



ESCOLA POLITÉCNICA DA UNIVERSIDADE DE SÃO PAULO
APLICATIVO DE GÊMEO DIGITAL EM REALIDADE AUMENTADA
PCS5112

Ricardo Gomes Miguel

RESUMO

O conceito de simulação na indústria, consiste em realizar ações de validação e otimização de produtos e processos. Quando o conceito de simulação é expandido, sendo aplicado de forma integrada a algum dispositivo físico por meio das tecnologias de comunicação, surge o termo Gêmeo Digital. Enquanto isso, as tecnologias imersivas por meios de aplicações de Realidade Aumentada tornam a experiência do usuário mais interativas e intuitivas. Sendo assim, as tecnologias de Gêmeo Digital e Realidade Aumentada, ambas tecnologias habilitadoras da Indústria 4.0, podem atuar de forma conjunta, enriquecendo a qualidade da interface entre Gêmeo Digital e usuário. Visando aplicar tal integração, o objetivo deste trabalho é apresentar uma aplicação de Gêmeo Digital imersivo com Realidade Aumentada, que permita o espelhamento dos estados de um equipamento, no caso, uma estação de automação, além de permitir o comissionamento de lógicas de controle de sistemas discretos. A aplicação proposta é composta por três subsistemas, um servidor OPC-UA representando um Controlador Programável, a aplicação mobile desenvolvida com o Google AR Core e Unity 3D, e um sistema intermediário que tem a função de estabelecer a comunicação entre o servidor e a aplicação, convertendo o padrão OPC-UA para o protocolo MQTT e vice-versa. Os testes da aplicação demonstraram que a interação imersiva foi satisfatória, com destaque para estabilidade do elemento virtual no ambiente físico, e com a ressalva de que a latência na transmissão de dados influenciou diretamente o espelhamento dos estados, sendo necessário realizar ações para mitigar seus efeitos.

Palavras-chave: Digital Twin, Augmented Reality, OPC-UA, MQTT, Industry 4.0.

ABSTRACT

The simulation concept in the industry, consists of carrying out validation and optimization actions of products and processes. When the concept of simulation is expanded, being applied in an integrated way to some physical device through communication technologies, the term Digital Twin arises. Meanwhile, immersive technologies through Augmented Reality applications make the user experience more interactive and intuitive. Thus, the technologies of Digital Twin and Augmented Reality, both enabling technologies of Industry 4.0, can work together, enriching the quality of the interface between Digital Twin and user. Aiming to apply such integration, the objective of this work is to present an application of immersive Digital Twin with Augmented Reality, which allows the mirroring of the states of an equipment, in this case, an automation station, besides allowing the commissioning of control logic of discrete systems. The proposed application consists of three subsystems, an OPC-UA server representing a Programmable Controller, the mobile application developed with Google AR Core and Unity 3D, and an intermediate system that has the function of establishing communication between the server and the application, converting the OPC-UA standard to the MQTT protocol and vice versa. The application tests showed that the immersive interaction was satisfactory, especially the stability of the virtual element in the physical environment, and with the exception that the latency in the data transmission directly influenced the state mirroring, being necessary to carry out actions to mitigate its effects.

1 INTRODUÇÃO

O conceito de simulação nas indústrias é antigo, e se baseia na ideia de realizar tarefas em um ambiente controlado e virtual, de modo a realizar testes de validação, e otimizar processos de forma segura (de Paula Ferreira et al., 2020).

Expandindo este conceito, realizando simulações de maneira vinculada à dispositivos físicos, surgiu o termo Gêmeo Digital, que de forma geral, consiste em uma réplica digital de um sistema, seja ele físico ou virtual.

Com o surgimento da Indústria 4.0, onde a principal característica é a digitalização e conectividade nas empresas, o conceito de Gêmeo Digital se enquadra como uma das tecnologias habilitadoras.

Paralelo a isso, surgem as tecnologias imersivas, como Realidade Virtual (RV) e Realidade Aumentada (RA), que promovem novas experiências aos usuários nas diversas atividades, sejam elas na educação, no entretenimento, na saúde e na indústria.

A interface entre usuários e Gêmeos Digitais costuma ser realizada através de monitores convencionais, que são importantes, no entanto não permitem uma interação natural com os modelos 3D de modo intuitivo, enquanto a RA combinada com recursos gráficos e sonoros permite que os usuários consigam interagir com Gêmeos Digitais em alto nível, além de prover uma visualização contínua e intuitiva do Gêmeo Digital (Zhu, 2019).

Sendo assim, o objetivo deste artigo é apresentar uma aplicação de Gêmeo Digital imersivo com Realidade Aumentada, que permita o espelhamento de estados de um equipamento, no caso, uma estação de automação, além de permitir o comissionamento de lógicas de controle de sistemas discretos.

A intenção é que a aplicação seja empregada na indústria, no entanto, devido suas características didáticas, pode ser empregada no ensino de engenharia.

Para que os estados sejam espelhados na aplicação RA, será necessário um sistema de comunicação sem fio para que os dados possam fluir de forma bidirecional, entre o dispositivo e a aplicação RA.

Os dispositivos de automação costumam trabalhar com diversos protocolos de comunicação industriais como Modbus, Profinet, entre outros. Buscando atender a necessidade de alcançar a interoperabilidade, uma das premissas da Indústria 4.0, foi criado o padrão OPC-UA, de modo que muitos fabricantes de Controladores Programáveis (CP) já contam com esse padrão.

No entanto, a maioria dos CPs com OPC-UA não possuem recursos de conectividade sem fio, sendo assim, foi desenvolvido um sistema intermediário que atua como um Gateway convertendo protocolos. Para a transmissão sem fio, foi escolhido o protocolo MQTT, por ser adequado em situações que a atualização dos dados em tempo real é prioridade. Desta forma, o Gateway fará a função de converter os dados de OPC-UA para MQTT e vice-versa.

