

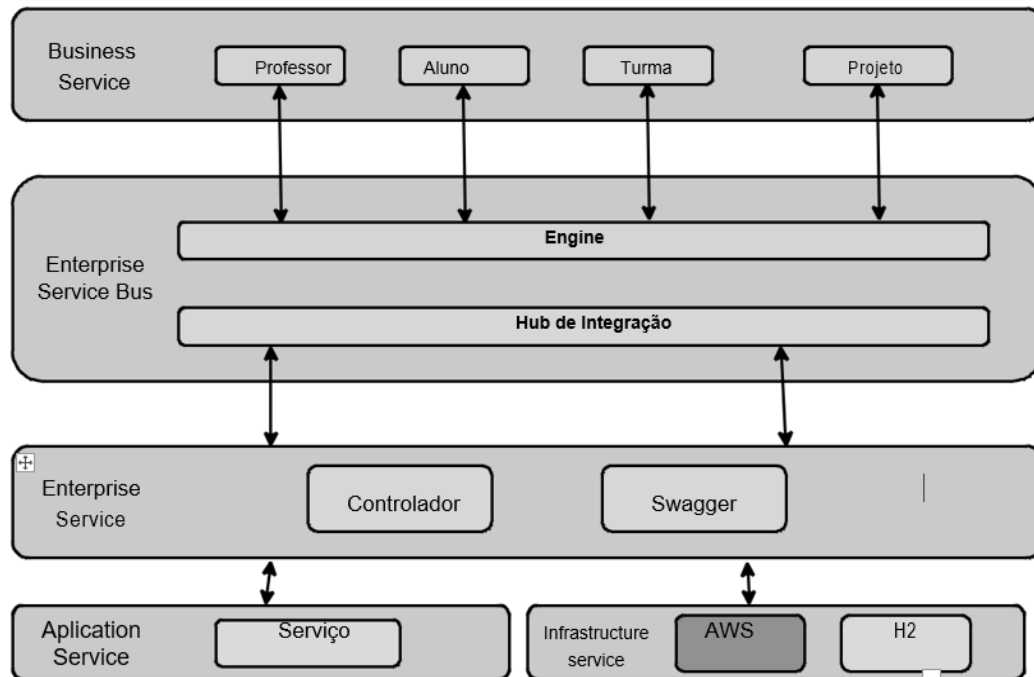
## Exercício 4

**Alunos:** Luanderlandy Fellipe e Matheus Possidônio

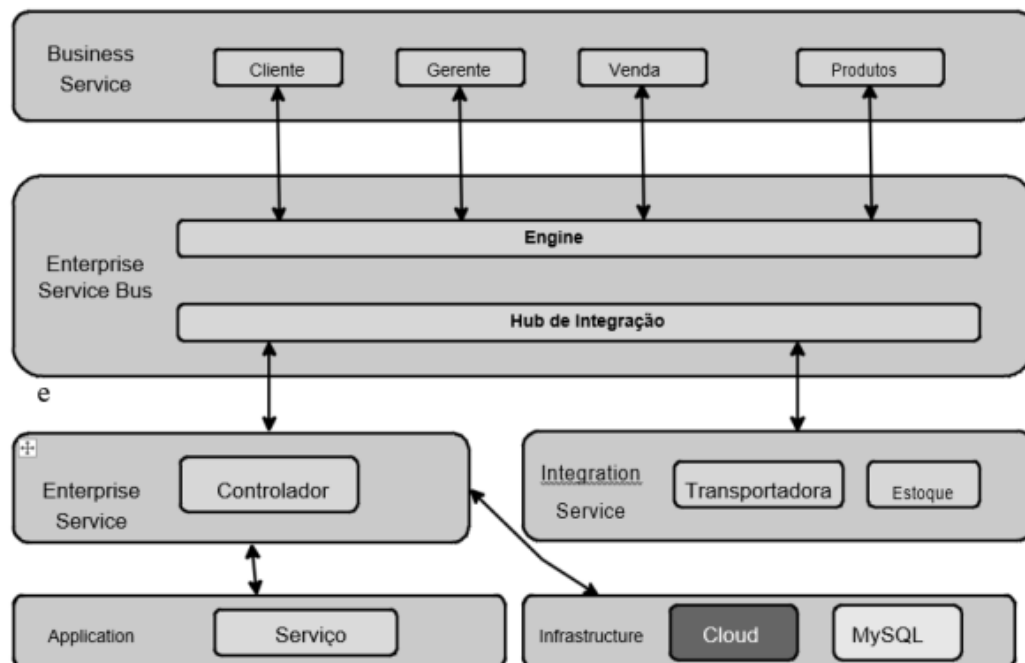
**Projetar dois modelos de aplicações fictícias para a Web utilizando os modelos SOA e Microserviços.**

