

Digital Twins: controle dos movimentos de um braço robótico utilizando MQTT e NGSI

Digital twins: controlling the movements of a robotic arms using MQTT and NGSI

Engenharia de Computação

Fábio Henrique Cabrini (fabio.cabrini@usp.br)

Doutor em Engenharia Elétrica pela Universidade de São Paulo (USP) e professor da Faculdade Engenheiro Salvador Arena.

Ysabela Akiyama Molero Rodrigues (ysabelamoleror@gmail.com)

Graduanda em Engenharia de Computação pela Faculdade Engenheiro Salvador Arena.

Caio Vinicius Magro (caio.magro21@gmail.com)

Graduando em Engenharia de Computação pela Faculdade Engenheiro Salvador Arena.

Matheus de Novais Sousa (math.novais@outlook.com)

Graduando em Engenharia de Computação pela Faculdade Engenheiro Salvador Arena.

Giovana Moreira da Silva (giovanamoreirasilva@outlook.com)

Graduanda em Engenharia de Computação pela Faculdade Engenheiro Salvador Arena.

FTT Journal of Engineering and Business

- SÃO BERNARDO DO CAMPO, SP JUN. 2024
- ISSN 2525-8729

Submissão: 14 dez.. 2023 Aceitação: 19 abr.2024

Sistema de avaliação: às cegas dupla (double blind review)

FACULDADE ENGENHEIRO SALVADOR ARENA, p.41 - 62

FTT JOURNAL
of Engineering and Business



Resumo

Este trabalho tem como objetivo desenvolver uma Prova de Conceito, do inglês *Proof of Concept* (PoC), envolvendo o conceito de *Digital Twin* (DT) no contexto de indústria inteligente que estabeleça a comunicação bidirecional entre um braço mecânico IoT (do inglês, *Internet of Things*) e um modelo computacional tridimensional intermediado por um *broker* de contexto baseado em padrões abertos. O sistema possibilita que o braço mecânico e seu gêmeo digital realizem todas as manobras previstas no projeto do braço mecânico. O DT é uma ferramenta de simulação que usa modelos matemáticos avançados para criar uma réplica virtual idêntica à do ambiente físico. Este trabalho apresenta a construção de um protótipo de gêmeo digital e sua modelagem, integrada às tecnologias de processamento de informações de contexto e IoT, visando a aplicação nos cenários industriais para potencializar melhorias no processo produtivo. O estudo foi conduzido com base no FIWARE para o desenvolvimento da PoC, cujos experimentos práticos realizados em laboratório apresentam o fluxo de comunicação bidirecional entre o robô e seu gêmeo digital, demonstrando a viabilidade da solução para aplicações didáticas.

Palavras-chave: *Digital Twin*. Fiware. Indústria inteligente. *Internet of Things*.

Abstract

This paper aims to develop a Proof of Concept (PoC) involving the concept of Digital Twin (DT) in the context of Smart Industry, capable of establishing bidirectional communication between a Robot Arm and a 3D computational model mediated by an open standard based context broker. The system enables the mechanical arm and its digital twin to perform all maneuvers envisaged in the mechanical arm's design. DT is a simulation tool that uses advanced mathematical models to create a virtual replica identical to the physical environment. This article shows the construction of a digital twin and its modeling, integrated with context information processing and IoT technologies, aiming its application in industrial scenarios to enhance improvements in the production process. The study was conducted based on FIWARE for the development of the PoC, which practical laboratory experiments presented the bidirectional communication between the robot and its digital twin, demonstrating the efficiency of the proposed solution for didactic application.

Keywords: Digital Twin. Fiware. Smart industry. Internet of Things.

Introdução

Embora as Indústrias Inteligentes tenham trazido muitos benefícios e oportunidades para as empresas, ainda existem alguns desafios a serem enfrentados, como a falta de previsibilidade da produção, alto custo para manutenção e realização de melhorias, falta de padronização e dificuldade para integrar diferentes tecnologias. A ausência de um modelo para realizar simulações que não coloquem em risco a produção ou os equipamentos dificulta o gerenciamento, a execução de testes e as inovações no âmbito industrial (Bajic, *et al.*, 2020).

Como solução, esta pesquisa propôs o desenvolvimento de uma PoC envolvendo o conceito de um gêmeo digital para aplicações didáticas, simplificando a introdução no campo de Pesquisa e Desenvolvimento (P&D) para esta tecnologia.

Os *digital twins* são modelos virtuais que reproduzem ambientes reais, e através da coleta de informações de contexto, permitem realizar simulações para prever o desempenho do mundo físico em diferentes cenários, englobando todo o ciclo de vida do processo (Bajic, *et al.*, 2020).

Nessa circunstância, este projeto apresentou o desenvolvimento de um protótipo que implementa o DT de um braço robótico, visando o contexto de indústrias inteligentes, para ser exercitado como um objeto de estudo, demonstrando dessa forma que o conceito aplicado em larga escala permite coletar métricas do ambiente produtivo real para acelerar a tomada de decisões nas indústrias. Essa técnica oferece a oportunidade de antecipar riscos que possam surgir na operação, permitindo que as empresas tomem medidas proativas sem desperdiçar seus insumos (Singh, *et al.*, 2021).

A criação do DT e sua modelagem foram realizadas através da integração de tecnologias de processamento de informações de contexto e a IoT para a prática em ambiente acadêmico. Para isso, foi necessária a utilização de um braço robótico equipado com servomotores que transmitem a informação de seus estados através de sensores internos, permitindo a reprodução do seu movimento pelo modelo virtual (Beek, Karkaria, Chen, 2023).

A avaliação dos resultados foi realizada a partir da transmissão bidirecional dos dados de movimento dos motores do braço robótico, através da criação de um modelo que implementa elementos que atuam como gerador de contexto baseado nos padrões abertos oferecidos pela

plataforma FIWARE e permitiram reproduzir os movimentos do robô virtual modelado com Unity, apresentando os conceitos estudados no artigo e demonstrados em trabalhos no estado da arte, como por exemplo, a perspectiva apontada por Rui Zhang na publicação “Digital twin and its applications: A survey” pelo periódico Springer Nature. O DT tem o potencial para melhorar a eficiência e a produtividade do negócio, ao mesmo tempo em que promove benefícios à empresa e ao meio ambiente em que está inserida (Zhang, *et al.*, 2022).

