# **Parte I. - Jugando a ser un cliente HTTP**

**3.**

**c) Código de error:** 400 Bad Request

**Significado:** El servidor no puede o no podrá procesar la solicitud debido a un aparente error del cliente (por ejemplo, sintaxis de solicitud con formato incorrecto, tamaño demasiado grande, enmarcado de mensaje de solicitud no válido o enrutamiento de solicitud engañoso).

**d)** Otros códigos de error

**401 Unauthorized**

Similar al *403 Forbidden*, pero específicamente para su uso cuando la autentificación es posible pero ha fallado o aún no ha sido provista. Vea autenticación HTTP básica y Digest access authentication.

**402 Payment Required**

La intención original era que este código pudiese ser usado como parte de alguna forma o esquema de Dinero electrónico o micropagos, pero eso no sucedió, y este código nunca se utilizó.

**403 Forbidden**

La solicitud fue legal, pero el servidor rehúsa responderla dado que el cliente no tiene los privilegios para hacerla. En contraste a una respuesta *401 No autorizado*, la autenticación no haría la diferencia.

**404 Not Found**

Recurso no encontrado. Se utiliza cuando el servidor web no encuentra la página o recurso solicitado.

**405 Method Not Allowed**

Una petición fue hecha a una URI utilizando un método de solicitud no soportado por dicha URI; por ejemplo, cuando se utiliza GET en un formulario que requiere que los datos sean presentados vía POST, o utilizando PUT en un recurso de solo lectura.

**406 Not Acceptable**

El servidor no es capaz de devolver los datos en ninguno de los formatos aceptados por el cliente, indicados por éste en la cabecera "Accept" de la petición.

**407 Proxy Authentication Required**

**408 Request Timeout**

El cliente falló al continuar la petición - excepto durante la ejecución de videos Adobe Flash cuando solo significa que el usuario cerró la ventana de video o se movió a otro.

**409 Conflict**

Indica que la solicitud no pudo ser procesada debido a un conflicto con el estado actual del recurso que esta identifica.

**410 Gone**

Indica que el recurso solicitado ya no está disponible y no lo estará de nuevo. Debería ser utilizado cuando un recurso ha sido quitado de forma permanente.

Si un cliente recibe este código no debería volver a solicitar el recurso en el futuro. Por ejemplo un buscador lo eliminará de sus índices y lo hará más rápidamente que utilizando un código 404.

**411 Length Required**

El servidor rechaza la petición del navegador porque no incluye la cabecera Content-Length adecuada.

**412 Precondition Failed**

El servidor no es capaz de cumplir con algunas de las condiciones impuestas por el navegador en su petición.

**413 Request Entity Too Large**

La petición del navegador es demasiado grande y por ese motivo el servidor no la procesa​.

**414 Request-URI Too Long**

La URI de la petición del navegador es demasiado grande y por ese motivo el servidor no la procesa (esta condición se produce en muy raras ocasiones y casi siempre porque el navegador envía como GET una petición que debería ser POST).

**415 Unsupported Media Type**

La petición del navegador tiene un formato que no entiende el servidor y por eso no se procesa.

**416 Requested Range Not Satisfiable**

El cliente ha preguntado por una parte de un archivo, pero el servidor no puede proporcionar esa parte, por ejemplo, si el cliente preguntó por una parte de un archivo que está más allá de los límites del fin del archivo.

**417 Expectation Failed**

La petición del navegador no se procesa porque el servidor no es capaz de cumplir con los requerimientos de la cabecera Expect de la petición.

**418 I'm a teapot**

"Soy una tetera". Este código fue definido en 1998 como una inocentada, en el Protocolo de Transmisión de Hipertexto de Cafeteras (RFC-2324). No se espera que los servidores web implementen realmente este código de error, pero es posible encontrar sitios que devuelvan este código HTTP.

**422 Unprocessable Entity (WebDAV - RFC 4918)**

La solicitud está bien formada pero fue imposible seguirla debido a errores semánticos.

**423 Locked (WebDAV - RFC 4918)**

El recurso al que se está teniendo acceso está bloqueado.

**424 Failed Dependency (WebDAV) (RFC 4918)**

La solicitud falló debido a una falla en la solicitud previa.

