

UNIVERSIDADE FEDERAL DO PARANÁ

ARIANE DE OLIVEIRA NEVES GRR20253191

YSIS BARDUCO STRAUB DE LIMA GRR20251303

TADS VESPERTINO

**DS010 – ADMINISTRAÇÃO DE SISTEMAS**  
**PROJETO FINAL**

CURITIBA/PR

2025

ARIANE DE OLIVEIRA NEVES  
YSIS BARDUCO STRAUB DE LIMA

**DS010 – ADMINISTRAÇÃO DE SISTEMAS**  
**PROJETO FINAL**

Trabalho apresentado à Universidade Federal do Paraná - UFPR, Setor de Educação Profissional e Tecnológica, com o projeto do Ubuntu para a obtenção de nota na disciplina de Administração de Sistemas – DS010, do curso de Tecnologia em Análise e desenvolvimento de Sistemas.

Docente: Prof. Mauro Castro.

CURITIBA/PR

2025

## **LISTA DE FIGURAS**

FIGURA 1 – UBUNTO ANTES.....	14
FIGURA 2 – UBUNTO DEPOIS.....	15
FIGURA 3 – UBUNTO RESULTADO.....	16

## SUMÁRIO

<b>1. INTRODUÇÃO .....</b>	<b>5</b>
1.1 OBJETIVO .....	5
1.2 ESCOPO DO PROGRAMA.....	5
<b>2. DESENVOLVIMENTO.....</b>	<b>6</b>
2.1 SCRIPT .....	6
2.1.1 backup.bash.....	6
2.1.2 Explicação linha a linha .....	9
3.1. Antes da execução .....	13
3.2. Executando backup.bash.....	13
3.3. Resultado .....	15
<b>4. CONCLUSÃO.....</b>	<b>16</b>
<b>Referências .....</b>	<b>17</b>

# **1. INTRODUÇÃO**

Este documento tem a finalidade de apresentar um programa desenvolvido para realização de backups, de documentos e arquivos. Esse programa proposto implementado utilizando o sistema operacional Ubuntu, sendo uma distribuição Linux, e utilizando suas ferramentas disponíveis.

Essa escolha nos permitiu um programa fluido, a integridade dos dados e sua recuperação quando necessário.

## **1.1 OBJETIVO**

Desenvolver um programa que faz backups de arquivos e documentos, em um diretório de armazenamento escolhida pelo usuário.

## **1.2 ESCOPO DO PROGRAMA**

Esse programa deverá executar backups em um armazenamento local ou remoto.

## 2. DESENVOLVIMENTO

O desenvolvimento do trabalho foi realizado com base nos princípios em Administração de Sistemas, buscando atender ao objetivo desse projeto de fazer um programa que faça backups de arquivos e diretórios em um armazenamento local ou remoto.

Esse programa foi feito utilizando o terminal do Ubuntu, com o auxílio de uma ferramenta que fica dentro do terminal chamada `./nano`, sendo uma programação em shell script. Isso nos permitiu criar as linhas de código que compõem todo o corpo do programa.

### 2.1 SCRIPT

#### 2.1.1 backup.bash

```
#!/bin/bash
```

#Script para executar um backup, sincronizando arquivos, verificando e validando a origem e o destino

```
origens=()
```

#Função que valida o diretório digitado

```
valD(){
```

```
if [ ! -d "$1" ]; then
```

```
    echo "Diretório $1 não encontrado."
```

```
    return 1
```

```
else
```

```
    return 0
```

```
fi
```

```
}
```

#O usuário decide quais pastas deseja salvar

```

Origem(){
while true; do
    echo "Para terminar, digite 'fim'."
    echo -n "Digite o caminho do diretório que deseja salvar: "
    read origem

    if [ "$origem" == "fim" ]; then
        break
    fi

    valD "$origem"
    if [ $? -eq 0 ]; then
        origens+=("$origem")
    fi
done

if [ ${#origens[@]} -eq 0 ]; then
    echo "Nenhum diretório válido informado."
    echo
    Origem
else
    echo
    Destino
fi
}

```

#O usuário seleciona o local onde irá salvar o backup

```

Destino(){
echo -n "Digite o caminho do local de destino: "
read destino

valD "$destino"

if [ $? -eq 0 ]; then
    Backup
else
    echo "Destino inválido."
    echo
    Destino
fi
}

#Executa o backup utilizando o rsync
Backup(){
data=$(date "+%d-%m-%Y_%H%M%S")
logfile="$destino/backup_$data.log"
echo "Log registrado em $logfile"
echo
echo "Iniciando o backup..."

for origem in ${origens[@]}; do
    nome=$(basename "$origem")
    subpasta="$destino/$nome"
    echo "Carregando de $origem para $destino..."

