**CENTRO ESTADUAL DE EDUCAÇÃO TECNOLÓGICA “PAULA SOUZA” FACULDADE DE TECNOLOGIA DE TAQUARITINGA**

**CURSO SUPERIOR DE TECNOLOGIA EM SISTEMAS PARA INTERNET.**

**UTILIZANDO O FRAMEWORK TAILWIND CSS**

**ORIENTADOR: Prof. Carlos Pereira de Castro Filho**

**Taquaritinga - SP**

**2022**

**Yan Mantovani da Rocha.**

**UTILIZANDO O FRAMEWORK TAILWIND CSS**

**Trabalho de pesquisa apresentado ao curso de Sistemas para Internet da Fatec Taquaritinga orientado pelo Prof. Carlos Pereira de Castro Filhocomo requisito parcial para a entrega de atividade.**

**Taquaritinga - SP**

**2022**

Dedico esse trabalho primeiramente a Deus, nossos pais e professores que nos auxiliou nesta jornada de conhecimento.

**Agradecimentos**

Agradeço ao meu orientador, Prof. Pereira de Castro Filho e a todos os professores do curso que tiveram extrema importância nessa trajetória.

Ao incentivo vindo de nossos familiares.

A disposição da secretaria.

A todos, meu muito obrigado!

“Nunca alguém tão grande se fez tão pequeno para tornar grandes os pequenos (CURY, 2012)”

**RESUMO**

Quando o CSS foi introduzido no desenvolvimento de páginas web, ele causou uma ruptura de paradigma, finalmente permitindo separar estrutura de apresentação. A cada nova versão do recurso, o CSS ganhou mais funcionalidades, animações e variáveis. Com isso, surgiu o Tailwind CSS, que é um framework desenvolvido para maximizar o potencial do CSS. Essa nova ferramenta oferece responsividade, códigos otimizados, customizações e integração com IDES. Da mesma forma que os frameworks de JavaScript mudaram a forma de interagir com o JavaScript, Tailwind pode ser o futuro do CSS.  
  
  
**Palavras-chave:**Tailwind; CSS; Framework; Javascript.

**Abstract**

When CSS was introduced in web page development, it caused a paradigm shift, finally allowing separate structure from presentation. With each new version of the feature, the CSS gained more functionality, animations and variables. With that came Tailwind CSS, which is a framework designed to maximize the potential of CSS. This new tool offers responsiveness, optimized codes, customizations and integration with IDES. In the same way that JavaScript frameworks have changed the way we interact with JavaScript, Tailwind could be the future of CSS.

**Keywords:** Tailwind; CSS; Framework; Javascript.

**SUMÁRIO**

[**INTRODUÇÃO** 7](#_Toc114688922)

[**DESENVOLVIMENTO** 8](#_Toc114688923)

[**Facilidade** 9](#_Toc114688924)

[**Redução de códigos** 9](#_Toc114688925)

[**Design responsivo** 10](#_Toc114688926)

[**Colocar os estilos inline reduz a nomenclatura** 10](#_Toc114688927)

[**Metodologia** 11](#_Toc114688928)

[**Discussão de resultados** 12](#_Toc114688929)

[**REFERÊNCIAS** 12](#_Toc114688930)

# **INTRODUÇÃO**

O Tailwind permite um maior controle na estilização da página. O Tailwind também consegue trabalhar com diversos plugins e ferramentas para facilitar o desenvolvimento, além de poder usar componentes criados por outras pessoas.

Como o Tailwind é um framework que usa classes de utilidades, os elementos podem ficar lotados de classes, sem contar a repetição e a manutenção para vários componentes ao mesmo tempo.

Por isso, o Tailwind permite que você crie classes customizadas que recebem as classes de utilidade, e depois basta chamas as classes criadas que os componentes vão receber todas as propriedades escolhidas.

E além disso, Tailwind permite maior controle na linguagem visual e é a solução mais leve por usar apenas as classes necessárias para desenvolver a página. Caso precise de outras ferramentas e plugins, você pode instalar por opção e não por obrigação. Por outro lado, ele requer um conhecimento maior das propriedades CSS e seus nomes de classes exigem uma curva de aprendizado maior comparado com Bootstrap.

**Objetivo Geral**

O objetivo dessa pesquisa é testar e analisar a usabilidade do framework Tailwind CSS.

**Objetivo específico**

Desenvolver uma página de login responsiva afim de aprender a como utilizar o framework Tailwind CSS.

# **DESENVOLVIMENTO**

Assim como o Bootstrap, Tailwind é um framework CSS que oferece a possibilidade de você criar layouts usando uma estrutura de CSS pronta. Isso permite que você otimize o tempo de criação de uma UI sem precisar fazer tudo novamente. Adotar essa estratégia tem seus prós e contras, como sempre se fala, nenhum framework, biblioteca ou tecnologia deve ser tratado como bala de prata. Qualquer um deles deve ser bem avaliado, são apenas meios para otimizar o desenvolvimento.

## **Facilidade**

Shawn Swyx Wang é um desenvolvedor ativo na comunidade de desenvolvimento web como blogueiro e palestrante frequente. Em um artigo publicado em um de seus blogs, ele explica que Tailwind CSS é mais fácil de aprender do que parece e desmistifica alguns preconceitos contra o framework.

