

CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA DE MINAS GERAIS

Luiza Andressa Vailante Silva  
Victória Fernanda Santos Rocha  
Victor Hugo Gonçalves da Luz Miranda

Laboratório de Linguagem e Técnicas de Programação II

### **TRABALHO PRÁTICO 3**

Contagem  
2023

Luiza Andressa Vailante Silva  
Victória Fernanda Santos Rocha  
Victor Hugo Gonçalves da Luz Miranda

## Relatório Trabalho Prático III Network Game

Relatório técnico apresentado para a disciplina de Laboratório de Linguagem de Programação, do curso de Informática do Centro Federal de Educação Tecnológica de Minas Gerais, unidade Contagem.

Professor Alisson Rodrigo dos Santos

Contagem  
2023

## RESUMO

Este relatório resume as principais características e funcionalidades do jogo Bomberman, desenvolvido com a biblioteca libGDX e programado em Java e tem como objetivos, aprender a lidar com conexões de rede e compreender a sincronização de aplicações *online* e *realtime*. O desenvolvimento do projeto foi dividido em etapas: período de pesquisas, desenvolvimento das aplicações e testagens periódicas. Por se tratar de um conteúdo consideravelmente complexo, foi necessário um estudo aprofundado para a ambientação dos desenvolvedores, para atender ao máximo as solicitações do trabalho prático. Sendo assim, por meio das pesquisas foram encontradas ferramentas fundamentais para execução do projeto, tais como a ferramenta de execução de códigos JavaScript, Node.js, e a biblioteca Socket.IO. Dessa forma, foi desenvolvido uma versão *multiplayer* do jogo Bomberman na qual é possível mais de um jogador ter acesso a estrutura do jogo, por meio de uma codificação Java inserida na rede, através do servidor, e acessada por mais de um cliente.

**Palavras-chaves:** *Multiplayer. Realtime. Online*

## SUMÁRIO

<b>1. INTRODUÇÃO.....</b>	<b>5</b>
<b>2. DESENVOLVIMENTO.....</b>	<b>5</b>
2.1 Node.js:.....	5
2.2 Socket.IO:.....	6
2.3 Testes:.....	6
Figura 1 - Servidor.....	7
Figura 2 - Clientes.....	7
Figura 3 - Jogo Multiplayer.....	7
2.4 Dificuldades:.....	8
<b>3. CONCLUSÃO.....</b>	<b>8</b>
<b>4. REFERÊNCIAS BIBLIOGRÁFICAS.....</b>	<b>9</b>

## 1. INTRODUÇÃO

Este trabalho prático consiste na implementação de uma funcionalidade de jogo em rede para o Trabalho Prático 2, que envolve a recriação do clássico jogo Bomberman, que em síntese é um tabuleiro labirinto, no qual o personagem principal, o “bomberman”, deve por meio da explosão de sua bomba, abrir caminhos e destruir os seus inimigos, com o objetivo de encontrar uma porta para prosseguir para a próxima fase.

O objetivo geral do trabalho é implementar a possibilidade de execução em rede, possibilitando a criação de um modo *multiplayer online*, que consiste na versão *offline* do jogo desenvolvido para uma versão *online*, proporcionando aos jogadores a opção de jogarem contra outros *players online*. Para o desenvolvimento, utilizamos a estrutura cliente-servidor que foi essencial na implementação deste projeto.

## 2. DESENVOLVIMENTO

Na estrutura cliente-servidor, o servidor atua como componente central que coordena a jogabilidade e armazena informações essenciais do jogo, como a disposição do mapa, o estado e as ações dos jogadores. Por outro lado, o cliente atua como jogadores individuais que se conectam ao servidor para conseguirem participar do jogo.

O servidor é o primeiro a ser inicializado, aguardando a conexão ser estabelecida com o cliente. Quando o jogo é inicializado, o servidor se conecta com o cliente, estabelecendo uma conexão. O servidor recebe as ações do cliente, como, movimentos do personagem, a posição dos powers e a colocação de bombas, processa essas ações e atualiza o estado do jogo. Logo em seguida, essas ações são enviadas para todos os clientes conectados, de modo que todos recebam as atualizações em tempo real. Toda essa interação é o que permite a jogabilidade de *multiplayer online*, onde os jogadores podem competir com outros *players* no ambiente virtual.

