



IMASUL MS

INSTITUTO DE MEIO AMBIENTE
DE MATO GROSSO DO SUL

Técnico em Informática
(Técnico Ambiental)

EDITAL N. 1/2024 – SAD/SEMADESC/IMASUL/
TECNICO/MEDIO

CÓD: SL-055MA-24
7908433253532

Língua Portuguesa

1. Compreensão Textual	7
2. Sílabas; Encontros Vocálicos e Consonantais; Dígrafos; Tonicidade.....	9
3. Reforma Ortográfica – 2009.....	9
4. Acentuação	11
5. Prosódia	12
6. Estrutura e Formação das Palavras	12
7. Classificação e Flexão das Palavras; Emprego de Tempos e Modos Verbais	14
8. Significação das Palavras;. Sinonímia, Antonímia, Polissemia, Emprego de Parônimos e Homônimos, Denotação e Conotação	26
9. Termos Essenciais, Integrantes e Acessórios da Oração; Vocativo.....	27
10. Crase	29
11. Pronomes: emprego, formas de tratamento, colocação.....	30
12. Pontuação	31
13. Coesão e coerência textual	33

Noções de Informática e Geoprocessamento

1. Introdução ao Geoprocessamento; Definição e importância do geoprocessamento na gestão ambiental	39
2. Histórico e evolução do geoprocessamento	39
3. Princípios básicos de cartografia e representação espacial	40
4. Conceitos e fundamentos básicos	40
5. Conhecimento e utilização dos principais softwares utilitários (compactadores de arquivos, chat, clientes de e-mails, reprodutores de vídeo, visualizadores de imagem, antivírus)	41
6. Backup de arquivos	42
7. Noções de sistema operacional: utilização do sistema operacional Windows. Identificação e manipulação de arquivos.....	42
8. Utilização dos editores de texto, planilhas e apresentações (ambientes Microsoft Office e LibreOffice)	63
9. Conceitos de tecnologias relacionadas à Internet e Intranet, busca e pesquisa na Web, mecanismos de busca na Web	97
10. Transferência de arquivos pela internet	103

Legislação Federal

1. Constituição da República Federativa do Brasil de 1988	109
2. Lei 6.938/1981 – Política Nacional de Meio Ambiente	196
3. Lei Federal 9.605/1998 - Lei de Crimes Ambientais.....	202
4. Decreto 6.514/2008 - Infrações Ambientais	209
5. Lei 12.651/2012 – Código Florestal.....	227
6. Lei Complementar 140/2011 – Competência	245
7. Lei 9.985/2000 – Sistema Nacional de Unidades de Conservação (SNUC)	249
8. Lei 9.985/2000 – Sistema Nacional de Unidades de Conservação (SNUC);	257

ÍNDICE

9. Lei nº 12.187/2009 – Política Nacional de Mudança Climática.....	260
10. Lei 9.433/1997 – Política Nacional de Recursos Hídricos.....	262
11. LGPD – Lei nº 13.709/2018.....	268

Legislação Estadual

1. Lei 90/1980 - Política Estadual.....	285
2. Lei 1.102/1990 – Estatuto do Servidor.....	290
3. Lei n. 5.287/2018 - Política Estadual de Educação Ambiental.....	316
4. Lei Estadual n. 2257/01 – Licenciamento Ambiental.....	317
5. Resolução Semade n. 9/2015 – Manual de Licenciamento Ambiental.....	319
6. Lei n. 5.673, de 8 de junho de 2021 - Dispõe sobre a Proteção à Fauna no Estado de Mato Grosso do Sul.....	334

Conhecimentos Específicos Técnico em Informática (Técnico Ambiental)

