Desafio da Ceos

Equipe: Ernesto Dalva, Gabriel Pinheiro, Rivânio Filho, Thiago Nogueira

1. *Sobre o Projeto*

No início dividimos a equipe em duas partes, Front-end: Ernesto e Thiago, e Back-end: Gabriel e Rivânio. Fizemos uma reunião, onde foram decididas as intenções do projeto, bem como as tecnologias e como faríamos as integrações. Quanto às funcionalidades, foi decidido que o projeto seria uma TO-DO List, contendo as atividades e seus atributos, uma aplicação para autenticação (registro e login), barra de pesquisa e as operações do CRUD (GET, POST, PUT, DELETE). Sobre as tecnologias, o back-end será implementado usando Django REST e o front-end em Angular.

2. *Progresso*

Criamos um repositório no GitHub (link_repositório), onde será feito os commits. Criamos a pasta para realizar o back-end, iniciamos um ambiente virtual, onde instalamos o Django e o Django REST e modelamos o objeto Task, que representa as nossas tarefas na TO-DO List.

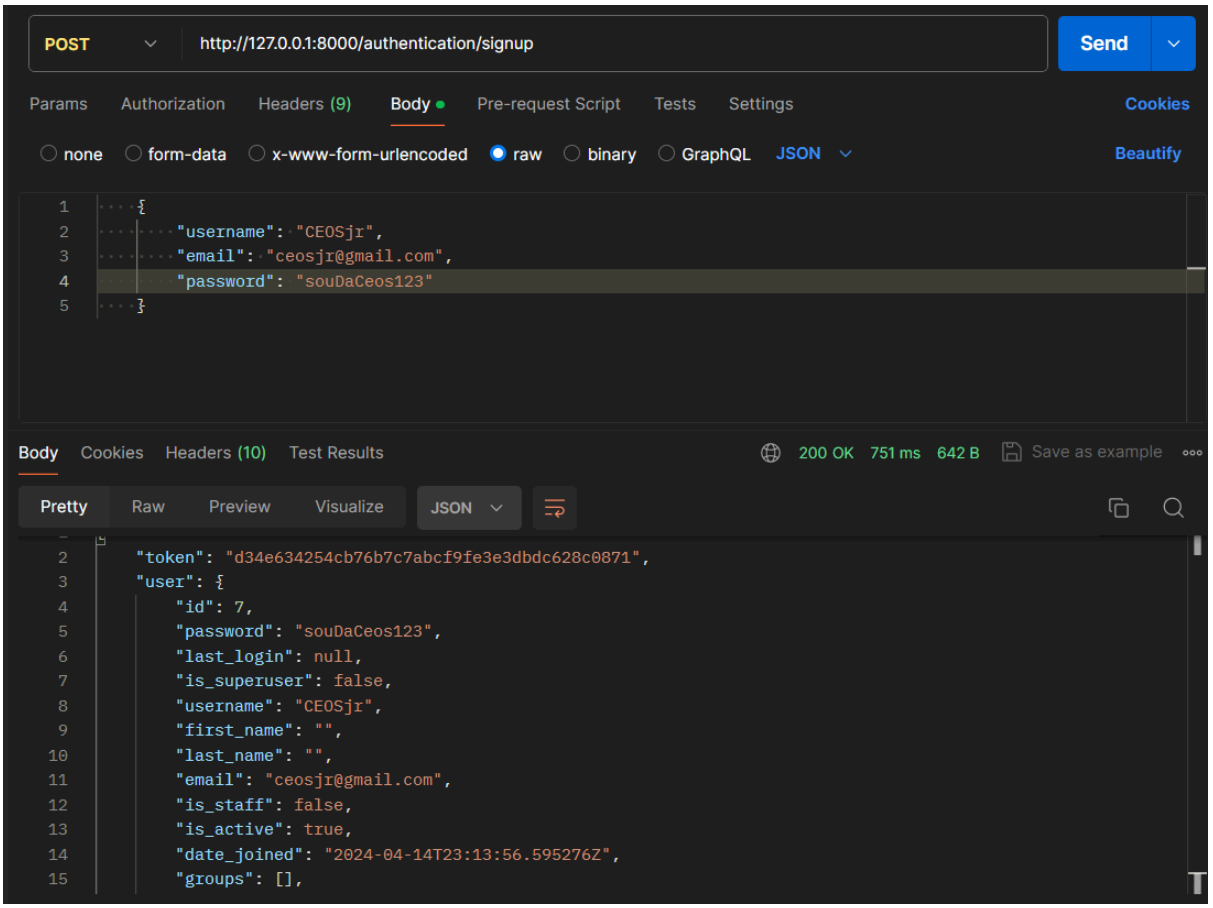
```
class Task(models.Model):
    user = models.ForeignKey(
        User, on_delete=models.CASCADE, null=True, blank=True)
    title = models.CharField(max_length=200)
    create = models.DateTimeField(auto_now_add=True)
    task_date = models.DateTimeField(default=timezone.now)
    category = models.CharField(max_length=150)
    complete = models.BooleanField(default=False)

    def __str__(self):
        return self.title

    class Meta:
        ordering = ['complete']
```