### 2.1 Node.js:

Para a realização de toda essa estrutura do projeto, utilizamos a plataforma de tempo de execução Node.js para desenvolver um servidor que foi utilizado como principal. Ele é responsável por gerenciar as conexões dos jogadores, coordenar a lógica do jogo, armazenar dados do jogo, como por exemplo, o estado atual do mapa e as posições dos jogadores, e atualizar os clientes com informações em tempo real.

Para sua utilização foi realizado um processo de instalação e configuração, no qual foi interligado a pasta do projeto.

Desta forma, o servidor gera um arquivo chamado `index.js` do tipo JavaScript, que é responsável por fazer a comunicação entre os arquivos java e o navegador.

## **2.2 Socket.IO:**

Socket.IO é uma biblioteca orientada a eventos para aplicativos da Web em tempo real. Ela permite a comunicação bidirecional em tempo real entre clientes e servidores da Web, consistindo em dois componentes: servidor e cliente.

Para a atualização em tempo real de todas as informações para todos os clientes, foi utilizada a biblioteca Socket.IO. A função dela no projeto é permitir que os jogadores interajam uns com os outros e com o servidor instantaneamente, além de contribuir para a aprimoração de aspectos do jogo, como segurança, jogabilidade e escalabilidade.

Para um bom funcionamento da biblioteca foi necessário uma implantação nas dependências da biblioteca `libgdx`, na qual foi possível utilizar de inúmeros recursos que possibilitam a ligação e comunicação entre cliente-servidor.

Para a conexão do Projeto estático anterior, a operação de cliente-servidor, foram criadas duas funções dentro da classe `Mapa()`, na qual é considerada a mais importante do projeto, pois engloba a criação do mapa e as chamadas de execução de todas as funções do jogo. Sendo extremamente importantes para aplicar o chamado aspecto do jogo *Multiplayer*, ao qual detecta a entrada de cada cliente novo que acessa o servidor, a `connectSocket()`, e a `config Socket Eventos()`.

## **2.3 Testes:**

Foram realizados vários testes durante o projeto para certificar-se de que a ordem cronológica do jogo estava funcionando. Primeiro, liga-se o servidor, e um cliente se conecta a cada hora que o jogo atualiza. Na Figura 1, "Servidor", ao rodar o código, o servidor é ligado e os clientes se conectam ao servidor. Na Figura 2, "Clientes", mostra-se a conexão dos clientes para ter acesso ao jogo e iniciar a partida. A Figura 3, "Jogo Multiplayer", mostra o acesso que os dois jogadores têm ao executar o código.

Figura 1 - Servidor

```
C:\Windows\System32\cmd.exe - node index.js
Microsoft Windows [versão 10.0.19045.3324]
(c) Microsoft Corporation. Todos os direitos reservados.

C:\BombermanINFO3-5-main\Trabalho\TP3-INFO3Bomberman\Server>node index.js
Servidor está rodando...
```

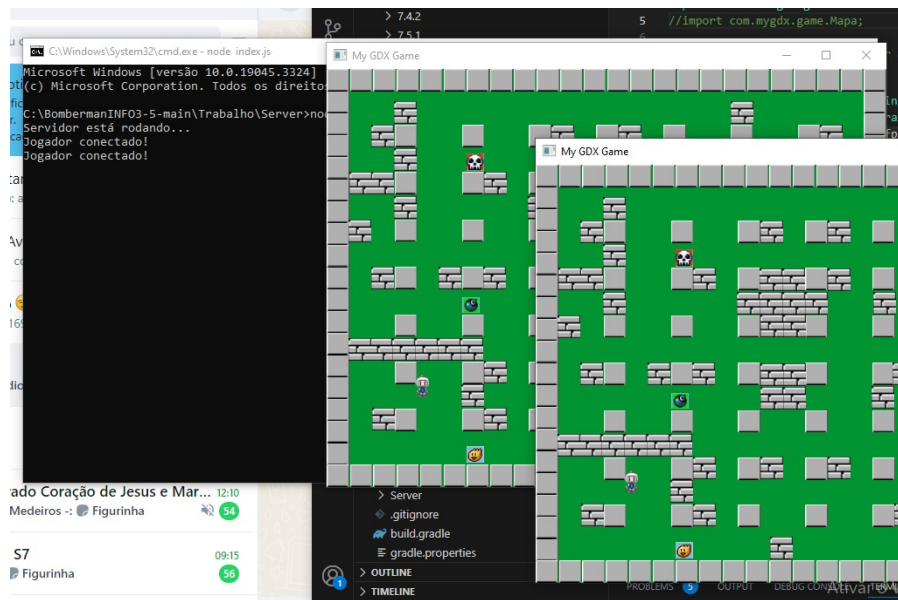
Fonte: Arquivo pessoal

Figura 2 - Clientes

```
[SocketIO] Connected
[SocketIO] My ID: ADHIWCIVmLusE_uYAAAF
```

