Safe Routes Suggestion App

T. A. Munarolo   *L. A. Villas*
F. G. Figueiredo
L. R. V. Madoreira

Technical Report - IC-PFG-18-07 - Relatório Técnico
July - 2018 - Julho

UNIVERSIDADE ESTADUAL DE CAMPINAS
INSTITUTO DE COMPUTAÇÃO

Safe Routes Suggestion App

Tiago Abreu Munarolo[*] Felipe Gustavo Figueiredo[†]
Luiz Renato Valadão Madoreira[‡] Leandro Villas[§]

July 15, 2018

Abstract

Urban violence was always a problem in Brazil, and recently it escalated even more in major cities. At the same time, a lot of navigation applications started to be more used by a great part of drivers, but those programs, when looking for the fastest route, can lead to dangerous places. Therefore those conventional solutions for routing are not sufficient for today's use because they do not consider the violence factor.

The developed application looks for the criminal rate of different districts in the city of Campinas and use this information to calculate not just the fastest route, but one that is also safe. The process can be replicated to many different cities that contains this kind of information.

1 Introduction

The current navigation applications were developed to help drivers to arrive in their destination as fast as possible, having in mind the highly rushed lives that is the reality of many great cities around the world. However, those apps were built in countries with less public safety problems comparing to Brazil. In the big cities of this country the reality is not as simple.

According to the Department of Public Safety of São Paulo (SSP-SP), there were almost one thousand vehicle robberies in the city of Campinas just on the first four months of 2018, that is one robbery each three hours. Furthermore there were many other crimes that can also affect drivers around the city, such as cargo theft or kidnaps. And this kind of situation occurs in almost any major city in Brazil.

Therefore, to think just about the fastest route is not sufficient in those cities. The developed navigation app was built to consider violence related to driving as an important factor on the route calculation. The purpose is to provide safe and fast routes with a focus on the former. To calculate these routes, the application utilizes an specific set of criminal data for the city of Campinas supplied by a third party API.

This third party API is named Senso Criminal. It provides the number of occurrences of each kind of crime in almost two thousand different sectors of Campinas. Using this information the app, between the fastest routes, chooses the one that passes on sectors with the lowest possible rate of crimes that can affect drivers.

[*]Inst. de Computação, UNICAMP, 13083-852 Campinas, SP. tiago.am94@gmail.com
[†]Inst. de Computação, UNICAMP, 13083-852 Campinas, SP. felipegusfigueiredo@gmail.com
[‡]Inst. de Computação, UNICAMP, 13083-852 Campinas, SP. luizrenatovmado@gmail.com
[§]Inst. de Computação, UNICAMP, 13083-852 Campinas, SP. leandro@ic.unicamp.br

2 Solution

2.1 Input data

The algorithm uses two input datas to calculate one safe and fast route between the user current location and the chosen destination. The first one is a set of almost two thousand different sectors of the city of Campinas, containing the level of insecurity in each one of them, based on the number of occurences of crimes that can affect drivers in each of the sectors. This information is provided by the third party API named Senso Criminal.

The second one is the first route between the starting point and the destination provided by Google Maps Directions API. Maps can provide more than one route (normally it provides three), so it was chosen to use the first one, because it is regurlaly considered the best one by the users.

2.2 The algorithm

In this section it is presented the algorithm used to solve the problem and its steps.

The application firstly makes a request to the Google Maps Directions API for routes to that destination. From these routes, we get the first one and show it to the user. The next step is to create a graph with the central points of the neighboring sectors on this route.

To discover which sectors contains these points, we use the ray casting algorithm to determine if the point is inside each sector polygon in a distance radius of one kilometer.

After discovering which sectors are part of this route, we use the centroid of these sectors to discover their neighbors, then all these points are connected to create edges for the graph that will be utilized to calculate the waypoints that will generate a safe and fast route.The cases for the existence of an edge between a vertex e and v are if are:

- There was a connection on the main route given by Google Directions API;
- Vertex e and v separation is smaller than 500 meters.

The weight of an edge e is calculated according to the following equation:

$$Weight_e = (Risk_{source} + Risk_{destination}) * ceil(log(Distance_{nextdestination})) * log(Distance_{finaldestination})$$

Where Risk-source is the risk level of the sector which is the source of the edge, and Risk-destination is the risk level of the sector which is the destination of the edge. Distance-next destination is the distance between the source and the destination point, and Distance-final destination is the distance from the destination point of the edge to the final destination of the route.

