Sistemas Distribuidos I
(75.74)

Middlewares y Documentación

Middlewares. Documentación Técnica. Diagramas de Diseño.

Docentes

- Pablo D. Roca
- Ezequiel Torres Feyuk
- Guido Albarello
- Ana Czarnitzki
- Cristian Raña

Agenda

- **Middlewares**
- Documentación
- Diagramas

Middlewares | Definiciones

"... software de conectividad que ofrece un conjunto de servicios que hacen posible el funcionamiento de aplicaciones distribuidas sobre plataformas heterogéneas..."

"... módulo intermedio que actúa como conductor entre sistemas permitiendo a cualquier usuario de sistemas de información comunicarse con varias fuentes de información que se encuentran conectadas por una red"

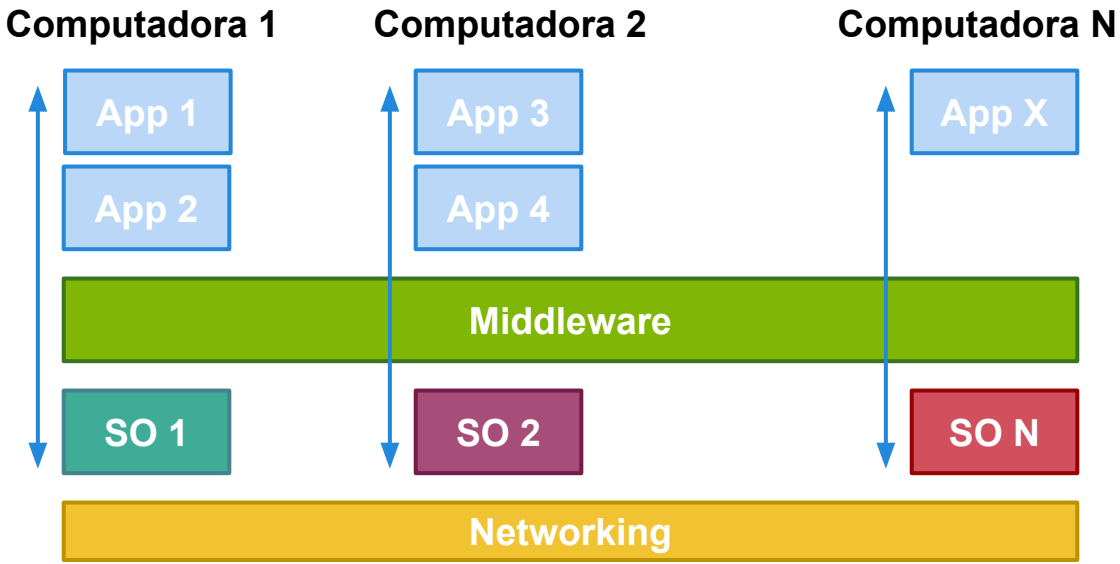
"... capa de software que se encuentra o sitúa entre el sistema operativo y las aplicaciones del sistema"

"... software que permite conectar componentes, softwares o aplicaciones. El mismo consiste en un conjunto de servicios que permiten que múltiples procesos corriendo en una o varias máquinas interactúen de un lado a otro de la red."

Source: Matías Leandro Varela, Introducción a Grid Computing

Middlewares | Vista lógica

Capa de software entre el sistema operativo y la capa de aplicación/usuario, para proveer una vista única del sistema.



Middlewares | Objetivos

Transparencia

- Se oculta la distribución y el sistema responde como si fuera una única computadora
- Respecto de: Acceso, Ubicación, Migración, Replicación, Concurrencia, Fallos, Persistencia

Tolerancia a Fallos

- Sistemas confiables, que se ejecuten y comporten de manera predecible incluso frente a la aparición de fallos.
- Abarcando: *Availability, Reliability, Safety, Maintainability*

Middlewares | Objetivos (II)

Acceso a recursos compartidos

- Eficiente, transparente y controlado

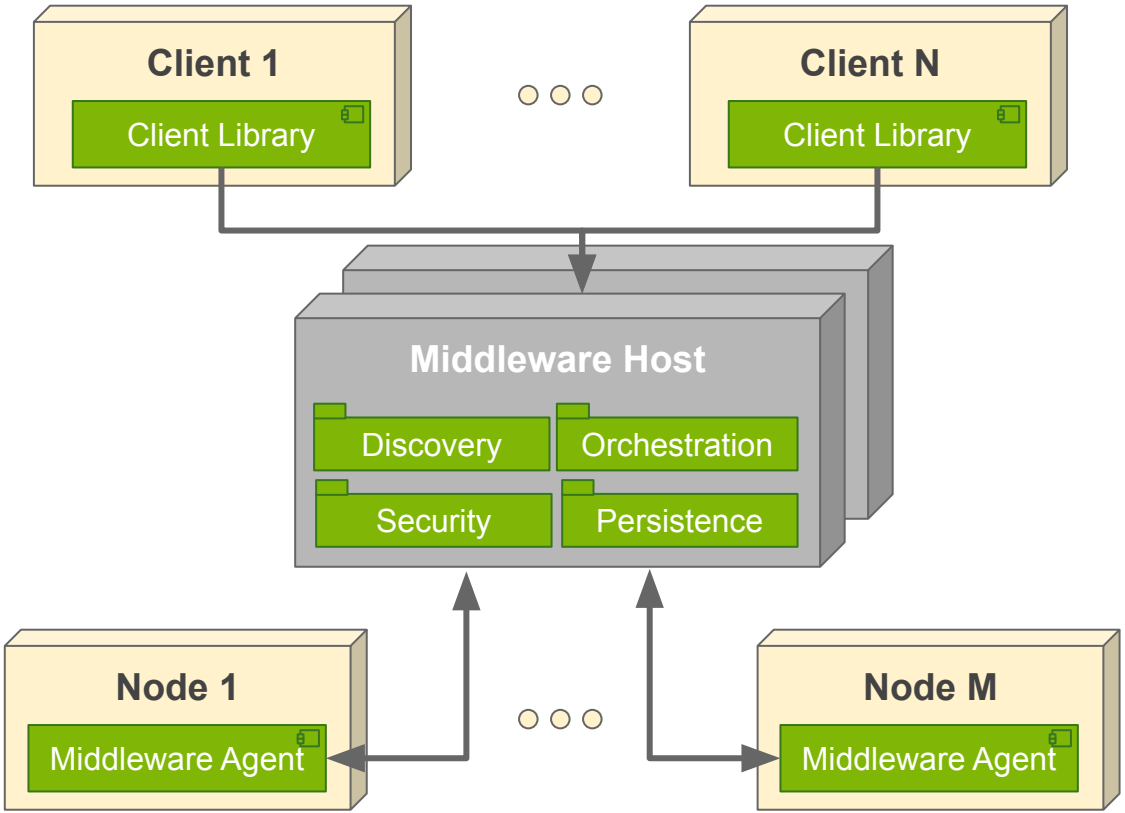
Sistemas distribuidos abiertos (Interfaces)

- Estándares claros sobre sintaxis y semántica de los servicios ofrecidos
- Interoperabilidad y portabilidad

