

Clase 3: Diagramas del Diseño

Rodrigo Arturo Saffie Kattan

Pontificia Universidad Católica de Chile

rasaffie@ing.puc.cl

9 de agosto de 2016

1 Repaso Clase Anterior

2 Vistas del Diseño

- Descripción del Diseño
- Modelo 4 + 1

El diseño se divide en 4 áreas:

- Diseño de componentes
- Diseño de arquitectura
- Diseño de clases/datos
- Diseño de interfaces

¿Qué cualidades debe tener la descripción del diseño de un *software*?

- Sin ambigüedades, preciso y objetivo
- Convenciones claras para los involucrados
- Modificable, con menor costo que la construcción

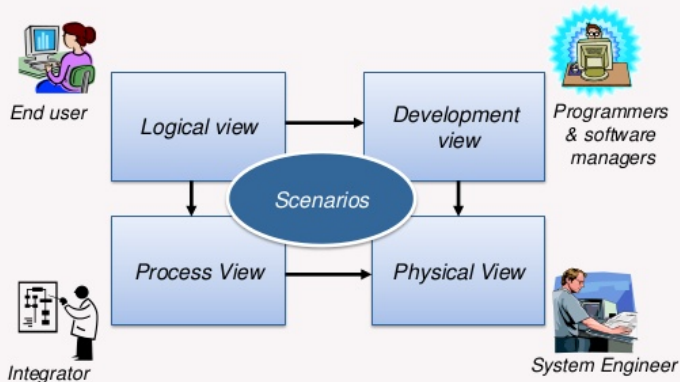
¿Quiénes están interesados en la descripción del diseño?

- Desarrolladores
- Líderes de proyectos
- Arquitectos de sistemas
- Analistas
- Clientes
- Usuarios

- Propuesto por Philippe Kruchten en 1995

Framework para describir la arquitectura de un *software*, basado en el uso de múltiples vistas concurrentes.

The 4+1 View model



- **Describes software architecture using five concurrent views.**

¿Qué es una vista?

Es una representación del sistema completo, enfocada desde una perspectiva con consideraciones específicas.

Unified Modeling Language 2.0

- Estándar para especificar un sistema
- 1º versión propuesta en 1997
- Ayuda a entender y explicar un sistema de manera consistente

Buena referencia: [Ambler, S. W., 2005]

Vista Lógica

Interesados : Usuarios Finales

Consideraciones : Requisitos Funcionales

¿Qué es lo que el sistema debería proveer a sus usuarios?

Diagramas : de Clases, de Comunicación, de Secuencia, de Estado

Diagrama de Clases

Modela las clases, sus relaciones, sus operaciones y sus atributos.

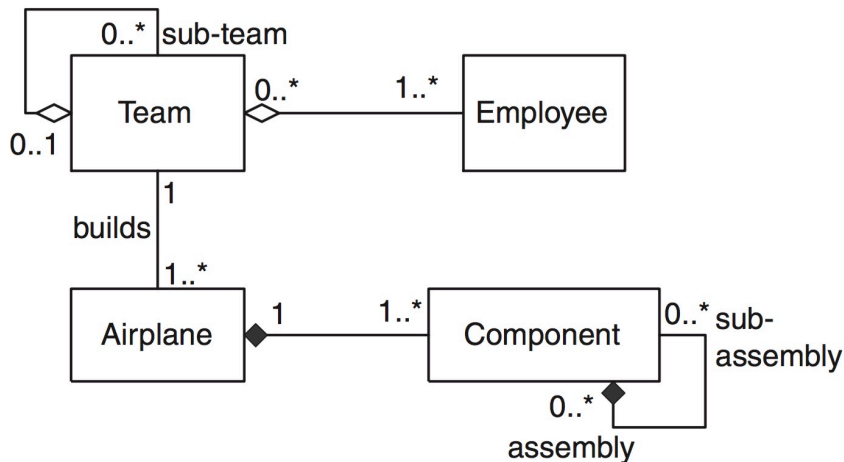


Diagrama de Comunicación

Modela la comunicación y dirección de los mensajes entre clases.

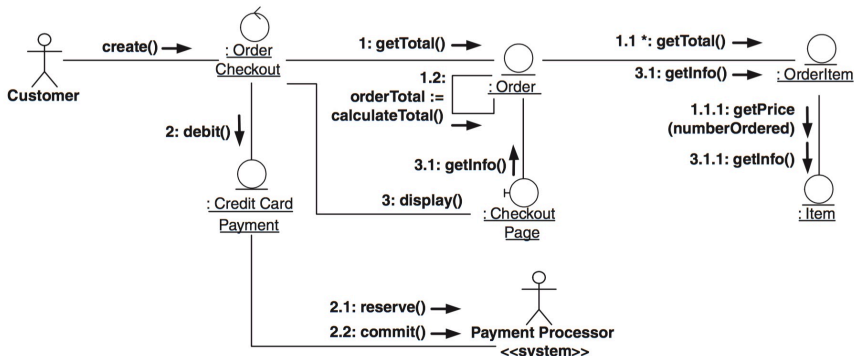


Figure 33. An instance-level UML communication diagram.

