

# Introducción

---

Javier Béjar

ECSDI - 2023/2024 2Q

CS-GEI-FIB 



## Objetivos del tema



- ⦿ Motivación de la necesidad de sistemas distribuidos
- ⦿ Características de los sistemas distribuidos
- ⦿ Orientación a servicios (SOA/Microservicios)
- ⦿ Inteligencia artificial como metodología de desarrollo de Sistemas Distribuidos

Motivación



Los sistemas software han dejado de ser entornos **cerrados** y **monolíticos**

- ⦿ El elemento del sistema es el **servicio**
- ⦿ Múltiples componentes y sistemas **colaboran** para realizar tareas complejas



- ⊙ Distribución geográfica
- ⊙ Necesidad de comunicarse de maneras complejas
- ⊙ Necesidad de organizarse/colaborar/coordinarse
- ⊙ Interacciones dinámicas y cambiantes (colaboración con múltiples entidades)
- ⊙ Autonomía (toman sus propias decisiones)

- ⊙ **Next Generation Internet Initiative** (2017-2030)
- ⊙ Estudiar los **cambios** que producirá en la **sociedad**
- ⊙ Estudiar el **impacto** que tendrá en como las **empresas** desarrollarán su actividad
- ⊙ Estudiar las **oportunidades** económicas y de innovación
- ⊙ Estudiar las **necesidades** tecnológicas y de investigación a distintos niveles (software, hardware, comunicaciones)
- ⊙ Promover **estándares** para facilitar la adopción de las tecnologías

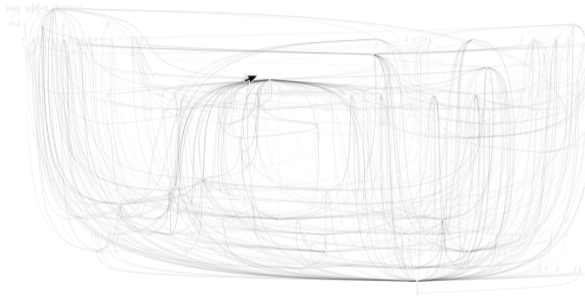
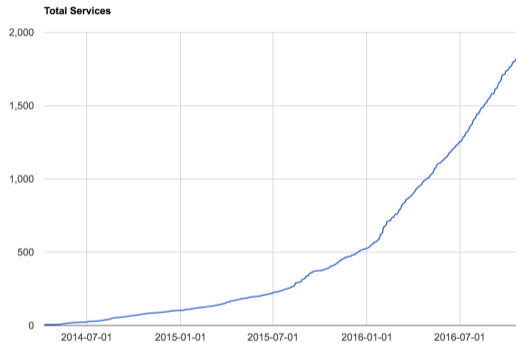
- ⊙ La **vida real y digital** estarán más **entrelazadas**
- ⊙ **Virtualización de las empresas**, transformando los roles de clientes y empleados
- ⊙ Nuevas herramientas y métodos para desarrollar y verificar sistemas interconectados y seguros
  - **Nuevas maneras de desarrollar software**



Mastering Chaos - A Netflix Guide to Microservices

<https://www.youtube.com/watch?v=CZ3wIuvmHeM>





What Comes after Microservices? (Uber)

<https://www.youtube.com/watch?v=UDC3kwbvka>

# Sistemas Distribuidos

---

- ⊙ Un **sistema distribuido** es un conjunto de sistemas computacionales que se comunican a través de una red
- ⊙ Los diferentes sistemas que los componen interactúan entre ellos para conseguir un objetivo (común)
- ⊙ La tarea a realizar no tiene por qué involucrar inteligencia o replicar mecanismos cognitivos complejos
- ⊙ La **inteligencia artificial distribuida** es la parte de la IA que aporta soluciones a la construcción de sistemas distribuidos complejos

- ⊙ Una forma de **repartir la carga** de trabajo entre un conjunto de computadores
- ⊙ Una manera de **ahorrar recursos**
- ⊙ Una posible manera de **mejorar la eficiencia** del hardware

- ⊙ Un sistema distribuido aparece como **una sola unidad** desde el exterior
- ⊙ Es **escalable**, se pueden añadir nuevos componentes y nuevas capacidades
- ⊙ Permiten **coordinar** sistemas que están separados físicamente
- ⊙ Permiten ser **tolerante a fallos** (los componentes pueden ser reemplazados)

- ⊙ **Heterogeneidad**, los componentes computaciones hardware y software pueden ser muy diferentes
- ⊙ Es difícil gestionar **fallos múltiples**
- ⊙ **Seguridad**, hay múltiples puntos de entrada
- ⊙ **Fiabilidad**, depende de la fiabilidad de la red de conexión
- ⊙ **Latencia**, la comunicación introduce retrasos y bloqueos
- ⊙ **Complejidad de diseño**

