



Asignatura..... **SISTEMAS INFORMÁTICOS II** ..... Grupo..... **231**  
 Apellidos ..... Nombre.....  
 Ejercicio del día..... **15 de marzo de 2017. Examen parcial.**.....

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

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

1. Describe en qué consiste el proceso de marshalling y unmarshalling en las RPCs.

2. Explica la diferencia entre un modelo de interacción síncrono y asíncrono.

3. Nombra los elementos principales de un fichero XML que siga el formato estándar WSDL.

4. Nombra los protocolos y componentes del estándar de directorios X.500.

5. Indica la utilidad de los acuerdos de nivel de servicio (service level agreements).



Asignatura..... **SISTEMAS INFORMÁTICOS II** ..... Grupo..... **231**  
 Apellidos ..... Nombre.....  
 Ejercicio del día..... **15 de marzo de 2017. Examen parcial.**.....

Teoría 6 (1)	Teoría 7 (1)	Teoría 8 (1)	Teoría 9 (1)	Teoría 10 (1)

6. Nombra las dos interfaces de invocación remota de CORBA.

7. Indica la principal utilidad del estándar UDDI.

8. Indica el principal problema de transparencia de las RPCs.

9. Nombra tres situaciones ideales para el uso de colas de mensaje como mecanismo de comunicación.

10. Indica qué ventajas tiene usar WS basados Rest frente a WS basados en SOAP.



Asignatura..... **SISTEMAS INFORMÁTICOS II** ..... Grupo..... **231**  
 Apellidos ..... Nombre.....  
 Ejercicio del día..... **15 de marzo de 2017. Examen parcial.**.....

2.1 (4)	3.1 (2)	4.1 (4)	Total Problemas (10)

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

1. Se quiere desarrollar un sistema cliente servidor para acceder al catálogo de productos de una empresa, realizar búsquedas, ventas de dichos productos a los clientes, gestionar envíos, etc. La empresa hará uso de un corta-fuegos para aumentar la seguridad de su red de área local. Se desea además que los clientes estén implementados en múltiples lenguajes de programación y se sabe que los sistemas que accederán al servicio serán heterogéneos. En el futuro se espera que aparezcan nuevas funcionalidades ofertadas como servicio y se desearía que los clientes pudiesen usarlas sin realizar cambios significativos en los mismos, a ser posible en tiempo de ejecución. La empresa cuenta con un ancho de banda que excede con creces las necesidades del servicio y no resulta crítico procesar los mensajes en poco tiempo.
2. Se quiere desarrollar un sistema cliente servidor que permita gestionar los servicios ofertados por un banco. Estos incluyen la realización de transferencias, consulta de saldo, adquisición de productos financieros, compras mediante tarjetas de crédito, etc. Los clientes que usarán el servicio serán heterogéneos y estarán programados en múltiples lenguajes de programación. La empresa no cuenta con ningún corta-fuegos. Sin embargo, el ancho de banda que tiene está limitado y resulta crítico procesar las solicitudes recibidas en poco tiempo. La complejidad del proceso de desarrollo no supone un problema. Sin embargo, se desearía mantener un nivel bajo de acoplamiento entre clientes y servidores.

**2.1. (4 puntos)** Para cada uno de ellos se pide elegir razonadamente el mecanismo de comunicación más adecuado entre los vistos en la parte de teoría de la asignatura (Java RMI, CORBA, WS-soap, WS-rest, RPC, UDP, TCP o Colas de Mensajes). Indicar así mismo si será necesario implementar algún **mecanismo adicional de traducción de datos**.

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

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. Descartaríamos UDP y TCP ya que la funcionalidad del sistema es complicada y un corta-fuegos podría filtrar los mensajes. Por esta misma razón descartamos RPC, CORBA y Java RMI. Propondríamos usar WS-soap ya que es estándar, soporta múltiples lenguajes de programación y sistemas heterogéneos, y pasa a través de cortafuegos. Frente a WS-Rest elegiríamos WS-soap ya que resultaría sencillo integrar nuevos servicios descubiertos en tiempo de ejecución mediante (UDDI y WSDL). Como la empresa tiene suficiente ancho de banda y no es necesario procesar rápidamente los mensajes WS-soap no estaría penalizado por usar mensajes grandes y costosos de procesar. Ya que los WS-soap usan XML para representar la información, que es independiente de la plataforma, no habrá que implementar ningún mecanismo adicional de traducción de datos.