A justificativa para o estudo dessa solução tem base nos objetivos propostos pela Agenda 2030 da Organização das Nações Unidas (ONU), através do Objetivo de Desenvolvimento Sustentável (ODS) número 4, que se refere à Educação de Qualidade. A implementação da PoC como MVP (*Minimum Viable Product*), ou Produto Mínimo Viável em português, nas empresas e instituições de ensino permite iniciar departamentos de Pesquisa e Desenvolvimento (P&D) que aperfeiçoem o tema internamente e promovam o desenvolvimento dos processos produtivos e tomada de decisões, auxiliando na formação de estudantes dos cursos de Engenharia (Nações Unidas Brasil, 2023).

Referencial teórico

Essa seção tem como objetivo apresentar alguns dos trabalhos, conceitos e ideias disponíveis na literatura específica, no estado da arte, que possibilitaram embasar o modelo tratado neste artigo, assim como uma base bibliográfica para validação das metodologias trabalhadas.

Nesse sentido, foram abordados o conceito de *digital twin* e sua modelagem, as tecnologias que permitem sua implementação, como o FIWARE, com foco de aplicação no cenário de Indústrias Inteligentes, o que é possível através da utilização de dispositivos que utilizam IoT para envio de informações de contexto em tempo de execução.

Digital twin

O DT é uma réplica virtual de um objeto ou ambiente físico que é utilizado para simular o comportamento e o desempenho de um determinado sistema em diferentes cenários. Este modelo digital pode promover uma redução do tempo de execução recebendo dados dos itens que o compõem, mantendo-se assim consistente com as entidades no ciclo de vida (Beek, Karkaria, Chen, 2023).

O gêmeo digital é um conjunto de construções de informação virtual que imita a estrutura, contexto e comportamento de um ativo físico único, ou um grupo de ativos físicos, e é atualizado dinamicamente com dados de seu gêmeo físico ao longo de seu ciclo de vida, transmitindo decisões que geram valor. Sua implementação ocorre com o suporte de tecnologias integradas que possibilitam desenvolver o modelo do DT e tratar seus dados para que sejam consumidos e analisados de acordo com seu objetivo de aplicação (Schroeder, *et al.*, 2016).

Conforme apresentado anteriormente, uma das principais aplicações do DT é no cenário de indústrias inteligentes em que ele pode ser modelado para o planejamento da produção, através de simulações que preveem o comportamento dos dispositivos de acordo com a projeção do teste, trazendo como resultado a quantidade de produtos fabricados, o tempo de vida das máquinas, alertas gerados sobre possíveis defeitos, orientações de quando devem ser feitas manutenções preventivas e até mesmo fornecer suporte de logística, monitorando o processo de transporte dos produtos e sugerindo melhores rotas de entrega (Tao, *et al.*, 2019).

Alguns exemplos na atualidade demonstram sua aplicabilidade e retorno de mercado para grandes empresas, como por exemplo, a SpaceX com a utilização de *digital twins* para mapear as trajetórias de lançamento de seus foguetes e satélites. A cada lançamento, centenas de dados são coletados de todos os sensores presentes no veículo de lançamento, para que métricas sejam geradas sobre as correções necessárias para manutenção das peças e até mesmo acelerar a tomada de decisão em tempo de execução. Além disso, sua utilização permite que as trajetórias de lançamento sejam exploradas em diferentes contextos, realizando simulações precisas sem a necessidade de relançar fisicamente o foguete, como foi apresentado no estudo de caso realizado pela Siemens, cujo resultado apontou um aumento de 50% na produtividade com a aplicação de plataformas digitais (Carlos, 2021).

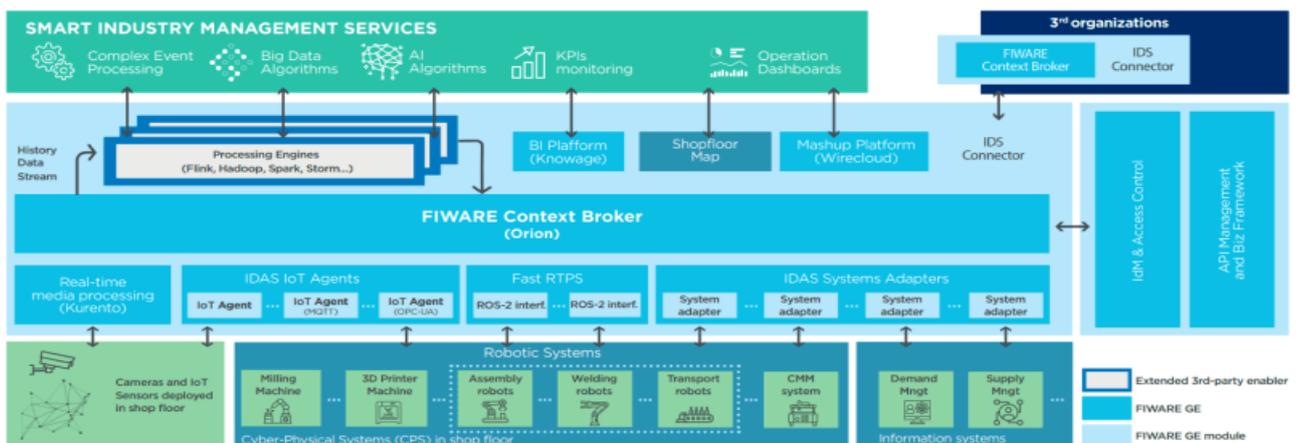
Outra inovação que utiliza DT para seu desenvolvimento é a iniciativa apresentada pela NVIDIA que utiliza *digital twins* para o planeta Terra, através do EARTH-2, para controle de mudanças climáticas, buscando acelerar o desenvolvimento de estratégias para controle dos efeitos causados à natureza e realizar projeções dos recursos disponíveis.

Fiware

O FIWARE é um *framework* composto por componentes modulares *open-source* cujos *Generic Enablers* (GEs) facilitam o desenvolvimento e a implementação de soluções inteligentes. Ele pode atuar como um *middleware* no gerenciamento dos dados em um DT e permitir o controle em tempo de execução (*runtime*) dos dados de contexto. Ele é fortemente escalável, porém sua implantação é altamente complexa, permitindo uma abordagem avançada de *Smart Data Models* (SDM) e interoperabilidade através do padrão NGSI (Next Generation Service Interfaces) (Gil, *et al.*, 2016).

A Figura 1 apresenta a arquitetura de referência utilizada pela FIWARE para aplicação em *Smart Industry* (Indústria Inteligente, em português), em que é possível observar em sua base a estrutura que recebe os dados dos dispositivos IoT e se comunica, de forma bidirecional, com os agentes que farão a conversão e o envio dos dados recebidos ao *broker* de contexto, Orion Context Broker, onde cada SDM irá realizar o consumo dos dados de acordo com sua aplicação, podendo alimentar também um DT, conforme evidenciado na arquitetura, motivo este que incentivou sua utilização como elemento central da solução, já que esta predisposição do Orion para aplicação em *digital twins* permite uma abordagem mais robusta na manipulação e gerenciamento dos dados que serão processados pelo *broker* (Bauer, *et al.*, 2021).