1. Sistemas operacionais (Windows, Linux, MacOS).....	339
2. Pacotes de escritório (Microsoft Office, LibreOffice).....	344
3. Internet e uso de ferramentas online.....	344
4. Suporte técnico a usuários.....	344
5. Diagnóstico e resolução de problemas de hardware e software.....	345
6. Linguagens de programação, como Python, Java, PHP, SQL.....	345
7. Scripts para automação de tarefas (Windows, Linux e MacOS).....	351
8. Redes de Computadores.....	352
9. Protocolos de comunicação.....	353
10. Configuração e manutenção de redes locais.....	353
11. Bancos de dados, como MySQL, PostgreSQL, Sql-Server.....	354
12. SQL simples.....	363

DIAGNÓSTICO E RESOLUÇÃO DE PROBLEMAS DE HARDWARE E SOFTWARE

O diagnóstico e resolução de problemas de hardware e software são atividades fundamentais no suporte técnico em TI. Aqui está uma visão geral de cada uma dessas áreas:

- Diagnóstico e resolução de problemas de hardware
- Identificação do problema: o processo começa com a identificação do problema relatado pelo usuário ou detectado por meio de monitoramento de sistemas.
- Isolamento da Causa: o técnico precisa isolar a causa raiz do problema. Isso pode envolver testes de hardware, verificação de conexões e análise de logs de erro.

— Testes e Verificação: são realizados testes nos componentes de hardware relevantes para confirmar o problema e determinar se é um defeito de hardware ou uma questão relacionada ao software.

— Substituição ou Reparo de Componentes: dependendo da natureza do problema, o técnico pode precisar substituir ou reparar componentes defeituosos, como placas de vídeo, discos rígidos, memória RAM, etc.

— Atualizações de Firmware/BIOS: em alguns casos, problemas de hardware podem ser resolvidos por meio da atualização do firmware ou BIOS dos dispositivos.

— Documentação e Registro: é importante documentar todos os passos tomados durante o diagnóstico e a resolução do problema para referência futura e para facilitar a resolução de problemas semelhantes.

— Diagnóstico e resolução de problemas de software

— Análise do Problema: o técnico analisa o problema relatado, verificando os sintomas e coletando informações relevantes, como mensagens de erro.

— Isolamento da Causa: semelhante ao diagnóstico de hardware, o técnico precisa isolar a causa raiz do problema de software. Isso pode envolver a verificação de configurações, arquivos de registro e comportamento do sistema.

— Resolução de Problemas de Software: dependendo da natureza do problema, as soluções podem variar desde a aplicação de patches e atualizações de software até a reconfiguração de configurações e a reinstalação de programas.

— Análise de Malware: se o problema estiver relacionado a malware ou vírus, será necessário executar uma varredura completa do sistema para identificar e remover as ameaças.

— Restauração de Backup: se o problema resultou na perda de dados, a restauração de backups recentes pode ser necessária para recuperar os arquivos perdidos.

— Educação do Usuário: em muitos casos, problemas de software podem ser evitados ou resolvidos mais rapidamente com a educação adequada do usuário sobre práticas seguras e eficientes de uso do sistema.

É essencial que os técnicos de suporte técnico em TI tenham habilidades de solução de problemas sólidas, conhecimento técnico abrangente e a capacidade de se comunicar efetivamente com os usuários para entender e resolver seus problemas.

LINGUAGENS DE PROGRAMAÇÃO, COMO PYTHON, JAVA, PHP, SQL

PHP

PHP é uma linguagem de script utilizada no desenvolvimento de sites, para aumentar a interatividade e usabilidade do usuário. O processamento do código PHP se dá no servidor, do qual é devolvida uma resposta ao cliente que está acessando o site. Desta forma o código PHP é mesclado ao código HTML trazendo ao usuário uma melhor experiência e acesso a funções mais específicas que vão além do HTML.

Características do PHP	
Simplicidade	PHP tem uma sintaxe simples, similar a outras linguagens populares no mercado. Pode ser utilizada em vários paradigmas tais como: Procedural e programação orientada a objeto, de acordo com o conceito do projeto.
Eficiência	Excelente desempenho com integração com diversos banco de dados, tais como: SQL Server, MySQL, MariaDB, Oracle e SQLite.
Multiplataforma	As aplicações escritas em PHP podem ser desenvolvidas e funcionar em Windows, Linux, UNIX e MacOS, além de serem hospedadas em servidores na nuvem, para serem utilizadas por diversos navegadores.
Código aberto	PHP, permite que se faça o download e inicie o trabalho sem custos para o desenvolvedor.

