

Projeto Final (M2)

Nome: Carlos Finney Silva de Souza	RGM: 11231103813
Nome: Gustavo Vinícius Ferraz de Oliveira	RGM: 11231102502
Nome: Gabriel Santanna Moraes dos Santos	RGM: 11231103641
Nome: Leonardo Giampaglia Gomes	RGM: 11231103169

Índice - Projeto FitLife

Sumário

Visão Geral do Projeto

1.1 Introdução

1.2 Objetivos

Arquitetura do Sistema

2.1 Tecnologias Utilizadas

2.2 Estrutura do Projeto

Componentes do Sistema

3.1 Banco de Dados

3.2 Conexão com Banco de Dados

3.3 Interface do Usuário

Funcionalidades Principais

4.1 Sistema de Autenticação

4.2 Cálculo de IMC

Segurança

5.1 Medidas Implementadas

5.2 Boas Práticas

Dependências

6.1 Maven Dependencies

Implantação

7.1 Requisitos

7.2 Passos para Implantação

Considerações Técnicas

8.1 Escalabilidade

8.2 Manutenibilidade

9.0 Conclusão

10.0 Fotos

Documentação Técnica - Projeto FitLife

1. Visão Geral do Projeto

O FitLife é uma aplicação web desenvolvida em Java que permite aos usuários calcular e acompanhar seu Índice de Massa Corporal (IMC). O sistema possui funcionalidades de cadastro, login e cálculo de IMC, utilizando tecnologias web modernas e um banco de dados relacional.

2. Arquitetura do Sistema

2.1 Tecnologias Utilizadas

Backend: Java EE (Servlets)

Frontend: HTML5, CSS3

Banco de Dados: MySQL

Servidor: Apache Tomcat

Gerenciador de Dependências: Maven

2.2 Estrutura do Projeto

trabalho/

|— src/

| └─ java/

| └─ com/

| └─ fitlife/

| └─ utils/

| └─ DatabaseUtils.java

| └─ servlets/

| └─ Login.java

```

|           |— RegistrarUsuario.java
|           |— CalcularIMC.java
|— web/
|   |— login.html
|   |— cadastro.html
|   |— CalcularIMC.html
|   |— WEB-INF/
|       |— web.xml
|— pom.xml

```

3. Componentes do Sistema

3.1 Banco de Dados

O banco de dados FitLifeDB possui três tabelas principais:

```

CREATE TABLE usuarios (
    id INT AUTO_INCREMENT PRIMARY KEY,
    nome VARCHAR(100) NOT NULL,
    email VARCHAR(100) NOT NULL UNIQUE,
    senha VARCHAR(255) NOT NULL,
    data_nascimento DATE,
    genero ENUM('M', 'F', 'Outro'),
    data_cadastro TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);

```

Armazena informações dos usuários

Utiliza chave primária autoincremental

Email único para cada usuário

Senha armazenada com hash por segurança

historico_imc

```

CREATE TABLE historico_imc (
    id INT AUTO_INCREMENT PRIMARY KEY,

```

```
usuario_id INT NOT NULL,  
altura DECIMAL(3,2) NOT NULL,  
peso DECIMAL(5,2) NOT NULL,  
imc DECIMAL(4,2) NOT NULL,  
classificacao VARCHAR(50),  
data_registro TIMESTAMP DEFAULT CURRENT_TIMESTAMP,  
FOREIGN KEY (usuario_id) REFERENCES usuarios(id)  
);
```

- Registra histórico de medições de IMC

Relacionamento com a tabela usuarios

Armazena dados precisos com casas decimais

Registro automático da data/hora

3.2 Conexão com Banco de Dados

O DatabaseUtils.java gerencia conexões com o banco:

```
public class DatabaseUtils {  
    private static final String DRIVER = "com.mysql.cj.jdbc.Driver";  
    private static final String URL = "jdbc:mysql://localhost:3306/FitLifeDB";  
    private static final String USER = "root";  
    private static final String PASSWORD = "";  
  
    // Método para obter conexão  
    public static Connection getConnection() throws SQLException {  
        return DriverManager.getConnection(URL, USER, PASSWORD);  
    }  
}
```

3.3 Interface do Usuário

Login (login.html)

Formulário de autenticação

Campos: email e senha

Redirecionamento para cadastro

Validação client-side

Cadastro (cadastro.html)

Formulário de registro

Campos: nome, email, senha, data nascimento, gênero

Validações de campos obrigatórios

Design responsivo

Calculadora IMC (CalcularIMC.html)

