ARQUITECTURAS DISTRIBUIDAS

PROBLEMAS

1. **Feb 06**. Dado el siguiente fragmento de HTML3.2, indique si es o no conforme al estándar. En caso de no ser conforme, enumere cada uno de los errores presentes. (1.5 puntos)

   Notas:

   - Responda esta pregunta en el propio enunciado.
   - Sólo se considerarán correctas las respuestas acordes a la especificación SGML.
   - Cualquier respuesta errónea descontará en la puntuación del ejercicio.

```
<h1> TABLA 1 </H1 >

<table align="right">
<tr>
<td>
<a href="http://www.vim.org">
<img border="0" src="vim_created_blue.gif"
     height="31" width="88" odd="Pagina creada con VIm">
</a>
<td>
<a href="http://validator.w3.org/check/referer">
<img border="0" src="http://www.w3.org/Icons/valid-html32"
alt="Valid HTML 3.2!" height="31" width="88">
<tr>
<a src="http://www.mozilla.org">
<img border="0" src="firefox.gif" alt="Firefox">
</a>

</table>
```

2. **Feb 06**. Una empresa posee tres bases de datos distintas y deslocalizadas que ofrecen datos referentes al histórico de clientes (se supone que cuesta mucho actualizarlas y unirlas en una sola). Estos datos deben ser accesibles desde todos los puestos de la empresa y desde el exterior, ya que los empleados pueden trabajar desde casa o en el emplazamiento de los clientes. El problema surge debido a que no todos los empleados tienen acceso a los datos de todos los clientes, ya que hay clientes especiales que tratan únicamente con el gerente y/o director de la empresa.

   De esta forma se pide que resuelva el problema de acceso de los empleados a las tres bases de datos. Para ello el empleado ha de identificarse con login y password para la base de datos a la que desee acceder. El acceso a cada una de las bases de datos se realiza de la misma forma para todos los empleados y bases de datos, utilizando

un método .open., al igual que para terminar la sesión, se utiliza un método .exit.. Estos métodos ya están programados en lenguaje Java.

Una vez el empleado ha accedido a la base de datos puede solicitar datos de los clientes, indicando el nombre del cliente, el parámetro a acceder (nombre_empresa, dirección, teléfono, fecha_ultima_operacion, capital_pagado, capital_pendiente) y, en su caso, el valor del parámetro en caso de cambio, (por ejemplo, cuando ha efectuado una compra). El sistema devuelve si la operación ha tenido éxito (por ejemplo esto no ocurre si se accede a un cliente especial), y los valores leídos.

De esta forma se pide:

- Diseñe una interfaz IDL que permita las operaciones de entrada, acceso y salida de los empleados. (1 punto)
- Codifique en Java la/s clase/s que permita el acceso de un empleado a los datos de los cliente. Para ello dispone de los métodos ya programados: (1,5 puntos)

```
Boolean open(String login, String password);   //Entrar Base de datos
Boolean exit(void);                            //Salir Base de datos
String read(String name, String field);        //Leer dato
Boolean write(String name, String field, String value);
                                               //Actualizar dato.
```

3. Una inmobiliaria va a establecer una nueva sede en Cartagena. Para organizar la información de los pisos disponibles quieren usar XML. Las características que deben guardar *para cada piso* son las siguientes:

   - Información general del piso: Metros cuadrados, altura (nº piso), coste estimado, estado.
   - Para cada una de sus habitaciones: Tipo (COCINA, SALON, DORMITORIO, BAÑO, DESPACHO), metros cuadrados, número de ventanas interiores, número de ventanas exteriores y fotografías.

   Se pide que:

   a) Diseñe un DTD adecuado a las especificaciones del problema. **NOTA**: Para las fotografías debe indicar una URL con el link.

   b) Instancie un documento con 2 pisos.

