International Journal of Applied Engineering Research ISSN 0973-4562 Volume 13, Number 6 (2018) pp. 4195-4202

Design of an Elevator Monitoring Application using Internet of Things

Alejandro Duarte Suárez[1], Octavio José Salcedo Parra[1, 2], Jhon Hernán Díaz Forero [1]

[1]*Faculty of Engineering - Universidad Distrital Francisco José de Caldas, Bogotá D.C., Colombia.*

[2] *Faculty of Engineering - Universidad Nacional de Colombia, Bogotá D.C., Colombia.*

Abstract

This article will be designed and an application that allows to inform the technicians or the maintenance company of elevators on the failures that they can present in their operation, thus avoiding that people are trapped in the elevators or have periods of outside of service very extensive. An application is implemented in the cloud that will be responsible for receiving the data from the elevator communications module, processing them and if necessary sending a notification to the technicians in elevators. It describes the current technologies and especially Microsoft Azure, which was used for this development.

Keywords: Cloud Computing, Elevator, Internet of Things, Motorization.

INTRODUCTIÓN

The elevator is a very used means of transport, especially in buildings with many floors, however these are not immune to failures and often people are trapped in them or simply do not climb. These problems are due to wear and poor condition of elevator components, although scheduled maintenance services are usually hired, but how do you prevent elevator failures if lifts suddenly wear out more?

Today the internet and in general networks are present in all situations, it is already daily that almost all people have and use smartphones, this keeps them connected. Recently the Internet of Things (IoT) has been spoken, which seeks to connect all the objects to the network, so that they can be monitored, used, prepared remotely and even inform the users About the state of something in particular. Using these approaches we can think of connecting the elevator to the network using the IoT, this in order to monitor the state and prevent failures in the elevators. The information obtained from the elevator must be processed by some application, this application is the one that will be in charge of informing the maintenance company about the problems in the elevator so as to reduce the events of people locked up or the time that an elevator is outside of service.

BACKGROUND

Xibo Wang and Hongshuai Ge and Wenbo Zhang and Yingzhen Li in the year 2015, carried out an investigation that titled Design of Elevator Running Parameters Remote Monitoring System Based on Internet of Things for a Chinese company, in which they design a system of monitoring of the elevators , First the communication component is installed in the electronic circuit of the elevator and connects to the network using GPRS technology, because the network does not support as much traffic the transport protocol used is UDP, reliable delivery and error correction Assign to the application layer, it is also proposed a thread method that handles multiple queues because not only will it be an elevator that sends information, but it will be a group of several. As a contribution to the current project is the general architecture proposal, which consists of three parts namely: The first is the data acquisition module, the second is the network module and the third is the remote monitoring center; The remote monitoring center manages a view-controller model and gives a description of the operation of this module which serves as reference for the present project.

Tundong Liu, Xiaosheng Liao and Jianping Zeng in 2010, made a design that entitled: Design of Intelligent Elevator Remote Monitoring System Based on Ethernet, in the article describe the functionality of a sensor connected to the elevator, consists of three circuits and one of They provide the connection to the network via Ethernet, then design the device driver, then develop the socket and database to use. Finally they make a simulation with a graphical interface to show how it would operate in monitoring device and how, through the interface, to control the elevator.

Hamza Ijaz Abbasi, Abdul Jabbar Siddiqui in 2011, implemented a Smart Elevator System based on Wireless Multi-hop AdHoc Sensor Networks, a system is proposed for intelligent elevator delivery in order to optimize time and energy in the Normal operation of the elevator, gives guidelines to simulate by means of an application in Java the operation of the elevator and the operation of the system, also shows us the information that should have information packets and gives a starting point for the choice of transport protocol.

Andrés Enrique, Rosso Mateus and José Jairo, Soriano Méndez in 2009 made the development of a building simulator with variable parameters, using java and several free libraries in such a way that data can be obtained to simulate, A posteriori idea is to implement it in groups of elevators for buildings in such a way that efficient management of time and energy is done; Gives a framework for the development of simulations and approaches the implementation of a simulated system so that it can be taken as the starting point in the simulation of the elevator.

