**PROJETO “Elenca Jogos”**

*Por “Bruno Coelho Perez”*

RONDONÓPOLIS – MT

2023

**HISTÓRICO DE ALTERAÇÕES**

|  |  |  |
| --- | --- | --- |
| DATA | ATIVIDADE | RESPONSÁVEL |
| 10/08 | Digitei as informações iniciais | Bruno Perez |
| 17/08 | Modelagem UML | Bruno Perez |
| 27/08 | Código e conclusão da fase BD do projeto | Bruno Perez |
| 23/09 | Revisão dos diagramas | Bruno Perez |
| 09/10 | Alteração dos diagramas e adição do código Java para os métodos do aplicativo | Bruno Perez |
| 16/10 | Alteração do Diagrama de Classe e criação do CRUD a ser apresentado em sala | Bruno Perez |
| 20/10 | Nova alteração no diagrama de classes | Bruno Perez |
| 26/10 | Alteração no diagrama de classes e MER | Bruno Perez |
| 21/11 | Alteração dos diagramas e demais partes do modelo | Bruno Perez |
| 26/11 | Alteração nos Diagramas | Bruno Perez |
| 27/11 | Nova alteração nos Diagramas e no campo do Banco de Dados | Bruno Perez |
| 30/11 | Finalização do modelo de projeto | Bruno Perez |

SUMÁRIO

[**1.** **DESCRIÇÃO DO CENÁRIO** 4](#_Toc142526958)

[1.1. REQUISITOS FUNCIONAIS 4](#_Toc142526959)

[1.2. REQUISITOS NÃO FUNCIONAIS 4](#_Toc142526960)

[**2.** **MODELAGEM UML** 5](#_Toc142526961)

[2.1. DIAGRAMA CASOS DE USO 5](#_Toc142526962)

[2.2. DIAGRAMA DE CLASSES 5](#_Toc142526963)

[2.3. DIAGRAMA DE SEQUÊNCIA 5](#_Toc142526964)

[2.4. DIAGRAMA DE ATIVIDADES 5](#_Toc142526965)

[**3.** **BANCO DE DADOS** 6](#_Toc142526966)

[3.1. MODELO CONCEITUAL 6](#_Toc142526967)

[3.2. MODELO LÓGICO 6](#_Toc142526968)

[**4.** **CODIFICAÇÃO** 6](#_Toc142526969)

[**5.** **CONCLUSÃO** 7](#_Toc142526970)

[**REFERÊNCIAS** 8](#_Toc142526971)

# **DESCRIÇÃO DO CENÁRIO**

# REQUISITOS FUNCIONAIS

* O usuário pode pesquisar os jogos por nome ou categoria. Os jogos são baseados na Play Store;
* Os jogos indicados são limitados, pois foram feitos com base na pesquisa do programador;
* Será criado um servidor do Discord para conversas entre usuários Android, contendo um link para ele;
* Possuirá login e senha;
* Possuirá um ranking anual com os Top 10 Jogos bem avaliados pelos usuários;
* Pode gerar relatórios;
* Também terá links com sites de empresas desenvolvedoras de jogos à Play Store;
* O sistema informará ao usuário jogos que não ocupem mais de 1gb de espaço no HD, por se tratar de um público que não possui um celular gamer;

# REQUISITOS NÃO FUNCIONAIS

* O sistema não indicará jogos adultos, por uma questão de ética profissional;
* O sistema não terá backups frequentes;
* O sistema não trabalhará com incrementações, devido a um único programador no projeto.

# **MODELAGEM UML**

# **DIAGRAMA CASOS DE USO**

Diagrama

Descrição gerada automaticamente

# 

# **DIAGRAMA DE CLASSES**

Diagrama

Descrição gerada automaticamente

# **DIAGRAMA DE SEQUÊNCIA 1**

Diagrama

Descrição gerada automaticamente

NOTA: o último campo chama-se PesquisaCategoria.

