



**INSTITUTO DO EMPREGO E FORMAÇÃO PROFISSIONAL**  
**DELEGAÇÃO REGIONAL DE LISBOA E VALE DO TEJO**  
CENTRO DE EMPREGO E FORMAÇÃO PROFISSIONAL DE LISBOA

---

## **Técnico especialista em Gestão de Redes e Sistemas Informáticos**

Programação em Python - TEGRSI08

**UFCD(s):** 10793-10793 | 5117-5091 | 5089

**Formador:** João Galamba

Efeitos.py  
Treep.py

**Formandos:**  
Carlos Venma  
Sérgio Cerqueira

---



## Introdução e Objetivos

O presente projeto visa consolidar os conhecimentos adquiridos em Python através da implementação de dois programas distintos.

O primeiro consiste na criação de algoritmos para aplicar efeitos visuais a texto, permitindo a prática de manipulação de strings e estruturas básicas da linguagem.

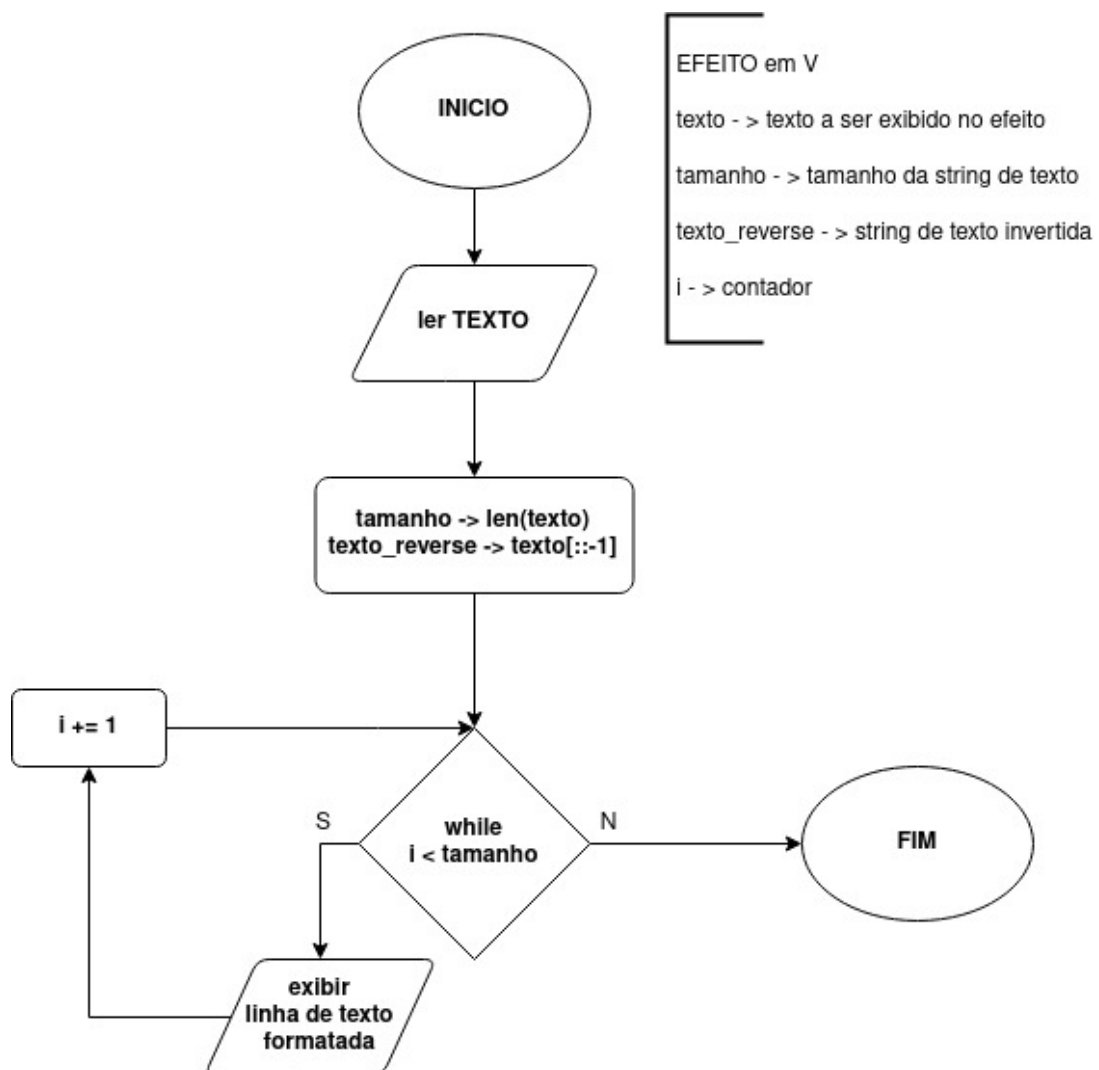
O segundo programa é um utilitário de linha de comandos, denominado *TREEP*, que simula o comportamento do comando *tree* do Linux. Este permite listar o conteúdo de uma diretoria em formato hierárquico. O programa deverá aceitar opções como -d, -f, -L e -H recorrendo à biblioteca *argparse* para o processamento dos argumentos.



