

At Manager

Documentação de visão

Versão 1.0

Tauan Santana da Silva

Sumário

1 - Introdução.....	3
2 – Arquitetura	3
3 - Diagramas.....	4
4 - Serviços disponíveis em cada microserviço	5
6 - Melhorias e sugestões.....	13

AT Manager

1 - Introdução

A plataforma visa facilitar a gestão de produtos no sistema, gestão de fornecedores e gestão de cotações com os fornecedores.

2 – Arquitetura

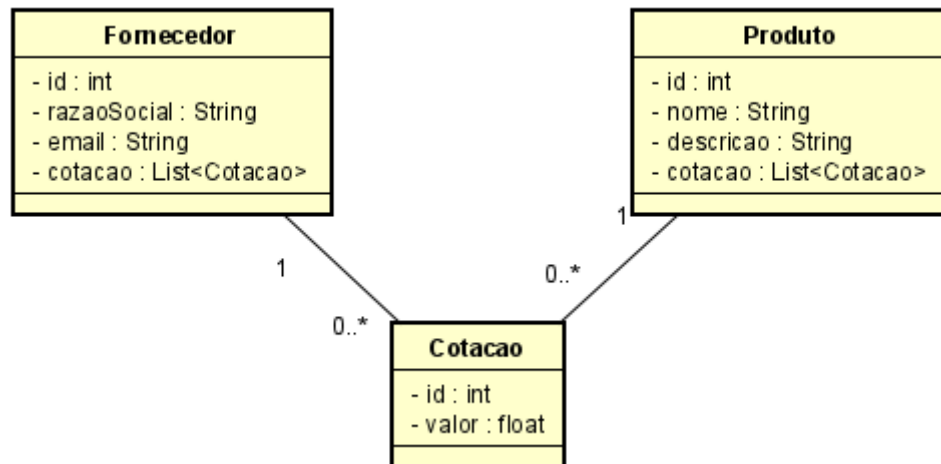
A At Manager foi desenvolvida pensando em escalabilidade e utilização de microsserviços e nuvem.

Todos os dois módulos (Produto e Cotação) são isolados desde sua concepção, permitindo desacoplamento, escalabilidade e facilidade em sua manutenção.

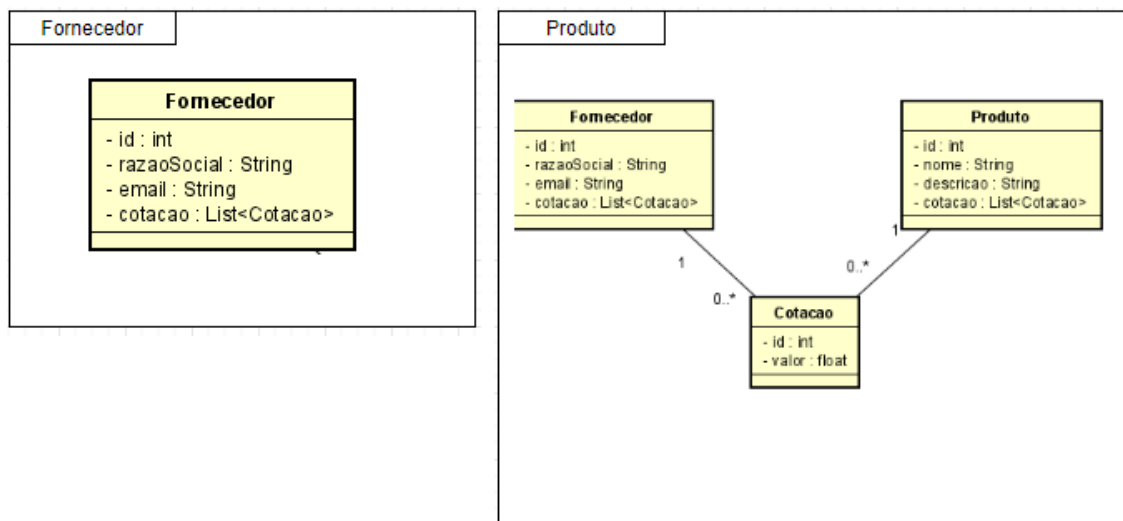
Além disso, a plataforma é hospedada na nuvem da Amazon e utiliza um banco de dados como serviço que proporciona escalabilidade e confiabilidade.

