

# TP-Sockets-DNS-Python

Nome: Márcio Ribeiro Júnior

Matrícula: 223116349

## Introdução

O objetivo da atividade foi desenvolver, em Python, uma ferramenta semelhante ao comando **dig**, amplamente utilizado para consultas DNS, com a finalidade de compreender os fundamentos do protocolo DNS e sua interação com servidores. A implementação envolveu a criação de sockets para comunicação direta com servidores DNS, permitindo consultas para registros IPv4 (A) e IPv6 (AAAA). Além de reforçar conceitos de redes e sistemas distribuídos, a atividade buscou aplicar práticas de programação para construir uma solução funcional e educativa, capaz de realizar consultas DNS personalizadas sem depender de bibliotecas externas ou métodos de alto nível.

## Metodologias

As metodologias técnicas empregadas na construção da ferramenta foram focadas na implementação de baixo nível do protocolo DNS utilizando Python. Inicialmente, foi feita a criação e configuração de **sockets UDP**, com suporte para IPv4 e IPv6, a fim de estabelecer comunicação direta com servidores DNS públicos, como o do Google (8.8.8.8). A estrutura de pacotes DNS foi montada manualmente, seguindo o formato definido pelo protocolo, incluindo a codificação de cabeçalhos, identificadores, flags, número de questões e o corpo da consulta. Esses pacotes foram enviados aos servidores utilizando o método *sendto* dos sockets, enquanto as respostas foram capturadas com o método *recvfrom* para posterior interpretação. A decodificação e interpretação dos pacotes retornados incluiu a análise de cabeçalhos, registros de resposta e metadados. Esse processo, conduzido sem bibliotecas externas específicas para DNS, permitiu um entendimento aprofundado da estrutura binária dos pacotes DNS e dos fluxos de comunicação cliente-servidor no contexto de sistemas distribuídos.

## Resultados

```
PS C:\Users\marci\Desktop\dns-redes\DNS-STUDENT> py dns.py --type "A" --name www.google.com --server 8.8.8.8
Server Response
-----
Message ID: 4660
Response code: No error
Counts: Query 1, Answer 1, Authority 0, Additional 0
Question 1:
  Name: www.google.com
  Type: A
  Class: IN
Answer 1:
  Name: 0xc00c
  Type: A, Class: IN, TTL: 158
  RLength: 4 bytes
  Addr: 172.217.29.68 (IPv4)
```

Figura 1: Aplicação resolvendo dns para ipv4 do www.google.com

Os resultados obtidos demonstraram o funcionamento eficaz da ferramenta na realização de consultas DNS para registros do tipo A (IPv4) e AAAA (IPv6). A aplicação foi capaz de enviar pacotes DNS construídos manualmente para servidores públicos, como o Google DNS (8.8.8.8 e 2001:4860:4860::8888), e receber respostas contendo informações precisas sobre os endereços IP associados aos domínios consultados, como [www.google.com](http://www.google.com). Além disso, a ferramenta mostrou-se robusta na manipulação de pacotes binários e na interpretação das respostas,

exibindo dados relevantes, como tempo de resposta e metadados do servidor DNS. A implementação bem-sucedida confirmou o aprendizado prático sobre o protocolo DNS, comunicação via sockets e a manipulação direta de dados de baixo nível, cumprindo o objetivo proposto pela atividade.

## Análise

O DNS (Domain Name System) é um protocolo essencial para a internet, responsável por traduzir nomes de domínio legíveis, como [www.google.com](http://www.google.com), em endereços IP, necessários para a comunicação entre dispositivos. Ele opera em um modelo hierárquico e distribuído, onde servidores DNS resolvem consultas encaminhando-as a servidores de nível superior até obterem a resposta ou retornarem um erro. O protocolo utiliza principalmente o transporte UDP na porta 53, devido à sua eficiência para consultas rápidas.

O programa desenvolvido implementa o funcionamento do DNS ao criar pacotes DNS manualmente, incluindo cabeçalhos e consultas formatados no padrão binário especificado pela RFC 1035. Esses pacotes são enviados diretamente para servidores DNS, como o Google (8.8.8.8), utilizando sockets UDP, com suporte para IPv4 e IPv6. Após o envio, o programa recebe a resposta do servidor, decodifica os pacotes e interpreta os registros retornados, como endereços IP ou mensagens de erro. Essa abordagem permite simular o comportamento básico do comando **dig**, reforçando a compreensão do protocolo DNS e sua comunicação em redes.

## Conclusão

A implementação de uma ferramenta para consultas DNS utilizando Python proporcionou um aprendizado significativo sobre os fundamentos do protocolo DNS, incluindo sua estrutura hierárquica, funcionamento com registros IPv4 (A) e IPv6 (AAAA) e a troca de dados via sockets UDP. A construção manual de pacotes DNS e a decodificação de respostas permitiram um entendimento prático da estrutura binária do protocolo e dos desafios envolvidos na comunicação de baixo nível em redes. Além disso, a atividade reforçou habilidades de programação relacionadas à manipulação de dados binários, uso de sockets e tratamento de exceções em redes. Como resultado, foi possível compreender não apenas os aspectos teóricos do DNS, mas também sua aplicação prática, preparando o terreno para o desenvolvimento de sistemas mais complexos que dependam de comunicação em redes distribuídas.

## Referências:

- [1] A. S. Tanenbaum and D. J. Wetherall, *Computer Networks*. 2013.
- [2] L. L. Peterson and B. S. Davie, *Computer Networks: A Systems Approach*. Elsevier, 2011.