

**Laboratorio de Aplicaciones Telemáticas**  
**Ingeniería Técnica de Telecomunicación**  
**Especialidad en Telemática**

Escuela Politécnica Superior. Universidad Carlos III de Madrid.  
Leganés, a 10 de Febrero de 2005.

**Duración de la prueba:** 2h.

Lea las preguntas con atención y sea concreto en sus respuestas. Si es necesario realizar alguna suposición, indíquelo claramente.

*Usted está desarrollando una aplicación denominada **ReservasVuelos** para gestionar reservas de vuelos de aviones. La aplicación se realizará en dos modalidades: una versión swing para las agencias de viajes y una versión web para los clientes finales.*

*Los clientes finales acceden a una primera página segura en la que se dan de alta o se identifican con su nombre de usuario y su clave (sin utilizar autenticación de HTTP ni SSL) y posteriormente son redirigidos a una página en la que pueden ver sus reservas, anularlas, o hacer nuevas reservas, indicando el número de vuelo, la fecha de salida, y la categoría (económica o preferente).*

*Por su parte las agencias de viajes acceden a una ventana que les permite indicar al cliente los vuelos existentes para realizar una reserva, y de otra ventana en la que pueden modificar las reservas efectuadas desde la agencia.*

1. La aplicación web requiere guardar una serie de variables de sesión. Los usuarios potenciales utilizan navegadores que no permiten la utilización de cookies, y por su parte el diseñador de la aplicación no quiere que se utilicen URLs específicas para cada cliente, ni las sesiones de SSL. ¿Cómo se implementaría el mecanismo de sesiones? (1 punto)

**Respuesta:**

Dadas las restricciones impuestas en el enunciado la única opción posible sería utilizar campos ocultos en los formularios. De esta forma todos los servlets y JSPs que devolvieran una página operativa al cliente, después que se hubiese registrado incluirán los campos de usuario y contraseña. Esto no plantea problemas de seguridad, puesto que las páginas se acceden en una sesión SSL, pero si se quisiera incrementar el nivel de seguridad para evitar que alguien deje un navegador abierto y otra persona pueda ver el fuente de la página, se debería implementar un mecanismo de sesiones paralelo al de java, utilizando retos y codificando un campo con un reto distinto en cada página.

2. Escriba el formulario (no es necesario escribir la página XHTML completa) para enviar los datos de creación de una reserva de la aplicación de **ReservasVuelos**, de forma que viajen codificados en la URI. Indique el método y la codificación HTTP de la petición. (1 punto)

**Respuesta:**

```
<form action='ReservasVuelos'  
      method='GET'  
      enctype='application/x-form-urlencoded'>  
  <input type='hidden' name='login' value='pepe' />  
  <input type='hidden' name='clave' value='miclave' />  
  
  Vuelo: <input type='text' name='vuelo' />  
  Fecha salida (dd/mm/aaaa): <input type='text' name='fecha' />  
  Hora salida (hh:mm): <input type='text' name='hora' />  
  Clase: <select name='clase'>  
    <option name='clase' value='pre'>preferente</option>  
    <option name='clase' value='eco'>económica</option>  
  </select>  
  Tarjeta (xxxx-xxxx-xxxx):<input type='text' name='tarjeta' />  
</form>
```

La codificación de los datos sería:

```
GET /ReservasVuelos?login=pepe&clave=miclave&...&tarjeta=.. HTTP/1.0
```

*Para el desarrollo de la aplicación se dispone de unas tablas en una base de datos que contienen:*

- *los vuelos ofrecidos por la compañía. Indexados por el número de vuelo y conteniendo información de origen y destino, fechas de salida y llegada, periodicidad del vuelo (diario, semanal o mensual), plazas preferentes y económicas no reservadas, y las tarifas.*
  - *las reservas efectuadas hasta la fecha, con el número de vuelo, la información del cliente (nombre y DNI/pasaporte), y el tipo de pago (agencia o tarjeta).*
3. Indique razonadamente qué campos faltan en las tablas anteriores y qué tablas es necesario añadir. Indique al menos una tabla y dos campos. (1 punto)

**Respuesta:**

En la tabla de reservas es necesario indicar el estado de la reserva (anulada o vigente), el localizador de la reserva, si una reserva pertenece a una agencia, el id de la agencia, y si la ha hecho directamente el cliente, el número de tarjeta, la fecha y hora de salida del vuelo, y el DNI/pasaporte del cliente.

