

FRAMEWORKS PARA HTML E JAVASCRIPT

ENGENHARIA DE COMPUTAÇÃO

7º SEMESTRE

Nome: Murilo Formaggio Feliciano PC3006875

Nome: Jefferson Roberto Piaia Martines PC3015602

Nome: Lucas Brancatti Pasqualetto PC3010546

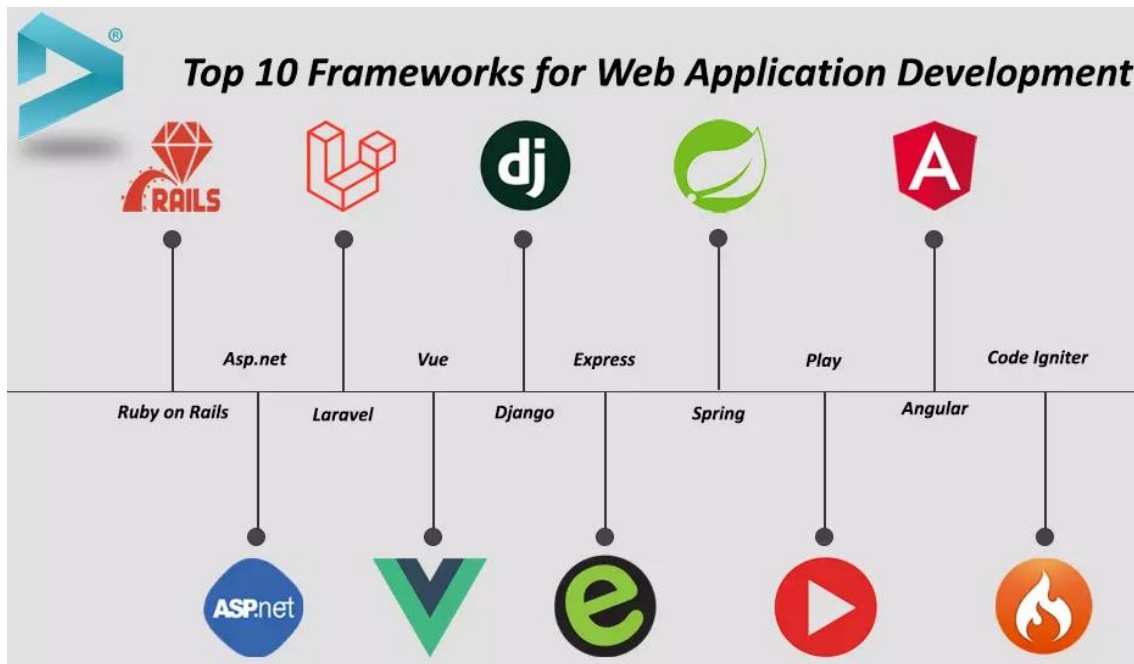
Nome: Thomas Dechen Ferreira PC3010732

Framework

Um framework é uma estrutura de software que fornece uma base para o desenvolvimento de aplicativos ou sistemas. É uma abstração de alto nível que oferece um conjunto de diretrizes, bibliotecas, componentes e funcionalidades comuns para facilitar o processo de desenvolvimento. Em essência, um framework é um esqueleto sobre o qual os desenvolvedores constroem suas aplicações. Ele define uma estrutura organizacional, fluxos de trabalho e convenções que ajudam a orientar o desenvolvimento de software. Um framework fornece uma arquitetura básica para resolver problemas específicos, eliminando a necessidade de começar do zero em cada projeto. Os frameworks podem ser genéricos, abrangendo diversos domínios, como web, mobile, desktop, jogos, machine learning, entre outros. Alguns exemplos populares de frameworks são o Django (framework web em Python), o React (framework JavaScript para interfaces de usuário), o Flutter (framework de desenvolvimento de aplicativos móveis multiplataforma) e o TensorFlow (framework para aprendizado de máquina). As principais vantagens do uso de frameworks incluem:

- 1 - Produtividade: Os frameworks fornecem estruturas e bibliotecas predefinidas que aceleram o desenvolvimento e permitem que os desenvolvedores se concentrem na lógica do aplicativo em vez de lidar com detalhes técnicos de baixo nível.
- 2 - Reutilização de código: Os frameworks geralmente possuem componentes reutilizáveis, o que reduz a duplicação de esforço e promove a consistência no código.
- 3 - Melhores práticas: Os frameworks geralmente incorporam boas práticas e padrões de desenvolvimento testados, permitindo que os desenvolvedores sigam diretrizes estabelecidas para obter um código mais organizado, legível e de alta qualidade.
- 4 - Comunidade e suporte: Frameworks populares geralmente têm uma comunidade ativa de desenvolvedores que compartilham conhecimento, resolvem problemas e fornecem suporte mútuo.

No entanto, é importante observar que os frameworks também podem ter algumas limitações. Eles podem impor certas restrições ou adotar abordagens específicas que podem não ser adequadas para todos os casos. Portanto, é importante selecionar um framework com base nos requisitos do projeto, no conhecimento da equipe e nos objetivos de longo prazo.



Exemplos:

Ruby on Rails

Também conhecido como Rails ou RoR é um framework de desenvolvimento de aplicações web escrito em Ruby. Ele segue o padrão Model-View-Controller (MVC) e fornece uma estrutura de trabalho para simplificar o processo de desenvolvimento de aplicativos web. O Ruby on Rails foi criado por David Heinemeier Hansson em 2004 e rapidamente ganhou popularidade devido à sua abordagem concisa e convenções que favorecem a produtividade do desenvolvedor. O framework enfatiza a convenção sobre a configuração, o que significa que, ao seguir determinadas convenções, grande parte da configuração manual é eliminada, tornando o desenvolvimento mais ágil. Principais características do Ruby on Rails:

1 - MVC: O Rails adota o padrão MVC, que separa as responsabilidades do aplicativo em Model (modelo), View (visualização) e Controller (controlador). Isso promove uma organização clara do código e facilita a manutenção e a extensibilidade do aplicativo.

2 - Convenções: O Rails possui um conjunto de convenções que padronizam a estrutura e a nomenclatura do código. Isso simplifica o desenvolvimento, pois muitas decisões de design são tomadas automaticamente, permitindo que os desenvolvedores se concentrem em resolver problemas específicos do domínio.

3 - ActiveRecord: O ActiveRecord é um componente-chave do Rails que facilita o mapeamento objeto-relacional (ORM). Ele fornece uma camada de abstração para interagir com o banco de dados, permitindo que os desenvolvedores trabalhem com dados de forma mais orientada a objetos.

4 - Scaffolding: O Rails oferece recursos de scaffolding que geram automaticamente código CRUD básico (Create, Read, Update, Delete) com base nas definições do modelo. Isso acelera a criação de protótipos e agiliza o desenvolvimento inicial do aplicativo.

5 - Gems: O Ruby on Rails possui um sistema de gerenciamento de pacotes chamado RubyGems, que permite adicionar funcionalidades extras ao aplicativo através de bibliotecas chamadas gems. Há uma ampla variedade de gems disponíveis, abrangendo diversos aspectos do desenvolvimento web.

O Ruby on Rails é amplamente utilizado para o desenvolvimento de aplicativos web devido à sua simplicidade, produtividade e ecossistema maduro. Ele é particularmente adequado para o desenvolvimento rápido de protótipos, startups e aplicações web de médio porte.

Laravel

É um framework de desenvolvimento de aplicativos web em PHP. Foi criado por Taylor Otwell e lançado em 2011. Assim como o Ruby on Rails, o Laravel segue o padrão Model-View-Controller (MVC) e oferece uma estrutura de trabalho elegante e intuitiva para o desenvolvimento de aplicativos web.

O Laravel é amplamente utilizado na comunidade PHP para o desenvolvimento de aplicativos web, desde pequenos projetos até aplicações empresariais complexas. Ele fornece uma experiência de desenvolvimento agradável, eficiente e segura, permitindo que os desenvolvedores construam aplicativos web de alta qualidade de forma rápida e eficaz.