4. **Jun 07**. Dado el siguiente fragmento de HTML3.2, indique si es o no conforme al estándar. En caso de no ser conforme, enumere cada uno de los errores presentes. (1.5 puntos) Notas:

   - Responda a la pregunta en el propio enunciado
   - Sólo se considerarán correctas las respuestas acordes a la especificación SGML.
   - Cualquier respuesta errónea descontará en la puntuación del ejercicio.

```
<H1>Presentación de redes inalámbricas
<TABLE WIDTH="50%" BORDER="1" CELLSPACING="large" CELLPADDING="2">
  <TR CAPTION="DAMOS SERVICIO">
    <TH COLSPAN="2" ALIGN="right">Servicio de Informática</TH>
    <TH>Redes y Comunicaciones</TH>
    <TH ROWSPAN="2">Técnicos:</TH>
  </TR>
  <TR >
     <TD>Amancio Redes</TD>
     <TD>Sofía Wifi</TD>
     <TD ALIGN="center">
     <IMG SRC="/graficos/bebe1.jpg" BORDER=0 TAG="Mi Foto" ALIGN="RIGHT">
     </TD>
  </TR>
</TABLE>
```

5. **Jun 07**. Un grupo de estudiantes de telemática ha desarrollado un rudimentario foro web (*www.forosteleco.com*) para debatir diversos temas. La página principal muestra una lista de temas hiperenlaces a distintos temas ("política", "deportes", "fiesta", etc.). Dentro de cada tema hay una lista de mensajes dejados por los usuarios. Los usuarios deben iniciar una sesión en el foro introduciendo su clave y nombre de usuario. Una vez iniciada la sesión, ésta durará hasta UNA SEMANA, pasado ese tiempo el usuario tendrá que volver a iniciar sesión. Además, cuando una sesión está iniciada, el foro muestra al usuario los temas en los que hay mensajes sin leer cuando accede a la página principal. Implemente usando PHP y COOKIES:

   a) (1 punto) Implemente la función *iniciarSesión()* que realiza las acciones necesarias para el establecimiento y control de sesiones. El nombre de usuario y la contraseña se comunica mediante el método POST en las variables *user* y *pass* respectivamente.

   b) (1 punto) Implemente la función *temaLeido()*. Esta función se ejecuta al pinchar en un tema y permitirá que el servidor informe al usuario de los temas sin leer cuando acceda a la página principal.

   c) (1 punto) Implemente la función *comprobarTemasConMensajesNoLeidos()*. Esta función se ejecuta en la página principal y comprueba los temas en los que hay mensajes sin leer. Como resultado mostrará el siguiente mensaje: "Hay mensajes sin leer en los temas: X, Y, Z, ..."

   NOTA: las siguientes funciones se suponen ya programadas y disponibles para su uso.

   - *bool comprobarClave(string usuario, string clave)*. Esta función devuelve "true"si la clave que se pasa como parámetro, coincide con la clave almacenada para el usuario que se pasa como parámetro.
   - *$nombreTema*. Esta variable almacena el nombre del tema que se está visitando en cada momento.
   - *int numeroMensajes(string tema)*. Esta función devuelve el número de mensajes que se han escrito para el tema que se pasa como parámetro.

6. La web de apuestas on-line TESACOLAPASTA.com desea guardar la información de las apuestas de fútbol mediante instancias en XML. Para ello, asociado a cada apostante, se guardará un BOLETO con cada una de las APUESTAS del mismo. En cada APUESTA el apostante debe decidir el resultado de tres partidos (indicando el número de goles que marcará cada equipo -*local* o *visitante*- en cada uno). En el BOLETO deben guardarse los datos del apostante (DNI y NOMBRE).

   Se le pide que:

   a) Escriba un DTD adaptado a las condiciones del problema.

   b) Escriba una instancia conteniendo una apuesta.

   c) Indique las tabla/s y estructura de las mismas que utilizaría para guardar la información de la aplicación en una BBDD. **Nota**: se necesitan dos tablas para resolver el problema.

   d) Codifique en PHP una función que devuelve una instancia XML con todas los boletos correspondientes a un apostante dado.

