

Instituto Mauá de Tecnologia

Amanda Carolina Ambrizzi Ramin 22.00721-0

André Felipe Silva Xavier 22.01339-3

Otto Camargo Kuchkarian 22.00571-4

Rafael Assanti 22.01464-0

Rodrigo Fernandes Faltz 21.01660-7

**Batalha Espacial**

Projeto de Linguagens de Programação I. – ECM251

São Caetano do Sul

2024

## **Resumo**

O seguinte trabalho propõe o desenvolvimento de um jogo semelhante ao famoso “Batalha Naval” com a temática da conhecida franquia Star Wars, utilizando a linguagem de programação Java e suas bibliotecas. Através desta implementação, busca-se explorar as diversas funcionalidades e capacidades de Java para a criar um jogo interativo e funcional. O projeto a seguir envolvera a criação de uma interface gráfica, a lógica de jogo necessária para simular a batalha dos navios, e a implementação de interação com um banco de dados. Este estudo tem como objetivo principal não apenas desenvolver um jogo, mas também demonstrar a aplicabilidade de Java em diversos projetos.

**Palavras-Chaves:** Batalha Naval, Star Wars, Java, Jogo

## **Abstract**

The following work proposes the development of a game similar to the famous "Battleship" with the theme of the well-known Star Wars franchise, using the Java programming language and its libraries. Through this implementation, we aim to explore the various functionalities and capabilities of Java to create an interactive and functional game. The project will involve the creation of a graphical interface, the game logic necessary to simulate the battle of the ships, and the implementation of interaction with a database. This study aims not only to develop a game but also to demonstrate the applicability of Java in various projects.

**Keywords:** Battleship, Star Wars, Java, Game

## Sumário

<b>1.Introdução .....</b>	<b>4</b>
<b>2.Metodologia .....</b>	<b>5</b>
<b>3.Planejamento e Desenvolvimento .....</b>	<b>6</b>
<b>4.Manual de Instruções .....</b>	<b>12</b>
<b>5.Resultados e Discussão .....</b>	<b>15</b>
<b>6.Conclusão .....</b>	<b>16</b>
<b>7.Referências Bibliográficas .....</b>	<b>17</b>

## 1. Introdução

O projeto a seguir é multidisciplinar, envolvendo diversas áreas do desenvolvimento de software, enquanto sua importância reside na aplicação prática de conceitos de programação, no desenvolvimento de habilidades essenciais e na demonstração da eficácia e versatilidade da linguagem Java. O projeto não só proporciona uma plataforma de aprendizado robusta, mas também cria uma aplicação funcional e iterativa que exemplifica os princípios e prática de desenvolvimento de *software* modernos.

A escolha de desenvolver um jogo com a linguagem Java é fundamentada na oportunidade de aprendizagem prática, nos desafios envolventes que estimulam o engajamento dos desenvolvedores e na demonstração da aplicabilidade real da linguagem em projetos complexos.

Por meio deste projeto, espera-se não apenas criar um jogo envolvente, mas também destacar o uso de Java em projetos complexos e sua eficiência em diversas aplicações de desenvolvimento de *software*.

O objetivo deste trabalho é desenvolver um jogo semelhante ao “Batalha Naval” utilizando a linguagem de programação Java e suas bibliotecas para criação de interfaces gráficas e se conectar com um banco de dados SQL. Especificamente, busca-se:

- ✓ **Explorar as funcionalidades da linguagem Java:** Demonstrar as diversas capacidades da linguagem de programação Java na criação de um jogo interativo e funcional;
- ✓ **Desenvolver uma interface gráfica:** Criar uma interface gráfica amigável e intuitiva que permita aos jogadores interagirem facilmente com o jogo;
- ✓ **Implementar a lógica de jogo:** Estabelecer as regras e a mecânica do “Batalha Naval” para simular a batalha dos navios;
- ✓ **Integrar com um banco de dados:** Implementar a interação do jogo com um banco de dados para salvar o estado do jogo, como se fosse um *pause*;
- ✓ **Demonstrar a aplicabilidade de Java:** Mostrar como Java pode ser utilizado em diversos projetos, exemplificando através do desenvolvimento de um jogo.

A mecânica do jogo é baseada no clássico “Batalha Naval”, em que os jogadores colocam as suas naves dentro de uma matriz e os jogadores precisam acertar as naves uns dos outros. Ao iniciar o jogo, o primeiro jogador seleciona onde ele quer colocar suas naves no tabuleiro. Os tamanhos das naves variam de 2 a 5. Ao terminar de colocar as naves, é a vez do outro jogador fazer o mesmo. Ao término da fase de preparação, o jogo se inicia.

O primeiro jogador tenta acertar a nave do adversário, se ele errar, a vez é passada para o segundo jogador. Caso o jogador acerte uma nave, ele poderá jogar de novo, até não acertar uma nave ou conseguir derrubar todas as naves, vencendo o jogo.

