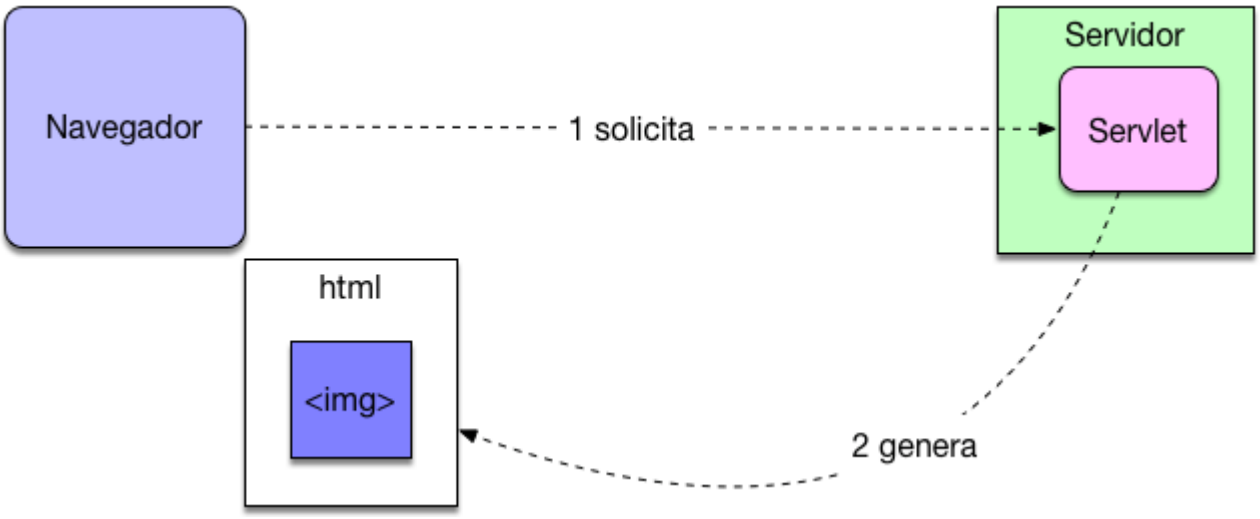
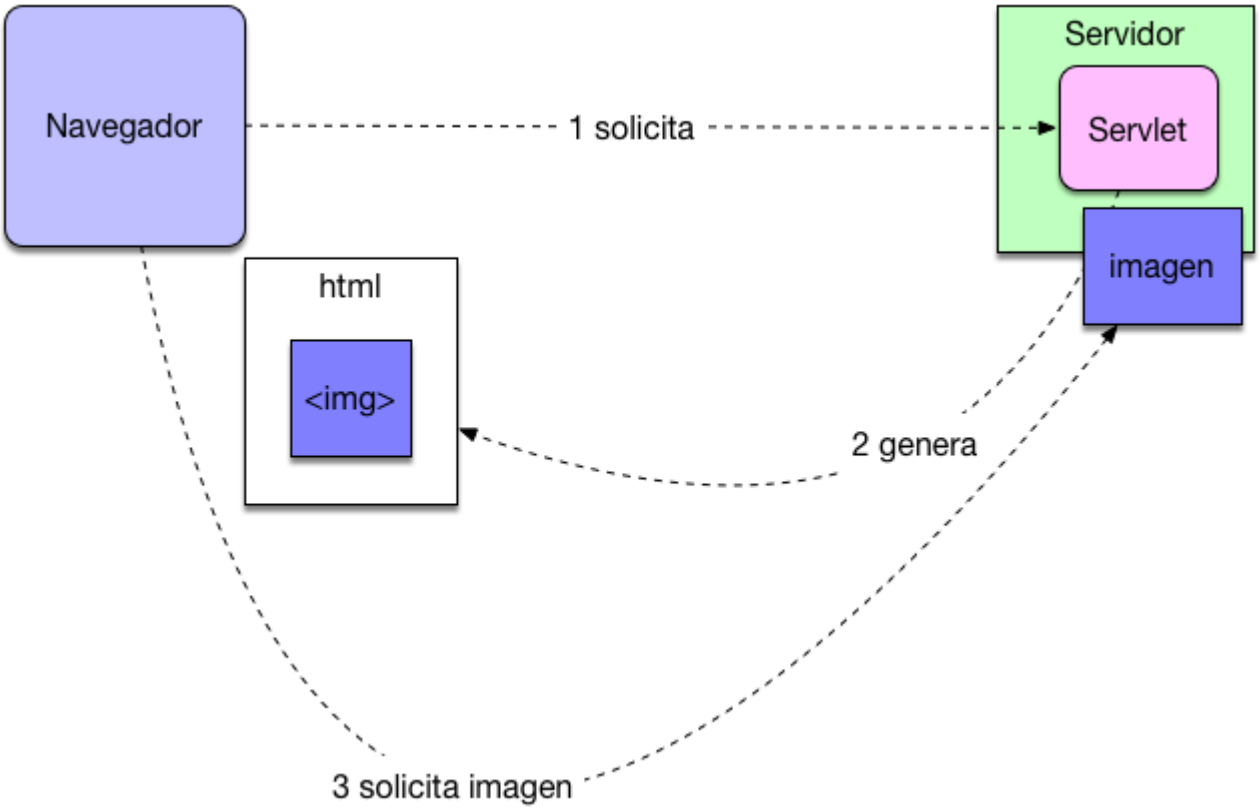
Poco a poco la especificación de servlets 4 irá llegando a nuestros servidores (Java EE 8). Una de sus grandes ventajas es la posibilidad de usar las capacidades de HTTP 2.0 para gestionar los recursos. HTTP 2.0 permite lo que se denomina operaciones push . Es decir el servidor puede enviar recursos al cliente antes de que este realmente los necesite . Esto puede suponer una mejora del rendimiento. Vamos a verlo a través de un ejemplo sencillo en el que un Servlets genera una página html que contiene una imagen. En un principio con la programación clásica de los servlets solicitaríamos al servidor el servlet y este generaría una página html que contiene un link a una imagen.