classe Task, representando as Tarefas e seus atributos

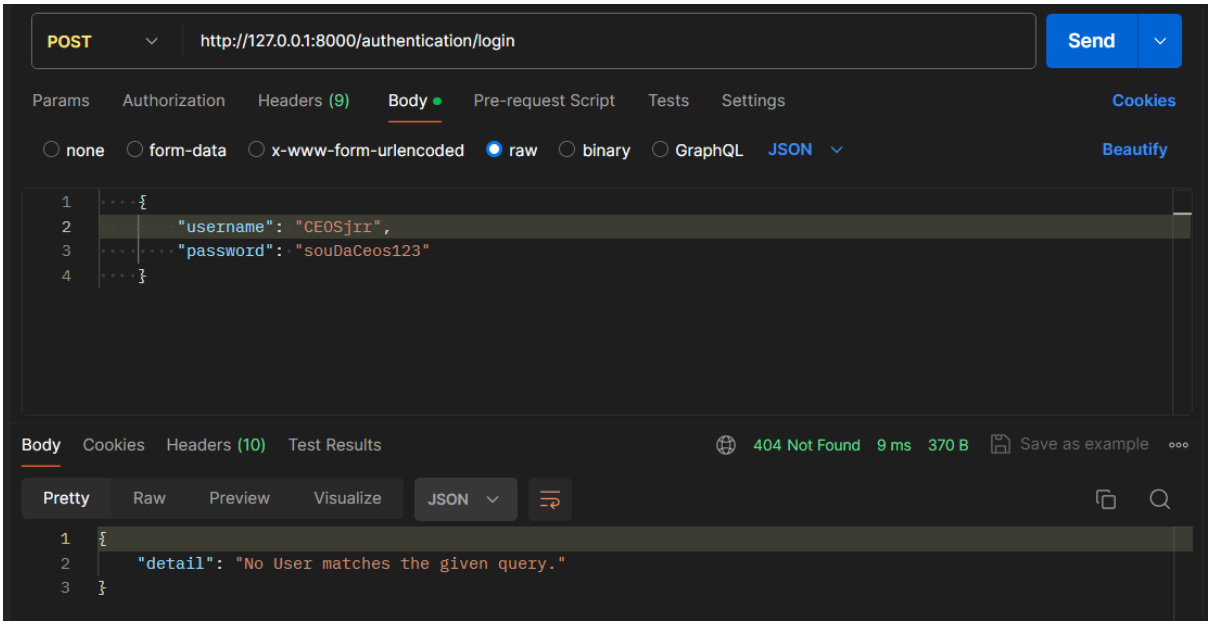
Os campos são: Nome do usuário(como Chave estrangeira, se relaciona com as tarefas pela forma many-to-one, se excluirmos o usuário, todas as tarefas dele serão excluídas também), Título da tarefa, Data de criação, Data da tarefa(por

padrão é a Data de criação, se o usuário não preencher o campo), Categoria da tarefa, Campo para verificação do estado da tarefa(completa ou não).

