INSTITUTO TECNOLÓGICO DE AERONÁUTICA

Marlon Wendrer Jacinto

CONSTRUCTION OF A METAVERSE INTEGRATED WITH DIGITAL TWINS AND PHYSICAL SENSORS

Final Paper
2025

Course of Aeronautical Engineering

CDU 681.3

Marlon Wendrer Jacinto

CONSTRUCTION OF A METAVERSE INTEGRATED WITH DIGITAL TWINS AND PHYSICAL SENSORS

Advisor

Maj. Av. Lucas Oliveira Barbacovi (ITA)

Co-advisor

Prof. Dr. Christopher Shneider Cerqueira (ITA)

AERONAUTICAL ENGINEERING

São José dos Campos
INSTITUTO TECNOLÓGICO DE AERONÁUTICA

2025

Cataloging-in Publication Data
Documentation and Information Division

Jacinto, Marlon Wendrer
Construction of a Metaverse Integrated with Digital Twins and Physical Sensors / Marlon Wendrer Jacinto.
São José dos Campos, 2025.
38p.

Final paper (Undergraduation study) – Course of Aeronautical Engineering– Instituto Tecnológico de Aeronáutica, 2025. Advisor: Maj. Av. Lucas Oliveira Barbacovi. Co-advisor: Prof. Dr. Christopher Shneider Cerqueira.

1. Metaverse. 2. Virtual Reality. 3. Digital Twins. I. Instituto Tecnológico de Aeronáutica. II. Construction of a Metaverse Integrated with Digital Twins and Physical Sensors.

BIBLIOGRAPHIC REFERENCE

JACINTO, Marlon Wendrer. **Construction of a Metaverse Integrated with Digital Twins and Physical Sensors**. 2025. 38p. Final paper (Undergraduation study) – Instituto Tecnológico de Aeronáutica, São José dos Campos.

CESSION OF RIGHTS

Marlon Wendrer Jacinto
Rua H8A, Ap. 127
12.228-460 – São José dos Campos–SP

CONSTRUCTION OF A METAVERSE INTEGRATED WITH DIGITAL TWINS AND PHYSICAL SENSORS

This publication was accepted like Final Work of Undergraduation Study

Marlon Wendrer Jacinto

Author

Maj. Av. Lucas Oliveira Barbacovi (ITA)

Advisor

Christopher Shneider Cerqueira (ITA)

Co-advisor

São José dos Campos: NOVEMBER 07, 2025.

My deepest gratitude to my family, friends, and everyone who was part of this long journey, for being present from the start until today.

Acknowledgments

I would like to begin by thanking my parents, Valdeli Jacinto and Maria Aparecida, for their unconditional support throughout my entire journey. Without your love, advice, and constant support, this path would have been much more difficult.

I also thank Turma 25 for the friendship, companionship, and resilience. We were together in happy and challenging times, and this bond made my academic life lighter, more fruitful, and memorable.

To Professor Christopher Shneider Cerqueira, I register my deep gratitude for the teachings shared throughout the course and for the careful and dedicated guidance during the completion of this work.

I thank my colleagues Mathias, Gustavo Gomes, José, and Barney for their friendship, the good times, and the undergraduate years; even in difficult moments, we were together.

"Ceux qui savent peu parlent beaucoup, et ceux qui savent beaucoup parlent peu."

— Jean-Jacques Rousseau

Abstract

This work presents a functional architecture for integrating the metaverse, digital twins, and physical sensors connected via the Internet of Things (IoT). The proposed system consists of a 3D immersive virtual environment developed in Unity, capable of real-time reflection of physical data such as luminosity. Using lightweight communication protocols like MQTT, the project ensures synchronization and responsiveness between physical devices and the digital model. The implementation offers a practical evaluation of the technical challenges involving latency, fidelity, and scalability, contributing to immersive applications in automation, urban monitoring, and predictive simulations. The results indicate that the solution is feasible and provides a replicable model for future use in educational, industrial, and urban contexts.

Resumo

Este trabalho apresenta uma arquitetura funcional para integração entre o metaverso, gêmeos digitais e sensores físicos conectados via Internet das Coisas (IoT). A proposta consiste no desenvolvimento de um ambiente virtual tridimensional imersivo, modelado na plataforma Unity, que reflete em tempo real os dados coletados por sensores físicos de luminosidade. Utilizando protocolos de comunicação leve, como o MQTT, foi possível garantir sincronização e responsividade na comunicação entre os dispositivos físicos e o modelo virtual. A implementação propicia uma análise prática dos desafios técnicos relacionados à latência, fidelidade de representação e escalabilidade, contribuindo para o avanço de aplicações imersivas em automação, monitoramento urbano e simulações preditivas. Os resultados indicam que a solução é viável e oferece um modelo replicável para futuras aplicações em contextos educacionais, industriais e urbanos.

List of Figures

List of Tables

List of Abbreviations and Acronyms

AI Artificial Intelligence.

AR *Augmented Reality*.

BIM *Building Information Modeling*.

IoT *Internet of Things*.

MQTT *Message Queuing Telemetry Transport*.

URP *Universal Render Pipeline*.

VR *Virtual Reality*.

Contents

1 Introduction

1.1 Motivation

The rapid digitalization of the global economy, coupled with the spread of emerging technologies like *Internet of Things* (IoT), Artificial Intelligence (AI), *Augmented Reality* (AR), and *Virtual Reality* (VR), is profoundly changing the way we interact with the environment around us. The concept of the *metaverse* has been widely explored as a three-dimensional, interactive, and immersive virtual environment that can deliver highly realistic experiences in real time (CAN..., 2023; Lv; Shang; Guizani, 2022).

At the same time, *digital twins* are being applied across diverse sectors, including smart cities, healthcare, logistics, and manufacturing, as dynamic replicas of physical systems. They allow for continuous monitoring and data-driven decisions, faithfully and responsively reflecting the state of real-world assets (Atalay, 2023; HOW..., 2021).

FIGURE 1.1 – An artistic illustration of an immersive virtual urban environment, representing a possible metaverse experience that integrates virtual reality, avatars, and digital simulations.

Source: Stacked Homes, 2021 (Homes, 2021)

The convergence between these two emerging fields, strongly supported by the use of physical sensors connected via AI, is ushering in a new paradigm: the possibility of virtual environments that don't just simulate, but actually react to the physical world in real time. This level of integration offers novel opportunities for applications in automation, predictive simulations, and smart urban management (Masubuchi; Ito; Nakamura, 2025; PIVOTAL..., 2024; SEMANTIC-AWARE..., 2023).

1.2 *Case Study: 51World*

The Chinese company 51World has stood out in developing digital twins of entire cities, applying technologies like geospatial data, BIM models, IoT sensors, and high-fidelity graphics engines like the Unreal Engine. The company has already developed digital replicas of cities like Shanghai and Singapore, with the goal of transforming how buildings, transport, and urban landscapes are designed and operated (Ravenscroft, 2021).

