

Sistemas Operativos  
2024/2025

Projeto1:

*Ferramenta de criação/atualização de cópias de  
segurança em Bash*

*Autores: João Pereira 120010 , Thiago Vicente 121497*

# Introdução

## Do que se trata?

Este projeto foi realizado no âmbito da disciplina de Sistemas Operativos cujo objetivo foi a criação de 3 scripts para criação e atualização de cópias de segurança.

- O backup\_files.sh tem o objetivo de criar e/ou atualizar uma cópia de segurança em que se assume que não existem diretórios.
  - O backup.sh tem o mesmo objetivo do backup\_files.sh, mas neste já se assume a possibilidade da existência de diretórios.
  - O backup\_summary.sh é similar ao backup.sh, mas este tem informação sobre quantos ficheiros foram copiados/eliminados e sobre o espaço que esses mesmos ficheiros ocupavam.
  - O backup\_check.sh serve para avaliar se os ficheiros na diretoria que foi copiada são iguais aos da diretoria que contém a sua cópia de segurança.
- Todos os scripts escrevem no terminal as operações de cópias ou de eliminação de ficheiros/diretorias que fizeram.

## Metodologia de Desenvolvimento

Para o desenvolvimento deste projeto, adotamos a ferramenta **Git** como controle de versão, permitindo que ambos os desenvolvedores trabalhassem de maneira independente em partes distintas do projeto. Essa abordagem facilitou a colaboração e o gerenciamento de alterações, especialmente ao modularizar as funcionalidades em arquivos separados.

Inicialmente, definimos as responsabilidades de cada função, especificando claramente os parâmetros de entrada, o comportamento esperado e o valor de retorno de cada uma. Com essa estruturação, decidimos a divisão das tarefas entre os membros da equipe, garantindo uma organização eficiente do trabalho.

As responsabilidades ficaram distribuídas da seguinte forma:

- **João Pereira (120010)** foi responsável por implementar a função `copy_file()`, pela lógica por trás da função `backup_check()` e pela criação dos testes do sistema.
- **Thiago Vicente (121497)** ficou encarregado de desenvolver as funções `compModeDate()`, `usage()`, `nfound()`, `find_element()`, de desenvolver e os arquivos de backup (`backup.sh`, `backup_files.sh`, `backup_summary.sh`) e de fazer os últimos retoques do `backup_check.sh`.

O trabalho foi realizado sem intercorrências, uma vez que cada desenvolvedor seguiu as orientações estabelecidas e contribuiu dentro de sua área de responsabilidade. Abaixo, apresentamos uma descrição detalhada das funcionalidades implementadas em cada função e arquivo.

## Testes

Para testar os scripts, iremos testar uma cópia em que a pasta destino está vazia, uma cópia em que a pasta destino tem alguns dos arquivos da pasta source, uma cópia em que a pasta destino tem arquivos que não pertencem à pasta source e vamos testar uma cópia em que a pasta destino tem os mesmos arquivos da pasta source mas alguns foram modificados. Também iremos testar pastas com ficheiros escondidos e com espaços nos nomes. Nos scripts que tiverem parâmetros opcionais iremos testá-los com e sem esses parâmetros.

Os teste apresentados neste relatório não representam a totalidade dos testes feitos, mas sim apenas os mais pertinentes de serem discutidos.

# Estruturas de dados

Nesta secção, vamos analisar as principais estruturas de dados utilizadas no script, que são fundamentais para o armazenamento e manipulação de informações durante o processo de backup. Veremos como arrays, variáveis inteiras e strings são usados para controlar a execução e otimizar a organização dos dados.

## Arrays

Os **arrays** são usados para armazenar listas de itens, como os arquivos a excluir ou os argumentos passados para o script.

### Uso:

```
exclude_list=() # Array para armazenar arquivos a serem excluídos
args=($@)       # Array que contém os parâmetros passados para o script
lst=(${args[@]}:${#args[@]}-1) # Array contendo todos os elementos de
args, exceto o último
```

## Strings

No Bash são tratadas como arrays de caracteres e são usadas para armazenar sequências de texto. Foram usadas para armazenar dados como *paths* e *regex*.

### Uso:

```
source_dir="" # Caminho do diretório de origem

backup_dir="" # Caminho do diretório de backup

regex=".*" # Expressão regular para filtrar arquivos

exclude_file="" # Arquivo contendo os arquivos a excluir
```

# Como dividimos o problema (funções)

O problema foi dividido de forma a torná-lo mais modular, visando a reutilização do código e facilitando a manutenção e a compreensão. Cada função desempenha um papel fundamental na execução do processo de backup. A seguir, iremos analisar as funções utilizadas e sua contribuição para a solução.

## **findElement()**

Esta função recebe 2 argumentos, um array e um valor a se procurar.

### **Entrada:**

- array: A lista de elementos para pesquisar
- elemento: O elemento que estamos procurando

.

### **Saída:**

- Retorna 0 se o elemento for encontrado
- Retorna 1 se o elemento não for encontrado

.

### **Execução**

```
args ← array
lst ← args[0 até comprimento(args)-1]
toFind ← args[último]
para cada item em lst faça:
    se item é igual a toFind então
        retorne 0
    fim se
fim para
retorne 1
```

## **compModDate()**

Esta função recebe 2 caminhos para ficheiros e compara a última data de modificação. usando o comando **"-nt"**.

### **Entrada:**

- file1: Caminho para o arquivo fonte
- file2: Caminho para o arquivo de backup

- .

### **Saída:**

- Retorna 0 se *file1* foi modificado depois de *file2*
- Retorna 1 se *file2* foi modificado depois de *file1*,  
ou se as datas de modificação são iguais

#### Execução

```
se file1 foi modificado depois de file2 então
    retorne 0
senão se file2 foi modificado depois de file1 então
    exiba mensagem de advertência
    retorne 1
fim se
retorne 1
```

## copyFile()

Esta função recebe um ficheiro ,um diretório para onde realizar a cópia e um valor **copy** que indica se a cópia deve ser realizada.

#### Entrada:

- file: Caminho do arquivo a ser copiado
- destination: Caminho do diretório de destino
- copy: Se 1, copia o arquivo; se 0, apenas verifica as datas

#### Saída:

- Retorna 0 se o arquivo for copiado com sucesso
- Retorna 1 se ocorrer um erro durante a cópia

#### Execução

```
se o arquivo já existe no destino então
    comparar a data de modificação entre o arquivo e o de
backup
    se o arquivo fonte for mais recente então
        se copy for 1 então
            copie o arquivo para o destino
            se ocorrer erro ao copiar então
                retorne 1
            fim se
            atualize os contadores de cópias
            retorne 0
        fim se
    fim se
fim se
fim se
.
se o arquivo não existe no destino e copy for 1 então
```

```
        copie o arquivo para o destino
        se ocorrer erro ao copiar então
            retorne 1
        fim se
        atualize os contadores de cópias
        retorne 0
    fim se
```

## usage()

Exibe a mensagem de uso do script.

### Entrada:

- Nenhuma

.

### Saída:

- Exibe a mensagem de uso do script

.

### Execução:

Exiba "[USAGE] ./backup.sh [-c] [-b excludefile] [-r regx] ...  
dir\_source dir\_backup"

## nfound()

Exibe mensagem de erro se o diretório ou arquivo não for encontrado.

### Entrada:

- field: O tipo de entidade (arquivo ou diretório) que não foi encontrado

- path: Caminho do arquivo ou diretório que não foi encontrado

.

### Saída:

- Exibe mensagem de erro e termina a execução do script

.

### Execução:

Exiba "[NOTFOUND]: campo "\$field" > "\$path"  
Terminar execução

## **end\_print()**

Exibe estatísticas do backup. É só usado no backup\_summary.sh.

