

UNIVERSIDADE FEDERAL DE SANTA CATARINA – UFSC  
CENTRO TECNOLÓGICO – CTC  
DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA – INE

INE5645 - PROGRAMAÇÃO PARALELA E DISTRIBUÍDA  
PROF. ODORICO MACHADO MENDIZABAL

GABRIEL CORREA TERRA  
PEDRO RUSCHEL BRESSAN  
VANESSA CUNHA

## TRABALHO 2: DNS PARTICIONADO

JUNHO, 2024

# Principais Decisões e Estratégias de Implementação

Neste projeto, implementamos um sistema de DNS particionado para associar nomes de domínios a endereços IP, distribuindo a carga de resolução entre múltiplos servidores. A principal decisão foi adotar um modelo de partição onde cada servidor é responsável por um conjunto específico de dados, com um nó roteador centralizando a comunicação entre clientes e servidores de partição. Utilizamos sockets UDP para garantir uma comunicação eficiente e direta entre os componentes e tentamos aplicar o padrão de projeto API Gateway conforme sugerido pelo professor em sala de aula. Utilizamos a linguagem Python para o desenvolvimento do trabalho e as seguintes bibliotecas:

- **socket:**
  - **Uso:** Permite a comunicação entre diferentes processos através da rede utilizando o protocolo UDP.
  - **Motivo:** Precisamos de uma forma de enviar e receber dados entre clientes, roteador e nós de partição. O módulo socket nos fornece essa capacidade.
- **threading:**
  - **Uso:** Criação e gerenciamento de múltiplas threads para execução paralela.
  - **Motivo:** Para lidar com múltiplas requisições simultâneas de clientes e para processar essas requisições em paralelo, melhorando a performance e a capacidade de resposta do sistema.
- **json:**
  - **Uso:** Manipulação de arquivos JSON para carregar configurações e dados DNS.
  - **Motivo:** JSON é um formato leve e de fácil manipulação para configuração e armazenamento de dados estruturados. Usamos essa biblioteca para ler arquivos de configuração e dados DNS.
- **time:**
  - **Uso:** Manipulação de tempo e criação de delays.
  - **Motivo:** Precisamos introduzir atrasos aleatórios entre as requisições dos clientes para simular um cenário mais realista e para medir a performance do sistema.
- **random:**
  - **Uso:** Geração de números aleatórios.
  - **Motivo:** Para criar intervalos aleatórios entre as requisições dos clientes, adicionando variabilidade à simulação e teste do sistema.
- **logging:**
  - **Uso:** Registro de logs de eventos e mensagens de erro.
  - **Motivo:** Para acompanhar o funcionamento do sistema, depurar problemas e manter um histórico de eventos e mensagens de erro que ocorrem durante a execução.

## Estrutura do Sistema

O sistema é composto pelos seguintes componentes principais:

- **Clientes:** Envia requisições de resolução de nomes de domínio ao roteador no método `enviar_requisicoes`.
- **Roteador (API Gateway):** Recebe as requisições dos clientes(`lidar_com_cliente`), autentica(`autenticar`), determina a partição correta(`determinar_particao`) e redireciona a requisição(`lidar_com_cliente`).

- **Nós de Partição:** Recebem as requisições do roteador, resolvem o domínio utilizando os dados armazenados localmente e respondem diretamente ao cliente(realiza as três ações em **lidar\_com\_requisicao**). Utilizamos a técnica de sharding vista na aula de padrões de projeto para setorizar as 26 partições em três nodos.

## Comunicação

Utilizamos sockets UDP para a comunicação ponto a ponto entre clientes, o roteador e os nós de partição, evitando o uso de recursos de comunicação coletiva. Cada requisição e resposta são exibidas na tela para acompanhamento (print).

## Arquivos de Configuração

- **configuracao\_particoes.json:** Contém a configuração dos nós de partição, incluindo endereços IP e intervalos de letras.
- **configuracao\_cliente.json:** Contém a configuração dos clientes, incluindo o endereço do roteador e a lista de requisições.

