

# Ingeniería de Software en Servicios

---

Javier Béjar

ECSDI - 2023/2024 2Q

CS-GEI-FIB 



## Objetivos del tema



- ⊙ Ingeniería de Servicios SOA
- ⊙ Modelo de referencia SOA (OASIS)
- ⊙ Arquitectura de referencia SOA (OASIS)
- ⊙ Arquitectura de servicios web (W3C)

## Directrices para la IS en SOA

---

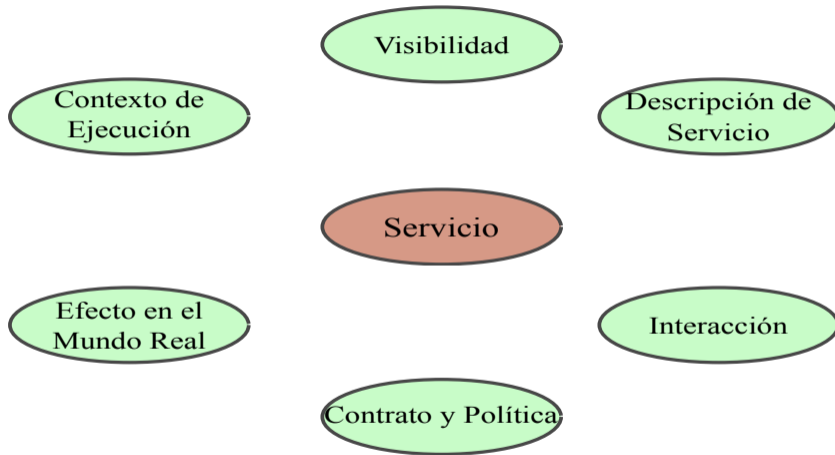
- ⊙ Hasta no hace mucho la Ingeniería de Software ha visto los sistemas computacionales como entidades monolíticas
  - El análisis ve el sistema completo como una caja negra
  - A partir de ahí se especifican las funcionalidades
  - A posteriori se pueden distribuir funcionalidades de acuerdo con requisitos funcionales o no funcionales
- ⊙ **Problema:** La distribución estructural y funcional se aborda demasiado tarde

- ⊙ Grupos de estandarización han definido directrices sobre el desarrollo de Arquitecturas Orientadas a Servicios de una manera abstracta
- ⊙ **OASIS**
  - Reference Model for Service Oriented Architecture 1.0 (10/2006)
  - Reference Architecture Foundation for Service Oriented Architecture 1.0 (12/2012)
- ⊙ **W3C**
  - W3C Web Services Architecture (2/2004)

# Modelo de referencia SOA (OASIS)

---

- ⊙ **Modelo abstracto** para el desarrollo de Arquitecturas Orientadas a Servicios
- ⊙ **Marco de referencia** para entender las relaciones entre los elementos de un entorno SOA
- ⊙ El modelo de referencia define SOA en un sentido abstracto
  - Por ejemplo, indica que se puede tener una descripción de servicio, pero no su realización concreta, por ejemplo usando WSDL (Web Service Description Language)





# Aquitectura de referencia SOA (OASIS)

---

- ⊙ Supone un paso de **refinamiento** sobre el reference model
- ⊙ Entra en los **detalles de los conceptos** de SOA-RM
- ⊙ Muestra como los sistemas SOA pueden ser **implementados**, planteado desde un **nivel abstracto** y de manera **independiente de las tecnologías** usadas
- ⊙ Da una perspectiva de SOA desde el punto de **sistemas dinámicos** en lugar de un sistema software estático

- ⊙ Uso de **recursos que están distribuidos** más allá del entorno cercano de los actores
- ⊙ **Personas y sistemas interaccionando** entre ellos también más allá de su entorno cercano
- ⊙ **Seguridad, manejo de recursos y gobernanza distribuidas**
- ⊙ Interacción a través de un **intercambio de mensajes** con fiabilidad y apropiado para las intenciones y usos establecidos

1. **Participación en un ecosistema SOA**, captura la visión como un entorno para personas que realizan actividades que involucran múltiples participantes
2. **Realización del ecosistema SOA**, captura los elementos de infraestructura que son necesarios para soportar la construcción de sistemas SOA
3. **Propiedad en un sistema SOA**, captura los elementos necesarios para definir la propiedad y la gestión del sistema, como se toman las decisiones, como se promulgan y como se hacen efectivas