Entrada:

- is\_recursive: Indica se a execução é recursiva ou não

.

Saída:

- Exibe o resumo das operações realizadas (Erros, Warnings, Arquivos Copiados, etc.)

- Finaliza a execução do script

.

Execução:

- Se is\_recursive for 1 então

- Exiba resumo das estatísticas de backup (Erros, Warnings, Arquivos Copiados, etc.)

- Senão

- Exiba resumo e encerre o script

- Fim se



# Ficheiros Principais

Todos os scripts têm 2 argumentos obrigatórios, sendo o primeiro deles a diretoria que vai ser copiada e o segundo a diretoria para onde a cópia deverá ir.

## backup\_files.sh

### Etapas:

- Importar funções

```
function_call="$0" # Usa-se o function_call para determinar o
caminho de acesso às funções
SCRIPT_DIR="$(dirname "$(realpath "$function_call")")"
DIR="$SCRIPT_DIR/functionsReal"
for file in "$DIR"/*; do
    if [[ -f "$file" ]]; then
        source "$file"
    fi
done
```

- Declaração de variáveis

```
checking=""
source_dir=""
backup_dir=""
```

- Ler as opções do usuário e processar as diferentes possibilidades.

```
while getopts ":c:" op; do
    case $op in
        ...
    esac
done
```

- Validar os diretórios ( e lidar com o possível erro do **realpath** )

```

source_dir=$(realpath "$1")
if [ $? -ne 0 ]; then
    echo "Can't resolve source directory path"
    exit 1
fi
backup_dir=$(realpath "$2")
if [ $? -ne 0 ]; then
    ...
fi
if [[ "$backup_dir" == "$source_dir"* ]]; then
    echo "[ERROR] $backup_dir is inside $source_dir"
    exit 1
fi

```

- Criar o diretório backup caso necessário

```

Se backup_dir não existe então
    criar backup_dir
fim se

```

- Copiar cada ficheiro

```

Para cada elemento de source_dir
    se é um ficheiro
        chamar copy_file( elemento, backup_dir )
    fim se
fim para

```

- Remover elementos do backup\_dir que não fazem parte do source\_dir

```

Para cada elemento de backup_dir
    se elemento não existe no source_dir
        remover elemento
    fim se
fim para

```

## Testes

Teste de cópia básica(foram copiados ficheiros para uma pasta vazia)

```
jpp9547@jpp-Vivobook: ~/Escola/S0/projeto1/Projeto1S0$ ls ~/Escola/S0/testeProjeto/
jpp9547@jpp-Vivobook: ~/Escola/S0/projeto1/Projeto1S0$ ls
backup_check.sh  backup_files.sh  backup_summary.sh  functions  output.txt  RelatorioSo  testeBackupFiles.py
backup_checkv2.sh backup.sh         filewithotherfuncs.sh functionsReal README.md    save
jpp9547@jpp-Vivobook: ~/Escola/S0/projeto1/Projeto1S0$ ./backup_files.sh ./ ~/Escola/S0/testeProjeto/
cp -a ././backup_check.sh /home/jpp9547/Escola/S0/testeProjeto/backup_check.sh
cp -a ././backup_checkv2.sh /home/jpp9547/Escola/S0/testeProjeto/backup_checkv2.sh
cp -a ././backup_files.sh /home/jpp9547/Escola/S0/testeProjeto/backup_files.sh
cp -a ././backup.sh /home/jpp9547/Escola/S0/testeProjeto/backup.sh
cp -a ././backup_summary.sh /home/jpp9547/Escola/S0/testeProjeto/backup_summary.sh
cp -a ././filewithotherfuncs.sh /home/jpp9547/Escola/S0/testeProjeto/filewithotherfuncs.sh
cp -a ././output.txt /home/jpp9547/Escola/S0/testeProjeto/output.txt
cp -a ././README.md /home/jpp9547/Escola/S0/testeProjeto/README.md
cp -a ././save /home/jpp9547/Escola/S0/testeProjeto/save
cp -a ././testeBackupFiles.py /home/jpp9547/Escola/S0/testeProjeto/testeBackupFiles.py
jpp9547@jpp-Vivobook: ~/Escola/S0/projeto1/Projeto1S0$ ls ~/Escola/S0/testeProjeto/
backup_check.sh  backup_files.sh  backup_summary.sh  output.txt  save
backup_checkv2.sh backup.sh         filewithotherfuncs.sh README.md    testeBackupFiles.py
```

Teste pastas com espaços

```
jpp9547@jpp-Vivobook: ~/Escola/S0/projeto1/Projeto1S0$ ls ~/Escola/S0/"teste Projeto"
jpp9547@jpp-Vivobook: ~/Escola/S0/projeto1/Projeto1S0$ ls
backup_check.sh  backup_files.sh  backup_summary.sh  functions  output.txt  RelatorioSo  testeBackupFiles.py
backup_checkv2.sh backup.sh         filewithotherfuncs.sh functionsReal README.md    save
jpp9547@jpp-Vivobook: ~/Escola/S0/projeto1/Projeto1S0$ ./backup_files.sh ./ ~/Escola/S0/"teste Projeto"
cp -a ././backup_check.sh /home/jpp9547/Escola/S0/teste Projeto/backup_check.sh
cp -a ././backup_checkv2.sh /home/jpp9547/Escola/S0/teste Projeto/backup_checkv2.sh
cp -a ././backup_files.sh /home/jpp9547/Escola/S0/teste Projeto/backup_files.sh
cp -a ././backup.sh /home/jpp9547/Escola/S0/teste Projeto/backup.sh
cp -a ././backup_summary.sh /home/jpp9547/Escola/S0/teste Projeto/backup_summary.sh
cp -a ././filewithotherfuncs.sh /home/jpp9547/Escola/S0/teste Projeto/filewithotherfuncs.sh
cp -a ././output.txt /home/jpp9547/Escola/S0/teste Projeto/output.txt
cp -a ././README.md /home/jpp9547/Escola/S0/teste Projeto/README.md
cp -a ././save /home/jpp9547/Escola/S0/teste Projeto/save
cp -a ././testeBackupFiles.py /home/jpp9547/Escola/S0/teste Projeto/testeBackupFiles.py
jpp9547@jpp-Vivobook: ~/Escola/S0/projeto1/Projeto1S0$ ls ~/Escola/S0/"teste Projeto"
backup_check.sh  backup_files.sh  backup_summary.sh  output.txt  save
backup_checkv2.sh backup.sh         filewithotherfuncs.sh README.md    testeBackupFiles.py
```

Teste cópia com todos os ficheiros já colocados na pasta destino e com a mesma data de alteração

```
jpp9547@jpp-Vivobook: ~/Escola/S0/projeto1/Projeto1S0$ ls ~/Escola/S0/testeProjeto/
backup_check.sh  backup_files.sh  backup_summary.sh  output.txt  save
backup_checkv2.sh backup.sh         filewithotherfuncs.sh README.md    testeBackupFiles.py
jpp9547@jpp-Vivobook: ~/Escola/S0/projeto1/Projeto1S0$ ls
backup_check.sh  backup_files.sh  backup_summary.sh  functions  output.txt  RelatorioSo  testeBackupFiles.py
backup_checkv2.sh backup.sh         filewithotherfuncs.sh functionsReal README.md    save
jpp9547@jpp-Vivobook: ~/Escola/S0/projeto1/Projeto1S0$ ./backup_files.sh ./ ~/Escola/S0/testeProjeto/
```

