

MBA⁺

Arquitetura Microserviços / Api e Digitalops

Diogo Borges - 354103
Eric Vieira - 356996
Tadeu Tanabe - 355034
Vinicius Dantas – 353546
Diego Farias – 355364
Erick Fonseca - 355898

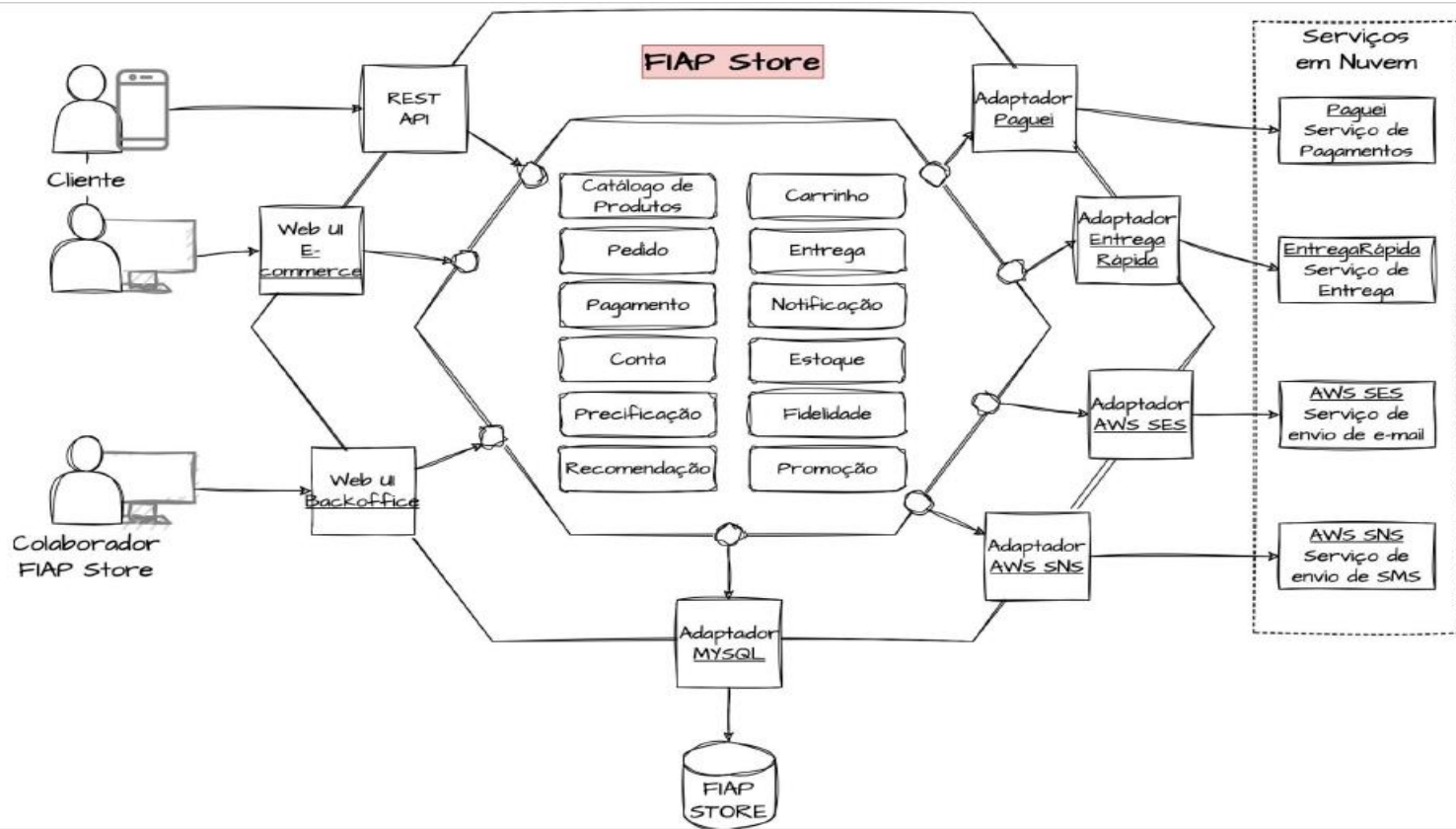
A FiapStore possui uma aplicação com arquitetura monolítica o que impacta diretamente no time to market, perdendo o diferencial competitivo e também a evolução tecnológica do parque.

