Mergulho Microserviços Spring – AlgaWorks

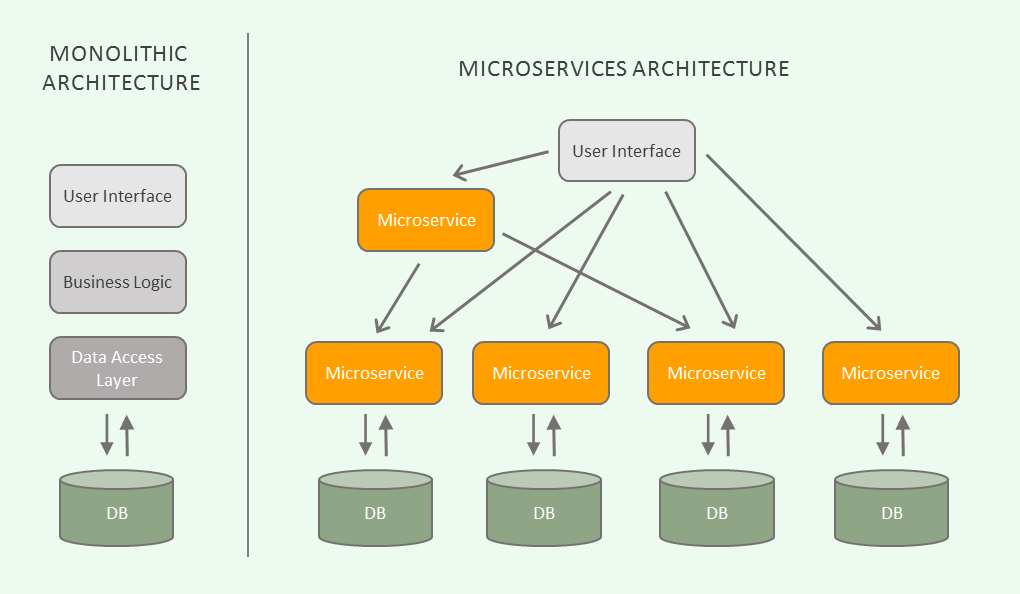
1 – A Arquitetura de microserviços

2 – Domain Driven Design (DDD)

3 – Preparando e modelando a arquitetura do projeto

1. A Arquitetura de Microserviços

Microserviços é uma arquitetura de desenvolvimento de software, que veio para dar uma opção além da arquitetura monolítica, ela segue um padrão de desenvolvimento baseado em módulos, que são partes importantes e que podem ser consideradas domínios dentro do domínio principal do negócio. Ela serve principalmente para desacoplar estruturalmente um software, dando liberdade para que sejam separadas aplicações que se sustentam, e que também podem ser dependentes de outros serviços dentro do mesmo software, o que pode ser visto como um acoplamento, mas muito mais livre e com uma observabilidade mais clara do que o sistema em monólito.



1. Domain Driven Design (DDD)

O DDD é a modelagem do negócio no desenvolvimento do software. Através dele podemos observar o negócio juntamente com os especialistas e criar a sua abstração na estrutura do software.

Descoberta do domínio é a prática de conversar com esses especialistas.

Subdomínios podem ser entendidos como partes que podem ter sua própria autonomia e responsabilidade dentro do negócio. Como departamentos, por exemplo, em um negócio de Seguros, teríamos os departamentos: Financeiro, Sinistros, Comercial, Atendimento, entre outros.

Em subdomínios, um cliente na parte comercial, pode ser um segurado na parte de sinistros. Isso mostra que em cada subdomínio podem existir entidades que representam o mesmo item no todo do negócio, mas que tem nomenclaturas diferentes em seu próprio subdomínio. Sendo que o inverso também é válido, onde um pagamento pode representar algo externo do negócio em um subdomínio e também algo interno em outro.

No DDD existem três classificações principais de subdomínios: Core Subdomain, Supporting Subdomain e Generic Subdomain.

Core Subdomain seria o coração do negócio, onde seriam alocados estrategicamente os melhores profissionais e os melhores recursos.

Supporting Subdomain daria suporte a outros domínios, seriam uma parte do negócio.

Generic Subdomain pode ser visto como algo genérico, que pode ser visto até mesmo com um certo padrão em vários negócios.

O domínio pode ser visto como uma arvore, o core seria a sua raíz, que deve receber os nutrientes e dela que tudo vai crescer e se desenvolver. O supporting pode ser visto como os galhos ou as folhas, que são estruturas que fazem parte dessa arvore, que podem suportar outros galhos, mas nunca serão essenciais para essa arvore, serão apenas parte de um todo dando mais abrangência para a arvore (negócio). Já o generic pode ser visto na arvore como o padrão das folhas e dos galhos, os formatos de cada um e tempo de crescimento, que qualquer outra arvore pode ter também.

Nem todo subdomínio vai ser implementado em código.

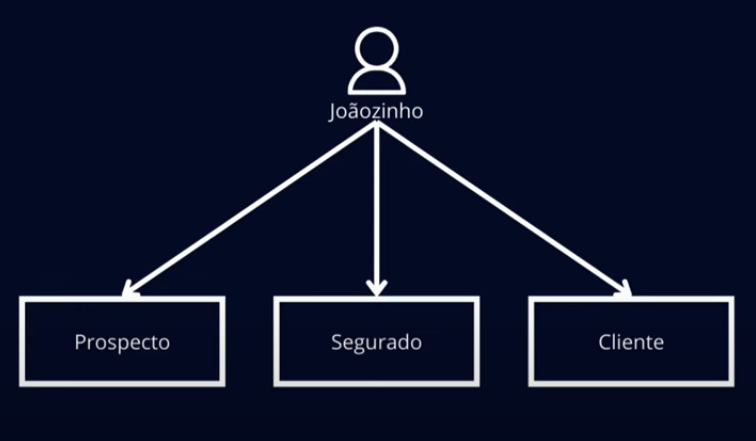
O foco do DDD é entender o domínio como um todo e não desenvolver tudo do negócio.

