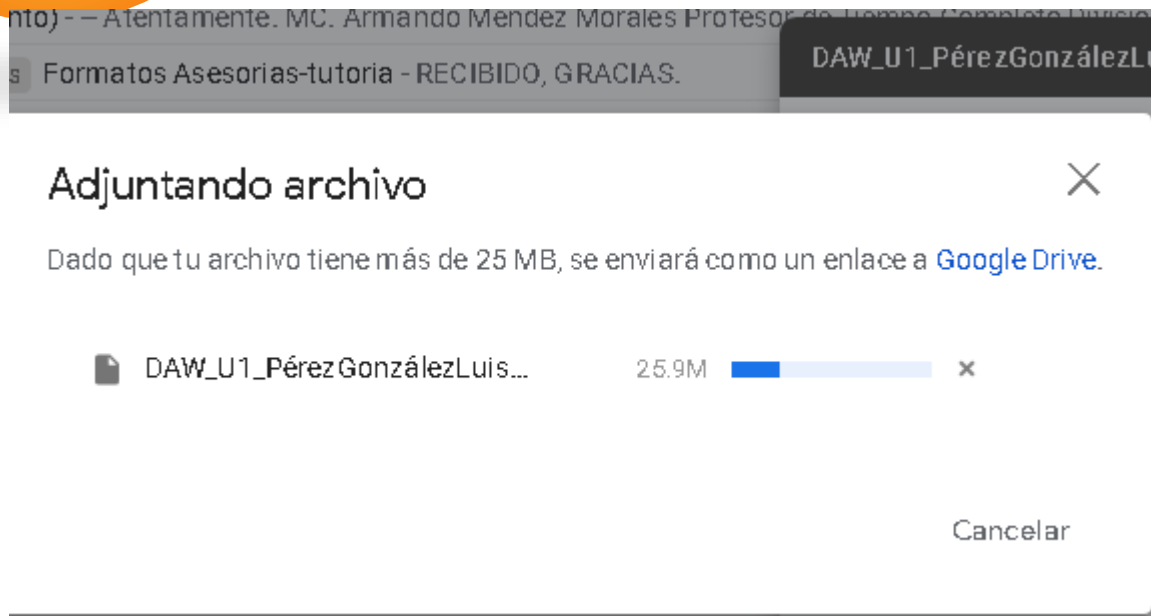
- ✚ **Obtén y Lee las rúbricas** en cada unidad.

- ✚ **Pregunta al profe, TODAS LAS DUDAS DE LAS RÚBRICAS Y DE LOS MATERIALES** de la materia.

- ✚ **Por favor... Hagan su tarea lo antes posible y no en el último día en el último minuto... Evita nivelaciones y GLOBALES... =(**



Importante: Compartir archivos de tamaño grande!!!



.... al finalizar te preguntará con quien quieres compartir el archivo, por lo que deberás **cambiar a la selección marcada en amarillo** en la imagen de la derecha **Activar la opción de compartir mediante enlace**, y finalizar con el botón **ENVIAR**.

Cuando envíes tus actividades por correo, con archivos de tamaño superior (al límite del servidor de correo electrónico), te enviará un mensaje similar al presentado, indicando que se enviará como un enlace externo a un servidor de almacenamiento (lo cual está bien), hay que esperar a que se termine la transferencia....





Contrato de aprendizaje

Como Instructor me comprometo a...

- ✓ Fomentar la participación e investigación.
- ✓ Aclarar todas las dudas.
- ✓ Tratarlos con respeto y amabilidad.
- ✓ Cumplir con lo programado en cada tema.
- ✓ Responder lo antes posible a las dudas de forma escrita o a regresarles una llamada telefónica si me tardo en ello.

Como Participante me comprometo a...

En tu libreta anota tu nombre completo, grado, grupo, materia y tus compromisos personales para/con esta materia y este cuatrimestre!!!

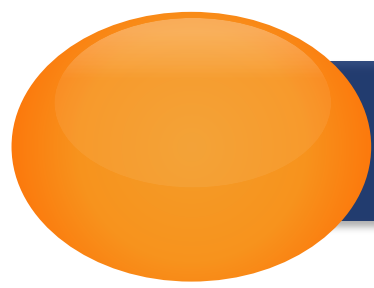


Evaluación diagnóstica

Contesta las preguntas que puedas, sin consultar en libros o internet, sino empleando tus conocimientos actuales a la fecha, y registra tus respuestas en tu libreta.



- + Que tecnologías empleaste en tu ultimo **desarrollo de software**?
- + Cómo explicas la **estructura de un software** a un compañero?
Representa un ejemplo visual de ello.
- + Cómo se puede ejemplificar un **Ambiente de desarrollo** cuando desarrollas software para una empresa?
- + Has desarrollado software con algún **framework**? Que características resaltarías de estas experiencias?
- + Menciona algún **patrón de diseño** que hayas empleado en algún software en el que hayas participado.



¡Manos a la
OBRA!





**UT
SELVA**

1. Fundamentos de arquitecturas de software

MC. Armando Méndez Morales

**Ingeniería en
DESARROLLO Y GESTIÓN DE SOFTWARE**

Materia

ARQUITECTURAS DE SOFTWARE



Objetivos

- El alumno **determinará la arquitectura de software** de acuerdo a requerimientos para guiar la construcción de los componentes de software.

TEMAS

- + Introducción a las arquitecturas de software
- + Frameworks de arquitecturas
- + Estilos de arquitectura de software





Evaluación del aprendizaje

+ Evaluación de la Unidad

+ **(40%) Saber.** Act 01 Conceptos

+ **(50%) Saber Hacer.** Act 02 Frameworks, Act 03 RA requisitos y estilos.

+ **(10%) Saber Ser.** Puntualidad en la entrega de trabajos, Respeto, Ética, Asesoramiento a compañeros.

Para la primera unidad, se considerará las actividades del contrato de aprendizaje y de su evaluación diagnóstica como parte de la calificación del SER. Enviarlos dentro de las próximas dos semanas.

