**[[LIBRO] Sistemas Distribuidos: Conceptos y Diseño](http://foro.gradoinformaticaua.es/viewtopic.php?f=42&t=95" \l "p433)**

<http://www.mediafire.com/?1yeiye4em14492p>

password: apuntesdesistemas.com.ar

**2012**

En el del miércoles por la mañana que es el que yo hice salió servicio de nombres y de directorio, aplicado a bases de datos o algo así, también que era SOA, y si mal no recuerdo tambíén entró WSDL, respecto al examen de la tarde no tengo mucha información, me dijeron que les entró UDDI y ya no se más.

**2013**

Pues despues de hacerlo, la cosa no a sido muy bestia.  
Lo que me acuerdo:  
  
-Explicar la arquitectura SOA  
-Explicar los Servlets, sus caracteristicas y compararlos con los CGI y las paginas dinamicas  
-Explicar que es un servicio de nombres y un servicio de dominios, relacionarlos con DNS, LDAP y UDDI, clasificar estos terminos y decir si son de uso general o especifico, con justificacion  
-Explicar la estructura de de los componentes de la practica no guiada de RMI, de forma esquematica, decir si es o no importante el registro rmi en la practica (rmiregistry)  
- 10 preguntas de test, algunas rebuscadillas

**[Examen Grupo ARA 2014]**  
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). ¿Qúe mecanismo de comunicación distribuida (RPC, RMI, ORB, Servicios Web) utilizarías para implementar MOM. Justifica tu respuesta (0,5).  
[2,5 puntos]  
  
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?.  
[2 puntos]  
  
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.  
[2 puntos]  
  
4. Respecto a la práctica no guiada de sockets y RMI que habéis realizado, explica esquemáticamente la arquitectura desde el navegador hasta el controlador. ¿Qué importancia tiene el controlador?  
[1 punto]  
  
Preguntas tipo test. 10 preguntas, 4 respuestas, las que falles no restan.  
[2,5 puntos]

Explica el paradigma de computación distribuida denominado SOA (Service Oriented Architecture) (0,5).  
Enumera y describe sus elementos (0,5). ¿Cuáles son las principales similitudes y diferencias entre SOA y  
el modelo cliente-servidor convencional (básate en los principios de SOA) (0,5)? ¿Qué relación existe  
entre el modelo arquitectónico MatchMaker de SOA y el de Servicios Web (0,5)? ¿Qué mecanismos de  
comunicación distribuida (RPC, RMI, ORB, Servicios Web) utilizarías para implementar SOA? Justifica la  
respuesta (0,5). [2,5 puntos]  
  
2. Dentro de las tecnologías web describe el modelo servlet (0,5) indicando sus principales características  
(0,5). Realiza una breve comparativa con el modelo CGI (0,5) y el de páginas activas atendiendo, en este  
último, a las recomendaciones de uso (0,5). [2 puntos]  
  
3. Explica qué es un servicio de nombres (0,4) y qué es un servicio de directorio (0,4) y cuáles son sus  
principales características (0,6). En relación con estos conceptos y de forma breve, clasifica y realiza una  
comparativa entre DNS, LDAP y UDDI atendiendo a qué tipo de servicio son y si son de propósito general  
o específico. Justifica la respuesta (0,6).

Examen SD 2015 primer parcial

1º Explica y describe la funcionalidad dentro de la tecnología de Servicios Web de WSDL,UDDI y SOAP (0,5p)

Explica y describe las partes de un documento WSDL(0,6p)

Que relación existe entre WSDL,UDDI y SOAP?(0,8p)

2º Explica SOA.(0,5p)

Enumera y describe sus elementos(0,5p)

Similitudes y diferencias entre SOA y el modelo cliente-servidor convencional?(0,5p)

Que relación existe entre el modelo arquitectónico MatchMaker de SOA y el Servicios web?(0,5p)

Que mecanismos de Comunicación distribuida(RCP,RMI,ORB, Servicios web) utilizarías para implementar SOA?(0,5p)

3º Explica esquemáticamente tu practica desde el controlador hasta sensores y actuadores(0,75p)

Que importancia tiene el RMIRegistry?(0,25p)

4º Explica los diferentes modos de resolución de nombres que pueden usar los DNS(resolución iterativa en cliente, resolución iterativa en el servidor y resolución recursia)(1,5p)

Realiza una comparativa entre ellos(0,5p)

Preguntas de test(son 10 preguntas pero solo apunte 6) (2,5p)

1º cual de los siguientes modelos tecnológicos web permiten generación dinámica de contenidos?

1. Servlets y CGI
2. Web básico
3. Paginas activas
4. A y c correctas

2º En que patrón de intercambio debería tener para comunicación síncrona en cliente/servidor donde el cliente inicia la comunicación?

1. Solicitud-Respuesta
2. Unidireccional
3. Petición-Respuesta
4. Notificación

3º Relación entre documento WSDL y mensaje SOAP?

