UNIVERSIDAD DE LIMA

60 años

DOCUMENTACIÓN DE LA ARQUITECTURA

UNIDAD 1: DISEÑO ARQUITECTÓNICO

Temario

- Documentación de la arquitectura:
  - Vistas de la arquitectura
  - Vista funcional
  - Vista, lógica
  - Vista de implementación
  - Vista de despliegue

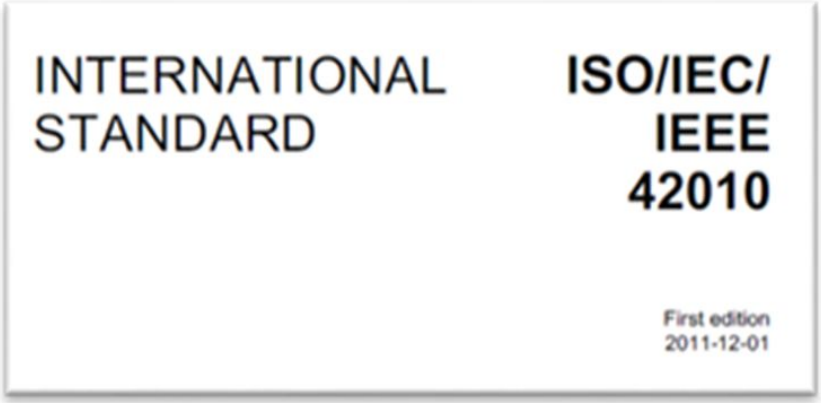
UNIVERSIDAD DE LIMA | 60 años

Documentación de la arquitectura

También denominado Descripción de Arquitectura que corresponde a un trabajo para expresar una arquitectura

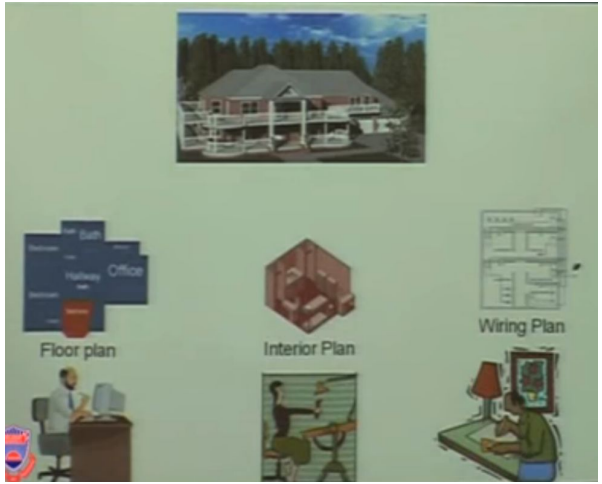
Importancia de un documento de arquitectura

- Comunica aspectos esenciales del sistema.
- Es el punto de partida para planificar el proyecto.
- Es la base para analizar y evaluar alternativas de implementación de la arquitectura.
- Es esencial para realizar las actividades de mantenimiento y soporte técnico.



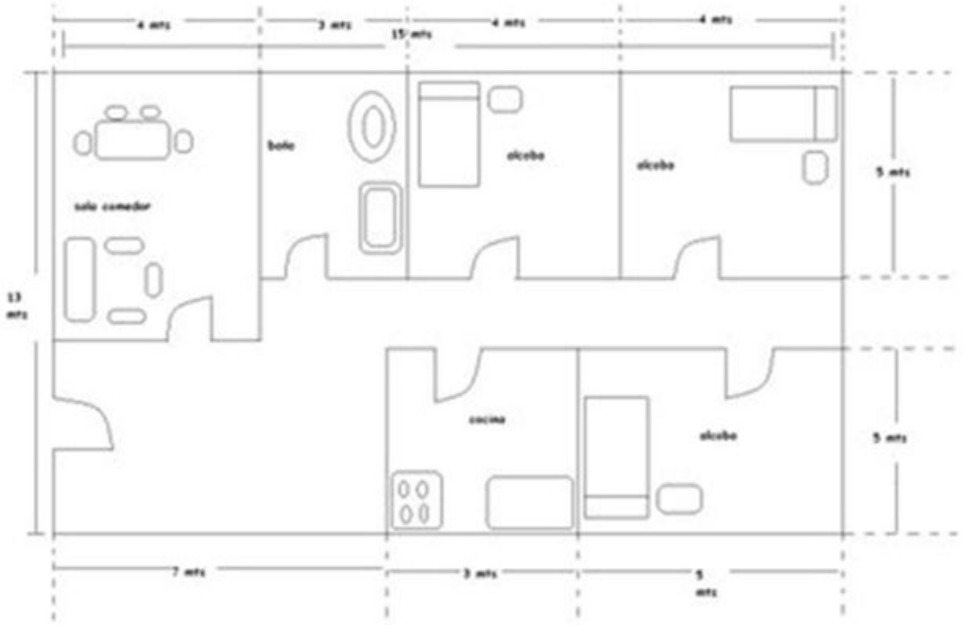
Definiciones en la norma ISO/IEC 42010:2011

- La Descripción de una Arquitectura está conformada por una o varias **Vistas**.
- Una **Vista de Arquitectura** presenta las decisiones diseño (en forma de Modelos) relacionadas con un **Punto de Vista** sobre la Arquitectura.
- Un **Punto de Vista** concentra un grupo específico de preocupaciones de **Stakeholders** sobre el sistema.
    - *Preocupaciones*: rendimiento, seguridad, escalabilidad, usabilidad, facilidad de modificación, escalabilidad, entre otras.