No caso do projeto, é possível salvar um jogo já iniciado a qualquer momento, clicando no botão “Salvar”. Os dados vão para a base de dados de um MySQL e o usuário pode voltar a jogar em outro momento.

## 2. Metodologia

Para atingir os objetivos propostos, seguiu-se as seguintes etapas:

### ✓ **Planejamento e Design:**

- a. Especificação: Definição das funcionalidades do jogo;
- b. Design da Interface do Usuário (UI): Criação das telas da interface gráfica;
- c. Modelagem do Banco de Dados: Definição das tabelas e relações necessárias para armazenar o estado do jogo.

### ✓ **Desenvolvimento:**

- a. Configuração do Ambiente de Desenvolvimento: Conexão com banco de dados e importação das bibliotecas necessárias;
- b. Implementação da Lógica do Jogo: Codificação das regras e mecânicas do “Batalha Naval”;
- c. Desenvolvimento da Interface Gráfica: Utilização de bibliotecas Java para criar a interface do usuário;
- d. Integração com o Banco de Dados: Implementação das funcionalidades de salvamento e carregamento do estado do jogo.

### ✓ **Testes e Validação:**

- a. Testes Unitários: Verificação da implementação das funções e métodos;
- b. Testes de Integração: Garantia de que os componentes funcionam juntos;
- c. Testes de Usabilidade: Avaliação da interface gráfica com usuários para assegurar a intuitividade do jogo.

### ✓ **Documentação e Apresentação:**

- a. Documentação do Código: Comentários e documentação das funções e métodos implementados;
- b. Manual do Usuário: Criação de um guia para os jogadores, explicando como utilizar o jogo;
- c. Apresentação Final: Demonstração do jogo e discussão dos resultados alcançados.

### ✓ **Especificações Tecnológicas:**

- a. Linguagem de Programação: Java;
- b. Biblioteca para Interface Gráfica: Swing;
- c. Banco de Dados: MySQL;
- d. IDE (Integrated Development Environment): VSCode.

### 3. Planejamento e Desenvolvimento

#### 3.1. Planejamento

Iniciou-se o projeto pelas especificações. Nessa etapa, foram detalhados os requisitos funcionais e não funcionais que o jogo deve atender. As funcionalidades incluem:

- ✓ Permitir que os jogadores posicionem suas naves;
- ✓ Alternação de turnos entre os jogadores;
- ✓ Interface gráfica intuitiva para facilitar a interação com o jogo;
- ✓ Opção de salvar o jogo para pausar e ser retomado depois;
- ✓ Permitir partidas entre dois jogadores na mesma máquina.

Após as especificações, iniciou-se o processo de design da interface gráfica. As telas incluem:

- ✓ Tela Inicial: tela com as opções de iniciar novo jogo, continuar jogo passado ou ler as regras de funcionamento (Figura 1);

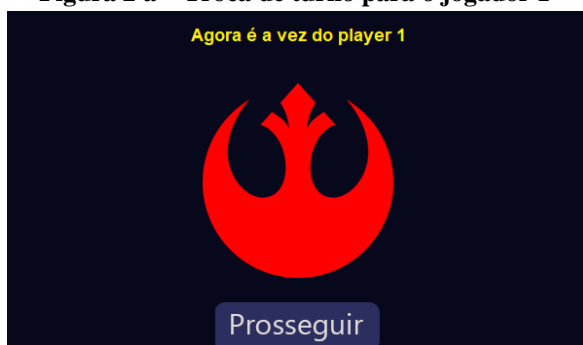
**Figura 1 – Tela Inicial**



Fonte: próprios autores

- ✓ Tela Intermediária: tela de aviso de troca de turno (Figura 2a e 2b);

**Figura 2 a – Troca de turno para o jogador 1**



Fonte: próprios autores

**Figura 2 a – Troca de turno para o jogador 2**



Fonte: próprios autores

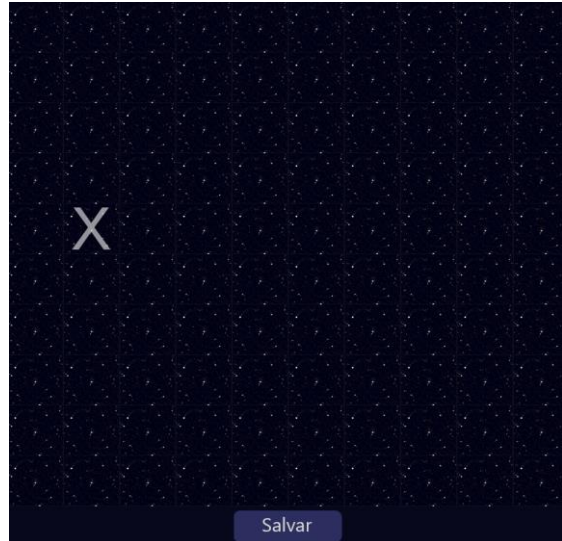
✓ Tela de Ataque: tela de jogo (Figura 3a, 3b, 3c e 3d);