**425 Unassigned**

Definido en los drafts de WebDav Advanced Collections, pero no está presente en "Web Distributed Authoring and Versioning (WebDAV) Ordered Collections Protocol" (RFC 3648).

**426 Upgrade Required (RFC 7231)**

El cliente debería cambiarse a TLS/1.0.

**428 Precondition Required**

El servidor requiere que la petición del navegador sea condicional (este tipo de peticiones evitan los problemas producidos al modificar con PUT un recurso que ha sido modificado por otra parte).

**429 Too Many Requests**

Hay muchas conexiones desde esta dirección de internet.

**431 Request Header Fields Too Large)**

El servidor no puede procesar la petición porque una de las cabeceras de la petición es demasiado grande. Este error también se produce cuando la suma del tamaño de todas las peticiones es demasiado grande.

**449**

Una extensión de Microsoft: *La petición debería ser reintentada después de hacer la acción apropiada*.

**451 Unavailable for Legal Reasons**

El contenido ha sido eliminado como consecuencia de una orden judicial o sentencia emitida por un tribunal.

**4.**

**- Resultado**

HTTP/1.1 200 OK

Access-Control-Allow-Credentials: true

Access-Control-Allow-Origin: \*

Content-Type: text/html; charset=utf-8

Date: Thu, 12 Sep 2019 12:40:10 GMT

Referrer-Policy: no-referrer-when-downgrade

Server: nginx

X-Content-Type-Options: nosniff

X-Frame-Options: DENY

X-XSS-Protection: 1; mode=block

Content-Length: 3741

Connection: keep-alive

<!DOCTYPE html>

<html>

<head>

</head>

<body>

<h1>Herman Melville - Moby-Dick</h1>

<div>

<p>

Availing himself of the mild, summer-cool weather that now reigned in these latitudes, and in preparation for the peculiarly active pursuits shortly to be anticipated, Perth, the begrimed, blistered old blacksmith, had not removed his portable forge to the hold again, after concluding his contributory work for Ahab's leg, but still retained it on deck, fast lashed to ringbolts by the foremast; being now almost incessantly invoked by the headsmen, and harpooneers, and bowsmen to do some little job for them; altering, or repairing, or new shaping their various weapons and boat furniture. Often he would be surrounded by an eager circle, all waiting to be served; holding boat-spades, pike-heads, harpoons, and lances, and jealously watching his every sooty movement, as he toiled. Nevertheless, this old man's was a patient hammer wielded by a patient arm. No murmur, no impatience, no petulance did come from him. Silent, slow, and solemn; bowing over still further his chronically broken back, he toiled away, as if toil were life itself, and the heavy beating of his hammer the heavy beating of his heart. And so it was.—Most miserable! A peculiar walk in this old man, a certain slight but painful appearing yawing in his gait, had at an early period of the voyage excited the curiosity of the mariners. And to the importunity of their persisted questionings he had finally given in; and so it came to pass that every one now knew the shameful story of his wretched fate. Belated, and not innocently, one bitter winter's midnight, on the road running between two country towns, the blacksmith half-stupidly felt the deadly numbness stealing over him, and sought refuge in a leaning, dilapidated barn. The issue was, the loss of the extremities of both feet. Out of this revelation, part by part, at last came out the four acts of the gladness, and the one long, and as yet uncatastrophied fifth act of the grief of his life's drama. He was an old man, who, at the age of nearly sixty, had postponedly encountered that thing in sorrow's technicals called ruin. He had been an artisan of famed excellence, and with plenty to do; owned a house and garden; embraced a youthful, daughter-like, loving wife, and three blithe, ruddy children; every Sunday went to a cheerful-looking church, planted in a grove. But one night, under cover of darkness, and further concealed in a most cunning disguisement, a desperate burglar slid into his happy home, and robbed them all of everything. And darker yet to tell, the blacksmith himself did ignorantly conduct this burglar into his family's heart. It was the Bottle Conjuror! Upon the opening of that fatal cork, forth flew the fiend, and shrivelled up his home. Now, for prudent, most wise, and economic reasons, the blacksmith's shop was in the basement of his dwelling, but with a separate entrance to it; so that always had the young and loving healthy wife listened with no unhappy nervousness, but with vigorous pleasure, to the stout ringing of her young-armed old husband's hammer; whose reverberations, muffled by passing through the floors and walls, came up to her, not unsweetly, in her nursery; and so, to stout Labor's iron lullaby, the blacksmith's infants were rocked to slumber. Oh, woe on woe! Oh, Death, why canst thou not sometimes be timely? Hadst thou taken this old blacksmith to thyself ere his full ruin came upon him, then had the young widow had a delicious grief, and her orphans a truly venerable, legendary sire to dream of in their after years; and all of them a care-killing competency.

