

Aluno: João Victor Fernandes da Silveira
Matrícula: 20230012298

Mapeamento de Requisitos para Código

1. Pthreads para clientes/conexões

O servidor cria uma thread dedicada para cada cliente conectado usando **pthread_create**, permitindo atendimento concorrente. Uma thread separada (**broadcast**) processa mensagens da fila.

Arquivos: server.c (threads por cliente + broadcast), client.c (thread para recebimento)

2. Exclusão Mútua

Múltiplos mutexes protegem estruturas compartilhadas: **clients_mutex** para a lista de clientes, mutex interno na **threadsafe_queue** para a fila, e mutex no **libtslog** para operações de logging.

Arquivos: server.c, threadsafe_queue.c, libtslog.c

3. Semáforos e Condvars

Semáforo **available_slots** controla o número máximo de conexões simultâneas. Condition variables **not_empty** e **not_full** na fila thread-safe permitem bloqueio eficiente quando vazia/cheia.

Arquivos: server.c (semáforo), threadsafe_queue.c (condvars)

4. Monitores

A **threadsafe_queue_t** é um monitor clássico que encapsula uma fila circular com sincronização interna. Oferece operações atômicas **queue_push** e **queue_pop** com espera bloqueante.

Arquivos: threadsafe_queue.h/c (implementação), server.c (uso)

5. Sockets

Socket TCP/IP na porta 8080 com todas as operações padrão: **socket()**, **bind()**, **listen()**, **accept()** no servidor; **socket()**, **connect()** no cliente.

Arquivos: server.c, client.c

6. Gerenciamento de Recursos

Limpeza adequada de todos os recursos: sockets fechados, memória liberada, semáforos e mutex destruídos. Uso de **queue_destroy** para o monitor.

Arquivos: server.c (principal), threadsafe_queue.c

7. Tratamento de Erros

Verificação sistemática de retornos de funções com mensagens de erro descritivas. Logs indicam falhas específicas e o sistema continua operando quando possível.

Arquivos: server.c, client.c

8. Logging Concorrente

Biblioteca libtslog com operações thread-safe usando mutex interno. Logs coloridos por nível (*MSG*, *WARNING*, *ERROR*) integrados em todo o sistema.

Arquivos: libtslog.c (implementação), server.c, client.c (uso)