

# **Gestão e armazenamento: Monitorização do espaço ocupado**

## **SISTEMAS OPERATIVOS RELATÓRIO**

Prof. José Nuno Panelas Nunes Lau

Departamento de Eletrónica, Telecomunicações e Informática

João Manuel Vieira Roldão	113920
Rui de Faria Machado	113765

Turma P6 - 2023/2024

# Índice

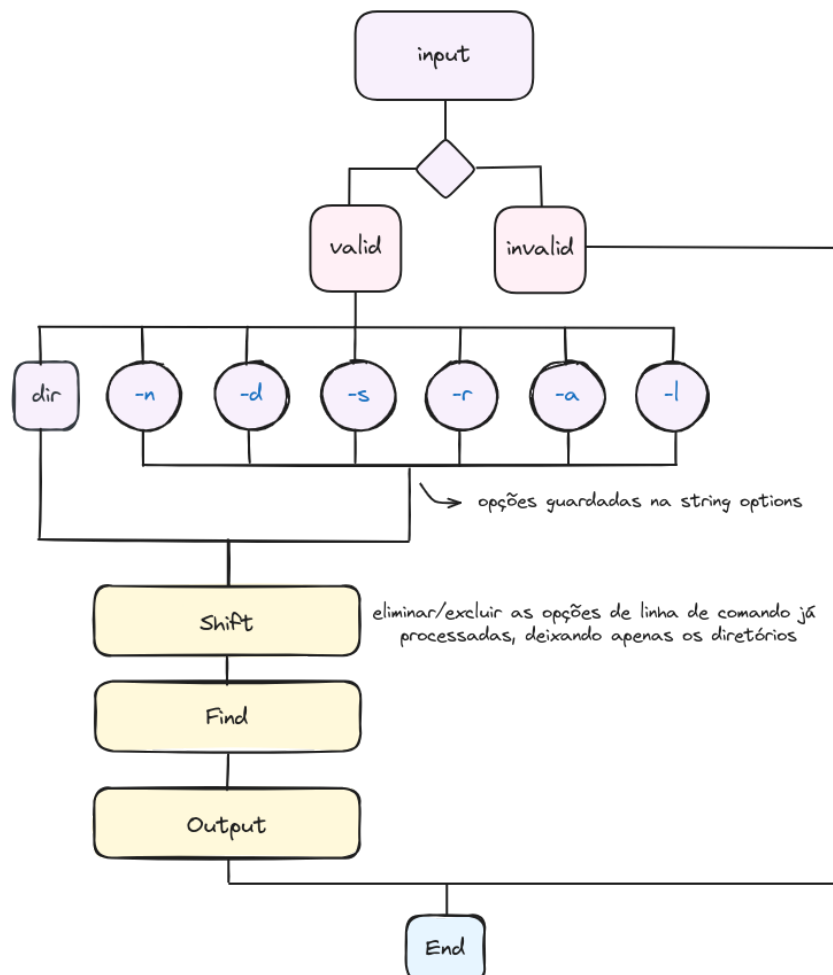
<b>Introdução</b>	3
<b>Objetivos</b>	4
Desenvolvimento de scripts em bash	5
Tratamento de nomes de arquivos com espaços	5
Espaço ocupado por arquivos com critérios específicos	5
Visualização flexível dos dados	5
Comparação dos resultados	5
Opções de ordenação na saída	5
Validação e Testes Abrangentes	5
<b>Validação e Testes</b>	6
Testes funcionais do spacecheck.sh	6
Testes de tratamento de espaços	6
Testes do spacerate.sh	6
Testes de integração com sistemas de arquivos reais	7
<b>Problemas e Soluções</b>	8
Filtragem por nome de arquivos com espaços	8
Manipulação de datas e horários	8
Tratamento de nomes de diretórios e arquivos com caracteres especiais	8
Comparação de saídas com grandes volumes de dados	9
Tratamento de erros e exceções	9
<b>Conclusão</b>	10
<b>Bibliografia</b>	11