Dr.Shaik Abdul Nabi, Dayakar Gurra and Mohd.Anwar Ali, in 2015 propose an educational system using cloud hybrid coimputing so that from the mobile phones can be accessed, they propose to provide the three types of service that are: software As service, infrastructure as service and platform as

service. Although the current project does not seek to make an educational platform, however if it provides information on how to structure the application in the cloud.

Yashpalsinh Jadeja and Kirit Modi, in 2012, at a conference showed Cloud Computing - Concepts, Architecture and Challenges. It gives important information to make the jump to the cloud, first showing the background for cloud computing, and then show us the characteristics, so that you can start to see the benefits of this technology. Cloud architecture is important in deciding which option we need to use, for example, whether we are going to use "Software as a Service" or perhaps a "platform as a service" or perhaps an "infrastructure as a service"; This choice is important, because first we can see what we need and second who will be the service provider. Another important concept is the kind of service, whether it is public cloud, private or hybrid, each has its advantages and disadvantages and it is necessary to make the decision. The advantages of cloud computing include: Easy administration, cost reduction, uninterrupted services, disaster management and green computing, all of these advantages are supported by the fact that the organization does not have physical equipment and servers, if Not that they are in the cloud and the provider is who has to worry about the administration and maintenance of them. Lastly, security and privacy issues in cloud computing are discussed and a possible way to solve them.

Liu Wenato in 2012, conducted Research on Cloud Computing Security Problem and Strategy. Explain that it is cloud computing, as it is composed, what types of service we can find in it, which providers exist. The most important part and what is mainly focused on the problems of cloud computing: it is exposed to all the problems of the internet, such as viruses and hacking. The main problem is that the user has no control over these aspects and can not prepare against them, who must take charge is the provider. The other problem is privacy, as secure and private data go through the network can be captured and altered and on the other hand can directly attack the server and steal or alter information. As a security strategy it is proposed to use the encryption tools in the data before they are stored in the cloud storage; Service providers must ensure and establish mechanisms to verify the privacy of information and as a last strategy is proposed to use the SOA architecture to use the security mechanisms that can be used in web services.

Yanuarizki Amanatullah, Charles Lim, Heru Purnomo Ipung and Arkav Juliandri, worked in Toward Cloud Computing Reference Architecture: Cloud Service Management Perspective. In the world of cloud computing there are several companies that have made an effort to propose an architecture for cloud computing, among them are IBM with CCRA, NIST and CSA (Cloud security alliance) with TCI (Trusted cloud initiative). It proposes an architecture that contemplates six types of actors, which are: Cloud customer, who uses and contracts the services; Cloud provider, who provides services; Cloud developer, is responsible for developing in the cloud; Cloud broker, who serves as a negotiator among the other actors; Cloud auditor, who evaluates, ensures and examines the security, privacy and performance of services; And the Cloud Carrrier, transportation and physical connectivity between actors and target services. There are guidelines for cloud service administration, generally what the Cloud Provider should do, to ensure the reliability, performance and use of the services.

Samah Ahmed Zaki Hassan in 2012, proposed: STAR: A Proposed Architecture for Cloud Computing Applications. This paper reviews existing architectures for cloud development such as IBM, NIST, and EAC. The author proposes an architecture called STAR (Stage architecture), which consists of 8 layers, which he calls 4S4A, in this new architecture in the form of stack, group the cloud functionalities and gives a way to choose how to develop a Solution in the cloud. Subsequently an explanation of the new concepts presented in this architecture is made.

METHODOLOGY

A back-end application was implemented in the cloud so that this is the means of communication and processing between the sensor to be installed in the elevator and the application on the mobile phone. To this end, we explored the cloud computing tools that allow this connection and deployment.

