



# Relatório para o projeto de Implementação de uma aplicação de Proxy Sever

Universidade de Brasília  
Departamento de Ciência da Computação  
Transmissão de Dados  
2017/1  
João Carlos Vasconcelos - 13/0062014

## Índice

Definições.....	3
Proxy Server Web.....	3
Um simples exemplo de utilização de um proxy.....	3
TCP.....	3
Exemplo de uso do TCP.....	3
HTTP.....	4
Exemplo de mensagem de requisição HTTP.....	4
Exemplo de mensagem de resposta HTTP.....	4
Implementação.....	5
Execução.....	5
Documentação.....	6
Funcionalidades Implementadas.....	6
Funcionalidades não implementadas.....	6

## Definições

### Proxy Server Web

Trata-se de um servidor dedicado ou um software que atua entre um host (como um computador) e um servidor web do qual o host está requisitando o serviço. O proxy pode inclusive estar no mesmo computador do host ou em um servidor à parte.

Uma vantagem do servidor de proxy é que o seu cache, gerado por vários usuários pode ser utilizado por também vários usuários o que diminui o consumo de banda e melhora o tempo de resposta para os usuários do proxy, além disso um proxy pode gerar log das requisições o que ajuda em caso de analisar eventuais problemas.

### Um simples exemplo de utilização de um proxy

Quando um servidor de proxy recebe uma requisição ele verifica em seu cache local se há conteúdo de resposta para a requisição, se houver e estiver válida a resposta é devolvida ao usuário, sem a necessidade de enviar nenhuma requisição na internet.

Caso não haja cache o servidor de proxy atua apenas como um repassador de conteúdo entre o cliente e o servidor web. Neste caso, se possível o servidor efetuará o cache para que em uma próxima requisição o cache seja utilizado.

## TCP

TCP é a sigla para *Transmission Control Protocol*, este é um padrão que define como se estabelece e é mantido a comunicação de aplicações para a transferência de dados. O TCP é um protocolo orientado à conexão, o que significa que uma conexão é estabelecida e mantida até que a camada de aplicação determine que o processo de troca de mensagens tenha terminado.

### Exemplo de uso do TCP

Quando um servidor web envia um conteúdo HTML para um cliente por meio de uma mensagem HTTP é solicitado ao TCP que crie uma conexão para o envio do arquivo, por sua vez o TCP divide o arquivo em pacotes e os enumera e encaminha cada pacote. O TCP no lado do cliente espera até que tenha recebido todos os pacotes e monta novamente o arquivo para entregá-lo a aplicação.

## HTTP

HTTP é a sigla para *Hypertext Transfer Protocol*, ou seja é uma série de regras para a transferência de arquivos (texto, imagens, som, vídeo e outros conteúdos multimídia) por meio da internet. Sempre que utilizamos nossos navegadores web estamos fazendo indiretamente requisições HTTP. O HTTP é um protocolo da camada de aplicação que roda por cima do TCP/IP.

### Exemplo de mensagem de requisição HTTP

```
GET /~caetano/ HTTP/1.1
Host: cic.unb.br
User-Agent: curl/7.47.0
Accept: */*
```

### Exemplo de mensagem de resposta HTTP

```
HTTP/1.1 200 OK
Date: Sun, 25 Jun 2017 20:10:21 GMT
Server: Apache/2.4.10 (Debian) OpenSSL/1.0.1t
Last-Modified: Tue, 02 Apr 2013 21:43:38 GMT
ETag: "245-4d967a14ed680"
Accept-Ranges: bytes
Content-Length: 581
Vary: Accept-Encoding
Content-Type: text/html

<!DOCTYPE HTML>
<html lang="en-US">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="refresh"
content="1;url=https://sites.google.com/site/mfcaetano">
  <script type="text/javascript">
    window.location.href = "https://sites.google.com/site/mfcaetano"
  </script>
  <title>Page Redirection</title>
</head>
<body>
<!-- Note: don't tell people to `click` the link, just tell them that it
is a link -->
If you are not redirected automatically, follow the <a
href='https://sites.google.com/site/mfcaetano'>https://sites.google.com/si
te/mfcaetano</a>
</body>
</html>
```

## Implementação

### Execução

A aplicação denominada ProxY foi desenvolvida utilizando Python 2.7 e não foi utilizada nenhuma biblioteca adicional para execução da aplicação. Todo o código está disponível em: <https://github.com/ninovasc/ProxY/>.

A aplicação é chamada por meio da execução do arquivo proxy.py:

```
python proxy.py
```

As configurações de comportamento da aplicação podem ser alteradas no arquivo `config.json` que é criado com valores padrão, os log da aplicação se localizam no diretório `log` e o cache no arquivo de banco de dados SQLite `cache.sqlite`. Todos esses arquivos, caso não existam são criados na primeira execução do aplicativo.

