FIRST LEGO LEAGUE 2021/2022

INNOVATION PROJECT

Selecting an optimal location for a logistic facility and finding an optimal path for delivery packages using artificial intelligence for small logistic companies in the city of Bogota

Group of participating students

Sara Sofía Bejarano Escobar
Alfredo Gómez Torres
Juan Camilo Contreras
Emilio Gómez Torres
Joseph Peñaranda

CIRAT
Club de Informática y Robótica Alan Turing
Tel: 3150550

Calle 57B # 37-21
Bogotá, Colombia

Table of Contents

1. Introduction

Our project is entitled: “Selecting an optimal location for a logistic facility and finding an optimal path for delivery packages using artificial intelligence for small logistic companies in the city of Bogota”.

The members of our team CIRAT live in Bogota, the capital of Colombia. Ours is a city with around 10.3 million habitants in the metropolitan area and It is located on a plateau in the Andes mountains at 2,600 meters above sea level. The density of the city is high with around 4,900 habitants/km^2. Those geographical and demographical characteristics, combined with the absence of efficient public transport, implies that Bogota has one of the worst traffic congestion in the world.

With more than 156 logistic companies working in the area of package delivery, the logistic sector has many small and medium size companies, most of them with low technological adoption.

Considering these characteristics of the city and the logistic market we have defined two problems: The first one consists in identifying the best place in Bogota in order to locate a package distribution logistics center (warehouse), and the second consists in establishing an optimal route for the distribution of packages in the city of Bogota. The international benchmark as well as the feedback of experts showed us the problems we selected are very relevant for those small and medium logistic companies as these technologies allow new kinds of business and services, reduces costs and improves competitiveness.

To solve the problems we follow a Design Thinking methodology, which comprises five big steps: Identify a problem, design a solution, share the solution with other people, improve the solution and prepare a presentation.

For the design of a solution we first improved our understanding of our city: how is it divided in zona planning units (UPZ), how is demography and land prices in those UPZ, how is the road infrastructure and the traffic. We converted this information through a quantitative approach in a set of two databases that we used as input for our proposed solution.

For the problem related with the location of an optimal place for the warehouse we used K-means, an artificial intelligence algorithm that regroupes the data in clusters, groups made with the properties that are put in for the user in the algorithm. Through our database, we were able to calculate 3 different scores: score for the package delivery (package demand), score for land price and total score (package demand + land price). It allows us to create a program in which users can select which of these 3 criterias to use and how many warehouses they want to locate. Through several simulations we got very good results as the algorithm suggestions for one warehouse are coincident with the expert opinion consulted.

For the problem of the optimal route for the distribution of packages we used Q-Learning, a method of machine learning based on reinforcement learning techniques. This kind of IA coding method can be synthesized correctly as the most optimal way to solve a determined

labyrinth, based on the scores that are given to all the parts of the paths that lead to the same result (or objective). Using this approach and the database we constructed to represent the streets of Bogota we built an algorithm that presented a solution for the optimal routing for the problem of package delivery. In our first approach we pretended to read real time traffic information from Azure platform. We got a successful connection with the platform, however during the development we were notified our license didn't let the lecture of this real time traffic information. This happened in the last week of our development so we were not able to solve it before the FLL CArgo Connect presentation.

2. Methodology

The following description presents the innovation methodology used to carry out this project.

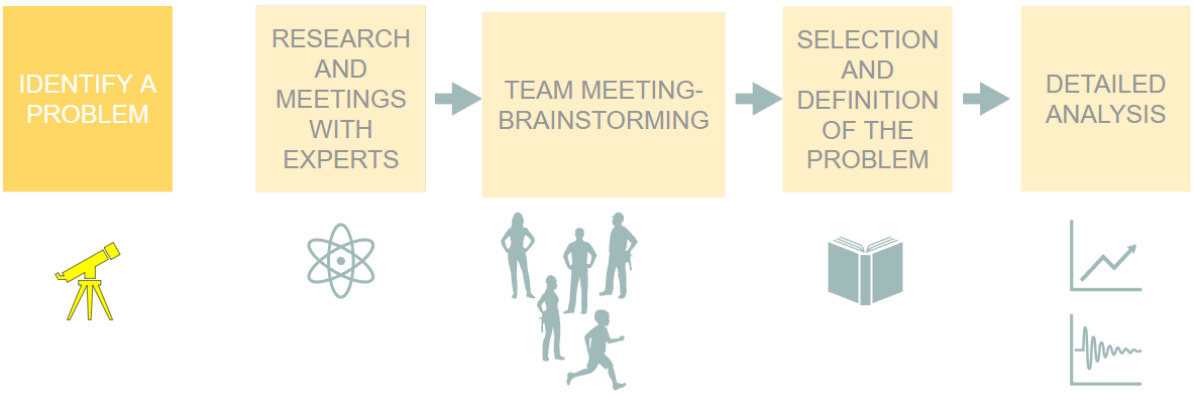
2.1 General description of the innovation methodology used.

Before starting our innovation project, we did a small research for the requirements of the competition in the subject of the project. For this, we focused on explaining what innovation is and how it can be innovated (this year the project is an innovation project). To understand the concept "innovate", we decided to do a research and we found that according to COLCIENCIAS innovation is the: "*introduction to the use of a product (good or service) or a process, new or significantly improved, or the introduction of a new marketing or organization method applied to business practices* ". Then we ask ourselves: How is innovation made? And we concluded that in order to innovate, you do not necessarily have to create a completely new solution to a given problem, you can also improve an existing solution.

Then we needed to define a methodology that would serve us for our innovation process. For this, we found *Design Thinking*, a method to generate innovative ideas that focuses its effectiveness on understanding and providing solutions. However, we decided to adapt this method to the needs of our team. For this, we define that our method would consist of 5 steps, which are presented below, graphically and explained in detail in the following sections.

2.2 First step: identify a problem

To carry out the first step of the methodology, which is the identification of the problem, we follow a series of our own stages that are presented graphically below.



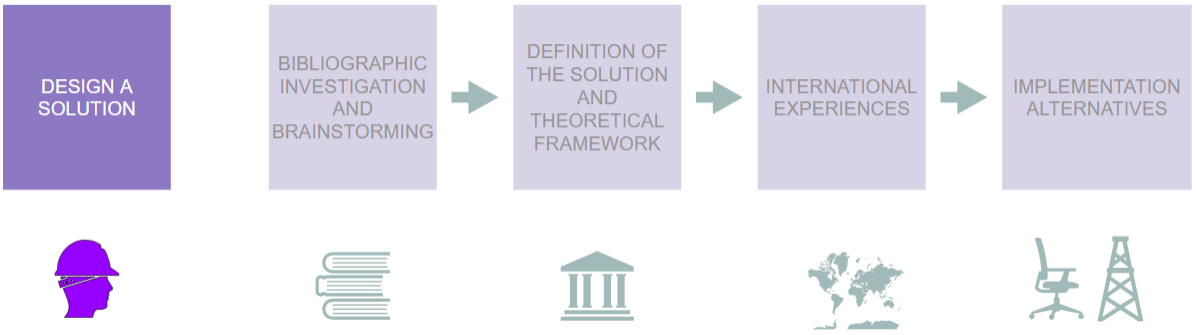
Source: Self made

- **Meetings with experts:** A general investigation of the subject and a series of meetings with experts were carried out that allowed us to build an overview of possible problems and solutions.
- **Team meeting and brainstorming:** Several meetings were held where all team members made presentations with specific proposals for possible problems.

● **Selection and definition of the problem:** Based on the proposals of the team members, through a vote, it was decided which was the general problem that we were interested in analyzing.
● **Detailed analysis:** The team investigated the problem in depth by searching and reading specialized bibliographic sources, especially detailed reports produced by universities and local government entities.

2.3 Second step: design a solution

To design the solution to our problem, we define four steps that are presented graphically below.



Source: Self made

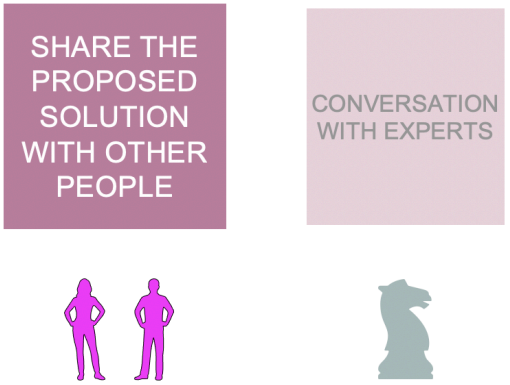
● **Bibliographic research, international benchmark and brainstorm:** The team read articles and saw videos. Since our solution would be an adaptation to the local reality of measures that have already been implemented in some cities of the world, we decided to explore how similar solutions have been implemented, to know their successes and difficulties.. From this work, several team members presented their proposals on how to solve the problem that had been identified in an innovative way.

● **Solution definition and theoretical framework:** The team had made a vote to define which of all the proposed solutions to the problem seemed the best. Next, the team began a process of theoretical deepening regarding the proposed problem, in order to understand the economic, social and technological foundations of the proposed solution.

● **Implementation alternatives:** The team investigated the most feasible way to implement the solution from a technological and economic point of view, so as to maximize its social impact and efficiency.

2.4 Third step: share the proposed solution with other people

To share the solution of the problem and see how viable it is, we decided to have 2 different perspectives: talk with experts and ask for their opinion to people who could become users of our innovation proposal.

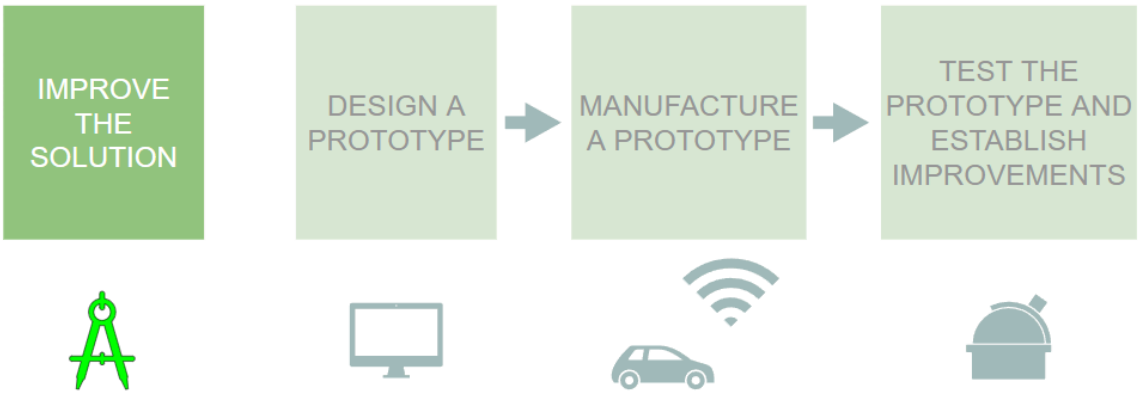


Source: Self made

● **Conversation with experts:** Conversations were developed with experts who gave us their criteria regarding the suitability and feasibility of the proposed solution, as well as their advice on how the proposed solution can be improved.

2.5 Fourth step: improve the solution

The team took into account this fundamental stage of the methodology. Three stages were defined to improve the solution; these can be seen better in the following graph:



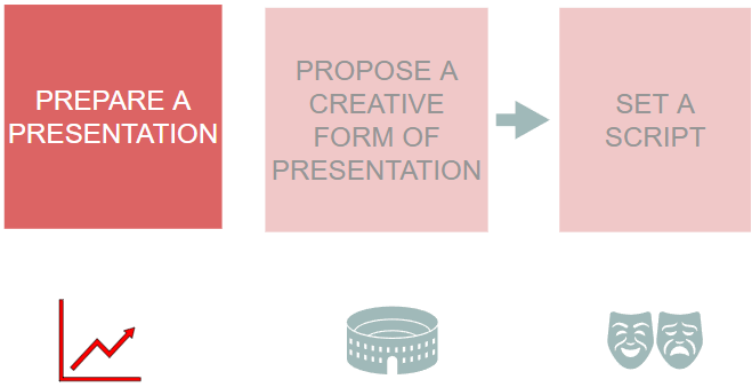
Source: Self made

● **Design a prototype:** At the beginning of this stage, one already has an idea of how the solution to our problem will be. To design the prototype, brainstorming is carried out and a way is found to make the prototype so that it develops the proposed solution in a technically and economically more optimal way.
● **Build a prototype:** During this stage, the ideas become reality and the prototype is built. For its manufacture, you must have all the materials and ideas must be well structured.
● **Test the prototype and establish improvements:** At this point, the prototype is tested to verify that everything is in place and well done. In case of problems, whether structural, functional or otherwise,

improvements must be made and a way to fix it be found so that it is not expensive, easy and simple, and does not take much time to execute.