EXEMPLO DE CÓDIGO HTML
<html>
<head>
<title> Como funciona o PHP em uma página HTML</title>
</head>
<body>
<?php echo "<p>Vamos programar em PHP?</p>";
/* Exemplo de um parágrafo em PHP */
?>
</body>
</html>

O que é possível fazer com PHP

— Desenvolvimento de sites dinâmicos: Sites dinâmicos são aqueles em que existe uma grande interatividade entre o cliente e o servidor, PHP permite que a resposta seja devolvida, evitando atualizações do usuário para obter respostas do servidor.

— Criação de plugins: Plugins na verdade são uma extensão, que podem ser adicionadas em outro programa para conceder mais recursos. PHP se torna ideal devido às suas características.

— Desenvolvimento de aplicações para a internet: As aplicações via Internet hoje são uma realidade, podemos utilizar recursos que vão além de um site para consulta.

Áudios e vídeos

```
<áudio src="mp3"></áudio>
```

São áudios que podem ser inseridos no site. Podemos usar vários formatos disponíveis tais como: mp3, wav, ogg. No caso dos vídeos o funcionamento é o mesmo, apenas devemos nos atentar ao tamanho do arquivo. Existem outras tecnologias que mescladas ao HTML promovem mais interatividade e usabilidade ao usuário do site.

Formulários

Formulários são telas que promovem a entrada de dados por parte do usuário, como demonstra a figura abaixo:

- Nome
- Email
- Sexo
 - Masculino Feminino
- Endereço
- Número
- Cidade
- Estado
-

HTML dispõe de um conjunto de instruções que permite esta entrada de dados, vejamos no quadro abaixo:

FORMULÁRIOS	
<code><input type="text" placeholder="stuff"></code>	Input type permite uma interatividade por parte do usuário. Neste caso Placeholder é uma dica que vai aparecer ao usuário. Desta forma o usuário é informado sobre o que vai digitar.
<code><input type="text" autofocus></code>	Neste caso o autofocus posiciona o cursor no local específico da digitação, evitando o uso do mouse por parte do usuário.
<code><input type="text" autocomplete="off"></code>	Ao digitarmos um campo em um formulário, podemos habilitar o autocompletar ou não para este campo.

Um conceito importante do HTML5 é o conceito de cache de aplicativo. Desta forma quando a conexão cai o site permanece acessível, cabendo ao usuário reatualizar para restabelecer a conexão.

A linguagem Python

Python é uma linguagem de propósito geral, que pode ser usada em diversos softwares, mas é aplicada também com muita efetividade em ciência de dados, coleta de informações, além de outras aplicações. A seguir temos aqui uma breve introdução à essa linguagem.

Instalando o Python

A instalação inicial se dá de maneira simples, como a maioria dos programas de computador. Verifique a seguir:

1 — Acessando <https://www.python.org/> você fará o Download para Windows;

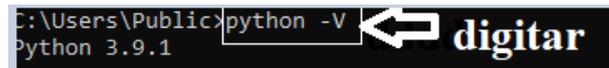


2 — A instalação se dá ao abrir o aplicativo baixado e selecionando a opção Install Python, de acordo com a versão mais recente.



Verificando a versão instalada

1 — Deve-se abrir o Command Prompt em seu computador;
 2 — Deve-se digitar `python -V` e a versão instalada será exibida, comprovando a instalação bem sucedida.