Introducción a las arquitecturas de software

Definition: software architecture

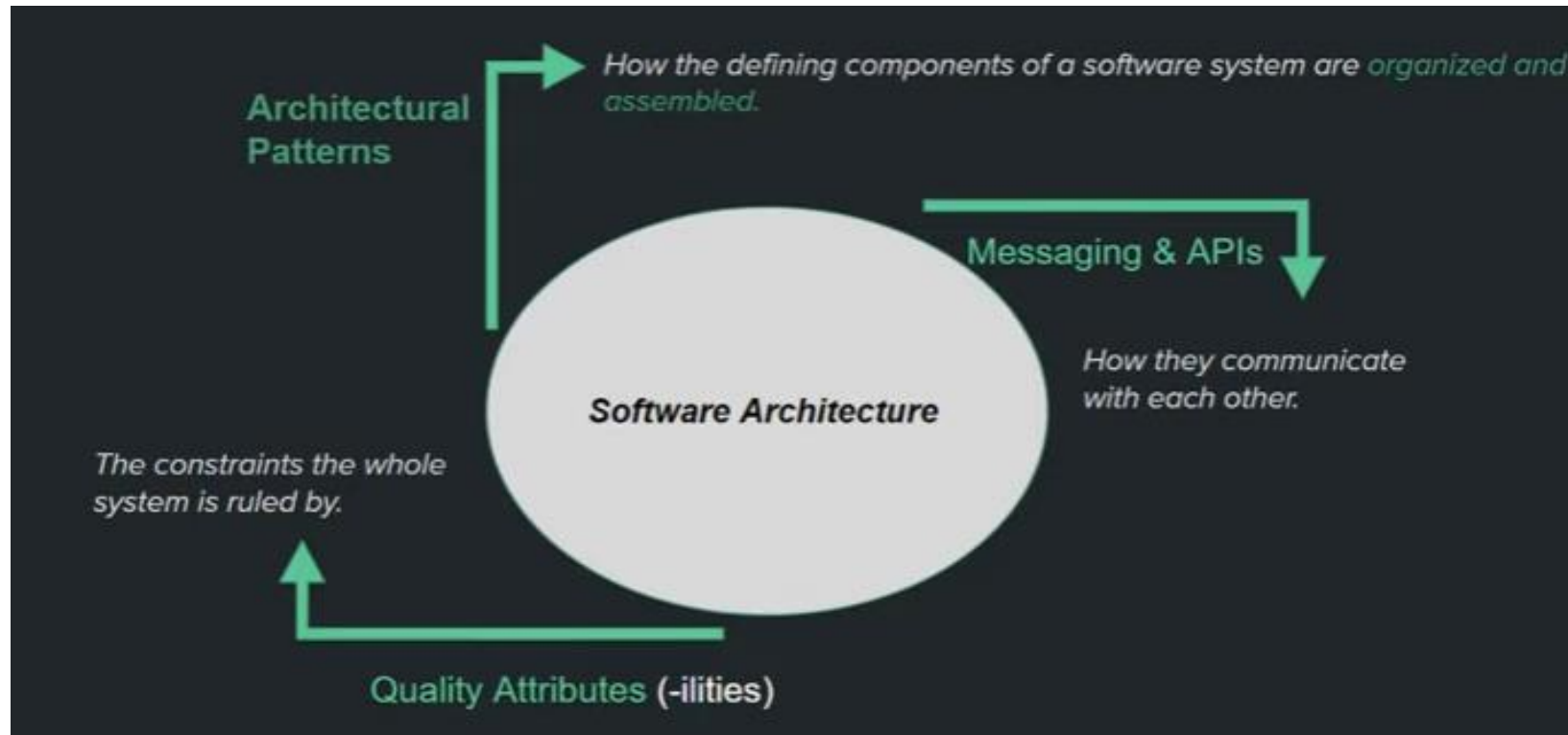
Architecture is the highest level concept of the expert developers.

“In most successful software projects, the expert developers working on that project have a shared understanding of the system design. This shared understanding is called **‘architecture.’** This understanding **includes** how the system is divided into **components and how** the components **interact through interfaces**. These components are usually composed of smaller components, but the architecture only includes the components and interfaces **that are understood by all the developers.**”

<http://martinfowler.com/ieeeSoftware/whoNeedsArchitect.pdf>



Introducción a las arquitecturas de software



Introducción a las arquitecturas de software

*The architectural pattern define the **granularity** of the components.*

Granularity = how small or big the component should be.

Architectural Pattern	Design Pattern
<ul style="list-style-type: none">• High-level, universal scope.• How components are organized and assembled	<ul style="list-style-type: none">• Lower-level scope• How components are built.



Introducción a las arquitecturas de software

Características de las arquitecturas de software

- **Representación de alto nivel** de la **estructura del sistema** describiendo las partes que lo integran.
- Puede incluir los **patrones** que supervisan la **composición de sus componentes** y las restricciones al aplicar los patrones
- Trata aspectos del diseño y desarrollo que no pueden tratarse adecuadamente dentro de los módulos que forman el sistema.
- Toda aplicación tiene una arquitectura, aunque no sea explícita.
- **Incluye modelos, lenguajes y herramientas** para la descripción y desarrollo práctico de arquitecturas software.

Consulta Fuente: (Moreno Navarro, 2020)



Introducción a las arquitecturas de software

Características de las arquitecturas de software

- ¿De qué se ocupa?
 - Diseño preliminar o de alto nivel.
 - **Organización a alto nivel del sistema**, incluyendo aspectos como la descripción y análisis de **propiedades relativas a su estructura** y control global, los **protocolos de comunicación** y sincronización utilizados, la **distribución física del sistema y sus componentes**, etc.
 - Otros **aspectos relacionados con el desarrollo** del sistema y su evolución y adaptación al cambio: **composición**, **reconfiguración**, **reutilización**, **escalabilidad**, **mantenibilidad**, etc.



