

README.MD

O projeto foi desenvolvido utilizando Python 3.10, com framework Django Rest Framework. Para o desenvolvimento, visando facilitar o setup inicial, foi utilizado o Sqlite3 (Não indicado para ambientes de produção).

O desenvolvimento foi orientado pelas definições do projeto. Visando entregar o mais próximo do solicitado, e atendendo o formato dos endpoints.

Alguns pontos não estão totalmente finalizados, ou possuem alguma melhoria, e foram descritos em Atividades pendentes.

Escopo

Público Alvo: Profissionais que atuam em uma rede de indicação de trabalhos freelance que precisam comprar, vender ou trocar produtos de tecnologia (notebooks, mouses, teclados, etc) que possibilitem a realização de seus trabalhos de forma rápida, otimizando seus custos com estruturas de trabalho.

- Deve ser desenvolvida uma API RESTful para atender os requisitos funcionais do projeto visual já apresentado.
- Nesta API temos ao todo 7 entidades, as 3 entidades principais são: usuário, negociação e oferta.
- Além das 3 principais temos: mensagens, frete, pagamento e convite. A relação seria:
 - Usuário pode efetuar gerar múltiplos convites;
 - Um convite é vinculado a um novo cadastro de usuário;
 - Um usuário efetua múltiplas negociações;
 - Um usuário envia múltiplas ofertas em uma negociação;
 - Um usuário envia e recebe múltiplas mensagens vinculadas a uma negociação;
 - Um pagamento é associado a uma negociação e é efetuado por um usuário;
- Cada endpoint criado, com seu respectivo método, deverá receber parâmetros, se for o caso, como Object JSON. Estes parâmetros estão disponíveis no descritivo dos endpoint.
- Todo endpoint devolverá um JSON, que também estará detalhado neste documento.
- A autenticação será feita utilizando JWT (JSON Web Token). Os endpoints que terão acesso restrito estarão apontados na documentação (Exemplo: "Public: Não").
- Você está livre para escolher o formato de grafia do seu endpoint, mas pedimos que ele seja versionado. Obviamente você deve usar os verbos da metodologia RESTful.
- Todos os retornos de endpoint deve possuir HTTP Code corretos para a ação de retorno. Usamos como referência este link:

Estimativa

Tarefa

	Estimativa em horas
Estudo e entendimento do escopo	2
CRUD Usuário	4
CRUD Endereço	2
CRUD Negociação	2
CRUD Produto da negociação	2
CRUD Foto do produto da negociação	4
CRUD Oferta em negociação	4
CRUD Mensagens	4
Integração com webservice correios	4
Consulta de Frete	2

CRUD Pagamento	2
CRUD Convite	3
Implementação de serviço em docker.	3
Documentação do projeto	4
Total	42 horas

Estimativa em dias: 7 (19/02/2023)

Atividades pendentes

- ☐ Corrigir a atualização de imagens de produtos, atualmente está duplicando a imagem, e não realiza a exclusão de imagens não utilizadas mais.
- ☐ Implementar o método de pagamento, e realizar o relacionamento com a oferta, sendo necessário o aceite do vendedor.
- ☐ Adicionar filtros para a listagem de itens. Especificamente na consulta de mensagens, permitir filtrar por origem, e alterar a ordenação.
- ☐ Retornar a listagem de todos os meios de frete. Atualmente só retorna o valor para sedex varejo.
- ☐ Padronizar mensagens de erro no retorno em um objeto padronizado.
- ☐ Implementar testes automatizados para a validação dos endpoints.

Execução da API

A seguir são descritas duas formas de execução para a API, utilizando ou não o docker.

Após ser efetuado os passos a seguir. A documentação Swagger deve estar acessível no endereço <http://127.0.0.1:8000/swagger/>.