Chamando o Python pelo Prompt de comando (CMD)

Chamando o Python pelo Windows (Forma 1): digitamos Python no menu Iniciar para fazer a pesquisa;

CONHECIMENTOS ESPECÍFICOS

Operadores condicionais

Os operadores condicionais servem para refinar a pesquisa juntamente com o operador WHERE, de acordo com a tabela abaixo:

Operador Lógico	Função	Exemplo
LIKE	O operador LIKE é utilizado para procurar uma parte do texto especificado em uma coluna	<pre>select * from products where ProductName like 'C%'</pre> <p>Procura todos os produtos que iniciam com a letra C.</p>
NOT LIKE	O operador NOT LIKE faz o oposto do LIKE, isto é, ele vai procurar o que não atende ao especificado.	<pre>select * from products where ProductName not like 'C%'</pre> <p>Procura todos os produtos que não iniciam com a letra C.</p>
IN	Determina se um valor especificado corresponde em qualquer valor em uma lista.	<pre>SELECT * from categories where CategoryName in ('Condiments','Confections')</pre> <p>Procura as categorias cujo nome é 'Condiments' e 'Confections'.</p>
NOT IN	O operador NOT IN faz o oposto do IN, isto é, ele vai procurar o que não atende ao especificado.	<p>Procura as categorias cujo nome não é 'Condiments' e 'Confections'.</p>

O caractere coringa % pode ser usado para substituir qualquer cadeia de caractere, conforme exemplificado no quadro acima.

SCRIPTS PARA AUTOMAÇÃO DE TAREFAS (WINDOWS, LINUX E MACOS)

Scripts de automação são extremamente úteis para simplificar tarefas repetitivas e aumentar a eficiência no trabalho. Aqui estão exemplos de scripts para automação de tarefas em sistemas Windows, Linux e macOS:

— Scripts para automação no Windows

Batch Script (.bat):

```
``batch
@echo off
echo Bem-vindo ao Script de Automação!
REM Comentário: Este script cria um diretório e move arquivos para ele.
mkdir C:\NovaPasta
move C:\Arquivos\*.txt C:\NovaPasta
echo Tarefas concluídas!
``
```

PowerShell Script (.ps1):

```
``powershell
Write-Host "Bem-vindo ao Script de Automação!"
# Comentário: Este script lista os arquivos em uma pasta.
Get-Childitem -Path "C:\Pasta" -File
Write-Host "Tarefas concluídas!"
``
```

— Scripts para automação no Linux

Bash Script (.sh):

```
``bash
#!/bin/bash
echo "Bem-vindo ao Script de Automação!"
# Comentário: Este script lista o conteúdo de um diretório.
ls /home/usuario/pasta
echo "Tarefas concluídas!"
``
```

Shell Script para Gerenciar Pacotes (.sh):

```
``bash
#!/bin/bash
echo "Bem-vindo ao Script de Automação!"
# Comentário: Este script atualiza os pacotes do sistema.
sudo apt-get update && sudo apt-get upgrade -y
echo "Tarefas concluídas!"
``
```

— SCRIPTS PARA AUTOMAÇÃO NO MACOS:

Shell Script (.sh):

```
``bash
#!/bin/bash
echo "Bem-vindo ao Script de Automação!"
# Comentário: Este script lista os arquivos em uma pasta.
ls /Users/usuario/pasta
echo "Tarefas concluídas!"
``
```

AppleScript (.applescript):

```
``applescript
display dialog "Bem-vindo ao Script de Automação!" buttons {"OK"} default button "OK"
-- Comentário: Este script envia um email usando o Mail.
tell application "Mail"
    set theMessage to make new outgoing message with properties {subject:"Assunto", content:"Conteúdo do email", visible:true}
    tell theMessage
        make new to recipient at end of to recipients with properties {address:"destinatario@example.com"}
    end tell
    send theMessage
end tell
display dialog "Tarefas concluídas!" buttons {"OK"} default button "OK"
``
```

Os exemplos citados podem ser personalizados e expandidos para atender às necessidades específicas de automação de tarefas em diferentes sistemas operacionais.

Além disso, é importante entender os princípios básicos de programação e scripting para criar scripts mais avançados e eficientes.

