Módulo: Gestor do sistema

A base de dados do sistema foi criada utilizando SQLite e é gerida por uma API construída em Flask. A API funciona assim como um gestor do sistema que permite a troca de dados entre a base de dados e a interface do utilizador.

Na inicialização da base de dados foram adicionados exercícios e planos padrão. Estes dados encontram-se disponíveis em três pastas, para imagens, vídeos e por último uma com informação em documentos Excel. Para inserir esses dados na base de dados, foram criados três ficheiros separados: um para processamento de dados dos documentos Excel, outro para processamento de imagens e um terceiro para processamento de vídeos.

Os vídeos necessitam de ser guardados no disco da máquina de cada utilizador (pasta de vídeos), sendo estes depois obtidos a partir da API, recorrendo ao seu nome mais o tipo de ficheiro, que se encontra na base de dados. Os restantes dados, incluindo as imagens, são guardados diretamente na base de dados.

Em ambas as tabelas de vídeo e imagens encontramos a chave estrangeira que faz referência ao exercício a que cada vídeo ou imagem está associado.

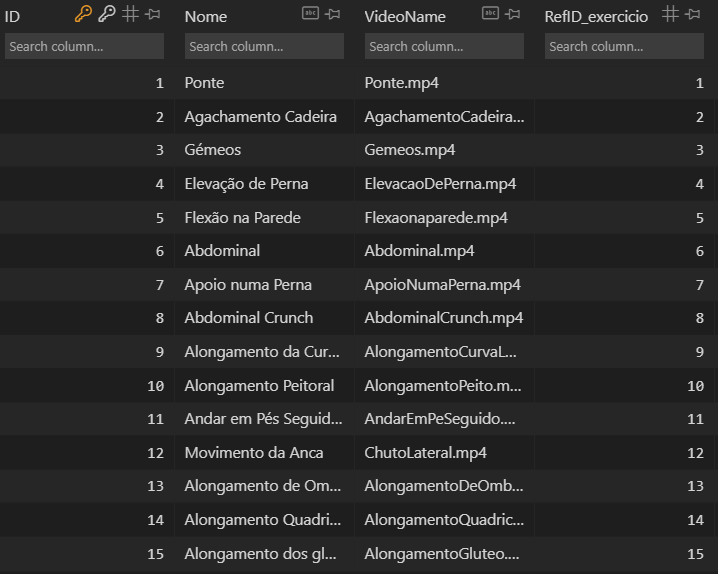


Figura - Tabela "Video"

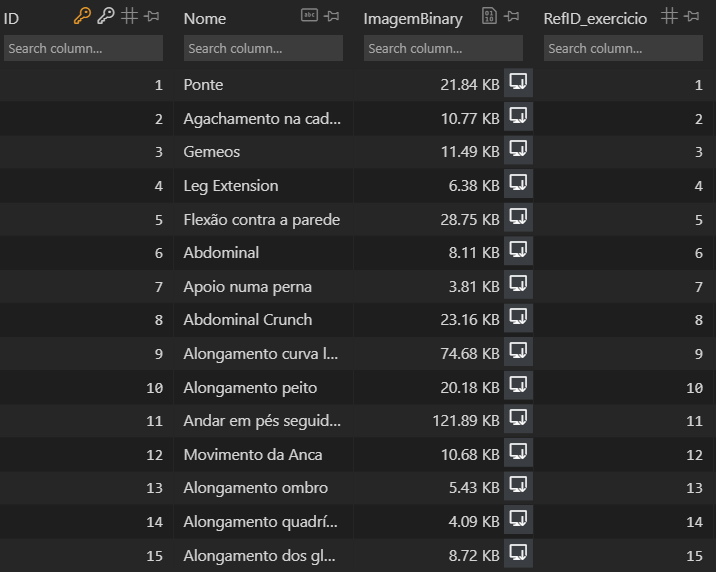


Figura - Tabela "Imagem"

Para iniciar a API, basta executar o ficheiro main.py, que iniciará o servidor Flask. A API oferece diversas operações para aceder e modificar os dados na base de dados usando queries SQL. Algumas das funcionalidades disponíveis incluem obter a lista completa de exercícios, as sessões e os planos associados a um usuário em específico, inserir e remover exercícios da base de dados, entre outras. Para aceder a estas funcionalidades, são enviadas solicitações HTTP para a API.

Por exemplo, para obter a lista completa de exercícios, presentes na tabela “Exercicio”, a interface envia uma solicitação GET para a rota /getExercises. Já para obter as sessões e o plano associado a cada sessão de um utilizador específico, pode ser enviada uma solicitação GET para a rota /profile/user\_id=<user\_id>, onde “user\_id” é o identificador único do utilizador na base de dados. Para a última solicitação, a api acede à tabela “Sessao”, que contém duas chaves estrangeiras, uma para o utilizador e outra para o plano. Sabendo o plano da sessão, falta obter os exercícios desse plano e para tal criámos uma tabela extra, “ExercicioPlano”, que faz a ligação da tabela “Plano” com a de “Exercicio”.