Introducción a las arquitecturas de software

Características de las arquitecturas de software

- ¿De qué **NO** se ocupa?
 - Diseño detallado
 - Diseño de algoritmos
 - Diseño de estructuras de datos.



Introducción a las arquitecturas de software

Características de las arquitecturas de software

*Una **arquitectura de software** bien configurada facilita a los usuarios y desarrolladores **la comprensión del sistema**. **Factores importantes** que influyen la aptitud de la arquitectura de software son la **planificación de proyectos**, el análisis de riesgo, la **organización**, el proceso de **desarrollo**, los ciclos de trabajo, el **hardware**, la garantía de **calidad** y los **requerimientos**.*

(voightmann, 2020)



Actividades de Evaluación

Realice las actividades descritas en:

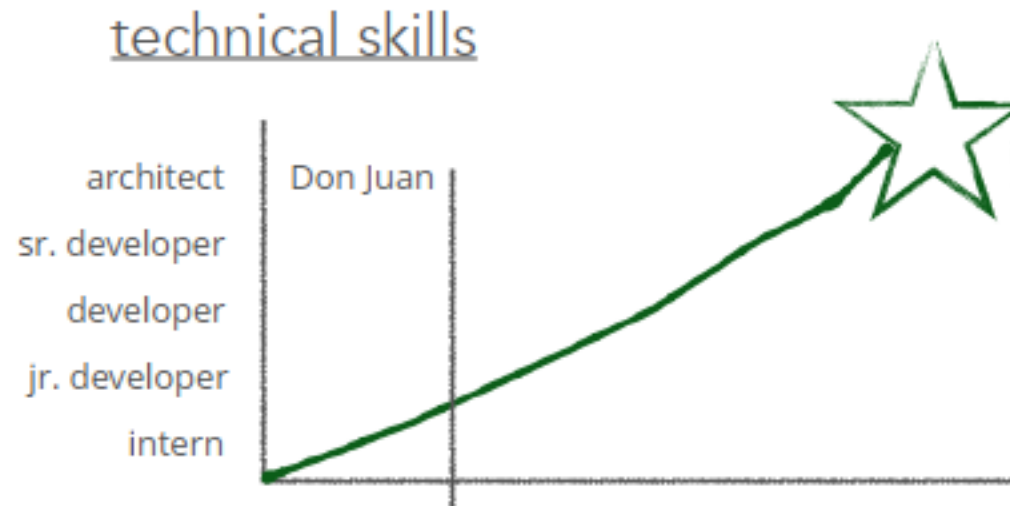
 ***ARCS - U1 Rúbrica Act 01 Conceptos.pdf***



Introducción a las arquitecturas de software

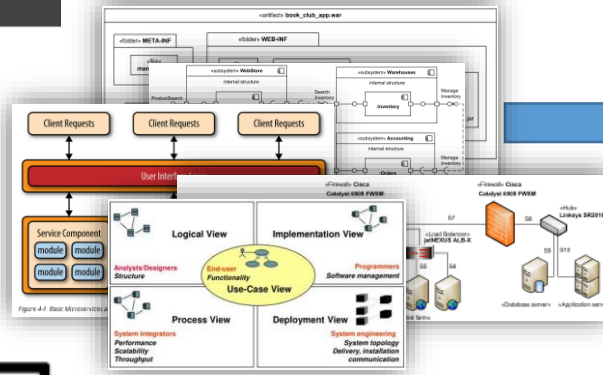
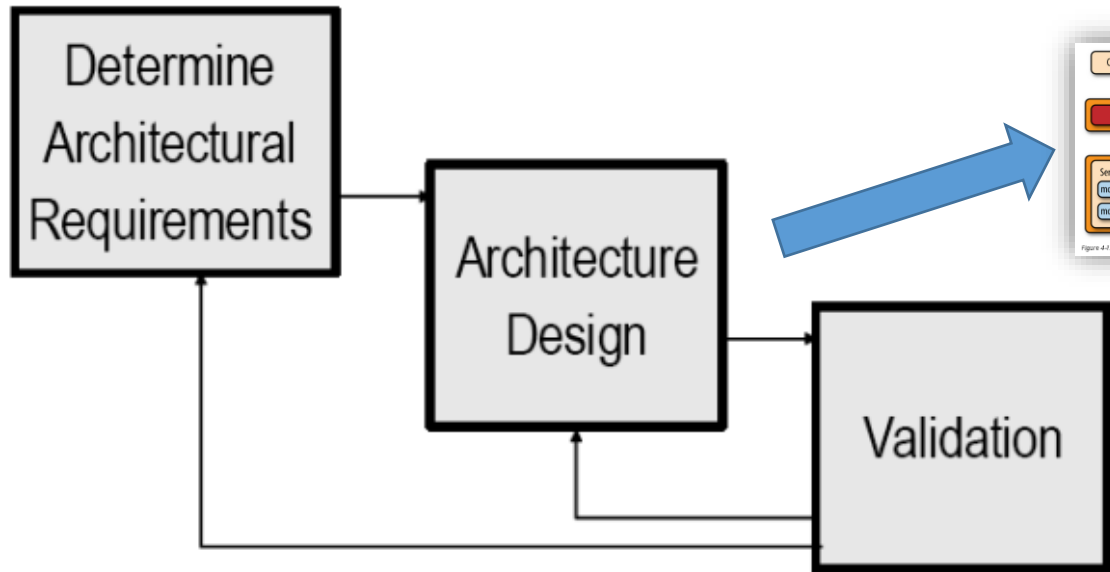
Ventajas de diseñar y documentar explícitamente una arquitectura de software:

- Comunicación entre stakeholders
- Decisiones tempranas de diseño
- Reuso a gran escala



Introducción a las arquitecturas de software

Esquema de proceso de modelado de AS



Para ***Diseñar la arquitectura del software***, se inicia con la ***determinación de los requisitos arquitecturales***, y posteriormente se realiza una validación de esta arquitectura, con la finalidad de verificar que se pueda cumplir con los criterios de calidad del software a desarrollar y/o implementar.