```



```
rsync -auv --progress --delete --exclude='.DS_Store' --delete-excluded --log-file="$logfile"  
"$origem/" "$subpasta/"
```

```
done
```

```
echo
```

```
echo "Backup finalizado."
```

```
}
```

```
#Inicia o programa
```

```
Origem
```

### 2.1.2 Explicação linha a linha

O programa inicia com a linha shebang, Essa linha, indica para a máquina que programa deve ser interpretado em bash (Bourne Again Shell)  
#!/bin/bash

Em seguida declaramos a variável global onde serão armazenados os diretórios de origem:

```
origens=()
```

Este array precisa ser declarado fora da função Origem, caso contrário ele será zerado sempre que a função for chamada.

A função valD verifica se o caminho para o diretório existe. Se não existe, o programa envia uma mensagem de erro e retorna o valor 1. Caso contrário, ele retorna o valor 0.

```
valD(){
```

```
if [ ! -d "$1" ]; then
```

```
    echo "Diretório $1 não encontrado."
```

```
    return 1
```

```
else
```

```
    return 0
```

```
fi  
}
```

A função Origem, pede para que o usuário digite o caminho do diretório que deseja salvar no destino. utiliza um laço de repetição while para que o usuário possa entrar com mais de um diretório, se o usuário digitar “fim” o laço de repetição é interrompido e o programa verifica se há algum diretório salvo no vetor origens, se não há ele exibe uma mensagem de erro e reinicia a função, caso contrário, executa a função Destino.

```
Origem(){  
while true; do  
    echo "Para terminar, digite 'fim'. "  
    echo -n "Digite o caminho do diretório que deseja salvar: "  
    read origem  
  
    if [ "$origem" == "fim" ]; then  
        break  
    fi  
  
    valD "$origem"  
    if [ $? -eq 0 ]; then  
        origens+=("$origem")  
    fi  
done  
  
if [ ${#origens[@]} -eq 0 ]; then  
    echo "Nenhum diretório válido informado."  
    echo  
    Origem
```

```

else
    echo
    Destino
fi
}

```

A função destino, pede para que o usuário digite o caminho do diretório onde será armazenado o backup e o guarda na variável destino. Em seguida, ele utiliza função valD para validar o diretório, se resposta retornada por valD for igual a 0, significa que o destino é valido e o programa prossegue com a função Backup, caso contrário o programa retorna uma mensagem de erro e reinicia a função Destino.

```

Destino(){
echo -n "Digite o caminho do local de destino: "
read destino

valD "$destino"

if [ $? -eq 0 ]; then
    Backup
else
    echo "Destino inválido."
    echo
    Destino
fi
}

```

A função backup, cria e registra o log em um arquivo no diretório de destino. Depois, utiliza um laço for para copiar o conteúdo de um diretório de origem do array origens, por vez, para o diretório de destino utilizando o comando rsync. O parâmetro —delete do rsync deleta os arquivos que não

estão presentes no diretório, portanto, para evitar que os arquivos do diretório anterior sejam apagados, o programa cria uma subpasta dentro do destino, para cada diretório.

```
Backup(){
```

```
data=$(date "+%d-%m-%Y_%H%M%S")
```

```
logfile="$destino/backup_$data.log"
```

```
echo "Log registrado em $logfile"
```

```
echo
```

```
echo "Iniciando o backup..."
```

```
for origem in ${origens[@]}; do
```

```
    nome=$(basename "$origem")
```

```
    subpasta="$destino/$nome"
```

```
    echo "Carregando de $origem para $destino..."
```

```
    rsync -auv --progress --delete --exclude='.DS_Store' --delete-excluded --log-file="$logfile"  
"$origem/" "$subpasta/"
```

```
done
```

```
echo
```

```
echo "Backup finalizado."
```

```
}
```

