# **Exercício Prático: Layout de Dashboard**

## 

## **Objetivo**

Criar uma página de dashboard que incorpore diferentes tipos de display e técnicas de posicionamento CSS.

## 

## **Tempo Estimado**

90 minutos

## 

## **Conceitos Abordados**

* Display block, inline e inline-block
* Position static, relative, absolute e fixed
* Flexbox
* Layout responsivo básico

## 

## **Requisitos do Projeto**

### **1. Estrutura Base HTML (15 minutos)**

<!DOCTYPE html>

<html lang="pt-BR">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Dashboard</title>

<link rel="stylesheet" href="styles.css">

</head>

<body>

<!-- Cabeçalho Fixo -->

<header>

<nav>

<!-- Menu de Navegação -->

</nav>

<!-- Ícone de Notificações -->

</header>

<!-- Menu Lateral -->

<aside>

<!-- Itens do Menu -->

</aside>

<!-- Conteúdo Principal -->

<main>

<!-- Cards de Estatísticas -->

<!-- Lista de Atividades Recentes -->

</main>

<!-- Rodapé -->

<footer>

<!-- Informações do Rodapé -->

</footer>

</body>

</html>

### 

### 

### **2. Elementos Requeridos**

## **Header (Cabeçalho Fixo)**

* Deve permanecer fixo no topo da página
* Menu de navegação com itens inline-block
* Ícone de notificações posicionado absolutamente no canto direito
* Badge numérico sobre o ícone de notificações

## **Menu Lateral**

* Fixo na lateral esquerda
* Lista de itens de menu empilhados verticalmente
* Ícones à esquerda do texto de cada item

## **Conteúdo Principal**

* Cards de estatísticas usando flexbox (4 cards por linha)
* Lista de atividades recentes com ícones de status
* Espaçamento adequado entre elementos

## **Rodapé**

* **Fixo** na parte inferior
* Links de navegação **inline**
* Texto de copyright **centralizado**

### 

### **3. Requisitos de Estilo**

## 

## **A. Layout Geral**

/\* Exemplo de estrutura base \*/

body {

margin: 0;

padding: 0;

min-height: 100vh;

}

header {

position: fixed;

top: 0;

width: 100%;

z-index: 100;

}

aside {

position: fixed;

left: 0;

height: 100%;

width: 250px;

}

main {

margin-left: 250px;

margin-top: 60px;

padding: 20px;

}

## 

## **B. Elementos Específicos**

1. Cards de Estatísticas:
   * Usar flexbox para organização
   * Dimensões fixas
   * Sombras e bordas arredondadas
2. Ícones de Status:
   * Posicionamento relative/absolute para badges
   * Cores diferentes para diferentes estados
3. Menu de Navegação:
   * Itens inline-block
   * Hover effects
   * Espaçamento consistente

### 

### **4. Passos de Desenvolvimento**

1. **Estrutura Base (15 minutos)**
   * Criar HTML básico
   * Definir estrutura de arquivos CSS
2. **Header e Navegação (20 minutos)**
   * Implementar cabeçalho fixo
   * Criar menu de navegação
   * Adicionar ícone de notificações com badge
3. **Menu Lateral (15 minutos)**
   * Criar estrutura do menu lateral
   * Estilizar itens do menu
   * Implementar ícones
4. **Área de Conteúdo (25 minutos)**
   * Criar cards de estatísticas com flexbox
   * Implementar lista de atividades
   * Adicionar ícones de status
5. **Rodapé (15 minutos)**
   * Implementar rodapé fixo
   * Organizar links e copyright

### 

### **5. Requisitos Extras (Bônus)**

1. Responsividade:
   * Menu lateral colapsável em telas menores
   * Cards reorganizados em menos colunas
   * Ajustes de fonte e espaçamento
2. Interatividade:
   * Hover effects nos cards
   * Menu dropdown no header
   * Tooltips nos ícones

### 

### **6. Critérios de Avaliação**

1. **Uso Correto de Display (25%)**
   * Implementação apropriada de block, inline e inline-block
   * Uso efetivo de flexbox
   * Organização lógica dos elementos
2. **Posicionamento (25%)**
   * Uso correto de position fixed, relative e absolute
   * Alinhamento adequado dos elementos
   * Sobreposições corretas (z-index)
3. **Estrutura e Organização (25%)**
   * HTML semântico
   * CSS bem organizado
   * Nomeação clara de classes
4. **Visual e Funcionalidade (25%)**
   * Aparência profissional
   * Funcionalidade em diferentes tamanhos de tela
   * Atenção aos detalhes

### 

### **7. Dicas de Desenvolvimento**

1. **Organização**
   * Comece com a estrutura HTML completa
   * Estilize um componente por vez
   * Teste frequentemente em diferentes tamanhos de tela
2. **Boas Práticas**
   * Use variáveis CSS para cores e valores comuns
   * Mantenha o CSS organizado por seções
   * Comente seções importantes do código
3. **Depuração**
   * Use as ferramentas de desenvolvedor do navegador
   * Verifique o comportamento de overflow
   * Teste a interação entre elementos posicionados

### 

### **8. Entrega**

1. **Arquivos Requeridos**
   * index.html
   * styles.css
   * README.md com instruções e comentários
2. **Formato**
   * Código formatado e comentado
   * Commits organizados e descritivos
   * Deploy em GitHub Pages (opcional)
3. **Prazo**
   * Data de entrega: 29/10