# **DIAGRAMA DE SEQUÊNCIA 2**

Diagrama

Descrição gerada automaticamente

# **DIAGRAMA DE ATIVIDADES**

Diagrama, Esquemático

Descrição gerada automaticamente

# **Nota:** última raia é Categoria.

# **BANCO DE DADOS**

# **MODELO CONCEITUAL**

Diagrama

Descrição gerada automaticamente

# **MODELO LÓGICO**

Diagrama, Tabela

Descrição gerada automaticamente

# **CODIFICAÇÃO**

**4.1 Conexão com o banco de dados**

Foi feita no Postgre. Apresentarei os seguintes código, presentes no NetBeans.

Na classe Conexao:

*private static final String URL = "jdbc:postgresql://localhost:5432/elenca\_jogos";*

*private static final String USER = "postgres";*

*private static final String PASSWORD = "1234";*

Esta classe é essencial para a utilização do Banco de Dados. Estes três campos fazem a conexão. O primeiro, URL: especifica o local onde está sendo usado o Postgre, no caso numa máquina qualquer, e o nome do banco chamado elenca\_jogos; O segundo, USER: o nome do usuário que faz o login no psql, chamado de postgres; Por fim, PASSWORD: a senha para acessar o banco Postgre. Optei por uma senha simples, pois ainda não sei utilizar hash.

O próximo é o método getConnection, visto nesta linha:

*public static Connection getConnection() {*

Este método é acionado por cada método que faz a conexão com o Banco de Dados. Por exemplo, o método para fazer login no sistema:

*public boolean verLogin(String email, String senha) {*

*Connection con = Conexao.getConnection();*

Note que getConnection() é acionado na linha abaixo do método verLogin; Logo, sempre que ele fizer a conexão ele deverá ser acionado.

Outro ponto importante de conexão é por este método:

*public static void closeConnection(Connection con) {*

Este closeConnection(Connection con) foi utilizado aqui:

*Connection con = Conexao.getConnection();*

Ele foi utilizado para iniciar a conexão.

O próximo passo é este:

public static void closeConnection(Connection con, PreparedStatement stmt) {

Note que este closeConnection aciona Connection con e PreparedStatement stmt. Este Segundo é utilizado assim na classe UsuarioDAO:

*PreparedStatement stmt = null;*

Sim, ele é iniciado como nulo e é utilizado durante o código. Veja o exemplo:

*stmt = con.prepareStatement("SELECT \* FROM cadastro WHERE "*

*+ "email = ? and senha = ?");*

Note que ele é feito em conjunto com con, um objeto de closeConnection.

Por fim, esta linha:

*public static void closeConnection(Connection con, PreparedStatement stmt,*

*ResultSet rs) {*

As três estão unidas para a função de fechar a conexão. Um exemplo da última função, chamada ResultSet rs:

ResultSet rs = null;

Sim, esta função também é iniciada como nula. Mas logo abaixo do código, ela é “chamada”:

*rs = stmt.executeQuery();*

*if (rs.next()) {*

*check = true;*

*}*

Ela faz a consulta do banco de dados e verifica se há uma próxima linha/coluna no banco de dados; Se verdadeiro, continua; Se falso, finaliza a consulta.

Finalizando, veja esta linha:

*stmt = con.prepareStatement("SELECT \* FROM cadastro WHERE "*

*+ "email = ? and senha = ?");*

Nela, são utilizadas as funções stmt e com e é feita uma varredura no banco de dados (novamente: elenca\_jogos, definido no começo) na tabela sublinhada onde verifica-se somente os campos/linhas da tabela “email” e “senha”. Como foi ensinado, utilizei id auto-gerado para facilitar o CRUD caso necessário. Outro ponto: nesta tabela contém: id, nome, data\_nasc, email e senha. Portanto, foi necessário especificar apenas e-mail e senha para o login e ocultando os outros campos – para facilidade do usuário.