With respect to the platforms it is convenient to name three of them that can be used for this development which are Microsoft Azure, Google App Engine and Amazon web services. All of them have back-end support for mobile applications, although all have a cost that varies depending on the plan that is acquired, however the collection is always done on demand, ie pay for what you use. All the reviewed platforms offer a free subscription so that the platform can be explored and subsequently purchased a plan, the amount of time and services available vary from one to another, from one month to one year, and from the basics to machine-learning . The platform chosen for development was Microsoft Azure.

On the other hand, a front-end application was implemented in the mobile device, this application receives from the cloud the information so that it can notify the user and in this way can carry out the maintenance and revisions required by the elevator by the technician.

MICROSOFT AZURE

Microsoft Azure is Microsoft's proposal for development in the cloud, they offer PaaS, and IaaS, can also be developed in the hybrid cloud.

Microsoft offers the ability to take business development to the cloud and pay only for what it consumes, among the benefits are the scalability of the application and not invest in equipment and / or maintenance of these, since this runs through Account of the platform. By deploying the application in the cloud you can choose the physical location of the services so that it can be deployed in the location closest to the users of the application.

Microsoft Azure offers a wide suite of services to deploy, including: Virtual Machines and Servers, Web and Mobile Development, Databases, Data Analysis, IoT Support, Networks, Media, Hybrid Cloud, Sessions, Artificial Intelligence and Containers.

To start using Microsoft Azure you need a subscription, some of them are: Free account (A free month with all services to explore), DreamSpark (Educational), Prepaid plans (6, 12 months), developer, and Type of technical support. In this development was used the license DreamSpark, which is provided by the agreement between Microsoft and the District University. The DreamSpark license has the services of Web and Mobile Development, AzureSQL and MySQL Databases and team projects.

DEVELOPMENT WITH AZURE

To start the development with Microsoft Azure creates a Microsoft account and in this case is linked to DreamSpark. Then go to https://portal.azure.com/, log in and show us the Azure main screen (See Fig. 1).

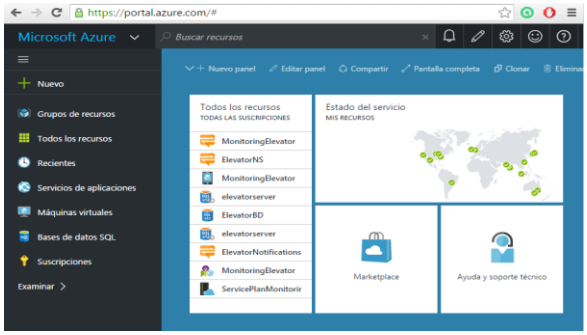

Figure 1. Main screen of Azure portal.
Source: Authors

Microsoft Azure provides all platform information and tutorials available at: https://azure.microsoft.com/en-us/. From that portal you can choose the services and platforms and follow the step by step for creating solutions.

The first time a development is made, it is necessary to create a resource group for the application. This resource group includes the location of the server, the type of subscription to be used, the database associated with the project and, depending on the Development, backend or solution resources. In this case it was chosen in the tab "New" and then MobileApp. The mobile application resource group is shown in the Figure 2.

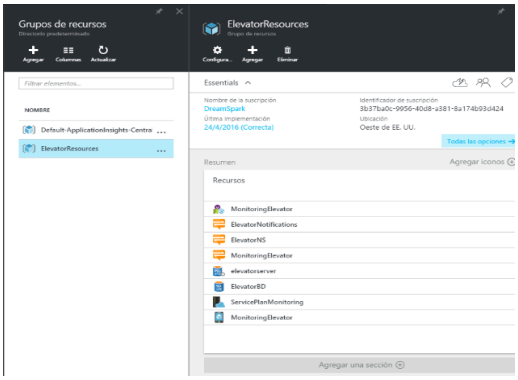

Figure 2. Mobile application resource group.
Source: Authors

In the application tab, we can see the state of execution and use of resources, and also offers the quick start tab in which we can choose the development platform (iOS, Android, Windows, Xamarin and Cordova). In the next section asks for the creation or selection of a database, also the choice of type of backend that can be Node.js and C # (.NET) and finally asks for the configuration of the client application that can be new or existing (See Figure 3).