Execução local (sem o uso do docker)

Para executar os seguintes procedimentos, certifique-se de estar dentro da pasta `ecommerce_api`

1. Crie um ambiente virtual de python, para a versão 3.10

```
python -m venv venv && source venv/bin/activate
```

2. Instale as dependências do projeto

```
pip install -r requirements.txt
```

3. Execute as “migrations”, para criar a estrutura de banco de dados

```
python manage.py makemigrations && \
python manage.py migrate
```

4. Cadastre um usuário admin, para iniciar a utilização da API, e acessar o painel de administração

```
python manage.py createsuperuser
```

5. Execute o projeto

```
python manage.py runserver
```

Execução utilizando Docker

Para executar os seguintes procedimentos, certifique-se de estar na raiz do projeto. E possuir o *docker* e *docker-compose* instalados no seu ambiente.

Execute o seguinte comando para subir o container da api.

```
docker-compose up --build
```

Este comando realiza todos procedimentos listados no passo **Execução local (sem o uso do docker)**.

O usuário padrão para acesso de administrador é o seguinte:

- **Usuário:** admin
- **Senha:** admin

Esta configuração pode ser alterada no arquivo `docker-compose.yml`, alterando os argumentos `DJANGO_ADMIN_USER` e `DJANGO_ADMIN_PASSWORD`.

Endereços úteis da API

Com a API rodando, alguns dos seguintes links podem ser úteis.

Painel de administrador: <http://127.0.0.1:8000/admin/>

* Painel de administrador, que permite a manipulação de dados do banco, por meio de uma interface.

Documentação Swagger: <http://127.0.0.1:8000/swagger/>

* Documentação interativa, que permite o envio de requisições, para realizar testes da API.

Documentação Redoc: <http://127.0.0.1:8000/redoc/>

* Documentação visual para os endpoints da API.

Endpoints

A api utiliza um versionamento para os endpoints, que permite a evolução de seus serviços com maior agilidade, sem danificar outros serviços dependentes.

As versões permitidas atualmente são `v1` e `v2`, informadas no path da url, antes do endereço desejado. Por exemplo, para realizar a autenticação, na versão 1, a url deve ficar neste formato `http://127.0.0.1:8000/api/v1/authenticate/`.

Usuários

GET: /users/

- Permissões necessárias: Administrador.
- Exemplo de resposta - 200:

```
{
  "count": 0,
  "next": "string",
  "previous": "string",
  "results": [
    {
      "id": 0,
      "username": "string",
      "email": "user@example.com",
      "password": "string",
      "confirm_password": "string",
      "first_name": "string",
      "last_name": "string",
      "is_active": true,
      "is_staff": true,
      "is_superuser": true,
      "date_joined": "2023-02-19T10:32:06.402Z",
      "last_login": "2023-02-19T10:32:06.402Z",
    }
  ]
}
```

```

    "location": {
      "id": 0,
      "created_at": "2023-02-19T10:32:06.402Z",
      "updated_at": "2023-02-19T10:32:06.402Z",
      "lat": 0,
      "lng": 0,
      "address": "string",
      "city": "string",
      "state": "string",
      "zip_code": "string"
    },
    "invite": 0
  }
]
}

```

POST: /users/

- Permissões necessárias: Pública.
- Exemplo de requisição:

```

{
  "username": "kaspary",
  "email": "user@example.com",
  "password": "kaspary",
  "confirm_password": "kaspary",
  "first_name": "João Pedro",
  "last_name": "Kaspary",
  "location": {
    "lat": -23.540652232340175,
    "lng": -46.63047949194844,
    "address": "R. da Cantareira, 306 - Centro Histórico de São Paulo, São Paulo - SP, 01024-900",
    "city": "São Paulo",
    "state": "SP",
    "zip_code": "01153000"
  },
  "invite": null
}