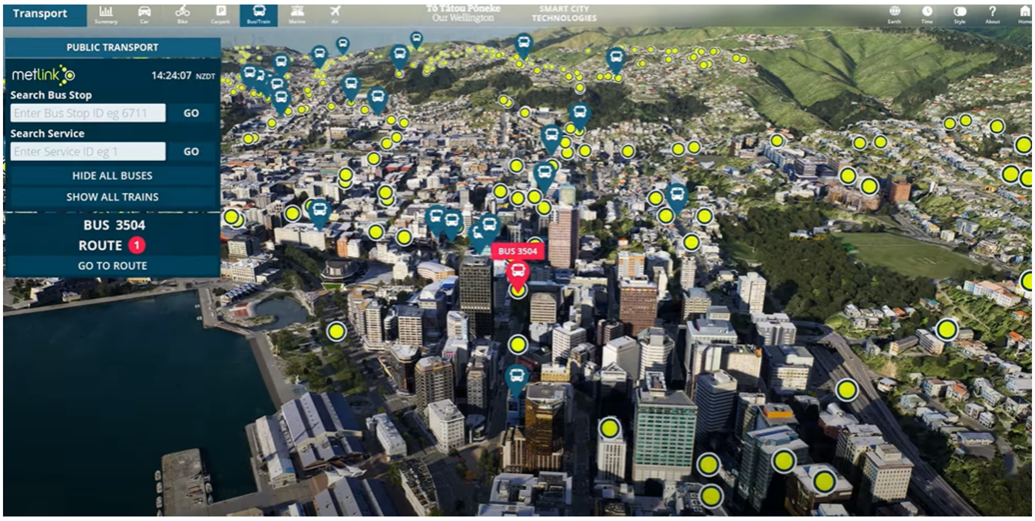

FIGURE 1.2 – An interactive 3D representation of an urban district using the digital twin platform developed by 51World. The image demonstrates the application of geospatial data, BIM models, and real-time visualization in a simulated urban environment.

Source: Adapted from 51World, 2021 (Ravenscroft, 2021)

Moreover, these digital twins offer interactive 3D visualizations of cities that can be updated in real time with sensor data, allowing for traffic simulations, energy consumption assessment, emergency response, and smart urban planning. In addition to supporting public governance, these platforms also enable applications in sectors like construction, urban mobility, and sustainability.

1.3 The Problem

While advances in digital technology have enabled innovations in virtual reality, digital twins, and the Internet of Things, significant technical challenges remain when it comes to efficiently integrating interactive environments with physical sensors. One of the main hurdles is the need for robust and scalable architectures that can handle real-time data collection, low latency, and a faithful digital representation of the physical environment (A..., 2024; CHALLENGES..., 2024). Beyond that, precise synchronization between multiple sensor source, ensuring data reaches the digital model with consistent timestamps, is recognized as a fundamental requirement for reactive digital twins to work properly (DIGITAL..., 2024; Ward *et al.*, 2023).

These technical requirements are confirmed in the literature: studies on the "real-time metaverse" point to low latency, high bandwidth, and synchronization as critical factors for the system's immersion and fidelity (A..., 2024). Moreover, in the digital twin field, keeping virtual models updated depends directly on a delay-free integration of the data flow (DIGITAL..., 2024).

1.4 Objectives

1.4.1 General Objective

To develop an interactive virtual environment within the metaverse, integrated with a digital twin system, which receives real-time data from physical sensors and faithfully reflects changes in the physical environment.

1.4.2 Specific Objectives

The development of this project will be guided by a set of interconnected actions aimed at consolidating the proposed integration of the metaverse, digital twins, and physical sensors. First, a 3D virtual environment will be modeled using the Unity game engine, seeking to accurately reflect a real physical space. In parallel, inputs will be selected to simulate the output of physical sensors capable of capturing environmental variables like temperature, humidity, light, and motion, using accessible platforms such as Arduino, Raspberry Pi, or similar IoT-supported devices.

The collected data will be connected to the virtual environment using an efficient and lightweight communication architecture, adopting protocols like MQTT, which is widely used in applications with real-time transmission requirements. Finally, the system will be submitted to performance and responsiveness tests to verify its ability to faithfully reflect, in the digital model, the changes occurring in the physical environment. This approach will allow for an evaluation of not only the technical feasibility of the proposed solution but also its potential for real-world scenarios.

1.5 Expected Contributions

The expected contribution of this work is to present a functional and replicable architecture that effectively integrates a virtual metaverse environment with a digital twin system fed by real-time physical sensors. The proposed implementation is expected to offer practical solutions for the main communication and synchronization challenges between the physical and digital worlds, based on accessible technologies and established protocols.

Additionally, the project aims to provide a critical assessment of the system's performance, identifying its limitations and opportunities for improvement. Ultimately, the goal is to establish theoretical and operational foundations that can contribute to future applications in environmental automation, intelligent monitoring, and immersive simulations within educational, urban, or industrial contexts.

2 Theoretical Foundation

2.1 The Unity Platform

As a highly flexible, cross-platform 3D engine, Unity has become a popular foundation for developing immersive metaverse environments. It facilitates the creation of detailed virtual worlds featuring physical simulation, real-time rendering, and support for extended reality (XR). This versatility has been explored in both industry and academia. A relevant example is the development of a virtual university campus model using Unity, aimed at integrating teaching, collaboration, and simulation activities into a unified metaverse environment (Luo *et al.*, 2024).

The *engine* has native support for languages like C#, making it easier to implement *scripts* that interpret data from physical sensors using protocols like MQTT. This allows changes in the real environment—such as temperature, light, or presence—to be immediately reflected in the virtual representation.

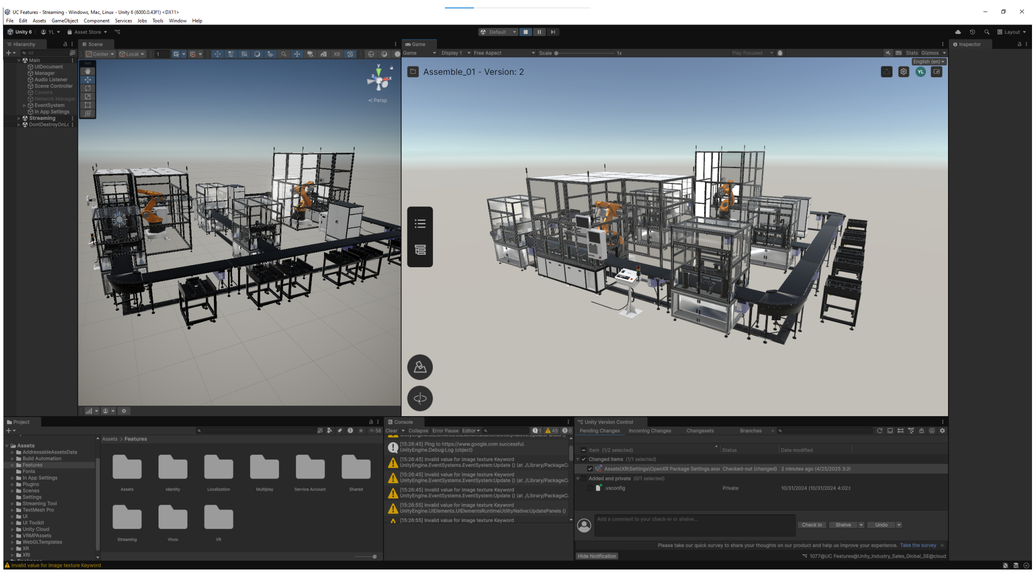

FIGURE 2.1 – Unity IDE interface (Industry Viewer template) showing the workspace layout with multiple specialized views (Scene, Hierarchy, Inspector, Project, and Console).

