

Programação Front-end – AULA 21

Frameworks e Bibliotecas Frontend

Matheus Moresco

Engenharia de Software - 3º Período

2025/01

Introdução

- Entender o que é um framework frontend
- Conhecer os principais frameworks do mercado
- Comparar vantagens e desvantagens
- Aprender como escolher um framework

O que é um Framework?

Um **framework** (em português, "estrutura" ou "arcabouço") é um conjunto de bibliotecas, ferramentas e boas práticas que fornece uma base estruturada para o desenvolvimento de software — neste caso, **interfaces de aplicações web** (frontend).

Características de um **framework frontend**:

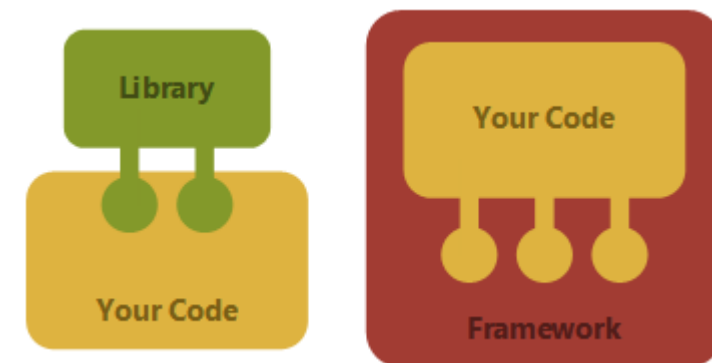
- **Organiza o código:** Impõe uma estrutura e arquitetura padrão ao projeto.
- **Acelera o desenvolvimento:** Já traz soluções prontas para tarefas comuns como manipulação de DOM, roteamento de páginas, e gerenciamento de estado.
- **Padroniza o trabalho em equipe:** Todos os desenvolvedores seguem um mesmo modelo de organização.
- **Promove reuso de código:** Por meio de componentes reutilizáveis.

Framework vs Biblioteca

A **chave da diferença** está em um conceito chamado **Inversão de Controle**

Imagine que você está cozinhando:

- **Biblioteca:** é como um **livro de receitas**.
Você escolhe a receita, adapta, decide como preparar.
→ **Você está no controle.**
- **Framework:** é como um **chefe de cozinha** que te dá ordens.
Ele diz o que fazer, quando e como. Você apenas preenche as partes que ele permite.
→ **O framework está no controle.**



Framework vs Biblioteca

Característica	Biblioteca	Framework
Controle do fluxo	O desenvolvedor chama a biblioteca	O framework chama o código do desenvolvedor
Flexibilidade	Alta: você monta sua estrutura	Baixa: estrutura imposta
Responsabilidades	Realiza tarefas específicas (ex: UI, HTTP)	Oferece uma solução completa
Curva de aprendizado	Geralmente mais baixa	Geralmente mais alta
Exemplo comum	React, Lodash, jQuery	Angular, Django (backend), Vue (parcialmente)

