



**SISTEMAS & ORGANIZACIONES**

**CLASE 7**

---

**Cátedra : SISTEMAS Y ORGANIZACIONES**

**TEORIA**

Docente : Prof. Mg. Sandra D'Agostino

Facultad de Informática

Universidad Nacional de La Plata

Año 2025



# **SISTEMAS & ORGANIZACIONES**

## **CLASE 7**

---

### **7.1- Evolución histórica de las aplicaciones**

**Aplicación Monolítica**

**Cliente-Servidor**

**Aplicación en 3 capas**

**Arquitectura en N capas**

**Arquitectura en N capas con objetos**

**Arquitectura en N capas, con objetos y sistemas heredados**

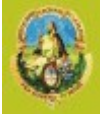
**Modelando Procesos de Negocio (BPM)**

### **7.2- Introducción a SOA**

### **7.3- Microservicios – Kubernetes (K8s) – Dockers – DevSecOps**

### **7.4- Ejemplos**

### **7.5- Conclusiones**



# **SISTEMAS & ORGANIZACIONES**

## **CLASE 7**

---

### **7.1- . Evolución histórica de las aplicaciones**



# SISTEMAS & ORGANIZACIONES

## CLASE 7

---

### ➤ **Aplicación monolítica**

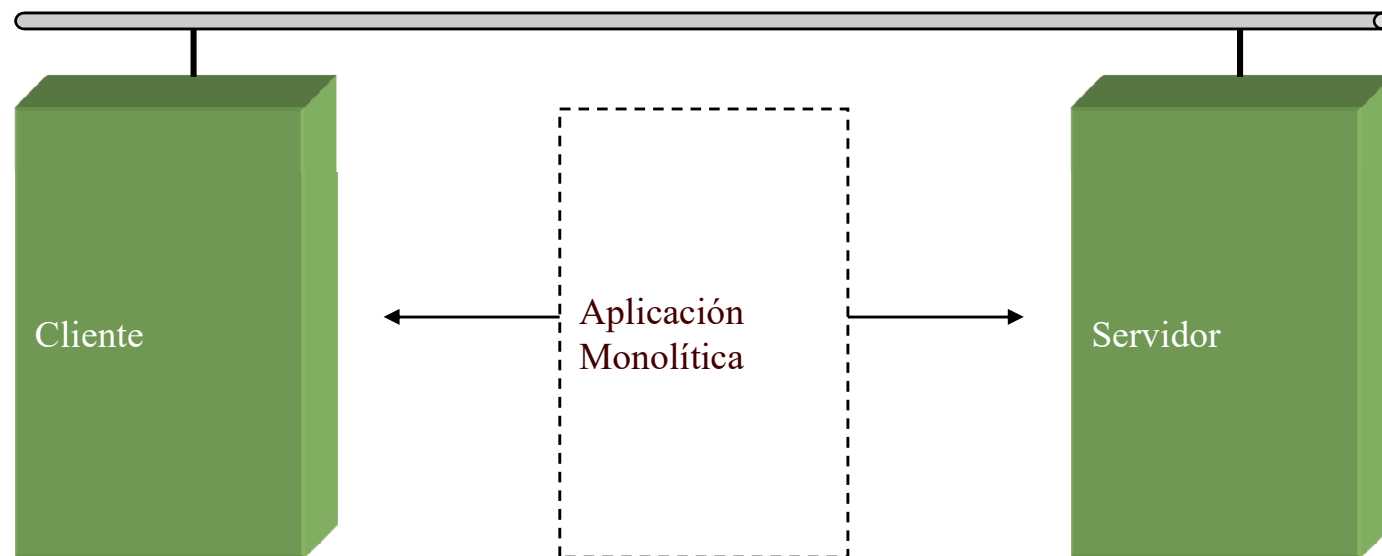




## CLASE 7

---

### ➤ Arquitectura Cliente-Servidor

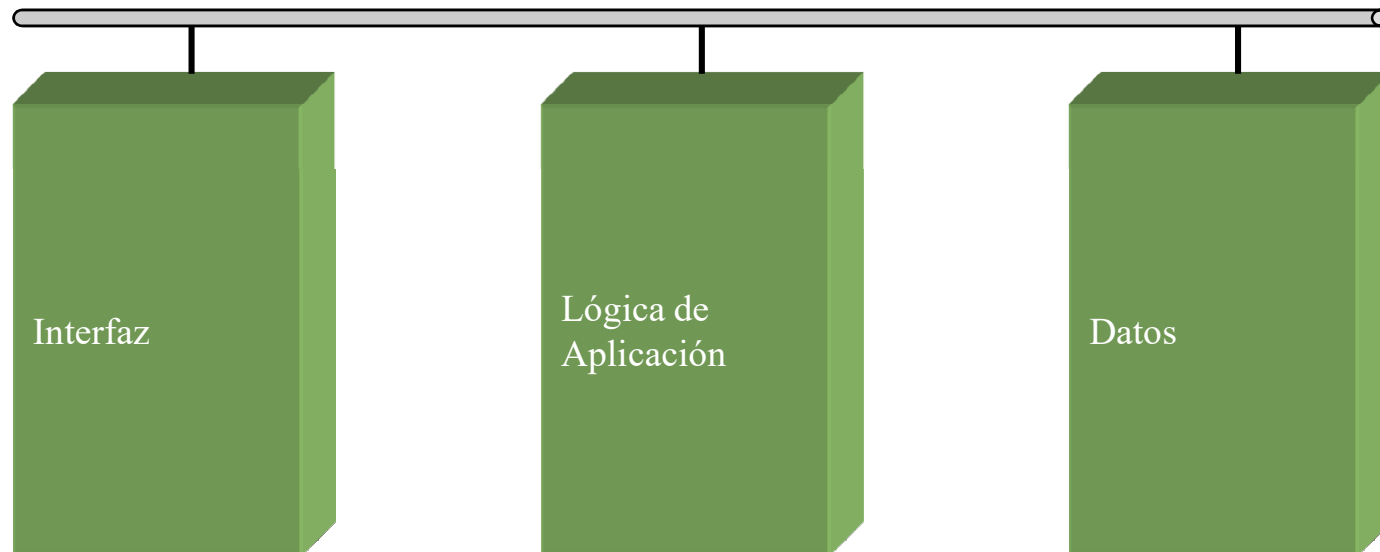