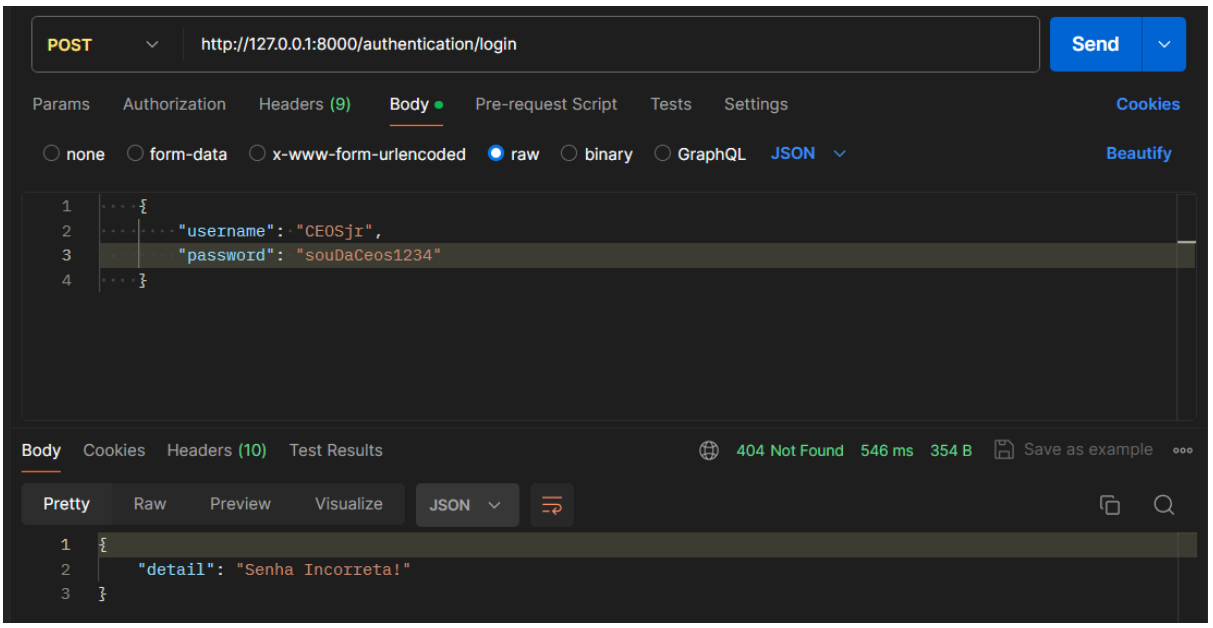
Ademais, foi criada uma app no Django para fazer a autenticação do usuário, com Sign-Up (que pede usuário, email e senha), login (que pede usuário e senha de uma conta já criada) e uma função para testar a autenticidade do Token do usuário.



