Article
Journal of Computational Technologies
June 2022, Vol.6 No.17 10-17

Multi-agent system integrated to a Smart campus for mobile device detection and lamp management

Sistema multiagente integrado a un Smart campus para la detección de dispositivos móviles y administración de lámparas

DE LA CRUZ-PONCE, José Fabricio†*, LINO-RAMÍREZ, Carlos, ZAMUDIO-RODRÍGUEZ, Víctor Manuel and GUTIÉRREZ-HERNANDEZ, David Asael

Tecnológico Nacional de México, Campus León, Graduate Studies and Research Division, León, Guanajuato, Mexico.

ID 1st Author: *José Fabricio, de la Cruz-Ponce* / **ORC ID:** 0000-0002-3581-9871, **CVU CONACYT ID:** 1031013

ID 1st Co-author: *Carlos, Lino-Ramírez* / **ORC ID:** 0000-0002-6415-8435, **CVU CONACYT ID:** 395781

ID 2nd Co-author: *Víctor Manuel, Zamudio-Rodríguez* / **ORC ID:** 0000-0002-9246-7999, **CVU CONACYT ID:** 70912

ID 3rd Co-author: *David Asael, Gutiérrez-Hernandez* / **ORC ID:** 0000-0002-9374-5110, **CVU CONACYT ID:** 167864

DOI: 10.35429/JOCT.2022.17.6.10.17 Received: January 15, 2022; Accepted: June 30, 2022

Abstract

It describes the progress that has been made in the "Smart Campus" project, which is being carried out at the Instituto Tecnológico de León. The objective of this project is to improve the stay of the people who walk through the institution on a daily basis, using intelligent agents, artificial intelligence, the internet of things, software, databases, hardware and platforms that the University already has, using the traditional architecture to its advantage and adding certain elements that allow us to make this proposal a reality. The project aims to census the classrooms to improve the stay of people who walk through the institution every day, obtain environmental data from the classrooms in real time and manipulate the current that can be delivered to a lamp. In this article, we talk in depth about the construction process of the architecture, in which the whole project will be based in a technical way, but at the same time making it as understandable as possible for any type of public.

IoT, Smart campus, Environmental, Agents, Platforms

Resumen

Se describe el avance que se ha tenido en el proyecto “Smart Campus”, el cual se está llevando a cabo en el Instituto Tecnológico de León. Este proyecto tiene como objetivo, mejorar la estancia de las personas que día a día recorren dicha institución, apoyándonos con agentes inteligentes, inteligencia artificial, el internet de las cosas, software, bases de datos, hardware y plataformas con las que ya cuenta la Universidad, usando a favor la arquitectura tradicional y agregando ciertos elementos que nos permiten hacer realidad esta propuesta. El proyecto tiene como objetivo, censar las aulas para mejorar la estancia de las personas que día a día recorren dicha institución, obtener datos ambientales de las aulas en tiempo real y manipular la corriente que se le puede entregar a una lampara. En este artículo, se habla a fondo sobre el proceso de construcción de la arquitectura, en la cual estará cimentado todo el proyecto de manera técnica, pero al mismo tiempo haciéndolo más entendible posible para cualquier tipo de público.

Internet de las cosas, Campus inteligente, Medio ambiente, Agentes, Plataformas

Citation: DE LA CRUZ-PONCE, José Fabricio, LINO-RAMÍREZ, Carlos, ZAMUDIO-RODRÍGUEZ, Víctor Manuel and GUTIÉRREZ-HERNANDEZ, David Asael. Multi-agent system integrated to a Smart campus for mobile device detection and lamp management. Journal of Computational Technologies. 2022. 6-17:10-17.

* Correspondence to Author: (E-mail: m14240570@leon.tecnm.mx)
† Researcher contributed as first author.

June 2022, Vol.6 No.17 10-17

Introduction

The word 'Smart' comes from the English language, which we could interpret as 'intelligent', nowadays it is used as a misleading marketing strategy, smart lamps are sold, smart locks, among other things, but we have stopped to think what is smart about them, how can we define that something is smart or not, and then, why this project has that name, maybe we put it because it is fashionable, like the Internet of Things and Industry 4.0.