Source: Adapted from Unity Technologies (2025) (Technologies, 2025).

Beyond just visualization, Unity enables the development of advanced features, such as dynamic behavior simulations, integration with augmented and virtual reality, and support for multiple simultaneous instances within the metaverse. As such, Unity acts not just as a viewer, but as a fundamental part of the infrastructure for interacting with digital twins and IoT.

2.2 MQTT as a Protocol for IoT and Virtual Environment Communication

MQTT is a lightweight communication protocol based on the publish/subscribe (*publish/subscribe*) model, widely adopted in IoT applications due to its efficiency on networks with limited bandwidth and variable latency (Arnold *et al.*, 2019). Its centralized architecture, featuring a *broker* that manages message exchange, allows physical sensors to publish data to specific topics. Meanwhile, client applications, like a virtual environment built in Unity, can subscribe to receive this information in real time.

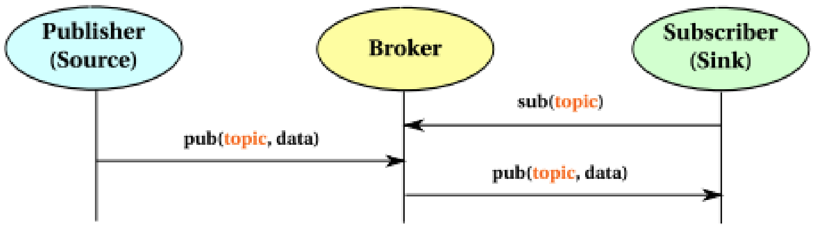

FIGURE 2.2 – Topic-based communication model of the MQTT protocol, where the *Publisher* sends data to a topic, the *Broker* manages the message exchange, and the *Subscriber* receives the data by subscribing to the topic.

Source: Adapted from (Arnold *et al.*, 2019)

This characteristic makes MQTT ideal for building digital twins that require continuous updates on the state of the physical world. It ensures a synchronized and interactive simulation, which is fundamental to developing the integrated metaverse proposed in this work.

2.3 Internet of Things (IoT)

IoT represents a technological revolution based on the interconnection of physical objects with the internet, allowing embedded devices—such as sensors, actuators, and microcontrollers—to collect, transmit, and process environmental data in real time. This capability enables task automation, remote system monitoring, and the integration of the physical and digital worlds.

The IoT architecture is generally composed of three layers: perception, network, and application. The perception layer involves the devices that collect environmental data; the network layer is responsible for transmitting this data using protocols like MQTT; and the application layer transforms the data into useful information, triggering automated responses or presenting visualizations to the user.

In the context of this work, IoT is fundamental for enabling communication between the physical environment and its virtual counterpart, or digital twin. Sensors installed in a real physical space capture data such as temperature, humidity, and light, which are then transmitted to a digital environment modeled in a game engine. This integration is essential for building interactive and responsive experiences in the metaverse.

IoT is, therefore, the foundation of sensing and real-time connectivity—key elements for any system seeking to integrate digital twins with immersive and interactive environments (Gubbi; Buyya; Marusic; Palaniswami, 2013).

2.4 Digital Twins

Digital Twins (*Digital Twins*) are an emerging technology that provides a precise digital representation of a physical object, system, or process. They have the ability to simulate, monitor, and predict real-world behavior based on data collected in real time (Grieves, 2014). This approach is characterized by the continuous interconnection between the physical and virtual worlds, allowing for the dynamic synchronization of states and operational variables.

Initially adopted in sectors like aerospace and advanced manufacturing, digital twins are now widely used in various fields, such as smart cities, healthcare, energy, and construction. Their application makes it possible to increase operational efficiency, reduce costs through predictive maintenance, and support data-driven decision-making (Mourtzis, 2023).

A digital twin is typically composed of three main layers:

1. **The real physical model:** the existing system or equipment in the real world.
2. **The digital representation:** the computational model that replicates the behavior and characteristics of the physical object.
3. **The bidirectional communication channel:** the infrastructure that allows the continuous exchange of data between the two domains, often via IoT and protocols like MQTT.

In the context of integration with game engines and virtual environments, digital twins allow for the immersive and interactive visualization of physical processes, providing greater intelligibility of the system's behavior in real time. 3D modeling, combined with sensor data from IoT devices, enriches the simulated experience, fostering a virtual representation that reacts instantly to changes in the real environment (Masubuchi; Ito; Nakamura, 2025).

2.5 The Metaverse

The metaverse can be defined as an immersive and persistent three-dimensional digital environment, where multiple users interact with each other and with digital objects through virtual representations. Unlike isolated simulations, the metaverse seeks to integrate multiple technologies—such as AR, VR, IoT, AI, and 5G networks—to provide continuous and interoperable experiences (Lee *et al.*, 2021).

Its architecture generally involves creating highly responsive virtual worlds where data from the physical world can be incorporated in real time, fostering new ways of visualizing, controlling, and analyzing real environments through digital twins. This is possible thanks to the bidirectional communication between physical sensors and the digital environment, which promotes dynamic updates in the virtual model as the monitored environment changes.

In the context of this work, the metaverse is used as an interactive 3D visualization platform, coupled with real sensors through asynchronous communication protocols like MQTT. The Unity game engine will be responsible for rendering the digital environment, while the physical sensors will allow for the continuous update of the virtual space in response to variations in the physical world.

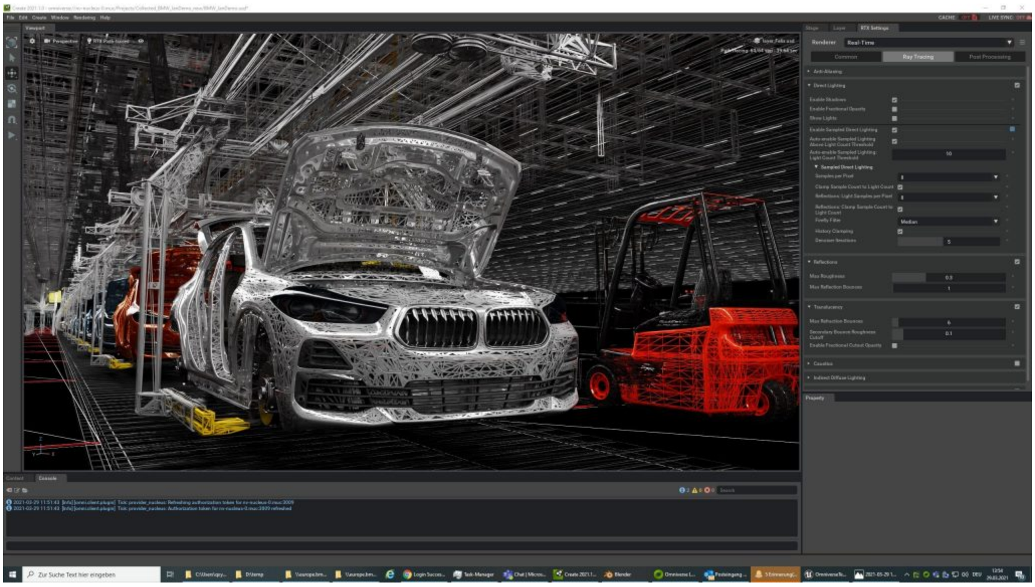

FIGURE 2.3 – BMW's digital factory built on NVIDIA Omniverse. The image represents a complete digital twin of the factory environment, used for production simulations, layout reorganization, and testing human-robot interactions.

