

SOFTWARE

DANT

Alunos: Arthur Santos, Diego Maia,
Henrique Nahim e Luis Felipe

PLANEJAMENTO

01

LOGIN

Página de login e
funções do back-end

02

CADASTRO DE USUÁRIO

Página de cadastro de
funções do back-end

03

ATUALIZAR USUARIO

Página de Update de
usuário por meio de
funções back-end

04

HOME PAGE

Estrutura da Home Page

05

SEARCH PAGE

Página de pesquisa

06

GAME PAGE

Página do Jogo e Fórum

REALIZE O LOGIN

Email

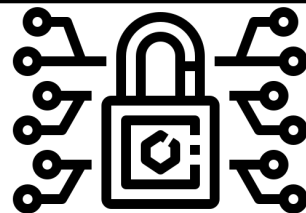
Senha

ENTRAR

NOVO USUARIO? [CADASTRAR](#)

ID
D.A.N.T

SEGURANÇA



```
public class Converter {  
    public static String CriptografarMd5(String senha) {  
        try {  
            MessageDigest md = MessageDigest.getInstance("MD5");  
            byte[] senhaBytes = senha.getBytes();  
            md.update(senhaBytes);  
            byte[] digest = md.digest();  
  
            StringBuilder hashString = new StringBuilder();  
            for (byte b : digest) {  
                hashString.append(String.format("%02x", b));  
            }  
  
            return hashString.toString(); // Retorna a senha em MD5 como uma string  
        } catch (NoSuchAlgorithmException e) {  
            e.printStackTrace();  
            return null; // Em caso de erro, retorna null ou lida com exceção de outra forma  
        }  
    }  
}
```

```
UserService.java 1 x
plmg-cc-2023-2-ti2-g08-dant > Codigo > MavenProject > inicio > src > main > java > Service > UserService.java > {} Service
26
29 public int auth(Request req, Response res) throws Exception {
30     String email = req.queryParams(queryParam:"email");
31     String senha = req.queryParams(queryParam:"senha");
32
33     senha = Converter.CriptografarMd5(senha);
34     int id = user.authentication(email, senha);
35
36     if (id == -1) {
37         return -1;
38         // retorna 0 se o usuario não for encontrado no BD
39     } else {
40         return id;
41         // retorna o ID do usuario caso o mesmo for encontrado
42     }
43 }
44 }
45
```

```
path("/Index", () -> {
    get("/login", (req, res) -> user.auth(req,res));
}
```



```
public int authentication(String email, String senha) throws SQLException {

    String sql = "SELECT * FROM usuario WHERE email = ? and senha = ?";
    PreparedStatement preparedStatement = conexao.prepareStatement(sql);
    preparedStatement.setString(1, email);
    preparedStatement.setString(2, senha);

    ResultSet resultSet = preparedStatement.executeQuery();

    // Verifica se o ResultSet contém alguma linha
    if (resultSet.next()) {
        return resultSet.getInt("id"); // Usuário encontrado
    } else {
        return 0; // Usuário não encontrado
    }
}
```

ID
D.A.N.T

Nome Completo

dd/mm/aaaa



Email

Senha

Confirmar Senha

[TERMOS DE USO](#) ☐

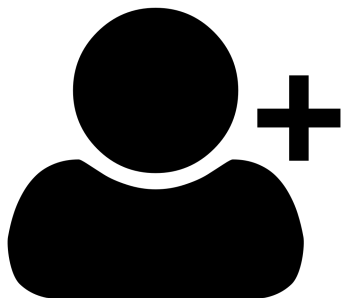
CADASTRAR

[VOLTAR](#)