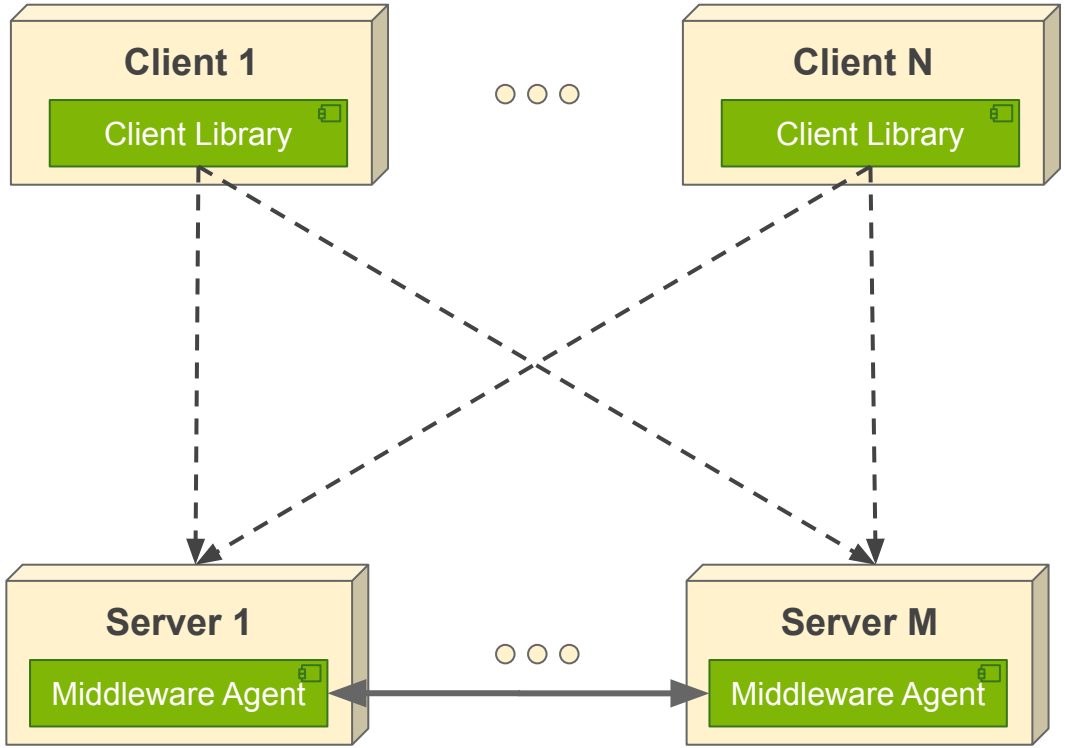
Comunicación de grupos

- Permite *broadcasting* y *multicasting*
- Facilita localización de elementos y coordinación de tareas

Middlewares | Vista Física (centralizado)



Middlewares | Vista Física (distribuido)



Middlewares | Clasificación

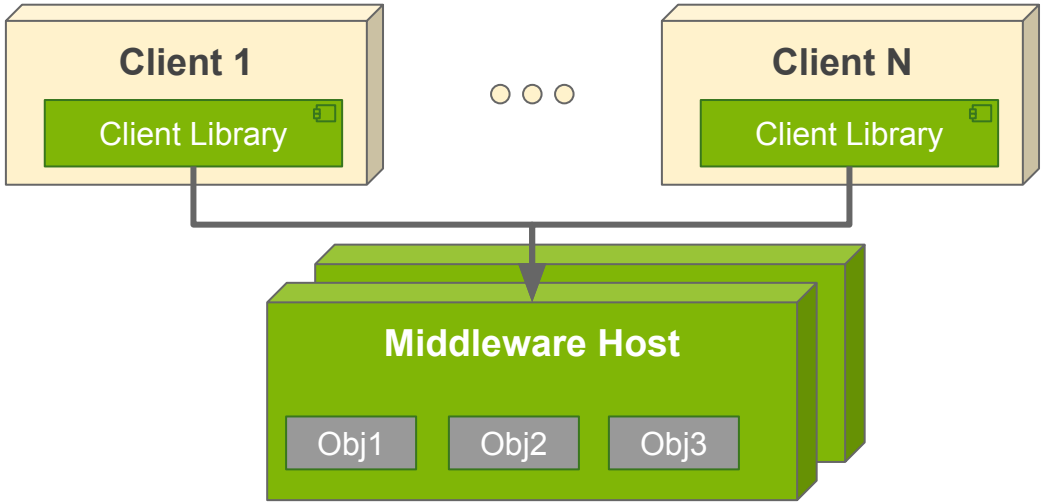
- *Transactional Procedure*
- *Procedure Oriented*
- *Object Oriented*
- *Message Oriented*
- *Reflective Middlewares* (de configuración dinámica)

Clasificación | *Transactional Processing*

- Permiten garantizar transaccionalidad de operaciones respecto de datos
- Conectan muchas fuentes de datos y permiten un acceso transparente al grupo
- Poseen políticas de reintentos y retención de datos frente a caídas internas

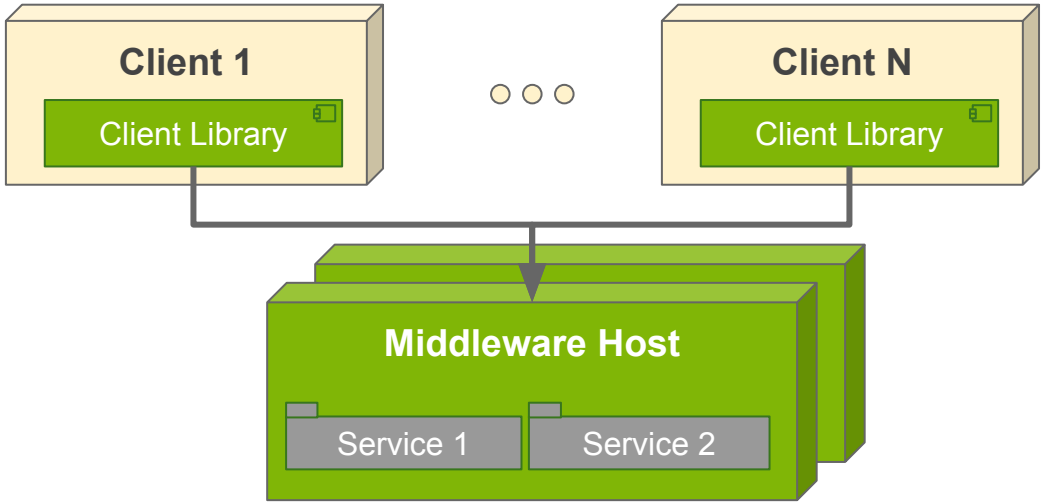
Clasificación | *Object Oriented*

- Mensajes hacia objetos distribuidos
- Los objetos viven dentro del *middleware*
- Esquema de 'marshalling' para transmitir la información