Interface para cálculo de IMC

Campos: altura e peso

Exibição do resultado e classificação

Histórico de medições

4. Funcionalidades Principais

4.1 Sistema de Autenticação

Login com email e senha

Sessão de usuário

Proteção contra SQL Injection

Hash de senha para segurança

4.2 Cálculo de IMC

Fórmula: $IMC = peso / (altura^2)$

Classificação automática

Armazenamento do histórico

Validação de dados

5. Segurança

5.1 Medidas Implementadas

Proteção contra SQL Injection

Senhas hasheadas

Validação de dados

Controle de sessão

5.2 Boas Práticas

Prepared Statements

Tratamento de exceções

Validação server-side

Sanitização de inputs

6. Dependências (pom.xml)

```
<dependencies>
  <!-- MySQL Connector -->
  <dependency>
    <groupId>mysql</groupId>
    <artifactId>mysql-connector-java</artifactId>
    <version>8.0.33</version>
  </dependency>
  <!-- Servlet API -->
  <dependency>
    <groupId>javax.servlet</groupId>
    <artifactId>javax.servlet-api</artifactId>
    <version>4.0.1</version>
  </dependency>
  <!-- Outras dependências... -->
</dependencies>
```

7. Implantação

7.1 Requisitos

JDK 8 ou superior

MySQL 5.7 ou superior

Apache Tomcat 9

Maven 3.6+

7.2 Passos para Implantação

Configurar banco de dados

Ajustar credenciais em DatabaseUtils.java

Compilar projeto com Maven

Deploy no Tomcat

8. Considerações Técnicas

8.1 Escalabilidade

Conexão pool configurável

Estrutura modular

Código reutilizável

8.2 Manutenibilidade

Código comentado

Padrões de projeto

Separação de responsabilidades

9. Conclusão

O projeto FitLife demonstra a implementação de um sistema web completo, utilizando tecnologias modernas e seguindo boas práticas de desenvolvimento. A arquitetura escolhida permite fácil manutenção e expansão futura.

Referências Bibliográficas

DEITEL, Paul; DEITEL, Harvey. Java: Como Programar. 10ª ed. São Paulo: Pearson Education do Brasil, 2016.

Referência fundamental para desenvolvimento Java e conceitos de programação orientada a objetos.

EVANS, Benjamin; WARBURTON, Richard. Java SE 8 Programming. Oracle Press, 2019.

3. SILBERSCHATZ, Abraham; KORTH, Henry F.; SUDARSHAN, S. Sistema de Banco de Dados. 6ª ed. Rio de Janeiro: Elsevier, 2012.

Fotos:

Visual

Imagem 1:

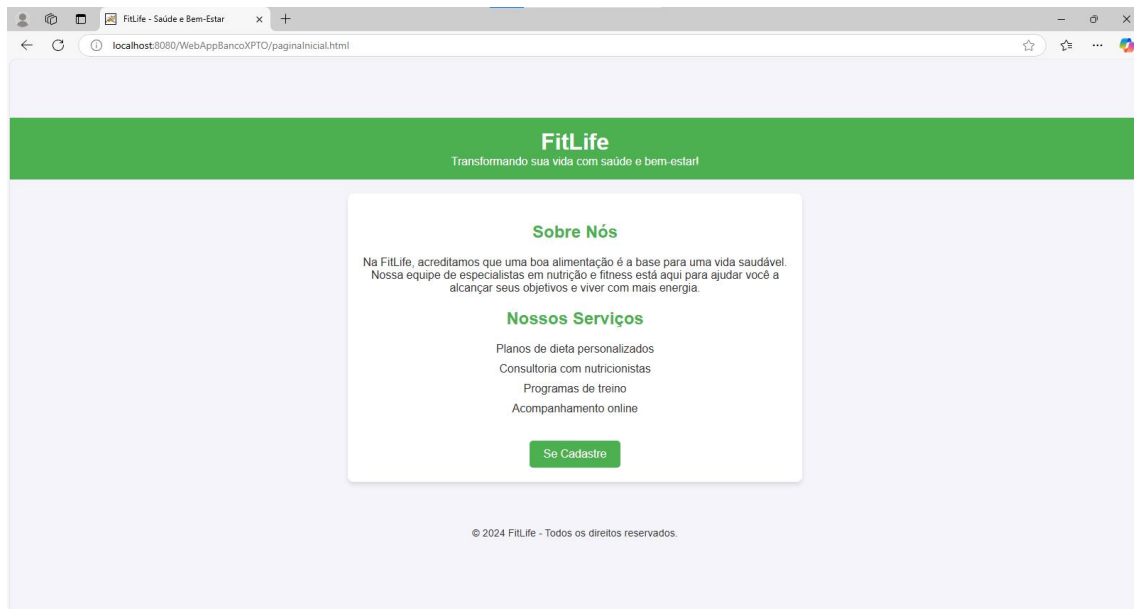


Imagem 2:

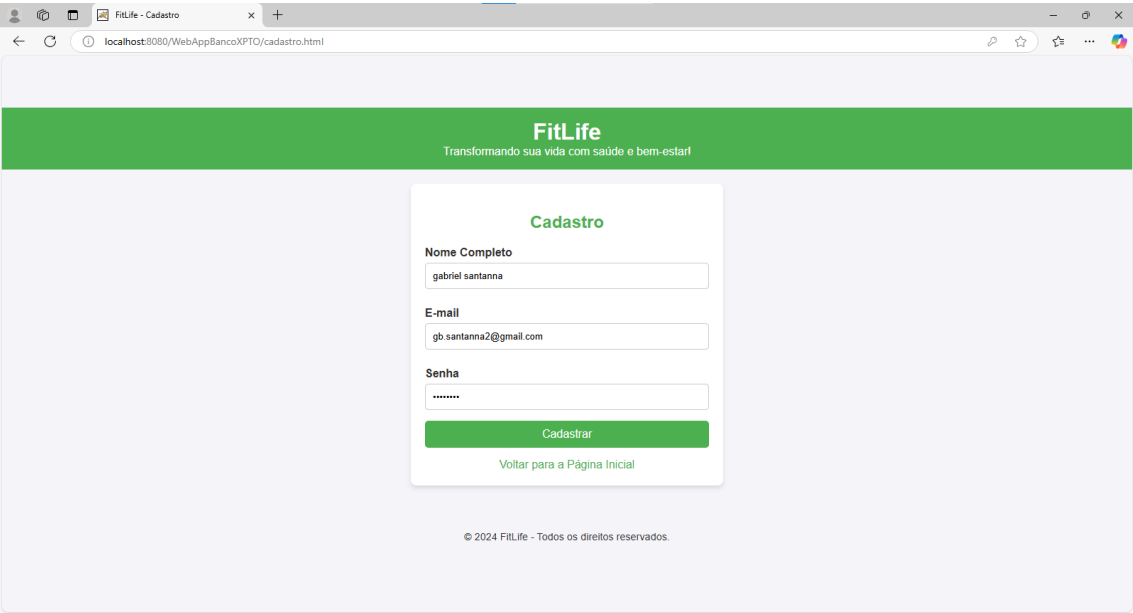


Imagem 3:



Fotos:

Html (imagem 01)

Imagem 1:

```
1 <!DOCTYPE html>
2 <html lang="pt-BR">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>FitLife - Saúde e Bem-Estar</title>
7   <style>
8     body {
9       font-family: Arial, sans-serif;
10      margin: 0;
11      padding: 0;
12      background-color: #f4f4f9;
13      color: #333;
14      display: flex;
15      flex-direction: column;
16      align-items: center;
17      justify-content: center;
18      min-height: 100vh;
19      text-align: center;
20    }
21
22    header {
23      background-color: #4caf50;
24      color: white;
25      padding: 1em 0;
26      width: 100%;
27    }
28
29    header h1, header p {
30      margin: 0;
31    }
32
33    .content {
34      background-color: white;
35      border-radius: 8px;
36      box-shadow: 0 4px 6px rgba(0, 0, 0, 0.1);
37      margin: 20px;
38      padding: 20px;
39      width: 90%;
40    }
41  </style>
42 </head>
43 <body>
44   <header>
45     <h1>FitLife</h1>
46     <p>Transformando sua vida com saúde e bem-estar!</p>
47   </header>
48   <div class="content">
49     <h2>Sobre Nós</h2>
50     <p>Na FitLife, acreditamos que uma boa alimentação é a base para uma vida saudável. Nossa equipe de especialistas em nutrição e fitness está aqui para ajudar você a alcançar seus objetivos e viver com mais energia.</p>
51     <h2>Nossos Serviços</h2>
52     <ul>
53       <li>Planos de dieta personalizados</li>
54       <li>Consultoria com nutricionistas</li>
55       <li>Programas de treino</li>
56       <li>Acompanhamento online</li>
57     </ul>
58     <a href="cadastro.html">Se Cadastre</a>
59   </div>
60   <footer>
61     <p>©copy; 2024 FitLife - Todos os direitos reservados.</p>
62   </footer>
63 </body>
64 </html>
```