Diagrama de Secuencia

Modela la interacción y orden en que las entidades se relacionan.

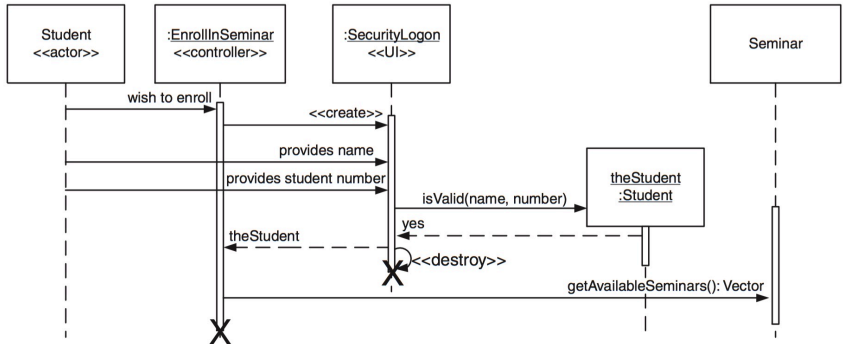
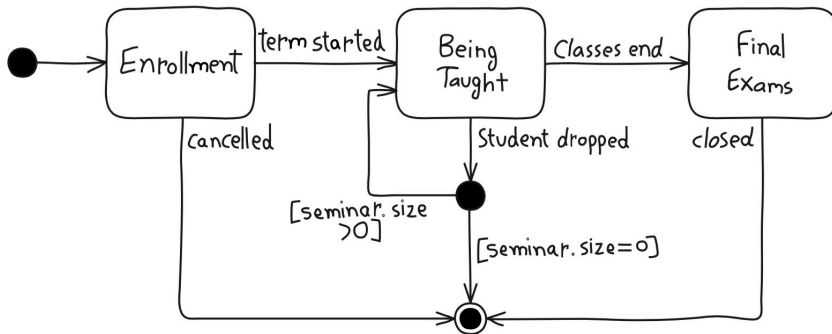


Diagrama de Estado

Modela el comportamiento de las entidades dado distintos estados, basado en sus respuestas a eventos.



Vista de Procesos

Interesados : Analistas

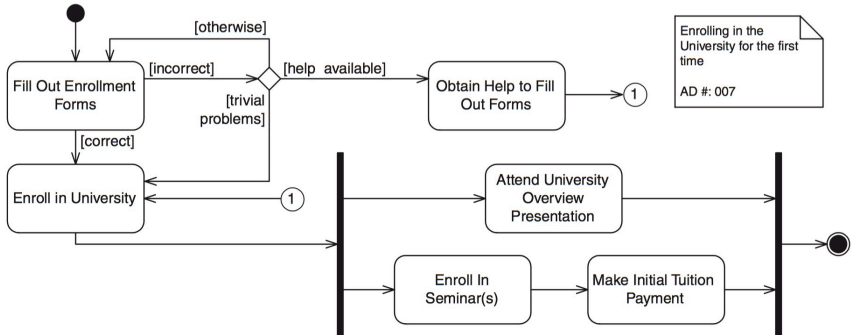
Consideraciones : Requisitos No Funcionales

¿Qué procesos y reglas tiene el sistema, y cómo interactúan los componentes?

Diagramas : de Actividad

Diagrama de Actividad (o Flujo)

Modela procesos en base a reglas.



Vista de Implementación

Interesados : Desarrolladores y Líderes de proyectos

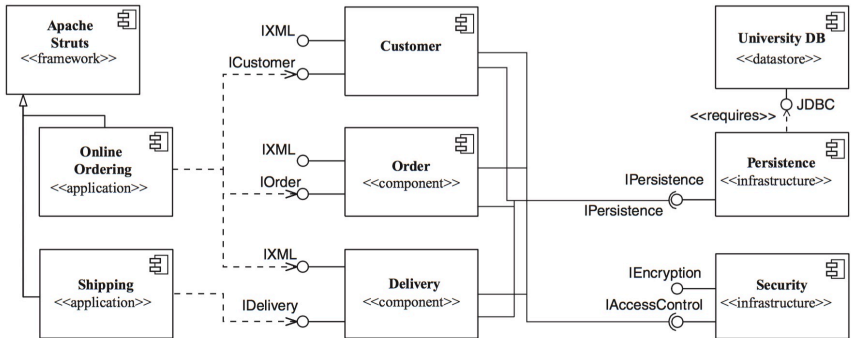
Consideraciones : Organización del Software

¿Cómo se organiza el software del sistema?