1. Ninguna
2. SOAP define como se genera WSDL
3. En WSDL se indica la estructura de SOAP
4. B y C son correctas

4º Objetivos del middleware

1. Abstraer y dar heterogeneidad a los integrantes de un sistema
2. Permitir distribuir recursos hardware a nivel local en diferentes maquinas para aumentar escalabilidad
3. Actuar de controlador para unificar operaciones comunes
4. Ninguna de las anteriores

5º Que es la convención de nombrado de un sistema de nombres?

1. Sintaxis para definir nombres
2. Reunión anual
3. Normativa estándar para servicios web
4. A y C son correctas

6º Un servicio de nombres o directorio:

1. Optimizado lecturas
2. Optimizado escritura
3. Sistema Escalable
4. A y C son correctas

**[Otros que hay por el foro]**

1 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? [2 puntos]  
  
2 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 [2 puntos]  
  
3 Comenta brevemente el modelo CGI indicando sus ventajas y desventajas. Justifica su aparición partiendo del modelo Web básico [2 puntos]  
  
4 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? [2 puntos]  
  
  
-Explicar la arquitectura SOA  
-Explicar los Servlets, sus caracteristicas y compararlos con los CGI y las paginas dinamicas  
-Explicar la estructura de de los componentes de la practica no guiada de RMI, de forma esquematica, decir si es o no   
-Explicar que es un servicio de nombres y un servicio de dominios, relacionarlos con DNS, LDAP y UDDI, clasificar estos terminos y decir si son de uso general o especifico, con justificacionimportante el registro rmi en la practica (rmiregistry).

-Explicar la arquitectura SOA  
-Explicar los Servlets, sus caracteristicas y compararlos con los CGI y las paginas dinamicas  
-Explicar la estructura de de los componentes de la practica no guiada de RMI, de forma esquematica, decir si es o no   
-Explicar que es un servicio de nombres y un servicio de dominios, relacionarlos con DNS, LDAP y UDDI, clasificar estos terminos y decir si son de uso general o especifico, con justificacionimportante el registro rmi en la practica (rmiregistry).

-Explicar la arquitectura SOA  
-Explicar los Servlets, sus caracteristicas y compararlos con los CGI y las paginas dinamicas  
-Explicar que es un servicio de nombres y un servicio de dominios, relacionarlos con DNS, LDAP y UDDI, clasificar estos terminos y decir si son de uso general o especifico, con justificacion  
-Explicar la estructura de de los componentes de la practica no guiada de RMI, de forma esquematica, decir si es o no importante el registro rmi en la practica (rmiregistry)  
- 10 preguntas de test, algunas rebuscadillas

1. Explica el paradigma de computación distribuida denominado SOA (Service Oriented Architecture) (0,5).  
Enumera y describe sus elementos (0,5). ¿Cuáles son las principales similitudes y diferencias entre SOA y  
el modelo cliente-servidor convencional (básate en los principios de SOA) (0,5)? ¿Qué relación existe  
entre el modelo arquitectónico MatchMaker de SOA y el de Servicios Web (0,5)? ¿Qué mecanismos de  
comunicación distribuida (RPC, RMI, ORB, Servicios Web) utilizarías para implementar SOA? Justifica la  
respuesta (0,5). [2,5 puntos]

2. Dentro de las tecnologías web describe el modelo servlet (0,5) indicando sus principales características  
(0,5). Realiza una breve comparativa con el modelo CGI (0,5) y el de páginas activas atendiendo, en este  
último, a las recomendaciones de uso (0,5). [2 puntos]  
  
3. Explica qué es un servicio de nombres (0,4) y qué es un servicio de directorio (0,4) y cuáles son sus  
principales características (0,6). En relación con estos conceptos y de forma breve, clasifica y realiza una  
comparativa entre DNS, LDAP y UDDI atendiendo a qué tipo de servicio son y si son de propósito general  
o específico. Justifica la respuesta (0,6).

Respuestas del foro, no tienen porque estar bien o perfectas(AVISO)

Ambos son modelos arquictónicos. Las diferencias son que MOM se trata de una comunicación asícrona (ya si eso llegará la respuesta) y SOA es petición/respuesta. Otra diferencia es que SOA sólo desacopla la localización, mientras que MOM desacopla todo, es decir, en MOM todo pasa por el intermediario mientras que en SOA una vez se ha descubierto el servicio ya son los nodos los que se comunican entre ellos. Básicamente todas las diferencias están en la estructura de ambos (los dibujicos de las diapositivas. Otra creo que es la interoperabilidad, SOA es interoperable, pero en MOM todo ha de estar en el mismo lenguaje. No se podría hacer lo que vamos a hacer en prácticas, comunicar una app en .NET con otras en Java.

