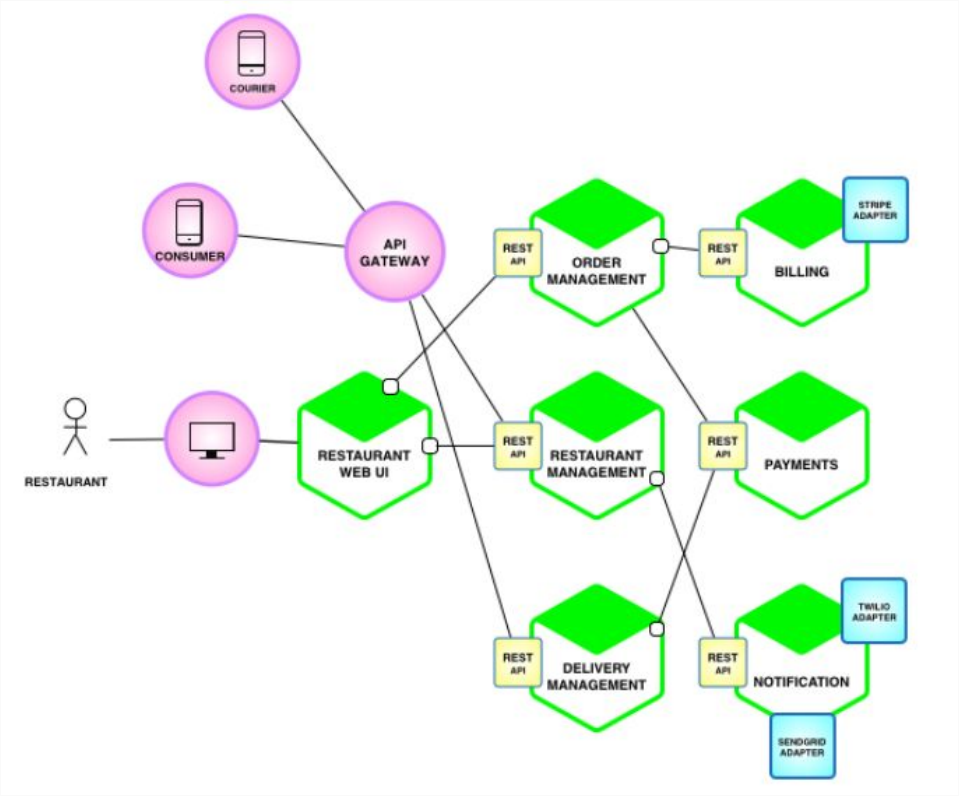
INSTITUTO FEDERAL
Triângulo Mineiro
Campus Uberlândia Centro