Django

É um framework de desenvolvimento web em Python que segue o padrão Model-View-Controller (MVC) e é conhecido por sua abordagem pragmática, eficiente e segura. Foi criado por Adrian Holovaty e Simon Willison e lançado em 2005. O Django é amplamente utilizado para construir aplicativos web de todos os tamanhos e complexidades.

O Django é amplamente utilizado por desenvolvedores Python devido à sua simplicidade, produtividade e robustez. Ele fornece uma base sólida para o desenvolvimento web, permitindo que os desenvolvedores criem aplicativos web escaláveis, seguros e de alta qualidade.

Angular

É um framework de desenvolvimento web para a criação de aplicativos de página única (SPA - Single-Page Applications) e aplicativos web dinâmicos. Foi

desenvolvido e é mantido pela equipe do Google. Angular é escrito em TypeScript e segue o padrão arquitetural Model-View-ViewModel (MVVM). Aqui estão algumas características e vantagens principais do Angular:

1 - Estrutura robusta: Angular fornece uma estrutura completa e abrangente para o desenvolvimento de aplicativos web. Ele inclui uma ampla gama de recursos, como manipulação do DOM, roteamento, gerenciamento de estado, validação de formulários, comunicação com APIs, testes automatizados e muito mais.

2 - Component-based: O Angular adota uma abordagem baseada em componentes, onde os aplicativos são construídos através da composição de componentes reutilizáveis. Os componentes encapsulam a lógica e a apresentação de uma parte específica da interface do usuário, tornando o código mais modular, reutilizável e fácil de manter.

3 - Two-way data binding: O Angular oferece vinculação de dados bidirecional, permitindo que as alterações nos dados do modelo sejam refletidas automaticamente na interface do usuário e vice-versa. Isso simplifica a manipulação e a sincronização de dados entre o modelo e a visualização, reduzindo a quantidade de código necessário para atualizar a interface.

4 - Injeção de dependência: Angular possui um sistema de injeção de dependência integrado que facilita a criação, gerenciamento e compartilhamento de objetos e serviços entre os componentes do aplicativo. A injeção de dependência ajuda a promover a modularidade, reutilização de código e testabilidade do aplicativo.

5 - CLI (Command Line Interface): O Angular possui uma poderosa interface de linha de comando chamada Angular CLI, que facilita a criação, a compilação e o teste de aplicativos Angular. A CLI automatiza tarefas comuns, como a geração de componentes, serviços, rotas e a execução de testes, acelerando o processo de desenvolvimento.

6 - Ecossistema e comunidade ativa: Angular possui uma comunidade ativa de desenvolvedores e um ecossistema próspero de bibliotecas e ferramentas de terceiros. Isso significa que você pode encontrar uma ampla variedade de recursos adicionais, pacotes de terceiros e suporte para estender as funcionalidades do Angular de acordo com as necessidades do seu projeto.

Angular é amplamente utilizado no desenvolvimento de aplicativos web escaláveis e complexos. Ele oferece uma arquitetura sólida, ferramentas poderosas e recursos avançados para criar aplicativos web modernos com uma excelente experiência do usuário.

Referências bibliográficas

Smith, J. (2022). Explorando Frameworks: Uma visão abrangente dos principais frameworks de desenvolvimento web. Nova York: Editora XYZ.

Freeman, A., & Robson, E. (2020). Head First Design Patterns: A Brain-Friendly Guide. Sebastopol, CA: O'Reilly Media.

Johnson, E. (2018). Designing with Web Standards. Berkeley, CA: New Riders.

Mead, A. (2019). JavaScript: The Good Parts. Sebastopol, CA: O'Reilly Media.

Powell, C. (2017). Learning JavaScript Design Patterns. Birmingham, UK: Packt Publishing.

Scott, J. (2021). Pro Angular 12. Berkeley, CA: Apress.