Figura 1 – Arquitetura de referência para solução de Indústria Inteligente estruturada pela FIWARE



Fonte: FIWARE (2022)

Next Generation Service Interface

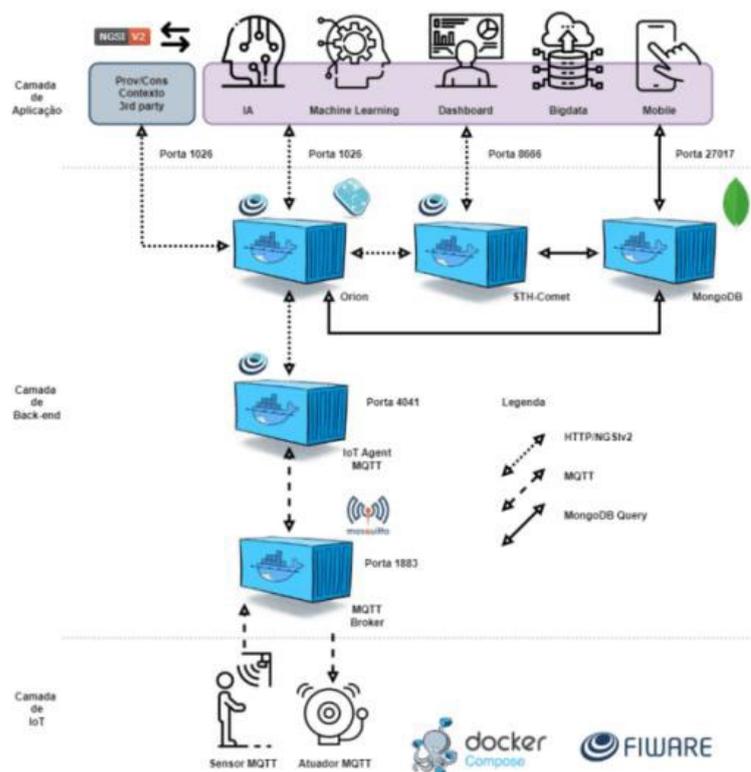
O padrão europeu de interface NGSI estabelece um modelo de dados consistente e uma API para troca de informação de contexto na plataforma FIWARE e outras aplicações, que podem ser utilizadas para a modelagem de DT em entidades, por exemplo, carros, pessoas, edifícios, entre outros. Os dados NGSI podem ser armazenados e disponibilizados utilizando *brokers* de contexto que implementam sua API, e têm como objetivo acessar informações de um DT específico, além de buscar ou descobrir novos DT disponíveis no ambiente (Conde et al., 2022).

O Fiware descomplicado

Outra tecnologia que deu apoio a este artigo foi o FIWARE Descomplicado, ferramenta que facilita o processo de instanciação dos GEs oferecidos pela FIWARE, destinado exclusivamente para atividades de PoCs que implementam soluções baseadas no processamento de informações de contexto que utilizam o NGSIv2 (Cabrini, 2023).

Na Figura 2, é possível visualizar as camadas da arquitetura dessa ferramenta; no topo, destaca-se a Camada de Aplicação, cuja função é voltada para o front-end tanto da aplicação quanto das ferramentas que irão interagir com os dispositivos IoT e os provedores de contexto. A camada seguinte é a de *back-end*, que abriga os GEs Orion Context Broker, STH-Comet, IoT Agent MQTT, *broker* Eclipse-Mosquitto, o banco de dados NoSQL MongoDB, que armazena as entidades e dados históricos. A última camada é a de IoT, que é responsável por estabelecer a comunicação com os dispositivos que se comunicam com a aplicação (Cabrini, 2023).

Figura 2 – Arquitetura da ferramenta FIWARE Descomplicado



Fonte: Cabrini (2023)

O grande diferencial do FIWARE Descomplicado é a utilização do STH-Comet, componente que permite o armazenamento de dados em larga escala e a consulta de séries temporais com maior eficácia, sendo um recurso mais aderente adequado à solução proposta por este estudo (Cabrini, 2023).

Componentes de comunicação

Para o desenvolvimento desta PoC foi importante identificar os componentes direcionados à comunicação entre as tecnologias envolvidas no processo, da coleta de informações dos sensores existentes no robô à transmissão desses dados ao Orion Context Broker presente no FIWARE Descomplicado. Dentre eles, estão as tecnologias existentes no braço robótico que permitiram a comunicação com os demais componentes da arquitetura deste protótipo.

IoT

Existem diferentes perspectivas para se analisar a definição de Internet das Coisas, e todas elas estão associadas à sua aplicação. Porém, no âmbito geral, IoT é uma tecnologia que possibilita a

conexão de dispositivos e objetos do cotidiano à internet, permitindo a troca de informações e ações automatizadas (Gil, *et al.*, 2016). Tendo como aliado o ambiente de computação em nuvem, seu escopo de aplicação é ampliado e as tecnologias se beneficiam entre si, permitindo uma gama de aplicações em múltiplas áreas, como casas, cidades inteligentes e indústrias inteligentes, bem como em áreas da saúde (Yassein, *et al.*, 2017)

Para garantir a segurança na troca de informações entre os dispositivos, foi proposta uma ontologia de segurança, projetada para ser leve e escalável, permitindo que dispositivos de IoT de baixo consumo de energia possam implementá-la sem problemas. Além disso, ela foi desenvolvida com base em padrões existentes de segurança de dados, como o *Web Ontology Language* (OWL) e o *Simple Knowledge Organization System* (SKOS), com o objetivo de fornecer uma estrutura comum para a troca segura e mais rápida de dados entre dispositivos IoT (Gil, *et al.*, 2016).

RoboARM

O RoboARM é um braço robótico desenvolvido para facilitar o aprendizado de robótica e programação, composto por 4 servomotores e projetado para evitar excesso de carga nos motores, concedendo mais torque para levantar objetos, suportando objetos de até 80g. Ele é desenvolvido e comercializado pela RoboCore e está disponível em um *kit* de montagem (Robocore, 2018).

Esse *kit* contém somente os componentes estruturais; para movimentá-lo, é preciso fazer a integração com uma placa auxiliar, cujas furações comportam a placa Arduino, Raspberry Pi ou a Vespa, uma placa própria do fabricante que pode ser comprada separadamente ou em um *kit* contendo todos os itens (Robocore, 2018).