The Internet of things and intelligent environments are issues that we see closer and closer in our daily lives, because they make use of technologies that years ago were little used in everyday life, such as an internet connection or simply a mobile phone in which the screen time was extremely minor compared to the use that is given to it today, As time goes by, what we saw in films and series such as working from home with video calls where people from all over the world interact and speak different languages or having a humanoid personal assistant, generated feelings of uncertainty and probably excitement, but things that in the last century were big problems, there are probably young people today who never had to know them and now they deal with other types of impediments.

New technologies are of great help and solve many of the problems we have, however, they bring us new problems, imagine the fact of building the nuclear bomb, and ending with an enemy and the problem of radiation that it will leave for years, likewise in this project we propose different ways in which we can help the people who belong to our institution with the use of technologies such as intelligent agents, sensors, databases, to say a few. For us it is important to point out that we use the word intelligent in reference to the fact that at some point we want our technological ecosystem to be able to remember and use these experiences to make decisions and generate recommendations for any individual belonging to the institution.

Intelligent agent

An intelligent agent is considered to be a system that can perform an automated process to achieve a goal, with the condition that the system recognises and acts on an environment, as it can perceive information, communicate or receive data and perform a corresponding action (Serna et al., 2019).

Multi-agent

Each agent is able to understand its situation and adapts to changing environments through self-conFiguration. Situational awareness is achieved through learning and contextual modelling of event data. Once a set of contextual bases is learned from the high-dimensional event data, different scenarios can be represented by the clustered contextual coefficients. Agents can then perceive the situation and locate regions of interest (RoIs) through the identified scenarios. Each agent has a behavioural state machine and a behavioural library; it chooses a certain behaviour according to the target individual and the behaviour of other agents (Sun et al., 2013).

Smart City

The concept of "Smart City" appeared as the application of automatic environmental data collection and processing to achieve efficient management of urban areas and their resources and assets. This approach is supported by the massive application of information and communication technologies (ICTs) and the Internet of Things (IoT) paradigm, in which a large number of distributed devices are connected to transfer the collected data (Fortes et al., 2019).

Smart Campus

A Smart Campus is a smart city-like infrastructure that makes use of Internet of Things (IoT) solutions to connect, monitor, control, optimise and automate a university's systems.

Today, a smart campus represents a challenging scenario for IoT networks, especially in terms of cost, coverage, availability, latency, security, energy consumption and scalability (Fraga-Lamas et al., 2019).

ISSN 2523-6814

A smart campus must immerse itself into the digital era and improve processes with the help of artificial intelligence (Cesur et al., 2019).

State of the art

In the work of the University of Coruña (UDC) they implement LoRaWAN which is a technology for low-power networks and wide areas. Its objective is to measure the radio frequency intensity in different areas and to propose an architecture for future works such as a mobility pattern detection system, a smart irrigation solution and an implementation of smart traffic monitoring (Fraga-Lamas et al., 2019).

The paper (Smart CEI Moncloa) presents an IoT-based platform for monitoring the flow of people and the environment in a Smart University Campus deployed in the CEI Moncloa campus, with special emphasis on the main technological challenges faced and solutions adopted, as well as the functionality, services and potential offered by the platform (Alvarez-Campana et al., 2017).

In another work it is proposed that a smart room is a way to create a smart campus, using ESP8266 and sensors to control the light in a room (Yuliansyah et al., 2019), in the same way it is possible to use ESP32 to monitor the environment (Babiuch & Postulka, 2021), based on these works a smart room can be considered as a smart classroom.

On the other hand, for the part of mobile device detections they use wifi waves (Gómez R. & Pedraza, 2018).

There are different ways in which a Smart Campus can be built our approach was to put together ideas from different works and create a multi-agent system focused on the Smart Campus, we get to use several technologies for communication such as mobile networks for end users and low-power networks for agents.

Agents

For the construction of each agent, 6 sensors, an actuator, an ESP32 with Arduino and a Raspberry Pi 3 were used, as shown in Figures 1 and 2. An ESP32 is used because the data provided by some sensors are analogue, which cannot be interpreted with a Raspberry Pi 3.

Sensors, actuator and communication

A presence or movement sensor (PIR), a light sensor (BH1750), a noise sensor (KY-037), a temperature and humidity sensor (DHT11), a gas sensor (Sensor MQ-7) and an air quality sensor (MQ-135) were used.

A dimmer module was used that works with a triac whose function is to detect the zero crossing of the alternating current in order to manipulate the light intensity of a lamp.