2.6 Fifth step: Prepare a presentation

During this stage the team devised a way to present the innovation project in such a way that it is easy to understand, didactic and enjoyable to the public.

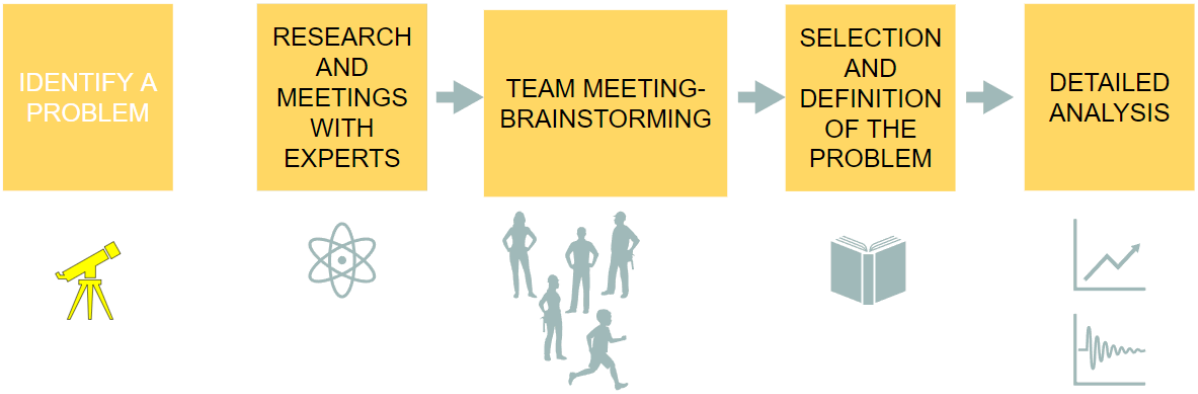


Source: Self made

- **Propose a creative form of presentation:** Different approaches were reviewed on how to present the problem and the solution, from those that are very formal, through a more academic presentation to others that were more playful. In the end we prefer to make a traditional academic presentation.

- **Establish a script:** Finally, once we selected our creative form of presentation, we designed a script that would explain the idea in a didactic and enjoyable way, so that we could communicate effectively what was the problem identified and what was the proposed solution, with its advantages and disadvantages. When establishing the script we took particular care to ensure that all members of the group could participate. Then we rehearse the script, prepare the props and have a lot of fun.

3. Problem identification

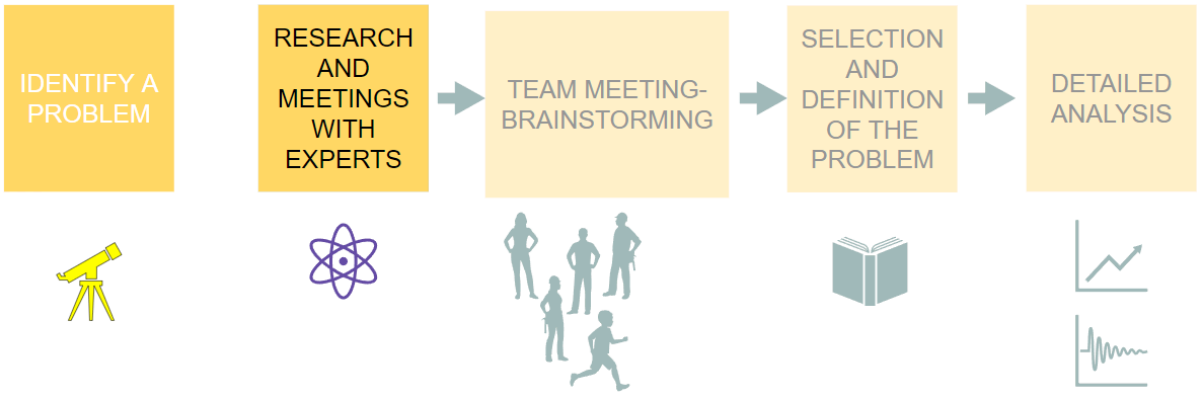


Source: Self made

As Chapter 2 explains, the methodology used to identify the problem consisted of four stages:

1. **Research and meetings with experts:** Before starting the problem identification process, we consider it necessary to have more information about cities, their creation, their operation and their importance. For this we use both primary and secondary sources. Our primary sources were seven experts who gave talks specialized in CIRAT. Our secondary sources were books and specialized articles we read.
2. **Team meeting - brainstorming:** Having read books and articles, and having had meetings with city experts, we believe that we already had the necessary knowledge to identify our problem. Then, the team met and we started making presentations of the ideas of each team member.
3. **Selection and definition of the problem:** After having listened to the ideas of all the members of the team, by means of direct voting we were defining the topics that interested us most until we reached the end of the identification of the problem.
4. **Detailed analysis:** Having already defined the problem we begin to analyze its relevance, both in the world and in our city (Bogotá). For this we deepen the two basic concepts of our problem: pollution and traffic.

3.1 Research and meetings with experts

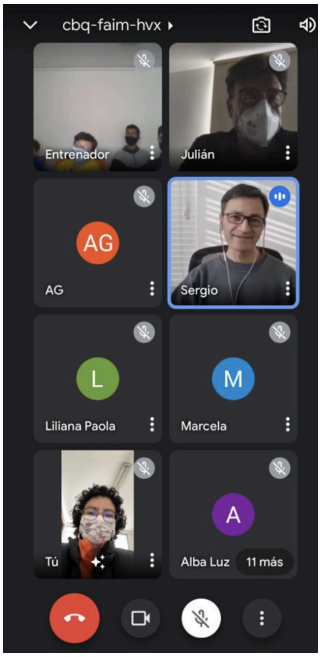


Source: Self Made

3.1.1 Meetings with experts

We had three meetings with experts in different aspects of logistics and we shared the presentations in real time using teleconferences with the other FLL teams in Colombia.

Sergio Torres, PhD in physics and fellow in Leidos, a US company, explained to us about new technologies for civilian air traffic management applied by the Federal Aviation Administration of the United States. He elaborated a speech about the complexity of the problem to handle thousands of planes in real time guaranteeing safety, respecting strict standards and at the same time optimizing paths so that the flights arrive according to their itinerary. And even more interesting, he explained to us in detail the science, technologies and software development behind that.

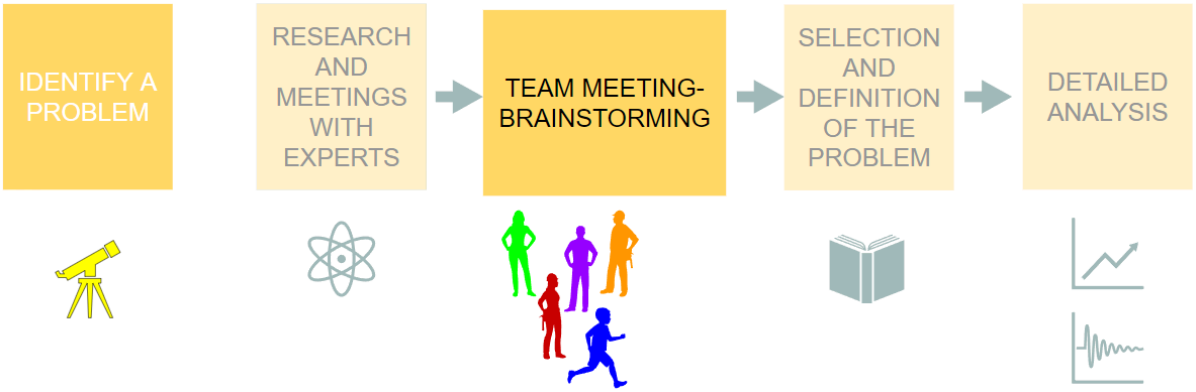


Daniel Villa, a logistic engineer working with DHL in Colombia explained to us that logistics is basically “Teamwork and coordination”. Subsequently, the importance of logistics, its usefulness, the way in which it is carried out, its future in the face of new technologies and needs of today's world, and international innovations facing common problems were discussed. Particularly noteworthy is the presence of robots capable of performing uncomplicated and repetitive tasks (bringing with it future job losses by the less-skilled/skilled population), climate impact and Big Data in terms of managing large inventories. And at the end, how a big multinational company such as DHL handles millions of packages in order to deliver all of them on time in so many different countries.

Daniel Garavito, an engineer and entrepreneur, talked about how to deliver high quality education using the technology to places which have difficulties in the access because they are located in places like isolated mountains or in the middle of the jungle which have many logistical problems as there are no roads or are very far away with not many people living there and because of that the transportation is very expensive and get teacher there is difficult too.

Germán Albornoz, the manager of a small logistic company explained to us how the customs works and what are the processes needed to import and export goods and merchandise in Colombia. There are a lot of legal aspects including free trade agreements, taxes and requirements, but also technology to handle and automatize part of the needed paperwork.

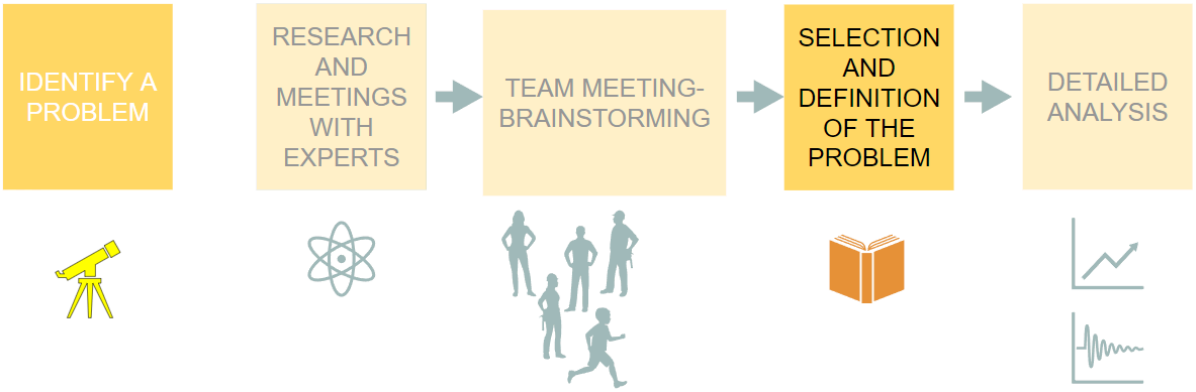
3.2 CIRAT team meeting – brainstorming



Source: Self made

After the meetings with the experts our team had several internal brainstorming meetings. Many ideas arised, for example: Regarding automation of the logistic process for the recollection of coffee on the slopes of the mountains in the Colombian coffee region; the solution of problems for small logistic companies in Bogota that don't have the technology needed to allocate their facilities or solve the optimal path to deliver their packages in the city; the possibility of improve the air traffic control for El Dorado airport in Bogota and work around how to create a robot to automate some basic problems for a small logistic companies in Bogota.