Imagem 2:

```
67   a:hover {
68     background-color: #45a049;
69   }
70
71   footer {
72     margin-top: 20px;
73     padding: 10px 0;
74     background-color: #f4f4f9;
75     width: 100%;
76     font-size: 0.9em;
77   }
78 </style>
79 </head>
80 <body>
81   <header>
82     <h1>FitLife</h1>
83     <p>Transformando sua vida com saúde e bem-estar!</p>
84   </header>
85   <div class="content">
86     <h2>Sobre Nós</h2>
87     <p>Na FitLife, acreditamos que uma boa alimentação é a base para uma vida saudável. Nossa equipe de especialistas em nutrição e fitness está aqui para ajudar você a alcançar seus objetivos e viver com mais energia.</p>
88     <h2>Nossos Serviços</h2>
89     <ul>
90       <li>Planos de dieta personalizados</li>
91       <li>Consultoria com nutricionistas</li>
92       <li>Programas de treino</li>
93       <li>Acompanhamento online</li>
94     </ul>
95     <a href="cadastro.html">Se Cadastre</a>
96   </div>
97   <footer>
98     <p>©copy; 2024 FitLife - Todos os direitos reservados.</p>
99   </footer>
100 </body>
101 </html>
```

Fotos:
Html (Imagem 2)



Imagem 1

Imagem 2



Fotos:

Calcular o IMC:

```
paginalnicial.html x cadastro.html x CalculaIMC.html x RegistrarUsuario.java x CalculaIMC.java x
Source History
import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

@WebServlet("/CalculaIMC")
public class CalculaIMC extends HttpServlet {
    @Override
    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");

        // Obtém os valores enviados pelo formulário
        String alturaStr = request.getParameter("altura");
        String pesoStr = request.getParameter("peso");

        try (PrintWriter out = response.getWriter()) {
            out.println("<html>");
            out.println("<head><title>Resultado do IMC</title></head>");
            out.println("<body style='font-family: Arial, sans-serif; text-align: center;'>");
            out.println("<h1>Resultado do IMC</h1>");

            try {
                // Converte os valores para números
                double altura = Double.parseDouble(alturaStr);
                double peso = Double.parseDouble(pesoStr);

                if (altura <= 0 || peso <= 0) {
                    throw new NumberFormatException("Valores inválidos");
                }

                // Calcula o IMC
                double imc = peso / (altura * altura);
                String classificacao;

                if (imc < 18.5) classificacao = "Abaixo do peso";
                else if (imc < 24.9) classificacao = "Peso normal";
            }
        }
    }
}
```

Fotos:

Conexão com o banco:

```
package com.fitlife.utils;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;

public class DatabaseUtils {
    private static final String URL = "jdbc:mysql://localhost:3306/FitLifeDB";
    private static final String USER = "root"; // Substitua pelo usuário do banco
    private static final String PASSWORD = "senha"; // Substitua pela senha do banco

    public static Connection getConnection() throws SQLException {
        try {
            Class.forName("com.mysql.cj.jdbc.Driver"); // Carregar o driver JDBC
        } catch (ClassNotFoundException e) {
            throw new SQLException("Driver JDBC não encontrado.", e);
        }
        return DriverManager.getConnection(URL, USER, PASSWORD);
    }
}
```

Fotos:

Registrar o usuário cadastrado:

```

import java.io.IOException;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.SQLException;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

@WebServlet("/RegistrarUsuario")
public class RegistrarUsuario extends HttpServlet {
    @Override
    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        // Receber os parâmetros do formulário
        String nome = request.getParameter("nome");
        String email = request.getParameter("email");
        String senha = request.getParameter("senha");

        // Validar senha

        // Inserir os dados no banco
        try (Connection conn = DatabaseUtils.getConnection()) {
            String sql = "INSERT INTO usuarios (nome, email, senha) VALUES (?, ?, ?)";
            try (PreparedStatement stmt = conn.prepareStatement(sql)) {
                stmt.setString(1, nome);
                stmt.setString(2, email);
                stmt.setString(3, senha);

                stmt.executeUpdate();

                // Redirecionar para uma página de sucesso
                response.sendRedirect("cadastro-sucesso.html");
            }
        } catch (SQLException e) {
            throw new ServletException("Erro ao salvar no banco de dados.", e);
        }
    }
}

```

Fotos:

Banco de dados (sql):

```

1 CREATE DATABASE FitLifeDB;
2
3 USE FitLifeDB;
4
5 CREATE TABLE usuarios (
6     id INT AUTO_INCREMENT PRIMARY KEY,
7     nome VARCHAR(100) NOT NULL,
8     email VARCHAR(100) NOT NULL UNIQUE,
9     senha VARCHAR(100) NOT NULL,
10    data_cadastro TIMESTAMP DEFAULT CURRENT_TIMESTAMP
11 );
12