# SISTEMAS & ORGANIZACIONES

## CLASE 7

- Arquitectura en 3 capas





# SISTEMAS & ORGANIZACIONES

## CLASE 7

### ➤ Arquitectura en N capas

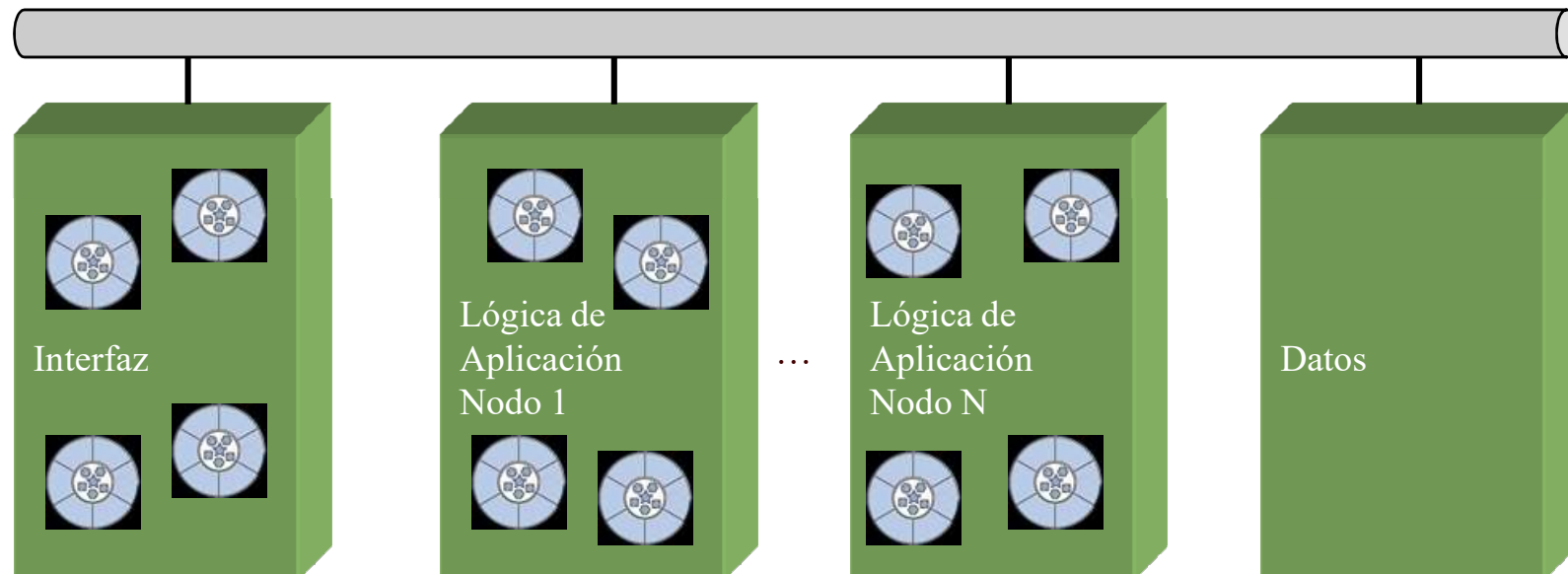




# SISTEMAS & ORGANIZACIONES

## CLASE 7

- Arquitectura en N capas con objetos



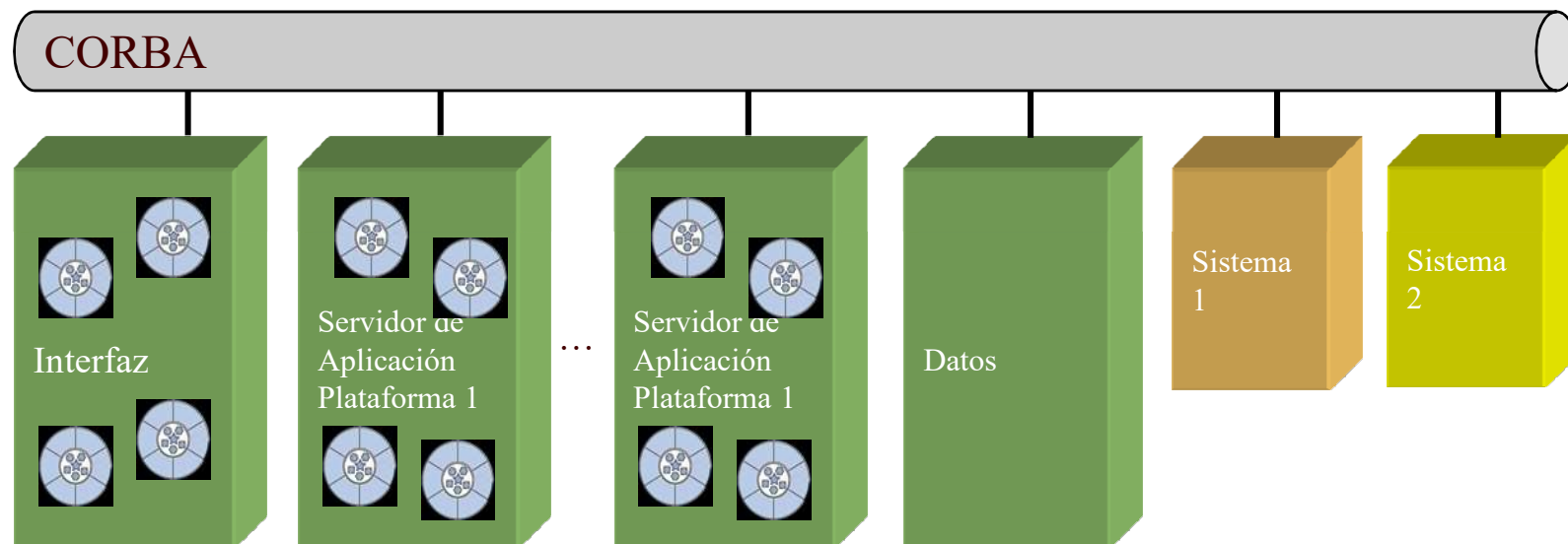




# SISTEMAS & ORGANIZACIONES

## CLASE 7

- Arquitectura en N capas con objetos y Sistemas Heredados

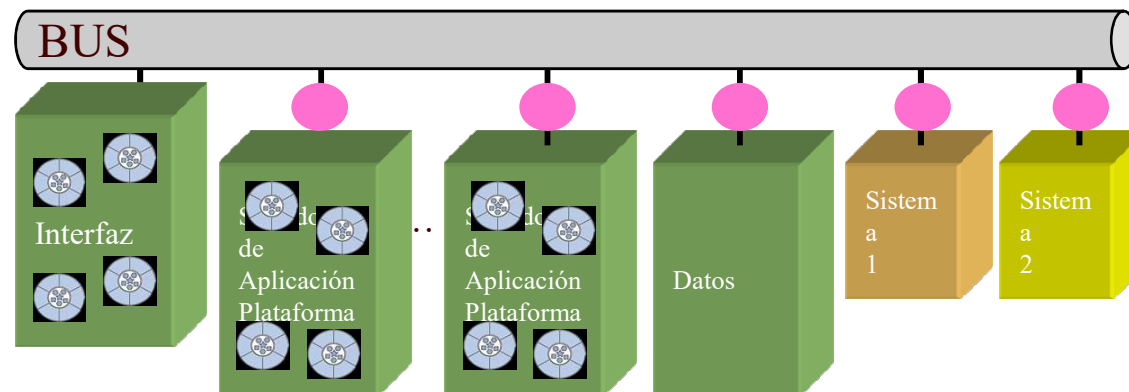
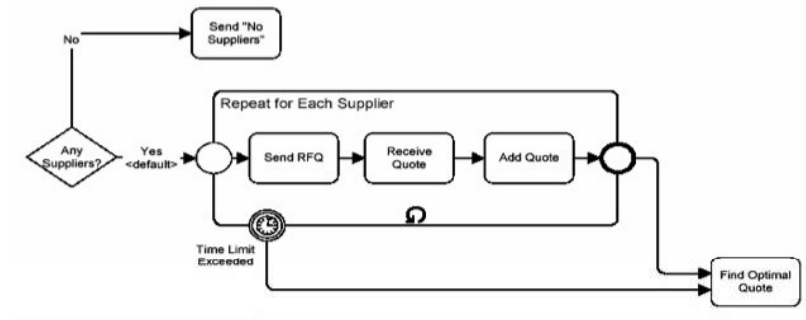




# SISTEMAS & ORGANIZACIONES

## CLASE 7

### ➤ Modelando Procesos de Negocio (BPM)

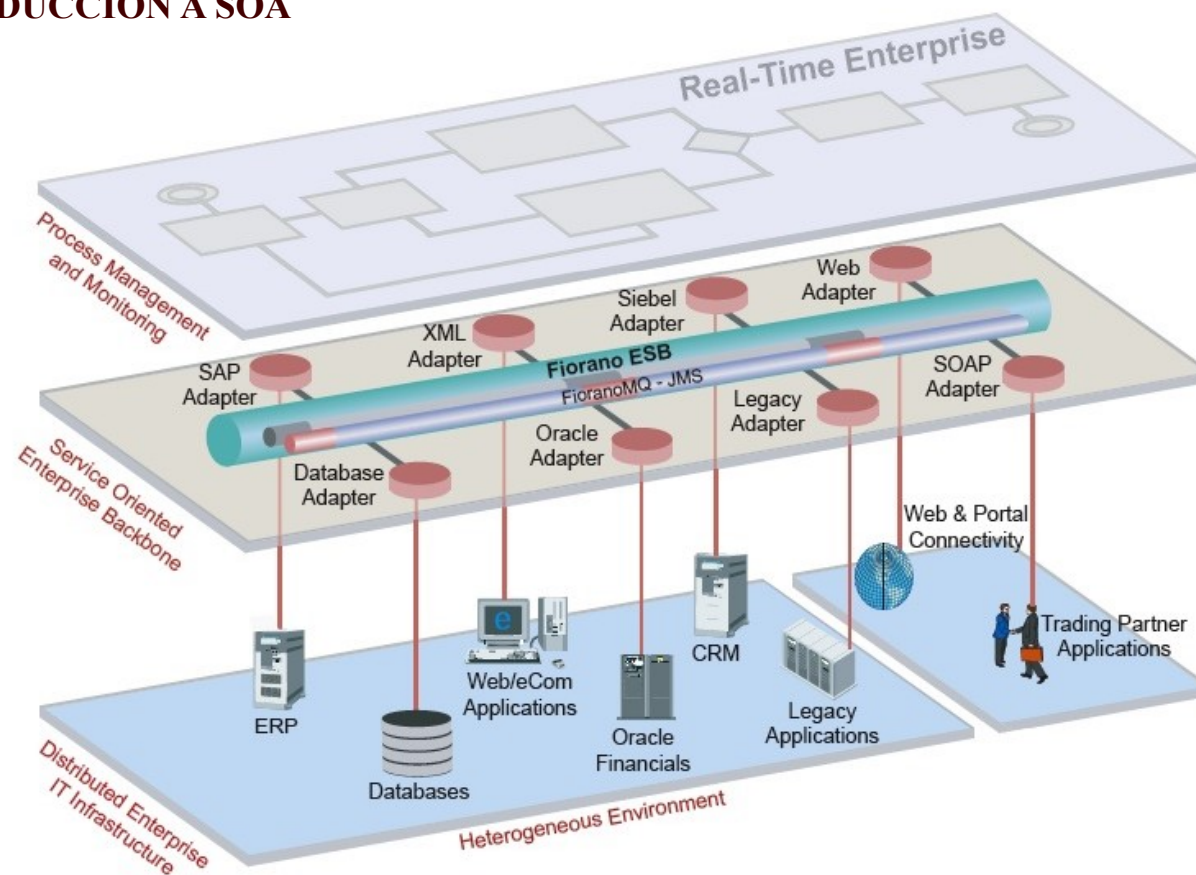




# SISTEMAS & ORGANIZACIONES

## CLASE 7

### 7.2- – INTRODUCCIÓN A SOA





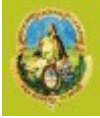
## CLASE 7

---

### 7.2 – INTRODUCCIÓN A SOA

#### ¿Qué no es SOA?

- SOA no es:
  - una tecnología
    - J2EE, .NET, Web Services
  - un producto
    - ESB's, SOA Fabrics
  - un protocolo
    - SOAP, HTTP, etc.
  - un estándar
    - Es un modelo de referencia.
  - una solución
    - SOA no es ejecutable y, por lo tanto, no produce resultados.