Parafraseando Shawn, ele diz o seguinte em seu artigo: “Não sou um devoto do Tailwind, eu sou alguém que mudou de ideia recentemente e um usuário feliz, apesar dos sacrifícios reconhecidos. “Pessoas que mudaram de lado” muitas vezes podem ser mais persuasivas para os céticos do que os crentes nascidos e criados naquilo”. Um internauta o questionou: “Você achava que o Tailwind era uma ideia horrível até que realmente construiu algo com ele?”. E a resposta de Shawn foi a seguinte: “Certa vez, reclamei com criador do Tailwind que ele causou uma sopa de nomes de classe ilegível e disse que o CSS-in-JS de tempo de execução zero poderia fazer mais com uma curva de aprendizado menor. Eu estava errado em 2 pontos: o Tailwind é mais fácil de aprender do que eu pensava e a flexibilidade do CSSinJS pode ser negativa.

Essa discussão que Shawn teve, pode nos passar a impressão de sensatez e uma oportunidade de aprendermos a como utilizar esse framework. Depois de testes, pude confirmar isso. A facilidade de encurtar ainda mais os códigos e já os deixando responsivos, faz com que o futuro do CSS esteja dentro dos vários frameworks que existem e ainda vão ser criados.

## **Redução de códigos**

CSS é extremamente flexível, o que o torna poderoso, mas também oferece muitas chances de o projeto sair errado. Restrições são necessárias, ou então espalhará números aleatórios por dentro da base do código.

Números mágicos em CSS são um coisa ruim, diz Chris Coyier. Ele os define como “valores que” funcionam “em algumas circunstâncias, mas são frágeis e propensos a quebrar quando essas circunstâncias mudam”, mas honestamente qualquer número fixo em hard code, como pixel nas margens e media-queries ou variantes de cor, é difícil de gerenciar bem em escala. A tentação de quebrar as regras apenas para enviar uma correção existe, e a dificuldade de refatoração quando os requisitos de design mudam é muito alta. Se você está trabalhando sozinho, não há nada que o obrigue a se limitar a um conjunto consistente de escalas numéricas bem projetadas, o que pode levar a um projeto de aparência ruim.

A solução, claro, é extrair apenas de um intervalo predefinido de valores numéricos, que chamo de “sistema”. O Tailwind vem com um bom conjunto de fontes e sistemas de cores por padrão. Mais uma vez, você poderia tentar lançar seu próprio sistema com variáveis ​​CSS, mas a sintaxe para isso é prolixa e você ainda tem que criar nomes para classes e variáveis, e você acaba com um sistema personalizado que não se transfere entre projetos e provavelmente não está bem documentado.

## **Design responsivo**

Este ponto é mais relevante para desenvolvedores que fazem o design: o melhor fluxo de trabalho de desenvolvimento é visualizar seu site no host local, fazer ajustes no navegador até que você esteja satisfeito com isso e, em seguida, copiar e colar suas alterações diretamente em seu código. Vamos chamar isso de fluxo de trabalho Design in Browser.

Se quiser esse fluxo de trabalho, você exclui o uso do estilo embutido do React (que faz com que você use a sintaxe de objeto). Mas digamos que você use alguma forma de solução “Write Real CSS” ™, como Styled-Components ou Vue ou Svelte, onde o design no navegador é possível.

## **Colocar os estilos inline reduz a nomenclatura**

Nomear Coisas é um problema difícil conhecido. Perdemos muito tempo discutindo sobre nomes de classes. Com Styled-Components, você geralmente escreve um monte de componentes com estilo intermediário que você precisa nomear. Com o BEM, substituímos um problema de nomenclatura por três problemas de nomenclatura e meio. Quantos milhões em horas de desenvolvedor desperdiçamos todos os anos discutindo sobre nomes?

Com o CSS utilitário, reduzimos significativamente o número total de nomes em nossa base de código e, talvez mais importante, o número de nomes que temos que inventar e lembrar de forma independente. Isso parece pequeno até que você trabalhe em uma base de código onde isso não foi usado. Que preço você está disposto a pagar para eliminar um dos problemas mais difíceis conhecidos? Os nomes não importam para as máquinas, mas importam para os humanos.

A desvantagem é que você precisa aprender os nomes da estrutura do CSS utilitário. -Mb-5 e space-x-reverse não são identificáveis ​​sem documentação. A diferença é que a nomenclatura CSS tradicional é feita sob medida por projeto, enquanto você aprende o Tailwind uma vez e pode usá-los em todos os projetos. Sim, você pode tentar lançar seus próprios utilitários, mas a nomenclatura do Tailwind é provavelmente projetada de forma mais cuidadosa do que qualquer coisa que você venha a criar.

## **Metodologia**

O início do estudo foi executado com a formação teórica sobre o tema, foi utilizado o site SciELO demais artigos para a realização da pesquisa, tendo em vista que já havia certo conhecimento sobre o tema abordado.

A pesquisa foi enviada com o objetivo de entrega de atividade, sendo divulgado nas tarefas de um aplicativo chamado Teams.

Consecutivamente, foi analisado os resultados e copilado os dados da pesquisa com base nas respostas obtidas.

Em seguida, foi criado uma página utilizando o framework Tailwind para estar comprovando as teorias relacionadas à documentação.

Por último, fez-se então a documentação para apresentar sobre as utilidades do Tailwind e suas partes benéficas.

## **Discussão de resultados**

Tailwind CSS é o novo framework do mercado que provavelmente veio para elevar o CSS ao limite. Sua capacidade de diminuir o tamanho dos códigos e já deixar as páginas responsivas devem ser exploradas ao máximo pelos desenvolvedores.

## **REFERÊNCIAS**

Max Stoiber: Why I Write CSS in JavaScript. Disponível em: <https://mxstbr.com/thoughts/css-in-js>

Johan Ronsse: Why you’ll probably regret using Tailwind. Disponível em: <https://johanronsse.be/2020/07/08/why-youll-probably-regret-using-tailwind/>

Jared White: Why Tailwind CSS isn’t for Me. Disponível em: <https://dev.to/jaredcwhite/why-tailwind-isn-t-for-me-5c90>