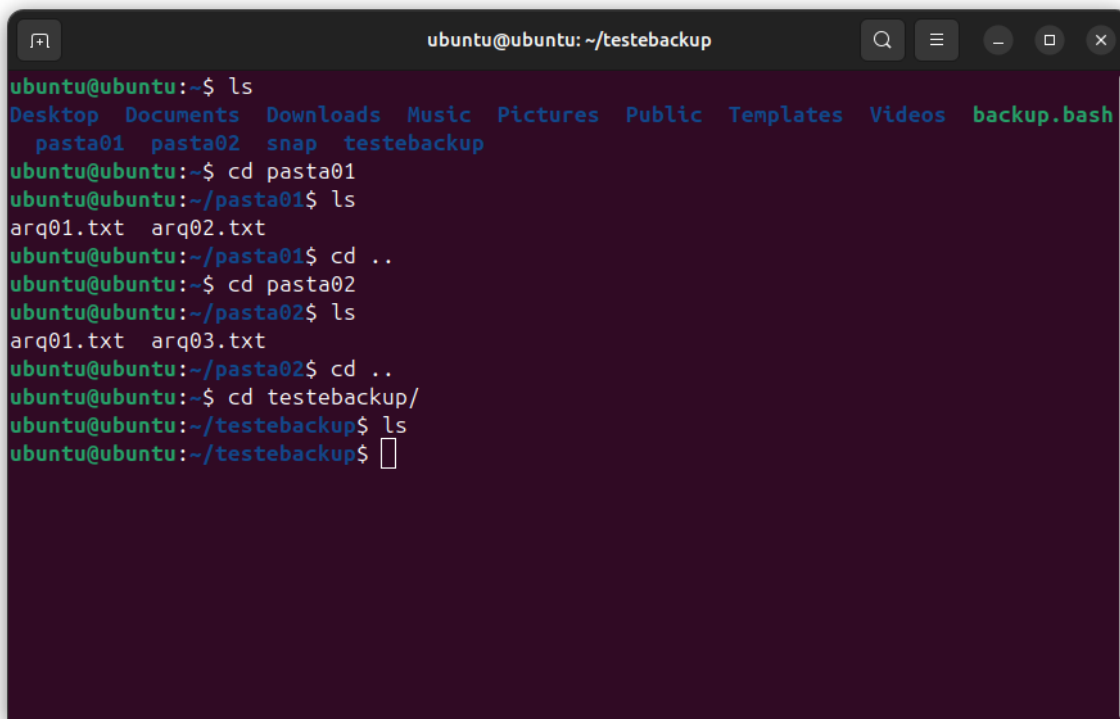
O programa é iniciado ao executar a função Origem

Origem

### 3. EXECUÇÃO DO PROGRAMA

#### 3.1. Antes da execução

FIGURA 1 – ANTES DE EXECUTAR O BACKUP

A terminal window titled 'ubuntu@ubuntu: ~/testebakup' with search, menu, and window control icons. The terminal shows the following commands and output:

```
ubuntu@ubuntu:~$ ls
Desktop  Documents  Downloads  Music  Pictures  Public  Templates  Videos  backup.bash
pasta01  pasta02  snap  testebakup
ubuntu@ubuntu:~$ cd pasta01
ubuntu@ubuntu:~/pasta01$ ls
arq01.txt  arq02.txt
ubuntu@ubuntu:~/pasta01$ cd ..
ubuntu@ubuntu:~$ cd pasta02
ubuntu@ubuntu:~/pasta02$ ls
arq01.txt  arq03.txt
ubuntu@ubuntu:~/pasta02$ cd ..
ubuntu@ubuntu:~$ cd testebakup/
ubuntu@ubuntu:~/testebakup$ ls
ubuntu@ubuntu:~/testebakup$
```

Vamos realizar um backup dos diretórios pasta01 e pasta02, com destino ao diretório testebakup. Na pasta01, contém os arquivos arq01.txt e arq02.txt. A pasta02, contém os arquivos arq01.txt e arq03.txt, note que um dos arquivos possui o mesmo nome de um dos arquivos da pasta01. O diretório de destino, testebakup, se encontra vazio.