Vistas Arquitectónicas

- Una vista es una perspectiva del sistema.
- Es un subconjunto de los elementos de la arquitectura.
- Describen un comportamiento particular del sistema.



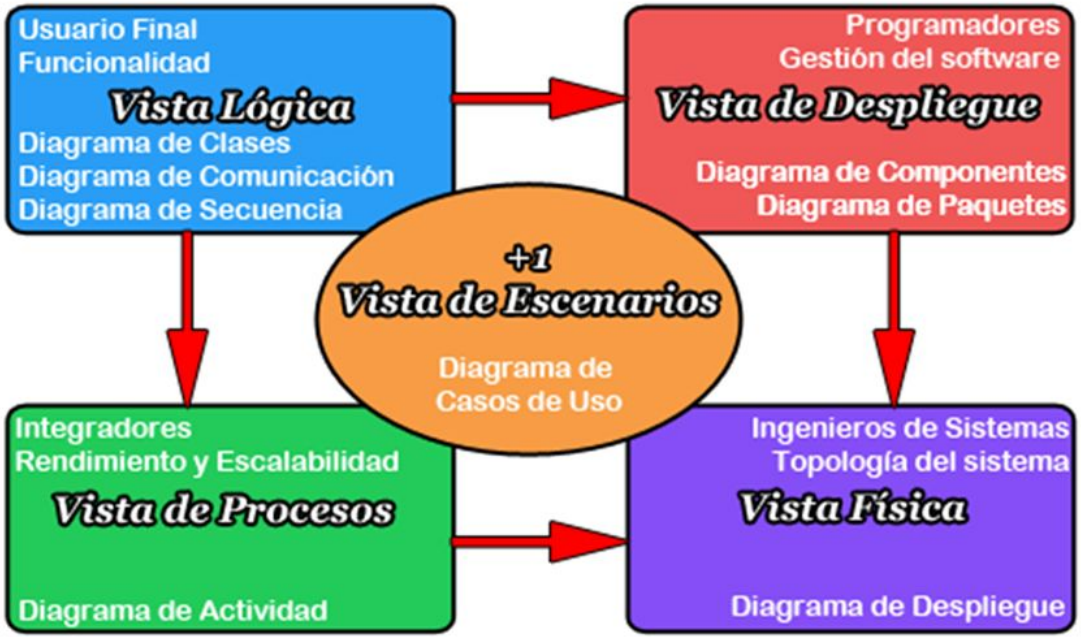
Preocupaciones

- Fuente: ISO/IEC 42010:2011

Name	Description	Use for Architecture Documentation
Architect	Responsible for the development of the architecture and its documentation. Focus and responsibility is on the system.	Negotiating and making trade-offs among competing requirements and design approaches. A vessel for recording design decisions. Providing evidence that the architecture satisfies its requirements.
Business manager	Responsible for the functioning of the business/organizational entity that owns the system. Includes managerial/executive responsibility, responsibility for defining business processes, and more.	Understanding the ability of the architecture to meet business goals.
Conformance checker	Responsible for assuring conformance to standards and processes to provide confidence in a product's suitability.	Basis for conformance checking, for assurance that implementations have been faithful to the architectural prescriptions.
Customer	Pays for the system and ensures its delivery. The customer often speaks for or represents the end user, especially in a government acquisition context.	Assuring required functionality and quality will be delivered, gauging progress, estimating cost, and setting expectations for what will be delivered, when, and for how much.
Database administrator	Involved in many aspects of the data stores, including database design, data analysis, data modeling and optimization, installation of database software, and monitoring and administration of database security.	Understanding how data is created, used, and updated by other architectural elements, and what properties the data and database must have for the overall system to meet its quality goals.

Modelo de Vista Arquitectónica 4+1

Fuente: https://www.cs.ubc.ca/~gregor/teaching/papers/4+1view-architecture.pdf



Múltiples vistas en una arquitectura

- **Vista Lógica**: Abstracciones Claves del Sistema. Relaciones requerimientos del sistema con entidades.
- **Vista de Proceso**: Procesos en interacción. Permite evaluar características no funcionales del sistema como rendimiento y disponibilidad.
- **Vista de Desarrollo**: Muestra la descomposición del software. A nivel de desarrollo, cada componente debería de ser implementado por un desarrollador o equipo de desarrollo.
- **Vista Física:** Expone el hardware del sistema. Muestra la relación entre componentes y hardware.

UNIVERSIDAD DE LIMA | 60 años

Arquitecturas de Transición

Estado actual y futuro de una arquitectura

AS-IS vs. TO-BE

- Arquitectura de Transición
- **AS - IS:** Estado actual de la arquitectura en cada una de las vistas arquitectónicas
- **TO - BE:** Estado futuro de la arquitectura en cada una de las vistas arquitectónicas
- Transición:
    - Cuando la evolución del AS-IS al TO-BE es complejo

Arquitecturas de Transición

- Conjunto de Arquitecturas intermedias entre el AS-IS y el TO-BE.
- Representa una arquitectura en un momento determinado.
- Utilizadas cuando el proceso de transición entre el AS-IS y el TO-BE es muy complejo
- Complejo a nivel:
  - Tecnológico
  - Gobierno
  - Tiempo
  - Recursos

- Arquitecturas de Transición
  - Implementación de Middleware.
  - Componentes redundantes (antiguo vs. nuevo).
  - Creación de nuevas interfaces.

