ISSN 1392-3730/eISSN 1822-3605

VILNIUS TECH

JOURNAL of CIVIL ENGINEERING and MANAGEMENT

2025
Volume 31
Issue 7
Pages 747–762
https://doi.org/10.3846/jcem.2025.24091

BIM-BASED DIGITAL TWIN FOR THE MANAGEMENT OF A RAILWAY STATION

Rubén Muñoz PAVÓN [1], Marcos García ALBERTI [1✉], Jorge Jerez CEPA [1], Tomás Luis RIPA ALONSO [1,2]

[1] *Departamento de Ingeniería Civil: Construcción, E.T.S de Ingenieros de Caminos, Canales y Puertos, Universidad Politécnica de Madrid, 28040 Madrid, Spain*
[2] *LRA Infrastructures Consulting, Avda. Concha Espina, 28016 Madrid, Spain*

Article History:
- received 3 November 2023
- accepted 24 February 2025

Abstract. Building Information Modelling (BIM) has positioned as one of the main project methodologies into the AECO (Architecture, Engineering, Construction and Operations) sector. However, its implementation entails important barriers such as hardware and software requirements or BIM skills and education. The main aim of this research was to overcome those implementation barriers providing full accessibility to BIM benefits, especially in the infrastructure management phase leading to a digital twin (DT) of the built environment. Furthermore, the paper shows the technical aspects of the implementation of BIM information as well as the external functionalities, i.e., real time information or data queries, which are commonly define in BIM-IoT topic rather than BIM-FM interoperability. The research is based on the project for the northern covering and new building of the Bidebieta-Basauri passenger station, located on the Bilbao-Orduña commuter line in the Autonomous Community of the Basque Country, Spain. The final result is a BIM-based online web platform with Full access to the BIM model of the infrastructure, linked with other functionalities based on technologies such as Internet of Things (IoT), Bigdata or Cloud Computing.

Keywords: BIM, computer programming, IoT, Cloud Computing, intelligent management system, railway stations.

✉Corresponding author. E-mail: *marcos.garcia@upm.es*

1. Introduction

Building Information Modeling (BIM) has proven experience in the Architecture, Engineering, Construction and Operations (AECO) sector, especially in the field of building, where the number of studies and applications is higher than in other fields such as civil engineering. Defined as the methodology that positively affects the management of the infrastructure throughout its life cycle (Cheng et al., 2016; Dibernardo, 2012; Mawlana et al., 2015; Miyamoto, 2016; Shim et al., 2011; Tanaka et al., 2016; Yabuki, 2016), providing a large repository of information (Redmond et al., 2012) and a digital infrastructure environment (Cerovsek, 2011), BIM increasingly has important applications in the Civil Engineering sector, as shown in literature reviews such as that by Costin et al. (2018).

The benefits of the application of this methodology are numerous and can be divided according to the project phase and the sector of application. Highlights include improved workflow, increased efficiency in resource management (Shim et al., 2011), traceability and location of construction elements (Akbarnezhad et al., 2014), and or simulation of emergency situations (Gu & London, 2010).

Nonetheless, there are also significant barriers to the implementation of the BIM methodology, including the following: lack of interoperability between the information and software used (Becerik-Gerber et al., 2012), common lack of standards (Ali et al., 2014; Aziz et al., 2017; Kim et al., 2016; Kurwi et al., 2017; Liu & Gao, 2017; Omoregie & Turnbull, 2016; Vitásek & Matějka, 2017), the need of training in BIM for the people in charge of using the methodology (Arayici et al., 2011; Becerik-Gerber et al., 2012), the initial outlay on necessary Hardware and Software (Aziz et al., 2017; Hüthwohl et al., 2016; Miyamoto, 2016; Omoregie & Turnbull, 2016; Tawelian & Mickovski, 2016) and the social resistance to change in methodology (Al-Shalabi et al., 2015; Aziz et al., 2017; Bradley et al., 2016; Liu & Gao, 2017; Sankaran et al., 2016). The research team did not only face the BIM implementation barriers with theoretical approaches, but also did with its expertise in other projects such as the one related to the Civil Engineering School of Madrid (Pavón et al., 2020). In this sense, the research team had previously experienced what to pass from a traditional infrastructure management system to a BIM based one entails. That is to say, for using BIM models not only software tools purchase was demanded, but also the knowledge needed on how to use them. Moreover, it must be considered the variety of technical profiles involved in the infrastructure management. In that phase, the resistance to change is bigger than in any other project phases such as the design one.

Although the use of BIM in the management and maintenance phase of infrastructure is still a field that is currently undergoing potential development, more and more studies are focusing their efforts on this aspect. The literature review carried out shows a growing trend in the use of BIM in the operation phase of any type of infrastructure (Atencio et al., 2022; Bazán et al., 2021; Byun et al., 2021; Dayan et al., 2022; Luo et al., 2022; Moshynskyi et al., 2022; Wang et al., 2022). It is meaningful the case of Civil Engineering infrastructures, such as bridges or dams, where the use of BIM is very useful for inspection, monitoring and maintenance tasks (Liu et al., 2019; Nguyen et al., 2022; Panah et al., 2021; Zhou et al., 2023; Zhou, 2022). In the case of the railway sector, the application of BIM has focused on rail infrastructure (Biancardo et al., 2021; Häußler et al., 2021; Neves et al., 2019; Pasetto et al., 2020; Xu et al., 2020), with railway station management being a field with fewer case studies to date (Carnevali et al., 2019; Noor et al., 2018). Those examples show the significance of BIM for Facility Management (FM) not only based on a BIM model, but also on public web platforms hosting those models as well as the information for their management. In this sense, studies such as the one applied to the maintenance of the Taiyuan Subway (Wang & Zhang, 2021) or the use of the so-called Common Data Environments (CDE) linking them with bridge BIM models (Ciccone et al., 2022; Dang et al., 2020; Jensen, 2020) stand out.

The most interesting results and definitions regarding software developments dealing with BIM data synchronization are published in the line of BIM combination with the usually named Internet of Things (IoT). In this line, outstanding results are provided by different authors to achieve the interoperability between BIM and real time data. For instance, the development of new multi-layer architecture named as BIM-IoTDI to achieve semantic interoperability between DT and external applications (Eneyew et al., 2022) or the collecting of most important IoT connectivity tools and their linking possibilities with the Digital Twins (DT) (Fortino & Savaglio, 2023).

The aim of this project was the development of an intelligent management system based on BIM methodology, which is capable of overcoming the implementation barriers of this methodology in the AECO sector, specifically in railway stations and leading to what could be considered as a complete DT management environment of the infrastructure. To achieve this, some important considerations must be taken into account based on previous research. According to literature, this type of BIM-based applications should provide specific functionalities on information sharing, communication and process management (Singh et al., 2011). In this regard, through the complete programming of the platform described, full interoperability between management data and the BIM model is sought. This interoperability is intended to be achieved through the application of technologies such as Internet of Things (IoT), Cloud Computing or Big Data. A real construction project in the Autonomous Community of the Basque Country in Spain, was chosen for the development of the platform. This public tender project proposed by the Spanish Ministry of Transport, Mobility and Urban Agenda (MITMA) includes the drafting of the project for the northern covering and new building of the passenger station in Bidebieta-Basauri, located on the Bilbao-Orduña commuter line of the Bilbao-Orduña network (Adif, 2021). This project, with an execution period of 14 months, has been awarded to the company L.R.A. Infrastructures Consulting, which must draw up the basic and construction project for the aforementioned actions.

The significance of this project relies on the actual proposal for its implementation. On the one hand, BIM methodology does not stand out for its applications in civil infrastructures, with the application of this methodology in railway stations being far behind. On the other hand, this project unifies several technologies (such as IoT or Cloud Computing) to try to overcome the barriers of BIM implementation in a real infrastructure, such as the Bidebieta-Basauri station.

To sum up, the aims of the project were: (1) development of a platform capable of overcoming BIM implementation barriers, (2) usage of different technologies for the application of BIM in railway infrastructures, and (3) study of the possible functionalities to be implemented in a real way in a Spanish railway station.

2. Methodology

According to the main objective of the paper this section shows the definition and technical development process for the creation of the BIM-based platform. Published literature on this topic is more oriented to provide with a general vision or future possibilities rather than to define technical aspects. Some authors have reported on the interoperability between stakeholders, IFC visualization or quantity and schedule information availability (Jang et al., 2021) or have detailed case studies and implementations (Lin et al., 2020). However, technical aspects such as the databases used, algorithms to link IFC elements with external information or with real time data are not commonly defined in publications about BIM-based FM. That level of definition about Databases and information management can only be found in BIM-IoT specific research (Eneyew et al., 2022; Fortino & Savaglio, 2023). The development of a complete management platform hosts multiple steps. Figure 1 shows the different phases followed to achieve

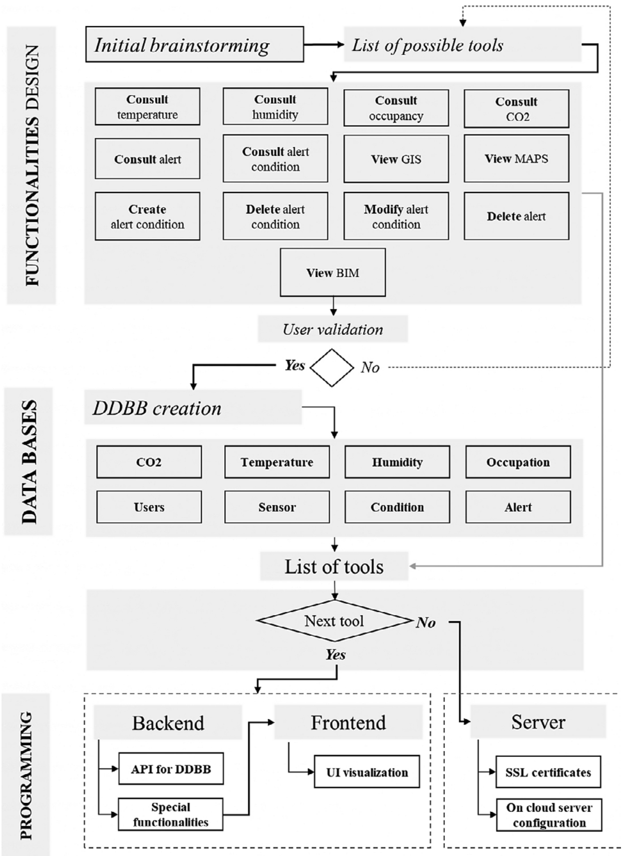


Figure 1. Methodology used in the project

the results shown in this project. The methodology itself can be divided in three main blocks: Design of functionalities, Databases and Programming.

