

MÓDULO INTERNET E PROGRAMAÇÃO WEB

Curso Técnico em Informática / Redes

Carga Horária: 80H

Ambiente das práticas: VPS Linux Debian + ttyd via Chromebook

Linguagens utilizadas: HTML, CSS, JavaScript, PHP, Python e MariaDB

Versão: 2026

Professor: Marcos Brandão

SUMÁRIO

I UNIDADE — FUNDAMENTOS DA WEB E FRONT-END

1. Introdução à Internet e Programação Web
 - 1.1 História da Internet
 - 1.2 Funcionamento da Web
 - 1.3 Cliente e servidor
 - 1.4 Estrutura do ambiente prático
2. HTML — Estrutura das Páginas Web
 - 2.1 Estrutura básica
 - 2.2 Tags HTML
 - 2.3 Links e imagens
 - 2.4 Formulários
3. CSS — Estilização de Páginas
 - 3.1 Seletores
 - 3.2 Cores e fontes
 - 3.3 Layout
 - 3.4 Responsividade
4. JavaScript
 - 4.1 Sintaxe básica
 - 4.2 Variáveis
 - 4.3 Funções
 - 4.4 Eventos
 - 4.5 Manipulação do DOM
5. Introdução ao PHP
 - 5.1 Conceitos

- 5.2 Variáveis
 - 5.3 Operadores
 - 5.4 Estruturas de decisão
 - 5.5 Funções
-

II UNIDADE — BACK-END E APLICAÇÕES DINÂMICAS

- 6. PHP Avançado
 - 6.1 Formulários
 - 6.2 Sessões
 - 6.3 Cookies
 - 6.4 Upload de arquivos
 - 7. Banco de Dados MariaDB
 - 7.1 Conceitos
 - 7.2 Instalação
 - 7.3 SQL básico
 - 7.4 Integração PHP + MariaDB
 - 8. Python para Web e Automação
 - 8.1 Conceitos
 - 8.2 Scripts Python
 - 8.3 Flask básico
 - 8.4 Automação no servidor
 - 9. Servidor Web Linux
 - 9.1 Apache
 - 9.2 PHP no Apache
 - 9.3 Diretórios web
 - 9.4 Hospedagem local
 - 10. Projeto Integrador
-

I UNIDADE

CAPÍTULO 1 — INTRODUÇÃO À INTERNET E PROGRAMAÇÃO WEB

1.1 História da Internet

A internet surgiu inicialmente para comunicação entre computadores militares e acadêmicos.

Com o tempo tornou-se a principal infraestrutura mundial de comunicação.

Evolução

Período	Evento
1969	ARPANET
1990	WWW
2000	Redes sociais
Atualidade	Cloud e IA

1.2 Funcionamento da Web

A web funciona no modelo:

Cliente ↔ Servidor

Exemplo

Cliente	Servidor
---------	----------

Navegador	Apache/Nginx
-----------	--------------

Fluxo de acesso

Chromebook

↓

Navegador

↓

Servidor VPS Debian

↓

Apache/PHP

1.3 Ambiente da disciplina

Toda prática será feita em:

- VPS Debian
 - acesso ttyd
 - navegador no Chromebook
-

Estrutura utilizada

Tecnologia	Função
HTML	estrutura
CSS	visual
JavaScript	interação
PHP	backend
Python	automação
MariaDB	banco de dados

PRÁTICA — Verificando servidor web

Comando

```
systemctl status apache2
```

Resultado esperado

```
active (running)
```

Explicação

Verifica se o servidor web Apache está ativo.

CAPÍTULO 2 — HTML

2.1 O que é HTML?

HTML é a linguagem usada para estruturar páginas web.