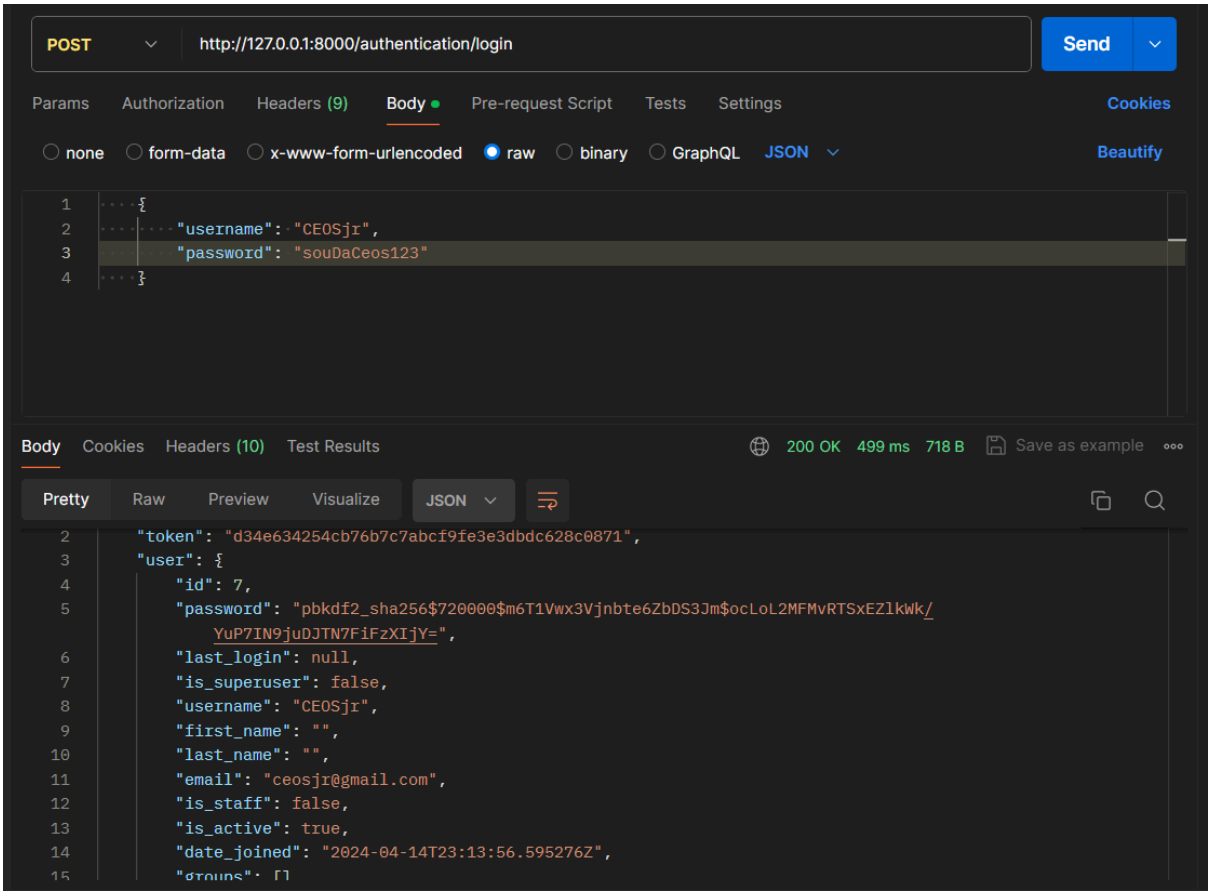
criando usuário



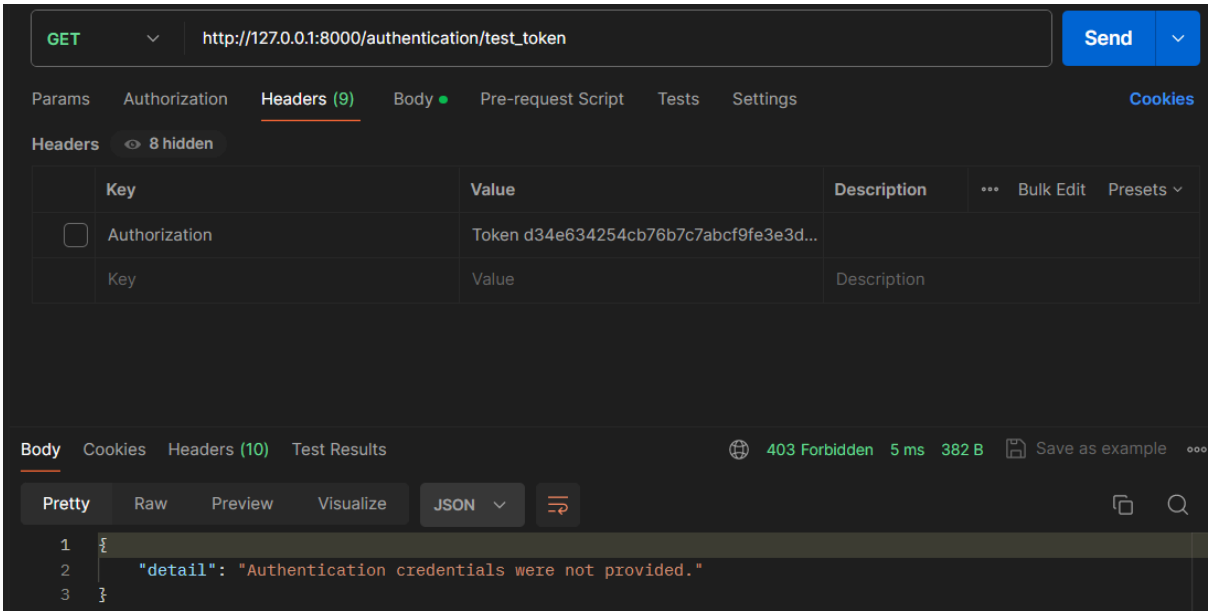
errando login por usuário



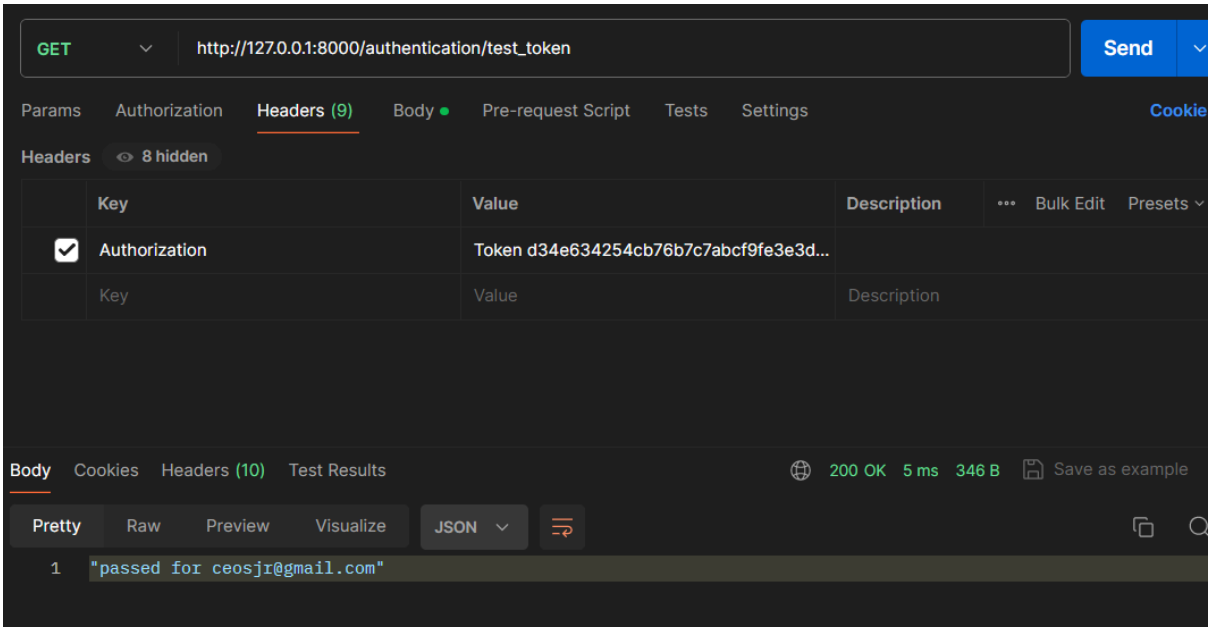
errando login por senha



efetuando login



utilizando a função de teste (test_token), mas não fornecendo as credenciais



autenticado pela função test_token

3. *Futuro*

Para o futuro, esperamos implementar toda a parte do front-end, bem como sua integração com o back-end. Além disso, iremos implementar o CRUD no back-end, fazer a ligação de um usuário com suas tarefas, adicionar a barra de pesquisa das tarefas e tentaremos filtrá-las por categoria.