Scientific Paper / Artículo Científico

UNIVERSIDAD POLITÉCNICA SALESIANA ECUADOR
INGENIUS
Revista de Ciencia y Tecnología
https://doi.org/10.17163/ings.n20.2018.10
pISSN: 1390-650X / eISSN: 1390-860X

Access System Using an RFiD Card and Face Verification

Sistema de acceso usando una tarjeta RFiD y verificación de rostro

José Ignacio Vega-Luna[1,*], Francisco Javier Sánchez-Rangel[1], Gerardo Salgado-Guzmán[1], Mario Alberto Lagos-Acosta[1]

Abstract

This paper presents the development of an access system to a data center using a RFiD card and verification of the user's face. The system consists of three input modules and a central module. The objective was to design a system to transmit, from each input module to the central module, the universal unique identifier of the RFiD card or UUID for its acronym in English and the user's face image to consult in a MySQL database and in a directory of photographs if the user can access the corresponding area of the input module. Each input module consists of a Raspberry Pi 3 B+ card, an RFiD card reader, a video camera and a liquid crystal display or LCD for its acronym in English. The central module is composed of the same elements as the input modules and has a touch screen used in the user interface instead of an LCD screen. The communication between the nodes is WiFi, achieving a precision of 99.2% in the verification of the face and a response time of 180 ms using 310 trained photographs.

Keywords: Face verification, MySQL, Raspberry Pi 3 B+, RFiD, touchscreen, video camera.

Resumen

En este trabajo se presenta el desarrollo de un prototipo de sistema de acceso a un centro de datos usando como identificación una tarjeta de radio frecuencia o RFiD y verificación del rostro del usuario. El sistema se compone de tres módulos de entrada y un módulo central. El objetivo fue diseñar un sistema para transmitir, desde cada módulo de entrada al módulo central, el identificador único universal de la tarjeta RFiD o UUID y la imagen del rostro del usuario para consultar en una base de datos MySQL y en un directorio de fotografías si el usuario puede acceder al área correspondiente del módulo de entrada. Cada módulo de entrada consta de una tarjeta Raspberry Pi 3 B+, un lector de tarjetas RFiD, una cámara de video y una pantalla de cristal líquido o LCD. El módulo central se compone de los mismos elementos que los módulos de entrada y cuenta con una pantalla táctil usada en la interfaz de usuario en lugar de una pantalla LCD. La comunicación entre los nodos es wifi, logrando una precisión del 99,2 % en la verificación del rostro y un tiempo de respuesta de 180 ms usando 310 fotografías entrenadas.

Palabras clave: cámara de video, MySQL, pantalla táctil, Raspberry Pi 3 B+, verificación de rostro, RFiD.

[1,*]Digital Systems Area, Department of Electronics, Universidad Autónoma Metropolitana-Azcapotzalco Cd. de México, Méxivo. Author for correspondence ✉: vlji@correo.azc.uam.mx.
https://orcid.org/0000-0002-4226-2936, https://orcid.org/0000-0002-4182-5856,
https://orcid.org/0000-0002-0581-7410, https://orcid.org/0000-0003-0455-007X.

Received: 14-05-2018, accepted after review: 21-06-2018

Suggested citation: Vega-Luna, J. I.; Sánchez-Rangel, F. J.; Salgado-Guzmán, G. and Lagos-Acosta, M. A. (2018). «Access System Using an RFiD Card and Face Verification». Ingenius. N.°20, (july-december). pp. 107-116. doi: https://doi.org/10.17163/ings.n20.2018.10.

1. Introduction

Data processing centers (DPC), also called data centers, are facilities that concentrate resources and equipment necessary for the processing and storage of information, as well as telecommunications equipment for companies and organizations. In data centers, different devices are used to access facilities, including electromagnetic locks, turnstiles, video cameras, motion detectors, identification cards, biometric systems and keyboards to enter a password, among others. Commonly, data centers are divided into sections called bunkers and are periodically audited in order to be certified. An important point that audits consider is the procedures and techniques used in security and access to facilities [1]. Currently there are different solutions for the identification of people to control the access to the bunkers of a data center. Some biometric solutions are based on the recognition of a person's fingerprints, face, hand geometry, iris, retina pattern, voice and signature [2].

This work presents the requirement of a data center operator company. The objective was to have an access system that uses a RFiD card and verification of the user's face to activate the actuator of the access door of the bunker the user is trying to access. Access must have two levels of security. The established requirements were to have a reliable system, easy to locate and use. The use of RFiD cards was required because they are cheap and easy to use. The maximum distance from the bunker farthest to the monitoring office is 65 meters, and 35 meters with line of sight to the wifi access point. The proposed solution consisted of a system composed of three input modules and a central module. The data center has three bunkers in whose entrance doors an input module was installed. The central module was installed in the data center monitoring office. The input modules are responsible for reading the information stored in the RFiD card, capturing the photograph of the user's face, and transmitting the information of the card and JPEG file with the photograph to the central module for validation, using Wi-Fi technology.

