



FACULDADE DE  
CIÊNCIAS E TECNOLOGIA  
UNIVERSIDADE DE  
**COIMBRA**

## Projeto RC

Alexandre José Martins Rodrigues nº2022249408

João Pedro Lopes Barraca nº2022244716

## Introdução

Este projeto implementa um sistema cliente-servidor utilizando sockets em C. O sistema permite a interação entre um cliente e um servidor através de comunicação UDP e TCP. O cliente pode autenticar-se, realizar operações administrativas e de gestão de utilizadores, enquanto o servidor gere as requests e realiza as operações solicitadas.

## Estrutura Geral

O sistema é composto por dois componentes principais:

1. **Cliente:** Programa que envia requests ao servidor e processa as respostas recebidas.
2. **Servidor:** Programa que recebe as requests dos clientes, processa-as, e envia as respostas apropriadas.

## Comunicação entre Cliente e Servidor

- **UDP:** Utilizado para autenticação de utilizadores e execução de comandos administrativos.
- **TCP:** Utilizado para operações específicas, como listagem e gestão de turmas.

## Cliente

O cliente é responsável por:

- Solicitar login ao utilizador.
- Enviar as credenciais de login para o servidor via UDP. Se autenticado, permitir que o utilizador insira comandos administrativos, como adicionar ou remover utilizadores, e listar utilizadores.
- Enviar os comandos administrativos para o servidor e exibir as respostas recebidas.

## Fluxo de Operação do Cliente

1. **Login:**
  - O utilizador é solicitado a inserir as suas credenciais de login.
  - As credenciais são enviadas ao servidor via UDP.
  - O cliente aguarda a resposta do servidor, que pode ser "passou" ou "parou".

## 2. Operações Administrativas:

- Após a autenticação, o cliente pode inserir comandos administrativos.
- Comandos válidos incluem: **ADD\_USER**, **DEL**, **LIST**, **QUIT\_SERVER**.
- Cada comando é enviado ao servidor via UDP, e a resposta do servidor é exibida ao utilizador.
- Para o comando **LIST**, o cliente continua a receber e exibir linhas até receber a mensagem "SAIR" do loop.

### Servidor

O servidor é responsável por:

- Receber e processar requests de clientes.
- Autenticar utilizadores com base num ficheiro de dados.
- Executar comandos administrativos.
- Gerir uma lista de turmas utilizando memória partilhada para operações de criação, listagem e inscrição de alunos.

### Estrutura do Servidor

#### 1. Processamento de Sinal:

- Utiliza sinais para gerir interrupções e encerrar o servidor corretamente.

#### 2. Autenticação de Utilizadores:

- Recebe as credenciais de login via UDP.
- Verifica as credenciais com base num ficheiro local (**dom.txt**).
- Envia a resposta de autenticação ao cliente.

#### 3. Execução de Comandos Administrativos:

- Recebe comandos administrativos via UDP.
- Processa cada comando, realizando operações como adicionar ou remover utilizadores, e listar todos os utilizadores.

#### 4. Gestão de Turmas:

- Utiliza memória partilhada para armazenar e gerir informações sobre turmas.
- Implementa funções para criação de turmas, inscrição de alunos, e listagem de turmas.

- Processa requisições específicas dos clientes via TCP.

### **Memória Partilhada**

- O servidor utiliza memória partilhada para armazenar a lista de turmas, permitindo acesso eficiente e sincronizado às informações.
- A memória é gerida através de funções para criação, busca, inserção e remoção de turmas e alunos.

### **Funções Específicas**

- **Adicionar Utilizador (ADD\_USER):** Adiciona um novo utilizador ao sistema, verificando se já existe um utilizador com o mesmo nome.
- **Remover Utilizador (DEL):** Remove um utilizador do sistema.
- **Listar Utilizadores (LIST):** Lista todos os utilizadores registados.
- **Criar Turma (CREATE\_CLASS):** Permite que professores criem turmas.
- **Inscriver Aluno (SUBSCRIBE\_CLASS):** Permite que alunos se inscrevam em turmas previamente criadas.
- **Listar Turmas (LIST\_CLASSES):** Lista todas as turmas disponíveis.
- **Listar Turmas Inscritas (LIST\_SUBSCRIBED):** Lista as turmas que o aluno está inscrito.

### **Conclusão**

O sistema cliente-servidor desenvolvido permite a realização de diversas operações administrativas e de gestão de turmas de forma eficiente e segura. A combinação de comunicação UDP e TCP proporciona flexibilidade e robustez na troca de informações entre cliente e servidor. A utilização de memória partilhada no servidor facilita a gestão das turmas, garantindo consistência e integridade dos dados.