7. **Jun 07**. Se desea crear un servicio de distribución de noticias empleando CORBA. Para ello, los clientes pueden acceder a un servidor centralizado para ESCRIBIR (una noticia) o LEER (el bloque de todas las noticias enviadas). Asimismo, tras haberlas leido, los clientes pueden PUNTUAR individualmente las noticias entre 0 y 10 puntos. Cuando un cliente lee el bloque de noticias se le indican, para cada una, los votos de la misma y su puntuación máxima (si los hay). Nota: Considere una noticia como una cadena de texto.

   a) (1 punto) Escriba un IDL para el objeto servidor de tal aplicación.

   b) (1.5 puntos) Codifique la implementación de los métodos remotos correspondientes al IDL anterior.

8. **Sep 07**. Dado el siguiente fragmento de HTML3.2, indique si es o no conforme al estándar. En caso de no ser conforme, enumere cada uno de los errores presentes. (1.5 puntos)

```
<H1>Presentación Plan Estratégico </H1 align="center">
<TABLE WIDTH="50%" INBORDER="1" CELLSPACING="3" CELLPADDING="2">
<TR CAPTION="DAMOS SERVICIO">
   <TH COLSPAN="2" ALIGN="right">
   Servicio de Gestión de Proyectos</TH>
   <TH>Marketing y Desarrollo empresarial</TH>
   <TH ROWSPAN="2">Coordinadores Jefes:</TH>
</TR>
<TR  >
   <TD>Amancio Perez</TD>
   <TD>Caridad Nula</TD>
   <TD ALIGN="center">
```

```
   <IMG ORIG="/graficos/consumo.jpg"
      BORDER=0 ALT="Gráfico diferencial" ALIGN="RIGHT">
   </TD>
</TR>
</TABLE>
```

9. **Sep 07** Una tienda online (*www.tienda.com*) ha implementado su servicio de ventas mediante el típico *carrito de la compra*. El usuario, a medida que navega por su catálogo, añade los objetos que desea comprar a su carrito de la compra. Cuando ha elegido todo lo que desea, efectúa la compra. Cada vez que el usuario selecciona productos, el carrito de la compra los incorpora. Además, la tienda ofrece descuentos del 10 % del precio total de la compra si: a) el usuario compra algún *producto estrella* o b) el usuario gasta más de 300 euros. Estos descuentos no son acumulables. Para simplificar, se supone que el usuario sólo puede seleccionar una unidad de un producto y no puede volver a seleccionar un producto ya seleccionado. Implemente con PHP Y COOKIES las funciones necesarias para ofrecer este servicio. NOTA: NO SE LIMITE A ESCRIBIR CÓDIGO PHP, EXPLIQUE TANTO LA SOLUCIÓN GLOBAL QUE PROPONE COMO LA SOLUCION A CADA APARTADO.

   a) (1 punto) Implemente la función *seleccionarProducto()* que realiza las acciones necesarias para actualizar el carrito del usuario. Esta función recibe el código del producto mediante el método GET, en la variable *id*. El carrito de la compra durará lo que dure la sesión. Cuando el usuario selecciona un nuevo producto recibirá el mensaje "Ha seleccionado el producto Y cuyo precio es Z".

   b) (1 punto) Implemente la función *eliminarProducto()*. Esta función elimina el producto que se indica mediante el método GET en la variable *id*, como en el caso anterior, y muestra el mensaje “Producto X eliminado”.

   c) (1 punto) Implemente la función *validarCompra()*. Esta función se ejecuta cuando el usuario decide aceptar su carrito de la compra. Esta función mostrará un mensaje "Va a comprar los siguientes productos: X, Y, Z, etc., cuyo precio total es W". Es decir, una lista de los productos a comprar. Si el usuario se beneficia de algún descuento, mostrará además el mensaje "Dispone de un descuento del 10 %, con lo que el precio total es T". Donde debe sustituir las variables X,Y,Z,W,T por su valor correspondiente.