A aplicação possui dois modos de funcionamento, o modo exibição offline onde um programa interno, contendo a lógica do controle de eventos discretos, é executado, e o modo de espelhamento online de um CP virtual, espelhando as entradas e saídas do controlador na aplicação RA.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 Gêmeo Digital (Digital Twin)

O conceito de Digital Twin, de acordo com Grieves (2014), se baseia na construção de um sistema Ciber-Físico composto por três partes principais:

- Dispositivo físico no espaço real;
- Dispositivo virtual no espaço virtual;
- Conectividade entre os dispositivos físicos e virtuais.

Este sistema funciona de modo que os eventos que ocorrem em um dispositivo sejam refletidos no outro.

No lado virtual, diversas melhorias foram implementadas com o passar do tempo, como a adição de características comportamentais que permitem, não só a visualização de dados do dispositivo físico em tempo real, como também a realização de testes de desempenho (Grieves, 2014).

Os Digital Twins modernos, são capazes de apresentar a versão virtual do dispositivo de maneira leve e otimizada, permitindo que a comunicação e atualização em tempo real de eventos ocorra reduzindo os esforços computacionais. Desta forma, é possível, por exemplo, realizar simulações dos processos de fabricação de um produto em ambientes virtuais.

O conceito de Digital Twin, de acordo com Kritzinger (2018) pode ser classificado em três diferentes níveis de integração:

Digital Model, que se trata da representação digital, na forma de um modelo, de uma entidade física, de modo que o modelo digital corresponde a uma versão futura da entidade física. Neste nível não existe comunicação entre as versões físicas e digitais, e os processos de integração são realizados de forma emulada.

Digital Shadow, que basicamente se trata da representação digital de uma entidade física, que estabelece comunicação entre as versões físicas e digitais, no entanto, esta comunicação não é bidirecional, ou seja, os dados fluem, de forma automática, da entidade física para a entidade digital, e a comunicação inversa é possível, mas ocorre de forma manual.

Digital Twin, propriamente dito, que se trata da representação da entidade física em um ambiente digital, com a possibilidade de monitorar, emular e controlar aspectos da entidade física. A comunicação flui de forma automática em ambas as direções e em tempo real.

2.2 Tecnologias Imersivas

As tecnologias imersivas permitem a visualização de informações e a interação com objetos virtuais. As principais tecnologias imersivas são a Realidade Virtual e Realidade Aumentada.

A RV, por meio de tecnologias computacionais, possibilita a criação de realidades alternativas, que simulam ambientes e sistemas reais, sendo uma importante ferramenta nas áreas da saúde, educação, entretenimento e indústria. O desenvolvimento dessa tecnologia surgiu na década de 1960, sendo que, até o início do século XXI se limitava a laboratórios de pesquisa, devido ao alto custo, o que inviabilizava a sua popularização (Tori et al., 2020). Com a evolução tecnológica, os equipamentos se tornaram mais acessíveis, e dispositivos móveis, como smartphones e tablets, se tornaram aptos a comportar aplicações imersivas.

As aplicações de RV possibilitam a criação de diversas experiências do mundo real de maneira simulada e segura. No segmento da indústria, a RV permite a visualização de modelos 3D durante a fase de desenvolvimento do produto, permitindo a interação com o protótipo virtual. Além dessa aplicação, a RV pode beneficiar sistemas de treinamento, simulações de montagens, simulações de dinâmica, análise estrutural, simulação de processos produtivos, entre outras.

Enquanto as aplicações de RV criam ambientes totalmente virtuais, e insere o usuário nestes ambientes, as aplicações de RA mantêm as referências do ambiente real, inserindo elementos virtuais, de modo que o usuário seja capaz de interagir tanto com o ambiente real quanto com os elementos virtuais, entregando uma experiência mais natural e intuitiva que a RV.

As aplicações de RA devem identificar o local onde serão inseridos os elementos virtuais, e definir como será a apresentação visual para o usuário com base na localização do seu ponto de vista. Tal processo é conhecido como rastreamento,

que é realizado seguindo diversas técnicas como reconhecimento de planos e rastreamento de marcadores.

A tecnologia RA pode ser aplicada em diversos segmentos, trazendo benefícios específicos, devido à sua natureza de combinar os objetos reais e virtuais, para qualquer atividade que tenha a necessidade de acesso às informações, seja em tempo real ou não. Quando a informação a ser exibida tiver características tridimensionais e, que de alguma forma, se relaciona com o ambiente real, a tecnologia RA provavelmente será a melhor opção (Tori et al, 2020). Entre as aplicações de RA se destacam, atividades de operação e manutenção de equipamentos, visualização de informações a respeito de dispositivos e instalações industriais, atividades de treinamento, entre outras.

2.3 Padrão OPC-UA

O padrão OPC, cuja sigla se refere a *Object linking and embedding for Process Control*, inicialmente era compatível apenas ao Windows, sendo denominada atualmente como OPC Clássico. Mesmo sendo restrito, foi largamente utilizado em diversos setores como manufatura, automação, petróleo e gás, energia renovável e serviços públicos (OPC Foundation, 2023).

Para expandir o padrão para outros sistemas operacionais atendendo requisitos de segurança, foi desenvolvido o padrão OPC-Unified Architecture.

O padrão OPC-UA promove a troca de informações de forma segura e confiável independentemente do sistema operacional e do fabricante dos dispositivos envolvidos.

Este padrão opera pelo modo Cliente-Servidor onde as informações pertencem ao servidor, e os clientes fazem requisições para terem acesso a essas informações, e pelo modo Publicador-Assinante, onde as informações são publicadas pelos publicadores, e os consumidores que estiverem conectados ao sistema de recebem automaticamente as informações publicadas.

