



Laboratorio de Aplicaciones Telemáticas Ingeniería Técnica de Telecomunicación – Telemática

Escuela Politécnica Superior. Universidad Carlos III de Madrid.
Leganés, a 3 de Febrero de 2003.
Duración de la prueba: 2h 30min

Cuestión 1 (1 pto.)

Existen múltiples tecnologías cliente-servidor. En cada una de ellas la ejecución del código que lo compone está distribuido entre el cliente (navegador Web) y el servidor.

1. Clasifica las siguientes tecnologías según el lugar donde ejecuta su código o donde están las rutinas para tratarlo (*cliente / servidor / cliente y servidor*):
 - a) XHTML
 - b) HTTP
 - c) Servlets
 - d) JSPs
 - e) Javascript
 - f) Scriptlets
 - g) Java Beans
 - h) Applets
 - i) Plugin

La empresa ACME S.A. ha desarrollado un novedoso y complejo sistema de codificación de señales de audio y vídeo, que reduce espectacularmente su tasa de transmisión, aunque haciendo un uso bastante elevado de recursos de CPU y memoria. El desarrollo es aprovechado para iniciar un negocio de transmisión de vídeo bajo demanda, y se ha programado un decodificador, en lenguaje C, con versiones para las arquitecturas más habituales. Para facilitar el acceso, pretenden que el usuario pueda comprar y ver los vídeos desde un navegador web. Por imposición del departamento de marketing, el plazo disponible para esta tarea es muy limitado.

2. De acuerdo con las restricciones anteriores, razona qué tecnología de programación web, de entre las mencionadas anteriormente, resultaría más adecuada para desarrollar el decodificador y su interfaz integrada en el navegador. Indica claramente el porqué.

Cuestión 2 (2 ptos.)

La división de aplicaciones móviles de la empresa ACME S.A. desea desarrollar un juego sencillo de aventura gráfica. El jugador maneja a un personaje con varias propiedades: puntos de vida (valor entre 0 y 100), ubicación (número de escenario) y bolsa de objetos (puede contener pócimas, mapas, llaves, armas, etc.) En función del contenido de la bolsa de objetos y de su ubicación, al jugador se le presentan o no determinadas opciones, imágenes, etc.

La aplicación debe cumplir los siguientes requisitos:

- El jugador interactúa con un navegador con soporte para XHTML Basic y cookies.
- La aplicación se ejecuta siempre a través de la URL *http://acme.com/aventura*.
- No se almacena ninguna información de sesión en el servidor, y ésta se gestiona mediante cookies.

En un momento dado el personaje está en el escenario 5 con 49 puntos de vida. Tiene en su bolsa tres pócimas (verde, azul y ámbar) y una espada. Entre otros, el documento representado en el móvil contiene el siguiente formulario:

```

<form action='aventura' method='get'
  enctype='application/x-www-form-urlencoded'>
  <p>
    Utilizar pócima (necesita un conjuro).
  </p>
  <p>
    Conjuro: <input type='text' name='conjuro' id='conjuro' />
  </p>
  <p>
    Pócima:
    <input type='checkbox' name='tipo' value='verde' />verde
    <input type='checkbox' name='tipo' value='azul' />roja
    <input type='checkbox' name='tipo' value='ámbar' />azul
  </p>
  <p>
    <input type='submit' value='Usar pócimas' />
  </p>
</form>

```

El usuario selecciona las pócimas verde y ámbar, conjuro “abracadabra” y envía el formulario.

1. Diseña un formato de cookies para esta aplicación. Descríbelo muy brevemente.
2. Escribe la petición HTTP 1.1 generada por el navegador y la respuesta de éste. Escribe en ambos casos la primera línea, las cabeceras más relevantes, y el cuerpo del mensaje (por simplicidad, pon “cuerpo” como valor del cuerpo de la respuesta). Se proporcionan pistas sobre el formato de un mensaje HTTP al final de la cuestión.
3. El formulario anterior se envía a una aplicación CGI, que funciona en Linux con un servidor Apache. Indica cómo el servidor le comunica a la CGI las cookies enviadas por el cliente.
4. Si desde el formulario hubiese que subir un fichero al servidor, ¿qué valores de método y codificación utilizarías? Razona la respuesta.

