



**UNIVERSIDADE FEDERAL DE SANTA CATARINA  
CENTRO DE CIÊNCIAS, TECNOLOGIAS E SAÚDE  
DEPARTAMENTO DE COMPUTAÇÃO  
CURSO DE ENGENHARIA DE COMPUTAÇÃO**

## **Servidor Web**

**Bernardo Pandolfi Costa**

Disciplina: Sistemas Operacionais  
Professor: Roberto Rodrigues Filho

Araranguá  
30 de outubro de 2023

## Conteúdo

<b>1</b>	<b>Exercício 1</b>	<b>2</b>
1.1	Servidor . . . . .	2
1.2	Cliente . . . . .	2
<b>2</b>	<b>Exercício 2</b>	<b>2</b>
<b>3</b>	<b>Exercício 3</b>	<b>2</b>
3.1	Servidor . . . . .	2
3.2	Cliente . . . . .	3

## 1 Exercício 1

Para este problema, foram feitas alterações aos códigos exemplos no arquivo da lista, sendo a alteração principal foi a adição de um loop, tanto para o cliente quanto para o servidor.

### 1.1 Servidor

No programa do servidor, o socket continuará ouvindo a porta 2023 até que o cliente envie uma mensagem. Ao receber uma mensagem, o código a imprimirá no terminal e uma resposta é enviada ao cliente e o programa retorna ao estado de ouvindo. Existe um comando que o cliente pode enviar que é o 'Stop'. O comando 'Stop' é interpretado pelo servidor para quebrar o loop e desligar o servidor.

### 1.2 Cliente

Para o programa do cliente, um loop similar foi implementado. Neste, o usuário será confrontado por um input para digitar a mensagem a ser enviada ao servidor. O programa também imprime a resposta enviada pelo servidor e possui um passo de verificação para analisar se o servidor foi desligado ou não. Esta verificação é feita da seguinte forma: se nenhuma resposta chegou, assume-se que o servidor não esteja no ar. Se a mensagem retornada é a resposta ao 'Stop', isto é, um aviso de desligamento do servidor, consideramos que o servidor será desligado. Quando esta verificação é verdadeira, o programa do cliente finaliza.

## 2 Exercício 2

A resolução deste exercício foi feita a partir de uma adaptação do código do servidor do exercício 1. O loop continua implementado, porém neste caso a resposta será sempre "HTTP/1.0 200 OK \n\n<h1>Hello World</h1>". Como o cliente neste caso é o browser, não foi implementado um programa para o cliente.

## 3 Exercício 3

Para o exercício 3, dois arquivos novos tiveram de ser criados: 404.html e index.html. Esses arquivos são usados para enviar a página html como resposta ao cliente com base na rota especificada na requisição. A documentação linha por linha do código está em forma de comentários no próprio código.

### 3.1 Servidor

O servidor implementado para este problema envolve quatro funções. Uma para cada rota e uma para lidar com a requisição do cliente. Existem três rotas no servidor, a home (acessada pelo / ou por /index.html), a /request que apenas apresenta a requisição do cliente no navegador e a 404, para qualquer rota inexistente. A função que

lida com a requisição é a função colocada em cada thread. Sempre que uma requisição chega ao servidor, uma nova thread é criada a partir desta função. Essa função checa a rota requisitada pelo cliente e retorna uma resposta chamando a respectiva função da rota.

## **3.2 Cliente**

No momento que o código do cliente é rodado, o usuário será pedido a quantidade de requisições que serão feitas ao mesmo tempo para o servidor. Cada requisição, automaticamente, pedirá por diferentes rotas para testar cada rota implementada.