### 1. Modelo SOA

#### 1.1 Controle Acadêmico (CA):

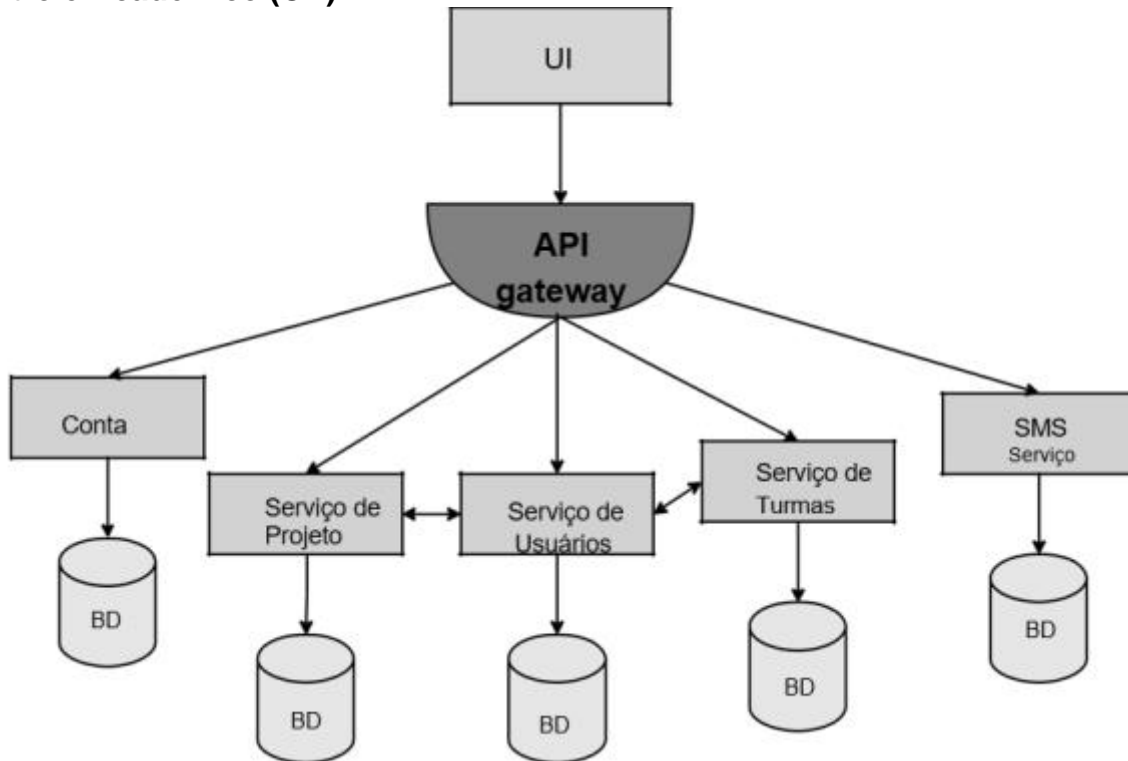


#### 1.2 Loja Virtual

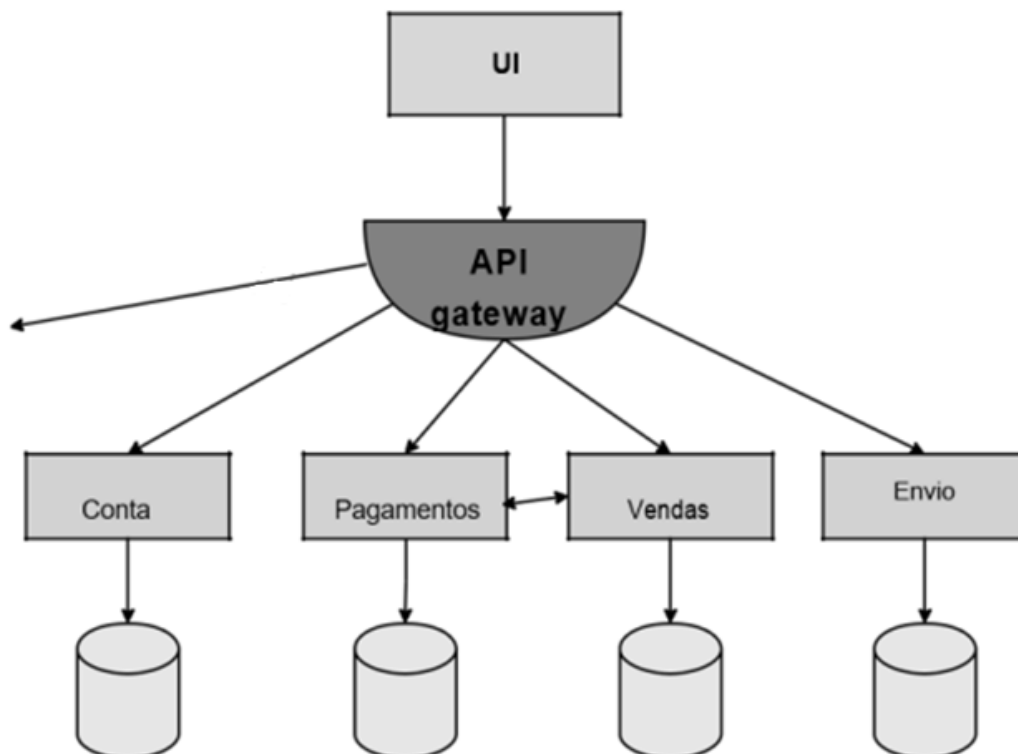


## 2. Modelo de Microserviços

### 2.1 Controle Acadêmico (CA):



### 2.2 Loja Virtual:



## **Comparando as aplicações utilizando os critérios discutidos e também do ponto de vista da evolução dos modelos básicos.**

**SOA(Arquitetura voltada a serviços):** Aqui temos um paradigma de programação onde o foco é delegar ações para determinados eventos, ou seja, de forma simplificada temos que para determinados eventos que aconteçam na nossa aplicação será aplicada uma lógica de gerenciamento e com isso uma determinada ação no programa irá ocorrer, por exemplo, nessa lógica podemos ter uma camada onde recebe os dados e de alguma forma ele decide onde alocar determinadas informações como gerenciamento, informações do banco de dados e por aí vai. Geralmente esse fluxo de informação se dá pelo fluxo de documento como por exemplo json ou soap. Esse organização se destacou principalmente pelas suas vantagens na sua capacidade de trabalho assíncrono sua flexibilidade de onde e como as ações do programa irão ocorrer e sua capacidade de trabalhar. Contudo temos algumas dificuldades nesse paradigma como o alto grau de complexidade exigido para o seu uso e os seus midwares não são padronizados o que pode dificultar a ligação entre vários serviços distintos.

