

# Inspetor HTTP baseado em Proxy Server

Wanderlan Alves - 16/0148782  
Miguel Barreto Rezende - 12/0130424

July 2019

## 1 Apresentação teórica

### 1.1 TCP/IP

O TCP/IP (Transmission Control Protocol/Internet Protocol) é o principal conjunto de protocolos de envio e recebimento de dados da internet.

Esses protocolos são divididos em cinco camadas: aplicação, transporte, rede, enlace e camada física. Cada uma delas é responsável pela execução de tarefas distintas.

#### 1.1.1 Camada de aplicação

A camada de aplicação é a camada que a maioria dos programas de rede usa de forma a se comunicar através de uma rede com outros programas. Processos que rodam nessa camada são específicos da aplicação.

O dado é passado do programa de rede, no formato usado internamente por essa aplicação, e é codificado dentro do padrão de um protocolo. Uma vez que o dado de uma aplicação foi codificado dentro de um padrão de um protocolo da camada de aplicação ele será passado para a próxima camada.

#### 1.1.2 Camada de transporte

Os protocolos na camada de transporte podem resolver problemas como confiabilidade (o dado alcançou seu destino?) e integridade (os dados chegaram na ordem correta?). Na suíte de protocolos TCP/IP os protocolos de transporte também determinam para qual aplicação um dado qualquer é destinado.

#### 1.1.3 Camada de rede

A camada de rede permite gerenciar o endereçamento e o encaminhamento dos dados, isto é, o seu encaminhamento através da rede.

#### 1.1.4 Camada de Enlace

A camada de enlace é o método usado para passar quadros da camada de rede de um dispositivo para a camada de rede de outro. Ela realiza processos como adicionar um header de pacote para prepará-lo para transmissão, então de fato transmitir o quadro através da camada física.

Do outro lado, a camada de enlace irá receber quadros de dados, retirar os headers adicionados e encaminhar os pacotes recebidos para a camada de rede.

#### 1.1.5 Camada física

Também chamada camada de abstração de hardware, tem como função principal ajustar a interface do modelo TCP/IP com os diversos tipos de redes (X.25, ATM, FDDI, Ethernet, Token Ring, Frame Relay, sistema de conexão ponto-a-ponto SLIP, etc.) e transmitir os datagramas pelo meio físico. Além de encontrar o caminho mais curto e confiável.

### 1.2 Protocolo HTTP

O HTTP (Hyper Text Transfer Protocol) é um protocolo utilizado pela World Wide Web que determina como as mensagens serão formatadas e transmitidas na rede e quais ações os servidores Web e os browsers devem tomar para diferentes situações.

Quando uma URL é inserida em um browser ele envia um pedido (Request) para o servidor web. Primeiramente, o browser descobre o endereço IP do servidor web usando o DNS. Em seguida, ele faz a conexão diretamente com o servidor solicitando uma página, imagem ou algum outro arquivo. Por sua vez, o servidor HTTP envia ao browser o conteúdo solicitado.

### 1.3 Proxy server

Um servidor proxy atua como um agente intermediário entre um cliente e um servidor web, fazendo com que o cliente se conecte apenas com o servidor proxy e não tenha que se conectar diretamente com o servidor web.

Quando um servidor proxy recebe uma solicitação para um recurso da Internet (como uma página da Web), ele procura em seu cache local de páginas anteriores. Se encontrar a página, ela retornará ao usuário sem precisar encaminhar a solicitação para a Internet. Se a página não estiver no cache, o servidor proxy, agindo como um cliente em nome do usuário, usa um dos seus próprios endereços IP para solicitar a saída da página do servidor na Internet. Quando a página é retornada, o servidor proxy a relaciona com a solicitação original e a encaminha para o usuário.

## 1.4 Spider e Dump

O Spider e o Dump são ambas funcionalidades de um servidor proxy. O Spider consiste em criar uma árvore hipertextual com todas as referências de uma página web, tendo como raiz a página inicial(home).

O Dump consiste em baixar todos os objetos presentes nessas referências em um arquivo local no computador.

## 2 Arquitetura do programa

O programa possui as seguintes classes principais :

### 2.1 Request

Consiste em verificar e montar as mensagens de request inspecionadas pelo servidor proxy.

### 2.2 Response

Consiste em verificar e montar as mensagens de response do servidor ao cliente. Funciona de forma semelhante ao Request.

### 2.3 Parser

Extraí as informações do corpo das mensagens trocadas entre cliente e servidor.

### 2.4 Servidor proxy

Inicia o servidor na porta TCP desejada (8228 é a porta padrão), cria o socket do servidor, monta o request do cliente e retorna um reply.

