



Asignatura..... **SISTEMAS INFORMÁTICOS II** Grupo..... **231**
 Apellidos Nombre.....
 Ejercicio del día..... **9 de marzo de 2016. Examen parcial.**.....

Teoría 1 (2)	Teoría 2 (2)	Teoría 3 (2)	Teoría 4 (2)	Teoría 5 (2)	Total Teoría (10)

1.- TEORÍA (10 puntos). Contesta de modo claro y conciso a las siguientes cuestiones.

1. Explica en qué consiste la transparencia de acceso.

2. Explica la principal utilidad de los servicios de directorio y nombres, y qué diferencia hay entre los servicios de directorio y los servicios de nombres.

3. Nombra las dos interfaces de invocación de métodos remotos disponibles en CORBA y describe sus ventajas y sus inconvenientes.



Asignatura..... **SISTEMAS INFORMÁTICOS II** Grupo..... **231**
 Apellidos Nombre.....
 Ejercicio del día..... **9 de marzo de 2016. Examen parcial.**.....

4. Explica las diferencias entre los servicios web basados en SOAP y servicios web basados en REST.

5. Explica todos los pasos que se llevan a cabo desde que se invoca una aplicación cliente que ejecuta un procedimiento remoto según la implementación RPC de SUN microsystems, hasta que se obtiene la respuesta correspondiente.



Asignatura..... **SISTEMAS INFORMÁTICOS II** Grupo..... **231**
 Apellidos Nombre.....
 Ejercicio del día..... **9 de marzo de 2016. Examen parcial.**.....

2.1 (6)	3.1 (2)	4.1 (2)	Total Problemas (10)

2. PROBLEMA. Considerar los siguientes casos particulares de sistemas distribuidos:

1. Se quiere desarrollar un sistema a través del cual los clientes se conecten a un servidor para realizar consultas a una base de datos: modificar las filas de una tabla, insertar nuevas filas, borrar las filas indicadas, etc. El cliente y el servidor serán desarrollados en lenguaje Java. Por la arquitectura de la red, es necesario que los mensajes no sean excesivamente grandes. El tráfico no será filtrado por ningún corta-fuegos. Se desea que la codificación de cliente y servidor sea lo más sencilla posible. Los clientes serán de arquitecturas heterogéneas.
2. Se quiere desarrollar un sistema cliente servidor donde los clientes soliciten al servidor información sobre una determinada ubicación: restaurantes, información meteorológica, información del tráfico rodado, etc. El tráfico de mensajes no será filtrado de ninguna forma, pero se desearía que el tamaño de los mensajes fuese pequeño. Los clientes serán de distintas arquitecturas y serán implementados en múltiples lenguajes de programación. La complejidad del proceso de desarrollo no supone un problema.
3. En este sistema los clientes son heterogéneos y están implementados en múltiples lenguajes de programación. Los clientes al acceder al servicio han de pasar por un corta-fuegos. La empresa cuenta con un ancho de banda considerable, por lo que el tamaño de los mensajes enviados y recibidos no será un problema. El servicio prestado incluye la posibilidad de realizar búsquedas en el catálogo de artículos de la empresa, registrar nuevos artículos, procesar la venta de un artículo, etc. En el futuro se espera que aparezcan nuevos servicios y se desearía que los clientes lo pudiesen utilizar de forma sencilla.
4. Sistema de computación distribuido. En este sistema los clientes pueden mandar trabajos a un servidor de computación. Cuando el servidor de cálculo recibe un trabajo, extrae la ruta al ejecutable, y lanza un proceso para ejecutarlo. Generalmente los trabajos enviados requieren varias horas de cálculo hasta terminar. El servidor solo puede ejecutar 32 trabajos de forma simultánea, y en caso de que haya más trabajos, los últimos en recibirse tendrán que esperar. Hay dos tipos de clientes, los normales y los preferentes. Estos últimos tendrán prioridad para ejecutar trabajos. Cada cliente tiene asociado un tiempo de cálculo. Una vez agotado, sus trabajos enviados al servidor serán ignorados. Los clientes que usarán el servicio son homogéneos. Se desearía evitar que la complejidad de desarrollo de clientes y servidor fuese elevada. En caso de que el servidor esté caído no queremos que se pierdan las solicitudes de cálculo de los clientes.

2.1. (6 puntos) Para cada uno de ellos se pide elegir razonadamente el mecanismo de comunicación más adecuado: colas de mensajes, objetos distribuidos basados en Java RMI, objetos distribuidos basados en CORBA y servicios web basados en SOAP.

No se tendrán en cuenta respuestas sin justificación (cuantas más justificaciones, mayor la puntuación).



SISTEMAS INFORMÁTICOS II

231

Asignatura..... Grupo.....

Apellidos..... Nombre.....