#### 3.2. Executando backup.bash

FIGURA 2 – EXECUTANDO O BACKUP

```
ubuntu@ubuntu: ~  
ubuntu@ubuntu:~$ chmod +x backup.bash  
ubuntu@ubuntu:~$ ./backup.bash  
Para terminar, digite 'fim'.  
Digite o caminho do diretório que deseja salvar: /home/ubuntu/pasta01  
Para terminar, digite 'fim'.  
Digite o caminho do diretório que deseja salvar: /home/ubuntu/pasta02  
Para terminar, digite 'fim'.  
Digite o caminho do diretório que deseja salvar: /home/ubuntu/pasta03  
Diretório /home/ubuntu/pasta03 não encontrado.  
Para terminar, digite 'fim'.  
Digite o caminho do diretório que deseja salvar: fim  
  
Digite o caminho do local de destino: /home/ubuntu/teste  
Diretório /home/ubuntu/teste não encontrado.  
Destino inválido.  
  
Digite o caminho do local de destino: /home/ubuntu/testebackup  
Log registrado em /home/ubuntu/testebackup/backup_16-06-2025_110526.log  
  
Iniciando o backup...  
Carregando de /home/ubuntu/pasta01 para /home/ubuntu/testebackup...  
sending incremental file list  
created directory /home/ubuntu/testebackup/pasta01  
./  
arq01.txt  
0 100% 0.00kB/s 0:00:00 (xfr#1, to-chk=1/3)  
arq02.txt  
0 100% 0.00kB/s 0:00:00 (xfr#2, to-chk=0/3)  
  
sent 191 bytes received 112 bytes 606.00 bytes/sec  
total size is 0 speedup is 0.00  
Carregando de /home/ubuntu/pasta02 para /home/ubuntu/testebackup...  
sending incremental file list  
created directory /home/ubuntu/testebackup/pasta02  
./  
arq01.txt  
0 100% 0.00kB/s 0:00:00 (xfr#1, to-chk=1/3)  
arq03.txt  
0 100% 0.00kB/s 0:00:00 (xfr#2, to-chk=0/3)  
  
sent 191 bytes received 112 bytes 606.00 bytes/sec  
total size is 0 speedup is 0.00  
  
Backup finalizado.  
ubuntu@ubuntu:~$
```

Agora, utilizaremos o programa backup.bash para realizar o backup.

Primeiro, através do comando `chmod +x backup.bash` o arquivo deve receber permissão de execução. Então, poderemos executá-lo com o comando `./backup.bash`

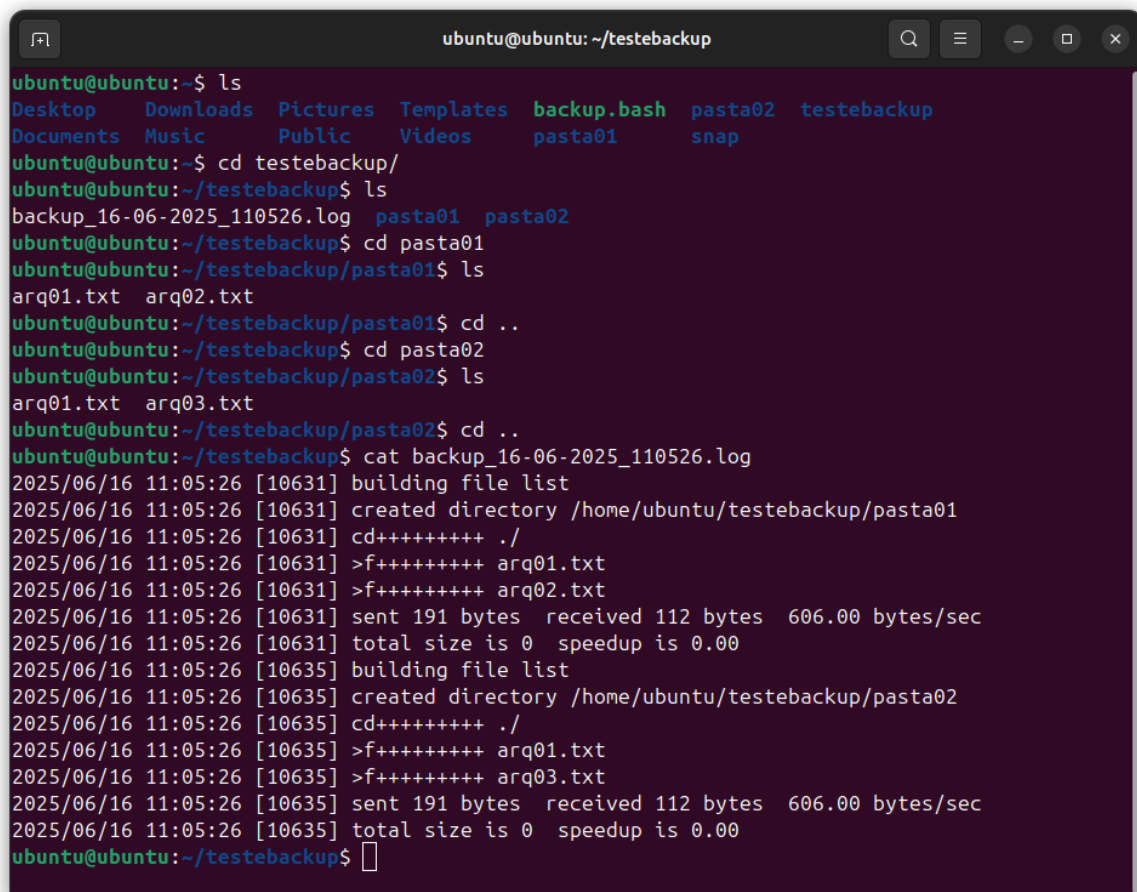
Assim que inicia, o programa pede para o usuário que digite o caminho dos diretórios de origem que deseja copiar para o destino, no nosso caso, a pasta01 se encontra em `/home/ubuntu/pasta01` e a pasta02 em `/home/ubuntu/pasta02`, quando o usuário digita um caminho que não existe na máquina, o programa exibe uma mensagem de erro e continua o perguntando pelos diretórios de origem até que o usuário digite “fim”.