### 2.5 Spider

Gera a árvore de referências da página web acessada e permite a realização do Dump.

### 2.6 Proxyui

É a classe responsável por implementar a interface gráfica do programa. Para este projeto, a interface foi criada utilizando o QT creator.

## 3 Funcionamento

Todo o código do programa junto com as instruções de como executá-lo está disponível em:

”<https://github.com/Wander-lan/Trabalho-TR2>”

Ao executar, a tela inicial do programa será a seguinte :

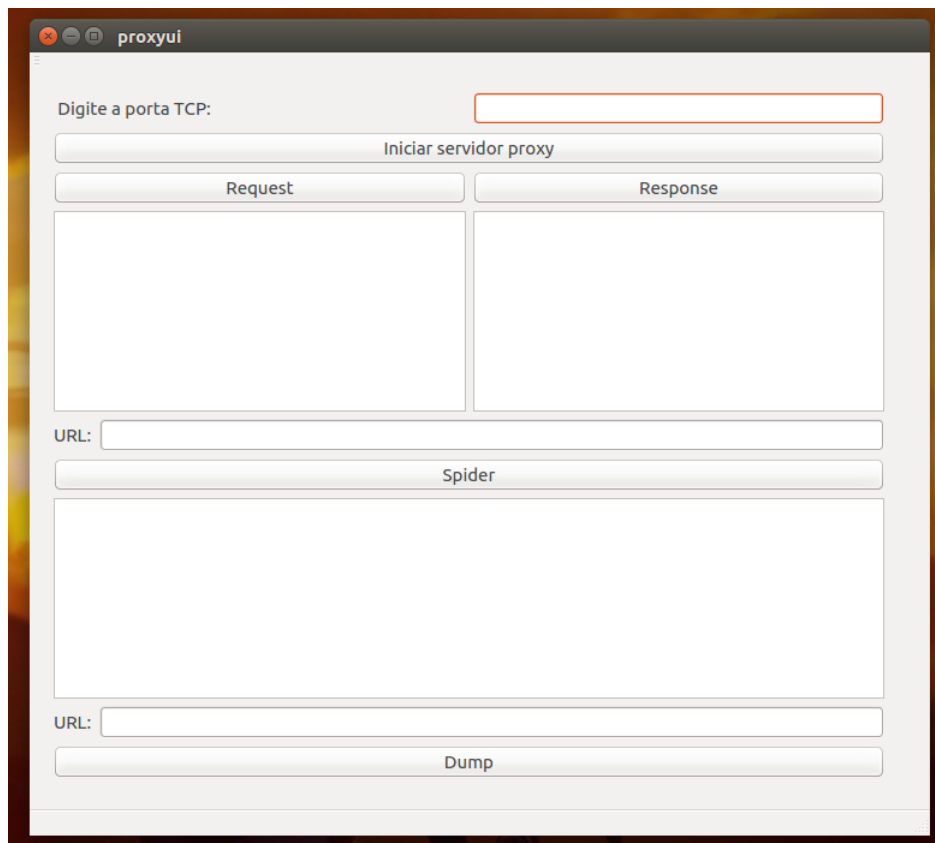


Figure 1: Tela inicial.

Após fornecer a porta TCP e clicar em "iniciar servidor proxy" o servidor proxy será iniciado e conectado na porta desejada. Caso nenhuma porta seja fornecida o servidor se conectará automaticamente na porta 8228.

O programa ficará estagnado até que o usuário tente acessar uma página web HTTP por um browser que esteja conectado na mesma porta que o servidor proxy. Após entrar com uma URL no browser o programa irá montar o request e mostrá-lo na tela:

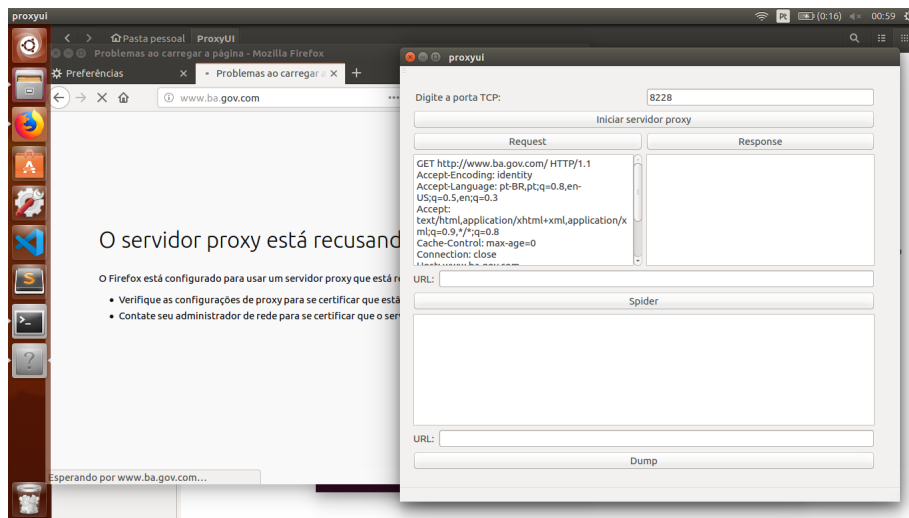


Figure 2: Request.