Fonte: Arquivo pessoal

Figura 3 - Jogo Multiplayer



Fonte: Arquivo pessoal

## 2.4 Dificuldades:

Durante o desenvolvimento do projeto, tivemos dificuldades em compreender como utilizar a biblioteca Socket.IO para implementar o servidor e suas funcionalidades. Encontramos como referência apenas uma playlist que utilizava um servidor com a linguagem de programação JavaScript, que pudesse ajudar no desenvolvimento do jogo. No entanto, ao decorrer do projeto, realizando outras pesquisas e buscando materiais de apoio, conseguimos obter um resultado que ajudasse a desenvolver o jogo.

Com o projeto em andamento, tivemos alguns problemas em baixar algumas bibliotecas que eram essenciais para a implementação do resto do projeto, o que acabou acarretando um atraso na codificação, pois alguns computadores dos membros do grupo, não estavam aceitando a instalação.

## 3. CONCLUSÃO

Portanto, conclui-se que o trabalho possui alguns objetivos específicos alcançados, como o entendimento do funcionamento de conexões de rede, a compensação da sincronização de aplicações *online* em *realtime* e a tentativa de implantação do modo *multiplayer online* no jogo desenvolvido no Trabalho Prático 2, para a completa realização do Trabalho Prático 3.

Neste relatório, apresentamos a nossa experiência na tentativa de desenvolvimento completo do projeto utilizando a biblioteca libGDX, com suas dependências que possibilitam acesso a rede, além da utilização da biblioteca Socket.IO e o Node.js.

Consideramos que a realização parcial do trabalho foi valiosa, pois permitiu a aplicação dos conhecimentos adquiridos em sala de aula e o desenvolvimento de habilidades na criação de jogos utilizando a biblioteca libGDX e as conexões de rede.



#### 4. REFERÊNCIAS BIBLIOGRÁFICAS

VEN, R. Java File Transfer - socket IO tutorial. Disponível em: <<http://www.youtube.com/playlist?list=PLyt2v1LVXYS0UFQQH7z9StTV76OFbD22->>.

Acesso em: 3 set. 2023.

CODES, B. A. **Creating Multiplayer Games with LibGDX and NodeJS - Part 1.**

Disponível em: <<https://www.youtube.com/watch?v=uIPAAdslhPM>>. Acesso em: 3 set. 2023.

**Networking.** Disponível em: <<https://libgdx.com/wiki/networking>>. Acesso em: 3 set. 2023.

MAURÍCIO, P. R. **Tutorial de Desenvolvimento de Jogos Online Multiplayer Em Python.**

Disponível em:

<<https://awari.com.br/tutorial-de-desenvolvimento-de-jogos-online-multiplayer-em-python/>>.

Acesso em: 3 set. 2023.

FILHO, S. **Como criar jogos para Android: Criando um jogo multiplayer.** Disponível em:

<<https://www.devmedia.com.br/android-games-desenvolvendo-um-jogo-multiplayer-com-java-e-android-parte-1/29696>>. Acesso em: 3 set. 2023.

**Como Configurar o SERVIDOR APACHE TOMCAT em sua máquina para PROJETOS JAWEB.** , 7 set. 2017. . Acesso em: 3 set. 2023

**Saiba o que é Socket.IO, como instalar e usar em suas aplicações Node.js.** Disponível em:

<<https://www.arquivodecodigos.com.br/dicas/3689-node-js-saiba-o-que-e-socket-io-como-instalar-e-usar-em-suas-aplicacoes-node-js.html>>. Acesso em: 3 set. 2023.

**node.js, Error: Cannot find module “express”.** Disponível em:

<<https://stackoverflow.com/questions/14949118/node-js-error-cannot-find-module-express>>.

Acesso em: 3 set. 2023.

WIKIPEDIA CONTRIBUTORS. **Socket.IO.** Disponível em:

<<https://en.wikipedia.org/w/index.php?title=Socket.IO&oldid=1173333199>>.

