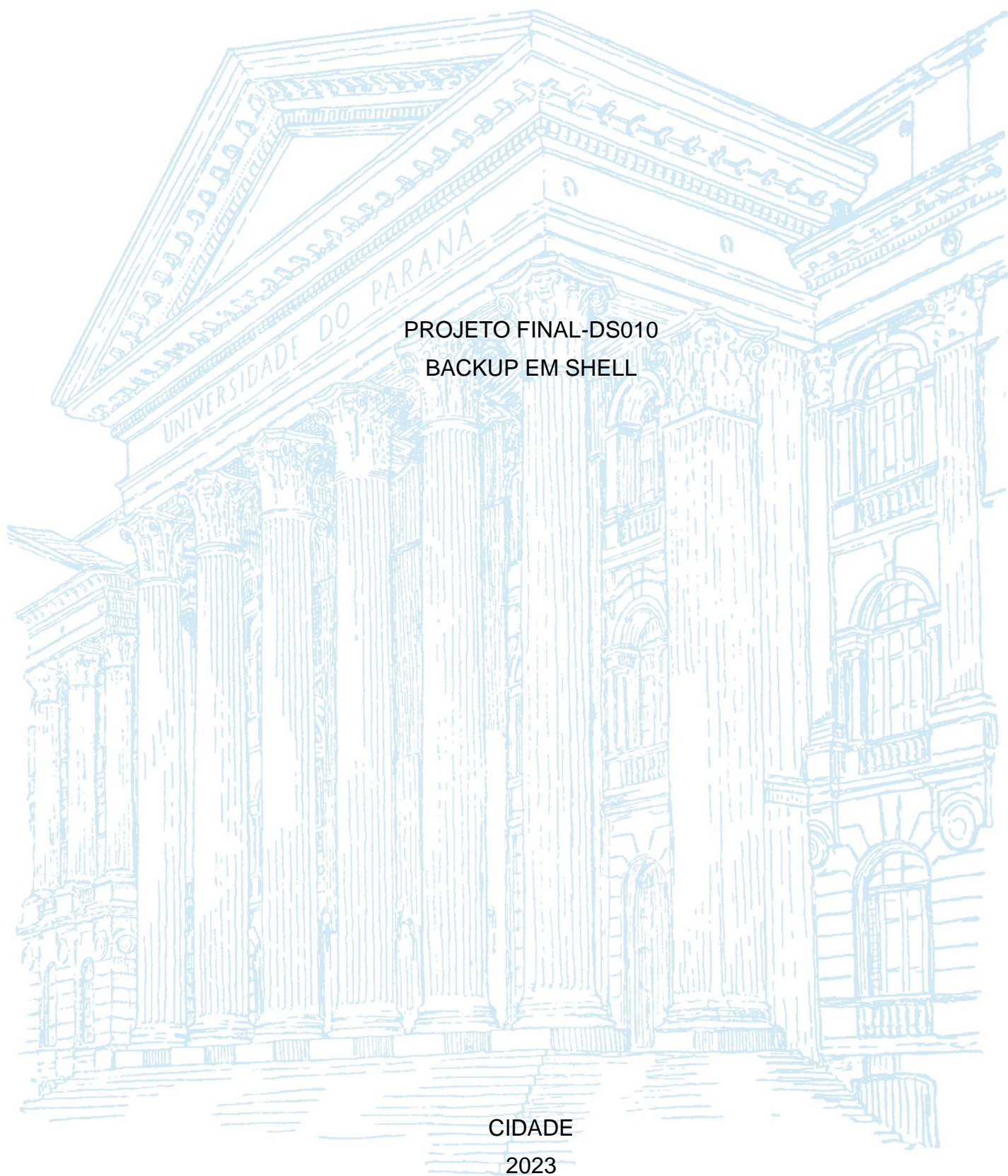


UNIVERSIDADE FEDERAL DO PARANÁ

JOÃO PEDRO QUEIROZ DEGER

PROJETO FINAL-DS010
BACKUP EM SHELL

CIDADE
2023



JOÃO PEDRO QUEIROZ DEGER

PROJETO FINAL-BACKUP

Trabalho do setor de tecnologia da Universidade Federal do Parana
Que visa a documentação e explicação de um
script em shell com o intuito de backup
Projeto Final- DS010 ADMINISTRAÇÃO DE SISTEMAS

Orientador: Prof.Dr. Mauro Antonio Alves Castro

CURITIBA

2023

INTRODUÇÃO

Documento elaborado para a disciplina DS010 - Administração de Sistemas, do curso de Análise e Desenvolvimento de Sistemas (TADS), da Universidade Federal do Paraná (UFPR), ministrada pelo Prof. Dr. Mauro Antônio Alves Castro. O documento foi feito pelo aluno:

João Pedro Queiroz Deger

O objetivo desse documento é explicar o processo de criação e execução de backups de arquivos em sistemas Linux utilizando o terminal e um editor neste caso o bash, levando em conta a sincronização de arquivos (rsync).

EXPLICANDO O ARQUIVO(Script)

O arquivo em questão é o backup_deger.sh, que foi disponibilizado juntamente com esse documento, a explicação será de forma passo-passo.

PASSO 1: SOLICITAR ORIGEM E DESTINO DO BACKUP.