## CLASE 7

---

### 7.2 – INTRODUCCIÓN A SOA

#### ¿Qué es SOA?

**La Arquitectura Orientada a Servicios (SOA)** es un modelo de diseño de software que organiza las funcionalidades de un sistema en **servicios independientes, reutilizables y con interfaces bien definidas**, que pueden comunicarse entre sí a través de protocolos estándar.

Su objetivo es lograr **interoperabilidad, flexibilidad y alineación entre los procesos de negocio y la infraestructura tecnológica**.



## CLASE 7

---

### 7.2 – INTRODUCCIÓN A SOA

#### ¿Qué es SOA?

- ❑ La Arquitectura Orientada a Servicios (SOA) es un paradigma de realización y mantenimiento de procesos de negocio que abarcan grandes sistemas distribuidos.
- ❑ Está basada en tres conceptos técnicos principales: servicios, interoperabilidad a través a de un bus de servicios de la empresa (ESB) y un acople débil.
  - Un *servicio* es una pieza de funcionalidad de negocio autocontenida. Puede ser simple (almacenar o recuperar datos de cliente) o complejos (un proceso de negocio para un pedido de cliente). Puesto que los servicios se concentran en el valor de negocio de una interfaz, pueden puentear el salto entre el negocio e IT.



# SISTEMAS & ORGANIZACIONES

## CLASE 7

---

### 7.2 – INTRODUCCIÓN A SOA

#### ¿Qué es SOA?

- Un *bus de servicio de la empresa (ESB)* es una infraestructura que habilita una alta interoperabilidad entre servicios de diversos sistemas distribuidos. Facilita la distribución de procesos de negocio entre múltiples sistemas usando plataformas y tecnologías diferentes.
- El **acople débil** es el concepto de reducir las dependencias del sistema. Puesto que los procesos de negocio están distribuidos entre múltiples backends, es importante **minimizar los efectos de modificaciones y fallos**. De otra forma, las modificaciones se vuelven muy arriesgadas y los fallos del sistema pueden derrumbar el completo andamiaje del sistema. Obsérvese, sin embargo, que existe un precio para el acople débil: la complejidad. Los sistemas distribuidos débilmente acoplados son más difíciles de desarrollar, mantener y depurar.



# SISTEMAS & ORGANIZACIONES

## CLASE 7

---

### 7.2 – INTRODUCCIÓN A SOA

#### Servicios Web y SOA

- Los servicios web son una forma técnica de implementar **SOA**.
- SOA es un **modelo de organización** de servicios, no solo una tecnología.

#### Limitaciones:

- No resuelven todos los problemas técnicos.
- Requieren **tiempo y recursos** adicionales para su correcta implementación.

#### Buenas prácticas:

- Evitar crear servicios **demasiado específicos** o difíciles de reutilizar.
- Usar servicios web solo cuando aporten **valor real** a la infraestructura.





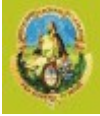
## SISTEMAS & ORGANIZACIONES

### CLASE 7

---

#### SOA en la práctica

- *En teoría, la teoría y la práctica son la misma cosa. En la práctica, no lo son.* -L. Berra
- Por supuesto, esto también **se aplica a SOA**. Los casos generales de negocio y los conceptos podrían no funcionar tan bien como se espera cuando intervienen factores tales como rendimiento y seguridad.
- Además, el hecho de que SOA es una estrategia para los sistemas existentes bajo mantenimiento **conduce a aspectos de estabilidad y compatibilidad hacia atrás**.



# SISTEMAS & ORGANIZACIONES

## CLASE 7

---

### 7.2 – INTRODUCCIÓN A SOA

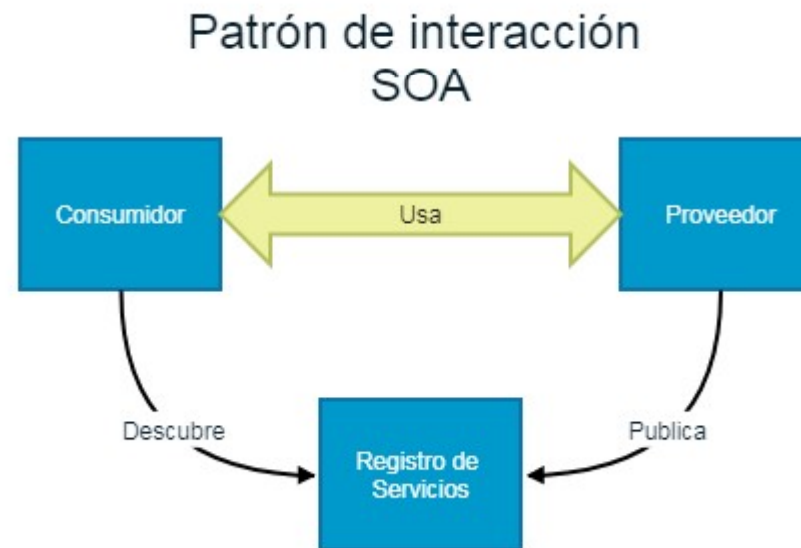
#### SOA en la práctica

- Y, en IT, **cada sistema es diferente**. En consecuencia, tendremos que **construir nuestro SOA específico**, no lo podemos comprar. Para elaborarlo, necesitaremos tiempo y un enfoque iterativo e incremental.
- Obsérvese, además, que la introducción de SOA no es lo importante. Lo verdaderamente importante es que la solución IT que introducimos sea la apropiada para nuestro contexto y requisitos.



## CLASE 7

### 7.2 – INTRODUCCIÓN A SOA



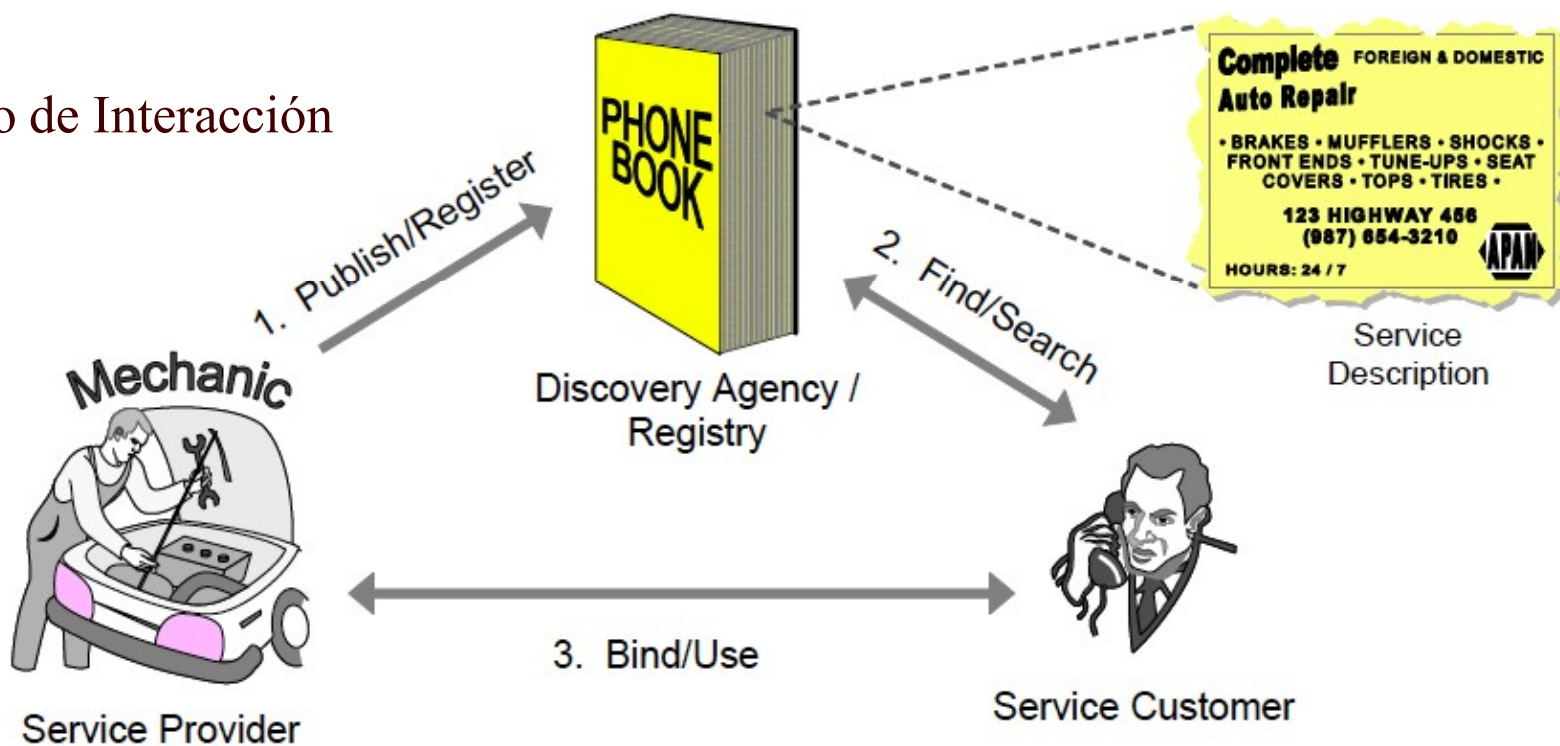


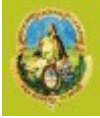
# SISTEMAS & ORGANIZACIONES

## CLASE 7

### 7.2 – INTRODUCCIÓN A SOA

#### Modelo de Interacción





## CLASE 7

---

### 7.2 – INTRODUCCIÓN A SOA

#### OBJETIVO DESDE EL PUNTO DE VISTA TECNOLÓGICO

##### ✿ Arquitectura SOA (Service-Oriented Architecture)

##### Concepto general:

- Enfoca las aplicaciones como servicios que integran datos y procesos de negocio en tiempo real.