UserService.java 1 X

plmg-cc-2023-2-ti2-g08-dant > Codigo > MavenProject > inicio > src > main > java > Service > UserService.java > {} Service

```
12
13 public Boolean registro(Request req, Response res) {
14     String nome = req.queryParams(queryParam:"nome");
15     String email = req.queryParams(queryParam:"email");
16     String senha = req.queryParams(queryParam:"senha");
17     String dataNasc = req.queryParams(queryParam:"nasc");
18
19     senha = conv.CriptografarMd5(senha);
20
21     try {
22         user.inserirUsuario(nome, senha, email, dataNasc);
23         return true;
24     } catch (Exception e) {
25         return false;
26     }
27 }
```



```
path("/Index", () -> {
    get("/login", (req, res) -> user.auth(req,res));
    post("/registro", (req, res) -> {
        boolean response = user.registro(req, res);
        return response;
    });
});
```

```
public boolean inserirUsuario(String nome, String senha, String email, String dataNascStr) {
    if(autenticandoEmail(email)) { return false;} //validando email

    try {

        // Converte a data de String para java.util.Date usando SimpleDateFormat
        SimpleDateFormat dateFormat = new SimpleDateFormat("yyyy-MM-dd");
        java.util.Date parsedDate = dateFormat.parse(dataNascStr);
        java.sql.Date dataNasc = new java.sql.Date(parsedDate.getTime());

        // SQL para inserir um novo usuário
        String sql = "INSERT INTO usuario (nome, senha, email, datanasc) VALUES (?, ?, ?, ?)";

        // Cria um PreparedStatement com a consulta SQL
        PreparedStatement preparedStatement = conexao.prepareStatement(sql);

        // Define os parâmetros da consulta com os valores passados como argumentos
        preparedStatement.setString(1, nome);
        preparedStatement.setString(2, senha);
        preparedStatement.setString(3, email);
        preparedStatement.setDate(4, dataNasc);

        // Executa a consulta de inserção
        preparedStatement.executeUpdate();
        return true;
    } catch (SQLException | ParseException e) {
        e.printStackTrace();
        return false;
    }
}
```

Nome:

Email:

Senha

Confirmar senha

Data

DELETAR

SALVAR

Insira uma Imagem (PNG, JPG)



1.

File name vegito.jpg, file size 260.7 KB.

R.EAD

```
public User GetUserByID(int id){
    User usuario = null;

    String sql = "SELECT * FROM usuario WHERE id = ?";
    try{
        PreparedStatement preparedStatement = conexao.prepareStatement(sql);
        preparedStatement.setInt(1, id);
        ResultSet resultSet = preparedStatement.executeQuery();
        if(resultSet.next()){
            int idUsuario = resultSet.getInt("id");
            String nome = resultSet.getString("nome");
            String senha = resultSet.getString("senha");
            String email = resultSet.getString("email");
            byte[] foto = resultSet.getBytes("foto");
            java.sql.Date dataNasc = resultSet.getDate("datanasc");
            boolean assinatura = resultSet.getBoolean("assinatura");
            usuario = new User(idUsuario,nome,senha,email,foto,dataNasc,assinatura);

            return usuario;
        }else{
            return null;
        }
    }catch(SQLException e){
        e.printStackTrace();
        return null;
    }
}
```

C.REATE

```
UserService.java 2 X
plmg-cc-2023-2-ti2-g08-dant > Codigo > MavenProject > inicio > src > main > java > Service > UserService.java > UserService

46
47 public Boolean update(Request req, Response res) throws Exception {
48     String nome = req.queryParams(queryParam:"nome");
49     String email = req.queryParams(queryParam:"email");
50     String senha = req.queryParams(queryParam:"senha");
51     String dataNasc = req.queryParams(queryParam:"nasc");
52
53     SimpleDateFormat dateFormat = new SimpleDateFormat("yyyy-MM-dd");
54     java.util.Date parsedDate = dateFormat.parse(dataNasc);
55     java.sql.Date dataNascDate = new java.sql.Date(parsedDate.getTime());
56
57     senha = conv.CriptografarMd5(senha);
58
59     int id = auth(req, res);
60
61     try {
62         user.updateUserName(id, nome);
63         user.updateUserEmail(id, email);
64         user.updateUserPassword(id, senha);
65         user.updateUserDateOfBirth(id, dataNascDate);
66         return true;
67     } catch (Exception e) {
68         return false;
69     }
70 }
```