```

- Exemplo de resposta - 201:

```

{
  "id": 2,
  "username": "kaspary",
  "email": "user@example.com",
  "first_name": "João Pedro",
  "last_name": "Kaspary",
  "is_active": true,
  "is_staff": false,
  "is_superuser": false,
  "date_joined": "2023-02-19T07:39:55.797944-03:00",
  "last_login": null,
  "location": {
    "id": 4,
    "created_at": "2023-02-19T07:39:55.795681-03:00",
    "updated_at": "2023-02-19T07:39:55.795744-03:00",
    "lat": -23.540652232340175,
    "lng": -46.63047949194844,
    "address": "R. da Cantareira, 306 - Centro Histórico de São Paulo, São Paulo - SP, 01024-900",
    "city": "São Paulo",
    "state": "SP",
    "zip_code": "01153000"
  }
}

```

GET: /users/{id}/

- Permissões necessárias: Autenticado, deve ser o próprio usuário ou administrador.
- Exemplo de resposta - 200:

```
{
  "id": 2,
  "username": "string",
  "email": "user@example.com",
  "first_name": "string",
  "last_name": "string",
  "is_active": true,
  "is_staff": false,
  "is_superuser": false,
  "date_joined": "2023-02-18T22:12:24.228897-03:00",
  "last_login": "2023-02-18T22:12:32.058008-03:00",
  "location": {
    "id": 1,
    "created_at": "2023-02-18T22:12:24.228021-03:00",
    "updated_at": "2023-02-19T07:58:29.436847-03:00",
    "lat": -23.540652232340175,
    "lng": -46.63047949194844,
    "address": "R. da Cantareira, 306 - Centro Histórico de São Paulo, São Paulo - SP, 01024-900",
    "city": "São Paulo",
    "state": "SP",
    "zip_code": "01153000"
  }
}
```

PUT: /users/{id}/

- Permissões necessárias: Autenticado, deve ser o próprio usuário ou administrador.
- Exemplo de requisição;

```
{
  "password": "12345",
  "confirm_password": "12345",
  "first_name": "João Pedro",
  "last_name": "Kaspary",
  "location": {
    "lat": -23.540652232340175,
    "lng": -46.63047949194844,
    "address": "R. da Cantareira, 306 - Centro Histórico de São Paulo, São Paulo - SP, 01024-900",
    "city": "São Paulo",
    "state": "SP",
    "zip_code": "01153000"
  }
}
```

- Exemplo de resposta - 200:

```
{
  "id": 2,
  "username": "string",
  "email": "user@example.com",
  "first_name": "Usuário",
  "last_name": "Kaspary",
  "location": {
    "id": 1,
    "created_at": "2023-02-18T22:12:24.228021-03:00",
    "updated_at": "2023-02-19T08:03:11.518595-03:00",
    "lat": -23.540652232340175,
    "lng": -46.63047949194844,
    "address": "R. da Cantareira, 306 - Centro Histórico de São Paulo, São Paulo - SP, 01024-900",
    "city": "São Paulo",
    "state": "SP",
    "zip_code": "01153000"
  },
  "date_joined": "2023-02-18T22:12:24.228897-03:00",
  "last_login": "2023-02-18T22:12:32.058008-03:00",
  "is_active": true,
  "is_staff": false,
  "is_superuser": false
}
```

DELETE: /users/{id}/

- Permissões necessárias: Autenticado, deve ser o próprio usuário ou administrador.
- Exemplo de resposta - 204:

```
204
```

Convites

GET: /users/invites

- Permissões necessárias: Autenticado.
- Exemplo de resposta - 200:

```
{
  "count": 1,
  "next": null,
  "previous": null,
  "results": [
    {
      "id": 2,
      "name": "convidado",
      "email": "user@example.com",
      "user_invited": null,
      "created_at": "2023-02-19T08:12:18.168744-03:00",
      "updated_at": "2023-02-19T08:12:18.168893-03:00"
    }
  ]
}
```

POST: /users/invites

- Permissões necessárias: Autenticado.
- Exemplo de requisição:

```
{
  "name": "convidado",
  "email": "user@example.com"
}
```