Es necesario incorporar la tabla de clientes con su login, nombre, clave y DNI/pasaporte.

Es necesario disponer de una tabla de agencias

Es interesante disponer de una tabla de ciudades que asigne un identificador a cada aeropuerto que pueda ser origen/destino de un vuelo.

4. Indique las clases comunes a la versión swing y la versión web que utilizará para realizar la aplicación. Enumere las propiedades y los métodos de cada una con sus parámetros. (2 puntos)

**Respuesta:**

Utilizaremos un bean para cada una de las tablas: `BeanUsuario`, `BeanVuelo`, `BeanReserva`. Los beans tendrán los métodos `set/get(is)` para cada propiedad, un constructor por defecto que las pone a null ó 0, y un constructor con parámetros. Las propiedades serían:

- `BeanUsuario`: `String login`, `String nombre`, `String clave`, `String númeroDocumento`, `String error` .
- `BeanVuelo`: `int númeroVuelo`, `int origen`, `int destino`, `Date fechaHoraSalida`, `Date fechaHoraLlegada`, `int plazasPre`, `int plazasEco`, `float precioPre`, `float precioEco`, `String error`.
- `BeanReserva`: `int localizador`, `int númeroVuelo`, `Date fechaSalida`, `int login`, `boolean agencia`, `int idAgencia`, `String númeroTarjeta`, `String clase`, `boolean anulada`, `String error`.

Además es muy conveniente utilizar una clase que recubra la base de datos y nos ofrezca las operaciones que vamos a necesitar para realizar las reservas, anulaciones, listar vuelos, listar reservas de usuario o agencias, listar aeropuertos, etc. Además en caso de encontrar un error lo introducirán en la propiedad `error` del bean que se pasa cómo parámetro.

```
class ConexionBD{
    private java.sql.Connection con= null;
    private java.sql.Statement st= null;

    \\ Si el objeto con es null se intenta conectar a la BD en
    \\ caso contrario no hace nada. Éste método es invocado en la
    \\ inicialización de los servlets
    void conectar(String dbURL, String user, string passwd)

    \\ Devuelve true si el cliente existe en la BD y actualiza el bean de
    \\ usuario con su DNI/pasaporte y su nombre
    boolean esCliente(BeansUsuario u);

    \\ Devuelve true o false si consigue realizar la reserva indicada en el
    \\ bean. Actualiza algunas de las propiedades del bean, como el
    \\ localizador de reserva
    boolean reservar(BeansReserva br);

    \\ Devuelven si existe el vuelo a la fecha y hora indicada teniendo en
    \\ cuenta la periodicidad de los vuelos
    boolean vueloExiste(BeansVuelo v);
```

```

boolean vueloExiste(BeanReserva br);

\\ Devuelve si hay plazas disponibles en el vuelo indicado
boolean hayPlazas(BeanReserva br);

\\ Anula una reserva
boolean anulareserva(BeanReserva br);

\\ Devuelve un vector de beans de vuelos para la fecha de salida pedida
Vector vuelosExistentes(Date fecha);

\\ Devuelve un Vector con los aeropuertos
Vector aeropuertosExistentes();

\\ Este método es invocado en la finalización de los servlets
void desconectar();
}

```

5. ¿Qué variables es necesario almacenar en la sesión de los clientes web?  
 ¿Qué variables es necesario almacenar en el contexto? (1 punto).