**Figura 3 a – tela de ataque**



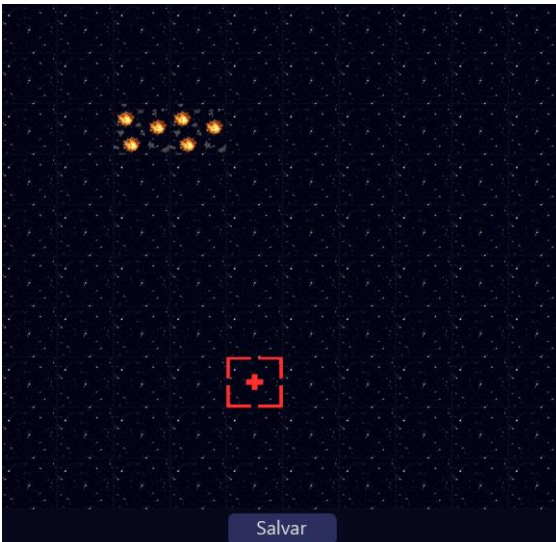
Fonte: próprios autores

**Figura 3 b – tela quando erra o ataque**



Fonte: próprios autores

**Figura 3 c – tela quando acerta o ataque**



Fonte: próprios autores

**Figura 3 d – tela quando detona a nave**

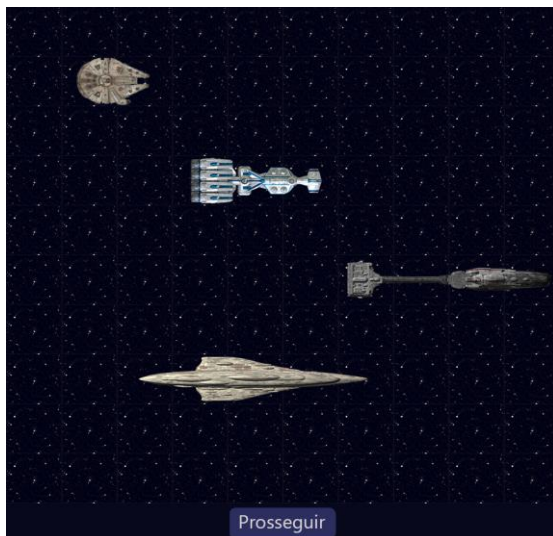


Fonte: próprios autores

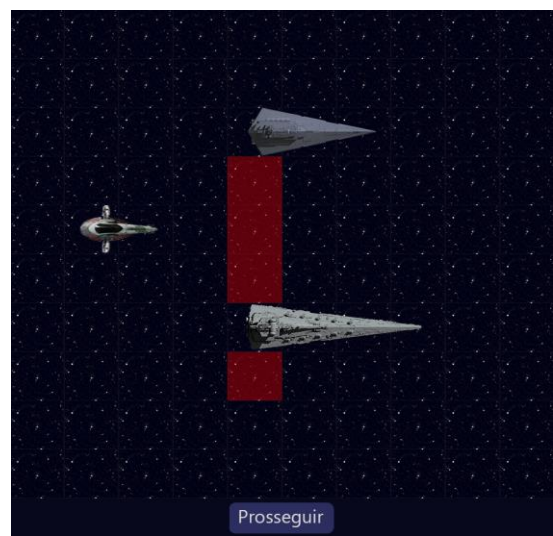
✓ Tela de Setup: tela de posicionar as naves (Figura 4a, 4b e 4c);

**Figura 4 a – Tela de Setup com exemplo e sem alerta de posicionamento**

**Figura 4 b – Tela de Setup com exemplo e com alerta de posicionamento**

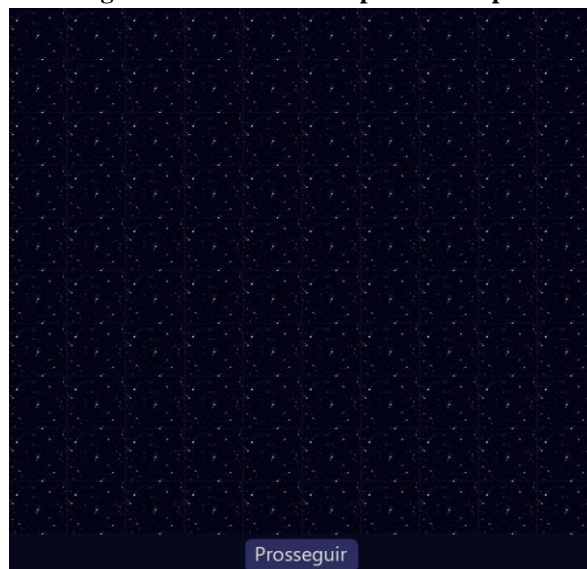


Fonte: próprios autores



Fonte: próprios autores

**Figura 4 c – Tela de Setup sem exemplo**



Fonte: próprios autores

- ✓ Tela Após Setup: tela de troca de jogador para posicionar as naves (Figura 5);

**Figura 5 – Tela Após Setup**

Obs.: jogador 1 sempre começa colocando as naves, por isso apenas uma tela após setup.