Projeto Backend Microsserviços e NoSQL

Decompondo Aplicações

Prof. Lucas Montanheiro
lucasmontanheiro@iftm.edu.br

Exemplo de Arquitetura de Microsserviços

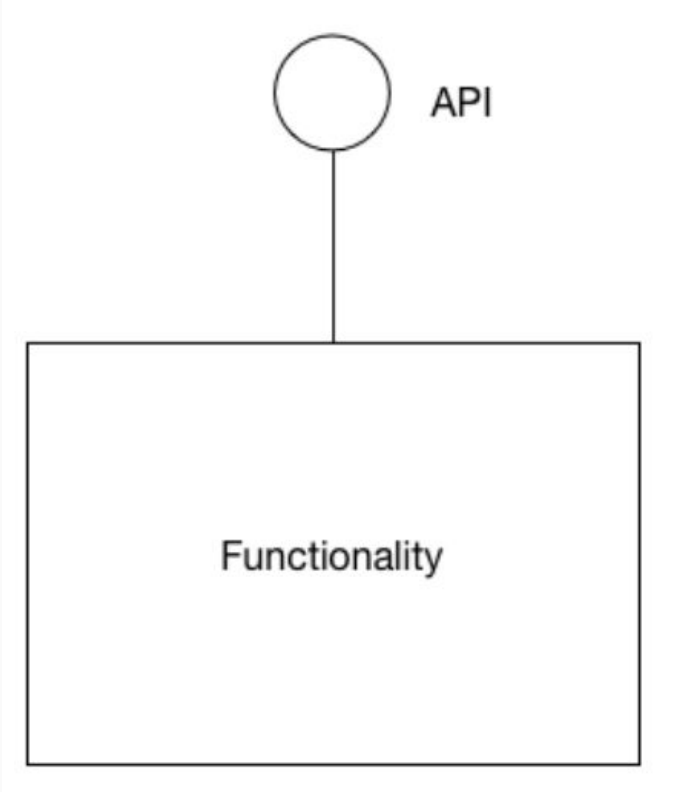


O que é um Serviço?

“Um mecanismo para permitir acesso a uma ou mais capacidades, onde o acesso é fornecido usando uma interface prescrita e é exercido consistentemente com restrições e políticas conforme especificado pela descrição do serviço”

OASIS - https://en.wikipedia.org/wiki/Service_(systems_architecture)

Visão Abstrata de um Serviço



Serviços Fracamente Acoplados

- Numa arquitetura de microsserviços, o serviços são fracamente acoplados;
- Toda interação com um serviço é feita via API, que encapsula os detalhes de implementação;
- Isto permite a execução de mudanças no serviço sem impactar seus clientes;

Serviços Fracamente Acoplados

- Serviços fracamente acoplados são a chave para melhorar atributos que impactam no tempo de desenvolvimento, como manutenibilidade e testabilidade;
- Serviços fracamente acoplados e pequenos são mais fáceis de entender, mudar e testar.

Serviços Fracamente Acoplados

- Cada serviço deve ser pequeno o suficiente para ser desenvolvido por uma equipe de “duas pizzas”, ou seja, uma equipe de 6 a 10 pessoas;
- Cada equipe que possui um ou mais serviços deve ser autônoma. Uma equipe deve ser capaz de desenvolver e implementar seus serviços com colaboração mínima com outras equipes.

