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

# INTEGRANTES DO TECH WARRIORS

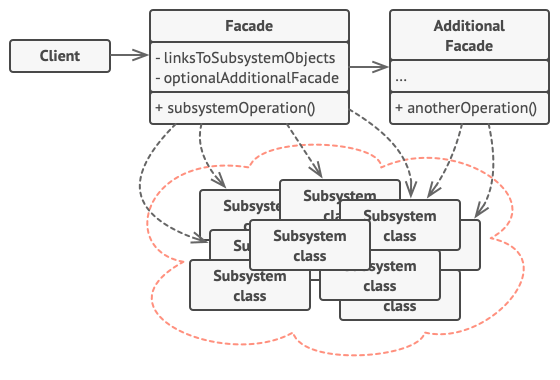
|  |
| --- |
| CAIO MORAES – RM 89262  DANILO PERESTRELO – RM 87813  FERNANDO SITTA – RM 87177  JOÃO PAULO HOTEQUIL – RM 87808  LUCAS NEIVA – RM 87300 |

# DESIGN PATTERNS

Analisamos e estudamos alguns designs patterns, encontramos dois mais interessantes que agregariam para a nossa aplicação, são eles: Facade e Saga. Citaremos as características que mais nos chamaram a atenção neles.

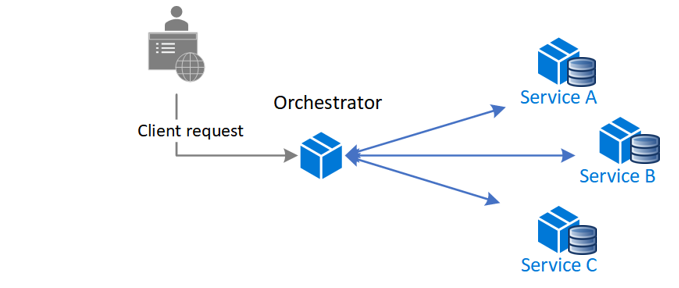
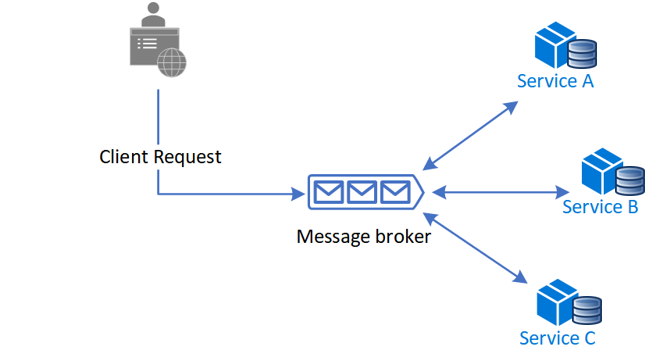
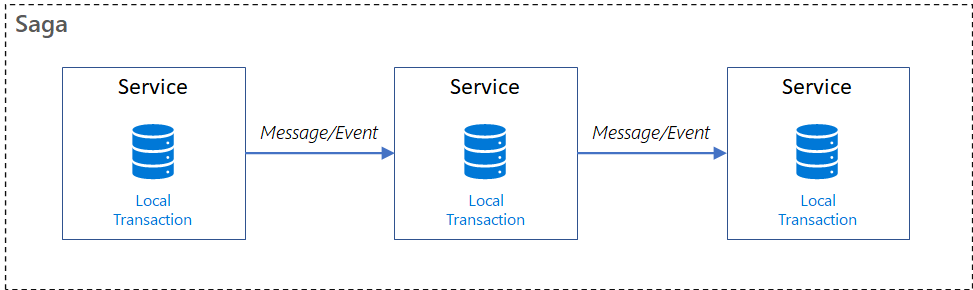
* **FACADE:**

O Facade nos ajuda a simplificar os sistemas. Ele funciona fornecendo uma interface clara para um conjunto de classes ou subsistemas. Esse padrão é especialmente útil em uma arquitetura **REST**, pois ajuda a melhorar a comunicação entre os clientes e os servidores. Ao fornecer uma interface de nível superior para os clientes, o Facade ajuda a ocultar a complexidade dos subsistemas subjacentes, reduzindo assim o acoplamento entre *client* e *server*, buscando mais escalabilidade e boa modularidade do projeto.



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

* **SAGA:**  
    
  O Saga auxilia na administração da consistência dos dados entre os microsserviços através de suas transações para atualizar cada serviço que estiver inscrito em seus eventos, eles receberão mensagens com qualquer tipo de informação. Essas transações devem seguir o acrônimo: **ACID** (atômicas, consistentes, isoladas e duráveis). Uma das grandes vantagens de utilizar o Saga é que caso uma transição falhar, irá compensar as anteriores desfazendo-as. Existem duas formas de trabalhar com Saga: **Coreografia** e **Orquestração**.



Concluímos que o Facade em alguns momentos acaba sendo refém de muitas dependências e a escalabilidade do projeto é prejudicada. Por este motivo, escolhemos o Saga que administra de uma forma menos centralizada e trabalha com eventos. Utilizaremos a maneira de **Coreografia** disponibilizada pelo Saga, pois irá se adaptar melhor no projeto.