An Ethernet segment was not used to transmit the user identification information to the monitoring office so as not to install additional wiring or modify the existing one. Once the information is received, the central module checks the user database to see if the UUID of the RFiD card is authorized to enter the bunker associated with the input module, verifies that the user's face is the one registered in the photograph directory, and registers the date and time of entry request in the database. If both of the above conditions are met, the central module transmits the command to the input module to activate the corresponding door actuator. The input modules and the central module were implemented using a Raspberry Pi 3 B+ card with Raspbian operating system as the basis. The main reason for using the Raspberry Pi card was because there is a large number of applications and libraries developed by the open source community that are easy to install, configure and use in Raspbian [3]. In the system presented here, the use of an RFiD card was implemented as the first security mechanism and the NFC/RFiD 532 device was used to read cards. The technology of near field communication, NFC, arose from the combination of RFiD technology and smart cards. It allows the identification and characterization of people or objects without physical contact using radio waves transmitted by a label. RFiD technology allows the exchange of information between objects located close to each other. The communication with NFC is safer than other technologies since the transmitter and receiver are closely coupled, with a maximum proximity of 10 centimeters, without the need to run an application. In recent years, NFC technology has been used in several ways with mobile phones, on the Internet of Things or IoT and in the field of sensors [4].

Although it was initially decided to use RFiD cards, alternative technologies were explored for the identification of users, including technologies such as rapid response codes or QR and the iBeacon system. QR codes are an improvement to bar codes, they store information in dot matrixes or barcodes in a two-dimensional way [5]. When a mobile device reads a QR code it executes an application to perform a specific action. In the development of this work a combination of RFiD technology and QR codes could be used, but it would be a slightly more expensive and slower system, since in addition to using a method of printing the QR code on RFiD cards, these could not be reused. On the other hand, iBeacon is a protocol used in indoor positioning systems, or IPS, patented by Apple Inc. It is based on low-cost transmitters and low power consumption that indicate their presence to a device with iOS operating system and some devices with Android operating system [6]. There are transmitter providers, called beacons, compatible with iBeacon. Beacons use Bluetooth technology transmitters with low power consumption or BLE for short, or Bluetooth 4.0, which transmit their UUID to mobile electronic devices, allowing a mobile phone or tablet to perform an action or application based on the location of the beacon upon receiving identification, or following up with clients or users of beacons. The iBeacon system is used in mobile commerce, where an application, running on a mobile phone, can find the location of a product associated with a beacon inside a store or a beacon can send offers or promotions to the mobile phone. In other applications, beacons transmit information about nearby stores and restaurants to the mobile phone, as well as waiting times or distribution of points of interest messages according to the phone's location. The iBeacon technology differs from others,

such as NFC/RFiD, in that the transmission made by the beacon is one-way and requires that an application be run on iOS or Android. It could have been an option to use iBeacon in the development of this work, which would imply using a beacon as the user's identifier and an iOS device at each access point to the data center, which would increase the complexity of use, installation and cost of the system [7].

With the explosion of services based on the Internet, or Internet of Things, RFiD technology continues to be used in different developments and applications of identification, including supply chain [8], health care, object localization, home automation, security systems and product delivery in restaurants [9]. Work has been done on access systems to Arduino-based facilities, RFiD cards and MySQL databases. The difference with respect to the presented here is that a Raspberry card of more recent technology and lower cost than Arduino is used [10]. Additionally, the works that have been developed use Ethernet communication to the database and in this work Wi-Fi wireless technology was used, whose implementation is non-intrusive to the data center facilities [11]. Similarly, work has been carried out on access systems for homes, offices, and even vehicles, which use smartphones to emulate NFC cards and NFC PN532 readers [12] such as the one used in this work. In these systems the user must carry a smart phone to identify himself, which is not feasible nor is it an option in the data centers due to the cost and the fact that sometimes the users are visitors. Various works have also been carried out using QR codes or a combination of these with RFiD cards to control access to facilities, for location and navigation systems [13] and for product identification [14] and medical images. Access systems to data centers have even been created combining QR codes and watermarks [15]. The use of QR codes provides a higher level of security than RFiD cards, but the cost of implementation and operation of these systems is high, since once a card with a QR code is used it can not be used for another user and the hardware for printing and reading QR codes is more expensive than an NFC reader. Other works recently carried out for identification, location and access control integrate iBeacon and wifi [16] or Bluetooth LE technologies. These systems have the limitation of using devices with iOS or Android operating system, which makes them more expensive than the one developed in this work.

The verification of the person's face is used as the second security mechanism. Facial recognition began to be used in the 60s. It was a semi-automatic process in which an operator identified the features of the person in two or more photographs and calculated the distances to reference points to compare them with each other. The technological advances of computing in recent years have created an explosion of algorithms, techniques and non-intrusive applications of automated facial recognition that run on a computer to identify a person in a digital image. Taking the image of an unknown person, a profile with the same face in a set of known images must be found, also called training images. This is done with one of two purposes: 1) Verification or authentication of faces, comparing an image of a person's face with another image. The application confirms or denies the identity of the face, the objective is to ensure that the person is who they say they are; and 2) identification or recognition of faces, comparing the image of an unknown face with the images of known faces stored in a database to determine someone's identity. Facial recognition is an area that integrates the following technologies: image processing, computer vision, pattern recognition, neural networks and machine learning [17]. The procedure used by facial recognition systems generally consists of five phases:

- Registration phase, the image of the face of the person to be identified is captured using a camera or a video camera.
- Phase of the image processing, the face alignment is carried out based on some geometric properties and an independent image of the illumination and color range of the original image is obtained.
- Phase of extraction of biometric information, facial characteristics are obtained as a biometric pattern.
- Comparison phase, the biometric pattern compares the pattern of faces stored in the database. It is a 1:N comparison where the percentage of similarity of the person to be identified is determined with respect to the photographs stored in the database.
- Decision making phase, using a matrix of similarities, the person that was found with the highest percentage of similarity of the database is identified using an established range.