UNIVERSIDAD DE LIMA | 60 años

Propuesta de un documento de arquitectura

Resaltar aspectos importantes de un documento de arquitectura

Documento de Arquitectura

1. Metas de la arquitectura
   - [código] meta 1
   - ….
2. Escenarios de atributos de calidad
   - [código] escenario de atributo de calidad 1
   - …
3. Restricciones
   - [código] restricción 1
   - ….
4. Estilos arquitecturales

Estilo	Justificación
<nombre>	<explicar las razones por las que se elige el estilo>

UNIVERSIDAD DE LIMA | 60 años

1. Meta de la arquitectura

Ejemplos de metas

- Los metas arquitecturales proporcionan la motivación y el fundamento de las decisiones de arquitectura.
- Definen cómo el sistema debe responder a los cambios a lo largo del tiempo.

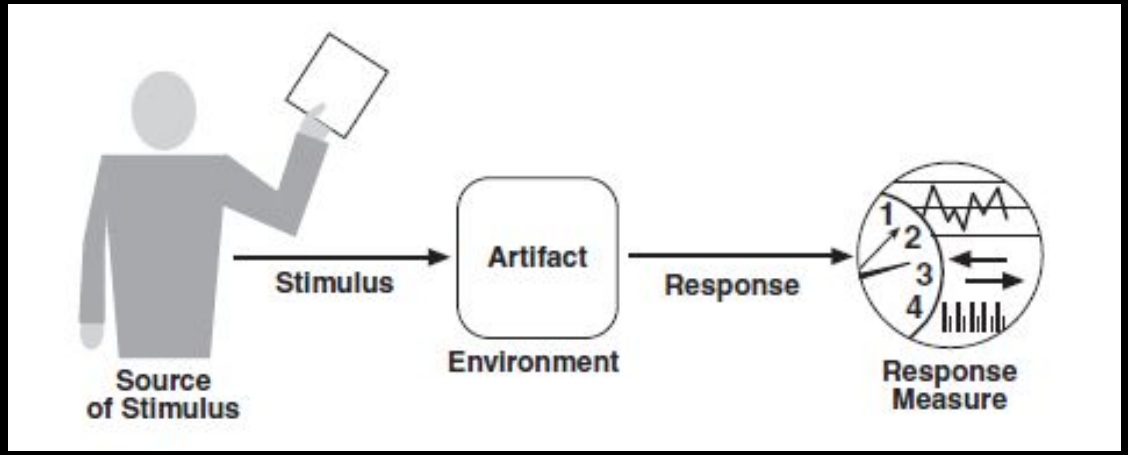
More about of The goals of Software Architecture

Código	Meta
AG001 Interoperability	The WSA should enable the development of interoperable Web services across a wide array of environments
AG002 Reliability	The WSA must be reliable and stable over time
AG003 Integration with the World Wide Web	The WSA must be consistent with the current and future evolution of the World Wide Web.

Source: WSA: Web Service Architecture

2. Escenarios de atributos de calidad

Escenarios de atributos de calidad



- Un atributo de calidad (QA) es una propiedad medible o comprobable de un sistema que se utiliza para indicar qué tan bien el sistema satisface las necesidades de los stakeholders.
- Atributos de calidad: usabilidad, confiabilidad, desempeño, etc.
- Tiene que ser descritos usando escenarios de atributos de calidad.

Ver video

Escenario de calidad	#1
Atributo de calidad	Desempeño
Fuente del estímulo	Cuando 1000 profesores de forma concurrente
Estímulo	Suben un archivo de 15 MB
Entorno	Con una velocidad de subida en internet entre 1 y 1.5 Mbps
Artefacto	A través de la página para subir archivos del sistema del aula virtual
Respuesta	El archivo es subido y puesto a disposición de lo estudiantes
Medida de la respuesta	95 % de los archivos están disponibles en menos de 5 segundos.

Ejemplo: Escenarios de atributos de calidad

UNIVERSIDAD DE LIMA | 60 años

3. Restricciones

Las restricciones son decisiones de diseño de la arquitectura que ya han sido tomadas.

Restricciones

Ejemplos de restricciones:

- Lenguaje de programación.
- Base de datos.
- Infraestructura.
- Estilo arquitectónico (Por ejemplo: SOA).
- Otros.

1. *Functional requirements.* These requirements state what the system must do, and how it must behave or react to runtime stimuli.
2. *Quality attribute requirements.* These requirements are qualifications of the functional requirements or of the overall product. A qualification of a functional requirement is an item such as how fast the function must be performed, or how resilient it must be to erroneous input. A qualification of the overall product is an item such as the time to deploy the product or a limitation on operational costs.
3. *Constraints.* A constraint is a design decision with zero degrees of freedom. That is, it's a design decision that's already been made. Examples include the requirement to use a certain programming language or to reuse a certain existing module, or a management fiat to make your system service oriented. These choices are arguably in the purview of the architect, but external factors (such as not being able to train the staff in a new language, or having a business agreement with a software supplier, or pushing business goals of service interoperability) have led those in power to dictate these design outcomes.