- Exemplo de resposta - 201:

```
{
  "id": 2,
  "name": "convidado",
  "email": "user@example.com",
  "user_invited": null,
  "created_at": "2023-02-19T08:12:18.168744-03:00",
  "updated_at": "2023-02-19T08:12:18.168893-03:00"
}
```

GET: /users/invites/{id}

- Permissões necessárias: Autenticado.
- Exemplo de resposta - 200:

```
{
  "id": 2,
  "name": "convidado",
  "email": "convidado@example.com",
  "user_invited": null,
  "created_at": "2023-02-19T08:12:18.168744-03:00",
  "updated_at": "2023-02-19T08:12:18.168893-03:00"
}
```

```

    "updated_at": "2023-02-19T08:14:27.315645-03:00"
  }
}

```

PUT: /users/invites/{id}/

- Permissões necessárias: Administrador.
- Exemplo de requisição;

```
{
  "name": "convidado",
  "email": "convidado@example.com"
}
```

- Exemplo de resposta - 200:

```
{
  "id": 2,
  "name": "convidado",
  "email": "convidado@example.com",
  "user_invited": null,
  "created_at": "2023-02-19T08:12:18.168744-03:00",
  "updated_at": "2023-02-19T08:14:27.315645-03:00"
}
```

DELETE: /users/invites/{id}/

- Permissões necessárias: Administrador.
- Exemplo de resposta - 204:

Autenticação

POST: /authenticate

- Permissões necessárias: Pública.
- Exemplo de requisição:

```
{
  "username": "admin",
  "password": "admin"
}
```

- Exemplo de resposta - 201:

```
{
  "token": {
    "access_token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ0b2tflbl90eXBIIjoieWVnZjZNZDcwODczLCJpYXQiOjE2NzY4MDU0NDMzMStkbnQyLnRlLW5kaS5jb2F0eSBvbiBvdmludGVuZS5kaSI6ImV4cCJ9.eyJ0b2tflbl90eXBIIjoicmVmcmVzaCI6ImV4cCMTY3NzQxMDI3MywiaWF0IjoxNjc2ODA1NDczfQ.",
    "refresh_token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ0b2tflbl90eXBIIjoicmVmcmVzaCI6ImV4cCMTY3NzQxMDI3MywiaWF0IjoxNjc2ODA1NDczfQ."
  }
}
```

DELETE: /authenticate/{id}/

- Permissões necessárias: Autenticado.
- Exemplo de resposta - 204:

POST: /authenticate/refresh/

- Permissões necessárias: Pública.
- Exemplo requisição - 200:

```
{
  "refresh": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ0b2t1b190eXB1IjoicmVmcmVzaCI6ImV4cCI6MTY3NzQxMDI3MywiaWF0IjoxNjc2ODAxNDczLCJqdGki
}
```

- Exemplo de resposta - 200:

```
{
  "token": {
    "access_token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ0b2t1b190eXB1IjoicmVmcmVzaCI6ImV4cCI6MTY3NzQxMDQ0NCwiaWF0IjoxNjc2ODAxNjQ0
    "refresh_token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ0b2t1b190eXB1IjoicmVmcmVzaCI6ImV4cCI6MTY3NzQxMDQ0NCwiaWF0IjoxNjc2ODAxNjQ0
  }
}
```

Negociação**GET: /deals**

- Permissões necessárias: Autenticado.
- Exemplo de resposta - 200:

```
{
  "count": 1,
  "next": null,
  "previous": null,
  "results": [
    {
      "id": 3,
      "location": {
        "id": 11,
        "created_at": "2023-02-19T08:28:20.114911-03:00",
        "updated_at": "2023-02-19T08:28:20.114957-03:00",
        "lat": -15.79412409308632,
        "lng": -47.86663336357427,
        "address": "Palácio do Congresso Nacional - Praça dos 3 Poderes, Brasília - DF, 70160-900",
        "city": "Brasília",
        "state": "DF",
        "zip_code": "70160900"
      },
      "product": {
        "id": 3,
        "pictures": [
          "http://127.0.0.1:8000/media/pictures/42981fc5-7147-418e-ab82-c358d202da84.jpg"
        ],
        "created_at": "2023-02-19T08:28:20.118260-03:00",
        "updated_at": "2023-02-19T08:28:20.118305-03:00",
        "description": "Mouse antigo",
        "weight": 50,
        "length": 50,
        "height": 50,
        "width": 100,
        "diameter": 100
      },
      "created_at": "2023-02-19T08:28:20.222895-03:00",
      "updated_at": "2023-02-19T08:28:20.223002-03:00",
      "type": 1,
      "value": 10,
      "trade_for": "Teclado",
      "urgency": 1,
      "limit_date": "2024-02-19"
    }
  ]
}
```



```
}
]
}
```

