

# 5<sup>th</sup> IEEE International Smart Cities Conference (IEEE ISC2 2019)

October 14-17, 2019 Casablanca, Morocco

[ieee-isc2.org](http://ieee-isc2.org)



## Organizers



## Partners & Sponsors



IEEE Catalog Number : CFP19B42-ART

ISBN: 978-1-7281-0846-9

Copyright and Reprint Permission : Abstracting is permitted with credit to the source. Libraries are permitted to photocopy beyond the limit of U.S. copyright law for private use of patrons those articles in this volume that carry a code at the bottom of the first page, provided the per-copy fee indicated in the code is paid through Copyright Clearance Center, 222 Rosewood Drive, Danvers, MA 01923. For reprint or republication permission, email to IEEE Copyrights Manager at [pubs-permissions@ieee.org](mailto:pubs-permissions@ieee.org). All rights reserved. Copyright ©2019 by IEEE.



[Date]

# Small steps towards a smart city: Mobile application that provides options for the use of public transport in Juarez City.

Luis Holguin  
Autonomous University of  
Juarez City  
Ciudad Juarez, Mexico  
al98616@alumnos.uacj.mx

Alberto Ochoa-Zezzatti  
Autonomous University of  
Juarez City  
Ciudad Juarez, Mexico  
alberto.ortiz@uacj.mx

Victor M. Larios  
Smart Cities Innovation Center  
CUCEA  
University of Guadalajara  
Guadalajara, Mexico  
victor.m.lariosrosillo@ieee.org

Edgar Cossio  
Instituto de Informacion  
Estadistica y Geografica de  
Jalisco  
Zapopan, Mexico  
edgar.cossio@jalisco.gob.mx

Rocio Maciel  
Smart Cities Innovation Center  
CUCEA  
University of Guadalajara  
Guadalajara, Mexico  
ma.maciel@academicos.udg.mx

Gilberto Rivera  
Autonomous University of  
Juarez City  
Ciudad Juarez, Mexico  
riveragil@gmail.com

**Abstract**— This project developed an Android application to provide users with options on which public transportation routes to use in Juarez City. To make this possible, we went to request information from the various lines of the city to the Transportation Office of the State Government. Subsequently, various tools were developed to support the process of capturing routes, editing, and obtaining intersections between them. The solution consisted of a client-server architecture, where the client is an application for mobile devices, which is used by users, and the server is the application responsible for returning possible routes to customers. We achieve this with the help of Quadtree algorithms to partition the space and thus obtain coordinates of routes near the origin and destination. It results in an efficient manner using Dijkstra for the search of roads using the routes near the beginning and the destination. Finally, to validate its usefulness, a survey was conducted to UACJ students, which consisted of three sections, the first on personal data of the respondent, the second on the use of public transport and the third about the Android application.

**Keywords**—component, formatting, style, styling, insert (key words) Graph theory, routing problems, pathfinding, Dijkstra.

## I. INTRODUCTION

For some people, public transport is the heart and arteries that distribute the working blood that moves the city. Mainly for being a means of transport so accessible for the population that does not have its vehicle. Also, by giving the possibility to its users to be able to approach their destinations and rid them of worries, for example, when the car fails. The population needs to know how to move between all the publicly available lines. Unfortunately, the public transport service is deficient in this city. Following this trend is the information regarding these since the concessionaires of these transport lines do not provide the corresponding authorities with the necessary information so

that they can provide accurate information about their routes, schedules, alternate routes, in another.

### A. Related research

At present, several applications contain functionality similar to that proposed by this project. Here are some of the most noteworthy:

Moovit - is an application belonging to an Israeli company, available for Android devices and IOS works in more than 70 countries. In Mexico, Moovit is present in cities such as Mexico City, Guadalajara, Monterrey, and Merida. It allows to show approximate times for the routes, shows the routes step by step, and tells the user how much to expect, where to change and where to go [1].

Google Maps - has the functionality Google Transit which by clicking on the icon of the bus and selecting a destination displays the route and the buses that must be taken to reach it, it is available for Android or IOS [2]. It is not yet available for Cd. Juárez.

Citymapper - is an application for Android, among its features can find the best route by combining the different means of transport available highlighting a variety of buses. It has the schedules of the metro, metro, among other functions. [3].