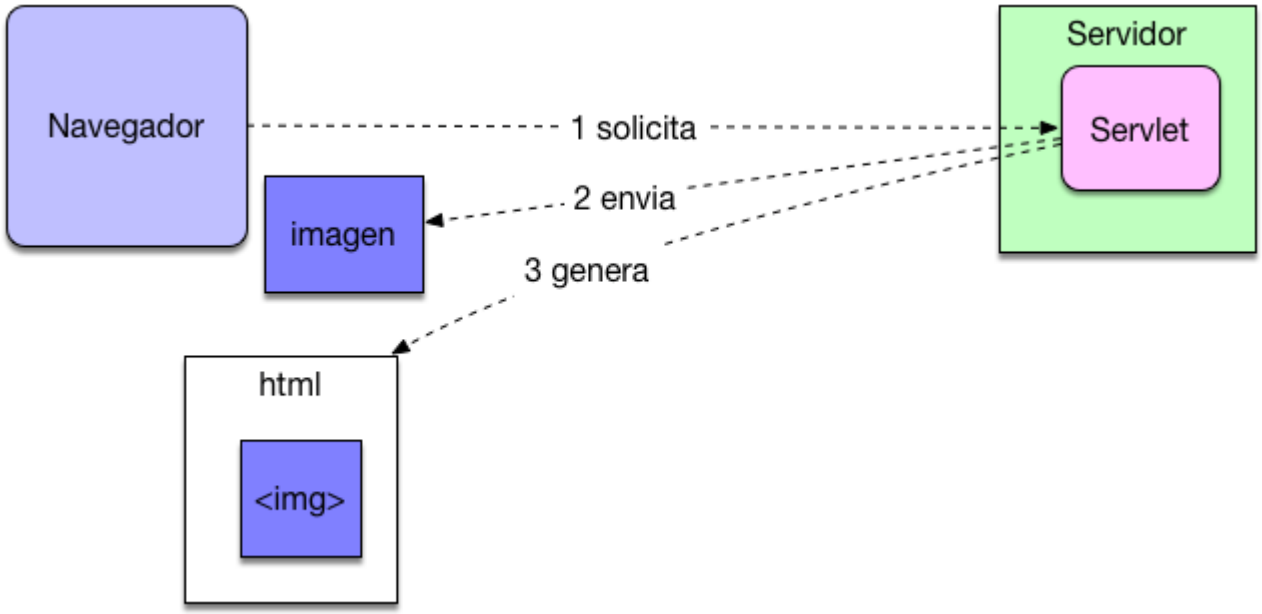


El navegador recibiría esta página y se encargaría de solicitar el recurso adicional que necesitamos (la imagen al servidor).



Servlets 4 y Http 2

Hasta aquí todo correcto sin embargo a partir de HTTP 2.0 podemos enviar de partida esos recursos al cliente con una operación push antes de que el tenga que solicitarlo.