POST: /deals

- Permissões necessárias: Autenticado.
- Exemplo de requisição:

```
{
  "location": {
    "lat": -15.79412409308632,
    "lng": -47.86663336357427,
    "address": "Palácio do Congresso Nacional - Praça dos 3 Poderes, Brasília - DF, 70160-900",
    "city": "Brasília",
    "state": "DF",
    "zip_code": "70160900"
  },
  "product": {
    "pictures": [
      "..."
    ],
    "description": "Mouse antigo",
    "weight": 50,
    "length": 50,
    "height": 50,
    "width": 100,
    "diameter": 100
  },
  "type": 1,
  "value": 10,
  "trade_for": "Teclado",
  "urgency": 1,
  "limit_date": "2024-02-19"
}
```

- Exemplo de resposta - 201:

```
{
  "id": 3,
  "location": {
    "id": 11,
    "created_at": "2023-02-19T08:28:20.114911-03:00",
    "updated_at": "2023-02-19T08:28:20.114957-03:00",
    "lat": -15.79412409308632,
    "lng": -47.86663336357427,
    "address": "Palácio do Congresso Nacional - Praça dos 3 Poderes, Brasília - DF, 70160-900",
    "city": "Brasília",
    "state": "DF",
    "zip_code": "70160900"
  },
  "product": {
    "id": 3,
    "pictures": [
      "http://127.0.0.1:8000/media/pictures/42981fc5-7147-418e-ab82-c358d202da84.jpg"
    ],
    "created_at": "2023-02-19T08:28:20.118260-03:00",
    "updated_at": "2023-02-19T08:28:20.118305-03:00",
    "description": "Mouse antigo",
    "weight": 50,
    "length": 50,
    "height": 50,
    "width": 100,
    "diameter": 100
  },
  "created_at": "2023-02-19T08:28:20.222895-03:00",
  "updated_at": "2023-02-19T08:28:20.223002-03:00",
  "type": 1,
  "value": 10,
  "trade_for": "Teclado",
  "urgency": 1,
  "limit_date": "2024-02-19"
}
```

GET: /deals/{id}/

- Permissões necessárias: Autenticado.
- Exemplo de resposta - 200:

```
{
  "id": 3,
  "location": {
    "id": 11,
    "created_at": "2023-02-19T08:28:20.114911-03:00",
    "updated_at": "2023-02-19T08:28:20.114957-03:00",
    "lat": -15.79412409308632,
    "lng": -47.86663336357427,
    "address": "Palácio do Congresso Nacional - Praça dos 3 Poderes, Brasília - DF, 70160-900",
    "city": "Brasília",
    "state": "DF",
    "zip_code": "70160900"
  },
  "product": {
    "id": 3,
    "pictures": [
      "http://127.0.0.1:8000/media/pictures/42981fc5-7147-418e-ab82-c358d202da84.jpg"
    ],
    "created_at": "2023-02-19T08:28:20.118260-03:00",
    "updated_at": "2023-02-19T08:28:20.118305-03:00",
    "description": "Mouse antigo",
    "weight": 50,
    "length": 50,
    "height": 50,
    "width": 100,
    "diameter": 100
  },
  "created_at": "2023-02-19T08:28:20.222895-03:00",
  "updated_at": "2023-02-19T08:28:20.223002-03:00",
  "type": 1,
  "value": 10,
  "trade_for": "Teclado",
  "urgency": 1,
  "limit_date": "2024-02-19"
}
```