Metro Metrobus México - this application provides information on the different Metro lines. Its features include the possibility of finding stations close to the user's position, map of all the subway lines, location of stations and transshipments, schedules, and costs [4]. The functionality for which this application is essential is to determine the route, taking into account the nearby stations to easily reach the destination.

Transit Metro, Bus, Tren - this application for Android devices is available in many of the largest cities in the world,

including Mexico City. Within the App, although it is not possible to automatically search for the best route, it allows being able to plan journeys using the different means of transport available in the city. It also enables the user to save routes, pinpoint locations on the map for accessible location, and plan routes faster [5].

We observe similar works implemented satisfactorily, and with this concept, we consider it possible to develop a similar tool for Juarez City.

### B. Problem

In Juarez City, information on the routes that public transport buses have is not non-existent, but it is difficult to access it. At present in Cd. Juarez, there are more than thirty public transport lines, which without taking into account its ramifications. To reach a new location, citizens and visitors ask, either to the drivers, or other users as the best practice.

## II. FRAMEWORK OF REFERENCE

In this section, we present the subtopics that are part of the referential framework. First is the theoretical framework, which will deal with several concepts that are useful to understand this work, as it is to know what a graph and its representation, path search algorithms and their usefulness in graph problems. Then there is the technological framework, where we focus on developing software for the Android system and the server platform into a private cloud. Some of the used libraries for the development of this project are also listed.

### A. Smart Cities

It is a term that began to be used in the 90s to refer to cities that thanks to the integration of information and communication technologies (ICT), energy, and transport have progressed [6]. They begin by modernizing their critical infrastructure to provide sustainable development, help improve the efficiency of available resources, and thereby improve the quality of life of citizens [7]. Among the advantages presented by Smart Cities are improving the efficiency of public administrations, interconnecting the different systems that make up the cities, facilitating the management of a higher number of public services, reducing the traffic of the transportation systems, improving the speed of response in emergencies [6].

### B. Graph theory

Every day in our lives, we face this type of connectivity structure, such as the telephone cables that hang on the poles, the electric power cables of the city, even on the Internet providers. In mathematics, we form a graph by a set of vertices or nodes and a set of edges, each edge connecting two nodes. Figure 1 shows the visual representation of a graph. We represent the nodes with the colored circles and the edges with the lines of the same color.

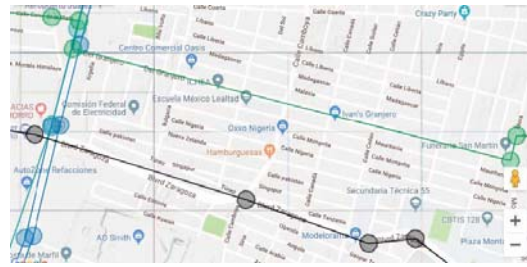


Fig. 1. Transport lines representing a graph in a Gmaps map.

### C. Routing problems

This type of problems has applications in planning, transport logistics, and distribution, including in the development of electronic circuits. The problems in which we look the optimal route from one or several origins to a group of locations are considered problems of vehicle routing [8]. The TSP is a routing problem considered NP-Hard, as the cities increase, the solution time increases exponentially. One way to obtain competitive paths for this problem is through the ant colony algorithm. This approach consists of using virtual ants that visit each city depending on a previously defined heuristic and for each edge through which they pass pheromones, the amount released will depend on the quality of the road. Just as real ants release pheromones on their way according to the quality of the food found, according to the number of ants that visit the road, the best route will have the highest number of pheromones [9].

### D. Dijkstra's algorithm

This algorithm arises in order to find the shortest distance from a node of origin to any other specified node, was created in 1959 by Edsger Wybe Dijkstra. Each node is assigned a weight; this weight is a value that represents how short the distance of that node is at the beginning. This algorithm is considered to belong to the algorithms  $O(n^2)$  where  $n$  is the number of nodes belonging to the network, given that increasing the number of nodes usually increases the solution time [10]. Consists of broad strokes, having initialized the distances between nodes, grouping the nodes or vertices into three types, the nodes marked, adjacent, and distant. The initial node is a marked node; the adjacent nodes are nodes connected to a marked node, and the distant nodes are nodes that have not yet taken into account. We must also consider that each adjacent node saves which node was accessed. The procedure is as follows, the initial node is a node marked as visited, and with zero cost, we mark the adjacent nodes to the current node and its cost is computed from the current node. At this point, the algorithm will cycle until all the nodes are marked nodes, from the marked nodes, the lowest cost is taken and becomes the current node. The nodes connected to the current node that was adjacent to another node become adjacent nodes of the current node if its cost from the current node is smaller than they already have [11].