Vantagens de se usar Frameworks e Bibliotecas

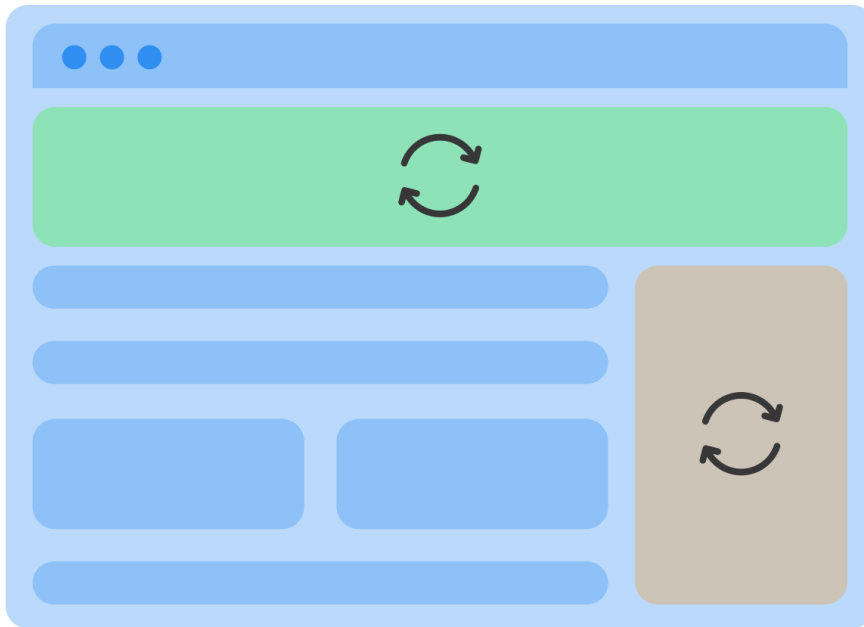
- **Produtividade aumentada:** Agilizam o desenvolvimento com ferramentas e componentes prontos.
- **Organização e manutenção:** Facilitam a estruturação do projeto e a leitura do código.
- **Recursos modernos:** Oferecem suporte nativo a roteamento, formulários, testes e mais.
- **SPAs rápidas:** Permitem criar interfaces dinâmicas sem recarregar a página.
- **Comunidade ativa:** Documentação ampla, tutoriais e fóruns com suporte constante.
- **Ecossistema rico:** Integrações fáceis com bibliotecas de UI, backend e autenticação.
- **Escalabilidade:** Projetos crescem de forma organizada e com bom desempenho.

O que são SPAs (Single Page Applications)?

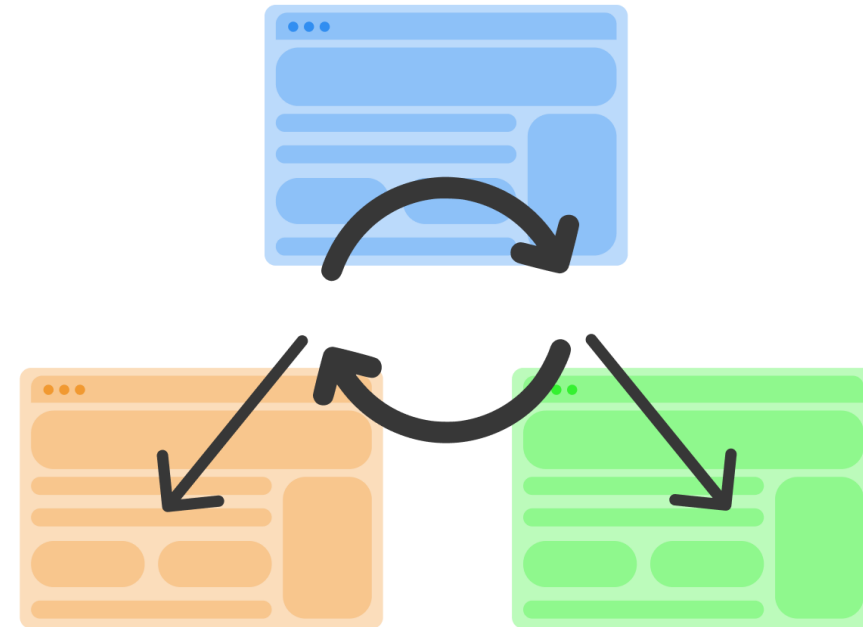
- **SPA (Single Page Application)** é um tipo de aplicação web que carrega uma única página HTML e atualiza dinamicamente o conteúdo conforme o usuário navega, sem recarregar a página inteira do navegador.
- Como funciona uma SPA:
 - Na primeira visita, o navegador carrega **todo o HTML, CSS e JavaScript principais**.
 - A partir daí, ao clicar em links ou menus, **somente partes da tela são atualizadas**, via JavaScript.
 - Os dados e novos conteúdos são carregados do servidor por meio de **requisições assíncronas** (como fetch ou axios).

SPA vs MPA

SPA



MPA



SPAs (Single Page Applications)

- **Vantagens das SPAs:**

- Navegação mais rápida e fluida
- Melhor experiência do usuário (UX)
- Redução no tráfego entre cliente e servidor
- Ideal para apps como Gmail, Trello, Facebook, etc.