In the functionalities design block, a hypothesis of the needs of the final user was made. Since the aim of the platform was the link between databases, sensors and BIM model in railway stations, a hypothesis of functionalities to be implemented in the platform was performed. In this sense, the possible tools focused on querying the parameters measured by the sensors, alerts and conditions, and BIM/GIS visualization in the platform. Through the query of parameters measured by the sensors, the user will be able to consult and filter any measurement of parameters such as temperature, humidity or CO_2. By consulting, creating and modifying alerts and conditions, the user can set alert conditions to each sensor in the BIM model, allowing the platform to automatically register alerts when the conditions are not met. Finally, through the BIM and GIS visualization functionalities, it will be possible to provide the user with a digital environment of the infrastructure, through which it will be possible to access the functionalities. These functionalities were proposed to different clients and infrastructure managers for testing their feasibility.

Subsequently, the design and development of the databases necessary to achieve the functionalities of the platform described above is carried out. For this purpose, eight different databases are considered: CO_2, Temperature, Humidity, Occupancy, Users, Sensors, Conditions and Alerts.

Once the databases have been developed, the necessary backend and frontend programming is carried out. This programming must be done independently for each of the functionalities described in the list of tools of the platform. Within the Backend programming, developed in Python, Database management algorithms are carried out, which interact with the information through the development of APIs (Application Programming Interface). However, a second layer of programming is needed to transform the raw information from the Database into the information that the Frontend must display. Therefore, Frontend programming focuses on the correct visualization of the information coming from the backend.

Finally, for full user accessibility to the developed platform, it is necessary to host all developments on a server. Using application containerization techniques, the developed application is uploaded to a cloud server with an Ubuntu operating system. Furthermore, the necessary SSL (Secure Socket Layer) certificates are configured to ensure the user a secure connection through any internet explorer.

3. Development

The development necessary to achieve the main aim of the project, the BIM-based intelligent web management platform, encompasses three main blocks, as shown in Figure 1: Functionality Design, Database Development and IT Developments. Moreover, this chapter includes a section dedicated to the flow of information inside the designed platform, detailing the workflow of the developed algorithms when working with the measurements, alerts or conditions.

3.1. Functionalities design

Railway stations have an extensive presence in Spanish transport systems. With more than 200 railway stations (Renfe, 2022), this type of Civil Engineering infrastructure is one of the most information-intensive to manage. Real-time information on train schedules, ticket sales and purchases, equipment maintenance, inventory control or measurements of different parameters are some examples of the information needed to manage this type of infrastructure.

For the development of the first initial version of the platform, a basic aim is established, the query of real-time information on temperature, humidity and occupancy, along with the possibility of creating conditions that register alerts for these parameters. Furthermore, these functionalities must be linked to information in BIM or GIS format. The scheme of functionalities proposed for this first version of the platform is shown in Figure 2.

In this initial release, the platform was able to differentiate two roles with different privileges or permissions. Firstly, the ordinary user. Secondly, the maintenance user or the user in charge of the platform management. On

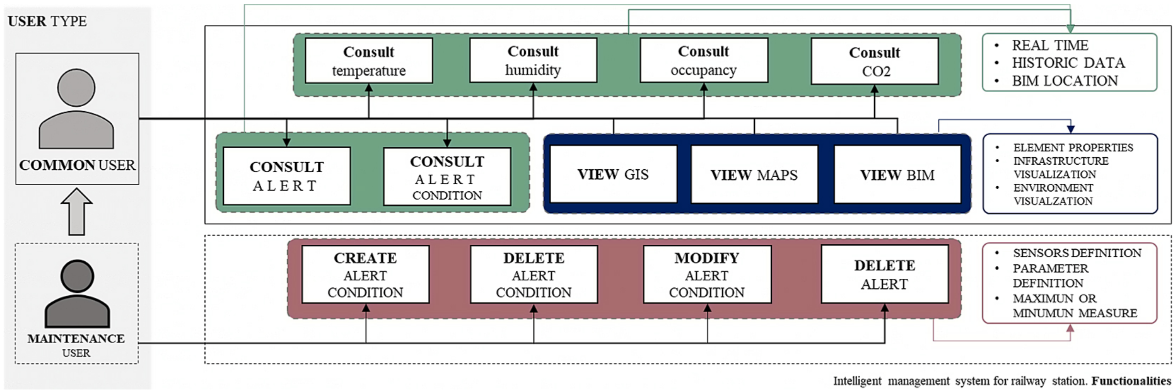


Figure 2. Outline of functionalities

the one hand, the common user will have access to all query and visualization functionalities, both of parameters and information in real time and of implemented BIM or GIS models. On the other hand, the maintenance user or infrastructure manager shall be an inherited user of the common user. That is to say, this last-mentioned user will have access to all the functionalities of the common user and will also have extra privileges for other added functionalities. The maintenance user can, in addition to querying parameters or viewing BIM or GIS models, create alert conditions, modify existing alert conditions and or delete alerts.

3.2. Databases

The databases chosen for the development of the platform are known as structured ones. In contrast to this typology, there are the so-called unstructured databases. Each of these has different advantages and disadvantages. On the one hand, unstructured databases provide with great flexibility in data management and even greater speed and efficiency at times of high data traffic. On the other hand, structured databases are much less flexible than unstructured ones, which implies the development of more complex data management algorithms than if unstructured databases were used. However, it is precisely this lack of flexibility that forces the development of much more orderly databases and management algorithms, which can subsequently translate into faster information retrieval. Structured Query Language (SQL) databases have been chosen for the development of the advances shown in this paper. The structure of the mentioned databases is defined in Figure 3.

The design and structure of the databases is obtained after careful analysis of the functionalities to be achieved in the platform. Considering the three main blocks: real-time information, management of alerts with conditions and BIM + GIS + Maps visualization, the previously mentioned databases are developed.

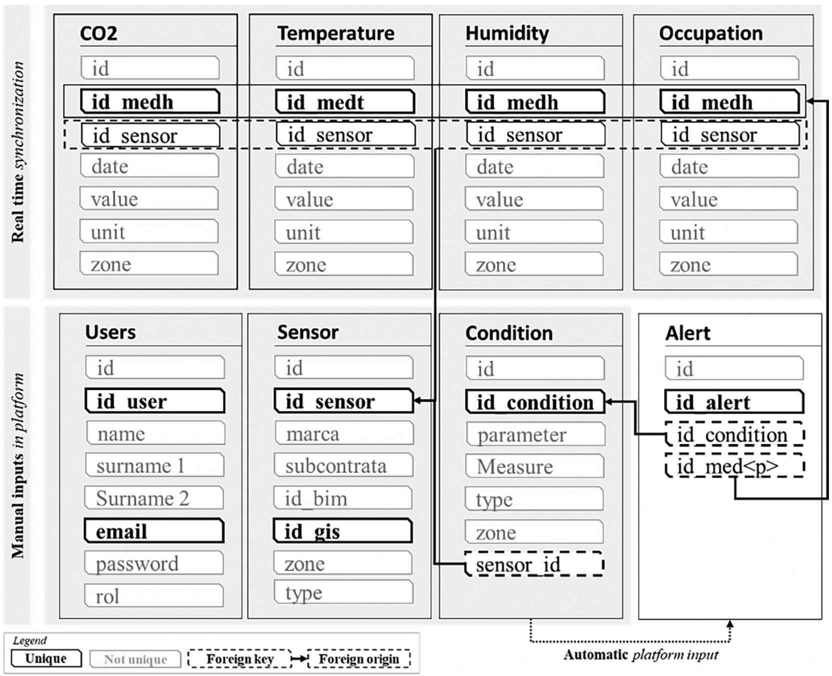


Figure 3. Designed database structure

3.2.1. Real-time information databases

To achieve the functionality related to the measurement and query of information in real time, the structured databases related to CO_2, Temperature, Humidity and Occupancy are developed. Each of them consists of the following parameters:

- *Identifier*. This is a unique parameter. It reflects the record order in the database and is auto incrementing. The data insertion algorithm proceeds to insert this parameter automatically.
- *Unique identifier*. Each measurement registered by each sensor will have a totally unique identifier. In this way, different alerts can be associated to certain measurements, thus being able to use all the information of the measurement in the alert database, such as Foreign Key.
- *Sensor identifier*. This is a Foreign Key that relates the sensor that has registered the measurement together with the inserted measurement. With this parameter, it will be possible to access all the information stored in the sensor database.
- *Date*. The algorithm developed, executed at the moment of receiving a measurement from the sensor, will proceed to save in the database the exact date on which it receives this information.
- *Value*. Similarly to the previous parameter, when the Backend algorithm receives the sensor information, it saves the received measurement as a "value" in the corresponding database.
- *Unit*. This parameter varies depending on the type of measurement considered. In the case of Temperature, it will be ºC, in the case of Humidity it will be a percentage, for Occupancy it will be units and for CO_2 it will be particles per million, ppm. Although it seems to be a fixed value for each type of measurement, it has been decided to introduce this parameter in these databases to accommodate the possibility of receiving different units of measurement by the sensor.
- *Zone*. This parameter records the area of the BIM model in which the measurement has been recorded or linked to.