Os código que utilizei no PostgreSQL (preferi utilizar o psql):

CREATE DATABASE elenca\_jogos;

CREATE TABLE cadastro (

id SERIAL PRIMARY KEY,

nome VARCHAR(100) NOT NULL,

data\_nasc VARCHAR(10) NOT NULL,

email VARCHAR(50) NOT NULL,

senha VARCHAR(50) NOT NULL

);

CREATE TABLE jogos (

id SERIAL PRIMARY KEY,

jogo VARCHAR(100) NOT NULL,

nota DOUBLE PRECISION NOT NULL,

produtora VARCHAR(100) NOT NULL,

classificacao VARCHAR(10) NOT NULL,

categoria\_id INTEGER,

FOREIGN KEY (categoria\_id) REFERENCES categoria\_elenca1(id)

);

CREATE TABLE categoria\_elenca1 (

id SERIAL PRIMARY KEY,

categoria VARCHAR(50) NOT NULL

);

**4.2 Visão do Usuário**

**Interface gráfica do usuário

Descrição gerada automaticamente**

Nesta tela, é utilizada a tabela cadastro do Postgre. Cada campo está presente no BD. Importante: na tela anterior, a de Login, são utilizados apenas email e senha valendo-se de um SELECT que busca apenas estes dois. Veja o código:

*stmt = con.prepareStatement("SELECT \* FROM cadastro WHERE "*

*+ "email = ? and senha = ?");*

Agora, na segunda tela que é a principal do programa:

Interface gráfica do usuário

Descrição gerada automaticamente

Nesta, é utilizado um *readJTable();* que, através de um método presente nesta InternalFrame gera tudo o que está na tabela jogos para visualização do usuário. Há também duas formas diretas do usuário encontrar seus jogos: pelo campo Jogos ou Categoria. Pelo Nome do Jogo:

Interface gráfica do usuário

Descrição gerada automaticamente

Noteque o botão BUSCAR está utilizando o método *readJTableForJogo(campoNomeJogo.getText());* que basicamente busca jogos com a palavra ball(como no exemplo). Agora veja pela Categoria:

*Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente*

No campo categoria é utilizado um código similar, mudando seu nome e utilizando o ID, que é a chave primária:

*String texto = campoCategoriaInterna.getText();*

*try {*

*int numero = Integer.parseInt(texto);*

*readJTableForCategoria(numero);*

*} catch (NumberFormatException e) {*

*System.out.println("Por favor, insira um número válido.");*

*}*

**4.3 Manipulação do Banco de Dados**

Utilizei os seguintes métodos para manipular os dados no BD elenca\_jogos: *inserirJogo(), inserirCategoria(), deletarJogo(), deletarCategoria(), readForJogo(), readForCategoria(), updateJogos(), updateCategoria()*. Estes estão presentes na classe DAO.

Já na classe UsuarioDAO: *verLogin(), inserir()*. Estes métodos são usados na tela de Cadastro do programa.

Detalhando os métodos do DAO: o CRUD está presente apenas na TelaADM e os métodos de procura, readForJogo e readForCategoria, são usados nos botões de buscar para jogo ou categoria.

Como exemplo, posso explicar sobre o método verLogin() presente na tela inicial do programa:

*public boolean verLogin(String email, String senha) {*

*Connection con = Conexao.getConnection();*

*PreparedStatement stmt = null;*

*ResultSet rs = null;*

*boolean check = false;*

*try {*

*stmt = con.prepareStatement("SELECT \* FROM cadastro WHERE "*

*+ "email = ? and senha = ?");*

*stmt.setString(1, email);*

*stmt.setString(2, senha);*

*rs = stmt.executeQuery();*

*if (rs.next()) {*

*check = true;*

*}*

*} catch (SQLException ex) {*

*Logger.getLogger(*

*UsuarioDAO.class.getName()).log*

*(Level.SEVERE, null, ex);*

*} finally {*

*Conexao.closeConnection(con, stmt, rs);*

*}*

*return check;*

*}*

Este método faz a verificação do email e senha cadastrados na tela de Cadastro “puxando” estes dois campos. Veja o código:

*stmt = con.prepareStatement("SELECT \* FROM cadastro WHERE "*

*+ "email = ? and senha = ?");*

No SELECT há uma especificação: ONDE email e senha sejam valores x. Ou seja, o programa faz uma varredura na tabela cadastro e busca nos campos email e senha valores iguais aos informados na tela Cadastro. Como exemplo, o meu login é brunopz@gmail.com senha 963. O programa verifica se foi isso mesmo que informei. Caso verdadeiro, é feito o login e o usuário pode utilizar o programa.