Fonte: próprios autores



- ✓ Tela de Vitória: tela de aviso de final do jogo (Figura 6a e 6b);

**Figura 6 a – Tela de Vitória do Jogador 1**



Fonte: próprios autores

**Figura 6 b – Tela de Vitória do Jogador 2**



Fonte: próprios autores

- ✓ Tela de Como Jogar: tela de instruções de como o jogo funciona;

Em seguida, modelou-se o banco de dados para salvar o jogo que será continuado. As tabelas e relações necessárias são:

- ✓ Tabela de Acertos com Player, número de acerto do Jogador 1 e do Jogador 2, quantidade de acertos em cada nave (Figura 7);

**Figura 7 – Criação da Tabela Acertos**

```
String criaTabelaDeAcertos = "CREATE TABLE Acertos ("
+ "player INT, "
+ "numeroDeAcertosPlayer1 INT, "
+ "numeroDeAcertosPlayer2 INT, "
+ "AcertosNave2P1 INT, "
+ "AcertosNave2P2 INT, "
+ "AcertosNave3P1 INT, "
+ "AcertosNave3P2 INT, "
+ "AcertosNave4P1 INT, "
+ "AcertosNave4P2 INT, "
+ "AcertosNave5P1 INT, "
+ "AcertosNave5P2 INT)";
```

Fonte: próprios autores

- ✓ Tabela de Posicionamento do Jogador 1 com as linhas, colunas e valor (Figura 8);

**Figura 8 – Criação da Tabela GridP1**

```
String criaGridP1 = "CREATE TABLE GridP1 ( linha INT, coluna INT, valor INT)";
```

Fonte: próprios autores

- ✓ Tabela de Posicionamento do Jogador 2 com as linhas, colunas e valor (Figura 9).

**Figura 9 – Criação da Tabela GridP2**

```
String criaGridP1 = "CREATE TABLE GridP1 ( linha INT, coluna INT, valor INT)";
```

Fonte: próprios autores

### 3.2. Desenvolvimento

Para iniciar o desenvolvimento e testes do projeto, a escolha de uma IDE adequada é fundamental. Nesse contexto, foi utilizada a IDE Visual Studio Code (VSCode), devido a sua ampla popularidade e interface intuitiva, que facilita a condução de projetos de maneira eficiente.

Além da escolha do VSCode como ambiente de desenvolvimento, foi crucial a decisão das bibliotecas Java para auxiliar no projeto. A biblioteca Swing proporciona uma ampla gama de componentes gráficos para a construção de interfaces de usuário (UI), facilitando a criação de aplicações visualmente atraentes e funcionais. Já o JDBC (Java Database Connectivity) foi essencial para realizar a comunicação entre o projeto e o banco de dados, permitindo o acesso e manipulação dos dados de forma eficiente. E, por fim, a integração com o MySQL como sistema gerenciador de banco de dados proporciona uma plataforma robusta e confiável para armazenamento e recuperação dos dados do projeto. Essas escolhas foram feitas visando garantir a eficiência, praticidade e robustez do desenvolvimento do projeto.

Com as bibliotecas separadas e com o ambiente de trabalho pronto para começar o desenvolvimento do projeto, deve-se estabelecer as mecânicas e a lógica por traz do jogo que será criado.

O jogo é baseado em "Batalha Naval", porém com a temática de Star Wars, em que os jogadores posicionam suas naves em uma grade e tentam adivinhar as localizações das naves adversárias. No início da partida, o primeiro jogador posiciona suas naves no tabuleiro. As naves variam de tamanho, ocupando entre 2 a 5 espaços. Ao terminar de colocar as naves, é a vez do segundo jogador fazer o mesmo.

Ao término da fase de preparação, o jogo se inicia. O jogador que inicia a rodada tenta adivinhar onde estão localizadas as naves do adversário. Se ele errar, a vez passa para o segundo jogador. Mas se o primeiro jogador acertar uma das naves do oponente, ele poderá jogar novamente. Isso continua até que ele erre ou até que todas as naves do adversário sejam derrubadas, vencendo o jogo.

Além disso, o progresso do jogo pode ser salvo a qualquer momento. Ao clicar no botão "Salvar", o estado atual do jogo é armazenado em um banco de dados MySQL. Isso permite que os jogadores interrompam a partida e retomem de onde pararam em outro momento, sem perder nenhum progresso. Essa funcionalidade é útil para partidas longas ou quando os jogadores não têm tempo suficiente para concluir o jogo de uma só vez. A capacidade de recuperar o estado do jogo é um diferencial em comparação com a versão clássica do jogo, que exigia que a partida fosse completada de uma só vez.