##### Tecnologías utilizadas:

- Basada en estándares como **XML**, **SOAP**, **WSDL** y **BPEL**.
- Facilitan la comunicación y cooperación entre aplicaciones.

##### Ventajas principales:

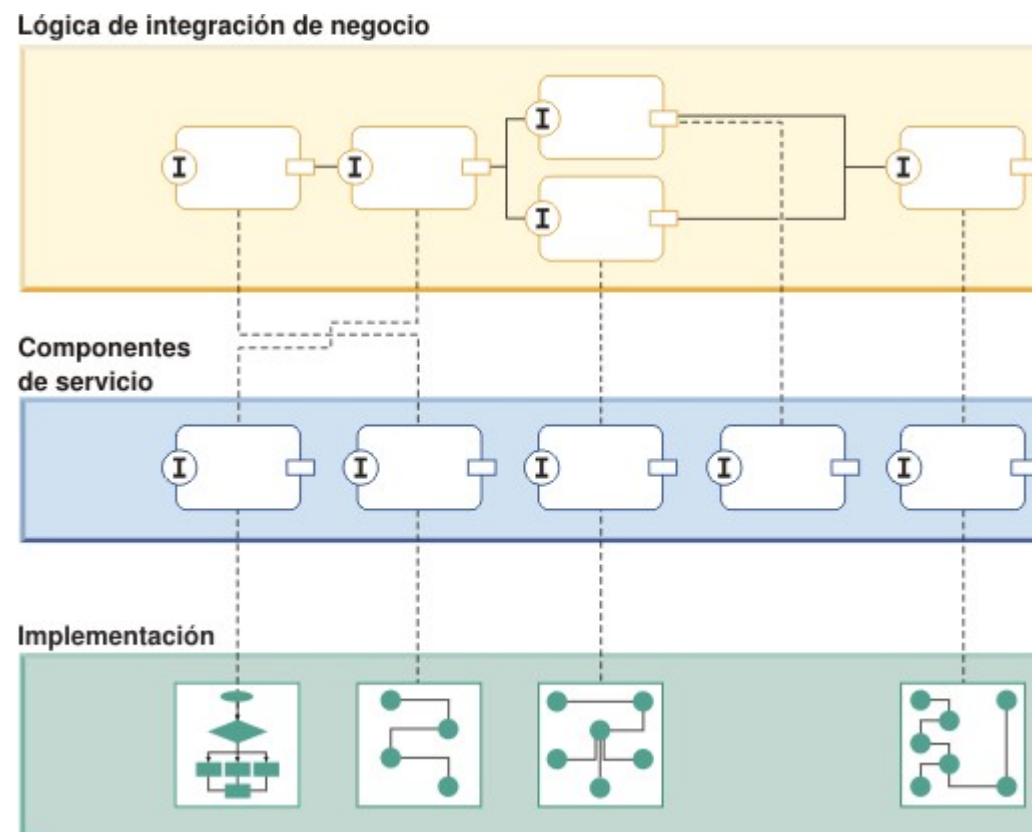
- Conecta aplicaciones heterogéneas.
- Aumenta la **flexibilidad** y la **interoperabilidad**.
- Permite interacción entre organizaciones sin conexiones permanentes.
- Disminuye el **tiempo de aprendizaje** por uso de tecnologías estándar.



## CLASE 7

### 7.2 – INTRODUCCIÓN A SOA

#### OBJETIVO DESDE EL PUNTO DE VISTA TECNOLÓGICO



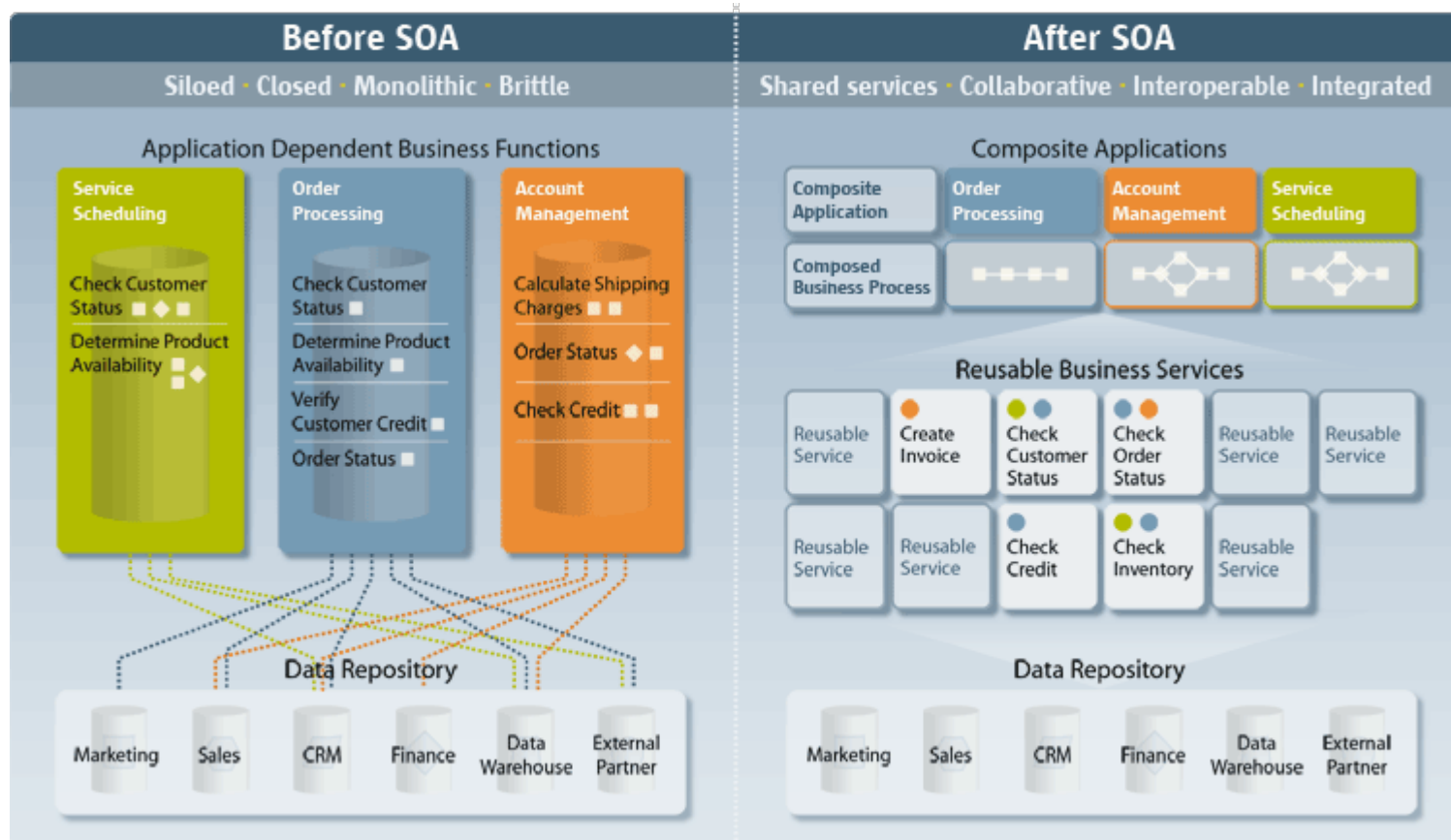


# SISTEMAS & ORGANIZACIONES

## CLASE 7

### 7.2 – INTRODUCCIÓN A SOA

#### OBJETIVO DESDE EL PUNTO DE VISTA TECNOLÓGICO





## SISTEMAS & ORGANIZACIONES

### CLASE 7

---

#### 7.2 – INTRODUCCIÓN A SOA

#### OBJETIVO DESDE EL PUNTO DE VISTA TECNOLÓGICO

La arquitectura orientada a servicios (SOA) **requiere un cambio cultural.**

SOA tendrá un impacto en muchos aspectos, desde la estrategia de negocio, los procesos de desarrollo de software, la infraestructura, el gobierno, la estructura de la organización y los roles hasta la colaboración en equipo

Para implementar de manera exitosa SOA en una organización, es importante establecer un roadmap para establecer y administrar expectativas, lograr ganancias a corto plazo, demostrar beneficios a largo plazo y mantener la iniciativa SOA encaminada.