# **CONCLUSÃO**

Elenca Jogos é um projeto simples que pensei em uma função para o meu cliente. Assim como qualquer programa, é uma alternativa para facilitar a vida do usuário.

Futuramente, penso em fazer uma aplicação ainda melhor, só alterando os requisitos, a documentação e funcionalidades para um programa mais robusto.

Acredito que novos aplicativos podem ser um diferencial para determinado grupo de pessoas. No nosso caso, o Brasil é carente de aplicativos/sites/fóruns e nós podemos ser a alternativa para mudar este cenário de limitações.

Quero também destacar a importância da IA, em especial o Bing que me ajudou muito. Apesar de ser uma tecnologia recente, ela possui grande potencial de crescimento. Para os trabalhadores da TI, será de grande valia para informação, revisão de erros e um facilitador para o Programador(a) – visto que ela tem sido amplamente usada por eles/elas segundo Zdnet: 92% dos programadores usam ferramentas de IA, diz pesquisa de desenvolvedores do GitHub (traduzido pelo autor. Link: <https://www.zdnet.com/article/github-developer-survey-finds-92-of-programmers-using-ai-tools/> ).

Estou teorizando, mas se tivermos que desenvolver programas para robôs, como os robôs-ajudantes do filme O Homem Bicentenário ou até mesmo do jogo Detroit: Become Human teremos sim uma longa jornada de trabalho e realizações futuras. Será exigido dos programadores que aperfeiçoem-se mais e mais para esta nova era da tecnologia.

# **REFERÊNCIAS**

Google. Elenca jogos. Disponível: <https://www.google.com/search?q=elenca+jogos&sca\_esv=555497980&sxsrf=AB5stBhyXD0UvAs3xhhh2GRfSBlypGGoCQ%3A1691688027118&source=hp&ei=WxzVZNeCBazQ1sQP-fi9mAI&iflsig=AD69kcEAAAAAZNUqa7Xv09kXWb04YAhwhM\_RZ\_CiZLKK&ved=0ahUKEwiXnLP\_zNKAAxUsqJUCHXl8DyMQ4dUDCAk&uact=5&oq=elenca+jogos&gs\_lp=Egdnd3Mtd2l6IgxlbGVuY2Egam9nb3MyBxAAGA0YgAQyBxAAGA0YgAQyBxAAGA0YgAQyBxAAGA0YgAQyBxAAGA0YgAQyBxAAGA0YgAQyBxAAGA0YgAQyBxAAGA0YgAQyBxAAGA0YgAQyBxAAGA0YgARIqg5QAFixDHAAeACQAQCYAa0BoAHzDaoBBDAuMTK4AQPIAQD4AQHCAhEQLhiABBixAxiDARjHARjRA8ICCxAuGIoFGLEDGIMBwgIFEAAYgATCAggQABiABBixA8ICCxAAGIAEGLEDGIMBwgILEC4YgAQYsQMYgwHCAhEQLhiKBRixAxiDARjHARjRA8ICCxAAGIoFGLEDGIMBwgIIEC4YsQMYgATCAggQLhiABBixA8ICChAAGIAEGLEDGArCAgcQABiABBgK&sclient=gws-wiz#ip=1>. Acesso em: 10 ago 2023.

Bing. Conversas/Pesquisas relacionadas ao Java, NetBeans e Postgre. Disponível: <https://www.bing.com/search? form=WSBCTB&toWww=1&redig=FEA32FE3DCFD48C5BD01C5EB7491FB24&q=Bing+AI&showconv=1>. Acesso em: 01 nov 2023.

Zdnet. 92% of programmers are using AI tools, says GitHub developer survey. Disponível: <https://www.zdnet.com/article/github-developer-survey-finds-92-of-programmers-using-ai-tools/>. Acesso em: 26 nov 2023.

Stack Overflow. Illegal forward reference in java. Disponível: <https://stackoverflow.com/questions/1746758/illegal-forward-reference-in-java>. Acesso em: 22 nov 2023.