In recent times, the use of facial recognition systems has experienced a boom in different types of applications, used to authenticate the owners of mobile devices, in the detection of sleepy or tired drivers, in human trafficking, in risk analysis and in situations in places with a high concentration of people [18]. Microsoft applies facial recognition to access a Windows computer [19], while Apple is trying to have a mechanism in which iOS users can automatically share photos with tagged friends. Facebook and Google have engaged in a war on the design and use of facial recognition algorithms to tag friends and find photos of a person. They aim to achieve the perfect algorithm, recognizing faces much better than the human being. Google introduced, in 2015, the facial recognition

system called FaceNet, with an accuracy of 99.63%, recognizing photos on Google+ [20]. This system uses machine learning, generating a map in a compact Euclidean space from the image of a human face, where the distances correspond directly to the measure of similarity of the face. With this space, the tasks of verification and recognition of an image can be easily performed using standard techniques such as FaceNet vector embeddings. The FaceNet system uses a deep convolutional neuronal network trained with more than 260 million face images. The authors of FaceNet indicate that they have developed the state of the art of facial recognition methods using only 128 bytes for each face and more than 13,000 face images of the Internet to verify if two images are the same person, while the system of recognition YouTube Faces achieves 95.12%. The technology used by Facebook for facial recognition is called DeepFace, it was developed by the Israeli company face.com and released in 2013 [21]. The creators of DeepFace indicate that they can achieve an accuracy of 97.25% when comparing two faces.

In recent years, facial recognition has been used in access systems in data centers. Reliable systems with an acceptable percentage of accuracy can be achieved without using algorithms as sophisticated as those developed by companies such as Google and Facebook, which are proprietary and patented algorithms. There are many open source algorithms that can be used in the operating system of a small, low-cost and powerful computer like the Raspberry Pi 3 B + card. One of these algorithms is the histogram of oriented gradients or HOG, called the HOG algorithm [22]. This algorithm was developed in 2005, it is one of the most advanced and it is continuously improved to optimize it and achieve greater precision. A HOG is a feature descriptor used in computer vision and image processing for the detection of objects. This counts the occurrences of gradient orientation in defined parts of an image. The descriptors can be used as input data or features for a machine learning algorithm. There are open source libraries that implement the phases of a facial recognition system with the HOG algorithm and deep machine learning, which are easy to install and use, significantly reducing the program code [23]. One of these libraries is Face_Recognition and is the one used in this work to verify the face of users. This library uses a trained neural network and is based on dlib, the state of the art tool in face recognition built with deep learning. The authors of Face_Recognition indicate that its accuracy is 99.38% and provides several functions with which you can perform some actions such as finding faces in a photograph, determining the location of the reference points of a face, manipulating the facial features of a face, biometrically coding a face, comparing two coded faces, recognizing faces in real time video and recognizing faces located in a photograph using a directory of photographs of people getting the name of each person. In order to use the Face_Recognition library, the following tools must be installed in Raspian: Python library for picamera (python3-picamera), dlib v19.6 and OpenCV.

On the other hand, a great variety of access systems to data centers has been made through biometric devices. Some of these systems carry out facial recognition using a desktop computer to implement the recognition process [24] and wired communication between the computer and the video camera [25] or webcam. They are efficient, but their cost and size is greater than the one developed in this paper. Other systems of this type are based on reading the iris of the eye [26] using a reader installed in the access door or by means of the user's smartphone. These systems are more secure than those of RFiD cards, QR codes, fingerprint reading or 2D facial recognition, but the cost of the reader is much higher.

2. Materials and methods

The methodology used in the design of this system consisted of dividing it into two components: the input modules and the central module. Subsequently, the system was implemented by choosing the appropriate elements and the lowest cost according to the established requirements. The functional block diagram of the system is shown in Figure 1.

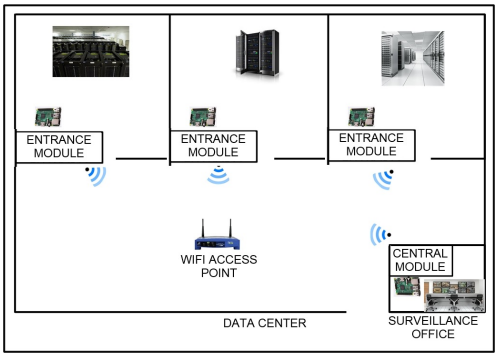

Figure 1. Functional block diagram of the access system

2.1. The input modules

Three input modules were built, all with the same architecture as the one shown in Figure 2. The main functions of these modules are the following: continuously explore if a card is found under the reach of the RFiD reader and read the UUID, capture the image of the face of the person trying to access, transmit the information read from the card and the photograph of the person in a JPEG file to the central module, and wait for the response from the central module to allow or deny access to the user. Each input module

consists of: a Raspberry Pi 3 B + card, an RFiD card reader, a video camera, a 2x16 LCD screen and an output interface.