</p>

</div>

</body>

</html>

**5.**

**a) Contar número de caracteres:** 3742

**b) ¿Cuál es la diferencia entre los verbos GET y POST?**

La diferencia entre los métodos get y post radica en la forma de enviar los datos a la página cuando se pulsa el botón “Enviar”. Mientras que el método GET envía los datos usando la URL, el método POST los envía de forma que no podemos verlos (en un segundo plano u "ocultos" al usuario).

**c) ¿Qué otros tipos de peticiones existen?**

**HEAD**

El método HEAD pide una respuesta idéntica a la de una petición GET, pero sin el cuerpo de la respuesta.

**PUT**

El modo PUT reemplaza todas las representaciones actuales del recurso de destino con la carga útil de la petición.

**DELETE**

El método DELETE borra un recurso en específico.

**CONNECT**

El método CONNECT establece un túnel hacia el servidor identificado por el recurso.

**OPTIONS**

El método OPTIONS es utilizado para describir las opciones de comunicación para el recurso de destino.

**TRACE**

El método TRACE realiza una prueba de bucle de retorno de mensaje a lo largo de la ruta al recurso de destino.

**PATCH**

El método PATCH es utilizado para aplicar modificaciones parciales a un recurso.

**6.**

**curl:** Es un proyecto de software que proporciona una biblioteca y una herramienta de línea de comandos para transferir datos usando varios protocolos.

**-i, --include** Incluye encabezados de respuesta de protocolo en la salida.

**-v, --verbose**  Make the operation more talkative.

# **Parte II. - Haciendo una aplicación Web dinámica a bajo nivel.**

**1.** La clase SampleServlet extiende a la clase abstracta HttpServelet, la cual lee la solicitud HTTP y determina si esta es HTTP GET, POST, PUT, DELETE, HEAD, etc. y llama al método correspondiente. De modo que quien la extiende, en este caso SampleServlet, debe sobreescribir al menos uno de estos métodos generalmente: doGet, doPost, doPut, doDelete.

**3.** TomCat: Puerto 8080.

# **Parte III.**

**20.** Se observa que la diferencia entre POST y GET, es que en GET se ve el valor del número en la URL, y por el contrario en POST se ocuta.

**21.** De la clase Service se están utilizando 2 métodos: getTodo y todosToHTMLTable. En el primero se le pasa como parámetro un número que vendría siendo el id, luego el método retorna a un objeto tipo Todo si lo encuentra; sino lo encuentra o está vacío puede generar excepciones. Por otro lado, el segundo método, lo que hace es insertar en un StringBuilder los datos de cada objeto Todo presente en la lista que se le pasa como parámetro, y lo organiza con estructura html, y lo retorna como una cadena.

# **Parte IV.**

**10.**

**b.** Abra la aplicación en dos computadores diferentes. Si no dispone de uno, hágalo en dos navegadores diferentes (por ejemplo Chrome y Firefox; incluso se puede en un único navegador usando una ventana normal y una ventana de incógnito / privada). Haga cinco intentos en uno, y luego un intento en el otro. ¿Qué valor tiene cada uno?

- En ambos aparecen los mismos valores.

**c.**

* ¿Coinciden los valores del premio?

No, todos los valores son distintos.

* Dado la anterior, ¿Cuál es la diferencia entre los backing-beans de sesión y los de aplicación?

**@ApplicationScoped:** Se comparten los estados entre todos los usuarios que interactúan con una aplicación web.

**@SessionScoped:** Los usuarios interactúan con una aplicación web a través de múltiples solicitudes HTTP.