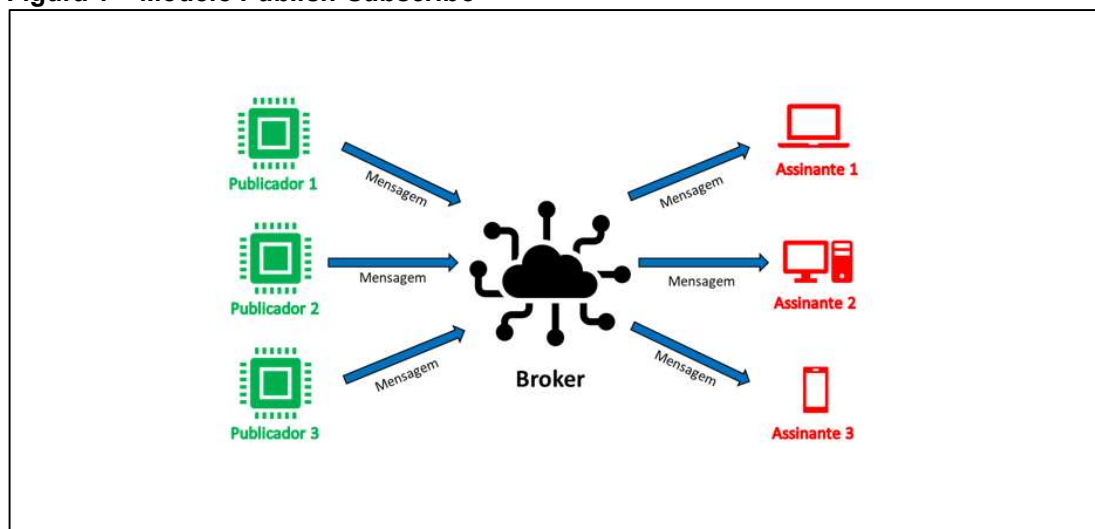
A camada de transporte do padrão OPC-UA é definida com base do modo de operação. Para o modo Cliente-Servidor as informações são trocadas através dos protocolos TCP e HTTPS, enquanto o modo Publicador-Assinante permite a utilização dos protocolos UDP e MQTT.

2.4 Protocolo MQTT

O protocolo denominado “Transporte de Telemetria do Serviço de Enfileiramento de Mensagens” (*Message Queuing Telemetry Transport* - MQTT) foi criado pela IBM para ser utilizado de maneira simples, dispensando a exigência de alta capacidade de processamento, o que viabiliza sua utilização em microcontroladores, e o torna o principal protocolo para a Internet das Coisas (IoT) (HiveMQ, 2023).

O MQTT opera no modo Publicador-Assinante, mostrado na figura 1, o servidor é chamado de *Broker*, e atua filtrando e direcionando as mensagens, de modo que o *Publisher* (Publicador) não precisa conhecer o *Subscriber* (Assinante) e vice-versa. Quando estiverem conectados ao mesmo *Broker*, as notificações de alterações de mensagens são enviadas a todos os assinantes inscritos sem a necessidade de requisição, enquanto a função de enviar os dados ao *Broker* fica destinada ao Publicador.

Figura 1 – Modelo *Publish-Subscribe*



Fonte: Elaborado pelo autor

O protocolo MQTT permite que os elementos conectados ao *Broker* possam atuar tanto como Publicador como Assinante, permitindo que dispositivos IoT possam receber e enviar dados.

Detalhando a ação do *Broker*, a distribuição das informações publicadas é organizada por tópicos, que podem ser definidos como locais onde as informações estarão disponíveis aos assinantes. Exemplificando, considere um dispositivo IoT que monitora a temperatura de um forno, e envia ao *Broker* o valor da temperatura

atualizada a cada cinco segundos em um tópico específico, nesse caso, o dispositivo IoT está atuando como Publicador. Os dispositivos que estiverem conectados ao mesmo *Broker*, e forem assinantes do mesmo tópico, receberão o valor da temperatura atualizado a cada cinco segundos, sem a necessidade de realizar uma requisição. Na prática o cliente Publicador, no caso o dispositivo IoT, não sabe quem são os assinantes, mas sabe que todos vão receber os dados atualizados. Enquanto os dispositivos clientes Assinantes não sabem a origem dos dados, mas sabem que estão sendo atualizados com o valor de interesse. Portanto o *Broker* é elemento principal que gerencia todo o processo de comunicação.

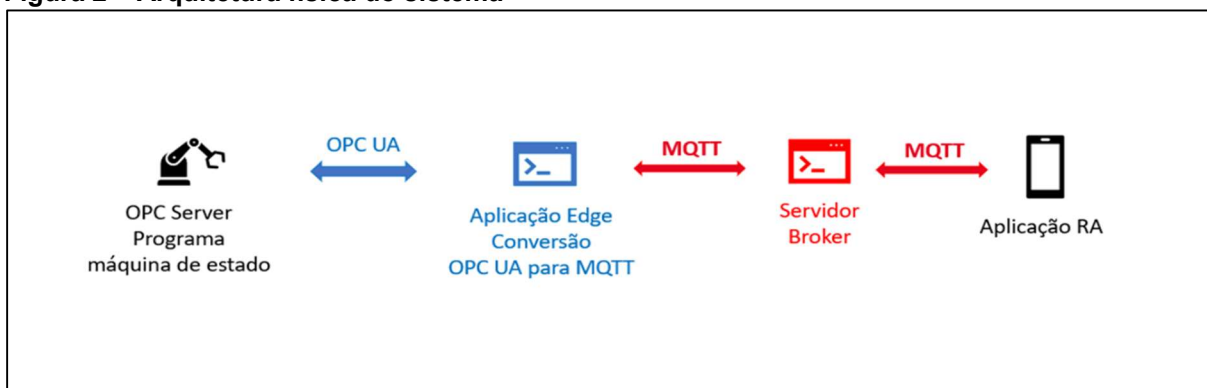
3 METODOLOGIA

Visando apresentar o conceito de Gêmeo Digital imersivo, foram desenvolvidos um aplicativo de Realidade Aumentada para interface com usuário, um sistema servidor OPC-UA e um sistema intermediário para integração.

