

Proposta de Arquitetura de Rede utilizando MQTT e WebSocket para aplicação em Gêmeo Digital com Minecraft

Thiago S. Prado¹, Lucas V. L. Assis¹, Higor F. Silva¹, Elizeu A. Jesus¹, Paulo S. R. Fraga¹, Jeohvana Vaz¹, Gustavo S. S. Souza², Gilberto L. Filho², Alisson A. Cardoso²

Resumo— Este trabalho apresenta uma rede de comunicação para a implementação de um Gêmeo Digital no Minecraft. Essa plataforma é proposta como uma alternativa acessível para a criação de Gêmeos Digitais e foi aplicada à modelagem do laboratório do grupo InComm. A arquitetura viabiliza a transferência bidirecional de dados entre dispositivos IoT e o Gêmeo, empregando os protocolos MQTT e WebSockets, além de uma interface que integra ambas as comunicações. Os resultados demonstram a viabilidade do intercâmbio de dados entre o Minecraft e dispositivos IoT, comprovando a eficácia da abordagem para a implementação de um Gêmeo Digital nesse ambiente.

Palavras-Chave— Gêmeos Digitais, Jogos Eletrônicos, Comunicação Bidirecional

Abstract— This paper presents a communication network for the implementation of a Digital Twin in Minecraft. This platform is proposed as an affordable alternative for the creation of Digital Twins and was applied to the modeling of the InComm group laboratory. The architecture enables the bidirectional transfer of data between IoT devices and the Twin, using the MQTT and WebSockets protocols, in addition to an interface that integrates both communications. The results demonstrate the feasibility of data exchange between Minecraft and IoT devices, proving the effectiveness of the approach for the implementation of a Digital Twin in this environment.

Keywords— Digital Twins, Electronic Games, Two-Way Communication

I. INTRODUÇÃO

Conforme *National Aeronautics and Space Administration* (NASA) no ano de 2012, gêmeos digitais são simulações virtuais que refletem, temporalmente, o estado de suas contrapartes físicas, tendo como base dados coletados em tempo real, dados históricos e modelagem física [1]. O conceito foi inicialmente sugerido por Michael Grieves em 2003, propondo que o gêmeo digital deveria incluir três partes: o produto físico, o produto virtual e as conexões dentre ambas as partes. Posteriormente, Fei Tao sugeriu cinco etapas de modelagem: as modelagens física, virtual, de conexão, de dados e de serviços.

- Modelagem física - envolve a seleção das informações da entidade a serem lidas, tendo em vista o objetivo do projeto;

¹, Escola de Engenharia Elétrica, Mecânica e de Computação - EMC, Universidade Federal de Goiás - UFG, CEP 74.605-010, Goiânia-GO.

², Centro de Excelência em Redes Inteligentes Sem Fio e Serviços Avançados (CERISE), EMC, UFG, CEP 74.605-010, Goiânia-GO.

- Modelagem virtual - aborda a elaboração de uma representação virtual que possa exibir os mesmos comportamentos da entidade física dado uma certa entrada;
- Modelagem de conexão - a criação de uma rede de comunicação que conecte ambas as partes física e virtual do projeto;
- Modelagem de dados - engloba a definição, os procedimentos de operação e o armazenamento dos dados trafegados entre as partes;
- Modelagem de serviços - trata sobre a aplicabilidade prática do projeto final.

Tal tecnologia possui uma ampla gama de aplicações [1], tendo sido consideradas ao menos catorze possíveis aplicações na indústria, em áreas como design de produtos, planejamento de linhas produtivas, interação homem-máquina, etc. Além disso, destacam-se aplicações no monitoramento de estruturas diversas. Por exemplo, [2] explorou essa tecnologia para acompanhar, em tempo real, as condições de um edifício físico, integrando os dados coletados a um modelo de *Building Information Modeling* (BIM).

Considerando a aplicabilidade da tecnologia de gêmeos digitais para o monitoramento em tempo real de entidades diversas, este projeto visa explorar o uso do jogo *Minecraft* como ambiente virtual para representar digitalmente uma sala real. A sala escolhida é a sala do grupo InComm, um grupo de pesquisa com foco em novas tecnologias para redes inteligentes e sistemas de comunicação; tal sala é sediada na Escola de Engenharia Elétrica, Mecânica e de Computação (EMC), uma unidade acadêmica da Universidade Federal de Goiás (UFG).

