

# Ejemplo examen → Temas 1, 2, 3 Sistemas Distribuidos

**1. Explica el paradigma de computación distribuida denominado MOM y sus tipos (1).**

**Enumera de forma general qué diferencias y coincidencias hay entre MOM y SOA (0,5).**

**Compara las funcionalidades del intermediario en cada una de ellas (0,5). ¿Qué mecanismo de comunicación distribuida (RPC, RMI, ORB, Servicios Web) utilizarías para implementar MOM. Justifica tu respuesta (0,5).**

- El modelo MOM (Middleware Orientado a Mensajes) es una evolución del sistema de paso de mensajes convencional. Este permite que la comunicación entre el emisor y el receptor se produzca de una manera completamente desacoplada. Se utiliza en sistemas asíncronos, es decir, donde el emisor no necesita una respuesta inmediata del receptor tras enviar el mensaje y puede continuar con su funcionamiento. Existe un intermediario, conocido como proceso intermediario, un emisor, proceso emisor y un receptor, proceso consumidor.

El proceso intermediario es el encargado de almacenar los mensajes del emisor, generalmente este es un middleware que incluye ciertos servicios para el tratamiento de los mensajes. Puede poseer las siguientes características: gestión de prioridades de mensajes, temporizadores para la gestión de mensajes, gestión de formatos de mensajes, gestión de seguridad, gestión de persistencia de los mensajes.

Es un modelo adecuado para aplicaciones y sistemas donde se quiere establecer un único punto de entrada a una comunicación asíncrona de los mismos.

Existen dos variantes de este modelo:

- Punto a punto (1:1) → cada mensaje enviado por el emisor únicamente será procesado por un proceso consumidor. El agente intermediario obtiene el mensaje, lo procesa y el mensaje es eliminado del intermediario.
- Publicación/Subscripción (1:M) → un mensaje publicado por un emisor será procesado por todos los agentes consumidores que se hayan suscrito a dicho proceso intermediario.

- Las principales diferencias entre SOA y MOM son:

- MOM es usado en comunicación asíncrona, mientras que SOA es petición/respuesta (síncrona).
- MOM consigue desacoplar el escenario completo gracias al proceso intermediario, en SOA únicamente se desacopla la localización de los servicios.
- MOM puede ser usado para la implementación de la arquitectura SOA.

- Debido a que MOM se apoya en un proceso intermediario para la gestión de los mensajes, una comunicación adecuada sería ORB. ORB dispone de un agente intermediario para la gestión de objetos ofreciéndonos la abstracción del acceso a objetos heterogéneos. Por otro lado, la arquitectura de servicios web también nos podría dar un enfoque MOM con una mayor interoperabilidad a más alto nivel.

## **2. Explica y describe la funcionalidad dentro de la tecnología de Servicios Web de WSDL, UDDI y SOAP. Explica y describe las partes de un documento WSDL. ¿Qué relación hay entre WSDL, UDDI y SOAP?.**

- Los servicios web son procesos o funciones de negocio significativas, con una interfaz bien definida y accesible a través de Internet, basada en el intercambio de mensajes en XML, pudiendo ser estos procesos o funciones combinados entre sí.

- WSDL (Web Service Description Language) es un formato XML que es utilizado para describir los mensajes SOAP que definen un servicio web en particular. Es un IDL (Interface Definition Language) para la comunicación con el servicio y definir de manera abstracta una interfaz pública del mismo. Describe principalmente el protocolo a utilizar y la implementación concreta del servicio.

- Anatomía WSDL:
  - <definitions> → definición de uno o más servicios.
  - <messages> y <portType> → operaciones que provee el servicio.
  - <binding> → cómo se invocan las operaciones.
  - <service> → dónde se ubica el servicio.
  - <documentation> → puede contener información del servicio para el usuario.
- Estructura WSDL:
  - Types → definiciones de los tipos de datos para describir los mensajes intercambiados.
  - Message → definición abstracta de los datos que se transmiten. Un mensaje es dividido en partes lógicas, cada una asociada a una definición de sistema de tipos.
  - PortType → operaciones abstractas que hacen referencia a un mensaje de entrada y uno de salida.
  - Binding → especifica el protocolo concreto y las especificaciones del formato de datos, de los mensajes definidos por un portType concreto.
  - Service → unir un conjunto de puertos relacionados.
    - Port → dirección para un Binding, para definir un único nodo de comunicación.