UNIVERSIDAD DE LIMA | 60 años

4. Estilos arquitecturales

Estilos arquitecturales

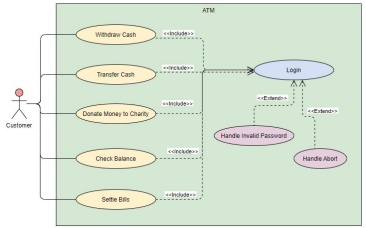
- Mencionar los estilos arquitecturales elegidos y la justificación.
- Estilos arquitecturales: SOA, N-capas, Microservicios, Otros

Estilo	Justificación
<nombre del estilo 1>	<explicar las razones por las que se elige el estilo>
<nombre del estilo 2>	<explicar las razones por las que se elige el estilo>

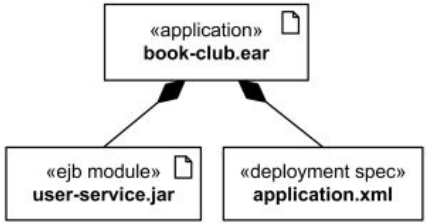
UNIVERSIDAD DE LIMA | 60 años

Vistas en el Documento de Arquitectura

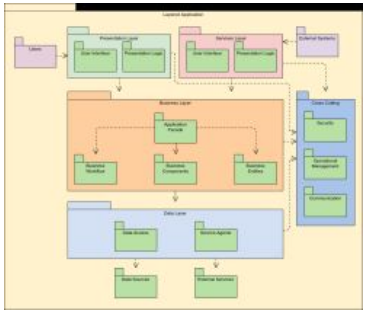
Documento de Arquitectura

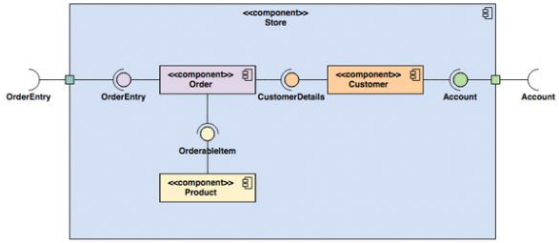


Vista funcional

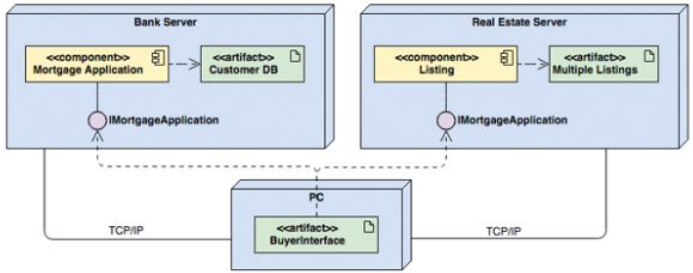


Vista de implementación





Vista lógica



Vista de despliegue

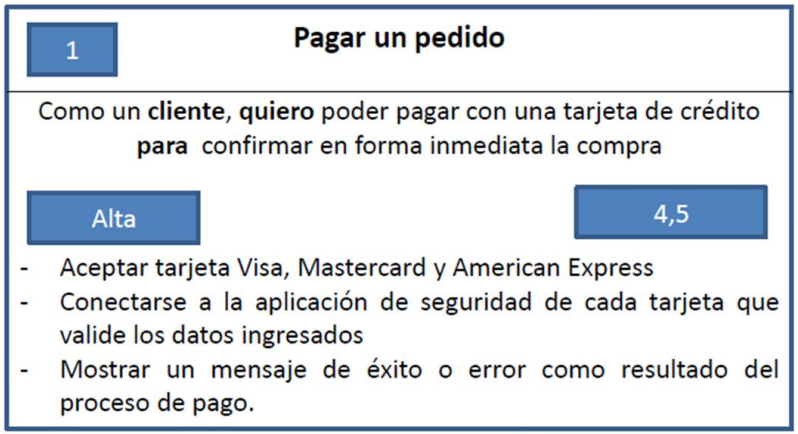
UNIVERSIDAD DE LIMA | 60 años

Vista funcional

Se utiliza para presentar los requisitos significativos para la arquitectura

Documentación de la vista funcional

- Uno o varios diagramas de casos de uso mostrando los requisitos funcionales (casos de uso o **historias**) significativas para la arquitectura.
- Información relevante sobre los requisitos: código, proposito, actores, etc.



Nota. - Para el caso del curso utilizamos **Historias de usuario**

UNIVERSIDAD DE LIMA | 60 años

Vista lógica

Se utiliza para mostrar **lógicamente** como está organizado y estructurado la arquitectura. También, para **satisfacer** los **requerimientos funcionales** y **los atributos de calidad**.

Consideraciones de diseño para *vista lógica*

Para el diseño de la vista lógica tener en cuenta:

- Las metas de la arquitectura.
- Los escenarios de atributos de calidad.
- Los requisitos funcionales significativos.
- Restricciones (Ejemplo Cloud con AWS).
- Estilos arquitectónicos seleccionados (capas, orientada a servicios, orientación a objetos, otros). **Los estilos arquitecturales guían el diseño**.