## Introdução

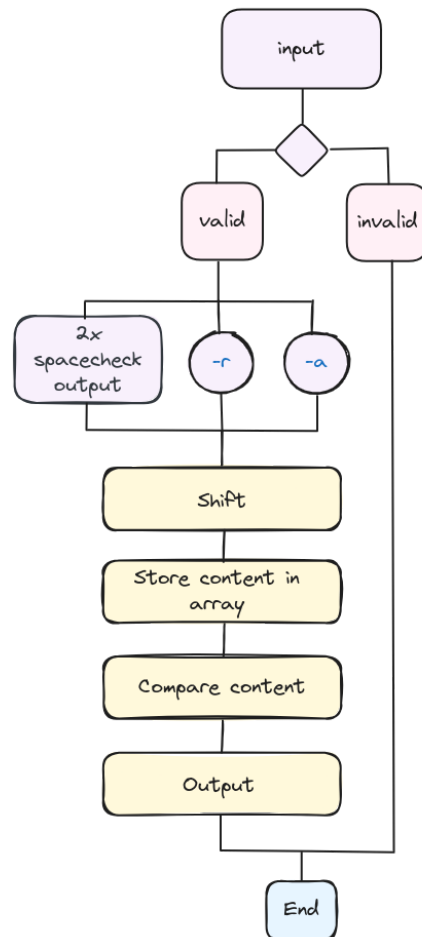
A eficiente gestão de armazenamento sempre foi um desafio em ambientes de computação, onde o volume de dados cresce exponencialmente. Nesse contexto, a criação de ferramentas capazes de monitorar e otimizar o espaço ocupado em disco torna-se fundamental para garantir o funcionamento adequado de sistemas e de recursos de armazenamento.

Este relatório apresenta o resultado de um projeto da UC de Sistemas Operativos que visa abordar essa necessidade por meio do desenvolvimento de dois scripts: `spacecheck.sh` e `spacerate.sh`.

O `spacecheck` deve ser capaz de analisar um diretório e aceder a todas as pastas e ficheiros desse mesmo diretório para depois expor a informação do tamanho ocupado por esses mesmos, tudo de acordo com critérios definidos pelo utilizador.



O spacerate.sh, por outro lado, analisa a informação que é dada pelo primeiro script e diz quais são as diferenças entre dois outputs, tirados em momentos diferentes, do spacecheck.sh.



Ao longo deste relatório, apresentaremos detalhes sobre a implementação de ambos os scripts, descrevendo a lógica por trás de cada funcionalidade e as opções disponíveis para personalização.

# Objetivos

Estes objetivos abrangem as principais metas do trabalho, fornecendo uma visão mais concisa e organizada das intenções do projeto.

## 1. Desenvolvimento de scripts em bash

- Desenvolver dois scripts em bash, `spacecheck.sh` e `spacerate.sh`, para monitorar e comparar o espaço ocupado em disco por arquivos e diretórios.

## 2. Tratamento de nomes de arquivos com espaços

- Garantir que o `spacecheck.sh` lida de forma adequada com arquivos e ficheiros que contenham espaços no seu nome, assegurando o seu correto processamento.

## 3. Espaço ocupado por arquivos com critérios específicos

- Implementar o `spacecheck.sh` para calcular o espaço ocupado por arquivos que atendam a critérios pré definidos pelo utilizador, como nome, data de modificação e tamanho, em diretórios e subdiretórios.

## 4. Visualização flexível dos dados

- Possibilitar a visualização dos resultados do `spacecheck.sh` de diferentes formas, incluindo ordenação por ordem alfabética, ordem reversa e limitação do número de linhas do output. Sendo que a ordem pré-definida é a ordem decrescente do tamanho dos subdiretórios.

## 5. Comparação dos resultados

- Implementar o `spacerate.sh` para permitir a comparação dos resultados de dois momentos diferentes obtidos com o `spacecheck.sh`, destacando alterações, acréscimos e remoções de arquivos e diretórios.

## 6. Opções de ordenação na saída

- Oferecer opções de ordenação para a saída do `spacerate.sh`, permitindo que o usuário escolha como deseja visualizar as diferenças nos dados de armazenamento.

## 7. Validação e Testes Abrangentes

- Validar a funcionalidade dos scripts por meio de testes abrangentes, garantindo que eles forneçam resultados precisos e úteis para o utilizador.