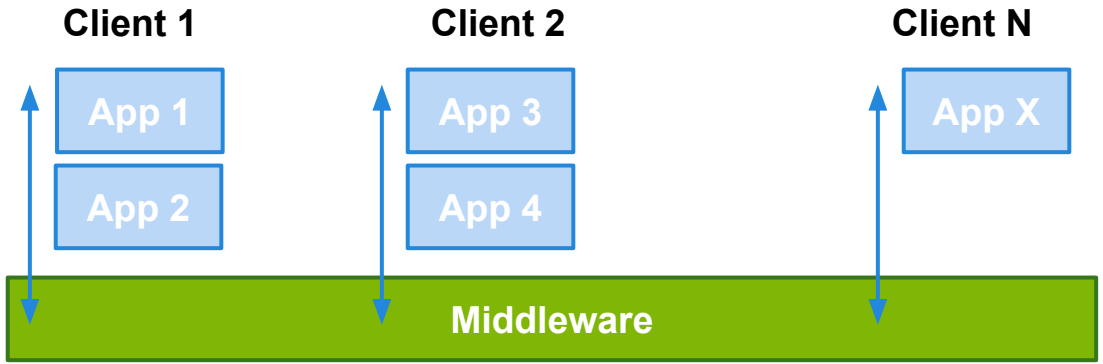
Clasificación | *Procedure Oriented*

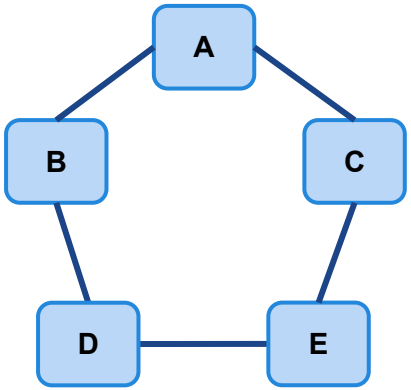
- El *middleware* trabaja como un servidor de funciones que se pueden invocar.
- Los servicios se pueden explorar y ejecutar pero no presentan estado para futuras invocaciones.



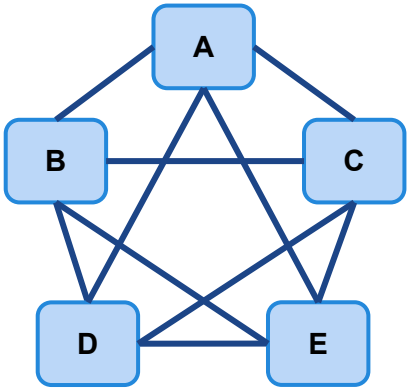
Clasificación | *Message Oriented*

- Funciona como un sistema de mensajería entre aquellas aplicaciones que utilizan el *middleware*
- Pueden enviarse mensajes bajo cierto 'tópico' para que aquellos interesados lo reciban (modo *Information Bus*)
- Pueden enviarse mensajes con un destinatario definido (modo *Queue*)

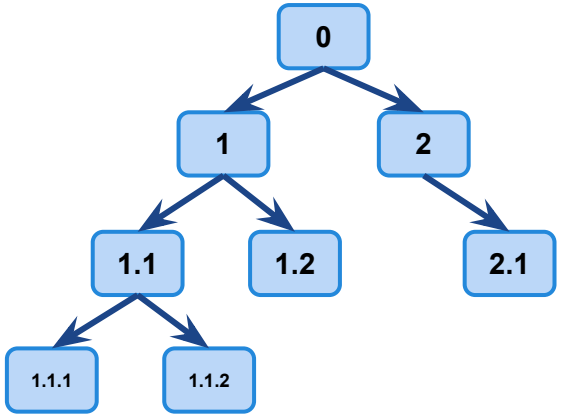




Anillos



Punto a Punto



Grupos Jerárquicos

Agenda

- Middlewares
- **Documentación**
- Diagramas

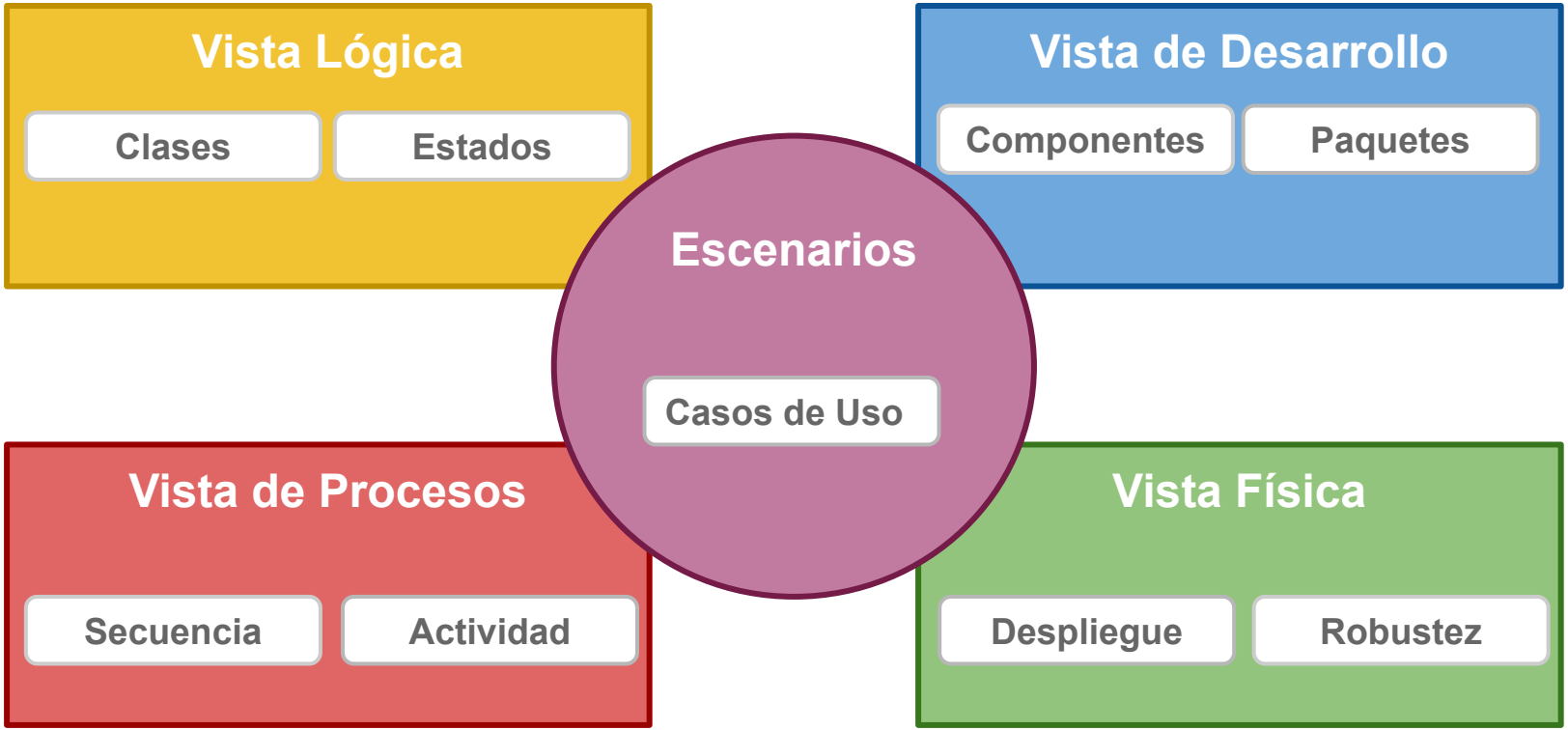
Documentación de Arquitectura

"La **arquitectura** representa aquellas decisiones de importancia, medidas de acuerdo al costo de modificarlas" (Grady Booch)

Diseño y Documentación:

- Evolutivo:
  - Adaptarse rápido, tomar feedback y aportar valor iterativamente
  - No buscar el entendimiento del todo y ni demorar la arquitectura
- Necesario para coordinación, coherencia y cohesión
  - Sin un diseño preliminar, probablemente jamás haya diseño.