Estrutura básica

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
  <title>Minha Página</title>
```

```
</head>
```

```
<body>
```

```
  <h1>Olá Mundo</h1>
```

```
</body>
```

```
</html>
```

Explicação

Tag	Função
-----	--------

html	documento
------	-----------

head	configurações
------	---------------

body	conteúdo
------	----------

h1	título
----	--------

2.2 Criando primeira página

Diretório web

```
/var/www/html
```

Criando arquivo

```
nano /var/www/html/index.html
```

Conteúdo

```
<h1>Servidor Web Funcionando</h1>
```

```
<p>Primeira página HTML</p>
```

Acesso

`http://IP_DA_VPS`

2.3 Links

Exemplo

```
<a href="https://google.com">Google</a>
```

2.4 Imagens

Exemplo

```

```

2.5 Formulários

Exemplo

```
<form>
```

```
<input type="text">
```

```
<button>Enviar</button>
```

```
</form>
```

CAPÍTULO 3 — CSS

3.1 O que é CSS?

CSS estiliza páginas HTML.

Exemplo

```
<style>
body{
  background:black;
  color:white;
}
</style>
```

Resultado

- fundo preto
 - texto branco
-

3.2 Seletores

Seletor	Exemplo
---------	---------

tag	h1
-----	----

classe	.menu
--------	-------

id	#topo
----	-------

Exemplo prático

```
<style>
```

```
h1{
```

```
  color:blue;
```

```
}
```

```
</style>
```

3.3 Layout

Centralizando conteúdo

```
body{
```

```
  text-align:center;
```

```
}
```

3.4 Responsividade

Meta viewport

```
<meta name="viewport" content="width=device-width, initial-scale=1">
```

CAPÍTULO 4 — JAVASCRIPT

4.1 Conceito

JavaScript adiciona interatividade.

Exemplo

```
<script>  
alert("Olá");  
</script>
```

4.2 Variáveis

```
let nome = "Marcos";
```

4.3 Funções

```
function soma(a,b){  
    return a+b;  
}
```

4.4 Eventos

Botão

```
<button onclick="mensagem()">Clique</button>
```

```
<script>
```

```
function mensagem(){  
    alert("Olá");  
}
```

```
</script>
```

4.5 DOM

Manipulação da página HTML.

Exemplo

```
document.getElementById("titulo").innerHTML="Novo Texto";
```

PRÁTICA — Página interativa

Arquivo

```
nano /var/www/html/teste.html
```

Código

```
<h1 id="titulo">Olá</h1>
```

```
<button onclick="trocar()">Alterar</button>
```

```
<script>
```

```
function trocar(){
```

```
    document.getElementById("titulo").innerHTML="Texto Alterado";
```

```
}
```

```
</script>
```

Resultado esperado

Ao clicar no botão o texto muda.

CAPÍTULO 5 — INTRODUÇÃO AO PHP

5.1 O que é PHP?

PHP é uma linguagem executada no servidor.

Fluxo

Navegador → Apache → PHP → HTML

Instalando PHP

Comando

```
apt install php libapache2-mod-php -y
```

Testando PHP

Arquivo

```
nano /var/www/html/info.php
```

Código

```
<?php  
phpinfo();  
?>
```

Acesso

http://IP_DA_VPS/info.php

5.2 Variáveis PHP

```
<?php
```

```
$nome = "Marcos";
```

```
echo $nome;
```

```
?>
```

5.3 Operadores

Operador	Função
+	soma
-	subtração
*	multiplicação
/	divisão

Exemplo

```
<?php
```

```
echo 10 + 5;
```

```
?>
```

5.4 Estruturas de decisão

```
<?php
```

```
$idade = 18;
```

```
if($idade >= 18){
```

```
    echo "Maior";
```

```
}else{
```

```
    echo "Menor";
```

```
}
```

```
?>
```

II UNIDADE

CAPÍTULO 6 — PHP AVANÇADO

6.1 Formulários

HTML

```
<form method="POST" action="recebe.php">
```

```
<input type="text" name="nome">
```

```
<button>Enviar</button>
```

```
</form>
```

PHP

```
<?php
```

```
$nome = $_POST['nome'];
```

```
echo $nome;
```

```
?>
```

Explicação

Parte	Função
POST	envio
\$_POST	recebe
T	dados

6.2 Sessões

Sessões armazenam informações do usuário.

Exemplo