3.2.2. User database

For the users' management, the structure of the "Users" database shall consist of the following parameters:

- *Identifier*. This is a unique parameter. It reflects the record order in the database and is auto incrementing. The data insertion algorithm proceeds to insert this parameter automatically.
- *Unique identifier and e-mail*. Each user shall have a unique identifier that cannot be shared with any other user. The same occurs with the user's "email" parameter. Therefore, each user will have two unique identifiers; on the one hand, the "id_user" and on the other hand, the "email".
- *First name*, *surname 1 y surname 2*. Parameters that collect first and last name of the user. Of course, this data is not unique in the database.
- *Password*. Password that the user has previously established to be able to access the platform. It should be noted that the Python algorithm of the developed Backend encrypts the password before inserting it into the database. Thus, the saved information is not identifiable by the platform development team.
- *Role*. This parameter is used to limit access to the functionalities of the platform. In this first version only two roles have been considered; Standard user and Management and maintenance user. The Frontend algorithms check the role of the active user of the platform. In this way, before opening a tool or offering a functionality, the algorithm checks if the user has the appropriate permissions.

3.2.3. Sensor database

Sensors are a fundamental aspect in the main platform functionalities. On the one hand, sensors will be fully linked to the recorded measurements. On the other hand, they will also be linked to the condition and alert databases. The structure of the sensor database is explained as follows:

- *Identifier*. This is a unique parameter. It represents the record order in the database and is auto-incrementing. The data insertion algorithm proceeds to insert this parameter in the database automatically.
- *Unique identifier and BIM identifier*. In this case, there are two unique identifiers for each sensor entered in the database. On the one hand, the common identifier, formed by a string of text. On the other hand, there is the BIM identifier of the sensor, in number format. This identifier will allow the sensor to be located within the BIM model itself.
- *Brand and subcontractor*. These parameters collect information related to the sensor. Data such as the brand of the device or the company in charge of its maintenance are collected in this parameter as a simple text string.
- *Zone and type*. These parameters are text strings. They define the zone or space of the BIM model in which the sensor is located and the type of measurement it collects. For example, locker area or temperature measurement.

For each measurement entered in the relevant database, whether it is Temperature, Humidity, Occupancy or CO_2, a connection is established with the sensor that recorded it. This connection is made possible by the use of the Foreign Key of the unique sensor identifier in the corresponding measurement database.

3.2.4. Condition database

The platform includes a specific module for conditions. So as to achieve the desired functionalities, the construction of the following database is proposed.

- *Identifier*. This is a unique parameter. It reflects the record order in the database and is auto incrementing. The data insertion algorithm proceeds to insert this parameter automatically.
- *Unique condition identifier*. This identifier shall be unique for each condition created. Thus, no other condition may have a similar value in this field.
- *Parameter*. This field will allow to record which type of measurement this condition is created. Therefore, it defines whether it is a condition for Temperature, Humidity, Occupancy or CO_2.
- *Measure*. Collects the value, maximum or minimum, that will trigger an alert.
- *Type*. This parameter defines whether the measurement refers to a maximum limit or a minimum limit. That is, whether an alert shall be registered or triggered considering if the incoming measure is higher or lower than the *measure* parameter limit saved in this database.
- *Zone*. This is a text string that contains the space or room of the BIM model in which the sensor located.
- *Unique sensor identifier*. It is a Foreign Key that allows to relate the condition with the sensor that will host it. In this way, the information of this sensor is also accessible from the present database.

The condition concept within the platform enables the automatic registration of alerts by the backend algorithms. Through the conditions' module, the user will be able to register different criteria linked with each parameter type. That is, conditions that will register an alert associated with a temperature, humidity, occupancy or CO_2 sensor.

3.2.5. Alert database

As with conditions, alerts have their own module within the platform. These alerts are entirely related to the conditions of the above-mentioned databases. The backend algorithms developed in Python are responsible for receiving the measurements from the different types of sensors. In this regard, upon receiving each measurement, the algorithm checks if there are conditions previously associated with the sensor. If so, the algorithms shall create a new alert database object, which consists of the following information:

- *Identifier*. This is a unique parameter. It reflects the record order in the database and is auto incrementing. The data insertion algorithm proceeds to insert this parameter automatically.
- *Unique alert identifier*. This will be a unique type of parameter in this database. That is, no other alert may have the same value as another alert.
- *Unique condition identifier*. This is a Foreign Key that links the alert database to the condition database. So, for each registered alert, there is always an associated alert condition.
- *Unique measurement identifier*. It is also a Foreign Key, as is the unique condition identifier. In this case, by using this Foreign Key, the registered alert is related to the measurement that triggered it. In turn, this measurement has another Foreign Key that relates it to the sensor that has emitted or registered this measurement.

Through the databases described above, the required information structure is provided to start the Frontend and other Backend programming works, with the aim of creating the measurement querying, condition management and alert management modules.

3.3. IT developments

The magnitude of the platform being presented in this project is significant. Backend and Frontend programming techniques are necessary for the complete functioning of the platform, along with important developments in database management algorithms. In addition, the accessibility via Internet has also made it necessary to set up servers in cloud. Specifically, servers based on Ubuntu operative systems. Furthermore, for secure access, SSL certificates have been incorporated and manually configured inside the server.

Since the main aim of the project is to detail the capability of relationship and interoperability with the BIM model, the following point focuses on collecting the class diagram necessary to achieve the real-time management system linked to the BIM model.

3.3.1. Class diagram

Object Oriented Programming (OOP) was selected for the development of the platform shown in this project. This OOP is found in both Backend and Frontend development. On the one hand, in the second case, the OOP is implemented through programming in Angular framework, using HTML, JavaScript and CSS languages. On the other hand, the Backend programming is developed Python. These developments provide the basis for subsequent client-side programming. Thus, this section shows the UML class diagram developed in the Backend, which allows to achieve the desired platform functionalities. The class diagram is shown in Figure 4.

The diagram firstly defines the name of the class (black box). Thereafter, by means of a box that distinguishes it from the name of the class, the attributes of the class can be found (dark grey). Finally, in the last box, the methods or functions of the class are shown (light grey). The attributes of each class are variables of the class, while the methods are functions that can be executed in that class.

The class diagram shown consists of 6 classes: CO_2, Sensors, Condition, Temperature, Occupancy and Alert. The CO_2, Temperature and Occupancy classes will record measurements of the parameters that give them their name. The Sensor class shall be focused on data from a sensor, irrespective of its location. The Conditions class will allow the creation of conditions through the platform. Finally, the Alert class will collect all the information related to alerts generated by the platform itself.

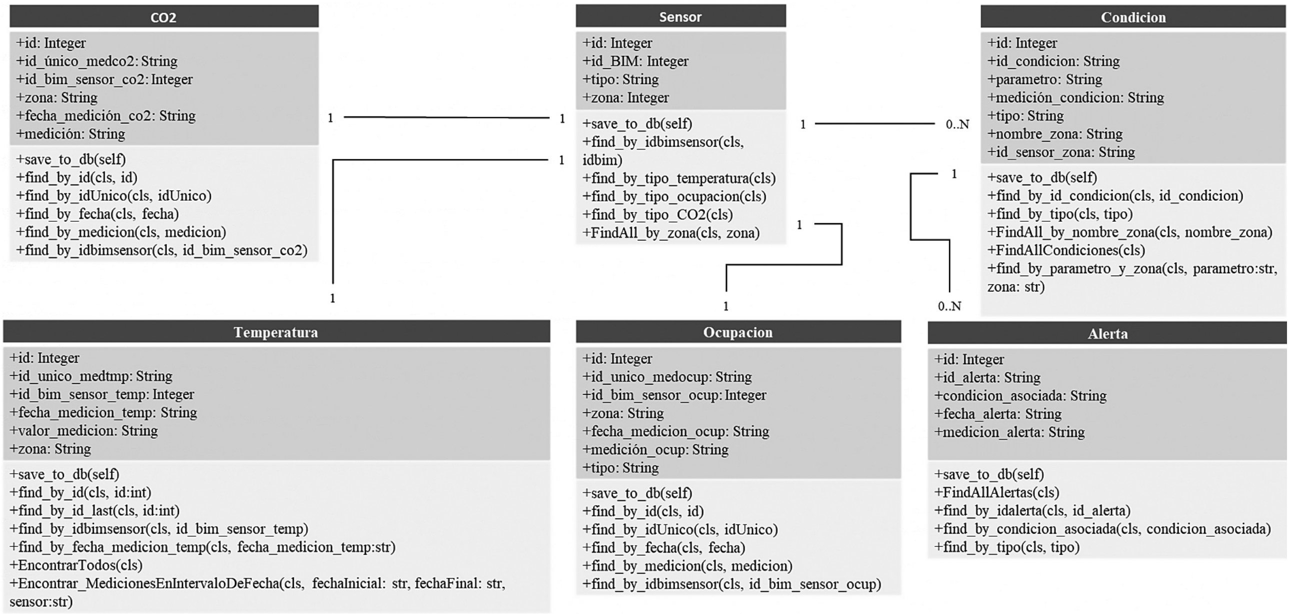


Figure 4. Backend UML class diagram

Collectively, the CO_2, Temperature and Occupancy classes hold data related to the measurements. That is, they do not focus on collecting information from the sensor that executes them. In this sense, the measurement classes consist of the same parameters as shown in Figure 3 on the database structure. The functions of these classes are:

- "*save_to_db*". Function that stores the object of the corresponding class in the database. It does not require any input.
- "*find_by_id*". Function that demands an "id" to search for, and returns the object of the class that meets that "id".
- "*find_by_idUnico*". Function that demands the "idUnique" of a measurement and returns the object of the class that meets this "idUnique". This function is extremely useful, as it allows interacting with the client side directly.
- "*find_by_fecha*". Function that demands a date and returns the object that matches the date input, or the nearest one.
- "*find_by_medicion*". Function that demands a measurement, either Temperature, Occupancy or CO_2, to return the complete object if it meets that measurement.
- "*find_by_idbimsensor*". Receiving a unique identifier from the sensor that registered the measurement, it returns all objects that meet this condition.

