### Projeto de Desenvolvimento de uma API para Agendamentos de Pagamentos Utilizando Django

#### Resumo

Este projeto tem como objetivo o desenvolvimento de uma API utilizando o framework Django para gerenciar agendamentos de pagamentos. A API permitirá a criação, listagem, consulta e exclusão de agendamentos.

#### 1. Introdução

##### 1.1 Contexto

Com o avanço da tecnologia e a necessidade de gerenciar pagamentos de maneira eficiente, o desenvolvimento de uma API para agendamentos de pagamentos torna-se crucial para empresas que lidam com diversos clientes e transações financeiras.

##### 1.2 Objetivo

O objetivo deste projeto é criar uma API RESTful utilizando Django que permita aos usuários gerenciar agendamentos de pagamentos de maneira eficiente e segura.

#### 2. Revisão da Literatura

##### 2.1 Django

Django é um framework web de alto nível para Python, que permite o desenvolvimento rápido e o design limpo e pragmático. Ele oferece uma arquitetura robusta para desenvolvimento de aplicações web.

##### 2.2 APIs RESTful

APIs RESTful são um padrão para a construção de APIs baseadas em HTTP. Elas utilizam métodos HTTP (GET, POST, PUT, DELETE) e permitem interações de maneira simples e escalável.

#### 3. Metodologia

##### 3.1 Ferramentas Utilizadas

- Python 3.x

- Django 3.x ou superior

- Django REST framework

- SQLite (para ambiente de desenvolvimento)

- VS Code (ou qualquer outro editor de texto)

- Postman (para testes de API)

#### 4. Desenvolvimento

##### 4.1 Configuração do Ambiente

1. \*\*Instalação do Python\*\*:

Baixe e instale a versão mais recente do Python em [python.org](https://www.python.org/).

2. \*\*Configuração do Virtual Environment\*\*:

```bash

python -m venv env

source env/bin/activate # No Windows: env\Scripts\activate

```

3. \*\*Instalação do Django e Django REST framework\*\*:

```bash

pip install django djangorestframework

```

##### 4.2 Criação do Projeto Django

1. \*\*Criar o projeto\*\*:

```bash

django-admin startproject PagamentosAPI

cd PagamentosAPI

```

2. \*\*Criar o aplicativo\*\*:

```bash

python manage.py startapp agendamentos

```

3. \*\*Adicionar o aplicativo no settings.py\*\*:

```python

INSTALLED\_APPS = [

...

'agendamentos',

'rest\_framework',

]

```

##### 4.3 Definição dos Modelos

Em `agendamentos/models.py`:

```python

from django.db import models

class Agendamento(models.Model):

data = models.DateField()

descricao = models.CharField(max\_length=255)

def \_\_str\_\_(self):

return f"{self.descricao} em {self.data}"

```

##### 4.4 Criação e Aplicação das Migrações

```bash

python manage.py makemigrations agendamentos

python manage.py migrate

```

##### 4.5 Configuração do Admin

Em `agendamentos/admin.py`:

```python

from django.contrib import admin

from .models import Agendamento

admin.site.register(Agendamento)

```

##### 4.6 Criação dos Serializers

Em `agendamentos/serializers.py`:

```python

from rest\_framework import serializers

from .models import Agendamento

class AgendamentoSerializer(serializers.ModelSerializer):

class Meta:

model = Agendamento

fields = '\_\_all\_\_'

```

##### 4.7 Criação das Views

Em `agendamentos/views.py`:

```python

from rest\_framework import viewsets

from .models import Agendamento

from .serializers import AgendamentoSerializer

class AgendamentoViewSet(viewsets.ModelViewSet):

queryset = Agendamento.objects.all()

serializer\_class = AgendamentoSerializer

```

##### 4.8 Configuração das URLs

Em `agendamentos/urls.py`:

```python

from django.urls import path, include

from rest\_framework.routers import DefaultRouter

from .views import AgendamentoViewSet

router = DefaultRouter()

router.register(r'agendamentos', AgendamentoViewSet)

urlpatterns = [

path('', include(router.urls)),

]

```

Em `PagamentosAPI/urls.py`:

```python

from django.contrib import admin

from django.urls import path, include

urlpatterns = [

path('admin/', admin.site.urls),

path('api/', include('agendamentos.urls')),

]

```

##### 4.9 Testes

Para testar a API, use o Postman ou Insomnia para fazer requisições para os endpoints:

- \*\*Listar agendamentos\*\*: `GET http://127.0.0.1:8000/api/agendamentos/`

- \*\*Criar agendamento\*\*: `POST http://127.0.0.1:8000/api/agendamentos/` com um corpo JSON como:

```json

{

"data": "2024-09-13",

"descricao": "Reunião com o cliente"

}

```

##### 4.10 Execução do Servidor

Para iniciar o servidor, use:

```bash

python manage.py runserver

```

#### 5. Conclusão

Este projeto demonstrou a criação de uma API RESTful utilizando Django, com funcionalidades completas para gerenciar agendamentos de pagamentos. O uso do Django REST framework facilitou a construção da API, proporcionando um desenvolvimento rápido e eficiente.

#### 6. Referências

- Documentação oficial do Django: [https://docs.djangoproject.com/](https://docs.djangoproject.com/)

- Documentação do Django REST framework: [https://www.django-rest-framework.org/](https://www.django-rest-framework.org/)

---