

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. (2 puntos) Definición de transparencia de datos. Enumera las tres posibles alternativas en el tipo de acuerdo a tomar para lograr la transparencia de datos.

Diapositiva 8 – Tema 1

2. (2 puntos) ¿Cuál es la funcionalidad del modelo de referencia de CORBA definido en la *Object Management Architecture*? Describir las características de los elementos que componen el modelo de referencia: *Object Request Broker*, *Common Object Services*, *Common Facilities*, *Domain Interfaces* y *Application Interfaces*.

Funcionalidad → diapositiva 43 – tema 1

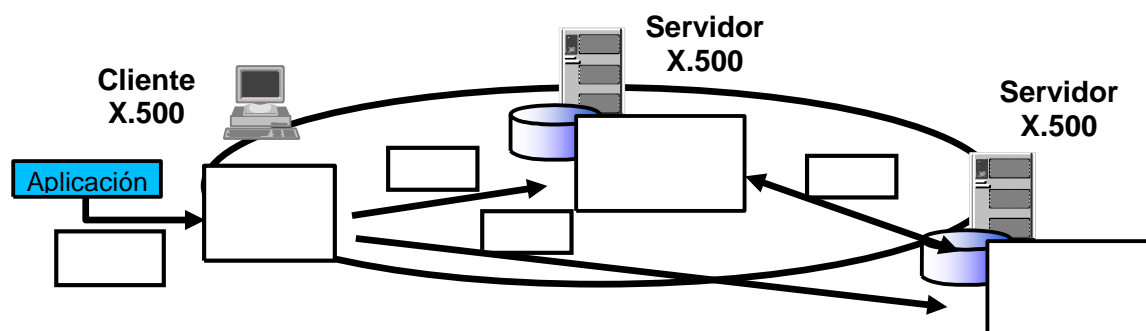
Elementos del modelo de referencia → diapositiva 47 – tema 1

3. (2 puntos) Definición del *Enterprise Service Bus* (ESB). ¿Cuáles son sus principales objetivos? ¿Y su aplicación principal? Enumera tres ventajas y tres inconvenientes del uso de ESB.

Definición, objetivos y aplicación: Diapositiva 75 – Tema 1

Ventajas e inconvenientes: Diapositiva 78 – Tema 1

4. (2 puntos) ¿Cuál es la funcionalidad principal de los servicios de directorio? ¿Qué tipo de transparencia garantizan? ¿Qué los diferencia de los servicios de nombres? Completar el siguiente diagrama de acuerdo a los elementos que componen en el estándar de directorios X.500.



Funcionalidad → Diapositiva 84 – Tema 1

Transparencia y diferencias con servicios de nombres → Diapositiva 80 – Tema 1

Estándar de directorios X.500 → Diapositiva 85 – Tema 1



Asignatura..... **SISTEMAS INFORMÁTICOS II** ..... Grupo..... **236 y 240** .....  
Apellidos ..... Nombre.....  
Ejercicio del día..... **10 de marzo de 2016. Examen parcial.** .....

5. **(2 puntos)** Pasos a seguir en el diseño orientado a rendimiento.

Diapositiva 9 – Tema 2



Asignatura..... **SISTEMAS INFORMÁTICOS II** ..... Grupo..... **236 y 240**  
 Apellidos ..... Nombre.....  
 Ejercicio del día..... **10 de marzo de 2016. Examen parcial.**.....

2 (6)	3 (4)	Total Problemas (10)

**2. PROBLEMA (6 puntos – 1.5 puntos por escenario).** Considerar los siguientes casos particulares de sistemas cliente / servidor:

- Servicio de monitorización de contaminación atmosférica.** Cada media hora, los clientes de una red homogénea (estaciones de control de calidad del aire) envían a un servidor central la información sobre el promedio las concentraciones en el aire de cuatro gases (dióxido de azufre, monóxido de carbono, monóxido de nitrógeno, dióxido de nitrógeno) durante la última media hora. El mensaje contiene además una marca de tiempo con la hora a la que se refieren las mediciones. El servidor registra estos datos en una base de datos para que puedan ser consultados en tiempo real y envía un mensaje de respuesta a los clientes indicando la correcta recepción del mensaje.
- Servicio de alertas bancarias.** Clientes (bancos) sobre arquitecturas heterogéneas solicitan a un servidor el envío de mensajes de texto y emails a los clientes del banco. Para ello, los clientes envían al servidor el texto del mensaje junto con un valor de prioridad del cliente (banca de particulares, banca personal, banca privada, banca de empresas) y un valor de prioridad del mensaje (ingreso de nómina, cobro de recibos, cancelación de préstamo, etc.). El servidor comienza a realizar sus envíos a las 6h de cada día, estableciendo el orden de envío en función de la prioridad del cliente y del mensaje.
- Servicio de organización de eventos.** Un servicio de organización de eventos actúa como elemento centralizador para la gestión de eventos y realización de reservas de recintos, hoteles, medios de transporte y restaurantes. Por tanto, el servicio de organización de eventos actúa como cliente, realizando peticiones a los servicios ofrecidos por diferentes recintos, hoteles, empresas de transporte y restaurantes, los cuales se implementan en arquitecturas y lenguajes heterogéneos (Java, C++ y Python) y el acceso a los mismos se realiza a través de una interfaz predefinida. La complejidad del proceso de desarrollo no supone un problema y se desearía que le tamaño de los mensajes fuese pequeño puesto que se espera una gran cantidad de peticiones.
- Servicio de localización de bicicletas.** Clientes de arquitecturas heterogéneas solicitan información sobre la localización de bicicletas libres del sistema de movilidad sostenible de un ayuntamiento. Los clientes también pueden reservar una bicicleta libre durante los próximos quince minutos con el fin de disponer del tiempo suficiente para llegar al punto de recogida. Estas reservas se pueden consultar, cancelar o modificar. Además, se espera que se faciliten nuevos servicios en el futuro. El acceso al servicio se realiza a través de una interfaz de programación predefinida. El coste de procesamiento de los mensajes no es un factor crítico ni tampoco el tamaño de los mensajes intercambiados. Los mensajes han de pasar a través de un corta-fuegos.