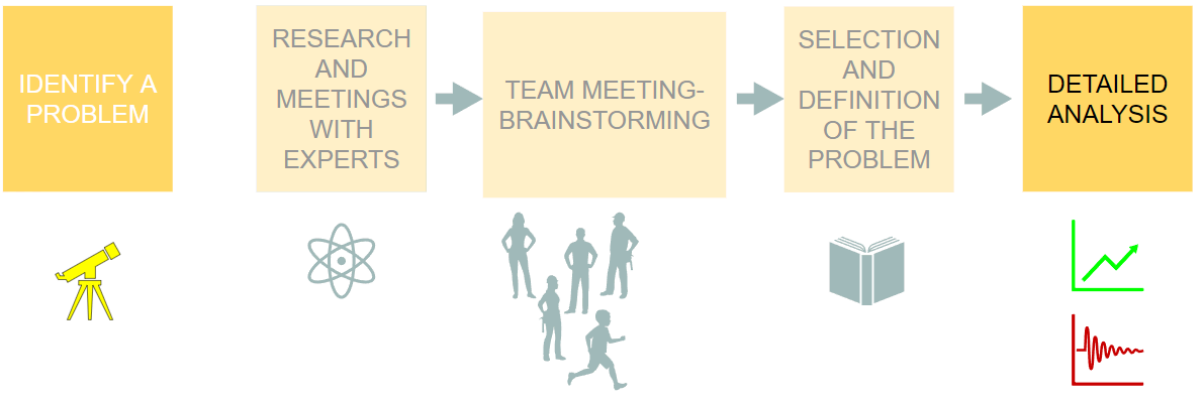
3.3 Selection and definition of the problem



Source: Self made

Based on previously submitted arguments, all the team members had a debate to decide which of the selected approaches was the best. At last, with slight changes, we select our problem: "To help small shipping companies to: (I) select an optimal location for a logistic facility distribution point in Bogota for a small logistic company. The place should be close to the potential customers but located in a not so expensive zone of the city; and (II) to help them to save money and time during their delivery travels while avoiding traffic".

3.4 Detailed analysis



Source: Self made

Bogota is a huge city, with 7.9 million habitants in the urban area but an expanded metropolitan area with around 10.3 million habitants. It is located in the Andes mountains at 2,600 meters above sea level. As it is located on a plateau in the mountains not too much space is available and therefore the density is high with around 4,900 habitants/km2. Those geographical characteristics, combined with the absence of efficient public transport implies Bogota has one of the worst traffic congestion in the world. According to the World Economic Forum a person in Bogota wastes 191 hours a year in traffic. In fact, according to Forbes, Bogotá was the most congested city in the world in 2020. Traffic jams are formed mostly on main roads and as a domino effect, secondary roads are also affected.

City	Country	2019
Bogota	Colombia	191 hours
Rio de Janeiro	Brazil	190 hours
Rome	Italy	166 hours
Paris	France	165 hours
Belo Horizonte	Brazil	160 hours
Mexico City	Mexico	158 hours
Dublin	Ireland	154 hours
Istanbul	Turkey	153 hours
Sao Paulo	Brazil	152 hours
Saint Petersburg	Russia	151 hours
Jakarta	Indonesia	150 hours
London	United Kingdom	149 hours
Boston	United States	149 hours
Chicago	United States	145 hours
Quito	Ecuador	144 hours
Philadelphia	United States	142 hours
New York City	United States	140 hours
Brussels	Belgium	140 hours

Quartz | qz.com | Data: INRIX Research

Source:Forbes

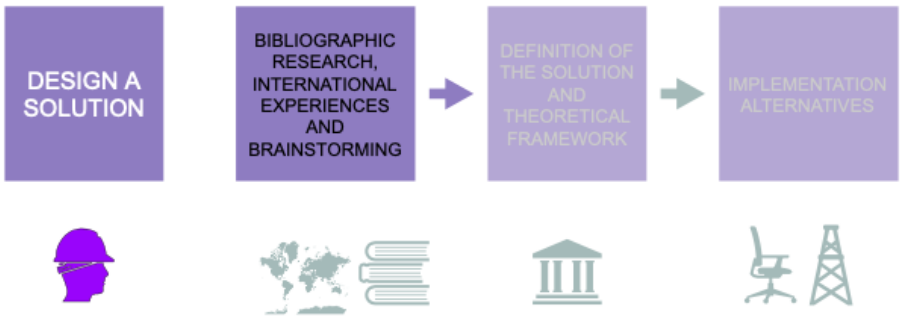
We have defined two problems: The first one consists in identifying the best place in Bogota in order to locate a package distribution logistics center, and the second consists in establishing an optimal route for the distribution of packages in the city of Bogota.

These two problems are relevant because the small and medium-sized logistics enterprises have not solved them yet, unlike large logistics companies around the world. According to the quarterly newsletter of the Colombian postal sector, which is published by the Ministry of Information and Communication Technologies (MICT): “The total number of operators that provide courier express service for the third quarter of 2019 was 156, of which 125 made shipments within the national scope, 118 operators locally and 62 at international level”. Consequently, there are more than one hundred small and medium-sized enterprises in the Colombian market and most of them operate in Bogota.

According to the research carried out and the meeting with experts these small and medium-sized enterprises dont have appropriate technological tools to carry out optimal solutions for both problems.

4. Initial solution design

4.1 Bibliographic research, international experiences and brainstorming



Source: Self made

Bibliographical Research and international experiences

The Latin American Development Bank has indicated the importance of the digitization for the logistic sector (CAF 2021) as these technologies allow the different actors involved in the value chain of this sector to detect and identify patterns that improve the efficiency of the different processes carried out in the different links of the chain. It also enables new kinds of business and services, reduces costs and improves competitiveness.

We searched for experiences in developed countries with problems similar to our approach and we found an application of IoT and Big Data to optimize package delivery made by DHL in India (DHL, 2018)

With 1.4 billion inhabitants, India is the second most populated country in the world with an area of 3,287,263 km².

The e-commerce sector has experienced exponential growth in recent years due, among other reasons, to the accelerated growth of Internet integration in populated areas, which currently stands at 50%. In the e-commerce supply chain, the last part generally starts at a warehouse or distribution center and ends at the end user's premises; this last step is called the "last mile" and optimizing this operation is essential in the rapid development of the e-commerce industry in the country.

DHL, which is the world's largest logistics company with a presence in more than 220 countries, implemented in 2018 the SmartTruck solution, previously developed and tested in Germany, which combines the Internet of Things (IoT) with Big Data to optimize the "last mile" operation. Dynamic delivery/pickup routes include the initial daily definition of an optimal route where the delivery/pickup sequence of the goods is defined.

Once on the road, and according to dynamic traffic conditions, weather, or customer availability, the route is updated in real-time, which is received by the driver through the

onboard navigation system. The integration of the Internet of Things, which facilitates real-time data collection, with data analysis through Big Data techniques, allows the development of solutions that optimize the operation of the last link in the supply chain related to the delivery of goods, known as the "last kilometer".

In general terms, an intelligent solution for the delivery of goods, such as the one implemented by DHL in India, allows improving several operational indicators, among which are: delivery time, asset utilization, kilometers traveled, travel speed, fuel consumption, CO2 emissions, fleet life, among others, which translate, for the logistics operator in cost reduction, and customers in a higher degree of satisfaction.

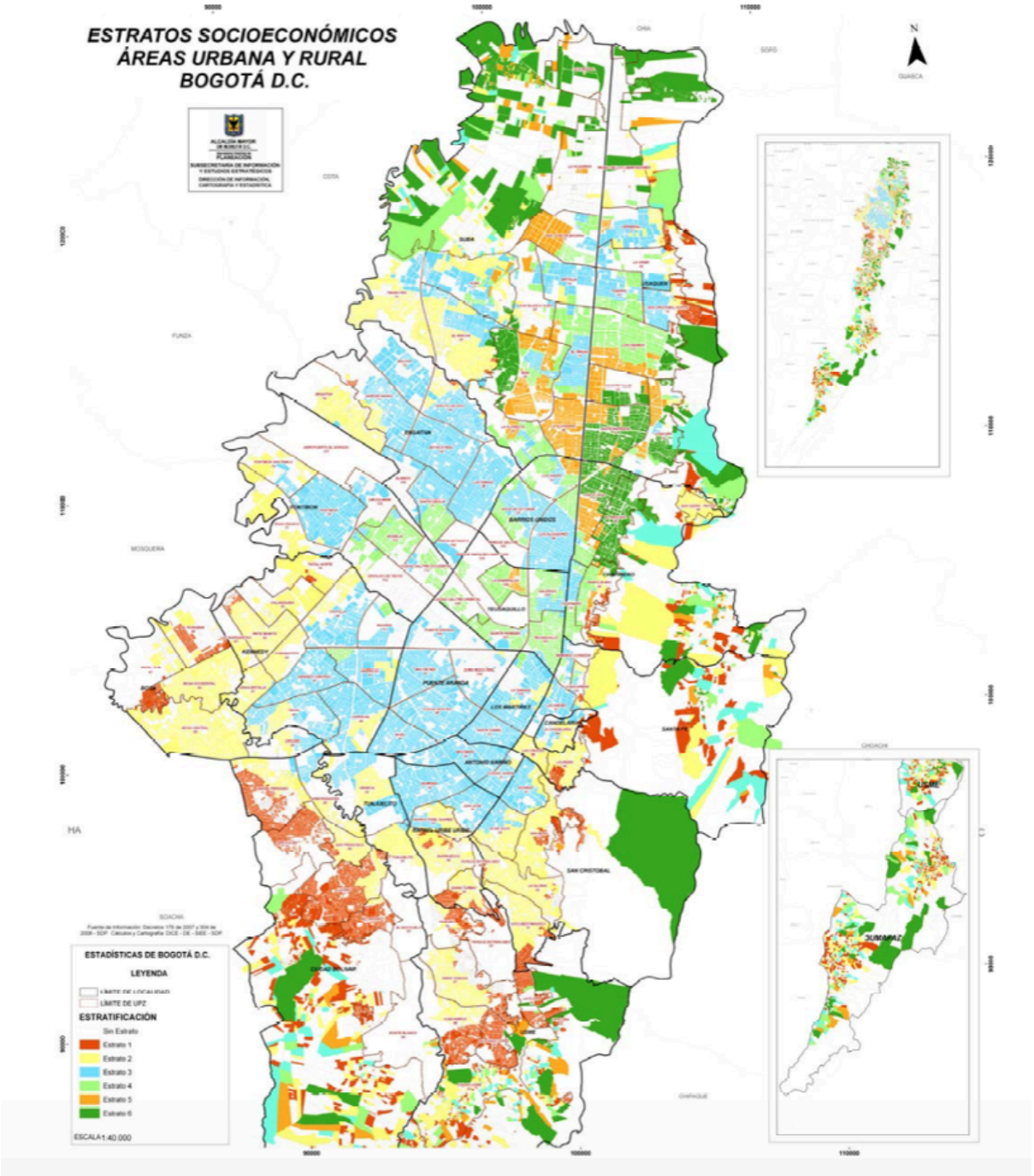
Optimizing the "last kilometer" not only lowers the logistics operator's costs but also improves the quality of service and therefore end-user satisfaction. The accelerated development of one industry, in turn, drives the development of another industry. In this case, the rapid adoption of e-commerce in India has driven the implementation of solutions in the logistics sector, as the logistics sector is an enabler of the e-commerce sector.

After these findings, we were convinced that the problems we want to solve are very relevant. Then we start with a brainstorm session to identify possible solutions.