Modelos de Doc. | Vistas de Arquitectura 4+1



4+1 *Views* | Objetivos de cada Vista

Vista Lógica

- Estructura y funcionalidad del sistema (Clases, Estados)

Vista de Física (o Despliegue)

- Topología y Conexiones entre componentes físicos (Despliegue)
- Expone la arquitectura del sistema (Robustez)

Vista de Desarrollo (o de Implementación)

- Artefactos que componen al sistema (Paquetes, Componentes)

Vista de Procesos (o Dinámica)

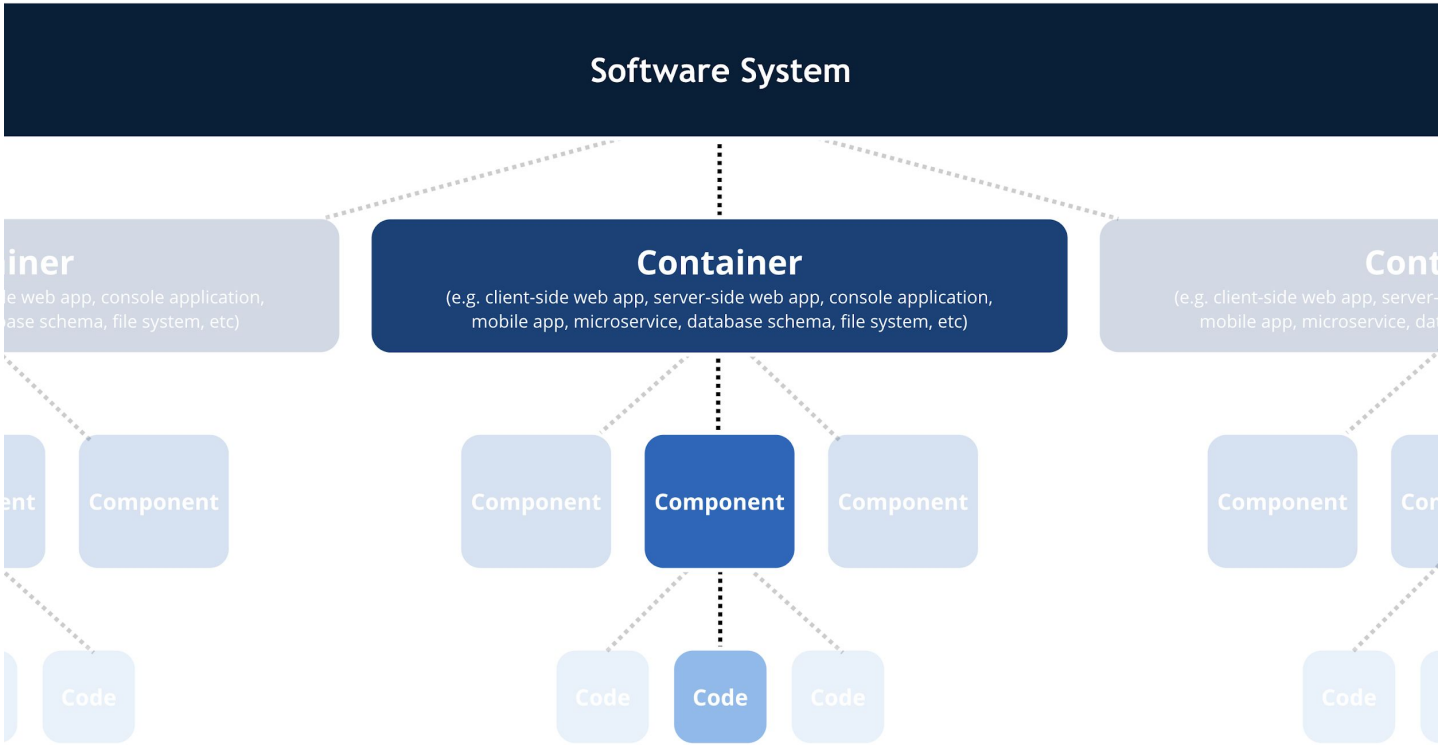
- Descripción de escenarios concurrentes (Actividades)
- Flujo de mensajes en el sistema (Colaboración)
- Flujo temporal de mensajes en el sistema (Secuencia)

4+1 *Views* | Contenido del documento

Ejemplo de Tabla de Contenidos:

1. Scope
2. Software Architecture
3. Architectural Goals & Constraints
4. Logical View
5. Process View
6. Development View
7. Physical View
8. Scenarios
9. Size and Performance
10. Quality Appendices
    - A. Acronyms and Abbreviations
    - B. Definitions
    - C. Design Principles

Modelos de Doc. | C4 *Model*

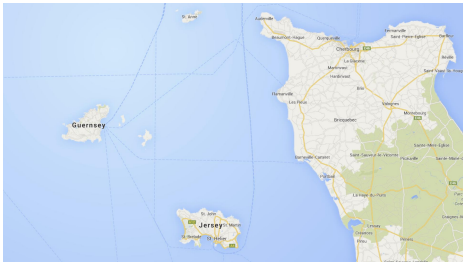


Source: Simon Brown, http://c4Model.com

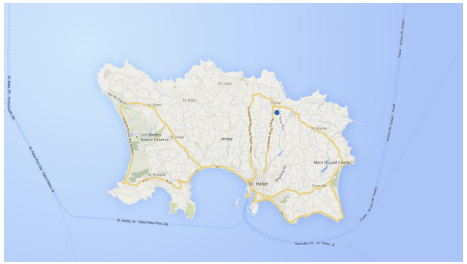
C4 Model | Niveles de Detalle

Nivel 1 - Contexto

Nivel 2 - *Containers*



Nivel 2 - *Components*



Nivel 4 - *Code*

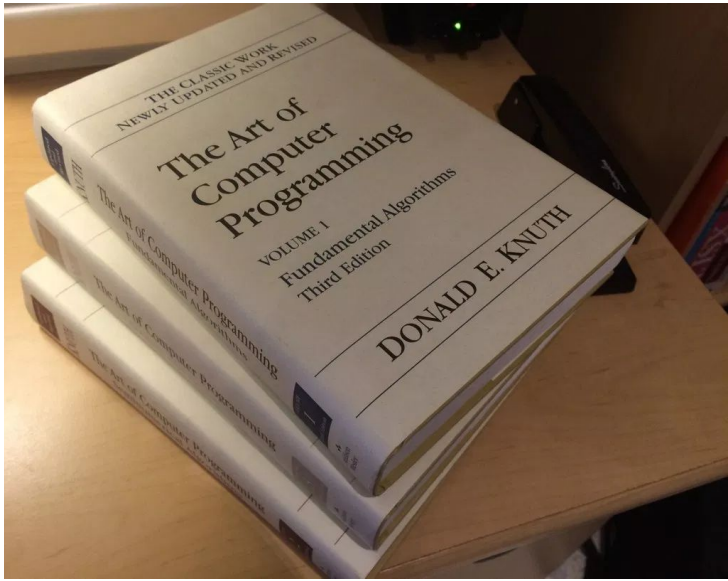
Source: Simon Brown, http://c4Model.com

Agenda

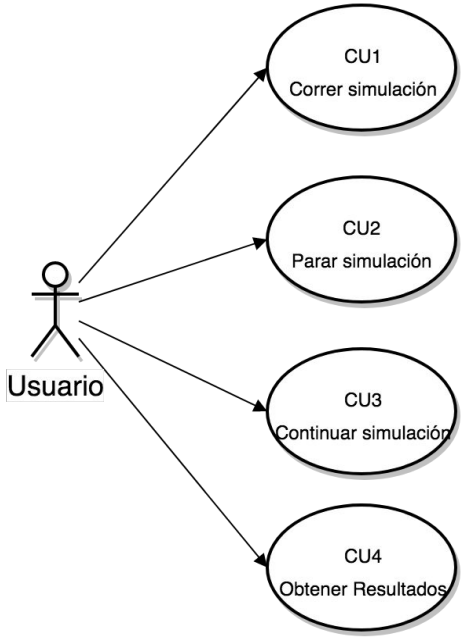
- Middlewares
- Documentación
- **Diagramas**