Source: Adapted from Time, 2021 (HOW..., 2021)

A notable example of this convergence between digital twins and the metaverse is the case of BMW, which, in partnership with NVIDIA, developed complete digital replicas of its factories using the Omniverse platform. In these virtual environments, engineers and operators can simulate production processes, reorganize industrial layouts, and visualize human-robot interactions in real time, based on data captured from the physical world. The initiative represents a significant step toward industrial metaverses, where immersive, digital-twin-based collaboration improves efficiency, reduces errors, and accelerates the development of industrial solutions (HOW..., 2021).

3 Methodology

This section details the research design, the selected technologies, and the experimental procedures used to develop and validate the integration system between a Digital Twin in Unity and a simulated IoT device.

3.1 Development Phases

The project's development was structured incrementally, allowing for independent testing and validation of each module before the final integration. The six main stages were:

1. **Architecture modeling:** defining the scope, communication requirements, and choice of technologies and protocols;
2. **IoT node development:** creating and programming the embedded circuit in a simulation environment (Wokwi);
3. **Communication channel setup:** establishing the MQTT broker and validating message traffic (MQTT Explorer);
4. **Digital twin development:** creating the interactive 3D environment and communication logic in the Unity *engine*;
5. **Integration testing:** performing functional tests to verify the bidirectional synchronism between the simulator and the virtual environment;
6. **Validation and performance analysis:** measuring the system's latency, stability, and bidirectionality.

This approach was key to efficiently isolating and debugging issues, ensuring that each component (simulated *hardware*, *broker*, *software*) was functional before the full system integration.

3.1.1 Wokwi Simulator

Wokwi is an online simulator for embedded systems, widely used in IoT prototyping projects. It reproduces the physical and electrical behavior of components like microcontrollers, sensors, and LEDs. It was chosen because it allows for rapid, reproducible, and low-cost testing, eliminating the complexities associated with assembling physical prototypes and reducing project costs.

Throughout the project, Wokwi was used to simulate an *ESP32 DevKit V1* connected to an LED, which functions as a representation of a physical actuator.

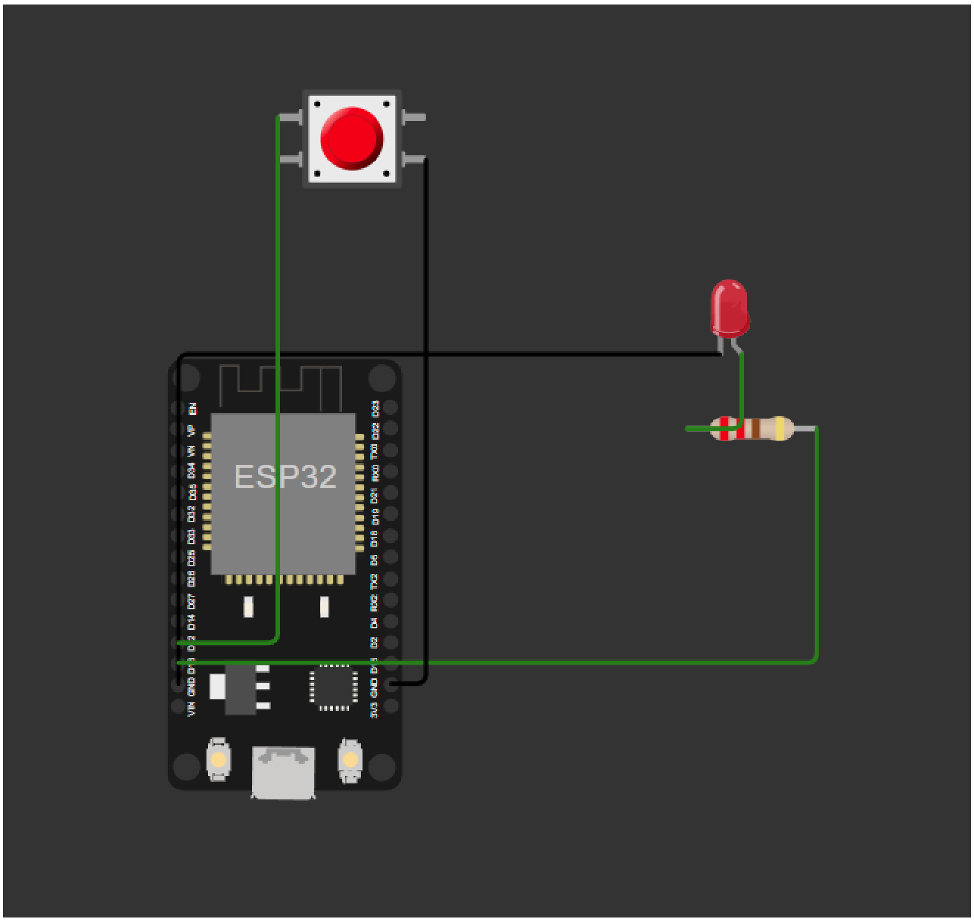

FIGURE 3.1 – Simulation of the ESP32 and LED with button activation in the Wokwi environment.

The simulator provided a reliable virtual environment to test the MQTT communication logic and device behavior without needing real components, making the validation process faster and more accessible.

3.1.1.1 IoT Node: ESP32

The *ESP32* microcontroller was chosen as the central unit for the IoT node because it combines features that make it widely used in Internet of Things applications. Its low cost, efficient performance, and native Wi-Fi support enable direct communication with local networks and, consequently, with the MQTT broker. This integration eliminates the need for intermediate *gateways*, simplifying the system architecture and making the development process faster and more accessible.

The embedded code (`sketch.ino`), developed in the Arduino language (C/C++), used the `WiFi.h` and `PubSubClient.h` libraries, which are responsible for network connectivity and MQTT communication. The embedded logic included:

- `reconnect()` function: automatic re-establishment of the MQTT connection if disconnected;
- `callback()` function: interpretation of messages received from the command topic;
- Periodic publishing of the LED's state to the *status* topic.

3.1.2 Protocol and Broker: MQTT

For the communication layer, the MQTT protocol was chosen. This choice is based on its core characteristics for the IoT world: it is a lightweight protocol (low overhead), asynchronous, and based on the *publish/subscribe* standard.

This model decouples the clients (the ESP32 and Unity), which do not need to know about each other directly. Communication is mediated by a *Broker*, which manages the distribution of messages into topics. For this experiment, the public *broker* `test.mosquitto.org`, maintained by the Eclipse Foundation, was used because it eliminates the need to configure a local MQTT server, speeding up the prototyping phase.