Brainstorming for the best location for a logistic distribution center considering the local conditions of Bogotá

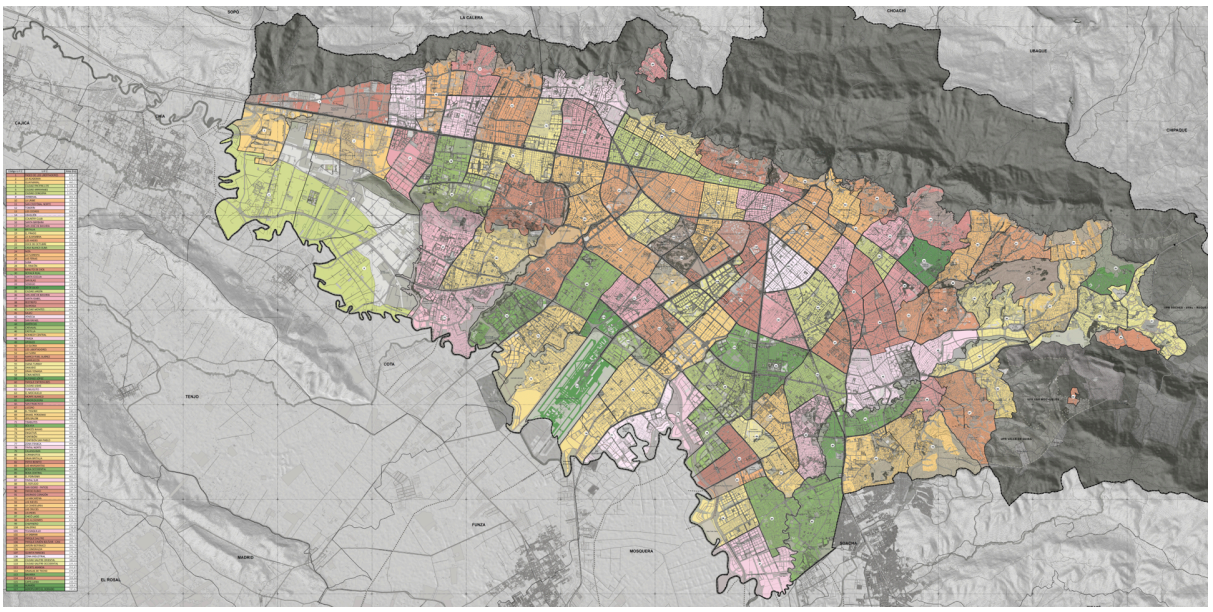
Regarding the first problem, during the brainstorm the team concluded the best location for a logistics distribution center requires the evaluation of two factors which are: The price of the land where its location should be relatively cheap and that the distribution center should be near the area where there is more demand for this service.

We decided we needed to better understand our city in order to identify those characteristics. So we consulted the official information of the District Planning Secretariat of Bogotá and we found that the geographic area of Bogotá is stratified. Each stratum (1 to 6) represents the relative economic capacities of the different areas of the city, with 1 being those with the least capacity and 6 being those with the greatest capacity. As can be seen in the following map, stratum 5 and 6 tend to be located in the northern and northeastern part of the city. The higher the stratum, the higher the price of the land.



Source: District Planning Secretariat of Bogotá

We also found the city of Bogota is divided into 20 localities, and each locality is divided into zonal planning units (UPZ). There are 112 UPZ in the city as is shown in the map below. We found there is good statistical information regarding each UPZ so we decided to use those UPZ as our minimum geographical unit of analysis.



Source: District Planning Secretariat of Bogotá

For each UPZ we found information regarding how many people of each stratum lived there. We also define a geographical location (latitude and longitude) for each UPZ approximating a description of its geographic center, so we built a table with this information for each UPZ as shown below.

UPZ y Localidad	Sin estrato *	1. Bajo - bajo	2. Bajo	3. Medio - bajo	4. Medio	5. Medio - Alto	6. Alto	Latitud	Longitud
1 Paseo de los Libertadores	198	10	2.587	0	0	0	702	4.796	-74.034
2 La Academia	221	0	8	0	0	334	310	4.786	-74.049
3 Guaymaral	14	0	0	0	2	0	152	4.814	-74.066
9 Verbenal	78	5.319	21.935	62.885	3.854	5.288	0	4.765	-74.038
10 La Uribe	135	7	543	12.136	4.566	0	0	4.752	-74.045
11 San Cristóbal Norte	4.482	12.596	5.036	51.088	0	0	131	4.734	-74.017
12 Toberín	159	0	565	16.955	34.597	0	0	4.747	-74.047
13 Los Cedros	328	114	0	0	81.283	17.760	3.312	4.724	-74.035
14 Usaquén	1.121	0	2.317	1.917	4.609	17.140	14.974	4.695	-74.030

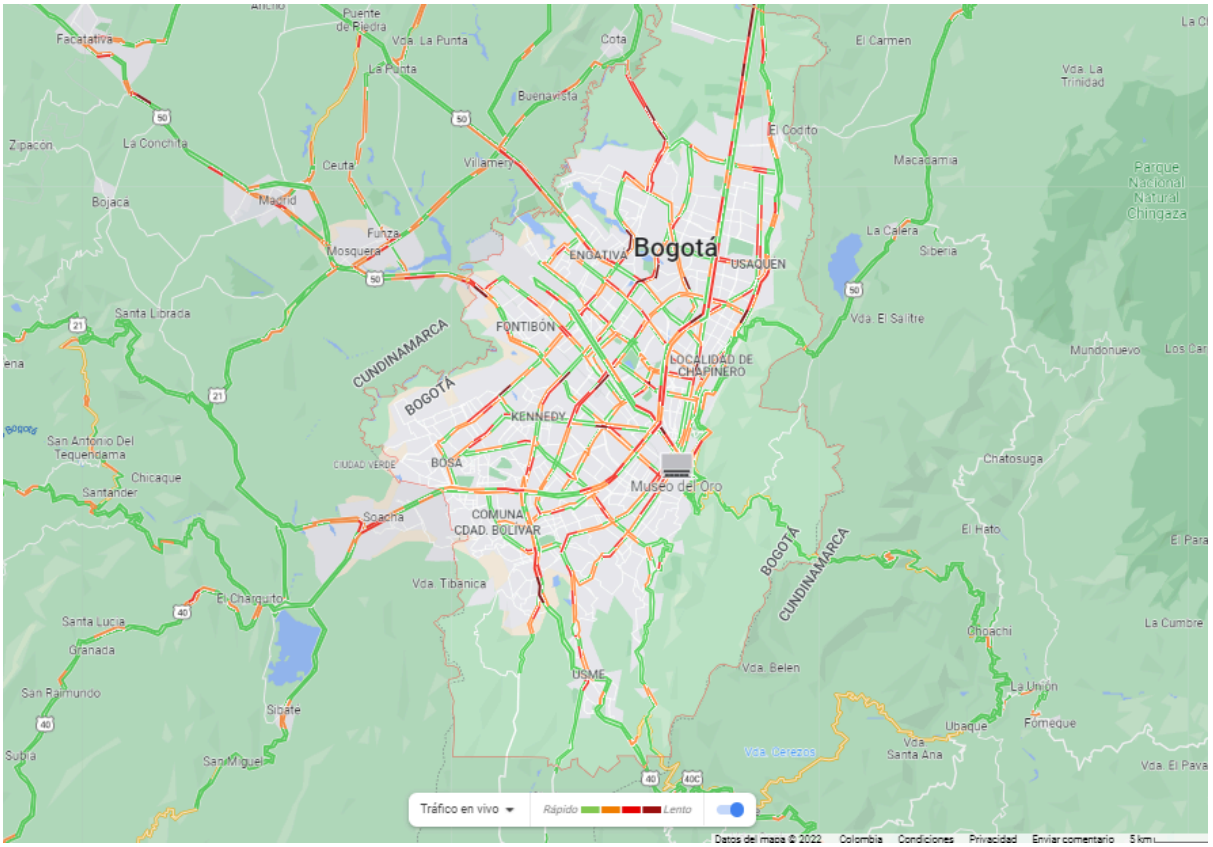
Source: Own elaboration from information of District Planning Secretariat of Bogotá

However, we have not found information regarding the detailed geographic location of the package demand in Bogotá as only each logistic company really knows how it is. Therefore in a real world application the company will provide us with the geographical demand. But for this exercise, we decided to use a combination of socioeconomic information (stratum) and population density as a criterion for package demand. The rationality for this approach is because the package demand is greater in the areas of Bogotá where people with greater economic resources are concentrated, but also the denser an area is, the greater number of people per square kilometer lives into it. Then both criteria are important.

Brainstorming for the optimal path for package distribution considering the local conditions of Bogotá

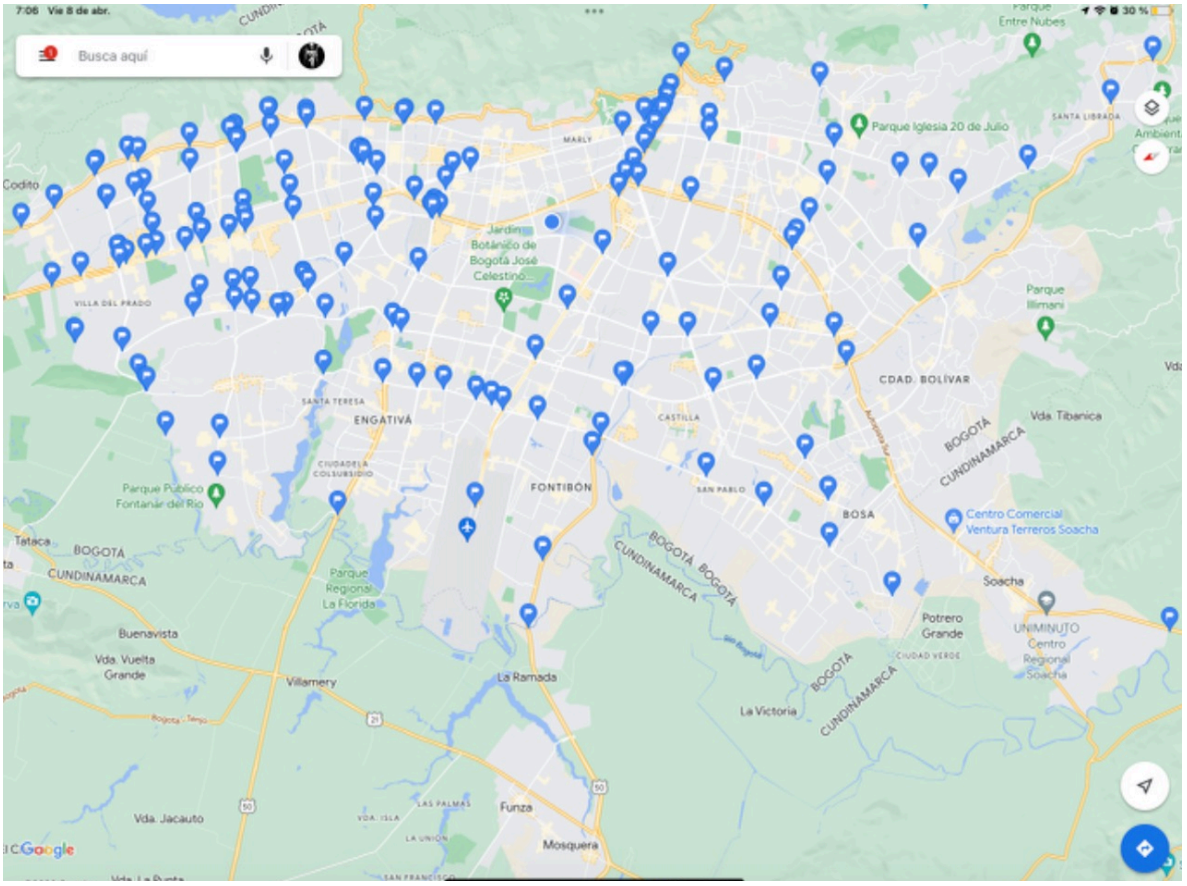
We also need to better understand the traffic behavior of the city. Traffic is variable: it depends on the day of the week, the hour of the day, the month of the year, and even

unpredictable facts such as the wheather or an accident in the street. It means we need to know the traffic in real time just at the moment a decision regarding the order of delivery of packages is going to be made. Just for illustration, this is a map of the main roads of the city on a common working day.



