

FIA/P GRADUAÇÃO

SISTEMAS DE INFORMAÇÃO

MICROSERVICE AND WEB ENGINEERING

PROF. Thiago Xavier

| Agenda

Aula de Hoje:

- Microsserviços!

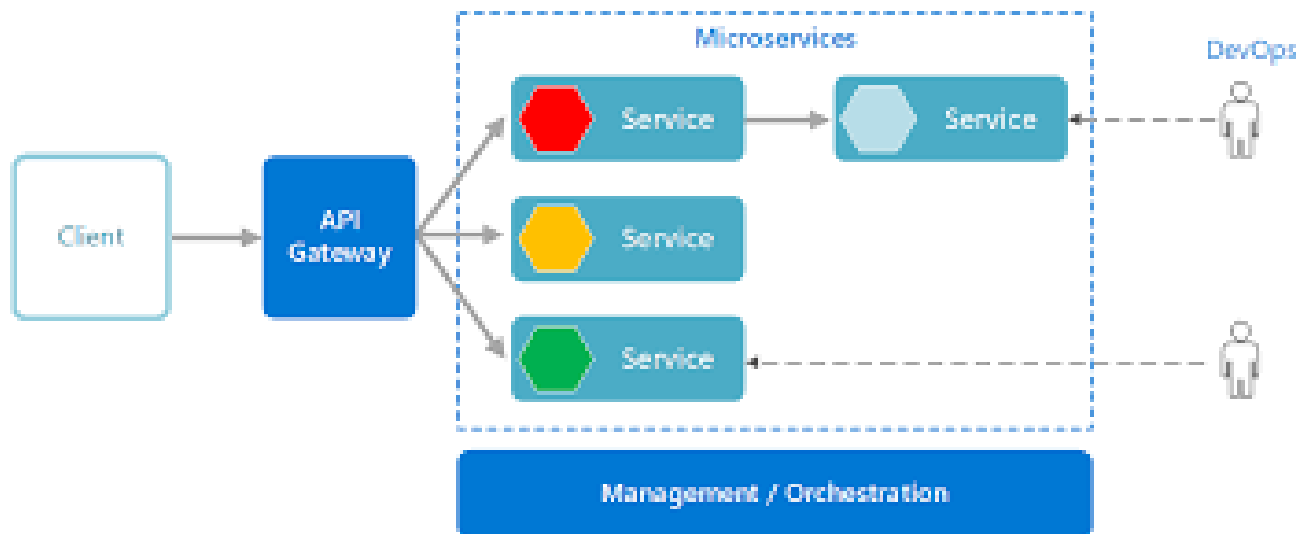
I Objetivo

- Entender os principais conceitos de um microserviço !



O que é um microserviço

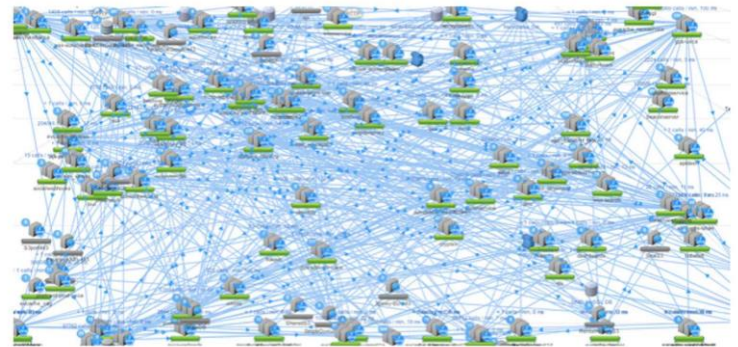
- Determina um modelo de desenvolvimento;
- Também é um conjunto de práticas;
- Aumenta a velocidade de desenvolvimento;
- Agnóstico de tecnologia;
- Permite escalabilidade;
- Princípios e padrões de tecnologias.



Qual o tamanho?