Considering the hardware processing limitations of mobile devices, We decided to use the Client / Server architecture. In figure 2, we present the architecture diagram. We represent the flow of information with arrows. This network architecture consists of two applications. First, the Client application used by the end-user placing queries to one or more servers. Second, The server is a system responsible for receiving queries from one or

several clients and serves to help them perform previously defined tasks during the system execution.

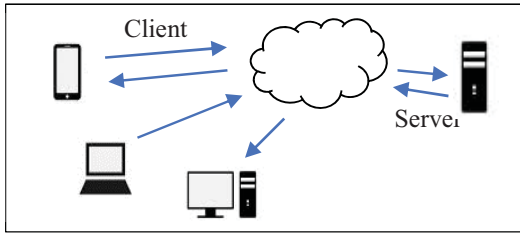


Fig. 2. Example diagram of the client/server architecture.

### III. DEVELOPMENT

In this section, the most outstanding aspects are presented during the development of the project, as they are: the diagram of the proposed solution, the methodology used, and the development process.

#### A. Proposed solution

In recent years, smart cities have become one of the most frequent topics in terms of technological development; such is the case of this project that aims to develop a mobile application to help the population of Juarez City in the planning of routes using transfers of public transport in the city. How the study was carried out to verify the benefits that the system can provide was through the results obtained after surveying 50 volunteers. Hence in Figure 3, we can see the diagram of the proposed solution.

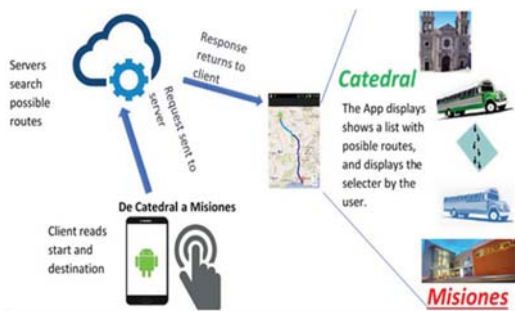


Fig. 3. Scheme of the planned solution.

This information had certain drawbacks. First, we delivered it in an unfriendly format for the needed purpose. The information consisted of PDF files, where lines represented the route of the routes and branches without any way of indicating their direction and drawn in different layers on the city map. Second, the personnel in charge reported that at that time, the files of all the city lines were not available. Figure 4 shows one of these files; in it, we can see the layers that compose it, each of the layers stores a different drawing.

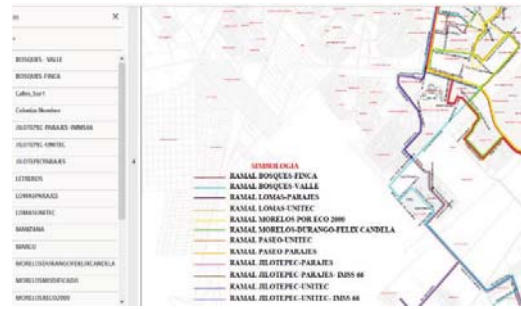


Fig. 4. Picture showing the PDF file containing bus line 1A.

For the reasons mentioned above, it was necessary to establish a format to be able to capture, store, and work with the information provided. The first attempt to capture the routes consisted of pressing the mouse pointer on a Google map, selecting and copying the address and coordinates of that point from a drop-down message and pasting it into an Excel file, the second step of the capture was to give the JSON format row by row, these procedures had to be done more than 300 times per route easily, later the relevant information was copied and pasted manually to a structure in JSON format, given the tediousness of this method, it was more useful to develop an application that accelerates the capture of directions and coordinates. Due to the previous situation, it was necessary to build a tool to facilitate the capture of routes of the transport lines. This capture tool was created with the help of the Gmaps API to show maps and get directions based on coordinates of map points and JavaScript to give it its functionality, but it was later changed to use the Here.com API to obtain the information of the coordinates due to the limit amount of Gmaps queries. The application consisted of three text fields for the name of the route, name of the branch, and another to show the route captured in JSON format. We can see the interface in figure 5. The files resulting in a start were separated into two sections manually to represent the direction of the route concatenating to the name of the route the characters "-a" if it goes from north to south and "-b" from south to north. In figure 6, we can see the structure storing the routes of the routes.

