**Aula 5**

**Signal R – Técnicas que surgiram nos navegadores**

- **Polling** (Em meados dos anos 2000)

Surgimento do AJAX (construído pela Microsoft) – Requisições feitas via javascript sem ser necessário recarregar a página.

Funciona como uma junção do Ajax com o setInterval() ou setTimeout().

**Desvantagens**

- Atraso nas atualizações (por conta dos intervalos de tempo para esperar a próxima requisição).

- Sobrecarregar o servidor com várias requisições HTTP. (Quanto menor o tempo, maior é o número de requisições).

- **Long** **Polling**

Uma evolução do Polling.

Não tem o uso do setInterval(), mas de requisição, aonde ela é enviada e o servidor vai segurar ela até que tenha uma resposta para ela (claro que as requisições tem um *timeout,* se não for definida, tem o padrão do navegador).

O navegador pode enviar várias requisições em paralelo (+- 8) para cada domínio.

**Vantagens**

- Sem atraso na requisição

- Suportado por vários navegadores

- Menor processamento noservidor

**Desvantagens**

- Múltiplos requests

- **Forever Frame (muito utilizado na internet discada)**

Técnica mais complexa.

Pegar uma página e colocar um iframe, esconde ele e o Javascript verifica esse frame, sendo um processamento do próprio navegador.

O iframe sempre é controlado pelo servidor e o servidor sempre vai deixá-lo em aberto, ou seja, a resposta nunca será finalizada.

Ao chegar um conteúdo, você escreve na página para ser retornado, ou seja, vai entregando pequenos pacotes bem pequenos para serem escritos na página. Então o javascript verifica que chegou algo novo no iframe e já manda a resposta.

A diferença para o long polling é que enquanto ele retorna a requisição toda, o forever frame retorna partes da resposta, conforme ela vai chegando.

(não há o uso do Ajax).

**Vantagens**

- Suportado por vários navegadores

**Desvantagens**

- Pode ter problemas com o navegador (peculiaridades).

- Dificuldade em tratar os erros

**Aula 6**

- **Server-Sent Events (SSE)**

A requisição sempre era feita pelo cliente para o servidor. Nesse caso foi a primeira vez que o servidor poderia fazer contato com o cliente de forma independente, ou seja, o cliente não fez uma requisição primeiro.

Claro, é necessário o cliente ter a conexão, as configurações e tudo mais, porém ele não envia uma requisição.

**Vantagens**

- Abre uma conexão com o servidor

**Desvantagens**

- Conexão unidirecional – servidor envia para cliente (não da para usar o mesmo canal de comunicação)

- Limitado a 6 conexões por navegador

- **WebSockets**

Foi construído justamente para superar a desvantagem do SSE, para a comunicação ser para os dois sentidos e não sobrecarrega por não haver limite de conexões.

Por utilizar TCP (protocolo de rede), ele trabalha com níveis mais baixos, mais simples, o que garante a performance dos websockets.

**Vantagens**

- Abre uma conexão com o servidor

- Bi-direcional

- Funciona via TCP

- Não tem limite de conexões

**Desvantagens**

- Navegador precisa suportar essa tecnologia (Não suportado por navegadores antigos).