Pense em um formulário de cadastro com os campos: nome, e-mail, idade e uma pergunta que admite como resposta sim ou não. O campo nome será um texto, uma sequência de letras com ou sem a presença de espaços em branco, que terá um limite máximo e não poderá conter números ou símbolos. O campo e-mail também terá o padrão textual, mas formado por uma sequência de caracteres (e não só letras, pois admitirá números e alguns símbolos) e terá que ter obrigatoriamente um arroba. Idade é um campo que aceita apenas um número inteiro positivo, enquanto o campo referente a pergunta armazena um valor binário (pense em 1 bit, que pode ser 0 ou 1. Valor 0 para não, 1 para sim). Assim, cada campo possui um padrão bem definido, que representa uma estrutura rígida e um formato previamente projetado para ele.

Os dados de um mesmo cadastro estão relacionados (dizem respeito a mesma pessoa). Em outras palavras, os dados estruturados de um mesmo bloco (registro) possuem uma relação.

Registros ou grupos de dados diferentes (como de pessoas diferentes), possuem diferentes valores, mas utilizam a mesma representação estrutural homogênea para armazenar os dados. Ou seja, possuem mesmo atributos (pense como sinônimo de campos no exemplo acima) e formatos, mas valores diferentes.

Agora, veja, banco de dados é um exemplo de dados estruturados, mas existem outros. O formulário de cadastro, mesmo que salvasse os dados em outro recurso fora banco de dados (como em um arquivo), também é um exemplo de dados estruturados por conter campos definidos por uma estrutura rígida e previamente projetada, se enquadrando na definição.

• Exemplos de dados estruturados

O exemplo mais típico de dados estruturados é um banco de dados. Nele, os dados são estruturados conforme a definição de um esquema, que define as tabelas com seus respectivos campos (ou atributos) e tipos (formato). O esquema pode ser pensado como uma meta-informação do banco de dados, ou seja, uma descrição sobre a organização dos dados que serão armazenados no banco. É exatamente como no exemplo do formulário que, normalmente, está interligado com um banco de dados.

Dados semiestruturados

Apresentam uma representação heterogênea, ou seja, possuem estrutura, mas ela é flexível. Facilita o controle por ter um pouco de estrutura, mas também permite uma maior flexibilidade.

Dados não estruturados

Qual é o oposto de uma estrutura rígida e previamente pensada? Uma estrutura flexível e dinâmica ou sem estrutura. Exemplo mais comum? Um documento ou um arquivo.

Pense em um arquivo feito em um editor de texto. Você pode adicionar quanto texto quiser, sem se preocupar com campos, restrições e limites. O arquivo pode conter também imagens, como gráficos e fotos, misturado com textos. Imagens, assim como vídeos ou arquivos de áudio, são também exemplos de dados não estruturados.

Assim, é fácil concluir que as redes sociais, as quais possuem um enorme volume de dados, como textos, imagens e vídeos criados diariamente por usuários, representam outro exemplo de dados não estruturados. Atualmente, mais de 80% do conteúdo digital gerado no mundo é do tipo não estruturado.

• Exemplos de dados não estruturados

Normalmente, basta pensar em uma situação de dados que não seguem estrutura para termos exemplos de dados não-estruturados, mas é preciso tomar um pouco de cuidado com essa análise.

Em computação, todo dado, seja ele um arquivo ou um campo rígido, terá que ter algum tipo de estrutura, mesmo que mínima. Um arquivo é um tipo de estrutura mínima, pois é a unidade básica de armazenamento de um sistema operacional, mas ela é genérica, pois aceita diferentes tipos de dados. Em resumo, quase tudo cairá em um arquivo, mesmo porque um vídeo tem que gravar em arquivo seus dados com um codificador (codec), um áudio também e assim por diante. Pensem, portanto, na estrutura interna do arquivo, se ela existe e é rígida, ou não.