Fuentes:

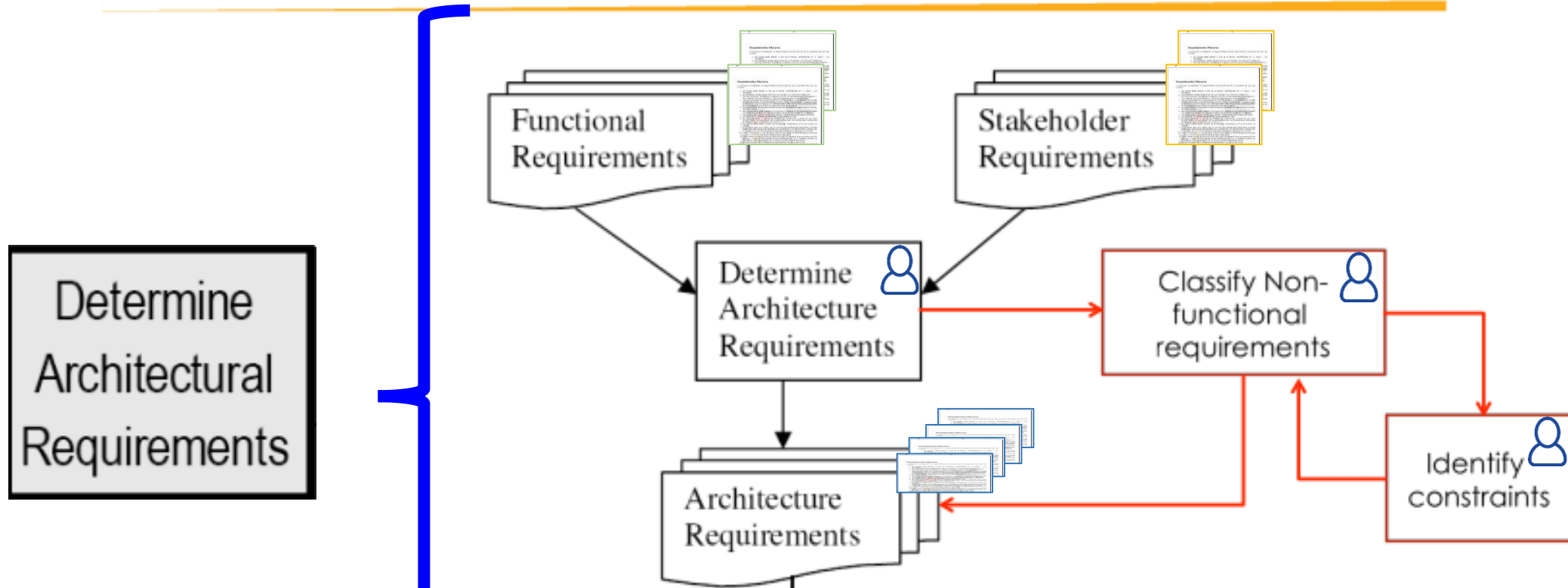
http://cic.iaverianacali.edu.co/wiki/lib/exe/fetch.php?media=materias:s2_conceptosde modelado.pdf
<https://www.oreilly.com/content/software-architecture-patterns/>
<https://www.uml-diagrams.org/index-examples.html>

Otro ejemplo según RUP

<https://www.slideserve.com/elvin/fundamentos-de-definici-n-de-arquitectura-de-software>



Introducción a las arquitecturas de software



Para **determinar los requisitos arquitecturales**, se deben obtener inicialmente los requisitos funcionales y de los interesados, posteriormente se debe clasificar a los requisitos NO Funcionales e identificar las restricciones.