The topic architecture was defined as shown in Table 3.1:

TABLE 3.1 – Structure of MQTT topics used

Topic	**Origin**	**Function**
`esp32/comando`	Unity	Publication of commands (ON/OFF)
`esp32/status`	ESP32	Publication of the LED's current state

3.1.3 Debugging Tool: MQTT Explorer

MQTT Explorer, shown in Figure 3.2, is an open-source graphical MQTT client used in this project as a diagnostic and debugging tool. Its function was to validate the integrity and flow of messages between the clients and the *broker*.

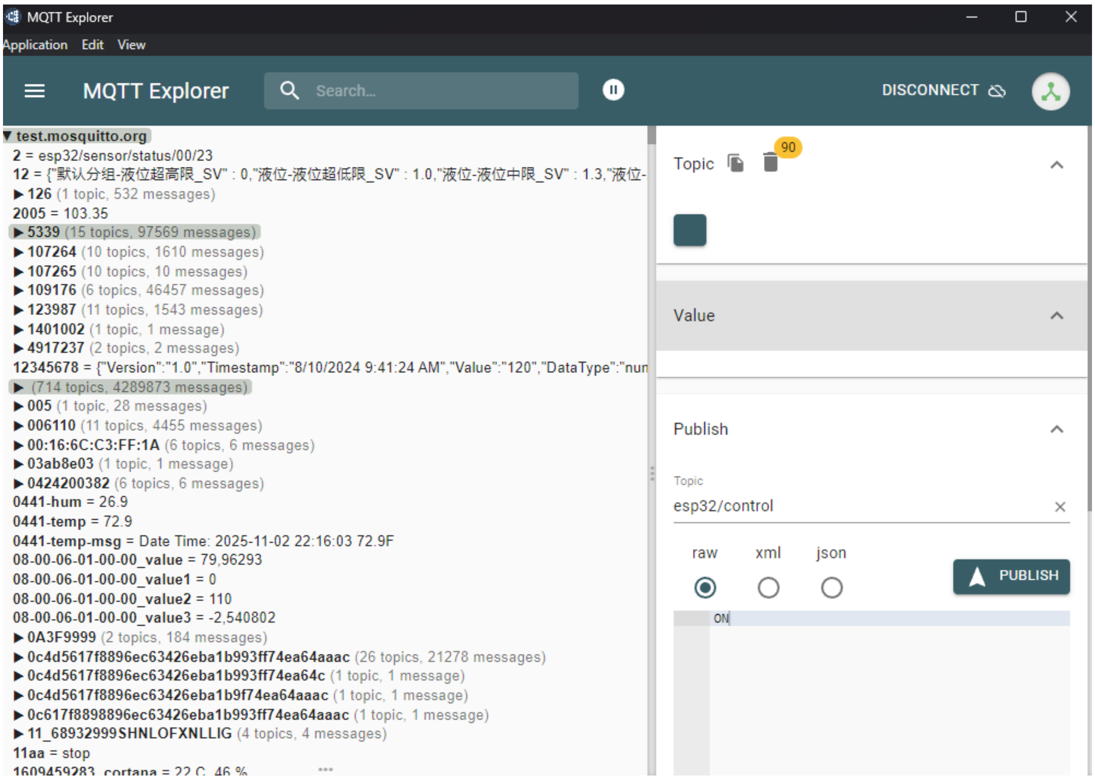

FIGURE 3.2 – MQTT Explorer.

Using this tool, it was possible to monitor the *payloads* (data) published to the status and command topics in real time, verifying that the ESP32 and Unity were sending and receiving data correctly. The tool was essential for identifying connection failures or message formatting issues before the final integration, and it allowed for measuring communication latency during tests.

3.1.4 Digital Twin Platform: Unity 3D

In this project, Unity was used to create a 3D scene containing a visual representation (a sphere with a light component), whose state (color, light intensity) is synchronized in real time with the state of the simulated LED in Wokwi.

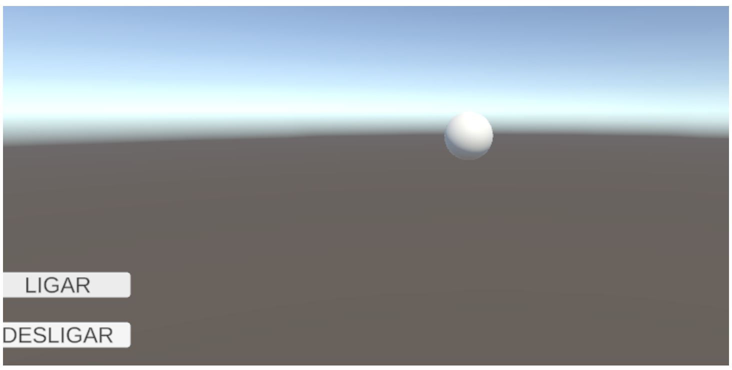

FIGURE 3.3 – Unity environment built for the project.

The Unity environment was configured with the pipeline, as shown in Figure *Universal Render Pipeline* (URP), and included:

- A 3D object representing the lamp with a light component;
- The `MQTT_Manager.cs` manager *script*;
- A graphical interface with *"Ligar"* (Turn On) and *"Desligar"* (Turn Off) buttons.

3.1.4.1 The Lamp

In Unity, the lamp was represented by a 3D *sphere* object, as shown in Figure 3.4, configured to simulate lighting behavior when activated, with the behavior of disappearing when turned off.

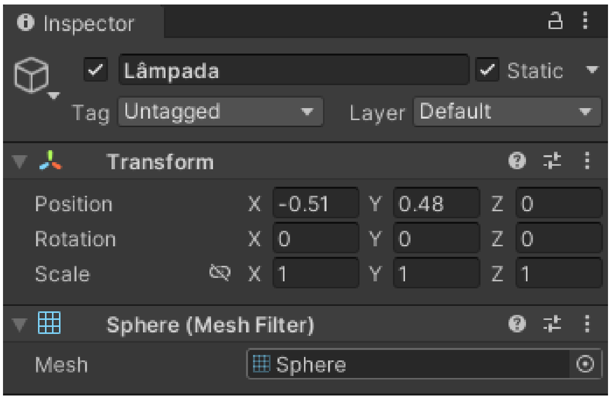

FIGURE 3.4 – Lamp represented in the Unity environment.

3.1.4.2 Creating the Control Buttons

To allow user interaction with the system, two buttons were developed in the Unity interface, labeled "Ligar" (On) and "Desligar" (Off). These buttons were created using the *engine's* native graphical interface system, via the `Button` component, found in the *UI* menu.