Assim, possivelmente, a maior parte dos arquivos que você pensar serão não-estruturados. Vamos aos exemplos:

- Textos diversos (páginas da internet, relatórios, documentos, e-mails, mensagens em aplicativos como WhatsApp, etc.)
- Imagens (fotos, gráficos, ilustrações, desenhos, etc.)
- Arquivos de áudio (música, streaming, etc.)
- Arquivos de vídeo (filmes, seriados, feitos por usuários, etc.)
- Redes sociais (Blogs, Facebook, Twitter, Instagram, LinkedIn, etc.)

Abstração de dados

Em virtude do grande número de usuários de BD que não são treinados em computação, faz-se necessário simplificar sua estrutura para melhor interação entre usuários e sistema⁷. O grande objetivo de um sistema de BD é oferecer uma visão “abstrata” dos dados aos usuários.

O conceito de abstração está associado à característica de se observar somente os aspectos de interesse, sem se preocupar com maiores detalhes envolvidos. No contexto de abstração de dados, um BD pode ser visto sem se considerar a forma como os dados estão armazenados fisicamente.

⁷ http://www.nrsystem.com.br/Fund_Banco_Dados.pdf

CONHECIMENTOS ESPECÍFICOS

Entretanto, para que possamos implementar, de forma correta, um BD utilizando algum SGBD, temos que passar por uma fase intermediária, chamada modelagem de dados. Observe o exemplo:

CLIENTES		
CÓDIGO	NOME	DATA DE NASCIMENTO
1	Regilan Meira Silva	13/02/1983
2	Aline Araujo Freitas	27/08/1986
3	Joaquim José Pereira da Silva	12/05/1967
4	Maria Aparecida Gomes da Costa	06/01/1995

TELEFONES		
CÓDIGO	NUMERO	TIPO
1	(73)9158-9683	Celular
2	(71)3458-5112	Residencial
3	(73)8874-9681	Celular
4	(77)8841-2563	Celular

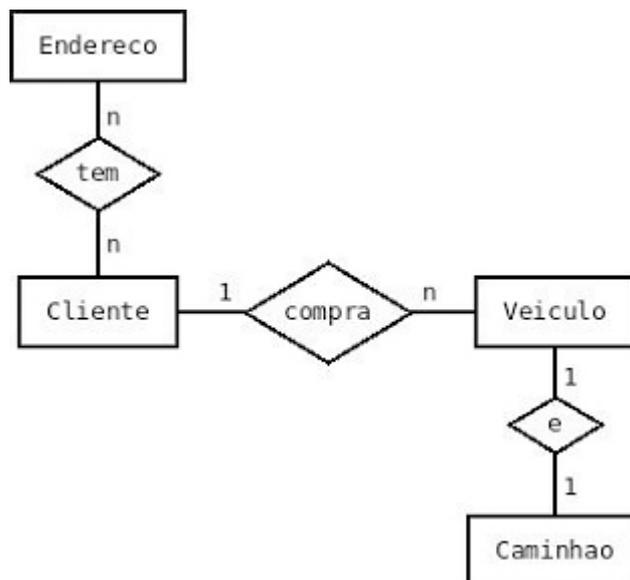
A tabela CLIENTES está relacionada com a tabela Telefones. O cliente Regilan Meira Silva possui dois telefones: um celular e um residencial. A cliente Aline Araujo Freitas possui um telefone celular, Maria Aparecida Gomes da Costa possui um celular e Joaquim José Pereira da Silva não possui telefone. Tal constatação é verificada após comparar a coluna CÓDIGO da tabela CLIENTES com a coluna CÓDIGO da tabela TELEFONES. A coluna CÓDIGO é utilizada para fazer o relacionamento entre as tabelas.

O modelo de dados mais adotado hoje em dia para representar e armazenar dados em um SGBD é o modelo relacional, onde as estruturas têm a forma de tabelas, compostas por linhas e colunas.

O Modelo Conceitual

Antes da implementação em um SGBD, precisamos de uma descrição formal da estrutura de um banco de dados, de forma independente do SGBD. Essa descrição formal é chamada modelo conceitual. Podemos comparar o modelo conceitual com o pseudocódigo/português estruturado em algoritmos, na qual construímos os algoritmos independentes de que linguagem de programação iremos desenvolver nossos programas.