## Principais Decisões

- **Uso de Sockets UDP:**
  - **Motivo:** Optamos pela UDP devido à sua simplicidade e eficiência para comunicações curtas e rápidas.
  - **Benefício:** Reduz a sobrecarga de comunicação, já que o UDP não requer a criação de uma conexão persistente entre cada nó. Isso resulta em menor latência e maior velocidade na troca de mensagens.
- **Particionamento dos Dados:**
  - **Motivo:** Cada partição é responsável por domínios que começam com uma letra específica, facilitando a distribuição e a consulta dos dados.
  - **Benefício:** Essa abordagem divide a carga de trabalho entre múltiplos servidores, evitando que um único servidor se torne um ponto de falha ou um gargalo. Isso melhora a escalabilidade e a eficiência do sistema.
- **Autenticação Simples:**
  - **Motivo:** Utilizamos um token estático para autenticação, o que é suficiente para os propósitos de simulação e testes.
  - **Benefício:** Proporciona uma camada básica de segurança, garantindo que apenas clientes autenticados possam enviar requisições, sem adicionar complexidade desnecessária ao sistema.
- **Criação de um API Gateway:**
  - **Motivo:** Implementamos o roteador como um API Gateway para centralizar o recebimento e o encaminhamento das requisições dos clientes para os nós de partição apropriados.
  - **Benefício:** O uso de um API Gateway traz diversos benefícios:
  - **Centralização da Lógica de Roteamento:** O API Gateway simplifica a arquitetura, centralizando a lógica de roteamento e autenticação em um único ponto.
  - **Escalabilidade:** Facilita a escalabilidade horizontal, pois novas partições podem ser adicionadas sem alterar a lógica do cliente.
  - **Segurança:** Melhora a segurança ao fornecer um ponto único para a autenticação e autorização.

- **Manutenção e Monitoramento:** Facilita a manutenção e o monitoramento do sistema, permitindo que o tráfego de rede e os logs sejam gerenciados de forma centralizada.
- **Desacoplamento:** Desacopla os clientes das partições, permitindo mudanças na estrutura interna do sistema sem afetar os clientes.

## Instruções de Compilação e Execução

**\*\*Essas informações constam no READE.me também, se já passou por lá, pode pular essa seção!**

**Passo 1:** Instale as bibliotecas dependentes necessárias:

```
Unset  
pip3 install psutil
```

**Passo 2:** Criar Partições: É necessário popular as partições com informações e como são 26 criar um script para nos ajudar na criação deles, para isso vamos usar o seguinte comando:

```
Unset  
python3 gerar_particoes.py
```

**Passo 3:** Iniciar Partições: Para conectar os nós as respectivas partições utilize o comando:

```
Unset  
python3 iniciar_particoes.py
```

**Passo 4:** Iniciar Roteador: Agora para colocar o sistema de pé utilize o seguinte comando:

```
Unset  
python3 dns_particionado.py roteador
```

**Passo 5:** Executar Clientes:

- Apenas 1 Cliente:

```
Unset  
python3 dns_particionado.py cliente
```

- Inúmeros Clientes:

Unset

```
python3 teste_dns_particionado.py <num_clientes>
```

## Exemplos de Saídas de Execução e Conclusões

### Teste com um cliente

No arquivo **configuracao\_cliente.json**, temos a configuração de um cliente com 100 requisições, essas requisições serão paralelizadas mas respeitando o tempo de 1 a 2 segundos conforme o requisito do trabalho e faz requisições de sites mapeados nas partições e também não mapeados, segue evidências dos outputs durante os testes:

### Cliente

```
Cliente 0 recebeu resposta para dailymail.co.uk: Domínio não encontrado
o vanessacunha@MacBook-Pro-de-Vanessa T2-INE5645 % python3 dns_particionado.py cliente

Cliente enviando requisição: amazon.com.br
Resposta para amazon.com.br: 192.0.2.1
Cliente enviando requisição: amaamarozon.com.br
Resposta para amaamarozon.com.br: Domínio não encontrado
Cliente enviando requisição: americanas.com.br
Resposta para americanas.com.br: 192.0.2.2
Cliente enviando requisição: apple.com
Resposta para apple.com: 192.0.2.3
Cliente enviando requisição: abril.com.br
Resposta para abril.com.br: 192.0.2.4
Cliente enviando requisição: zoom.com.br
Resposta para zoom.com.br: Domínio não encontrado
Cliente enviando requisição: google.com
Resposta para google.com: Domínio não encontrado
Cliente enviando requisição: netflix.com
Resposta para netflix.com: Domínio não encontrado
Cliente enviando requisição: globo.com
Resposta para globo.com: Domínio não encontrado
Cliente enviando requisição: uol.com.br
Resposta para uol.com.br: Domínio não encontrado
Cliente enviando requisição: terra.com.br
Resposta para terra.com.br: Domínio não encontrado
Cliente enviando requisição: facebook.com
Resposta para facebook.com: Domínio não encontrado
Cliente enviando requisição: instagram.com
```

## Nodo

```
2024-06-18 21:52:32,595 - INFO - Partição respondendo: cctv.com -> Domínio não encontrado
2024-06-18 21:52:32,832 - INFO - Partição respondendo: tmz.com -> Domínio não encontrado
2024-06-18 21:52:32,879 - INFO - Partição respondendo: cctv.com -> Domínio não encontrado
2024-06-18 21:52:33,177 - INFO - Partição respondendo: tmz.com -> Domínio não encontrado
2024-06-18 21:52:33,206 - INFO - Partição respondendo: huan.tv -> Domínio não encontrado
2024-06-18 21:52:33,750 - INFO - Partição respondendo: cctv.com -> Domínio não encontrado
2024-06-18 21:52:33,851 - INFO - Partição respondendo: hollywoodlife.com -> Domínio não encontrado
2024-06-18 21:52:33,858 - INFO - Partição respondendo: fun.tv -> Domínio não encontrado
2024-06-18 21:52:33,881 - INFO - Partição respondendo: cntv.cn -> Domínio não encontrado
2024-06-18 21:52:34,326 - INFO - Partição respondendo: hollywoodlife.com -> Domínio não encontrado
2024-06-18 21:52:34,378 - INFO - Partição respondendo: justjared.com -> Domínio não encontrado
2024-06-18 21:52:34,437 - INFO - Partição respondendo: cntv.cn -> Domínio não encontrado
2024-06-18 21:52:34,498 - INFO - Partição respondendo: hollywoodlife.com -> Domínio não encontrado
2024-06-18 21:52:34,522 - INFO - Partição respondendo: justjared.com -> Domínio não encontrado
2024-06-18 21:52:34,769 - INFO - Partição respondendo: cntv.cn -> Domínio não encontrado
2024-06-18 21:52:35,151 - INFO - Partição respondendo: hollywoodlife.com -> Domínio não encontrado
2024-06-18 21:52:35,357 - INFO - Partição respondendo: cctv.com -> Domínio não encontrado
2024-06-18 21:52:35,376 - INFO - Partição respondendo: tmz.com -> Domínio não encontrado
2024-06-18 21:52:35,724 - INFO - Partição respondendo: huan.tv -> Domínio não encontrado
2024-06-18 21:52:35,815 - INFO - Partição respondendo: perezhilton.com -> Domínio não encontrado
2024-06-18 21:52:36,042 - INFO - Partição respondendo: tmz.com -> Domínio não encontrado
2024-06-18 21:52:36,089 - INFO - Partição respondendo: tmz.com -> Domínio não encontrado
2024-06-18 21:52:36,160 - INFO - Partição respondendo: perezhilton.com -> Domínio não encontrado
2024-06-18 21:52:36,352 - INFO - Partição respondendo: cctv.com -> Domínio não encontrado
2024-06-18 21:52:36,442 - INFO - Partição respondendo: hollywoodlife.com -> Domínio não encontrado
2024-06-18 21:52:36,622 - INFO - Partição respondendo: cctv.com -> Domínio não encontrado
2024-06-18 21:52:37,100 - INFO - Partição respondendo: tmz.com -> Domínio não encontrado
2024-06-18 21:52:37,199 - INFO - Partição respondendo: justjared.com -> Domínio não encontrado
```

