Webology (ISSN: 1735-188X)
Volume 19, Number 6, 2022

Iot Architecture For Automated Parking Systems

Luis Eduardo Ramírez Carvajal [1], Msc. Karla Cecilia Puerto López[2], PhD. Francisco Ernesto Moreno Garcia[3], Msc. Haidee Yulady Jaramillo [4], Msc. July Andrea Gomez Camperos[5*]

[1,2,3] faculty of engineering, Universidad Francisco de Paula Santander, Cucuta, Colombia.
[4] Department of Civil Engineering, Faculty of Engineering, University Francisco de Paula Santander Sectional Ocaña, Ocaña, Norte de Santander.

[5] Mechanical Engineering Department, Faculty of Engineering, University Francisco de Paula Santander Sectional Ocaña, Ocaña, Norte de Santander.

Abstract
The following paper presents the development of an IoT architecture for autonomous parking systems. The research is performed using a DFSS (Design for Six Sigma) methodology composed of 4 phases: requirements identification, design characterization, design optimization and design verification. It is implemented in a scaled prototype using Arduino for data acquisition, Android language for user interface development and Java language for control software development. The architecture is composed of three main blocks: User, System and Parking. The results show that the implementation of parking systems is presented as an alternative to optimize mobility by reducing the time required for parking and minimizing vehicular congestion. The prototype allowed concluding that the architecture is functional in a business context if it is to be implemented on a large scale as a massive mobile application. The interface optimizes communication allowing better interactivity between the user and the parking lot.

Keywords: Architecture, IoT, Systems, Parking, Autonomous, Software.

1. Introduction

The automotive sector in Colombia recorded significant growth in production and sales in 2012 [1]; and although in recent years this has decreased, the vehicle fleet continues to be a challenge for the road infrastructure we currently have [2]. This causes demand for parking lots, making them an important element within the transportation policy [3]. Needs such as these, which demand the construction of large and complex software systems, sometimes requiring the combination of different technologies and platforms, have imitated methodologies similar to those used in construction. Hence, computer science adopted the concept of "architecture". As in large construction projects, to guarantee the success in the development of a software application, it is first of all required a good definition of the structure to be followed, of the different elements or modules to be built and how they interact with each other in a safe and efficient way [4].

The term "software architecture" officially appeared in 1992 with the work of Perry and Wolf [5] and over time has been implemented in projects in various areas such as business intelligence [6], medical prescription systems [7], critical railway systems [8], cultural tourism applications [9], video game development [10], among others. Considering the excellent results that a software architecture provides to a project and in view of the latest research on IoT in the optimization of parking lots in large cities such as Bogota [11] and Medellin [12], the objective of the research was to develop an IoT architecture for autonomous parking systems.

2. Methodology

The research was developed using a methodology based on DFSS (Design for Six Sigma) [13] composed of 4 phases: requirements identification, design characterization, design optimization and design verification. This methodology can be seen in Figure 1.