## Validação e Testes

A validação e os testes foram essenciais para garantir que os scripts `spacecheck.sh` e `spacerate.sh` funcionassem conforme o esperado, fornecendo resultados precisos e úteis. Estes asseguram que as ferramentas são confiáveis e capazes de lidar com uma grande variedade de cenários.

- **Testes funcionais do `spacecheck.sh`:**

- Cada opção disponível no script foi testada de forma isolada e em combinação para garantir que a filtragem de arquivos e a geração da saída ocorresse conforme o esperado. Os prints que seguem demonstram apenas alguns dos testes realizados.

```
laptop-johnny@laptop-johnny-OMEN-Laptop-15-ek0xxx:~/Secretária/Uni/2ºAno/so-projeto01$ ./spacecheck.sh -d "Oct 10 10:00" -n ".sh" -l "3" /home/laptop-johnny/Secretária/Uni/2ºAno/SO
SIZE NAME 20231113 -d "Oct 10 10:00" -n ".sh" -l "3" /home/laptop-johnny/Secretária/Uni/2ºAno/SO
6200 /home/laptop-johnny/Secretária/Uni/2ºAno/SO
3647 /home/laptop-johnny/Secretária/Uni/2ºAno/SO/P3
2553 /home/laptop-johnny/Secretária/Uni/2ºAno/SO/P4
```

```
laptop-johnny@laptop-johnny-OMEN-Laptop-15-ek0xxx:~/Secretária/Uni/2ºAno/so-projeto01$ ./spacecheck.sh -s "2500" -r -a /home/laptop-johnny/Secretária/Uni/2ºAno/SO
SIZE NAME 20231113 -s "2500" -r -a /home/laptop-johnny/Secretária/Uni/2ºAno/SO
215706 /home/laptop-johnny/Secretária/Uni/2ºAno/SO/P6
80576 /home/laptop-johnny/Secretária/Uni/2ºAno/SO/P5
0 /home/laptop-johnny/Secretária/Uni/2ºAno/SO/P4
0 /home/laptop-johnny/Secretária/Uni/2ºAno/SO/P3
0 /home/laptop-johnny/Secretária/Uni/2ºAno/SO/P2
2567470 /home/laptop-johnny/Secretária/Uni/2ºAno/SO/Guiões
2863752 /home/laptop-johnny/Secretária/Uni/2ºAno/SO
```

- Para além disso, foram realizados testes automatizados fornecidos pelo professor da disciplina. Estes foram passados com sucesso.

```
● laptop-johnny@laptop-johnny-OMEN-Laptop-15-ek0xxx:~/Secretária/Uni/2ºAno/so-projeto01
$ ./test_a1.sh
OK
```

- **Testes de tratamento de espaços no nome de arquivos:**

- Foram criados arquivos com espaços no nome para que fosse possível a sua execução no script de modo a garantir que esses elementos fossem processados adequadamente.

- **Testes do spacerate.sh:**

- O script spacerate.sh foi validado por meio de testes que compararam as saídas de dois momentos diferentes do spacecheck.sh. Isso envolveu a criação de arquivos de saída simulados do spacecheck.sh em momentos distintos e a execução do spacerate.sh para verificar se as diferenças eram destacadas de maneira precisa. Os testes também incluíram a ordenação da saída do spacerate.sh usando opções como -r e -a.

```
laptop-johnny@laptop-johnny-OMEN-Laptop-15-ek0xxx:~/Secretária/Uni/2ºAno/so-projeto01$ ./spacerate.sh -r -a spacecheck_20231101.txt spacecheck_20231102.txt
SIZE NAME
217312 /home/laptop-johnny/Secretária/Uni/2ºAno/SO/P6 NEW
0 /home/laptop-johnny/Secretária/Uni/2ºAno/SO/P5
0 /home/laptop-johnny/Secretária/Uni/2ºAno/SO/P4
0 /home/laptop-johnny/Secretária/Uni/2ºAno/SO/P3
0 /home/laptop-johnny/Secretária/Uni/2ºAno/SO/P2 REMOVED
1234356 /home/laptop-johnny/Secretária/Uni/2ºAno/SO/Guiões
1252854 /home/laptop-johnny/Secretária/Uni/2ºAno/SO
```

- **Testes de integração com sistemas de arquivos reais:**

- Para validar a eficácia dos scripts em cenários do mundo real, foram realizados testes em sistemas de arquivos reais, incluindo diretórios e arquivos existentes. Isso ajudou a verificar a capacidade dos scripts de lidar com volumes de dados significativos e com situações complexas de armazenamento.