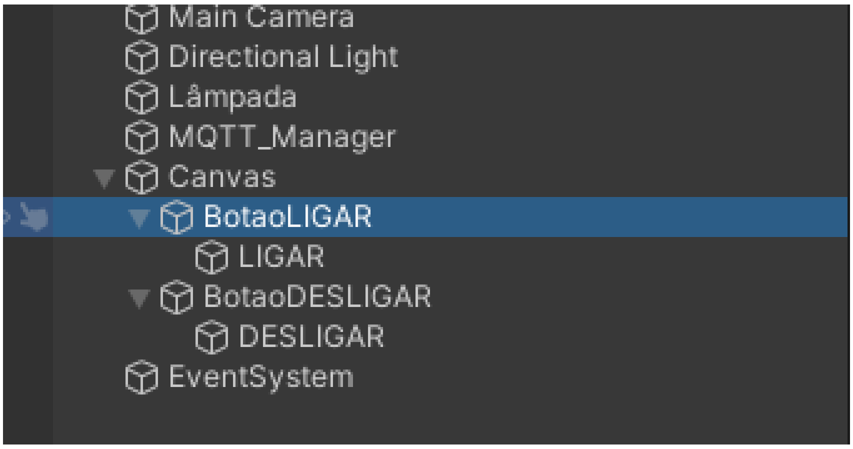

FIGURE 3.5 – Hierarchy interface of the elements.

Each button was configured to execute a specific function in the `MQTTManager.cs` *script*, as illustrated in Figure 3.6. The "Ligar" (On) button sent the `ON` message to the MQTT broker's command topic, while the "Desligar" (Off) button published the `OFF` message. These functions were associated with the buttons through the *OnClick()* field in the Unity editor, allowing the messages to be published automatically upon a user's click.

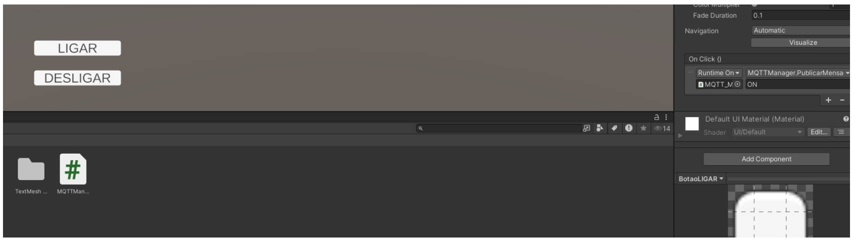

FIGURE 3.6 – Button configuration.

The creation of these buttons aimed to make the game engine interactive for sending commands from Unity to the IoT node.

3.1.5 Connection Library: MQTTnet

To implement the MQTT client logic within Unity, the *MQTTnet* library was used. It is a high-performance MQTT protocol implementation for .NET, chosen for its robustness, active community maintenance, and native compatibility with Unity's scripting environment (C#).

The `MQTTManager.cs` *script* was created to manage the communication between Unity and the ESP32. It handles the connection to the *broker*, subscribes to the control and button topics (`"esp32/control"` and `"esp32/button"`), and sends the commands to turn the virtual LED on and off based on user interaction. Additionally, the code includes the

”ping-pong” communication test, used to measure the average response time between Unity and the device.

3.2 Test and Validation Procedures

The system validation process was divided into independent stages, ensuring each component functioned correctly before the final integration.

3.2.1 IoT Node Validation

The first test consisted of running the ESP32 simulation in Wokwi and, using MQTT Explorer, manually publishing messages to the `esp32/comando` topic. Success was verified by observing the LED light up in Wokwi and receiving the confirmation message in the `esp32/status` topic, which was monitored by the Explorer.

3.2.2 Digital Twin Validation

The second test was performed with the Unity scene. By interacting with the interface (On/Off buttons), we verified in MQTT Explorer that Unity was correctly publishing the command messages and reacting visually (changing the virtual lamp’s color and intensity) upon receiving status messages published manually via the Explorer.

3.2.3 System Integration Test

The final validation involved running both Wokwi and Unity simultaneously, both connected to the broker. The goal was to verify the complete bidirectional communication loop:

1. **Action in Unity:** clicking the ”Ligar” (On) button should publish the ”ON” command;
2. **Reception by ESP32:** The ESP32 would receive the message and turn on the simulated LED;
3. **Status Publication:** The ESP32 would publish the ”LED_ON” state to the status topic;
4. **Update in Digital Twin:** Unity would receive the message and confirm the visual state of the virtual lamp;

5. **Reverse Synchronization:** The same process would occur in reverse for the ”OFF” command.

3.2.4 “Ping-Pong” Communication Test

To complement the communication analysis between the Digital Twin and the IoT node, an additional test called ”ping-pong” was implemented. This experiment aimed to measure the response time and stability of the MQTT message exchange in a complete round-trip cycle.

The procedure consisted of creating a mechanism for automatically sending and responding to messages between the two environments:

- Unity periodically published a test message (*ping*) to the control topic;
- The ESP32, upon receiving the command, immediately responded with a return message (`pong`) to the status topic;
- Unity recorded the time between sending the `ping` and receiving the `pong`, thus calculating the average communication latency.

This functionality was incorporated into the `MQTTManager.cs` *script*, ensuring the test ran automatically in parallel with the system's normal operation. The results helped validate the network's responsiveness and confirmed that the MQTT channel maintained stable communication, with no perceptible message loss over long periods of operation.

3.2.5 Performance Metrics

The final tests included measuring the following metrics:

- **Stability:** duration of the connection to the broker without packet loss;
- **Latency:** average response time between publishing a command and the system's reaction (measured in milliseconds);
- **Bidirectionality:** verification that actions in both environments (Wokwi and Unity) were correctly reflected in real time;
- **Data Integrity:** validation that MQTT messages were transmitted without corruption.

3.3 Logical Architecture and Data Flow

Table 3.2 summarizes the role of each component in the implemented *publish/subscribe* communication architecture.

TABLE 3.2 – MQTT system responsibility matrix

Component	**Function**	**Publishes to**	**Subscribes to**
ESP32 (Wokwi)	IoT Actuator	`esp32/status`	`esp32/comando`
Unity	Interface and control	`esp32/comando`	`esp32/status`
Broker MQTT	Intermediary	Manages topics and messages	
MQTT Explorer	Monitoring	Manual	Both

The data flow follows the typical decoupling pattern of *publish/subscribe* architectures. Neither the ESP32 nor Unity needs to know the location or identity of the other; communication is mediated exclusively by the MQTT broker, which ensures the delivery of messages to interested subscribers.

4 Results and Discussion

The experiments confirmed the technical feasibility of integrating a Digital Twin developed on the Unity platform with a simulated IoT node using the ESP32 microcontroller, using the MQTT protocol as the communication layer. The proposed system demonstrated satisfactory performance in terms of stability and bidirectional synchronization, proving that it is possible to establish a consistent data flow between the virtual and physical environments.

4.1 Integrated System Validation

Figure 4.1 shows the 3D environment developed in Unity, where the system's Digital Twin was created. The graphical interface contains control buttons (*Ligar/Desligar* - On/Off) and a virtual lamp representing the state of the physical LED. The `MQTT_Manager.cs` script was responsible for connecting to the public *broker* `test.mosquitto.org`, managing the publishing and subscription of topics, and logging the communication behavior in real time.