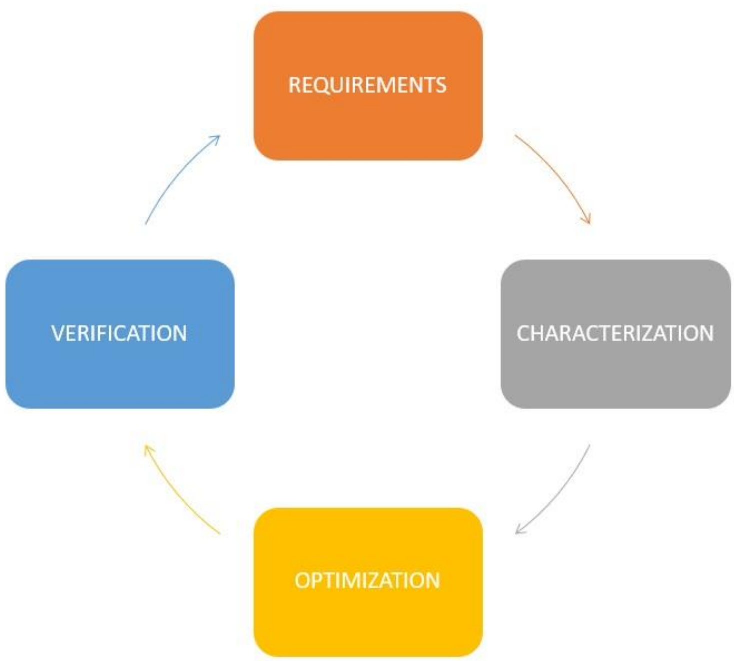

Figura 1. Metodología utilizada en la investigación.
Fuente: Elaboración de los autores

2.1 Requirements identification

The description of the services provided by the system and their operational limits are identified, taking into account the need and the order of priorities of each one of them. It is important to highlight that the process is qualitative and analytical, taking into account that the basis of this item is the classification of functional requirements and measurable quality attributes [13] whose organizational foundation is oriented by the general establishment of: specific objectives, the scope of the architecture, acceptance criteria and risk consideration.

2.2 Design characterization

With the clear information of the requirements process, a systematic analysis is carried out to develop an algorithm represented by a diagram. The main objectives are the development of architectural design alternatives based on the requirements and quality attributes, as well as the verification of the architecture. The activities that allow fulfilling this objective are: the review of the scenarios, the inductive decomposition of concepts that underlie in concrete modules and represent specific elements of the architecture, the selection of architectural styles and patterns of reference and finally, the qualitative and quantitative consideration of the final interfaces.

2.3 Design optimization

It is a process of continuous improvement that allows constant reflection on the effectiveness, scope and feasibility of the architecture design throughout the process. DFSS methods are dynamic and very practical, oriented to a repetitive cycle of optimization based on analysis of results and tuning parameters, among which improvement optimization is highlighted as a fundamental step to achieve that purpose. Tools such as the Taguchi method [14], critical thinking as robustness assessment and reliability-based design [13] are used at this point to carry out the optimization purpose.

2.4 Design verification

Finally, the design verification process is carried out, which consists of technical tests and evaluation of final results, measuring the effectiveness of the architecture through the implementation at scale of a prototype based on free-use platforms. Taking into account the processes and analysis on which this research is based, this point is approached taking as a reference the methodology used by Cubillos and Rodriguez [11] who established as primary components, the application architecture, infrastructure, data and business. Also based on the analysis performed by Pérez and Posada [12] it is considered of utmost relevance to perform the validation of the design through the implementation of a mobile application using Android language. And finally, based on the excellent results obtained by Rosales [15], the data acquisition through the Arduino platform is developed.

3. Results and discussion

Figure 2 shows the general diagram of the proposed architecture.