3.1 Arquitetura do sistema desenvolvido

A aplicação desenvolvida, como um todo, se baseia na integração de diversas tecnologias ilustradas na arquitetura exibida na figura 2. O objetivo é levar os dados gerados por um servidor OPC-UA (Controlador Programável de um equipamento industrial) para um broker MQTT, e disponibilizá-los à uma aplicação de Realidade Aumentada.

Figura 2 – Arquitetura física do sistema



Fonte: Elaborado pelo autor

Apesar da aplicação desenvolvida ter sido idealizada com apenas um equipamento industrial e uma aplicação Gateway, a arquitetura pode ser estendida com diversos equipamentos industriais se comunicando via OPC-UA à uma mesma

aplicação Gateway. Neste novo cenário, a aplicação Gateway teria que realizar o trabalho de organizar os dados e publicá-los ao broker de forma adequada, especificando a origem dos dados.

3.1.1 Comunicação entre o servidor OPC-UA e o Gateway OPCUA/MQTT

A ideia de integrar a aplicação móvel com um Controlador Programável compatível com OPC-UA exige a presença de um servidor OPC-UA, que fornece as variáveis na rede, para serem lidas e/ou editadas. Em uma situação real, o servidor OPC-UA é o próprio Controlador Programável, que detém o código responsável por gerir o funcionamento do dispositivo físico. No entanto, existem alternativas para emular o papel do Controlador Programável, como controladores virtuais, e softwares específicos para este fim. Neste projeto foi decidido criar um servidor OPC-UA em Node.js utilizando a linguagem TypeScript, com a biblioteca node-opcua, visando ter maior liberdade nas configurações do servidor.

No servidor OPC-UA foram criados dois códigos, um responsável por realizar as configurações do servidor como nome do servidor, IP, porta, e as variáveis, e outro para realizar o controle da dos eventos discretos da estação. Neste projeto, visando emular o funcionamento de um Controlador Programável, foram criadas 18 variáveis booleanas com acesso à leitura e escrita via OPC-UA, sendo 8 entradas para os sensores da estação, 8 saídas para os atuadores da estação, e uma entrada e uma saída para teste de conectividade e latência.

O sistema foi criado de forma que o servidor OPC-UA possa ser substituído por um outro servidor definido por algum software ou por um controlador programável físico, bastando criar as variáveis OPC-UA seguindo a mesma nomenclatura utilizada neste projeto.

3.1.2 Comunicação entre o Gateway OPCUA/MQTT e o Broker MQTT

Os Controladores Programáveis não costumam ter recursos de conectividade sem fio, mesmo os controladores que operam no padrão OPC-UA. Desta forma, foi desenvolvido, em Node.js com TypeScript, um sistema intermediário com a finalidade de atuar como um Gateway entre o servidor OPC-UA e o aplicativo de Realidade Aumentada.

Para permitir o tráfego dos estados das variáveis booleanas através de comunicação sem fio, foi utilizado o protocolo MQTT, largamente utilizado em sistemas de Internet das Coisas.

De modo geral, este sistema Gateway, estabelece comunicação com o servidor OPC-UA via TCP na rede cabeada local. No experimento, tanto o sistema do servidor OPC-UA quanto o Gateway OPC-UA/MQTT são executados na mesma máquina.

Além disso, o Gateway estabelece comunicação com um broker MQTT Mosquitto, um broker de código aberto, amplamente utilizado em projetos desta natureza, se inscrevendo nos respectivos tópicos das variáveis.

O modo de operação do Gateway é bidirecional, ou seja, realiza tanto a conversão de OPC-UA para MQTT quanto vice-versa, logo, quando uma variável OPC-UA de saída é alterada pelo programa do controlador, esta mudança é detectada pelo Gateway, e imediatamente publica este valor no respectivo tópico do broker MQTT. Por outro lado, quando uma variável de entrada é alterada pelo aplicativo de Realidade Aumentada, sendo publicada em seu respectivo tópico no broker MQTT, o Gateway também detecta tal mudança e imediatamente escreve o novo estado na respectiva variável OPC-UA.