El arte de realizar diagramas

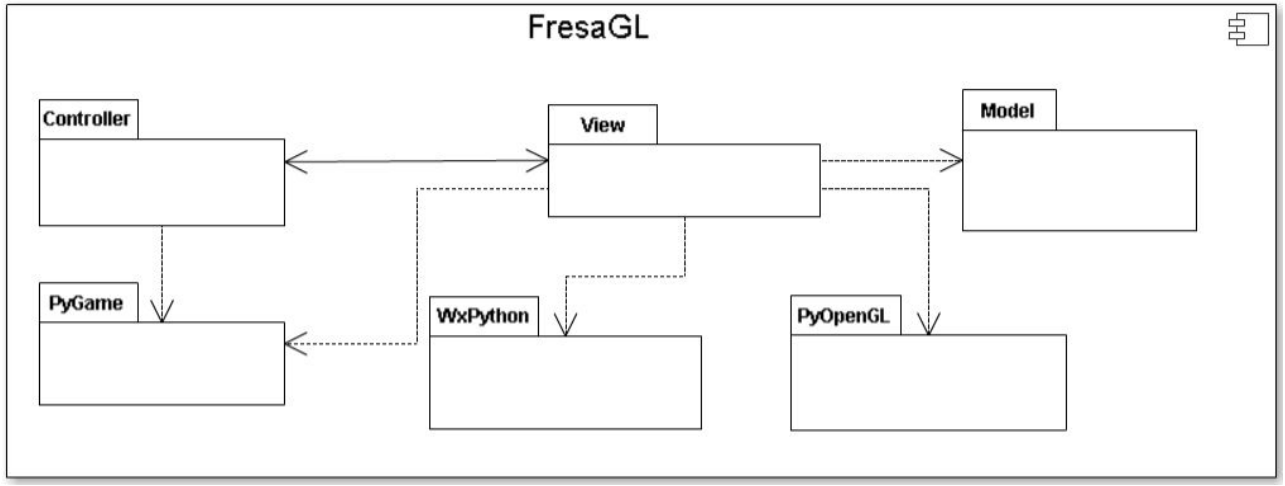
- Un buen diagrama es una obra de arte
  - Balanceados, destacan lo importante, elegantes
- Requieren un contexto
  - Breve parrafo explicativo
  - Título que sintetice un escenario
- Información relevante
  - Concentrados en una dimensión del problema
  - Deben ser complementarios
- Principio de Miller: 7+/- 2...
  - "... *Some Limits on Our Capacity for Processing Information*"
  - Trivial si posee menos de 5 entidades
  - Complejo si posee más de ~9