Diagramas : de Componentes, de Paquetes

Diagrama de Componentes

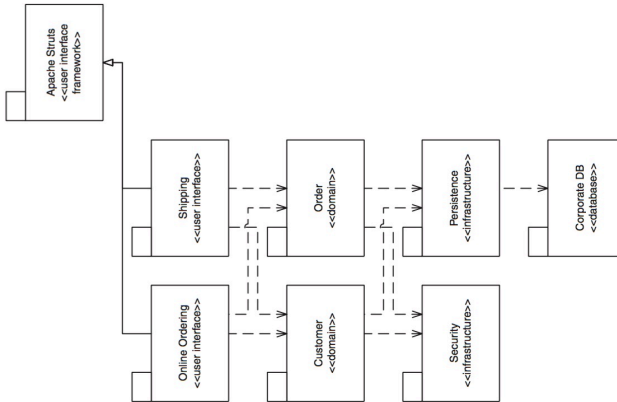
Modela la dependencia entre los componentes del *software*.



Vista de Implementación

Diagrama de Paquetes

Modela la dependencia entre los paquetes del *software*. Un paquete se puede ver como un carpeta del proyecto.



Vista Física

Interesados : Arquitectos de Sistemas

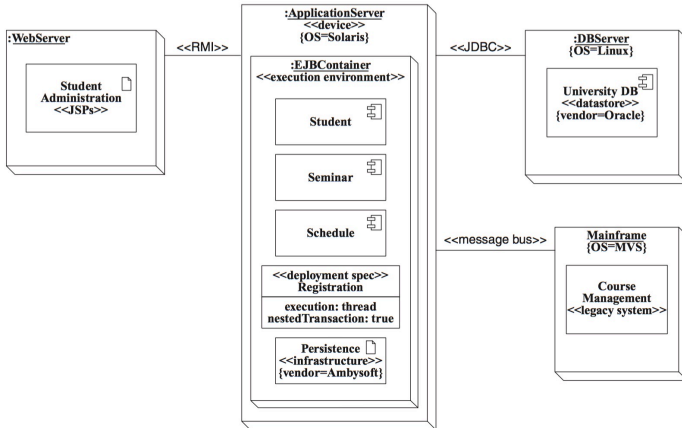
Consideraciones : Requisitos no Funcionales

¿Cómo es el ambiente de ejecución del sistema?

Diagramas : de Despliegue

Diagrama de Despliegue

Modela la configuración en tiempo de ejecución del *hardware*, y el *software* que se ejecuta.



Vista de Escenarios

Interesados : Todos

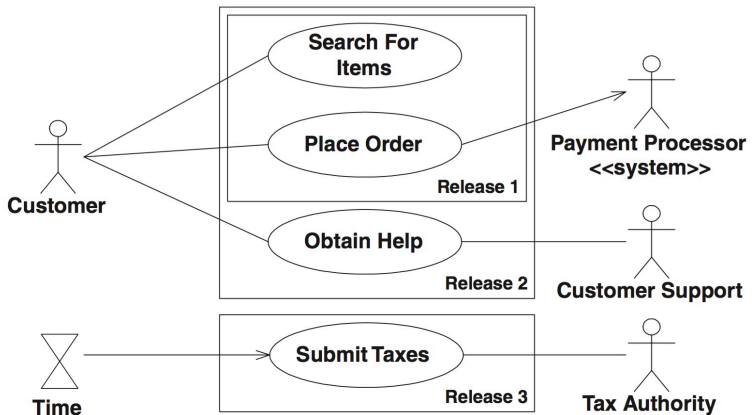
Consideraciones : Consistencia, Validez

¿Qué es lo que el sistema debería hacer?

Diagramas : de Casos de Uso

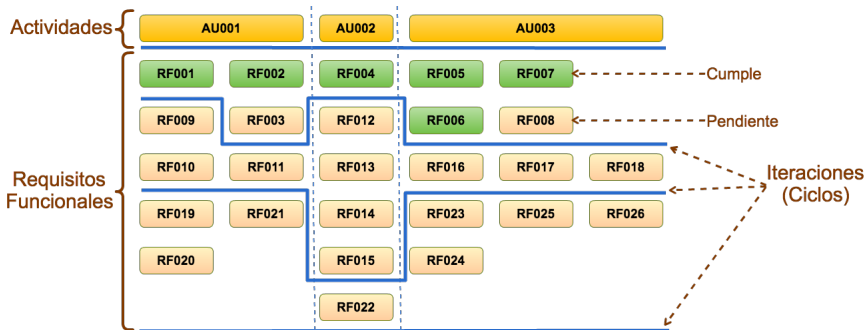
Diagrama de Casos de Uso

Modela la interacción entre los actores y las funcionalidades de un sistema.



Mapa de Historias de Usuarios [Patton, J., 2014]

Alternativa a los diagramas de casos de uso. (No es UML)



References



Pressman, R. S. (2009)

Software Engineering: A Practitioner's Approach

7th ed., *McGraw-Hill Education*



Kiruthika, R.S., Shobana, C.

4+1 View Model

<http://es.slideshare.net/shobanachokkalingam/41-view-model>



Kruchten, P. B. (1995)

The 4 + 1 view model of architecture

IEEE software, 12(6), 42-50.



Ambler, S. W. (2005)

The Elements of UML 2.0 Style

Cambridge University Press



Patton, J. (2014)

User Story Mapping

O'Reilly

Fin