O modelo conceitual é a análise dos elementos e fenômenos relevantes de uma realidade observada ou imaginada e a posterior formação de um modelo abstrato do corpo de conhecimento adquirido: o Modelo Entidade-Relacionamento ou MER¹⁰. É frequentemente documentado de forma visual em um diagrama, quando passa a ser conhecido como Diagrama Entidade-Relacionamento ou DER.



Representação do MER como um Diagrama Entidade-Relacionamento.

¹⁰ <https://medium.com/@felipeozalmeida/guia-da-modelagem-de-dados-introdu%C3%A7%C3%A3o-modelo-conceitual-238c1f8be48>

– **DQL - Linguagem de Consulta de Dados:** define o comando utilizado para que possamos consultar (SELECT) os dados armazenados no banco;

– **DML - Linguagem de Manipulação de Dados:** define os comandos utilizados para manipulação de dados no banco (INSERT, UPDATE e DELETE);

– **DDL - Linguagem de Definição de Dados:** define os comandos utilizados para criação (CREATE) de tabelas, views, índices, atualização dessas estruturas (ALTER), assim como a remoção (DROP);

– **DCL - Linguagem de Controle de Dados:** define os comandos utilizados para controlar o acesso aos dados do banco, adicionando (GRANT) e removendo (REVOKE) permissões de acesso;

– **DTL - Linguagem de Transação de Dados:** define os comandos utilizados para gerenciar as transações executadas no banco de dados, como iniciar (BEGIN) uma transação, confirmá-la (COMMIT) ou desfazê-la (ROLLBACK).

SQL SIMPLES

Prezado Candidato, o tema supracitado, já foi abordado nos tópicos anteriores

QUESTÕES

- Qual é a finalidade do DHCP em uma rede local?
 - Controlar o acesso à Internet.
 - Atribuir endereços IP automaticamente aos dispositivos na rede.
 - Fornecer segurança de rede contra ataques cibernéticos.
 - Criar uma conexão VPN para acesso remoto.
 - Gerenciar o tráfego de rede.
- Qual é a principal medida de segurança para proteger uma rede local contra acessos não autorizados?
 - Criptografia de dados
 - Uso de VLANs
 - Firewall de rede
 - Autenticação de dispositivo
 - Uso de endereços IP estáticos
- O que significa VLAN em uma rede local?
 - Virtual Local Area Network.
 - Very Large Area Network.
 - Virtual Link Access Network.
 - Versatile LAN.
 - Variable LAN.
- Qual protocolo é comumente usado para configuração dinâmica de endereços IP em uma rede local?
 - HTTP
 - DNS
 - SMTP
 - DHCP
 - FTP

5. O que é necessário para estabelecer uma conexão VPN em uma rede local?

- Servidor DNS.
- Firewall de rede.
- Roteador Wi-Fi.
- Protocolo DHCP.
- Software cliente VPN.

6. Qual conjunto de protocolos é amplamente usado para comunicação em redes IP, como a Internet?

- HTTP.
- FTP.
- SMTP.
- TCP/IP.
- DNS.

7. Qual protocolo é usado para transferência de arquivos entre um cliente e um servidor?

- HTTP.
- FTP.
- SMTP.
- TCP.
- UDP.

8. Qual protocolo é usado para enviar e-mails?

- HTTP.
- FTP.
- SMTP.
- TCP.
- DNS.

9. Qual protocolo é usado para traduzir nomes de domínio em endereços IP?

- HTTP.
- FTP.
- SMTP.
- TCP.
- DNS.

10. Qual protocolo é usado para transferência de páginas da web?

- HTTP.
- FTP.
- SMTP.
- TCP.
- UDP.

11. Qual é a extensão de arquivo comum para scripts de automação no Windows?

- .sh
- .py
- .bat
- .ps
- .exe

12. Qual linguagem de script é comumente usada para automação de tarefas em sistemas Linux?

- PowerShell
- Bash
- JavaScript