* 1. Domain Model

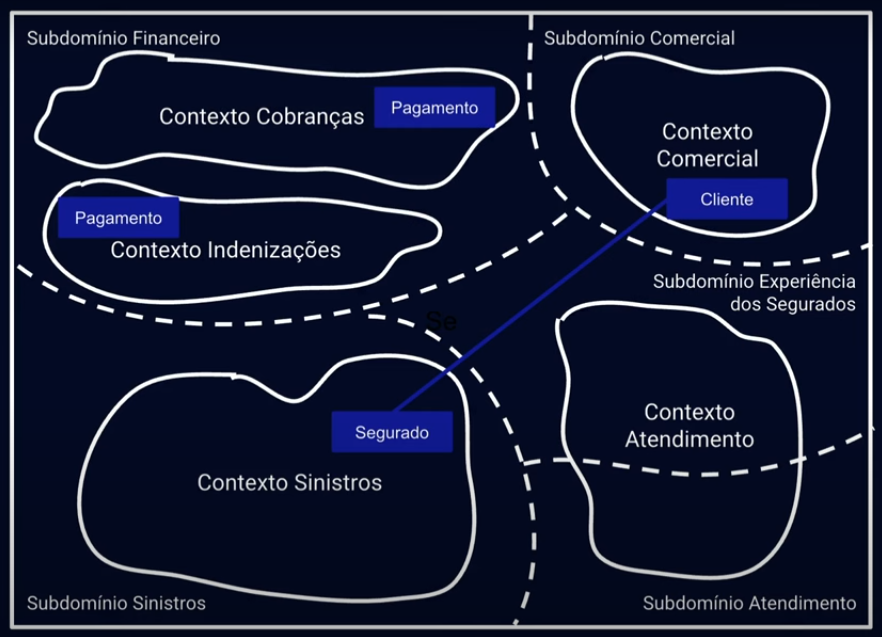
Centraliza as regras de negócio e reflete o nosso entendimento do domínio dentro do software. Pode ser representado como um diagrama:



O significado das coisas muda conforme o contexto, como já tratado anteriormente.



Assim, o DDD deve ser aplicado tendo em mente que cada subdomínio tem o seu contexto próprio ou até mais de um contexto, para evitar o problema da Big Ball of Mudge, um exemplo seria um aglomerado de nomenclaturas e definições da mesma entidade em vários subdomínios, gerando conexões bizarras e difíceis de compreender.



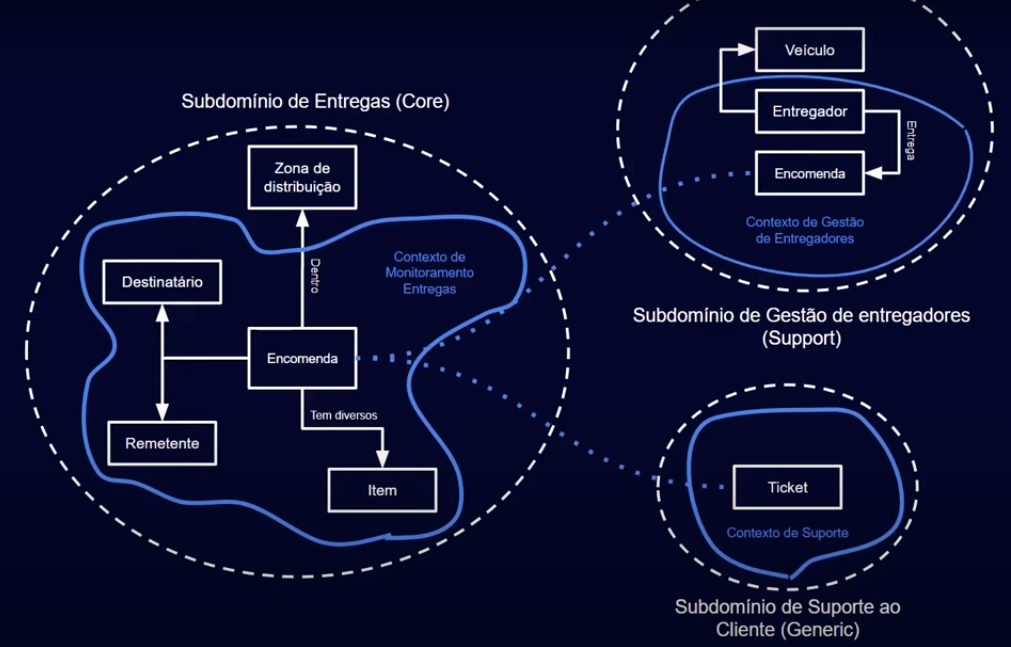
1. Preparando e modelando a arquitetura do projeto

AlgaDelivery

Subdominios identificados:  


Subdominio de Entregas, seria o Core do negócio.

A gestão de entregadores seria o Support e a parte de Tickets seria o Generic.



Analisando o domínio, os microserviços ficariam assim:  