An ESP32 was used to read the sensors and manipulate the light intensity.

Processing

The ESP32 gets the data from the environment with the sensors and the data is sent to the Raspberry. The Raspberry calculates the required intensity and sends the intensity values to the ESP32 which the ESP32 then sends to the dimmer.

The intensity calculation needs 2 inputs which is the ambient light intensity and the current time. The calculation gives an intensity between 20 and 80, which are values where the lamp works correctly and these same values the ESP32 converts them to milliseconds to send pulses to the dimmer and this in turn turns on the lamp.

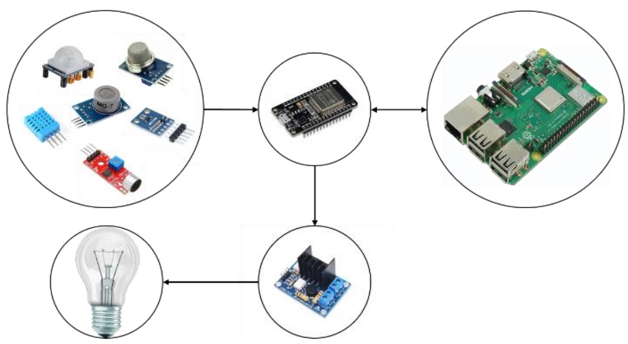


Figure 1 Agent structure
Source: Own elaboration

ISSN 2523-6814

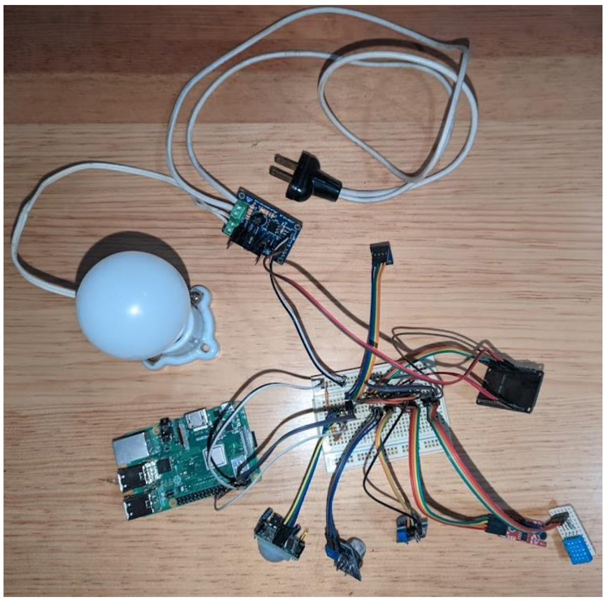


Figure 2 Assembled agent
Source: Own elaboration

Development

The system is based on the architecture proposed in the work "Propuesta de una arquitectura para un Smart Campus Universitario" (CRUZ-PARADA et al., 2020), using sockets, agents, database and end devices, as shown in Figure 3.

Socket

We worked at the same time in the construction of the different actors that we have in the Smart Campus, as shown in Figure 3, we proposed the use of a "Socket" server which will allow us to communicate all our actors within the project, so we will begin by showing its process.

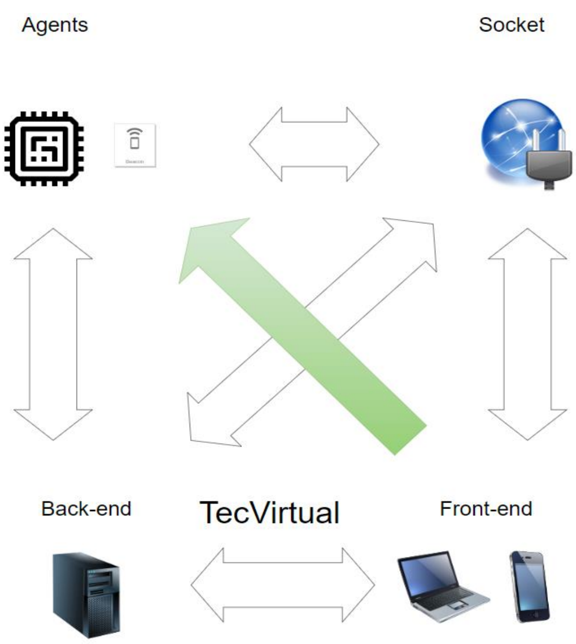


Figure 3 Architecture proposal for the Smart campus
Source: (Cruz-parada et al., 2020)

The first step in the construction was to define which language our socket server would be built on and which library would be used as a base in order to obtain a stable platform that would allow us to have shorter development times using the tools that the repository already has. There are as many alternatives as there are for each language, each one already has specific functions to carry out these tasks. We chose to use the socket.io library, given that it uses the JavaScript programming language and that a large part of the architecture is based on this language.