A aplicação foi testada utilizando o navegador Mozilla Firefox utilizando apenas como servidor HTTP, conforme figura 1. Foi utilizado o site do Instituto de Computação da Unicamp (<http://www.ic.unicamp.br/>) para a execução dos testes.

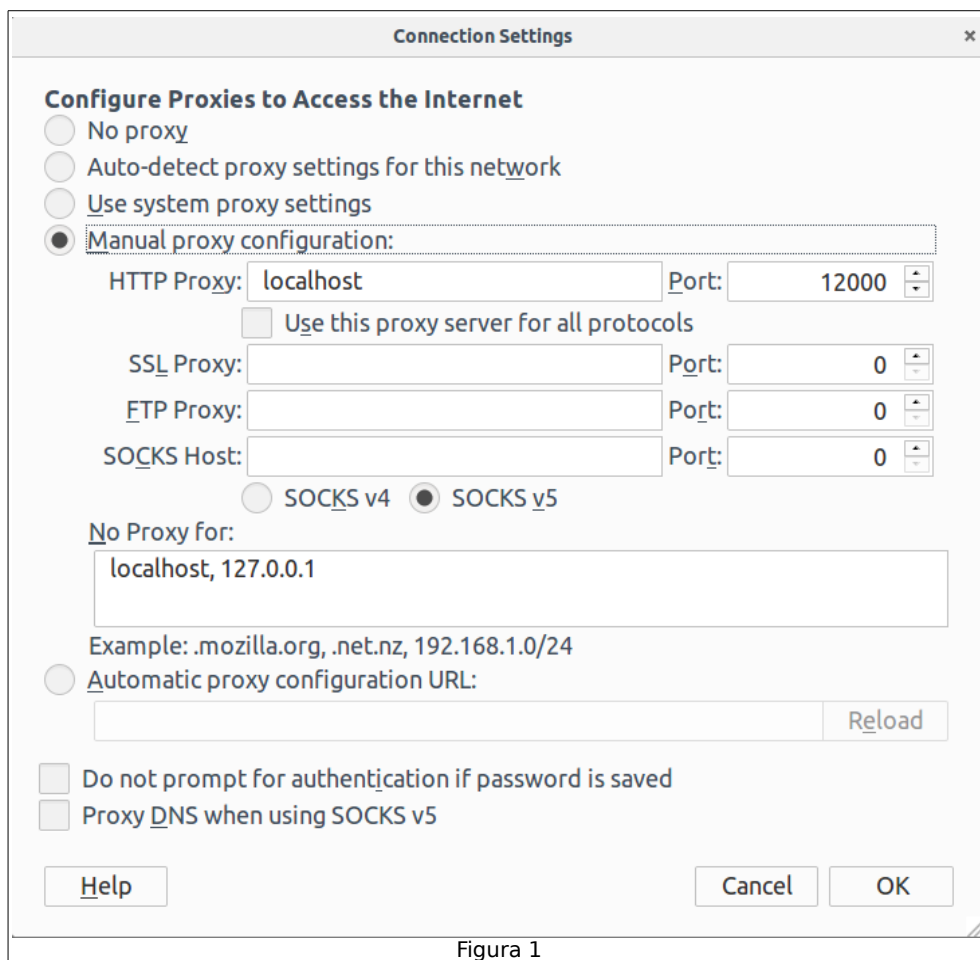


Figura 1

## Documentação

A documentação do código foi gerada utilizando a ferramenta Sphinx (site oficial da ferramenta: <http://www.sphinx-doc.org>) e está disponível no arquivo [Documentação.pdf](#). Neste arquivo estão descritos todos os conceitos das classes e métodos utilizados na aplicação.

## Funcionalidades Implementadas

Foram implementadas as seguintes funcionalidades:

1. Intermediação de requisições;
2. Parser HTTP;
3. Cache; e
4. Log.

Na Figura 2 observamos o console da primeira execução da aplicação, não há nesse relatório outras imagens da tela, pois, não há interface de execução para as funcionalidades.

```
2017-06-25 21:31:22 - log directory created  
  
2017-06-25 21:31:22 - config file created with default values  
  
2017-06-25 21:31:22 - cache DB created  
  
2017-06-25 21:31:22 - ProxY port load from file, trying to listening on 12000  
  
ProxY port load from file, trying to listening on 12000  
2017-06-25 21:31:22 - ProxY Started Sucessfully [ 12000 ]  
  
ProxY Started Sucessfully [ 12000 ]
```

Figura 2

## Funcionalidades não implementadas

Neste projeto não foram implementadas as requisições de filtro de requisições, de “*whitelist*”, “*blacklist*” e “*deny terms*”.