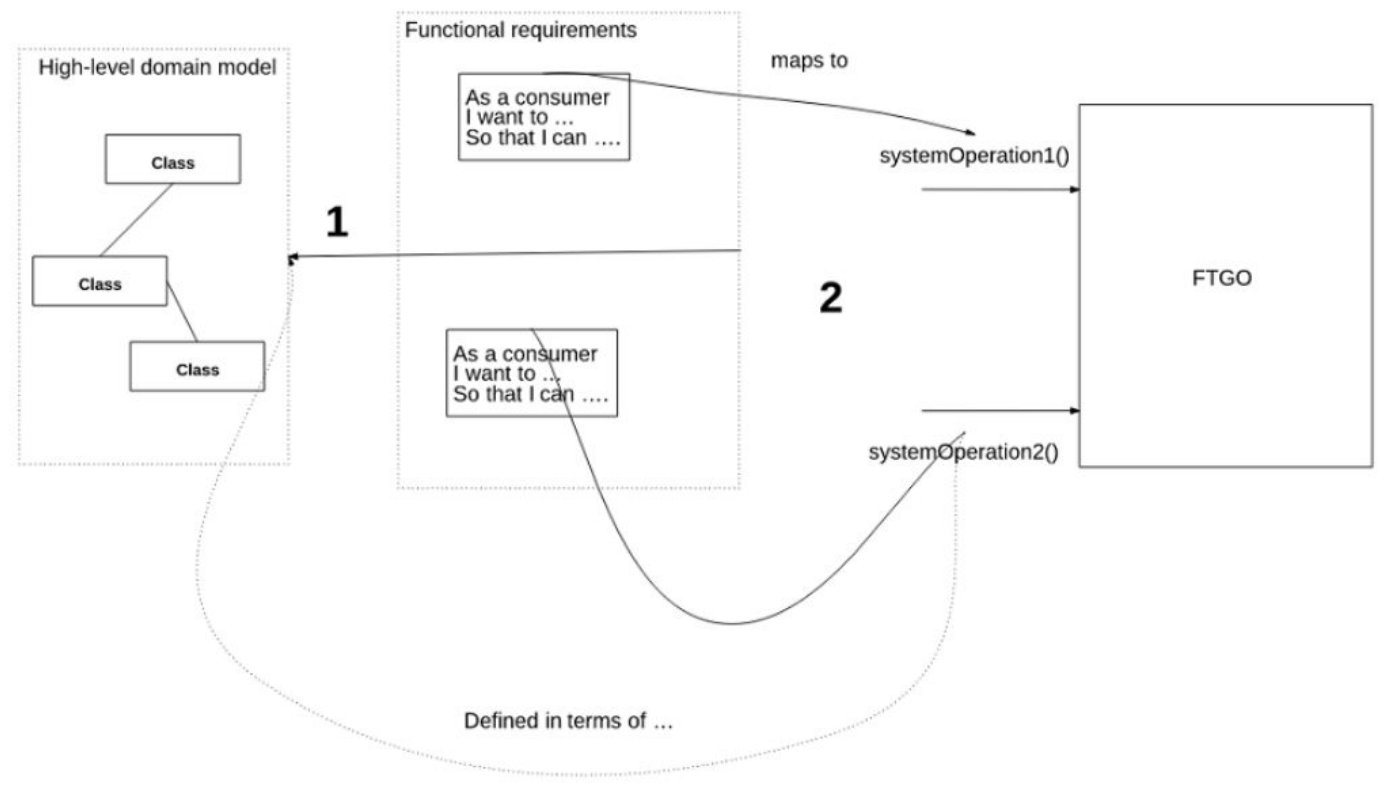
Identificando Operações do Sistema

- O primeiro passo da definição da arquitetura da aplicação é definir as operações do sistema, através da interpretação dos requisitos da aplicação;
- As operações do sistema são identificadas e definidas usando um processo em duas etapas, inspirado pelo processo de design orientado a objetos descrito por Craig Larman's no livro "*Applying UML and Patterns*"

Identificando Operações do Sistema

- O primeiro passo é criar um modelo do domínio consistindo das classes principais, o que irá prover um vocabulário com o que será possível descrever as operações do sistema.
- O segundo passo é identificar as operações do sistema e descrever o comportamento de cada um em termos de modelo de domínio.

Identificando Operações do Sistema



Criando um Modelo de Domínio de Alto Nível

Pedido Solicitado

Dado um cliente

E um restaurante

Quando o cliente solicita um pedido para o restaurante com um endereço de entrega que poderá ser servidor por tal restaurante

Então o cartão de crédito do cliente é autorizado

E um pedido é criado no estado “ACEITAÇÃO PENDENTE”