Once the tool has been defined, we start by making the connection as shown in Figure 4, we define the functions with which we will detect who is connected, differentiating between users and agents, with the possibility of adding more functionality later.

```
io.sockets.on('connection', function (socket) {

    // When human user join to the socket service
    socket.on('join', (user) => {
        removeUser(user.id);
        socket.join(user.type);
        addUser(socket.id, user.id, user.type);
        getAgentsOnlineDB(socket.id);
    });

    // Update the position of the user's match with an agent
    socket.on('here',(macAddressAgent)=> {
        try{
            agentId = getIdFromMac(macAddressAgent);
            user = getUserBySocketId(socket.id);
            updateUserPosition(agentId,user.id);
        } catch (e) {}
    });

    // When agent user join to the socket service
    socket.on('joinAgent', (agentId) => {
        removeAgent(agentId);
        addAgent(socket.id, agentId, 0);
        setAgentOnline(agentId,true);
    });
```

Figure 4 Socket source code
Source: Own elaboration

Mobile application

At the same time we started with the development of the mobile application, which will serve as a link between the user and the Smart Campus, the decision on which development platform we would use was simple, we needed a fast and multiplatform development so we opted for the use of Ionic Cordova, which allows us to develop an application in Angular and reuse the code by compiling it on both the Android and iOS platforms.

We used as a basis the test application with which Ionic shows the full functionality that the platform has, this framework, so it was enough to add our functionality that makes the connection with the backend, which already has the institution, to the views and change some texts, to quickly have a robust application, once we have a functional app we start connecting it with the socket previously described as shown in Figure 5.

```
start() {
    this.socket = io(environment.urlServerSocket);
    this.startScaning()
    console.log('empieza conexión');
    this.socket.on('connect', ()=> {
        this.userData.getUsername().then(username => {
            this.socket.emit('join', {curp:username,sala:1} , (error) => {
                if (error) {} else {}
            });
        });
    });

    this.socket.on('connect_error', async (error) => {
      this.scanning = false;
    });

    this.socket.on('setAgentOnline', (list)=>{
        this.userData.setAgentsOnlineList(list).then(()=>{
            this.scanning = true;
        })
    });
}
```

Figure 5 Source code for connecting application to socket
Source: Own elaboration

Agent

At the same time we worked on the development of the agent, although later we will try to have different types, for the moment we are working on a base agent, which can be used to later add enough code to change the type without losing the structure of the other agents. We will mount our agent on a Raspberry Pi 3 device since it already has the bluetooth connectivity we need for the project and is using a very suitable operating system for this type of task based on Linux. Together with this device we will use the Python programming language and a set of libraries which allow us to create an agent that is constantly listening and at the same time we can add artificial intelligence modules, modules that we will use in future works in order to achieve the objective of the Smart Campus. We start by generating an id with which we can identify our agent and then start listening as shown in Figure 6, once the agent started to run the first thing it does is to establish communication with our socket server as shown in Figure 7, and once the connection to the socket is made, the socket function is called which joins us to the list of agents and at the same time updates the status of the same in the database. As shown in Figure 8, part of the socket code.

```
# Get ID
mac = bluetooth.read_local_bdaddr()
my_id = int(mac[-1].replace(':', ''), 16)

# Turn On Agent
agents_list = list()
agente = Agente(AID(name=str(my_id)))
agents_list.append(agente)

start_loop(agents_list)
```

Figure 6 Obtain identifier to turn on the agent
Source: Own elaboration

```
def on_start(self):
    super(ComportTemporal, self).on_start()

    # Socket
    sio.connect(os.getenv('SOCKET_URI'))
```

Figure 7 Connecting the agent to the socket
Source: Own elaboration

```
// When agent user join to the socket service
socket.on('joinAgent', (agentId) => {
    removeAgent(agentId);
    addAgent(socket.id, agentId, 0);
    setAgentOnline(agentId,true);
});
```

