

Global Solution 2025 – O Futuro do Trabalho

Curso: Engenharia de Software – 2ESR

Disciplina: Domain Driven Design – Java

Tema: Plataforma de Upskilling / Reskilling 2030+

Professor: Anderson Fonseca

Ferramenta: Java + Spring Boot

1. CONTEXTO DO PROJETO

O futuro do trabalho está sendo impulsionado por Inteligência Artificial, automação, análise de dados e ambientes híbridos/remotos. Essas tecnologias vêm transformando carreiras e exigindo novas formas de aprendizado, tornando essencial:

Requalificação (reskilling)

Adaptação de profissionais cujas funções podem ser automatizadas.

Aperfeiçoamento contínuo (upskilling)

Aprimoramento de competências técnicas e humanas.

Educação permanente

Alinhada às demandas de 2030+ e aos Objetivos de Desenvolvimento Sustentável (ODS 4, 8, 9 e 10).

A proposta da Global Solution 2025 é desenvolver uma API RESTful em Java Spring Boot que apoie pessoas em seu processo de requalificação profissional, simulando uma plataforma de Upskilling / Reskilling voltada para o futuro do trabalho.

2. OBJETIVO DA SOLUÇÃO

O objetivo é construir uma API RESTful que permita:

- 1 Cadastrar usuários e trilhas de aprendizagem
- 2 Realizar inscrições (matrículas) de usuários em trilhas
- 3 Consultar, atualizar e remover dados via endpoints RESTful
- 4 Persistir informações em banco relacional (H2 ou MySQL)
- 5 Executar o projeto com o comando `mvn spring-boot:run`
- 6 Garantir a qualidade do código com testes unitários cobrindo no mínimo 80% da base desenvolvida

3. ESTRUTURA DE PACOTES DO PROJETO

```
futuro-do-trabalho/  
├─ pom.xml  
├─ README.md  
├─ src/  
│   ├─ main/java/com/fiap/futurotrabalho/  
│   │   ├─ controller/  
│   │   ├─ domain/  
│   │   ├─ dto/  
│   │   ├─ repository/  
│   │   ├─ service/  
│   │   ├─ exception/  
│   │   └─ util/  
│   └─ main/resources/  
│       ├─ application.yml  
│       ├─ data.sql  
│       └─ schema.sql  
└─ test/java/com/fiap/futurotrabalho/  
    ├─ service/  
    └─ controller/
```

4. REQUISITOS TÉCNICOS

Linguagem / Framework

Java 21 + Spring Boot 3.3.x – versões mínimas obrigatórias.

Banco de Dados

H2 (em memória) ou MySQL – configurado no arquivo application.yml.

Execução

Comando mvn spring-boot:run – a aplicação deve iniciar sem erros.

Arquitetura

Padrão em camadas Controller → Service → Repository, garantindo separação de responsabilidades.

Validação

Bean Validation (jakarta.validation) – uso de anotações como @NotBlank, @Email e @Positive.

Tratamento de Erros

Centralização de exceções e mensagens padronizadas.

Seeds

Arquivos data.sql e/ou uso de CommandLineRunner – inserção automática de dados iniciais.

Documentação

Arquivo README.md – deve conter instruções de execução e exemplos de requisições.

Testes Unitários

Implementação com JUnit 5 + Mockito – cobertura mínima de 80% do código entregue.

Entrega

Arquivo compactado .zip contendo o projeto completo, pronto para execução.

5. CRITÉRIOS DE AVALIAÇÃO (PONTUAÇÃO)

25

Modelagem de Domínio e Aderência ao Tema

Entidades coerentes e conectadas ao contexto de Upskilling/Reskilling 2030+.

20

Arquitetura e API RESTful

Camadas bem definidas, rotas REST corretas e status HTTP adequados.

15

Persistência e Seeds

Banco de dados funcional, entidades persistidas e dados iniciais válidos.

15

Validações e Exceções

Uso consistente de validações e tratamento centralizado de erros.

10

CRUDs Completos

Dois CRUDs (Usuário e Trilha) implementados ponta a ponta.

10

Testes Unitários

Cobertura mínima de 80% com JUnit/Mockito.

5

Documentação e Execução

README claro e execução via Maven.



Total Máximo: 100 pontos

6. FORMA DE ENTREGA

Formato

Arquivo .zip

Prazo

19/11/2025 – até 23h59

Conteúdo

- Projeto completo em Java + Spring Boot.
- Documento PDF com identificação dos integrantes.
- Scripts e dados iniciais (data.sql, schema.sql).
- Testes unitários com cobertura mínima de 80%.

7. OBSERVAÇÕES FINAIS

→ Boas Práticas

O projeto deve refletir boas práticas de Orientação a Objetos e arquitetura limpa.

→ Execução Maven

O comando `mvn spring-boot:run` deve iniciar a aplicação sem ajustes adicionais.

→ Entrega Pronta

A entrega .zip deve estar pronta para execução imediata.

→ Cobertura de Testes

Os testes unitários devem ter cobertura mínima de 80%.

→ Documentação Obrigatória

A documentação é obrigatória e compõe a nota final.



Sucesso na Entrega!

Lembre-se: qualidade, documentação e testes são fundamentais para o sucesso deste projeto. Boa sorte!