Introducción a las arquitecturas de software

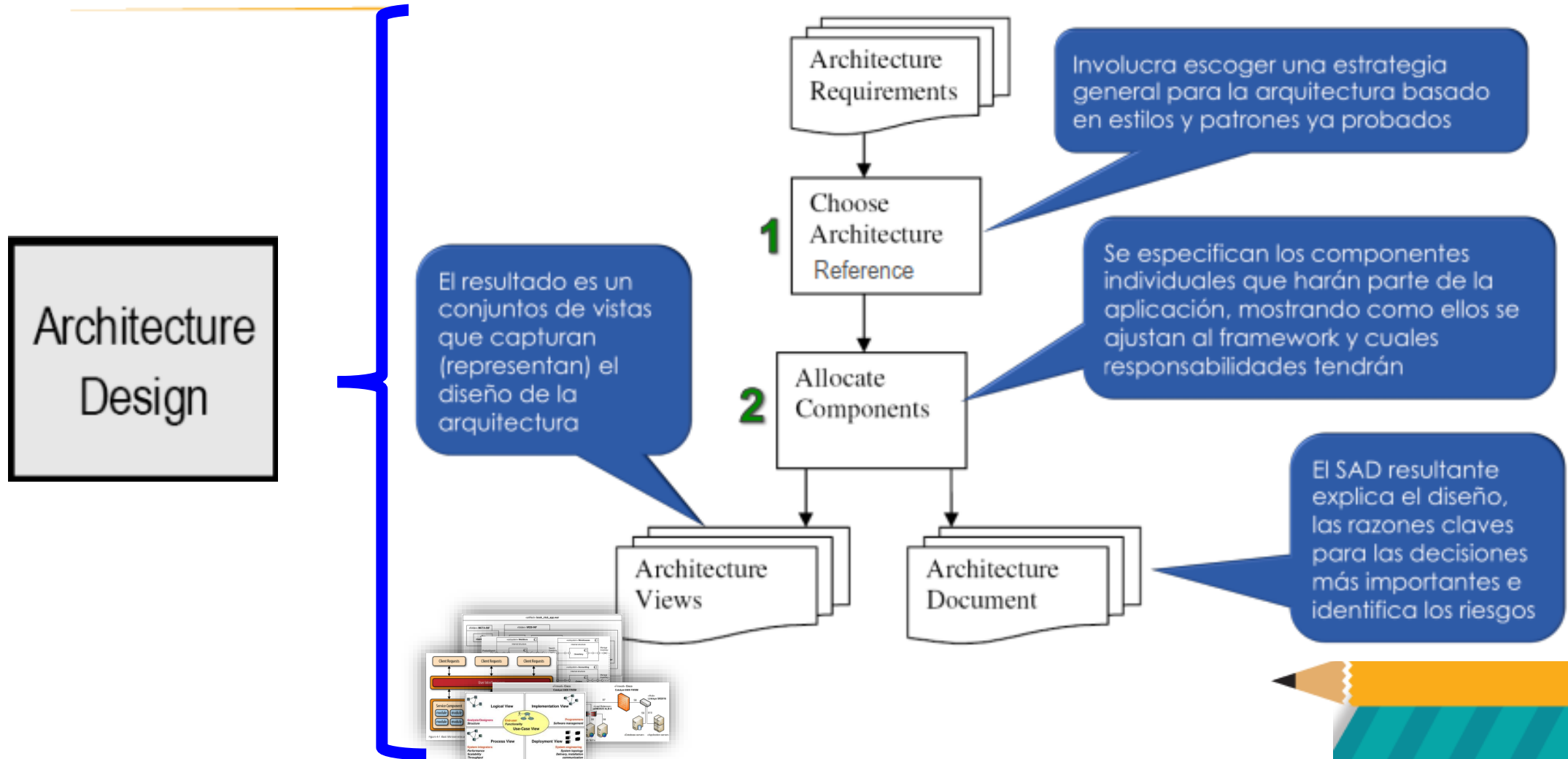
La arquitectura de software afecta la

- ← Performance
- ← Seguridad (security y safety)
- ← Disponibilidad
- ← Mantenibilidad

• Entonces, el estilo y estructura particular elegido para una aplicación dependen fuertemente de **los requerimientos no funcionales**



Introducción a las arquitecturas de software



Introducción a las arquitecturas de software

Validation

Durante el proceso de creación de la arquitectura, el objetivo de la fase de validación consiste en aumentar la confianza del equipo de diseño con respecto a que la arquitectura es adecuada para cumplir con los requerimientos del sistema. Se puede escoger entre dos técnicas: ***Pruebas manuales o Prototipos***.

1. Prueba manual: Involucra la prueba de la AS usando escenarios
2. Prototipo: Involucra la construcción de un prototipo que crea un arquetipo de la aplicación deseada, de esta forma su capacidad para satisfacer las necesidades se pueden evaluar con más detalle a través de prototipos.

Evaluación de Arquitecturas

1. Software Architecture Analysis Method (SAAM)
2. Architecture Tradeoff Analysis Method (ATAM)



Frameworks de arquitecturas

*“Un **framework** busca **establecer y organizar la información arquitectónica** para buscar una **evolución sobre los procesos internos** de la **organización**. En la actualidad existen diferentes frameworks, dependiendo de las necesidades de la empresa se puede seleccionar el más apto para ser implementado..”*

(Sarasty España, 2015)

*“**Enterprise Architecture** is an approach to **aligning business and technology resources to achieve strategic outcomes**. It creates a **Roadmap** for how to **deliver strategic goals in the best, fastest and cheapest way**. By creating this roadmap, it **bridges the gap between strategy and effective delivery**.”*

(Terra Firma, 2020)

Para mas detalles revisar:

(Sarasty España, 2015): http://sedici.unlp.edu.ar/bitstream/handle/10915/52183/Documento_completo..pdf?sequence=3&isAllowed=y

(Terra Firma, 2020): <https://www.terrafirma.com.au/our-thinking/top-10-enterprise-architecture-frameworks/>



Frameworks de arquitecturas

1. **TOGAF's** ADM
2. **Zachman**
3. **ITABOK**
4. **Gartner's** Enterprise Architecture Method
5. **Federal** Enterprise Architecture (**FEAF**)
6. **Dept of Defence** Architecture Framework (**DoDAF**)
7. **Australian Government AGA**
8. **SABSA** – Enterprise Security Architecture
9. **Business** Architecture Body of Knowledge (**BizBoK**)
10. **NIST** Enterprise Architecture Model
11. **ISO** Standard for Enterprise Modelling (**ISO19439**)
12. **Integrated** Architecture Framework (**IAF**)
13. Framework **ATOM**

Evaluación de Arquitecturas

1. **Software** Architecture Analysis Method (**SAAM**)
2. **Architecture** Tradeoff Analysis Method (**ATAM**)

Mas referencias: <https://www.avolutionsoftware.com/abacus/frameworks/>
https://www.youtube.com/watch?v=_mcTTGi6dTA&list=PLgn00WwJcEmo-m6nKFz2gh_r5l3jyqtGI&index=7



Australian Government
Architecture (AGA)



PESTLE Framework



IT4IT Reference Architecture



Zachman Framework



Actividades de Evaluación

Realice las actividades descritas en:

 ***ARCS - U1 Rúbrica Act 02 Frameworks.pdf***



Estilos de arquitectura de software/ estilos arquitectónicos de software

- **Clasificación de los sistemas software en grandes familias** cuyos integrantes comparten un patrón estructural común
- **Ejemplos:** Basados en capas, Basados en servicios, Distribuidos, etc.
- **Aspectos:** Patrón de organización general, tipos de componentes presentes habitualmente, interacciones entre ellos.
- **Ejemplos de componentes:** clientes, servidores, filtros, niveles, bases de datos, ...
- **Ejemplos de mecanismos de interacción:** RPC, paso de mensajes, protocolos de comunicación, ...
- El campo de aplicación no es relevante.



