

Curso/Oferta: Arquitetura de Software Distribuído – Oferta 7

Disciplina: Arquitetura de Backend e Microserviços

Professor: Marco Aurélio S. Mendes

Data: 30/05/2017

Trabalho Final

Escopo - API de controle de pedidos entre lojistas e atacadistas.

Considere um cenário de negócio onde lojistas e atacadistas precisam realizar transações de compras de produtos.

O fluxo de negócio é exemplificado abaixo:

1. O lojista percebe que o seu estoque está baixo e que ele precisar comprar novos produtos.
2. O lojista realiza uma solicitação de pedidos para o seu atacadista preferido. Cada solicitação deve ter uma ou mais entradas com os seguintes campos:
 - Código do produto (número)
 - Quantidade de itens (número)
 - Observações (texto)
3. O atacadista recebe o seu pedido solicitações através de notificações.
4. O atacadista informa o orçamento (número) e a data de entrega (campo data) e envia uma notificação para o lojista.
5. O lojista recebe a notificação da proposta do orçamento e aceita ou rejeita a mesma.
6. Em caso de rejeição, ele envia uma notificação para o atacadista e o fluxo é finalizado
7. Em caso de aceitação, ele envia uma notificação de confirmação do pedido.
8. Um pedido aprovado terá três estados: *Solicitado*, *Em fabricação*, *Finalizado*, *Despachado*. Toda vez que o atacadista mudar o estado do pedido, o lojista deve ser sinalizado da mudança.

A partir desse fluxo de negócio, você deve prover uma implementação mínima de uma prova de conceito com o uso de APIs e microserviços.

A sua implementação não precisa prover mecanismos de segurança nem nenhum tipo de interface gráfica, mas deve ser minimamente funcional do ponto de vista da API e prover pelo menos dois microsserviços que operam em ambientes containerizados distintos (do lojista e atacadista)

A sua linguagem de implementação é livre (exemplos incluem PHP, C#, Java ou Node). Ao mesmo tempo, você deve garantir o uso dos seguintes elementos tecnológicos:

- Microsserviços que operem em ambientes autocontidos (ex. contêineres leves como o [ASP.NET](#) Core, Spring Boot, Service Fabric, Pivotal Cloud Foundry, Amazon API Gateway)
- Distribuição dos microsserviços em Docker
- Criação de uma API RESTful (nível 3 do modelo de maturidade de Richardson) para cada microsserviço
- Teste de unidade que exercite o fluxo de negócio do enunciado
- Teste de unidade que garanta que o microsserviço está disponível em produção
- Geração de documentação Swagger para as APIs

Opcionalmente, você pode incorporar distribuição em nuvem e a criação de algum microsserviço serverless para uma função acessória que você mesmo pode decidir. Neste caso, você pode escolher entre o Amazon Lambda, Azure Functions ou Google Functions.

Entregáveis do Trabalho

Os seus entregáveis são:

- Código Fonte no GitHub
- Arquivo de configuração docker para montar as imagens de cada microsserviço
- Testes de unidade automatizados.
- Documentação Swagger

Valor – 30 pontos

Data de entrega – 30 de Junho de 2017

Entrega – Usar o Edmodo com os links do GitHub e/ou Docker

Times - Pode ser realizado individualmente ou em duplas.