



**SENAI**





# PROGRAMAÇÃO FRONT END II

---



# FRAMEWORK CSS

## DEFINIÇÃO



Um conjunto de código pronto (biblioteca) que fornece estilos, classes e componentes reutilizáveis para facilitar e acelerar o desenvolvimento de interfaces web.

Em vez de escrever todo o CSS manualmente, você pode usar um framework para aplicar estilos de forma rápida e consistente.

Reset/normalize → ajusta estilos básicos para que fiquem iguais em todos os navegadores.

- Sistema de grid → facilita a criação de layouts responsivos.
- Componentes prontos → botões, menus, tabelas, cards, modais etc.
- Utilitários → classes rápidas para margens, cores, fontes, etc.
- Responsividade → já vêm preparado para funcionar em qualquer tela

# TAILWIND CSS

## DEFINIÇÃO



Framework CSS **utility-first** — cada classe faz apenas **uma função**.

Em vez de criar classes como `.btn-vermelho`, você usa classes utilitárias

```
<button class="bg-red-500 text-white px-4 py-2 rounded">
```

Comprar

```
</button>
```

# TAILWIND CSS

## VERSÃO 4



- Novo motor “Oxide”: reescrito em Rust + Lightning CSS, entrega builds até 5× mais rápidos e rebuilds incrementais até 100× mais rápidos
- Instalação simplificada: só um `@import "tailwindcss"` — não precisa mais das diretrizes `@tailwind base/components/utilities`
- Configuração zero-config e detection automática de conteúdo
- Configuração “CSS-first”, com `@theme { ... }` dentro do próprio CSS, substituindo `tailwind.config.js`

# TAILWIND CSS

## CONSIDERAÇÕES



	Bootstrap	Tailwind CSS
Estilo	Framework de <b>componentes prontos</b> (botões, cards, modais).	Framework <b>utility-first</b> (classes atômicas para montar tudo do zero).
Abordagem	“Coloque o HTML, já fica bonito.”	“Você estiliza cada elemento com utilitários no HTML.”
Dependências	CSS + JS (para alguns componentes).	Apenas CSS, sem dependência de JS para estilos.
Responsividade	Sistema de grid fixo baseado em breakpoints pré-definidos.	Breakpoints móveis, flexíveis e personalizáveis via classes.



# TAILWIND CSS

## CONSIDERAÇÕES



	Bootstrap	Tailwind CSS
Início rápido	Mais rápido para iniciante → basta usar componentes prontos.	Precisa entender bem o HTML e utilitários para compor o layout.
Customização	Mais difícil alterar sem sobrescrever CSS existente.	Altamente customizável desde o início.
Complexidade com o tempo	Fica repetitivo se quiser fugir do “padrão Bootstrap”.	Pode gerar HTML com muitas classes ( <i>class soup</i> ).

# TAILWIND CSS CONSIDERAÇÕES



## Bootstrap

- Sites institucionais rápidos, protótipos, CRUDs administrativos, MVPs.
- Equipes que não querem criar design do zero.
- Desenvolvedores com pouca experiência em CSS.

## Tailwind CSS

- Projetos com identidade visual única, aplicativos complexos, dashboards modernos.
- Equipes que querem total controle do visual.
- Desenvolvedores confortáveis com HTML e responsividade.



# TAILWIND CSS

## EXTENÇÕES PARA VSCODE



- Tailwind CSS IntelliSense
  - É a extensão oficial da equipe do Tailwind.
  - Preview da cor.
  - Validação: avisa se a classe não existe no Tailwind.
  - Config-aware: reconhece suas customizações.
- Tailwind Snippets
  - Fornece atalhos de código prontos (snippets) para Tailwind no HTML, JSX, TSX etc.
- Tailwind Fold
  - Permite “dobrar/colapsar” (fold) classes longas do Tailwind em uma linha mais curta.

# TAILWIND CSS

## PADDING & MARGING



**p** = padding

**m** = margin

- Sufixos:
  - 1 a 99 - cada número é um multiplicador de 0.25rem (4px)
- Direções:
  - x - no eixo x
  - y - no eixo y
  - s - esquerda (start)
  - e - direita (end)
  - t - topo (top)
  - b - em baixo (bottom)

# TAILWIND CSS

## TIPOGRAFIA



Tamanho:

- text-xs, text-sm, text-base, text-lg, text-2xl, etc.