Teste cópia em que existem ficheiros na pasta de destino que não existem na pasta que irá ser copiada

```
jpp9547@jpp-Vivobook: ~/Escola/S0/projeto1/Projeto1S0$ ls
backup_check.sh  backup_files.sh  backup_summary.sh  functions  RelatorioSo
backup_checkv2.sh backup.sh         filewithotherfuncs.sh functionsReal
jpp9547@jpp-Vivobook: ~/Escola/S0/projeto1/Projeto1S0$ ls ~/Escola/S0/testeProjeto/
backup_check.sh  backup_checkv2.sh  backup_files.sh  backup.sh  backup_summary.sh  filewithotherfuncs.sh  save
jpp9547@jpp-Vivobook: ~/Escola/S0/projeto1/Projeto1S0$ ./backup_files.sh ./ ~/Escola/S0/testeProjeto/
rm -r /home/jpp9547/Escola/S0/testeProjeto/save
jpp9547@jpp-Vivobook: ~/Escola/S0/projeto1/Projeto1S0$ ls ~/Escola/S0/testeProjeto/
backup_check.sh  backup_checkv2.sh  backup_files.sh  backup.sh  backup_summary.sh  filewithotherfuncs.sh
jpp9547@jpp-Vivobook: ~/Escola/S0/projeto1/Projeto1S0$
```

Teste cópia em que existem um ficheiro com data de alteração mais recente na pasta destino do que na pasta que vai ser copiada

```
jpp9547@jpp-Vivobook: ~/Escola/S0/projeto1/Projeto1S0$ ls
backup_check.sh  backup_files.sh  backup_summary.sh  functions  RelatorioSo
backup_checkv2.sh backup.sh         filewithotherfuncs.sh functionsReal teste.txt
jpp9547@jpp-Vivobook: ~/Escola/S0/projeto1/Projeto1S0$ ls ~/Escola/S0/testeProjeto/
backup_check.sh  backup_checkv2.sh  backup_files.sh  backup.sh  backup_summary.sh  filewithotherfuncs.sh  teste.txt
jpp9547@jpp-Vivobook: ~/Escola/S0/projeto1/Projeto1S0$ ./backup_files.sh ./ ~/Escola/S0/testeProjeto/
[WARNING] Backed file (/home/jpp9547/Escola/S0/testeProjeto/teste.txt) is newer than source file (././teste.txt) (SHOULD NOT HAPPEN)
jpp9547@jpp-Vivobook: ~/Escola/S0/projeto1/Projeto1S0$
```

## Teste do parametro -c

```
jpp9547@jpp-Vivobook:~/Escola/S0/projeto1/Projeto1S0$ ls ~/Escola/S0/testeProjeto/
jpp9547@jpp-Vivobook:~/Escola/S0/projeto1/Projeto1S0$ ls
backup_check.sh  backup_files.sh  backup_summary.sh  functions  RelatorioSo
backup_checkv2.sh  backup.sh  filewithotherfuncs.sh  functionsReal  teste.txt
jpp9547@jpp-Vivobook:~/Escola/S0/projeto1/Projeto1S0$ ./backup_files.sh -c ./ ~/Escola/S0/testeProjeto/
checking activated
cp -a ././backup_check.sh /home/jpp9547/Escola/S0/testeProjeto/backup_check.sh
cp -a ././backup_checkv2.sh /home/jpp9547/Escola/S0/testeProjeto/backup_checkv2.sh
cp -a ././backup_files.sh /home/jpp9547/Escola/S0/testeProjeto/backup_files.sh
cp -a ././backup.sh /home/jpp9547/Escola/S0/testeProjeto/backup.sh
cp -a ././backup_summary.sh /home/jpp9547/Escola/S0/testeProjeto/backup_summary.sh
cp -a ././filewithotherfuncs.sh /home/jpp9547/Escola/S0/testeProjeto/filewithotherfuncs.sh
cp -a ././teste.txt /home/jpp9547/Escola/S0/testeProjeto/teste.txt
jpp9547@jpp-Vivobook:~/Escola/S0/projeto1/Projeto1S0$ ls ~/Escola/S0/testeProjeto/
jpp9547@jpp-Vivobook:~/Escola/S0/projeto1/Projeto1S0$
```

Teste em que ambas as pastas têm os mesmos ficheiros, mas a pasta que vai ser copiada tem um arquivo que foi alterado

```
jpp9547@jpp-Vivobook:~/Escola/S0/projeto1/Projeto1S0$ ./backup_files.sh ./ ~/Escola/S0/testeProjeto/
cp -a ././backup_check.sh /home/jpp9547/Escola/S0/testeProjeto/backup_check.sh
cp -a ././backup_checkv2.sh /home/jpp9547/Escola/S0/testeProjeto/backup_checkv2.sh
cp -a ././backup_files.sh /home/jpp9547/Escola/S0/testeProjeto/backup_files.sh
cp -a ././backup.sh /home/jpp9547/Escola/S0/testeProjeto/backup.sh
cp -a ././backup_summary.sh /home/jpp9547/Escola/S0/testeProjeto/backup_summary.sh
cp -a ././filewithotherfuncs.sh /home/jpp9547/Escola/S0/testeProjeto/filewithotherfuncs.sh
cp -a ././teste.txt /home/jpp9547/Escola/S0/testeProjeto/teste.txt
jpp9547@jpp-Vivobook:~/Escola/S0/projeto1/Projeto1S0$ vim teste.txt
jpp9547@jpp-Vivobook:~/Escola/S0/projeto1/Projeto1S0$ ./backup_files.sh ./ ~/Escola/S0/testeProjeto/
cp -a ././teste.txt /home/jpp9547/Escola/S0/testeProjeto/teste.txt
jpp9547@jpp-Vivobook:~/Escola/S0/projeto1/Projeto1S0$
```

# backup.sh

-b [file], com este parametro o script irá ignorar os ficheiros cujo nome estão no ficheiro file

-r [regex\_expression], com este parametro o script irá ignorar os ficheiros cujo nome se inclui na expressão regex [regex-expression]

## A seguir só as etapas que diferem do backup\_files.sh:

- Inicializar variáveis

```
...
filter=""
is_recursive=0
regex=".*"
hasExclude=0
...
```

- Leitura de Opções/Flags

```
# a option z recursividade e não deve ser usada pelo usuário
while getopts ":czb:r:" op; do
    case $op in # Processar as opções
        ...
    esac
done
```

- Carregar Arquivo de Exclusões

```
if [ -f "$exclude_file" ]; then
    mapfile -t exclude_list "$exclude_file"
else exclude_list=()
fi
```

- Iniciar o backup dos arquivos

**Para** cada item no diretório de origem:

Obter o nome base do item (sem o caminho)

Se o item corresponder à expressão regular fornecida:

Se o item for um arquivo e não estiver na lista de exclusões:

Se o modo de verificação estiver ativado:

Realizar a verificação (sem copiar o arquivo)

Caso contrário:

Copiar o arquivo para o diretório de backup

Se o item for um diretório e o backup for recursivo:

Realizar o backup recursivo para o diretório  
Fim para

## Testes

### Backup.sh

Teste parametro -b

```
jpp9547@jpp-Vivobook:~/Escola/S0/projeto1/Projeto1S0$ cat teste.txt
abc1
abc3
abc5
jpp9547@jpp-Vivobook:~/Escola/S0/projeto1/Projeto1S0$ ls testeProjeto/
abc1 abc2 abc3 abc4 abc5 abc6 abc7 abc8 abc9 file1 file2 file3 file4 file5 file6 file7 file8 file9
jpp9547@jpp-Vivobook:~/Escola/S0/projeto1/Projeto1S0$ ls ~/Escola/S0/testeProjeto/
jpp9547@jpp-Vivobook:~/Escola/S0/projeto1/Projeto1S0$ ./backup.sh -b teste.txt ./testeProjeto/ ~/Escola/S0/testeProjeto/
cp -a /home/jpp9547/Escola/S0/projeto1/Projeto1S0/testeProjeto/abc2 /home/jpp9547/Escola/S0/testeProjeto/abc2
cp -a /home/jpp9547/Escola/S0/projeto1/Projeto1S0/testeProjeto/abc3 /home/jpp9547/Escola/S0/testeProjeto/abc3
cp -a /home/jpp9547/Escola/S0/projeto1/Projeto1S0/testeProjeto/abc4 /home/jpp9547/Escola/S0/testeProjeto/abc4
cp -a /home/jpp9547/Escola/S0/projeto1/Projeto1S0/testeProjeto/abc5 /home/jpp9547/Escola/S0/testeProjeto/abc5
cp -a /home/jpp9547/Escola/S0/projeto1/Projeto1S0/testeProjeto/abc6 /home/jpp9547/Escola/S0/testeProjeto/abc6
cp -a /home/jpp9547/Escola/S0/projeto1/Projeto1S0/testeProjeto/abc7 /home/jpp9547/Escola/S0/testeProjeto/abc7
cp -a /home/jpp9547/Escola/S0/projeto1/Projeto1S0/testeProjeto/abc8 /home/jpp9547/Escola/S0/testeProjeto/abc8
cp -a /home/jpp9547/Escola/S0/projeto1/Projeto1S0/testeProjeto/abc9 /home/jpp9547/Escola/S0/testeProjeto/abc9
cp -a /home/jpp9547/Escola/S0/projeto1/Projeto1S0/testeProjeto/file1 /home/jpp9547/Escola/S0/testeProjeto/file1
cp -a /home/jpp9547/Escola/S0/projeto1/Projeto1S0/testeProjeto/file2 /home/jpp9547/Escola/S0/testeProjeto/file2
cp -a /home/jpp9547/Escola/S0/projeto1/Projeto1S0/testeProjeto/file3 /home/jpp9547/Escola/S0/testeProjeto/file3
cp -a /home/jpp9547/Escola/S0/projeto1/Projeto1S0/testeProjeto/file4 /home/jpp9547/Escola/S0/testeProjeto/file4
cp -a /home/jpp9547/Escola/S0/projeto1/Projeto1S0/testeProjeto/file5 /home/jpp9547/Escola/S0/testeProjeto/file5
cp -a /home/jpp9547/Escola/S0/projeto1/Projeto1S0/testeProjeto/file6 /home/jpp9547/Escola/S0/testeProjeto/file6
cp -a /home/jpp9547/Escola/S0/projeto1/Projeto1S0/testeProjeto/file7 /home/jpp9547/Escola/S0/testeProjeto/file7
cp -a /home/jpp9547/Escola/S0/projeto1/Projeto1S0/testeProjeto/file8 /home/jpp9547/Escola/S0/testeProjeto/file8
cp -a /home/jpp9547/Escola/S0/projeto1/Projeto1S0/testeProjeto/file9 /home/jpp9547/Escola/S0/testeProjeto/file9
jpp9547@jpp-Vivobook:~/Escola/S0/projeto1/Projeto1S0$ ls ~/Escola/S0/testeProjeto/
abc2 abc3 abc4 abc5 abc6 abc7 abc8 abc9 file1 file2 file3 file4 file5 file6 file7 file8 file9
jpp9547@jpp-Vivobook:~/Escola/S0/projeto1/Projeto1S0$
```

Teste parametro -r

```
jpp9547@jpp-Vlvobook: ~/Escola/S0/projeto1/Projeto1S0$ ls testeProjeto/
abc1 abc2 abc3 abc4 abc5 abc6 abc7 abc8 abc9 file1 file2 file3 file4 file5 file6 file7 file8 file9
jpp9547@jpp-Vlvobook: ~/Escola/S0/projeto1/Projeto1S0$ ls ~/Escola/S0/testeProjeto/
jpp9547@jpp-Vlvobook: ~/Escola/S0/projeto1/Projeto1S0$ ./backup.sh -r abc.* ./testeProjeto/ ~/Escola/S0/testeProjeto/
cp -a /home/jpp9547/Escola/S0/projeto1/Projeto1S0/testeProjeto/abc1 /home/jpp9547/Escola/S0/testeProjeto/abc1
cp -a /home/jpp9547/Escola/S0/projeto1/Projeto1S0/testeProjeto/abc2 /home/jpp9547/Escola/S0/testeProjeto/abc2
cp -a /home/jpp9547/Escola/S0/projeto1/Projeto1S0/testeProjeto/abc3 /home/jpp9547/Escola/S0/testeProjeto/abc3
cp -a /home/jpp9547/Escola/S0/projeto1/Projeto1S0/testeProjeto/abc4 /home/jpp9547/Escola/S0/testeProjeto/abc4
cp -a /home/jpp9547/Escola/S0/projeto1/Projeto1S0/testeProjeto/abc5 /home/jpp9547/Escola/S0/testeProjeto/abc5
cp -a /home/jpp9547/Escola/S0/projeto1/Projeto1S0/testeProjeto/abc6 /home/jpp9547/Escola/S0/testeProjeto/abc6
cp -a /home/jpp9547/Escola/S0/projeto1/Projeto1S0/testeProjeto/abc7 /home/jpp9547/Escola/S0/testeProjeto/abc7
cp -a /home/jpp9547/Escola/S0/projeto1/Projeto1S0/testeProjeto/abc8 /home/jpp9547/Escola/S0/testeProjeto/abc8
cp -a /home/jpp9547/Escola/S0/projeto1/Projeto1S0/testeProjeto/abc9 /home/jpp9547/Escola/S0/testeProjeto/abc9
jpp9547@jpp-Vlvobook: ~/Escola/S0/projeto1/Projeto1S0$ ls ~/Escola/S0/testeProjeto/
abc1 abc2 abc3 abc4 abc5 abc6 abc7 abc8 abc9
jpp9547@jpp-Vlvobook: ~/Escola/S0/projeto1/Projeto1S0$
```

## Teste com 2 parametros ao mesmo tempo

```
jpp9547@jpp-Vivobook: ~/Escola/s0/projeto1/Projeto1S0$ ls testeProjeto/
abc1 abc2 abc3 abc4 abc5 abc6 abc7 abc8 abc9 file1 file2 file3 file4 file5 file6 file7 file8 file9
jpp9547@jpp-Vivobook: ~/Escola/s0/projeto1/Projeto1S0$ ls ~/Escola/S0/testeProjeto/
jpp9547@jpp-Vivobook: ~/Escola/S0/projeto1/Projeto1S0$ ./backup.sh -c -r abc.* ./testeProjeto/ ~/Escola/S0/testeProjeto/
Checking activated
cp -a /home/jpp9547/Escola/S0/projeto1/Projeto1S0/testeProjeto/abc1 /home/jpp9547/Escola/S0/testeProjeto/abc1
cp -a /home/jpp9547/Escola/S0/projeto1/Projeto1S0/testeProjeto/abc2 /home/jpp9547/Escola/S0/testeProjeto/abc2
cp -a /home/jpp9547/Escola/S0/projeto1/Projeto1S0/testeProjeto/abc3 /home/jpp9547/Escola/S0/testeProjeto/abc3
cp -a /home/jpp9547/Escola/S0/projeto1/Projeto1S0/testeProjeto/abc4 /home/jpp9547/Escola/S0/testeProjeto/abc4
cp -a /home/jpp9547/Escola/S0/projeto1/Projeto1S0/testeProjeto/abc5 /home/jpp9547/Escola/S0/testeProjeto/abc5
cp -a /home/jpp9547/Escola/S0/projeto1/Projeto1S0/testeProjeto/abc6 /home/jpp9547/Escola/S0/testeProjeto/abc6
cp -a /home/jpp9547/Escola/S0/projeto1/Projeto1S0/testeProjeto/abc7 /home/jpp9547/Escola/S0/testeProjeto/abc7
cp -a /home/jpp9547/Escola/S0/projeto1/Projeto1S0/testeProjeto/abc8 /home/jpp9547/Escola/S0/testeProjeto/abc8
cp -a /home/jpp9547/Escola/S0/projeto1/Projeto1S0/testeProjeto/abc9 /home/jpp9547/Escola/S0/testeProjeto/abc9
jpp9547@jpp-Vivobook: ~/Escola/s0/projeto1/Projeto1S0$ ls ~/Escola/S0/testeProjeto/
jpp9547@jpp-Vivobook: ~/Escola/S0/projeto1/Projeto1S0$
```

