

MÓDULO PROGRAMAÇÃO VISUAL

Curso Técnico em Informática / Desenvolvimento Web

Carga Horária: 80H

Ambiente das práticas: VPS Linux Debian + ttyd via Chromebook

Ferramentas e tecnologias utilizadas: HTML, CSS, JavaScript, PHP, Python, Figma, GIMP, Canva, SVG, Bootstrap

Versão: 2026

Professor: Marcos Brandão

SUMÁRIO

I UNIDADE — FUNDAMENTOS DO DESIGN E INTERFACE VISUAL

1. Introdução à Programação Visual
 - 1.1 Conceito de programação visual
 - 1.2 Design aplicado a software
 - 1.3 UX e UI
 - 1.4 Estrutura visual de aplicações
2. Elementos Visuais
 - 2.1 Cores
 - 2.2 Tipografia
 - 2.3 Espaçamento
 - 2.4 Contraste
 - 2.5 Hierarquia visual
3. Componentes Visuais para Web
 - 3.1 Botões
 - 3.2 Menus
 - 3.3 Cards
 - 3.4 Formulários
 - 3.5 Ícones
4. Layout e Estrutura de Interfaces
 - 4.1 Wireframes
 - 4.2 Grid systems
 - 4.3 Responsividade
 - 4.4 Mobile First
5. HTML e CSS para Design de Interfaces
 - 5.1 Estrutura visual

- 5.2 CSS moderno
 - 5.3 Flexbox
 - 5.4 Animações básicas
-

II UNIDADE — MULTIMÍDIA E PROJETOS VISUAIS

- 6. Imagens e Recursos Gráficos
 - 6.1 Fotografia
 - 6.2 Ilustração
 - 6.3 SVG
 - 6.4 Compressão de imagens
 - 7. Áudio, Vídeo e Animação
 - 7.1 Vídeos para web
 - 7.2 Efeitos visuais
 - 7.3 Transições
 - 7.4 Motion design básico
 - 8. Ferramentas de Programação Visual
 - 8.1 Figma
 - 8.2 Canva
 - 8.3 GIMP
 - 8.4 Ferramentas web
 - 9. Criação de Interfaces para Sistemas
 - 9.1 Dashboards
 - 9.2 Painéis administrativos
 - 9.3 Interfaces modernas
 - 10. Projeto Visual Completo
-

I UNIDADE

CAPÍTULO 1 — INTRODUÇÃO À PROGRAMAÇÃO VISUAL

1.1 O que é Programação Visual?

Programação visual é a criação da parte visual de sistemas e aplicações.

Ela envolve:

- aparência
 - organização
 - experiência do usuário
 - interatividade
-

Objetivo principal

Criar interfaces:

- bonitas
 - funcionais
 - intuitivas
 - organizadas
-

Exemplos de aplicações

Sistema	Elementos Visuais
Instagram	feed, ícones

Netflix cards, banners

Sistema
escolar formulários

1.2 UX e UI

UI — User Interface

Parte visual da aplicação.

UX — User Experience

Experiência do usuário ao utilizar o sistema.

Exemplo

Situação	Problema
botão escondido	ruim UX
letras pequenas	baixa usabilidade
cores confusas	dificuldade visual

Estrutura visual de aplicações

Topo
↓
Menu
↓
Conteúdo
↓

PRÁTICA — Estrutura básica de interface

Criando diretório

```
mkdir /var/www/html/interface
```

Criando página

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

Código

```
<!DOCTYPE html>
<html>
<head>
<title>Sistema</title>
</head>

<body>

<header>
<h1>Painel Administrativo</h1>
</header>

<nav>
Menu
</nav>

<main>
Conteúdo
</main>

<footer>
Rodapé
</footer>
```

```
</body>  
</html>
```

Resultado esperado

Estrutura inicial de um sistema web.

CAPÍTULO 2 — ELEMENTOS VISUAIS

2.1 Cores

As cores influenciam:

- percepção
 - atenção
 - emoção
-

Exemplos

Cor	Sensação
Azul	confiança
Verde	segurança
Vermelho	alerta
Preto	elegância

Exemplo CSS

```
body{  
  background:#f5f5f5;  
}
```

2.2 Tipografia

Escolha das fontes.

Fontes comuns

Fonte	Uso
Arial	sistemas
Roboto	aplicações modernas
Verdan	leitura

Exemplo

```
body{  
  font-family:Arial;  
}
```

2.3 Espaçamento

Espaçamento melhora organização visual.

Exemplo

```
.card{  
  padding:20px;  
  margin:10px;  
}
```

Explicação

Propriedade	Função
padding	espaço interno
margin	espaço externo

2.4 Hierarquia Visual

Destaca informações importantes.

Exemplo

```
<h1>Título Principal</h1>  
<h3>Subtítulo</h3>  
<p>Texto</p>
```

CAPÍTULO 3 — COMPONENTES VISUAIS

3.1 Botões

Botões precisam ser claros e acessíveis.

Exemplo moderno

```
<button class="btn">  
Salvar  
</button>
```

CSS

```
.btn{  
  background:#007bff;  
  color:white;  
  border:none;  
  padding:12px 20px;  
  border-radius:8px;  
}
```

Resultado esperado

Botão moderno semelhante a sistemas profissionais.

3.2 Cards

Cards organizam conteúdo.

Exemplo

```
<div class="card">  
  
<h2>Servidor</h2>
```

```
<p>Status online</p>
```

```
</div>
```

CSS

```
.card{  
  background:white;  
  padding:20px;  
  border-radius:10px;  
  box-shadow:0 0 10px #ccc;  
}
```

3.3 Formulários

Muito utilizados em sistemas web.