Modernizar as arquiteturas da aplicação

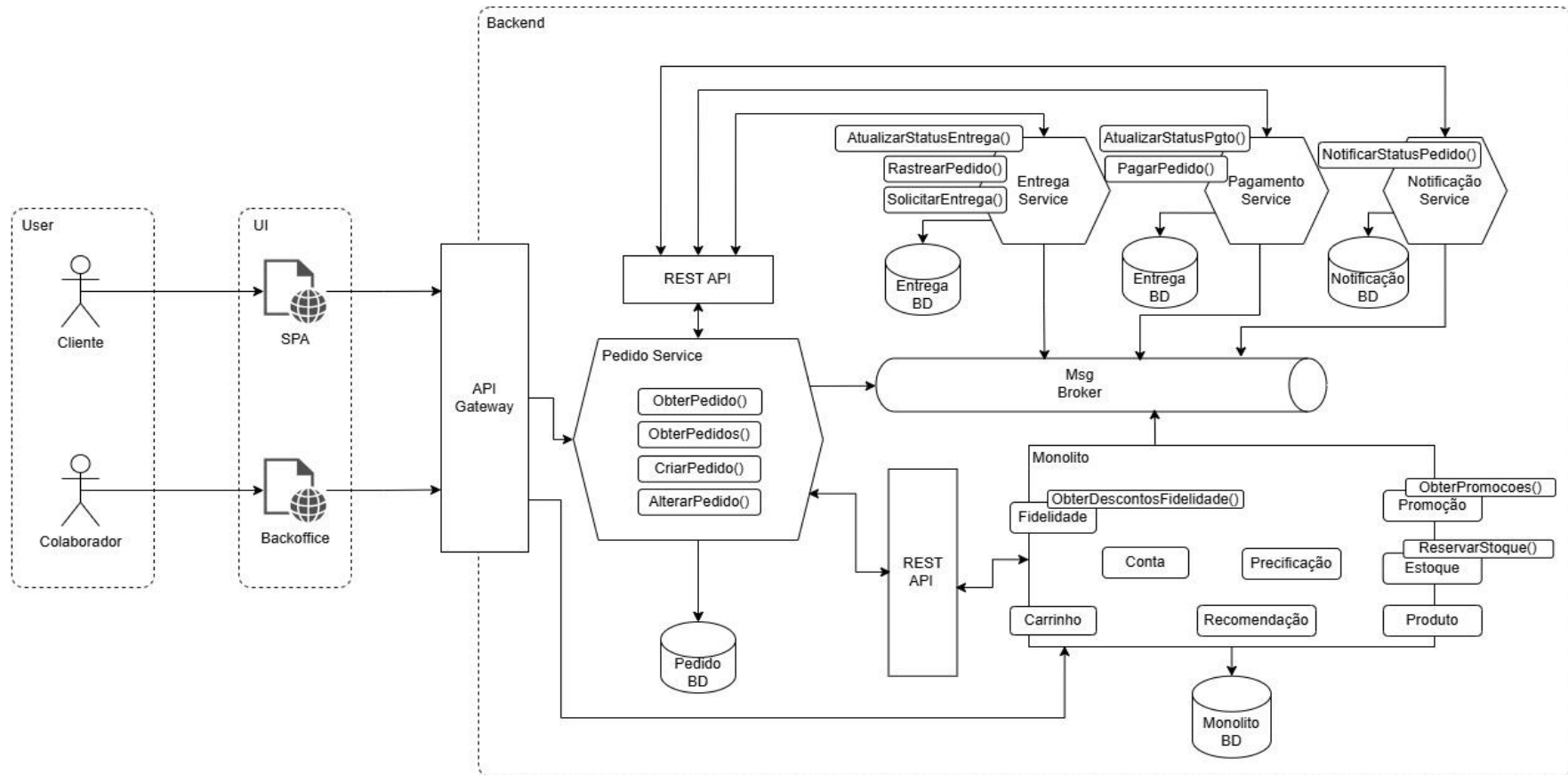
Migrar WebUI para SPA

Criar microsserviços para o módulo de pedido

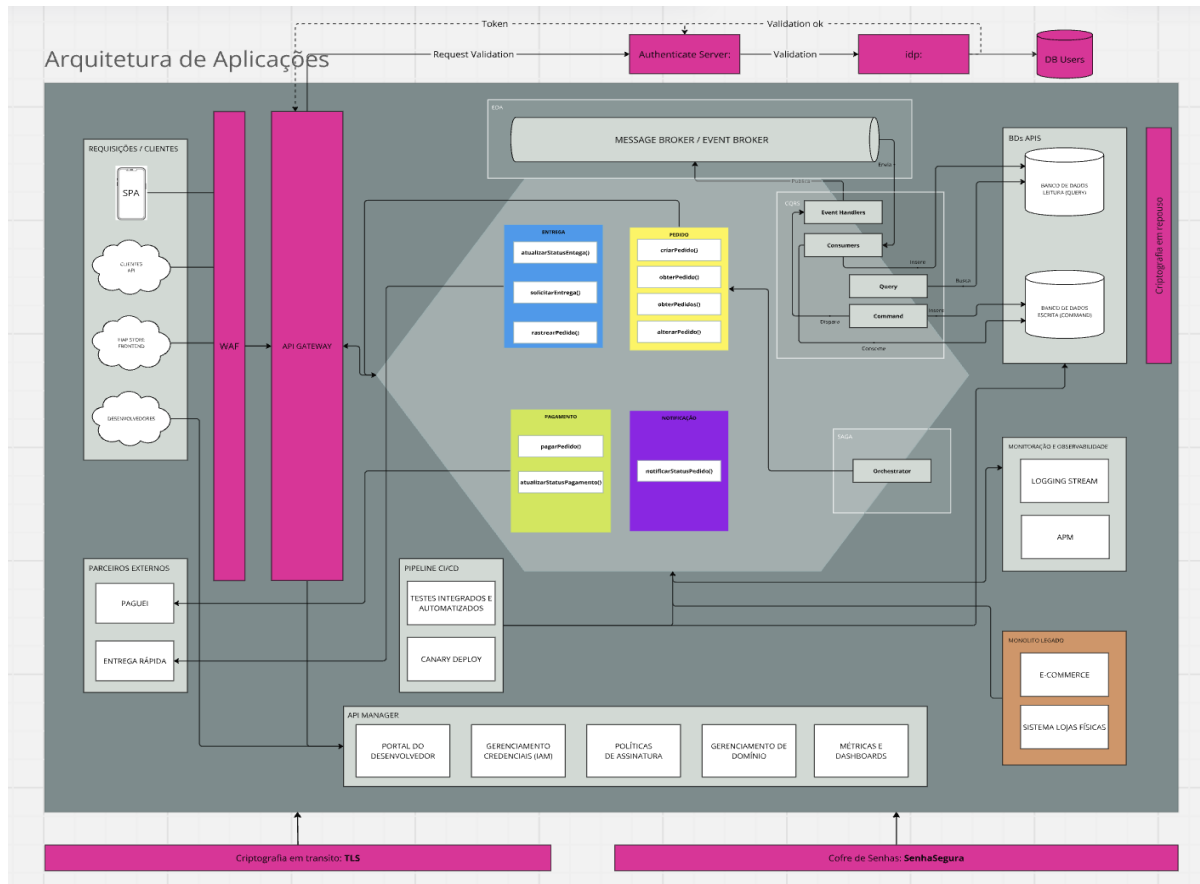
Arquitetura de Aplicação AS-IS



Arquitetura de Transição

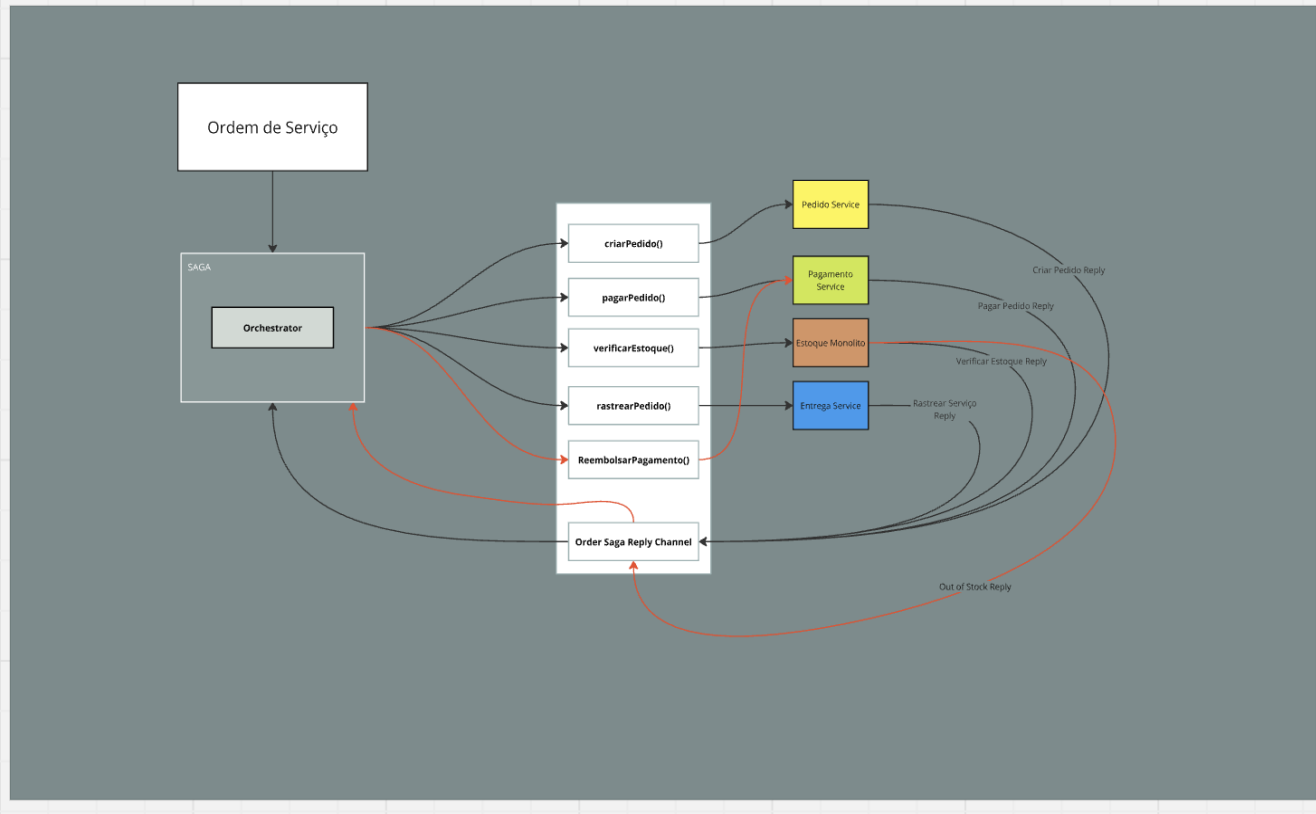


Arquitetura de Aplicação TO-BE