Following, is the "Sensor" class, which is responsible for collecting location data from devices, without recording measurements. Methods of this class are:

- "*save_to_db*". Function that stores the object of the corresponding class in the database. It does not require any input.
- "*find_by_idbimsensor*". This function requires a unique identifier of the sensor so as to locate it in the BIM model. The function shall return the object that meets this identifier.
- "*find_by_tipo_temperatura*", "*find_by_tipo_ocupación*", *"find_by_tipo_CO2"*. These functions do not require any parameter as input. They will simply return all sensor objects that are of type Temperature, Occupancy and or CO2 respectively. That is, it returns the sensors according to the type of measurement they perform.
- "*findAll_by_zona*". This function asks for the name of a zone in the BIM model. It will return all sensors that are in the zone introduced as input.

The "Condition" class shall hold all condition related data on the platform. It should be noted that the conditions are criteria that will cause an alert to be registered, or not, when a measurement is inserted into the database or is emitted by a sensor. The "Condition" class consists of the same attributes as indicated in Figure 3. The methods of this class are:

- "*save_to_db*". Function that stores the object of the corresponding class in the database. It does not require any input.
- "*find_by_id_condicion*". As input, the function demands a unique condition identifier so as to return the condition that matches the input entered.
- "*find_by_tipo*". This function demands the type of condition to search for. That is, Temperature, Occupancy or CO_2 condition. Therefore, it will return all conditions depending on the type entered.
- "*find_all_by_zona*". Function that demands the name of a zone in the BIM model. On its part, it shall return all existing conditions associated with that zone, regardless of the type of measurement.

Finally, the "Alert" class, like the previous ones, has the same attributes as those shown in Figure 3. These alerts will be automatically registered by the platform whenever a measurement does not fulfil any of the associated conditions. This class consists of the following methods or functions:

- "*save_to_db*". Function that stores the object of the corresponding class in the database. It does not require any input.
- "*findAllAlertas*". No input required. It will return all alerts registered in the database.
- "*find_by_idalerta*". The function demands a unique alert identifier, so it will return the alert that meets this data.
- "*find_by_condicion_asociada*". Function that demands as input a condition and returns the alerts generated when this condition is fulfilled. That is to say, if there is an alert condition that indicates that a temperature of 25 ºC cannot be exceeded, this function will return the alerts generated due to this condition.
- "*find_by_tipo*". This function demands the type of measurement to return all alerts generated by that condition. That is, it will return all Temperature, Occupancy or CO_2 alerts.

With all the class structure described, the Backend has the necessary tools to be able to provide the Frontend with all the necessary information, depending on each module or section of the platform. These classes make it possible to query, create and or delete measurements, conditions or alerts.

3.4. Platform workflow

To achieve the functionalities described above, three main development blocks are required: databases, Backend developments and Frontend developments. After these blocks have been programmed and executed, the platform presents the workflow shown in Figure 5, where the aforementioned blocks can be appreciated.

Firstly, a distinction has been made between two main methods of accessing or interacting with the platform. On the one hand, the functionalities developed in the Frontend. These tools will allow platform users to modify alerts, conditions or consult parameters such as Temperature, Humidity, CO_2 or Occupancy. These utilities are based on algorithms located in both the Backend or the Frontend. On the other hand, there are the algorithms in charge of interacting with the information collected by different sensors. These algorithms are located only in the Backend as they interact directly with the sensors and not with the users through the user interface (UI) of the platform.

If the user consults a measurement, alert or condition, a connection to the query algorithms is made from the frontend. These algorithms will return the required information to the UI. Therefore, from the client side the visualization in the desired format will be provided to the user. For example, in table format.

If the user wants to modify a condition, it is the Frontend that must execute this command. For example, if the user wants to modify that an alarm is triggered when 25 ºC is exceeded in the room "Lockers" of the BIM model, a connection will be established from the Frontend with the Backend to make the desired modification.

Finally, there are modifications made by the sensors. These modifications are just insertions in the database corresponding to the parameter they measure. These can be Temperature, Occupancy, CO_2 or Humidity. To do so, the sensor would directly trigger its information against the API created in the Backend, without the need to go through the client or Frontend side of the platform.

4. Results

The result is the intelligent web management platform, consisting of all the functionalities described above. This platform will be divided into three main blocks or sections, as Figure 6 shows. First of all, the block related to visualization is located. To do this, the platform uses two currently emerging technologies, BIM and GIS. The second block is related to real-time information on Temperature, CO_2, Humidity and Space Occupancy in the infrastructure. The

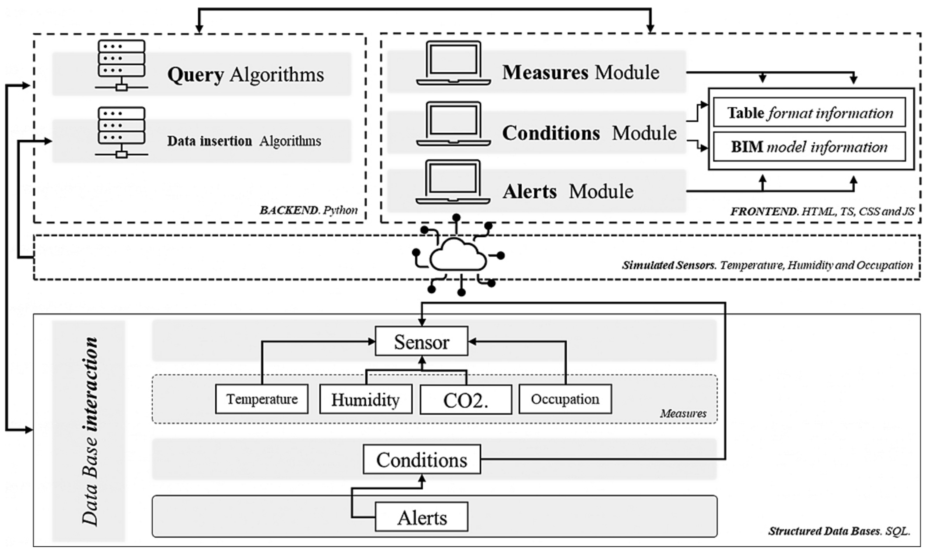


Figure 5. Internal platform workflow

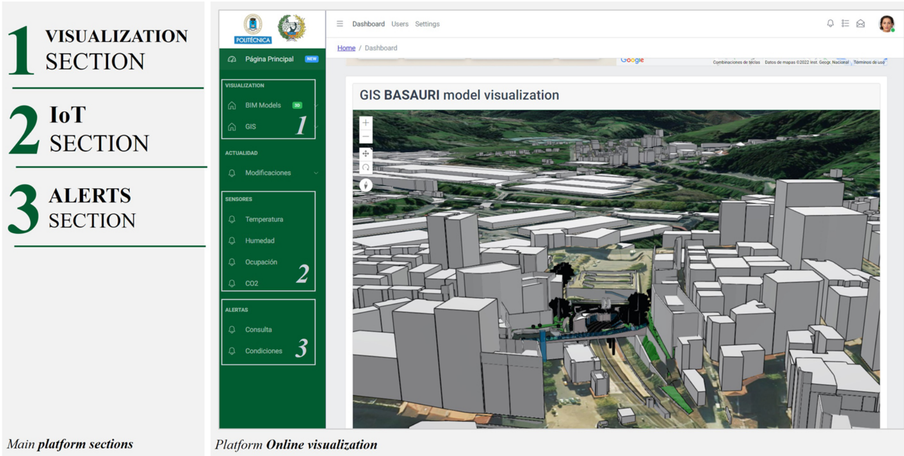


Figure 6. Main blocks of the platform

results of this block allow access to real-time information and data recording from sensors communicating through IoT technology.

The third block provided by this initial version of the platform is related to the management of alerts and conditions. Through access to this content, the user will be able to query and/or modify alerts, as well as manage the conditions of these alerts.

4.1. Real-time parameter management system

Real-time information is an indispensable parameter to be considered nowadays. In civil infrastructures, such as railway stations, there is a large volume of interesting data for further analysis. Train timetables, passenger flows within the station, origin and destination of passengers, or the measurement of parameters such as temperature, CO_2 and or humidity can be of great interest, not only for current management also for future research possibilities.

It is on the measurement of these basic parameters that this first version of the platform focuses. Establishing a relationship between people flows and parameters such as temperature or humidity, it is possible to design predictive maintenance (PdM). Even more interesting if additional parameters such as the relationship between station flows and train arrivals, timetable and/or weather conditions are implemented. Analysis of as many variables as possible would allow the design of highly accurate predictive maintenance tools. In this regard, and as a basis for achieving these concepts, it is proceeded to simulate a platform capable of hosting real-time information from different sensors. Temperature, humidity, CO_2 and people counting are the simulated parameters used in the platform.

It is important to note the actual state of the Basauri railway station reconstruction project at the time of writing this paper. Awarded and in the design phase, the current state of the railway station is characterized by its antiquity, awaiting renovation owing to the upcoming start of works. This means that no sensors are installed, so the reading of the measurements is simulated, thanks to the development of algorithms that carry out the task of measuring these parameters. Python is the programming language used for this purpose.

