

# Proxy HTTP

---

Grupo: Renan Benatti dias, Victor Pereira de Paula

## Introdução

O proxy HTTP foi implementado para a disciplina de Redes de Computadores. A linguagem escolhida para implementação foi Java. Apenas os comandos GET são tratados pelo proxy.

## Configuração do navegador

Ensinaremos de forma breve a configuração do Firefox para utilizar o proxy:

1. Selecione a opção **Preferências**
2. Na sessão **Configurações de rede** clique em **Configurar conexões...**
3. Na sessão **Configuração do proxy de acesso à internet** selecione a opção **Configuração manual de proxy**
4. No campo **Proxy HTTP** insira *localhost* e no campo **Porta** insira a porta de sua preferência
5. Clique em **OK** e pronto

## Executar o proxy

1. Na pasta **src/**, abra o terminal e compile o código com o comando `javac MainServer.java`
2. Após compilado, para utilizar execute o seguinte comando no terminal `Java MainServer <número_da_porta> <tamanho_da_cache_em_MB>`
3. Abra o navegador já configurado e navegue

## Implementação

### Problemas e soluções

1. Cabeçalhos HTTP: Nas primeiras fases da implementação não havíamos implementado a adição de cabeçalhos às respostas, o que acarretava a não renderização das páginas que testávamos, mesmo quando recebíamos conteúdo nas respostas.
2. Cache e LRU (Least Recently Used): Como temos uma limitação do tamanho da cache, temos a necessidade de remover itens conforme há demanda por espaço. A solução encontrada para implementação de forma mais fácil, foi criar uma **LinkedList** para armazenar os **URLs** das páginas acessadas, desta forma as páginas mais antigas sempre estarão no fim da lista. Também criamos um **HashMap** para salvar os elementos cacheados, com a **URL** como chave, desta forma quando precisamos remover itens, removemos a partir da última posição da **LinkedList**. O uso do **HashMap** além da **LinkedList** foi para facilitar a obtenção dos dados da cache, por meio do **URL** (chave).
3. Requisição de imagens: Inicialmente tratamos todas as requisições como se estas respondessem sempre com arquivos HTML. Então passamos a tratar os URLs das requisições a fim de saber quais são as requisições de imagem e desta forma receber os dados de acordo com o tipo do arquivo, embora os itens da cache sejam salvos em bytes, a forma que o java lê a imagem é diferente da forma como lê arquivos de texto, desta forma faz-se necessário o tratamento diferenciado.

4. Requisição de vídeos: Não foram realizados os tratamentos necessários para requisições de vídeos, nem o armazenamento destes na cache.

## Conclusão

O Proxy funcionou corretamente para sites simples como <http://pudim.com.br/> ou <http://example.com/>, no entanto em páginas com estruturas de arquivos mais complexas como <http://dados.gov.br/dataset/dominios-gov-br> ou <http://abelhasbrasil.myspecies.info/> não renderizam corretamente.

Na versão atual foram tratados apenas requisições de texto e imagens (**JPG** e **PNG**), desta forma conteúdos em formatos diferentes não são baixados, alterando como a página é exibida.

Outro ponto de atenção é a necessidade de tratamento dos arquivos de texto, para que a renderização de caracteres especiais seja feita corretamente, atualmente letras acentuadas e outros caracteres especiais são substituídos pelo carácter ❖.

Criamos uma página de erro para ser exibida quando algum erro ocorre durante a requisição da página, no entanto algumas páginas tem problemas em requisições de itens secundários, mas que sem eles não é possível visualizar ou navegar pela página, no entanto nesses casos não há renderização da página de erro.