

Modelo Computacional Distribuído em Ambiente Multijogador Massivo Online de Tempo Real

Anderson S. Carapiá^{1,2}, Márcio R. B. Soussa^{1,2}, Leonardo S. A. da Silva¹

¹ Centro Universitário Jorge Amado (UNIJORGE)
41.680-400 – Salvador – BA – Brasil

² Centro Universitário SENAI CIMATEC
Salvador – BA – Brasil.

andy.carapia@gmail.com, marcio.soussa10@gmail.com, leoalm@gmail.com

Abstract. *The digital games universe has been gaining even more space and fans, and among various styles is worth noticing the importance of those which are multiplayer. In general terms, those games depend on complex server infrastructures and online availability across the planet to attend and distribute all of their players' information. This implies, beside the delay due to geographic distance itself between player and server, the dependence of availability. The objective of this work is to present an alternative to these problems, by using group communication techniques on distributed systems, bringing the game experience closer to the users, without losing the Multiplayer Massive Online (MMO) game aspect. In one of the tests carried out, using the prototype developed and presented in this work, it was possible to notice a reduction in the message exchange time, from approximately 262 ms (106 + 156 ms) to approximately 190 ms, representing a gain of around 27%.*

Resumo. *O universo de jogos digitais vem conquistando cada vez mais espaço e adeptos, e dentre os vários estilos destacam-se, principalmente, aqueles que são multijogador. De uma forma geral, estes jogos dependem de infraestruturas complexas de servidores online espalhados ao redor do planeta para atender e distribuir as informações de seus jogadores. Isso implica, além de atrasos decorrentes da própria distância geográfica entre jogador e servidor, na dependência de disponibilidade deste último. Este trabalho tem como objetivo apresentar uma alternativa para os problemas citados, através de técnicas de comunicação em grupo sobre sistemas distribuídos, trazendo a experiência de jogo mais próxima para os usuários, sem perder o aspecto Multijogador Massivo Online (MMO) do mesmo. Em um dos testes realizados, por meio do protótipo desenvolvido e apresentado neste trabalho, foi possível notar a redução no tempo de troca das mensagens, de aproximadamente 262 ms (106 + 156 ms) para aproximadamente 190 ms, um ganho em torno de 27%.*

1. Introdução

Os jogos digitais estão presentes no nosso cotidiano, tornando-se cada vez mais complexos e interativos. Newzoo (2020) apresentou uma projeção para 2020 com o total de jogadores ao redor do planeta ultrapassando 2,7 bilhões, um crescimento de aproximadamente 6% em relação ao ano anterior, enquanto Laurence (2024) estima que a indústria

dos jogos deve faturar mais de 213 bilhões de dólares no mundo até 2027, sendo mais de 9 bilhões apenas na América Latina em 2024.

No universo dos jogos digitais, uma categoria que se destaca e é objeto de estudo neste trabalho são os jogos *online*. Eles permitem que múltiplos jogadores interajam simultaneamente dentro de um ambiente virtual, especialmente nos jogos de cunho massivo, ou seja, que não definem limite na quantidade de jogadores. Needleman (2020) aborda sobre o aumento no número de jogadores em função da pandemia de COVID-19 que, por conta das medidas de segurança e do isolamento social, resultou em um crescimento econômico para as empresas desenvolvedoras de jogos digitais.

Entretanto, dois problemas comuns podem afetar jogos *online* massivos: aumento da latência e alta instabilidade na conexão, caracterizada pelo aumento na perda de pacotes e do *Jitter*. Latência, comumente indicada pelo RTT (*Round Trip Time*), pode ser compreendida como o intervalo de tempo entre o envio de um pacote na rede e o recebimento de uma resposta. Pacotes são enviados para transmitir informações sobre as ações dentro do jogo.

O problema da latência é frequentemente percebido em jogos do estilo RPG (*Role-Playing Game*), que se caracterizam por um mundo virtual persistente onde jogadores se conectam e interagem através de avatares, no intuito de cumprir missões e desafios, a exemplo de *World of Warcraft*¹.