Teste com pasta que existe na pasta destino mas que não existe na pasta que vai ser copiada

```
jpp9547@jpp-Vlvoobook:~/Escola/S0/projeto1/Projeto150$ ls testeProjeto/
file1 file2 file3 file4 file5 file6 file7 file8 file9
jpp9547@jpp-Vlvoobook:~/Escola/S0/projeto1/Projeto150$ ls ~/Escola/S0/testeProjeto/
abc file1 file2 file3 file4 file5 file6 file7 file8 file9
jpp9547@jpp-Vlvoobook:~/Escola/S0/projeto1/Projeto150$ ./backup.sh testeProjeto/ ~/Escola/S0/testeProjeto/
rm -r /home/jpp9547/Escola/S0/testeProjeto/abc
jpp9547@jpp-Vlvoobook:~/Escola/S0/projeto1/Projeto150$ ls ~/Escola/S0/testeProjeto/
file1 file2 file3 file4 file5 file6 file7 file8 file9
jpp9547@jpp-Vlvoobook:~/Escola/S0/projeto1/Projeto150$
```

Teste com pasta que existe na pasta que vai ser copiada mas não existe na pasta destino

```
jpp9547@jpp-Vivobook:~/Escola/S0/projeto1/Projeto1$ ls testeProjeto/
abc file1 file2 file3 file4 file5 file6 file7 file8 file9
jpp9547@jpp-Vivobook:~/Escola/S0/projeto1/Projeto1$ ls ~/Escola/S0/testeProjeto/
file1 file2 file3 file4 file5 file6 file7 file8 file9
jpp9547@jpp-Vivobook:~/Escola/S0/projeto1/Projeto1$ ./backup.sh testeProjeto/ ~/Escola/S0/testeProjeto/
mkdir /home/jpp9547/Escola/S0/testeProjeto/abc
jpp9547@jpp-Vivobook:~/Escola/S0/projeto1/Projeto1$ ls ~/Escola/S0/testeProjeto/
abc file1 file2 file3 file4 file5 file6 file7 file8 file9
jpp9547@jpp-Vivobook:~/Escola/S0/projeto1/Projeto1$
```



## Teste com pastas escondidas

```
jpp9547@jpp-Vlvobook:~/Escola/S0/projeto1/Projeto1S0$ ls ~/Escola/S0/testeProjeto/
jpp9547@jpp-Vlvobook:~/Escola/S0/projeto1/Projeto1S0$ ls
backup_check.sh  backup_files.sh  backup_summary.sh  functions  RelatorioSo
backup_checkv2.sh  backup.sh  filewithotherfuncs.sh  functionsReal  teste.txt
jpp9547@jpp-Vlvobook:~/Escola/S0/projeto1/Projeto1S0$ ./backup.sh ./ ~/Escola/S0/testeProjeto/
cp -a /home/jpp9547/Escola/S0/projeto1/Projeto1S0/backup_check.sh /home/jpp9547/Escola/S0/testeProjeto/backup_check.sh
cp -a /home/jpp9547/Escola/S0/projeto1/Projeto1S0/backup_checkv2.sh /home/jpp9547/Escola/S0/testeProjeto/backup_checkv2.sh
cp -a /home/jpp9547/Escola/S0/projeto1/Projeto1S0/backup_files.sh /home/jpp9547/Escola/S0/testeProjeto/backup_files.sh
cp -a /home/jpp9547/Escola/S0/projeto1/Projeto1S0/backup.sh /home/jpp9547/Escola/S0/testeProjeto/backup.sh
cp -a /home/jpp9547/Escola/S0/projeto1/Projeto1S0/backup_summary.sh /home/jpp9547/Escola/S0/testeProjeto/backup_summary.sh
cp -a /home/jpp9547/Escola/S0/projeto1/Projeto1S0/filewithotherfuncs.sh /home/jpp9547/Escola/S0/testeProjeto/filewithotherfuncs.sh
mkdir /home/jpp9547/Escola/S0/testeProjeto/functions
cp -a /home/jpp9547/Escola/S0/projeto1/Projeto1S0/functions/compModDate.sh /home/jpp9547/Escola/S0/testeProjeto/functions/compModDate.sh
cp -a /home/jpp9547/Escola/S0/projeto1/Projeto1S0/functions/copyFile.sh /home/jpp9547/Escola/S0/testeProjeto/functions/copyFile.sh
cp -a /home/jpp9547/Escola/S0/projeto1/Projeto1S0/functions/nfound.sh /home/jpp9547/Escola/S0/testeProjeto/functions/nfound.sh
mkdir /home/jpp9547/Escola/S0/testeProjeto/functionsReal
cp -a /home/jpp9547/Escola/S0/projeto1/Projeto1S0/functionsReal/compModDate.sh /home/jpp9547/Escola/S0/testeProjeto/functionsReal/compModDate
.sh
cp -a /home/jpp9547/Escola/S0/projeto1/Projeto1S0/functionsReal/copyFile.sh /home/jpp9547/Escola/S0/testeProjeto/functionsReal/copyFile.sh
cp -a /home/jpp9547/Escola/S0/projeto1/Projeto1S0/functionsReal/findElement.sh /home/jpp9547/Escola/S0/testeProjeto/functionsReal/fndElement
.sh
cp -a /home/jpp9547/Escola/S0/projeto1/Projeto1S0/functionsReal/nfound.sh /home/jpp9547/Escola/S0/testeProjeto/functionsReal/nfound.sh
mkdir /home/jpp9547/Escola/S0/testeProjeto/.git
mkdir /home/jpp9547/Escola/S0/testeProjeto/.git/branches
cp -a /home/jpp9547/Escola/S0/projeto1/Projeto1S0/.git/COMMIT_EDITMSG /home/jpp9547/Escola/S0/testeProjeto/.git/COMMIT_EDITMSG
cp -a /home/jpp9547/Escola/S0/projeto1/Projeto1S0/.git/config /home/jpp9547/Escola/S0/testeProjeto/.git/config
cp -a /home/jpp9547/Escola/S0/projeto1/Projeto1S0/.git/description /home/jpp9547/Escola/S0/testeProjeto/.git/description
cp -a /home/jpp9547/Escola/S0/projeto1/Projeto1S0/.git/FETCH_HEAD /home/jpp9547/Escola/S0/testeProjeto/.git/FETCH_HEAD
cp -a /home/jpp9547/Escola/S0/projeto1/Projeto1S0/.git/HEAD /home/jpp9547/Escola/S0/testeProjeto/.git/HEAD
mkdir /home/jpp9547/Escola/S0/testeProjeto/.git/hooks
cp -a /home/jpp9547/Escola/S0/projeto1/Projeto1S0/.git/hooks/applypatch-msg.sample /home/jpp9547/Escola/S0/testeProjeto/.git/hooks/applypatch
-msg.sample
cp -a /home/jpp9547/Escola/S0/projeto1/Projeto1S0/.git/hooks/commit-msg.sample /home/jpp9547/Escola/S0/testeProjeto/.git/hooks/commit-msg.sam
ple
cp -a /home/jpp9547/Escola/S0/projeto1/Projeto1S0/.git/hooks/fsmonitor-watchman.sample /home/jpp9547/Escola/S0/testeProjeto/.git/hooks/fsmoni
tor-watchman.sample
cp -a /home/jpp9547/Escola/S0/projeto1/Projeto1S0/.git/hooks/post-update.sample /home/jpp9547/Escola/S0/testeProjeto/.git/hooks/post-update.s
ample
cp -a /home/jpp9547/Escola/S0/projeto1/Projeto1S0/.git/hooks/pre-applypatch.sample /home/jpp9547/Escola/S0/testeProjeto/.git/hooks/pre-applyp
atch.sample
cp -a /home/jpp9547/Escola/S0/projeto1/Projeto1S0/.git/hooks/pre-commit.sample /home/jpp9547/Escola/S0/testeProjeto/.git/hooks/pre-commit.sam
ple
cp -a /home/jpp9547/Escola/S0/projeto1/Projeto1S0/.git/hooks/pre-merge-commit.sample /home/jpp9547/Escola/S0/testeProjeto/.git/hooks/pre-merg
```