**Arquitetura de microservices:** Nesse paradigma temos o esforço voltado para a separação e independência de serviços e ações dentro de um programa, ou seja, invés de termos um programa com maior nível de código engessado e entrelaçado temos porções menores de código que executam ações independentes entre si e que se comunicam de forma padronizada. Com isso podemos ter uma grande melhora na área de manutenção pois quando formos alterar serviços podemos trabalhar apenas em um serviço específico e não no programa todo, além da geração de testes que se torna mais fácil e eficiente, e por fim temos a capacidade de reuso desses serviços. Contudo temos alguns problemas como por exemplo, devido a muitas vezes termos comunicação entre vários micro serviços podemos ter uma alta latência de transferência de dados além do trabalho exigido para manter os serviços se comunicando entre si, para que eles enviem e recebam o que eles esperam.

Em método de comparação somos capazes de distinguir também os dois paradigmas pelo seu tamanho e escopo, pois como mostrado os micro serviços possuem um maior detalhamento das execuções e executam tarefas de forma individual e única com uma baixa granularidade, já o paradigma orientado a serviços por sua vez possui um detalhamento menor e uma alta granularidade. Concluindo podemos definir que os dois paradigmas são extremamente interessantes e tem espaço no mercado de desenvolvimento, contudo assim como qualquer outro assunto de tecnologia a distinção de melhor e pior irá ficar a cargo do projeto a qual foi definido pois de acordo com o projeto solicitado e dos seus requisitos determinado paradigma irá se sobressair.