E o pedido é associado ao cliente

E o pedido é associado ao restaurante

Pedido Aceito

Dado um pedido ou um restaurante que está esperando ser aceito

E um correio que está disponível para entregar o pedido

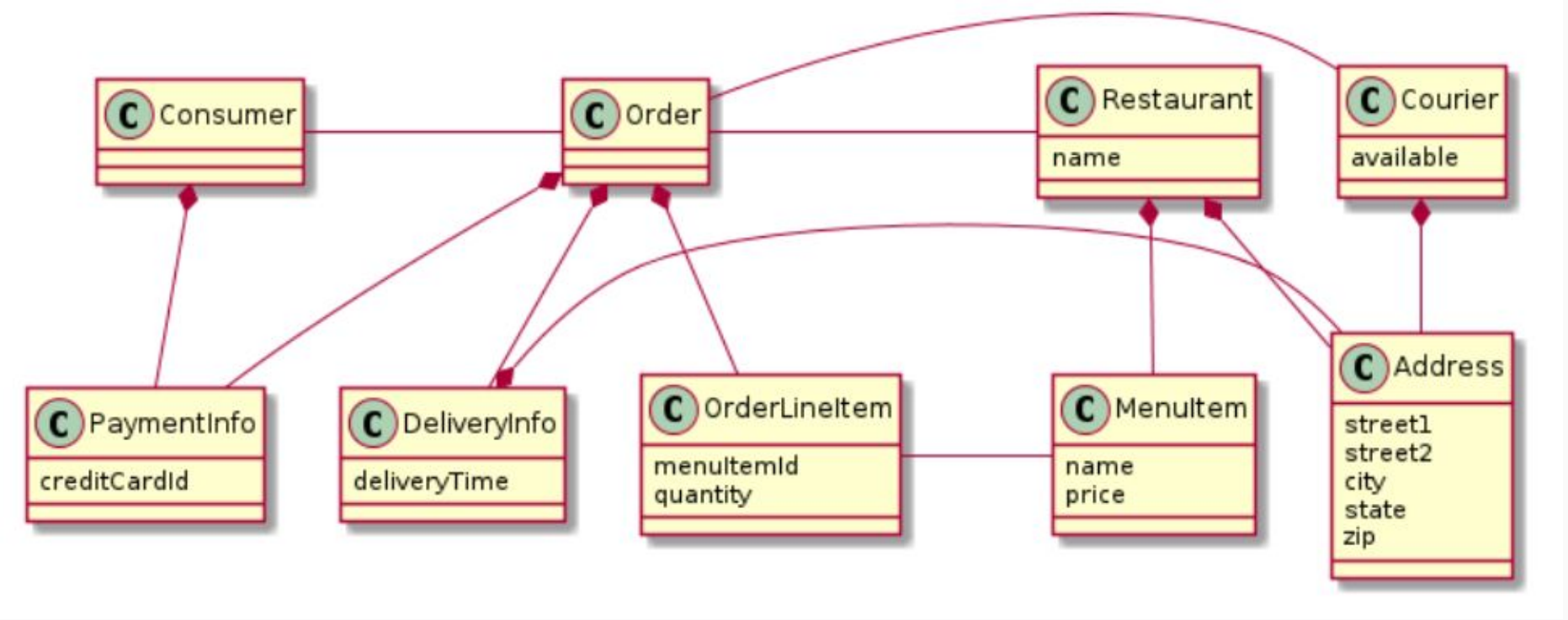
Quando um restaurante aceita um pedido com a promessa de prepará-lo com um tempo estimado de entrega

Então o estado do pedido é mudado para “ACEITO"

E o tempo estimado de entrega do pedido é atualizado

E o correio é designado para entregar o pedido

Criando um Modelo de Domínio de Alto Nível



Definindo as Operações do Sistema

Actor	Story	Command	Description
Consumer	Create Order	createOrder()	creates an order
Restaurant	Accept Order	acceptOrder()	indicates that the restaurant has accepted the order and is committed to preparing it by the indicated time
Restaurant	Order Ready for Pickup	noteOrderReadyForPickup()	indicates that the order is ready for pickup
Courier	Update Location	noteUpdatedLocation()	updates the current location of the courier
Courier	Order picked up	noteOrderPickedUp()	indicates that the courier has picked up the order
Courier	Order delivered	noteOrderDelivered()	indicates that the courier has delivered the order

Estratégia 1: Decompor por regra de negócio

- Defina serviços correspondentes a capacidades de negócios.
- Uma capacidade de negócios é um conceito da modelagem de arquitetura de negócios.
- É algo que um negócio faz para gerar valor.
- Uma capacidade de negócios geralmente corresponde a um objeto de negócios, por exemplo:
  - O gerenciamento de pedidos é responsável pelos pedidos
  - A Gestão de Clientes é responsável pelos clientes