Todos os testes que foram feitos para o backup\_files.sh também foram feitos para este script com algumas adaptações(assumindo já a existência de pastas e de ficheiros lá dentro incluindo pastas com espaços)



# backup\_summary.sh

-b [file], com este parametro o script irá ignorar os ficheiros cujo nome estão no ficheiro file

-r [regex\_expression], com este parametro o script irá ignorar os ficheiros cujo nome se inclui na expressão regex [regex-expression]

## A seguir só as etapas que diferem do backup.sh:

- Inicialização de variáveis

```
...  
# Contadores de status  
cError=0  
cWarnings=0  
cUpdated=0  
cCopied=0  
cDeleted=0  
# Tamanho dos arquivos copiados e excluídos  
sizeCopied=0  
sizeDeleted=0  
...
```

- Exibir estatísticas finais

```
# Exibir estatísticas detalhadas sobre o backup  
se modo recursivo:  
    exibir erro, aviso, arquivos copiados, excluídos, etc.  
senão:  
    exibir resumo final com o total de erros, avisos, arquivos  
copiados, deletados fim
```

- Neste código, após cada **cp** e **rm** as variáveis de contagem e tamanho são alterados

```

...
cp -a "$file" "$destination"
if [[ $? -ne 0 ]]; then
    ((cError++))
    return 1
else
    file_size=$(stat -c %s "$file")
    ((cCopied++))
    sizeCopied=$((sizeCopied + file_size))
    return 0
fi
...
if [ -d "$file" ]; then
    num_files=$(find "$file" -type f | wc -l)
    dir_size=$(du -sb "$file" | cut -f1)
    ((cDeleted += num_files))
    ((sizeDeleted += dir_size))
else
    file_size=$(stat --format=%s "$file")
    ((sizeDeleted += file_size))
    ((cDeleted++))
fi
if [ -z "$checking" ];then
    rm -r "$file"
fi
...

```

- Após as chamadas as variáveis são atualizadas

```

...
# res é o output de parametros da chamada recursiva
cError=$((cError + res[0]))
cWarnings=$((cWarnings + res[1]))
cUpdated=$((cUpdated + res[2]))
cCopied=$((cCopied + res[3]))
sizeCopied=$((sizeCopied + res[4]))
cDeleted=$((cDeleted + res[5]))
sizeDeleted=$((sizeDeleted + res[6]))
...

```

## Testes

Foram utilizados os mesmos testes que os do backup.sh devido à similaridade do código, acrescentando-se apenas alguns testes

Foi usado um teste em que a pasta destino tem 2 arquivos que não estão na pasta que vai ser copiada (um na raiz e outro em uma subpasta) e em que 1 arquivo na pasta destino é mais novo do que um da pasta que vai ser copiada.

O resto dos arquivos será igual

```
jpp9547@jpp-Vivobook:~/Escola/S0/projeto1/Projeto1S0$ ls testeProjeto/
abc1 abc2 abc3 abc4 abc5 abc6 abc7 abc8 abc9 teste
jpp9547@jpp-Vivobook:~/Escola/S0/projeto1/Projeto1S0$ ls testeProjeto/teste/
abc1 abc2 abc3 abc4 abc5 abc6 abc7 abc8 abc9 teste1 teste2 teste3 teste4 teste5 teste6 teste7 teste8 teste9
jpp9547@jpp-Vivobook:~/Escola/S0/projeto1/Projeto1S0$ ls ~/Escola/S0/testeProjeto/
abc2 abc3 abc4 abc5 abc6 abc7 abc8 abc9 teste
jpp9547@jpp-Vivobook:~/Escola/S0/projeto1/Projeto1S0$ ls ~/Escola/S0/testeProjeto/teste/
abc1 abc2 abc3 abc4 abc5 abc6 abc7 abc8 abc9 teste1 teste2 teste3 teste4 teste5 teste6 teste7 teste8 teste9
jpp9547@jpp-Vivobook:~/Escola/S0/projeto1/Projeto1S0$ ./backup_summary.sh testeProjeto/ ~/Escola/S0/testeProjeto/
cp -a /home/jpp9547/Escola/S0/projeto1/Projeto1S0/testeProjeto/abc1 /home/jpp9547/Escola/S0/testeProjeto/abc1
[WARNING] Backed file (/home/jpp9547/Escola/S0/testeProjeto/abc3) is newer than source file (/home/jpp9547/Escola/S0/projeto1/Projeto1S0/testeProjeto/abc3) (SHOULD NOT HAPPEN)
cp -a /home/jpp9547/Escola/S0/projeto1/Projeto1S0/testeProjeto/teste/teste8 /home/jpp9547/Escola/S0/testeProjeto/teste/teste8
While backing up /home/jpp9547/Escola/S0/projeto1/Projeto1S0/testeProjeto/teste: 0 Errors; 0 Warnings; 0 Updated; 1 Copied (0B); 0 deleted (0B)
While backing up /home/jpp9547/Escola/S0/projeto1/Projeto1S0/testeProjeto: 0 Errors; 1 Warnings; 0 Updated; 2 Copied (5B); 0 deleted (0B)
jpp9547@jpp-Vivobook:~/Escola/S0/projeto1/Projeto1S0$
```