## Roteador

```
2024-06-18 22:03:02,879 - INFO - Requisição recebida: baidu.com
2024-06-18 22:03:02,879 - INFO - Requisição encaminhada para baidu.com: {'intervalo': ['A', 'H'], 'ip': '127.0.0.1', 'porta': 10000}
2024-06-18 22:03:04,735 - INFO - Requisição recebida: sogou.com
2024-06-18 22:03:04,735 - INFO - Requisição encaminhada para sogou.com: {'intervalo': ['Q', 'Z'], 'ip': '127.0.0.1', 'porta': 10016}
2024-06-18 22:03:06,133 - INFO - Requisição recebida: weibo.com
2024-06-18 22:03:06,133 - INFO - Requisição encaminhada para weibo.com: {'intervalo': ['Q', 'Z'], 'ip': '127.0.0.1', 'porta': 10016}
2024-06-18 22:03:07,281 - INFO - Requisição recebida: qq.com
2024-06-18 22:03:07,281 - INFO - Requisição encaminhada para qq.com: {'intervalo': ['Q', 'Z'], 'ip': '127.0.0.1', 'porta': 10016}
2024-06-18 22:03:09,022 - INFO - Requisição recebida: wechat.com
2024-06-18 22:03:09,023 - INFO - Requisição encaminhada para wechat.com: {'intervalo': ['Q', 'Z'], 'ip': '127.0.0.1', 'porta': 10016}
2024-06-18 22:03:10,595 - INFO - Requisição recebida: alibaba.com
2024-06-18 22:03:10,596 - INFO - Requisição encaminhada para alibaba.com: {'intervalo': ['A', 'H'], 'ip': '127.0.0.1', 'porta': 10000}
2024-06-18 22:03:11,741 - INFO - Requisição recebida: jd.com
2024-06-18 22:03:11,742 - INFO - Requisição encaminhada para jd.com: {'intervalo': ['I', 'P'], 'ip': '127.0.0.1', 'porta': 10008}
2024-06-18 22:03:13,153 - INFO - Requisição recebida: taobao.com
2024-06-18 22:03:13,154 - INFO - Requisição encaminhada para taobao.com: {'intervalo': ['Q', 'Z'], 'ip': '127.0.0.1', 'porta': 10016}
2024-06-18 22:03:14,957 - INFO - Requisição recebida: tmall.com
2024-06-18 22:03:14,957 - INFO - Requisição encaminhada para tmall.com: {'intervalo': ['Q', 'Z'], 'ip': '127.0.0.1', 'porta': 10016}
```

## Teste com inúmeros clientes

No arquivo **teste\_dns\_particoes.py**, temos a definição de threads que representam clientes independentes que consomem o arquivo **configuracao\_cliente.json** e realiza as mesmas requisições pelo número de clientes que especificarmos na chamada **python3**