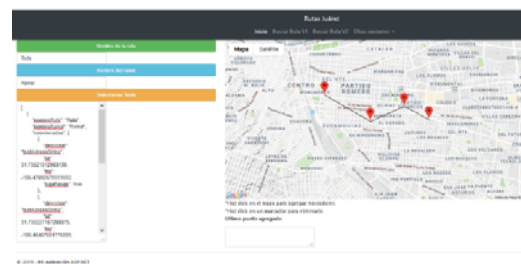


Fig. 5. Route capture module V1.

```

[
  {
    nombreRuta:"2-L-a",
    nombreRamal:"perio",
    color:"FF00FF",
    sentifo:122,
    coordenadas:[
      {
        direccion: "Privada General Martín
        Triana 3209, Emiliano Zapata, 32210 Juárez, CHIH, México",
        lat: 31.72151,
        lng: -106.50622,
        bajaPasaje: true
      }
    ]
  }
]

```

Fig. 6. Captured route line file.

In order to perform the route search, we treated the problem as a routing problem. We can visualize into a graph the different lines of the city and their connection. Each coordinate of a route corresponds to a node, and the proximity between them is an edge. Hence, we decided to use families of algorithms searching for roads. We opted for the algorithm of Dijkstra to try to find the solutions. This method serves to find the shortest route in a graph from a given node to another one, that is to say, it would only be obtained a solution, to obtain more than one solution. Selecting the roads from the nodes connecting to the destination node, we can use as another possible solution the path traveled until the beginning of each of these nodes and was added. It was decided to develop a client-server model, where the client was designed as the Android application that was responsible for sending the requests to the server and show the result returned by the server, the server is an application hosted on the Microsoft Azure platform, responsible for performing the search of roads, and return the result in JSON format to the respective clients. Given a large number of points that build the routes, it was considered necessary an effective way obtain the points near the start and endpoints to generate the connection between them because initially the origin node and the destination are not part of the graph. With this objective, it was decided to use a quaternary tree (*Quadtree*), which is a useful data structure to partition the space and thus obtain points close to a given point without the need to iterate them all. Next, figure 7 shows the preview of a route file, where each circle represents a point that shapes it and the squares visually represent a *Quadtree*, each of these subdivisions encloses a group of points.



Fig. 7. Preview of the captured route coordinates.

### B. Prototype evaluation by the client

We developed the first server prototype fulfilling the functionality of showing routes, but results were not feasible because the server application not necessarily returned the shortest route. This issue caused an excessive number of transshipments. It also resulted in hardly granted routes of two or three bus transfers and some cases they were more than four. We created the first client along with the first server prototype. The server was deployed on the web with HTML and JavaScript due to the required speed to update. Its functionalities were simple, fill two text fields with the name of the location to which we wanted to go, save our coordinates in JavaScript variables with the help of the Here.com API. After these, we send the request to the server, and we draw a line of a single color the received coordinates. This first client was useful to visualize the results, but it was not helpful to visually distinguish between routes because they could not differentiate between one another, in figure 8 we present the result of the first client.

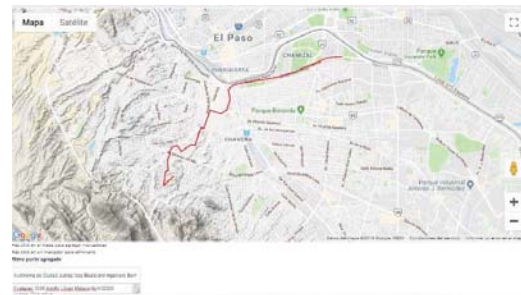


Fig. 8. First client prototype results.

