1- Uma aplicação web encontra-se alojada, por norma, num servidor, e providencia, remotamente, serviços para os seus clientes. Ainda que os recursos da aplicação se encontrem armazenados no servidor, é necessário um mecanismo que detalhe como o acesso a estes será efetuado por um cliente. Esse mecanismo denomina-se HTTP e utiliza um sistema de pedido/resposta, em que o cliente envia uma mensagem ao servidor a requisitar um recurso, e o servidor responde, geralmente enviando esse mesmo recurso.

2 - Quando clicamos numa hiperligação, o browser envia um pedido ao servidor adequado a requisitar a página web correspondente. Quando este recebe a resposta, é necessário processar a página HTML e procurar referências para outros recursos, como imagens ou scripts. Esta função é da responsabilidade do módulo de interpretação de conteúdo. Este módulo vasculha a página por recursos adicionais, e para cada um que encontra, gera um pedido adicional a ser enviado ao servidor, de modo a poder juntar todo o conteúdo necessário para o browser renderizar a página web completa.

3 – O header Host é um cabeçalho obrigatório em todos os pedidos HTTP, na medida em que possibilita o conceito de virtual hosting. Um servidor só pode alojar vários domínios da mesma aplicação, contudo apenas pode ter atribuído um endereço IP. O virtual hosting permite ao servidor processar os pedidos que recebe de forma diferente, consoante o valor presente nesse cabeçalho.

4 – O módulo State Maintenance é responsável por verificar se existem cookies a serem incluídas no cabeçalho da resposta, antes da mesma ser gerada. Dado que o HTTP é um protocolo stateless, o mecanismo que se usa para manter uma sessão, ainda que lógica, aberta são as cookies. Ao processar uma resposta, o cliente web consulta o cabeçalho desta e, caso encontre o cabeçalho Set-Cookie (ou Set-Cookie2), o módulo State Maintenance é responsável por processar o conteúdo destes, guardando , em memória ou no disco, a informação associada às cookies recebidas ou atualizando a informação, caso estas já existam no cliente.

5 – Ainda que os browsers modernos estejam preparados para renderizar a maioria dos conteúdos das respostas, como texto, imagens, áudio ou vídeo, existem certos conteúdos que não são nativamente compatíveis. Isto pode verificar-se por duas razões: a capacidade renderizar nativamente esse conteúdo vai para além das funcionalidades do browser; os componentes necessário para fazer a renderização podem complicar o processamento da resposta. Por isso mesmo, os browsers atuais suportam plugins. Plugins são pequenos programas ou aplicações incorporadas no próprio browser que se encarregam de renderizar conteúdos mais complexos, mas frequentes, por exemplo, um leitor de PDFs.

6 – Caso o cliente faça gestão local de contéudos cache (cliente-side cache), esta pode influenciar como este processa os pedidos de recursos aos servidores. Quando um browser necessita de um recurso, este consulta a sua cache à procura duma cópia que possa usar. Se essa cópia existir e ainda não tiver expirado, o browser evita fazer um pedido desnecessário ao servidor e usa-a. Se a sua cópia tiver expirado, o browser pode enviar um pedido do recurso ao servidor incluindo o cabeçalho If-Modified-Since ou If-Unmodified-Since. Ao processar o pedido, o servidor consulta a data de última modificação do recurso (Last-Modified) e responde de acordo com esta. Se for usado o cabeçalho If-Modified-Since, o servidor pode responder com um status code 304 (Unmodified) ou 200 (OK), devendo neste último incluir na resposta o recurso atualizado. Já se for usado o cabeçalho If-Unmodified-Since e o recurso já tiver sido modificado desde essa data, o servidor responderá com um 412 (Precondition Failed).

7 – Caso o utilizador especifique no seu browser que não pretende usar web cookies, pode ser necessário encontrar uma alternativa para manter uma sessão entre o cliente e o servidor. Este tipo de sessões só é possível usando aplicações com configuração cookieless, que pode consistir em recorrer, por exemplo, à query string do URL para enviar informação relativa à sessão (ID) no mesmo. Assim, é possível identificar a sessão em questão sem recorrer a informação armazenada sob a forma de cookies.