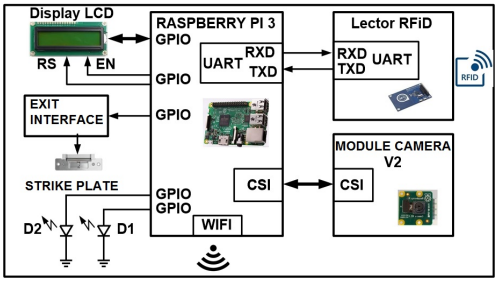

Figure 2. Block diagram of the input modules

The Raspberry Pi 3 B + card used in this module has the following hardware resources: 1 GB of RAM, 40 GPIO terminals, serial interface for camera, or CSI, DSI port for touch screen, Gigabit Ethernet port, SD memory slot and a Wi-Fi interface. The RFiD card reader used is the NFC/RFiD PN532 device. This reader is one of the most used in applications that use NFC technology, cards and RFiD tags of 13.56 MHz, since its main integrated circuit is embedded in many smartphones. It can write RFiD cards and labels type 1 to 4 and integrate an antenna whose range is 10 centimeters.

There is a large number of open source tools to make applications with the NFC/RFiD PN532. One of these tools is the libnfc library. In both the input modules and the central module, the RFiD reader was connected to the UART port of the Raspberry Pi and version 1.7.0 of the libnfc library was downloaded. Before installing and configuring libnfc, in the core of the Raspberry Pi operating system, the UART was disabled as a console port using the paspi-config tool and editing the /boot/config.txt file. Next, the libnfc library was installed and built using the following commands: sudo make clean and sudo make install all, which created the corresponding drivers, documentation files, binaries and executables. The input modules also contain a camera module for Raspberry V2 connected to the CSI interface of the Raspberry Pi 3 B+. This camera module has a high-resolution Sony IMX219 sensor of 8 megapixels. It allows the capture of photographs with a maximum resolution of 3238 x 2464 and high definition video.

There are open source libraries to use the camera and manipulate photos and video that can be invoked from Raspbian or from a program in Python. The camera can be controlled using the raspinstall command. However, in this work the Python python-picamera library was used in case that, in the future, it is necessary to modify the capture characteristics of photographs or video in the system. The camera of the input modules was enabled through Raspbian's raspi-config tool and later the python-picamera library was installed using the command: sudo apt-get install python3-picamera. Once the above was done, the camera.capture function ('file.jpg') could be used to capture an image in a JPEG file. The program that runs on the capture nodes was made in Phyton 3.6 and performs the following actions: configures timers, the UART port, Wi-Fi interface, GPIO terminals and peripheral devices, RFiD reader, video camera and LCD screen, display the message that tells the user to place the RFiD card in the reader on the LCD screen, and then enters a continuous cycle where the RFiD reader scans every 0.5 seconds by executing the nfc-pool_8c function.

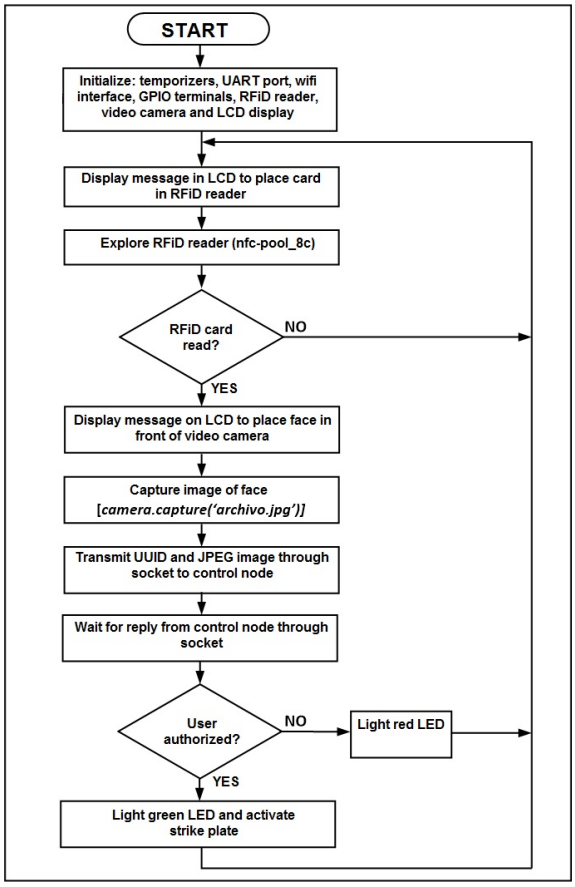

Figure 3. Flow diagram of the program of the input modules

The communication between the input and the control modules was carried out using message exchange with sockets under the client-server scheme. The input modules are the clients and the control module is the server. When the reader detects a card, it displays a message on the LCD screen asking the user to stand in front of the video camera and captures the image of the person's face in a JPEG file. Subsequently, the program transmits the UUID of the RFiD card and the JPEG file to the central module through a socket. Once the above is done, the program waits for the response of the central module in the socket. If the an-

swer indicates that the user is authorized to enter, the input module activates the actuator of the access door, through the interface connected to a GPIO terminal of the Raspberry card, and turns on a green LED (D1), connected to another GPIO terminal, for 3 seconds. If the user is not authorized, it lights a red LED (D2) intermittently for 5 seconds. Figure 3 shows the flow diagram of the program.

To be able to use sockets from Python, the corresponding library must be installed by executing the following command: sudo apt-get install socket. The output interface that controls the input gate actuator was connected to a GPIO terminal on the Raspberry card as shown in Figure 4.