The second server prototype solved the problem of the first; this granted routes with fewer transfers. We separate the route files into two items to represent a round trip, and a file corresponded to a node in the graph. However, this method caused another problem, made unnecessary roundabouts for the following reasons, only returned a path regardless of whether there were others with the same cost of transshipment, ultimately returned the coordinates of the files of the nodes used, it was unknown where the transfer occurred. With this second server, we developed two client prototypes. The first client is for the webserver and the second client is for the Android application. Therefore, we modified on the web, only the interface, and the structure of the responses received. We copied the web functionality to draw the results received from the second server prototype and in the Android client. In figure 9, we show the second version prototypes.

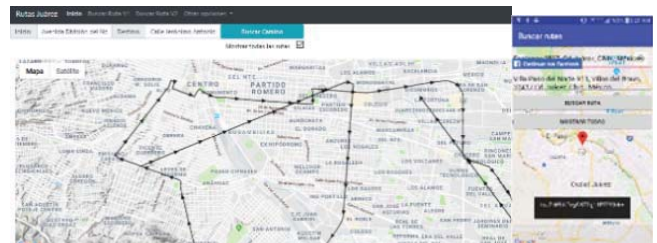


Fig. 9. Web (left) and Android (right) prototypes.

### C. Engineering product

With the development of the V3 prototypes, we introduce differences concerning the previous versions. In the first instance, we produced the client application for Android with all the functionality of its web version. The notable difference was the improvement in the user interface. The application named Rutas Juárez, not found in the Google store has five windows. Main screen. We created this window with the purpose that by pressing the black/green buttons for start and black/red for the destination, we would see a cursor that the user can address in the position of his choice. By pressing the text fields, we display the autocomplete search window. Once we fill the text fields, we place the query to the server. Figure 10 shows the main screen, with an interface similar to other transport applications. On the left side is the state with which it starts, on the right side as we see when we press the buttons to position us on a location manually.

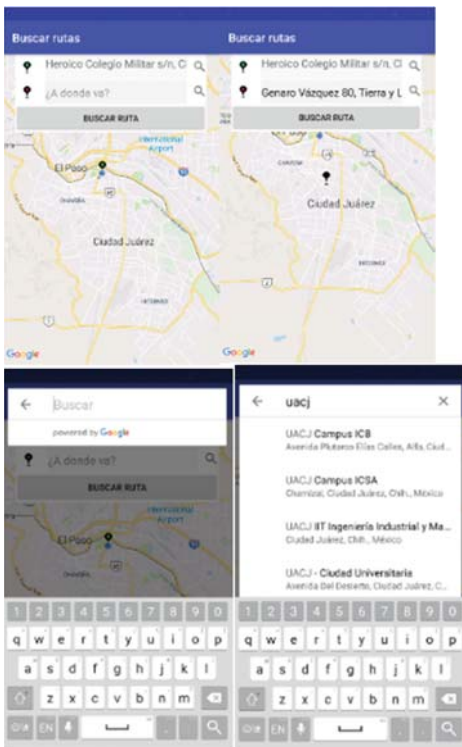


Fig. 10. Main window before and after selecting one of the text fields.

Once the query is received, the drop-down list is displayed. The list is filled after making a query and obtaining a result from the server. In figure 11 it can be seen that a list of indications that can be hike or name of the route is shown, in the lower part of the route, the number of transshipments, the distance walked in meters and the bus travel in kilometers.

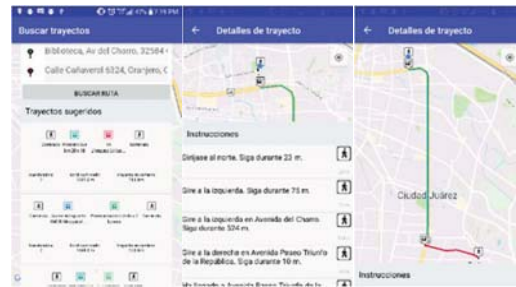


Fig. 11. Search results.

