

**GRUPO 04**

**INTEGRANTES**

Beatriz Santina

Caroline Ribeiro

Izabelly Gutierres

Luan Oliveira

Lucas Manhães

**PROJETO INTEGRADOR**

**ESCOPO DO PROJETO**

**Tema: Aplicativo Fitness Personalizado**



**GRUPO 04**

**INTEGRANTES**

Beatriz Santina

Caroline Ribeiro

Izabelly Gutierres

Luan Oliveira

Lucas Manhães

**PROJETO INTEGRADOR**

**ESCOPO DO PROJETO**

**Tema:Aplicativo Fitness Personalizado**

Relatório solicitado pela Generation Brasil para compor o projeto final.O relatório refere-se ao escopo do projeto integrador.

1. **Visão Geral do Projeto**

Desenvolver um aplicativo fitness com plano gratuito e pago, com personalização de treinos e acompanhamento de IMC.

1. **Modelo de Negócio**

**Descrição:**

* **Tipo de Plataforma:**Aplicativo para acompanhamento fitness.
* **Público-alvo:** O aplicativo fitness personalizado inclui indivíduos que desejam melhorar sua saúde física, manter uma rotina de exercícios e acompanhar seu progresso de forma prática e personalizada.
* **Proposta de Valor**: Oferecer uma solução digital acessível, eficiente e motivadora para indivíduos de todos os níveis de condicionamento físico que desejam melhorar sua saúde e alcançar seus objetivos pessoais.

1. **Entidades e Atributos**

**Usuário:**

* ID do usuário - Long Int
* Nome - String
* Usuario - String
* Senha - String
* Foto - String
* Peso - Double
* Altura - Double
* Treinos - Treino id (OneToMany)
* Tipo - Tipo id (ManyToOne)

**Plano:**

* ID - Long id
* Nome - String

**Treino:**

* ID - Long id
* Exercício - String
* Num\_serie - Integer
* Carga - String
* Descanso - String
* Usuario - usuario

1. **Principais Funcionalidades (CRUD):**

**Usuário:**

**Listar Todos:** Lista todos os usuários.

* **Endpoint**: GET

**Listar Todos por ID:** Lista todos os usuários por ID.

* **Endpoint**: GET

**Criar/Adicionar**: Cadastro de novos usuários no aplicativo.

* **Endpoint**: POST
* **Dados de Entrada**: {nome, usuario, senha, foto, peso, altura}

**Ler**: Consulta de informações de usuários.

* **Endpoint**: GET
* **Parâmetros de Consulta**: {nome, usuario, senha, foto, peso, altura}

**Atualização**: Atualizar dados do usuário.

* **Endpoint**: PUT
* **Dados de Entrada**: {nome, usuario, senha, foto, peso, altura}

**Listar:** Listar todos os usuários.

* **Endpoint**:GET

**Deletar**: Excluir usuário.

* **Endpoint**: DELETE

**Plano:**

**Listar todos:**: Listar todos os planos no aplicativo.

* **Endpoint**: GET
* **Dados de Entrada**: {id, nome}

**Listar por ID:**: Lista os planos por ID.

* **Endpoint**: GET
* **Parâmetros de Consulta**: {nome, usuario, senha, foto, peso, altura}

**Listar por nome:**: Listar os planos por nome.

* **Endpoint**: GET
* **Dados de Entrada**: {nome, usuario, senha, foto, peso, altura}

**Criar um novo Tipo:** Cria um novo tipo de plano.

* **Endpoint**:POST

**Atualizar:** Atualizar um tipo.

* **Endpoint**: PUT

**Deletar:** Deletar um tipo por ID.

* **Endpoint**: DELETE

**Treino:**

**Listar todos:**: Listar todos os treinos no aplicativo.

* **Endpoint**: GET
* **Dados de Entrada**: {id, exercicio, num\_serie, carga, descanso, usuario}

**Listar por ID:**: Lista os planos por ID.

* **Endpoint**: GET
* **Parâmetros de Consulta**: {id, exercicio, num\_serie, carga, descanso, usuar

**Listar por exercício:**: Listar os treinos por nome.

* **Endpoint**: GET
* **Dados de Entrada**: {id, exercicio, num\_serie, carga, descanso, usuario}

**Criar um novo Tipo:** Cria um novo treino no aplicativo.

* **Endpoint**:POST

**Atualizar:** Atualiza um treino no aplicativo.

* **Endpoint**: PUT

**Deletar:** Deletar um treino por ID.

* **Endpoint**: DELETE

1. **Tecnologias Utilizadas**

**Backend**

* **Java:** Linguagem de programação robusta e amplamente utilizada para desenvolvimento backend.
* **Spring Framework:** Framework para desenvolvimento de aplicações Java, fornecendo suporte para injeção de dependência, segurança, e muito mais.
* **Spring Boot:** Extensão do Spring Framework que simplifica a configuração e o desenvolvimento de aplicações standalone e produção-ready.

**Banco de Dados**

* **MySQL:** Sistema de gerenciamento de banco de dados relacional, ideal para armazenar dados estruturados.

**Ferramentas de Desenvolvimento e Teste**

* **Spring Tool Suite (STS)**: IDE baseada no Eclipse, otimizada para desenvolvimento com Spring Framework.
* **Insomnia:** Ferramenta de teste de APIs que facilita a criação, envio e gerenciamento de requisições HTTP.
* **Hibernate:** Framework de mapeamento objeto-relacional (ORM) para Java, utilizado para facilitar a interação com o banco de dados**.**
* **Maven:** Gerenciador de dependências e ferramenta de automação de build para projetos Java.