**teste\_dns\_particionado.py** <num\_clientes>, conforme a evidência:

## Cliente

```
o vanessacunha@MacBook-Pro-de-Vanessa T2-INE5645 % python3 teste_dns_particionado.py 10
Cliente 0 enviando requisição: amazon.com.br
Cliente 1 enviando requisição: amazon.com.br
Cliente 2 enviando requisição: amazon.com.br
Cliente 3 enviando requisição: amazon.com.br
Cliente 4 enviando requisição: amazon.com.br
Cliente 5 enviando requisição: amazon.com.br
Cliente 6 enviando requisição: amazon.com.br
Cliente 8 enviando requisição: amazon.com.br
Cliente 7 enviando requisição: amazon.com.br
Cliente 9 enviando requisição: amazon.com.br
Cliente 0 recebeu resposta para amazon.com.br: 192.0.2.1
Cliente 1 recebeu resposta para amazon.com.br: 192.0.2.1
Cliente 2 recebeu resposta para amazon.com.br: 192.0.2.1
Cliente 3 recebeu resposta para amazon.com.br: 192.0.2.1
Cliente 4 recebeu resposta para amazon.com.br: 192.0.2.1
Cliente 5 recebeu resposta para amazon.com.br: 192.0.2.1
Cliente 6 recebeu resposta para amazon.com.br: 192.0.2.1
Cliente 8 recebeu resposta para amazon.com.br: 192.0.2.1
Cliente 7 recebeu resposta para amazon.com.br: 192.0.2.1
Cliente 9 recebeu resposta para amazon.com.br: 192.0.2.1
Cliente 0 enviando requisição: amaamarozon.com.br
Cliente 0 recebeu resposta para amaamarozon.com.br: Domínio não encontrado
Cliente 1 enviando requisição: amaamarozon.com.br
Cliente 1 recebeu resposta para amaamarozon.com.br: Domínio não encontrado
Cliente 9 enviando requisição: amaamarozon.com.br
Cliente 9 recebeu resposta para amaamarozon.com.br: Domínio não encontrado
Cliente 8 enviando requisição: amaamarozon.com.br
```

## Nodo

```
2024-06-18 22:10:08,051 - INFO - Partição respondendo: facebook.com -> Domínio não encontrado
2024-06-18 22:10:08,065 - INFO - Partição respondendo: instagram.com -> 192.0.2.3
2024-06-18 22:10:08,129 - INFO - Partição respondendo: linkedin.com -> Domínio não encontrado
2024-06-18 22:10:08,372 - INFO - Partição respondendo: instagram.com -> 192.0.2.3
2024-06-18 22:10:08,435 - INFO - Partição respondendo: facebook.com -> Domínio não encontrado
2024-06-18 22:10:08,455 - INFO - Partição respondendo: instagram.com -> 192.0.2.3
2024-06-18 22:10:08,850 - INFO - Partição respondendo: instagram.com -> 192.0.2.3
2024-06-18 22:10:09,066 - INFO - Partição respondendo: linkedin.com -> Domínio não encontrado
2024-06-18 22:10:09,328 - INFO - Partição respondendo: twitter.com -> Domínio não encontrado
2024-06-18 22:10:09,543 - INFO - Partição respondendo: instagram.com -> 192.0.2.3
2024-06-18 22:10:09,551 - INFO - Partição respondendo: linkedin.com -> Domínio não encontrado
2024-06-18 22:10:09,613 - INFO - Partição respondendo: twitter.com -> Domínio não encontrado
2024-06-18 22:10:09,701 - INFO - Partição respondendo: linkedin.com -> Domínio não encontrado
2024-06-18 22:10:09,768 - INFO - Partição respondendo: linkedin.com -> Domínio não encontrado
2024-06-18 22:10:10,003 - INFO - Partição respondendo: linkedin.com -> Domínio não encontrado
2024-06-18 22:10:10,145 - INFO - Partição respondendo: twitter.com -> Domínio não encontrado
2024-06-18 22:10:10,201 - INFO - Partição respondendo: instagram.com -> 192.0.2.3
2024-06-18 22:10:10,551 - INFO - Partição respondendo: linkedin.com -> Domínio não encontrado
2024-06-18 22:10:10,809 - INFO - Partição respondendo: twitter.com -> Domínio não encontrado
2024-06-18 22:10:10,954 - INFO - Partição respondendo: github.com -> Domínio não encontrado
2024-06-18 22:10:10,990 - INFO - Partição respondendo: linkedin.com -> Domínio não encontrado
2024-06-18 22:10:11,061 - INFO - Partição respondendo: github.com -> Domínio não encontrado
2024-06-18 22:10:11,380 - INFO - Partição respondendo: twitter.com -> Domínio não encontrado
2024-06-18 22:10:11,575 - INFO - Partição respondendo: twitter.com -> Domínio não encontrado
2024-06-18 22:10:11,669 - INFO - Partição respondendo: twitter.com -> Domínio não encontrado
2024-06-18 22:10:11,699 - INFO - Partição respondendo: github.com -> Domínio não encontrado
2024-06-18 22:10:11,842 - INFO - Partição respondendo: linkedin.com -> Domínio não encontrado
2024-06-18 22:10:11,890 - INFO - Partição respondendo: twitter.com -> Domínio não encontrado
```