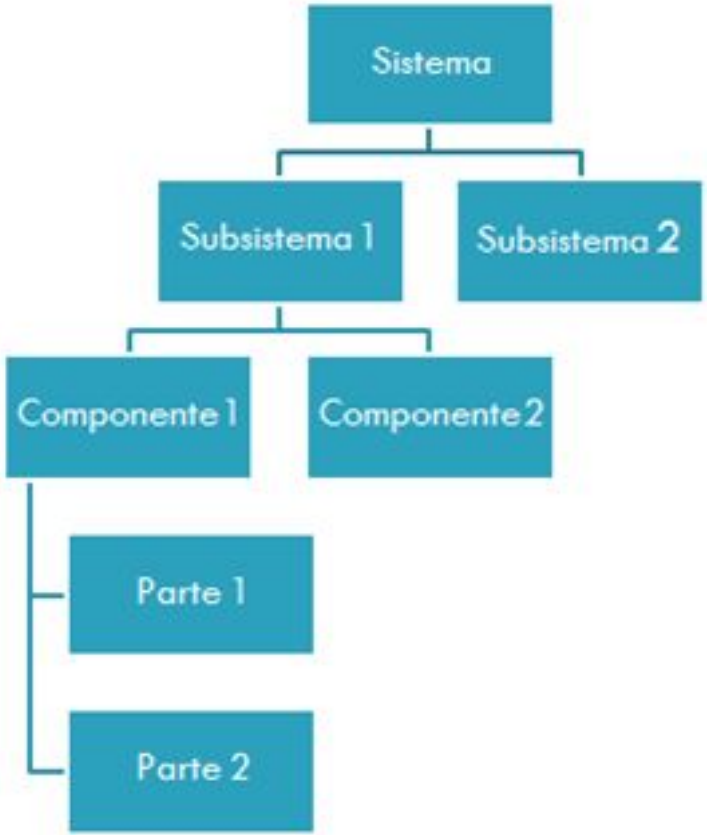
Niveles de descomposición - *vista lógica*

Útil para determinar el nivel de detalle de la arquitectura.

- Sistema
- Subsistema
- Componentes
- Parte
  - Interfaz en Java
  - Clase en Java
  - Archivo XML
  - Archivo de propiedades