Para cada uno de ellos se pide:



**SISTEMAS INFORMÁTICOS II** 236 y 240  
 Asignatura..... Grupo.....  
 Apellidos..... Nombre.....  
 Ejercicio del día **10 de marzo de 2016. Examen parcial.**

- a) Elegir razonadamente el mecanismo de comunicación cliente-servidor (comunicación no orientada a conexión (UDP), comunicación orientada a conexión (TCP), servicios web basados en SOAP, CORBA o colas de mensajes) que sería aconsejable utilizar para conectar a ellos los sistemas clientes.
- b) Indicar, si es necesario, las funciones adicionales que habría que implementar sobre el protocolo elegido para garantizar que el *middleware* resultante garantizara la transparencia de datos del sistema distribuido.

**No se tendrán en cuenta respuestas sin justificación.**

### 1. Servicio de monitorización de contaminación atmosférica.

- a. El mecanismo más adecuado sería comunicación no orientada a conexión (UDP) debido a las siguientes razones:
  - Las interacciones serían acopladas en el tiempo (servidor y cliente conectados a la vez).
  - El cliente podría quedarse o no bloqueado a la espera de la respuesta del servidor.
  - Los clientes son homogéneos, por lo que no habría que hacer marshalling/unmarshalling de los datos.
  - La funcionalidad es muy elemental, no merece la pena usar CORBA o servicios web. La información a enviar es muy pequeña, sólo los cuatro valores de concentración y la marca de tiempo. La respuesta del servidor también es sencilla, indicando la correcta recepción del mensaje.
  - No merece la pena establecer una conexión para el envío de un mensaje pequeño cada media hora, por lo que se descarta TCP. Además, en caso de pérdida de un mensaje, éste se podría reenviar sin problema puesto que las interacciones son del tipo petición/respuesta e idempotentes, dado que los mensajes contienen la marca de tiempo
- b. No es necesario implementar ningún tipo de funcionalidad adicional para garantizar la transparencia de datos al tratarse de sistemas homogéneos.

### 2. Servicio de alertas bancarias.

- a. La solución más apropiada serían las colas de mensajes ya que:
  - Cliente y servidor no necesitan estar conectados en el mismo instante de tiempo (desacoplados en el tiempo).
  - El cliente no queda bloqueado tras la petición al servidor.
  - Además, las colas de mensajes hacen posible que las peticiones sean procesadas de acuerdo a ciertas prioridades (tipo de cliente y de alerta) y la temporización en el procesamiento de mensajes.
- b. La cola de mensajes debería encargarse de la traducción de datos para garantizar la transparencia de acceso puesto que se trata de sistemas heterogéneos.



Asignatura..... **SISTEMAS INFORMÁTICOS II** ..... Grupo..... **236 y 240**  
 Apellidos ..... Nombre.....  
 Ejercicio del día..... **10 de marzo de 2016. Examen parcial.**.....

### 3. Servicio de organización de eventos.

- a. La solución más adecuada sería CORBA por los siguientes motivos.
- Los dos sistemas tiene que estar conectados a la vez (acoplado en el tiempo) y el cliente quedaría bloqueado a la espera de la respuesta del servidor: las colas de mensajes pierden su atractivo.
  - El cliente (servicio de gestión de eventos) accede al servicio a través de una interfaz de programación definida.
  - Se esperaría que las peticiones incluyan cierta lógica de proceso compleja.
  - Usando WSDL y UDDI podríamos registrar y localizar de forma sencilla los nuevos servicios.
  - Es necesaria la traducción de datos (arquitecturas heterogéneas).
  - Todo esto hace indicar que lo más apropiado serían los servicios web o CORBA. Sin embargo, es crítico que los mensajes no sean muy pesados. Esto hace que los servicios web basados en SOAP (XML) estén en desventaja respecto a CORBA, teniendo en cuenta que la complejidad en el desarrollo no es inconveniente.
- b. Sería necesario llevar a cabo operaciones de marshalling/unmarshalling para garantizar la transparencia de acceso a datos. Sin embargo, al usar CORBA, este proceso sería transparente y no haría falta añadir funcionalidad adicional al middleware.

