

Módulo Lógica de Programação para Redes de Computadores

Professor: Marcos Brandão

Versão: 2026

APRESENTAÇÃO

Os profissionais de Redes de Computadores administram servidores, roteadores, serviços de rede, sistemas operacionais e ambientes de computação em nuvem. Embora muitas tarefas possam ser realizadas por interfaces gráficas, a automação por meio de scripts tornou-se uma habilidade essencial para administradores de redes.

Criar um script que configure usuários, realize backups, monitore serviços ou automatize atualizações economiza tempo, reduz erros e aumenta a segurança operacional.

Antes de aprender linguagens como **Bash Script** e **PowerShell**, é necessário compreender os fundamentos da lógica de programação, dos algoritmos e das estruturas de dados.

Nesta disciplina, o estudante aprenderá a construir algoritmos, desenvolver o raciocínio lógico e aplicar esses conhecimentos na automação de servidores Linux (Debian) e Windows, utilizando scripts para administração de serviços e infraestrutura de redes.

UNIDADE I

FUNDAMENTOS DA LÓGICA DE PROGRAMAÇÃO E ESTRUTURAS DE DADOS

(Páginas 1 a 5)

1. O que é Lógica de Programação?

Lógica de programação é a capacidade de organizar informações e definir uma sequência de passos para resolver um problema de forma eficiente.

Na administração de redes, diversos problemas exigem soluções lógicas, como:

- criar contas de usuários;
- verificar espaço em disco;
- reiniciar serviços automaticamente;
- monitorar servidores;
- realizar backups;
- identificar falhas de rede.

Antes de escrever um script, é necessário planejar a solução.

2. Algoritmos

Um algoritmo é uma sequência ordenada de instruções para resolver um problema.

Todo algoritmo deve possuir:

- início;
- processamento;
- resultado;
- fim.

Exemplo

Problema: verificar se um servidor está respondendo.

Algoritmo:

1. Informar o endereço IP.
2. Enviar um teste de conectividade.
3. Receber a resposta.
4. Exibir "Servidor Online" ou "Servidor Offline".
5. Encerrar.

3. Formas de Representação

Os algoritmos podem ser representados por:

- linguagem natural;
- fluxogramas;
- pseudocódigo (Portugol);
- código em linguagem de programação.

O pseudocódigo é utilizado para desenvolver a lógica antes da implementação em Bash ou PowerShell.

4. Constantes, Variáveis e Tipos de Dados

Uma variável armazena informações que podem mudar durante a execução.

Exemplo:

```
nomeServidor
ipServidor
portaSSH
```

Uma constante possui valor fixo.

Exemplo:

```
PORTA_HTTP = 80
PORTA_HTTPS = 443
```

Principais tipos de dados:

Tipo	Exemplo
Inteiro	22
Real	3.14
Texto	"Servidor"
Lógico	Verdadeiro/Falso

5. Operadores

Aritméticos

-
-
-

- /
- %

Utilizados em cálculos.

Relacionais

- ==
 - !=
 - <
 - =
-
- <=

Utilizados em decisões.

Lógicos

- AND
- OR
- NOT

Muito utilizados em scripts de administração.

Exemplo:

"O serviço está ativo **E** o servidor responde ao ping."

6. Entrada e Saída de Dados

Todo programa recebe informações e produz resultados.

Entrada:

- teclado;
- arquivos;
- parâmetros;
- rede.

Saída:

- tela;
 - arquivo;
 - banco de dados;
 - log.
-

7. Estruturas de Dados

As estruturas de dados permitem organizar informações na memória.

Arrays (Vetores)

Armazenam vários elementos do mesmo tipo.

Exemplo:

Lista de servidores.

Servidor1

Servidor2

Servidor3

Pilhas (Stacks)

Funcionam segundo o princípio:

Último que entra, primeiro que sai (LIFO).

Aplicações:

- histórico de chamadas;
 - execução de funções;
 - recursividade.
-

Filas (Queues)

Funcionam segundo:

Primeiro que entra, primeiro que sai (FIFO).

Aplicações:

- impressão;
 - processamento de pacotes;
 - filas de atendimento.
-

Listas Encadeadas

Cada elemento aponta para o próximo.

Utilizadas quando há necessidade de inserir ou remover elementos com frequência.

Árvores

Estruturas hierárquicas.

Aplicações:

- sistemas de arquivos;
 - DNS;
 - Active Directory.
-

Recursividade

Uma função chama ela mesma.

Exemplo:

Percorrer todos os diretórios de um servidor.

Atividades da Unidade I

1. Defina lógica de programação.
 2. Explique o conceito de algoritmo.
 3. Diferencie variável e constante.
 4. Cite duas aplicações de filas em redes.
 5. Explique onde árvores são utilizadas em infraestrutura de TI.
-

UNIDADE II

AUTOMAÇÃO DE SERVIDORES COM BASH SCRIPT E POWERSHELL

(Páginas 6 a 10)

1. Estruturas de Controle

Estrutura Sequencial

Comandos executados em ordem.

Estrutura Condicional

Permite tomar decisões.

Exemplo:

Se o serviço estiver parado, iniciar automaticamente.