Esto es arquitectura

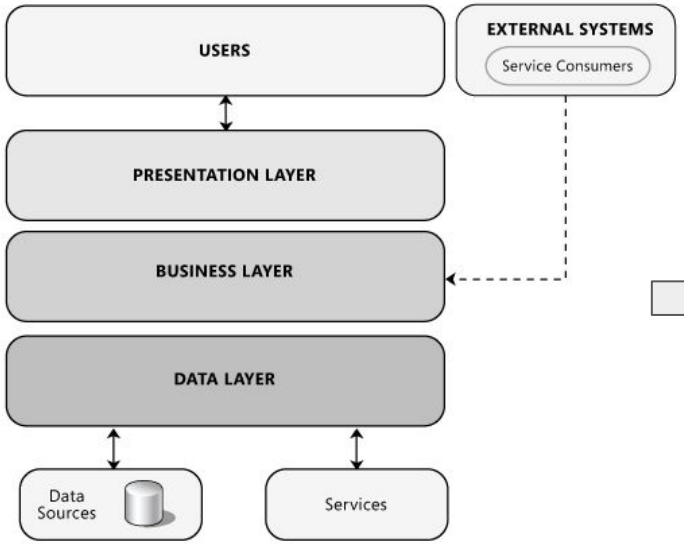
Esto no es arquitectura



Fuente: Nasa

Cómo aplicar un estilo de arquitectura a la vista lógica

Arquitectura N-capas aplicada a una aplicación web



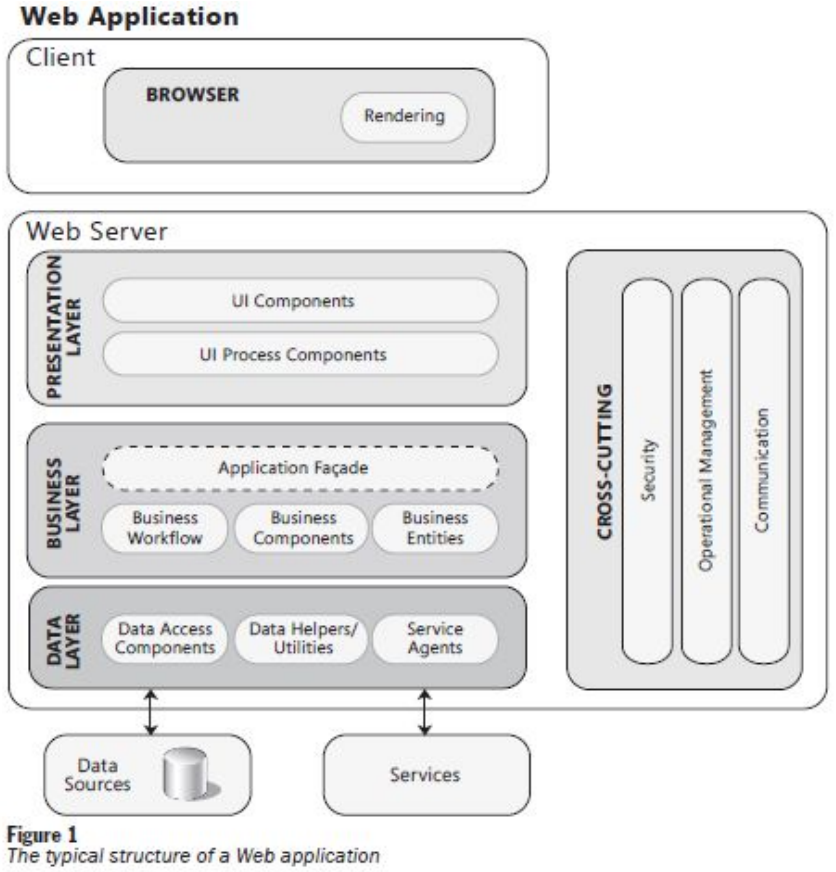
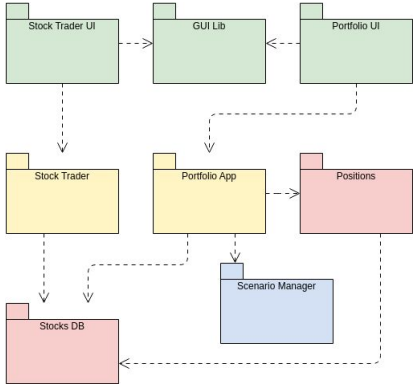


Figure 1
The typical structure of a Web application

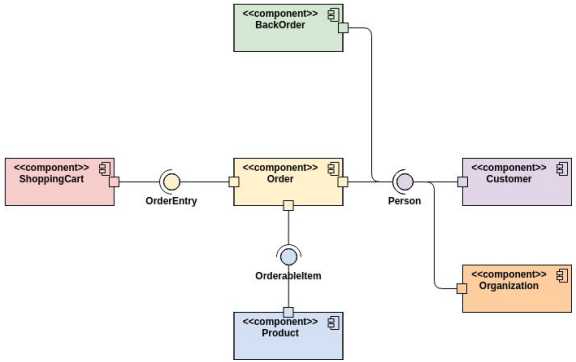
Fuente: [Microsoft Application Architecture Guide, 2nd Edition]

¿Cómo modelar la estructura de la solución?

- Diagrama de paquetes de UML
- Diagrama de componentes de UML



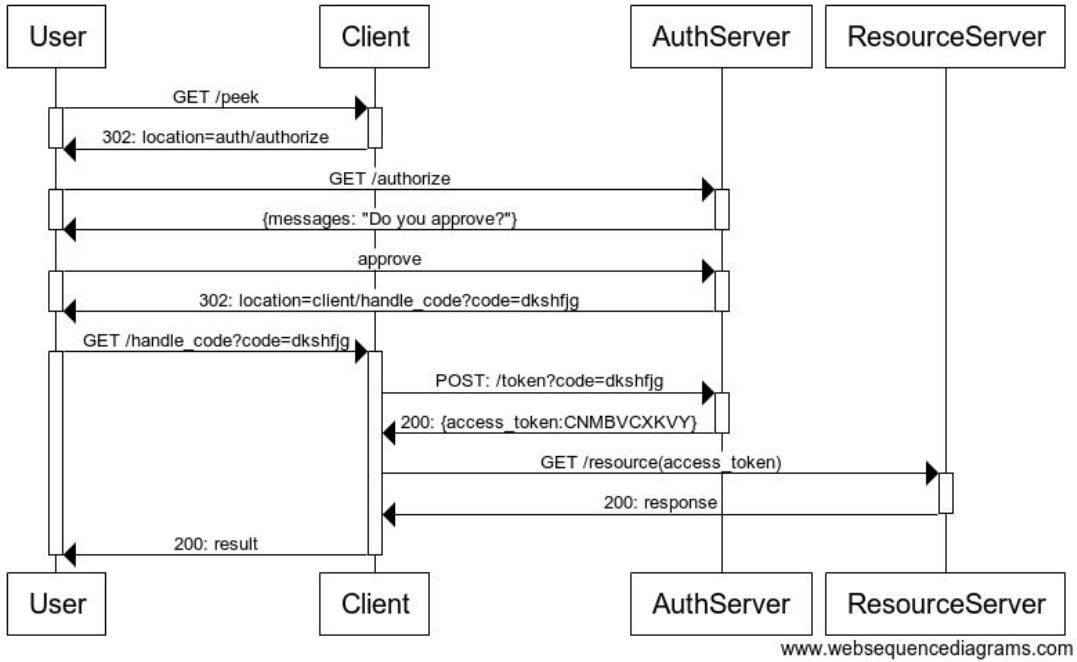
https://online.visual-paradigm.com/diagrams/templates/package-diagram/