# Desenho e Estrutura

## Fluxograma do efeito em V:

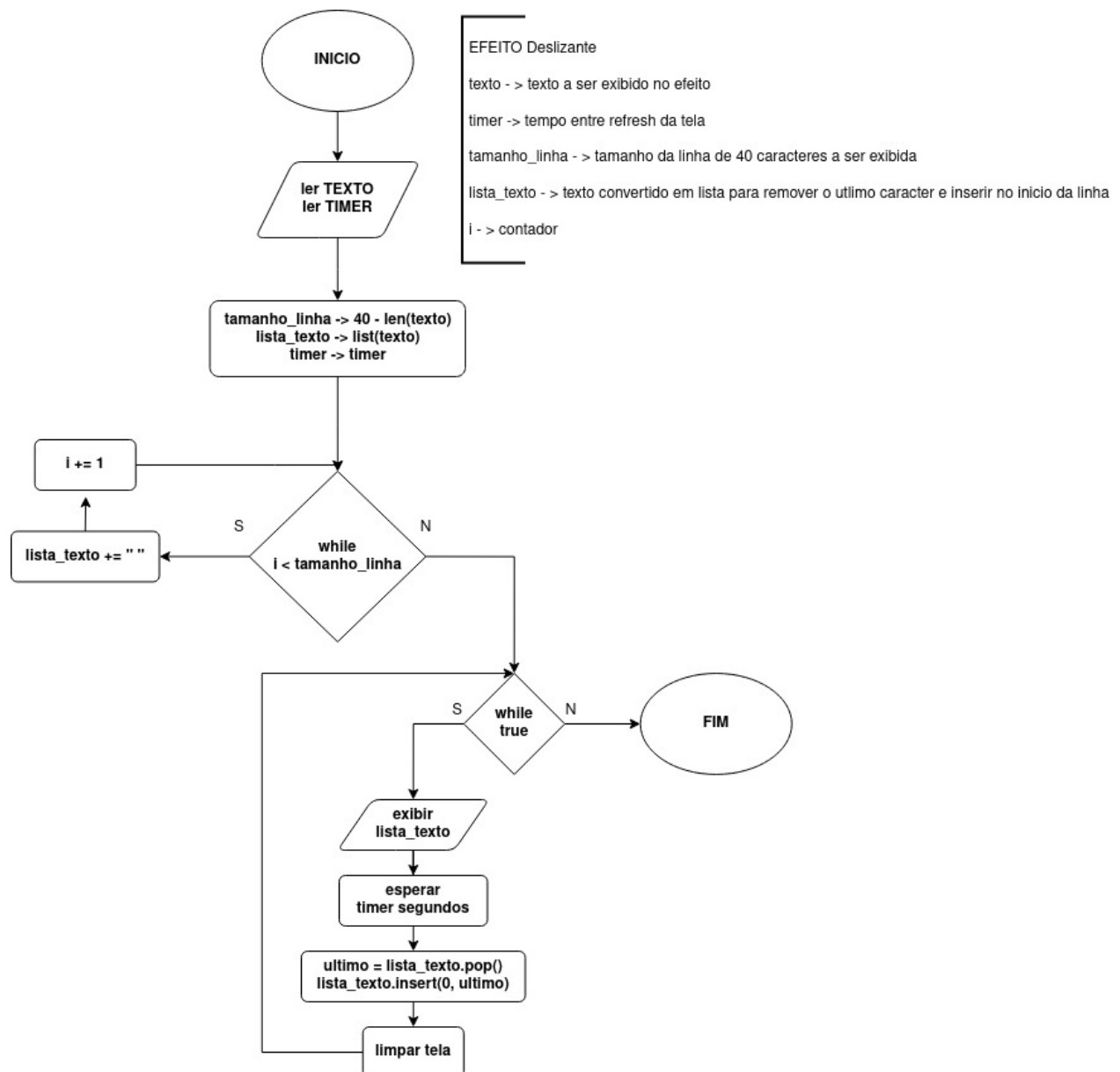
Este fluxograma representa o algoritmo do "efeito em V", que exibe uma string duas vezes por linha — uma em ordem normal e outra invertida — formando visualmente um "V".