U.PDATE

UserService.java 2 X

plmg-cc-2023-2-t2-g08-dant > Codigo > MavenProject > inicio > src > main > java > Service > UserService.java > UserService

```
46
47 public Boolean update(Request req, Response res) throws Exception {
48     String nome = req.queryParams("nome");
49     String email = req.queryParams("email");
50     String senha = req.queryParams("senha");
51     String dataNasc = req.queryParams("nasc");
52
53     SimpleDateFormat dateFormat = new SimpleDateFormat("yyyy-MM-dd");
54     java.util.Date parsedDate = dateFormat.parse(dataNasc);
55     java.sql.Date dataNascDate = new java.sql.Date(parsedDate.getTime());
56
57     senha = conv.CriptografarMd5(senha);
58
59     int id = auth(req, res);
60
61     try {
62         user.updateUserName(id, nome);
63         user.updateUserEmail(id, email);
64         user.updateUserPassword(id, senha);
65         user.updateUserDateOfBirth(id, dataNascDate);
66         return true;
67     } catch (Exception e) {
68         return false;
69     }
70 }
```

```
public boolean UpdateUser(User user) {
    String sql = "UPDATE usuario SET nome = ?, senha = ?, email = ?, foto = ?, datanasc = ? WHERE id = ?";
    try {
        PreparedStatement preparedStatement = conexao.prepareStatement(sql);
        preparedStatement.setString(1, user.getNome());
        preparedStatement.setString(2, user.getSenha());
        preparedStatement.setString(3, user.getEmail());
        preparedStatement.setBytes(4, user.getFoto());
        preparedStatement.setDate(5, user.getDataNasc());
        preparedStatement.setInt(6, user.getId());
        preparedStatement.executeUpdate();
        return true;
    } catch (SQLException e) {
        e.printStackTrace();
        return false;
    }
}
```

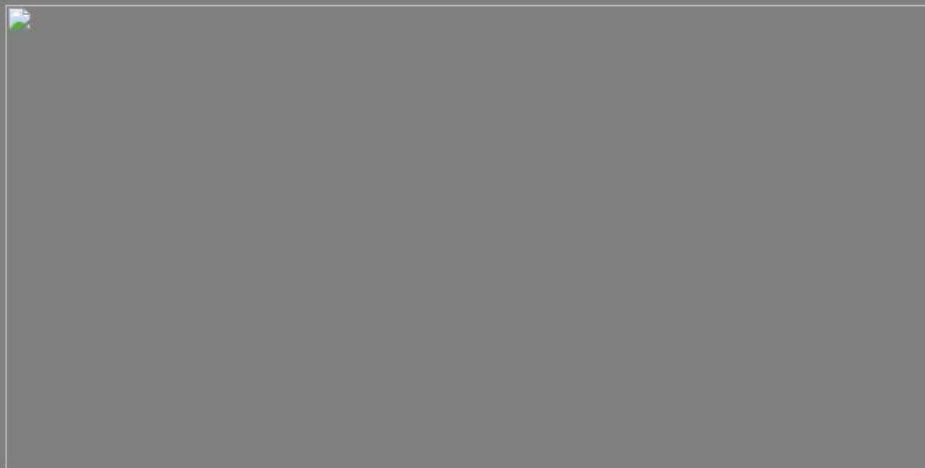
D.ELETE

```
public boolean deleteUserByID(int id){  
    String sql = "DELETE FROM usuario WHERE id = ?";  
    try{  
        PreparedStatement preparedStatement = conexao.prepareStatement(sql);  
        preparedStatement.setInt(1, id);  
        preparedStatement.executeUpdate();  
        return true;  
    }catch(SQLException e){  
        e.printStackTrace();  
        return false;  
    }  
}
```