Foi usado uma pasta em que uma subpasta e um ficheiro têm espaços no nome para testar como o programa trata estes ficheiros

```
jpp9547@jpp-Vivobook:~/Escola/S0/projeto1/Projeto1S0$ ls ~/Escola/S0/testeProjeto/
abc1 abc2 abc3 abc4 abc5 abc6 abc7 abc8 abc9 teste 'teste 2'
jpp9547@jpp-Vivobook:~/Escola/S0/projeto1/Projeto1S0$ ls ~/Escola/S0/testeProjeto/"teste 2"
'espaco 1'
jpp9547@jpp-Vivobook:~/Escola/S0/projeto1/Projeto1S0$ ls ~/Escola/S0/testeProjeto/teste
abc1 abc2 abc3 abc4 abc5 abc6 abc7 abc8 abc9 teste1 teste2 teste3 teste4 teste5 teste6 teste7 teste8 teste9
jpp9547@jpp-Vivobook:~/Escola/S0/projeto1/Projeto1S0$ ./backup_summary.sh ~/Escola/S0/testeProjeto/ ~/Escola/S0/"teste Projeto"
cp -a /home/jpp9547/Escola/S0/testeProjeto/abc1 /home/jpp9547/Escola/S0/teste Projeto/abc1
cp -a /home/jpp9547/Escola/S0/testeProjeto/abc2 /home/jpp9547/Escola/S0/teste Projeto/abc2
cp -a /home/jpp9547/Escola/S0/testeProjeto/abc3 /home/jpp9547/Escola/S0/teste Projeto/abc3
cp -a /home/jpp9547/Escola/S0/testeProjeto/abc4 /home/jpp9547/Escola/S0/teste Projeto/abc4
cp -a /home/jpp9547/Escola/S0/testeProjeto/abc5 /home/jpp9547/Escola/S0/teste Projeto/abc5
cp -a /home/jpp9547/Escola/S0/testeProjeto/abc6 /home/jpp9547/Escola/S0/teste Projeto/abc6
cp -a /home/jpp9547/Escola/S0/testeProjeto/abc7 /home/jpp9547/Escola/S0/teste Projeto/abc7
cp -a /home/jpp9547/Escola/S0/testeProjeto/abc8 /home/jpp9547/Escola/S0/teste Projeto/abc8
cp -a /home/jpp9547/Escola/S0/testeProjeto/abc9 /home/jpp9547/Escola/S0/teste Projeto/abc9
mkdir /home/jpp9547/Escola/S0/teste Projeto/teste
cp -a /home/jpp9547/Escola/S0/testeProjeto/teste/abc1 /home/jpp9547/Escola/S0/teste Projeto/teste/abc1
cp -a /home/jpp9547/Escola/S0/testeProjeto/teste/abc2 /home/jpp9547/Escola/S0/teste Projeto/teste/abc2
cp -a /home/jpp9547/Escola/S0/testeProjeto/teste/abc3 /home/jpp9547/Escola/S0/teste Projeto/teste/abc3
cp -a /home/jpp9547/Escola/S0/testeProjeto/teste/abc4 /home/jpp9547/Escola/S0/teste Projeto/teste/abc4
cp -a /home/jpp9547/Escola/S0/testeProjeto/teste/abc5 /home/jpp9547/Escola/S0/teste Projeto/teste/abc5
cp -a /home/jpp9547/Escola/S0/testeProjeto/teste/abc6 /home/jpp9547/Escola/S0/teste Projeto/teste/abc6
cp -a /home/jpp9547/Escola/S0/testeProjeto/teste/abc7 /home/jpp9547/Escola/S0/teste Projeto/teste/abc7
cp -a /home/jpp9547/Escola/S0/testeProjeto/teste/abc8 /home/jpp9547/Escola/S0/teste Projeto/teste/abc8
cp -a /home/jpp9547/Escola/S0/testeProjeto/teste/abc9 /home/jpp9547/Escola/S0/teste Projeto/teste/abc9
cp -a /home/jpp9547/Escola/S0/testeProjeto/teste/teste1 /home/jpp9547/Escola/S0/teste Projeto/teste/teste1
cp -a /home/jpp9547/Escola/S0/testeProjeto/teste/teste2 /home/jpp9547/Escola/S0/teste Projeto/teste/teste2
cp -a /home/jpp9547/Escola/S0/testeProjeto/teste/teste3 /home/jpp9547/Escola/S0/teste Projeto/teste/teste3
cp -a /home/jpp9547/Escola/S0/testeProjeto/teste/teste4 /home/jpp9547/Escola/S0/teste Projeto/teste/teste4
cp -a /home/jpp9547/Escola/S0/testeProjeto/teste/teste5 /home/jpp9547/Escola/S0/teste Projeto/teste/teste5
cp -a /home/jpp9547/Escola/S0/testeProjeto/teste/teste6 /home/jpp9547/Escola/S0/teste Projeto/teste/teste6
cp -a /home/jpp9547/Escola/S0/testeProjeto/teste/teste7 /home/jpp9547/Escola/S0/teste Projeto/teste/teste7
cp -a /home/jpp9547/Escola/S0/testeProjeto/teste/teste8 /home/jpp9547/Escola/S0/teste Projeto/teste/teste8
cp -a /home/jpp9547/Escola/S0/testeProjeto/teste/teste9 /home/jpp9547/Escola/S0/teste Projeto/teste/teste9
While backing up /home/jpp9547/Escola/S0/testeProjeto/teste: 0 Errors; 0 Warnings; 0 Updated; 18 Copied (3B); 0 deleted (0B)
mkdir /home/jpp9547/Escola/S0/teste Projeto/teste 2
cp -a /home/jpp9547/Escola/S0/testeProjeto/teste 2/espaco 1 /home/jpp9547/Escola/S0/teste Projeto/teste 2/espaco 1
While backing up /home/jpp9547/Escola/S0/testeProjeto/teste 2: 0 Errors; 0 Warnings; 0 Updated; 1 Copied (0B); 0 deleted (0B)
While backing up /home/jpp9547/Escola/S0/testeProjeto: 0 Errors; 0 Warnings; 0 Updated; 28 Copied (8B); 0 deleted (0B)
jpp9547@jpp-Vivobook:~/Escola/S0/projeto1/Projeto1S0$ ls ~/Escola/S0/"teste Projeto"
abc1 abc2 abc3 abc4 abc5 abc6 abc7 abc8 abc9 teste 'teste 2'
jpp9547@jpp-Vivobook:~/Escola/S0/projeto1/Projeto1S0$
```

Foram usadas duas pastas com os mesmos ficheiros

```
jpp9547@jpp-Vivobook:~/Escola/S0/projeto1/Projeto1S0$ ./backup_summary.sh ~/Escola/S0/testeProjeto/ ~/Escola/S0/"teste Projeto"
While backing up /home/jpp9547/Escola/S0/testeProjeto/teste: 0 Errors; 0 Warnings; 0 Updated; 0 Copied (0B); 0 deleted (0B)
While backing up /home/jpp9547/Escola/S0/testeProjeto/teste 2: 0 Errors; 0 Warnings; 0 Updated; 0 Copied (0B); 0 deleted (0B)
While backing up /home/jpp9547/Escola/S0/testeProjeto: 0 Errors; 0 Warnings; 0 Updated; 0 Copied (0B); 0 deleted (0B)
jpp9547@jpp-Vivobook:~/Escola/S0/projeto1/Projeto1S0$
```

# Backup\_check.sh

## Etapas

- Validar inputs do script

```
if [[ $# != 2 ]]; then
    echo "Script has to receive 2 arguments"
    return 1
fi

if [[ ! -d $1 && -d $2 ]]; then
    echo "Arguments have to be directories"
    return 1
fi
```

- Retirar a "/" (caso exista) do fim do path da diretoria (para normalizar o comportamento do código caso o input seja com ou sem a "/" no fim)

```
work_dir=$1
backup_dir=$2

# removes last bar(/) from backup_dir path (for formatting reasons)
if [[ $work_dir == */ ]]; then
    work_dir="${work_dir:0:-1}"
fi

if [[ $backup_dir == */ ]]; then
    backup_dir="${backup_dir:0:-1}"
fi
```

- Iterar pelos ficheiros e ver se são diferentes

```

se backup_dir existe e não está vazio, então
    para cada ficheiro em backup_dir
        se ficheiro é um directorio
            chamar funcao para com argumentos ,
            "work_dir/nome do ficheiro" e ficheiro
            senao
                se ficheiro é um ficheiro normal
                    se md5sum(ficheiro) !=
md5sum(work_dir/nome de ficheiro)
                                print("os ficheiros sao
diferentes")