   NOTA: Tenga en cuenta que el usuario puede seleccionar más de un producto estrella. Las siguientes funciones ESTÁN A DISPOSICION DEL PROGRAMADOR Y PERMITEN OBTENER INFORMACIÓN DEL PRODUCTO:

   - string nombreCompletoProducto($idProducto)
   - double precioUnitario($idProducto)
   - bool esEstrella($idProducto)

10. **Sep 07**. Un periódico deportivo desea implementar un sistema distribuido para que los comentaristas puedan narrar los partidos de fútbol de modo sencillo. Para ello, han elegido implementar una solución basada en CORBA. Un servidor en la sede del diario recibirá los mensajes de la crónica de todos los partidos en juego y los irá almacenando junto con información de la hora a la que es recibido cada mensaje y el nombre del comentarista que lo ha incluido. Asimismo, en cualquier momento puede solicitarse al servidor la crónica completa hasta el momento actual de cualquier partido, para ser enviada a los clientes.

    En resumen, el servidor debe implementar las siguientes operaciones remotas:

    - COMENTARIO: añade un comentario en el servidor al partido indicado.
    - CRONICA: devuelve la crónica del partido indicado hasta el momento actual, es decir, *todos* los mensajes recibidos de los comentaristas, junto con la información de la hora y el nombre del comentarista que ha añadido cada uno de ellos.

    Se le solicita que:

    a) Escriba una interfaz IDL adecuada para el problema anterior. (1 punto).

    b) Codifique en Java, siguiendo la estructura de la interfaz IDL creada, el objeto de implementación. (1.5 puntos).

11. **Sep 07**. La firma de inversiones AKIMFORROYO desea crear diversas aplicaciones para gestionar las carteras de sus clientes. Para ello se ha optado por organizar la información de cada cartera mediante XML. Cada cartera contiene la información de las acciones poseídas actualmente por un cliente, junto con los datos personales del mismo. De cada grupo de acciones se guarda como información el nombre de la empresa (por ejemplo, “Apple Inc.”), el símbolo de la misma (p. ej., “AAPL”), el precio de compra de las acciones, el día de compra, el mercado de compra (uno de los tres siguientes: DOWJONES, NASDAQ o IBEX35) y el número de acciones. Notese que podrían existir grupos de acciones distintos de la misma empresa. De cada cliente se debe guardar su nombre, NIF y cuenta bancaria de intercambio de capitales. Se pide que:

    a) Diseñe el DTD que permita definir un lenguaje XML para contener la cartera de un cliente. (1 punto).

    b) Escriba una instancia del lenguaje anterior. (0.5 puntos).

12. Se desea desarrollar una aplicación para gestionar el trabajo en una empresa. La información se guardará en una BBDD MySQL. Se le pide que:

    a) Diseñe una tabla con la información de los empleados: DNI (sin letra), nombre, apellidos, dirección, edad título.

b) Implemente 4 funciones en PHP: *insertar, eliminar, modificar y seleccionar*. Todas reciben la información necesaria mediante el método GET (elija los nombres de variables que considere adecuados). Las función *seleccionar* recibe una cadena de búsqueda y el campo sobre el que se realiza la búsqueda o bien, si no se indica cadena de búsqueda, se muestra toda la tabla en la BBDD.

c) Diseñe una tabla adicional de tareas: nombre, contratista, tiempo y coste. Puede serle útil un campo de

d) Diseñe una tabla que relacione tareas con empleados. Un empleado puede tener más de una tarea asociada.

e) Repita el apartado b) para la tabla de tareas.

f) Implemente funciones que permitan mostrar las tareas asociadas a cada empleado o el empleado asociado a una tarea determinada.

g) Diseñe un DTD adecuado para mostrar la información de empleados en XML.

h) Implemente un script que exporte a XML la información de empleados, mediante las funciones DOM.

i) Diseñe un DTD adecuado para mostrar la información de tareas en XML.

j) Implemente un script que exporte a XML la información de tareas, mediante las funciones DOM.

k) Implemente un script que, dados un documento XML con tareas y otro con empleados, muestre por pantalla las tareas asociadas a cada empleado.