Em, seguida o programa pede para que o usuário digite o caminho do diretório de destino onde será realizado o backup, no nosso caso /home/ubuntu/testebackup.

Então, o programa cria um arquivo log e inicia o backup, copiando os arquivos em subpastas para o destino.

### 3.3. Resultado

FIGURA 3 - RESULTADO



```
ubuntu@ubuntu: ~/testebackup
ubuntu@ubuntu:~$ ls
Desktop  Downloads  Pictures  Templates  backup.bash  pasta02  testebackup
Documents Music      Public    Videos     pasta01      snap
ubuntu@ubuntu:~$ cd testebackup/
ubuntu@ubuntu:~/testebackup$ ls
backup_16-06-2025_110526.log  pasta01  pasta02
ubuntu@ubuntu:~/testebackup$ cd pasta01
ubuntu@ubuntu:~/testebackup/pasta01$ ls
arq01.txt  arq02.txt
ubuntu@ubuntu:~/testebackup/pasta01$ cd ..
ubuntu@ubuntu:~/testebackup$ cd pasta02
ubuntu@ubuntu:~/testebackup/pasta02$ ls
arq01.txt  arq03.txt
ubuntu@ubuntu:~/testebackup/pasta02$ cd ..
ubuntu@ubuntu:~/testebackup$ cat backup_16-06-2025_110526.log
2025/06/16 11:05:26 [10631] building file list
2025/06/16 11:05:26 [10631] created directory /home/ubuntu/testebackup/pasta01
2025/06/16 11:05:26 [10631] cd+++++++ ./
2025/06/16 11:05:26 [10631] >f+++++++ arq01.txt
2025/06/16 11:05:26 [10631] >f+++++++ arq02.txt
2025/06/16 11:05:26 [10631] sent 191 bytes  received 112 bytes  606.00 bytes/sec
2025/06/16 11:05:26 [10631] total size is 0  speedup is 0.00
2025/06/16 11:05:26 [10635] building file list
2025/06/16 11:05:26 [10635] created directory /home/ubuntu/testebackup/pasta02
2025/06/16 11:05:26 [10635] cd+++++++ ./
2025/06/16 11:05:26 [10635] >f+++++++ arq01.txt
2025/06/16 11:05:26 [10635] >f+++++++ arq03.txt
2025/06/16 11:05:26 [10635] sent 191 bytes  received 112 bytes  606.00 bytes/sec
2025/06/16 11:05:26 [10635] total size is 0  speedup is 0.00
ubuntu@ubuntu:~/testebackup$
```

Como resultado, o diretório testebackup possui uma cópia dos diretórios pasta01 e pasta02 e seus respectivos arquivos, arq01.txt, arq02.txt, arq03.txt e arq01.txt. Note que mesmo com nomes duplicados os arquivos foram mantidos, pois se encontram em subpastas diferentes, evitando interferências

## **4. CONCLUSÃO**

Nosso projeto foi concluído com sucesso, atingindo o objetivo, fazendo backups de arquivos ou diretórios em um diretório de destino. Uma ferramenta que foi utilizada e teve como papel principal para a conclusão desse projeto, foi o rsync.

O comando rsync permite sincronizar arquivos e diretórios entre locais, podendo ser em um armazenamento local ou remoto.



## Referências

Slides das aulas de Administração de Sistemas:

Mauro Castro. Apresentação **P02.slides\_unix**.

Mauro Castro. Apresentação **P03.slides\_unix**

Mauro Castro. Apresentação **P04.slides\_unix**

Mauro Castro. Apresentação **P05.slides\_unix**

Mauro Castro. Apresentação **P06.slides\_unix**

Mauro Castro. Apresentação **P07.slides\_unix**

Mauro Castro. Apresentação **P08.slides\_unix**

Mauro Castro. **UsoDoRsync**

Slackjeff. Vídeo **Criação e automação de BACKUPS com Shell Script – Linux.**

[https://youtu.be/NjVtMaZR47Q?si=kggsmOMb1H\\_2bAnt](https://youtu.be/NjVtMaZR47Q?si=kggsmOMb1H_2bAnt)