Ejercicio del día **9 de marzo de 2016. Examen parcial.**

1. Hay que implementar funcionalidades síncronas, por lo que descartaríamos las colas de mensajes. Este mecanismo también quedaría descartado, ya que los clientes son heterogéneos, y no incorpora nada para la traducción de datos. Como el tamaño de los mensajes debe ser pequeño y el tráfico no será filtrado, descartaríamos web servicios basados en SOAP. Como cliente y servidor serán implementados en Java, y queremos que ambos sean fáciles de implementar elegiríamos Java RMI y descartaríamos CORBA por ser más complicado.
2. Hay que implementar funcionalidades síncronas, por lo que descartaríamos las colas de mensajes. Este mecanismo también quedaría descartado, ya que los clientes son heterogéneos, y no incorpora nada para la traducción de datos. Como el tamaño de los mensajes debe ser pequeño y el tráfico no será filtrado descartaríamos web servicios basados en SOAP. Como los clientes serán implementados en múltiples arquitecturas y múltiples lenguajes de programación, y la complejidad de desarrollo no penaliza, elegiríamos CORBA frente a RMI.
3. Hay que implementar funcionalidades síncronas, por lo que descartaríamos las colas de mensajes. Este mecanismo también quedaría descartado, ya que los clientes son heterogéneos, y no incorpora nada para la traducción de datos. En este caso los mensajes han de pasar a través de un corta-fuegos, lo que sugiere servicios web basados en SOAP y descarta CORBA y Java RMI. También elegimos este último mecanismo ya que podríamos usar UDDI para buscar nuevos servicios en tiempo de ejecución y mediante el fichero WSDL podríamos obtener la información para invocarlos de forma automática. Como SOAP, UDDI y WSDL son estándar, no habría problema en usar múltiples lenguajes de programación, y tendríamos transparencia de datos.
4. Debido a que los trabajos enviados al servidor requieren varias horas para terminar lo más adecuado sería usar comunicación asíncrona. Esto sugeriría colas de mensajes frente a los demás mecanismos (RMI o WS-SOAP). Razones adicionales para usar colas de mensajes frente a los otros mecanismos son: la existencia de prioridades en los trabajos (se implementa con prioridades en los mensajes de ejecución de trabajo), la necesidad de filtrar trabajos, y el hecho de que si el servidor está caído, y mandamos un trabajo, esta solicitud no se perdería. Como los sistemas son homogéneos no habría que preocuparse por la traducción de datos. Como el desarrollo no debe ser complicado, descartamos CORBA.



Asignatura..... **SISTEMAS INFORMÁTICOS II** Grupo..... **231**
 Apellidos Nombre.....
 Ejercicio del día..... **9 de marzo de 2016. Examen parcial.**.....

3. PROBLEMA Un servidor B recibe un mensaje de otro servidor A a las 12:34:16.35 (las dos últimas cifras son centésimas de segundo). El mensaje lleva una marca de tiempo igual a las 12:34:12.32. Tras esto el servidor A recibe un mensaje del servidor B a las 12:34:20.82. La marca de tiempo del mensaje es igual a las 12:34:16.92.

3.1. (2 puntos) Determina la deriva entre los relojes de ambos servidores e indica la precisión de la estimación realizada.

$T_{i-3}=12:34:12.32$

$T_{i-2}=12:34:16.35$

$T_{i-1}=12:34:16.95$

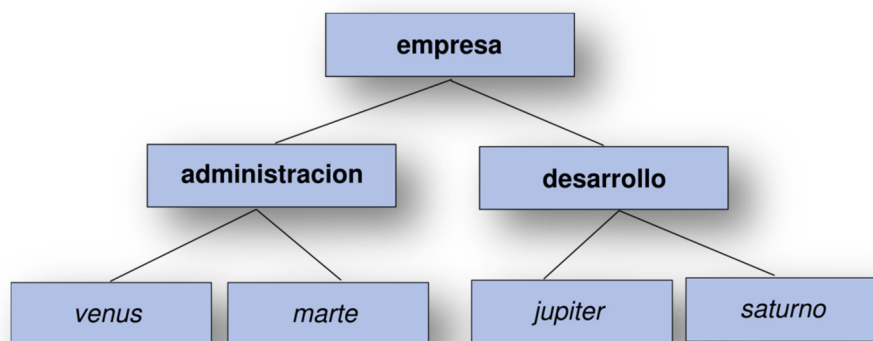
$T_i=12:34:20.82$

$$o_i = ((T_{i-2} - T_{i-3}) - (T_i - T_{i-1})) / 2 = ((16.35 - 12.32) - (20.82 - 16.92)) / 2 = (4.03 - 3.9) / 2 = 0.13 / 2 = 0.065 \text{ segundos}$$

$$d_i / 2 = ((T_{i-2} - T_{i-3}) + (T_i - T_{i-1})) / 2 = ((16.35 - 12.32) + (20.82 - 16.92)) / 2 = (4.03 + 3.9) / 2 = 7.93 / 2 = 3.967 \text{ segundos}$$

La deriva entre los relojes es de 0.065 s +/- 3.967 segundos.

4. PROBLEMA A efectos de asignación de nombres a los distintos recursos que lo componen, un determinado sistema distribuido se encuentra dividido en distintos dominios administrativos, organizados de modo jerárquico según el siguiente esquema, donde los nodos en cursiva representan servidores, y los nodos en negrita, los dominios administrativos. Cada dominio administrativo tiene un servidor de directorio cuyo nombre coincide con el dominio.



4.1 (2 puntos) Suponiendo que el servicio de directorio está implementado según el estándar X.500, indicar en qué ordenadores se ejecutará el Directory User Agent, y en qué ordenadores se ejecutará el Directory System Agent. Indicar los protocolos de comunicación utilizados para intercambiar información de directorio entre los distintos ordenadores.

DUA: venus, marte, júpiter, Saturno.

DSA: administración, desarrollo, empresa.

Directory Access Protocol: venus-administración, marte-administración, júpiter-desarrollo, Saturno-desarrollo.

Directory System Protocol: administración-empresa, desarrollo-empresa.