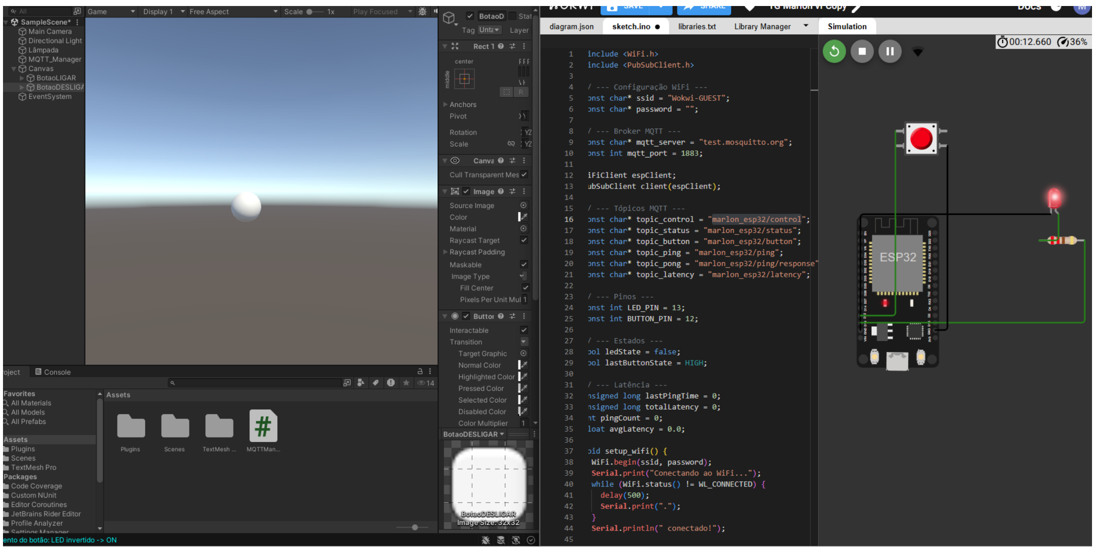

FIGURE 4.1 – Unity environment configured with the Digital Twin and control interface.

During the integration tests, pressing the button in the Unity interface resulted in the immediate sending of a command message to the `control` topic. The ESP32 would interpret the message, execute the corresponding action, and publish the new state to the `status` topic, which caused the virtual lamp to reflect the change in the Unity environment. This cycle confirmed the bidirectional synchronism between the two domains.

4.1.1 Communication Performance and Stability

To evaluate the system's responsiveness, the ”ping-pong” communication test was implemented, where Unity periodically sent a `ping` message to the ESP32 and awaited the `pong` response. The time difference between sending and receiving was used to calculate the system's average latency.

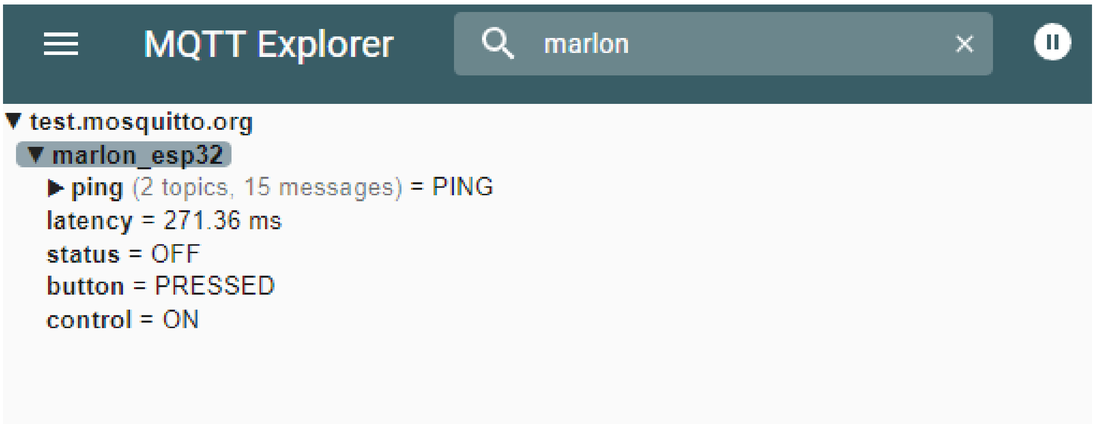

FIGURE 4.2 – Simulation of the ESP32 circuit with an LED and activation button.

The results showed an average response time of around 135 milliseconds. This is the round-trip time, but it was considered high for real-time monitoring and basic remote control applications. During continuous testing, no significant message loss was observed, and the connection to the *broker* remained stable for long periods, demonstrating the reliability of the MQTT protocol even on public networks.

Furthermore, the high latency is attributed to the lightness of MQTT, which uses asynchronous communication and small packet sizes, in addition to the simplicity of the test environment (a single publishing client and a single subscriber). However, in contexts with multiple simultaneous devices, a local MQTT server would be necessary to ensure greater predictability and security.

4.2 Critical Analysis and Limitations

Despite the positive results, some limitations were identified. Using a public *broker* implies a dependency on external availability and potential performance variations. Additionally, the Wokwi simulation environment does not fully reproduce physical aspects like electrical noise, real network latency, or hardware interference, which could impact time measurements in a physical setup.

Another point observed was the lack of authentication and encryption in the communication channel, which, while adequate for a test environment, would not be acceptable in industrial or commercial applications. For future work, adopting security protocols and credential-based MQTT authentication is recommended.

From the Digital Twin's perspective, the 3D model used was intentionally simple, serving only as a proof of concept. However, Unity allows for the creation of highly realistic 3D environments that could incorporate virtual sensors, animations, and simulated physics, expanding the application's potential.

4.3 Potential Applications and Future Work

The developed system shows potential for application in multiple domains. In industrial environments, it can be used for real-time machine monitoring, predictive maintenance, and remote device control. In educational contexts, the solution serves as a teaching tool for IoT communication and cyber-physical system integration.

The main possibilities for improvement include:

- Implementation of additional sensors (temperature, humidity, motion) to create more complex scenarios;
- Integration with cloud databases for storage and historical analysis of operational data;
- Use of machine learning techniques for fault detection and behavior prediction;
- Replacement of the public *broker* with a local or cloud-hosted server, complete with authentication and encryption;
- Expansion of the Digital Twin to multiple devices and data visualization on interactive dashboards.

In summary, the results prove that the system is functional, scalable, and provides a good foundation for evolution toward real-world applications.

5 Conclusion

This project successfully achieved its objective of integrating a 3D virtual environment, developed in Unity, with a simulated IoT device using the MQTT protocol as the communication medium. The proposed system demonstrated stable and responsive performance, enabling bidirectional control between the Digital Twin and the IoT node with an acceptable average latency of around 150 ms.

The results confirmed that it is possible to develop low-cost and easily reproducible digital twin applications using only free and open-source tools. This characteristic makes the solution accessible for academic and prototyping purposes, without requiring advanced infrastructure.

Despite the good performance, the use of a public *broker* and a simulated environment introduces limitations, particularly regarding security and the fidelity of the physical behavior. For future work, it is recommended to implement the system on real hardware, optimize latency, and integrate multiple sensors to evaluate the scalability of MQTT communication.

Therefore, it is concluded that the proposed architecture represents a solid foundation for the study and development of digital twin systems applied to the Internet of Things. It can be expanded into areas such as home automation, industrial monitoring, and technological education.