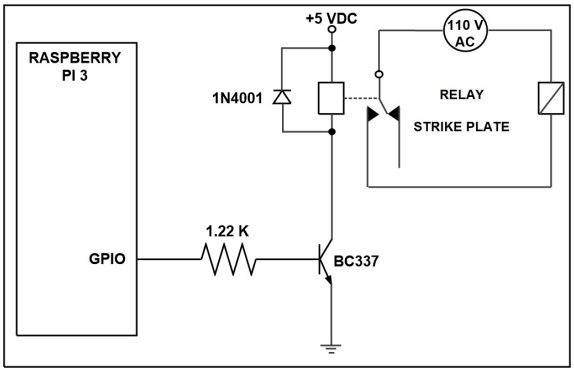

Figure 4. Entrance door actuator exit interface.

2.2. The central module

The central module consists of the following components: a Raspberry Pi 3 B+ card, an RFiD card reader, a video camera and a 3.5" Pi+TFT touch screen. Figure 5 shows the block diagram of the architecture of the central module.

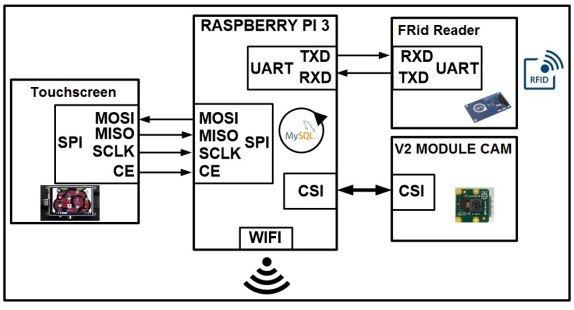

Figure 5. Block diagram of the central module.

The programming of the central module was carried out in Phyton 3.6 and it is divided into three parts: the main program, the communication routine with the input modules and the routine of the user interface. The main program configures timers, the UART port, the Wi-Fi interface and peripheral devices, RFiD reader, video camera and touch screen and invokes the two routines of the system, as indicated in the flow diagram in Figure 6.

In this module a database was created, managed with MySQL, which stores the information of authorized users to access the bunkers and a directory with the photographs of the face of previous users.

The communication routine with the input modules executes a program in the background that performs the following functions: 1) Create a socketthrough which it receives from the input modules the UUID and the JPEG file. 2) Access the MySQL database to determine if the user is authorized to enter the corresponding area. 3) Invokes the routine that verifies that the user's face is in the photo directory. 4) Update the user record in the MySQL database with date and time of entry. 5) Transmit the message to the input module to activate the door actuator or deny entry. 6) Update the log of access attempts by storing the JPEG file.

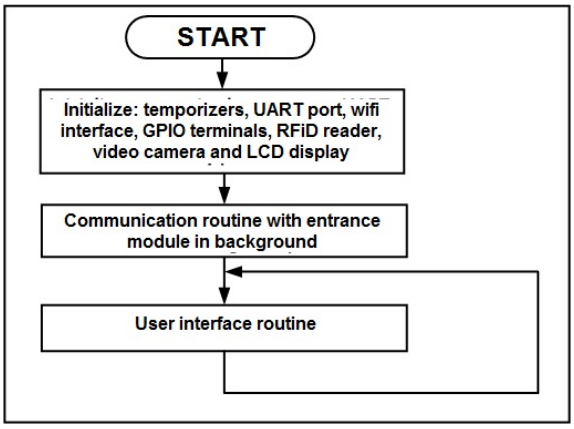

Figure 6. Flow chart of the main program of the central module.

Figure 7 shows the flow diagram of this routine. Both the database and the directory of coded and trained photographs reside in the Raspberry Pi 16 GB SD card. A table containing user records was created in the database. Each record stores the UUID of the assigned RFiD card, number of doors to which it has access, name, company and user email. To create the database and user table the following tasks were carried out:

1.- Installation of the MySQL server and client, as well as the Python API to access MySQL.

2.- Creation of the database executing the following commands: mysql -u root -p, mysql> CREATE DATABASE RFID_DB; CREATE TABLE users_tbl (id INT NOT NULL PRIMARY KEY AUTO_INCREMENT, UUID VARCHAR(20), doors VARCHAR(20), name VARCHAR(20), last name VARCHAR(30), company VARCHAR(20), email VARCHAR(30)).

Once the database was created, the program in Python was made to access it. Python uses an object or data structure, called a cursor, to access the data in the table. This object allows operations to create, read, update and remove records in the database. The program executes the following general actions:

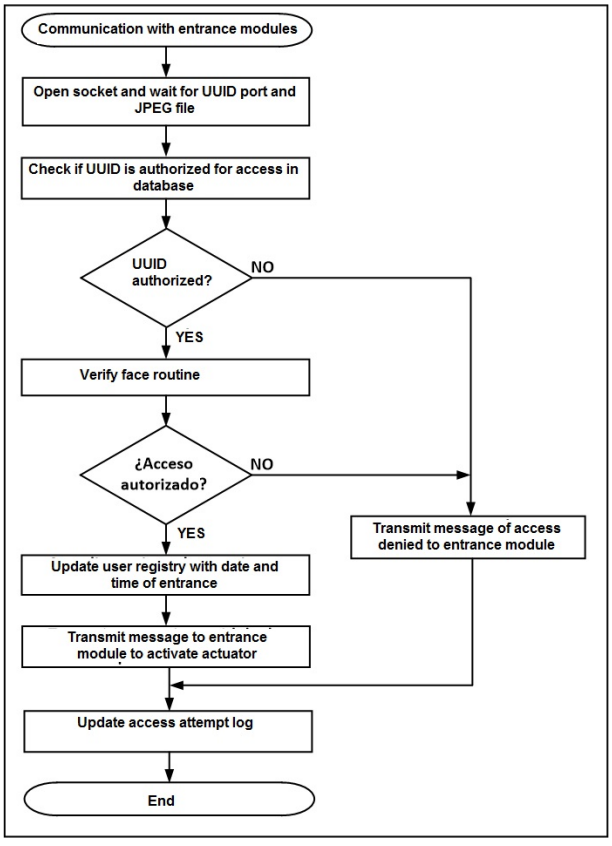

Figure 7. Flow diagram of communications with the input modules

1.- Import the Python API for MySQL: import MySQLdb.

