Neste projeto utilizamos **Microsserviços**, que é uma arquitetura de desenvolvimento que consiste em pequenos serviços independentes que se comunicam usando APIs bem definidas. Esses serviços pertencem a pequenas equipes autossuficientes.

Nesta classe usamos **Service Discovery** para descobrir dinamicamente a localização da rede (endereço IP e porta) de outro serviço para se comunicar com ele.  
O **Service Discovery** se conecta com o Config Server e busca o arquivo de configuração (service-discovery) no local: https://github.com/oswaldoneto/dio-experts-config/blob/master/service-discovery.yml que foi informado no arquivo application.yml da classe config-server.

Nele utilizamos o **Eureka** que é um serviço REST (Representational State Transfer), usado para que serviços sejam registrados através do Eureka Server e descobertos através do Eureka Client, facilitando esse controle de aplicação distribuída.

Utilizamos o **Elasticsearch** que é um mecanismo de busca e análise de dados distribuído que possui velocidade, escalabilidade e capacidade de indexar muitos tipos de conteúdo.

Utilizamos o **Redis**, que serve para armazenamento de estrutura de dados de chave-valor e oferece um conjunto de estruturas versáteis de dados na memória que permite a fácil criação de várias aplicações personalizadas.

Utilizamos o **ConfigServer** para distribuir configurações para todos os microsserviços da nossa aplicação.