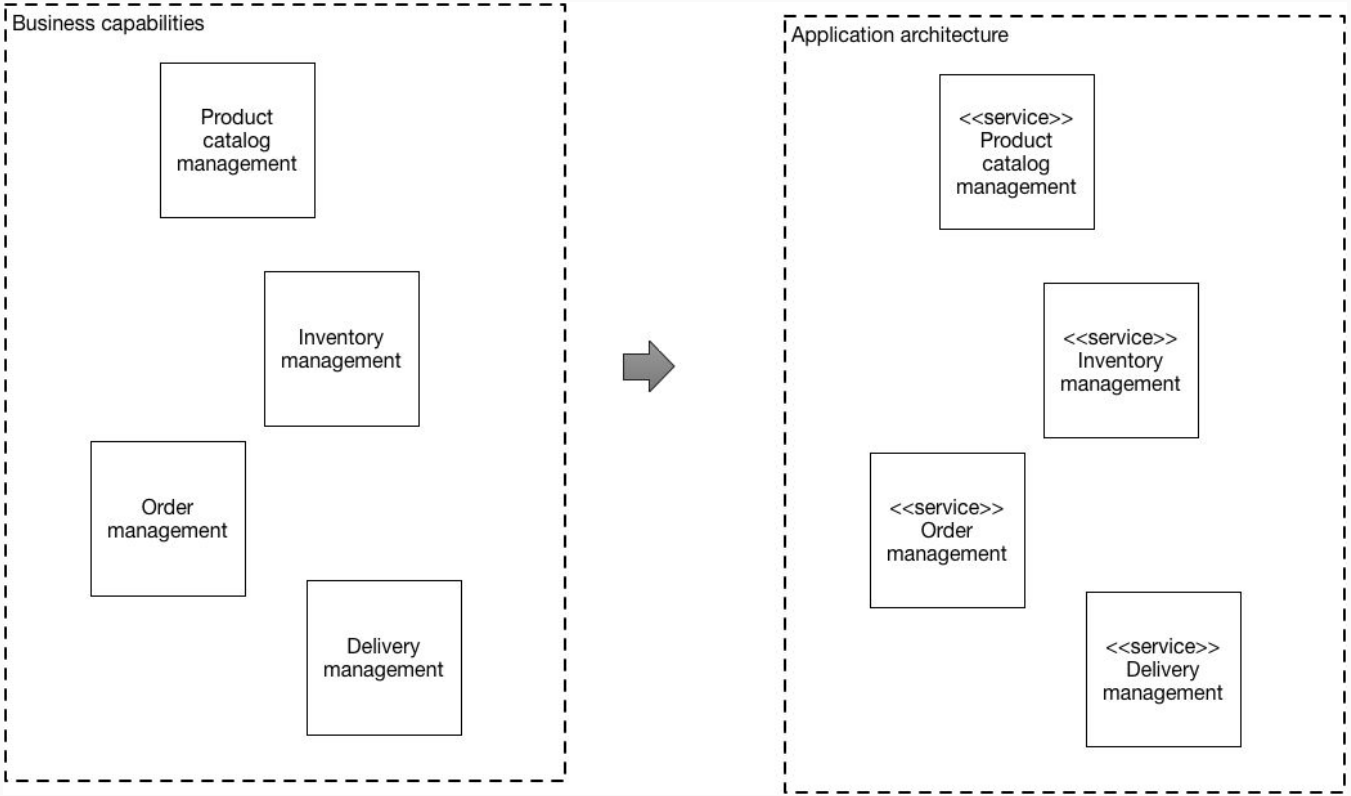
Estratégia 1: Decompor por regra de negócio

- As capacidades de negócios são frequentemente organizadas em uma hierarquia multinível.
- Por exemplo, um aplicativo empresarial pode ter categorias de nível superior:
  - Desenvolvimento de produto/serviço
  - Entrega de produto/serviço
  - Geração de demanda
  - etc.

Estratégia 1: Decompor por regra de negócio

- Os recursos comerciais de uma loja online incluem:
  - Gestão de catálogo de produtos
  - Gestão de inventário
  - Gestão de pedidos
  - Gestão de entregas

Estratégia 1: Decompor por regra de negócio



Estratégia 1: Decompor por regra de negócio

Este padrão tem os seguintes benefícios:

- Arquitetura estável, pois as capacidades de negócios são relativamente estáveis;
- As equipes de desenvolvimento são multifuncionais, autônomas e organizadas em torno da entrega de valor comercial em vez de recursos técnicos;
- Os serviços são coesos e fracamente acoplados.