```

SELECT * FROM usuarios

Perfil [Editar em linha] [Editar] [Demonstrar SQL] [Criar código PHP] [Atualizar]

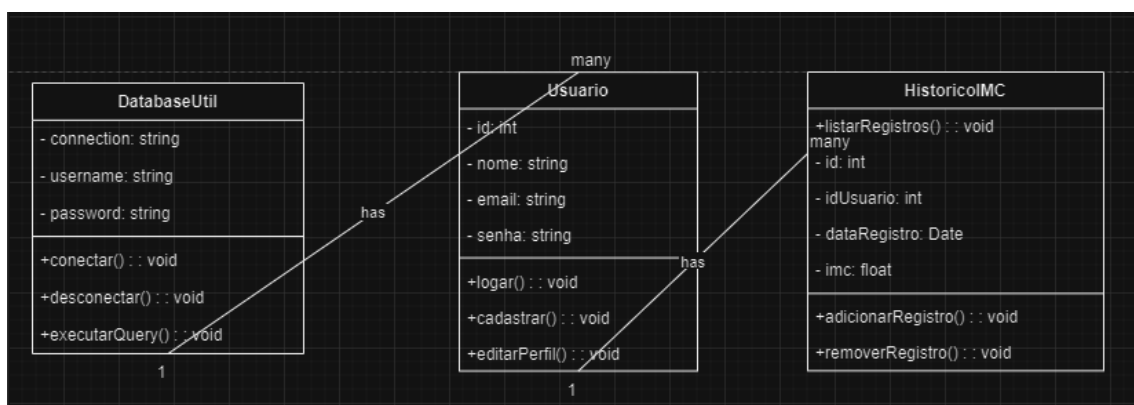
☐ Mostrar tudo | Número de linhas: 25 | Filtrar linhas: Procurar nesta tabela

Opções extras

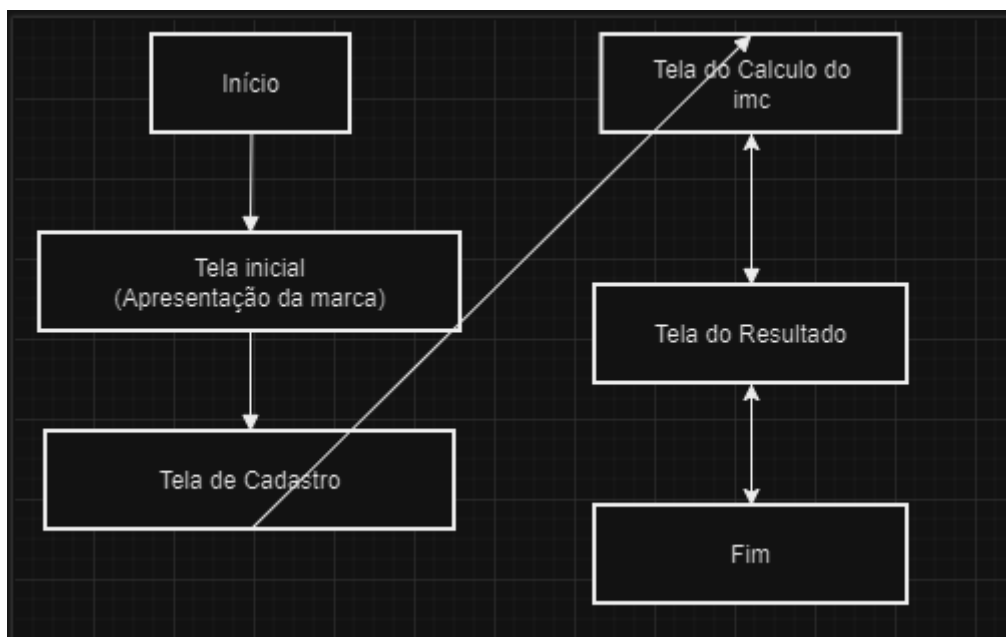
	id	nome	email	senha	data_cadastro
1	gabriel santanna	gb.santanna2@gmail.com	hsegb366	2024-11-29 23:53:32	

☐ Marcar todos | Com marcados: ☐ Editar ☐ Copiar ☐ Remover ☐ Exportar

Diagrama de classe



Fluxo de Telas:



LINK DO PROJETO NO GITHUB:

<https://github.com/santanna3/TrabalhoJava>