



**UNINASSAU**

# FRONT-END FRAMEWORKS

Análise e Desenvolvimento de Sistemas

Prof. Jonas Bernardino

João Pessoa-PB  
- Agosto de 2025 -

# O QUE É UM FRAMEWORK?

Um **framework** é uma estrutura ou conjunto de ferramentas que fornece uma base para o desenvolvimento de software. Ele inclui bibliotecas, componentes, regras, e convenções que ajudam os desenvolvedores a construir aplicações de forma mais eficiente e consistente



# PRINCIPAIS FRAMEWORKS FRONT-END

## JavaScript



## CSS



# PROPÓSITO DOS FRAMEWORK



## Produtividade Aumentada

- **Componentização:** Frameworks como React, Angular, e Vue.js permitem a criação de componentes reutilizáveis. Isso significa que você pode construir elementos de UI uma vez e usá-los em várias partes da aplicação, economizando tempo e esforço



## Consistência no Código

- **Padrões e Convenções:** Frameworks impõem uma estrutura consistente para o código, o que facilita a colaboração em equipe. Todos os desenvolvedores seguem as mesmas práticas e convenções, resultando em um código mais limpo e fácil de manter.



## Manutenção Facilitada

- **Código Modular:** Com a abordagem de componentes, o código é naturalmente modular, facilitando a manutenção e a atualização de partes específicas da aplicação sem afetar o restante.

# PROPÓSITO DOS FRAMEWORK



## Desempenho Otimizado

- **Renderização Eficiente:** Frameworks modernos como React e Vue.js utilizam técnicas avançadas de renderização, como o DOM virtual, que melhora significativamente o desempenho da aplicação, garantindo que apenas as partes da interface que precisam ser atualizadas sejam renderizadas novamente.



## Melhor Experiência de Usuário (UX)

- **Interatividade Aprimorada:** Frameworks front-end permitem a criação de interfaces interativas e dinâmicas, que podem responder rapidamente às ações do usuário sem a necessidade de recarregar a página inteira.



## Facilidade de Integração

- **Integração com APIs:** Frameworks front-end facilitam a integração com APIs e serviços de backend, simplificando a comunicação entre o front-end e o backend e permitindo a criação de aplicações full-stack robustas.

# PROPÓSITO DOS FRAMEWORK



## Documentação e Recursos

- **Documentação Completa:** Frameworks bem estabelecidos geralmente têm documentação abrangente e bem mantida, o que facilita o aprendizado e a solução de problemas.

# DESAFIOS AO USAR FRAMEWORKS

## Curva de Aprendizado

- **Complexidade Inicial:** Frameworks populares como Angular, React, e Vue.js podem ter uma curva de aprendizado íngreme, especialmente para desenvolvedores que são novos na programação front-end ou que estão acostumados a trabalhar sem frameworks.
- **Atualizações e Mudanças:** À medida que frameworks evoluem, novas versões podem introduzir mudanças significativas. Manter-se atualizado e adaptar-se a essas mudanças pode ser desafiador, especialmente em projetos de longo prazo.



# DESAFIOS AO USAR FRAMEWORKS

## Sobrecarga e Complexidade

- Overhead de Configuração: Muitos frameworks vêm com um conjunto completo de ferramentas e bibliotecas, o que pode ser útil, mas também pode adicionar complexidade desnecessária em projetos menores ou mais simples.
- Complexidade Adicional: Usar um framework pode introduzir uma camada adicional de complexidade, tornando mais difícil depurar problemas ou entender completamente como o código está funcionando sob o capô.





# IMPORTÂNCIA DOS FRAMEWORKS NO FRONT-END

Padronização e Consistência

Produtividade Acelerada

Desempenho e Otimização

Manutenção e Escalabilidade

Suporte a Complexidade

Ecosistema Rico e Comunidade Ativa

Inovação Contínua

# PRODUTIVIDADE COM FRAMEWORKS

## Pequenos Projetos:

- Para projetos menores, a aceleração pode ser de cerca de **20-30%**. O impacto é menor porque a complexidade é reduzida e o overhead de aprender ou configurar um framework pode não ser totalmente justificado.

## Projetos Médios:

- Em projetos de média complexidade, a aceleração geralmente fica em torno de **30-50%**. A reutilização de componentes, padrões estabelecidos e a estruturação consistente começam a mostrar benefícios significativos.

## Grandes Projetos:

- Para projetos maiores e mais complexos, a aceleração pode atingir **50-70%**. Aqui, o framework proporciona maior valor, não só em termos de desenvolvimento inicial, mas também em manutenção e escalabilidade.

# FRAMEWORK VS BIBLIOTECA

## O que é uma Biblioteca?

- **Definição:** Conjunto de funções e métodos prontos que você pode **chamar diretamente** no seu código, conforme necessário.
- **Analogia:** Uma biblioteca é como uma **caixa de ferramentas** — você escolhe qual ferramenta usar e quando.

## O que é um Framework?

- **Definição:** Um ambiente estruturado que fornece funções e regras de desenvolvimento. Diferente da biblioteca, no framework **é ele quem chama seu código**, seguindo um fluxo pré-definido.
- **Analogia:** Um framework é como uma **casa já construída**, onde você só escolhe como decorar e adaptar os cômodos — a estrutura principal já está definida.



Duvidas?



**UNINASSAU**

# FRONT-END FRAMEWORKS

Análise e Desenvolvimento de Sistemas

Prof. Jonas Bernardino

João Pessoa-PB  
- Agosto de 2025 -