Nota:

- El formato de la línea de solicitud es “método uri HTTP/1.1”
- El formato de la línea de estado de una respuesta es “HTTP/1.1 estado frase-explicativa”. El estado OK tiene código 200.
- Las cabeceras HTTP relevantes para este ejercicio son: Content-Type, Content-Length, Cookie, Set-Cookie y Host.
- Puedes especificar varias cookies siguiendo el formato del siguiente ejemplo: “nombre1=valor1,nombre2=valor2,...”

Cuestión 3 (1 pts.)

Supón que se define una extensión a HTTP con tres nuevos métodos para facilitar el soporte de transacciones (TSTART, TCOMMIT y TCANCEL). Se desarrolla un nuevo servlet genérico con soporte para estos métodos:

```
public class TransactionServlet extends HttpServlet ...
```

Gracias a estos métodos, se pueden escribir servlets como el siguiente:

```

public class MiServlet extends TransactionServlet {
  ...
  public doTStart(HttpServletRequest req, HttpServletResponse res) {...}
  public doTCommit(HttpServletRequest req, HttpServletResponse res) {...}
  public doTCancel(HttpServletRequest req, HttpServletResponse res) {...}
}

```

1. ¿Qué método es necesario reescribir en la clase *TransactionServlet* para que, cuando llegue una petición, se ejecute el método adecuado en el servlet *MiServlet*? Describe qué modificaciones habría que hacer con respecto al mismo método de *HttpServlet*.

Bajo ciertas circunstancias, la operación TCOMMIT no puede ser ejecutada con éxito, y es necesario cancelar la transacción (método TCANCEL). Para ello, desde el método *doTCommit* se puede invocar al método *TCancel*, pero es necesario comunicarle a éste un código explicativo del error encontrado. Los diseñadores han optado por introducir este código como atributo de instancia de *MiServlet*, ya que de esta forma será visible desde ambos métodos.

2. ¿Es correcto el diseño anterior? Razona clara y brevemente por qué. Si no es correcto, propón un mecanismo alternativo para hacerlo.

Cuestión 4 (1 pto.)

Responde las siguientes preguntas sobre el uso de Servlets:

1. Un determinado servlet devuelve un fichero de texto en el cuerpo de la respuesta HTTP (tipo MIME *text/plain*) con el texto "Eureka". Describe clara y brevemente la secuencia de acciones que debe realizar el servlet para establecer el cuerpo y el tipo MIME de la respuesta.
2. En un servlet se toma la decisión de que una determinada petición debe ser atendida por otro servlet más adecuado. Indica brevemente las diferencias que hay entre utilizar `sendRedirect()` y `forward()` para ello.

Cuestión 5 (1 pto.)

El la práctica de la AgendaWeb la aplicación debía mantener cierta información además de la almacenada en la Base de Datos. Dicha información debía mantenerse en la forma de JavaBeans para permitir su intercambio entre los elementos de la aplicación. Indique que información contenían los JavaBeans y el contexto (aplicación/sesión/petición) de los mismos que empleaba su aplicación.

Cuestión 6 (1 pto.)

Responda a las siguientes cuestiones referidas al uso de JSPs y JavaBeans:

1. Describa brevemente las tres operaciones definidas en la especificación de las JSP con las que es posible definir y manipular JavaBeans y sus atributos más habituales.
2. Diseñe un sencillo JavaBean con una sólo propiedad de lectura/escritura denominada `Quantity` y que tiene como valor por defecto 1.

Cuestión 7 (1 pto.)

¿Existe alguna forma de que un diseñador Web pueda configurar el comportamiento o la apariencia de un applet sin necesidad de modificar el código fuente del mismo? Si ha utilizado esta técnica en la práctica de Applets, descríbala brevemente e indique qué información era configurable en dicha práctica.

Cuestión 8 (1 pto.)

Responda a las siguientes cuestiones referidas a la práctica de AgendaSwing:

1. Describa brevemente el método `actionPerformed()` de la interfaz `ActionListener` y bajo qué circunstancias se puede ejecutar dicho método en la práctica.
2. ¿Qué patrón de diseño emplea el método citado en el apartado anterior? Indique el nombre del patrón y cómo debe implementarse en Java.

Cuestión 9 (1 pto.)

Describa brevemente los modelos de la clase `JTable` que ha empleado en la práctica de AgendaSwing. Indique su propósito general, si ha sido necesario extender el modelo por defecto y al menos una operación de cada uno de ellos.