## IV. RESULTS

After presenting the application to 50 UACJ students, they were given a questionnaire with 11 questions where the first 3 were their general data (sex, age, marital status), four were related to their use of public transport and the rest about your opinion after using the application. We take into account only the results of questions 1 to 8, which correspond to the use of public transport and the use of the App. We can observe that slightly more than half of the students use less than three routes daily. The most frequent destinations among the surveyed students are the school and the home easily surpassing the work and recreational zones. We also can look in figure 24 that, to the majority of the volunteers, were sometimes happened to them that they needed to move towards some destination, but they did not know buses needed to take. The results obtained show that for most users, the application was easy to use. Few responses claimed that the App was challenging to manipulate. We can see a positive trend at the level of utility concerning the results suggested by the application according to the people surveyed. Regarding the last question, where the respondents should answer on what would improve the application, and we obtained 36 comments, of which six are not taken into account because they are not useful.

### A. Discussions

Taking into account the results obtained from questions 5 to 8, we can consider that the application complies with helping citizens by providing travel options to reach their destinations through public transport. According to some users, the addition of more routes is an essential improvement for the application. This issue is a requirement already taken into account as a possible improvement. Another suggestion given by the users was to deploy the application on IOS. We discard for the moment such a proposal because it is outside the limits of this project. One of the most notable comments in the recommended improvements was the addition of photos of the buses since not everyone knows them. The most recurrent comments were the improvement of the interface, in that they did not know how to use the application. We agreed with such remarks because it was decided not to devote much time to the design of the interface. Although we can solve this problem by adding a small tutorial, at the beginning of the App.

## V. CONCLUSIONS AND FUTURE RESEARCH

### A. Regarding the objective

At present, there is no accessible way to consult the information in Juárez City. To solve this problem, we developed

this project, and a survey was subsequently carried out in order to know the opinions of the people who used it and to verify whether it fulfills its objective or not. Thanks to the results given by the volunteers, there is evidence that the application meets its objective of helping citizens obtain routes for mobility through the different public transport lines of Juarez City.

### B. Recommendations in the future

As possible future work, we consider some of the answers given by the users. We describe as follows, the improvement works we consider of relevance. First, we need to filter results so that we do not display those that are not feasible. Second, we need to contact the different concessionaires to obtain the times of travel of the lines and their branches, the times between bus exits, and even the possibility of in the future having satellite locators in all the units. Third, we require to add points of interest throughout the map of the city but showing only those located around the route selected by the user. Fourth, we need to explore the use of other algorithms or methods to attempt to obtain better results. Fifth, add pictures of the buses. Finally, the sixth relates to adding more routes.

### C. Future Research

Being a project that improves a route, we can use it not only for schools; we can use it can also for parcel delivery or transport companies. Using a mobile device associated with a more detailed framework, one could even determine specific days with more transport objects associated with each student, as can be seen in figure 12:

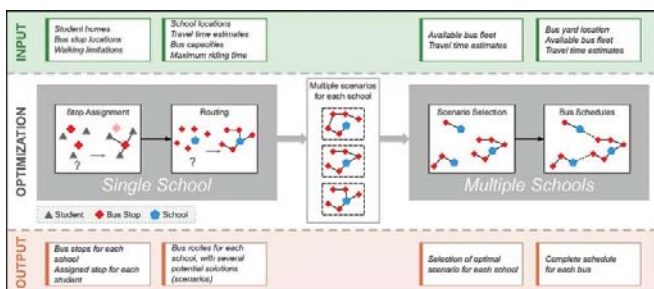


Fig. 12. Intelligent Design of a mobile application based on Ubiquitous Computing that would allow each student to send information associated with the transport of musical objects of a band and online specifications of each particular situation, of the dimensions and care of their respective musical instruments.



Fig. 13. The OECD considers that by 2027, the association of seven of the 12 components of a Smart City with improving school performance, including student mobility.

An aspect of utmost importance will be to adapt over time, the specifications of each individual in childhood that allows it to be competitive, helping with mobility in a Smart City, as can be seen in figure 13.