3.1.3 Comunicação entre o Broker MQTT e a aplicação de Realidade Aumentada

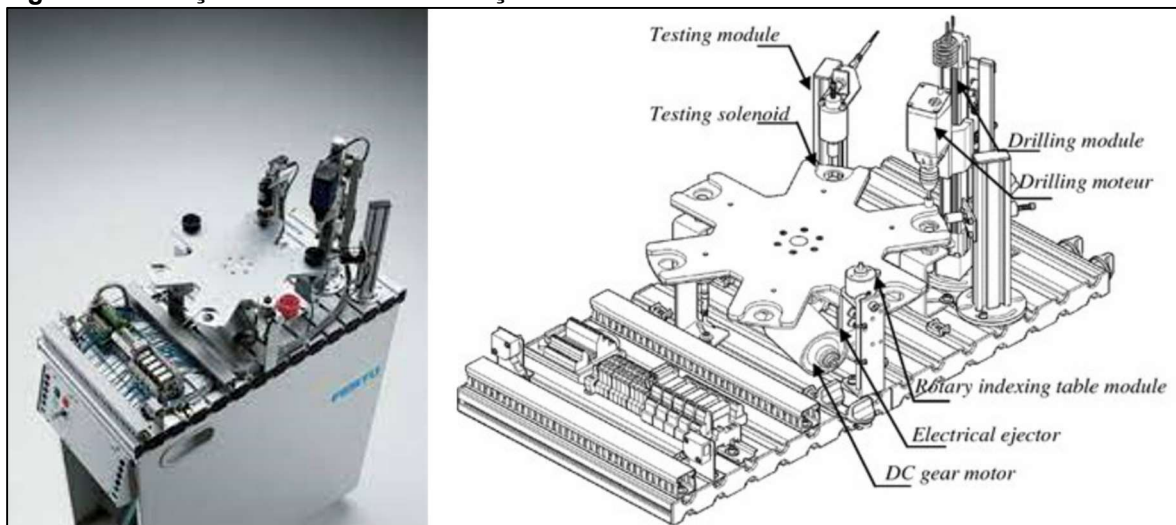
O aplicativo de Realidade Aumentada foi desenvolvido no Unity 3D, operando com diversos scripts desenvolvidos na linguagem C#. O script de comunicação via MQTT, estabelece comunicação com o broker MQTT por meio do IP da rede WiFi. O broker MQTT Mosquitto, por padrão, não permite que aplicações estabeleçam conexão por meio de redes que não sejam a rede local, portanto, nos testes locais a conexão ocorreu sem problemas, porém, ao tentar estabelecer a conexão com o aplicativo pelo WiFi, foi necessário editar as configurações do Mosquitto, para permitir tais conexões externas.

De modo geral, o aplicativo de Realidade Aumentada, após estabelecer conexão com o broker MQTT, se inscreve nos tópicos das variáveis de saídas recebendo os comandos provenientes do controlador, e de acordo com os eventos detectados pelos sensores presentes na cena, publica os novos estados nos seus respectivos tópicos.

3.2 Gêmeo Digital com Realidade Aumentada

O aplicativo de Realidade Aumentada foi desenvolvido com o objetivo de apresentar ao usuário os detalhes relevantes do funcionamento do equipamento físico, que se trata de uma estação modular de automação destinada a realizar uma simulação de furação de uma peça, através da manipulação da peça por meio de uma mesa rotativa indexadora, e a execução de ações em 4 estágios, o estágio inicial, estágio de teste, estágio de furação e estágio de extração da peça.

Figura 3 – Estação modular de automação



Fonte: Adaptado de FESTO

A mesa indexadora possui 6 acomodações para a peça distribuídas igualmente, movida por um motor elétrico rotativo, e controlada pelos sinais de um sensor de proximidade que detecta a rotação de 60° da mesa, ou seja, o controle da mesa se resume em acionar o motor e aguardar que o sensor gere um sinal, desligando então o motor.

O primeiro estágio possui um sensor de proximidade que detecta se uma peça nova foi inserida na mesa.

A peça a ser processada é cilíndrica com um furo central, que deve estar voltado para cima. O segundo estágio é destinado a realizar um teste com o objetivo de verificar se a peça está na posição correta. Tal tarefa é realizada por meio de um atuador elétrico linear, contendo um sensor para verificar se tal atuador está recuado, e um sensor para detectar se a peça está neste estágio. Resumindo, quando a peça é detectada, o atuador avança contra a peça, caso o furo esteja voltado para cima, o sensor do atuador não estará detectando a haste do atuador, indicando que a peça

está na posição correta, e poderá ser processada, caso após o avanço, o sensor detectar a haste, então a peça estará na posição incorreta, não deve ser processada, sendo conduzida diretamente para o último estágio para ser extraída.

No terceiro estágio, a peça é processada por meio de um conjunto composto de uma furadeira, e uma junta prismática vertical, acionada por duas saídas, uma para avançar e uma para recuar, de modo que as duas não devem ser acionadas simultaneamente. O conjunto ainda conta com um par de chaves fim de cursos para detectar se o conjunto está avançado ou recuado. Antes da furação ser realizada, a peça deve ser fixada por meio de um atuador elétrico linear sem sensores. Sendo assim, quando a peça alcança este estágio, e está na posição correta, o atuador de fixação avança, e a furadeira é acionada, o conjunto então avança contra a peça, até alcançar a chave fim de curso inferior, quando retorna até alcançar a chave fim de curso superior. A furadeira então é desligada, e a peça é liberada.

O último estágio é destinado a extrair a peça, e é composta apenas por um atuador elétrico rotativo com rotação limitada em 60° . Quando a peça alcança o último estágio, o atuador é acionado para extrair a peça da mesa, retornando em seguida.

A aplicação de RA foi desenvolvida utilizando recursos do Google AR Core, e o método utilizado para vincular os elementos virtuais no ambiente físico foi o de reconhecimento de planos, ou seja, após a câmera do dispositivo ser aberta, é iniciado o rastreamento do ambiente, gerando uma nuvem de pontos, e a partir dessas nuvens formando planos, tanto horizontais quanto verticais, conforme a Figura 4. Após a criação dos planos, um indicador é exibido, e quando o usuário toca a tela, a estação é instanciada no ponto central do indicador, atrelada ao plano, que é excluído da cena. Deste modo, o usuário pode se mover livremente ao redor da estação, que estará ancorada na posição de origem. O usuário pode reiniciar o processo quando julgar necessário.