Já em relação a interface gráfica, foram usadas as bibliotecas Swing e AWT próprias da linguagem Java, sendo criadas as telas iniciais, intermediárias e de jogo. A AWT (Abstract Window Toolkit) fornece um conjunto básico de componentes GUI que dependem dos recursos nativos do sistema operacional, como janelas, botões e caixas de texto. Já o Swing é uma biblioteca que oferece maior variedade de componentes, como botões, tabelas, listas, painéis com abas etc. O Swing desenha todos os componentes

utilizando Java, resultando em uma aparência e comportamento consistentes em diferentes sistemas operacionais.

As telas iniciais, intermediárias e de jogo foram criadas utilizando essas bibliotecas, resultando em uma interface intuitiva para o usuário. As classes de interface gráfica desenvolvidas são: ComoJogar, TelaInicial, TelaIntermediaria, TelaDeSetup, TelaDeAtaque, TelaAposSetup e TelaVitoria.

Cada uma dessas classes possui seus atributos, construtores e métodos específicos para criar interfaces funcionais e esteticamente agradáveis. Esses componentes garantem que o usuário possa interagir com o jogo de maneira fluida e intuitiva.

Para a integração e interação com o banco de dados, foi necessário criar as classes: CriaDBs, EnviaProDB, RecebeDoDB e LimpaTabelasDoDB. Cada uma dessas classes desempenha um papel no gerenciamento dos dados e na manutenção do banco de dados utilizado para armazenar informações relacionadas aos acertos e aos grids do jogo. Todas as classes citadas acima fazem a conexão com o banco de dados MySQL usando usuário e senha estabelecidos, as lógicas estão implementadas dentro dos construtores, que também tratam as exceções.

A classe CriaDBs é responsável por criar as tabelas necessárias. Esta classe cria as tabelas que serão utilizadas para armazenar os dados dos acertos e dos grids do jogo. Isso é fundamental para iniciar o jogo.

A classe EnviaProDB é utilizada para inserir dados nas tabelas criadas. Esta classe executa instruções SQL INSERT para adicionar registros nas tabelas. Isso é importante para armazenar os resultados do jogo, os acertos e seguimento do jogo.

A classe RecebeDoDB tem a função de selecionar e recuperar dados das tabelas no banco de dados. Esta classe executa instruções SQL SELECT para obter os registros desejados. Isso é útil para visualizar o progresso do jogo e os acertos registrados.

A classe LimpaTabelasDoDB é responsável por apagar as tabelas do banco de dados. Esta classe se conecta ao banco de dados e executa instruções SQL DROP para deletar as tabelas. Isso é essencial para reiniciar o jogo.

Por fim, também foram criadas as classes Imagens e Grid. A classe Grid cria a grade do jogo, a lógica de acertos de cada nave e troca de turno entre os jogadores. Enquanto a classe Imagens carrega os ícones a partir do caminho especificado no código, por exemplo: `public ImageIcon batalha = new ImageIcon ("Lib\\Bataalha.jpg");`.

