

---

# Roteiro diagnóstico

---

Este roteiro diagnóstico contém a descrição de uma aplicação que você deverá implementar ao longo do horário disponibilizado para a avaliação. Esta aplicação deverá ser entregue como um projeto compilável. O objetivo principal deste roteiro é delimitar sua capacidade de implementar um programa simples. No entanto, caso você tenha estudado os conceitos complementares abaixo, tente aplicá-los no contexto da sua aplicação:

- Além de fornecer sua aplicação como um **conjunto de arquivos bem organizado**, forneça um arquivo de **compilação automatizada**.
- Modele o seu programa usando os conceitos da programação **orientada a objetos**.
- Use os **tipos abstratos de dados** mais adequados para o contexto da sua aplicação.

## 1 Aplicação – cut

Ambientes UNIX fornecem a ferramenta `cut`, um programa simples, porém bastante útil, que permite filtrar fluxos ou arquivos, especificando delimitadores e selecionando apenas os campos desejados. O exemplo abaixo ilustra o uso desta ferramenta, onde a opção `-d` designa o delimitador (entre aspas simples) a ser usado para identificação dos campos e a opção `-f` designa quais campos deverão ser selecionados.

### Entrada

```
1,14.23,2.43,15.6,2.8,3.06,0.28,2.29
1,13.2,2.14,11.2,2.65,2.76,0.26,1.28
1,13.16,2.67,18.6,2.8,3.24,0.3,2.81
1,14.37,2.5,16.8,3.85,3.49,0.24,2.18
1,13.24,2.87,21,2.8,2.69,0.39,1.82
```

### Comando

```
cut -d ',' -f -2,4-5,7- < entrada.in
```

### Saída

```
1,14.23,15.6,2.8,0.28,2.29
1,13.2,11.2,2.65,0.26,1.28
1,13.16,18.6,2.8,0.3,2.81
1,14.37,16.8,3.85,0.24,2.18
1,13.24,21,2.8,0.39,1.82
```

Neste roteiro, você deverá reimplementar a ferramenta `cut`. Além disso, sua aplicação deverá fornecer as opções de linha de comando `--std` e `--file arquivo`:

`--std` especifica que a entrada e a saída padrões serão utilizadas.

`--file arquivo` indica a leitura de um arquivo de entrada `arquivo.in` e a escrita em um arquivo de saída `arquivo.out`.