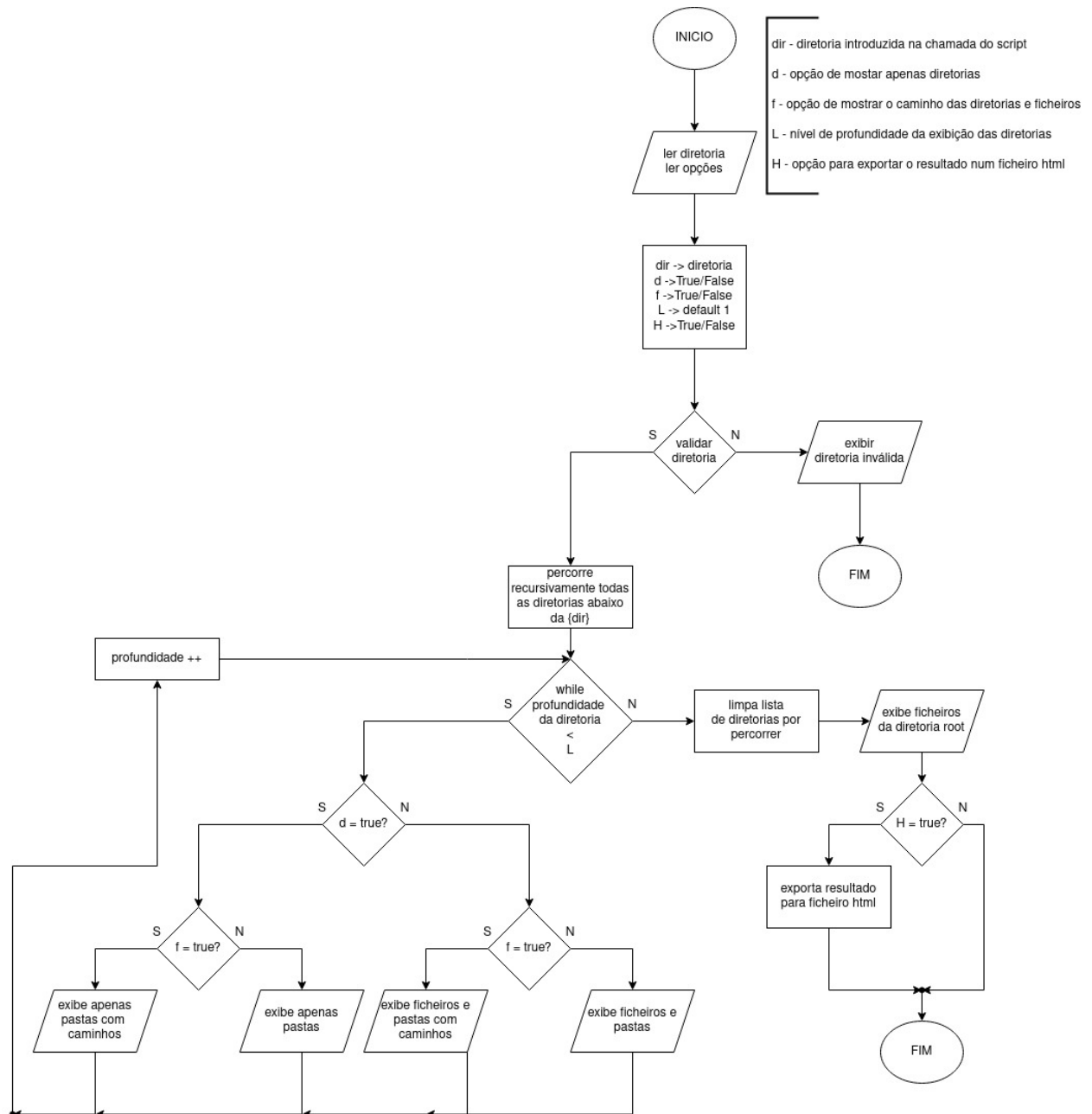
## Fluxograma do efeito Deslizante:

Este fluxograma representa um algoritmo de texto animado, onde a *string* fornecida se move da esquerda para a direita num intervalo de tempo especificado pelo utilizador, dando um efeito de "deslizamento" infinito.





## Fluxograma do script Treep:





---

**Técnico especialista em Gestão de Redes e Sistemas Informáticos**

Programação em Python - TEGRSI 08

**Início:** O script começa por ler os argumentos passados na linha de comandos (diretoria, opções -d, -f, -L, -H).

- **Validação:** Verifica se a diretoria fornecida é válida. Caso não seja, termina com uma mensagem de erro.
- **Percurso Recursivo:** Utiliza uma função de varrimento recursivo (`os.walk`) para explorar todas as subdiretórias e ficheiros a partir da raiz.
- **Controlo de Profundidade:** Para cada diretoria visitada, calcula-se a sua profundidade relativa à raiz. Se a profundidade ultrapassar o limite definido por -L, o percurso nessa ramificação é interrompido.
- **Formatação da Saída:**
- Dependendo das opções -d e -f, o script pode:
  - Exibir apenas pastas
  - Exibir pastas e ficheiros
  - Incluir ou não os caminhos completos
- **Exibição da raiz:** Se existirem ficheiros diretamente na raiz da diretoria, estes são exibidos separadamente.
- **Exportação HTML:** Se a opção -H for ativada, o resultado final é exportado para um ficheiro HTML, mantendo a estrutura hierárquica da árvore.



# Implementação

## Efeitos.py

A implementação do programa encontra-se no *script* `efeitos.py`, desenvolvido em *python* e executado em terminal. O objetivo do programa é aplicar diferentes efeitos visuais com texto, como diagonais, cruzamentos e deslizamentos animados, diretamente na linha de comandos.

A estrutura principal baseia-se na função `main()`, que apresenta um menu interativo para o utilizador selecionar qual dos efeitos deseja visualizar. Dependendo da escolha, uma função específica é chamada, como `efeito_1()`, `efeito_2()` até `efeito_6()`.

### Plataforma e Execução

O script foi desenvolvido e testado em ambiente *Unix (Linux)*, utilizando o terminal. É necessário dar permissão de execução com `chmod u+x efeitos.py`. A execução do script é feita através do terminal, por exemplo:

```
chmod u+x efeitos.py  
/efeitos.py -i 0.2 Hello World
```

### Compilação e Instalação

Não é necessária compilação. Basta ter o interpretador Python 3 instalado no sistema. O script é auto executável com as permissões corretas (*chmod*). Não requer bibliotecas externas.