Exemplo

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

Melhorando visual

```
input{  
  width:100%;  
  padding:10px;  
  border-radius:5px;  
}
```

CAPÍTULO 4 — LAYOUT E ESTRUTURA

4.1 Wireframes

Wireframe é o esboço da interface.

Exemplo

[LOGO]

[MENU]

[CONTEÚDO]

[RODAPÉ]

4.2 Grid Layout

Organiza elementos na tela.

Exemplo Flexbox

```
.container{  
  display:flex;  
  gap:20px;  
}
```

Resultado esperado

Elementos lado a lado.

4.3 Responsividade

Adaptação para celular.

Exemplo

```
@media(max-width:600px){  
  
.container{  
  flex-direction:column;  
}  
  
}
```

Explicação

No celular os elementos ficam empilhados.

CAPÍTULO 5 — HTML E CSS MODERNO

Estrutura profissional

HTML

```
<div class="dashboard">
```

```
<div class="sidebar">
```

```
Menu
```

```
</div>
```

```
<div class="content">
```

```
Conteúdo
```

```
</div>
```

```
</div>
```

CSS

```
.dashboard{  
  display:flex;  
}
```

```
.sidebar{  
  width:250px;  
  background:#222;  
  color:white;  
}
```

```
.content{  
  flex:1;  
  padding:20px;  
}
```

Resultado esperado

Layout de painel administrativo moderno.

Animações

Exemplo

```
button:hover{  
  transform:scale(1.05);  
}
```

Explicação

Botão aumenta ao passar mouse.

II UNIDADE

CAPÍTULO 6 — IMAGENS E RECURSOS VISUAIS

6.1 Imagens para web

Imagens devem ser:

- leves
 - organizadas
 - responsivas
-

Formatos

Formato	Uso
JPG	fotografias
PNG	transparência
SVG	vetorial
WEBP	otimizado

Exemplo SVG

```

```

6.2 Compressão

Reduz tamanho das imagens.

Ferramentas

- TinyPNG
 - GIMP
 - Canva
-

PRÁTICA — Organizando imagens

Criando pasta

```
mkdir /var/www/html/sistema/img
```

CAPÍTULO 7 — ÁUDIO, VÍDEO E ANIMAÇÃO

Vídeos na web

Exemplo

```
<video controls width="400">
```

```
<source src="video.mp4">
```

```
</video>
```

Áudio

```
<audio controls>
```

```
<source src="audio.mp3">
```

```
</audio>
```

Transições CSS

```
.card{  
  transition:0.3s;  
}
```

Efeito hover

```
.card:hover{  
  transform:translateY(-5px);  
}
```

Resultado esperado

Card sobe suavemente.

CAPÍTULO 8 — FERRAMENTAS DE PROGRAMAÇÃO VISUAL

8.1 Figma

Ferramenta para prototipação.

Usos

- wireframes
 - telas
 - protótipos
-

8.2 Canva

Criação rápida de elementos visuais.

8.3 GIMP

Editor de imagens open source.

Fluxo profissional

Protótipo → Design → Código HTML/CSS

CAPÍTULO 9 — INTERFACES PARA SISTEMAS

Dashboard

Painel administrativo.

Elementos comuns

- menu lateral
 - cards
 - gráficos
 - tabelas
-

Exemplo HTML

```
<div class="menu">
```

```
<a href="#">Dashboard</a>
```

```
<a href="#">Usuários</a>
```

```
</div>
```

Exemplo CSS

```
.menu{
  width:250px;
  background:#111;
  height:100vh;
}
```

Tabelas modernas

```
<table>
```

```
<tr>
<th>Usuário</th>
<th>Status</th>
</tr>
```

```
<tr>
<td>Marcos</td>
<td>Online</td>
</tr>
```

```
</table>
```

Melhorando visual

```
table{
  width:100%;
  border-collapse:collapse;
}
```

CAPÍTULO 10 — PROJETO FINAL

Objetivo

Criar interface completa de sistema web.

Projeto sugerido

Sistema de monitoramento da VPS.

Funcionalidades

- painel administrativo
 - cards de CPU/RAM
 - usuários online
 - menu lateral
 - gráficos
 - layout responsivo
-

Estrutura do projeto

/var/www/html/dashboard

Etapas

1. Criar estrutura

mkdir css js img

2. Criar HTML

nano index.html

3. Criar CSS

nano css/style.css

4. Criar JavaScript

nano js/script.js

Resultado esperado

Sistema visual semelhante a aplicações modernas.

Tecnologias utilizadas

Tecnologia	Função
HTML	estrutura
CSS	design
JavaScript	interação
PHP	backend
Python	integração
VPS Debian	hospedagem

EXEMPLO DE PROJETO VISUAL COMPLETO

HTML

```
<div class="card">  
  
<h2>CPU</h2>  
  
<p>15%</p>  
  
</div>
```

CSS

```
.card{  
  background:#1f2937;  
  color:white;  
  padding:20px;  
  border-radius:10px;  
}
```

Resultado

Card moderno estilo dashboard profissional.

CONCLUSÃO

Ao final da disciplina o aluno será capaz de:

- desenvolver interfaces modernas
- aplicar conceitos de UX/UI
- criar layouts responsivos
- utilizar HTML e CSS profissionalmente
- trabalhar com multimídia
- criar dashboards e sistemas visuais
- aplicar animações e efeitos
- desenvolver design para aplicações web hospedadas em VPS Debian