

**GRUPO 06**

**INTEGRANTES**

Guilherme Kaludin

João Oliveira

Juliana Monteiro

Lucas Daniel

Tainá Lara

**PROJETO INTEGRADOR**

**ESCOPO DO PROJETO**

**Tema: Aplicativo fitness**



**GRUPO 06**

**INTEGRANTES**

Guilherme Kaludin

João Oliveira

Juliana Monteiro

Lucas Daniel

Tainá Lara

**PROJETO INTEGRADOR**

**ESCOPO DO PROJETO**

**Tema: Aplicativo fitness**

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

1. **Título do Projeto e Modelo de Negócio escolhido**

Aplicativo fitness.

Uma plataforma fitness para cálculo do IMC, onde usuários podem inserir seus dados e receber uma classificação baseada no índice de massa corporal.

1. **Descrição Geral do Projeto**

O projeto aplicativo fitness é uma aplicação desenvolvida em Spring Boot que gerencia as operações relacionadas ao cálculo do IMC, permitindo que usuários cadastrem seus exercícios e acompanhem sua evolução. O Sistema conta com um mecanismo de autenticação, onde cada usuário precisa se identificar por meio de um nome de usuário e senha para acessar a plataforma.

1. **Descrição da Entidade/Model foi criada e seus atributos**

Criamos as Model’s Usuário, Exercicio e Categoria.

(tb\_usuario) com os atributos id, nome, usuario, senha, foto, peso, altura, e idade.

(tb\_exercicio) com os atributos id, nome, descricao, aparelho, serie, repeticao, FK id usuario e FK id categoria.

(tb\_categoria) com os atributos id, tipo e grupo.

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

Criamos as seguintes funcionalidades:

Getters and Setters

findAll()

findById()

findAllByNome()

get(),

post()

put()

delete()

Método utilitário para calcular o IMC com base no peso e altura do usuário:

calcularIMC()

Método utilitário para classificar o IMC do usuário:

classificarIMC()

Métodos de segurança:

**userDetailsService()**

**passwordEncoder()**

**authenticationManager**

filterChain(HttpSecurity http)

extractAllClaims(String token)

extractUsername(String token)

extractExpiration(String token)

i**sTokenExpired(String token)**

**generateToken(String username)**

**validateToken(String token, UserDetails details)**

getAuthorities()

isAccountNonExpired()

isAccountNonLocked()

isCredentialsNonExpired()

1. **Tecnologias Utilizadas (banco de dados e backend)**

Java 17+

Maven Plugin

MySql Driver

JPA/Hibernate

Spring Boot Dev Tools

Spring Web

Spring Validation

Spring Security

JSON Web Token

Swagger

PostgreSQL

Docker

Render