- **Desvantagens:**

- Pode ser mais complexa de configurar (ex: rotas, SEO)
- Primeira carga pode ser mais pesada
- Necessita de JavaScript ativo no navegador

Principais Ferramentas Frontend

- **ReactJS** (Meta/Facebook)
- **Angular** (Google)
- **Vue.js** (Comunitário)
- Outros: Svelte, SolidJS, Alpine.js

Angular

- **Angular é um framework frontend completo, criado e mantido pelo Google, voltado para o desenvolvimento de aplicações web modernas, especialmente Single Page Applications (SPA).**



```
// app.component.ts
import { Component } from '@angular/core';

@Component({
  selector: 'app-root',
  template: `<h1>Olá {{ nome }}</h1>`,
})
export class AppComponent {
  nome = 'Mundo';
}
```

Principais Características do Angular

Framework completo

- Já vem com tudo incluído: roteamento, formulários, injeção de dependência, testes, serviços, etc.
- Não há necessidade de instalar muitas bibliotecas externas.

Baseado em TypeScript

- Usa **TypeScript** (um superconjunto de JavaScript com tipagem estática), o que ajuda a evitar erros e melhora a produtividade em equipes grandes.

Componentização

- Assim como React e Vue, Angular é baseado em **componentes reutilizáveis**.

MVVM (Model-View-ViewModel)

- Segue um padrão arquitetural claro, separando bem as responsabilidades.

CLI poderosa

- Angular CLI (Command Line Interface) permite gerar, testar e construir aplicações com comandos simples.

Angular

Vantagens do Angular

- Completo: já vem com tudo necessário
- Escalável: ideal para grandes aplicações e equipes
- Muita documentação e suporte do Google
- Alta produtividade com TypeScript e Angular CLI

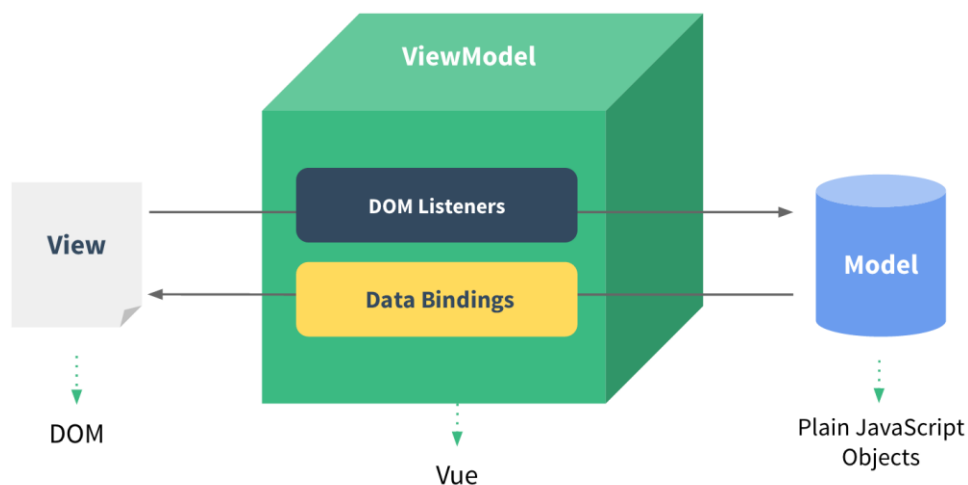
Desvantagens do Angular

- Curva de aprendizado íngreme (conceitos como módulos, decorators, DI)
- Verbosidade no código
- Mais "pesado" que bibliotecas como React ou Vue

Vue.js



- **Vue.js** é um **framework progressivo** para construção de interfaces de usuário. Ele foi criado por **Evan You** (ex-Google) e lançado em 2014. É leve, simples de aprender e muito flexível — ideal tanto para projetos pequenos quanto grandes.



```
<!-- this is our View -->
<div id="example-1">
  Hello {{ name }}!
</div>
```

```
// this is our Model
var exampleData = {
  name: 'Vue.js'
}
// create a Vue instance, or, a "ViewModel"
// which links the View and the Model
var exampleVM = new Vue({
  el: '#example-1',
  data: exampleData
})
```