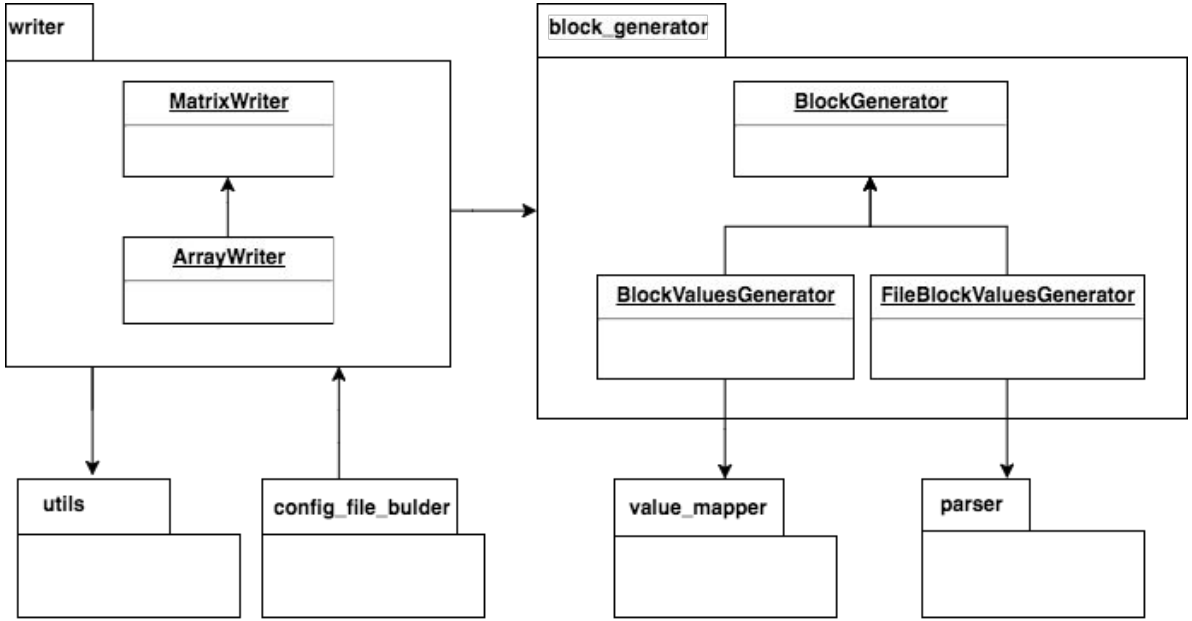
UML | Diagrama de Casos de Uso



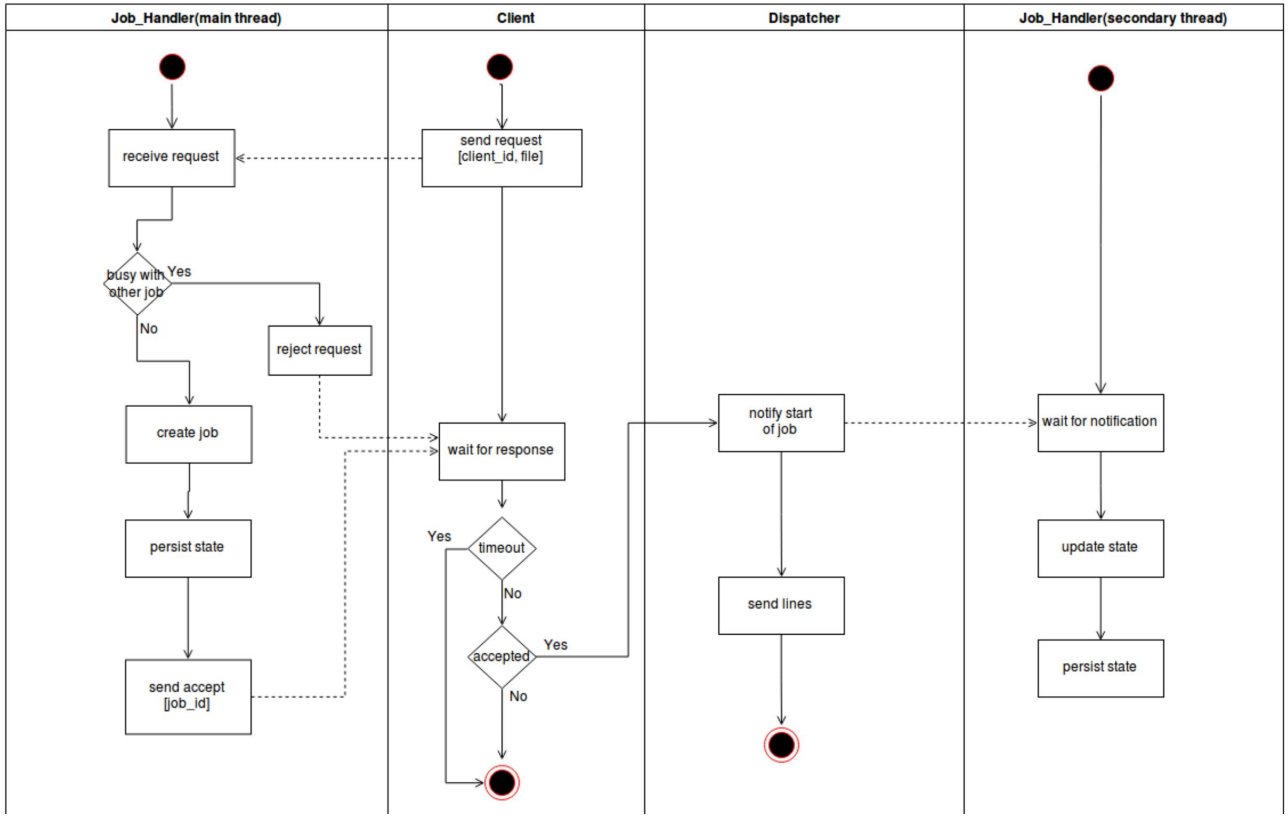
UML | Diagrama de Paquetes



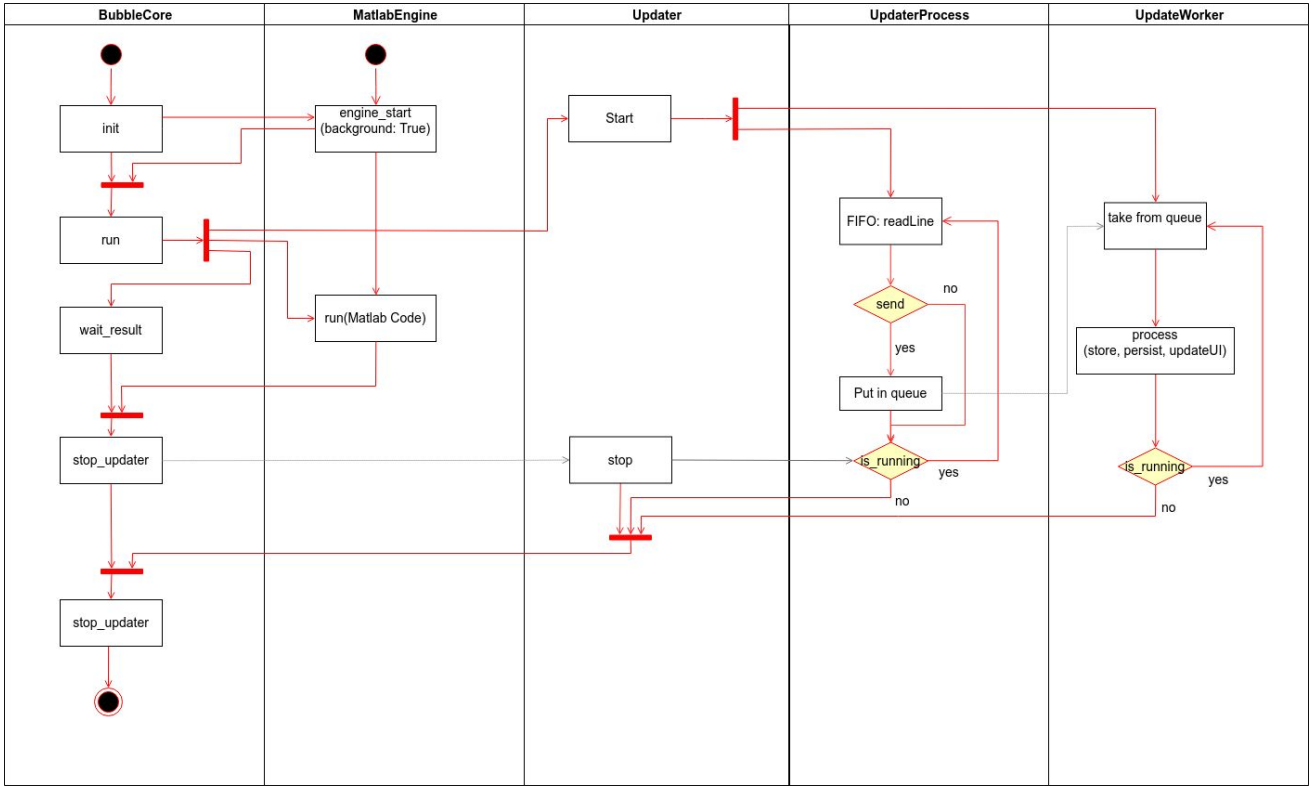
UML | Diagrama de Paquetes



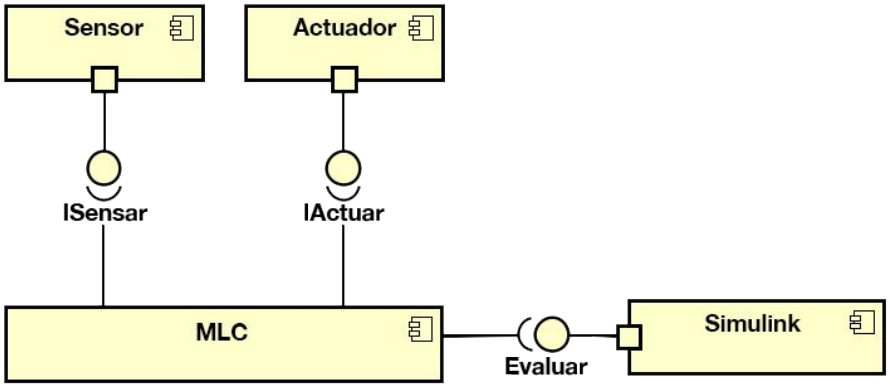
UML | Diagrama de Actividades