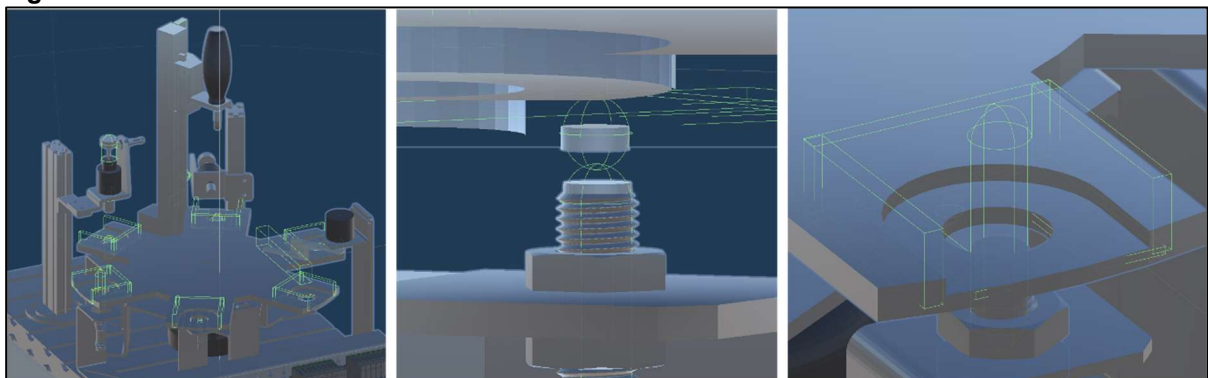
Figura 4 – Reconhecimento de planos



Fonte: Elaborado pelo autor

Para a implementação do funcionamento da estação, foram inseridos colisores na cena com o intuito de promover a movimentação da peça bem como as diversas interações com os atuadores. Para a implementação dos sensores e chaves fim de curso, foram inseridos colisores no modo gatilho (Trigger), que permitem que outros colisores passem por eles, gerando um evento, que no caso significa a detecção do respectivo sensor ou chave fim de curso, conforme a Figura 5.

Figura 5 – Colisores



Fonte: Elaborado pelo autor

Para enriquecer a visualização do funcionamento da estação, foram inseridos botões na tela para acompanhar o estado das entradas, e acionar as saídas de forma individual. Além disso, conforme a Figura 6, foram inseridas na cena, indicadores da posição de cada entrada e saída na estação, que alteram a sua transparência de acordo com seu estado, e foram implementados LEDs nos relês presentes na base da estação, e no controlador presente no painel inferior, que acendem ou apagam de acordo com os estados das suas respectivas entradas e saídas, da mesma forma que ocorre no dispositivo físico.

Figura 6 – Indicadores e LEDs

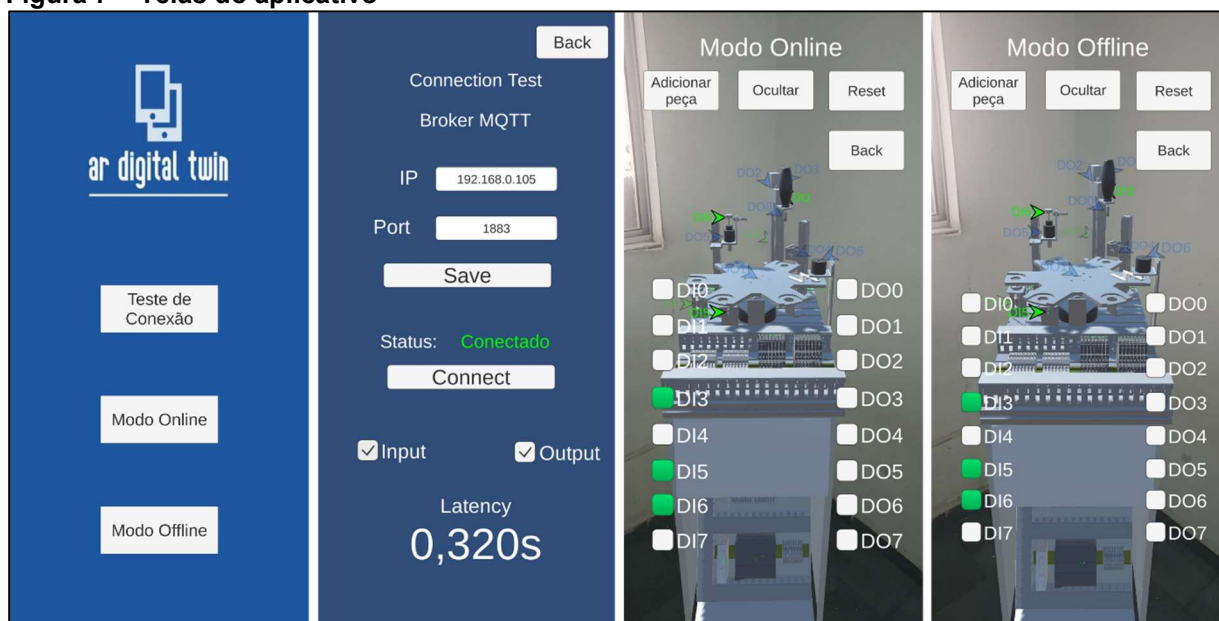


Fonte: Elaborado pelo autor

Considerando que os efeitos sonoros enriquecem a experiência do usuário, foram inseridos áudios com ruídos de funcionamento dos atuadores, que são reproduzidos quando são acionados.

Foram desenvolvidas 3 telas, conforme a Figura 7, que são selecionadas na tela principal, uma tela contendo o modo de exibição offline, uma tela contendo o modo online integrado ao servidor OPC-UA e uma tela contendo recursos para teste de conectividade e latência.

Figura 7 – Telas do aplicativo



Fonte: Elaborado pelo autor

3.2.1 Modo exibição

O modo de exibição offline foi desenvolvido com o propósito de entregar ao usuário uma versão do Gêmeo Digital imersivo sem integração com o servidor OPC-UA, permitindo a utilização do aplicativo sem a necessidade da instalação e execução dos sistemas servidor e Gateway, e dispensando conexão com a internet. Neste modo, a lógica de funcionamento da estação é definida por um código executado no próprio aplicativo. Quando o usuário clica no botão de adicionar peça, o modelo da peça é instanciado na cena na posição do primeiro estágio podendo estar com o furo central voltado para cima ou para baixo. O que define a posição da peça é uma variável booleana aleatória, ou seja, de acordo com o valor dessa variável, às vezes a peça será instanciada com o furo voltado para cima, às vezes voltado para baixo. Desse modo é possível testar o comportamento da lógica de funcionamento da estação.