Placas auxiliares

Arduíno

O Arduíno é uma plataforma *open source* desenvolvida para facilitar a prototipação e integração do hardware e software. Seu hardware é capaz de interpretar as requisições recebidas dos sensores e dispositivos conectados a ela e gerar uma resposta. Seu software, através de sua interface e linguagem de programação com suporte para C/C++, possibilita que o usuário estabeleça instruções que serão interpretadas pelos microcontroladores existentes na placa e transmitirão o comando específico para que as ações sejam executadas (Arduino, 2020).

Existem diferentes modelos de placa de acordo com a finalidade de sua aplicação. Para a aplicação neste estudo, foi utilizada a placa Arduino Mega 2560, cujo hardware também é *open source*, baseado no microcontrolador ATmega2560. Ela possui características encontradas no Arduino UNO R3 e R4, tendo algumas diferenças técnicas relacionadas ao seu tamanho, quantidade de pinos, além de apresentar maior capacidade de memória RAM e demais componentes (Arduíno, 2020).

Arduíno Ethernet Shield

A *shield* Ethernet é uma placa que permite conectar o Arduíno à Internet através do cabo Ethernet RJ45. Compatível com Arduíno UNO ou MEGA, ela é um hardware *open-source*; há diversas placas similares no mercado que foram baseadas em sua arquitetura e na interface Ethernet W5500 da Wiznet, o *upgrade* da versão W5100 utilizada nos primeiros Shields Ethernet fabricados pela marca Arduino.

Através do barramento SPI (*Serial Peripheral Interface*), ela utiliza os pinos 10, 11, 12 e 13 no UNO e 50, 51 e 52 no MEGA, para a conexão SPI com a interface WizNet, e suas versões mais recentes possuem cartão SD integrado (Arduíno, 2023).

Arduíno Ethernet Shield V5

O Arduíno Sensor Shield V5 é um *shield* do tipo expansor, de entradas e saídas, que permite a conexão de componentes e até outros módulos a placas como o Arduíno UNO ou MEGA, proporcionando maior área para se trabalhar com os componentes que serão integrados ao sistema. Ele é muito utilizado para projetos com servomotores, já que algumas características específicas para essa finalidade tornam a integração dos servomotores muito mais fácil, como por exemplo, barramentos tipo alimentação-sinal de 3 pinos e seu conector de alimentação auxiliar que permite uma alimentação separada aos pinos D0 ao D13 (Robocore, 2023).

Neste trabalho, ela foi conectada à placa *shield* Ethernet, que por sua vez estava conectada ao Arduíno MEGA, para que recebesse os pinos conectores dos 4 servomotores, permitindo a conexão do sistema como um todo.

Message Queuing Telemetry Transport (MQTT)

O protocolo de comunicação MQTT tem como base o protocolo TCP/IP, utiliza o modelo cliente-servidor e surgiu como uma forma de simplificar essa comunicação para sensores, estruturado em uma arquitetura de três componentes: *publisher*, *broker* e *subscriber*. Criado pela IBM nos anos 1990, ele é um protocolo M2M (do inglês, *Machine to Machine*), orientado para mensagens, que utiliza o método *Publish-Subscribe* através do *broker*, que atua como um servidor intermediário, recebendo os dados e os trabalha para envio ao cliente e publicando a mensagem em um endereço popularmente chamado de tópico (Muneer et al., 2017).

O MQTT é um protocolo bastante usado na manipulação de dispositivos de IoT e recebimento de informações em tempo de execução, pelo seu baixo consumo de memória e pouca necessidade de processamento para envio das mensagens. Um exemplo de *broker* muito utilizado é o Mosquitto, por ser uma ferramenta *open source* de fácil implementação, o que auxilia na execução de testes (Muneer et al., 2017).

Este tipo de *broker* oferece suporte para implantação da versão com segurança *Transport Layer Security* (TLS), mais utilizada em aplicações executivas; dessa forma, para estudos e aplicações menores, o Mosquitto acaba sendo utilizado na versão MQTT padrão, por não exigir um nível elevado de protocolos de segurança, além de ser mais fácil de configurar (Muneer, et al., 2017).

Para os casos de aplicações mais robustas, principalmente para ambientes de produção, outros *brokers* como o EMQX acabam sendo utilizados por terem a opção embutida e oferecerem uma versão executiva com recursos específicos para o nível de complexidade exigido (EMQ Technologies Inc., 2023).

Unity

O Unity é uma plataforma, inicialmente desenvolvida para a criação de jogos 2D e 3D, que teve um grande desenvolvimento em modelagens para diversas aplicações como nas áreas automotiva, industrial, energética, aeroespacial e até para arquitetura e construção, pela facilidade na criação de componentes 3D, renderização e animação dos modelos. Seu desenvolvimento tem suporte para a linguagem C# e uma vasta comunidade que impulsionam o mercado, assim como sua capacidade de integração com múltiplas plataformas (Unity, 2022).

Sua criação veio com o objetivo de melhorar a indústria dos jogos, trazendo um novo motor gráfico focado na produção 3D. A grande razão para o sucesso da Unity foi o suporte aos pequenos produtores, já que lhes faltavam recursos para licenciar o desenvolvimento. Um clássico exemplo é o jogo GooBall, publicado em março de 2005, que trouxe a oportunidade ao time Unity de trazer melhorias de modo geral (Haas, 2014).

Recentemente, o Unity deixou de ser utilizado apenas para jogos; o motor 3D vem ganhando credibilidade em diversos meios: arquitetos podem desenvolver suas ideias de forma simples, artistas têm a capacidade de desenvolver exposições de artes interativas e pesquisadores encontram utilidade para a visualização de dados (Haas, 2014).

Blender

O Blender, de acordo com seus criadores, é uma “suíte de criação 3D gratuita e de código aberto” que permite a modelagem de objetos 3D, animações e efeitos especiais, podendo ser utilizado em multiplataformas, como Linux, macOS e Windows. Utilizando o OpenGL, consegue fornecer uma interface simples e consistente, sendo possível sua personalização com scripts em Python (Blender, 2023).

Metodologia

O desenvolvimento dessa PoC foi baseado nas arquiteturas propostas pela FIWARE para implementação de DT no cenário de indústrias inteligentes, buscando unir seus principais pontos e reproduzi-los para a construção de um gêmeo digital de um braço robótico. Esse robô é equipado com dispositivos de IoT que permitem alimentar o sistema com dados em tempo de execução, além de simular o ambiente produtivo em operação nas mais diversas situações, utilizando os dados obtidos para auxiliar na tomada de decisão.