Estilos de arquitectura de software

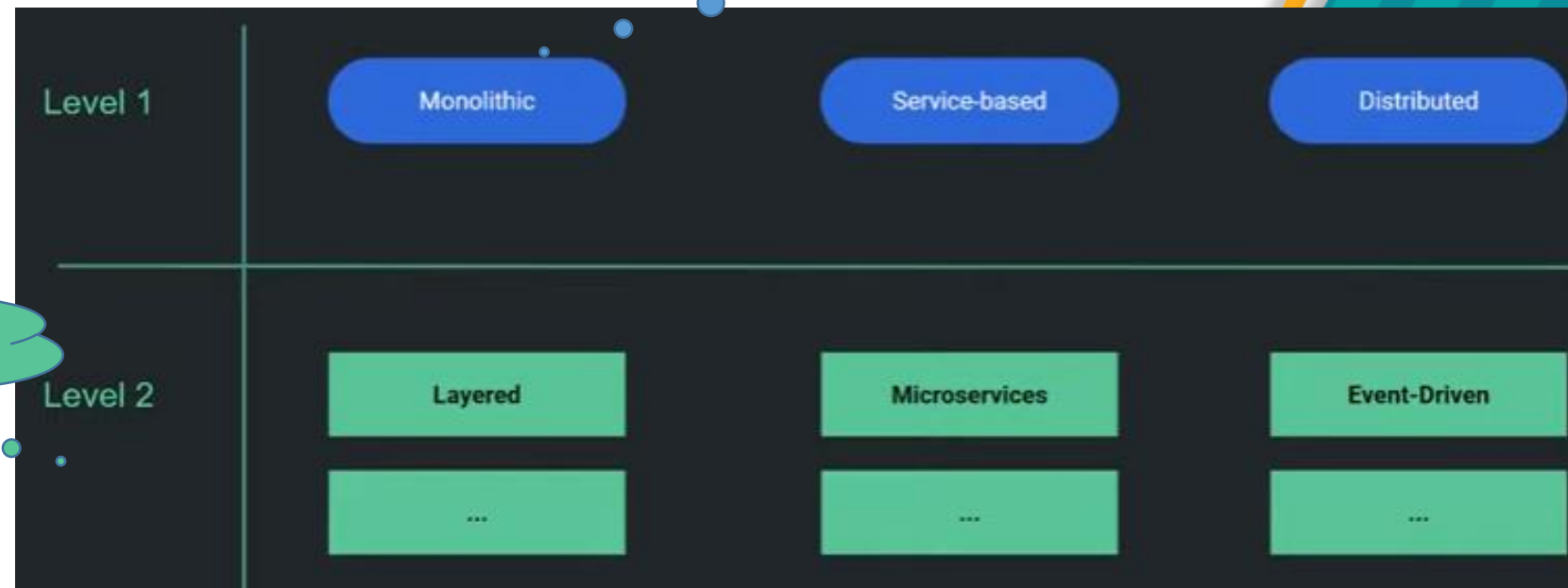
Ejemplos de estilos de arquitecturas de software:

- Cliente-Servidor
- N capas
- Orientada a servicios
- Microservicios
- Dirigida a eventos
- Microkernel
- Basada en espacio
- Serverless