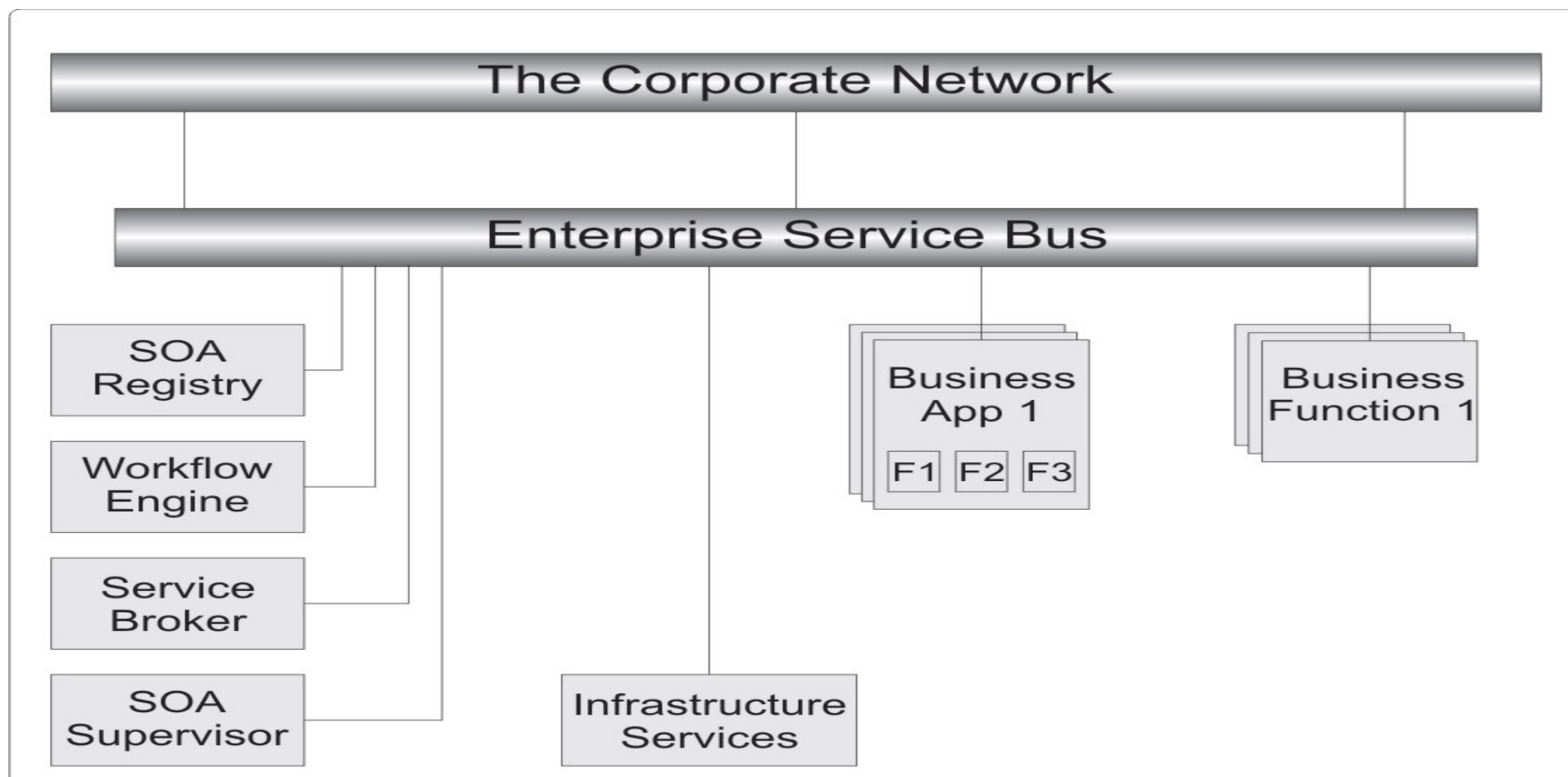


# SISTEMAS & ORGANIZACIONES

## CLASE 7

### 7.2 – INTRODUCCIÓN A SOA

#### INFRAESTRUCTURA PARA SOPORTAR SOA





## **SISTEMAS & ORGANIZACIONES**

### **CLASE 7**

---

#### **7.2 – INTRODUCCIÓN A SOA**

##### **INFRAESTRUCTURA PARA SOPORTAR SOA**

El Bus de servicios “ESB” (Enterprise Service Bus), asegura el intercambio de mensajes entre los componentes de los sistemas SOA

El Registro SOA (SOA registry) contiene información relativa a los servicios específicos de los sistemas SOA, y a su localización.

El Intermediario de Servicios (Service Broker), conecta entre sí los servicios requeridos por los procesos de negocio.

El Motor de procesos (workflow engine), se encarga de la orquestación completa de un proceso de negocio, incluyendo la participación de tareas humanas y provistas por servicios.



# **SISTEMAS & ORGANIZACIONES**

## **CLASE 7**

---

### **7.2 – INTRODUCCIÓN A SOA**

#### **INFRAESTRUCTURA PARA SOPORTAR SOA**

El Supervisor SOA (SOA supervisor), monitorea la ejecución de los procesos de negocio, controlando que se cumplan los niveles de servicio acordados (SLA).

Actualmente, el Bus de Servicios es uno de los elementos que más está evolucionando, con diversas implantaciones propuestas por los distintos proveedores de tecnología SOA. Tiende a convertirse en el elemento central, y a asumir varias de las funciones de los demás elementos, (además de sus funciones básicas de intercambio y transformación de mensajes).



# SISTEMAS & ORGANIZACIONES

## CLASE 7

---

### RETOS POR RESOLVER EN LOS SISTEMAS SOA

- 1- **Seguridad** : ¿Cómo asegurar el manejo de la identidad de un usuario, la autenticación y la autorización, para invocar servicios provistos por múltiples aplicaciones, cada una con sus propias políticas de seguridad?
- 2- **Gestión de datos de Distintas Bases de Datos:** para asegurar su integración y consistencia, en el contexto de los procesos de negocio. Para lograrlo, se requiere construir un repositorio de metadatos con las reglas de extracción y transformación de los datos, para que puedan ser utilizados en los procesos de negocio
- 3- **Auditoria:** Control de Transacciones
- 4- **Cultura Organizacional:** Es un nuevo paradigma donde el cambio organizacional, sobretodo en las áreas de sistemas debe ser adoptado.



# SISTEMAS & ORGANIZACIONES

## CLASE 7

---

### 7.2 – INTRODUCCIÓN A SOA

#### ¿Qué hace diferente a SOA?

- No hay elementos radicalmente nuevos:
  - ✓ La orientación a servicios se ha aplicado con éxito durante años
  - ✓ Los servicios se aplican a entornos concretos creando sistemas aislados
- Las diferencias se encuentran en:
  - ✓ La forma de comunicarse
  - ✓ Los servicios dejan de utilizarse de forma aislada
  - ✓ La utilización de estándares es parte esencial del éxito
  - ✓ Se aplican las técnicas exitosas basadas en experiencias pasadas



# SISTEMAS & ORGANIZACIONES

## CLASE 7

---

### 7.2 – INTRODUCCIÓN A SOA

#### Beneficios para implementar SOA

- Es **adaptable**
  - Se adapta más fácilmente a los cambios del negocio.
  - Facilita la integración de sistemas.
  - Es independiente de la plataforma y lenguajes de programación.
- Es **ágil**
  - Resulta más fácil añadir nueva funcionalidad y hacerla llegar a los canales, ya sean nuevos o existentes.
  - Facilita la creación de nuevos servicios haciendo uso de elementos existentes (propios o ajenos).
- Puede **ahorrar costes**
  - Facilita la reutilización de inversiones realizadas en SI.
  - Facilita la reducción de los costes de desarrollo.