Figure 8 Adding an agent to the connected list of the socket
Source: Own elaboration

Once everything is connected, the socket sends the application a list of identifiers, once the application receives the list, it starts searching for devices using bluetooth as shown in Figure 9, and once it detects a match with an agent it makes the call to update the user's position associated with an agent, which would help for future specialised services, such as automatic roll call, among others.

```
start() {
    this.devices = [];
    this.ble.scan([],20).subscribe(device => {
        this.onDeviceDiscovered(device)
    });
}

onDeviceDiscovered(device){
    this.ngZone.run(()=>{
        this.devices.push(device);
        this.checkAgentList(device)
    });
}

checkAgentList(device) {
    this.userData.getAgentOnlineList().then((list)=>{
        list.forEach((agentId,index)=>{
            if (agentId == device.id) {
                this.socket.here(agentId);
                return;
            }
        })
    })
}
```

Figure 9 Search agents (Application source code)
Source: Own elaboration

Finally, it detects when a user disconnects and updates their status within this ecosystem as shown in Figure 10.

```
// Triggered when a user disconect
socket.on('disconnect', () => {
    let user = getUserBySocketId(socket.id);

    if (user) {
        users.removeUser(user.id);
    } else {
        let agent = getAgentBySocketId(socket.id);

        if(agent) {
            agents.removeUser(agent.id);
            setAgentStatus(agent.id,false);
        }
    }
})
```

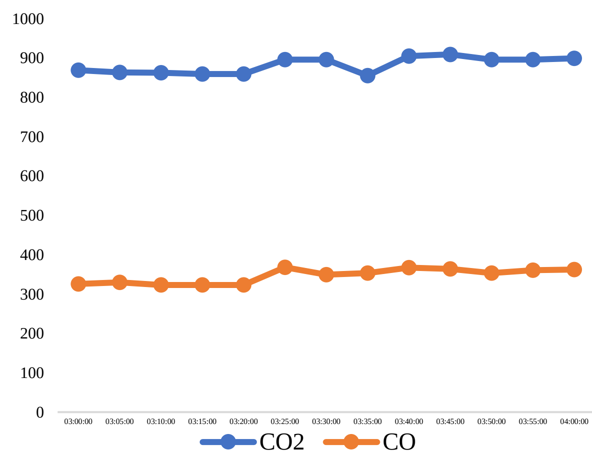
Figure 10 Detecting disconnected agents (Socket source code)
Source: Own elaboration

Results

The results of this stage of the project are in line with what was planned from the beginning, and leave everything ready to start with the next phases of the project, we can see in Figures 11 and 14, we have in real time the status of our agents and they can be consulted at any time, besides being able to generate graphs of environmental data such as air quality as shown in Figure 1, which could be useful in times of COVID.

_id: 242606372691727
status: "online"
build: "Postgrado"
location: "Laboratorio 1"
dates: Array
  0: Object
    date: 2022-02-28T02:55:48.537+00:00
    co2: "869"
    co: "326"
    noise: "59"
    humidity: "21.00"
    temperature_c: "25.50"
    temperature_f: "77.90"
    heat_index_c: "24.65"
    heat_index_f: "76.38"

Figure 11 Real-time agent status
Source: Own elaboration



Graphic 1 Air quality graph for 1 hour on February 28, 2022
Source: Own elaboration

Likewise, as shown in Figure 12, we have the last registered position of the users which is linked to the agent where they were and in Figure 15 we have the position of the users in the application.

_id: ObjectId("621c9c8a2f84b1e21da05a33")
agent: 354653214445
user: "M14240580"
updatedAt: 2022-02-28T09:57:30.531+00:00

_id: ObjectId("621ca5bdad563eed1f632969")
agent: 242606372691727
user: "M14240570"
updatedAt: 2022-02-28T10:36:45.398+00:00

Figure 12 Last location of a user
Source: Own elaboration

ISSN 2523-6814

We have a functional application on Android and iOS, which already has a connection with the services that the institute's old platforms already had, such as logging into the system, checking the student's timetable, to mention a couple, thanks to the architecture that was planned based on the strategy of using one or more api, and it is also prepared to add the services that are necessary as the project progresses, as shown in Figures 13, 14 and 15.