13. Se desea implementar una tienda electrónica, mediante una aplicación en PHP. La tienda cuenta con un catálogo de productos diferenciados en 3 categorías (Literatura, Música y Video), que se ha implementado sobre una BBDD MySQL. Se dispone de una BBDD llamada ”ad” y de un usuario llamado ”aduser” y clave ”ad”. Se pide que:

    a) Diseñe una tabla para la BBDD que recoja la información del catálogo. Para cada producto se guardará la siguiente información: precio, nombre, descripción breve, unidades disponibles y categoría. Puede serle útil un campo adicional de identificación.

    b) Cree un formulario HTML para introducir la información de cada producto. La información se enviará mediante el método POST a un script PHP llamado *procesarCat.php*. La categoría del producto se elige mediante una lista desplegable.

    c) Implemente una función en PHP que inserta un nuevo elemento en el catálogo. La información se recoge mediante el formulario anterior. El script muestra el mensaje *Nuevo producto añadido*.

    d) Cree un documento HTML que permite visualizar el catálogo. Se puede elegir la categoría que se quiere visualizar.

    e) Implemente la función *mostrarCatalogoAdmin($cat)* que muestra el catálogo según la categoría. El catálogo se muestra en forma de tabla. En cada fila de

la tabla, además de la información del catálogo, habrá un botón de selección (tipo radio). Al final de la página del catálogo habrá tres dos botones *Eliminar* y *Editar*. Además se incluirá un mini-formulario que permita cambiar de categoría. Decida e incluya en los formularios el método que utilizará para enviar la información y los script que la procesarán.

f) Implemente la una función en PHP que elimine un producto del catálogo y muestre *Producto eliminado*.

g) Implemente un formulario HTML que permita actualizar los campos de un producto del catálogo. Este formulario se mostrará cuando el usuario pinche el botón *Editar* del formulario mostrado en el apartado e). Para ello, el script debe buscar en la BBDD la información del producto que se desea editar. Hecho esto, se utiliza esa información para rellenar los campos que se le muestran al usuario en el formulario. Además se incluye un botón *Guardar*.

h) Implemente una función PHP que actualiza el valor de los campos de un producto según la información recibida del formulario del apartado anterior.

i) Implemente una formulario HTML y un script PHP que permita buscar y mostrar al usuario todos los productos de una determinada categoría cuyo precio sea inferior o superior a uno dado.

j) Diseñe una tabla adicional con información extra asociada a cada producto: Autor, Género y año de publicación.

k) Implemente una función que, dado un producto, muestre su información extra asociada.

l) Implemente un script que exporte la información del catálogo a un documento XML. Para ello: a) diseñe un DTD adecuado, teniendo en cuenta la información de las dos tablas; b) Utilice las funciones DOM para mostrar esa información en forma de XML.

14. **Feb 05**. Dado el siguiente fragmento de HTML3.2, indique si es o no conforme al estándar. En caso de no ser conforme, enumere cada uno de los errores presentes. (2 puntos)

    Notas:

    - Sólo se considerarán correctas las respuestas acordes a la especificación SGML.
    - Cualquier respuesta erronea descontará en la puntuación del ejercicio.

```
<h2>Enlaces de interes</h2>

<form>
<select>
<option> 1
<option value="2"> 2
</select>
</form>
```

```
<ol>
<li> <a href="Teoria">Teor&iacute;a y <a href="Apuntes"> apuntes </a>
     <a> </a>
<li> <a href="practicas"> Practicas </a>
<li> <a href="Boletines"> Boletines y Ejercicios Puntuables </a>
<li> <a href="ejemplos.html"> Ejemplos de arquitecturas distribuidas
     </a> </li>
</ol>

<url href="http://www.google.com">

<table align="right">
<tr> <img border="0" src="vim_created_blue.gif">
</table>
```

15. **Feb05**. La Administración Nacional de Loterías desea crear un sistema distribuido para la gestión de un nuevo tipo de apuestas. La administración quiere, por una parte, permitir la generación automática de boletos y por otra parte, facilitar la comprobación de los resultados de cada boleto.