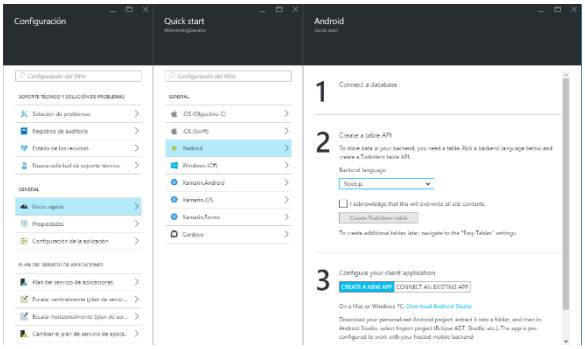

Figure 3.View of creating a mobileApp with Azure.

Source: Authors

In the quick start it is requested to download the backend, compile it and publish it with VisualStudio (See Figure 4).

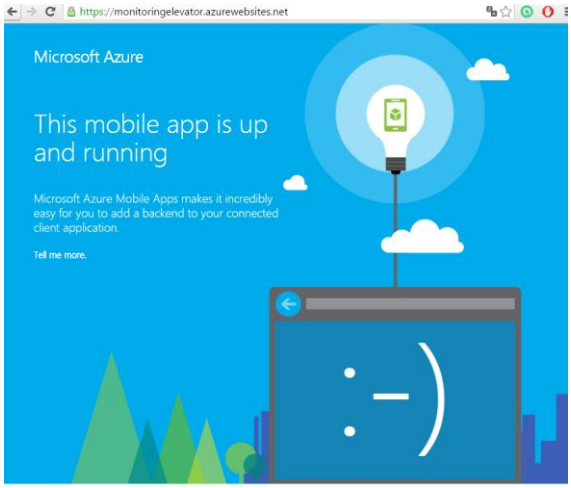

Figure 4. Backend deployed.

Source: Authors

The client application download is edited with Android Studio and customized as needed.

SYSTEM FLOW DIAGRAM

The flow diagram (See Figure 5) is based on what was presented in [1]. First, the cloud application initiates the database, which contains all the information concerning the elevators, and is waiting to receive from the sensor installed in the elevators information about detected faults; If the sensor

detects a fault, it sends the type of fault and the id of the elevator. Upon receiving such information, the information of the elevator is consulted in the database from the id received from the sensor, and a notification is built that reaches the application installed on the company's mobiles or maintenance technicians. To indicate to the technician more appropriate information for them, such as: the address of the building where the elevator is installed, date of last maintenance, brand and type of elevator, etc. After the fault has been solved, the information is updated in the database and the process is finished.