Arquitetura de Aplicação / Zoom Saga

Zoom SAGA



Saga Orquestrador

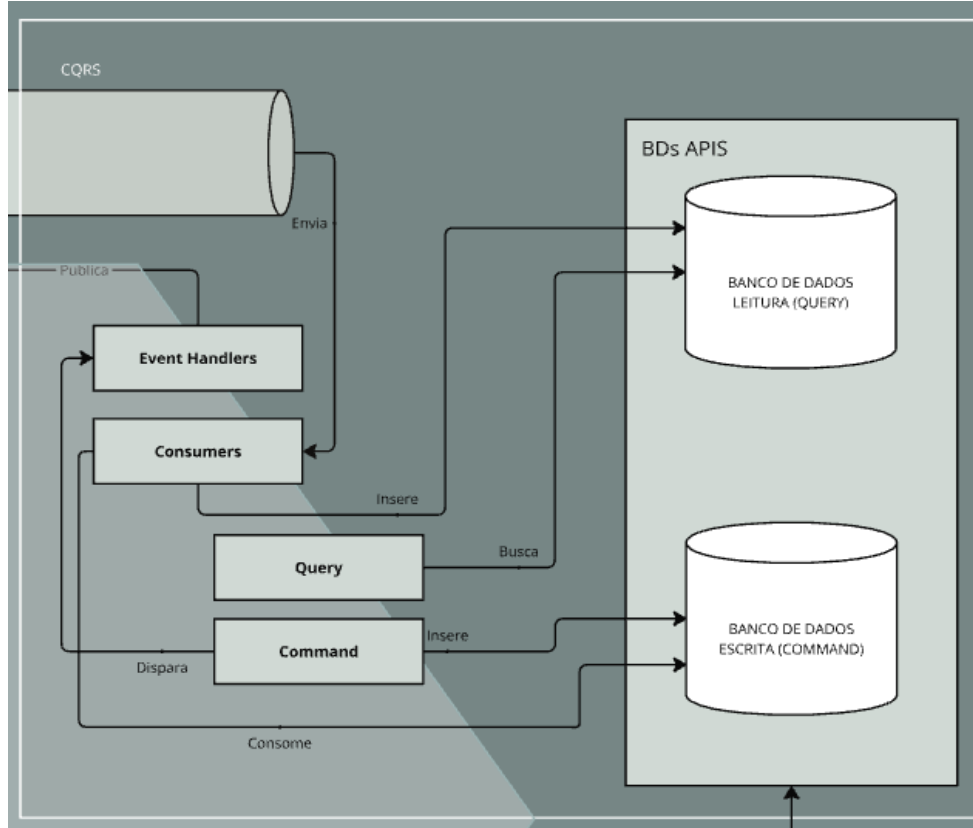
Gerencia transações distribuídas garantindo consistência em sistemas distribuídos. Ele coordena uma sequência de etapas compensáveis, garantindo que, se uma falhar, ações de compensação sejam executadas para manter a integridade dos dados. Isso é útil para evitar inconsistências em microservices.



API Manager

Gerencia o ciclo de vida de APIs, controlando acesso, segurança, monitoramento e desempenho. Ele facilita a exposição, consumo e governança de APIs em um ecossistema digital. Isso garante escalabilidade, segurança e controle sobre integrações entre sistemas.