PUT: /deals/{id}/

- Permissões necessárias: Autenticado.
- Exemplo de requisição;

```
{
  "location": {
    "lat": -15.79412409308632,
    "lng": -47.86663336357427,
    "address": "Palácio do Congresso Nacional - Praça dos 3 Poderes, Brasília - DF, 70160-900",
    "city": "Brasília",
    "state": "DF",
    "zip_code": "70160900"
  },
  "product": {
    "pictures": [],
    "description": "Mouse antigo",
    "weight": 50,
    "length": 50,
    "height": 50,
    "width": 100,
    "diameter": 0
  },
  "type": 1,
  "value": 10,
  "trade_for": "Teclado",
  "urgency": 1,
  "limit_date": "2024-02-19"
}
```

- Exemplo de resposta - 200:

```
{
  "id": 3,
  "location": {
    "id": 11,
    "created_at": "2023-02-19T08:28:20.114911-03:00",
    "updated_at": "2023-02-19T08:35:31.381797-03:00",
    "lat": -15.79412409308632,
    "lng": -47.86663336357427,
    "address": "Palácio do Congresso Nacional - Praça dos 3 Poderes, Brasília - DF, 70160-900",
    "city": "Brasília",
    "state": "DF",
    "zip_code": "70160900"
  },
  "product": {
    "id": 3,
    "pictures": [
      "http://127.0.0.1:8000/media/pictures/42981fc5-7147-418e-ab82-c358d202da84.jpg"
    ],
    "created_at": "2023-02-19T08:28:20.118260-03:00",
    "updated_at": "2023-02-19T08:35:31.390300-03:00",
    "description": "Mouse antigo",
    "weight": 50,
    "length": 50,
    "height": 50,
    "width": 100,
    "diameter": 0
  },
  "created_at": "2023-02-19T08:28:20.222895-03:00",
  "updated_at": "2023-02-19T08:35:31.399495-03:00",
  "type": 1,
  "value": 10,
  "trade_for": "Teclado",
  "urgency": 1,
  "limit_date": "2024-02-19"
}
```

DELETE: /deals/{id}

- Permissões necessárias: Autenticado.
- Exemplo de resposta - 204:

```
204
```

Ofertas

GET: /deals/{deal_id}/bids

- Permissões necessárias: Autenticado.
- Exemplo de resposta - 200:

```
{
  "count": 1,
  "next": null,
  "previous": null,
  "results": [
    {
      "id": 1,
      "accepted": false,
      "value": 25,
      "description": "faço 25",
      "user": 1,
      "created_at": "2023-02-19T08:37:18.285865-03:00",
      "updated_at": "2023-02-19T08:37:18.285932-03:00"
    }
  ]
}
```

POST: /deals/{deal_id}/bids

- Permissões necessárias: Autenticado.
- Exemplo de requisição:

```
{
  "accepted": false,
  "value": 25,
  "description": "faço 25"
}
```

- Exemplo de resposta - 201:

```
{
  "id": 1,
  "accepted": false,
  "value": 25,
  "description": "faço 25",
  "user": 1,
  "created_at": "2023-02-19T08:37:18.285865-03:00",
  "updated_at": "2023-02-19T08:37:18.285932-03:00"
}
```

GET: /deals/{deal_id}/bids/{id}/

- Permissões necessárias: Autenticado.
- Exemplo de resposta - 200:

```
{
  "id": 1,
  "accepted": false,
  "value": 25,
  "description": "faço 25",
  "user": 1,
  "created_at": "2023-02-19T08:37:18.285865-03:00",
  "updated_at": "2023-02-19T08:37:18.285932-03:00"
}
```

PUT: /deals/{deal_id}/bids/{id}/

- Permissões necessárias: Autenticado.
- Exemplo de requisição;

```
{
  "accepted": true,
  "value": 40,
  "description": "faço 40"
}
```