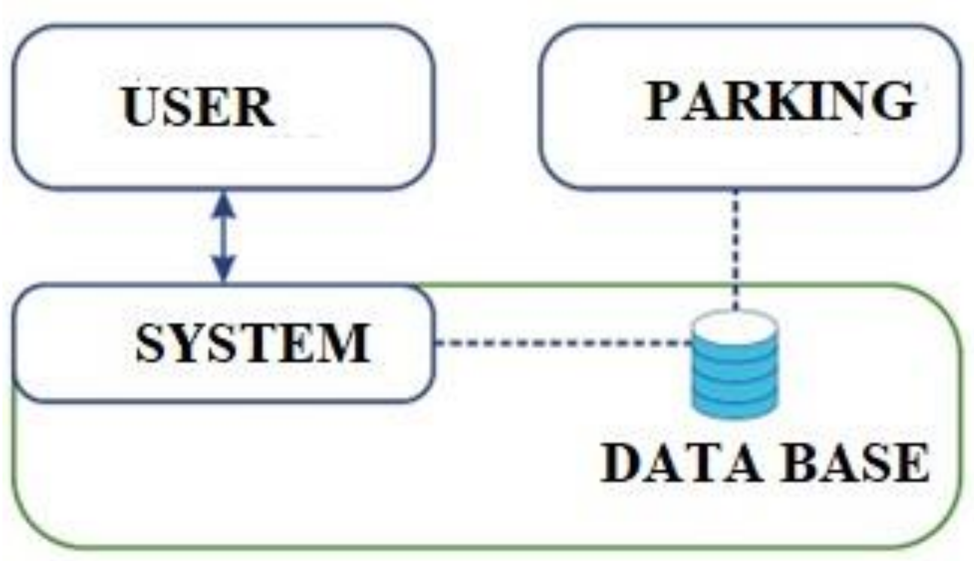

Figure 2. General architecture diagram

The architecture consists of three main blocks: User, System and Parking. The user interface allows five actions to be performed: search for parking, reserve parking space, check in, check out, and make payment.

The parking block has a control system, either manual, automated or mixed, where the information on available parking spaces is sent to the system and connected to the electronic actuator system to control the entry and exit of vehicles by means of hydraulic arms or mechanisms available on site.

The system interconnects the parking lot with the user and a database that keeps track of available parking spaces. It allows the parking lot to control the parking time and execute the corresponding charge, offering support to communicate the existing charging system or implement a new one, in case it does not have one.

The architecture allows the integration of the technology that best suits the environment or the availability of the area where it is implemented. In the case of this research, the design verification is developed using Arduino, Android and Java, as shown in Figure 3.

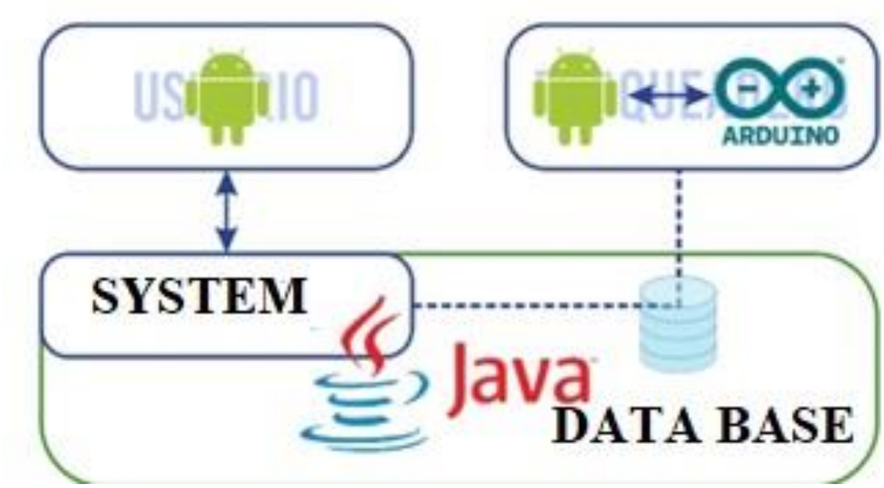

Figure 3. Technology used for design verification

The database is fed and controlled only by the parking lot and the system. The system, developed in Java, displays information to the user through an interface programmed in Android language. The parking lot has an interface, also developed in Android, which communicates with the data acquisition system developed in Arduino to obtain the information and feed the database.

Using the scaled prototype, quantitative tests are carried out and it is established that parking time is reduced by 30%. The time it takes for cars to leave the parking lot is also reduced by 10%. This reduction is due to the time the driver spends looking for an available parking space and, when leaving, the time required to pay manually.

By performing tests with a scaled prototype that allows estimating the real time a car spends in a parking lot both at the entrance and exit, qualitative analyses are performed that allow concluding that at a methodological scale, the architecture could be implemented through the development of a massive application that allows not only finding a parking space within a parking lot, but also an available parking space within a determined perimeter, which, added to the factors that affect road mobility in a city, replicates its effectiveness at a macro level and allows reducing the measured times by the same or higher percentage.

4. Conclusions

The implementation of parking systems optimizes mobility by reducing the time required for parking and minimizes vehicular congestion considering that, by finding a parking space and available spot quickly, the time and number of vehicles that travel the roads of a city are reduced.