# Orientación a servicios - SOA/Microservicios

---

- ⊙ Perspectiva de desarrollo de sistemas distribuidos desde el punto de vista de la ingeniería de software
- ⊙ La **orientación a servicios** está pensada a la vez como diseño arquitectónico y metodología de desarrollo de software.
- ⊙ Basada en el concepto de **servicio** como elemento de diseño
- ⊙ **Servicio**: Componente software que provee sus funcionalidades a otras aplicaciones
- ⊙ Metodología **independiente de la tecnología** de implementación



- ⊙ Cada servicio está diseñado para hacer **una actividad**:

Servicio = Un fragmento de código

- ⊙ El **reuso** solo implica **cambiar cómo se interacciona** con otros servicios (vs cambiar el código del servicio)
- ⊙ En lugar de una API, un **servicio define una interfaz** en términos de protocolos y funcionalidad
- ⊙ Se puede ver SOA como una **evolución natural** de la computación distribuida y la programación modular

- ⊙ Un servicio es **auto descriptivo**
- ⊙ Implementa **una funcionalidad** bien definida
- ⊙ Usa el **encapsulamiento** de información como principio de diseño (caja negra)
- ⊙ Es **independientes de su plataforma** de implementación
- ⊙ Pensado para operar **en red**
- ⊙ Pensado para **facilitar la integración** de aplicaciones/organizaciones

- ⊙ **Acoplamiento débil**: Minimización de dependencias, conocimiento básico sobre la funcionalidad de los otros
- ⊙ **Abstracción**: Lógica de funcionamiento no visible más allá de entradas y salidas
- ⊙ **Contratos entre servicios**: Comunicación entre servicios descrita de manera global mediante un documento público
- ⊙ **Reusabilidad**: La funcionalidad está diseñada para poder formar parte de diferentes problemas

- ⊙ **Composabilidad**: El acoplamiento de servicios permite proveer servicios más complejos
- ⊙ **Autonomía**: El servicio tiene control sobre la funcionalidad que provee
- ⊙ **Sin estado**: El servicio guarda la mínima información para su funcionamiento (estado a nivel superior es responsabilidad de otros)
- ⊙ **Descubrimiento**: La descripción de los servicios provee la información que les permite ser descubiertos (qué hacen) e interpretados (cómo invocarlos)

## Una arquitectura orientada a servicios define

- ⊙ Un conjunto de buenas prácticas
- ⊙ Un conjunto de principios de diseño
- ⊙ Un conjunto de patrones de diseño

**Objetivo:** Guiar el diseño orientado a servicios para poder desarrollar software más flexible y complejo

(!) Microservicios vs SOA

# Inteligencia Artificial y Servicios

---

## Definición

*“Un campo de estudio que busca explicar y emular el comportamiento inteligente en términos de procesos computacionales” (Schalkoff, 1990)*

- ⊙ Adoptando **metáforas** de IA que acerquen el desarrollo **a como pensamos**
- ⊙ Acercando la **interacción** entre sistemas software **a como interactuamos**
- ⊙ Adoptando una **visión declarativa** en la resolución de problemas (qué hacer vs cómo hacerlo)
- ⊙ Permitiendo sistemas **software adaptativos** que puedan resolver requerimientos/situaciones/interacciones no previstas



La **Inteligencia Artificial Distribuida** (DAI) y los **Sistemas Multiagente** aporta soluciones a problemas complejos de los sistemas distribuidos, Puede utilizarse como marco general de desarrollo de software

- ⊙ Semántica de comportamiento
- ⊙ Descripción, organización y coordinación de componentes
- ⊙ Toma de decisiones en entornos heterogéneos y abiertos
- ⊙ Descubrimiento y composición dinámica
- ⊙ Recuperación de fallos

- ⊙ La complejidad de los sistemas distribuidos exige que tratemos los datos de manera más compleja y desde diferentes puntos de vista
- ⊙ La **representación del conocimiento** estudia los **formalismos para representar la información** en sistemas inteligentes
- ⊙ La representación del conocimiento permite describir:
  - La comunicación, interacción, objetivos y funcionamiento de los servicios
  - El contexto del servicio
  - El razonamiento sobre todos estos elementos

- ⊙ Los sistemas distribuidos se componen de elementos heterogéneos con cambios dinámicos en sus interacciones (no podemos preprogramar todo)
- ⊙ La toma de decisiones es compleja y requiere el uso de conocimiento declarativo
- ⊙ La inteligencia artificial provee de mecanismos que permiten tratar esas interacciones y la toma de decisiones
  - **Planificación Automática:** Generación de composiciones de tareas simples en tareas más complejas dado un objetivo, programación automática
  - **Razonamiento Automático:** Toma de decisiones dinámica a partir de conocimiento declarativo

- ⊙ Las aplicaciones evolucionan, las interacciones con otras entidades cambian, las interacciones requieren personalización/adaptación
- ⊙ No podemos programar para cambios que no podemos prever, ni programar todas las posibles variedades de interacciones existentes
- ⊙ **Aprendizaje Automático**: adaptación del comportamiento a diferentes niveles, por ejemplo:
  - Perfilado y preferencias de los servicios que interaccionan
  - Aprendizaje de resolución de tareas sin programación previa
  - Adaptación a los cambios en el contexto/entorno