

Projeto Final – Linguagem de Programação 2 (Java + SPA + Webservices)

Tema: Portal de Estágios

Contextualização:

Sua equipe de desenvolvimento foi contratada para criar um **Portal de Estágios**, cujo objetivo é facilitar a conexão entre **empresas ofertantes de vagas** e **estudantes em busca de oportunidades**. A solução deve ser composta por uma **aplicação backend** (API RESTful em Java + Spring Boot) e uma **Single Page Application (SPA)** desenvolvida com framework JavaScript de sua escolha (React, Angular, Vue.js etc.).

A aplicação será utilizada por três perfis de usuários:

- **Administrador do sistema**
 - **Representante de empresa**
 - **Estudante**
-

Requisitos Funcionais e Regras de Negócio

1. **Cadastro e autenticação de usuários**
 - Deve existir login individual para **estudantes, empresas e administradores**.
 - A senha deve ser armazenada com criptografia.
2. **Cadastro de áreas de interesse (admin)**
 - Somente administradores podem cadastrar, editar e remover áreas de interesse.
3. **Cadastro de empresas**
 - Empresas devem preencher: nome, CNPJ, e-mail, telefone, endereço e área(s) de atuação.
 - Após o cadastro, a empresa poderá ofertar vagas.
4. **Cadastro de estudantes**
 - Estudantes devem preencher: nome, CPF, curso, e-mail, telefone, e selecionar suas áreas de interesse.
5. **Ofertas de vagas**
 - Cada vaga deve conter: título, descrição, área, localização, modalidade (remoto/presencial/híbrido), carga horária e requisitos.
 - Apenas empresas logadas podem criar vagas.
 - Vagas devem ser listadas publicamente para estudantes com base nas áreas de interesse.
6. **Inscrição em vagas**
 - Estudantes podem se inscrever em qualquer vaga que não estejam encerradas.

- Uma vaga pode ter múltiplos candidatos; o estudante pode se inscrever em várias vagas.
7. **Painel personalizado**
- Estudante logado: deve visualizar na página inicial as **vagas em aberto** relacionadas às suas **áreas de interesse**.
 - Empresa logada: deve visualizar na página inicial os **estudantes que se inscreveram** nas suas vagas.
 - Administrador: pode visualizar estatísticas gerais do portal.
8. **Encerramento de vagas**
- A empresa pode encerrar uma vaga a qualquer momento, impedindo novas inscrições.
9. **Dashboard administrativo (mínimo)**
- Quantidade de empresas cadastradas, estudantes, vagas abertas e encerradas.
 - Gráfico com quantidade de vagas por área.
-

Entregas Técnicas

a) Backend em Spring Boot

- Criar projeto com APIs REST para todas as entidades do diagrama de domínio.
- Utilizar JPA e PostgreSQL para persistência.
- Documentar a API (Swagger ou equivalente).

b) Frontend SPA

- Criar interface web para todas as funcionalidades.
- Operações CRUD para entidades principais.
- Autenticação e autorização com controle de acesso baseado em perfil.

c) Hospedagem opcional

- Extra: grupos que entregarem a aplicação funcionando em ambiente cloud (Railway, Render, Vercel, Netlify, Heroku, etc.) ganharão até 1 ponto adicional na nota final.
-

Funcionalidade Inovadora Obrigatória (por grupo)

Cada grupo deverá implementar **uma funcionalidade adicional e inovadora, diferente das demais equipes**, previamente validada com o professor. Exemplos possíveis:

- Integração com LinkedIn para importar currículo do estudante.
- Chat interno entre empresa e candidato.
- Algoritmo de recomendação de vagas usando machine learning simples.

- Módulo de avaliação da vaga por ex-estagiários.
- Currículo em PDF gerado automaticamente para cada estudante.
- Sistema de entrevistas agendadas com calendário e notificações.
- Gamificação do portal com pontuação e conquistas para os estudantes.

Avaliação

Critério	Peso
Funcionalidades obrigatórias	4,0
Interface e usabilidade (SPA)	2,0
Estrutura e qualidade do código	2,0
Funcionalidade inovadora	2,0
Apresentação final e documentação	1,0
Bônus: Hospedagem online	+1,0