

## Documento de Visão do Projeto (DVP)

Disciplina: Sistemas distribuídos

Projeto: **Tank Front** - Jogo online de Tanque em tempo real

### 1. Introdução

#### 1.1. Propósito do documento

Este documento tem como propósito apresentar a visão geral, os objetivos e os detalhes técnicos do projeto **TankFront**, desenvolvido como trabalho final da disciplina de Sistemas Distribuídos.

O documento descreve o funcionamento do jogo, sua arquitetura, os componentes de software utilizados e a forma como os conceitos de sistemas distribuídos foram aplicados ao longo do desenvolvimento.

#### 1.2. Visão Geral do Projeto

O TankFront é um jogo online de tanques em tempo real inspirado na dinâmica dos jogos do gênero **battle royale**. Nele, vários jogadores entram simultaneamente em um mesmo campo de batalha e competem entre si até que apenas um tanque permaneça ativo. O mapa do jogo diminui progressivamente de tamanho à medida que a partida avança, forçando os jogadores a se aproximarem e se enfrentarem diretamente. Essa mecânica cria um ambiente de constante tensão e estratégia, em que cada movimento pode determinar a vitória ou a eliminação. Cada jogador controla um tanque capaz de se mover em diferentes direções e disparar projéteis, exigindo reflexos rápidos, precisão e tomada de decisão sob pressão. Assim como nos jogos battle royale, o posicionamento e o tempo de reação são fatores cruciais para a sobrevivência. O TankFront busca unir competitividade e sincronização em tempo real.

#### 1.3. Escopo do projeto

O objetivo do projeto é criar uma experiência competitiva, interativa e dinâmica, demonstrando na prática os conceitos de comunicação em sistemas distribuídos utilizando a tecnologia WebSocket. O sistema é dividido em duas camadas principais:

- **Cliente**

A interface do jogo é executada diretamente no navegador, sendo responsável por:

- Renderizar os tanques, tiros e zona de combate, controlando toda a parte visual e animada do jogo;
- Aplicar as regras locais, como o limite do mapa e a redução progressiva da zona segura;
- Capturar as ações do jogador (movimento e disparo) e enviá-las ao servidor via WebSocket;
- Atualizar a posição dos tanques inimigos com base nas mensagens recebidas do servidor.

O cliente é disponibilizado na rede local (LAN) por meio de um HTTP Server, permitindo que outros dispositivos conectados na mesma rede acessem o jogo diretamente pelo navegador.

- **Servidor**

O servidor, implementado em Node.js, é responsável por.

- Gerenciar as conexões WebSocket e identificar cada jogador conectado;
- Sincronizar o estado do jogo entre todos os clientes, garantindo consistência nas posições e disparos;
- Distribuir eventos de jogo (como novos tiros, movimentação e destruição de tanques) em tempo real para todos os jogadores;
- Manter a lógica central de comunicação, sem processar animações ou renderizações, que são executadas exclusivamente no cliente.

Dessa forma, o TankFront demonstra de forma prática a arquitetura de um sistema distribuído cliente-servidor, no qual o processamento é compartilhado: o servidor coordena e sincroniza, enquanto o cliente executa e exibe a experiência de jogo.

#### 1.4. Tecnologias Utilizadas

Camada	Tecnologia	Função
<b>Servidor</b>	Node.js	Plataforma para execução do servidor e gerenciamento das conexões WebSocket.
<b>Comunicação</b>	WebSocket (biblioteca ws)	Canal de comunicação bidirecional e em tempo real entre cliente e servidor.
<b>Hospedagem Local</b>	HTTP Server (módulo do Node.js)	Disponibiliza o cliente na rede LAN, permitindo o acesso ao jogo a partir de outros dispositivos via navegador.
<b>Cliente</b>	HTML5, CSS3, JavaScript ES6	Responsável pela renderização do jogo, controle do jogador e interação com o servidor.

<b>Renderização</b>	DOM e CSS	Representação visual dos tanques, tiros e da zona de combate.
<b>Controle de Estado</b>	Estruturas de objetos JavaScript	Armazenamento e atualização dos estados locais e remotos (posições, disparos e status dos jogadores).

## 2. Necessidades e Funcionalidades

Esta seção apresenta as principais necessidades do projeto e as funcionalidades desenvolvidas para atendê-las.

<b>Necessidade</b>	<b>Descrição</b>	<b>Funcionalidade Implementada</b>
<b>N1</b> - Comunicação em tempo real	Permitir que os jogadores troquem informações instantaneamente durante a partida.	Uso de WebSocket para enviar e receber dados entre o cliente e o servidor em tempo real.
<b>N2</b> - Sincronização entre jogadores	Garantir que todos vejam os tanques e tiros nas posições corretas.	O servidor Node.js coordena os dados e atualiza todos os clientes conectados.
<b>N3</b> - Acesso em rede local (LAN)	Possibilitar que o jogo funcione entre vários computadores na mesma rede.	Utilização de um HTTP Server para disponibilizar o jogo na LAN.
<b>N4</b> - Campo de batalha visível e dinâmico	Exibir os tanques, tiros e a zona que diminui com o tempo.	Implementação de renderização e animações no cliente (HTML, CSS e JavaScript).
<b>N5</b> - Controle do jogador	Permitir que cada jogador mova seu tanque e dispare tiros.	Captura de eventos de teclado e envio das ações ao servidor.
<b>N6</b> - Partida competitiva	Tornar o jogo divertido e estratégico até restar apenas um vencedor.	Zona de combate que reduz de tamanho para forçar o confronto entre os jogadores.