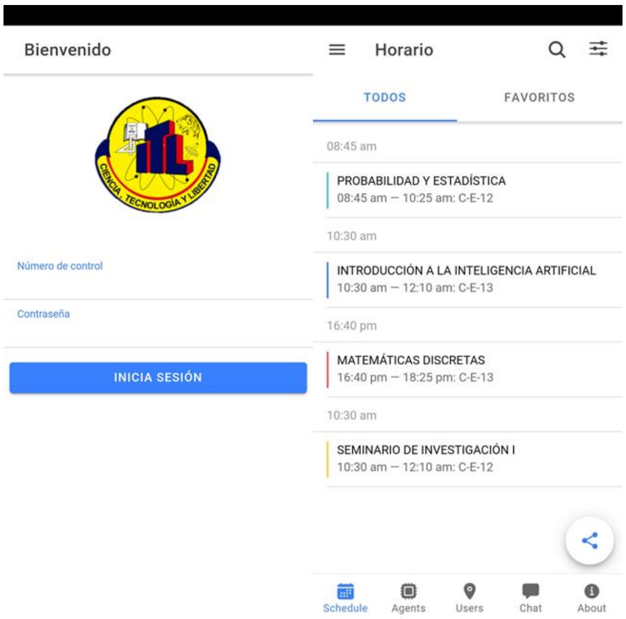


Figure 13 Login application and class schedule on Android
Source: Own elaboration

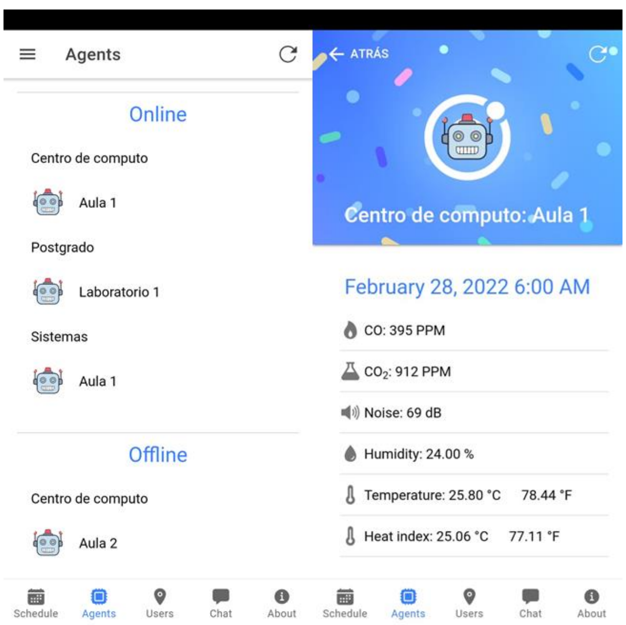


Figure 14 Real-time agent status in Android application
Source: Own elaboration

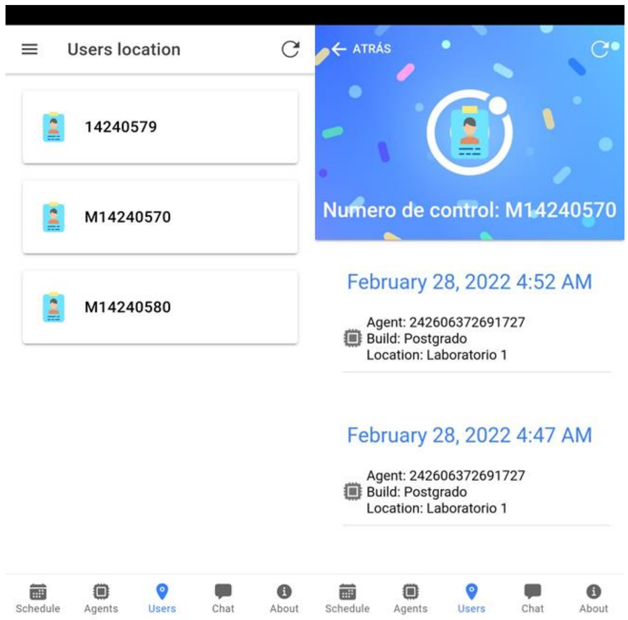


Figure 15 Location of the users application on Android
Source: Own elaboration

Acknowledgements

I would like to thank the Graduate Studies and Research Division of the Tecnológico Nacional de México in León for allowing us to develop this project.