Baseando-se na proposta de Fei Tao [1], o presente trabalho discorrerá majoritariamente sobre a modelagem de conexão do gêmeo digital proposto. Na seção posterior será apresentada a arquitetura de rede utilizada para comunicação, interrelacionando o mundo real com a contraparte virtual do gêmeo digital proposto, de forma a criar um ambiente responsivo da sala designada. Por fim, são apresentados alguns exemplos práticos e perspectivas futuras para o projeto.

II. PROPOSTA DE REDE PARA COMUNICAÇÃO

Tendo em vista a modelagem do espaço virtual e a implementação dos dispositivos IoT para coleta dos dados do mundo físico, é necessário a existência de uma rede que realize o transporte dos dados entre os dispositivos físicos e suas

contrapartes digitais, tanto no sentido objeto-gêmeo quanto no sentido gêmeo-objeto. Além disso, é necessária definição dos tipos e protocolos além dos procedimentos a serem executados com os dados trafegados. Alguns aspectos desejáveis para o projeto são: escalabilidade, consistência no transporte da informação e segurança contra possíveis ataques externos. Ao longo desta seção será discutida a arquitetura de rede utilizada para a realização do projeto.

A. Protocolos MQTT e WebSocket

Para realizar a comunicação dos dados entre os dispositivos IoT, foi utilizado o protocolo MQTT (*Message Queuing Telemetry Transport*) [3], um protocolo amplamente utilizado em aplicações IoT. A escolha do MQTT se dá pela sua eficiência, voltada para cenários com mínima conectividade e poder de processamento limitados. Adicionalmente, sua escalabilidade, que possibilita fazer uma rede com até milhões de dispositivos conectados, e sua segurança, devido a facilidade na implementação de métodos de criptografia e autenticação, são importantes para a modelagem de dados e de serviços sem deixar de lado a proteção da rede contra ataques externos.

Para receber as informações no *Minecraft* optou-se pelo uso do *CC:Tweaked* [4], um *mod* (modificação do jogo feita por terceiros que adiciona, altera ou remove elementos do jogo) do *Minecraft* que oferece aos jogadores maneiras de interagir com as mecânicas do jogo através da programação, permitindo a criação de sistemas e automações programáveis em linguagem Lua. A escolha do *mod* se deve à sua capacidade de interagir com sistemas fora do jogo através do protocolo de redes *WebSocket*.

O protocolo MQTT utiliza um servidor central denominado *Broker*, responsável por redistribuir as mensagens publicadas em cada tópico a todos os clientes inscritos, de modo que cada tópico funcione como um canal de controle de um dispositivo. Em função de sua simplicidade, o *Broker* utilizado é o *Mosquitto* [5].

B. STEVE: Ponte de Comunicação

Em razão do *mod CC:Tweaked* não ser capaz de realizar comunicações MQTT nativamente, foi criada a ponte *Server-to-Things Event-driven Virtual Engine* (STEVE), sendo ela um software escrito em *Python* que atua como intermediário entre o *Broker* MQTT e os computadores no *Minecraft*. Explicando melhor, esse componente foi criado para receber, via MQTT, as mensagens do *Broker* e repassá-las via *WebSocket* ao jogo, e vice-versa. Para o desenvolvimento da ponte STEVE foram utilizadas as bibliotecas *paho-mqtt* [6] e *websockets* [7], na linguagem *Python*. Os passos do funcionamento da ponte STEVE são representados na Figura 1 e serão descritos em maiores detalhes na sequência.

Ao ser iniciada, a ponte STEVE conecta-se como um cliente ao *Broker* MQTT, inscreve-se em todos os tópicos e, ao mesmo tempo, hospeda um servidor *WebSocket* e aguarda para conectar aos computadores do *mod*. Em seguida, quando um computador do *mod* faz conexão com a ponte, é protocolado que sua primeira mensagem deve ser uma identificação, no formato JSON, com as seguintes informações: tipo de conexão

(consumidor ou produtor) e o tópico ao qual se refere. Dessa forma, os atores estão conectados e a comunicação pode ocorrer.

