

Desenvolvimento Web II

Aula 08 - Linguagens e frameworks modernos para desenvolvimento web

Prof. Fabricio Bizotto

Instituto Federal Catarinense

fabricio.bizotto@ifc.edu.br

Ciência da Computação

12 de abril de 2024

1 Linguagens e frameworks modernos para desenvolvimento web

2 Experimentos

Linguagens e frameworks modernos para desenvolvimento web

O que é framework e para que serve?

Um framework é um conjunto de classes e funções que auxiliam o desenvolvimento de um software. Esses códigos trazem funcionalidades já determinadas para agilizar o processo e evitar que as pessoas tenham que reescrever essas funções frequentemente.

Benefícios

- Agilidade no desenvolvimento
- Padronização
- Segurança
- Manutenção
- Comunidade ativa

Desvantagens

- **Código Inchado:** muitas vezes o framework traz muitas funcionalidades que não serão utilizadas no projeto.
- **Problemas com suporte:** se o framework não for bem documentado, pode ser difícil encontrar soluções para problemas.
- **Overhead de recursos:** o framework pode consumir muitos recursos do servidor, o que pode ser um problema em projetos com muitos acessos.


Linguagens e frameworks modernos para desenvolvimento web

Principais frameworks para desenvolvimento web




Figura: Principais frameworks para desenvolvimento web

JQuery


- Desenvolvido por John Resig em 2006.
- É uma biblioteca JavaScript que facilita a manipulação do DOM, o desenvolvimento de animações e a interação com o servidor.
- É uma das bibliotecas JavaScript mais utilizadas no mundo.
- Pode ser substituída pela biblioteca **Vannila JS**, que é mais leve e performática.
-  Site oficial

Angular


- Desenvolvido pelo Google em 2010. Popular no desenvolvimento de Single Page Applications (SPA)^a.
- Utilizava inicialmente a linguagem JavaScript, mas atualmente utiliza TypeScript
- Utiliza o padrão de projeto MVC (Model-View-Controller).
- Utiliza o conceito de **two-way data binding**, que permite que a alteração de um elemento na interface reflita automaticamente no modelo de dados e vice-versa.
- Usa o conceito de **diretivas**, que são atributos HTML que permitem a criação de novos elementos HTML.
- A partir de 2016, a versão 2 do Angular foi lançada, com grandes mudanças em relação à versão anterior. Atualmente, a versão mais recente é a 17.
-  Site oficial

^aaplicativo web que carrega uma única página HTML e atualiza dinamicamente o conteúdo conforme o usuário interage, proporcionando uma experiência mais fluida e rápida.


React

- Desenvolvido pelo Facebook em 2011. Popular no desenvolvimento de Single Page Applications (SPA).
- Utiliza a linguagem JavaScript.
- Utiliza o padrão de projeto MVC (Model-View-Controller).
- Utiliza o conceito de **componentes**, que são elementos que encapsulam o código HTML, CSS e JavaScript.
- Modernizou a sintaxe do JavaScript, permitindo a manipulação do DOM de forma mais simples.
-  Site oficial


Vue

- Desenvolvido por Evan You em 2014.
- Também traz alguns elementos que já comentamos, como data binding, DOM virtual e suporte a SPAs.
- Vue é um framework progressivo que pode ser adotado gradualmente em partes de um projeto, facilitando a integração com projetos existentes.
- O ciclo de vida de um componente Vue é composto por 8 etapas: **beforeCreate**, **created**, **beforeMount**, **mounted**, **beforeUpdate**, **updated**, **beforeDestroy** e **destroyed**.
- Outros frameworks JavaScript: Ember, Backbone, Meteor, Aurelia, Svelte, etc.
-  Site oficial


Ember

- Desenvolvido por Yehuda Katz em 2011.
- É um framework JavaScript que segue o princípio de **convenção sobre configuração**. Isso significa que o desenvolvedor não precisa configurar muitas coisas, pois o framework já traz uma série de convenções que facilitam o desenvolvimento.
- Utiliza o padrão de projeto MVC (Model-View-Controller).
- Usa Handlebars como template engine.
- Podemos criar aplicações web completas com Ember, incluindo o backend e Single Page Applications (SPA).
-  Site oficial


Svelte

- Desenvolvido por Rich Harris em 2016.
- É um framework JavaScript que utiliza o conceito de **Build Time Framework**. Isso significa que o código é compilado para JavaScript puro, sem a necessidade de uma biblioteca de terceiros.
- **Zero Runtime**: o código gerado é muito menor do que o código gerado por outros frameworks, o que resulta em uma aplicação mais leve.
- **Programação Declarativa**: o Svelte utiliza uma sintaxe declarativa para a criação de componentes.
- Outros frameworks JavaScript: Backbone, Meteor, Aurelia, etc.
-  Site oficial