I would like to thank the Consejo de Ciencia y Tecnología (CONACYT) for the financial support which allows me to study a Master's degree in Computer Science.

Funding

Funding: The present work has been funded by CONACYT [grant number 774653].

Conclusions

The development of the project described in the document shows how we acted on the need to create an intelligent university campus using agents installed in different areas of the institute, this will allow us in the future to develop modules with artificial intelligence in the agents already installed to improve security, improve welfare, improve academic performance, etc.. And all of this focused on the users of the environment. On the other hand, the location of the devices within the institution may or may not be shared due to privacy issues. Showing the location in the application is provisional and for testing purposes since the location of the users will not be visible to any user.

Some of the future works are the automatic roll call, regulating the temperature of an environment, detecting if a student or teacher is isolated and providing support, making intelligent recommendations according to their subjects, etc.

References

CRUZ-PARADA, J. M., ZAMUDIO-RODRIGUEZ, V. M., LINO-RAMÍREZ, C. & GUTIERREZ-HERNANDEZ, D. A. (2020, 30 junio). Propuesta de una arquitectura para un Smart Campus Universitario. Revista de Innovación Sistemática, 21-27. https://doi.org/10.35429/jsi.2020.14.4.21.27

Serna, B., Baltazar, R., Cruz-Parada, P., Meza, J., Manríquez, J. & Zamudio, V. (2019, 20 junio). Use of Intelligent Agent Through Low-Cost Brain–Computer Interface to Analyze Attention and Meditation Levels by Gender. Agents and Multi-agent Systems: Technologies and Applications 2019, 163-174. https://doi.org/10.1007/978-981-13-8679-4_14

Sun, Q., Yu, W., Kochurov, N., Hao, Q. & Hu, F. (2013, 19 agosto). A Multi-Agent-Based Intelligent Sensor and Actuator Network Design for Smart House and Home Automation. Journal of Sensor and Actuator Networks, 2(3), 557-588. https://doi.org/10.3390/jsan2030557

Fraga-Lamas, P., Celaya-Echarri, M., Lopez-Iturri, P., Castedo, L., Azpilicueta, L., Aguirre, E., Suárez-Albela, M., Falcone, F. & Fernández-Caramés, T. M. (2019, 26 julio). Design and Experimental Validation of a LoRaWAN Fog Computing Based Architecture for IoT Enabled Smart Campus Applications. Sensors, 19(15), 3287. https://doi.org/10.3390/s19153287

Alvarez-Campana, M., López, G., Vázquez, E., Villagrá, V. & Berrocal, J. (2017, 8 diciembre). Smart CEI Moncloa: An IoT-based Platform for People Flow and Environmental Monitoring on a Smart University Campus. Sensors, 17(12), 2856. https://doi.org/10.3390/s17122856

Fortes, S., Santoyo-Ramón, J., Palacios, D., Baena, E., Mora-García, R., Medina, M., Mora, P. & Barco, R. (2019, 18 marzo). The Campus as a Smart City: University of Málaga Environmental, Learning, and Research Approaches. Sensors, 19(6), 1349. https://doi.org/10.3390/s19061349

Yuliansyah, H., Corio, D., Yunmar, R. A. & Kahar Aziz, M. R. (2019, 10 mayo). Smart-Room Technology Implementation Based on Internet of Things Toward Smart Campus in Institut Teknologi Sumatera. IOP Conference Series: Earth and Environmental Science, 258, 012053. https://doi.org/10.1088/1755-1315/258/1/012053

Gómez R., C. A. & Pedraza, L. F. (2018, junio). Ubicación de dispositivos móviles en ambientes interiores por medio de análisis de radiación de redes WiFi y deformaciones de campo magnético. Ingeniare. Revista chilena de ingeniería, 26(2), 203-212. https://doi.org/10.4067/s0718-33052018000200203

Cesur, R., Torkul, O., Cedimoğlu, S. H. & Uçar, S. (2019, 1 diciembre). Intelligent Campus Implementation For Smart Cities. Sakarya University Journal of Science, 1218-1224. https://doi.org/10.16984/saufenbilder.376218

Babiuch, M. & Postulka, J. (2021, 18 agosto). Smart Home Monitoring System Using ESP32 Microcontrollers. Internet of Things. https://doi.org/10.5772/intechopen.94589