    Los boletos de apuesta constarán de 7 números entre 1 y 53. Existen premios a los 7, 6 y 5 aciertos. El servicio de generación automática de boletos debe devolver al cliente una combinación. El servicio de comprobación debe indicar al cliente *cuantos* y *que* números ha acertado, a partir de los números del boleto y la fecha del sorteo.

    Para ello han decidido utilizar un servidor centralizado y terminales clientes en cada uno de los puntos de venta. El servicio se implementará usando Java y CORBA. Se le pide que:

    a) Diseñe una interfaz IDL adecuada a los requisitos del problema. (1.25 puntos)

    b) Realice las clases de implementación en Java. Suponga que dispone del siguiente objeto de Java ya implementado en el servidor. (1.25 puntos)

```
public class Sorteo {

        // Devuelve una apuesta aleatoria
        int[] Apuesta(void);

        // Devuelve números acertados
        int[] ComprobarApuesta(int [] apuesta, string fecha);
}
```

16. Feb 06. Un sitio de Internet (*www.widgets.com*) ofrece a sus usuarios una serie de pequeños programas para su descarga. Si el usuario se registra, le ofrece un servicio adicional de actualizaciones: cada vez que el usuario accede al sitio web, es informado sobre actualizaciones disponibles para los programas que el usuario ha descargado.

Implemente en PHP las acciones necesarias para que este mecanismo funcione, sin que sea necesario almacenar información en el servidor de *www.widgets.com*. Para ello:

a) Implemente un script PHP llamado *registro.php*. Este script se encarga de procesar los datos del usuario cuando rellena el formulario de registro, que se envían mediante el método GET. El nombre de usuario se envía en un campo llamado *name*. Es decir, realiza las acciones para que el servidor pueda saber que un usuario se ha registrado e imprime un mensaje de bienvenida al usuario.

b) Implemente la función *descarga(string programa, int version)*. Esta función PHP es llamada cada vez que el usuario descarga un nuevo programa. Recibe como parámetros el nombre del programa descargado y la versión del mismo. Esta función toma las acciones necesarias para que el servidor pueda saber los programas que el usuario ha descargado e imprime un mensaje por pantalla "Aquí tiene su programa".

c) Implemente la función *actualizaciones*(). Esta función PHP comprueba si existen actualizaciones de los programas que el usuario ha descargado. Para ello hace uso de otra función *int versión(programa, versionprevia)*, que recibe como parámetros el nombre del programa y la versión descargada por el usuario y devuelve 0 si no hay una versión nueva o un entero con el nuevo número de versión. NO DEBE IMPLEMENTAR ESTA ULTIMA FUNCION, SOLO UTILIZARLA SI LE ES NECESARIO. Por cada programa en el que exista una nueva versión, se escribe un mensaje que dice "La versión X del programa Y está disponible para descargar".

17. Feb 06. Una web comercial (*www.musicapami.com*) permite descargar gratuitamente hasta 10 Mbytes durante una semana para que los usuarios comprueben la calidad de sus productos. Si el usuario compra algún producto de la web, se le pemite comenzar a descargar desde cero. Implemente en PHP las acciones necesarias para que este mecanismo funcione. Para ello:

a) Implemente la función *descarga(string file, int kbytes)*. Esta función se encuentra en la página *www.musicapami.com/descarga.php* a la cual es dirigido un usuario cuando quiere descargar un archivo. La función recibe como parámetro el nombre y el nombre y el tamaño en Kbytes del archivo que pretende descargar el usuario. Esta función comprueba si el usuario ha superado el límite de descargas, en cuyo caso muestra un mensaje del tipo *"Lo siento. Ha superado el límite. Espere un tiempo o compre uno de nuestros productos"*. En caso contrario simplemente muestra un mensaje *"Su archivo. Ya lleva descargados X bytes"*.

b) Implemente la función *comprar()*. Esta función se encuentra en la página *www.musicapami.com/compra.php*. Esta función lemuestra al usuario unmensaje *"Gracias por la compra. Ahora puede disfrutar de otros 10 Mbytes de descargas"*. Y toma las acciones necesarias para que el usuario pueda seguir descargando.