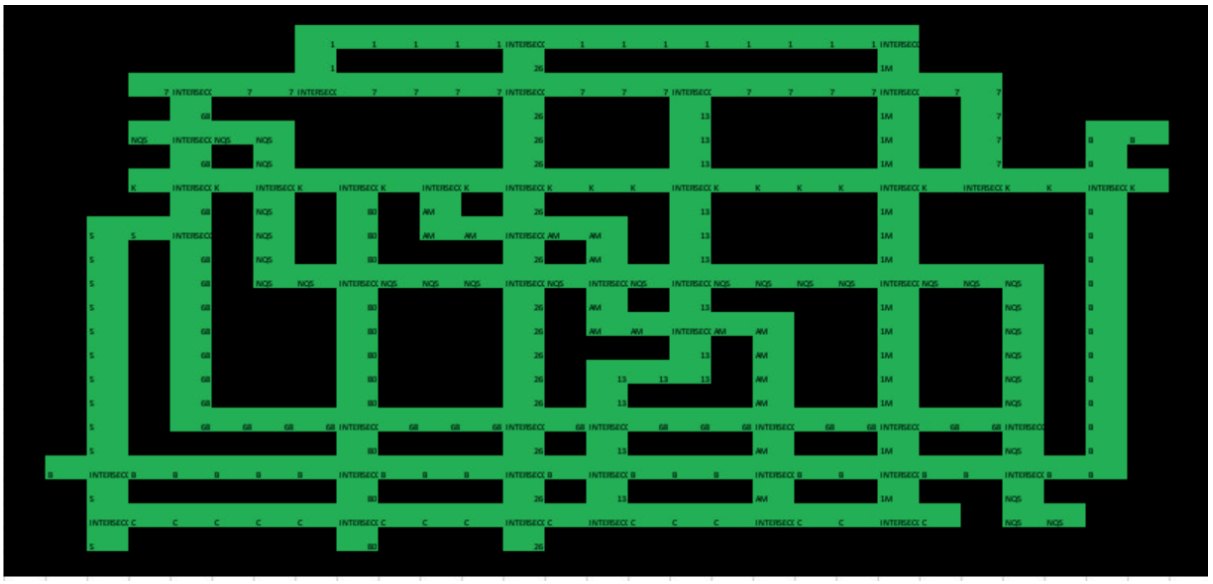
Source: Google maps

During the brainstorming we realized that it was essential to have real time information of the traffic but as Bogota is too big, we concluded we need a geographical representation of the streets of the city. Not of each street but the most important. Therefore we worked an exercise to identify those key streets in order to get that representation. The results of our exercise were a set of points with geographical location on a map as shown below.



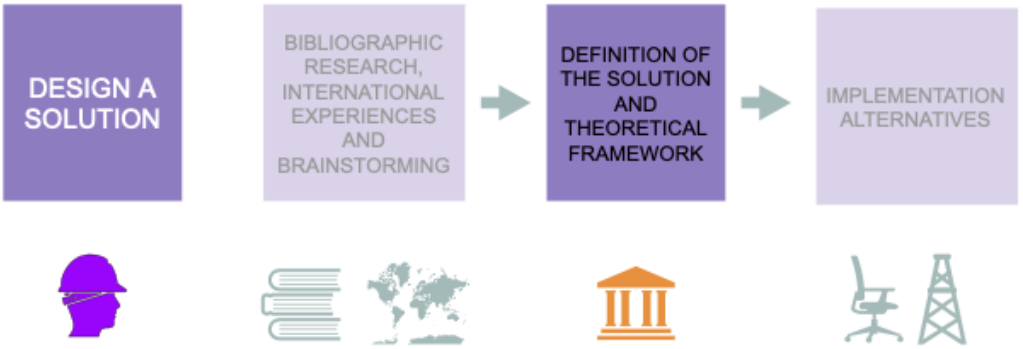
Source: Own elaboration with information of Google Maps

Then we use our points in the map to create our own representation of the main roads of Bogotá. Our results are shown below. The numbers and the names in the geographical representation are to describe the streets (i.e. 7th avenue, NQS, 80th street etc). Our idea was to search for a traffic real time provider in order to read key indicators of the traffic (i.e. average speed) only in those specific points.



Source: Own elaboration

4.2 Solution definition and theoretical framework



Source: Self made

To solve our problem, we decided to use AI (Artificial Intelligence), using k-means (that uses clusters) to identify the best places to put up a delivery company service point, and Q-learning to determine the best route in a city. We can't use equations or simple methods to solve this problem, since the city is always in constant change (traffic, routes, etc.) and taking advantage of the fact that Q-learning it is an artificial intelligence that uses rewards and punishments to determine the best solution to a possible problem, we will use it to identify the route.

How to decide the best route and the best place to put a service point?

With the k-means algorithm we can calculate exactly where the best place will be to put a service point. This algorithm groups different sets of data into Clusters that define a subset where the data best fit, and like that we can identify the best cluster for the location of a logistic facility, for example a warehouse to be defined as a service point. We will have to define some evaluation criteria, like: The population density, the cost of the land and the demand for package delivery. Our scope is to be sure that the little delivery companies choose an optimal place that they can afford.

With Q-learning we will use the rewards and punishments learning technique to teach the AI how to choose the best routes to deliver a set of packages by a truck during a single delivery trip. In this case we also have some evaluation criteria: The traffic, the distance, the time for delivery and unforeseen setbacks (landslides, road failures, etc.).

It's also important to mention that these IA algorithms are made to cope and adapt random and dynamic situations, so they are the best algorithms to make this solution

What is K-means?

K-means is an artificial intelligence algorithm that regroupes the data in clusters, groups made with the properties that are put in for the user in the algorithm. K-means creates the most precise and accurate groups, that we can use for our investigation.

What is a supervised and unsupervised algorithm

A supervised algorithm is an IA algorithm that has a reference point, for example, a goal that has to be done, and a punishment if it doesn't meet the goal, a very good example of a supervised algorithm is Q-learning, because it learns doing a lot of attempts to solve a situation/problem (for example a game) and trying to learn of every attempt to make it better in the next one.

An unsupervised algorithm is an algorithm that DOESN'T have a reference point, thus if the program does something, it has to be well done simply because the program is programmed so that this action is perfect, and it doesn't have to learn to do it, for example K-means. K-means just regroup some information, in clusters, that is put by the user in the program.

What is a Cluster?

A cluster is a group that is done based on the detection of "Points" that are observations/data, which are found in different areas. This program uses algorithms that detect patterns in spatial location based on the number of properties that are in the program, and in this way clusters are created.

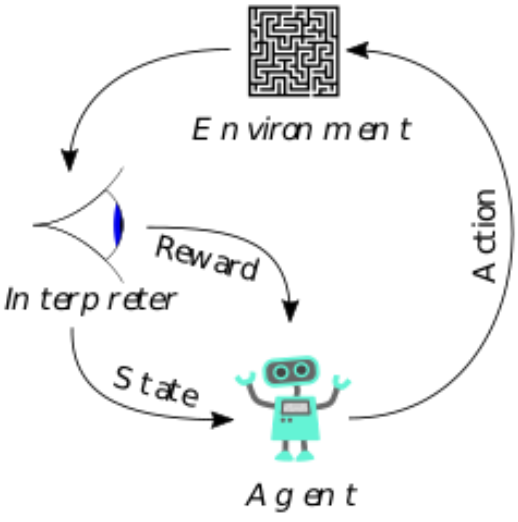
What are the properties and the observations?

A property its the data that its put by the user for creating clusters, and the observations are the criterias that are out for the user, this criteria is traduced by the program to create dimensions like X, Y, Z, but in this case with the criterias: The population density, the cost of the land and package delivery demand.

What is Q-Learning?

Q-Learning is a method of machine learning based on reinforcement learning techniques. This kind of IA coding method can be synthesize correctly as the most optimal way to solve a determined labyrinth, based on the scores that are given to all the parts of the paths that lead to the same result (or objective); in other words, given the current state in which the agent currently is, it chooses the optimal action that will maximize the long-term expected reward provided by the environment.

However, to understand correctly how it works, is fundamental to fully understand the following terms:



1. Agent: the abstract kind of character that does all the actions and interacts with the environment in which it is.
2. Environment: the world in which the agent acts and interacts with. When the agent takes a decision and by making an action interacts with the environment, the environment returns to a new state, having an impact again in the future agent choices.
3. Action: what the agent can do in a determined state. Condition that, another again, would lead to a different reward and a different state.
4. State: the condition or situation in which the agent is. It is not unique, since it can change as the agent makes actions andenters in different states.

5. Reward: what the agent earns with every action and in every state.

Additionally, it is important to mention that this particular coding method has two different approaches: a deterministic one, in which all the actions are done, simply because they are the most optimal ones based on the rewards the agent would have and the state it is; and a no deterministic one in which probabilities and statistic also affects the actions of the agent. This last option is mostly used in video games, but obviously it would not be taken into consideration in this project.

With all the above, understanding the Q-Learning coding shouldn't be too difficult. However, there is a fundamental factor that wasn't openly said before: this method is what is called an iterative method. That is to say, that the program (agent) doesn't know the best way to solve a problem at first, but only after many attempts, mistakes and negative values can it arrive at the ideal result. Consequently, while solving the problem, the states and rewards would change constantly.

Finally, in the environment there is always a defined terminal state, which means that if the agent arrives at that precise situation the simulation finishes and the agent has to restart all the process (with the values that it had given to the states in which it was along the way…)

What is dynamic programming?

Technically, dynamic programming is defined as a class of algorithms which seek to simplify a complex problem(s) by breaking it up into sub-problems that are solved recursively; using a function that calls itself numerous times. This concept is the base of Q-Learning's equation/functioning and was introduced by Dr Richard Bellman in 1954. In fact, without Bellman's equation Q-Learning itself wouldn't be possible.

How does it works?

The main purpose of Bellman's equation is to calculate the value of a certain state, depending on whether this state is closer or further from reaching a specific goal. It is a recursive equation, because a specific value (V(s)) depends on the previous states (V(s')).

$$V(s) = max_a(R(s,a) + \gamma V(s'))$$

Where:
$V(s)$: The value of the actual state.
$V(s')$: The value of the next state.
$R(s,a)$: The reward of a specific action, executed from the actual state.
$max_a()$: The maximum function. It selects the highest value, out of all possible options.

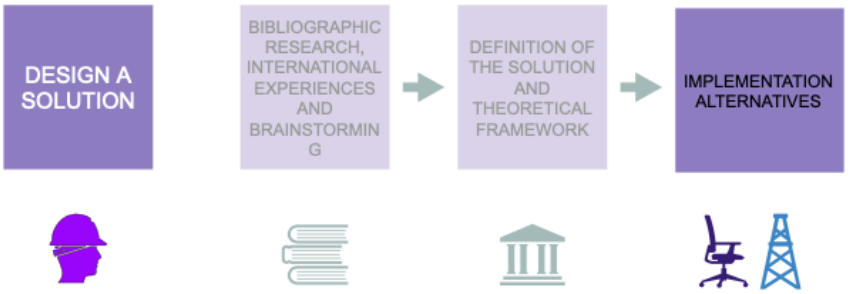
This equation allows us to describe how close a state is from the final target state. We can add some other features in order to consider stochastic elements in the environment. And we can prioritize the best action from a state. Q-learning final equation takes this characteristics into account:

$$Q(s,a) = R(s,a) + \gamma \sum_{s'}(P(s,a,s')\, max_{a'} Q(s',a'))$$

Where:
$Q(s,a)$: Reflects how good an action is executed from the actual state.
$Q(s',a')$: Is the Q value of the actions of the near states.

$P(s, a, s')$: Is the stochastic element, and adds some random behavior to the agent (this could be removed if the agent is deterministic).
$R(s, a)$: The reward of a specific action, executed from the actual state.
$max_a()$: The maximum function. It selects the highest value, out of all possible options.

4.4 Alternatives for the implementation



Source: Self made

For the implementation alternatives, two ideas that were developed using the Python programming language were thought of, these two ideas were K-means and Q-learning as explained previously.

For both K-means and Q-learning, the software Anaconda was used to manage environments, Jupyter for the programming environment, and Github to carry out collaborative work through a repository.

Alternative implementation for K-Means

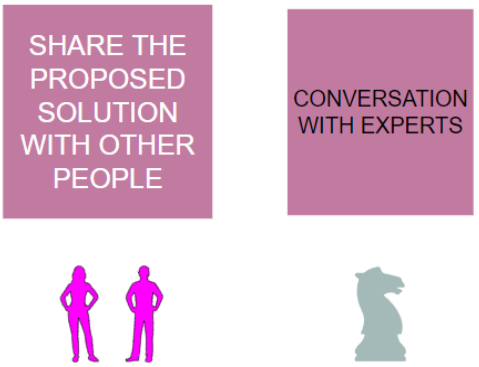
As mentioned above, the programming language for the implementation of K-means was Python; the libraries that we used in this programming language were Pandas, Random, Numpy, SKlearn and Matplotlib.

Pandas was used to import the CSV database, Random to generate some random algorithms, Numpy for array and multidimensional array, Matplotlib to create graphs and through one of the SKlearn modules, the artificial intelligence algorithm was included to execute K-means. It should be noted that all these libraries are open source and free software libraries.

