

**Cenário:**

A Universidade de Fortaleza (Unifor) pretende implantar uma plataforma unificada de gestão acadêmica, integrando alunos, professores, coordenadores e secretarias acadêmicas.

**A solução deve permitir:**

- Cadastro e gestão de usuários (alunos, professores, coordenadores);
- Gestão de cursos, semestres e matrizes curriculares;
- Visualização de dados acadêmicos pelos alunos e docentes;
- Integração segura entre módulos frontend (Angular) e backend (Java + Quarkus);
- Autenticação e autorização via Keycloak;
- Deploy e orquestração via Docker e Docker Compose.

**Etapa 1 - Análise e Arquitetura:**

- Propor uma arquitetura de referência para o sistema (diagramas e descrição dos componentes).
- Apresentar o modelo de domínio e principais entidades.
- Definir os requisitos não funcionais (performance, segurança, escalabilidade, observabilidade).
- Descrever políticas de segurança e controle de acesso via Keycloak (roles e realms).

**Entregáveis:**

- Documento de arquitetura (README detalhado ou documento separado);
- Diagramas de componentes, classes e implantação (PlantUML ou C4 Model).

## **Etapa 2 – Prototipagem e Implementação:**

- Implementar um MVP funcional com os módulos básicos de backend e frontend;
- Backend em Java + Quarkus (API REST) a partir da versão 3.15 LTS;
- Frontend em Angular 18+ (standalone);
- Banco de dados relacional (PostgreSQL, MySQL, etc.);
- Autenticação integrada ao Keycloak a partir da versão 26;
- Containers Docker e orquestração com docker-compose.

### **Entregáveis:**

- Código-fonte versionado no Git (histórico relevante e commits claros);
- Dockerfile e docker-compose.yml funcionando;
- Documentação de build e execução.

## **Etapa 3 – Mentoria e Comunicação Técnica:**

- Criar uma breve **apresentação (5-10 slides)** explicando:
  - A visão arquitetural;
  - Decisões e trade-offs;
  - Estratégias de evolução do sistema;
  - Diretrizes de boas práticas e padrões para o time de desenvolvimento.

### **Entregáveis:**

- Apresentação em PDF ou PowerPoint.

## **Critérios de Avaliação**

<b>Critério</b>	<b>Descrição</b>
Visão Sistêmica	Clareza e coerência da arquitetura proposta
Decisão Arquitetural	Justificativas técnicas e trade-offs bem documentados
Qualidade Técnica	Clareza, manutenibilidade e boas práticas de código
Segurança e Acesso	Implementação adequada de autenticação e roles
Documentação	Documentação técnica e diagramas
Liderança Técnica	Comunicação clara e capacidade de orientar o time
Inovação e UX	Soluções criativas e experiência do usuário

## Diferenciais

Candidatos que demonstrarem os seguintes diferenciais terão destaque na avaliação:

- Uso de arquitetura reativa (Mutiny);
- Implementação de comunicação orientada a eventos (ex.: Apache Kafka ou RabbitMQ);
- Adoção de arquitetura limpa ou hexagonal para separação de domínios e camadas;
- Integração assíncrona entre microserviços ou módulos;
- Implementação de observabilidade reativa (tracing, métricas e logs estruturados);
- Uso de padrões de documentação: OpenAPI/Swagger, README e diagramas C4;
- Aplicação de Design System no frontend (Material ou PrimeNG).

## Observações:

Enviar o link do repositório Git (GitHub, GitLab, Bitbucket etc.) contendo:

- Código-fonte completo;
- Documentação e diagramas;
- Scripts de execução e build;
- Apresentação final.

Em caso de dúvidas, envie um email para [rafael.reis@unifor.br](mailto:rafael.reis@unifor.br), [caiocolares@unifor.br](mailto:caiocolares@unifor.br) e [ronaldoalm@unifor.br](mailto:ronaldoalm@unifor.br).