Lebres et al. (2018) afirmam que uma das principais causas de evasão nos jogos *online* é de fato a latência elevada, característica que influencia negativamente na performance *in-game*. Um estudo empírico realizado por BarraDois (2019) informa que a latência média alcançada na conexão entre jogadores no Brasil e servidor localizado nos Estados Unidos variou de 130 ms a 400 ms, enquanto a latência entre jogadores nos Estados Unidos foi inferior a 30 ms. Este atraso é justificado, principalmente, pela infraestrutura de rede e a distância geográfica entre o computador do jogador e os servidores onde estão armazenados os jogos, e pode ser decisivo durante combates e ações de tempo real dentro do jogo.

No que tange à instabilidade da comunicação, ela pode ser ocasionada pela perda de pacotes e por grande variação no atraso (*Jitter*). A demanda pelo envio e recebimento de mensagens que devem ser gerenciadas pelo servidor cresce proporcionalmente à medida que novos jogadores ingressam no ambiente virtual. Desta forma, problemas de rede como enfileiramento e perda de pacotes são potencializados. Além disso, uma aplicação em tempo real demanda grande volume de pequenas mensagens, em sua maioria associadas a ações do usuário que, por este motivo, não podem ser previamente armazenadas, caso que não ocorre por exemplo em serviços de *streaming* de vídeo.

Portanto, percebe-se que a qualidade da conexão e a latência estão frequentemente associadas à infraestrutura de rede e à distância geográfica entre o servidor do jogo e o dispositivo do cliente. Por isso, entende-se a importância de projetos que busquem soluções para redução da latência, considerando a infraestrutura de rede existente e a distância geográfica entre servidores e jogadores.

A proposta deste trabalho é desenvolver um modelo computacional para

¹Disponível em: <https://www.worldofwarcraft.com>

comunicação em grupo, com o objetivo de reduzir a latência na troca de mensagens e pacotes de rede entre jogadores. Como resultado, foi possível implementar um protótipo funcional, que valida a decisão técnica escolhida, reduzindo o atraso na comunicação entre jogadores dentro de um ambiente controlado. Além disso, este trabalho se difere dos demais ao criar um ambiente virtual que permite a troca de mensagens entre jogadores, sem necessariamente depender de um servidor centralizado. As seções seguintes apresentam trabalhos correlatos que servem de base para este projeto, descrição do conceito de Grupo Lógico e resultados obtidos por meio de testes reais com jogadores em diferentes regiões do planeta. Ao final, são apresentadas as conclusões do trabalho e ideias para trabalhos futuros.

2. Trabalhos Correlatos

Abdulazeez and El Rhalibi (2019) apresentam uma solução para o problema de sobrecarga no servidor, por meio de uma arquitetura híbrida entre o modelo cliente-servidor tradicional e conexões P2P (*peer-to-peer*). Entretanto, seu foco está apenas na redução da carga no lado do servidor, uma vez que a estratégia adotada é redistribuir o custo computacional de microrregiões virtuais entre os dispositivos dos jogadores. Esta estratégia também é apresentada por Kim (2017), entretanto, não resolve diretamente o problema da distância geográfica.

Uma abordagem semelhante está presente no trabalho de Reis (2018), que, neste caso, utiliza uma arquitetura P2P como rede auxiliar para reduzir latência, sem, entretanto, reduzir a dependência de um servidor centralizador.

Distintamente dos trabalhos citados, este projeto tem por objetivo apresentar um modelo computacional híbrido e escalável para jogos massivos de tempo real, que reduza o atraso na troca de mensagem entre jogadores conectados ao sistema quando, em comparação ao servidor central, apresentarem menor latência entre si. E como prova de conceito, desenvolver um protótipo de jogo virtual que implemente o modelo proposto.

O modelo tradicional Cliente-Servidor, como López et al. (2016) apresentam em seu trabalho, é frequentemente utilizado em jogos *multiplayer* e possui arquitetura semelhante à apresentada na Figura 1 (lado esquerdo). Nela, é possível identificar um elemento centralizador denominado Servidor, que é responsável por estabelecer a troca de mensagens entre cada cliente. Neste trabalho, em busca de uma solução com menor latência, será proposta uma nova arquitetura, conforme detalhado na Seção 3.

