

# Examen de Software de Comunicaciones

5º Ingeniero de Telecomunicación

21 de enero de 2009

Departamento de Ingeniería Telemática  
Universidad Carlos III de Madrid

Nombre: \_\_\_\_\_

Duración: 2,5 horas

Puntuación: 8 puntos (+ 2 puntos por la presentación de tecnologías Web)

## Parte 1: test

Duración: 1 hora

Puntuación: 4 puntos

No se permite el uso de libros ni apuntes.

### Instrucciones:

- Para cada pregunta hay **una y sólo una** respuesta correcta
- Marque la respuesta correcta rodeando la letra de la respuesta con un círculo
- En caso de que se equivoque en una pregunta, tache toda la columna de las letras de las respuestas de la pregunta y escriba una nueva columna de letras al lado
- Cada **tres respuestas incorrectas** se resta **una buena** y las no contestadas ni suben ni bajan

Bien	Mal	En blanco	TOTAL

Nombre: \_\_\_\_\_

Duración: 2,5 horas

Puntuación 8 puntos (+ 2 puntos por la presentación de tecnologías Web)

**Parte 2:** problemas

Duración: 1,5 horas

Puntuación: 4 puntos

Conteste a **dos de las siguientes tres problemas**, cada problema en una hoja separada. No se permite el uso de libros ni apuntes.

**Problema 1 (2 puntos)**

(a) La base de datos de un servicio de actividades deportivas de una universidad contiene las siguientes tablas (este ejemplo debería serle familiar de la práctica):

Cientes

Login	Password	Nombre	Apellido	Direccion	Telefono

Actividades

ID	Nombre	Descripcion	Fecha_Inicio	Precio	Nombre_Pavellon	Plazas

Pavellones

Nombre	Localidad

Inscripciones

Login_Cliente	ID_Actividad

Proporcione las sentencias SQL que implementan las siguientes instrucciones:

- Liste el login, nombre y apellido de cada cliente junto con el coste total de todas las actividades en las que cada uno de ellos está inscrito.
- Liste los nombres de todas las actividades que tienen lugar en Leganés y que tienen plazas libres.
- Modifique el precio de algunas actividades, restando 10 Euros del precio de aquellas que tienen lugar en Leganés y que cuestan más de 110 Euros.
- Borre las actividades en las que no hay ningún cliente inscrito.

Suponga que "Leganés" es el nombre de una localidad utilizado en la tabla `Pavellones` y que la columna `Plazas` de la tabla `Actividades` contiene el máximo número de clientes que

Nombre: \_\_\_\_\_

pueden estar inscritos en la actividad. Obsérvese que, al contrario de la situación en la práctica, la tabla `Actividades` **no** tiene una columna que contenga el número de plazas libres.

- (b) Estudie el diagrama de clases UML que aparece en la Figura 1.
- Describa en lenguaje natural el dominio modelado por esta especificación UML; si no tiene tiempo para proporcionar una descripción exhaustiva, al menos ilustre cada uno de los distintos elementos sintácticos que en ella aparece.
  - ¿Qué otra información podría haberse proporcionado dentro de las cajas?
  - ¿Qué tiene de particular la caja que contiene el texto "IntervaloTiempo"?

## Problema 2 (2 puntos)

El consorcio del próximo año santo Xacobeo santiagués desea utilizar tecnología Java EE en sus servidores corporativos y le solicitan su asesoramiento sobre un componente, AÑOSANTO, que está desarrollando una compañía auxiliar. Dicho componente consta de dos interfaces Java que se **detallan** a continuación:

### Interfaz remota

```
package swc.xacobeo;
public interface AnhoSanto extends javax.ejb.XXX_1 {
    public Description getDescription() throws XXX_2Exception;
    public int getAnho() throws XXX_2Exception;
}
```

### Interfaz home

```
package swc.xacobeo;
public interface AnhoSantoHome extends javax.ejb.XXX_3 {
    public AnhoSanto create(Description desc, int anho) throws XXX_4,
        XXX_2Exception;
    public AnhoSanto findByPrimaryKey(int anho) throws XXX_5, XXX_2Exception;
}
```

Conteste a las siguientes cuestiones:

- (0,25 puntos)** Viendo las interfaces, ¿ante qué tipo de EJB cree que nos encontramos? Justifique su respuesta de forma argumentada.
- (0,25 puntos)** Teniendo en cuenta que dicho componente accede a cierto soporte persistente, describa el esquema mínimo de la base de datos necesario para que el componente pueda funcionar.
- (0,1 puntos cada uno)** Teniendo en cuenta sus conocimientos, diga cuál es el valor de los cinco identificadores marcados con `XXX_#` en el código.
- (1 punto)** Implemente el código de la clase Bean teniendo en cuenta que es de tipo CMP. Acuérdesse de diferenciar entre (i) los métodos de acceso a datos persistentes de la base de datos, (ii) los de la lógica de negocio, (iii) los relacionados con la creación del componente [PISTA: hay 2] y (iv) los de retrollamada (*callback*) [PISTA: hay 7].

## Problema 3 (2 puntos)

El código del método `doGet` de un servlet Java que aparece a continuación procesa peticiones del cliente de añadir o borrar un libro (clase `Book`) de una lista de libros (clase `BookList`), o editar los detalles de un libro de la lista. En la petición, el cliente manda el `id` del libro y el

Nombre: \_\_\_\_\_

nombre de la operación que se llevará a cabo. En el caso de la adición o la edición, los detalles de la operación se llevan a cabo a través de otras páginas Web. La lista de libros está accesible mientras dura una sesión y se manipula mediante los métodos de la clase `BookListManager`. Responda a las preguntas que aparecen después del código. Note que ciertas partes del código han sido reemplazadas por el texto `ClassX`, `objectX` y `methodX`.

```
public void doGet(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
```

```
    Class1 object3 = object1.method1();
    String action = object1.method2("operation");
    BookList mylist = (BookList) object3.method4("booklist");
    if (mylist == null) {
        mylist = new BookList();
        object3.method5("booklist", mylist);
    }

    BookListManager manager = new BookListManager();

    long id = 0;
    try{
        id = Long.parseLong(object1.method2("id"));
    }catch(NumberFormatException e){}

    if(action.equals("add")){
        RequestDispatcher dispatcher = method6.method7("/jsp/bookAdd.jsp");
        dispatcher.method8(object1, object2);

    }else if(action.equals("edit")){
        Book book = manager.findBookById(id, mylist);
        object1.method3("book", book);
        RequestDispatcher dispatcher = method6.method7("/jsp/bookEdit.jsp");
        dispatcher.method8(object1, object2);

    }else if(action.equals("delete")){
        manager.deleteBookById(id, mylist);
        object2.sendRedirect(object1.getContextPath() + "/bookList");
    }
}
```

- Explique brevemente el papel que desempeñan los dos parámetros de los métodos `doPost` y `doGet` en el caso general, esto es, los objetos de las clases `HttpServletRequest` y `HttpServletResponse` (en este ejemplo se denominan `request` y `response` respectivamente).
- Proporcione los nombres de la clase `Class1`, los objetos `object1` y `object2`, los métodos `method1`, `method2`, `method3`, `method4`, `method5`, `method6`, `method7` y `method8`, y el nombre utilizado por convención para `object3`. Si no se acuerda del nombre de un método, en vez de dar su nombre, explique brevemente lo que hace.
- Hasta donde se puede ver, ¿cumple la aplicación de la que forma parte este código con el patrón de diseño MVC? Debe justificar (brevemente) su respuesta. Explique brevemente cómo modificaría este código para tratar el caso en que el cliente elige editar un libro pero proporciona un `id` que no se corresponde con ningún libro de la lista (puede suponer que `findBookById` devuelve `null` en este caso).

Nombre: \_\_\_\_\_

- (d) ¿Cuál es la diferencia entre el método `sendRedirect`, utilizado en el tratamiento de la acción "delete", y el `method8`, utilizado en el tratamiento de las otras dos acciones? ¿Cuál sería el resultado de sustituir las dos últimas líneas del tratamiento de la acción "edit" por una llamada al método `sendRedirect` como la del tratamiento de la acción "delete" (pero con parámetro `/jsp/bookEdit.jsp`)? ¿Considera que el uso que se hace de `sendRedirect` en este código un uso adecuado de este método? Justifique su respuesta.