https://online.visual-paradigm.com/diagrams/templates/component-diagram/

El comportamiento

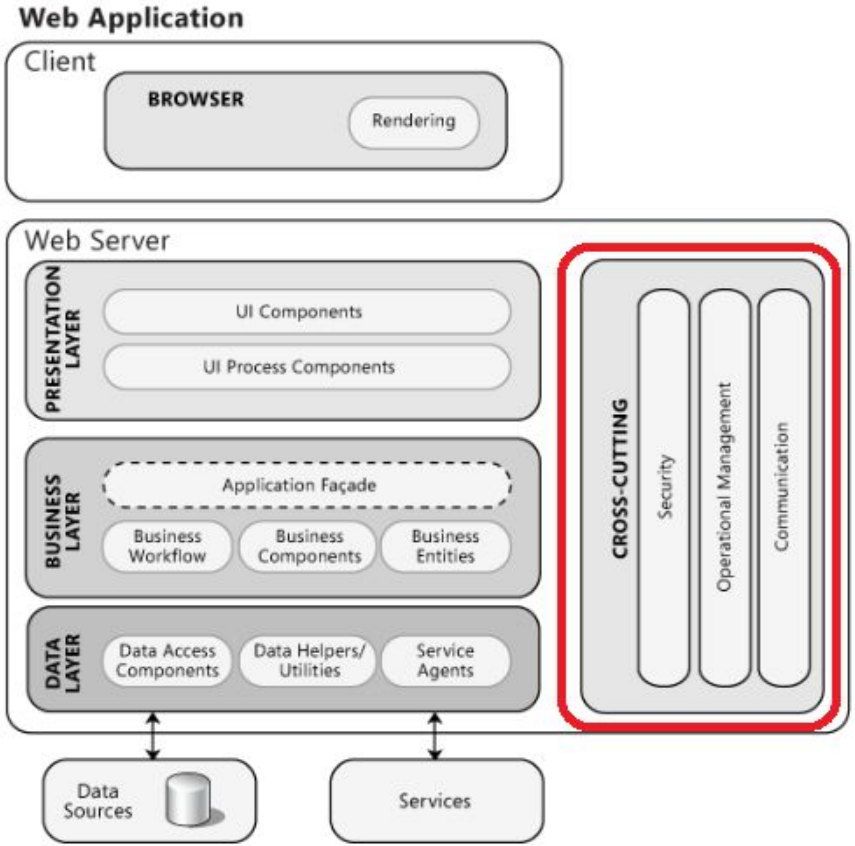
- El comportamiento es parte esencial de la arquitectura
- ¿Cómo modelar el comportamiento?
  - Diagrama de secuencia en UML



Otros ejemplos:
https://online.visual-paradigm.com/diagrams/templates/sequence-diagram/

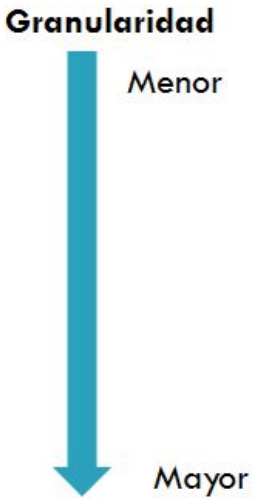
Requisitos técnicos

- Requisito significativos
  - **De negocio** (Ejem. Subir documento, Registrar notas, Registrar asistencia)
  - **Técnicos** (Ejem. Autenticación, Autorización, Enviar correo, Ayuda en línea)
- Los requisitos técnicos influyen en la vista lógica.
- Los requisitos técnicos atraviesan toda la aplicación: autenticación, autorización, almacenamiento en caché, comunicación, administración de excepciones, auditoría, persistencia etc.



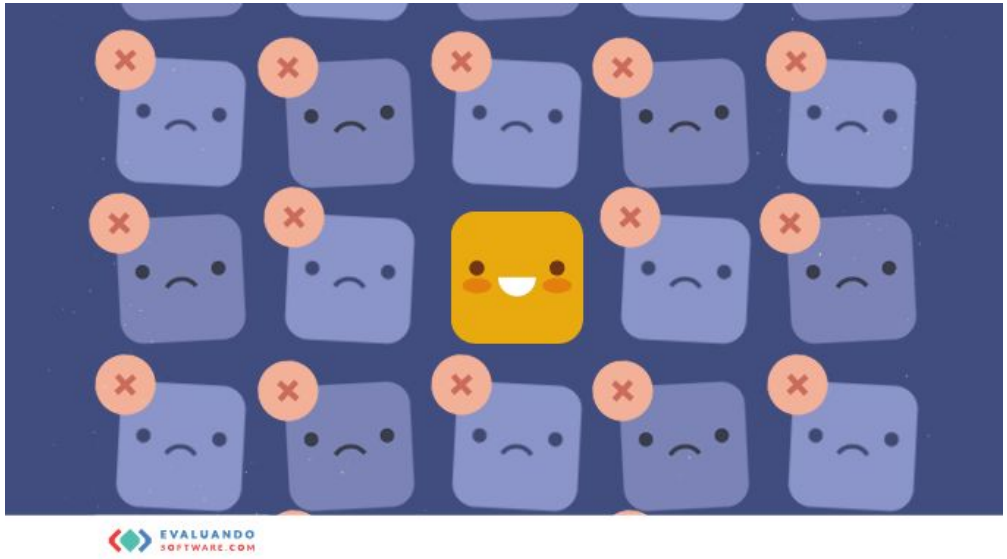
Contenido referencial de la vista lógica

Tema	Nivel	Contenido
Visión general de la arquitectura	Sistema	Diagramas de paquetes mostrando los subsistemas (según los estilos arquitecturales), principios de diseño, información general sobre cada subsistema, decisiones de arquitectura, restricciones, etc.
Arquitectura de cada subsistema	Subsistema	Diagramas de paquetes mostrando la organización de cada subsistema (según los estilos de arquitectura), principios de diseño, información relevante sobre cada subsistema, decisiones de arquitectura, restricciones, etc.
Comportamiento	Componente	Diagramas de secuencia y de componente (respetando los estilos arquitecturales) para cada requisito de negocio o técnico, escenarios de atributos de calidad se deben tomar en cuenta, información relevante sobre cada componente, decisiones de arquitectura, restricciones, etc.