3 - Diagramas

Diagrama de domínio



Delimitação de contexto de cada microserviço



4 - Serviços disponíveis em cada microserviço

- Fornecedor

GET /Fornecedor/{id}

Endpoint responsável por listar um fornecedor por Id

Entrada:

Id: campo do tipo inteiro e deve ser enviado na URL

Saída: {

```
"id": 2,  
"cnpj": "546564654645645",  
"razaoSocial": "Razao Social ",  
"email": "T..@gmail.com"  
}
```

DELETE /Fornecedor/{id}

Endpoint responsável por deletar um fornecedor por Id

Entrada:

Id: campo do tipo inteiro e deve ser enviado na URL

Saída: *não se aplica*

GET /Fornecedor/

Endpoint responsável por listar todos os fornecedores

Entrada: *não se aplica*

Saída: [{

```
"id": 1,  
"cnpj": "123456789",  
"razaoSocial": "Razao Social 1",  
"email": "T..@gmail.com"  
}  
, {  
  
"id": 2,  
"cnpj": "546564654645645",  
"razaoSocial": "Razao Social 2",  
"email": "T..@gmail.com"  
}]
```

POST /Fornecedor/

Endpoint responsável por incluir um fornecedor

Entrada: {

```
"cnpj": "456654546",  
"razaoSocial": "Fornecedor 01",  
"email": "fornecedor010@gmail.com",  
}
```

Saída: {

```
"cnpj": "456654546",  
"razaoSocial": "Fornecedor 01",  
"email": "fornecedor010@gmail.com",  
"id": 1  
}
```

PUT /Fornecedor/

Endpoint responsável por alterar um fornecedor

Entrada: {

```
"cnpj": "456654546",  
"razaoSocial": "Fornecedor 01",  
"email": "fornecedor010@gmail.com",  
"id": 1  
}
```

Saída: {

```
"cnpj": "456654546",  
"razaoSocial": "Fornecedor 01",  
"email": "fornecedor010@gmail.com",  
"id": 1  
}
```

- Produto

GET /Produto/{id}

Endpoint responsável por listar um produto por Id

Entrada:

Id: campo do tipo inteiro e deve ser enviado na URL

Saída: {

```
"id": 2,  
"cnpj": "546564654645645",  
"razaoSocial": "Razao Social ",  
"email": "T..@gmail.com"  
}
```

DELETE /Produto/{id}

Endpoint responsável por deletar um produto por Id

Entrada:

Id: campo do tipo inteiro e deve ser enviado na URL

Saída: *não se aplica*

GET /Produto/

Endpoint responsável por listar todos os produtos

Entrada: *não se aplica*

Saída: [{

```
"id": 1,  
"cnpj": "123456789",  
"razaoSocial": "Razao Social 1",  
"email": "T..@gmail.com"  
}  
, {  
"id": 2,  
"cnpj": "546564654645645",  
"razaoSocial": "Razao Social 2",  
"email": "T..@gmail.com"  
}]
```

POST /Produto/

Endpoint responsável por incluir um produto

Entrada: {

```
"id": 1,  
"nome": "NomeProduto",  
"descricao": "Produto novo"  
}
```

Saída: {

```
"id": 1,  
"nome": "NomeProduto",  
"descricao": "Produto novo"  
}
```

PUT /Produto/

Endpoint responsável por alterar um produto

Entrada: {

```
"id": 1,  
"nome": "NomeProduto",  
"descricao": "Produto novo"  
}
```

Saída: {

```
"id": 1,  
"nome": "NomeProduto",  
"descricao": "Produto novo"  
}
```


- Cotação

GET /Cotacao/{id}

Endpoint responsável por listar uma cotação por Id

Entrada:

Id: campo do tipo inteiro e deve ser enviado na URL

Saída: {

```
"id": 2,  
"valor": 0.0,  
"fornecedor": {  
  "id": 2,  
  "cnpj": "546564654645645",  
  "razaoSocial": "Razao Social Exemplo",  
  "email": "T..@gmail.com",  
  "cotacao": []  
},  
"produtos": [  
  {  
    "id": 1,  
    "nome": "Produto1",  
    "descricao": "descricao produto",  
    "cotacao": []  
  }  
]  
}
```

DELETE /Cotacao/{id}

Endpoint responsável por deletar uma cotação por Id

Entrada:

Id: campo do tipo inteiro e deve ser enviado na URL

Saída: *não se aplica*

GET /Cotacao/

Endpoint responsável por listar todas as cotações

Entrada: *não se aplica*

Saída: [{

```
    "id": 2,
    "valor": 0.0,
    "fornecedor": {
        "id": 2,
        "cnpj": "546564654645645",
        "razaoSocial": "Razao Social Tauan",
        "email": "tauansantana@gmail.com",
        "cotacao": []
    },
    "produtos": [
        {
            "id": 1,
            "nome": "Produto1",
            "descricao": "descricao produto",
            "cotacao": []
        }
    ]
},
{
    "id": 3,
    "valor": 10.0,
    "fornecedor": {
        "id": 2,
        "cnpj": "1234567879",
        "razaoSocial": "Razao Social Tauan 2",
        "email": "T..@gmail.com",
        "cotacao": []
    },
    "produtos": [
        {
            "id": 2,
            "nome": "Produto2",
            "descricao": "descricao produto 2",
            "cotacao": []
        }
    ]
}]
```

POST /Cotacao/

Endpoint responsável por incluir uma cotação

Entrada: {

```
    "valor": 10.0,
    "fornecedor": {
        "id": 2,
        "cnpj": "546564654645645",
        "razaoSocial": "Razao Social Tauan",
        "email": "tauansantana@gmail.com",
        "cotacao": []
    },
    "produtos": [
        {
            "id": 1,
            "nome": "Produto1",
            "descricao": "descricao produto",
            "cotacao": []
        }
    ]
}
```

Saída: {

```
    "id": 2,
    "valor": 0.0,
    "fornecedor": {
        "id": 2,
        "cnpj": "546564654645645",
        "razaoSocial": "Razao Social Tauan",
        "email": "tauansantana@gmail.com",
        "cotacao": []
    },
    "produtos": [
        {
            "id": 1,
            "nome": "Produto1",
            "descricao": "descricao produto",
            "cotacao": []
        }
    ]
}
```

PUT /Cotacao/

Endpoint responsável por alterar um produto

Entrada: {

```
    "id": 2,
    "valor": 0.0,
    "fornecedor": {
      "id": 2,
      "cnpj": "546564654645645",
      "razaoSocial": "Razao Social Tauan",
      "email": "tauansantana@gmail.com",
      "cotacao": []
    },
    "produtos": [
      {
        "id": 1,
        "nome": "Produto1",
        "descricao": "descricao produto",
        "cotacao": []
      }
    ]
  }
```

Saída: {

```
    "id": 2,
    "valor": 0.0,
    "fornecedor": {
      "id": 2,
      "cnpj": "546564654645645",
      "razaoSocial": "Razao Social Tauan",
      "email": "tauansantana@gmail.com",
      "cotacao": []
    },
    "produtos": [
      {
        "id": 1,
        "nome": "Produto1",
        "descricao": "descricao produto",
        "cotacao": []
      }
    ]
  }
```

6 - Melhorias e sugestões

- É preciso incrementar a quantidade de testes dos dados que chegam nas API para que evite problemas de dados vazios ou replicados no banco de dados.
- O banco de dados na Amazon RDS é muito interessante, mas é preciso ficar atento às camadas de preço. Na camada gratuita foi possível detectar, por muitas vezes, timeout na aplicação.
- A interface de interação com o usuário é muito básica, recomendo a reformulação para tornar os formulários mais intuitivos, interativos e seguros
- É preciso implementar uma camada de segurança para as API's e aplicação Web.