Fuente: <https://www.youtube.com/watch?v=ITkL1oIMiaU>

Macro clasificación

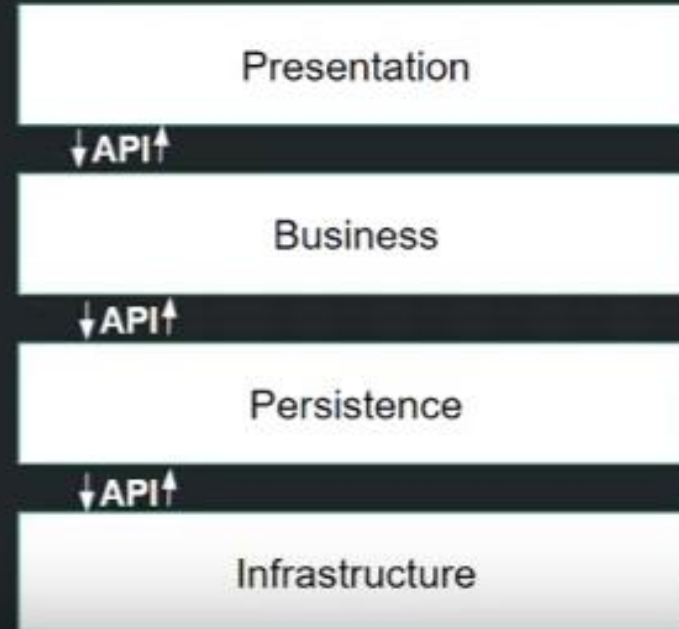
Micro clasificación



Estilos de arquitectura de software

Layered Architectural pattern

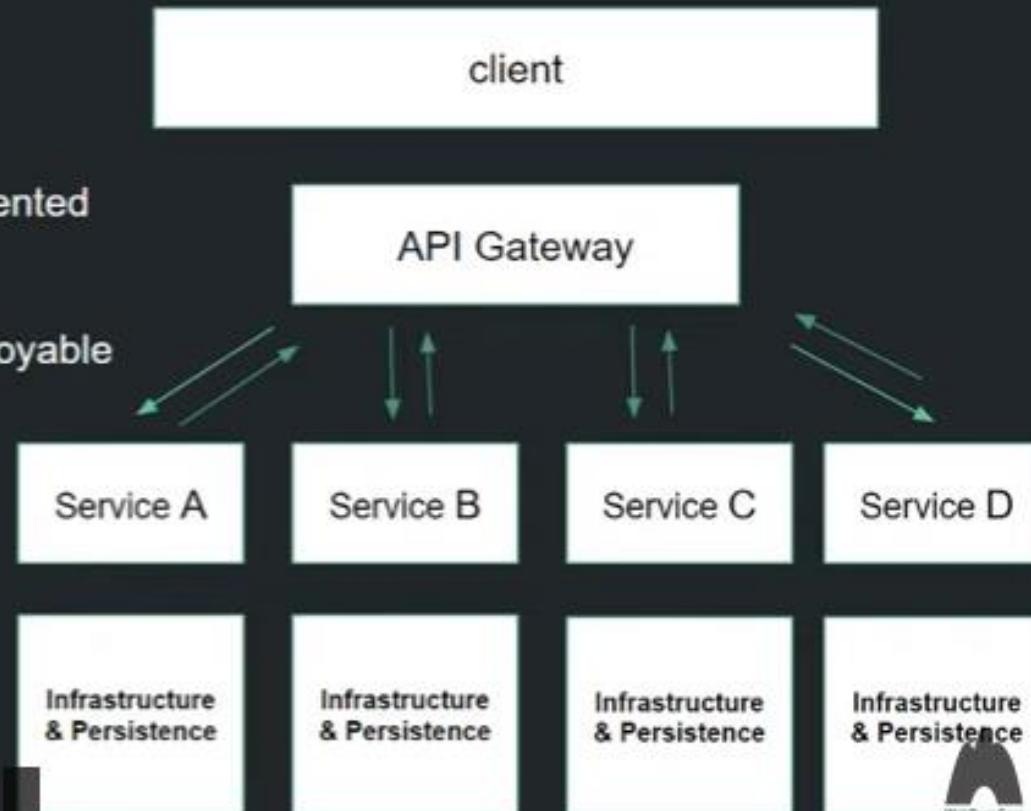
- Also known as the N-tier pattern.
- Monolithic
- Separation of Concerns
- Layers as major components



Estilos de arquitectura de software

Microservices Architectural pattern

- Service-based type of pattern.
- Microservices is NOT Service-Oriented Architecture.
- Independently evolvable and deployable units



Estilos de arquitectura de software

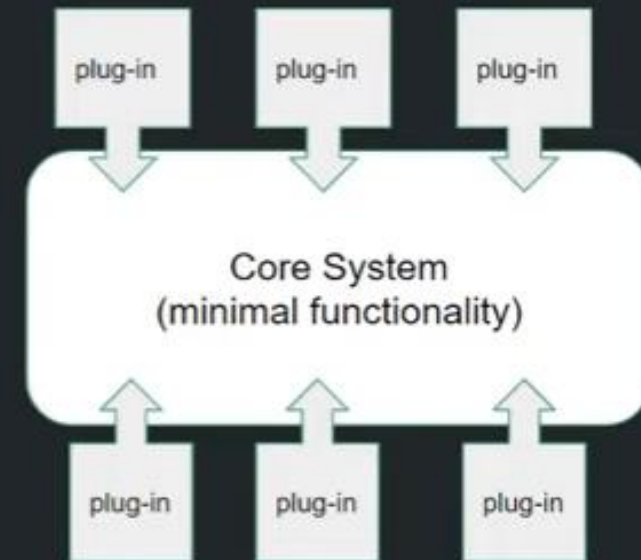
Event-Driven Architectural pattern

- Distributed architecture
- Event Processing units coordinated in a **Mediator**, or **Broker** topology
- Asynchronous nature
- Overall Structure depends on the topology chosen.

1

MicroKernel Architectural pattern

- Also known as the Plug-in architectural pattern
- Major components:
 - Core system (like a kernel)
 - plug-ins
- Monolithic
- Examples: Browser, Text Editors, OSs...



Estilos de arquitectura de software

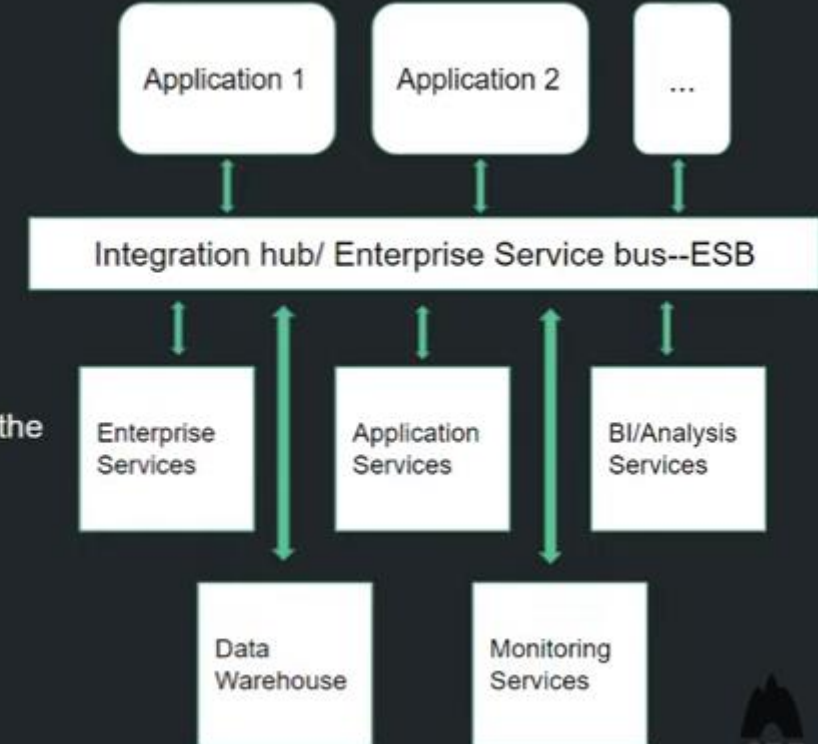
Event-Driven Architectural pattern

- Distributed architecture
- Event Processing units coordinated in a **Mediator**, or **Broker** topology
- Asynchronous nature
- Overall Structure depends on the topology chosen.

2

Service-Oriented Architectural pattern

- Large, enterprise systems.
- Coarse-grained services.
- Integrating different heterogeneous components together.
- There are many implementations of the pattern.



