

UNIVERSIDADE SÃO JUDAS TADEU
ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

MIRIAM GAMA
FELIPE PETRASSI
GABRIEL CAIQUE

PROJETO A3
Aplicativo MetaFit

SÃO PAULO - SP
2024

MIRIAM GAMA
FELIPE PETRASSI
GABRIEL CAIQUE

PROJETO A3

Aplicativo MetaFit

Trabalho A3 da UC Programas de soluções computacionais, apresentação à Universidade São Judas Tadeu, com o objetivo de conclusão do semestre 2024.2 do curso Análise e Desenvolvimento de Sistemas.

Orientador: Robson Calvetti

SÃO PAULO - SP

2024

Sumário

1. INTRODUÇÃO	4
1.1 TECNOLOGIAS UTILIZADAS	4
1.2 ARQUITETURA MVC(MODEL-VIEW-CONTROLLER).....	5
2. FLUXO	6
3. ESTRUTURA DE PASTAS.....	8
4. REFERÊNCIAS.....	8
5. CONCLUSÃO	8

1. INTRODUÇÃO

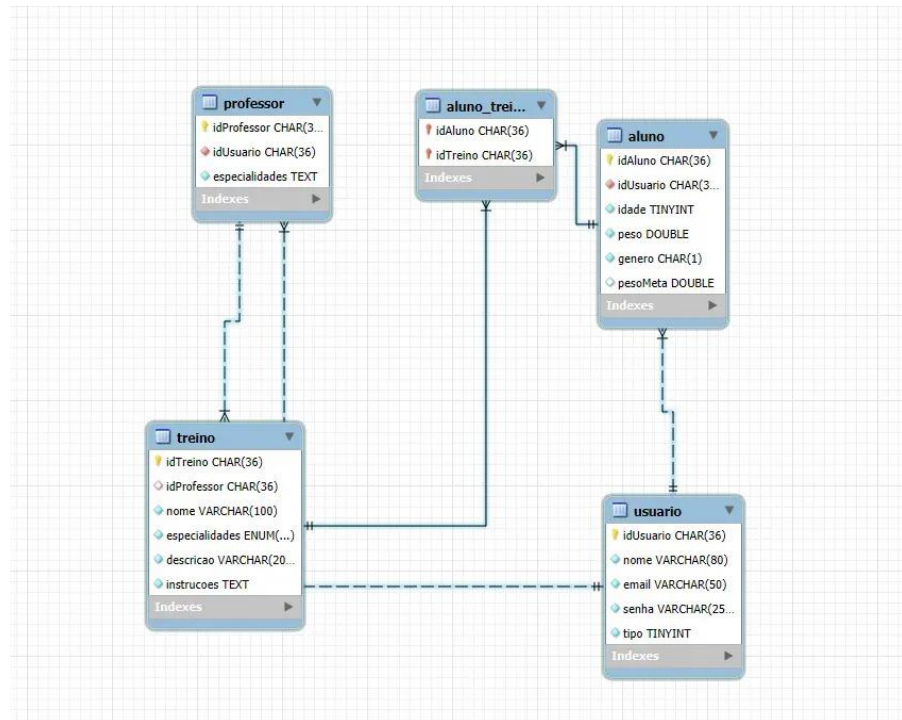
O avanço da tecnologia tem transformado a forma como as pessoas buscam alcançar objetivos relacionados à saúde e ao bem-estar. Nesse cenário, o aplicativo MetaFit apresenta-se como uma ferramenta inovadora para promover o acompanhamento de metas e treinos personalizados, atendendo às necessidades específicas de alunos e professores em um ambiente digital. Com funcionalidades voltadas para o controle e monitoramento de progresso, o MetaFit permite que alunos definam metas, registrem dados como peso inicial e peso desejado, e acessem treinos desenvolvidos por profissionais especializados. Paralelamente, professores encontram na plataforma recursos para gerenciar e cadastrar treinos.

