LADR

GitHub - peter-evans/lightweight-architecture-deci...

Template

- Title
- Context
- Decision
- Status [Proposed, Accepted, Deprecated, Superseded]
- Consequences

Arquitetura de processo

- Context: Qual arquitetura utilizar em termos de processos de Sistema Operacional (Monolito ou Microserviço)
- Decision: Adotamos arquitetura de microsserviços
- Status: Accepted
- Consequences:
  - Exige integrações entre microsserviços
  - Necessidade maior de gerenciamento, incluindo eventuais deploys, monitoramento e orquestração
  - Maior facilidade de escalabilidade granular

Tecnologia base back-end

- Context: Qual linguagen/plataforma base utilizar para servir requisições
- Decision: C#/ASP NET core
- Status: Accepted
- Consequences:
  - Maior facilidade de integração com Azure, Microsoft
  - Suporte da Microsoft por vários anos

- Familiaridade do time com OO

Abordagem transacional para confiabilidade

- Context: Devido a natureza do sistema - compra de produtos - se faz necessário confiabilidade transacional entre operações, mais voltada a parte de pedidos
- Decision: Decidimos adotar o padrão transacional distribuído SAGA, com a implementação tecnológica em C#, MassTransit
- Status: Accepted
- Consequences:
  - Entender o funcionamento do MassTransit, já que não temos tão conhecimento
  - Todo a gestão de transacionalidade fica com o framework, o que por um lado dificulta a depuração, mas facilita a gestão

---

Qual arquitetura distribuída de mensagens utilizar para Holder de pedidos

Context: Tendo em vista que utilizamos o padrão SAGA para transações, se faz necessário uma tecnologia pra suportar essas mensagens que carregarão as requisições de pedidos

Decision: Decidimos utilizar mensageria, tendo em vista a sua vantagem de permitir mensagens serem consumidas via a necessidade permitindo a microsserviços escalarem sem gerenciamento direto dos desenvolvedores para trackear as mensagens

Status: Accepted

Consequences:

Devemos decidir se usaremos algum framework para gerenciar o broker ou faremos tudo na mão

---

Qual arquitetura distribuída de mensagens utilizar para Holder de pedidos

- Context: Tendo em vista que utilizamos o padrão SAGA para transações, se faz necessário uma tecnologia pra suportar essas mensagens que carregarão as requisições de pedidos

- Decision: Decidimos utilizar mensageria, tendo em vista a sua vantagem de permitir mensagens serem consumidas via a necessidade permitindo a microsserviços escalarem sem gerenciamento direto dos desenvolvedores para trackear as mensagens
- Status: Accepted
- Consequences:
    - Devemos decidir se usaremos algum framework para gerenciar o broker ou faremos tudo na mão

---

Gerenciar o broker de mensageria diretamente ou usar um framework?

- Context: Tendo em vista que utilizaremos mensageria, é necessário decidir se iremos gerenciar (criar, deletar, etc) as filas, ou se delegaremos o gerenciamento para um framework
- Decision: Decidimos usar um framework. Utilizaremos o MassTransit, tendo em vista que ele oferece suporte ao padrão SAGA também
- Status: Accepted
- Consequences:
    - Menos gerenciamento das filas
    - Maior facilidade na utilização de mensagens, tendo em vista o apoio de criação de tipos criados diretamente pelo framework, facilitando a comunicação, bastando seguir o mesmo padrão nos serviços que utilizarão as mensagens

---

Qual arquitetura usar para módulos nos serviços

- Context: Se faz necessário, para padronizar o projeto e facilitar a comunicação entre desenvolvedores, a escolha de uma arquitetura para organização dos módulos e relacionamento entre componentes
- Decision: Decidimos usar o Clean Architecture
- Status: Accepted
- Consequences:
    - Complexidade maior devido a configuração das camadas
    - Menor acoplação entre componentes que estão em diferentes camadas, facilitando mudanças futuras