Errores más comunes en la vista lógica

- **No conocer en profundidad los estilo arquitectónicos**.
- Diagramas genéricos.
- Considerar sólo los requisitos de negocio.
- Uso incorrecto de la notación UML.
- No explicar los diagramas.



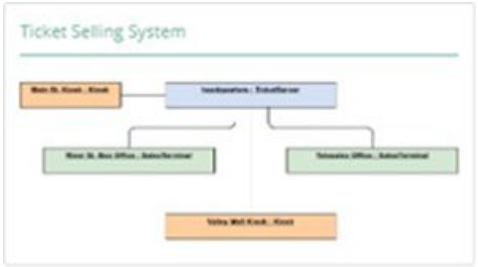
UNIVERSIDAD DE LIMA | 60 años

Vista de implementación

Presenta las **unidades de despliegue** (una por subsistema) que conformarán finalmente el sistema y sus dependencias.
Una **unidad de despliegue** contiene una parte del código compilado de la aplicación y/o archivos. También, otros archivos necesarios para su ejecución.

¿Cómo modelar la implementación?

- Contenido:
  - Unidades de despliegue y sus interdependencias.
    - Diagrama de despliegue en UML 2.0



Contenido referencial de la vista de implementación

Tema	Contenido
Restricciones a la implementación	Relación de restricciones consideradas en la vista de implementación
Escenarios de atributos de calidad	Relación de escenarios de atributos de calidad considerados en la vista de implementación
Unidades de despliegue y sus Interdependencias	Diagramas de despliegue mostrando las unidades de despliegue (artefactos) y sus interdependencias, información relevante sobre cada artefacto, etc.

UNIVERSIDAD DE LIMA | 60 años

Vista de despliegue

Presenta el **entorno de ejecución** necesario para ejecutar el software conforme a los requerimientos de disponibilidad, rendimiento, escalabilidad, entre otros; así como la distribución de las unidades de despliegue en el entorno de ejecución.

Patrón N-Tier

- El patrón de N-Niveles representa un patrón general de distribución donde los componentes de la aplicación se separan a través de uno o más nodos físicos.

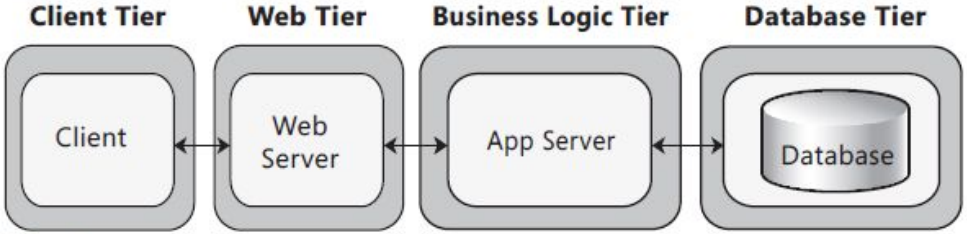


Figure 6
4-tier deployment with the Web code and the business logic on separate tiers

Consider the 4-tier pattern if security requirements dictate that business logic cannot be deployed to the perimeter network, or you have application code that makes heavy use of resources on the server and you want to offload that functionality to another server.

Fuente: [Microsoft Application Architecture Guide, 2nd Edition]

¿Cómo modelar el despliegue?

- Contenido
  - Infraestructura de software y hardware necesaria para la ejecución del sistema, así como la ubicación de las unidades de despliegue
- Representación gráfica
  - Diagrama de despliegue en UML 2.0

Contenido referencial de la vista de despliegue

Tema	Contenido
Restricciones a la implementación	Relación de restricciones consideradas en la vista de despliegue
Escenarios de atributos de calidad	Relación de escenarios de atributos de calidad considerados en la vista de despliegue
Principios de diseño	Principios de diseño empleado en la vista de despliegue
Infraestructura de ejecución	Diagramas de despliegue mostrando la infraestructura de software y hardware necesaria para ejecutar la solución, información relevante sobre cada nodo, protocolos de comunicación, explicación de los diagramas

Referencias Bibliográficas

1. Joel Moreno. (2018). Construcción de Software 2 Documentación de Arquitectura. Maestría en Informática en la PUCP.
2. Sommerville, I. (2015). Software engineering 10th Edition. ISBN-10, 137035152, 18.
3. Bennett, S., McRobb, S., & Farmer, R. (2010). Object-oriented systems analysis and design using UML. McGraw-Hill.
4. Pressman, R. S., & Troya, J. M. (2010). Ingeniería del software. McGraw-Hill.