

Instituto Superior de Engenharia de Lisboa  
Licenciatura em Engenharia Informática e de Computadores  
**Programação de Sistemas Computacionais**  
Inverno de 2025/2026  
Segundo Trabalho Prático

---

Construa os programas e as bibliotecas indicados na Parte II, usando a linguagem C, tendo o cuidado de eliminar repetições no código fonte e de isolar funcionalidades distintas em diferentes ficheiros fonte. Entregue o código desenvolvido, devidamente indentado e comentado, bem como os *makefile* para gerar os executáveis e as bibliotecas a partir do código fonte. Assegure-se de que o compilador não emite qualquer aviso sobre o seu código com a opção **-Wall** activa, e de que no final da execução do programa não existem recursos por libertar (memória alocada dinamicamente e ficheiros abertos). Deve verificar essa situação com o utilitário Valgrind. Em caso de erro irrecuperável, o programa não deve terminar descontroladamente, deve, no mínimo, emitir uma mensagem informativa e em seguida terminar. O código desenvolvido será valorizado segundo os seguintes critérios, em importância decrescente: correção, eficiência, clareza.

A realização deste trabalho consiste na programação dos artefactos especificados ao longo das alíneas, destinados à construção de um programa para manipulação de dados em tabela, com inspiração em folhas de cálculo como Microsoft Excel ou LibreOffice Calc.

Como o trabalho está organizado de modo que os exercícios seguintes utilizem artefactos desenvolvidos atrás, é possível que a dada altura, se constate a inadequação dos desenvolvimentos anteriores. Se isso acontecer, deve, num processo iterativo, proceder à alteração dos exercícios já realizados, de modo a adequá-los às necessidades dos exercícios seguintes.

A entrega deve consistir num ficheiro em formato **zip** contendo uma diretoria como raiz, com quatro diretórios no seu interior, uma por cada exercício. O conteúdo de cada diretoria deve ser composto pelo código fonte, *makefile* e eventuais bibliotecas necessárias para gerar o programa executável especificado nesse exercício.

---

## Parte I - Preparação do ambiente de desenvolvimento

### libcsv

Para auxiliar no carregamento de dados a partir de ficheiros em formato CSV deverá utilizar a biblioteca de fonte aberta **libcsv**.

Instalação: `$ sudo apt install libcsv-dev`

Documentação: [github.com/rgamble/libcsv/](https://github.com/rgamble/libcsv/) ; `man csv`

### Valgrind

Para verificar se um programa libera toda a memória alocada dinamicamente deverá utilizar a ferramenta **valgrind**.

Instalação: `$ sudo apt-get install valgrind`

Documentação: `$ man valgrind`

---

## Parte II - Realização

1.
  - a. Defina a **struct table** para suporte em memória dos dados de uma tabela. A tabela tem duas dimensões, as colunas são identificadas por letras e as linhas são identificadas por números. O número de colunas é fixado na criação da tabela, num máximo de 26 colunas. O número de linhas é dinâmico, pode aumentar ou diminuir ao longo do processamento. Na concretização da **struct table** deve-se minimizar a memória alocada e não utilizada.

```
struct table { ...campos a definir... };
```

- b. Programe a função **table\_load\_csv** que cria uma tabela (uma instância de **struct table**) preenchida com os dados lidos de um ficheiro em formato CSV. Considere que o ficheiro está sempre bem formado.

```
struct table *table_load_csv(const char *filename);
```

- c. Programe a função **table\_save\_csv** que escreve em ficheiro, em formato CSV, o conteúdo de informação de uma tabela **struct table**.

```
void table_save_csv(const struct table *table, const char *filename);
```

- d. Para teste das funções anteriores realize um programa que crie uma réplica de um ficheiro CVS. O programa tem dois parâmetros: o ficheiro criado e o ficheiro fonte.

- e. Programe a função **table\_filter** que cria uma nova tabela com as linhas que satisfazem um predicado. O parâmetro **predicate** recebe o ponteiro para a função onde é programada a verificação do predicado; o parâmetro **context** recebe dados de contexto necessários à verificação.

```
struct table *table_filter(const struct table *table,  
                           bool (*predicate)(const void *row, const void *context),  
                           const void *context);
```

- f. Para teste da função **table\_filter** realize um programa que recebendo o ficheiro **frutas.csv** produza um novo ficheiro com os frutos de preço inferior a um Euro.

Os programas pedidos nas alíneas d) e f) devem ser gerados sob o controle do programa **make**. Os respetivos ficheiros (ou ficheiro) **makefile** devem acompanhar a entrega.

2. Construa uma biblioteca de ligação dinâmica (*shared object*) com as funções definidas no ponto anterior e com as funções auxiliares necessárias. Na organização do código, tenha em consideração que deve evitar repetições de código fonte.

Utilize o programa de teste da alínea f) para verificar o correto funcionamento da biblioteca.

3. Desenvolva um programa que permanece em execução, aceitando comandos no terminal.

O programa deve aceitar os seguintes comandos:

**help** Listar comandos existentes.

**exit** Terminar o programa.

**load <filename>** Carregar na tabela o conteúdo do ficheiro **filename**.

**save <filename>** Salvar a tabela no ficheiro **filename**.

**show <col><row>:<col><row>**

Mostrar o conteúdo da sub-tabela definida pelas coordenadas **<col><row>:<col><row>**.

Exemplo: show E1:K10

**filter <column> <data>** Eliminar as linhas da tabela com conteúdo na coluna **column** diferente de **data**.

Na construção deste programa deve-se utilizar a biblioteca produzida no exercício anterior.

Ao terminar a execução, o programa deve explicitamente libertar a memória alocada dinamicamente.

4. Adapte o programa anterior à aceitação de *plug-in* de modo que possam ser acrescentados novos comandos. Por exemplo, incorporar um comando que permita carregar um *shared object* com o código do novo comando.

**command <libfile>** Carregar um novo comando programado no *shared object* **libfile**.

Exemplifique a utilização dessa funcionalidade adicionando um novo comando que modifique a tabela. Por exemplo, a eliminação de uma linha dado o seu número.

Data limite de entrega: 22 de dezembro de 2025

ISEL, 17 de novembro de 2025