Por fim, Amir et al. (2021) desenvolveram um *framework* de comunicação em grupo chamado *Spread*, uma ferramenta *Open Source* de alta performance para troca de mensagens com tolerância à falhas. Sendo uma solução de código aberto, é possível avaliar pontos de melhoria no código fonte, e realizar adaptações, sendo assim escolhida como uma das bases deste trabalho. Ela está disponível em várias linguagens de programação de forma gratuita ².

3. Materiais e Métodos

No modelo aqui proposto (Figura 1, lado direito) são adicionados novos elementos à arquitetura considerada tradicional, responsáveis por diminuir a dependência dos clientes em relação ao servidor, promovendo também redução na latência.

²Disponível em: <https://www.spread.org/download.html>

A arquitetura proposta possui três elementos principais: um computador **servidor** responsável pela recepção dos jogadores no mundo virtual, e persistência de suas informações em banco de dados; um ou mais dispositivos **clientes** que desejam jogar, sendo que cada cliente se conecta inicialmente ao servidor e aguarda até que um segundo dispositivo cliente esteja também disponível para trocar mensagens; e, por fim um *framework* de comunicação em grupo chamado *Spread*, desenvolvido por Amir et al. (2021). Ele é responsável por estabelecer a comunicação entre clientes de forma ordenada e confiável. Este conjunto formado por clientes que trocam entre si mensagens diretas será então chamado **Grupo Lógico**, conforme apresentado na Figura 1.

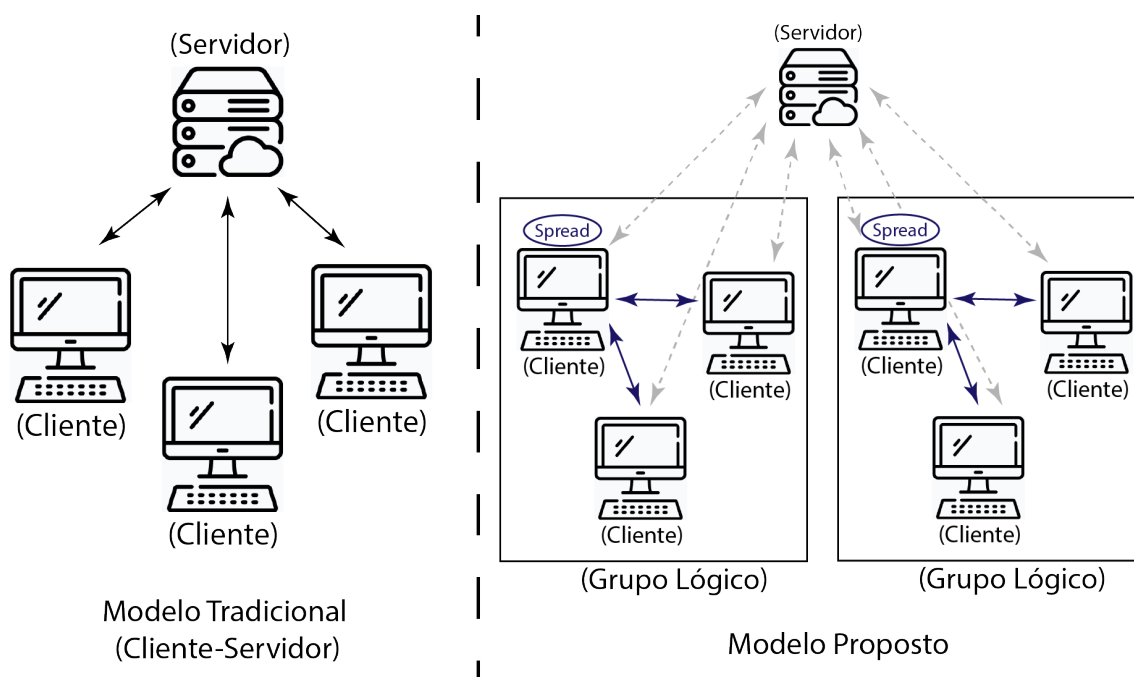


Figura 1. Diagrama comparativo entre a arquitetura tradicional e arquitetura proposta neste projeto