Alternative implementation for Q Learning

The tools used for this section of the software were very similar to those needed for K-Means since it was also necessary to import the logical matrix of the routes in Bogota to the program and create and edit various arrays. However, a graphical interface was added, via the Pygame library. In addition, for the "display" of some actions, it was necessary to use "delays", for which the Time library was added.

5. Socialization of the proposed solution



Source: Self made

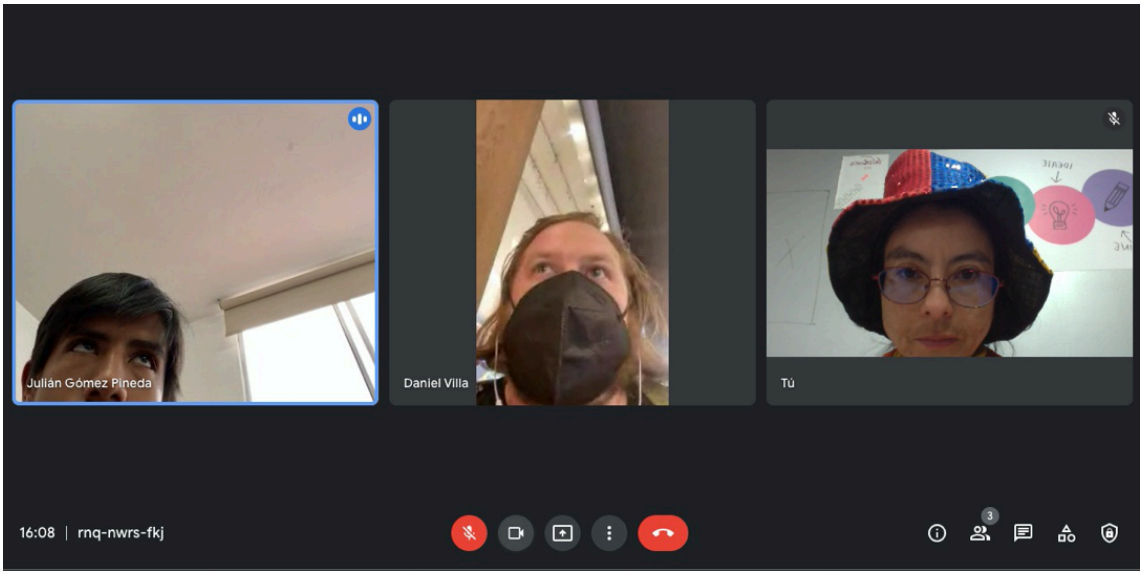
5.1 Conversation with experts

We prepared a set of questions for the conversation with experts. However

We prepare this questions to our expert:

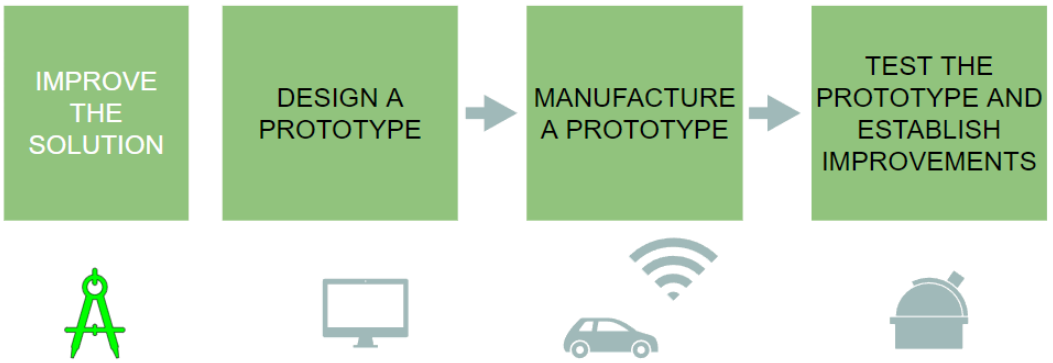
1. What parameters do they use to determine the optimal routes for delivery truck trips in Bogota?
2. What is the best place to set up a package delivery center in Bogotá?
3. What other uses or scope do you think this system could have?
4. Do you have any recommendations or corrections about our project?
5. Do you think there is demand between the logistic companies in Bogota for a software as we developed for the service in a city?

Then we have a teleconference with Daniel Villa the first expert we contacted for this innovation project, we explained him the two problems we selected and the focus of our solution and these were his answers to our questions:



1. Daniel explained to us in the DHL operation the city of Bogota is divided in loops and the packages are divided according to its geographical location on these loops. Then the package assignment is made truck by truck according to the loops each truck covers. However the programming of a route is a process that still has manual steps as there is complexity with the address provided by the final customer for Bogotá. It needs a human being making validation and then the routing is made.
2. Most of the hubs of DHL are close to the airport as the business is mostly for international transport. If it is a domestic company, then in his opinion the warehouse should be close to the points where the industry is located and safety is also very important. A local company should be possibly close to Puente Aranda locality.
3. For routing, to ensure the way the addressing is completely parameterizable is key as it needs taking into account problems with the way people write addresses. It is also important how these software tools can make measurements useful for the logistic companies in terms of productivity.
4. Artificial Intelligence makes a suggestion but the final decision should be made by a person. The route is a suggestion but the human beings should be the last word. Is the same with the proposed location for a warehouse: the tools make suggestions but the humans make the final decision. Then the raised issues are pertinent and the solution adopted is very good as a tool to improve the decision process.
6. Even for a company as DHL the routing solving problem is a Challenge in a city as Bogota. Both problems CIRAT is solving are very important for logistic companies, even more for small logistic companies with low technology adoption. To have these kinds of tools available in a logistic company determines the competitive advantages and improves the operational efficiency. In fact these kinds of solutions are very important because the two selected problems are vital problems for a logistic company.

6. Improving the solution: Development and general operation of the prototype



Source: Self made.

6.1 Design a prototype

K-means

It was thought to make an AI that could identify and classify the areas of Bogota based on a database, in such a way that it had a kind of map, which not only takes into account the coordinates, but also the population by strata of each place, the price of the land, the demand for package delivery among others. As explained before, Bogota is divided into localities, which in turn are divided into zonal planning units (UPZ), and thanks to the fact that each of these UPZ has a count of inhabitants per km² and inhabitants per strata division, it was much easier for us to prepare the database.

Therefore, using this database, we were able to calculate 3 different scores: score for the package delivery (package demand), score for land price and total score. Using these data, we can identify the best points for each UPZ.

Through all this data, we can make the AI identify the clusters. In order to provide flexibility in the programming, the number of desired clusters can be selected by the user at will, and in this way the AI can identify the number of centroids according to the scores of the selected criteria, which show us the best places (geographical coordinates) for the location of the logistic distribution center (i.e. warehouse) in Bogotá.

Q-Learning

The second step of our solution was to develop a program that indicates the user the optimal route to deliver a given number of packages.

First, we reduced the complex road structure of Bogotá into a logic matrix, so it simplifies the management of the data. For this part, we use only the main avenues of the city, in order to simplify the steps of the algorithm. However, adding new rows or columns to the bidimensional array, the program could consider in the future additional streets and paths.

Each cell in the matrix would have a reward score, depending on the real traffic flow in the respective road. To access the traffic information, we consider using an API of Microsoft Azure that provides the user specific information about the vehicular mobility of a provided coordinate in a city. Also, the API allows the user to request the information through a command line in any programming language (using the correct libraries) and download it in real time in a *json* file. The data update time in the API is 1.1 seconds.

6.2 Creation of the prototype

K-means

As a first step, the libraries that will be needed in the project are imported, and the table that was made in Excel is called by the code which contains the necessary data to make the clusters, and the table is inserted "physically" in the code (lines 1, 2 and 3):

```
import pandas as pd
import random
```

```
In [2]: import numpy as np
        from sklearn.cluster import KMeans
        import matplotlib.pyplot as plt
        import matplotlib.image as img
```

```
In [3]: plt.rcParams['figure.figsize']=(16,9)
        plt.style.use('ggplot')
        dataframe = pd.read_csv(r"Base de Datos/Hoja de cálculo UPZ y localidaes.csv", sep = ';')
        dataframe.head()
```

Out[3]:

		Sin estrato *	1. Bajo - bajo	2. Bajo	3. Medio - bajo	4. Medio	5. Medio - Alto	6. Alto	Latitud	Longitud	...	Estrato 3 a\u00f1o 2018	Estrato 4 A\u00f1o 2018	Estrato 5 a\u00f1o 2018	Estrato 6 A\u00f1o 2018	Paquetes por UPZ	Precio promedio por m\u00f1 por UPZ	Paquetes
0	1 Paseo de los Libertadores	198.0	10.000	2.587	0.000	0.000	0.000	702.0	4.796	-74.034	...	0	0	0	504	13283	2775093	0,0
1	2 La Academia	221.0	0.000	8.000	0.000	0.000	334.000	310.0	4.786	-74.049	...	0	0	197	183	4867	5211913	0,0
2	3 Guaymaral	14.0	0.000	0.000	0.000	2.000	0.000	152.0	4.814	-74.066	...	0	24	0	1835	27837	7369048	0,0
3	9 Verbenal	78.0	5.319	21.935	62.885	3.854	5.288	0.0	4.765	-74.038	...	71191	4363	5986	0	522606	2099151	0,2
4	10 La Uribe	135.0	7.000	543.000	12.136	4.566	0.000	0.0	4.752	-74.045	...	21509	8092	0	0	159236	2501438	0,0

After this, the user is asked to choose the number of clusters and with which one of the three previously mentioned possibilities of scores he wants the clusters to be made. After this, the commands are initialized so that K-means is executed and as a result it defines the clusters (in4, in5, in6):

```
In [4]: Seleccion = int(input("Señor usuario, porfavor ingrese el numero 1 si desea escoger los centros de envío solo teniendo
        if(Seleccion == 1):
            dataframe2 = dataframe[['Latitud',"Longitud","Puntaje paqueteria"]].to_numpy()
        elif(Seleccion == 2):
            dataframe2 = dataframe[['Latitud',"Longitud","Puntaje precio"]].to_numpy()
        elif(Seleccion == 3):
            dataframe2 = dataframe[['Latitud',"Longitud","Puntaje total"]].to_numpy()

        Señor usuario, porfavor ingrese el numero 1 si desea escoger los centros de envío solo teniendo en cuenta la cantidad
        de paquetes de la zona, 2 para tener en cueta solo el precio de arrendamiento de la zona y 3 si quiere tener en cuent
        a ambos factores: 2

In [5]: matriz = []
        for i in range(112):
            fugaz = [0.0,0.0]
            for j in range(3):
                if(j == 2):
                    for b in range(int(dataframe2[i][j])):
                        matriz.append(fugaz)
                else:
                    fugaz[j] = float(dataframe2[i][j])

In [6]: numero_clusters = int(input("Ingrese el numero de centros de envío que desea instalar en la ciudad: "))
        kmeans = KMeans(n_clusters = numero_clusters, random_state = 0).fit(matriz)
        kmeans.labels_
        kmeans.cluster_centers_
        centros = kmeans.cluster_centers_

        Ingrese el numero de centros de envío que desea instalar en la ciudad: 8
```