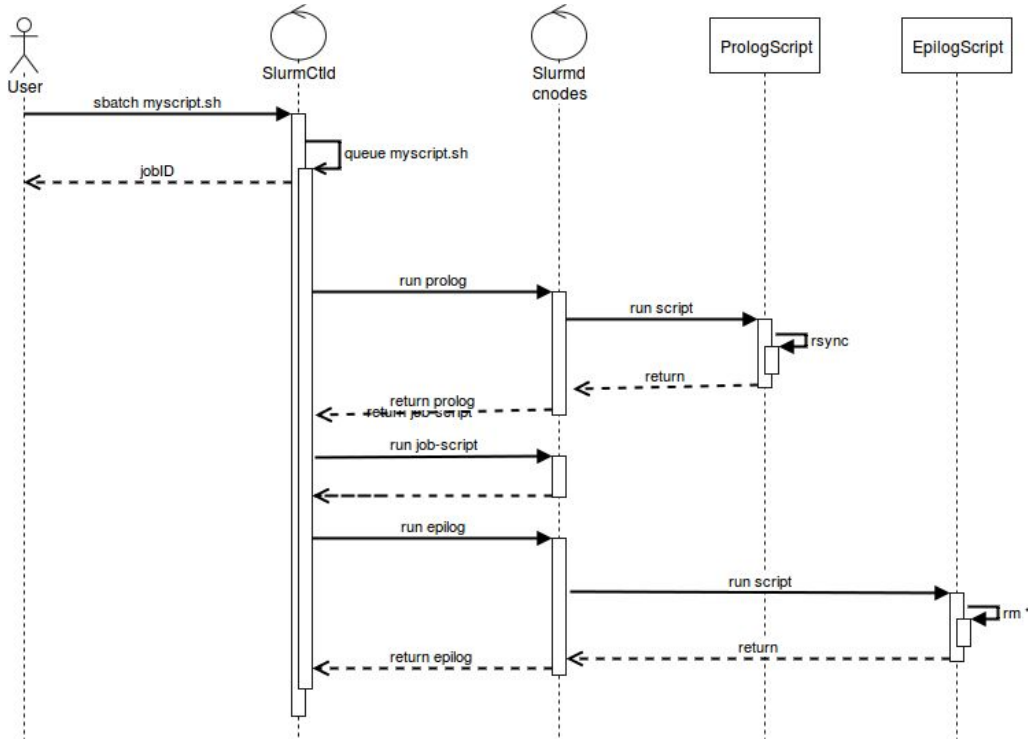
UML | Diagrama de Actividades (II)



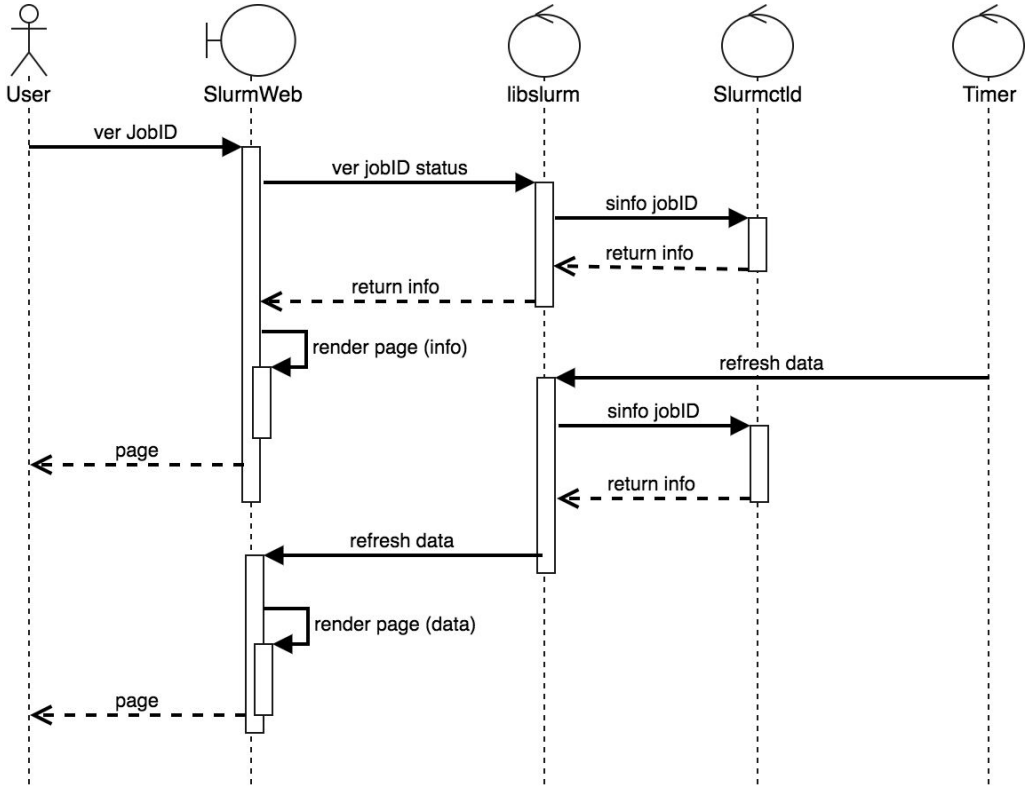
UML | Diagrama de Componentes



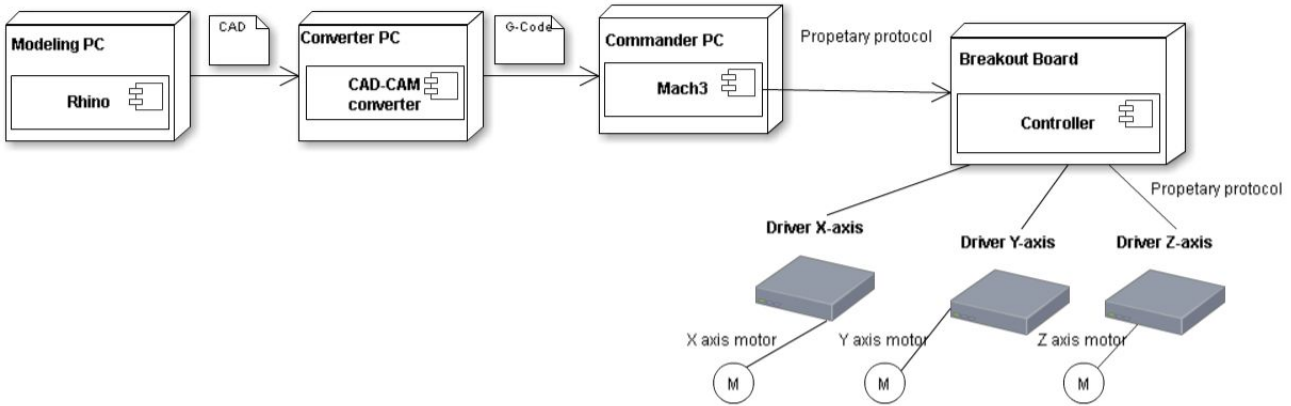
UML | Diagrama de Secuencia



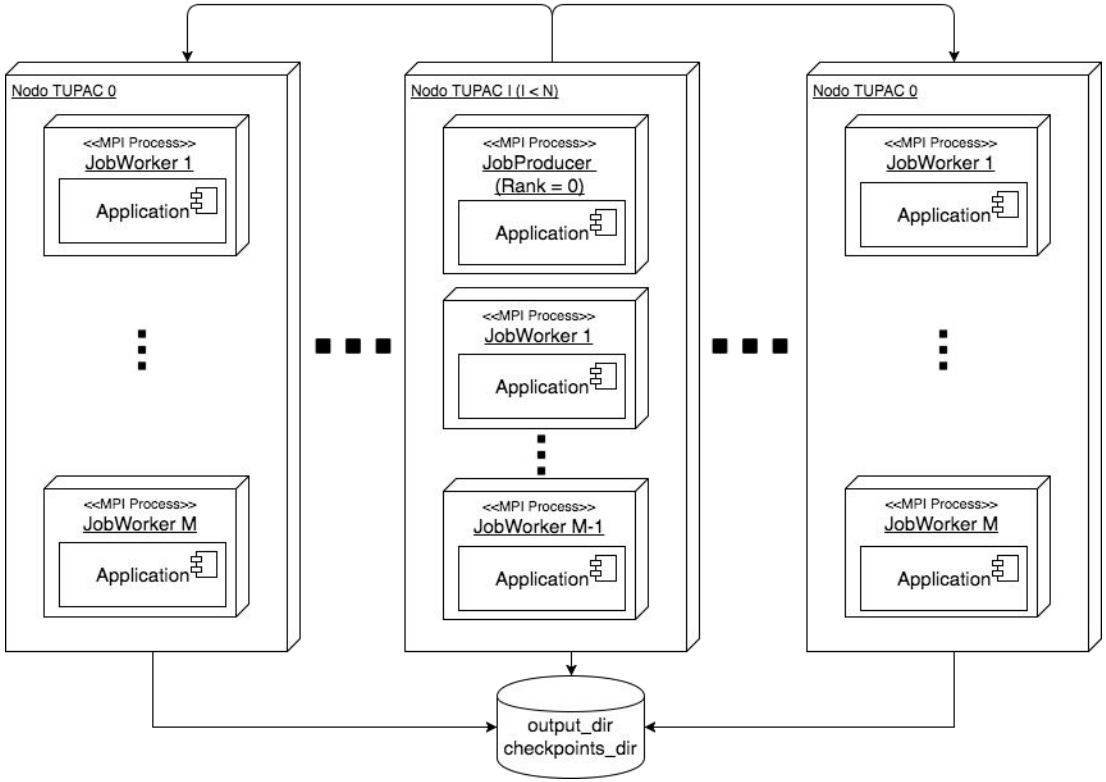
UML | Diagrama de Secuencia | Multiples hilos (?)