Estratégia 1: Decompor por regra de negócio

Problemas:

- Como identificar capacidades de negócios?
- Identificar capacidades de negócios e, portanto, serviços requer uma compreensão do negócio.
- As capacidades de negócios de uma organização são identificadas pela análise do propósito, estrutura, processos de negócios e áreas de especialização da organização.
- Contextos limitados são melhor identificados usando um processo iterativo.

Estratégia 1: Decompor por regra de negócio

- Bons pontos de partida para identificar capacidades de negócios são:
  - estrutura organizacional - diferentes grupos dentro de uma organização podem corresponder a capacidades de negócios ou grupos de capacidades de negócios.
  - modelo de domínio de alto nível - as capacidades de negócios geralmente correspondem a objetos de domínio.

Estratégia 2: Decompor por subdominío

- Defina serviços correspondentes a subdomínios Domain-Driven Design (DDD).
- DDD se refere ao espaço problemático do aplicativo - o negócio - como o domínio.
- Um domínio consiste em vários subdomínios. Cada subdomínio corresponde a uma parte diferente do negócio.

Estratégia 2: Decompor por subdominío

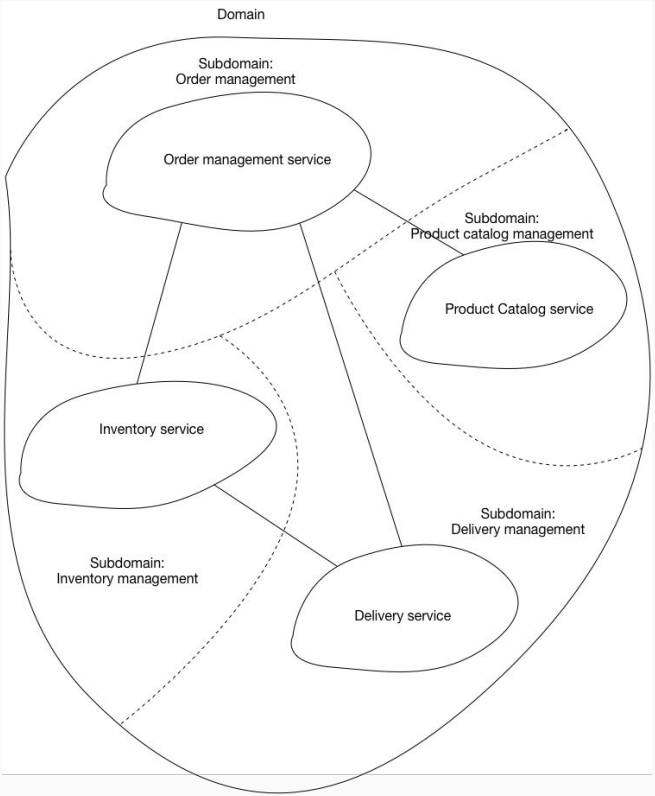
- Os subdomínios podem ser classificados da seguinte forma:
  - Núcleo - principal diferencial para o negócio e a parte mais valiosa do aplicativo
  - Suporte - relacionado ao que o negócio faz, mas não é um diferenciador. Podem ser implementados internamente ou terceirizados.
  - Genéricos - não são específicos para o negócio e são idealmente implementados usando software pronto para uso