4.1.1. Temperature, Occupancy and CO_2

The client side of the platform, also known as the Frontend, provides the user with a similar structure for the measurement and query cases of Temperature, Occupancy and CO_2. The visualization shown in Figure 7 is achieved by using TypeScript, HTML, CSS and JavaScript programming languages. Within this section of the platform, there are two types of information sources; one in table format and other in visual format, both supported by the BIM methodology to provide the real environment of the platform.

The platform is in permanent synchronization with the platform Backend, allowing the Frontend, or client side, to be updated continuously if any event of new temperature, occupancy and CO_2 measurements is triggered. On the one hand, the table format information shows data such as the space of the BIM model in which the measurement was recorded, the date of recording, the value of the measurement, a unique identifier of the recorded measurement, the identifier of the sensor that registered the measurement and a unique identifier of the zone. Each of these fields is completely filterable, thus allowing significant accessibility to all the data recorded in the database. On the other hand, there is the visualization block of the platform, which is linked to the information displayed in table format, as shown in Figure 8.

The information displayed to the user in table format is the information recorded in databases. For the development of the platform shown, it was decided to use structured databases, as opposed to the possibility of using unstructured ones. According to the parameter selected in the side menu; Temperature, Occupancy or CO_2, the table and the BIM model will query a specific database, thanks to the query algorithm made in Python.

The structure of the information displayed to the user is similar, although with slight differences according to the parameter selected. In the information displayed in a table format, parameters such as zone, recording date, Measurement, Measurement Identifier, Sensor Identifier and Zone Identifier are shown. In this way, the user can query and filter according to a specific sensor, a specific area, a measured value or a recording date. It should be pointed the significance of the parameters Sensor Identifier, Measurement Identifier and Zone Identifier. These are unique columns in their respective databases. That is, no two objects or rows in such database can share this parameter. This unique identifier consists of the following structure "PAR_AAAAAA", where: "PAR" is the parameter being considered. In the case of temperature, it shall be "TEM", in the case of CO_2 it is "CO_2" and in the case of occupancy it is "OCU". Also, the text string "AAAAAA" is made up of six letters of the alphabet, registered randomly, differentiating between lower- and upper-case letters.

The information gathered in table format is fully synchronized with the BIM model of the infrastructure itself. Once the user has filtered the desired information and clicks on the row of a certain measurement, the platform will be updated consequently. Thus, as shown in Figure 8, the BIM model will automatically search for the sensor that has recorded the selected measurement. At the same time, the BIM environment itself will show the last recorded measurement for that sensor and the register of the last recorded measurements, sorted by date of insertion in the database.

4.1.2. Condition management system

The platform included a special module, dedicated exclusively to the creation and/or modification of conditions. Furthermore, by means of a table format or directly through the BIM model, the conditions can be queried, as shown in Figure 9. The conditions will have a direct influence on alerts, as these conditions define the criteria that a measurement must meet in order not to be considered an alert. For example, if a maximum temperature condition of 25 ºC was defined in a zone, as soon as a sensor reg-

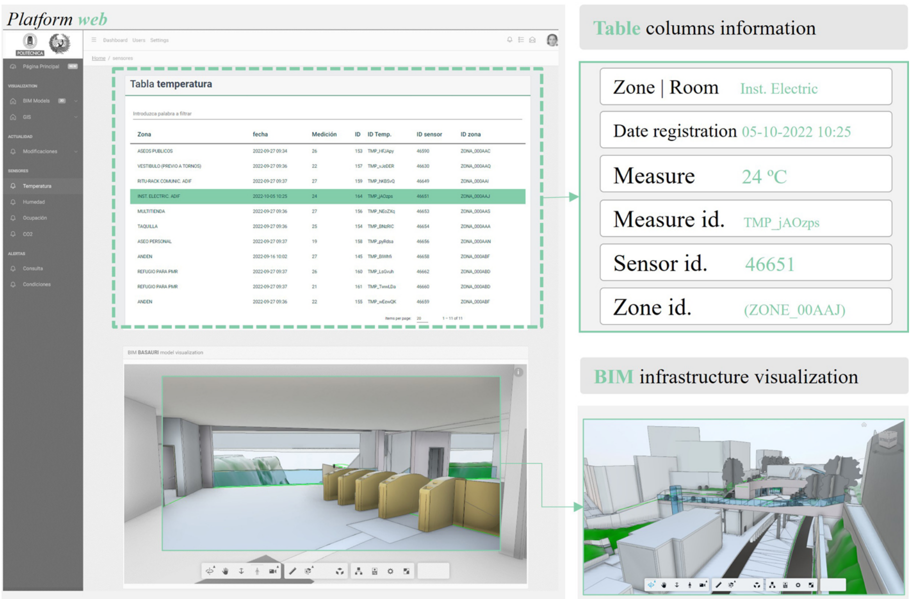


Figure 7. Functionalities of the parameter registration block on the platform

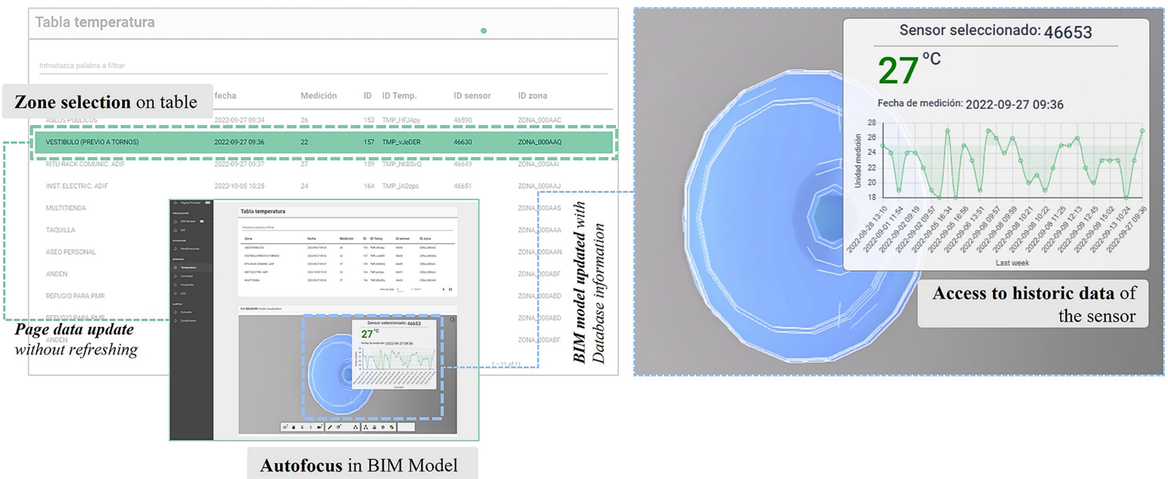


Figure 8. Synchronisation between BIM model and database information

istered a temperature higher than this, the Backend algorithms of the platform would register an alert associated to this measurement and to the condition that does not fulfil.

Figure 9 outlines the functionalities of this module in the platform. First of all, all existing conditions in the databases are provided in a table format, regardless of the associated zone or sensor. On the one hand, clicking on a specific row, that is, on a condition, allows the user to choose between modifying the condition or displaying the sensor to which the condition refers in the BIM model. At the same time, if the user selects a sensor from the BIM model, the platform renders a new table summarizing all the conditions associated with the selected sensor.

On the other hand, in case the user wants to create or modify an existing condition, the parameters shown in Table 1 must be considered. When a condition is created, the platform algorithm will automatically generate the condition number and unique condition identifier parameters. Furthermore, the user can choose whether to set the condition for Temperature, Occupancy or CO_2. It may also indicate whether the alert is to be registered when the value of the condition is exceeded or falls below the value set in condition. In the same way, the user can indicate the zone in which the condition is to be applied, and within that zone, a specific sensor. When modifying an existing condition, the user shall only be able to edit the value that will cause the alert to be registered. The identification or location parameters shall not be modifiable.

4.1.3. Alert management system

The alert management system module was linked to the condition management module. As mentioned above, AI developed using Python algorithms will be in charge of analyzing each recorded measurement. In this way, each time a temperature, occupancy or CO_2 measurement is recorded in the databases, the AI checks the existing conditions and verifies whether the measurement meets the recorded conditions or not. In case of failure to fulfil them, a new alert will be registered in the relevant database.

The alert management module is shown in Figure 10. This section of the platform has a similar structure to the management of alert conditions. Nonetheless, there are minor differences. First, the user must access this module using the side browser. Afterwards, the user is provided with information of all registered alerts. This information is defined by the name of the zone, the name of the measured parameter (Temperature, Occupancy or CO_2), date of recording and value of the recorded measurement.