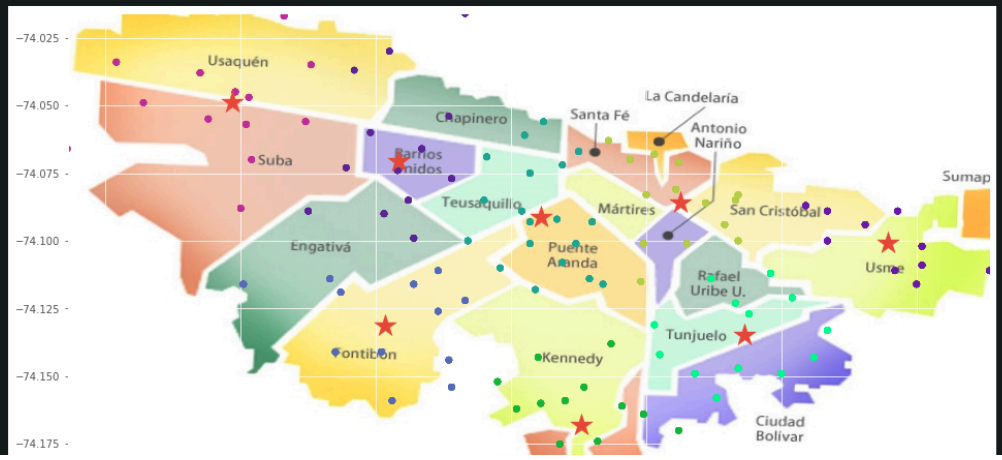
Finally, the code is written so that the AI can define the clusters. The output of the program is shown: The map appears in the screen as well as the clusters, as red stars icons (in7, in8, in9, out9):

```
In [7]: matriz1=[]
        for i in range(len(matriz)):
            matriz1.append(matriz[i][0])

        matriz2=[]
        for i in range(len(matriz)):
            matriz2.append(matriz[i][1])

        centros1=[]
        for i in range(len(centros)):
            centros1.append(centros[i][0])

        centros2=[]
        for i in range(len(centros)):
            centros2.append(centros[i][1])
```

```
In [8]: labels = kmeans.predict(matriz)
        colores = []
        for i in range(numero_clusters):
            colores.append([random.randint(0, 255)/255, random.randint(0, 255)/255, random.randint(0, 255)/255])
        asignar=[]
        for row in labels:
            asignar.append(colores[row])
```

```
In [9]: fig = plt.figure()

        image = img.imread('mapa.jpg')
        imagen = plt.imshow(image, extent = (min(matriz1),max(matriz1),min(matriz2), max(matriz2)))
        ax0 = imagen.axes
        ax0.invert_xaxis()
```

```
ax1 = fig1.axes
ax1.invert_xaxis()
fig2 = plt.scatter(centros1,centros2,marker = '*', s = 500)
ax2 = fig2.axes
ax2.invert_xaxis()
```



Q-Learning

After importing the libraries, we proceed to create the reward array. Also, we define the coordinates where the agent will terminate the process. In this case, it will stop after reaching all the target states (where the company has to deliver the packages).

```
In [1]: import numpy as np
        import pygame
        from time import time,sleep
        from random import randint as r
        import random
        import pandas as pd
```

```
pygame 2.1.2 (SDL 2.0.18, Python 3.8.8)
Hello from the pygame community. https://www.pygame.org/contribute.html
```

```
In [2]: n = 29
        # Número de columnas en la matriz.
```

```
In [3]: m = 24
        # Número de filas en la matriz.
        # La matriz es de 29*24 --> n*m
```

```
In [4]: reward = np.zeros((m,n))
```

```
In [5]: terminals = []
        reward[m == 21, n == 25] = 1 #El objetivo se pondrá al final de la NQS
```

```
In [6]: states = {}
        k = 0
        for i in range(m):
            for j in range(n):
                states[(i,j)] = k
                k+=1
```

Then we create the states array, that represents the 695 possible locations in which we divide Bogotá.

```
In [6]: states = {}
        k = 0
        for i in range(m):
            for j in range(n):
                states[(i,j)] = k
                k+=1
```

```
In [7]: print (states)
```

```
{(0, 0): 0, (0, 1): 1, (0, 2): 2, (0, 3): 3, (0, 4): 4, (0, 5): 5, (0, 6): 6, (0, 7): 7, (0, 8): 8, (0, 9): 9, (0, 10): 10, (0,
11): 11, (0, 12): 12, (0, 13): 13, (0, 14): 14, (0, 15): 15, (0, 16): 16, (0, 17): 17, (0, 18): 18, (0, 19): 19, (0, 20): 20,
(0, 21): 21, (0, 22): 22, (0, 23): 23, (0, 24): 24, (0, 25): 25, (0, 26): 26, (0, 27): 27, (0, 28): 28, (1, 0): 29, (1, 1): 30,
(1, 2): 31, (1, 3): 32, (1, 4): 33, (1, 5): 34, (1, 6): 35, (1, 7): 36, (1, 8): 37, (1, 9): 38, (1, 10): 39, (1, 11): 40, (1, 1
2): 41, (1, 13): 42, (1, 14): 43, (1, 15): 44, (1, 16): 45, (1, 17): 46, (1, 18): 47, (1, 19): 48, (1, 20): 49, (1, 21): 50,
(1, 22): 51, (1, 23): 52, (1, 24): 53, (1, 25): 54, (1, 26): 55, (1, 27): 56, (1, 28): 57, (2, 0): 58, (2, 1): 59, (2, 2): 60,
(2, 3): 61, (2, 4): 62, (2, 5): 63, (2, 6): 64, (2, 7): 65, (2, 8): 66, (2, 9): 67, (2, 10): 68, (2, 11): 69, (2, 12): 70, (2,
13): 71, (2, 14): 72, (2, 15): 73, (2, 16): 74, (2, 17): 75, (2, 18): 76, (2, 19): 77, (2, 20): 78, (2, 21): 79, (2, 22): 80,
(2, 23): 81, (2, 24): 82, (2, 25): 83, (2, 26): 84, (2, 27): 85, (2, 28): 86, (3, 0): 87, (3, 1): 88, (3, 2): 89, (3, 3): 90,
(3, 4): 91, (3, 5): 92, (3, 6): 93, (3, 7): 94, (3, 8): 95, (3, 9): 96, (3, 10): 97, (3, 11): 98, (3, 12): 99, (3, 13): 100,
(3, 14): 101, (3, 15): 102, (3, 16): 103, (3, 17): 104, (3, 18): 105, (3, 19): 106, (3, 20): 107, (3, 21): 108, (3, 22): 109,
(3, 23): 110, (3, 24): 111, (3, 25): 112, (3, 26): 113, (3, 27): 114, (3, 28): 115, (4, 0): 116, (4, 1): 117, (4, 2): 118, (4,
3): 119, (4, 4): 120, (4, 5): 121, (4, 6): 122, (4, 7): 123, (4, 8): 124, (4, 9): 125, (4, 10): 126, (4, 11): 127, (4, 12): 12
8, (4, 13): 129, (4, 14): 130, (4, 15): 131, (4, 16): 132, (4, 17): 133, (4, 18): 134, (4, 19): 135, (4, 20): 136, (4, 21): 13
7, (4, 22): 138, (4, 23): 139, (4, 24): 140, (4, 25): 141, (4, 26): 142, (4, 27): 143, (4, 28): 144, (5, 0): 145, (5, 1): 146,
(5, 2): 147, (5, 3): 148, (5, 4): 149, (5, 5): 150, (5, 6): 151, (5, 7): 152, (5, 8): 153, (5, 9): 154, (5, 10): 155, (5, 11):
156, (5, 12): 157, (5, 13): 158, (5, 14): 159, (5, 15): 160, (5, 16): 161, (5, 17): 162, (5, 18): 163, (5, 19): 164, (5, 20): 1
65, (5, 21): 166, (5, 22): 167, (5, 23): 168, (5, 24): 169, (5, 25): 170, (5, 26): 171, (5, 27): 172, (5, 28): 173, (6, 0): 17
4, (6, 1): 175, (6, 2): 176, (6, 3): 177, (6, 4): 178, (6, 5): 179, (6, 6): 180, (6, 7): 181, (6, 8): 182, (6, 9): 183, (6, 1
0): 184, (6, 11): 185, (6, 12): 186, (6, 13): 187, (6, 14): 188, (6, 15): 189, (6, 16): 190, (6, 17): 191, (6, 18): 192, (6, 1
9): 193, (6, 20): 194, (6, 21): 195, (6, 22): 196, (6, 23): 197, (6, 24): 198, (6, 25): 199, (6, 26): 200, (6, 27): 201, (6, 2
8): 202, (7, 0): 203, (7, 1): 204, (7, 2): 205, (7, 3): 206, (7, 4): 207, (7, 5): 208, (7, 6): 209, (7, 7): 210, (7, 8): 211,
(7, 9): 212, (7, 10): 213, (7, 11): 214, (7, 12): 215, (7, 13): 216, (7, 14): 217, (7, 15): 218, (7, 16): 219, (7, 17): 220,
(7, 18): 221, (7, 19): 222, (7, 20): 223, (7, 21): 224, (7, 22): 225, (7, 23): 226, (7, 24): 227, (7, 25): 228, (7, 26): 229,
(7, 27): 230, (7, 28): 231, (8, 0): 232, (8, 1): 233, (8, 2): 234, (8, 3): 235, (8, 4): 236, (8, 5): 237, (8, 6): 238, (8, 7):
```

We also need a Q-table, that will store the values of each action, and, iteratively, change them with the Q equation.

```
In [8]: qTable = np.zeros((4,m*n))
        print (qTable)

[[0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]]

In [9]: states2 = {}
        o = 0
        for i in range(4):
            for j in range(n*m):
                states2[(i,j)] = o
                o+=1

In [10]: print (states2)

256, (0, 257): 257, (0, 258): 258, (0, 259): 259, (0, 260): 260, (0, 261): 261, (0, 262): 262, (0, 263): 263, (0, 264): 264,
(0, 265): 265, (0, 266): 266, (0, 267): 267, (0, 268): 268, (0, 269): 269, (0, 270): 270, (0, 271): 271, (0, 272): 272, (0, 2
73): 273, (0, 274): 274, (0, 275): 275, (0, 276): 276, (0, 277): 277, (0, 278): 278, (0, 279): 279, (0, 280): 280, (0, 281):
281, (0, 282): 282, (0, 283): 283, (0, 284): 284, (0, 285): 285, (0, 286): 286, (0, 287): 287, (0, 288): 288, (0, 289): 289,
(0, 290): 290, (0, 291): 291, (0, 292): 292, (0, 293): 293, (0, 294): 294, (0, 295): 295, (0, 296): 296, (0, 297): 297, (0, 2
98): 298, (0, 299): 299, (0, 300): 300, (0, 301): 301, (0, 302): 302, (0, 303): 303, (0, 304): 304, (0, 305): 305, (0, 306):
306, (0, 307): 307, (0, 308): 308, (0, 309): 309, (0, 310): 310, (0, 311): 311, (0, 312): 312, (0, 313): 313, (0, 314): 314,
(0, 315): 315, (0, 316): 316, (0, 317): 317, (0, 318): 318, (0, 319): 319, (0, 320): 320, (0, 321): 321, (0, 322): 322, (0, 3
23): 323, (0, 324): 324, (0, 325): 325, (0, 326): 326, (0, 327): 327, (0, 328): 328, (0, 329): 329, (0, 330): 330, (0, 331):
331, (0, 332): 332, (0, 333): 333, (0, 334): 334, (0, 335): 335, (0, 336): 336, (0, 337): 337, (0, 338): 338, (0, 339): 339,
(0, 340): 340, (0, 341): 341, (0, 342): 342, (0, 343): 343, (0, 344): 344, (0, 345): 345, (0, 346): 346, (0, 347): 347, (0, 3
48): 348, (0, 349): 349, (0, 350): 350, (0, 351): 351, (0, 352): 352, (0, 353): 353, (0, 354): 354, (0, 355): 355, (0, 356):
356, (0, 357): 357, (0, 358): 358, (0, 359): 359, (0, 360): 360, (0, 361): 361, (0, 362): 362, (0, 363): 363, (0, 364): 364,
(0, 365): 365, (0, 366): 366, (0, 367): 367, (0, 368): 368, (0, 369): 369, (0, 370): 370, (0, 371): 371, (0, 372): 372, (0, 3
73): 373, (0, 374): 374, (0, 375): 375, (0, 376): 376, (0, 377): 377, (0, 378): 378, (0, 379): 379, (0, 380): 380, (0, 381):
```

