**Case técnico DevOps Junior – Desafio teórico.**

**1º Etapa:**

No desenvolvimento de microsserviços em cloud existe uma separação de cada serviço da aplicação em módulos que desempenham atividades específicas e não dependem de outros módulos para funcionar. Trazendo para a realidade do anotaai, podemos ter um exemplo com um serviço para o modo garçom, outro para o cardápio digital e um outro para os pedidos. Cada um destes serviços funciona de forma independente, com suas próprias rotas, banco de dados e sistema de logs, e caso necessário podem se comunicar de forma interna uns com os outros via requisições diretas ou por serviços de mensageria como o apache kafka. Uma das vantagens deste tipo de aplicação é que se um dos módulos não estiver respondendo corretamente, os outros continuam ativos e funcionando, e caso utilize serviços de mensageria, ao retornar o funcionamento o módulo pode ler as mensagens na fila para retornar a responder as requisições recebidas.

Este tipo de arquitetura facilita o desenvolvimento da aplicação pois permite isolar equipes específicas para cada serviço, permitindo uma rápida compreensão do projeto para possíveis novos integrantes da equipe. Também, por serem serviços menores, facilita o processo de implementação através de pipelines CI/CD pois o processo de build, teste e implantação se torna mais rápido.

Para o monitoramento da aplicação podemos utilizar padrões como o Log aggregation pattern, que permite centralizar todos os dados de logs dos serviços em uma base de dados, proporcionando um controle mais preciso dos logs e também permitindo a criação de alertas para determinados logs, e a visualização da aplicação por meio de dashboards que podem ser criados em plataformas como Grafana.

**2º Etapa:**

Atualmente, faço parte de um projeto voluntário como desenvolvedor backend. Considero o projeto em que estamos trabalhando no momento muito significativo na minha experiência, pois foi proposto o uso de uma stack que eu nunca havia utilizado: NestJS, TypeORM e um banco de dados PostgreSQL no Docker.   
  
No início do projeto, enfrentei alguns problemas com as configurações do banco de dados PostgreSQL que estava rodando no Docker, pois não tinha familiaridade com essa ferramenta e o TypeORM não conseguia fazer o mapeamento correto das entidades para o banco de dados. Após consultar a documentação do NestJS, alguns tutoriais no Medium e realizar um minicurso no YouTube, consegui resolver o erro e o projeto continuou.   
  
Apesar do desafio inicial, por já possuir experiência em desenvolvimento backend utilizando Java e Spring Boot, pude abstrair muitos dos conceitos do backend e trazer para a nova ferramenta, tendo uma curva de aprendizado rápida. Com o auxílio da documentação do framework NestJS, foi possível desenvolver as tarefas que me foram atribuídas e também auxiliar meus colegas de time, que possuíam menos experiência, com revisão dos códigos e uso do Git.