```
<?php
```

```
session_start();
```

```
$_SESSION["usuario"]="admin";
```

```
?>
```

6.3 Cookies

Armazenam dados no navegador.

Exemplo

```
<?php
```

```
setcookie("nome","Marcos",time()+3600);
```

```
?>
```

CAPÍTULO 7 — MARIADB

7.1 Conceito

MariaDB é um banco de dados relacional.

Instalação

```
apt install mariadb-server -y
```

Verificando status

```
systemctl status mariadb
```

Entrando no banco

```
mysql -u root -p
```

Criando banco

```
CREATE DATABASE escola;
```

Criando tabela

```
CREATE TABLE alunos(  
id INT AUTO_INCREMENT PRIMARY KEY,
```

```
nome VARCHAR(100)
```

```
);
```

Inserindo dados

```
INSERT INTO alunos(nome)
```

```
VALUES('Marcos');
```

Consultando

```
SELECT * FROM alunos;
```

7.2 PHP + MariaDB

Exemplo conexão

```
<?php
```

```
$con = mysqli_connect(
```

```
"localhost",
```

```
"root",
```

```
"senha",
```

```
"escola"
```

```
);
```

```
?>
```

Exibindo dados

```
<?php

$sql = "SELECT * FROM alunos";

$resultado = mysqli_query($con,$sql);

while($linha = mysqli_fetch_assoc($resultado)){

    echo $linha['nome'];

}

?>
```

CAPÍTULO 8 — PYTHON PARA WEB

8.1 Conceito

Python pode ser usado para:

- automação
 - APIs
 - servidores web
 - scripts
-

Verificando Python

```
python3 --version
```

Primeiro script

Arquivo

```
nano teste.py
```

Código

```
print("Olá Mundo")
```

Executando

```
python3 teste.py
```

8.2 Flask

Microframework web Python.

Instalação

```
apt install python3-flask -y
```

Aplicação simples

```
from flask import Flask
```

```
app = Flask(__name__)
```

```
@app.route('/')
```

```
def inicio():
```

```
    return "Servidor Python"
```

```
app.run(host='0.0.0.0',port=5000)
```

Executando

```
python3 app.py
```

Acesso

```
http://IP_DA_VPS:5000
```

CAPÍTULO 9 — SERVIDOR WEB LINUX

Apache

Servidor web principal da disciplina.

Instalação

```
apt install apache2 -y
```

Iniciando serviço

```
systemctl enable apache2
```

```
systemctl start apache2
```

Diretório padrão

```
/var/www/html
```

Verificando portas

```
ss -tulnp
```

Resultado esperado

```
*:80
```

CAPÍTULO 10 — PROJETO FINAL

Objetivo

Criar sistema web completo contendo:

- HTML
 - CSS
 - JavaScript
 - PHP
 - MariaDB
 - Python
-

Projeto sugerido

Sistema de cadastro de alunos.

Funcionalidades

- login
 - cadastro
 - listagem
 - banco de dados
 - painel web
-

Estrutura do projeto

/var/www/html/projeto

Etapas

1. Instalar Apache

```
apt install apache2 -y
```

2. Instalar PHP

```
apt install php libapache2-mod-php -y
```

3. Instalar MariaDB

```
apt install mariadb-server -y
```

4. Criar banco

```
CREATE DATABASE sistema;
```

5. Criar página inicial

```
nano /var/www/html/index.php
```

Resultado esperado

Os alunos terão experiência prática com:

- desenvolvimento web
 - Linux servidor
 - HTML/CSS/JS
 - PHP
 - Python
 - MariaDB
 - hospedagem web
 - aplicações dinâmicas
-

CONCLUSÃO

Ao final da disciplina o aluno será capaz de:

- desenvolver páginas web
- utilizar HTML, CSS e JavaScript
- programar em PHP
- integrar banco de dados MariaDB
- criar aplicações Python
- administrar servidor web Linux
- hospedar sistemas em VPS Debian
- trabalhar com aplicações dinâmicas reais