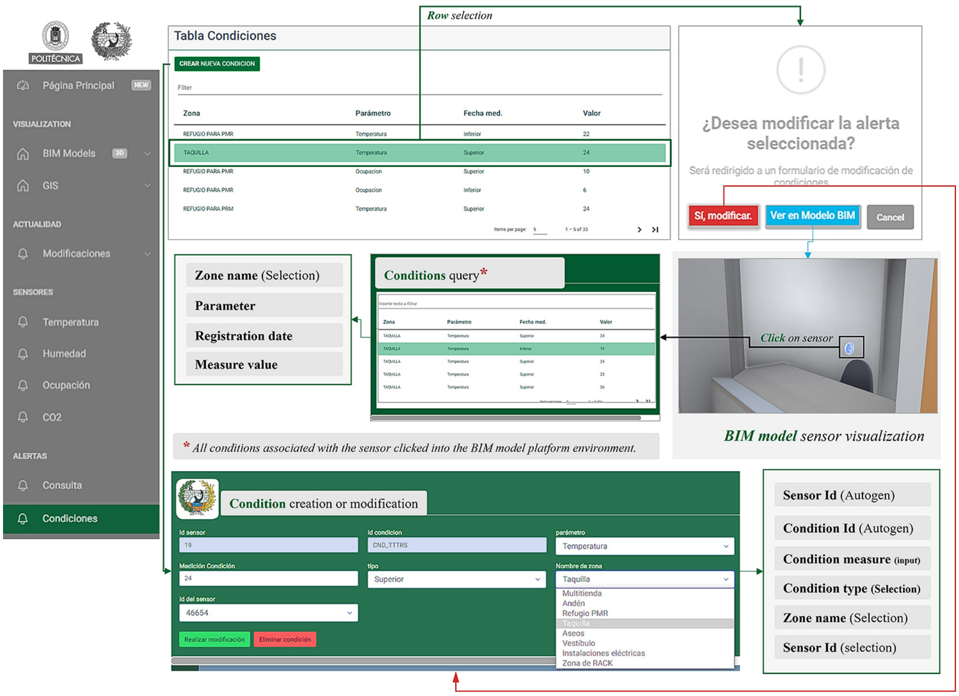


Figure 9. Conditions module on platform

Table 1. Parameters manipulation on condition creation or modification

Type	Parameter	On condition creation	On condition modification
Identification	Condition number	Autogenerated by the platform	Not editable
	Condition Id	Autogenerated by the platform	Not editable
Characterization	Condition Measure	Temperature, CO_2 or occupation	Editable
	Condition Type	Maximum or minimum	Not editable
Localization	Zone name	Selected by the user	Not editable
	Sensor BIM id	Selected by the user	Not editable

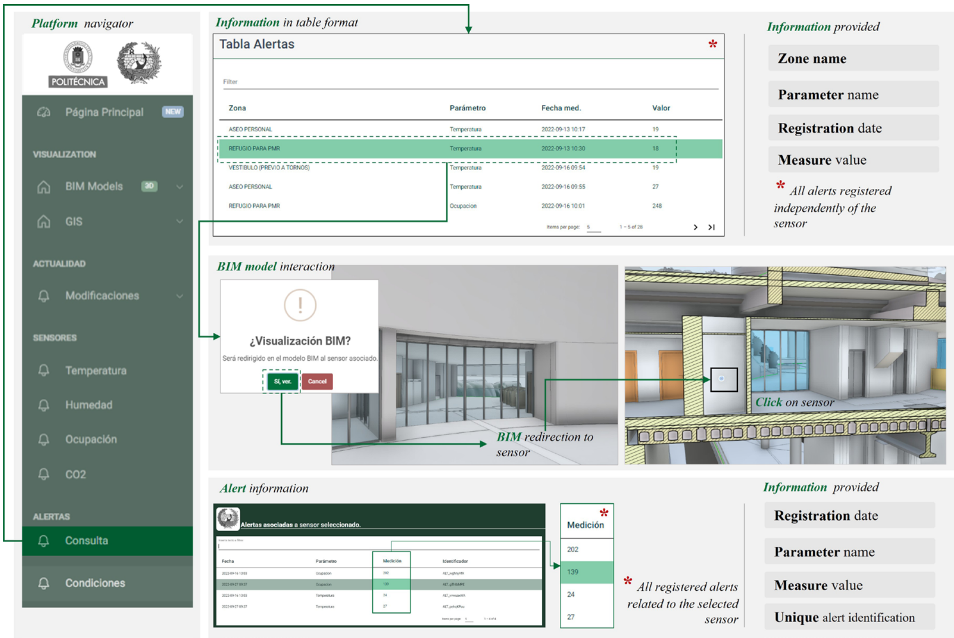


Figure 10. Alert management system module in platform

As opposed to the condition management module, the alert management module does not allow the modification of alerts. This module only allows to query and/or delete registered alerts. Therefore, if the user clicks on a specific alert in the table, the platform will redirect the user through the BIM model to the sensor that has recorded the measurement that triggers the alert. Simultaneously, the user will be able to select the sensor in the BIM model. Subsequently, the platform will show the user all alerts related only to that sensor. In this case, the information is provided to the user in table format and is defined by the date in which the alert was recorded, the parameter name, the measurement value and a unique alert identifier.

4.2. Asynchronous data

Although real-time information is one of the most significant parts of the platform, static information also has an important role to play in the developments of the platform. To provide real-time information, the platform is based on technologies such as BIM, IoT or Cloud Computing, together with structured databases. In order to provide the user with information at a lower update rate, GIS has been used in combination with other technologies or software tools, as shown in Figure 11.

In this case, the platform has an independent module for the visualization of this type of information. First, and separately from the sensor data collection, a BIM model is shown as a query of building elements. In this way, the alert management, query and modification functionalities are separated from the mere building element data query functionalities. In turn, map visualization is provided thanks to the algorithms of the platform, which interact with the Google Maps APIs that offer this service. Finally, the algorithms of the platform also interact with ESRI servers, capable of providing both GIS and BIM visualization in a single viewer. For this purpose, an environment has been created in advance, in which several layers are introduced.

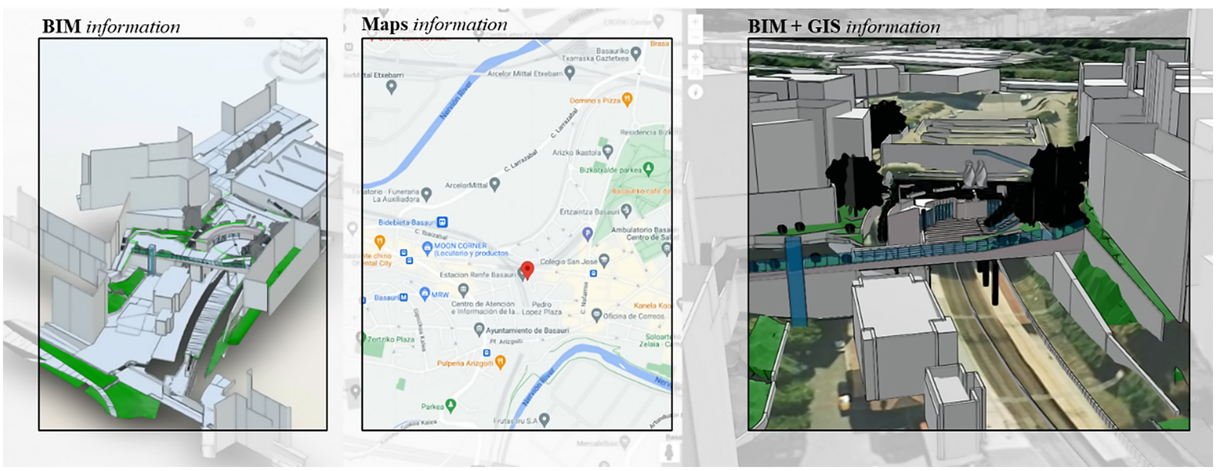
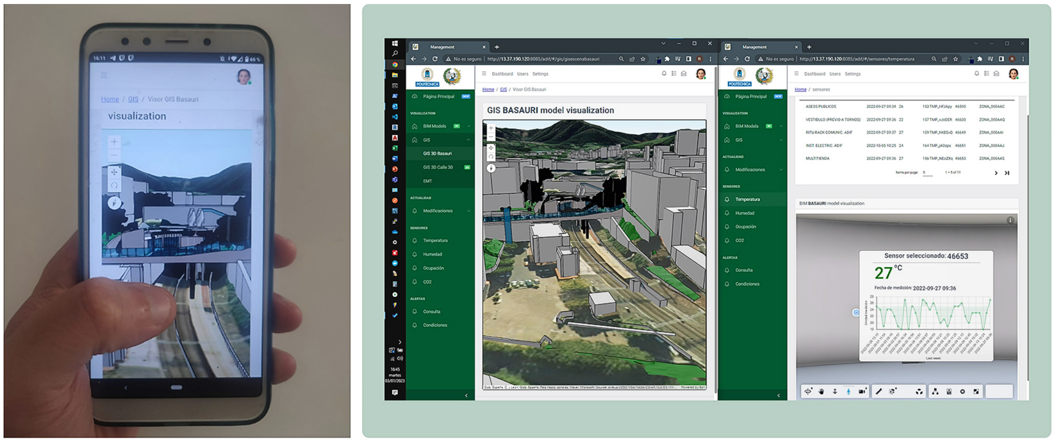


Figure 11. Static information provided by the platform. BIM, Maps and BIM + GIS



***Mobile device** visualization* ***Computer** visualization*

Figure 12. Online platform visualization

First, the digital terrain model is introduced. In this case, downloaded thanks to the public portal of the Autonomous Community of the Basque Country. A second layer is downloaded and inserted from the same portal, which provides a simple modelling of the buildings in the area under study. Finally, the BIM model of the new infrastructure is incorporated as the last layer. The final result of the implementation of all layers is shown in Figure 11. The whole platform is accessible through any device, as it is shown in Figure 12.

5. Discussion and future implementations

The developments and results shown in this project prove the real application of the BIM methodology in railway stations. In this sense and, in order to achieve the highest possible utility, a complete intelligent management platform has been developed, based not only on BIM methodology, but also on other technologies such as IoT, Cloud Computing or Big Data. This differentiates it from previous research, encompassing tools beyond those provided by the BIM methodology. Another important key is the actual implementation of the project in an existing infrastructure. Thanks to the construction and remodeling project of the Bidebieta-Basauri station by L.R.A. Infrastructures Consulting, which is the public tender awarded, It is provided to the research real feedback on the proposed functionalities, thus avoiding developments that in the end would not be useful for the real management of public administrations infrastructure.

The kind of results detailed in this paper provides one step forward through a total interoperability and an integrated management system. However, the application of this kind of tool is not easy. It must be considered that most of the built environment consist of lack of BIM or digital models. Even if they exist, is very likely that the modelling standard was different for each building. That is why one of the greatest BIM implementation barriers is the lack of interoperability between BIM models and management tools. In this sense, in order to improve the scalability of this platform and achieve a complete implementation in other infrastructures, a standardization of BIM definitions, elements characterization, communication protocols and databases typology usage is required.