Figura – Tabela “Exercicio”



Figura – Tabela “Sessao”

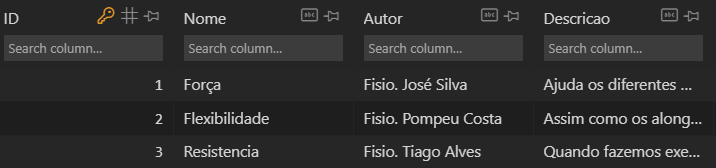


Figura – Tabela “Plano”

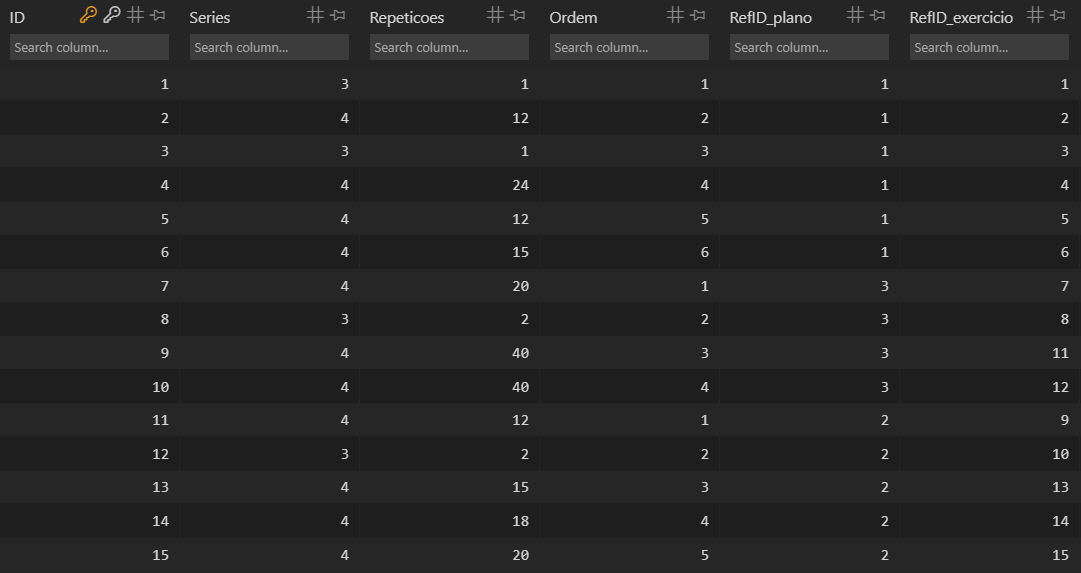


Figura - Tabela "ExercicioPlano"

Para inserir um novo exercício na base de dados, pode ser enviada uma solicitação POST para a rota /addExercise com as informações do exercício no corpo da solicitação. Da mesma forma, para remover um exercício existente, pode ser enviada uma solicitação POST para a rota /removeExercise com o “id” do exercício a remover, sendo este o identificador único do exercício na base de dados, e depois serão realizados vários DELETE’s para a apagar toda a informação do exercício. Estas duas funcionalidades estão disponíveis, para já, numa interface à parte da principal, acessível pelo link constituído pelo endereço público ou privado da máquina em conjunto com /addExercise ou /removeExercise.

O processo de inserção de exercícios é uma função importante para o sistema. Este representa a possibilidade de fisioterapeutas adicionarem exercícios adequados para cada idoso que possui o sistema, individualmente.

O servidor está instalado localmente em cada casa de um idoso. Para acessá-lo remotamente, é necessário utilizar o endereço IP público da máquina na qual o servidor está hospedado. Para isso, é preciso configurar o router da casa do idoso por forma a aceder ao servidor em execução. Para tal, deve-se inserir as informações sobre o endereço IP privado da máquina e o número da porta que serão utilizados para permitir a ligação ao servidor.

Depois do endereço IP e a porta terem sido configurados no router, o tráfego será então encaminhado para o servidor em execução. Ao aceder à página com o endereço público e a porta correta o router realiza o NAT para o transformar no endereço privado, sendo assim possível aceder ao servidor remotamente.

Por último, as funcionalidades de login, registo, acesso a planos individuais e adição de planos e sessões personalizados já estão implementados, em parte, na API, embora ainda não estejam disponíveis na interface atual. Estas poderão ser usadas futuramente. Foi inclusive feita uma tabela de “Utilizador” para testes:

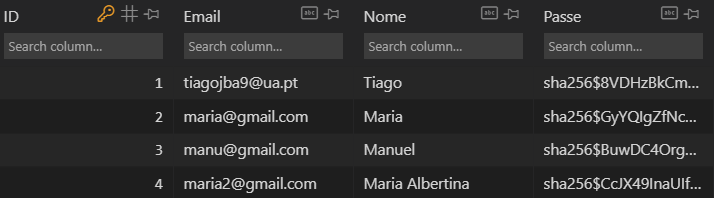


Figura - Tabela "Utilizador"