- UDDI (Universal Description, Discovery and Integration) es un mecanismo que nos permite localizar saber con quién comunicarse para acceder a un servicio web concreto y dónde hacerlo. Su objetivo es ser accedido por los mensajes SOAP y dar paso a documentos WSDL. Posee dos categorías de API:

- De publicación → para que los proveedores de servicios se registren (ellos y sus servicios).

- De consulta → permite al subscriptor buscar los servicios disponibles y obtenerlos una vez localizados.

- SOAP (Simple Object Access Protocol) es un protocolo ligero basado en XML para el intercambio de la información en un entorno descentralizado y distribuido, digamos que es el protocolo sobre el que hablan los servicios. Este protocolo deriva de XML-RPC. Nos facilita la comunicación entre objetos de cualquier tipo, sobre cualquier lenguaje y plataforma. Permite la comunicación máquina a máquina débilmente acoplada y el intercambio de mensajes a través de firewalls. Consta de las partes: Sobre o envoltura, reglas de codificación y representación RPC/Document.

- La relación entre las tecnologías WSDL, SOAP y UDDI radica en que son los estándares o protocolos que permiten la comunicación entre los negocios a nivel de aplicaciones. Constando de las partes de: Publicación, Búsqueda, Descubrimiento y Consumo.

**3. Explica qué es un servicio de nombres y qué es un servicio de directorios y cuales son sus principales características. De forma breve, clasifica y separa DNS, LDAP y UDDI en servicios de nombre o directorio y si son de propósito general o específico. Justifica la respuesta.**

- Un servicio de nombres provee a los clientes información sobre elementos de un sistema distribuido de manera legible para el ser humano, con el fin de identificar los mismo. Es usado para referencia recursos y usuarios, así como comunicar y compartir recursos. Este servicio almacena colecciones de pares <nombre, atributo> y busca atributos a partir de nombres. Podemos hacer un símil con las páginas blancas.

Las características principales de los servicios de nombre son:

- Utilizan el paradigma cliente/servidor.
- Es un servicio independiente fácilmente escalable.
- Independencia de su ubicación.
- Alta disponibilidad.
- La información se almacena jerárquicamente.
- Débil consistencia de replicación.
- Flexibilidad.
- BD optimizada: orientada a la lectura de información, datos de una entrada en un único registro, no necesita transacciones y tampoco bloqueos.

- Un servicio de directorio provee a los clientes información sobre objetos que satisfacen una determinada descripción. Es similar al servicio de nombre pero este nos permite buscar nombres a través de atributos. Podemos hacer un símil con las páginas amarillas.

Además de las características de los servicios de nombre, los servicios de directorio también poseen las siguientes:

- Información acerca de objetos relacionados (recursos de red, personas..).
- Refuerza la seguridad para proteger a los objetos de intrusos.
  - Servicios de nombres (páginas blancas): DNS (específico)
- DNS es un servicio de nombre ya que nos permite buscar un atributo a través de un nombre, y específico debido a que nos permite crear un espacio y extender la funcionalidad, guardando la información en la estructura que se crea.
  - Servicio de directorio (páginas amarillas): UDDI (específico) y LDAP (general).
- UDDI es un servicio de directorio ya que nos permite realizar búsquedas por una descripción dada, además es específico por lo mismo que lo es el DNS.
- LDAP es un servicio de directorio ya que nos permite realizar búsquedas por atributos y/o distinta información, además es de propósito general ya que solo nos permite crear un espacio de nombres.