The last part of the algorithm runs Dijkstra algorithm over the graph we acquired on the previous step. The result of the shortest paths to the destination is a set of vertexes with the latitude and longitude that form a path from the user current current location to the chosen final destination passing through the safest possible sectors. This is the output of the algorithm.

2.3 Output

The algorithm outputs a set of waypoints that is sent to Google Maps Directions API again, serving as a guide to generate a new route passing through all of them. Google will optimize this route to reach all the waypoints on the fastest way. This route will be the safe and fast route that the application will be able to show the user if he/she requests it.

3 Related Work

There are a lot of related works in this area, but each of them try to solve the problem in a different way, using different data or in a different platform. Souza et al., for example, proposes in [3] a hybrid architecture, composed by a server and a software executed in On Board Units (OBUs) of the participant vehicles. In this system, called #PAS, the vehicles would report traffic information to the server, so that it could combine these data with its own information about criminal events (gathered from SSP-SP). Afther that, the server can detected congested roads and ,as soon as one of them is detected, it sends a routing notification for the vehicles going to that congestion containing the knowledge about traffic conditions and insecure areas. With that, the vehicles can calculate a new route to improve mobility and security. The main problem with #PAS is that intelligent OBUs, capable of doing that, are not common around the majority of user vehicles yet. So, it is hard for the system to be massively used and really resolve the problem today.

Shah et al. [4], in the other hand, tried to solve the problem by creating a system based on crowdsourcing called CROWDSAFE. It allows users to report criminal incidents using smartphones to its central server. Processing the incoming data, the server builds unsafe areas. In this way, CROWDSAFE can suggest safer routes to the users. However, as the system does not have information about traffic conditions, the routes can not be fast enough and it can create congestion in unsafe areas. Another problem is that being purely crowdsourcing, the system may be limited, because the reports can be insufficient or false.

Another approach to this can be found in Galbrun et al. [5], where they also consider this as a biobjective shortest path problem using a tradeoff model between distance and risk associated to find a safe route to the destination. The problem of this solution is that it can be costly depending on the number of routes generated that are going to be generated. Besides that, in a real urban environment taking into consideration only distances and risk is not enough to provide a good route since vehicles could end in areas with lots of traffic when trying to avoid an area which is not safe.

4 Results

4.1 Routing

The results obtained on the new routing system taking into account sectors with criminal data can be considered successful. The application calculates a new safe route based on the ones provided by Google Maps Directions API trying to avoid sectors with high criminal rates when possible.