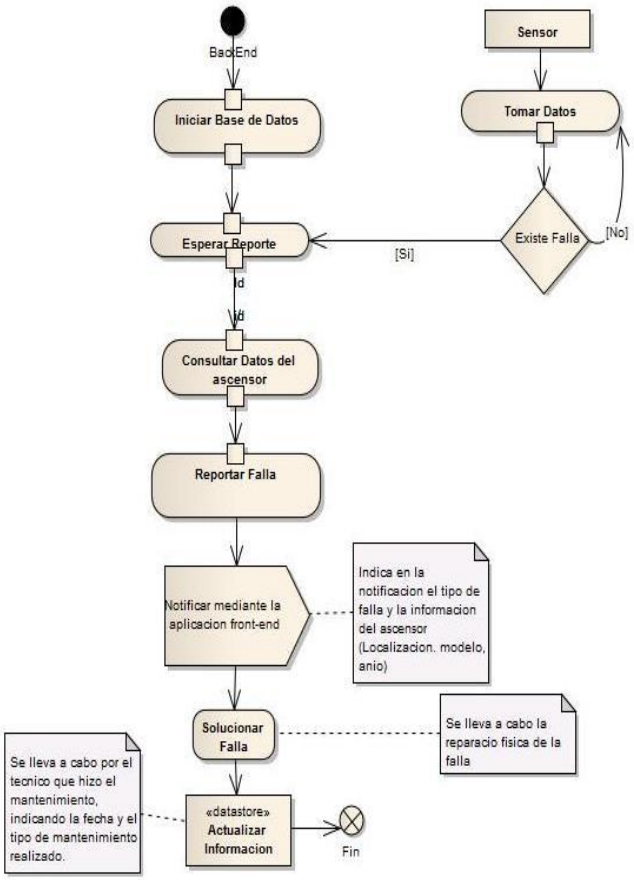

Figure 5. Flow diagram of the system operation.

Source: Authors

SYSTEM OPERATING ARCHITECTURE

Microsoft Azure, it should be noted that Microsoft Azure provides support for IoT applications.

For the operation of the system the sensor installed in the elevator must have a communication module, either via Ethernet, wi-fi, among others, and to send information to

When a fault is detected the sensor adds a new record within the table of reported problems and faults, this table is deployed and available in the backend of the application. Using the Microsoft Azure portal, the Notification Hub is created, which will provide support for sending notifications to the user (See Figure 6).

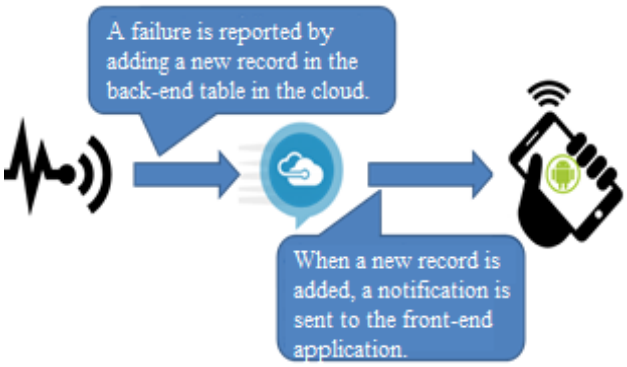

Figure 6. Aggregated records

Source: Authors

Thus, each time a new record is inserted the back-end in the cloud sends a notification to the smart-phone of the technician and / or maintenance company (See Figure 7).

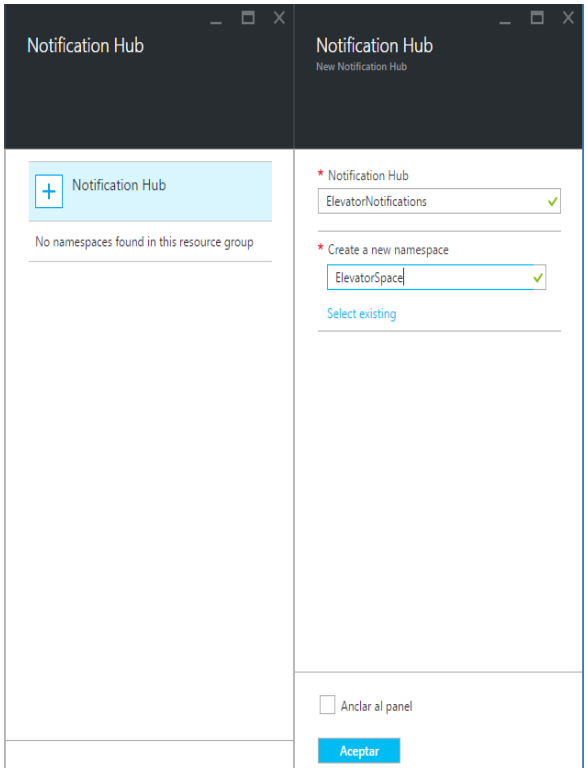

Figure 6. Creating the Notification Hub from the Azure portal.

Source: Authors

SYSTEM DRAWING SCHEME

The Figure 7 shows the data schema that was used for the application, the application requires a data store and Azure provides such support, for the creation of the table in the cloud, it is defined in the mobile application and then referenced in the backend (Algorithm 1).