**Explica el modelo arquitectónico denominado middleware orientado a mensajes (MOM). Enumera y describe sus elementos. ¿Cuáles son las principales similitudes y diferencias entre MOM y el modelo cliente-servidor convencional?**

- Mientras que el modelo C/S actúa sobre el paradigma de paso de mensajes, la arquitectura MOM es una elaboración más extensa del mismo. En el modelo MOM nos encontramos con un proceso intermediario, mientras que en la arquitectura C/S solo existen dos procesos (cliente y servidor). Ambos nos permiten la abstracción del acceso a recursos de red.

**Uno de los principales problemas en la comunicación entre entidades de un sistema distribuido es la representación de la información. Enumera y describe el funcionamiento de las tres posibles tácticas que tratan de resolver esta problemática.**

- Debido a la heterogeneidad de la información (diferentes representaciones de las mismas) surgen las siguientes tácticas para resolver esta problemática:

- Emisor se adapta a la representación del receptor: esta es la técnica menos utilizada, debido a que tiene una gran carga sobre la red debido a la traducción de los mensajes.
- Receptor adapta la representación del emisor: esta técnica, aunque es más usada que la anterior, tampoco es ampliamente aceptada. El mensaje que recibe el receptor debe contener información sobre cómo tratar dicha información y tiene la misma problemática que la anterior táctica.
- Representación externa común: esta técnica es la más utilizada actualmente, ambos se comunican de la misma manera, “usando el mismo lenguaje”.

**Comenta brevemente el modelo CGI indicando sus ventajas y desventajas. Justifica su aparición partiendo del modelo Web básico**

- CGI (Common Gateway Interface) especifica una manera de transferir datos entre un cliente y un programa alojado en un servidor. El servidor web pasará las solicitudes que reciba del cliente a un programa externo (CGI) y la salida de dicho programa será enviada al cliente.

Sus principales ventajas son:

- Capacidad de respuesta dinámica.
- Libertad de elección del lenguaje de programación.

Y sus principales desventajas:

- No existe relación entre el programa CGI y el servidor web, por lo que no existe un control sobre la ejecución del mismo, ni del resultado.
- Se crea una nueva instanciación del CGI por cada solicitud produciendo una sobrecarga de recursos

La principal justificación de su aparición es la necesidad de ofrecer contenido web dinámico, capaz de interactuar con el usuario a través de una aplicación web sobre el protocolo HTTP.

**Justifica la gestión distribuida de LDAP, define las ventajas sobre otros sistemas de información tradicionales y enumera y describe los motivos que pueden llevar a esta gestión distribuida. ¿Qué técnica es utilizada para relacionar las diferentes partes del espacio de nombres cuando es distribuido?**

- WTF?

**Explicar arquitectura SOA**

- La arquitectura SOA (Service Oriented Architecture) es un paradigma de arquitectura para desarrollar sistemas distribuidos. Permite la abstracción de acceso a actividades de negocio, conocidas como servicios.

Sus principales características son:

- Localización, descubrimiento y publicación.
- Interoperabilidad.
- Composición.
- Autonomía y autocontenidos.
- Reusabilidad y desacoplamiento.
- Contrato bien definido.

- Sin estado

Sus principales componentes son un proveedor de servicios, un consumidor y un registro de los mismos.

Los pasos para su utilización serían: Publicación, búsqueda, descubrimiento y consumo.

Las herramientas utilizadas en este paradigma serían los servicios web, JINI y UPnP entre otros.

### **Explicar los Servlets, sus características y compararlos con los CGI y las paginas dinámicas.**

- Los Servlets son una tecnología utilizada para la ampliación del modelo web básico desde el lado del servidor. El uso más común de los Servlets es generar webs de forma dinámica a partir de parámetros de una petición enviada por un cliente.

Sus principales características son:

- Portabilidad → solo precisa de JVM.
- Rendimiento → única instanciación.
- Sesión → mantiene información en diferentes conexiones al servlet.
- Software Distribuido → permite comunicación entre servlets.
- Multithread → permite concurrencia