# SOA y microservicios

---

- ⊙ SOA es complicada de llevar a la práctica
- ⊙ Muchos problemas no requieren la complejidad que representa
- ⊙ Las implementaciones de SOA son complicadas de desplegar y mantener
- ⊙ Las empresas líder en desarrollo de software no han convergido en una estandarización suficiente

- ⊙ Un microservicio resuelve una **tarea simple** (¡deja vu!)
- ⊙ Están al **mínimo nivel de abstracción**/granularidad (no son composición de otros)
- ⊙ Son **autónomos**, deben necesitar mínima coordinación para realizar su tarea (o ninguna)
- ⊙ No dependen de una infraestructura que los conecta (middleware)

- ⊙ Los microservicios son los elementos sobre los que se construyen servicios más complejos
- ⊙ **Necesitarán:** coordinación, comunicación, interoperabilidad, seguridad, contratos, regulaciones
- ⊙ Implementaciones de microservicios **utilizan:** sistemas de mensajes (comunicación), servicios de descubrimiento/coordinación, seguridad/criptación, ...



# Arquitectura de servicios web (W3C)

---

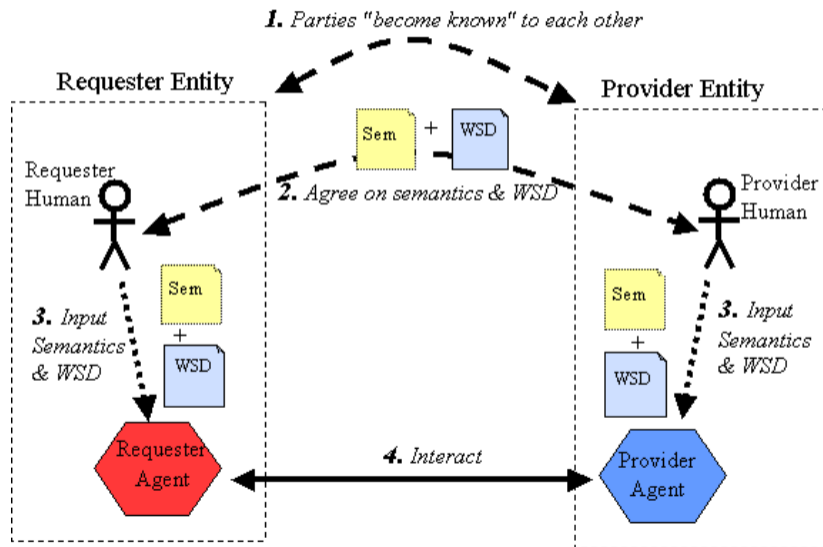
- ⊙ Los servicios web pueden verse como **un paso más en la concreción** del desarrollo
- ⊙ Un **servicio web** es una noción abstracta que se implementa a través de un **agente**
- ⊙ **Servicio web**: Recurso descrito por un conjunto de funcionalidades abstractas
- ⊙ **Agente**: código software/hardware concreto que envía y recibe mensajes
  - **Proveedor (provider)**: Organización que pone a disposición el agente que presta un servicio
  - **Solicitador (requester)**: Organización que desea usar un servicio y provee el agente que intercambia mensajes con el agente del proveedor

## ⊙ Descripción de servicio

- Los intercambios de mensajes entre servicios están documentados en una descripción de servicio
- Este documento define todo lo necesario para la interacción
- Esta descripción se hace utilizando WSDL
- Opcionalmente puede incluir una descripción funcional

## ⊙ Semántica

- La semántica está definida por las expectativas acerca del comportamiento del servicio
- Supone un contrato entre partes sobre el propósito y consecuencias de la interacción
- No se define la forma de su realización, pudiendo estar documentada o no, ser implícita o explícita, formal o informal, negociada o no negociada



- ⊙ SOA-RM, SOA-RAF y W3C-WSA solo describen los conceptos y elementos que aparecen en SOA
- ⊙ No dan ninguna guía sobre cómo distribuir funcionalidades
- ⊙ Las metodologías de ingeniería de software estándar no dan una guía de como desarrollar sistemas completamente distribuidos
- ⊙ ¿Dónde debemos mirar para obtener esas guías?