Pela grande variedade de cenários de aplicação para a tecnologia a ser implantada, houve uma longa análise cujo caminho seria seguido como foco do estudo. Inicialmente, os autores consideraram utilizar como base as cidades inteligentes, mas a dificuldade para a manutenção dos sensores que

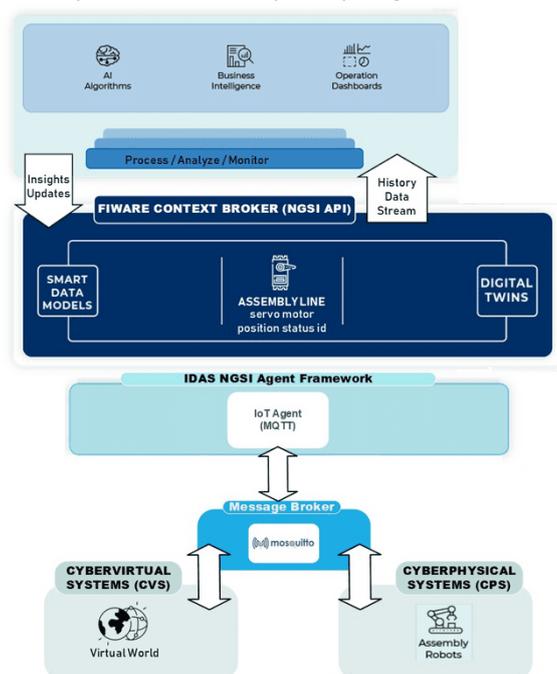
iriam coletar as informações de contexto para esse caso e a realização de testes inviabilizaram o estudo.

Diante disso, cogitou-se estudar o cenário de agricultura inteligente, para a análise do processo produtivo em fazendas, visando quantificar a previsibilidade de produção, mas as mesmas dificuldades acabaram por motivar a troca do escopo de aplicação.

Por fim, o cenário de aplicação escolhido foram as indústrias inteligentes, inspirado em um *Computer Integrated Manufacturing* (CIM) que a princípio serviria como modelo para o DT, mas por ser um sistema mais antigo, cujo compartilhamento dos dados estava disponível apenas via comunicação serial, foi substituído por um braço robótico individual.

Para que os dados gerados pelo robô fossem transmitidos ao FIWARE Descomplicado, num fluxo de comunicação bidirecional que permitisse provar o conceito a ser modelado, foi necessário aplicar uma arquitetura conforme a demonstrada na Figura 3. Nela é possível identificar os protocolos de comunicação escolhidos para a coleta dos dados no dispositivo IoT, visualizar o fluxo de processamento da informação e a alimentação do gêmeo digital suportados pelas tecnologias FIWARE, assim como os *frameworks* de modelagem do robô que permitiram o desenvolvimento de um modelo virtual que replicará o comportamento real.

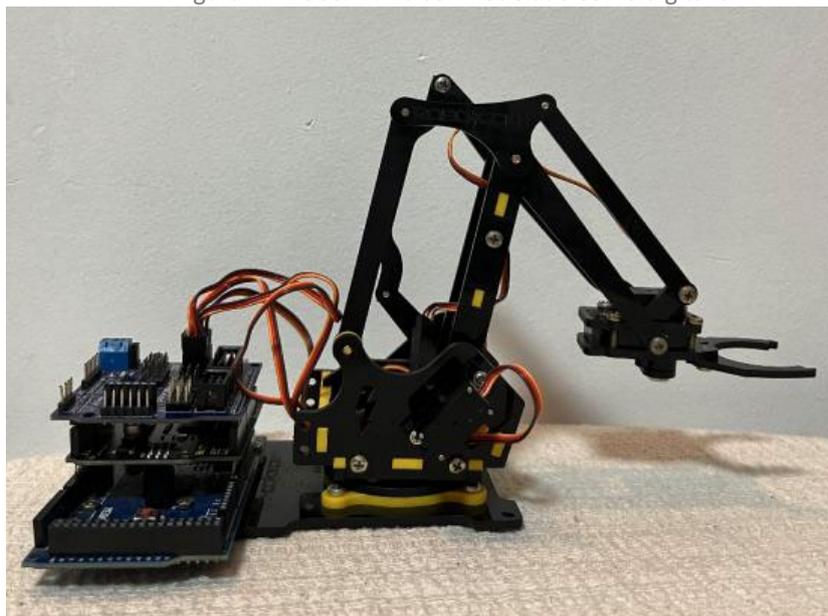
Figura 3 – Arquitetura utilizada para aplicação do DT de um braço robótico



Fonte: Elaboração dos autores (2023)

O robô em questão é um RoboARM, e sua montagem foi o ponto de partida para o desenvolvimento desta PoC. Montado pelos próprios autores, é possível observá-lo na Figura 4. Ele é controlado por uma placa Arduino MEGA integrada a uma Sensor Shield V5 e placa Ethernet que permitem a manipulação da garra e do braço robótico, assim como sua comunicação e envio de dados, através de dispositivos de IoT, ao *context broker*, por meio do protocolo MQTT.

Figura 4 – RoboARM a ser modelado como digital twin



Fonte: Elaboração dos autores (2023)

Para que isso fosse possível, foi utilizado o código disponível no repositório TCC-DT-Robot do GitHub através do *link*: <https://github.com/mathk9/TCC-DT-Robot/blob/master/roboArmMQTT.cpp>. Este script, escrito em linguagem C++, foi adaptado para esta finalidade e inserido na placa Arduino Mega acoplada no robô. Ele permite que o Arduino interprete os comandos enviados e transmita a posição escolhida ao servomotor, que irá se mover para a posição correta.

Com o robô funcionando, para dar continuidade no desenvolvimento do protótipo de gêmeo digital, a próxima etapa a ser definida é a integração com o *middleware* FIWARE Descomplicado. Para isso, foi criada uma conta estudantil no portal Azure, provedor de cloud da Microsoft, para a criação de uma máquina virtual (VM) de acordo com as especificações de configuração do *framework* disponíveis no arquivo ReadMe.md do GitHub, através do *link*: <https://github.com/fabioabrini/fiware/blob/main/README.md>.

Para a modelagem do DT, foi adquirida uma licença estudantil gratuita do Unity Hub, na categoria *Professional*, com validade de um ano. Nela, foi construído o modelo virtual do braço robótico idêntico ao real, cujo desenho 3D foi realizado no Blender, uma ferramenta *open source* que possibilita esse tipo de montagem.

Uma vez finalizado o desenho em 3D e seus eixos permitindo a movimentação do robô, foram incluídos no *script* os comandos que realizam a animação do modelo através do Unity, diferenciando as camadas do desenho e seus eixos para que, a cada requisição recebida pelo robô físico, o robô virtual saiba qual movimento deve realizar.