3.2.2 Modo espelhamento de entradas e saídas de Controlador Programável virtual

O modo de espelhamento das entradas e saídas se trata do modo online do aplicativo. Caso os sistemas do servidor OPC-UA e Gateway estejam sendo

executados, e o aplicativo esteja conectado na mesma rede WiFi, este modo funcionará normalmente.

O funcionamento deste modo é similar ao modo offline, no entanto, ao invés do código que controla os eventos discretos da estação estar sendo executado no próprio aplicativo, este será executado no servidor OPC-UA que representa o Controlador Programável físico. Sendo assim, quando a peça é instanciada, o sinal do sensor que detecta a peça é publicado no seu respectivo tópico no broker MQTT, que por consequência reflete na respectiva variável OPC-UA pela ação do Gateway. O código do servidor OPC-UA processa o sinal, e gera sinais de saída fazendo o caminho inverso, e refletindo no aplicativo de Realidade Aumentada.

3.2.3 Tela de configuração e teste de conectividade

A comunicação do aplicativo com o broker MQTT deve ser configurada, em especial o IP e a porta da rede. Para facilitar futuras alterações de IP e porta, foi desenvolvida uma tela de configuração, onde o usuário pode inserir os parâmetros e salvar na memória do dispositivo, logo, uma vez configurada, a conexão se manterá. Além disso, foi inserida uma indicação do estado do aplicativo, se está conectado ou desconectado, e um botão para realizar a conexão quando necessário.

A comunicação no modo online, do aplicativo com o Gateway e servidor, assim como qualquer transmissão de dados, ocorre apresentando um atraso de resposta denominado de latência, que é prejudicial neste sistema, atrasando a resposta dos atuadores da estação virtual após o envio dos sinais dos sensores. Para mensurar a latência foram inseridos nesta tela um botão de entrada e um de saída, de modo que quando o usuário pressiona o botão de entrada, o sinal trafega até o servidor, passando pelo Gateway, e o servidor devolve o sinal em uma saída de teste, que trafega de volta até o aplicativo. O período, em segundos, entre o envio do sinal e o recebimento da resposta é mensurado e exibido na tela.

3.3 Testes realizados

Com o sistema desenvolvido foram realizados alguns testes, visando observar a estabilidade do vínculo dos elementos virtuais no ambiente físico, o comportamento dos colisores e efeitos da latência. Para a realização dos testes, o servidor OPC-UA

e o Gateway foram instalados localmente enquanto o aplicativo foi instalado em um smartphone Android.

3.3.1 Teste do modo exibição (modo offline)

O teste do modo offline foi realizado em um ambiente físico livre de objetos. Após a câmera do smartphone ser aberta, o reconhecimento de planos detectou o chão, e tocando no indicador de posição, a estação foi instanciada, permitindo o usuário se mover ao seu redor. Clicando o botão para instanciar a peça, o processo foi iniciado, e de acordo com a posição da peça, a peça foi processada ou excluída.

3.3.2 Teste do modo Controlador Programável virtual (modo online)

O teste do modo online, se iniciou com a configuração dos IPs e portas tanto do servido OPC-UA quanto do Gateway, seguindo pela execução dos dois sistemas utilizando o terminal do Visual Studio Code. No sistema servidor OPC-UA foram inseridos logs para indicar que o servidor está ativo, e para indicar a fase atual do controle dos eventos discretos para acompanhar o andamento. No sistema Gateway foram inseridos logs para indicar que o sistema está ativo, e para indicar todas as alterações de variáveis que ocorreram. Desse modo é possível monitorar os dois sistemas durante o funcionamento.

Quanto ao aplicativo de Realidade Aumentada, o teste foi similar ao teste offline, com a diferença que antes de iniciar o processo foi realizada a configuração da rede, e confirmada que a conexão com o broker está ativa. Sendo assim, o teste ocorreu da mesma forma, detectando o plano, instanciando a estação, e adicionando uma peça para iniciar o processo.

4 RESULTADOS OBTIDOS

A estabilidade dos elementos virtuais no ambiente físico foi satisfatória, permitindo que o usuário se movesse no ambiente, inclusive se afastando consideravelmente da estação, e mesmo assim, os elementos virtuais se mantiveram ancorados na posição em que foi instanciada, com poucas ocorrências em que os elementos virtuais se moveram com relação ao piso.

Foi possível observar que em ambos os testes, os colisores apresentaram respostas satisfatórias, com exceção para alguns casos isolados, onde a peça se

movia, e ficava em posição incorreta, mas no geral as colisões entre a peça e a mesa, entre a peça e o atuador de teste, entre a peça e o atuador de extração, e todos os sensores funcionaram bem.

Quanto ao atraso na transmissão de dados, realizando testes pela tela de configuração e testes, foi observada uma latência variável entre 0,2s e 1,4s, mesmo operando no modo QoS 0 (Quality of Service), onde o dado é enviado apenas uma vez pelo protocolo MQTT sem confirmação, o que tornou o sistema online instável em um primeiro momento. Foi observado que o movimento mais prejudicado pela latência foi a rotação da mesa indexadora, pelo fato de depender do sinal do seu sensor para ter o seu movimento interrompido, posicionando a peça na posição adequada. Na ocorrência de alta latência, ocorreu um atraso de desligamento da mesa mais significativo, causando um desalinhamento da peça com relação à posição desejada. Para mitigar tal efeito, a velocidade da mesa foi reduzida, e os efeitos da latência foram suprimidos, melhorando a precisão de posicionamento da peça.