Principais Características do Vue.js

- **Reatividade:** Atualiza automaticamente a interface sempre que os dados mudam.
- **Componentização:** A interface é dividida em componentes reutilizáveis e organizados.
- **Simplicidade:** Sintaxe intuitiva e acessível, baseada em HTML, CSS e JavaScript.
- **Framework progressivo:** Pode ser usado como uma simples biblioteca ou como um framework completo, conforme a necessidade do projeto.
- **Ferramentas integradas:** Suporte nativo para roteamento, gerenciamento de estado (Pinia/Vuex), testes e CLI.
- **Vue DevTools:** Ferramenta poderosa para depuração e visualização da árvore de componentes e estado em tempo real.

Vue.js

Vantagens

- Leve e rápido
- Fácil de aprender e começar
- Ótima documentação oficial
- Pode ser adotado aos poucos (ex: apenas para uma parte do site)
- Comunidade ativa e em crescimento

Desvantagens

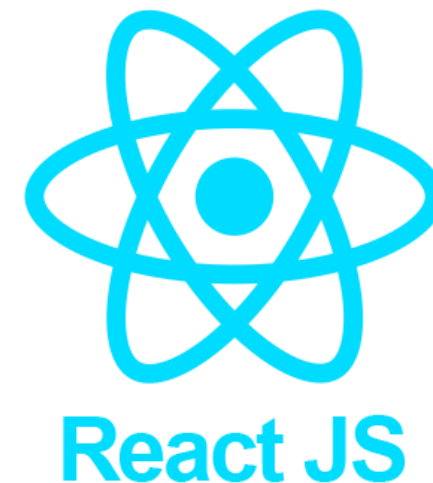
- Menor uso em grandes corporações comparado ao React/Angular
- Menor mercado de trabalho em alguns países
- Projetos muito grandes podem exigir mais configuração manual

React JS

- React é uma **biblioteca** JavaScript criada pelo **Facebook (Meta)** em 2013 para construção de interfaces de usuário. Seu foco principal é ser eficiente, declarativo e baseado em componentes. Ele é usado para criar aplicações web interativas, especialmente SPAs (Single Page Applications).

Por que React é uma biblioteca (e não um framework)?

- Porque ele se concentra apenas na camada de visualização da aplicação (UI). Para funcionalidades como roteamento, gerenciamento de estado ou testes, o desenvolvedor escolhe e integra outras ferramentas, enquanto um framework já traz tudo isso pronto e integrado.



Principais Características do React JS

- **Componentes reutilizáveis:** A UI é dividida em pequenas peças independentes e reutilizáveis.
- **JSX (JavaScript XML):** Sintaxe que permite escrever HTML dentro do JavaScript de forma mais legível.
- **DOM Virtual (Virtual DOM):** React atualiza apenas o que mudou na tela, tornando a aplicação mais rápida.
- **Unidirecionalidade dos dados:** Os dados fluem em uma única direção, tornando o código mais previsível.
- **Ecossistema flexível:** Pode ser usado com diversas bibliotecas, ferramentas e até combinado com outros frameworks.

React JS

- **Vantagens**

- Fácil de aprender se você já conhece JavaScript
- Alto desempenho com o Virtual DOM
- Reutilização de código com componentes
- Grande comunidade, ótima documentação e muitas vagas no mercado
- Muito usado por empresas como Meta, Instagram, Airbnb, Netflix, etc.

- **Desvantagens**

- Precisa de configuração (como Webpack, Babel) para projetos do zero
- Exige bibliotecas externas para funcionalidades completas (ex: roteamento, estado)
- JSX pode ser estranho no início para quem vem do HTML puro

React vs Angular vs Vue

Característica	React	Angular	Vue
Tipo	Biblioteca	Framework	Framework
Linguagem	JavaScript/TypeScript	TypeScript	JavaScript/TypeScript
Curva de aprendizado	Média	Alta	Baixa
Flexibilidade	Alta	Baixa	Alta
Popularidade	Muito alta	Alta	Média/Alta