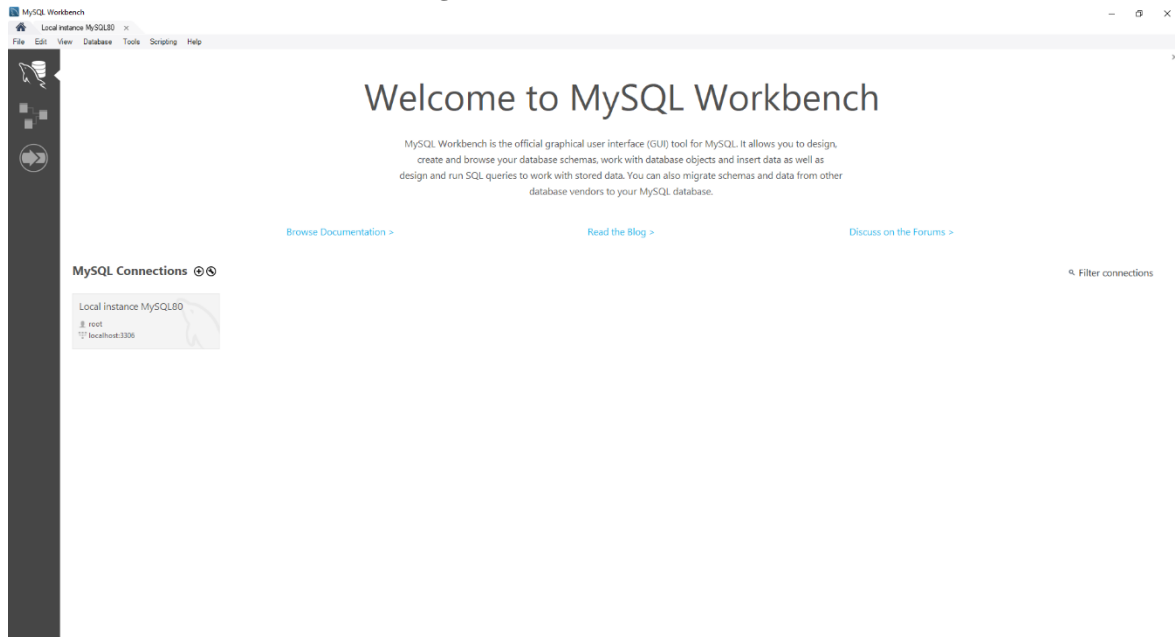
## 4. Manual de Instruções

### 4.1 Iniciar o MySQL

Para começar a jogar, é essencial configurar corretamente o banco de dados MySQL para garantir que o salvamento do jogo ocorra sem erros. Siga os passos abaixo para criar o banco de dados e o usuário MySQL necessários:

1. Abra o MySQL Workbench.
2. Crie ou abra uma conexão “localhost” (Figura 10).

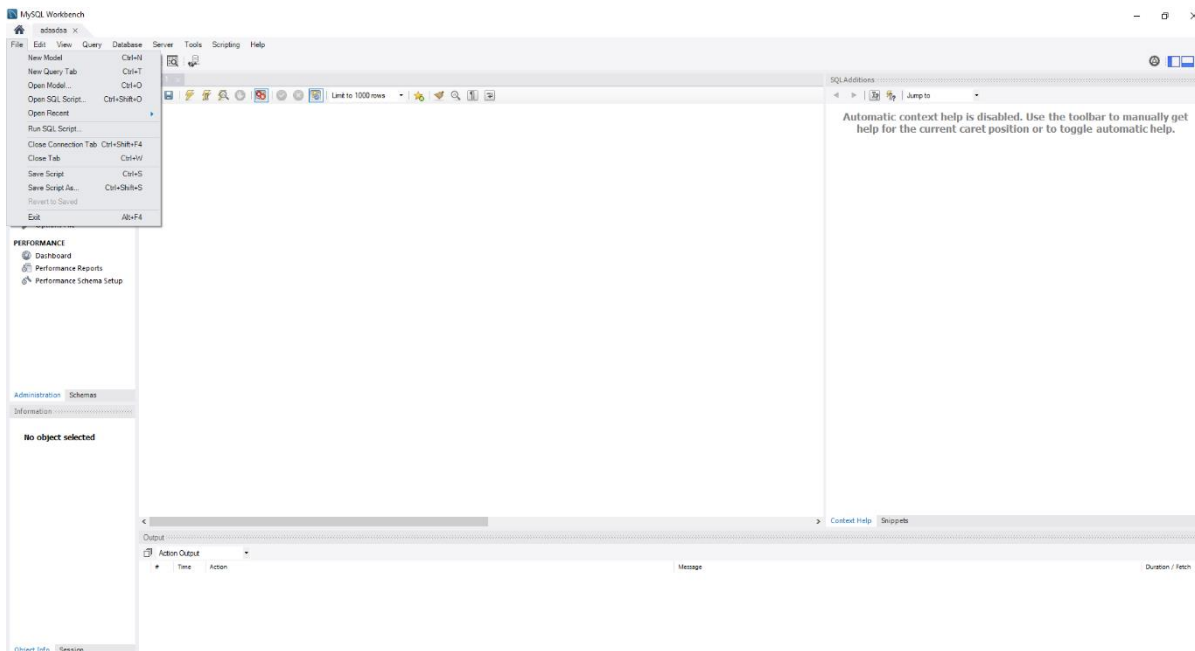
**Figura 10 – Tela Inicial do SQL**



Fonte: próprios autores

3. Clique para abrir um “Script SQL” (Figura 11):

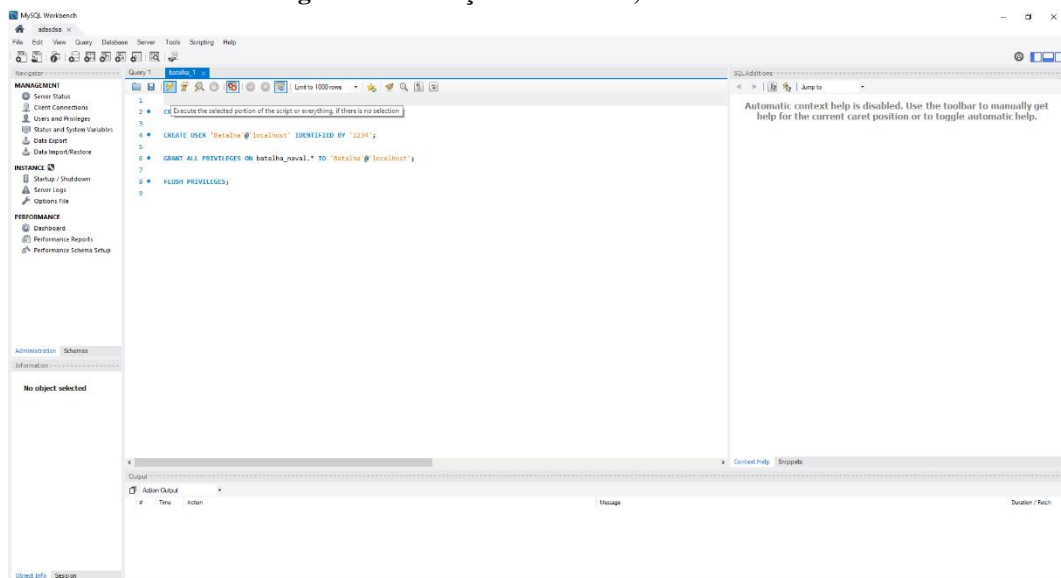
**Figura 11 – Tela Inicial do SQL**



Fonte: próprios autores

4. Abra o arquivo “batalha\_1.sql”:
5. Rode os comandos e feche o MySql Workbench (Figura 12).

**Figura 12 – Criação do Usuário, database e tabelas**



Fonte: próprios autores

## 4.2 Iniciando o Jogo

Após configurar o sql, para começar o jogo, siga as etapas abaixo:

Clique duas vezes no arquivo executável .jar para abrir a janela de menu do jogo.

Na janela de menu, você terá duas opções: criar um novo jogo ou continuar um jogo existente. Se esta for sua primeira vez jogando, clique em "Novo Jogo" para iniciar uma nova partida. Se não for e quiser continuar de onde parou, clique em "Continuar Jogo" para carregar o jogo antigo.

#### **4.3 Objetivo**

O objetivo do jogo é localizar e destruir todas as naves do oponente antes que suas próprias naves sejam destruídas. O primeiro jogador a atingir esse objetivo é declarado o vencedor. Para facilitar a identificação, os jogadores são designados com papéis específicos: o Jogador 1 é a "Aliança Rebelde" e o Jogador 2 é o "Império Intergaláctico".

#### **4.4 Rodada de Preparação**

Uma vez iniciado o jogo, haverá uma fase de preparação onde os jogadores organizam suas naves conforme suas estratégias:

1. Cada jogador terá 4 tipos de naves com tamanhos 2, 3, 4 e 5;
2. O Jogador 1 posiciona suas naves primeiro e, em seguida, clica no botão inferior para passar a vez ao Jogador 2;
3. O Jogador 2 então posiciona suas naves;

Para mudar a orientação de uma nave, clique com o botão direito do mouse.

#### **4.5 Salvar o Jogo**

Se os jogadores desejarem salvar o progresso do jogo para continuar posteriormente, devem clicar no botão de salvar localizado na parte inferior da janela. Isso garantirá que a partida seja salva corretamente, permitindo que os jogadores fechem o jogo sem perder o progresso.

Implementando essas instruções, você garantirá que o banco de dados esteja configurado corretamente e que os jogadores possam desfrutar de uma experiência de jogo contínua e sem interrupções.

## **5. Resultados e Discussão**

### **5.1 Resultados**

A interface gráfica desenvolvida em Java mostrou-se ser intuitiva e atraente para os usuários. Os testes de usabilidade indicaram que os jogadores puderam interagir com o jogo sem dificuldades significativas. A integração com o banco de dados MySQL foi realizada com sucesso, permitindo que o progresso dos jogadores fosse salvo e recuperado eficientemente.

O desempenho do jogo foi estável durante os testes, sem travar. O feedback dos usuários foi positivo. Os jogadores destacaram a fluidez da interface e a confiabilidade do sistema de salvamento de progresso. Comentários específicos dos usuários reforçaram que a interface é agradável e o jogo é divertido.

### **5.2 Discussão**

A análise dos resultados mostra que a interface gráfica cumpriu seu objetivo de ser intuitiva e atraente. A integração do banco de dados MySQL foi eficaz, garantindo que o progresso dos jogadores fosse salvo e recuperado sem problemas. O desempenho estável do jogo indica uma boa otimização do código.

Uma limitação foi a falta de um sistema de redimensionamento responsivo para as imagens. Quando a janela é redimensionada, as imagens não se ajustam adequadamente, prejudicando a experiência visual. Esta limitação afetou os resultados, principalmente na percepção de qualidade da interface pelos usuários.

Para atualizações, é preciso aprimorar o sistema de redimensionamento das imagens. Implementar um redimensionamento tornaria a experiência visual melhor e garantiria que as imagens mantenham suas proporções e qualidade, independentemente do tamanho da janela.