A função "FUNCAO_ORIGEM_DESTINO" é responsável por solicitar ao usuário a origem e o destino dos arquivos a serem salvos. O usuário deve fornecer os caminhos absolutos para a origem e o destino do backup, estes que são registrados respectivamente nas variáveis origem e destino1.

```
3 FUNCAO_ORIGEM_DESTINO() {
4     # Função para solicitar a origem e o destino do backup
5     echo "Digite a origem dos arquivos a serem salvos (caminho absoluto):"
6     read -r origem
7     echo "Digite o destino para o backup (caminho absoluto):"
8     read -r destino1
9     # Verifica se os diretórios de origem e destino existem
10    if [[ ! -d "$origem" || ! -d "$destino1" ]]; then
11        echo "Diretório de origem ou destino inválido."
12        return 1
13    fi
14 }
```

O script verifica se os diretórios de origem e destino existem utilizando um teste condicional "-d \$diretório" que retorna algo se o diretório existe, neste caso o número

1 que mais pra frente será utilizado. Caso algum deles não exista, será exibida a mensagem "Diretório de origem ou destino inválido." e o script será interrompido.

PASSO 2: SOLICITAR O LOCAL DO BACKUP

A função "FUNCAO_DESTINO_BACKUP" solicita ao usuário o local para o script de backup e a salva na variável destino2. O usuário deve fornecer o caminho absoluto para o destino do script que então é salvo na outra variável localsalvo. Se nenhum destino for fornecido, o script será gerado na pasta atual. Esta etapa é um pouco complicado então segue o passo a passo dos comandos utilizados:

`$(dirname "${BASH_SOURCE[0]}")`: A parte `$(dirname "${BASH_SOURCE[0]}")` obtém o diretório do script em execução. `${BASH_SOURCE[0]}` retorna o caminho do script atual, e `dirname` extrai o diretório do caminho, ou seja, remove o nome do arquivo do caminho.

A expressão `cd` altera o diretório atual para o diretório do script obtido na etapa anterior. O redirecionamento `>/dev/null 2>&1` é usado para descartar a saída do comando `cd` e redirecionar possíveis mensagens de erro para o mesmo destino. `pwd`: Após o comando `cd`, a expressão `pwd` é usada para obter o diretório atual atualizado. Isso retorna o caminho absoluto do diretório atual, que é o diretório em que o script está localizado após a alteração feita pelo `cd` na etapa anterior.

```
16 FUNCAO_DESTINO_BACKUP(){
17     # Função para solicitar o destino do script de backup
18     echo -e "Onde deseja que este backup seja gerado (o script)?"
19     echo "Deixe em branco para gerar na pasta atual."
20     read -r destino2
21     if [[ ! -z $destino2 ]]; then
22         localsalvo="$destino2/backup_script.sh"
23     else
24         localsalvo="$( cd "$( dirname "${BASH_SOURCE[0]}" )" >/dev/null 2>&1 && pwd )/backup_script.sh"
25     fi
26 }
27
```

PASSO 3: REALIZAR O BACKUP

A função `FUNCAO_BACKUP_RSYNC` realiza o backup dos arquivos utilizando o comando `rsync` com as opções `-avh --delete`.

-a: opção para preservar as permissões, proprietários, grupos e timestamps dos arquivos.

-v: opção para exibir o progresso detalhado do backup.

-h: opção para exibir o tamanho dos arquivos em um formato legível para humanos (por exemplo, "1.5M" para 1,5 megabytes).

--delete: opção para excluir arquivos no destino que não existem na origem.

Após a conclusão do backup, é exibida a mensagem "Backup concluído com sucesso!".

```
28 FUNCAO_BACKUP_RSYNC() {  
29     # Função para realizar o backup utilizando o comando rsync  
30     rsync -avh --delete "$origem/" "$destino1/"  
31     echo "Backup concluído com sucesso!"  
32 }  
33
```

PASSO 4: GERAR O SCRIPT DE BACKUP

Após o usuário fornecer a origem e o destino do backup, bem como o destino para o script de backup, o script irá gerar o arquivo de script backup_script.sh contendo o comando rsync com os parâmetros fornecidos pelo usuário.

O script define a variável localsalvo com o caminho para o script de backup. Caso o usuário tenha fornecido um destino, o valor será concatenado com /backup_script.sh. Caso contrário, o valor será gerado automaticamente na pasta atual.

```
# Execução do script  
echo "Script de Backup de Arquivos"  
  
FUNCAO_ORIGEM_DESTINO  
  
if [[ $? -eq 0 ]]; then  
    FUNCAO_DESTINO_BACKUP  
  
    echo "#!/bin/bash" >> "$localsalvo"  
    echo >> "$localsalvo"  
    echo "rsync -avh --delete \"$origem/\" \"$destino1/\" >> \"$localsalvo"  
    chmod +x "$localsalvo"  
  
    FUNCAO_BACKUP_RSYNC  
  
    echo "Script de backup gerado com sucesso em: $localsalvo"  
fi
```

O comando echo "#!/bin/bash" >> "\$localsalvo" adiciona a linha #!/bin/bash ao início do arquivo de script.

O comando echo >> "\$localsalvo" adiciona uma linha em branco no arquivo de script.

O comando echo "rsync -avh --delete \"\$origem/\" \"\$destino1/\" >> \"\$localsalvo" adiciona o comando rsync com os parâmetros fornecidos pelo usuário no arquivo de script.

O comando chmod +x "\$localsalvo" concede permissões de execução ao arquivo de script.

Após a geração do script de backup, o backup é executado utilizando o comando `FUNCAO_BACKUP_RSYNC`. Se o backup for concluído com sucesso, será exibida a mensagem "Script de backup gerado com sucesso em: [caminho do script]".

CONCLUSÃO

Este é o documento com a explicação e apresentação do arquivo `backup_deger.sh`.