3.1. Comunicação em Grupo e Grupo Lógico

O **Grupo Lógico** é a estrutura fundamental deste modelo, responsável por definir o estado consistente de comunicação entre jogadores quando conectados entre si. De forma abstrata, cada grupo lógico armazena uma lista com o ID único de cada jogador conectado àquele grupo lógico. Através dessa lista, todos os jogadores passam a compartilhar a mesma noção sobre quem está *online* em seu grupo. Ao se conectar no servidor, cada novo jogador recebe seu respectivo ID único para que possa ingressar em algum grupo lógico. É papel de cada dispositivo solicitar ao servidor a relação dos jogadores atualmente conectados, e descobrir se existem divergências na sua percepção de estado do mundo virtual.

Um grupo lógico é formado por membros que se conectam à ele por meio de uma mensagem do tipo *Join*, composta por seu endereço IP e o nome do grupo do qual deseja participar. Além disso, um grupo lógico possui um espaço de memória compartilhada por todos os membros. Este espaço de memória é responsável por armazenar informações dos elementos instanciados na sessão e é de uso geral, podendo gerenciar quaisquer outras

informações relevantes ao jogo como, por exemplo, resultados de batalhas travadas por jogadores daquele grupo.

Outro objetivo na formação de um grupo lógico é, além da redução na latência, permitir que um jogador possa ser desconectado do servidor sem perder completamente a experiência *online*, tendo em vista que poderá continuar trocando mensagens com os jogadores do seu próprio grupo.

4. Resultados e Discussão

Com o objetivo de validar a arquitetura proposta, foi desenvolvido um protótipo fictício de jogo chamado "Mythus"³, que estabelece comunicação inicial entre jogadores por meio de um servidor localizado na região de Iowa, nos Estados Unidos. Na Figura 2, é possível identificar o jogador principal representado pelo avatar vermelho, sua latência em relação a outro jogador representado pelo avatar branco, e sua latência em relação ao servidor no canto superior esquerdo. A latência é calculada com base na diferença entre os momentos de envio e resposta de uma mensagem.



Figura 2. Jogo Mythus com o diagrama de latência nas localidades testadas

A partir do modelo apresentado, foram realizadas 4 sessões de teste no dia 17/11/2021, por volta das 15h (horário de Brasília) com jogadores voluntários em regiões distintas do planeta (2 entre os jogadores Bahia-Hamburgo, e 2 entre Bahia-São Paulo), com aproximadamente 100 mensagens trocadas em cada sessão. A Figura 3 ilustra a distribuição de latência identificada nos testes. Foram analisados os intervalos de tempo na troca de mensagens, calculados através da diferença de tempo percebida pelo sistema operacional no momento de envio e recebimento das mesmas, e possui valores cruzados com estatísticas de *sites* que apresentam dados históricos de latência, à exemplo de Reineimer et al. (2021). Vale ressaltar que as linhas contínuas indicam a direção de envio de mensagem, enquanto as linhas tracejadas representam o seu retorno.