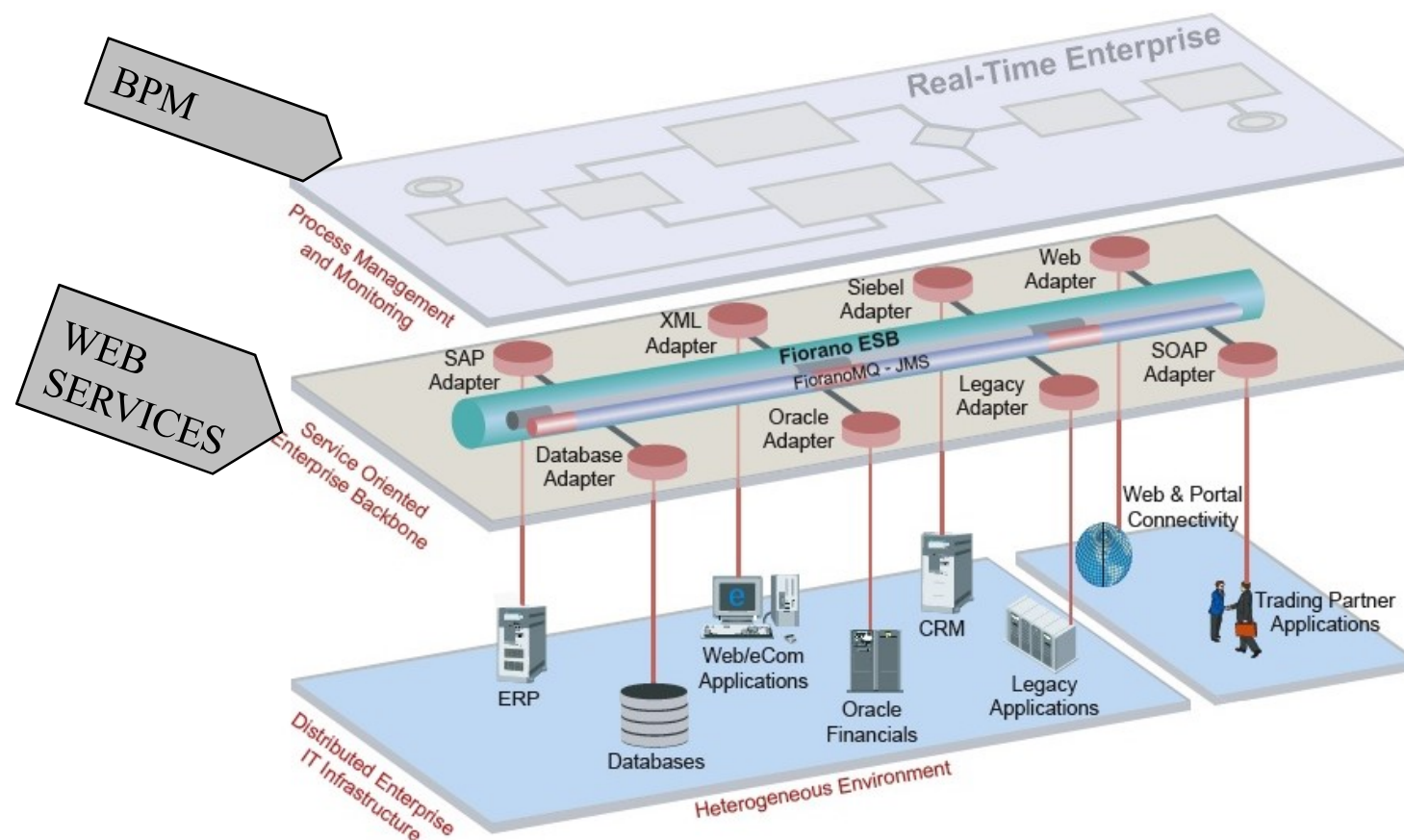


# SISTEMAS & ORGANIZACIONES

## CLASE 7

### 7.2 – INTRODUCCIÓN A SOA

#### RESUMIENDO

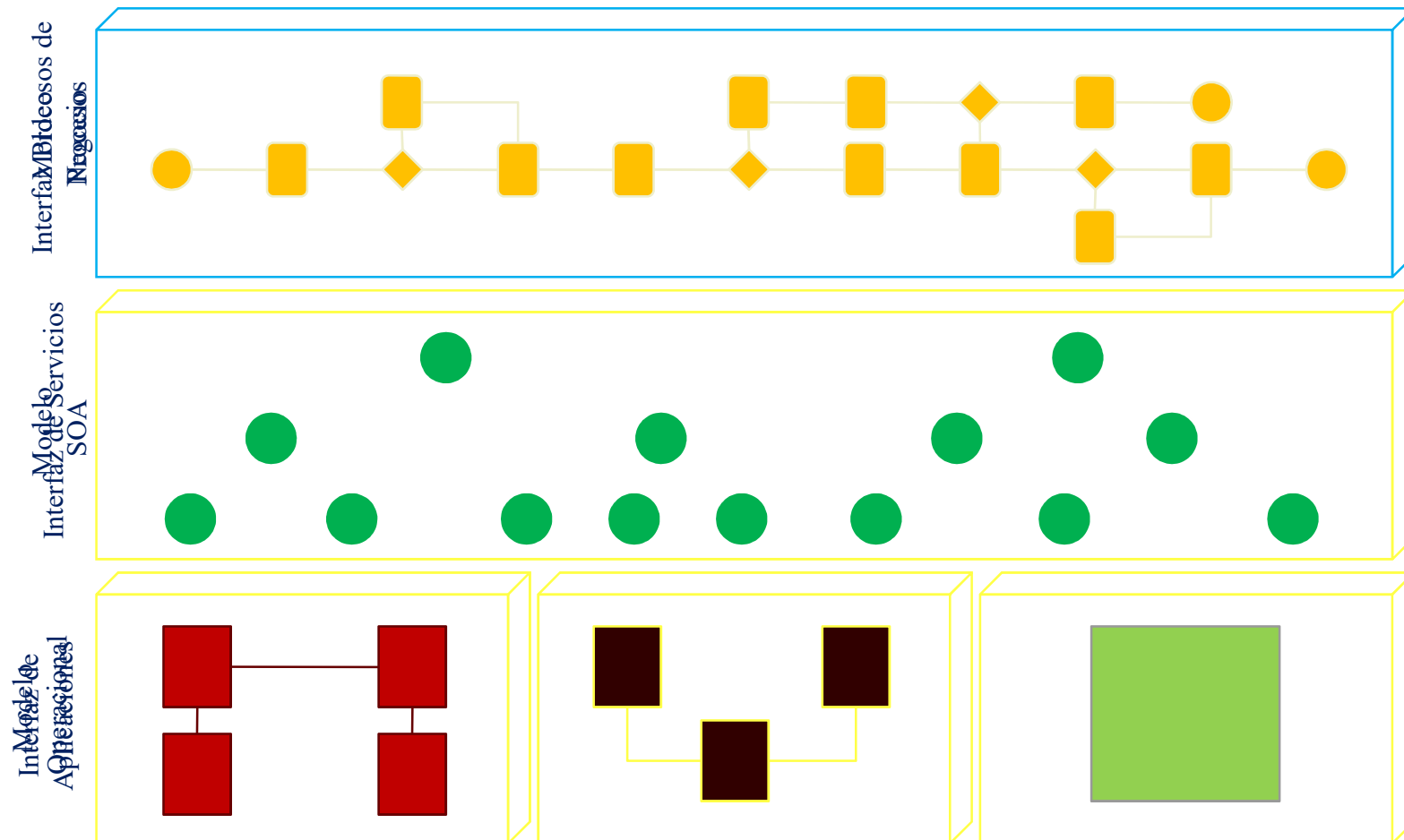




# SISTEMAS & ORGANIZACIONES

## CLASE 7

### Modelo de implementación BPM/SOA - Identificación



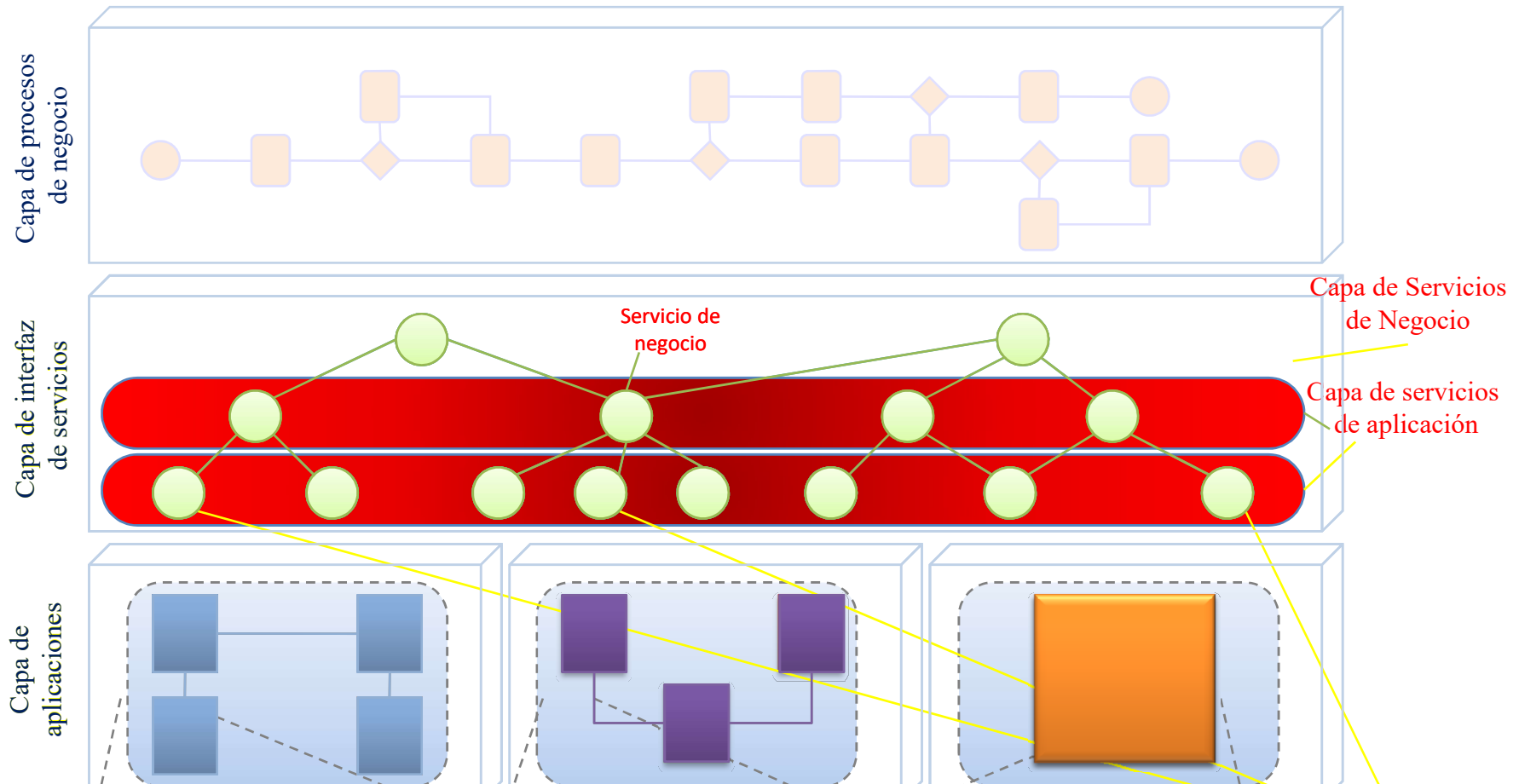




# SISTEMAS & ORGANIZACIONES

## CLASE 7

### Modelo de implementación BPM/SOA - Secuencia

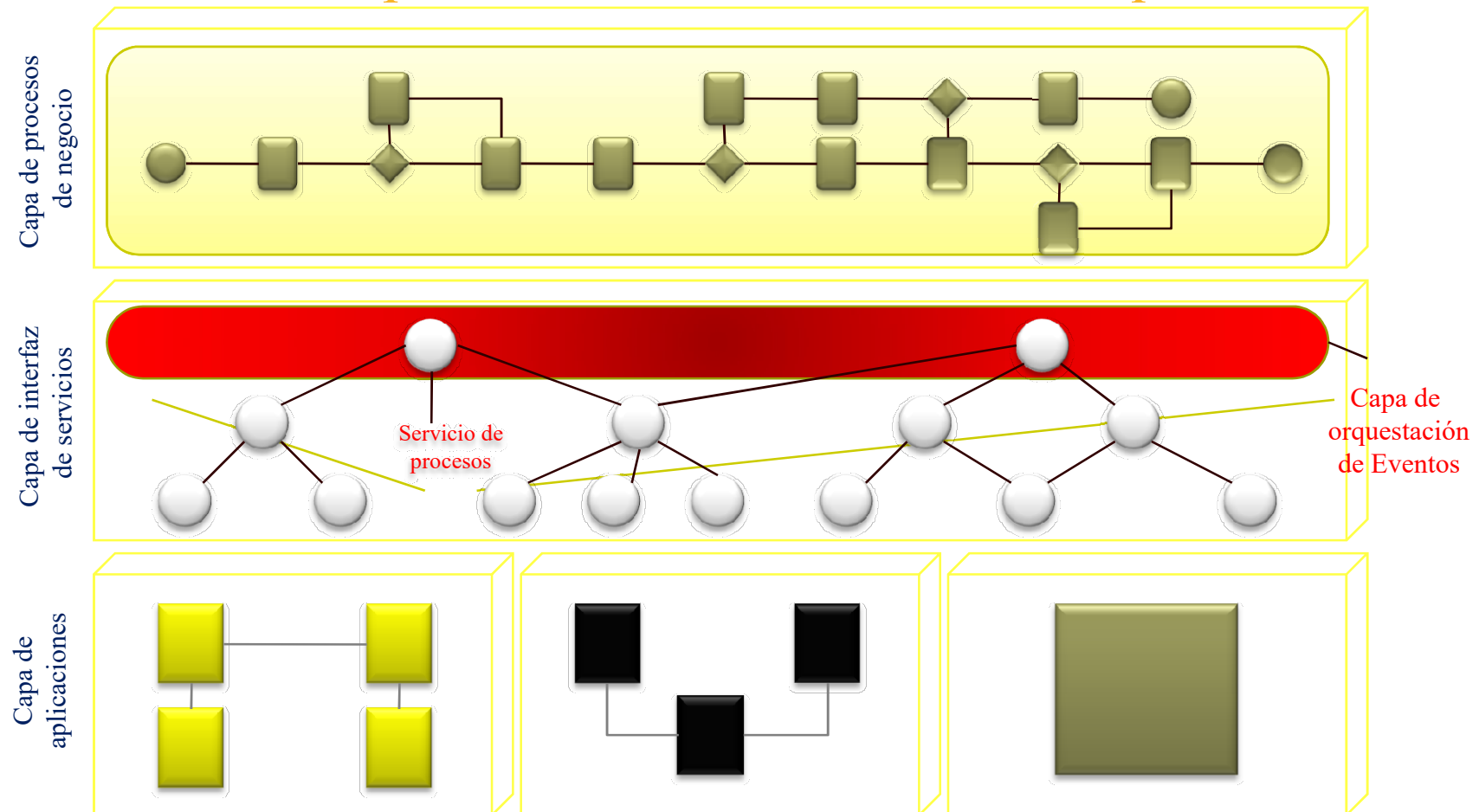




# SISTEMAS & ORGANIZACIONES

## CLASE 7

### Modelo de implementación BPM/SOA - Orquestación





### CONCLUSIÓN





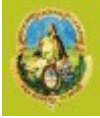
### CLASE 7

---

## CONCLUSIÓN

La relación entre BPM y SOA es fundamental, entendiendo que las herramientas de software para SOA son las que soportan los procesos de negocio diseñados con BPM. Esta relación se traduce igualmente, en el trabajo conjunto de expertos de negocio que modelan los procesos de negocio, y de expertos técnicos informáticos que detectan los servicios nuevos o de aplicaciones existentes, que hacen parte de esos procesos de negocio, (y los logran acoplar preservando las diversas condiciones de seguridad, auditoría, transaccionalidad...).

Las herramientas de infraestructura para soportar los sistemas SOA son numerosas y están en plena evolución, a juzgar por las variadas propuestas de los proveedores de tecnología. Igualmente, hay múltiples propuestas de metodologías para manejar el proceso de desarrollo de los sistemas SOA. -



### CLASE 7

---