2.- Make the connection to the database: db = MySQLdb.connect ("localhost", "root", "password", "RFID_DB").

3.- Define the cursor object: cursor = db.cursor ().

4.- Wait for the option selected by the user in the graphic interface.

5.- Depending on the option, define one of the following SQL query's: cursor.execute ("INSERT INTO users-tbl"), cursor.execute ("SELECT * FROM users_tbl"), cursor.execute ("UPDATE users_tbl SET") or cursor.execute ("DELETE FROM users_tbl WHERE")

6.- Execute the query: db.commit ().

In the photo directory, the name of each file corresponds to the name of the user registered in the MySQL database. The routine that checks if the user's face is in the photo directory performs the following actions: loads the image of the face received from an input module in a buffer using the face_recognition.load_image_file function, encodes and learns to recognize the stored image in the buffer using the function face_recognition.face_encodings and enters a cycle where it compares the encoded image of the buffer with each image of the directory of coded photographs. The cycle ends when it finds equivalence between the two images analyzed or when it explored the entire directory without finding equivalence. The comparison is made through the function face_recognition.compare_faces, which obtains, if successful, the name of the user of the photograph. If the name obtained is equal to the name read from the user's record in the database, it returns to the routine that invoked it authorizing access to the user, as shown in the flow diagram in Figure 8. It was considered that the image received from the input module contains only one face, otherwise the face_recognition.face_locations function would have to be used to find the faces in the image and code them individually.

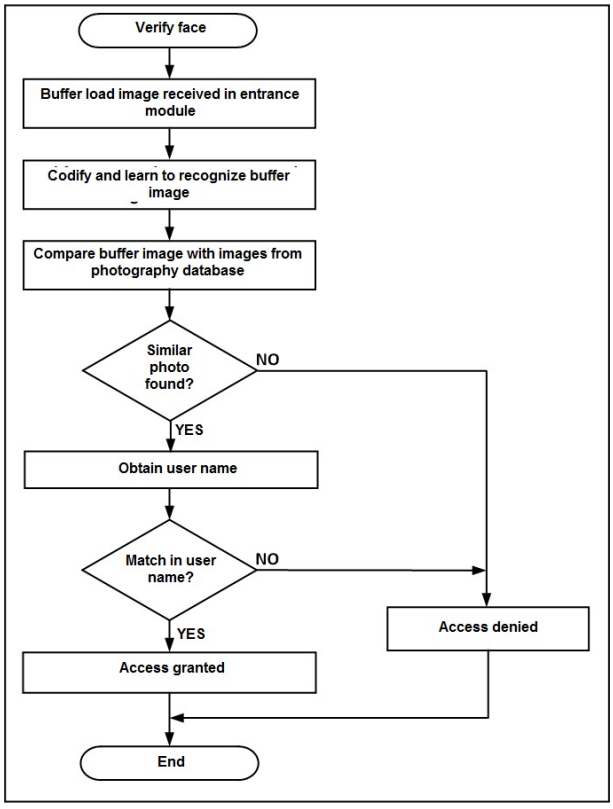

Figure 8. Flow diagram of the face verification routine

The routine that implements the graphical user interface, allows for access and management of the

database using the touch screen. The screen used in the central module is the 3.5" Pi+TFT device which has a resolution of 480 x 320 and was connected to the SPI port of the Raspberry Pi card. In the user interface, the administrator can perform the following operations: uploads, deletions and changes of users, as well as showing the registered users and the log of access attempts. The RFiD card reader and the validation module camera are used when registering or making changes to a user's registry. The user interface was made using pygame. The pygame tool is a set of libraries that can be used in a Python program for the implementation of videogames, multimedia programs and graphical user interfaces, since it allows to display text, images and sounds on a touch screen and control the position of the cursor. This tool is installed by default with the Raspbian version for Raspberry Pi. The IP address of the Wi-Fi interface of each input module is fixed and is used by the central module to determine the door number which the user is trying to access.

3. Results and discussion

Four groups of tests were carried out. The first group aimed to measure the RFiD reader's reach for the input modules. By placing 50 cards in the module reader, it was determined that the range is 14 centimeters, a little more than indicated in the manufacturer's specifications. The second group of tests aimed to store the photographs of 50 users in the central module's directory and train the neural network. The average size of each photograph was 110 KB. The third group of tests aimed to determine the accuracy of the face verification system of users registered in the database. This group of tests was carried out in several phases. In the first phase, the directory of trained photographs stored 50 faces. In each subsequent phase, 20 photographs were added, leading to a total of 310. In each phase, 40 different faces were verified. With some faces, the recognition was not successful despite it being registered in the central module. The number of unsuccessful matches led, as a consequence, to an accuracy of 96.3% in the first phase, which increased as the number of photographs trained increased until reaching 99.2% as shown in the graph in Figure 9.