## 6. Conclusão

Após realizar uma série de testes, foi possível desenvolver um jogo que não apenas é interessante e divertido para os usuários, mas também serve como uma aplicação prática dos conhecimentos adquiridos em banco de dados MySQL e das diversas bibliotecas da linguagem Java para a criação de uma interface gráfica amigável. A experiência de desenvolvimento foi desafiadora, especialmente na criação de uma interface que fosse realmente intuitiva e atraente para os jogadores, além da complexa tarefa de integrar um banco de dados que permitisse salvar e recuperar o andamento das partidas.

Apesar desses desafios, a implementação foi um sucesso. O jogo não só atende às expectativas dos jogadores em termos de diversão e envolvimento, como também demonstra um uso eficiente das tecnologias envolvidas. A interface gráfica, desenvolvida em Java, permite uma interação fluida e agradável, enquanto o banco de dados MySQL garante que os progressos dos jogadores sejam armazenados com segurança, proporcionando uma experiência contínua e personalizada.

Para futuras atualizações, é altamente recomendável aprimorar o redimensionamento das imagens no jogo. Atualmente, quando o usuário altera o tamanho da janela, as imagens não se ajustam adequadamente, o que compromete a experiência visual. Este é um ponto crítico, pois uma interface visualmente agradável e responsiva é fundamental para manter os jogadores engajados. Implementar um redimensionamento responsivo garantirá que as imagens se ajustem corretamente ao tamanho da janela, mantendo suas proporções e qualidade independentemente das dimensões definidas pelo usuário. Isso não apenas melhorará a experiência visual, mas também contribuirá para uma sensação de profissionalismo e cuidado com os detalhes no desenvolvimento do jogo.

Em suma, a experiência de desenvolvimento deste jogo foi enriquecedora e os resultados obtidos são promissores. Com as melhorias propostas, especialmente no que diz respeito ao redimensionamento das imagens e outras funcionalidades futuras, o jogo tem o potencial de se tornar ainda mais popular e apreciado pelos usuários, proporcionando horas de diversão e aprendizado contínuo.



## 7. Referências Bibliográficas

- Batalha naval (jogo).** Disponível em: <[https://pt.wikipedia.org/wiki/Batalha\\_naval\\_\(jogo\)#:~:text=Seu%20objetivo%20%C3%A9%20derrubar%20os](https://pt.wikipedia.org/wiki/Batalha_naval_(jogo)#:~:text=Seu%20objetivo%20%C3%A9%20derrubar%20os)>. Acesso em: 30 maio. 2024.
- BENGUITAR90. Star Wars IV: A new hope - Final Scene (The Throne Room) and End Title.** Disponível em: <<https://youtu.be/yixG8pfncOs?si=JJ5v5BiizavSHOtL>>. Acesso em: 1 jun. 2024.
- COLLECTIVE CULTURE. Darth Vader Kills Emperor Palpatine Scene [4k UltraHD] - Star Wars: Return of The Jedi.** Disponível em: <[https://youtu.be/IMzg2X18rm0?si=kwRiVprE\\_mq23lvZ](https://youtu.be/IMzg2X18rm0?si=kwRiVprE_mq23lvZ)>. Acesso em: 31 maio. 2024.
- Como surgiu o jogo Batalha Naval?** Disponível em: <<https://super.abril.com.br/mundo-estranho/como-surgiu-o-jogo-batalha-naval/>>. Acesso em: 30 maio. 2024.
- ESCOLA@WEB.** Disponível em: <<https://escolaweb.educacao.al.gov.br/roteiro-de-estudo/batalha-naval-56721#:~:text=Batalha%20naval%20%C3%A9%20um%20jogo>>. Acesso em: 30 maio. 2024.
- Escravo I.** Disponível em: <[https://starwars.fandom.com/pt/wiki/Legends:Escravo\\_I](https://starwars.fandom.com/pt/wiki/Legends:Escravo_I)>. Acesso em: 29 maio. 2024.
- LIANG, Y. Daniel. Introduction to Java: programming and data structures comprehensive version.** 11. ed. New York: Pearson, c2015. 1210 p. ISBN 9780134670942.
- Millennium Falcon** Disponível em: <<https://br.pinterest.com/pin/528680443769855416/>>. Acesso em: 29 maio. 2024.
- Pick holes in popular ship designs - Part Deux.** Disponível em: <<https://forums.spacebattles.com/threads/pick-holes-in-popular-ship-designs-part-deux.309248/page-544>>. Acesso em: 29 maio. 2024.
- Pin by Ivan Kaidan on token Star Wars tabletop | Star Wars rpg, Star Wars vehicles, Star Wars universe.** Disponível em: <<https://www.pinterest.com/pin/624874517024540081/>>. Acesso em: 29 maio. 2024.
- Star Jedi Font | dafont.com.** Disponível em: <<https://www.dafont.com/star-jedi.font>>. Acesso em: 31 maio. 2024.
- unusalsuspex - Hobbyist, Digital Artist | DeviantArt.** Disponível em: <<https://www.deviantart.com/unusalsuspex>>. Acesso em: 29 maio. 2024.