## Roteador

```
2024-06-18 22:11:52,605 - INFO - Requisição recebida: people.com.cn
2024-06-18 22:11:52,606 - INFO - Requisição encaminhada para people.com.cn: {'intervalo': ['I', 'P'], 'ip': '127.0.0.1', 'porta': 10008}
2024-06-18 22:11:52,672 - INFO - Requisição recebida: people.com.cn
2024-06-18 22:11:52,672 - INFO - Requisição encaminhada para people.com.cn: {'intervalo': ['I', 'P'], 'ip': '127.0.0.1', 'porta': 10008}
2024-06-18 22:11:52,759 - INFO - Requisição recebida: people.com.cn
2024-06-18 22:11:52,760 - INFO - Requisição encaminhada para people.com.cn: {'intervalo': ['I', 'P'], 'ip': '127.0.0.1', 'porta': 10008}
2024-06-18 22:11:52,893 - INFO - Requisição recebida: people.com.cn
2024-06-18 22:11:52,893 - INFO - Requisição encaminhada para people.com.cn: {'intervalo': ['I', 'P'], 'ip': '127.0.0.1', 'porta': 10008}
2024-06-18 22:11:53,049 - INFO - Requisição recebida: xinhuanet.com
2024-06-18 22:11:53,049 - INFO - Requisição encaminhada para xinhuanet.com: {'intervalo': ['Q', 'Z'], 'ip': '127.0.0.1', 'porta': 10016}
2024-06-18 22:11:53,062 - INFO - Requisição recebida: sina.com.cn
2024-06-18 22:11:53,062 - INFO - Requisição encaminhada para sina.com.cn: {'intervalo': ['Q', 'Z'], 'ip': '127.0.0.1', 'porta': 10016}
2024-06-18 22:11:53,252 - INFO - Requisição recebida: eastday.com
2024-06-18 22:11:53,252 - INFO - Requisição encaminhada para eastday.com: {'intervalo': ['A', 'H'], 'ip': '127.0.0.1', 'porta': 10000}
2024-06-18 22:11:53,406 - INFO - Requisição recebida: 163.com
2024-06-18 22:11:53,406 - WARNING - Nenhuma partição disponível para 163.com
2024-06-18 22:11:53,606 - INFO - Requisição recebida: eastday.com
2024-06-18 22:11:53,606 - INFO - Requisição encaminhada para eastday.com: {'intervalo': ['A', 'H'], 'ip': '127.0.0.1', 'porta': 10000}
2024-06-18 22:11:53,803 - INFO - Requisição recebida: people.com.cn
2024-06-18 22:11:53,803 - INFO - Requisição encaminhada para people.com.cn: {'intervalo': ['I', 'P'], 'ip': '127.0.0.1', 'porta': 10008}
```

## Conclusões

A implementação do serviço de DNS particionado demonstrou a eficácia do uso de sharding para distribuir a carga de trabalho entre múltiplos servidores. Utilizando sockets UDP para comunicação, garantimos uma transmissão rápida e eficiente entre clientes, roteador e nós de partição. O API Gateway centralizou o ponto de entrada das requisições, garantindo autenticação consistente e determinação correta das partições. A configuração flexível através de arquivos JSON facilitou a simulação de diferentes cenários de teste, permitindo ajustar rapidamente o ambiente e avaliar a performance sob variadas condições.