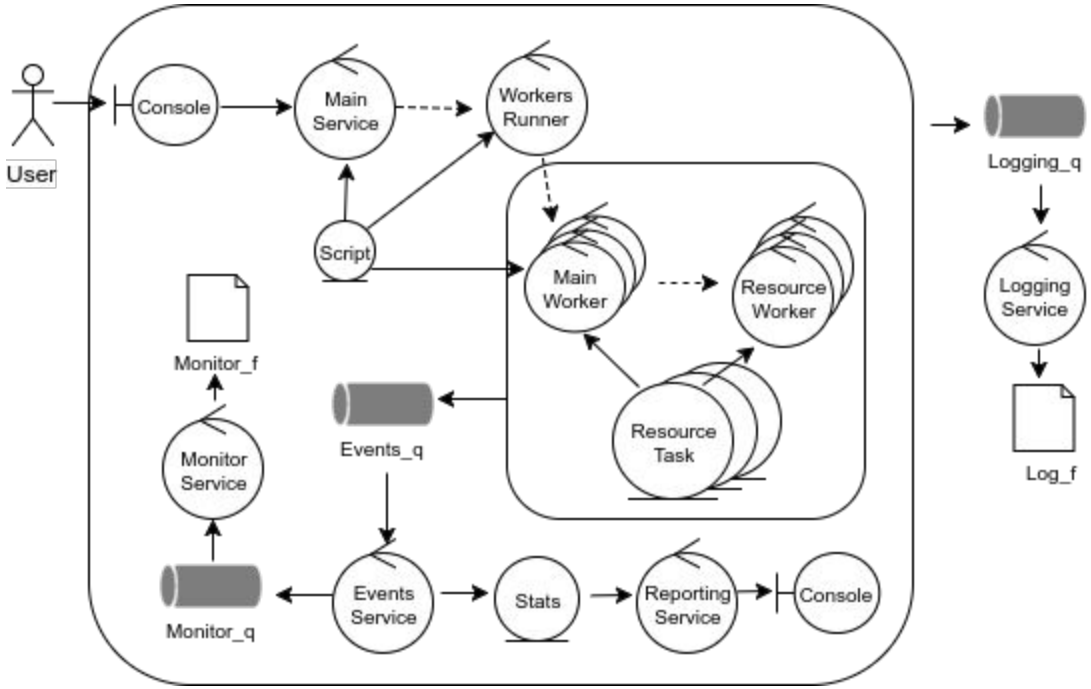
UML | Diagrama de Despliegue



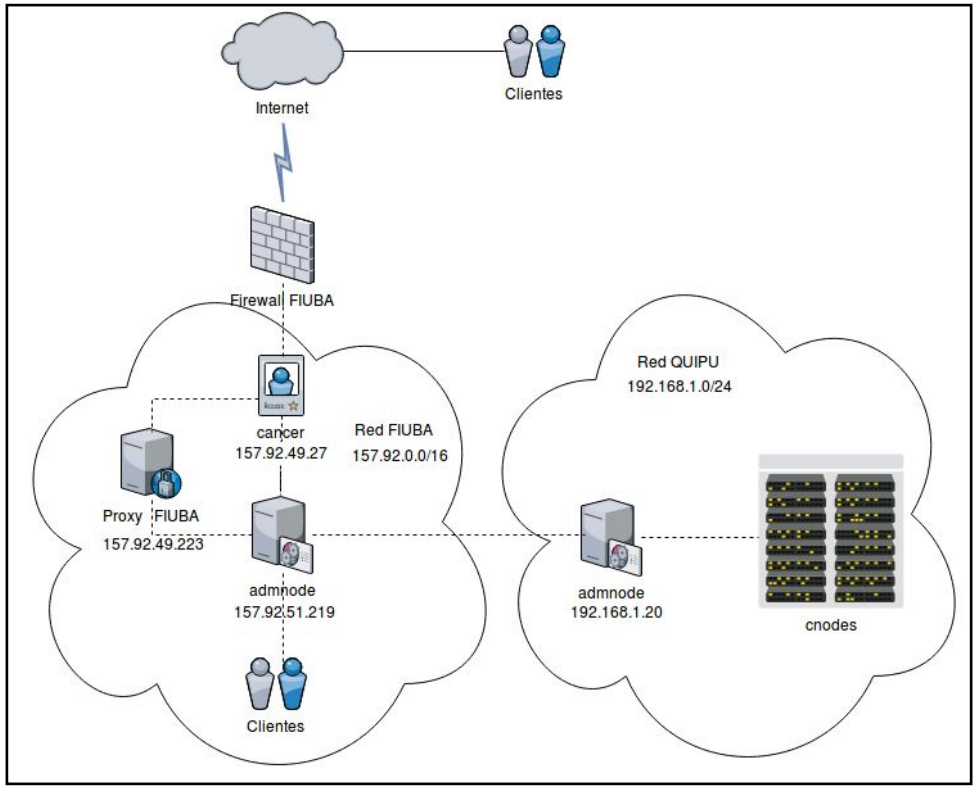
UML | Diagrama de Despliegue (II)



UML | Diagrama de Robustez



non-UML | Diagrama de Topología de Red



Bibliografía

- P. Bernstein. Middleware - An Architecture for Distributed System Services, Digital Equipment Corporation. Cambridge Research Lab. 1993.
- T. Bishop and R. Karne. "A Survey of Middleware." Computers and Their Applications. 2003.
- Gomaa, Hassan: Software Modeling & Design. UML, Use cases, pattern & software architectures. Cambridge, 2011.
- Fowler, UML Distilled, 1997.
- Rumbaugh, James, Jacobson, Ivar, Booch, Grady. The Unified Modeling Language Reference Manual, 1999.
- Philippe Kruchten, Architectural Blueprints - The “4+1” View Model of Software Architecture, 1995
- Simon Brown, The C4 model for software architecture
  - https://c4model.com/
- Simon Brown, Software Architecture for Developers
  - https://www.youtube.com/watch?time_continue=433&v=YmdTyFRBGxE