Estilos de arquitectura de software

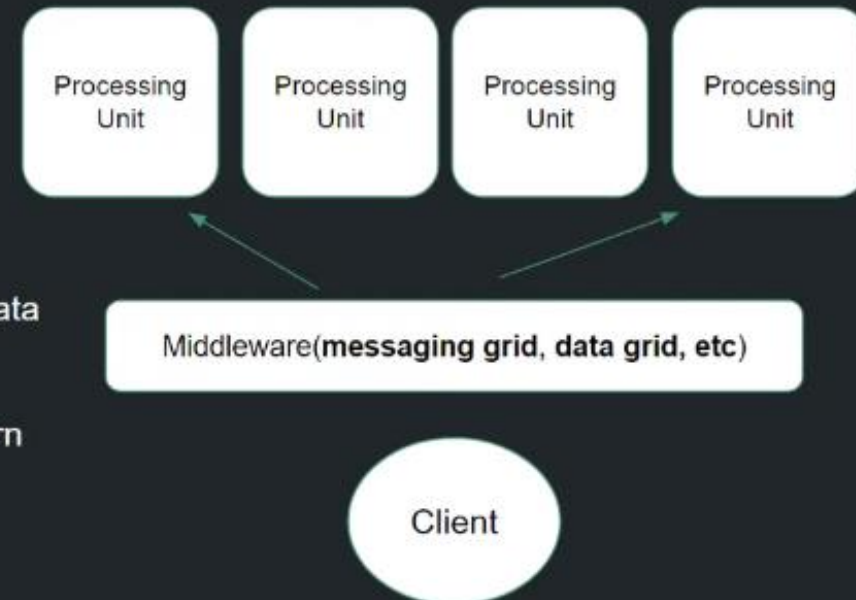
Event-Driven Architectural pattern

- Distributed architecture
- Event Processing units coordinated in a **Mediator**, or **Broker** topology
- Asynchronous nature
- Overall Structure depends on the topology chosen.

3

Space-based Architectural pattern

- Hybrid of Architectural patterns
- Inspired by Space tuples
- Distributed caching - In Memory Data Grid
- Elastic scalable architectural pattern



Actividades de Evaluación

Realice las actividades descritas en:

 ***ARCS - U1 Rúbrica Act 03 RA requisitos y estilos.pdf***



Conclusión

- Dependiendo de la problemática, infraestructura, equipamiento del cliente y condiciones (todas como parte de las restricciones), así como de sus requisitos, se establecerá un diseño arquitectónico que tratará sobre todo de cumplir con los requisitos NO Funcionales, considerando desde **una a varias arquitecturas (que contienen frameworks y estilos arquitectónicos)**, para cumplir con los ***criterios de calidad*** del software a desarrollar o implementar.



Referencias

Autor	Año	Título del Documento	Ciudad	País	Editorial
Mark Richards, Neal Ford	2020 ISBN:97:81-492043-454	Fundamentals of Software Architecture	New York	United States	O'Reilly Media
Robert C. Martin	2018 ISBN:013-449416-4-860	Clean Architecture	Boston	United States	Addison - Wesley - Pearson
Angel Arias, Alicia Durango	2016 ISBN:97:81-52336-5-487	Ingeniería y Arquitectura del Software	Vigo	España	IT Campus Academy
Humberto Cervantes, Rick Kazman	2016 ISBN:97:80-13439-0-789	Designing Software Architectures: A Practical Approach	Boston	United States	Addison - Wesley - Pearson
Laurent Debrauwer	2018 ISBN:97:82-40901-6-349	Patrones de Diseño en Java	Barcelona	España	Eni Ediciones
David Roldan Martinez, Pedro J. Valderas	2018 ISBN:97:88-49964-7-654	Microservicios, un enfoque integrado	Jarama	España	Ra-Ma
Fien Van Der Heyde	2019 ISBN:97:82-40901-8-046	UML 2.5 Domine el diseño con los Patrones de diseño	Barcelona	España	Eni Ediciones
IEEE/ISO/IEC	2011 ISBN(s):9781504413671, 9780738171678, 9780738171425	ISO/IEC/IEEE Std 42010:2011(E), Systems and software engineering -- Architecture description	New York	United States	ISO/IEC/IEEE

Referencias

- *danielprimo*. (23 de enero de 2018). Obtenido de WR 19: **Microframeworks, microservicios y la corrala de vecinos**: <https://www.danielprimo.io/blog/microframeworks-microservicios-y-la-corrala-de-vecino>
- *Moreno Navarro, J. J.* (septiembre de 2020). *http://babel.ls.fi.upm.es*. Obtenido de **Arquitecturas Software**: http://babel.ls.fi.upm.es/~fred/sbc/arquitecturas_sw.pdf
- *Sarasty España, H. F.* (octubre de 2015). **DOCUMENTACIÓN Y ANÁLISIS DE LOS PRINCIPALES FRAMEWORKS DE ARQUITECTURA DE SOFTWARE EN APLICACIONES EMPRESARIALES**. Obtenido de sedici.unlp.edu.ar: http://sedici.unlp.edu.ar/bitstream/handle/10915/52183/Documento_completo..pdf?sequence=3&isAllowed=y
- *Terra Firma*. (02 de septiembre de 2020). **Top 10 Enterprise Architecture Frameworks**. Obtenido de <https://www.terrafirma.com.au/our-thinking/top-10-enterprise-architecture-frameworks/>
- *voightmann*. (2020). *Arquitectura de software*. Obtenido de **Características de una arquitectura de software exitosa**: <https://www.voightmann.de/es/desarrollo-de-software/arquitectura-de-software/>

Otros ejemplos de estilos arquitectónicos: <https://www.slideshare.net/CathyGuevara/arquitectura-software-capitulo-i>

Diseño de arquitectura de software con UML. <https://www.slideserve.com/elvin/fundamentos-de-definici-n-de-arquitectura-de-software>



Muchas gracias

**FORMANDO PROFESIONISTAS
DE VERDAD**