Nevertheless, the result of the paper demonstrates the capability of linking real time data, BIM models and a total interoperability based on an own modelling standard. In this sense, the developed platform is able to provide all the functionalities for different models, with no more programming or modelling work needed. In this sense, the infrastructure user or manager will be able to access BIM or GIS data visualization, creation of alerts system based on real time measurements or get accurate material quantity take-offs for any BIM model of the infrastructure network. All in a total accessible way neither with Software installation needed nor BIM skills. However, more important than achieved results are the avenues of research that open up. For example, the possibility of applying AI linking BIM and GIS data with historic information saved into the databases like incidents, Temperatures, Occupancy and others.

More in depth regarding future developments and given the current tendency of public organizations to provide open data of their territory, this type of platform can achieve exponential growth, feeding on such data. This data provided by public portals can be found of various topics. Examples include traffic conditions, weather conditions or land mapping. In the particular case of railway stations, especially those managed by the Spanish administration, the platform could grow considerably with greater access to data that is currently closed. That is, access via API or JSON to data from already installed sensors, train timetables, timetable incidents or inventory and or technical data sheets of the infrastructure elements can lead to new functionalities in the platform. These new tools can focus on: (1) Management of elements for maintenance, being able to attach technical data sheets, revision schedules, incident systems or BIM location from the platform, (2) Staff management, being able to relate technical profiles with outstanding tasks, incidents or schedules, and (3) Management for the user of the infrastructure, being able to provide him with real-time information from sensors in the station, actual occupancy of the platform to be used and or temperature in the different spaces.

6. Conclusions

The construction project for the remodeling of the Bidebieta-Basauri station includes the northern covering and the new passenger building of the station, located on the Bilbao-Orduña commuter line. Being the contractor L.R.A. Infrastructures Consulting, a new management system for this infrastructure, based on BIM methodology, was carried out.

In order to avoid the usual barriers to BIM implementation, the development of a web management system was chosen in order to setup the basis for a DT to manage this type of infrastructure. This involves the development of Frontend and Backend programming, in order to create tools that would be useful for managing the infrastructure. To officially open the platform in production environment, the following methodology has been applied. In the first place, a brainstorming of possible functionalities that could be of interest to the company when it comes to managing the station. These functionalities are previously validated before starting the design and development of databases. In this regard, the necessary databases are developed for the previously agreed functionalities. Finally, to achieve real and accessible operation by the client or final user, it is essential to develop the necessary Backend and Frontend for each utility or tool described. Once completed, the platform is uploaded to a hired cloud server so that the client, company or user can check the results obtained. The project status at the time of writing did not reach the construction phase. Thus, the platform has been tested with all the models and versions that make up the total infrastructure and environment of the railway station of Bidebieta-Basauri. In that sense, the tool stands out for its accessibility, interoperability and simulated real time information.

The main result of the project is a web platform for BIM management that is fully accessible to any user. In this way, the main barriers to BIM implementation are broken down: hardware and software outlay, need of previous BIM knowledge or lack of data interoperability. The platform is designed for easy handling, with no need for special training in its use. The basic tools introduced have been: Querying historical and real-time measurements and creating alerts and conditions for these measurements. These functionalities allow linking Temperature, Humidity, Occupancy and or CO_2 measurements with BIM models of the infrastructure itself. These tools are accessible to any user through a simple registration on the platform, without any previous license or initial outlay.

In short, the final results obtained were: (1) Web platform fully accessible from any device, (2) Real-time sensor measurement functionality, (3) Functionality to create alerts and conditions for measurements, (4) BIM and GIS visualization linked to the previously described functionalities. These results provide a great basis for further development of new tools to be implemented in the platform, such as: inventory of elements, incident management systems, staff control, study of people flows or efficient energy management.

Acknowledgements

The authors gratefully acknowledge the financial support provided by Madrid Calle 30 for supporting the Enterprise University Chair Madrid Calle30-UPM.

Funding

This work received no external funding.

Author contributions

Conceptualization: R. M. P., T. L. R. A. and M. G. A.; Data curation: R. M. P. and J. J. C.; Formal analysis: R. M. P., M. G. A., T. L. R. A. and J. J. C.; Investigation: R. M. P., T. L. R. A. and M. G. A.; Methodology: R. M. P., M. G. A. and J. J. C; Resources: M. G. A.; Software: R. M. P., M. G. A. and J. J. C.; Supervision: T. L. R. A. and M. G. A.; Validation: M. G. A. and J. J. C.; Visualization: R. M. P. and M. G. A.; Writing – original draft: R. M. P. and M. G. A.; Writing – review & editing: T. L. R. A. and M. G. A. All authors have read and agreed to the published version of the manuscript.

Disclosure statement

The authors declare no conflict of interest.

References

Adif. (2021, February 11). *Adif adjudica la redacción del proyecto de cubrimiento y nuevo edificio de viajeros en la estación de Bidebieta-Basauri*. https://www.adif.es/w/adif-adjudica-la-redacci%C3%B3n-del-proyecto-de-cubrimiento-y-nuevo-edificio-de-viajeros-en-la-estaci%C3%B3n-de-bidebieta-basauri-

Akbarnezhad, A., Ong, K. C. G., & Chandra, L. R. (2014). Economic and environmental assessment of deconstruction strategies using building information modeling. *Automation in Construction*, *37*, 131–144. https://doi.org/10.1016/j.autcon.2013.10.017

Ali, N., Chen, S., Srikonda, R., & Hu, H. (2014). Development of concrete bridge data schema for interoperability. *Transportation Research Record*: *Journal of the Transportation Research Board*, *2406*(1), 87–97. https://doi.org/10.3141/2406-10

Al-Shalabi, F. A., Turkan, Y., & Laflamme, S. (2015). BrIM implementation for documentation of bridge condition for inspection. In *Proceedings of ICSC15*: *The Canadian Society for Civil Engineering 5th International/11th Construction Specialty Conference*, University of British Columbia, Vancouver, Canada. https://doi.org/10.14288/1.0076437

Arayici, Y., Coates, P., Koskela, L., Kagioglou, M., Usher, C., & O'Reilly, K. (2011). Technology adoption in the BIM implementation for lean architectural practice. *Automation in Construction*, *20*(2), 189–195. https://doi.org/10.1016/j.autcon.2010.09.016

Atencio, E., Araya, P., Oyarce, F., Herrera, R. F., Muñoz-La Rivera, F., & Lozano-Galant, F. (2022). Towards the integration and automation of the design process for domestic drinking-water and sewerage systems with BIM. *Applied Sciences*, *12*(18), Article 9063. https://doi.org/10.3390/app12189063

Aziz, Z., Riaz, Z., & Arslan, M. (2017). Leveraging BIM and Big Data to deliver well maintained highways. *Facilities*, *35*(13–14), 818–832. https://doi.org/10.1108/F-02-2016-0021

Bazán, Á. M., Alberti, M. G., Arcos Álvarez, A. A., Pavón, R. M., & Barbado, A. G. (2021). BIM-based methodology for the management of public heritage. Case study: Algeciras Market Hall. *Applied Sciences*, *11*(24), Article 11899. https://doi.org/10.3390/app112411899

Becerik-Gerber, B., Jazizadeh, F., Li, N., & Calis, G. (2012). Application areas and data requirements for BIM-enabled facilities management. *Journal of Construction Engineering and Management*, *138*(3), 431–442. https://doi.org/10.1061/(ASCE)CO.1943-7862.0000433

Biancardo, S. A., Intignano, M., Viscione, N., Guerra De Oliveira, S., & Tibaut, A. (2021). Procedural modeling-based BIM approach for railway design. *Journal of Advanced Transportation*, *2021*, Article 8839362. https://doi.org/10.1155/2021/8839362

Bradley, A., Li, H., Lark, R., & Dunn, S. (2016). BIM for infrastructure: An overall review and constructor perspective. *Automation in Construction*, *71*, 139–152. https://doi.org/10.1016/j.autcon.2016.08.019

Byun, N., Han, W. S., Kwon, Y. W., & Kang, Y. J. (2021). Development of BIM-based bridge maintenance system considering maintenance data schema and information system. *Sustainability*, *13*(9), Article 4858. https://doi.org/10.3390/su13094858

Carnevali, L., Lanfranchi, F., & Russo, M. (2019). Built information modeling for the 3D reconstruction of modern railway stations. *Heritage*, *2*(3), 2298–2310. https://doi.org/10.3390/heritage2030141

Cerovsek, T. (2011). A review and outlook for a 'Building Information Model' (BIM): A multi-standpoint framework for technological development. *Advanced Engineering Informatics*, *25*(2), 224–244. https://doi.org/10.1016/j.aei.2010.06.003

Cheng, J. C. P., Lu, Q., & Deng, Y. (2016). Analytical review and evaluation of civil information modeling. *Automation in Construction*, *67*, 31–47. https://doi.org/10.1016/j.autcon.2016.02.006

Ciccone, A., Stasio, S. D., Asprone, D., Salzano, A., & Nicolella, M. (2022). Application of openBIM for the management of existing railway infrastructure: Case study of the Cancello–Benevento railway line. *Sustainability*, *14*(4), Article 2283. https://doi.org/10.3390/su14042283

Costin, A., Adibfar, A., Hu, H., & Chen, S. S. (2018). Building Information Modeling (BIM) for transportation infrastructure – Literature review, applications, challenges, and recommendations. *Automation in Construction*, *94*, 257–281. https://doi.org/10.1016/j.autcon.2018.07.001

Dang, N. S., Rho, G. T., & Shim, C. S. (2020). A master digital model for suspension bridges. *Applied Sciences*, *10*(21), Article 7666. https://doi.org/10.3390/app10217666