cuales son las 3 tácticas en un sistema distribuido para resolver la problemática de representación de información? Ahora mismo no caigo jaja

La problemática de representación viene por la heterogeneidad del medio, el emisor y el receptor no tienen porqué usar el mimismo lenguaje o tener la misma arquitectura (little y big endian), las tres tácticas que existen son:

* **Emisor se adapta al receptor** (menos usado): tiene una gran latencia por preguntar y traducir.
* **Receptor se adapta al emisor** (bastante utilizado): el mensaje inicial contiene información para que el receptor pueda interpretar la información. Por ejemplo, el protocolo del correo electrónico.
* **Representación externa común** (el más utilizado): Seguro que lo entenderá porque ambos hablan ese lenguaje. Por ejemplo, Corba, servicios web con xml o RMI para java.

tipos de P2P

- Centralizado: El cliente se conecta a un nodo privilegiado que se encarga de localizar el recurso, pedirlo y mandárselo al cliente. Todo pasa por este nodo privilegiado.   
- Descentralizado: Es el P2P puro, no existen nodos privilegiados y los datos se encuentran distribuidos por toda la red. A veces unos nodos actúan como clientes y a veces como servidor indistintamente. Por ejemplo, este es el método usado para videconferencia.  
- Híbrido: Existe un nodo privilegiado donde se conectan los clientes cuando necesitan un recurso, este nodo localiza el recurso y se encarga de decirle al cliente la localización. La transferencia es desde el servidor hasta cliente sin pasar por el nodo central. Usado por ejemplo en el protocolo Torrent.

Otra pregunta ¿entendeis bien la comunicación en el Midleware en un server de aplicaciones? En los apuntes que tome en clase hablaba de comunicación textual o binaria. Luego te hablan de RPC y RMI y supuse que con binaria se refería a éstos 2. la pregunta es si a parte de objetos remotos también hay petición y respuesta como en HTTP.

Middleware es un concepto abstracto. Define una arquitectura donde unos clientes utilizan unos servicios remotos y el middleware se encarga de que esa comunicación sea transparente.  
Se puede implementar mediante ORB o mediante RMI o mediante web services...  
Como decía la mayor ventaja es la transparencia, así pues, si una tecnología pretende hacer uso del paradigma middleware debe aportar herramientas para proporcionarla, por ejemplo a través de los stub y skeletons.   
Realmente los stub y skeletons son clases (estoy hablando de la implementación en Java) que implementan los métodos de más bajo nivel para comunicar con los objetos remotos.  
Como ves, no es que middleware sea algo tangible, es solamente un paradigma susceptible de ser implementado de muchas maneras y con muchas tecnologías.  
  
Otra cuestión relacionada es el papel del Midleware usando RPC Y RMI. Supongo que los componentes SW disponen de los stub y skeleton que serían los que solventarían los problemas de la comunciación pero ¿que papel tiene el MW aquí? ¿simplemente hacer posible pasar los datos serializados o eske los stub y skeleton forma parte de él? aquñi me hago un buen lio.

Respecto a la comunicación binaria o textual hace referencia a los mensajes que se mandan en la tecnología que subyace bajo RMI, RPC, Web Services o lo que sea.  
Como sabes los componentes remotos se comunican entre sí serializando los datos y mandándolos mediante mensajes, pues bien estos mensajes pueden ser de tipo binario o tipo textual. Por poner un ejemplo, RPC utiliza un sistema binario y web services uno textual (ya sabes que un mensaje SOAP es realmente un XML). Evidentemente, que sea textual acarrea la gran ventaja de ser más interoperable, que es lo que se busca en este tipo de tecnologías.

Por una parte los equipos son heterogeneos (64-32 bits, litlle o big endian...). Luego se programa en distintos lenguajes. Bien, proponen que lo solvente las diversidades usando una representación externa común. Pero yo aquí veo que esto solo solventan el hecho de que use distintos lenaguajes de programación.

El hecho de que se comuniquen con una representación externa común ya tiene en cuenta ese tipo de problemas.  
Por ejemplo un lenguaje de representación externa puede ser XML.   
Si en un mensaje XML se especifica que existe una variable cuyo tipo de dato es entero, tanto el emisor como el receptor, en sus distintos lenguajes de programación saben que es un entero (para c es int, para Java es Integer). Que sea big endian o little endian no nos importa porque ese aspecto está a mucho más bajo nivel (el int de c en una máquina puede estar como big endian y en Integer de Java en little endian, pero eso da igual, el lenguaje toma el dato, entero, y cómo lo almacene en memoria ya es cosa suya).   
Y lo mismo para si el sistema es de 32 o 64 bits, eso está a muy bajo nivel.  
  
El matiz es que no es el mismo tipo de dato, no se envía digamos el churro de memoria correspondiente a un dato, sino que se envían los datos en string (en el XML) y cada sistema infiere el tipo de dato en función de lo que ponga en el mensaje.