A etapa final do desenvolvimento consiste na integração dos ambientes físico e virtual; para isso, os dados são modelados em 4 subscrições diferentes, como se fossem *triggers*, ou seja, gatilhos que disparam a atualização dos motores para que os movimentos sejam realizados. Conforme as alterações são realizadas no robô físico, as subscrições são atualizadas e o robô virtual, que fica constantemente atualizando o status delas, reproduz o movimento da versão atual. E assim sucessivamente; no caso de o movimento ser realizado no robô 3D, a atualização da subscrição ocorre em tempo de execução e os movimentos são replicados pelo robô real.

É possível visualizar, na Figura 5, a consolidação de todas as etapas apresentadas neste capítulo para a criação do gêmeo digital do RoboARM, cujo fluxo de desenvolvimento do protótipo é exemplificado através de um diagrama, a fim de facilitar a replicação desse experimento em trabalhos futuros.

Figura 5 – Diagrama de desenvolvimento do protótipo de DT



Fonte: Elaboração dos autores (2023)

Os dados trabalhados neste estudo não se caracterizam como dados pessoais ou que manipulem informações sensíveis ou plausíveis de aplicação da Lei Geral de Proteção de Dados (LGPD); trata-se de dados públicos provenientes de experimentos realizados pelos próprios autores.

Resultados e discussão

Os testes iniciais com o robô foram realizados via comunicação serial; o envio era feito diretamente para o Arduino e o movimento era executado. Esses *setups* de calibração evoluíram e os ajustes finos foram aplicados no controle operacional do robô, assim como o refinamento do *script*, para que a comunicação fosse feita através da API, do inglês *Application Programming Interface*, e que os dados fossem armazenados no banco de dados MongoDB, que seria administrado pelo *middleware* FIWARE Descomplicado.

Para o envio das requisições ao robô foi utilizado o Postman, uma plataforma para criação e utilização de APIs, que centraliza e facilita o gerenciamento das requisições. A Figura 13 mostra parte do JSON responsável por criar a *collection* no Postman, disponível no repositório do GitHub com o *link*: [https://github.com/mathk9/TCC-DT-Robot/TCC Collection.postman_collection.json](https://github.com/mathk9/TCC-DT-Robot/TCC%20Collection.postman_collection.json) at master · mathk9/TCC-DT-Robot · GitHub.

Uma *collection* é uma série de requisições pré-definidas que padronizam o envio das solicitações para o robô. Através dos comandos definidos nela, é criada a entidade do robô que será

trabalhada e armazenada no banco, definindo o padrão que será utilizado para armazenar os dados no banco. Ela foi criada com base na *collection* disponível no repositório do FIWARE Descomplicado, através do *link*:

https://github.com/fabioabrini/fiware/blob/main/FIWARE%20Descomplicado.postman_collection.json, sendo adaptada para as necessidades do projeto.

Tendo concluído a etapa de testes para a manipulação do robô através da API, deu-se sequência aos testes da animação do modelo virtual. O objetivo desta fase era separar as camadas do desenho 3D e agrupá-las no código conforme o robô real se movimentava, permitindo que as peças fossem ordenadas de acordo com o tipo de movimento realizado por cada peça. Este código foi escrito em linguagem C#, na versão suportada pelo Unity e está disponível na pasta “Scripts” através do *link*: https://github.com/mathk9/TCC-DT-Robot/tree/master/Robot_3D/Assets/roboarm/Scripts. A Figura 6 mostra o robô virtual disponível no Unity.

Figura 6 – Robô virtual no Unity



Fonte: Elaboração dos autores (2023)

Com os dois robôs funcionando, os testes finais foram iniciados para estabelecer a comunicação de ponta a ponta, permitindo o envio de dados do robô para a replicação de seu movimento pelo seu gêmeo digital, cuja manipulação das requisições através do Postman foi feita utilizando os métodos da TCC Collection. No método POST, chamado “Provisionar um dispositivo”, é criado o dispositivo “DeviceRoboArm001”, que armazena a entidade “RoboArm” e seus atributos para cada motor. Ele é executado inicialmente para criar esse dispositivo no MongoDB e definir a forma com que os dados serão armazenados.

Os métodos GET dessa *collection* permitem a consulta das entidades e atributos armazenados no banco de dados e facilitam o gerenciamento e visualização das informações. Para a movimentação do robô, é utilizado o método POST da categoria *Subscribe*, criado para cada um dos 4 servomotores. Através do *design pattern Publish-Subscribe*, conhecido como Pub-Sub, ele estabelece uma conexão onde há um ouvinte, aguardando as atualizações daquele que envia as requisições.

Dessa forma, a cada envio do POST com o método *Subscribe* do motor correspondente, ocorre a atualização no tópico MQTT, e aquele que está de como ouvinte no fluxo de comunicação atualiza seus parâmetros. Se as publicações são enviadas do robô físico, o modelo virtual se enquadra como ouvinte, e a cada atualização do tópico ele reproduz o movimento em tempo de execução. Se o cenário for inverso, a publicação é disparada pelo robô virtual e o ouvinte passa a ser o robô físico, concluindo o cenário de fluxo bidirecional.

Isso permite provar o conceito proposto por este estudo, demonstrando que, ao enviar um comando para o braço robótico executar, o seu movimento também é realizado por seu gêmeo digital através da validação dos dados de posição determinado pelo ângulo dos servomotores, fazendo com que o modelo virtual reproduza o cenário real, conforme evidenciado no vídeo em alta resolução disponível através do *link*: <https://youtu.be/azPzDzYlbdS>.

Esses dados históricos contendo o ângulo de posição dos motores são armazenados no MongoDB através da integração dos componentes FIWARE STH-Comet e Orion Context Broker, que permitem o registro dos referidos dados em série temporal, ou seja, eles registram o “*timestamp*” do atributo e seu valor em ordem cronológica, permitindo que aquele movimento seja recriado em análises futuras, já que se tem a sequência das posições que foram realizadas pelos motores. Conforme apresentado na Figura 7, em que é possível visualizar os registros dos itens 440 ao 432 através da ferramenta MongoDB Compass, um *dashboard* permite visualizar os *logs* armazenados no banco de dados, o que indica que seria possível simular o movimento exato realizado pelo robô seguindo essa sequência de registros.