The developed architecture is useful for the implementation of autonomous parking systems, being effective in reducing time and allowing multiplatform incorporation depending on the availability of technologies in the market or specific geographic area.

The prototyping at scale allowed concluding that the research based on software development had a multidisciplinary scope, because the results and analysis led the architecture to be functional in a business context if it is to be implemented on a large scale as a massive mobile application.

Finally, it is important to conclude that the interfaces optimize the communication allowing a better interactivity between the user and the parking lot.

Referencias

[1] N. Cantor, J. P. Herrera y D. Sánchez, «Estudio del sector automotor en Colombia,» Superintendencia de Industria y Comercio, Colombia, 2012.

[2] BBVA, Situación Automotriz 2018 Colombia, BBVA, 2018.

[3] L. Macea, L. Márquez y J. Soto, «Elasticidad de la demanda de estacionamiento ante variaciones sistemáticas del ingreso en Montería, Colombia,» Ingeniería y desarrollo, vol. 35, nº 2, pp. 513-532, 2017.

[4] G. A. Hernández Cabrera, «Arquitectura de software para la construcción de un sistema de cuadro de mando integral como herramienta de inteligencia de negocios,» Tecnología, Investigación y Academia, vol. 5, nº 2, pp. 143-152, 2017.

[5] L. F. Fernández, «Arquitectura de software,» Software Guru, vol. 2, nº 3, pp. 40-42, 2006.

[6] G. A. Hernández Cabrera, «Arquitectura de software para la construcción de un sistema de cuadro de mando integral como herramienta de inteligencia de negocios,» Tecnología, Investigación y Academia, vol. 5, nº 2, pp. 143-152, 2017.

[7] T. Ariza, F. Fernández Jiménez, J. Muñoz Calle, A. Sierra y M. Mora, «Arquitectura software para la prescripción de ejercicio físico personalizado,» Revista Internacional de Medicina y Ciencias de la Actividad Física y el Deporte, vol. 19, nº 73, pp. 137-150, 2019.

[8] J. Acevedo Duprat y E. Irrazábal, «Evaluación de la calidad en patrones de arquitecturas software para sistemas críticos ferroviarios: un enfoque basado en la mantenibilidad,» de XXV Congreso Argentino de Ciencias de la Computación, Río Cuarto, 2019.

[9] G. E. Chanchí, M. Saba y M. E. Monroy, «Propuesta de una arquitectura software basada en realidad virtual para el desarrollo de aplicaciones de turismo cultural,» RISTI - Revista Ibérica de Sistemas e Tecnologias de Informação, pp. 157-170, 2020.

[10] A. Hernández Paez, J. A. Domínguez Falcón y A. A. Pi Cruz, «Arquitectura de software para el desarrollo de videojuegos sobre el motor de juego Unity 3D,» Revista de I+D Tecnológico, vol. 14, nº 1, pp. 54-64, 2018.

[11] N. Cubillos García y J. S. Rodríguez Mora, Arquitectura IoT para parqueaderos inteligentes en la ciudad de Bogotá, Bogotá: Universidad Católica de Colombia, 2018.

[12] P. Pérez y N. Posada Matickis, Estudio de viabilidad de una aplicación móvil para parqueaderos públicos en la ciudad de Medellín, Envigado: Universidad EIA, 2018.

[13] A. Núñez Mora, Proceso para el desarrollo de Arquitecturas de Software basado en DFSS, Guanajuato: Centro de Investigación en Matemáticas, 2006.

[14] G. Del Carmen Urbina, «Creación de un algoritmo autoajustable mediante el método Taguchi aplicado a un sistema de segundo orden,» Academia Journals, vol. 11, nº 2, pp. 605-608, 2019.

[15] L. N. Rosales Lindao, Diseño e implementación de un parqueo inteligente utilizando Arduino YUN basado en internet de las cosas (IoT), Guayaquil: Universidad Politécnica Salesiana, 2016.