

# Ejercicios de cabeceras y códigos

## Índice

1 Distinguir el navegador.....	2
2 Redirecciones con retardo.....	2
3 (*) Loggear variables CGI.....	3

## 1. Distinguir el navegador

Muchas veces, cuando escribamos una aplicación Web, nos va a interesar poder distinguir el tipo de navegador que está utilizando el cliente, porque en función del mismo se podrán hacer o no determinadas acciones. Por ejemplo, el código Javascript que se emplea en un navegador Internet Explorer es diferente a veces del que se emplea en uno Netscape. Para probar a distinguir entre navegadores, vamos a completar el servlet **ejercicios.NavServlet** de la aplicación **appcab** para que identifique si el cliente accede desde un tipo de navegador u otro. Para ello leemos la cabecera `User-Agent` con el método `getHeader(...)` de la petición, y comprobamos su valor. Mostrad la cadena en una página, y cargad dicha página desde dos o tres navegadores diferentes. Cada uno mostrará algún rasgo distintivo en dicha cadena, que lo identifique de los demás. Una vez tengáis una parte de texto que los diferencia (por ejemplo, imaginemos que en Netscape el `User-Agent` tiene la cadena "Netscape", y en el otro navegador (Konqueror, por ejemplo), no la tiene) haríamos con algo como:

```
public void doGet(HttpServletRequest req, ...) throws ...
{
    String nav = req.getHeader("User-Agent");
    if (nav.indexOf("Netscape") != -1)    // Cambiar "Netscape" por el
texto que sea
        ... // Netscape
    else
        ... // Otro navegador (Konqueror)
    ...
}
```

Una vez distinguido el navegador, ya se podría hacer algo que sólo sirviera para ese navegador. En este caso, por simplificar, vamos a cargar como imagen el logo del navegador que hayamos detectado. Tenéis en la plantilla las imágenes GIF correspondientes a tres navegadores o entornos diferentes. Colocad como imagen de la página la del navegador que hayáis detectado (con una etiqueta `<img>` de HTML).

## 2. Redirecciones con retardo

El formulario **form\_datos.html** de la aplicación **appcab** se envía al servlet **ejercicios.CompruebaServlet**. Se pide implementar dicho servlet para comprobar que los datos sean correctos (que no haya ningún campo de texto vacío). En el caso de que no haya errores el servlet simplemente mostrará un mensaje indicando que todo ha ido bien. Si hay algún error, el servlet debe redirigir al servlet **ejercicios.ErrorCompruebaServlet**, que deberéis completar para que muestre un mensaje con el error producido (indicando qué campo está incompleto), y a los 5 segundos redirija al formulario anterior (utilizando una cabecera de respuesta `Refresh`).

### 3. (\*) Loggear variables CGI

---

Sobre el servlet **ejercicios.CompruebaServlet** anterior vamos a añadir mensajes de log de tipo INFO, para que:

- Tras cada petición (por `doGet` o `doPost`), genere un mensaje de tipo INFO que indique la IP del cliente que solicitó la petición (variable CGI `REMOTE_ADDR`), el tipo de petición (variable CGI `REQUEST_METHOD`), y el tipo de navegador (cabecera `User-Agent`).
- El mensaje en conjunto deberá tener el formato siguiente:  
día/mes/año hora:minuto:segundo - texto del mensaje - nueva línea

Haced que los mensajes de log se guarden en un fichero `visitas.txt`, en la carpeta que preferáis. Para ello utilizad un fichero `log4j.properties` colocado en `WEB-INF/classes`.