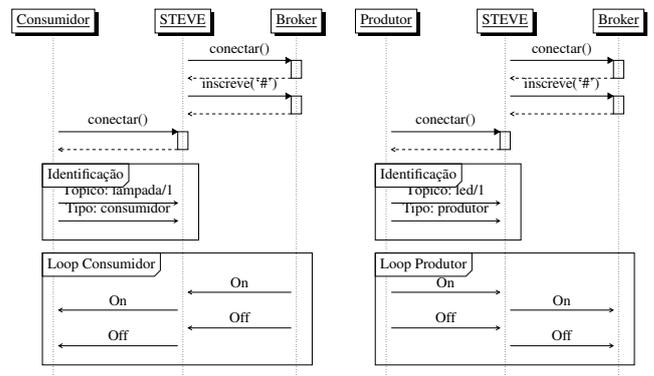


Fig. 1: Funcionamento da ponte STEVE com um consumidor e um produtor

No sentido objeto-gêmeo, quando um sensor detectar alguma alteração, ela é enviada ao *Broker* pelo tópico correspondente ao mesmo sensor. Então, logo em seguida, é levada até a STEVE, pois a ponte está inscrita em todos os tópicos. Ao receber a mensagem, STEVE verifica se algum de seus consumidores está inscrito no tópico, se sim, repassa via *WebSocket* o conteúdo da mensagem MQTT ao consumidor em questão; caso contrário, a mensagem é descartada.

No sentido gêmeo-objeto, um computador do *CC: Tweaked*, conectado como um produtor, envia a mensagem via *WebSocket* à ponte STEVE. Então, a ponte verifica o tópico do produtor e publica o conteúdo, no tópico verificado, no *Broker*. Dessa forma, a mensagem chega ao microcontrolador inscrito no tópico, o qual executará a resposta correspondente.

No intuito de conectar as informações do mundo real com o *Minecraft*, buscando escalabilidade, foi proposta a utilização de uma arquitetura de rede composta por: sensores e atuadores no mundo físico, um *Broker* MQTT, computadores do *mod CC:Tweaked* no *Minecraft* e a ponte de comunicação STEVE. A descrição do processo pode ser visualizada na Figura 2.

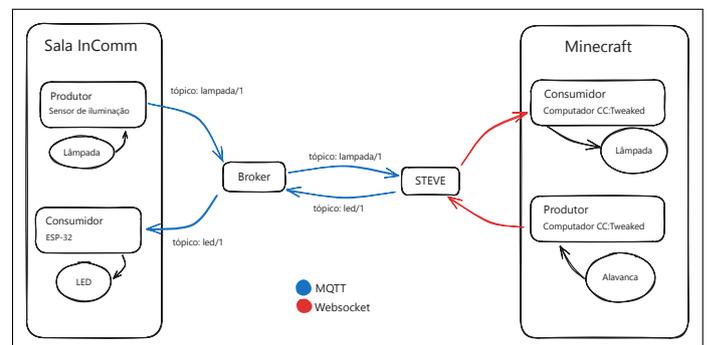


Fig. 2: Topografia de Rede do sistema implementado

III. RESULTADOS

Esta Seção apresenta os resultados obtidos deste trabalho. Inicialmente, foi realizada a modelagem do laboratório Incomm no ambiente do *Minecraft*. Foram representados elementos estruturais e equipamentos computacionais, além do

estado de lâmpadas e portas, que são sincronizados com o ambiente externo. Esses dispositivos podem ser acionados tanto pelo Gêmeo Digital quanto pelo próprio ambiente externo, garantindo uma interação bidirecional entre os dois sistemas. A representação física da sala é retratada nas Figuras 3a e 3c; e a representação virtual da sala é retratada nas Figuras 3b e 3d.



(a) Sala do InComm



(b) Sala do InComm - Minecraft



(c) Sala de Reuniões



(d) Sala de Reuniões - Minecraft

Fig. 3: Imagens do gêmeo digital da sala no Minecraft

Para comunicação gêmeo-objeto, é possível transmitir dados a partir de um computador do mod *CC:Tweaked* e recebê-los em um microcontrolador. Para isso, é necessário acoplar os atuadores adequados, permitindo a execução da resposta conforme o sinal recebido. Essa abordagem viabiliza uma integração eficiente entre o ambiente virtual e o mundo real, possibilitando a automação e o controle remoto de dispositivos físicos.