References

A Survey on Real-Time Metaverse: Challenges and Opportunities. **ResearchGate**, 2024. Real-Time Metaverse enfrenta desafios como baixa latência e sincronização. Cit. on p. 17.

ARNOLD, M. *et al.* MQTT-SN—A Publish-Subscribe Protocol for Wireless Sensor Networks. **IEEE Communications Magazine**, IEEE, v. 57, n. 3, p. 64–69, 2019. Cit. on p. 20.

ATALAY, S. Digital Process Twins as Intelligent Design Technology for Engineering Metaverse/XR Applications. **Sustainability**, v. 15, n. 22, p. 16062, 2023. Cit. on p. 15.

CAN the Internet of Things Make the Metaverse More Real? **IEEE Innovate**, 2023. Cit. on p. 15.

CHALLENGES in Metaverse Research: An Internet of Things Perspective. **NSF**, 2024. Destaca a confiança de baixa latência e sincronização. Cit. on p. 17.

DIGITAL twin: Data exploration, architecture, implementation and future. **PMC**, 2024. Reforça a necessidade de atualização virtual em tempo real. Cit. on p. 17.

GRIEVES, M. Digital Twin: Manufacturing Excellence through Virtual Factory Replication, 2014. White paper, available at: `https://www.researchgate.net/publication/275211047`. Cit. on p. 21.

GUBBI, J.; BUYYA, R.; MARUSIC, S.; PALANISWAMI, M. Internet of Things (IoT): A vision, architectural elements, and future directions. **Future Generation Computer Systems**, Elsevier, v. 29, n. 7, p. 1645–1660, 2013. Cit. on p. 21.

HOMES, S. **The Future Of Digital Real Estate And The Metaverse – Should You Invest?** Acesso em: 10 jun. 2025. 2021. Available from: `https://stackedhomes.com/the-future-of-digital-real-estate-and-the-metaverse-should-you-invest/#gs.mdjy2e`. Cit. on p. 15.

HOW Digital Twins Are Transforming Manufacturing, Medicine and More. **TIME**, 2021. Cit. on pp. 15, 23.

LEE, L.-H.; BRAUD, T.; ZHOU, P.; WANG, L.; XU, D.; LIN, Z.; KUMAR, A.; BERMEJO, C.; HUI, P. All One Needs to Know about Metaverse: A Complete Survey on Technological Singularity, Virtual Ecosystem, and Research Agenda. **arXiv preprint arXiv:2110.05352**, 2021. Available from: `https://arxiv.org/abs/2110.05352`. Cit. on p. 22.

LUO, D.; ZHANG, T.; HE, S.; SUN, W.; SUN, X. Research on the Metaverse Model of a University Based on Unity. **ACM Digital Library**, 2024. Construção de ambiente metaverso de campus universitário usando Unity. Cit. on p. 19.

LV, Z.; SHANG, W.-L.; GUIZANI, M. Impact of Digital Twins and Metaverse on Cities: History, Current Situation, and Application Perspectives. **Applied Sciences**, v. 12, n. 24, p. 12820, 2022. DOI: `10.3390/app122412820`. Cit. on p. 15.

MASUBUCHI, Y.; ITO, R.; NAKAMURA, H. **Development of Digital Twin Environment through Integration of Commercial Metaverse Platform and IoT Sensors of Smart Building**. [*S. l.: s. n.*], 2025. arXiv preprint arXiv:2505.12345. Available from: `https://arxiv.org/abs/2505.12345`. Cit. on pp. 16, 22.

MOURTZIS, D. Digital Twin Inception in the Era of Industrial Metaverse. **Frontiers in Manufacturing Technology**, v. 3, n. 1, p. 1–10, 2023. DOI: `10.3389/fmtec.2023.1154321`. Cit. on p. 21.

PIVOTAL role of digital twins in the metaverse: A review. **Digital Communications and Networks**, 2024. Cit. on p. 16.

RAVENSCROFT, T. **How digital twins are helping to develop the cities of the future**. Accessed: 2025-06-08. 2021. Available from: `https://www.dezeen.com/2021/07/09/digital-twins-develop-cities-digital-design-architecture/`. Cit. on p. 16.

SEMANTIC-AWARE Digital Twin for Metaverse: A Comprehensive Review. **IEEE Wireless Communications**, 2023. Cit. on p. 16.

TECHNOLOGIES, U. **Unity Industry Viewer Template**. [*S. l.: s. n.*], 2025. Acesso em: YYYY-MM-DD. Available from: `https://github.com/Unity-Technologies/unity-industry-viewer-template`. Cit. on p. 19.

WARD, R.; CHOUDARY, R.; J. SINGH, M. J.; ROUMPANI, F.; LAZAUSKAS, T.; YONG, M.; BARLOW, N.; HAURU, M. The challenges of using live-streamed data in a predictive digital twin. **Journal of Building Performance Simulation**, 2023. Salienta a importância da consistência temporal dos timestamps. Cit. on p. 17.

FOLHA DE REGISTRO DO DOCUMENTO

1. CLASSIFICAÇÃO/TIPO	2. DATA	3. DOCUMENTO Nº	4. Nº DE PÁGINAS
TC	12 de novembro de 2025	DCTA/ITA/TC-052/2025	38

5. TÍTULO E SUBTÍTULO:

Construction of a Metaverse Integrated with Digital Twins and Physical Sensors

6. AUTOR(ES):

Marlon Wendrer Jacinto

7. INSTITUIÇÃO(ÕES)/ÓRGÃO(S) INTERNO(S)/DIVISÃO(ÕES):

Instituto Tecnológico de Aeronáutica – ITA

8. PALAVRAS-CHAVE SUGERIDAS PELO AUTOR:

Realidade virtual; Metaverso; Gêmeos Digitais

9. PALAVRAS-CHAVE RESULTANTES DE INDEXAÇÃO:

Realidade virtual; Internet das coisas; Protocolo de comunicação; Simulação computadorizada; Sensores; Engenharia de sistemas; Computação.

10. APRESENTAÇÃO: **(X) Nacional** **( ) Internacional**

ITA, São José dos Campos. Curso de Graduação em Engenharia de Aeronúutica. Orientador: Maj. Lucas Oliveira Barbacovi; Coorientador: Prof. Dr. Christopher Shneider Cerqueira. Publicado em 2025.

11. RESUMO:

This work presents a functional architecture for integrating the metaverse, digital twins, and physical sensors connected via the Internet of Things (IoT). The proposed system consists of a 3D immersive virtual environment developed in Unity, capable of real-time reflection of physical data such as luminosity. Using lightweight communication protocols like MQTT, the project ensures synchronization and responsiveness between physical devices and the digital model. The implementation offers a practical evaluation of the technical challenges involving latency, fidelity, and scalability, contributing to immersive applications in automation, urban monitoring, and predictive simulations. The results indicate that the solution is feasible and provides a replicable model for future use in educational, industrial, and urban contexts.

12. GRAU DE SIGILO:

(X) OSTENSIVO **( ) RESERVADO** **( ) SECRETO**