# Problemas e Soluções

## 1. Manipulação de datas e horários:

Problema: A conversão de datas e horários para um formato consistente, a fim de comparar a data de modificação dos arquivos com a data especificada nas opções, foi uma tarefa complexa.

Solução: Foi necessário criar a função `parse_date()` que fosse capaz de manipular a data inserida no formato “MMM DD HH:MM” (por exemplo, SEP 09 15:30). Esta formata a data de modo a incluir o ano atual e, em seguida, converte essa data formatada para segundos desde época (UNIX timestamp).

```
parse_date() {          #função para manipular a data inserida
    input_date="$1"
    # extrair o mês, o dia e a hora
    month=$(echo "$input_date" | awk '{print $1}')
    day=$(echo "$input_date" | awk '{print $2}')
    time=$(echo "$input_date" | awk '{print $3}')
    # mudar a data para o formato "MMM DD HH:MM YYYY"
    formatted_date="$month $day $time $(date +%Y)"
    # usar a data para converter a formatada em segundos
    date -d "$formatted_date" +%s
}
```

## 2. Tratamento de nomes de diretórios e arquivos com caracteres especiais:

Problema: Além de espaços, nomes de diretórios e arquivos podem conter outros caracteres especiais, o que pode dificultar o processamento correto.

Solução: A solução envolveu o uso de aspas e caracteres de escape, como aspas duplas (") e barras invertidas (\), para garantir que nomes de diretórios e arquivos com caracteres especiais fossem interpretados corretamente.

```
for dir in $@; do
    # pesquisa do diretório, atendendo critérios
    find "$dir" -type d | \
        while read -r folder; do
            size=0
            while IFS= read -r -d $'\0' file; do
```



### 3. Comparação de saídas com grandes volumes de dados:

Problema: A comparação de saídas do spacecheck.sh com grandes volumes de dados pode ser demorada e exigir muita memória.

Solução: Para abordar esse problema, os scripts foram otimizados para minimizar a sobrecarga de recursos. Além disso, a possibilidade de dividir a tarefa em várias execuções menores foi considerada para lidar com grandes volumes de dados de forma mais eficaz.

### 4. Tratamento de erros e exceções:

Problema: A falta de permissões de leitura em diretórios ou arquivos inacessíveis pode gerar erros não tratados.

Solução: Foi implementada uma lógica para tratar de forma adequada essas situações, marcando o espaço ocupado como "NA" e garantindo que o script continue a funcionar mesmo em cenários de erro.

```
if [ -n "$file_size" ]; then
    if [ "$file_size" -ge "$min_size" ]; then
        # manipulação de acordo com a flag "-s" (size)
        size=$((size + file_size))
    fi
else
    size="NA" # não foi possível determinar o tamanho do arquivo
fi
```

## **Conclusão**

De modo geral, os scripts `spacecheck.sh` e `spacerate.sh` mostraram-se ferramentas eficazes e flexíveis para lidar com o gerenciamento do armazenamento e espaço ocupado em disco em computadores. Eles facilitam o monitoramento e a análise do espaço em disco, lidando bem com nomes de arquivos e diretórios complexos. Os testes comprovaram sua confiabilidade e eficiência, tornando-os valiosos para a gestão de armazenamento em ambientes de computação atuais.

Para concluir, é importante destacar que o trabalho proposto foi concluído com êxito, atendendo a todos os parâmetros e requisitos estabelecidos inicialmente. A execução demonstrou conformidade com as diretrizes solicitadas, garantindo um bom resultado final.

## **Bibliografia**

1. Stack Overflow - Comunidade de programadores  
Acesso em outubro de 2023  
<https://stackoverflow.com/>
2. Página oficial do GNU Bash  
Acesso em outubro de 2023  
<https://www.gnu.org/software/bash/>
3. Unix Time Stamp  
Acesso em novembro de 2023  
<https://www.unixtimestamp.com/>