### Elementos Opcionais

O argumento `-i` permite ao utilizador definir o intervalo de tempo (em segundos) entre cada movimento do efeito deslizante (`efeito_6`). O valor padrão é 0.5 segundos, mas pode ser personalizado conforme desejado.



## Treep.py

O script Treep foi desenvolvido em Python com o objetivo de simular o comando *tree* presente em sistemas Unix. A aplicação permite listar de forma hierárquica o conteúdo de uma diretoria, incluindo sub-diretorias e ficheiros, com diversas opções.

### Bibliotecas utilizadas

- **os**: usada para navegar no sistema de ficheiros, obter listas de ficheiros/diretorias e construir caminhos (`os.walk`, `os.path.join`, `os.access`, etc.).
- **sys**: permite a saída do programa em caso de erro e o acesso a argumentos do sistema.
- **argparse**: utilizada para tratar os argumentos passados pela linha de comandos de forma estruturada.
- **io** e **contextlib.redirect\_stdout**: usadas para manipular um ficheiro em memória e redirecionar a saída do programa para uma string (no caso da exportação HTML).
- **re**: usada para remover códigos ANSI do texto exportado, garantindo compatibilidade com HTML.

### Funcionalidades

O programa aceita os seguintes argumentos:

- **-d**: mostra apenas diretorias;
- **-f**: mostra os caminhos completos de ficheiros e diretorias;
- **-L N**: limita a profundidade da árvore a N níveis (por omissão: 1);
- **-H**: exporta o resultado para um ficheiro HTML.

As cores no terminal são geridas com códigos ANSI: diretorias são apresentadas a verde e em negrito, caminhos completos a ciano e entradas sem permissão são destacadas a vermelho.

### Exemplo de utilização

```
/treep.py ~/Documentos/Projetos -f -L 2
```

Este comando mostra a estrutura da pasta Projetos, incluindo caminhos completos até uma profundidade de 2 níveis.

```
./treep.py ~/Desktop -H
```

Neste caso, a estrutura da diretoria Desktop é apresentada no terminal e também exportada automaticamente para um ficheiro HTML chamado `Desktop_export.html`.





# Criação de .exe

Transformar um script Python num ficheiro executável (.exe) permite distribuir a aplicação de forma mais prática e profissional. Um executável pode ser executado diretamente, sem necessidade de instalar o Python nem configurar dependências, tornando o programa acessível a qualquer utilizador, mesmo sem conhecimentos técnicos.

Para obter um executável a partir de um script python, podemos correr os seguintes comandos:

1. Instale o PyInstaller

```
pip install pyinstaller
```

2. Navegue até o diretório do seu script

```
cd ./caminho/ficheiro.py
```

3. Com o powershell corra o pyinstaller

```
pyinstaller --onefile ficheiro.py
```

4. Teste o executável.



## Conclusão

Este projeto permitiu consolidar conhecimentos de programação em Python através da implementação de dois programas com naturezas distintas, mas complementares em termos de competências desenvolvidas.

O script `Efeitos.py` centrou-se na manipulação de texto, aplicando efeitos visuais através de algoritmos simples. Esta parte reforçou o domínio de estruturas básicas da linguagem, como listas, ciclos, funções e formatação de texto para a apresentação do output.

O segundo programa, `Treep.py`, teve um carácter mais técnico e próximo de uma aplicação real de linha de comandos. A sua implementação envolveu a interação com o sistema de ficheiros, o processamento de argumentos e a exportação de resultados para HTML. Para isso, foi necessário o uso de bibliotecas da standard *library* do Python, o que proporcionou um contacto mais profundo com ferramentas comuns em programação orientada ao sistema.