#### 7.3- Microservicios – Kubernetes (K8s) – Dockers - DevSecOps

Microservicios: La arquitectura de microservicios es un método para desarrollar una sola aplicación como un conjunto de pequeños servicios. Cada aplicación se ejecuta en su propio proceso y se comunica con un mecanismo ligero (generalmente API de recursos HTTP). Estos servicios se basan en funciones comerciales y se pueden implementar de forma independiente a través de un mecanismo de implementación totalmente automatizado. Estos servicios tienen una gestión centralizada mínima, se pueden escribir en diferentes lenguajes de programación y utilizan diferentes tecnologías de almacenamiento de datos.



## SISTEMAS & ORGANIZACIONES

### CLASE 7

---

#### 7.3- Microservicios – Kubernetes (K8s) – Dockers - DevSecOps

**Kubernetes** es una herramienta de gestión de contenedores que automatiza la implementación.

Es una plataforma portátil de código abierto diseñada por Google y ahora administrada por una base informática nativa de la nube. Ayuda a actualizar las aplicaciones de una manera más fácil y rápida sin ningún tiempo de inactividad. Maneja el trabajo de programar contenedores en el clúster y también administra la carga de trabajo.

Esta plataforma de orquestación automatiza muchos procesos manuales, como la implementación, la administración y el escalado de las aplicaciones en el contenedor.



## SISTEMAS & ORGANIZACIONES

### CLASE 7

---

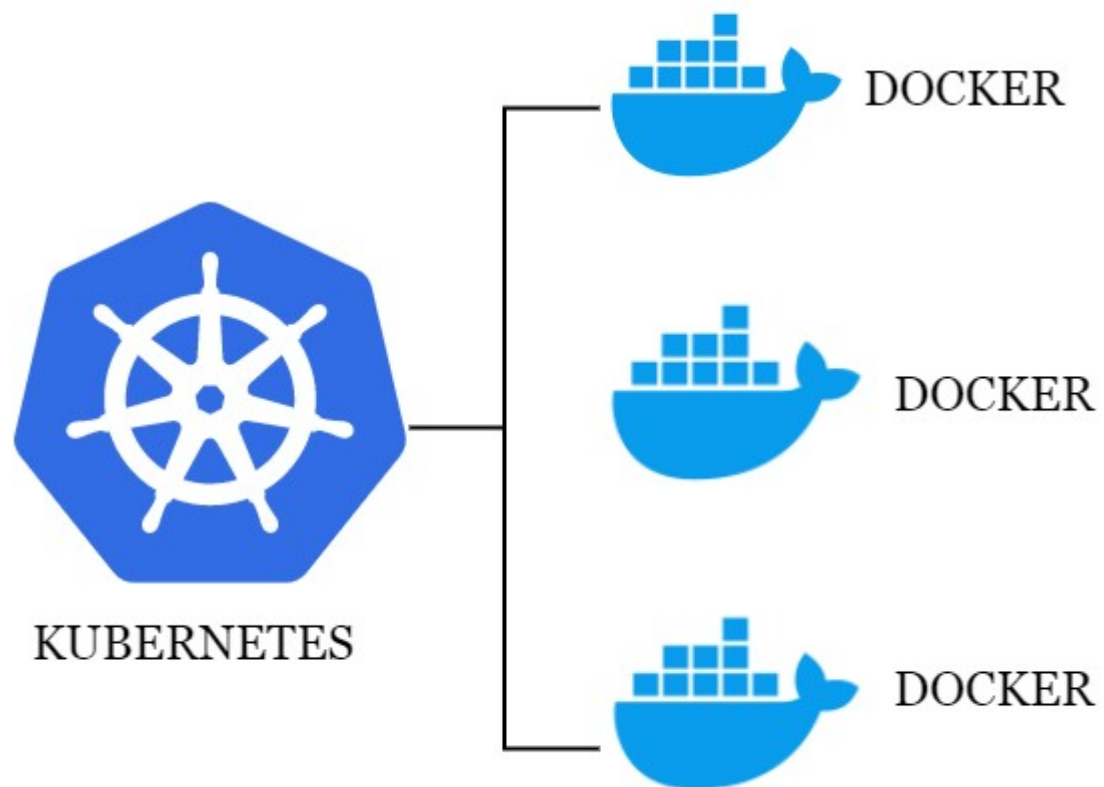
#### 7.3- Microservicios – Kubernetes (K8s) – Dockers - DevSecOps

**Docker** es una plataforma de contenedores, se usa para aislar su aplicación en contenedores, mientras que Kubernetes es una herramienta de orquestación / programador de contenedores y se usa para implementar y escalar su aplicación mediante la administración de múltiples contenedores implementados en múltiples máquinas host.



