

## Job Description

Buscamos o estado da arte no que se refere às práticas de desenvolvimento de software, nas quais incluem uma variedade de tecnologias. Valorizamos os candidatos que estão familiarizados com estas tecnologias, mas também estamos confiantes de que os engenheiros de software interessados em ingressar na Zup Innovation poderão aprender com nossa equipe.

- Microsserviços escaláveis, resilientes e performáticos escritos principalmente em Java, usando Spring Cloud e aproveitando técnicas de programação funcional e arquitetura hexagonal.
- APIs com contratos fortes e versionamento consistentes para o consumo em cenários de alto throughput e baixa latência.
- Sistemas de Mensageria com SQS, SNS e Kafka com padrões de arquitetura orientada a eventos.
- Containers Docker e Orquestração de containers
- Cloud Native e os principais serviços da AWS como ECS, lambdas, IAM, RDS, S3, CodeDeploy, Api Gateway, DynamoDB
- Continuous Integration e Continuous Deployment
- Monitoramento e Observability

Como Engenheiro de Software Senior nós esperamos que você:

- Desenvolva aplicações com qualidade, utilizando testes automatizados e as melhores práticas e design patterns de Java moderno (versões 11, 13, 17).
- Experiência com metodologias ágeis e processos de engenharia de software.
- Seja capaz de realizar code reviews com objetivo de melhoria contínua do produto e troca de conhecimento das pessoas.
- Tenha conhecimentos práticos de Cloud Computing e Arquitetura distribuída.
- Trabalhe de maneira colaborativa se expondo para elevar a régua das pessoas e melhorar os produtos de tecnologia.

Responda de 1 a 5 para as perguntas, sendo:

- 1) Não sei o que é
- 2) Conheço de maneira básica. Já implementou? Faz quantos anos que aprendeu sobre?
- 3) Conheço de maneira intermediária. Já implementou? Utiliza atualmente e Como? Faz quanto tempo que aprendeu sobre? Quais problemas já resolveu?
- 4) Conheço de maneira avançada. Quais problemas já resolveu? Soluções complexas que criou com esse conhecimento?
- 5) Tenho um conhecimento profundo e completo.

Cinco pilares das habilidades de uma pessoa engenheira/desenvolvedora de software:

- 1 - Desenvolvimento de software
- 2 - Engenharia de software
- 3 - Arquitetura de software
- 4 - Testes de software
- 5 - DevOps e Observabilidade

## **1 - Desenvolvimento de software**

### **1.1 - Java**

#### **1.2 - Orientação a Objetos.**

- 1.2.1 - Estrutura vs Comportamento
- 1.2.2 - Conhece sobre Herança, reescrita e polimorfismo
- 1.2.3 - Classes Abstratas
- 1.2.4 - Interfaces
- 1.2.5 - Composição de objetos
- 1.2.6 - Encapsulamento e visibilidade
- 1.2.7 - Construtores e membros estáticos

#### **1.3 - Exceções e controle de erros**

- 1.3.1 - Tratamento e lançamento de Exceptions
- 1.3.2 - Checked e Unchecked

#### **1.4 - Collections framework**

- 1.4.1 - Listas
- 1.4.2 - Sets e Iterators
- 1.4.3 - Equals e Hashcode
- 1.4.5 - Maps

#### **1.5 - Threads**

- 1.5.1 - Modelo de execução
- 1.5.2 - Coleções thread-safe
- 1.5.3 - Deadlock

#### **1.6 - Funcional**

- 1.6.1 - Lambdas
- 1.6.2 - Method references
- 1.6.3 - Streams e Collectors
- 1.6.4 - Map, Reduce, Filter
- 1.6.5 - Nova API de datas
- 1.6.6 - Optional
- 1.6.7 - Design Patterns

## **1.7 - Reflection**

- 1.7.1 - Anotações
- 1.7.2 - Generics
- 1.7.3 - Injeção de Dependência

## **1.8 - Reactive Streams**

- 1.8.1 - WebFlux

## **1.9 - JDBC, JPA e Hibernate**

## **2 - Spring**

### **2.1 - Spring Framework**

- 2.2 - Spring Web
- 2.3 - Spring Data
- 2.4 - Bean Validation
- 2.5 - Dependency Injection

### **2.2 - Spring Data JPA**

### **2.3 - Spring Cloud**

## **3 - APIs REST**

- 3.1 - HTTP
- 3.2 - URIs/URLs
- 3.3 - Métodos
- 3.4 - Payloads e JSON
- 3.5 - Headers
- 3.6 - Segurança - CORS e OAuth
- 3.7 - Idempotência
- 3.8 - HATEOAS
- 3.9 - Versionamento
- 3.10 - Documentação - Swagger e Open API
- 3.11 - API Gateway

## **4 - Mensageria**

4.1 - Pub/Sub (Kafka, SNS)

4.2 - Queues (SQS)

## **5 - Banco de dados**

5.1 - Relacional

5.2 - Não Relacional

## **5 - Testes Automatizados**

## **6 - SOLID**

## **7 - Design Patterns**

## **8 - Microserviços**