- Determina um modelo de desenvolvimento;
- Não dá para mensurar o tamanho;
- Faz apenas uma coisa bem-feita;
- Delimitação de negócio; (Bounded context);
- Agir independentemente;
 - Incluir o deploy.
- É o ato de dividir o domínio em subdomínios.

NETFLIX



Serviços

- São componentes independentes;
- Interoperabilidade;
- SOA (Arquitetura orientada a serviço).



Microserviços e DDD (domain driven design)

- Podemos relacionar o microserviço ao DDD;
- Domínio do negócio;
- Sub-domínio;
 - Uma parte do domínio: usuário, pedido, produto etc.
- Contexto delimitado. (Bounded contexto);
 - Capacidade de negócio;

.

Rotas e padrão REST

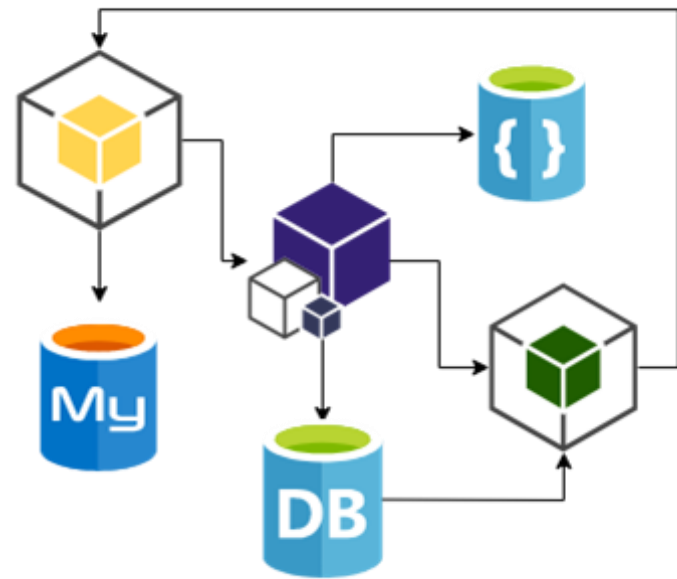
- Podemos relacionar o microserviço ao DDD;
- Definir um padrão para que a equipe tenha um caminho a seguir;
- Por falta de padrões, endpoints podem deixar de seguir boas práticas.
 - Exemplo de endpoints: getAllStudents ou get-all-students.

.

Microserviço

Seguem alguns conceitos fundamentais para a criação de um serviço de qualidade !

- Verbos do Protocolo HTTP
- Acesso aos Recursos
- HTTP Status Code
- Json



Microserviço

Verbos do protocolo HTTP

GET	<ul style="list-style-type: none">• Buscar recursos• Cache
POST	<ul style="list-style-type: none">• Criar um novo recurso
PUT	<ul style="list-style-type: none">• Atualizar um recurso existente
DELETE	<ul style="list-style-type: none">• Remover um recurso

Microserviço

Acesso ao Recursos

- Utilize paths legíveis ao olho humano sem colocar verbos.
- Exemplos mais utilizado:

Roteamento	
Path	Tipo
Plural	/api/products/
Singular	/api/product/
Camel Case	/api/institutionalProducts
Snake Case	/api/institutional_products
Spinal Case	/api/institutional-products

Microserviço

Http Status Code: Toda requisição feita por um **cliente** a um **servidor** deve resultar em uma resposta representado por um **código HTTP**, utilizado para informar o resultado da requisição, tenha ela sido processada com sucesso ou não.

Classe	Semântica
2xx	Indica que a requisição foi processada com sucesso.
3xx	Indica ao cliente uma ação a ser tomada para que a requisição possa ser concluída.
4xx	Indica erro(s) na requisição causado(s) pelo cliente.
5xx	Indica que a requisição não foi concluída devido a erro(s) ocorrido(s) no servidor.

