

Requisitos Funcionais

1. Criar um novo treinamento:

- O requisito é capaz de criar um novo treinamento com base nos parâmetros fornecidos.
- Ele usa uma função para validar e criar o treinamento no banco de dados.

2. Criar um novo usuário:

- O requisito é capaz de criar um novo usuário com base nos parâmetros fornecidos.
- Ele usa uma função para validar e criar o usuário no banco de dados.

3. Obter informações de um usuário:

- O requisito é capaz de obter informações de um usuário com base em um ID fornecido.
- Ele verifica se o ID é um UUID válido e, em seguida, busca o usuário correspondente no banco de dados.
- Se o usuário for encontrado, carrega os treinamentos associados a ele e retorna os dados.

4. Calcular o IMC a partir de um arquivo CSV:

- O requisito é capaz de ler um arquivo CSV e calcular o IMC (Índice de Massa Corporal) de cada pessoa no arquivo.
- O arquivo CSV é fornecido através do parâmetro "filename".

5. Listar todos os treinamentos disponíveis:

- O sistema é capaz de listar todos os treinamentos disponíveis.
- Ele recupera todos os registros de treinamentos do banco de dados e retorna os dados correspondentes.
- Os dados podem incluir informações como nome do treinamento, descrição, instrutor, horário, etc.

6. Agendar um treinamento para um usuário:

- O requisito é capaz de agendar um treinamento para um usuário específico.
- Requer o ID do usuário e o ID do treinamento como parâmetros.
- Verifica se o usuário e o treinamento existem no banco de dados antes de criar o agendamento.
- Cria uma nova entrada de agendamento associando o usuário ao treinamento.

7. Registrar histórico de treinamento de um usuário:

- O requisito é capaz de registrar o histórico de treinamento de um usuário.
- Permite adicionar entradas de histórico de treinamento para um usuário específico.
- O histórico pode incluir informações como data do treinamento, duração, desempenho, etc.
- Armazena as informações no banco de dados, associadas ao usuário correspondente.

8. Gerar relatório de desempenho do usuário:

- O requisito é capaz de gerar um relatório de desempenho para um usuário.
- Utiliza os dados do histórico de treinamento do usuário para calcular métricas relevantes, como progresso, frequência, calorias queimadas, etc.
- Gera um relatório formatado contendo as informações de desempenho e estatísticas.
- O relatório pode ser exportado em diferentes formatos, como PDF ou CSV.

9. Gerenciar planos de pagamento:

- O requisito é capaz de gerenciar os planos de pagamento oferecidos pela academia.
- Permite a criação, atualização e exclusão de planos de pagamento.
- Os planos podem incluir informações como preço, duração, serviços incluídos, etc.
- Armazena os planos de pagamento no banco de dados e os associa aos usuários conforme necessário.

10. Atualizar informações de um usuário:

- O requisito é capaz de atualizar as informações de um usuário com base em um ID fornecido.
- Ele verifica se o ID é um UUID válido e, em seguida, busca o usuário correspondente no banco de dados.
- Permite a modificação dos campos relevantes, como nome, idade, altura, peso, etc.
- Utiliza uma função para validar as alterações e atualizar os dados do usuário no banco de dados.

Requisitos não funcionais

- Os requisitos funcionais que envolvem criar e obter informações de usuários e treinamentos dependem de uma conexão funcional com o banco de dados.
- Os requisitos funcionais que envolvem criar usuários, criar treinamentos e obter informações de usuários dependem de uma validação adequada dos dados fornecidos.
- Os requisitos funcionais que envolvem obter informações de usuários podem depender do desempenho do banco de dados, especialmente se houver uma grande quantidade de dados.
- Os requisitos funcionais que envolvem a leitura de um arquivo CSV e o cálculo do IMC devem ser capazes de lidar com erros de leitura de arquivo e outros possíveis erros durante o processamento dos dados.
- O software deve garantir a segurança dos dados do usuário, protegendo informações pessoais e confidenciais.
- Deve ser implementada uma autenticação robusta para garantir que apenas usuários autorizados possam acessar e modificar os dados.
- O acesso ao banco de dados e a transmissão de dados devem ser protegidos por criptografia.
- O software deve ser intuitivo e fácil de usar para os usuários, independentemente do seu nível de experiência em tecnologia.
- A interface do usuário deve ser bem projetada, com fluxos de trabalho claros e navegação intuitiva.

- A documentação e a ajuda do sistema devem estar disponíveis para orientar os usuários em relação às funcionalidades do software.
- O software deve ser capaz de lidar com um aumento significativo no número de usuários e volume de dados sem comprometer o desempenho.
- Deve ser implementada uma arquitetura escalável, capaz de dimensionar horizontalmente conforme necessário.
- O software deve estar disponível e acessível aos usuários durante a maior parte do tempo.
- Deve ser implementado um plano de backup e recuperação de desastres para garantir a continuidade dos serviços em caso de falhas de hardware ou outros eventos inesperados.
- O software deve ser projetado de forma modular e bem estruturada, facilitando a manutenção e a incorporação de futuras melhorias.
- Deve ser adotada uma abordagem de desenvolvimento orientada a testes para garantir a estabilidade do sistema e facilitar a introdução de alterações.
- O software deve ser capaz de integrar-se com outros sistemas relevantes, como sistemas de pagamento, aplicativos móveis ou dispositivos de rastreamento de atividades.
- Deve ser fornecida uma API (Interface de Programação de Aplicativos) ou outros mecanismos de integração para permitir a comunicação e a troca de dados entre sistemas.

Atributos de qualidade

- Validação de dados
- Desempenho
- Tratamento de erros
- Segurança
- Usabilidade
- Escalabilidade
- Disponibilidade
- Manutenibilidade
- Integração com outros sistemas