PROJETO, IMPLEMENTAÇÃO E TESTE DE SOFTWARE

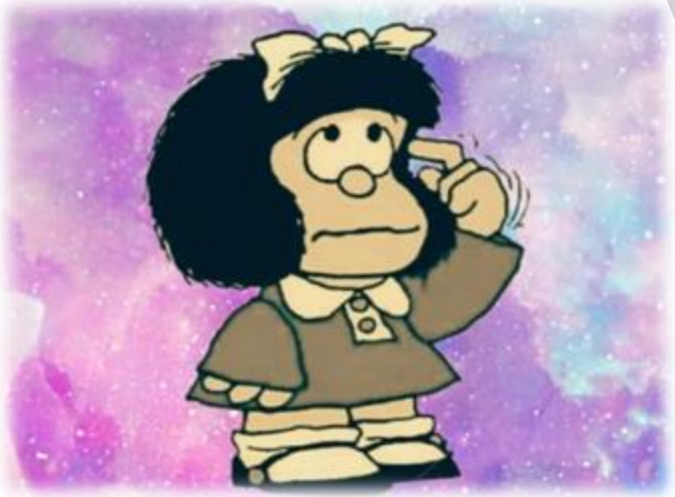
Aula 07 – Tópicos Especiais

Professor Fabricio Freire

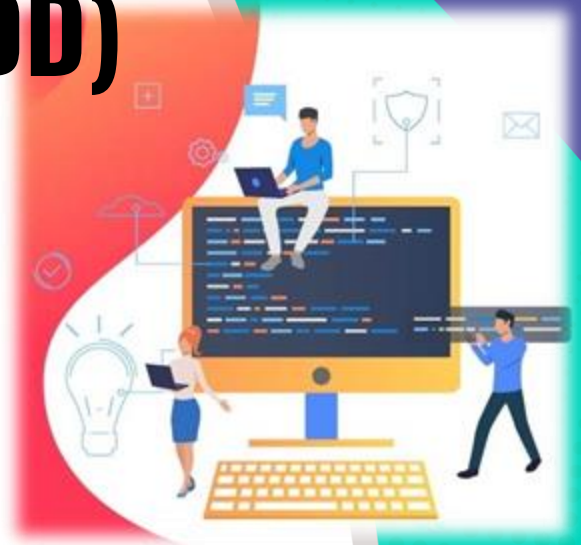
METAS DE APRENDIZAGEM

- **Domain-Driven Design (DDD)**
- **Arquitetura Hexagonal**
- **Microsserviços**
- **Spring Framework & Spring Boot**
- **Integração entre os Conceitos**
- **Conclusões e Aplicações Práticas**

- Arquitetura Ortogonal
- Depuração de Código
- Asserções e Programação Defensiva
- Otimização de Desempenho
- Refatoração



Domain-Driven Design (DDD)



O que é DDD?

Abordagem de desenvolvimento focada no **domínio do negócio**

Colaboração próxima entre especialistas técnicos e de domínio

Criação de um **modelo de domínio rico** e expressivo

Conceitos Principais:

Ubiquitous Language: Linguagem comum entre equipe e stakeholders

Bounded Context: Delimita onde um modelo se aplica

Aggregate: Grupo de objetos tratados como unidade

Domain Services: Lógicas que não pertencem a entidades específicas

ARQUITETURA HEXAGONAL

Princípios Fundamentais:

- Também conhecida como **"Ports and Adapters"**
- Isola a lógica de negócio das dependências externas
- Permite **testabilidade** e **flexibilidade** elevadas

ARQUITETURA HEXAGONAL

Componentes:

Centro (Core): Lógica de negócio pura

Ports: Interfaces que definem contratos

Adapters: Implementações concretas dos ports

Externa: Base de dados, APIs, interfaces de usuário

Benefícios:

- ✓ Desacoplamento de infraestrutura
- ✓ Facilita testes unitários
- ✓ Flexibilidade para mudanças tecnológicas

MICROSSERVIÇOS

Arquitetura que estrutura aplicações como **coleção de serviços pequenos**, **independentes** e **fracamente acoplados**.

Características:

Autonomia: Cada serviço pode ser desenvolvido e “deployado” independentemente

Responsabilidade única: Cada microsserviço tem uma função específica

Comunicação via APIs: Geralmente REST ou messaging

Tecnologia diversa: Cada serviço pode usar stack diferente

MICROSSERVIÇOS

Vantagens:

- ✓ Escalabilidade granular
- ✓ Resiliência (falhas isoladas)
- ✓ Desenvolvimento paralelo por equipes
- ✓ Flexibilidade tecnológica

SPRING FRAMEWORK & SPRING BOOT

Spring Framework:

- Framework **abrangente** para desenvolvimento Java
- **Inversão de Controle (IoC)** e **Injeção de Dependências**
- Suporte para diferentes camadas da aplicação

SPRING FRAMEWORK & SPRING BOOT

Spring Boot:

- **Facilita** a configuração e inicialização de projetos Spring
- **Convenção sobre configuração**
- **Starters**: Dependências pré-configuradas
- **Auto-configuration**: Configuração automática baseada no classpath

Características Principais:

- **Embedded Servers**: Tomcat, Jetty, Undertow
- **Actuator**: Monitoramento e gerenciamento
- **Spring Security**: Segurança robusta
- **Spring Data**: Acesso facilitado a dados

INTEGRAÇÃO ENTRE CONCEITOS

DDD e Arquitetura Hexagonal

Contexto Delimitado ——► Serviço Hexagonal
Modelos de Domínio ——► Núcleo/Lógica de Negócio
Repositórios ————————► Portas (Interfaces)

INTEGRAÇÃO ENTRE CONCEITOS

Arquitetura Hexagonal e Microsserviços

- Cada microsserviço implementa arquitetura hexagonal
- Isolamento da lógica de negócio
- Facilita testes e manutenção

INTEGRAÇÃO ENTRE CONCEITOS

Spring Boot + Microsserviços:

- **Spring Cloud**: Ferramentas para microsserviços
- **Service Discovery**: Eureka, Consul
- **Load Balancing**: Ribbon, Spring Cloud LoadBalancer
- **Circuit Breaker**: Hystrix, Resilience4j

PRÁTICA

Vamos exercitar os conceitos da aula

BONS ESTUDOS