Figura 7 – Registro de dados históricos de movimentação dos robôs

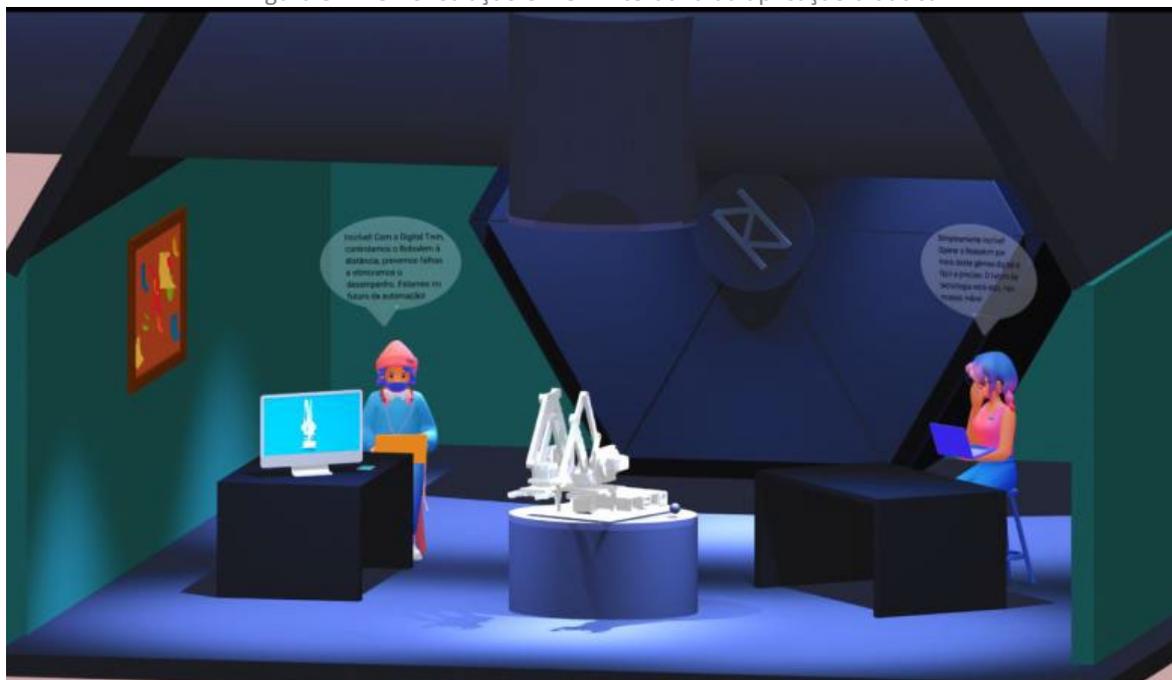
sth/_urn:ngsi-Id:RoboArm:001_RoboArm	_id	Objectid	recvTime Date	attrName String	attrType String	attrValue Int32
432	Objectid('655bde94e3aa900077...)	2023-11-20T22:32:45.640+00:00	"motor4"	"Integer"	57	
433	Objectid('655bde94e3aa900077...)	2023-11-20T22:32:49.797+00:00	"motor3"	"Integer"	122	
434	Objectid('655bde94e3aa900077...)	2023-11-20T22:32:52.511+00:00	"motor3"	"Integer"	82	
435	Objectid('655bde94e3aa900077...)	2023-11-20T22:32:56.915+00:00	"motor3"	"Integer"	42	
436	Objectid('655bde94e3aa900077...)	2023-11-20T22:32:59.917+00:00	"motor1"	"Integer"	62	
437	Objectid('655bde94e3aa900077...)	2023-11-20T22:33:03.760+00:00	"motor1"	"Integer"	91	
438	Objectid('655bde94e3aa900077...)	2023-11-20T22:33:06.885+00:00	"motor1"	"Integer"	64	
439	Objectid('655bde94e3aa900077...)	2023-11-20T22:33:10.661+00:00	"motor3"	"Integer"	80	
440	Objectid('655bde94e3aa900077...)	2023-11-20T22:33:13.394+00:00	"motor3"	"Integer"	105	

Fonte:

Fonte: Elaboração dos autores (2023)

Para demonstração da aplicação didática desta PoC foi desenvolvida uma experiência interativa em 3D, no Spline, que assim como o Blender, é uma ferramenta gratuita e *on-line* que possibilita criar modelos tridimensionais. Essa demonstração permite a navegação em um laboratório onde os alunos estão estudando um braço robótico e seu gêmeo digital. Nela é possível visualizar o robô se movendo no centro, e o seu DT no computador, replicando os movimentos, conforme demonstrado na Figura 8.

Figura 8 – Demonstração em 3D interativa da aplicação didática



Fonte: Elaboração dos autores (2023)

Além disso, é possível andar pelo ambiente através das teclas do computador ou do celular, e realizar a leitura dos textos de apoio que explicam a tecnologia. Esta experiência está disponível no [link: https://mathk9.github.io/Relatorio-TCC-Spline/labDTWebSite](https://mathk9.github.io/Relatorio-TCC-Spline/labDTWebSite).

Considerações finais

Este estudo delineou as fases de desenvolvimento de um protótipo de *digital twin* de um braço robótico, por meio de uma PoC que visou demonstrar o fluxo de comunicação bidirecional entre o robô físico e sua representação virtual, sendo este o objetivo central da pesquisa. A análise detalhada dos resultados apresentados na seção 4 (Resultados e discussão) ratifica o alcance do objetivo principal proposto.

Adicionalmente, foi possível atingir objetivos específicos, tais como a formulação de uma arquitetura simplificada, fundamentada na estrutura FIWARE para aplicações em indústrias inteligentes, que viabiliza a comunicação bidirecional entre o gêmeo digital e o robô físico. Além disso, foi demonstrada a aplicabilidade didática da solução, evidenciada pela experiência interativa em 3D, que apresenta a facilidade agregada à introdução do tema no âmbito acadêmico e implementação eficiente do modelo proposto.

O principal desafio enfrentado durante a pesquisa concentrou-se na integração entre o robô físico e seu modelo em 3D. Enquanto as etapas funcionavam separadamente, a integração exigiu uma atenção mais detalhada para garantir que o envio das requisições e a replicação dos movimentos fossem fiéis ao conceito proposto. Algumas limitações foram identificadas durante a condução do estudo, e os autores optaram por abordá-las em pesquisas futuras. Entre elas, destaca-se a complexidade na manipulação da comunicação MQTT com o protocolo TLS, que, por caracterizar-se como um protótipo de aplicação didática, não exigiu uma abordagem de segurança mais robusta. Como diretrizes para futuras pesquisas, está prevista a inclusão de elementos de previsibilidade nas métricas associadas aos movimentos do robô, bem como o aprimoramento da autonomia do sistema. Além disso, planeja-se a implementação de um ambiente que permita a realização de simulações comportamentais no contexto em que o robô está inserido, integrando-o a novos sensores capazes de agregar informações de contexto para enriquecer o conjunto de dados avaliados.