³Disponível em: <https://github.com/andysampaio/MythusTCC>

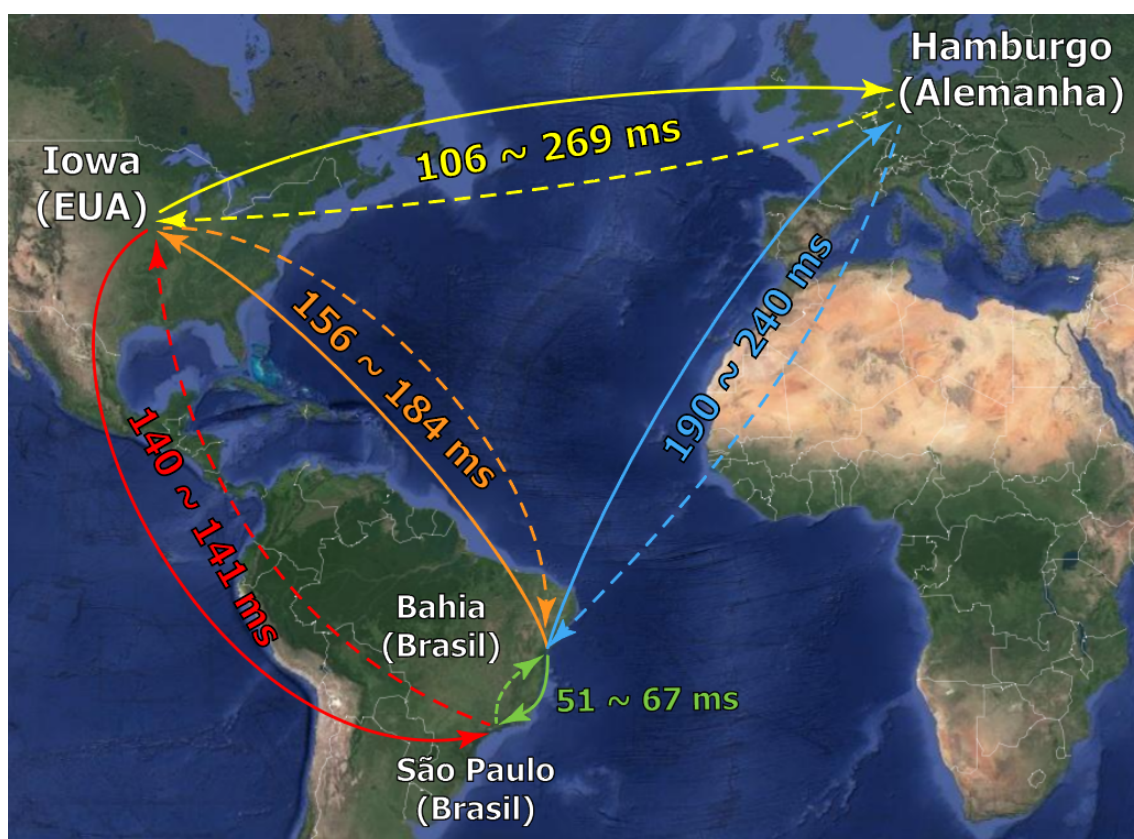


Figura 3. Diagrama de latência nas localidades testadas

O primeiro teste foi realizado com três jogadores, sendo um jogador localizado na Bahia (Brasil) e dois jogadores localizados em Hamburgo (Alemanha), enquanto o servidor do jogo estava localizado em Iowa (Estados Unidos). As linhas laranja (156 à 184 ms aproximadamente) e amarela (106 à 269 ms aproximadamente), na Figura 3 apresentam os valores da latência utilizando a arquitetura tradicional enquanto as linhas na cor azul (190 à 240 ms aproximadamente) apresentam a latência utilizando a arquitetura aqui proposta.

Nesse ponto, fica evidente a diferença de latência no modelo tradicional em relação à nova abordagem proposta. Mesmo no melhor caso do modelo tradicional, a troca de mensagens entre o jogador no Brasil e os jogadores na Alemanha sofreu um atraso de aproximadamente 262 ms (156 ms + 106 ms), uma vez que as mensagens são encaminhadas de um jogador para o outro através do servidor localizado nos Estados Unidos.

Já no modelo proposto que estabelece comunicação direta, representada pela rota azul, ainda que considerado o pior caso, há ganho de desempenho, com o atraso de aproximadamente 240 ms. Entretanto, analisando no melhor caso do modelo proposto, essa diferença se torna mais evidente, pois a conexão do Brasil para a Alemanha passando pelos Estados Unidos sofreu atraso superior à 450 ms, enquanto a conexão direta esteve abaixo de 200 ms.

O segundo teste foi realizado com dois jogadores, onde o primeiro estava também

localizado na Bahia (Brasil), e o segundo em São Paulo (Brasil). As linhas laranja (156 à 184 ms aproximadamente) e vermelha (140 à 141 ms aproximadamente), na Figura 3, apresentam a latência utilizando também a arquitetura tradicional, enquanto as linhas verdes (51 à 67 ms aproximadamente) mostram a latência obtida com a arquitetura proposta.