Peso:

- font-light, font-normal, font-semibold, font-bold

Cor:

- text-gray-700, text-blue-500

Alinhamento:

- text-left, text-center, text-right



# TAILWIND CSS

## CORES



Tailwind tem paleta pré-definida (gray, red, etc., com variações 50 a 950):

```
<div class="bg-blue-500 text-white">Azul</div>
```

```
<div class="bg-green-200 text-green-800">Verde claro</div>
```

- Transparência com `{0-100}`
- Cores personalizadas com variáveis ou utilizando `-[#0000]`

# TAILWIND CSS

## VARIÁVEIS



Tailwind permite a criação e utilização de cores via variáveis alterando o arquivo de estilos principal, exemplo:

```
@theme {  
  --color-brand: #ff5722;  
}
```

-----

```
<div class="bg-[var(--color-brand)]">Cor customizada</div>
```

# TAILWIND CSS

## LARGURA E ALTURA



### - Largura:

- w-4 (1rem),
- w-1/2,
- w-full,
- max-w-xl

### - Altura:

- h-8,
- h-screen,
- min-h-full



# TAILWIND CSS

## FLEX



### Ativando Flex

- flex → ativa o flex container
- inline-flex → flex em linha

### 2. Direção (flex-direction)

- flex-row → elementos na horizontal (padrão), '-reverse' para inverte
- flex-col → elementos na vertical, '-reverse' para inverter

### 3. Alinhamento no eixo principal (justify-content)

- justify-start → início
- justify-end → fim
- justify-center → centro
- justify-between → espaçamento entre elementos

# TAILWIND CSS

## FLEX



### 4. Alinhamento no eixo cruzado (align-items)

- items-start → no topo/esquerda
- items-end → embaixo/direita
- items-center → centralizados

### 5. Controle dos itens

- flex-1 → ocupa todo espaço disponível
- flex-none → não cresce nem encolhe
- flex-auto → cresce automaticamente
- grow / grow-0 → permite ou não expandir
- shrink / shrink-0 → permite ou não encolher
- order-{n} → muda a ordem dos itens

# TAILWIND CSS

## FLEX



```
<div class="flex items-center justify-between">  
  <span>Logo</span>  
  <nav class="flex gap-4">  
    <a href="#">Home</a>  
    <a href="#">Sobre</a>  
  </nav>  
</div>
```



# TAILWIND CSS

## GRID



### Ativando Grid

- `grid` → ativa o grid container
- `inline-grid` → grid em linha

### 2. Definindo colunas e linhas

- `grid-cols-{n}` → define número de colunas (ex: `grid-cols-3`)
- `grid-rows-{n}` → define número de linhas
- `grid-cols-none` → remove configuração de colunas

### 3. Espaçamento (gap)

- `gap-{n}` → espaço entre linhas e colunas
- `gap-x-{n}` → espaço só nas colunas
- `gap-y-{n}` → espaço só nas linhas

# TAILWIND CSS

## GRID



### Posicionamento dos itens

- `col-span-{n}` → item ocupa várias colunas
- `row-span-{n}` → item ocupa várias linhas
- `col-start-{n}` / `col-end-{n}` → posição inicial/final da coluna
- `row-start-{n}` / `row-end-{n}` → posição inicial/final da linha

### 5. Alinhamento

- `justify-items-{start|end|center|stretch}` → eixo horizontal
- `items-{start|end|center|stretch}` → eixo vertical
- `justify-content-{start|end|center|between|around|evenly}` → distribui colunas/linhas
- `content-{start|end|center|between|around|evenly}` → distribui conteúdo

# TAILWIND CSS

## GRID



```
<div class="grid grid-cols-3 gap-4">  
  <div class="bg-gray-200">1</div>  
  <div class="bg-gray-200">2</div>  
  <div class="bg-gray-200">3</div>  
</div>
```



# TAILWIND CSS

## FLEX & GRID



Recomendação:

- Use Flex para alinhamento e distribuição linear (uma dimensão).
- Use Grid para layouts estruturados em linhas e colunas (duas dimensões).

# TAILWIND CSS

## USO DOS COLCHETES



Permite valores livres não definidos no tema:

```
<div class="bg-[#1da1f2] text-[18px] p-[10px]">
```

Twitter

```
</div>
```

- Pode ser cor, tamanho, font, espaçamento, etc
- Útil para medidas exatas ou cores específicas.

# TAILWIND CSS

## RESPONSIVIDADE



Tailwind segue mobile-first:

Prefixos:

- sm:  $\geq 640\text{px}$
- md:  $\geq 768\text{px}$
- lg:  $\geq 1024\text{px}$
- xl:  $\geq 1280\text{px}$
- 2xl:  $\geq 1536\text{px}$

```
<div class="text-sm md:text-lg lg:text-2xl">
```

Texto que aumenta conforme tela

```
</div>
```

# TAILWIND CSS

## ESTADOS



Tailwind segue mobile-first:

Prefixos:

- hover: — quando o mouse passa
- focus: — quando o elemento recebe foco
- active: — enquanto clicado
- disabled: — elemento desativado

```
<button class="bg-blue-500 hover:bg-blue-700 active:scale-95">
```

Clique aqui

```
</button>
```

# TAILWIND CSS

## CRIANDO VARIÁVEIS



```
@theme {  
  /* Definição de cores */  
  --color-primary: #0ea5e9;  
  --color-secondary: #9333ea;  
  
  /* Espaçamentos */  
  --spacing-container: 2rem; /* Usado pelo container */  
}
```

Tudo que é definido aqui pode ser usado nas classes do Tailwind com `var(--...)` e `[]`.