IV. CONSIDERAÇÕES FINAIS

O presente trabalho estudou a viabilidade de implementação do jogo *Minecraft* como plataforma para um gêmeo digital funcional, desbravando o seu ambiente intuitivo, customizável e sustentado por uma comunidade ativa. Foram efetuadas demonstrações práticas que comprovam a eficácia da rede proposta. No sentido objeto-gêmeo, considerando o sensor de luminosidade e a rede de comunicação estabelecida, houve resposta da luz ambiente da sala do InComm no gêmeo digital, estando uma lâmpada acesa no ambiente do jogo em sincronia com o cenário real.

O reconhecimento da abertura e fechamento de uma das portas no ambiente real foi bem-sucedido, garantindo que as portas correspondentes no Gêmeo Digital fossem acionadas em sincronia. A atualização do estado ocorreu conforme os dados recebidos do sensor, assegurando uma interação precisa entre o ambiente físico e o digital.

As aplicações futuras visam ampliar o ambiente replicado e integrar visão computacional para detectar indivíduos, portas e janelas; combinados à comunicação bidirecional, essas estratégias elevarão a precisão e a funcionalidade.

REFERÊNCIAS

- [1] F. Tao, H. Zhang, A. Liu, and A. Y. C. Nee, "Digital twin in industry: State-of-the-art." *IEEE Transactions on Industrial Informatics*, vol. 15, no. 4, pp. 2405–2415, 2019.
- [2] G. A. d. S. Galvão, H. G. d. T. Castro, B. R. d. B. Costa, G. Miceli Junior, and P. C. Pellanda, "Aplicação de gêmeos digitais em um ambiente bim de monitoramento de estruturas de edificações." *SIMPÓSIO BRASILEIRO DE TECNOLOGIA DA INFORMAÇÃO E COMUNICAÇÃO NA CONSTRUÇÃO*, vol. 4, p. 1–11, out. 2023.
- [3] A. Banks, E. Briggs, K. Bordengale, and R. Gupta, "Mqtt version 5.0 - oasis standard." Disponível em: <https://mqtt.org/mqtt-specification/>, 2019. Acesso em: 2 mar. 2025.
- [4] J. Coates, "Cc:tweaked." Disponível em: <https://tweaked.cc/>, 2025. Acesso em: 1 mar. 2025.
- [5] R. Light and Eclipse Foundation, "Mosquitto broker documentation." Disponível em: <https://mosquitto.org/documentation/>, 2025. Acesso em: 1 mar. 2025.
- [6] Eclipse Foundation, "Paho-mqtt documentation." Disponível em: <https://eclipse.dev/paho/files/paho.mqtt.python/html/index.html>, 2024. Acesso em: 4 mar. 2025.
- [7] c. Aymeric Augustin, "Websockets documentation." Disponível em: <https://websockets.readthedocs.io/en/stable/>, 2025. Acesso em: 4 mar. 2025.
- [8] M. Grieves, *Virtually Intelligent Product Systems: Digital and Physical Twins*, pp. 175–200. American Institute of Aeronautics and Astronautics, 07 2019.
- [9] Espressif Systems, "Esp32 series datasheet." Disponível em: https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf, 2025. Acesso em: 11 fev. 2025.
- [10] A. S. Sedra and K. C. Smith, *Circuitos Microeletrônicos*. LTC, 8th ed., 2023.
- [11] Y. Wu, K. Zhang, and Y. Zhang, "Digital twin networks: A survey," *IEEE Internet of Things Journal*, vol. 8, no. 18, pp. 13789–13804, 2021.
- [12] C. Lesjak, D. Hein, M. Hofmann, M. Maritsch, A. Aldrian, P. Priller, T. Ebner, T. Rupprechter, and G. Pregartner, "Securing smart maintenance services: Hardware-security and tls for mqtt," in *2015 IEEE 13th International Conference on Industrial Informatics (INDIN)*, pp. 1243–1250, 2015.
- [13] Oracle, "Transport layer security (tls) protocol overview." Disponível em: <https://docs.oracle.com/javase/8/docs/technotes/guides/security/jsse/tls.html>, 2025. Acesso em: 4 mar. 2025.