**SISTEMAS INFORMÁTICOS II**

Asignatura..... Grupo..... 231

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

Ejercicio del día..... 15 de marzo de 2017. Examen parcial.

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. Java RMI lo descartaríamos al tener que usar múltiples lenguajes de programación. También preferiríamos CORBA frente a RPC, ya que CORBA es estándar, y RPC depende de la versión utilizada. CORBA tendría un abanico de compatibilidad mucho más grande. Descartaríamos WS-rest ya que queremos un grado bajo de acoplamiento entre cliente y servidor. Descartaríamos TCP y UDP ya que la funcionalidad a desarrollar es complicada y hay que traducir datos. CORBA ya proporciona un mecanismo propio de traducción de datos, por lo que no será necesario implementar ninguno adicional.

**3. PROBLEMA** Un servidor B recibe un mensaje de otro servidor A a las 10:34:16.35 (las dos últimas cifras son centésimas de segundo). El mensaje lleva una marca de tiempo igual a las 10:34:12.32. Tras esto el servidor A recibe un mensaje del servidor B a las 10:34:12.97. La marca de tiempo del mensaje es igual a las 10:34:16.95.

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

$T_{i-3}=10:34:12.32$   
 $T_{i-2}=10:34:16.35$   
 $T_{i-1}=10:34:16.95$   
 $T_i=10:34:12.97$

$$o_i = ((T_{i-2} - T_{i-3}) - (T_i - T_{i-1})) / 2 = ((16.35 - 12.32) - (12.97 - 16.95)) / 2 = (4.03 + 3.98) / 2 = 4.005 \text{ segundos}$$

$$d_i / 2 = ((T_{i-2} - T_{i-3}) + (T_i - T_{i-1})) / 2 = ((16.35 - 12.32) + (12.97 - 16.95)) / 2 = (4.03 - 3.98) / 2 = 0.05 / 2 = 0.025 \text{ segundos.}$$

La deriva entre los relojes es de 4.005+-0.025 segundos.

**4. PROBLEMA.** Se desea construir un servicio web basado en REST para gestionar un sistema de alquiler de bicicletas. El sistema se compone por una serie de estaciones que se identifican por un nombre. Cada estación tiene asignado un determinado número de bicicletas, y cada bicicleta se identifica por un número entero.

**4.1 (2 puntos) Indicar de forma razonada una posible URI (Uniform Resource Identifier) y el posible método HTTP (GET, POST, DELETE, PUT) a utilizar para ofrecer las siguientes funciones:**

- a) Obtener las bicicletas disponibles para ser alquiladas en la estación "atocha".

GET <http://www.mysite.com/estaciones/atocha>



**SISTEMAS INFORMÁTICOS II**

Asignatura..... Grupo..... **231**

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

Ejercicio del día..... **15 de marzo de 2017. Examen parcial.**

- b) Alquilar la bicicleta con número 73201234 localizada en la estación "bilbao".  
POST <http://www.mysite.com/estaciones/bilbao/73201234>
- c) Dar de baja la bicicleta con número 6323421 en la estación "sol".  
DELETE <http://www.mysite.com/estaciones/sol/6323421>
- d) Dar de alta la bicicleta con número 2313412 en la estación "goya".  
PUT <http://www.mysite.com/estaciones/goya/73201234>

**4.2 (1 punto) Utilizando XML, dar una posible representación de la bicicleta 2313412 localizada en la estación "goya" teniendo en cuenta que ésta debe incluir información sobre su ubicación y estado "alquilada" o "disponible".**

```
<bicicleta>
  <numero>2313412</numero>
  <estado>disponible</estado>
  <localizacion>http://www.mysite.com/estaciones/goya/</localizacion>
</bicicleta>
```

**4.2 (1 punto) Utilizando XML, dar una posible representación de la estación "goya" teniendo en cuenta que ésta debe incluir información sobre las bicicletas disponibles para alquilar en esta estación.**

```
<estacion>
  <nombre>atocha</nombre>
  <numero_bicicletas>2</numero_bicicletas>
  <bicicletas>
    <bicicleta href ="http://www.mysite.com/estaciones/goya/2313412">
      2313412
    </bicicleta>
    <bicicleta href ="http://www.mysite.com/estaciones/goya/343234178">
      343234178
    </bicicleta>
  </bicicletas>
</estacion>
```