### 4. Servicio de localización de bicicletas.

- a. La solución más apropiada sería servicios web basados en SOAP dado que:
- Los dos sistemas tiene que estar conectados a la vez (acoplado en el tiempo) y el cliente quedaría bloqueado a la espera de la respuesta del servidor: las colas de mensajes pierden su atractivo.
  - Los clientes acceden al servicio a través de una interfaz de programación definida.
  - Se esperaría que las peticiones incluyan cierta lógica de proceso compleja.
  - Es necesaria la traducción de datos (arquitecturas heterogéneas).
  - Los mensajes deben pasar por un corta-fuegos, su tamaño no es muy importante, y en el futuro aparecerán nuevos servicios. Esto sugiere WS-SOAP, ya que pasaríamos a través del corta-fuegos, y usando WSDL y UDDI podríamos registrar y localizar de forma sencilla los nuevos servicios.
- b. Sería necesario llevar a cabo operaciones de marshalling/unmarshalling para garantizar la transparencia de acceso a datos. Sin embargo, al usar servicios web basados en SOAP, este proceso sería transparente y no haría falta añadir funcionalidad adicional al middleware.

**3. PROBLEMA (4 puntos)** Se quiere prestar un servicio que permita gestionar el préstamo de libros de la biblioteca de la Escuela Politécnica Superior utilizando servicios web REST. Dicho servicio



Asignatura..... **SISTEMAS INFORMÁTICOS II** ..... Grupo..... **236 y 240**  
Apellidos ..... Nombre.....  
Ejercicio del día..... **10 de marzo de 2016. Examen parcial.**

podrá ser útil para consultar información de un usuario (nombre, apellidos, tipo de usuario – profesor, estudiante, PAS–, posibles penalizaciones por demora), para consultar los libros prestados a un usuario, para consultar y cancelar las reservas de libros de un usuario, y para renovar los préstamos de un usuario. Cada usuario se identifica por su nombre y cada libro por un código numérico de cuatro cifras.

**3.1. (2.5 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) Actualizar la penalización por demora del usuario con nombre “Andrew Tanenbaum”.

URI: <http://www.biblos.eps.uam.es/usuarios/andrewtanenbaum>

Método: POST

- b) Indicar que el usuario “Andrew Tanenbaum” ha tomado prestado el libro con identificador 3141.

URI: <http://www.biblos.eps.uam.es/usuarios/andrewtanenbaum/prestamos/3141>

Método: PUT

- c) Renovar el préstamo del libro con identificador 9876 que el usuario “Andrew Tanenbaum” tiene prestado.

URI: <http://www.biblos.eps.uam.es/usuarios/andrewtanenbaum/prestamos/9876>

Método: POST

- d) Obtener las reservas del usuario “Andrew Tanenbaum”.

URI: <http://www.biblos.eps.uam.es/usuarios/andrewtanenbaum/reservas>

Método: GET

- e) Eliminar la reserva del libro con identificador 1234 de las reservas del usuario “Andrew Tanenbaum”.

URI: <http://www.biblos.eps.uam.es/usuarios/andrewtanenbaum/reservas/1234>

Método: DELETE

**3.2 (1 punto)** Dar un ejemplo razonado de una posible representación del usuario “Andrew Tanenbaum”.



Asignatura..... **SISTEMAS INFORMÁTICOS II** ..... Grupo..... **236 y 240**  
 Apellidos ..... Nombre.....  
 Ejercicio del día..... **10 de marzo de 2016. Examen parcial.**.....

La representación podría tener cualquier formato, pero debería incluir sus datos personales y enlaces las listas de préstamos y reservas. Elegimos XML.

```
<usuario>
  <nombre> Andrew Tanenbaum </nombre>
  <email>andrew.tanenbaum @uam.es </email>
  <categoria> profesor </categoria>
  <penalizacion>0</penalizacion>
<lista_prestados>      http://www.biblos.eps.uam.es/usuarios/andrewtanenbaum/prestamos
</lista_prestados>
  <lista_reservas>      http://www.biblos.eps.uam.es/usuarios/andrewtanenbaum/reservas
</lista_reservas>
</usuario>
```

**3.3 (0.5 puntos)** Dar un ejemplo razonado de una posible representación de la lista de libros prestados al usuario "Andrew Tanenbaum".

La representación podría tener cualquier formato, pero debería incluir referencias a los libros prestados. Elegimos XML.

```
<prestamos>
<item href="http://www.biblos.eps.uam.es/usuarios/andrewtanenbaum/prestamos/9876"> Coulouris,
George, Jean Dollimore, and Tim Kindberg. Sistemas distribuidos. </item>
<item href="http://www.biblos.eps.uam.es/usuarios/andrewtanenbaum/prestamos/7649">
Richardson, Leonard and Ruby, Sam. RESTful web services.</item>
</prestamos>
```