

# Manual do Utilizador

## 1. Visão Geral

O **Servidor HTTP Concorrente** é um servidor HTTP multi-processo e multi-thread implementado em C que utiliza:

- Um **processo master** para gerenciar os workers e recursos compartilhados
- Vários **processos worker**, cada um a executar um pool de threads para tratar requisições de forma concorrente
- **Memória compartilhada e semáforos** para estatísticas e gerenciamento da fila de requisições
- **Parâmetros configuráveis** através do `server.conf`

O servidor atende arquivos estáticos do diretório `www/` e suporta páginas de erro personalizadas (403.html, 404.html, 500.html, 503.html).

---

## 2. Compilação

### 2.1 Pré-requisitos

Certifique-se de que as seguintes ferramentas estão instaladas:

- `gcc` (suporta C99 ou superior)
- `make`
- `bash`

### 2.2 Compilar o Servidor

1. Abra um terminal no diretório raiz do projeto:

```
cd path/to/concurrent-http-server
```

2. Execute make para compilar o projeto:

```
make(apenas compila) ou make run(compila e inicia o servidor)
```

3. Após a compilação, um executável chamado server será criado.
4. Opcional: Para remover arquivos compilados:

```
make clean
```

---

## 3. Configuração

O servidor lê a sua configuração a partir de server.conf. Valores padrão:

```
PORT=8080
DOCUMENT_ROOT=./www
NUM_WORKERS=4
THREADS_PER_WORKER=10
MAX_QUEUE_SIZE=100
LOG_FILE=server.log
CACHE_SIZE_MB=10
TIMEOUT_SECONDS=30
```

```
# Hosts Virtuais
DEFAULT_VHOST=localhost
VHOST_localhost=./www
```

### 3.1 Editando a Configuração

Abra server.conf num editor de texto:

```
nano server.conf
```

Modifique os parâmetros conforme necessário:

- PORT – porta do servidor
- NUM\_WORKERS – número de processos worker
- THREADS\_PER\_WORKER – threads por processo worker
- MAX\_QUEUE\_SIZE – tamanho máximo da fila de requisições
- CACHE\_SIZE\_MB – tamanho do cache LRU em megabytes
- DOCUMENT\_ROOT – pasta que contem os arquivos web

Guarde e feche o ficheiro.

---

## 4. Executar o Servidor

Inicie o servidor:

```
./server
```

- O processo master criará os processos worker e gerenciará os recursos compartilhados.
- Os workers tratam das conexões usando `SO_REUSEPORT`.
- Pressione **Ctrl+C** para encerrar o servidor de forma segura.

### 4.1 Aceder ao Servidor

Abra um navegador e visite:

```
http://localhost:8080
```

Os arquivos estáticos serão servidos a partir do diretório `www/`.

---

## 5. Comportamento do Servidor

- **Encerramento seguro:** Ao pressionar `Ctrl+C` ou enviar `SIGTERM`, os workers terminam, semáforos e memória compartilhada são libertados, e o servidor fecha os sockets abertos.
- **Estatísticas:** A cada 30 segundos, o servidor imprime estatísticas:

```
Total de requisições
Requisições completadas
Bytes transferidos
Tempo médio de resposta
Códigos de status (200, 403, 404, 500, 503)
Conexões ativas
Tamanho da fila
```

- **Logs:** As ações são registadas no arquivo definido em `server.conf` (`LOG_FILE`).
- 

## 6. Testes

### 6.1 Testes via Makefile (recomendado)

**Teste básico** e funcionalidade essencial:

`make testSimple`

Este alvo compila o servidor, arranca-o e executa a suite de testes funcionais e de concorrência mais rápidos.

**Teste completo** (inclui carga, sincronização e stress):

`make testFull`

Este alvo executa todos os testes, incluindo os mais longos (Helgrind/DRD, Valgrind, stress de vários minutos).

## 6.2 Testes por script (opcional)

Caso seja preciso fazer **testes individuais**:

1. Dê permissão de execução ao script de teste:

`chmod +x testdesejado.sh`

2. Compile o código:

`make`

3. Abra o servidor:

`./server`

4. Execute o teste desejado:

`./testdesejado.sh`

## 6.3 Testes manuais (opcional)

1. Inicie o servidor:

`./server` ou `make run`

2. Abra o navegador e aceda:

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

3. Teste entrar em páginas inexistentes ou em “privado.html”.

---

## 7. Observações

- Certifique-se de que o diretório `www/` contém um arquivo `index.html`.
- As páginas de erro personalizadas (`403.html`, `404.html`, `500.html`, `503.html`) são usadas automaticamente.
- Evite usar portas abaixo de 1024 a menos que esteja executando como root.
- Tenha a certeza que o ficheiro `privado.html` não tem permissões, executar este código para tirar as permissões `"chmod 000 www/privado.html"`