# TAILWIND CSS

## EDITANDO VARIÁVEIS PADRÃO

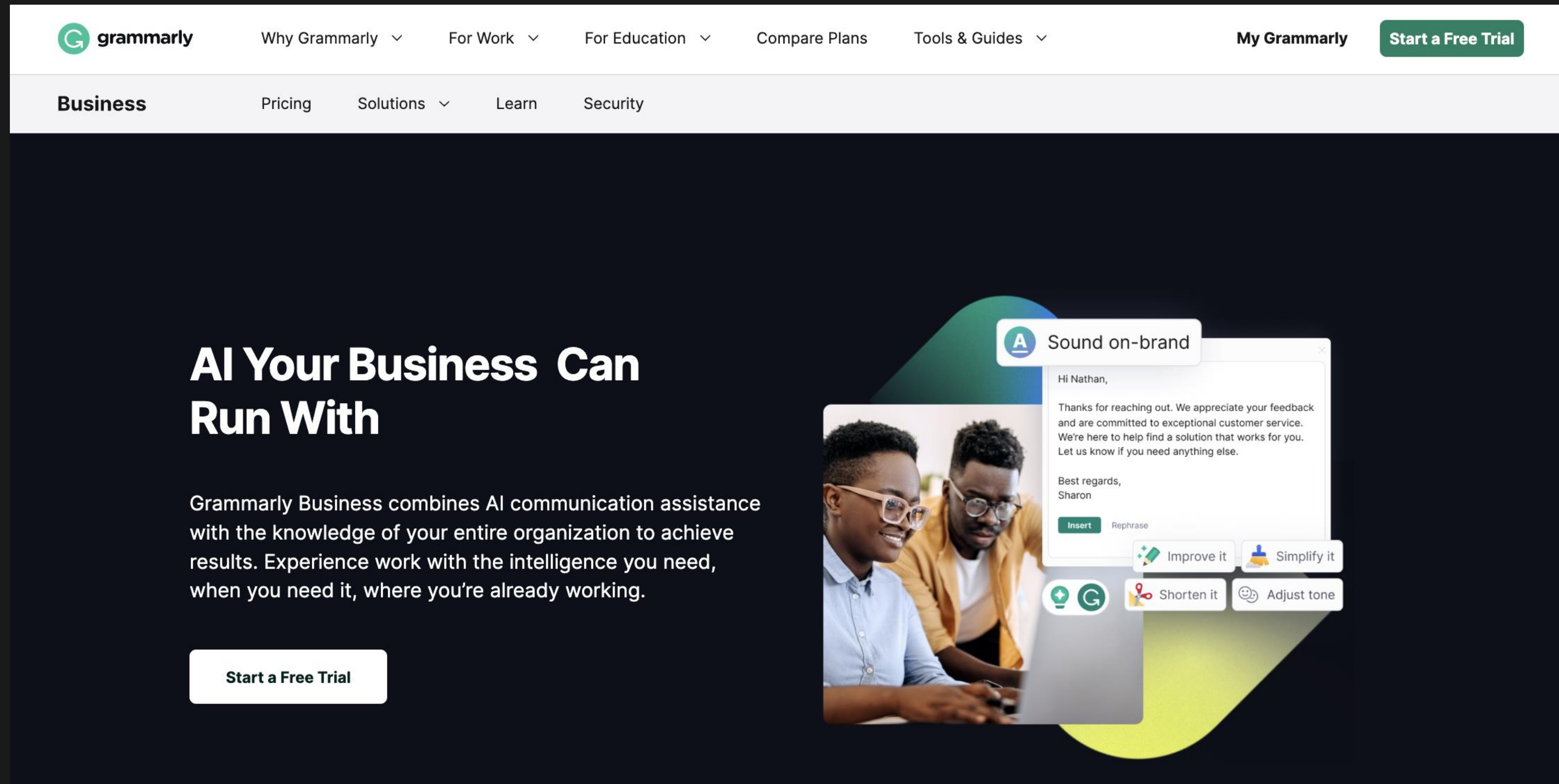


O `container` no Tailwind é uma classe utilitária para centralizar conteúdo. No v4, você pode sobrescrever o espaçamento máximo dele via variável:

```
@theme {  
  --container-max: 1200px; /* largura máxima */  
  --container-padding: 2rem; /* padding interno */  
}
```

```
<div class="container bg-gray-100">  
  <p>Conteúdo centralizado com largura e padding customizados</p>  
</div>
```

# TAILWIND CSS EXERCÍCIO





A nighttime photograph of a city street, likely in São Paulo, featuring tall buildings and light trails from moving vehicles. The image is dark, with the primary light sources being the city lights and the long-exposure light trails. A red banner with the SENAI logo is positioned on the left side of the image.

**SENAI**

DEPARTAMENTO REGIONAL  
DE SÃO PAULO

[www.sp.senai.br](http://www.sp.senai.br)