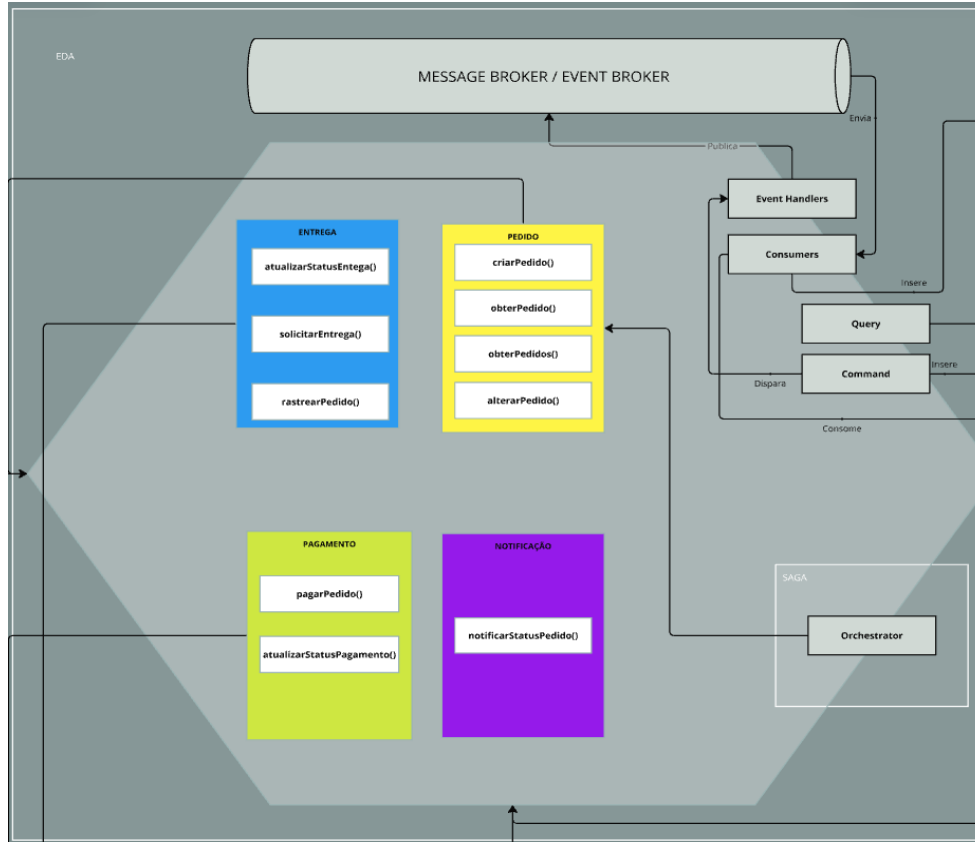
Arquitetura de Aplicação



CQRS - Command Query Responsibility Segregation

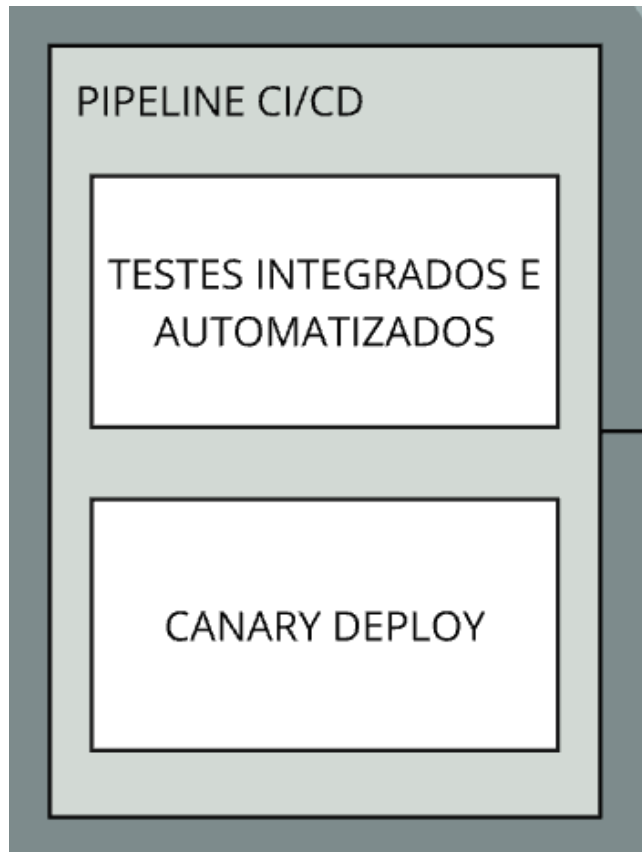
Complementa naturalmente a separação entre produção e consumo de eventos. Os "commands" representam intenções de alterar o estado do sistema e podem resultar na geração de eventos que refletem essas mudanças. Já as "queries" operam em modelos de dados otimizados para leitura, que podem ser atualizados com base nos eventos gerados pelas operações de escrita, dessa forma, evitando que tenhamos problemas de performance utilizando essa segregação.

Arquitetura de Aplicação



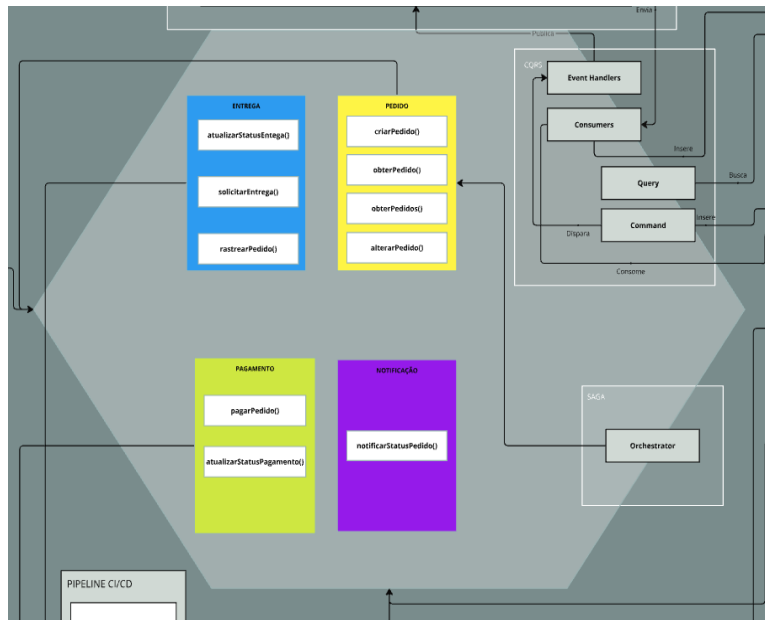
EDA

É um paradigma de design de software que enfoca a produção, detecção, consumo e reação a eventos. Adequado para ambientes distribuídos, dinâmicos e assíncronos. Permite resposta ágil a mudanças de estado ou ações específicas.



Pipeline CI/CD

Automatiza a integração e entrega contínua de software. Ele passa por etapas como build, testes e deploy, garantindo que novas alterações sejam verificadas e implantadas de forma segura e eficiente. Isso acelera o desenvolvimento e reduz erros em produção, gerando um time to market bem menor.



Arquitetura de MicroServiços

A partir da dificuldade apresentada, foi decidido uma solução utilizando uma Arquitetura de Microserviços em um provedor Cloud, onde estrangulamos os módulos de Pedidos e relacionados, aplicando as melhores práticas de arquitetura, como EDA, CQRS, SAGA, além de tecnologias apoiando a centralização de serviços (API Manager), Monitoração e Observabilidade, além de um fluxo de Pipelines CI/CD, garantindo um time to market muito mais competitivo.

Arquitetura de Segurança



SHIELD: Mitigação à ataques na camada 7 protegendo o banco de dados contra SQL Injections e ataques a disponibilidade com DDoS e pode criar e aplicar regras de mitigação para bloquear o tráfego malicioso.

API Gateway: Garante o controle de requisições externas, validando solicitação de acessos, log, controle de requisições.

Criptografia in Rest: Para os ambientes de produção é necessário que haja a criptografia AES 256 dos dados (principalmente se houver dados pessoais e ou sensíveis) garantindo a confidencialidade dos dados caso sejam vazados.

Criptografia em Transito: É necessária para evitar que os dados que estejam sendo trafegados dentro da rede tenham sua confidencialidade e para atendimento de regulamentações.

Cofre de Senhas: Nos garante a confidencialidade das credenciais usadas pelas aplicações e serviços e a guarda das chaves de criptografia, usando criptografia forte e rotação de senha.

IDP: Porque é necessário que tenhamos um gerenciamento e controle dops acessos a toda nossa infraestrutura de rede.

Authentication Server: Nos garante que não haja o acesso direto ao provedor de identidades servindo como um intermediário de liberações de acesso.