1.1 TECNOLOGIAS UTILIZADAS

Java: O MetaFit foi desenvolvido utilizando a linguagem de programação Java para criar tanto a interface do usuário quanto o back-end da aplicação.

MySQL: No desenvolvimento do MetaFit, utilizamos o MySQL como sistema de gerenciamento de banco de dados e criação de tabelas, para armazenar e gerenciar as informações dos usuários, treinos e metas.

Figura 1 – Tabela do banco de dados



1.2 ARQUITETURA MVC(MODEL-VIEW-CONTROLLER)

No desenvolvimento do MetaFit, foi adotada a arquitetura MVC (Model-View-Controller), que permite uma estrutura organizada e escalável para o aplicativo.

A arquitetura MVC é um padrão de design de software amplamente utilizado no desenvolvimento de sistemas, especialmente em aplicações web. Sua principal característica é a separação de responsabilidades em três componentes principais, cada um com um propósito bem definido:

1. Model (Modelo):

Representa a lógica de negócios e os dados da aplicação. É responsável por gerenciar as regras de negócio, acessar o banco de dados e notificar a *view* sobre alterações.

2. View (Visão):

É a camada responsável pela interface do usuário e pela apresentação dos dados. Ela exibe informações fornecidas pelo modelo e captura as interações dos usuários, como cliques e entradas de dados.

3. Controller (Controlador):

Atua como intermediário entre o modelo e a visão. Ele processa as entradas do usuário, invoca os métodos apropriados no modelo e define qual *view* será apresentada ao usuário.

Vantagens do MVC:

- Separação de responsabilidades: facilita a manutenção e a escalabilidade do sistema.
- Reutilização de componentes: o mesmo modelo pode ser utilizado com diferentes *views*.
- Facilidade na colaboração: equipes de design e desenvolvimento podem trabalhar de forma independente.

Exemplo Prático: responsabilidade das demais camadas da aplicação. É especialmente aplicado em sistemas que utilizam persistência de dados, como aqueles que interagem com bancos de dados.

A principal finalidade do DAO é fornecer uma interface clara e específica para realizar operações relacionadas aos dados, como inserção, consulta, atualização e remoção. Com isso, a lógica de negócios fica desacoplada dos detalhes técnicos de acesso ao banco de dados, promovendo maior organização e manutenção do código.

2. FLUXO

Fluxo de navegação

O usuário ao fazer a inicialização do aplicativo, será diretamente direcionado a tela de login, onde colocará suas informações de cadastro, como e-mail e senha, sendo direcionado a tela principal, de acordo com seu tipo de usuário.

Caso o usuário não tenha cadastro, ele será direcionado para uma tela de cadastro, onde haverá duas opções: Aluno; Professor.

Escolha de cadastro Aluno:

Caso o usuário escolha se cadastrar como aluno, ele deverá digitar as seguintes informações: Nome, e-mail, idade, peso, gênero, e senha.

Escolha de cadastro Professor:

Caso o usuário escolha se cadastrar como professor, ele deverá digitar as seguintes informações: Nome, e-mail, especialidade e senha.

Após feito o cadastro, o usuário(aluno) será direcionado para a tela de Home Aluno, que é a página inicial, onde serão exibidas as funções de Metas, Treinos e configurações. Caso o aluno escolha a opção de “Meta”, ele será direcionado a Tela de Metas, onde especificará o seu objetivo de peso.

Feito isso, será oferecida uma lista de treinos com o determinado objetivo para que o Aluno escolha um que atenda melhor a sua necessidade, com isso, ele voltará a tela HomeAluno.