HOME PAGE

SEARCH PAGE



```
async function sendMessage() {
  const userMessage = document.querySelector(".recebe").value;
  document.querySelector(".recebe").value = '';
  document.querySelector(".insere").innerHTML += `<div style="margin-top: 10px">Usuário: ${userMessage}</div>`;
  const Mensagem = `Você é um atendente de uma loja virtual de jogos no qual detem conhecimentos a cerca de tendencia de mercado sugestoes e banco de dados de jogos`;
  console.log(Mensagem);
  const response = await fetch('https://api.openai.com/v1/engines/text-davinci-003/completions', {
    method: 'POST',
    headers: {
      'Content-Type': 'application/json',
      'Authorization': `Bearer ${apiKey}`
    },
    body: JSON.stringify({
      prompt: `0 usuário diz: ${Mensagem}\nChatGPT:`,
      max_tokens: 150
    })
  });

  const data = await response.json();
  const chatGPTResponse = data.choices[0].text.trim();
  document.querySelector(".insere").innerHTML += `<div style="margin-top: 10px; text-align: justify; font-size: 10px;"> ChatGPT: ${chatGPTResponse}</div>`;
}
```

Categoria: 0

Categoria: 1

Categoria: 2

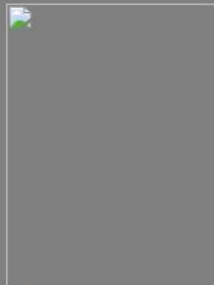
Categoria: 0



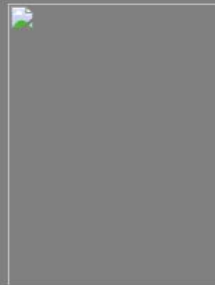
titulo 0



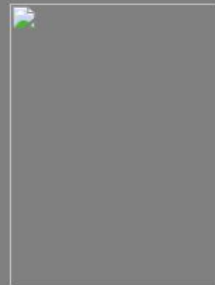
titulo 1



titulo 2



titulo 3



titulo 4



titulo 5



titulo 6



```
9  import com.google.gson.Gson;
10
11  You, 18 seconds ago | 1 author (You)
12  public class GamesService {
13      public String getGames() throws Exception {
14          Gson gson = new Gson();
15          GameDAO game = new GameDAO();
16
17          LinkedList<Game> gamearr = new LinkedList<>();
18
19          gamearr = game.GetAllGames();
20
21          return gson.toJson(gamearr);
22      }
23  }
```

```
path("/HomePage", () -> {
    get("/", (req, res) -> game.getGames());
});
```

```
public LinkedList<Game> GetAllGames() throws Exception {
    LinkedList<Game> games = new LinkedList<Game>();

    String sql = "SELECT * FROM games where appid > 0 ORDER BY appid;";

    PreparedStatement preparedStatement = conexao.prepareStatement(sql);
    ResultSet resultSet = preparedStatement.executeQuery();

    while (resultSet.next()) {
        games.add(new Game(resultSet.getInt("appid"), resultSet.getString("nome"), resultSet.getString("json")));
    }

    return games;
}
```