Durante a exposição dos resultados na banca, algumas sugestões de implementação futuras foram apresentadas e dentre elas houve destaque para a aplicação de um *dashboard* de visualização, em tempo de execução, do posicionamento, velocidade e força para os 4 eixos do RoboARM, assim

como a possibilidade de inclusão do comissionamento dos movimentos em um CLP (Controlador Lógico Programável), o que possibilitaria a utilização em um robô industrial, como o CIM.

Esses trabalhos futuros visam não apenas aprimorar a precisão e eficácia do modelo proposto, mas também ampliar suas capacidades ao introduzir elementos preditivos e simulações contextualizadas, a fim de oferecer uma representação mais fiel e útil do ambiente em que o robô opera além de trazer um viés de interdisciplinaridade ao permitir sua integração com o curso de Engenharia de Controle e Automação.

Uma importante contribuição deste projeto foi a publicação do artigo de título “Desenvolvimento de um protótipo de Digital Twin através do processamento de informações de contexto para identificar potenciais melhorias numa planta industrial”, submetido e aceito como trabalho em andamento, sem modificações, no V SIMAC (Simpósio Acadêmico da Faculdade Engenheiro Salvador Arena), apresentado no dia 11 de novembro de 2023 e indicado a premiação.

Referências

- ARDUINO. Arduino Ethernet Shield. **Arduino Docs**, 2023. Disponível em: <https://docs.arduino.cc/retired/shields/arduino-ethernet-shield-without-poe-module>. Acesso em: 25 nov. 2023.
- ARDUINO. Arduino Mega 2560 Rev3. **Arduino Documentation**, 2023. Disponível em: <https://docs.arduino.cc/hardware/mega-2560>. Acesso em: 21 out. 2023.
- BAJIC, Bojana et al. Industry 4.0 Implementation Challenges and Opportunities: A Managerial Perspective. **IEEE Systems Journal**, v. 15, n. 1, p. 546-559, 2020.
- BAUER, Martin et al. Urban Digital Twins—a fiware-based model. **Automatisierungstechnik**, v. 69, n. 12, p. 1106-1115, 2021.
- BEEK, Anton V.; KARKARIA, Vispi N.; CHEN, Wei. Digital twins for the designs of systems: a perspective. **Springer Nature**, v. 66, n. 3, p. 49, 2023.
- BLENDER. Introduction. **Docs Blender**, 2023. Disponível em: https://docs.blender.org/manual/en/latest/getting_started/about/introduction.html. Acesso em: 13 nov. 2023.
- BRASIL, Nações Unidas. Objetivos de Desenvolvimento Sustentável. **Nações Unidas Brasil**. Disponível em: <https://brasil.un.org/pt-br/sdgs>. Acesso em: 15 set. 2023.
- CABRINI, Fábio Henrique et al. Enabling the Industrial Internet of Things to Cloud Continuum in a Real City Environment. **Sensors**, v. 21, n. 22, p. 7707, 2021.
- CABRINI, Fábio Henrique. fiware. **GitHub**, 2023. Disponível em: <https://github.com/fabiocabrini/fiware>. Acesso em: 20 ago. 2023.
- CARLOS, Juan. SpaceX: Enabling Space Exploration through Data and Analytics. **Harvard Digital Innovation and transformation**, 2021. Disponível em: <https://d3.harvard.edu/platform-digit/submission/spacex-enabling-space-exploration-through-data-and-analytics/#>. Acesso em: 29 nov. 2023.

CONDE, Javier et al. Modeling Digital Twin Data and Architecture: A Building Guide with FIWARE as Enabling Technology. **IEEE Internet Computing**, v. 26, n. 3, p. 7-14, 2021.

ENTERTAINMENT, Over T. E. GooBall. **Macintosh Repository**, 2023. Disponível em: <https://www.macintoshrepository.org/26983-gooball>. Acesso em: 15 nov. 2023.

GIL, David et al. Internet of Things: A Review of Surveys Based on Context Aware Intelligent Services. **Sensors**, v. 16, n. 7, p. 1069, 2016.

GONZALEZ-GIL, Pedro; MARTINEZ, Juan A.; SKARMETA, Antonio F. Lightweight Data-Security Ontology for IoT. **Sensors**, v. 20, n. 3, p. 801, 2020.

INC., EMQ T. EMQX Overview. **EMQX**, 2023. Disponível em: <https://www.emqx.io/docs/en/latest/>. Acesso em: 23 out. 2023.

NVIDIA. NVIDIA'S Earth -2: Digital Twins for Weather and Climate. **NVIDIA On-Demand**, 2022. Disponível em: <https://www.nvidia.com/en-us/on-demand/session/gtcfall22-a41326/>. Acesso em: 29 nov. 2023.

ROBOCORE. Braço Robótico RoboARM. **RoboCore**, 2018. Disponível em: <https://www.robocore.net/>. Acesso em: 15 abr. 2023.

ROBOCORE. Shields. **Robocore**, 2023. Disponível em: <https://www.robocore.net/shields-arduino/arduino-shield-sensor-shield-v5>. Acesso em: 21 nov. 2023.

SCHROEDER, Greyce N. et al. Digital Twin Data Modeling with AutomationML and a Communication Methodology for Data Exchange. **IFAC-PapersOnLine**, v. 49, n. 30, p. 12-17, 2016.

SIEMENS. PLM Software. **Siemens**, 2023. Disponível em: <https://www.geoplms.com/knowledge-base-resources/GEOPLM-Siemens-PLM-NX-SpaceX-cs-Z10.pdf>. Acesso em: 29 nov. 2023.

SINGH, Maulshree et al. Digital Twin: Origin to Future. **Applied System Innovation**, v. 4, n. 2, p. 36, 2021.

SOUZA, Matheus de Novais. TCC-DT-Robot. **GitHub**, 2023. Disponível em: <https://github.com/mathk9/TCC-DT-Robot>. Acesso em: 20 ago. 2023.

TAO, Fei et al. Digital Twins and Cyber-Physical Systems toward Smart Manufacturing and Industry 4.0: Correlation and Comparison. **Engineering**, v. 5, n. 4, p. 653-661, 2019.

TECHNOLOGIES, Unity. Unity User Manual 2022.3 (LTS). **Unity Documentation**, 2023. Disponível em: <https://docs.unity3d.com/Manual/index.html>. Acesso em: 12 out. 2023.

YASSEIN, Muneer Bani et al. Internet of Things: Survey and open issues of MQTT protocol. In: 2017 international conference on engineering & MIS (ICEMIS). **IEEE Xplore**, 2017. p. 1-6

ZHANG, Rui et al. Digital twin and its applications: A survey. **The International Journal of Advanced Manufacturing Technology**, v. 123, n. 11-12, p. 4123-4136, 2022.