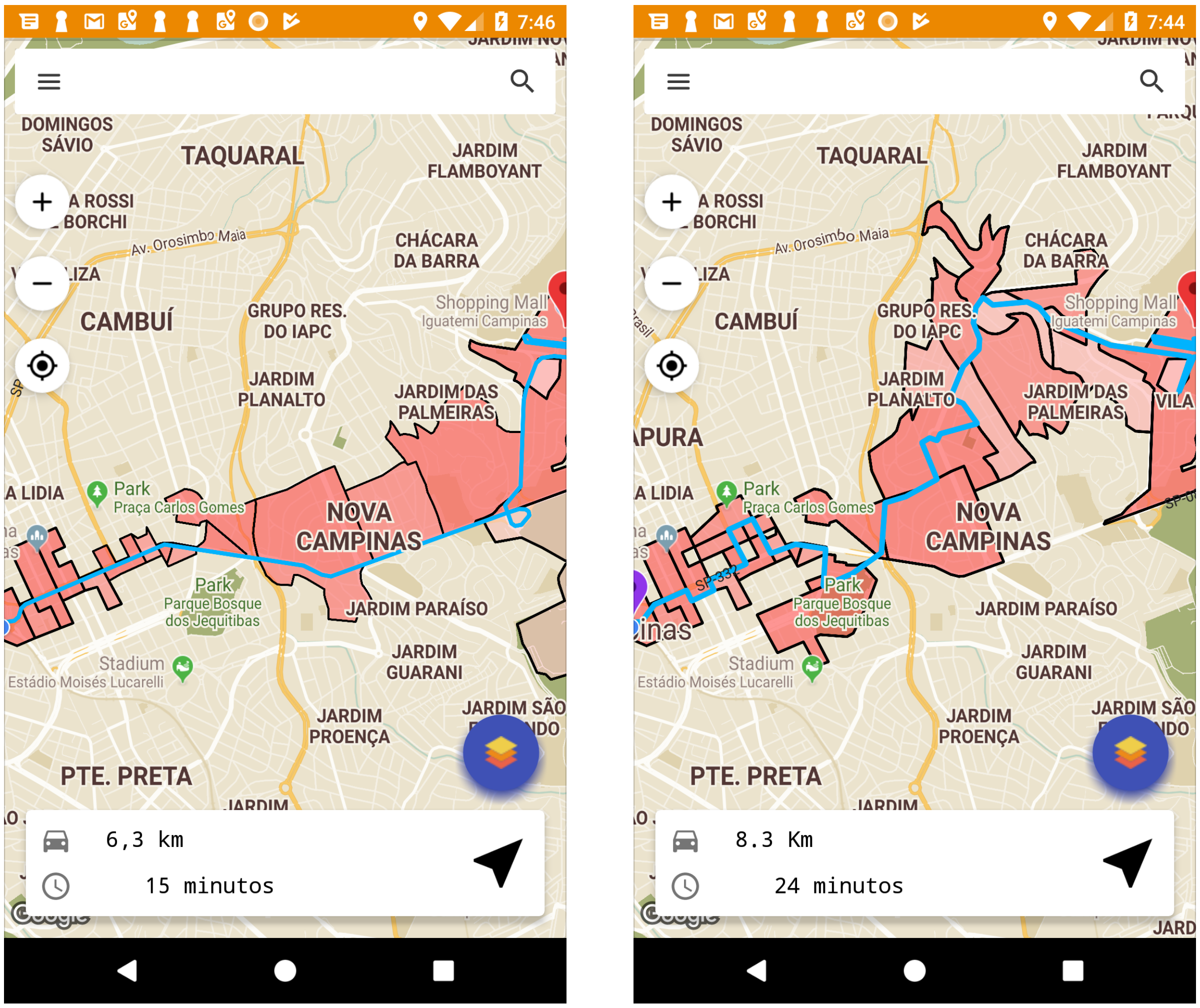


Figure 1: Comparison between Google fastest route (left) and Application safest route (right).

4.2 Interface

The application was built considering Material Design patterns. The main screen shows the map (using Google Maps API) and a search bar:

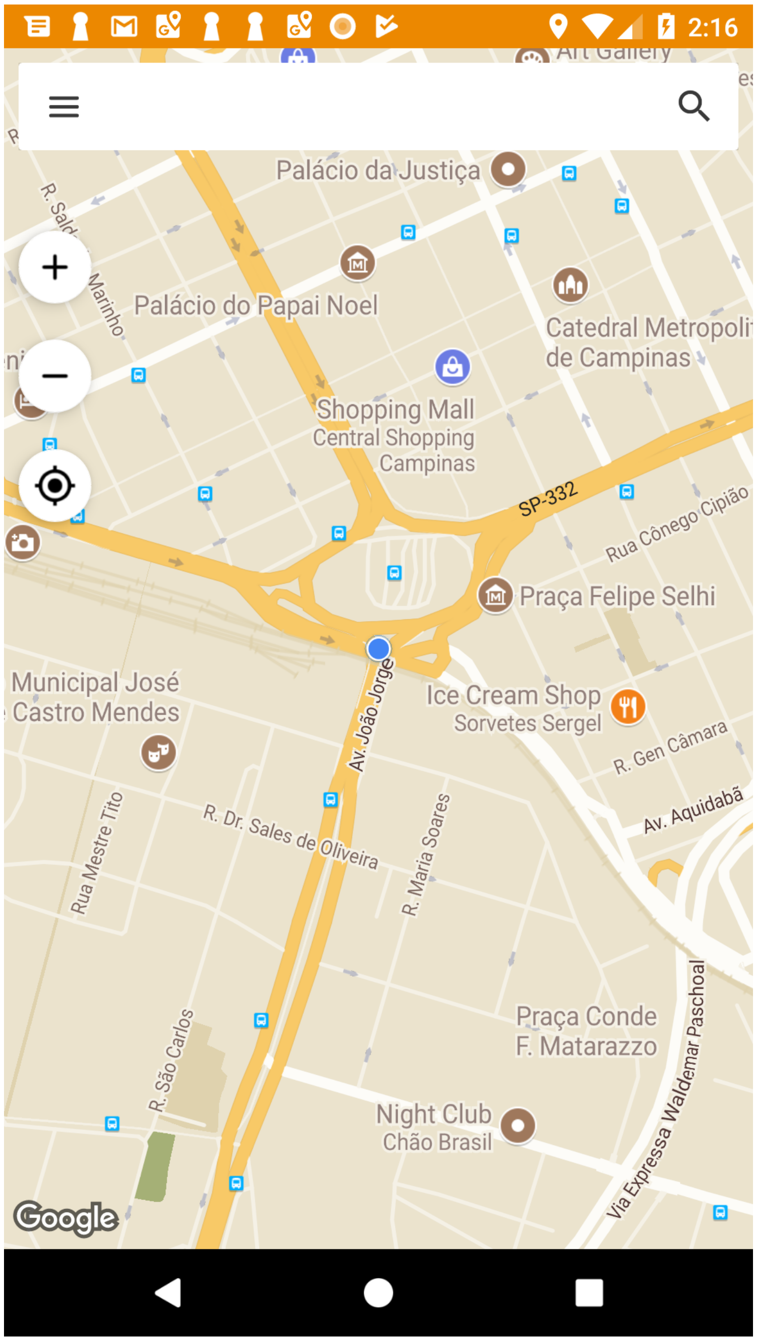
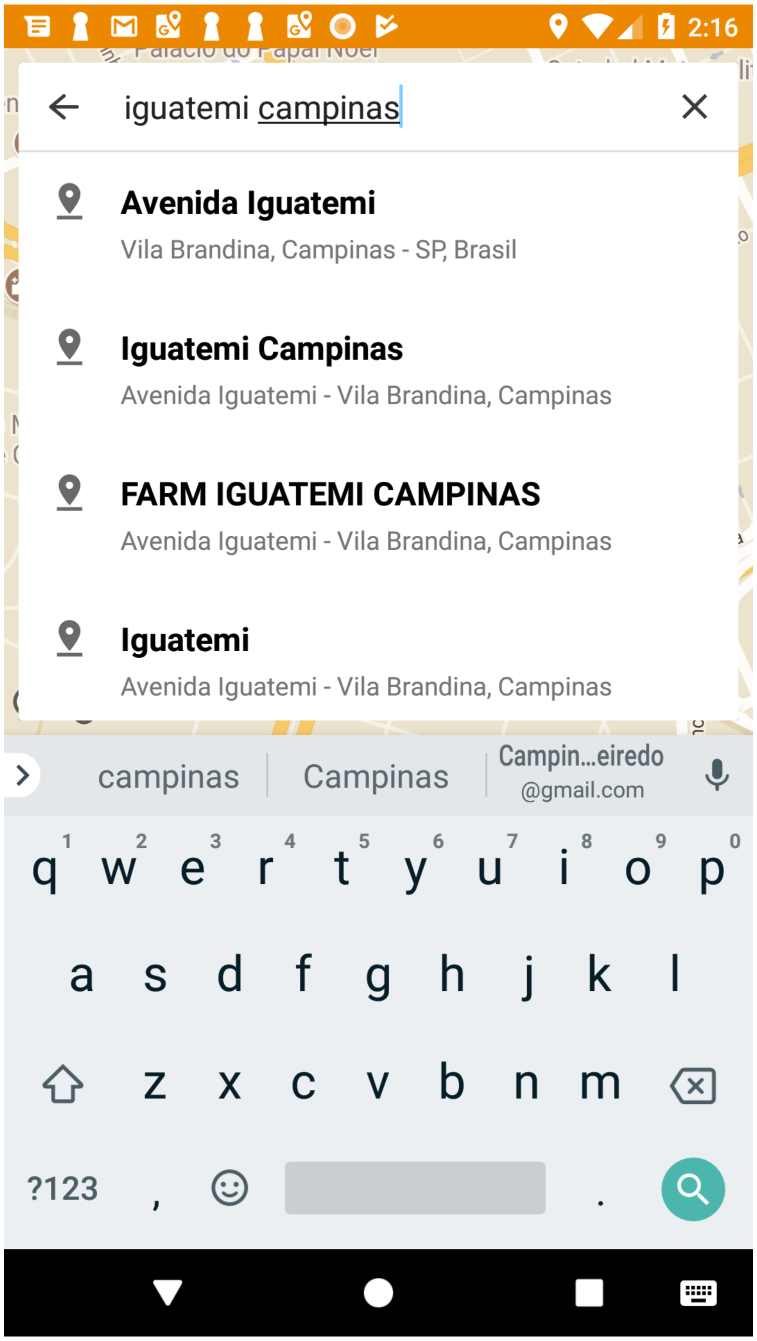