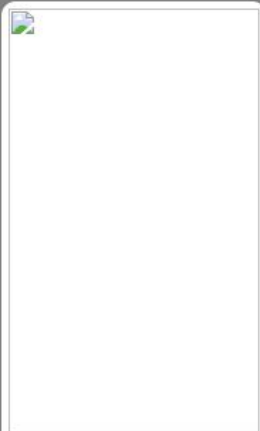

Algo

Algo

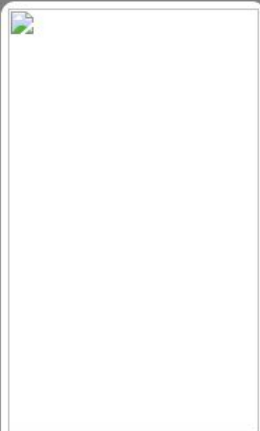
Algo

Algo

Filtrar



Nome: 0
Algo mais



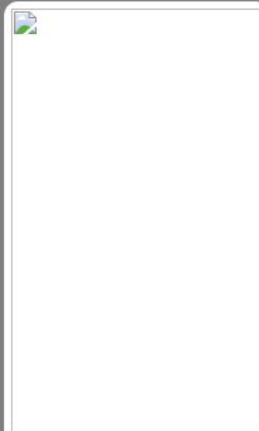
Nome: 1
Algo mais



Nome: 2
Algo mais



Nome: 3
Algo mais



Nome: 4
Algo mais



```
public LinkedList<Game> getGamesBy(String genero, boolean free,String nome){
    LinkedList<Game> games = new LinkedList<Game>();
    String sql = "SELECT * FROM GAMES WHERE 1=1 ";

    if(genero != null || genero != ""){
        sql += "AND json like '%" +genero+"%' ";
    }
    if(free){
        sql += "AND NOT  json like '%price%'";
    }
    if(nome != null || nome != ""){
        sql += "AND nome like '%" +nome+"%' ";
    }

    sql += "ORDER BY appid;";
    try (PreparedStatement preparedStatement = conexao.prepareStatement(sql)) {
        ResultSet resultSet = preparedStatement.executeQuery();
        while (resultSet.next()) {
            games.add(new Game(resultSet.getInt("appid"), resultSet.getString("nome"), resultSet.getString("json")));
        }
    } catch (SQLException e) {
        e.printStackTrace();
        throw new RuntimeException("Erro ao buscar o jogo por appid", e);
    }
    return games;
}
```



FORUM

Lorem ipsum dolor sit amet.



Lorem ipsum dolor, sit amet consectetur adipisicing elit. Consequatur, excepturi harum maxime esse illo sequi fugit consequuntur tempora! Vitae, aliquid.

Algo mais

PReco XXXXX em X e X

```
public Game getGameByID(int appid) {
    Game game = null;
    if (appid >= 0) {

        String sql = "SELECT * FROM games WHERE appid = ?";

        try (PreparedStatement preparedStatement = conexao.prepareStatement(sql)) {

            preparedStatement.setInt(1, appid);

            try (ResultSet resultSet = preparedStatement.executeQuery()) {
                if (resultSet.next()) {
                    game = new Game();
                    game.setAppid(resultSet.getInt("appid"));
                    game.setNome(resultSet.getString("nome"));

                    game.setJson(resultSet.getString("json"));
                }
            }
        } catch (SQLException e) {
            e.printStackTrace();
            throw new RuntimeException("Erro ao buscar o jogo por appid", e);
        }

    }

    return game;
}
```

[HOME PAGE](#)[SEARCH PAGE](#)[FORUM](#)

Genero

Genero

Genero



Lorem ipsum dolor sit amet consectetur adipisicing elit. Vero, corrupti. Quae voluptates veniam tenetur quo aliquid alias! Quam est exercitationem debitis aliquam, repudiandae perferendis quidem, voluptas dolore vel dolores natus?

responder +



Lorem ipsum dolor sit amet consectetur adipisicing elit. Vero, corrupti. Quae voluptates veniam tenetur quo aliquid alias! Quam est exercitationem debitis aliquam, repudiandae perferendis quidem, voluptas dolore vel dolores natus?

responder +



Lorem ipsum dolor sit amet consectetur adipisicing elit. Vero, corrupti. Quae voluptates veniam tenetur quo aliquid alias! Quam est exercitationem debitis aliquam, repudiandae perferendis quidem, voluptas dolore vel dolores natus?

responder +



Lorem ipsum dolor sit amet consectetur adipisicing elit. Vero, corrupti. Quae voluptates veniam tenetur quo aliquid alias! Quam est exercitationem debitis aliquam, repudiandae perferendis quidem, voluptas dolore vel dolores natus?

responder +

```
public boolean insertForum(int gameId, String nome ){  
  
    String sql = "INSERT INTO forum (jogos_appid,nome) VALUES (?,?)";  
    try{PreparedStatement preparedStatement = conexao.prepareStatement(sql);  
        preparedStatement.setInt(1, gameId);  
        preparedStatement.setString(2, nome);  
        preparedStatement.executeUpdate();  
        return true;}  
    catch(SQLException e){  
        e.printStackTrace();  
        return false;  
    }  
}
```

FIM!!!!

OBRIGADO PELA ATENÇÃO!!!!