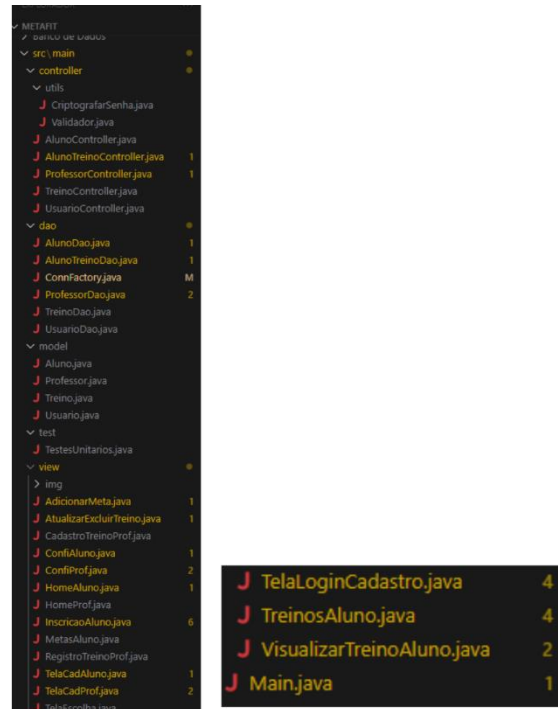
Caso o usuário escolha se cadastrar como professor, ele deverá digitar as seguintes informações: Nome; e-mail; especialidade e senha.

Após feito o cadastro ele será direcionado a tela HomeProfessor, onde mostrará as funções de criar treino e configurações. Após o usuário escolher a função de criar treino ele será direcionado a tela de cadastro de treinos, onde ele irá criar treinos específicos para cada tipo de objetivo.

Feito isso, ele retornará para a tela principal do professor e lá irá encontrar todos os treinos cadastrados por ele.

3. ESTRUTURA DE PASTAS

Figura 2 - Modelagem



4. REFERÊNCIAS

Para o desenvolvimento do aplicativo MetaFit, utilizamos como referência o aplicativo oficial da Smart Fit, que apresenta funcionalidades voltadas para gestão de treinos, acompanhamento de metas e integração com os usuários. Essa análise permitiu identificar boas práticas de design, navegação e funcionalidades, servindo como base para adaptar e aprimorar a experiência oferecida pelo MetaFit, focando em personalização e interatividade entre alunos e profissionais capacitados.

Referência bibliográfica

SMART FIT. *Aplicativo Smart Fit*. Disponível em: <https://www.smartfit.com.br/>. Acesso em: 04 dez. 2024.

5. CONCLUSÃO

O desenvolvimento do MetaFit representou uma solução inovadora para o monitoramento de metas de saúde e desempenho físico, atendendo tanto alunos quanto professores por meio de uma plataforma digital personalizada. Utilizando as

tecnologias de Java para a criação da interface e do backend, além de MySQL para o gerenciamento de dados, o aplicativo oferece uma experiência eficiente e dinâmica para os usuários. A arquitetura MVC foi fundamental para organizar o código e garantir a separação clara entre a lógica de negócios, a interface e o controle de fluxo, o que facilitou a manutenção e expansão do sistema.

Através de suas funcionalidades, como o acompanhamento de metas de peso, registro de treinos e a interação entre alunos e professores, o MetaFit cumpre seu objetivo de promover uma jornada mais personalizada e eficaz para aqueles que buscam melhorar sua saúde e bem-estar. A plataforma oferece uma interface amigável, integra dados em tempo real e permite uma gestão eficiente dos treinos e progresso dos alunos, tornando-se uma ferramenta valiosa tanto para iniciantes quanto para aqueles com experiência em atividades físicas.

No entanto, como qualquer aplicação, o MetaFit pode ser aprimorado. A adição de novas funcionalidades, como a integração com dispositivos de monitoramento de saúde e a personalização ainda maior dos treinos, poderia ampliar sua utilidade e engajamento. O desenvolvimento de futuras versões também pode explorar novas tecnologias e métodos para oferecer um serviço ainda mais completo e eficaz, atendendo às necessidades em constante evolução dos usuários.