Dayan, V., Chileshe, N., & Hassanli, R. (2022). A scoping review of information-modeling development in bridge management systems. *Journal of Construction Engineering and Management*, *14*8(9), Article 03122006. https://doi.org/10.1061/(ASCE)CO.1943-7862.0002340

Dibernardo, S. (2012). Integrated modeling systems for bridge asset management - Case study. In *Proceedings of the 2012 Structures Congress* (pp. 483–493). https://doi.org/10.1061/9780784412367.043

Eneyew, D. D., Capretz, M. A., & Bitsuamlak, G. T. (2022). Toward smart-building digital twins: BIM and IoT data integration. *IEEE Access*, *10*, 130487–130506. https://doi.org/10.1109/ACCESS.2022.3229370

Fortino, G., & Savaglio, C. (2023). Integration of Digital Twins & Internet of Things. In N. Crespi, A. T. Drobot, & R. Minerva (Eds.), *The digital twin* (pp. 205–225). Springer. https://doi.org/10.1007/978-3-031-21343-4_8

Gu, N., & London, K. (2010). Understanding and facilitating BIM adoption in the AEC industry. *Automation in Construction*, *19*(8), 988–999. https://doi.org/10.1016/j.autcon.2010.09.002

Häußler, M., Esser, S., & Borrmann, A. (2021). Code compliance checking of railway designs by integrating BIM, BPMN and DMN. *Automation in Construction*, *121*, Article 103427. https://doi.org/10.1016/j.autcon.2020.103427

Hüthwohl, P., Lu, R., Brilakis, I., & O'rourke, L. (2016). *Challenges of bridge maintenance inspection*. https://repository.lboro.ac.uk/articles/Challenges_of_bridge_maintenance_inspection/9425816/files/17046737.pdf

Jang, K., Kim, J., Ju, K., & An, Y. (2021). Infrastructure BIM platform for lifecycle management. *Applied Sciences*, *11*(21), Article 10310. https://doi.org/10.3390/app112110310

Jensen, J. S. (2020). Innovative and sustainable operation and maintenance of bridges. *Structure and Infrastructure Engineering*, *16*(1), 72–83. https://doi.org/10.1080/15732479.2019.1604772

Kim, J. U., Kim, Y. J., Ok, H., & Yang, S. H. (2016). A study on the status of infrastructure BIM and BIM library development. In *Proceedings of 2015 International Conference on Computational Science and Computational Intelligence* (*CSCI 2015*) (pp. 857–858), Las Vegas, NV, USA. IEEE. https://doi.org/10.1109/CSCI.2015.52

Kurwi, S., Demian, P., & Hassan, T. M. (2017). Integrating BIM and GIS in railway projects: A critical review. In P. W. Chan, & C. J. Neilson (Eds.), *33rd Annual ARCOM Conference* (pp. 45–53), Cambridge, UK.

Lin, Y., Lo, N., Hu, H., & Hsu, Y. (2020). Collaboration-based BIM model development management system for general contractors in infrastructure projects. *Journal of Advanced Transportation*. https://doi.org/10.1155/2020/8834389

Liu, Q., & Gao, T. (2017). The information requirements for transportation industry's facilities management based on BIM. *The Open Construction* & *Building Technology Journal*, *11*(1), 136–141. https://doi.org/10.2174/1874836801711010136

Liu, Q., Liu, Y., Chen, X., Ji, S., & Hu, T. (2019). Construction simulation and real-time monitoring research of concrete dam based on BIM. *IOP Conference Series*: *Earth and Environmental Science*, *304*(5), Article 052056. https://doi.org/10.1088/1755-1315/304/5/052056

Luo, S., Yao, J., Wang, S., Wang, Y., & Lu, G. (2022). A sustainable BIM-based multidisciplinary framework for underground pipeline clash detection and analysis. *Journal of Cleaner Production*, *374*, Article 133900. https://doi.org/10.1016/j.jclepro.2022.133900

Mawlana, M., Vahdatikhaki, F., Doriani, A., & Hammad, A. (2015). Integrating 4D modeling and discrete event simulation for phasing evaluation of elevated urban highway reconstruction projects. *Automation in Construction*, *60*, 25–38. https://doi.org/10.1016/j.autcon.2015.09.005

Miyamoto, K. (2016). A framework for data coordination method of maintenance data and 3D conceptual model on CIM based database. In *Proceedings of the International Conference on Computing in Civil and Building Engineering* (*ICCCBE*) (pp. 6–8).

Moshynskyi, V., Striletskyi, P., & Trach, R. (2022). Application of the building information modelling (BIM) for bridge structures. *Acta Scientiarum Polonorum Architectura*, *20*(4), 3–9. https://doi.org/10.22630/aspa.2021.20.4.29

Neves, J., Sampaio, Z., & Vilela, M. (2019). A case study of BIM implementation in rail track rehabilitation. *Infrastructures*, *4*(1), Article 8. https://doi.org/10.3390/infrastructures4010008

Nguyen, D. C., Nguyen, T. Q., Jin, R., Jeon, C. H., & Shim, C. S. (2022). BIM-based mixed-reality application for bridge inspection and maintenance. *Construction Innovation*, *22*(3), 487–503. https://doi.org/10.1108/CI-04-2021-0069

Noor, B. A., Yi, S., & Kazmi, S. H. A. (2018). Revit-based automation modeling for intermediate railway station. In *Proceedings of 2nd International Conference on Cybernetics*, *Robotics and Control* (pp. 162–166), Chengdu, China. IEEE. https://doi.org/10.1109/CRC.2017.34

Omoregie, A., & Turnbull, D. E. (2016). Highway infrastructure and building information modelling in UK. *Proceedings of the Institution of Civil Engineers*: *Municipal Engineer*, *169*(4), 220–232. https://doi.org/10.1680/jmuen.15.00020

Panah, R. S., Kioumarsi, M., Lichti, D., & Maalek, S. (2021). Application of Building information modelling (BIM) in the health monitoring and maintenance process: A systematic review. *Sensors*, *21*(3), Article 837. https://doi.org/10.3390/s21030837

Pasetto, M., Giordano, A., Borin, P., & Giacomello, G. (2020). Integrated railway design using Infrastructure-Building Information Modeling. The case study of the port of Venice. *Transportation Research Procedia*, *45*, 850–857. https://doi.org/10.1016/j.trpro.2020.02.084

Pavón, R. M., Arcos Alvarez, A. A., & Alberti, M. G. (2020). BIM-based educational and facility management of large university venues. *Applied Sciences*, *10*(22), Article 7976. https://doi.org/10.3390/app10227976

Redmond, A., Hore, A., Alshawi, M., & R West. (2012). Exploring how information exchanges can be enhanced through Cloud BIM. *Automation in Construction*, *24*, 175–183. https://doi.org/10.1016/j.autcon.2012.02.003

Renfe. (2022). Estaciones de Tren en España. https://www.renfe.com

Sankaran, B., France-Mensah, J., & O'Brien, W. (2016). Data integration challenges for CIM in large infrastructure: a case study of Dallas Horseshoe Project. In *Proceedings of the 16th International Conference on Computing in Civil and Building Engineering* (*ICCCBE 2016*), Tokyo, Japan.

Shim, C., Yun, N., & Song, H. (2011). Application of 3D bridge information modeling to design and construction of bridges. *Procedia Engineering*, *14*, 95–99. https://doi.org/10.1016/j.proeng.2011.07.010

Singh, V., Gu, N., & Wang, X. (2011). A theoretical framework of a BIM-based multi-disciplinary collaboration platform. *Automation in Construction*, *20*(2), 134–144. https://doi.org/10.1016/j.autcon.2010.09.011

Tanaka, F., Hori, M., & Onosato M. (2016). Bridge information model based on IFC standards and web content providing system for supporting an inspection process. In *16th International Conference on Computing in Civil and Building Engineering* (*ICCCBE2016*), Osaka, Japan.

Tawelian, L. R., & Mickovski, S. B. (2016). The implementation of geotechnical data into the BIM process. *Procedia Engineering*, *143*, 734–741. https://doi.org/10.1016/j.proeng.2016.06.115

Vitásek, S., & Matějka, P. (2017). Utilization of BIM for automation of quantity takeoffs and cost estimation in transport infrastructure construction projects in the Czech Republic. *IOP Conference Series*: *Materials Science and Engineering*, *236*(1), Article 012110. https://doi.org/10.1088/1757-899X/236/1/012110

Wang, G., & Zhang, Z. (2021). BIM implementation in handover management for underground rail transit project: A case study approach. *Tunnelling and Underground Space Technology*, *108*, Article 103684. https://doi.org/10.1016/j.tust.2020.103684

Wang, B., Wang, Q., Cheng, J. C. P., Song, C., & Yin, C. (2022). Vision-assisted BIM reconstruction from 3D LiDAR point clouds for MEP scenes. *Automation in Construction*, *133*, Article 103997. https://doi.org/10.1016/j.autcon.2021.103997

Xu, X., Wang, G., Cao, D., & Zhang, Z. (2020). BIM adoption for facility management in urban rail transit: An innovation diffusion theory perspective. *Advances in Civil Engineering*, *2020*, Article 886422. https://doi.org/10.1155/2020/8864221

Yabuki, N. (2016). Current status of civil construction information modeling (CIM) initiative and projects in Japan. In *Civil Engineering Conference in the Asian Region* (*CECAR 7*), Waikiki, Oahu, Hawaii.

Zhou, Z. (2022). An intelligent bridge management and maintenance model using BIM technology. *Mobile Information Systems*, *2022*, Article 7130546. https://doi.org/10.1155/2022/7130546

Zhou, Y., Bao, T., Shu, X., Li, Y., & Li, Y. (2023). BIM and ontology-based knowledge management for dam safety monitoring. *Automation in Construction*, *145*, Article 104649. https://doi.org/10.1016/j.autcon.2022.104649