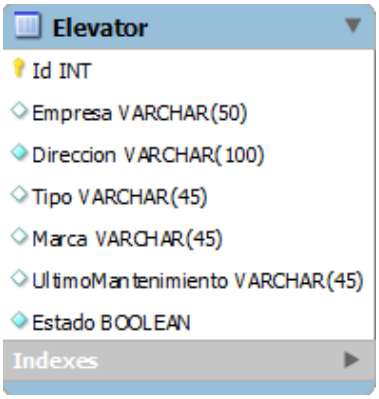

Figure 7. Data schema of the Elevator table.

Source: Authors

```
public class ElevatorItem {
    //Fields in the DB
    private String id;
    private String companyN;
    private String address;
    private String type;
    private String mark;
    private String dateLastMaintenance;
    private boolean state;
```

Algorithm 1. Definition of the table in the application.

Source: Authors

The Figure 7. is an alternative schema that extends and enhances searches and data handling by deleting repeating data in the table.

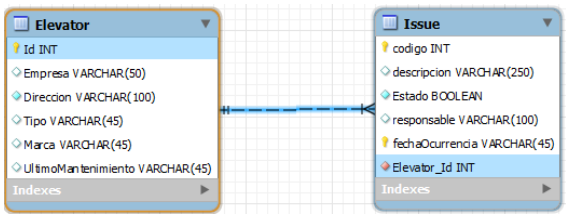

Figure 7. Second scheme of data proposed.

Source: Authors

With this scheme we facilitate the search of problems in an elevator and we eliminate the replication of information such as the name of the company, address, etc., and a field is added to indicate who was the person who performed the maintenance. This is the schema that the application has at this moment.

For the implementation of the data schema it is also necessary to define it in the back-end (See Algorithm 2).

```
using Microsoft.Azure.Mobile.Server;

namespace MonitoringElevatorService.DataObjects
{
    public class ElevatorItem: EntityData
    {
        public string IDElevator { get; set; }
        public string Company { get; set; }
        public string Address { get; set; }
        public string TypeElevator { get; set; }
        public string MarkElevator { get; set; }
        public string DateLastMaintenance { get; set; }
    }
}
```

Algorithm 2. Defining the back end data schema in VisualStudio. Source: Authors

In the previous image it is shown how the definition of a table is made that will be deployed in the back-end, in addition to that it is necessary to create a controller for the table, VisualStudio gives the facility to create it automatically, the controller serves to give the support To actions on the database, such as insertion, update, deletion and consultation.

FUNCTIONING OF THE APPLICATION

In the client application should have the instructions that link the application of the mobile with the back-end (Algorithm 3).

```
private MobileServiceClient mClient;
private MobileServiceTable<ElevatorItem> mElevatorTable;
```

Algorithm 3. Declaration of connection variables. Source: Authors

In addition, they must be instantiated in the method of creating the application using the Algorithm 4.

```
mClient = new MobileServiceClient(
        "https://monitoringelevator.azurewebsites.net",
        this);
    mElevatorTable = mClient.getTable("ElevatorItem", ElevatorItem.class);
```

Algorithm 4. Instantiation of the connection variables. Source: Authors

The main screen of the application shown below, allows to see the id of the elevators and the companies in which they are installed, also allows to go to the interface that provides the mechanism to add new elevators to the database (See Figure 8).

Figure 8. Application main screen. Source: Authors

When selecting any elevator, the application will take the user to another screen that will display the information of the elevator and also show a list of problems that has and had the elevator (See Figure 9).

Figure 9. Lift information. Source: Authors

Finally you have the screen that allows the insertion of a new lift the database (See Figure 10).

