

Go and Slay - Especificação e Modelagem do Sistema

Gabriel Ribeiro Filice Chayb – Teste para vaga de Desenvolvedor Web Jr.

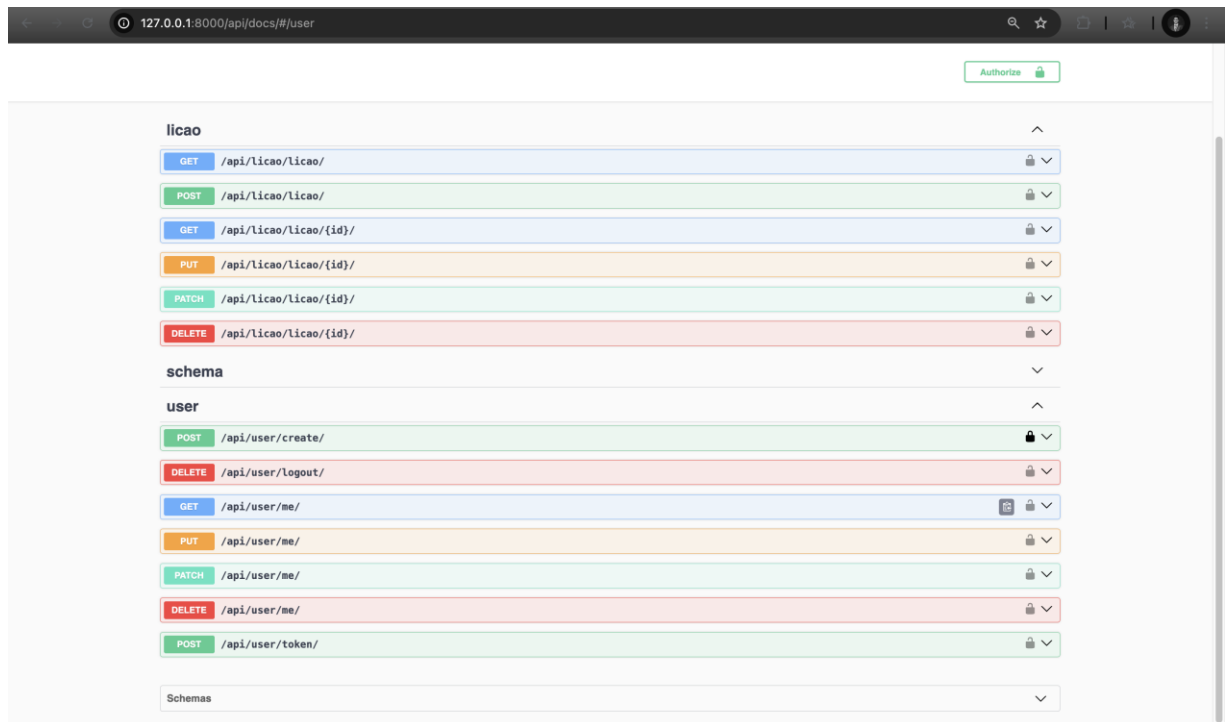
Avaliadores: Pedro e João

Tema: REST APIs para cadastro de usuário no sistema e lições de inglês

Descrição do sistema:

Uma plataforma de ensino de inglês que permite aos *users* fazerem login usando autenticação baseada em tokens para gerenciar suas *licoes* e visualizar as *licoes* de outros *users*, incluindo um id único para cada *licao*, título, conteúdo, tempo em minutos para ler e aprender, preço em R\$ e um link para acessar mais informações. Além disso, a plataforma permite, através de seus endpoints, que:

- O *user* se cadastre e faça login autenticado via Django Token Auth;
- Que o *user* autenticado, na home, consiga visualizar, editar suas informações de perfil e deletar seu perfil.
- Que o *user* autenticado, consiga visualizar as *licoes* cadastradas por qualquer *user* na plataforma
- Que o *user* autenticado, consiga editar ou deletar as *licoes* que ele mesmo cadastrou, via id;
- Que ao editar uma *licao* com sucesso, um script Javascript aparece na sua tela em forma de alert, avisando assim o *user* de sua bem-sucedida alteração.
- Na rota /api/docs , por meio da ferramenta SWAGGER UI, é possível ver todos os endpoints, simular todas as requisições reais via curl, autenticar manualmente o login de um *user* com um tokenAuth. Ela se comporta da seguinte maneira:

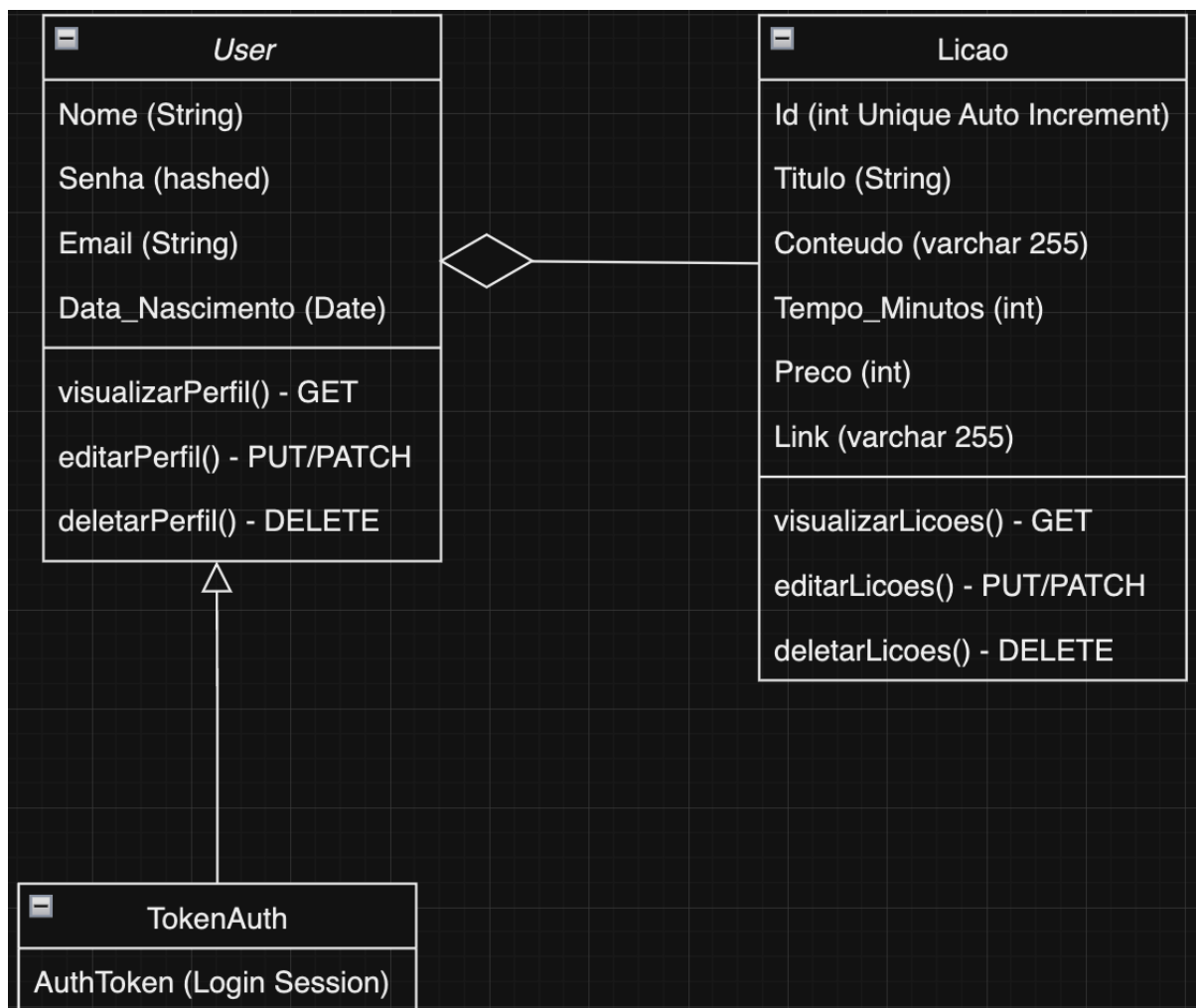


Processo de desenvolvimento:

Além de todas as funcionalidades descritas acima, pedidas no teste, incluindo os requisitos obrigatórios e adicionais, também foram desenvolvidas outras partes como forma de **overdelivery**. A seguir, uma lista delas:

- **BÔNUS:** A aplicação inteira está containerizada via Docker, o que permite um ambiente de desenvolvimento local e em produção de qualidade, segurança e sem erros de versionamento ou importação.
- **BÔNUS:** A aplicação também está configurada para o deploy, pois já conta com todas as configurações necessárias para ser servida, como: uWSGI, configurações de proxy, Dockerfile de proxy, docker-compose-deploy e configurações de host em settings.py.
- **BÔNUS:** A aplicação conta com um front-end personalizado para a Go And Slay, com copy e cores características de acordo com a identidade visual da empresa.
- **BÔNUS:** A aplicação também está responsiva e otimizada para todas as faixas de tela: desktop, tablet e dispositivos móveis.

- **BÔNUS:** O repositório conta com uma documentação completa além da tradicional (README.md). Ela inclui o arquivo DOC.md, que explica cada pasta e arquivo dentro do sistema, além de uma documentação completa em PDF.
- **BÔNUS:** Foi desenvolvido e explicado um diagrama de classes com base nos paradigmas da programação orientada a objetos, que se encontra logo abaixo:



JUSTIFICATIVA BASEADA EM POO:

- O modelo *User* possui 4 atributos principais: nome, senha, e-mail e data de nascimento. Assim que ele realiza o login, ele recebe um Token via Django Token Auth, que é validado enquanto a sessão dele é ativa.
- O modelo *licao* possui 5 atributos principais e um ID: título, conteúdo, o tempo em minutos, o preço em R\$ e o link para mais informações.

- O modelo *licao* se enquadra como uma composição do modelo *User* (por isso ele implica a composição em *User*, e não o contrário, pois não existe *licao* sem *User*, mas existe *User* sem *licao*). Isso ocorre pois toda *licao* é cadastrada por um *User*, sendo identificada da seguinte maneira: se o login do *user* criador da *licao* for o mesmo login do *user* que quer editar ou deletar a *licao*, a requisição é aceita. Caso contrário, é negada.