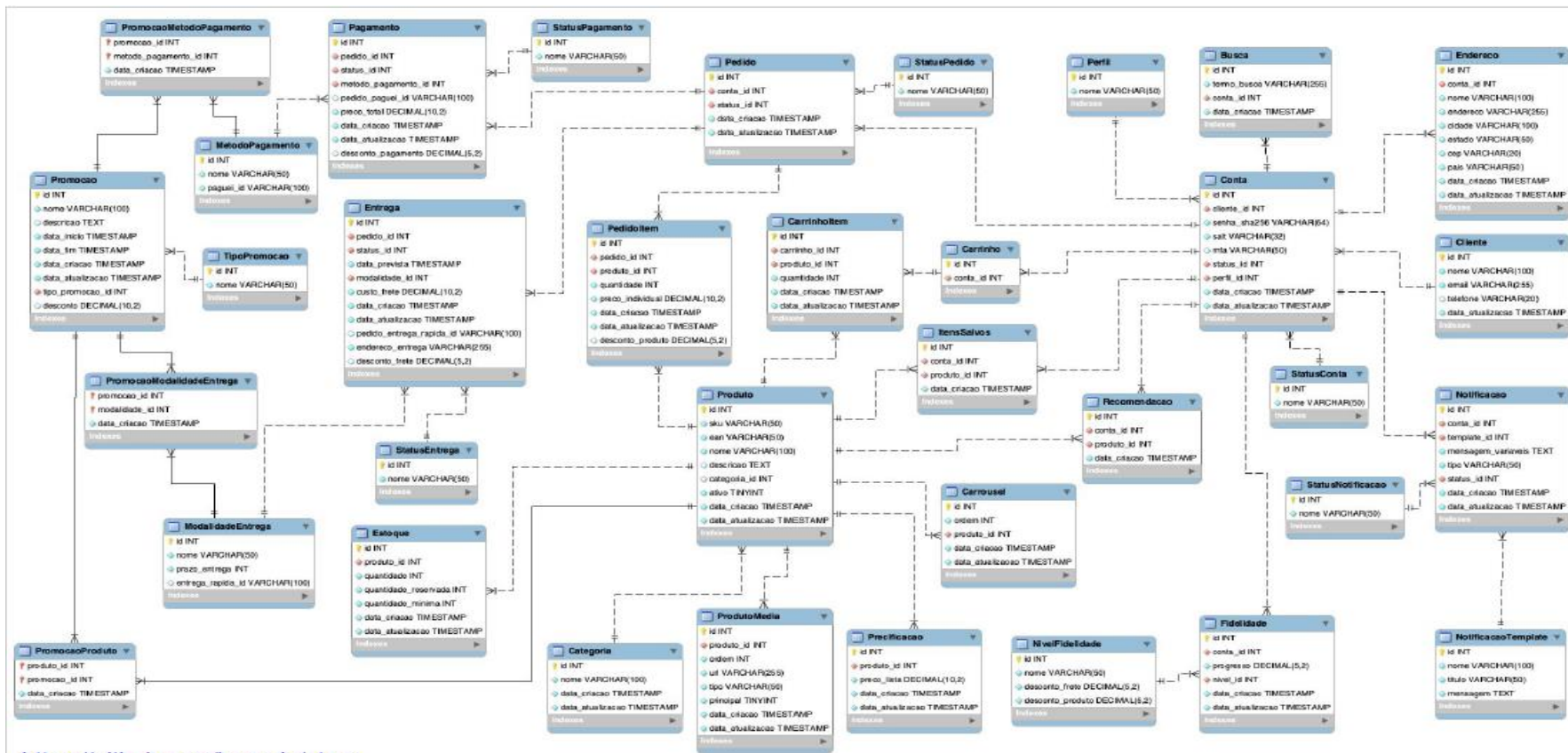
Monitoração e observância: Nos ajudam a identificar o mal funcionamento das aplicações, sobrecarga de recursos, obter rastreabilidade e auditoria sobre as ações dentro do sistema.

A arquitetura de dados atual foi construída usando banco de dados relacional que continha todas as entidades do monolito.

Com a modernização da FiapStore, os microserviços terão seus próprios bancos de dados.

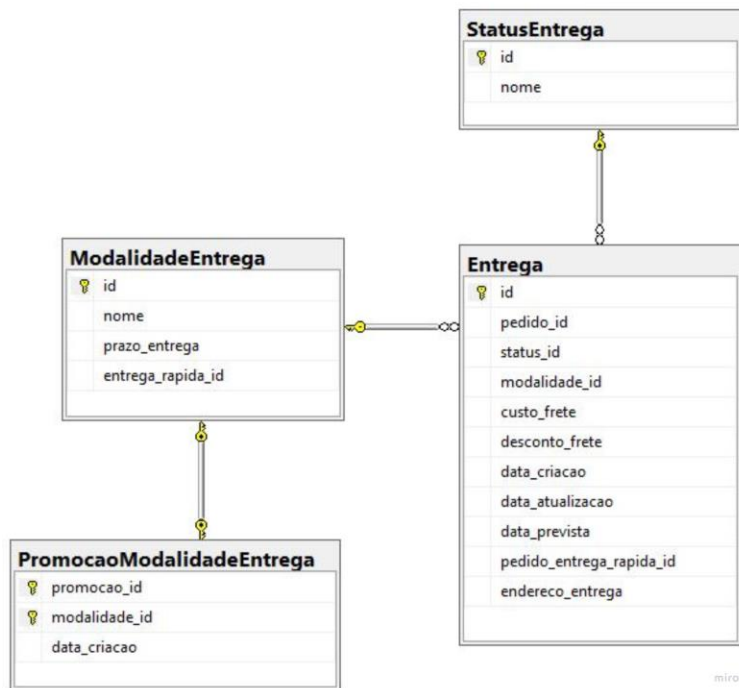
Na arquitetura de transição os demais módulos continuam no monolito.