**Respuesta:**

No se puede almacenar nada en la sesión puesto que no existen sesiones según el apartado 1. Si se hiciera un mecanismo de sesiones ad-hoc, se almacenaría el bean de usuario y el de reserva.

En el contexto es necesario tener una referencia al objeto conexionBD.

6. Escriba el método `doGet` del servlet `RealizarReserva` que recoge los datos de reserva del cliente, realiza la reserva en la base de datos y pasa el control a `MostrarReservas.jsp` que permite mostrar la información de una reserva de un cliente. En caso de que no se pueda realizar la reserva, se pasa el control a `Error.jsp`. (2 puntos)

**Respuesta:**

```

public void doGet(HttpServletRequestRequest req,
                  HttpServletResponse res)
                  throws IOException, ServletException{
String númeroVuelo= req.getParameter('vuelo');
String fecha= req.getParameter('fechaSalida');
String hora= req.getParameter('hora');
SimpleDateFormat sdf= new SimpleDateFormat('ddmmyyyyHH:MM');
Date fechaSalida= sdf.parse(fecha+hora,0);
String login= req.getParameter('login');
String clave= req.getParameter('clave');
String clase= req.getParameter('clase');

```

```

String númeroTarjeta= req.getParameter("tarjeta");
String nombre= null;
String númeroDocumento= null;
String error= null;
BeanUsuario u= new
    BeanUsuario(login, nombre, clave, númeroDocumento, error);
int localizador= 0;
boolean isAgencia=false;
int idAgencia= 0;
boolean anulada= false;
BeanReserva br= new
    BeanReserva(localizador, númeroVuelo, fechaSalida,
        login, isAgencia, idAgencia, númeroTarjeta,
        clase, anulada, error);
ServletContext sc= ServletConfig.getServletContext();
RequestDispatcher rd=null;

con= (ConexionBD) sc.getAttribute("con");
if (con.esCliente(u) && con.vueloExiste(br) &&
    con.hayPlazas(br) && con.reservar(br))
    rd= sc.getRequestDispatcher("/MostrarReserva.jsp");
else
    rd= sc.getRequestDispatcher("/Error.jsp");

req.setAttribute("BeanUsuario", u);
req.setAttribute("BeanReserva", br);
rd.forward();
}

```

7. Suponga que para la versión swing, la clase equivalente a **RealizarReserva** de la versión web tiene una caja combo para seleccionar el número de vuelo y que cada vez que se selecciona un vuelo distinto, el botón **Reservar** se activa o desactiva dependiendo de si hay plazas o no en la categoría elegida del vuelo. ¿Qué método utilizaría para hacer dichos cambios en el botón? Proporcione el código del mismo. (1 punto)

**Respuesta:**

En el método `itemStateChanged` o en el `actionPerformed` del `JComboBox`.

\\ Suponemos que `reserva` es un objeto de tipo `BeanReserva`  
 \\ que se va actualizando a medida que el usuario interactúa  
 \\ con el interfaz.

```

void actionPerformed(ActionEvent e){
    if (e.getSource() == micombo){
        String
        vuelo= micombo.getSelectedItem();
    }
}

```

```
    reserva.setVuelo(vuelo);
    if (hayPlazas(reserva))
        botónReserva.setEnabled(true);
    else
        botónReserva.setEnabled(false);
}
..
}
```

8. En el análisis de la aplicación swing se detecta que las consultas que las agencias realizan a la base de datos pueden ser costosas en tiempo ralentizando en exceso el interfaz gráfico. ¿Qué solución propone para evitar bloquear la interfaz gráfica mientras se está realizando la consulta? Obtenida la información, ¿cómo haría para actualizar el interfaz gráfico de manera consistente? (1 punto)

**Respuesta:**

Se puede separar el acceso a la base de datos en un hilo diferente, y utilizar el método `invokeLater(Runnable)` de las utilidades de Swing para actualizar el interfaz gráfico con el resultado.