5 CONSIDERAÇÕES FINAIS E TRABALHOS FUTUROS

De forma geral, o objetivo de entregar um aplicativo de Gêmeo Digital imersivo foi alcançado, tornando a experiência de interação com uma versão virtual de um dispositivo físico de modo satisfatório.

Ficou claro que a biblioteca Google AR Core no Unity 3D se mostrou estável, apresentando uma experiência do usuário interessante na interação com os elementos virtuais.

A comunicação entre o servidor OPC-UA e o aplicativo de Realidade Aumentada pelo protocolo MQTT de mostrou estável, com a ressalva da latência, que impactou diretamente no funcionamento da estação de automação virtual. No entanto, vale ressaltar que em uma situação de integração com um Controlador Programável físico controlando uma estação de automação física, este inconveniente não terá efeitos significativos, uma vez o processamento dos eventos discretos da estação será realizado localmente no equipamento, sendo que os sinais dos sensores físicos não serão enviados pela rede com o intuito de controlar algum sistema, mas simplesmente para visualização na interface, enquanto os sinais de saída gerados pelo controlador será enviado tanto para o dispositivo físico quanto para a aplicação em RA, portanto, em um situação deste tipo, a aplicação RA iria imitar o equipamento real, e a latência

não iria impactar no funcionamento da estação real, o impacto da latência seria apenas no atraso da atualização das animações no aplicativo de RA.

Concluindo, o objetivo de desenvolver uma aplicação de Gêmeo Digital imersivo com Realidade Aumentada, que permita o espelhamento de estados de um equipamento, e concede aos usuários uma interface contínua e intuitiva do Gêmeo Digital, foi alcançado.

Quanto aos próximos passos, naturalmente seria adaptar o sistema desenvolvido para ser validado em um equipamento físico, além de realizar ajustes para aprimorar o realismo dos elementos virtuais presentes na cena como materiais, texturas e iluminação, e aprimorar os efeitos sonoros, em especial criar sons específicos para aceleração e desaceleração dos motores.

Quanto aos trabalhos futuros, seria interessante expandir o conceito para Gêmeos Digitais de outros equipamentos como robôs colaborativos, que ultimamente demandam uma interface com os programadores e operadores cada vez mais precisa e interativa, e implementar funcionalidades de registro de informações do funcionamento do equipamento, para futuras análises e tomadas de decisão, sendo um dos fatores importantes do Gêmeo Digital no contexto da Indústria 4.0.

REFERÊNCIAS

BUCHGEHER, Georg et al. **Migrating Cyber-Physical Systems to OPC UA**. Procedia Computer Science, v. 200, p. 276-283, 2022.

DE PAULA FERREIRA, William; ARMELLINI, Fabiano; DE SANTA-EULALIA, Luis Antonio. Simulation in industry 4.0: A state-of-the-art review. **Computers & Industrial Engineering**, v. 149, p. 106868, 2020.

GRIEVES, Michael. Digital twin: manufacturing excellence through virtual factory replication. White paper, v. 1, n. 2014, p. 1-7, 2014.

HiveMQ; **MQTT Essentials**. Disponível em: <https://www.hivemq.com/blog/mqtt-essentials-part-1-introducing-mqtt/> Acesso em 23 de agosto 2023.

MCKENDRICK, R.; **Mastering Docker**. 4ª edição. Birmingham. Packt Publishing. 2020.

JO, Guejong; JANG, Su-Hwan; JEONG, Jongpil. **Design and implementation of cpps and edge computing architecture based on opc ua server**. Procedia Computer Science, v. 155, p. 97-104, 2019.

KRITZINGER, Werner et al. Digital Twin in manufacturing: A categorical literature review and classification. **Ifac-PapersOnline**, v. 51, n. 11, p. 1016-1022, 2018.

OASIS. **MQTT Version 5 Specification**. 2019. Disponível em: <https://docs.oasis-open.org/mqtt/mqtt/v5.0/mqtt-v5.0.html>. Acesso em: 25 agosto 2023

OPC FOUNDATION; **OPC Unified Architecture Interoperability for Industrie 4.0 and the Internet of Things**. Disponível em: <https://opcfoundation.org/wp-content/uploads/2023/05/OPC-UA-Interoperability-For-Industrie4-and-IoT-EN.pdf>. Acesso em: 23 agosto 2023.

OPC FOUNDATION. **What is OPC?**. 2023. Disponível em: <https://opcfoundation.org/about/what-is-opc/>. Acesso em: 25 agosto 2023

ROCHA, Murilo Silveira et al. **On the performance of OPC UA and MQTT for data exchange between industrial plants and cloud servers**. Acta IMEKO, v. 8, n. 2, p. 80-87, 2019.

SILVA, Daniel et al. **A performance analysis of internet of things networking protocols: Evaluating MQTT, CoAP, OPC UA**. Applied Sciences, v. 11, n. 11, p. 4879, 2021.

TORI, Romero; DA SILVA HOUNSELL, Marcelo. Introdução a realidade virtual e aumentada. **Sociedade Brasileira de Computação**, 2020.

ZHU, Zexuan; LIU, Chao; XU, Xun. Visualisation of the digital twin data in manufacturing by using augmented reality. **Procedia Cirp**, v. 81, p. 898-903, 2019.

ZVEI. **Status Report. Reference Architecture Model Industrie 4.0 (RAMI4.0)**. 2015. Disponível em: <https://www.zvei.org/en/press-media/publications/gma-status-report-reference-architecture-model-industrie-40-rami-40>. Acesso em: 27 agosto 2023