Um pouco mais de DDD

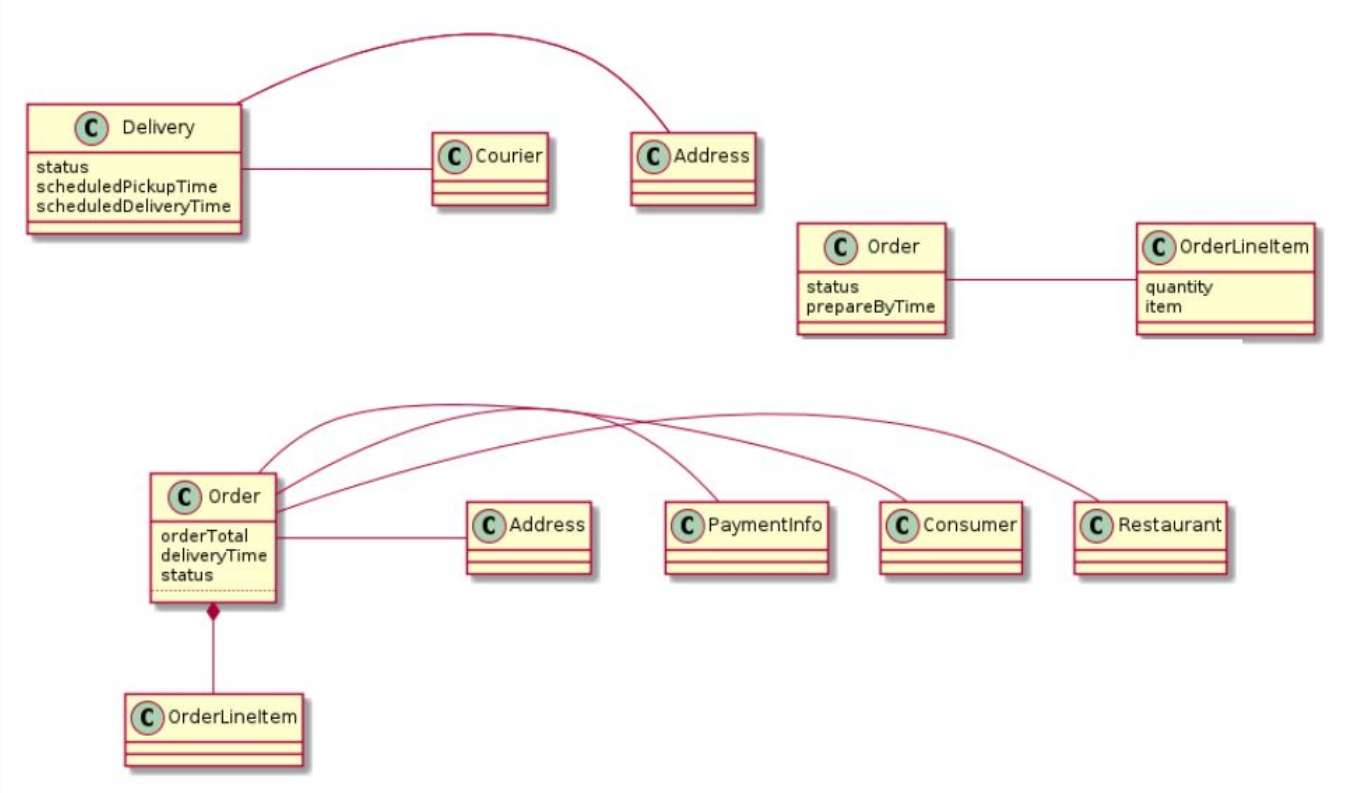
Estratégia 2: Decompor por subdominío

- Os subdomínios de uma loja online incluem:
  - Catálogo de produtos
  - Gestão de inventário
  - Gestão de pedidos
  - Gestão de entregas
  - ...

Estratégia 2: Decompor por subdominío



Estratégia 2: Decompor por subdomínio



Estratégia 2: Decompor por subdominío

Este padrão tem os seguintes benefícios:

- Arquitetura estável, pois os subdomínios são relativamente estáveis
- As equipes de desenvolvimento são multifuncionais, autônomas e organizadas em torno da entrega de valor comercial em vez de recursos técnicos
- Os serviços são coesos e fracamente acoplados

Estratégia 2: Decompor por subdominío

Problemas:

- Como identificar os subdomínios?
- Identificar subdomínios e, portanto, serviços requer uma compreensão do negócio.
- Assim como as capacidades do negócio, os subdomínios são identificados pela análise do negócio e sua estrutura organizacional e pela identificação das diferentes áreas de especialização.
- Os subdomínios são melhor identificados usando um processo iterativo.

Estratégia 2: Decompor por subdominío

- Bons pontos de partida para identificar subdomínios são:
  - estrutura organizacional - diferentes grupos dentro de uma organização podem corresponder a subdomínios
  - modelo de domínio de alto nível - subdomínios geralmente têm um objeto de domínio chave