The fourth group of tests aimed to measure the response time of the system. To carry out these tests in each of the phases of the previous test group, the face capture time of a person registered in the database and the time until the central module received a response were both recorded in a file in the input module once the authorized person was verified. The response time in the first phase was 132 ms on average. It increased to 180 ms in the last phase, an almost imperceptible change for the user, as indicated in the graph in Figure 10. The photographs of the directory of the central module were taken with enough ambient light, from the front, without glasses, poses or objects preventing a clear view of the face. It is recommended that when registering new users, several photographs of the face are captured using different expressions, allowing the system to be more tolerant and in order to improve both accuracy and response time. The implementation of this work did not require installing additional wiring for data transmission or modifying the existing one. The central module is installed in a data center control office, this makes it more practical than the commercially available alternatives that use wired communication. The cost of the system is $350.00 USD, lower than existing commercial alternatives, which cost $1700.00 USD on average.

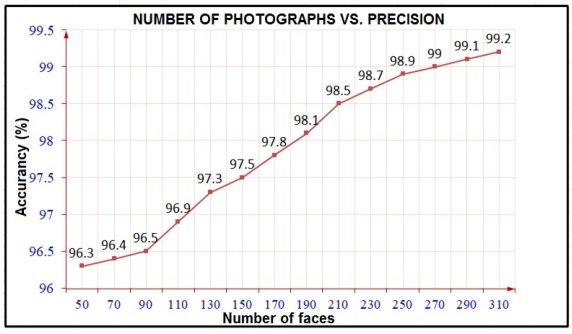

Figure 9. Accuracy of the system with different amounts of faces included in training

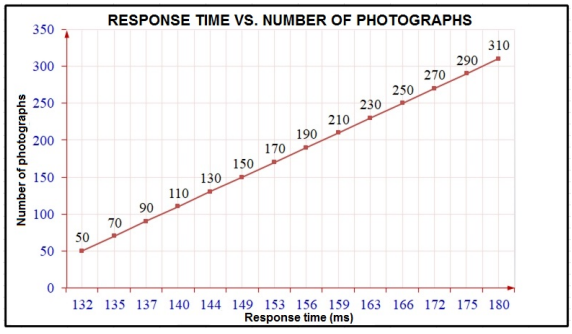

Figure 10. System response time

4. Conclusions

The result of this work was an access system with a double safety mechanism which is more robust than those commercially available that use only one mechanism. It was built using the latest technology and low-cost components, open source software and communication via WiFi, which does not impact the data center facilities, resulting in a practical application that meets the established requirements. The reading range of RFiD cards achieved was 14 centimeters. In

face verification, 99.2% accuracy and 180 ms response time were achieved using 310 trained photographs.

Future works

With the percentage of accuracy and response time achieved, the data center requested a second version that incorporates the following features: 1) Incorporate a web server to the central module and a touch screen in the input modules so that the administrator can access the user database and photo directory from any input module and 2) Incorporate a fingerprint reader in all modules to have an additional level of security. These functionalities are feasible to perform with the current architecture of the system modules.

Acknowledgements

We appreciate the support provided by the Electronics Department at Universidad Autónoma Metropolitana-Azcapotzalco.

References

[1] M. V. M. Lima, R. M. F. Lima, and F. A. A. Lins, "A multi-perspective methodology for evaluating the security maturity of data centers," in *2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, Oct 2017. DOI: https://doi.org/10.1109/SMC.2017.8122775, pp. 1196–1201.

[2] M. Levy and J. O. Hallstrom, "A new approach to data center infrastructure monitoring and management (dcimm)," in *2017 IEEE 7th Annual Computing and Communication Workshop and Conference (CCWC)*, Jan 2017. textscdoi: https://doi.org/10.1109/CCWC.2017.7868412, pp. 1–6.

[3] I. B. Mustaffa and S. F. B. M. Khairul, "Identification of fruit size and maturity through fruit images using opencv-python and rasberry pi," in *2017 International Conference on Robotics, Automation and Sciences (ICORAS)*, Nov 2017. DOI: https://doi.org/10.1109/ICORAS.2017.8308068, pp. 1–3.

[4] J. Mihal'ov and M. Hulič, "Nfc/rfid technology using raspberry pi as platform used in smart home project," in *2017 IEEE 14th International Scientific Conference on Informatics*, Nov 2017. DOI: https://doi.org/10.1109/INFORMATICS.2017.8327257, pp. 259–264.

[5] N. Goel, A. Sharma, and S. Goswami, "A way to secure a qr code: Sqr," in *2017 International Conference on Computing, Communication and Automation (ICCCA)*, May 2017. DOI: https://doi.org/10.1109/CCAA.2017.8229850, pp. 494–497.

[6] S. Menon, A. George, N. Mathew, V. Vivek, and J. John, "Smart workplace – using ibeacon," in *2017 International Conference on Networks Advances in Computational Technologies (NetACT)*, July 2017. DOI: https://doi.org/10.1109/NETACT.2017.8076803, pp. 396–400.

[7] X. Li, D. Xu, X. Wang, and R. Muhammad, "Design and implementation of indoor positioning system based on ibeacon," in *2016 International Conference on Audio, Language and Image Processing (ICALIP)*, July 2016. DOI: https://doi.org/10.1109/ICALIP.2016.7846648, pp. 126–130.

[8] M. Chamekh, S. E. Asmi, M. Hamdi, and T. H. Kim, "Context aware middleware for rfid based pharmaceutical supply chain," in *2017 13th International Wireless Communications and Mobile Computing Conference (IWCMC)*, June 2017. DOI: https://doi.org/10.1109/IWCMC.2017.7986576, pp. 1915–1920.