### References

- [1] M. Inc., «www.company.moovitapp.com,» [En línea]. Available: <https://play.google.com/store/apps/details?id=com.tranzmate>.
- [2] Google Inc., «https://play.google.com/store/apps/details?id=com.google.android.apps.maps,» Google, 20 Septiembre 2017. [En línea]. Available: <https://play.google.com/store/apps/details?id=com.google.android.apps.maps>.
- [3] I. S. Jacobs and C. P. Bean, “Fine particles, thin films and exchange anisotropy,” in Magnetism, vol. III, G. T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271–350.
- [4] yalla ya! , «https://www.yalla-ya.com/,» [En línea]. Available: <https://play.google.com/store/apps/details?id=com.yallaya.metrotrob.us>. [last access: September 21, 2017].
- [5] Transit App, Inc., «https://transitapp.com/,» 24 Agosto 2017. [En línea]. Available: <https://play.google.com/store/apps/details?id=com.thetransitapp.droid>. [Las access: September 21, 2017].
- [6] J. M. Fernández Güell, «Ciudades Inteligentes: La mitificación de las nuevas tecnologías como respuesta a los retos de las ciudades contemporáneas,» Economía Industrial, n° 395, pp. 17-28, 2015.
- [7] J. Domingue, A. Galis, A. Gavras, T. Zahariadis, D. Lambert, F. Cleary, P. Daras, S. Krco, H. Müller, M.-S. Li, H. Schaffers, V. Lotz, F. Alvarez, B. Stiller, S. Karnouskos, S. Avesta y M. (. Nilsson, The Future Internet Future Internet Assembly 2011: Achievements and technological promises, Springer, 2011.
- [8] I. Kara, B. Y. Kara y M. K. Yetis, «https://cdn.intechopen.com/pdfs-wm/4579.pdf,» 1 Septiembre 2008. [En línea]. Available: <https://cdn.intechopen.com/pdfs-wm/4579.pdf>. [Último acceso: 24 10 2017].
- [9] B. Bullnheimer, R. Hartl y C. Strauss, «An improved Ant System algorithm for the Vehicle Routing Problem,» Annals of Operations Research, vol. 89, n° 0, pp. 319-328, 1999.
- [10] D. B. Johnson, «https://dl.acm.org/citation.cfm?id=321768,» Julio 1973. [En línea]. Available: [http://delivery.acm.org/10.1145/330000/321768/p385-johnson.pdf?ip=148.210.81.189&id=321768&acc=ACTIVE%20SERVICE&key=6F4CCF05E2930152%2E107F762BF030A7FE%2E4D4702B0C3E38B35%2E4D4702B0C3E38B35&CFID=998291177&CFTOKEN=83887922&\\_acm\\_=1508876511\\_cc6be9453b01d2](http://delivery.acm.org/10.1145/330000/321768/p385-johnson.pdf?ip=148.210.81.189&id=321768&acc=ACTIVE%20SERVICE&key=6F4CCF05E2930152%2E107F762BF030A7FE%2E4D4702B0C3E38B35%2E4D4702B0C3E38B35&CFID=998291177&CFTOKEN=83887922&_acm_=1508876511_cc6be9453b01d2). [Las access: October 24, 2017].
- [11] «https://www.ssucet.org/old/pluginfile.php/2121/mod\_resource/content/1/21-dijkstra.pdf,» 1990. [En línea]. Available: [https://www.ssucet.org/old/pluginfile.php/2121/mod\\_resource/content/1/21-dijkstra.pdf](https://www.ssucet.org/old/pluginfile.php/2121/mod_resource/content/1/21-dijkstra.pdf). [Último acceso: 25 10 2017].
- [12] E. T. Bray, «https://buildbot.tools.ietf.org/html/rfc7158,» Marzo 2013. [En línea]. Available: <https://buildbot.tools.ietf.org/html/rfc7158>. [Último acceso: 23 10 2017].
- [13] T. Bray, J. Paoli, C. M. Sperberg-McQueen, E. Maler, F. Yergeau y J. Cowan, «https://www.w3.org/TR/xml11/,» 29 Septiembre 2006. [En línea]. Available: <https://www.w3.org/TR/xml11/>. [Último acceso: 19 10 2017].
- [14] N. Nurzhan, P. Michael y R. Reynol, «https://pdfs.semanticscholar.org/8432/1e662b24363e032d680901627aa1bfd6088f.pdf,» 2009. [En línea]. Available: <https://pdfs.semanticscholar.org/8432/1e662b24363e032d680901627aa1bfd6088f.pdf>. [Último acceso: 23 10 2017].