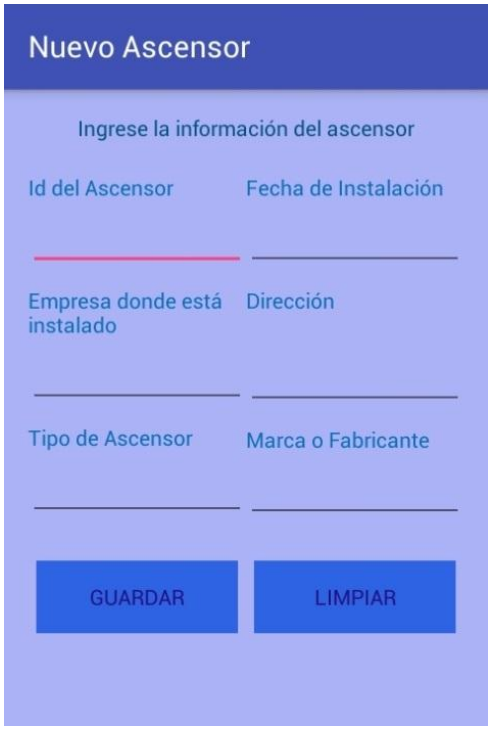

Figure 10. Enter new elevator. Source: Authors

All these actions make querys to the database, these querys are made by asynchronous methods that allow the execution of processes in the background so that the implementation of the application is not degraded (Algorithm 5).

```
public void obtenerAscensores(){
    AsyncTask<Void, Void, Void> task = new AsyncTask<Void, Void, Void>() {
        @Override
        protected Void doInBackground(Void... params) {
            try {
                final List<ElevatorItem> eItems = getItemsFromMobileServiceTable();
                runOnUiThread(new Runnable() {
                    @Override
                    public void run() {
                        adapter.clear();
                        for(ElevatorItem elev: eItems){
                            ElevatorInfo item = new ElevatorInfo(elev.getmId(),elev.getCompany());
                            adapter.add(item);
```

Algorithm 5. Asynchronous query method. Source: Authors

The code above shows the implementation of the asynchronous method that performs the query of all the elevators and shows them in the main screen of the application.

NOTIFICATIONS AND PERFORMANCE

To verify the performance of the application, a series of fault reports were made and the time it took to notify the front-end application was observed. While the response is quite fast around five seconds, in the Figure 11 obtained from the Azure portal

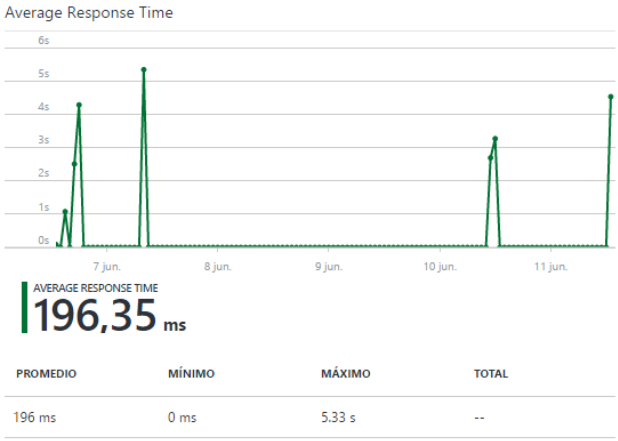

Figure 11. Metric obtained from the Azure portal. Source: Authors

With respect to the application, it takes about 10 seconds to start the connections and make the necessary queries for the operation of the application, there are cases where it is delayed a little more, for this the application is configured to wait at startup A maximum of 20 seconds (Algorithm 6).

```
mIClient.setAndroidHttpClientFactory(new OkHttpClientFactory() {
    @Override
    public OkHttpClient createOkHttpClient() {
        OkHttpClient client = new OkHttpClient();
        client.setReadTimeout(20, TimeUnit.SECONDS);
        client.setWriteTimeout(20, TimeUnit.SECONDS);
        return client;
```

Algorithm 6. Method to change the timeout when starting the application. Source: Authors

DISCUSSION OF RESULTS

In [1] a connection model is proposed between the sensor and the monitoring application, this article uses this idea giving good results by including the three components differently but using the same principle which is to report the fault through Of the network and be received by a client application.