```

## Testes

Foi usada uma pasta com todos os ficheiros iguais menos 2, um na raiz e outro em uma subpasta

```

jpp9547@jpp-Vivobook:~/Escola/S0/projeto1/Projeto1S0$ ls testeProjeto/
abc1 abc2 abc3 abc4 abc5 abc6 abc7 abc8 abc9 teste
jpp9547@jpp-Vivobook:~/Escola/S0/projeto1/Projeto1S0$ ls testeProjeto/teste/
abc1 abc2 abc3 abc4 abc5 abc6 abc7 abc8 abc9 teste1 teste2 teste3 teste4 teste5 teste6 teste7 teste8 teste9
jpp9547@jpp-Vivobook:~/Escola/S0/projeto1/Projeto1S0$ ls ~/Escola/S0/testeProjeto/
abc1 abc2 abc3 abc4 abc5 abc6 abc7 abc8 abc9 teste
jpp9547@jpp-Vivobook:~/Escola/S0/projeto1/Projeto1S0$ ls ~/Escola/S0/testeProjeto/teste
abc1 abc2 abc3 abc4 abc5 abc6 abc7 abc8 abc9 teste1 teste2 teste3 teste4 teste5 teste6 teste7 teste8 teste9
jpp9547@jpp-Vivobook:~/Escola/S0/projeto1/Projeto1S0$ ./backup_checkv2.sh testeProjeto/ ~/Escola/S0/testeProjeto/
jpp9547@jpp-Vivobook:~/Escola/S0/projeto1/Projeto1S0$ vim
jpp9547@jpp-Vivobook:~/Escola/S0/projeto1/Projeto1S0$ ./backup_checkv2.sh testeProjeto/ ~/Escola/S0/testeProjeto/
/home/jpp9547/Escola/S0/testeProjeto/abc1 and testeProjeto/abc1 differ
/home/jpp9547/Escola/S0/testeProjeto/teste/teste2 and testeProjeto/teste/teste2 differ
jpp9547@jpp-Vivobook:~/Escola/S0/projeto1/Projeto1S0$ 

```

Foram usadas subpastas com espaços e um ficheiro diferentes também ele com espaços para testar o comportamento do programa quer com subpastas quer com nomes que tenham espaços

```

jpp9547@jpp-Vivobook:~/Escola/S0/projeto1/Projeto1S0$ ls ~/Escola/S0/testeProjeto/
abc1 abc2 abc3 abc4 abc5 abc6 abc7 abc8 abc9 teste 'teste 2'
jpp9547@jpp-Vivobook:~/Escola/S0/projeto1/Projeto1S0$ ls ~/Escola/S0/testeProjeto/teste
abc1 abc2 abc3 abc4 abc5 abc6 abc7 abc8 abc9 teste1 teste2 teste3 teste4 teste5 teste6 teste7 teste8 teste9
jpp9547@jpp-Vivobook:~/Escola/S0/projeto1/Projeto1S0$ ls ~/Escola/S0/testeProjeto/"teste 2"
'espaco 1'
jpp9547@jpp-Vivobook:~/Escola/S0/projeto1/Projeto1S0$ ls ~/Escola/S0/"teste Projeto"
abc1 abc2 abc3 abc4 abc5 abc6 abc7 abc8 abc9 teste 'teste 2'
jpp9547@jpp-Vivobook:~/Escola/S0/projeto1/Projeto1S0$ ls ~/Escola/S0/"teste Projeto/teste"
abc1 abc2 abc3 abc4 abc5 abc6 abc7 abc8 abc9 teste1 teste2 teste3 teste4 teste5 teste6 teste7 teste8 teste9
jpp9547@jpp-Vivobook:~/Escola/S0/projeto1/Projeto1S0$ ls ~/Escola/S0/"teste Projeto/teste 2"
'espaco 1'
jpp9547@jpp-Vivobook:~/Escola/S0/projeto1/Projeto1S0$ ./backup_checkv2.sh ~/Escola/S0/"teste Projeto" ~/Escola/S0/testeProjeto/
/home/jpp9547/Escola/S0/testeProjeto/teste 2/espaco 1 and /home/jpp9547/Escola/S0/teste Projeto/teste 2/espaco 1 differ
jpp9547@jpp-Vivobook:~/Escola/S0/projeto1/Projeto1S0$ 

```

# Como resolvemos certos problemas

Durante o desenvolvimento, surgiram alguns desafios que puderam ser superados através de pesquisas.

## Backup de ficheiros escondidos (.file)

O bash por padrão só inclui no *globbing* os arquivos que não começam com ponto, logo os ficheiros escondidos não são normalmente incluídos em comandos como *ls*,... . Este desafio foi resolvido com uma simples linha de comando.

```
shopt -s dotglob # Faz com que os ficheiros começados com '.' sejam incluídos no globbing
```

## Verificar se o diretório destino estava dentro da source

Verificar se o diretório de destino estava dentro da source é um passo importante para o **backup.sh** e **backup\_summary.sh** visto que estes 2 utilizam recursividade para iterar sobre todos os diretórios da pasta source e caso o diretório destino estivesse dentro da source aconteceria um loop infinito.

A função **realpath** do bash foi de grande ajuda para a solução que encontramos.

```
source_dir = argumento 1 (diretório de origem)
backup_dir = argumento 2 (diretório de backup)

.
Se source_dir não for um caminho válido:
    Exibir : "Não foi possível resolver o caminho do diretório de origem"
    Finalizar o processo
Se backup_dir não for um caminho válido:
    Exibir "Não foi possível resolver o caminho do diretório de backup"
    Finalizar o processo
Se backup_dir é um subdiretório de source_dir:
    Exibir: "[ERRO] backup_dir está dentro de source_dir"
    Finalizar o processo
```

**[NOTA]** no backup\_summary.sh o "Finalizar o processo" é feito através da função **end\_print()**.

### Usar *realpath* para chamadas recursivas

A solução que encontramos para este ponto foi justamente **não** usar o *realpath* em todas as chamadas recursivas. De facto pela forma como o código está estruturado o *realpath* só precisa ser utilizado na primeira chamada. Para controlar em qual chamada é que estavamos foi fácil, só foi preciso adicionar uma nova **flag ( opção )** no *getopts* que assinala se uma chamada é recursiva ou não.

### Display dos passos efetuados na chamada recursiva

Ao fazer a chamada recursiva nós guardamos todos os outputs numa variável para termos um melhor controlo sobre os displays.

```
...  
output = "output da chamada recursiva"  
...
```

Depois escolhemos quais linhas do output é que queremos dar display (**echo**)

```
...  
output = "output da chamada recursiva"  
echo $output" | grep -E '^(cp|mkdir|rm|While)' # display só às informações importantes  
...
```

### Summary das alterações no backup\_summary.sh

A diferença entre o backup.sh e o backup\_summary.sh é a disponibilização de informação à cerca das alterações realizadas pelo script. Para superar esta etapa a solução que encontramos foi de usar uma versão diferente das funções que são usadas no backup.sh.

Então no ficheiro backup\_summary.sh contêm uma copia definição de todas as funções no próprio ficheiro. Essas cópias contam com alterações para que a contagem possa ser feita.

Alterações do ficheiro:

Variáveis para contagem (**counters**)

```
# contadores de ocorrências
cError=0
cWarnings=0
cUpdated=0
cCopied=0
cDeleted=0
# contadores de tamanho
sizeCopied=0
sizeDeleted=0
```

## Manter a contagem através de chamadas recursivas

```
output= "Saídas da chamada Recursiva"
...
res= "última linha do output da recursiva" # A última linha do output
corresponde ao end_print() do diretório filho
# Adicionar à contagem atual a contagem dos seus subdiretórios
cError += res->cErros
cWarnings += res->cWarnings
cUpdated += res->cUpdated
cCopied += res->cCopied
sizeCopied += res->sizeCopied
cDeleted += res->cDeleted
sizeDeleted += res->sizeDeleted
```

**[NOTA]** os contadores foram colocados a seguir a todas as funções *rm*, *cp*.



# Bibliografia

- Stack Overflow. (n.d.). *Stack Overflow: Where developers learn, share, & build careers*. Recuperado de <https://stackoverflow.com/>
- GNU Project. (n.d.). *Bash manual*. Recuperado de (<https://www.gnu.org/software/bash/manual/bash.html>)
- Linux Die. (n.d.). *Bash man page*. Recuperado de <https://linux.die.net/man/1/bash>