- Exemplo de resposta - 200:

```
{
  "id": 1,
  "accepted": true,
  "value": 40,
  "description": "faço 40",
  "user": 1,
  "created_at": "2023-02-19T08:37:18.285865-03:00",
  "updated_at": "2023-02-19T08:41:40.180818-03:00"
}
```

DELETE: /deals/{deal_id}/bids/{id}/

- Permissões necessárias: Autenticado.
- Exemplo de resposta - 204:

```
204
```

Mensagens

GET: /deals/{deal_id}/messages

- Permissões necessárias: Autenticado.
- Exemplo de resposta - 200:

```
{
  "count": 1,
  "next": null,
  "previous": null,
  "results": [
    {
      "id": 1,
      "title": "Oferta de 25",
      "message": "25 é muito pouco",
      "user": 1,
      "created_at": "2023-02-19T08:43:02.980750-03:00",
      "updated_at": "2023-02-19T08:43:02.980850-03:00"
    }
  ]
}
```

POST: /deals/{deal_id}/messages

- Permissões necessárias: Autenticado.
- Exemplo de requisição:

```
{
  "title": "Oferta de 25",
  "message": "25 é muito pouco"
}
```

- Exemplo de resposta - 201:

```
{
  "id": 1,
  "title": "Oferta de 25",
  "message": "25 é muito pouco",
  "user": 1,
  "created_at": "2023-02-19T08:43:02.980750-03:00",
  "updated_at": "2023-02-19T08:43:02.980850-03:00"
}
```

GET: /deals/{deal_id}/messages/{id}

- Permissões necessárias: Autenticado.
- Exemplo de resposta - 200:

```
{
  "id": 1,
  "title": "Oferta de 25",
  "message": "25 é muito pouco!!!",
  "user": 1,
  "created_at": "2023-02-19T08:43:02.980750-03:00",
  "updated_at": "2023-02-19T08:43:02.980850-03:00"
}
```

```
"updated_at": "2023-02-19T08:45:14.494391-03:00"
}
```

PUT: /deals/{deal_id}/messages/{id}/

- Permissões necessárias: Autenticado.
- Exemplo de requisição;

```
{
  "title": "Oferta de 25",
  "message": "25 é muito pouco!!!"
}
```

- Exemplo de resposta - 200:

```
{
  "id": 1,
  "title": "Oferta de 25",
  "message": "25 é muito pouco!!!",
  "user": 1,
  "created_at": "2023-02-19T08:43:02.980750-03:00",
  "updated_at": "2023-02-19T08:45:14.494391-03:00"
}
```

DELETE: /deals/{deal_id}/messages/{id}/

- Permissões necessárias: Autenticado.
- Exemplo de resposta - 204:

```
204
```

Frete

GET: /deals/{deal_id}/deliveries

- Permissões necessárias: Autenticado.
- Exemplo de resposta - 200:

```
{
  "code": "40010",
  "value": 869.8,
  "own_hand_value": 0,
  "value_notice_receipt": 0,
  "declared_value": 0,
  "deadline": 1,
  "value_without_additional": 869.8,
  "home_delivery": true,
  "delivery_in_saturday": true
}
```

Pagamento

GET: /deals/{deal_id}/payments

- Permissões necessárias: Autenticado.
- Exemplo de resposta - 200:

```
{
  "id": 1,
  "created_at": "2023-02-19T09:02:24.843446-03:00",
  "updated_at": "2023-02-19T09:02:24.843589-03:00",
  "user": 1,
  "deal": 3
}
```

POST: /deals/{deal_id}/payments

- Permissões necessárias: Autenticado.
- Exemplo de resposta - 200:

```
{
  "id": 1,
  "created_at": "2023-02-19T09:02:24.843446-03:00",
  "updated_at": "2023-02-19T09:02:24.843589-03:00",
  "user": 1,
  "deal": 3
}
```