

Frameworks de Desenvolvimento Back-end

Uma jornada pelo mundo dos frameworks que transformam o desenvolvimento de aplicações modernas

ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

Bem-vindos

Apresentação da Disciplina



Fundamentos

Teoria essencial de back-end e arquitetura

Prática

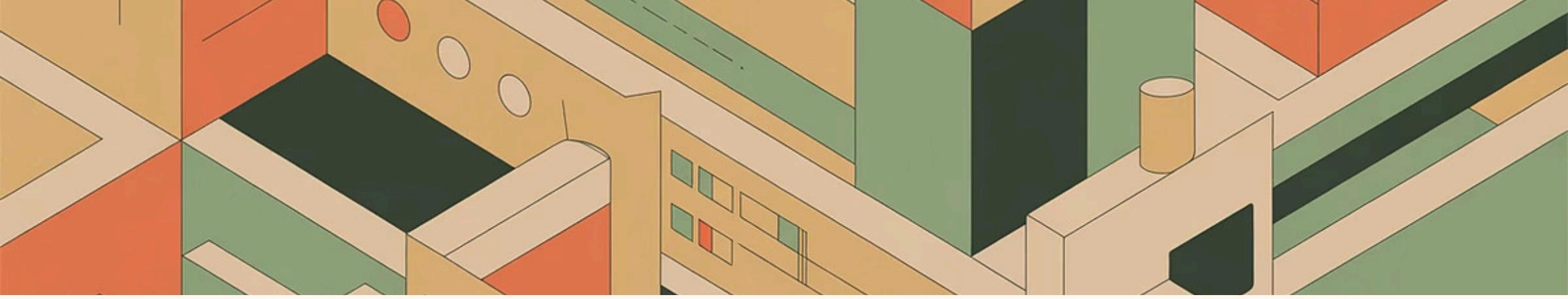
Uso de frameworks populares em exercícios

Mercado

Preparação de projetos e competências profissionais

Objetivos do Semestre

- Compreender o papel dos frameworks no desenvolvimento back-end
- Explorar frameworks modernos e suas aplicações práticas
- Desenvolver habilidades técnicas alinhadas ao mercado
- Construir projetos reais utilizando as melhores práticas



O que é um Framework?

Conceito Simples

Um framework é como uma **fundação pré-construída** para sua aplicação. Em vez de começar do zero, você trabalha sobre uma estrutura que já resolve problemas comuns.

Definição Técnica

Conjunto de **componentes, bibliotecas e padrões arquiteturais** que fornecem uma base reutilizável para desenvolver aplicações, invertendo o controle do fluxo de execução.

Frameworks Back-end Populares

Conheça as principais tecnologias que dominam o mercado de desenvolvimento



Java

Spring Boot — Robusto, escalável e amplamente usado em sistemas empresariais de grande porte



Node.js

Express.js, NestJS — Rápido, assíncrono e ideal para aplicações em tempo real



Python

Django, Flask — Elegante, produtivo e perfeito para prototipagem rápida e IA



PHP

Laravel, Symfony — Sintaxe expressiva e ecossistema maduro para web



C#

ASP.NET Core — Performance excepcional e integração completa com Microsoft Azure

Biblioteca vs Framework

Entendendo a diferença fundamental

A Analogia Perfeita

Biblioteca: Você está no comando. É como ter uma caixa de ferramentas — você escolhe qual ferramenta usar e quando usar.

Framework: O framework está no comando. É como uma linha de montagem — você segue o processo estabelecido e preenche as lacunas.

Comparação Técnica

| Característica | Biblioteca | Framework |
|----------------------|---------------|----------------|
| Controle do fluxo | Desenvolvedor | Framework |
| Inversão de controle | Não | Sim |
| Arquitetura definida | Não | Sim |
| Flexibilidade | Alta | Moderada |
| Curva de aprendizado | Baixa | Alta |
| Exemplos | Lodash, Axios | Spring, Django |

Vantagens do Uso de Frameworks

Por que a indústria adotou frameworks como padrão de desenvolvimento

1 Produtividade Acelerada

Reutilização de código testado e otimizado reduz drasticamente o tempo de desenvolvimento. Foque na lógica de negócio, não em problemas já resolvidos.

2 Padrões e Boas Práticas

Frameworks impõem arquiteturas comprovadas (MVC, DDD, Clean Architecture) garantindo código organizado, manutenível e escalável desde o início.

3 Segurança Reforçada

Proteções contra vulnerabilidades comuns (SQL Injection, XSS, CSRF) vêm integradas e são constantemente atualizadas pela comunidade.

4 Comunidade Ativa

Documentação extensa, tutoriais, plugins, e suporte da comunidade facilitam a resolução de problemas e o aprendizado contínuo.

5 Manutenibilidade

Estrutura padronizada facilita a entrada de novos desenvolvedores no projeto e reduz o custo de manutenção a longo prazo.