Arquitetura de Dados Relacional AS-IS

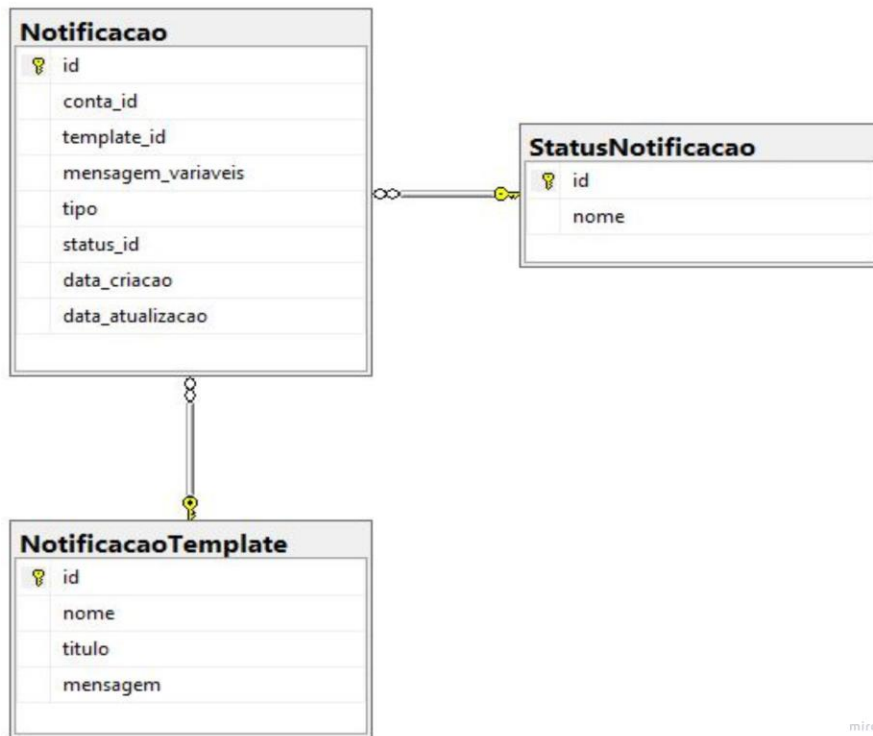


Arquitetura de Dados / Micro Serviço Entrega TO-BE

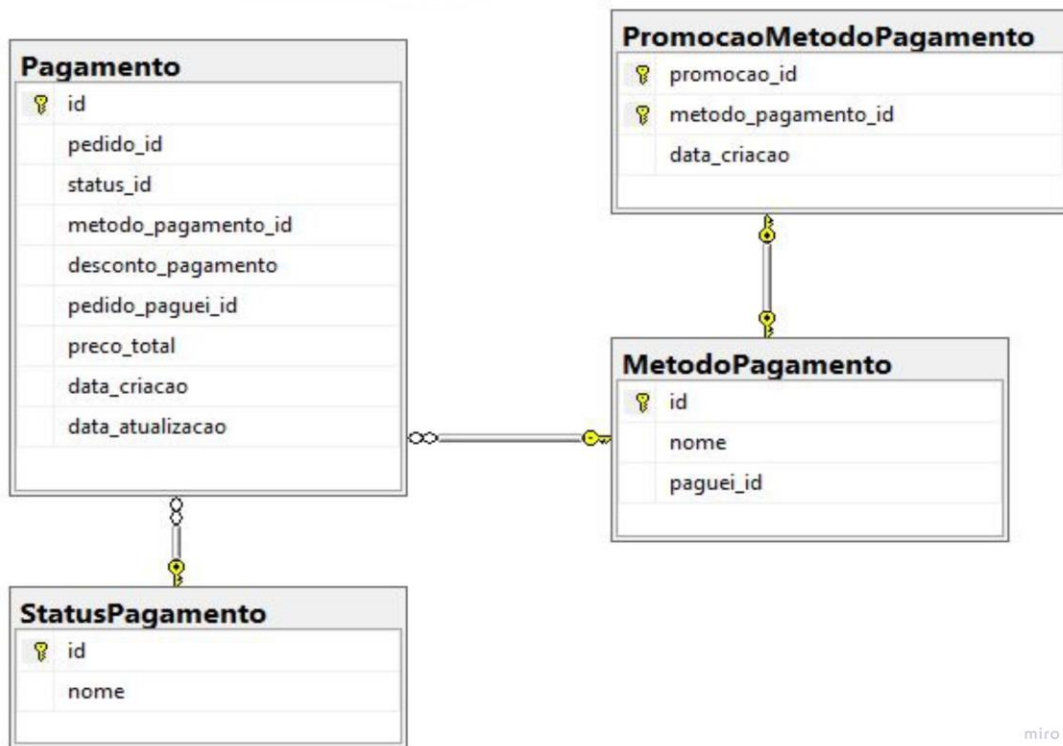
FIAP



Arquitetura de Dados / Micro Serviço Notificação TO-BE

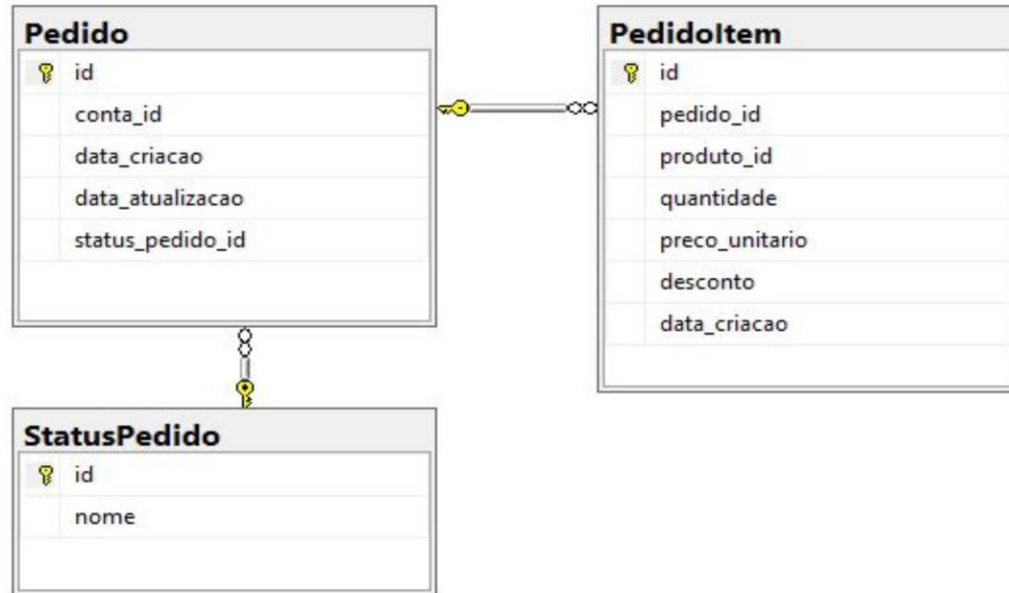


Arquitetura de Dados / Micro Serviço Pagamento TO-BE

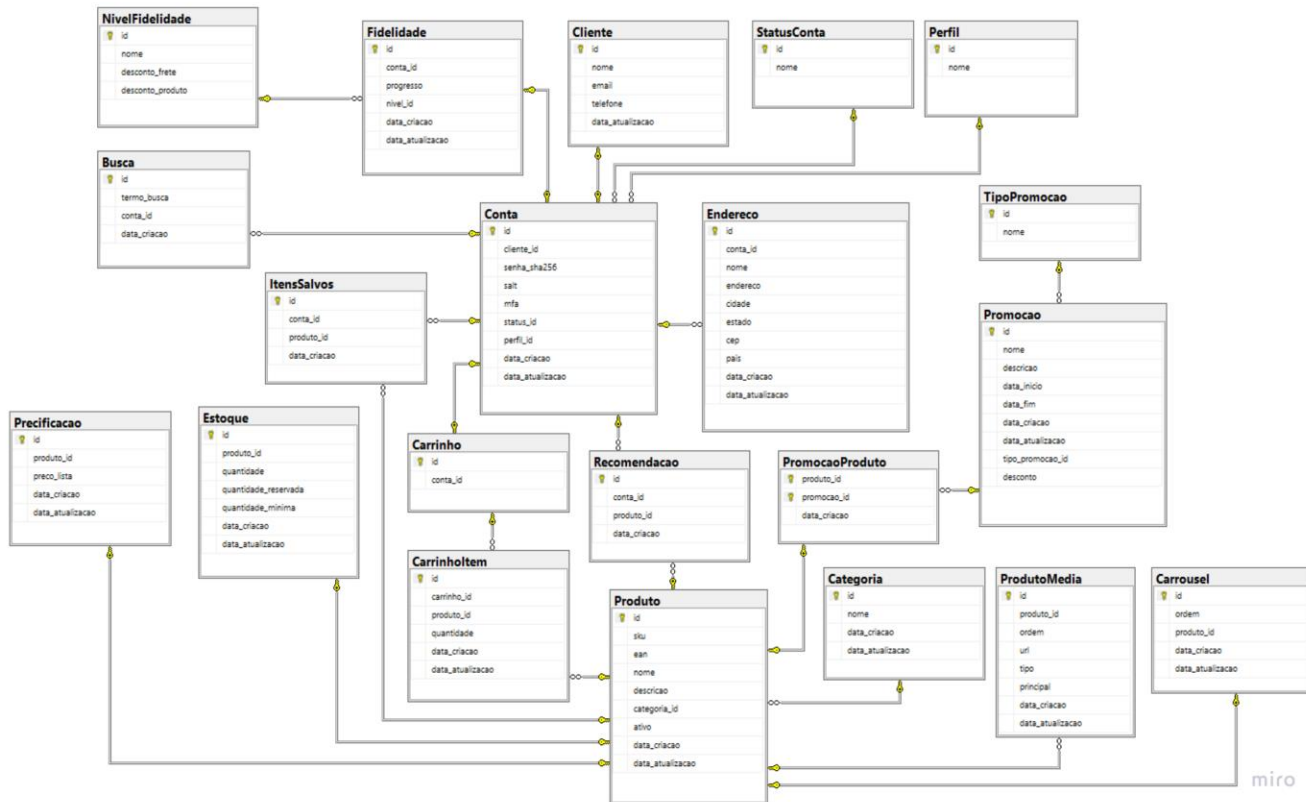


Arquitetura de Dados / Micro Serviço

Pedido TO-BE



Arquitetura de Dados / Monolito TO-BE



- A arquitetura de tecnologia atual do monolito, utiliza tecnologias defasadas e isso impacta diretamente em mão de obra especializada e senioridade maior.
- A proposta da modernização é utilizar novas tecnologias tais como: computação em nuvem, ferramentas de segurança, esteira CI/CD, ferramentas de APM para garantir o monitoramento e observabilidade da saúde da aplicação entre outras.

Arquitetura de Tecnologia TO-BE

FIAP

