Relatório Python-Neo4J

Aluno	Guilherme Feitosa
Disciplina	Novas Tecnologias em BD
Data	@22/05/2023

Descrição da aplicação

A aplicação é uma API desenvolvida com o framework FastAPI e utiliza um banco de dados Neo4j como SGBD (Sistema de Gerenciamento de Banco de Dados). A aplicação permite a criação e recuperação de usuários, juntamente com a adição de amigos para cada usuário.

Funcionalidades

- `GET /users`: Recupera todos os usuários cadastrados juntamente com suas listas de amigos.
- `POST /users`: Cria um novo usuário com os campos "name" (nome) e "age" (idade).
- `POST /users/{userName}/add-friend`: Adiciona amigos para um usuário existente. Os amigos são fornecidos como uma lista no corpo da solicitação.
- `PUT /users/{name}`: Atualiza os dados de um usuário existente com base no nome.
- `DELETE /users/{name}`: Deleta um usuário existente com base no nome.

Descrição da arquitetura da solução

A aplicação utiliza o FastAPI para lidar com as requisições HTTP e criar uma API RESTful. O banco de dados Neo4j é acessado por meio do driver oficial neo4j-python. O código define cinco rotas: `getAllUsers` para recuperar todos os usuários, `createUser` para criar um novo usuário, `addFriend` para adicionar amigos a um usuário existente, `updateUser` para atualizar os dados de um usuário existente e `deleteUser` para deletar um usuário existente.

Quando a rota `getAllUsers` é acessada, uma conexão é estabelecida com o banco de dados Neo4j usando o driver, e uma consulta Cypher é executada para recuperar todos

os usuários e seus respectivos amigos. Os resultados são retornados como uma lista de dicionários, onde cada dicionário representa um usuário e sua lista de amigos.

Na rota `createUser`, é verificado se os campos obrigatórios "name" e "age" estão presentes nos dados recebidos. Se estiverem, uma conexão é estabelecida com o banco de dados e uma consulta Cypher é executada para criar um novo nó de usuário com os valores fornecidos.

Na rota `addFriend`, é feito um loop nos amigos fornecidos no corpo da solicitação e, para cada amigo, uma conexão é estabelecida com o banco de dados e uma consulta Cypher é executada para criar uma relação "Friend" entre o usuário existente e o amigo.

Na rota `updateUser`, os dados fornecidos são usados para atualizar os campos "name" e "age" de um usuário existente. Uma conexão é estabelecida com o banco de dados e uma consulta Cypher é executada para realizar a atualização.

Na rota `deleteUser`, um usuário existente é deletado com base no nome fornecido. Uma conexão é estabelecida com o banco de dados e uma consulta Cypher é executada para deletar o nó correspondente ao usuário.

Descrição da interação entre a aplicação e o SGBD:

A aplicação se conecta ao banco de dados Neo4j usando o driver neo4j-python. Quando uma rota é acessada, uma sessão é aberta com o driver para executar as consultas Cypher. As consultas Cypher são enviadas para o banco de dados Neo4j por meio da sessão, e os resultados são retornados para a aplicação. A aplicação utiliza os resultados para construir as respostas adequadas, que são retornadas como JSON nas rotas definidas.

Em resumo, a aplicação interage com o SGBD Neo4j por meio do driver neo4j-python, enviando consultas Cypher para recuperar, criar e modificar dados no banco de dados, conforme necessário, para fornecer as funcionalidades da API.