Desvantagens e Cuidados

O outro lado da moeda



Curva de Aprendizado

Frameworks complexos exigem tempo significativo para domínio completo de conceitos, convenções e melhores práticas.



Dependência Tecnológica

Forte acoplamento ao framework pode dificultar migrações futuras e limitar escolhas arquiteturais específicas.



Overhead de Performance

Frameworks trazem funcionalidades que você pode não usar, resultando em aplicações mais pesadas do que o necessário.



Atualizações e Breaking Changes

Versões novas podem quebrar compatibilidade, exigindo refatorações custosas em projetos legados.

Boas Práticas

- **Escolha com critério:** avalie o tamanho do projeto, requisitos e expertise da equipe
- **Não abuse:** nem todo projeto precisa de um framework robusto
- **Entenda os fundamentos:** conheça a linguagem antes de depender do framework
- **Mantenha-se atualizado:** acompanhe mudanças e tendências da tecnologia escolhida
- **Evite over-engineering:** simplicidade muitas vezes supera complexidade

Quando Usar um Framework?

Projetos Médios/Grandes

Aplicações empresariais que demandam escalabilidade, manutenibilidade e trabalho em equipe

Longo Prazo

Projetos com ciclo de vida extenso que precisarão de manutenção e evolução contínuas



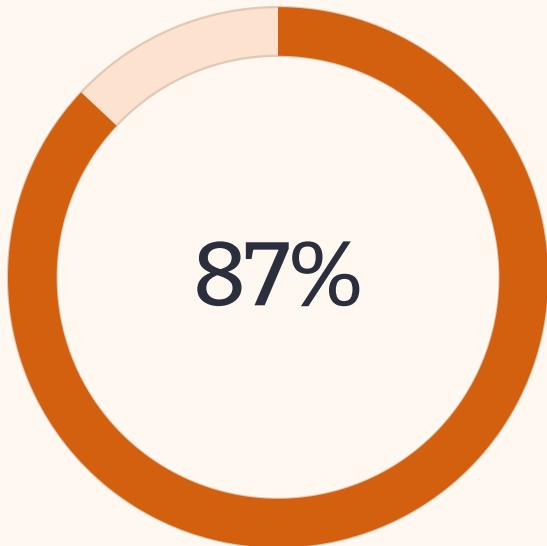
Padrões Estabelecidos

Quando você precisa seguir arquiteturas conhecidas e facilitar onboarding de desenvolvedores

Projetos Simples

Scripts pequenos, protótipos rápidos ou aplicações com requisitos muito específicos

Preparando-se para o Mercado



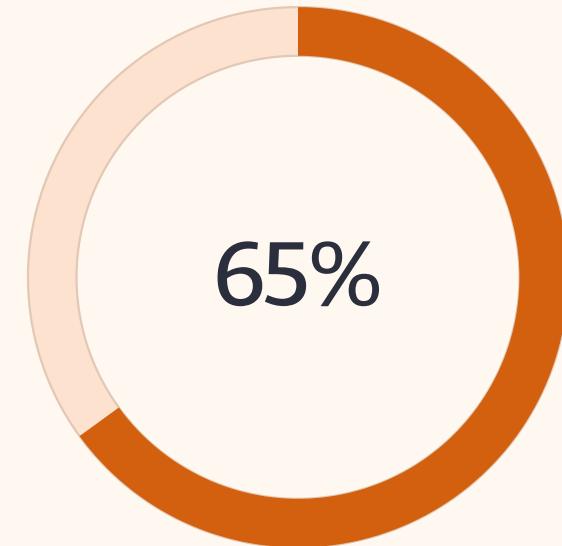
Das vagas back-end

Exigem conhecimento em pelo menos um framework popular



Meses de estudo

Tempo médio para alcançar proficiência em um framework moderno



Aumento salarial

Diferença média entre profissionais com e sem domínio de frameworks enterprise

Dica profissional: Comece dominando os fundamentos da linguagem, depois mergulhe em um framework amplamente usado no mercado. A expertise em um framework facilita o aprendizado de outros.

Perguntas para Reflexão

Reflexão Individual

- Qual framework você já conhece ou gostaria de aprender? Por quê?
- Como você avaliaria se um projeto precisa de um framework ou não?
- Quais habilidades fundamentais você precisa desenvolver antes de mergulhar em um framework?

Para Discussão

- Um desenvolvedor deve aprender vários frameworks ou se especializar em um?
- Frameworks facilitam ou dificultam o aprendizado de programação?
- Como equilibrar produtividade com entendimento profundo da tecnologia?

Próxima aula: Mergulharemos no primeiro framework prático, explorando instalação, configuração e desenvolvimento do nosso primeiro projeto.

OBRIGADO PELA ATENÇÃO!