|  |
| --- |
| **CAP 9 – ARQUITETANDO UM MICROSSERVIÇO** |

# INTEGRANTES DO TECH WARRIORS

|  |
| --- |
| CAIO MORAES – RM 89262  DANILO PERESTRELO – RM 87813  FERNANDO SITTA – RM 87177  JOÃO PAULO HOTEQUIL – RM 87808  LUCAS NEIVA – RM 87300 |

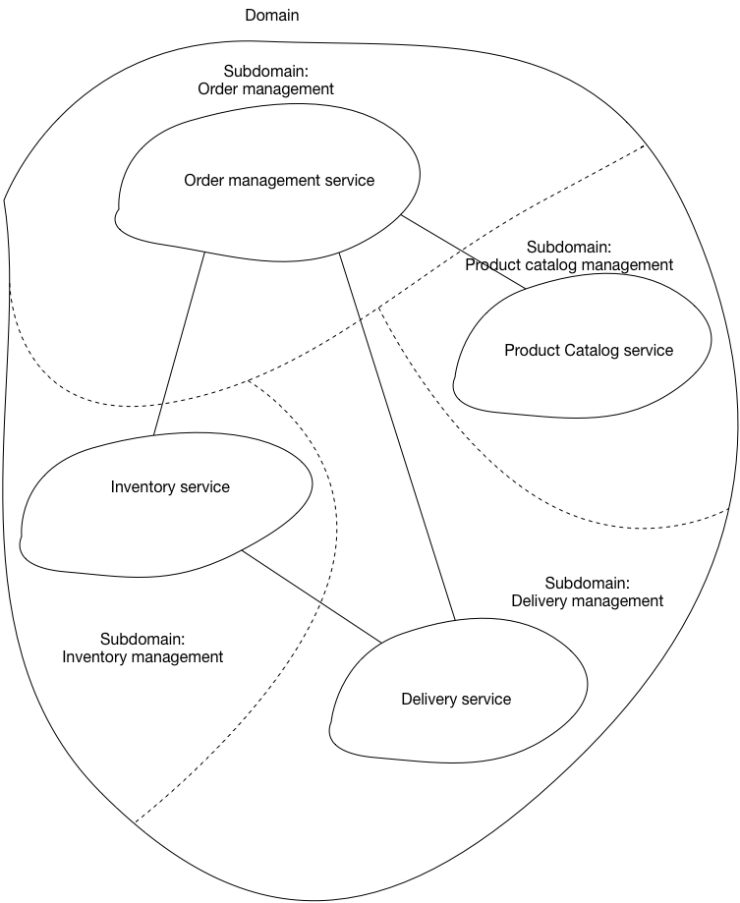
# DESIGN PATTERNS

Escolhemos dois designs patterns para fazer parte da nossa aplicação, abaixo falaremos um pouco sobre eles e ditaremos suas características indicando para o que serão utilizados.

**DECOMPOSE BY SUBDOMAIN:**

Esse design pattern é voltado para a decomposição da estrutura. Visa uma arquitetura com pouca acoplação e o desenvolvimento é totalmente voltada para a regra de negócio, pois utiliza o padrão DDD (Domain-driven design). Outro ponto que o pattern soluciona é ter uma arquitetura estável.

O trabalho em equipe também é beneficiado, pois cada serviço deve ser autônomo e pequeno para que um time desenvolva e dê manutenção. Por não ser um serviço grande, é mais simples de ser testado pelo time. Com a desacoplação da arquitetura, permite com que o sistema não seja totalmente impactado.



|  |
| --- |
| **CAP 9 – ARQUITETANDO UM MICROSSERVIÇO** |

**API COMPOSITION:**

Esse pattern trabalha na camada mais próxima do banco de dados, fazendo a junção dos dados na memória dos microsserviços envolvidos. Utiliza da sua simplicidade para fazer a consulta dos dados nos bancos, porém, o programador deve ficar atento com os dados sensíveis da aplicação, podendo prejudicar o usuário final.