Next.js

- Desenvolvido por Vercel em 2016.
- É um framework JavaScript que utiliza o conceito de **Server Side Rendering (SSR)**. Isso significa que o código é executado no servidor e o HTML é gerado no servidor, o que resulta em uma aplicação mais rápida.
- **Static Site Generation:** o Next.js permite a geração de sites estáticos, o que resulta em uma aplicação mais leve.
- **Roteamento Automático:** as rotas são geradas automaticamente com base na estrutura de pastas do projeto.
- **Hot Module Replacement:** o Next.js permite a atualização de módulos durante o desenvolvimento sem a necessidade de recarregar a página.
- Outros frameworks JavaScript: Nuxt.js, Gatsby, Sapper, etc.
-  Site oficial


Laravel

- Desenvolvido por Taylor Otwell em 2011.
- É um framework PHP que utiliza o padrão de projeto MVC (Model-View-Controller).
- Possui uma sintaxe simples e elegante, que permite o desenvolvimento de aplicações web de forma rápida e fácil.
- Possui uma comunidade ativa e uma documentação completa.
- Outros frameworks PHP: Symfony, CodeIgniter, CakePHP, Zend Framework, Yii, Phalcon, FuelPHP, Slim, etc.
-  Site oficial


Django

- Desenvolvido por Adrian Holovaty e Simon Willison em 2003.
- É um framework Python que utiliza o padrão de projeto MVT (Model-View-Template).
- Pode ser usado para criar uma aplicação com backend e frontend, ou apenas o backend (API).
- Outros frameworks Python: Flask, Pyramid, Bottle, CherryPy, Tornado, etc.
-  Site oficial


Spring

- Desenvolvido por Rod Johnson em 2002.
- É um framework Java que utiliza o padrão de projeto MVC (Model-View-Controller).
- Usa Gradle ou Maven para gerenciar dependências.
- Outros frameworks Java: JSF, Struts, Vaadin, GWT, Wicket, Hibernate, JPA, Maven, etc.
-  Site oficial


Ruby on Rails

- Desenvolvido por David Heinemeier Hansson em 2004.
- É um framework Ruby que utiliza o padrão de projeto MVC (Model-View-Controller).
- Possui uma sintaxe simples e elegante, que permite o desenvolvimento de aplicações web de forma rápida e fácil.
- Possui uma comunidade ativa e uma documentação completa.
- Outros frameworks Ruby: Sinatra, Padrino, Hanami, Devise, etc.
-  Site oficial

Express

- Desenvolvido por TJ Holowaychuk em 2010.
- É um framework JavaScript que utiliza o padrão de projeto MVC (Model-View-Controller).
- É um dos frameworks mais populares para desenvolvimento de aplicações web com Node.js.
- Outros frameworks Node.js: Koa, Hapi, Sails, Meteor, Nest, etc.
-  Site oficial

ASP.NET

- Desenvolvido pela Microsoft em 2002.
- É um framework que utiliza o padrão de projeto MVC (Model-View-Controller).
- É um dos frameworks mais populares para desenvolvimento de aplicações web com .NET.
- Outros frameworks .NET: ASP.NET Core, Nancy, ServiceStack, etc.
-  Site oficial

Considerações finais

- A escolha de um framework depende do tipo de aplicação que será desenvolvida, da linguagem de programação que será utilizada, do conhecimento da equipe, do tempo disponível para o desenvolvimento, entre outros fatores.
- A escolha de um framework deve ser feita com cuidado, pois uma vez que o projeto é iniciado com um framework, é difícil mudar para outro.
- A escolha de um framework deve ser feita com base em critérios técnicos, e não apenas por modismo ou por ser o mais popular.

- 1 Pesquise sobre frameworks web e escolha um deles para estudar.
- 2 Crie um repositório no GitHub
- 3 Crie um arquivo README.md com as informações sobre o framework escolhido, contendo: **Nome do framework, Ano de lançamento, Desenvolvedor, Linguagem de programação, Quando é recomendado utilizar o framework, Principais características (mínimo de 3), Um vídeo, podcast ou artigo sobre o framework, Site oficial.**
- 4 Crie um exemplo prático de uso do framework escolhido e compartilhe no repositório. Pode ser uma aplicação web simples para:
 - Manipulação do DOM, Consumo de uma API, Criação de um componente, etc
 - Converter texto para voz
 - Sistema de votação para uma enquete em tempo real
 - Outra funcionalidade que você achar interessante
- 5 Compartilhe o link do repositório com o colega e peça para ele revisar o README e seu exemplo prático.
- 6 **Encerramento:** reflexão entre a dupla sobre o que foi desenvolvido na atividade.
- 7 **Entrega:** Entrega do link do repositório no SIGAA.