

UNIVERSIDADE DE AVEIRO

SISTEMAS OPERATIVOS

# Multi-Threaded Web Server

Manual de Utilizador

**Autores:**

Diogo Ruvio (NMec: 126498)  
David Cálix (NMec: 125043)

Dezembro 2025

# Conteúdo

<b>1</b>	<b>Introdução</b>	<b>2</b>
<b>2</b>	<b>Pré-requisitos</b>	<b>2</b>
2.1	Essenciais (Para Executar) . . . . .	2
2.2	Ferramentas de Teste (Recomendado) . . . . .	2
2.2.1	Comando de Instalação Rápida . . . . .	2
<b>3</b>	<b>Compilação</b>	<b>2</b>
3.1	Compilar o Projeto . . . . .	3
3.2	Limpar Ficheiros Temporários . . . . .	3
<b>4</b>	<b>Configuração (server.conf)</b>	<b>3</b>
<b>5</b>	<b>Execução</b>	<b>3</b>
<b>6</b>	<b>Como Testar</b>	<b>3</b>
6.1	Via Browser . . . . .	3
6.2	Via Linha de Comandos . . . . .	3
<b>7</b>	<b>Resolução de Problemas</b>	<b>4</b>

# 1 Introdução

O **Multi-Threaded Web Server** é um servidor web de alto desempenho desenvolvido em C, capaz de processar múltiplos pedidos simultaneamente utilizando uma arquitetura Master-Worker e *thread pools*.

Este manual descreve os passos necessários para compilar, configurar e executar o servidor em ambiente Linux.

## 2 Pré-requisitos

Para garantir o funcionamento correto do servidor e da suite de testes, certifique-se que o sistema possui o software listado abaixo.

### 2.1 Essenciais (Para Executar)

Software necessário para compilar e iniciar o servidor:

- **Sistema Operativo:** Linux (Ubuntu 20.04+ ou similar)
- **Compilador:** GCC (com suporte a threads)
- **Build System:** Make

### 2.2 Ferramentas de Teste (Recomendado)

Para executar os scripts de teste automatizados (`run_all_tests.sh`), é necessário instalar:

- **curl:** Para realizar pedidos HTTP de teste.
- **apache2-utils:** Contém a ferramenta `ab` para testes de carga.
- **valgrind:** Para deteção de fugas de memória.
- **bc:** Calculadora de linha de comandos (necessária para os scripts de teste).

#### 2.2.1 Comando de Instalação Rápida

Pode instalar todas as dependências necessárias no Ubuntu/Debian com o seguinte comando:

```
$ sudo apt-get update
$ sudo apt-get install build-essential curl apache2-utils valgrind
  bc
```

## 3 Compilação

O projeto inclui um `Makefile` para automatizar o processo de compilação.

### 3.1 Compilar o Projeto

Abra o terminal na raiz do projeto e execute:

```
$ make
```

### 3.2 Limpar Ficheiros Temporários

Para remover os executáveis e ficheiros objeto (.o):

```
$ make clean
```

## 4 Configuração (server.conf)

O comportamento do servidor é controlado pelo ficheiro `server.conf`. Abaixo descrevem-se os parâmetros disponíveis:

Parâmetro	Padrão	Descrição
PORT	8080	Porta TCP de escuta
NUM_WORKERS	4	Número de processos trabalhadores
THREADS_PER_WORKER	10	Threads por processo
DOCUMENT_ROOT	./www	Pasta dos ficheiros HTML
LOG_FILE	access.log	Caminho do ficheiro de registo

Tabela 1: Parâmetros de Configuração

## 5 Execução

Para iniciar o servidor com a configuração padrão:

```
$ ./server
```

O servidor iniciará e apresentará o PID do processo Master. Para encerrar o servidor pressione **Ctrl+C**.

## 6 Como Testar

Uma vez o servidor em execução, pode testar o funcionamento de duas formas:

### 6.1 Via Browser

Abra o seu navegador e aceda a:

`http://localhost:8080/index.html`

### 6.2 Via Linha de Comandos

Pode utilizar o comando `curl` para ver os cabeçalhos da resposta:

```
$ curl -v http://localhost:8080/index.html
```

## 7 Resolução de Problemas

- **Erro "Address already in use":** Aguarde alguns segundos ou altere a PORT no ficheiro de configuração.
- **Erro 403 Forbidden:** Verifique se o ficheiro tem permissões de leitura no sistema operativo.