CLASE 7

7.3- Microservicios – Kubernetes (K8s) – Dockers - DevSecOps



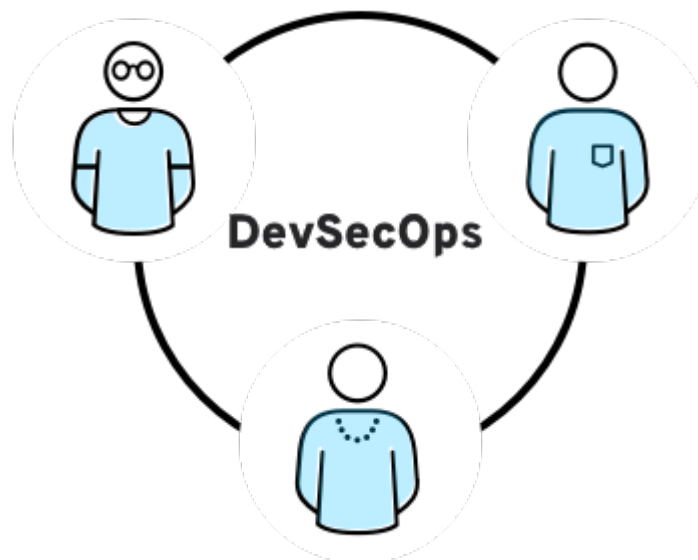




### CLASE 7

#### 7.3- Microservicios – Kubernetes (K8s) – Dockers - DevSecOps

DevSecOps significa desarrollo, seguridad y operaciones. Se trata de un enfoque que aborda la cultura, la automatización y el diseño de plataformas, e integra la seguridad como una responsabilidad compartida durante todo el ciclo de vida de la TI.





## CLASE 7

---

### Casos reales

#### 1. IBM – Pionera y evangelizadora de SOA

- ✓ **Contexto:** IBM fue una de las empresas que más promovió SOA en la década de 2000, con su *IBM SOA Reference Architecture*.
- ✓ **Aplicación:** Implementó SOA en soluciones de integración empresarial, banca y gobierno (por ejemplo, IBM WebSphere ESB, IBM BPM).
- ✓ **Actualidad:** Hoy evolucionó esos productos hacia plataformas **cloud-native** como **IBM Cloud Pak for Integration** y **Red Hat OpenShift**, manteniendo la filosofía SOA pero sobre contenedores y Kubernetes.
  - ➡ □ *SOA clásica transformada en microservicios y APIs gestionadas.*



# SISTEMAS & ORGANIZACIONES

## CLASE 7

---

### Casos reales

#### 2. Amazon – De un monolito a miles de microservicios

- ✓ **Contexto:** A comienzos de los 2000, Amazon migró su monolito e-commerce hacia una **arquitectura orientada a servicios interna (SOA)**.
- ✓ **Resultado:** Cada equipo pasó a manejar servicios independientes, con interfaces bien definidas.
- ✓ **Evolución:** Ese modelo fue el punto de partida para lo que hoy es la **arquitectura de microservicios de AWS**.
- ✓ **Tecnologías actuales:** Docker, Kubernetes (EKS), Lambda (serverless) y DevSecOps integrado.  
➡ □ *SOA → Microservicios → Cloud-native (AWS).*



## CLASE 7

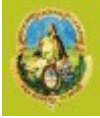
---

### Casos reales

#### 3. Netflix – De SOA a arquitectura de microservicios resiliente

- ✓ **Contexto:** Netflix implementó una arquitectura basada en SOA para soportar su expansión global.
- ✓ **Evolución:** Pasó a más de 1.000 microservicios desplegados en la nube de AWS.
- ✓ **Tecnologías:** Docker, Kubernetes, **Service Mesh propio (Netflix OSS stack: Eureka, Zuul, Hystrix).**
- ✓ **Aporte:** pionera en observabilidad y resiliencia distribuida, base de muchas prácticas DevOps actuales.

➡ □ *SOA → Microservicios → Service Mesh + Observabilidad.*



# SISTEMAS & ORGANIZACIONES

## CLASE 7

---

### Casos reales

#### 4. PayPal – Integración SOA en servicios financieros

- ✓ **Contexto:** Adoptó SOA para integrar sistemas de pago, antifraude y gestión de usuarios.
  - ✓ **Evolución:** Modernizó su stack a **microservicios desplegados con Kubernetes** en entornos híbridos.
  - ✓ **Objetivo:** escalabilidad y disponibilidad global.
- ➡ □ *SOA evolucionado con contenedores y orquestación cloud.*



# SISTEMAS & ORGANIZACIONES

## CLASE 7

---

### Casos reales

#### 5. Lufthansa Systems – SOA en el transporte aéreo

- ✓ **Contexto:** Implementó SOA para integrar múltiples sistemas de reservas, mantenimiento y logística.
- ✓ **Resultado:** interoperabilidad entre aerolíneas y socios tecnológicos.
- ✓ **Evolución:** adopción de **Kubernetes (OpenShift)** y pipelines **DevSecOps** para automatizar despliegues.
  - ➡ ☐ *SOA empresarial + automatización cloud-native.*



# SISTEMAS & ORGANIZACIONES

## CLASE 7

---

### Casos reales

#### 6. Mercado Libre

- ✓ **Contexto:** Nació con un enfoque monolítico en los 2000 y evolucionó hacia una **arquitectura de microservicios inspirada en SOA**.
- ✓ **Evolución tecnológica:**
  - Miles de microservicios desplegados en contenedores Docker.
  - Orquestación en **Kubernetes (propia y en la nube de AWS)**.
  - Infraestructura gestionada con **DevOps y observabilidad avanzada (Prometheus, Grafana)**.
- ✓ **Aporte:** es **uno de los referentes latinoamericanos** en aplicar los principios de SOA de forma moderna, escalable y cloud-native.



# SISTEMAS & ORGANIZACIONES

## CLASE 7

---

### Casos reales

#### 5. YPF Tecnología (Y-TEC)

- ✓ **Contexto:** En su plataforma de innovación y datos industriales, adoptó **arquitecturas SOA y microservicios** para integrar sistemas de monitoreo, IoT y análisis.
- ✓ **Evolución:** despliegue en **entornos cloud híbridos con Kubernetes**, priorizando interoperabilidad entre servicios científicos y de producción.
- ✓ **Objetivo:** mejorar la trazabilidad de datos industriales y acelerar el desarrollo de soluciones tecnológicas.





# SISTEMAS & ORGANIZACIONES

## CLASE 7

---

### ✿ Conclusión sobre la importancia y evolución de SOA

La **Arquitectura Orientada a Servicios (SOA)** marcó un punto de inflexión en la historia del desarrollo de software empresarial. Su aporte fue **conceptual y estructural**: introdujo la idea de construir sistemas **basados en servicios autónomos, reutilizables e interoperables**, capaces de integrar plataformas y tecnologías diversas bajo un mismo modelo lógico de negocio.

SOA permitió **romper el paradigma monolítico**, promoviendo el desacoplamiento, la estandarización de interfaces (como SOAP y WSDL) y la comunicación mediante mensajes, lo que facilitó la integración entre aplicaciones corporativas complejas. Fue el primer gran intento de **alinear la tecnología con los procesos de negocio**, un paso decisivo hacia la ingeniería del software moderna.



## SISTEMAS & ORGANIZACIONES

### CLASE 7

---

#### ✿ Conclusión sobre la importancia y evolución de SOA

Con el tiempo, las limitaciones operativas de SOA —como la rigidez del ESB o la complejidad del gobierno centralizado— impulsaron su evolución natural hacia **arquitecturas más livianas y distribuidas**, dando origen a los **microservicios**, las **APIs RESTful**, el **Service Mesh** y, en última instancia, a las **arquitecturas cloud-native**.

Hoy, los principios de SOA siguen vigentes, reencarnados en nuevas prácticas: **desacoplamiento, reutilización, automatización y gobierno distribuido**. En definitiva, **SOA fue el puente entre la integración empresarial del siglo XX y las arquitecturas ágiles, escalables y resilientes del siglo XXI**.



# SISTEMAS & ORGANIZACIONES

## CLASE 7

---

### □ Bibliografía:

Sam Newman, Chris Richardson, Red Hat, CNCF, IBM.

Stephen A. White. *“Introduction to BPMN”*. IBM Corporation

Martin Owen- Jog. Raj. “BPMN and Business Process Management. Introduction to the New Business Process Modeling Estándar”

*“Model Driven Engineering Aplicado a Business Process Management”* José Manuel perez, Francisco Ruiz, Mario Piattini Universidad de Castilla.

Las Organizaciones y los sistemas – Ing. Lorenzón

<https://IBM.com>

María Consuleo Franky – [http://www.acis.org.co/fileadmin/Revista\\_111/editorial.pdf](http://www.acis.org.co/fileadmin/Revista_111/editorial.pdf)

<https://www.redhat.com/es/topics/devops/what-is-devsecops>