The design of the sensor can be based on what is exposed in [2], however the most important thing exposed in [2] is the use of the existing Ethernet network, in the current article there are no restrictions with respect to the architecture of the network However it is necessary to clarify that measurements were made with Ethernet and 3G mobile networks.

The current system only deals with faults, although it could extend its functionality to also function as elevator manager as explained in [3] and [4], since nowadays everything is wanted to be optimized by including a Fault reporting and a usage report can improve the operation of elevators.

The system proposed in [5] is quite large to include the different types of cloud implementations (PaaS, SaaS, IaaS) in the present development only PaaS was used, but also gives arguments and results to consider the hybrid cloud as a Cloud computing, although only the public cloud was used in this development and it was quite efficient to consider whether organizations want everything in the public cloud or if they want to keep important data within their infrastructure using the hybrid cloud.

CONCLUSIONS

Cloud architecture is a good choice when a scalable, reliable, always-on application is required; It is also an appropriate solution for companies that do not have the necessary equipment for development, but nevertheless want to provide web-based services. With the services provided by the public cloud all the deployment and equipment run on the part of the provider so that companies focus on the application and services provided and not on maintenance and purchase infrastructure.

Cloud platforms provide the mechanisms to make the transition to the cloud not so difficult, they also provide the tools to deploy everything that is required, for example in a mobile application notifications are required, this is supported by platforms in Cloud.

Another important aspect in the cloud is that it facilitates the ways of developing any type of application and for any type of client technology, so it is quite flexible with respect to this.

The cloud architecture is quite appropriate for the Internet deployment of things, since it provides the bridge between the devices that send information and the application that will receive it, either to be analyzed by a person or to perform Machine Learning.

The databases and their multiple applications and uses are essential for the management of information, this is why all the development of this application is based on the tools they provide, the database is stored all the information of the lifts And also the entire log of fault information, also provides the mechanism for the implementation and operation of the notifications. Another important factor provided by databases and that could be used for future development is the ability to analyze data through data mining and obtain patterns of use and improvement, however data analysis could lead to problems of privacy and security as Are discussed and analyzed in [6] and [7].

REFERENCES

[1] Xibo Wang and Hongshuai Ge; Wenbo Zhang; Yingzhen Li. Design of Elevator Running Parameters Remote Monitoring System Based on Internet of Things. China, 2015.

[2] Tundong Liu; Xiaosheng Liao; Jianping Zeng. Design of Intelligent Elevator Remote Monitoring System Based on Ethernet. China, 2010.

[3] Hamza Ijaz Abbasi; Abdul Jabbar Siddiqui. realizaron Implementation of Smart Elevator System based on Wireless Multi-hop AdHoc Sensor Networks. India, 2011.

[4] Rosso Mateus, Andrés Enrique; Soriano Méndez, José Jairo. DESARROLLO DE UN SIMULADOR DE EDIFICIO CON PARÁMETROS VARIABLES. Colombia, 2009.

[5] Shaik Abdul Nabi; Dayakar Gurra; Mohd.Anwar Ali. Mobile Hybrid Cloud Computing for Educational Institutions: Mobihybrid EduCloud. India, 2015.

[6] Yashpalsinh Jadeja; Kirit Modi. Cloud Computing - Concepts, Architecture and Challenges. India, 2012.

[7] Liu Wenato. realizó Research on Cloud Computing Security Problem and Strategy. China. 2012.

[8] Yanuarizki Amanatullah; Charles Lim; Heru Purnomo Ipung; Arkav Juliandri. Toward Cloud Computing Reference Architecture: Cloud Service Management Perspective. Indonesia.

[9] Samah Ahmed Zaki Hassan. STAR: A Proposed Architecture for Cloud Computing Applications. Egipto, 2012.

[10] Yong Pan; Ning Hu. Research on Dependability of Cloud Computing Systems. China, 2014.

[11] Tutorial y documentación de Azure. https://azure.microsoft.com/es-es/documentation/services/app-service/mobile/.

[12] Windows Azure:

http://www.desarrolloweb.com/manuales/windows-azure-dotnet.html