Microserviço

Os status mais utilizados para cada classe HTTP são:

Código	Descrição	Quando utilizar
200	OK	Em requisições GET, PUT e DELETE executadas com sucesso.
201	Created	Em requisições POST, quando um novo recurso é criado com sucesso.
206	Partial Content	Em requisições GET que devolvem apenas uma parte do conteúdo de um recurso.
302	Found	Em requisições feitas à URI's antigas, que foram alteradas.

Microserviço

400	Bad Request	Em requisições cujas informações enviadas pelo cliente sejam invalidas.
401	Unauthorized	Em requisições que exigem autenticação, mas seus dados não foram fornecidos.
403	Forbidden	Em requisições que o cliente não tem permissão de acesso ao recurso solicitado.
404	Not Found	Em requisições cuja URI de um determinado recurso seja inválida.
405	Method Not Allowed	Em requisições cujo método HTTP indicado pelo cliente não seja suportado.

Microserviço

406	Not Acceptable	Em requisições cujo formato da representação do recurso requisitado pelo cliente não seja suportado.
415	Unsupported Media Type	Em requisições cujo formato da representação do recurso enviado pelo cliente não seja suportado.
429	Too Many Requests	No caso do serviço ter um limite de requisições que pode ser feita por um cliente, e ele já tiver sido atingido.
500	Internal Server Error	Em requisições onde um erro tenha ocorrido no servidor.
503	Service Unavailable	Em requisições feitas a um serviço que esta fora do ar, para manutenção ou sobrecarga.

Microserviço

Sempre que implementar algum serviço, pense em qual o melhor HTTP Status Code retornar !!!





Microserviço

Json

Representação de dados baseado na linguagem de programação Javascript, daí o nome JavaScript Object Notation.

Sintaxe básica do Json:

- { e }** - Delimita um objeto
- [e]** - Delimita um array
- :** - Separa chaves (atributos) de valores
- ,** - Separa os atributos chave/valor

Tipos de dados:

```
{  
  "texto" : "Brasil",  
  "numero" : 23,  
  "numeroReal" : 54.87,  
  "booleano": true,  
  "nulo": null  
}
```

OBS: Para o tipo data, anote a propriedade do Model com:
@JsonFormat(pattern="dd/MM/yyyy")

Microserviço

```
{  
  "company": "mycompany",  
  "companycontacts": {  
    "phone": "123-123-1234",  
    "email": "myemail@domain.com"  
  },  
  "employees": [  
    {  
      "id": 101,  
      "name": "John",  
      "contacts": [  
        "email1@employee1.com",  
        "email2@employee1.com"  
      ]  
    },  
    {  
      "id": 102,  
      "name": "William",  
      "contacts": null  
    }  
  ]  
}
```

JSON Object → {

String Value → "mycompany"

Object Inside Object → {

JSON Array → [

Array Inside Array → [

Number Value → 102

Null Value → null

| Microserviço - Vamos praticar

Para praticar, vamos montar alguns grupos para criarmos alguns serviços. Todos os serviços devem retornar os seguintes verbos: get, post, put e delete. Este exercício será realizado de forma “mockada”(<https://www.mockable.io/>), ou seja, não vamos construir nenhum serviço.

Serviços internos:

- Departamentos (codigo, nome e setor). Um departamento pode ter vários setores.
- Cadastro de pessoas (nome, documento, idade e endereço, telefone e e-mail);
- Cadastro de folha salarial (código da pessoa, salario e data de cadastro, codigo do departamento, código do setor);

| Microserviço - Vamos praticar

Serviços para atendimento ao cliente:

- Tickets aberto para atendimento (identificador da pessoa, título, telefone);
- Cadastro de atendentes: identificador da pessoa, identificador do departamento e identificador do setor;
- Registros de atendimento: ticket, texto para salvar histórico de conversa, identificador do atendente e motivos do atendimento (ex: cliente reclamando, cliente solicitando documento);

Entregável:

- Links dos mock;
- Desenho da arquitetura.



Copyright © 2023
Profº. Thiago Xavier