[9] K. B. Eric and W. H. Ya, "Iot based smart restaurant system using rfid," in *4th International Conference on Smart and Sustainable City (ICSSC 2017)*, June 2017. DOI: https://doi.org/10.1049/cp.2017.0123, pp. 1–6.

[10] M. Andriansyah, M. Subali, I. Purwanto, S. A. Irianto, and R. A. Pramono, "e-ktp as the basis of home security system using arduino uno," in *2017 4th International Conference on Computer Applications and Information Processing Technology (CAIPT)*, Aug 2017. DOI: https://doi.org/10.1109/CAIPT.2017.8320693, pp. 1–5.

[11] S. Nath, P. Banerjee, R. N. Biswas, S. K. Mitra, and M. K. Naskar, "Arduino based door unlocking system with real time control," in *2016 2nd International Conference on Contemporary Computing and Informatics (IC3I)*, Dec 2016. DOI: https://doi.org/10.1109/IC3I.2016.7917989, pp. 358–362.

[12] J. Cui, D. She, J. Ma, Q. Wu, and J. Liu, "A new logistics distribution scheme based on nfc," in *2015 International Conference on Network and Information Systems for Computers*, Jan 2015. DOI: https://doi.org/10.1109/ICNISC.2015.48, pp. 492–495.

[13] W. Xiao-Long, W. Chun-Fu, L. Guo-Dong, and C. Qing-Xie, "A robot navigation method based on rfid and qr code in the warehouse," in *2017 Chinese Automation Congress (CAC)*, Oct 2017. DOI: https://doi.org/10.1109/CAC.2017.8244199, pp. 7837–7840.

[14] H. Keni, M. Earle, and M. Min, "Product authentication using hash chains and printed qr codes," in *2017 14th IEEE Annual Consumer Communications Networking Conference (CCNC)*, Jan 2017. DOI: https://doi.org/10.1109/CCNC.2017.7983126, pp. 319–324.

[15] P. Pramkeaw, T. Ganokratanaa, and S. Phatchuay, "Integration of watermarking and qr code for authentication of data center," in *2016 12th International Conference on Signal-Image Technology Internet-Based Systems (SITIS)*, Nov 2016. DOI: https://doi.org/10.1109/SITIS.2016.111, pp. 669–672.

[16] H. Zou, Z. Chen, H. Jiang, L. Xie, and C. Spanos, "Accurate indoor localization and tracking using mobile phone inertial sensors, wifi and ibeacon," in *2017 IEEE International Symposium on Inertial Sensors and Systems (INERTIAL)*, March 2017. DOI: https://doi.org/10.1109/ISISS.2017.7935650, pp. 1–4.

[17] Z. Yu, F. Liu, R. Liao, Y. Wang, H. Feng, and X. Zhu, "Improvement of face recognition algorithm based on neural network," in *2018 10th International Conference on Measuring Technology and Mechatronics Automation (ICMTMA)*, Feb 2018. DOI: https://doi.org/10.1109/ICMTMA.2018.00062, pp. 229–234.

[18] N. Mokoena, H. D. Tsague, and A. Helberg, "2d methods for pose invariant face recognition," in *2016 International Conference on Computational Science and Computational Intelligence (CSCI)*, Dec 2016. DOI: https://doi.org/10.1109/CSCI.2016.0163, pp. 841–846.

[19] D. Goldman. (2015) Microsoft will let you unlock windows 10 with your face. CNN tech. [Online]. Available: https://goo.gl/tgo8pM

[20] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015. DOI: https://doi.org/10.1109/CVPR.2015.7298682, pp. 815–823.

[21] S. Srisuk and S. Ongkittikul, "Robust face recognition based on weighted deepface," in *2017 International Electrical Engineering Congress (iEECON)*, March 2017. DOI: https://doi.org/10.1109/IEECON.2017.8075885, pp. 1–4.

[22] M. Wiglasz and L. Sekanina, "Evolutionary approximation of gradient orientation module in hog-based human detection system," in *2017 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, Nov 2017. DOI: https://doi.org/10.1109/GlobalSIP.2017.8309171, pp. 1300–1304.

[23] J. Zeng, X. Zhao, C. Qin, and Z. Lin, "Single sample per person face recognition based on deep convolutional neural network," in *2017 3rd IEEE International Conference on Computer and Communications (ICCC)*, Dec 2017. DOI: https://doi.org/10.1109/CompComm.2017.8322819, pp. 1647–1651.

[24] X. Chen, L. Qing, X. He, J. Su, and Y. Peng, "From eyes to face synthesis: a new approach for human-centered smart surveillance," *IEEE Access*, vol. 6, pp. 14 567–14 575, 2018. DOI: https://doi.org/10.1109/ACCESS.2018.2803787.

[25] A. H. M. Amin, N. M. Ahmad, and A. M. M. Ali, "Decentralized face recognition scheme for distributed video surveillance in iot-cloud infrastructure," in *2016 IEEE Region 10 Symposium (TENSYMP)*, May 2016. DOI: https://doi.org/10.1109/TENCONSpring.2016.7519389, pp. 119–124.

[26] Ş. Karahan and Y. S. Akgül, "Eye detection by using deep learning," in *2016 24th Signal Processing and Communication Application Conference (SIU)*, May 2016. DOI: https://doi.org/10.1109/SIU.2016.7496197, pp. 2145–2148.