Vamos a verlo en código modificando nuestro servlet de inicio.

```
package com.arquitecturajava;

import java.io.IOException;
import java.io.PrintWriter;

import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.PushBuilder;
```

```
/**
 * Servlet implementation class ServletHola
 */
@WebServlet("/ServletHola")
public class ServletHola extends HttpServlet {
        private static final long serialVersionUID = 1L;
         @Override
            protected void doGet(HttpServletRequest request,
            HttpServletResponse
            response) throws ServletException, IOException {

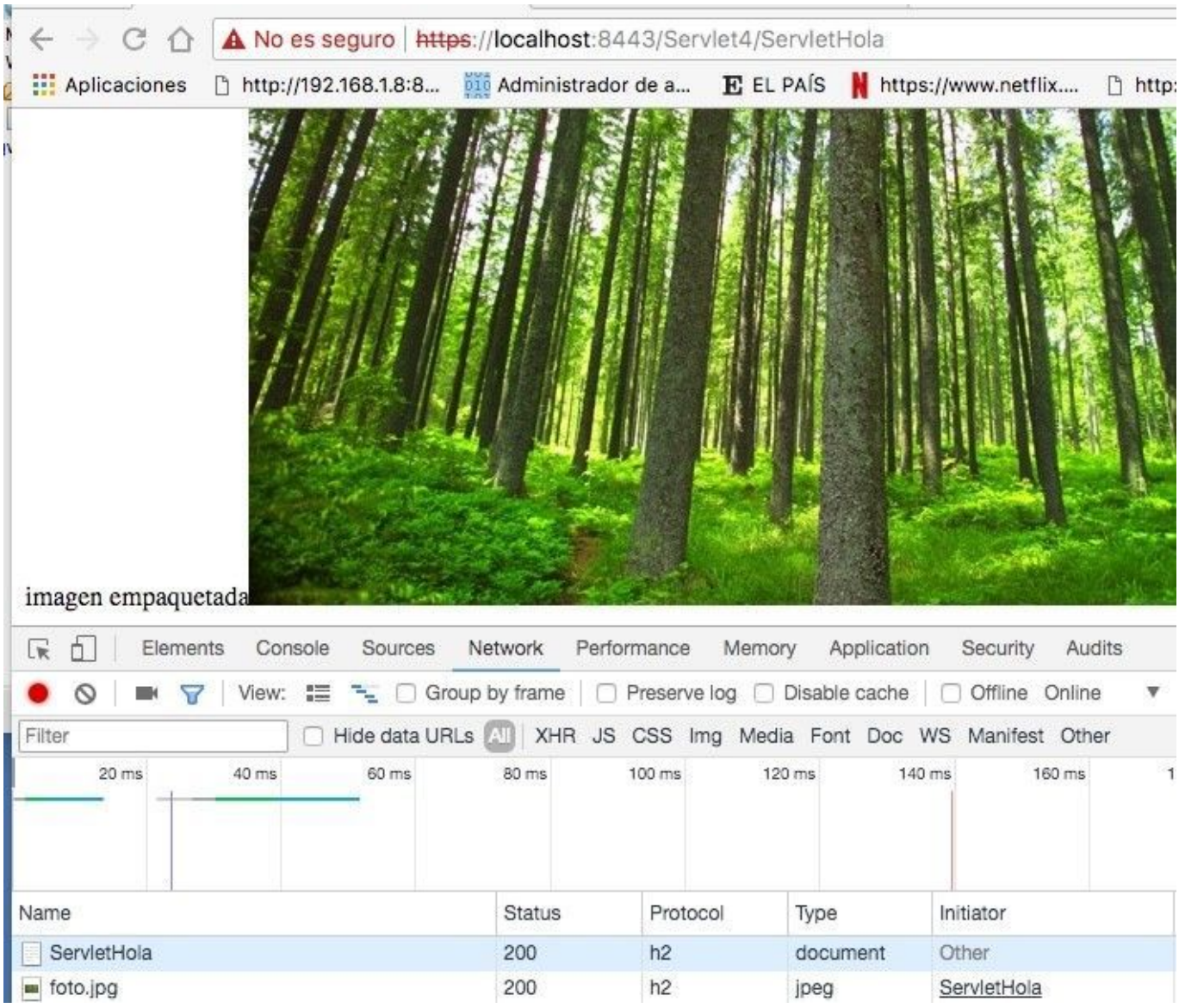
                PushBuilder pb = request.newPushBuilder();
                if (pb != null) {
                    pb.path("imagenes/foto.jpg")
                      .addHeader("content-type", "image/jpg")
                      .push();
                }

                try (PrintWriter writer = response.getWriter();) {
                    StringBuilder html = new StringBuilder();
                    html.append("<html>");
                    html.append("<body>");
                    html.append("imagen empaquetada");
                    html.append("<img src='imagenes/foto.jpg>");
                    html.append("</body>");
                    html.append("</html>");
                    writer.write(html.toString());
                }
            }

}
```

Servlets 4 y Chrome

El servlet es muy sencillo podemos ver claramente como lo primero que hacemos es comprobar que podemos enviar esa imagen que la página va a necesitar. Si la operación de push esta soportada enviamos la imagen que la tenemos ubicada en la carpeta. El resultado será que la página es cargada con la imagen y que el protocolo de comunicación es h2 tanto para la imagen como para el Servlet.



Acabamos de hacer nuestro primer ejemplo con Servlets 4.0 . Hay que recordar que para conseguir que nos funcione a nivel de HTTP 2 el servidor web tenemos que tener instalado el conector de SSL en mi caso he usado tomcat.

Otros artículos relacionados

1. Tomcat Java DataSource y @Resource
2. Tomcat context xml y su configuración
3. Eclipse Timeout Servidor Tomcat/JBoss Etc