Estruturas de Repetição

Utilizadas para repetir tarefas.

Aplicações:

- verificar diversos servidores;
 - criar vários usuários;
 - atualizar diversos arquivos.
-

2. Introdução ao Bash Script

O Bash é o interpretador de comandos padrão do Linux.

Permite automatizar tarefas administrativas.

Exemplo:

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

```
echo "Servidor iniciado."
```

Variáveis

```
NOME="ServidorWeb"
```

Entrada

```
read servidor
```

Condição

```
if [ -f backup.tar ]
```

```
then
```

```
echo "Arquivo encontrado"
```

```
fi
```

Repetição

```
for servidor in srv1 srv2 srv3
```

```
do
```

```
echo $servidor
```

```
done
```

3. Introdução ao PowerShell

O PowerShell é a linguagem de automação da Microsoft.

Muito utilizada para administrar:

- Windows Server;
 - Active Directory;
 - Hyper-V;
 - Microsoft 365.
-

Exemplo:

```
Write-Host "Servidor iniciado"
```

Variável:

```
$usuario="Administrador"
```

Repetição:

```
foreach ($pc in $computadores)
```

```
{
```

```
Write-Host $pc
```

```
}
```

4. Arquivos de Dados

Scripts frequentemente manipulam arquivos.

Exemplos:

- logs;

- backups;
- listas de usuários;
- arquivos de configuração.

No Linux:

cat
grep
awk
sed

5. Automação de Serviços

Exemplos de tarefas automatizadas:

- reiniciar Apache;
 - verificar uso de memória;
 - monitorar espaço em disco;
 - criar usuários;
 - alterar permissões;
 - copiar backups;
 - atualizar pacotes.
-

Exemplo:

```
systemctl status apache2  
systemctl restart apache2
```

6. Manipulação pela Linha de Comando

A linha de comando permite administrar servidores remotamente utilizando:

- SSH;
- PowerShell Remoting.

Vantagens:

- rapidez;
 - automação;
 - menor consumo de recursos;
 - administração remota.
-

7. Bibliotecas

As linguagens utilizam bibliotecas para ampliar funcionalidades.

Exemplos:

Bash:

- curl
- jq
- rsync

PowerShell:

- ActiveDirectory
 - NetTCPIP
 - CimCmdlets
-

Boas Práticas

- ✓ Escrever scripts simples.
 - ✓ Utilizar comentários.
 - ✓ Testar antes da produção.
 - ✓ Evitar duplicação de código.
 - ✓ Fazer tratamento de erros.
 - ✓ Registrar eventos em logs.
 - ✓ Automatizar tarefas repetitivas.
-

Resumo Geral

Ao concluir esta disciplina, o estudante deverá compreender que:

- lógica de programação é a base da automação de redes;
 - algoritmos descrevem soluções para problemas computacionais;
 - estruturas de dados organizam informações de forma eficiente;
 - comandos condicionais e de repetição controlam o fluxo dos programas;
 - Bash Script e PowerShell permitem automatizar tarefas administrativas;
 - scripts reduzem erros e aumentam a produtividade na administração de servidores;
 - a linha de comando é uma ferramenta essencial para profissionais de infraestrutura e redes.
-

Exercícios Finais

1. Explique a importância da lógica de programação para um administrador de redes.
2. Diferencie pilha e fila, apresentando um exemplo de uso em redes.
3. Escreva um algoritmo para verificar se um servidor responde ao comando de teste de conectividade (ping).
4. Explique quando utilizar uma estrutura de repetição.

- 5. Pesquise três comandos do Bash utilizados na administração de servidores Debian e descreva suas funções.
 - 6. Pesquise três cmdlets do PowerShell utilizados na administração do Windows Server.
 - 7. Desenvolva um script em Bash que:
 - o solicite o nome de um usuário;
 - o crie um diretório com esse nome;
 - o exiba uma mensagem de confirmação.
 - 8. Desenvolva um script em PowerShell que exiba os nomes dos usuários cadastrados no computador.
-

Glossário

Termo	Significado
Lógica de Programação	Processo de organizar instruções para resolver problemas computacionais.
Algoritmo	Sequência finita e ordenada de passos para atingir um objetivo.
Variável	Espaço de memória utilizado para armazenar dados que podem ser alterados durante a execução do programa.
Array (Vetor)	Estrutura que armazena diversos elementos do mesmo tipo em posições indexadas.
Pilha (Stack)	Estrutura de dados do tipo LIFO (último a entrar, primeiro a sair).
Fila (Queue)	Estrutura de dados do tipo FIFO (primeiro a entrar, primeiro a sair).
Lista Encadeada	Estrutura formada por nós conectados por referências, permitindo inserções e remoções eficientes.
Árvore	Estrutura hierárquica utilizada para representar relações entre elementos, como diretórios e registros de DNS.
Recursividade	Técnica em que uma função chama a si própria para resolver problemas repetitivos.
Bash Script	Linguagem de scripts utilizada para automação de tarefas em sistemas Linux.
PowerShell	Shell e linguagem de automação da Microsoft para administração de sistemas Windows e ambientes híbridos.
Script	Conjunto de comandos executados automaticamente para realizar uma tarefa específica.