Ao clicar em "response" o programa mostra a mensagem de response do servidor e libera a conexão do computador do usuário com o servidor da página web desejada:

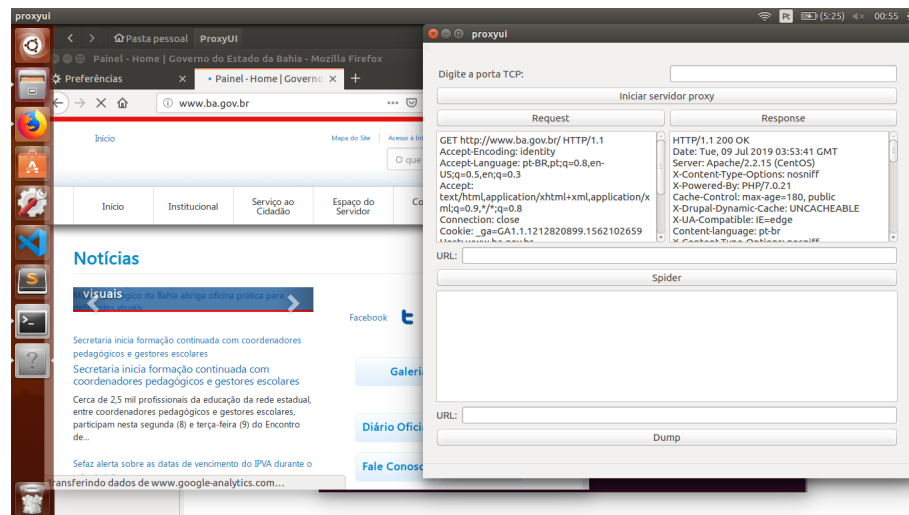


Figure 3: Response.

Para realizar o Spider é necessário colocar o endereço da página web desejada e clicar em "spider". Então a árvore será gerada:

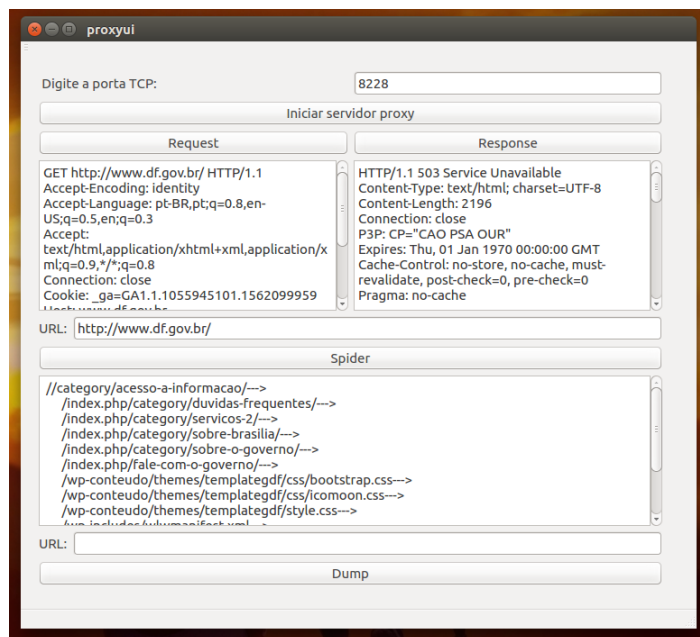


Figure 4: Spider.

Para realizar o Dump é necessário colocar o endereço da página web desejada e clicar em "dump". Então será criada uma pasta com uma cópia de cada um dos objetos presentes na página web desejada.

### 3.1 Bugs

O programa possui alguns bugs, a maioria deles relacionado a fazer o programa travar. Se o usuário tentar clicar em qualquer um dos botões antes de clicar em "iniciar o servidor proxy" o programa irá travar.

Se a página acessada for do tipo HTTPS, o programa irá travar ao tentar criar o request.

Se o usuário clicar no botão de request mais vezes do que a quantidade de requests existentes, o programa irá travar.