Desta forma, é possível notar que a diferença mais significativa nessa imagem está na conexão com jogadores geograficamente próximos. Analisando o melhor caso do modelo tradicional entre os jogadores da Bahia e São Paulo, a latência continua por volta de 300ms (156 ms + 140 ms). Já a conexão direta representada pela rota verde reduziu a latência em mais de 75%, estando na casa de 51 ms à 67 ms, aproximadamente.

5. Trabalhos Futuros

Quando um jogador ingressa no mundo virtual, ele se torna apto a formar grupos lógicos com outros jogadores que também estejam *online*. A formação de grupos neste trabalho acontece de forma manual, uma vez que este é um protótipo para validar o modelo. Como trabalho futuro, a formação dos grupos deve acontecer de forma automática, sem qualquer intervenção humana, seguindo 3 critérios de proximidade: Virtual, Social e Física.

Virtual: entende-se por proximidade virtual a distância entre avatares dentro do mundo virtual. Essa distância é importante por questões de otimização, uma vez que não é necessário receber o estado de avatares fora do campo de visão de cada jogador.

Social: é a distância estabelecida entre jogadores por meio de suas relações. Um jogador pode escolher adicionar um outro jogador à sua rede de contatos, tornando-o seu amigo. Assim, mesmo que estejam virtualmente separados poderão trocar mensagens.

Física: também referida como geográfica, é a distância física que cada dispositivo apresenta entre si, do ponto de vista da topologia de rede. Essa distância é a mais importante pois determinará o critério principal de formação dos grupos lógicos, uma vez que possui relação direta com o atraso na troca de mensagens entre os pares.

Uma vez identificada a necessidade de formar um grupo lógico, cada par inicia seu processo de descoberta e entrada em algum grupo por meio do *framework Spread*. Ele é composto por 2 módulos principais: um processo em segundo plano responsável pela comunicação em grupo, garantindo ordenação na entrega de mensagens e manutenção da visão consistente; e a implementação do cliente que se conecta nesse serviço e está embutido no jogo.

Assim como a formação automática de grupos, uma questão importante que por motivos técnicos não foi desenvolvida neste primeiro trabalho é a travessia de **NAT** (*Network Address Translation*). Atualmente, o cenário de redes mais comum é que cada usuário esteja conectado à internet por meio de um roteador doméstico que implementa NAT, um mecanismo que traduz IPs dos dispositivos de uma rede local para um único IP público. Neste cenário, nenhum dispositivo expõe seu endereço IPv4 real para o mundo, tornando impossível a conexão direta entre dispositivos de sub-redes distintas.

Este é um desafio importante para qualquer serviço do tipo P2P, uma vez que a solução convencional é manter um servidor físico denominado *Relay*, responsável por realizar a troca de informação dos seus respectivos IPs e portas reais, como Baset and Schulzrinne (2004) apresentam em sua análise do protocolo utilizado no serviço do *Skype*.

Portanto, será considerada a criação manual de grupos lógicos com dispositivos dentro de uma mesma sub-rede local ou por meio de VPN (*Virtual Private Network*). Neste trabalho, foi utilizada a VNP gratuita *ZeroTier*⁴.

Por fim, este trabalho também se limita à comparação entre a arquitetura híbrida aqui proposta, e a arquitetura tradicional Cliente-Servidor. Porém, é necessária ainda uma comparação com outras soluções híbridas.

6. Conclusão

A partir do estudo realizado no campo da computação distribuída, foi possível concretizar uma possível solução para resolver um dos problemas mais antigos em jogos *online*. O protótipo aqui apresentado consegue se comunicar com um servidor real, mas também realiza operações desconectadas dentro de seu grupo, por meio de mecanismos que garantem a troca ordenada de mensagens. Nos testes realizados estabelecendo conexão com um servidor localizado nos Estados Unidos, foi possível reduzir a latência de aproximadamente 300 ms para pouco menos de 60 ms, através da criação de grupos lógicos dentro de uma rede virtual.