Stack Overflow. [why the public static void main(String args[]) error and it say illegal start of expression?](https://stackoverflow.com/questions/37098278/why-the-public-static-void-mainstring-args-error-and-it-say-illegal-start-of) Disponível: <https://stackoverflow.com/questions/37098278/why-the-public-static-void-mainstring-args-error-and-it-say-illegal-start-of>. Acesso em: 22 nov 2023.

Stack Overflow. [How ca i set Background Color in my JInternalFrames?](https://stackoverflow.com/questions/41197425/how-ca-i-set-background-color-in-my-jinternalframes) Disponível: <https://stackoverflow.com/questions/41197425/how-ca-i-set-background-color-in-my-jinternalframes>. Acesso em: 21 nov 2023.

Youtube. How do you add a jFrame to your main class in Netbeans. Disponível: <https://www.youtube.com/watch?v=cxM87izbEMo>. Acesso em: 20 nov 2023.

Stack Overflow. Populate a jCombobox in netbeans 7.1. Disponível: <https://stackoverflow.com/questions/9535260/populate-a-jcombobox-in-netbeans-7-1>. Acesso em: 15 nov 2023.

Stack Overflow. [How to fix 'Exception in thread "AWT-EventQueue-0" java.lang.NullPointerException' on code generated by NetBeans?](https://stackoverflow.com/questions/55957753/how-to-fix-exception-in-thread-awt-eventqueue-0-java-lang-nullpointerexceptio) Disponível: <https://stackoverflow.com/questions/55957753/how-to-fix-exception-in-thread-awt-eventqueue-0-java-lang-nullpointerexceptio>. Acesso em: 8 nov 2023.

Stack Overflow. [Java NetBeans Error: run: Exception in thread "AWT-EventQueue-0" java.awt.IllegalComponentStateException: The frame is decorated](https://stackoverflow.com/questions/55021462/java-netbeans-error-run-exception-in-thread-awt-eventqueue-0-java-awt-illega). Disponível: <https://stackoverflow.com/questions/55021462/java-netbeans-error-run-exception-in-thread-awt-eventqueue-0-java-awt-illega>. Acesso em: 8 nov 2023.

Stack Overflow. [Postgres table visible in psql, not from jdbc](https://stackoverflow.com/questions/21270970/postgres-table-visible-in-psql-not-from-jdbc). Disponível: <https://stackoverflow.com/questions/21270970/postgres-table-visible-in-psql-not-from-jdbc>. Acesso em: 8 nov 2023.

Stack Overflow. [How do I implement a DAO manager using JDBC and connection pools?](https://stackoverflow.com/questions/12812256/how-do-i-implement-a-dao-manager-using-jdbc-and-connection-pools) Disponível: <https://stackoverflow.com/questions/12812256/how-do-i-implement-a-dao-manager-using-jdbc-and-connection-pools>. Acesso em: 31 out 2023.

DevMedia. Modelagem de sistemas através de UML: uma visão geral. Disponível: <https://www.devmedia.com.br/modelagem-de-sistemas-atraves-de-uml-uma-visao-geral/27913>. Acesso em: 17 ago 2023.

Lucid Software Português. Tutorial de Caso de Uso UML. Disponível: <https://www.youtube.com/watch?v=ab6eDdwS3rA>. Acesso em: 17 ago 2023.

LORENZ, Wagner Gadea. UML – Diagrama de Classe – resumo. Disponível: <https://www.youtube.com/watch?v=guH37e4MPLs>. Acesso em: 17 ago 2023.

Boson Treinamentos. Curso de UML – Diagrama de Sequência UML – Exemplo Básico. Disponível: <https://www.youtube.com/watch?v=LeV6RO-6Tn4>. Acesso em: 17 ago 2023.

PICCIN, Jonas. Exemplo de modelagem do diagrama de Sequência. Disponível: <https://www.youtube.com/watch?v=SsAzPzgrmsw&t=46s>. Acesso em: 17 ago 2023.

CAETANO, Gustavo. Diagrama de Atividades. Disponível: <https://www.youtube.com/watch?v=DcsGnz7Kg5o>. Acesso em: 17 ago 2023.