Then, we import the logic matrix that reflects the accessible states. With this array, we avoid the possibility of the agent accessing restricted cells. In the Q-table, all the forbidden actions will be represented as a constant zero.

```
In [11]: dataframe = pd.read_csv(r"Libro1.csv", sep = ';', header=None)
         print (dataframe)

     0   1   2   3   4   5   6   7   8   9  ...  19  20  21  22  23  24  25  \
0    0   0   0   0   0   0   0   0   0   0  ...   0   0   0   0   0   0   0
1    0   0   0   0   0   0   0   1   1   1  ...   1   1   1   0   0   0   0
2    0   0   0   0   0   0   0   1   0   0  ...   0   0   1   0   0   0   0
3    0   0   0   1   1   1   1   1   1   1  ...   1   1   1   1   1   0   0
4    0   0   0   0   1   0   0   0   0   0  ...   0   0   1   0   1   0   0
5    0   0   0   1   1   1   1   0   0   0  ...   0   0   1   0   1   0   0
6    0   0   0   0   1   0   1   0   0   0  ...   0   0   1   0   1   0   0
7    0   0   0   1   1   1   1   1   1   1  ...   1   1   1   1   1   1   1
8    0   0   0   0   1   0   1   0   1   0  ...   0   0   1   0   0   0   0
9    0   0   1   1   1   0   1   0   1   0  ...   0   0   1   0   0   0   0
10   0   0   1   0   1   0   1   0   1   0  ...   0   0   1   0   0   0   0
11   0   0   1   0   1   0   1   1   1   1  ...   1   1   1   1   1   1   0
12   0   0   1   0   1   0   0   0   1   0  ...   0   0   1   0   0   1   0
13   0   0   1   0   1   0   0   0   1   0  ...   0   0   1   0   0   1   0
14   0   0   1   0   1   0   0   0   1   0  ...   0   0   1   0   0   1   0
15   0   0   1   0   1   0   0   0   1   0  ...   0   0   1   0   0   1   0
16   0   0   1   0   1   0   0   0   1   0  ...   0   0   1   0   0   1   0
17   0   0   1   0   1   1   1   1   1   1  ...   1   1   1   1   1   1   0
18   0   0   1   0   0   0   0   0   1   0  ...   0   0   1   0   0   1   0
19   0   1   1   1   1   1   1   1   1   1  ...   1   1   1   1   1   1   1
20   0   0   1   0   0   0   0   0   1   0  ...   0   0   1   0   0   1   0
21   0   0   1   1   1   1   1   1   1   1  ...   1   1   1   1   0   1   1
22   0   0   1   0   0   0   0   0   1   0  ...   0   0   0   0   0   0   0
23   0   0   0   0   0   0   0   0   0   0  ...   0   0   0   0   0   0   0
```

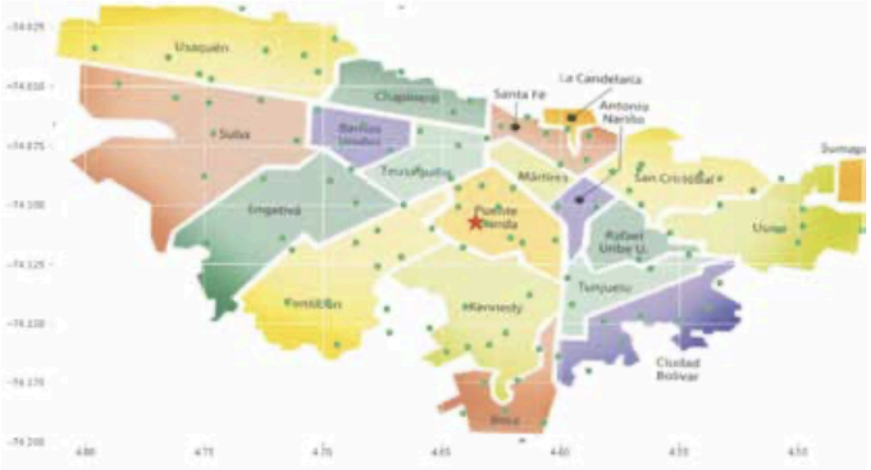
Because of the problems we had with the database in the API of Azure, we couldn't download the traffic flow data and we had to stop this part of our project.

6.3 Test the prototype and establish improvements

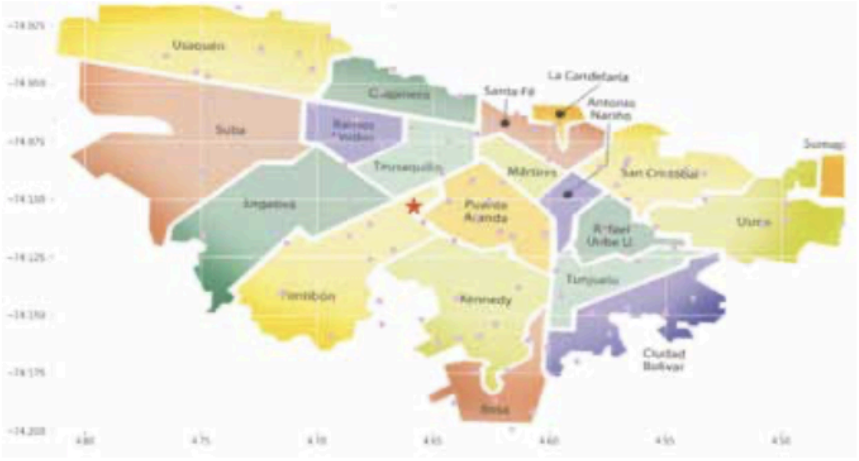
On one hand, we run our K-means prototype for different kind of configurations (i.e. number of warehouses, criteria of selection of the place according to package demand, land price or a combination of both properties) and we got very good results according to the feedback we received from the expert in logistics Daniel Villa.

For example, we tested the algorithm using package demand and land cost and asking for five places for warehouse location and we got as result suggestions of implementation as shown below. From our knowledge of the city locations have a good balance between both choosed criteria.

Then we asked for one single place using package demand and land cost as criteria and we got a location in the locality of Puente Aranda which is the same location the expert in logistics told us possibly is the optimal place to put a logistic distribution point (warehouse) in the city as shown below. It reinforced our confidence in our algorithm



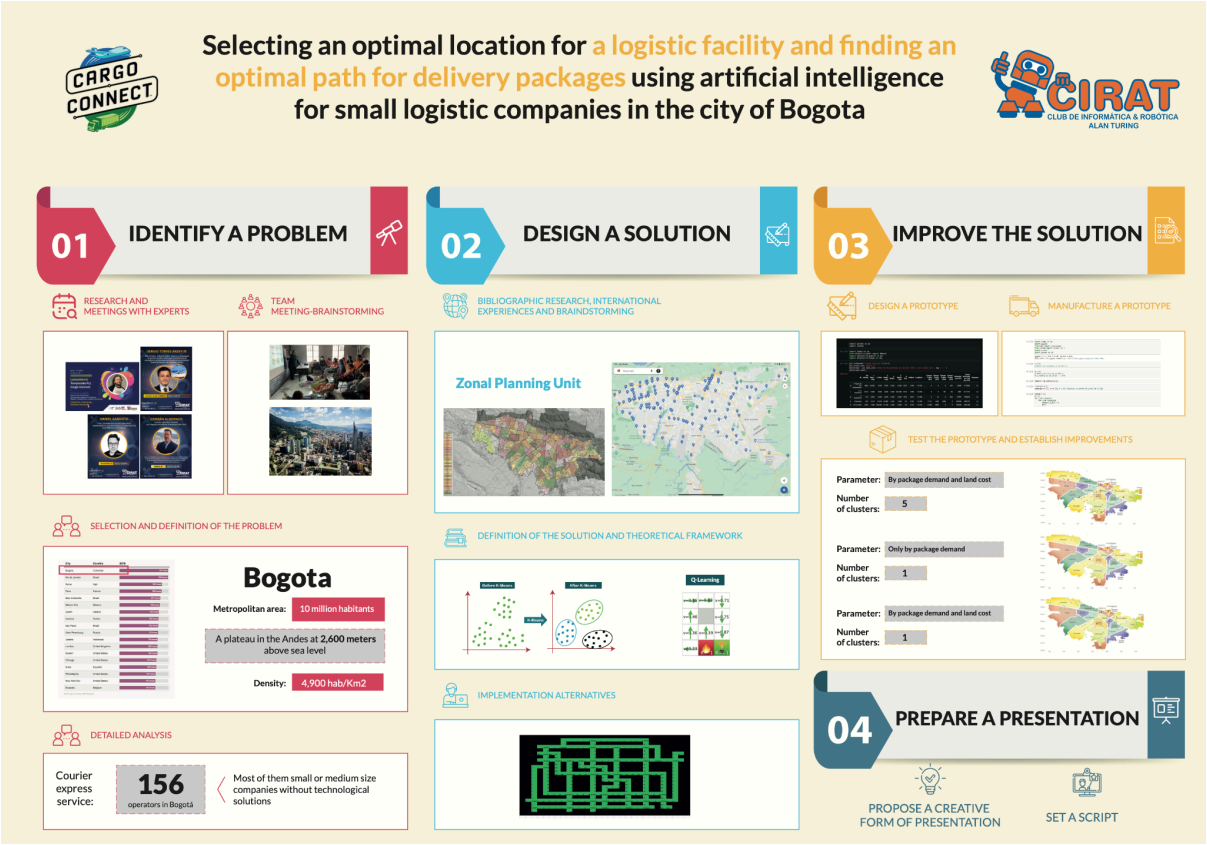
Finally we use as only criteria package demand and we got a location in the locality of Fontibón, close to the airport as shown below.



On the other hand, due to the problems with the licensing for accessing Azure real time traffic information we were unable to test or establish improvements for our delivery package routing algorithm.

7. Presentation

For the presentation of our project, we worked in an infographic approach that could comprise in one place, the different steps of our methodology as well as some iconic pictures, diagrams or results. The result is shown below. Letter looks small in this document but for the presentation we printed it in color in a 1 m x 0,7 m poster to use it to help us to communicate ideas. We also will be using it in our stand to explain the public curious about our innovation project.



8. Conclusions

The conclusions we got from our innovation project are the following:

1. We selected two problems for our innovation project: The first one consists in identifying the best place in Bogota to locate a package distribution logistics center (warehouse), and the second consists in establishing an optimal route for the distribution of packages in the city of Bogota.
    a. According to the international benchmark as well as the feedback of experts both problems are very relevant for small and medium logistic companies as most of them don't have technological solutions available and to get optimal ways to solve those problems are essential for the business core: provides new kinds of business and services, reduces costs, improves operational efficiency and competitiveness.
2. For the problem related with the location of an optimal place for one or more package distribution logistic centers we used K-means, an artificial intelligence algorithm. The implementation was very successful:
    a. The results we got provide accurate alternatives for the location under different criteria of optimization (package delivery demand, land cost or a combination of both criteria). It works properly for the selection of one or more locations for the distribution logistic center (warehouse).
3. For the problem of the optimal route for the distribution of packages we used Q-Learning, a method of machine learning based on reinforcement learning techniques. Even though we prepared the algorithm and think our idea is good, the

implementation was not successful as we had last minute problems to get the real time traffic information we needed from Azure platform, as we were notified of a problem with the license conditions.

   a. We realized how important the data to be used in any artificial intelligence development are. For example, the traffic information of Bogota is given for each person driving a car in the city but the aggregation of the data converges to a limited number of platforms and access that is difficult and expensive.

4. The usage of the Design Thinking methodology provided us with a very good tool in order to have a comprehensive and organized set of steps to define and solve the problem.
5. We improved our understanding of our city: how it is geographically divided, how is the demography, how is the infrastructure and the traffic. We realized several problems but also opportunities to use the technology to improve the conditions of operation of the small and medium enterprises in the logistic sector, which at the end, means improvements in the efficiency and finally in the economic growth of Bogota and Colombia.
6. Even though our solution is specific for Bogota, the concepts involved have the potential to be used for other cities in Colombia, in Latin America or even in the developing countries.

9. References

CAF. (2021). *Hacia el desarrollo de infraestructuras eficientes y sostenibles en América Latina: Oportunidades y beneficios de la digitalización. Resumen ejecutivo*. Banco de Desarrollo de América Latina. https://scioteca.caf.com/handle/123456789/1799

DHL. (2018, May 23). *DHL Launches Innovative Road Transportation Across India*. DHL. Retrieved April 9, 2022, from https://www.dhl.com/in-en/home/press/press-archive/2018/dhl-launches-innovative-road-transportation-across-india.html