Entretanto, essa não é a etapa final no desenvolvimento do jogo. Garantir a consistência e ordenação de mensagens permite que todas as ações básicas do jogador sejam mapeadas para um conjunto reduzido de mensagens, mas como citado anteriormente é uma etapa simples se comparada ao estabelecimento da própria comunicação em grupo.

Portanto, foi possível encontrar boas evidências de uma solução prática que torna viável a construção de jogos distribuídos, com características escaláveis, tolerante a falhas e que poderá trazer um desempenho geral melhor, se comparado ao modelo tradicional cliente-servidor.

Além disso, duas questões importantes para validar a eficiência deste projeto são a realização de testes envolvendo um conjunto maior de jogadores, uma vez que será possível analisar o impacto real da comunicação concorrente entre centenas de jogadores conectados, e a implementação de uma solução para o problema da travessia de NAT previamente citado.

Referências

- Abdulazeez, S. A. and El Rhalibi, A. (2019). *Dynamic Load Balancing for Massively Multiplayer Online Games Using OPNET*. Computer Science: Transactions on Education, XV. pp. 177-191. ISSN 1867-7207.
- Amir, Y., Amir-Miskin, M., Stanton, J., and Schultz, J. (2021). Spread toolkit. Acesso em 27/11/2021. Disponível em <http://www.spread.org/SpreadOverview.html>.
- BarraDois (2019). Portal de notícias sobre world of warcraft. Acesso em 18/04/2021. Disponível em <https://barradois.com/guia-latencia/>.
- Baset, S. A. and Schulzrinne, H. (2004). An analysis of the skype peer-to-peer internet telephony protocol. Acesso em 05/04/2021. Disponível em https://www.researchgate.net/publication/1957641_An_Analysis_of_the_Skype_Peer-to-Peer_Internet_Telephony_Protocol.

⁴Disponível em: <https://www.zerotier.com>

- Kim, Y. (2017). A study of optimal spatial partition size and field of view in massively multiplayer online game server. In *International Journal of Applied Engineering Research*. Research India Publications. Acesso em 18/04/2021. Disponível em https://www.ripublication.com/ijaer17/ijaerv12n19_24.pdf.
- Laurence, F. (2024). Faturamento do setor de jogos eletrônicos deve crescer 2,1. Acesso em 13/01/2025. Disponível em <https://bit.ly/globo-valor-faturamento-games-2024>.
- Lebres, I., Rita, P., Moro, S., and Ramos, P. (2018). Factors determining player drop-out in massive multiplayer online games. Acesso em 03/05/2021. Disponível em <https://www.sciencedirect.com/science/article/abs/pii/S1875952117300770>.
- López, C. A. M., Parra, O. J. S., and Torres, A. G. (2016). Networks and their traffic in multiplayer games. In *International Journal of Applied Engineering Research*. Revista Científica. Acesso em 18/04/2021. Disponível em https://www.researchgate.net/publication/311975618_Networks_and_their_traffic_in_multiplayer_games.
- Needleman, S. E. (2020). From 'fall guys' to 'among us,' how america turned to video-games under lockdown. Acesso em 10/05/2021. Disponível em <https://on.wsj.com/3lz43Eg>.
- Newzoo (2020). Global players per year. Acesso em 18/04/2021. Disponível em <https://newzoo.com/resources/blog/games-market-engagement-revenues-trends-2020-2023-gaming-report>.
- Reinheimer, P., Roberts, W., Moore, A., Gemma, A., Vihuri, M., and McKewan, L. (2021). Global ping statistics. Acesso em 27/11/2021. Disponível em <https://wondernetwork.com/pings>.
- Reis, R. H. (2018). *Diminuição de Latência em Jogos Multijogador Utilizando Conexões Auxiliares P2P*. 46 p. Monografia (Graduação em Ciência da Computação) - Universidade Federal do Rio Grande do Sul, Rio Grande do Sul. Acesso em 18/04/2021. Disponível em <https://www.lume.ufrgs.br/bitstream/handle/10183/175108/001065239.pdf>.