Figure 2: Main screen.

The search uses Google PlaceAutocomplete API to suggest places for the user based on the real-time input:


iguatemi campinas

Avenida Iguatemi
Vila Brandina, Campinas - SP, Brasil

Iguatemi Campinas
Avenida Iguatemi - Vila Brandina, Campinas

FARM IGUATEMI CAMPINAS
Avenida Iguatemi - Vila Brandina, Campinas

Iguatemi
Avenida Iguatemi - Vila Brandina, Campinas

Figure 3: Main screen.

When the user chooses a place, information about that is shown in a window, with the option to trace the route to that place:

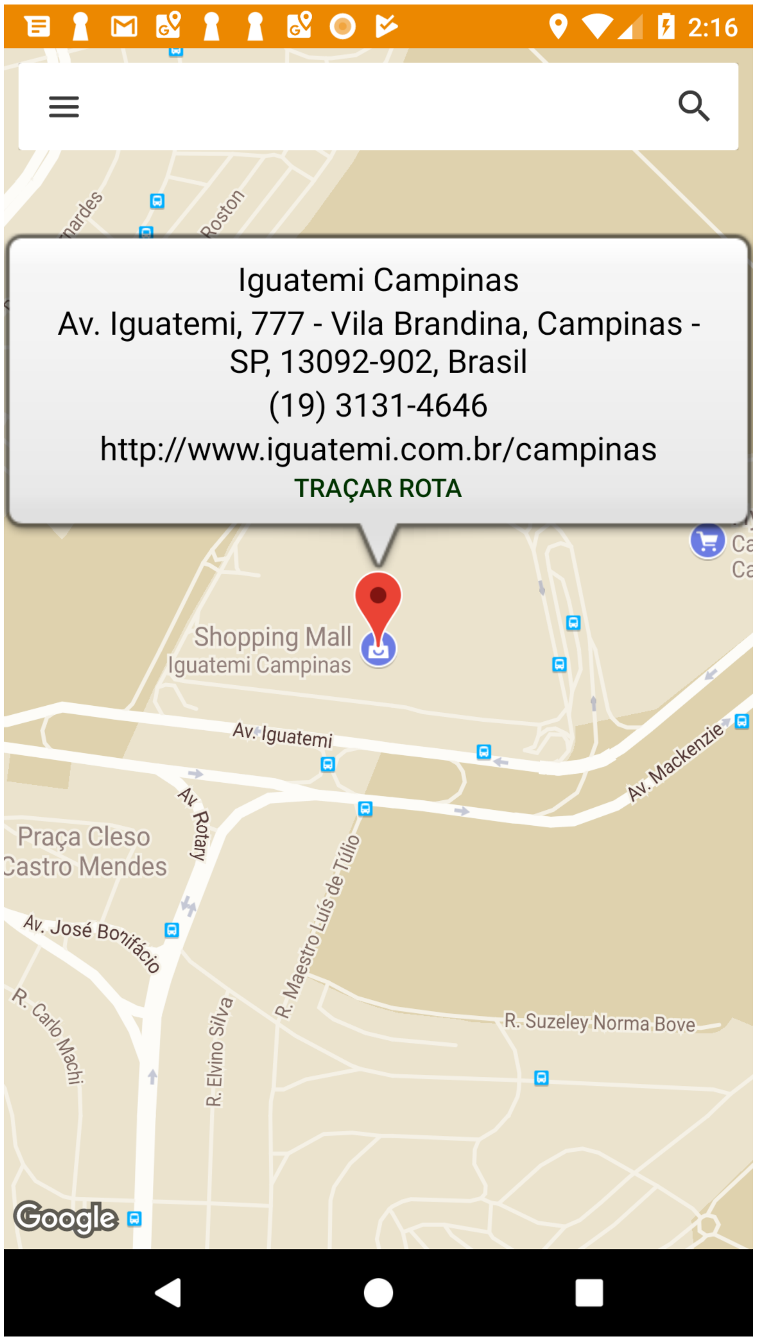

Iguatemi Campinas
Av. Iguatemi, 777 - Vila Brandina, Campinas - SP, 13092-902, Brasil
(19) 3131-4646
http://www.iguatemi.com.br/campinas
TRAÇAR ROTA

Figure 4: Main screen.

The button on the upper left corner brings a drawer with various options for the user:

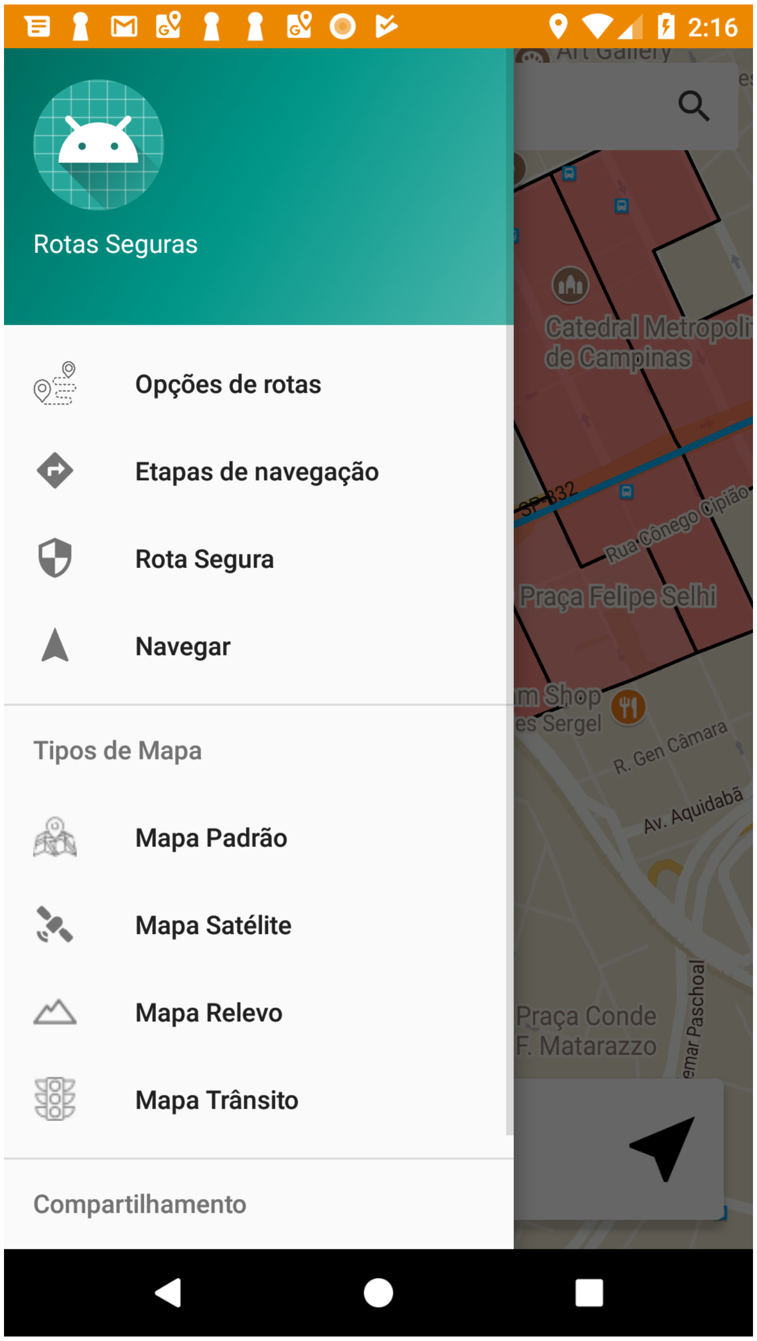
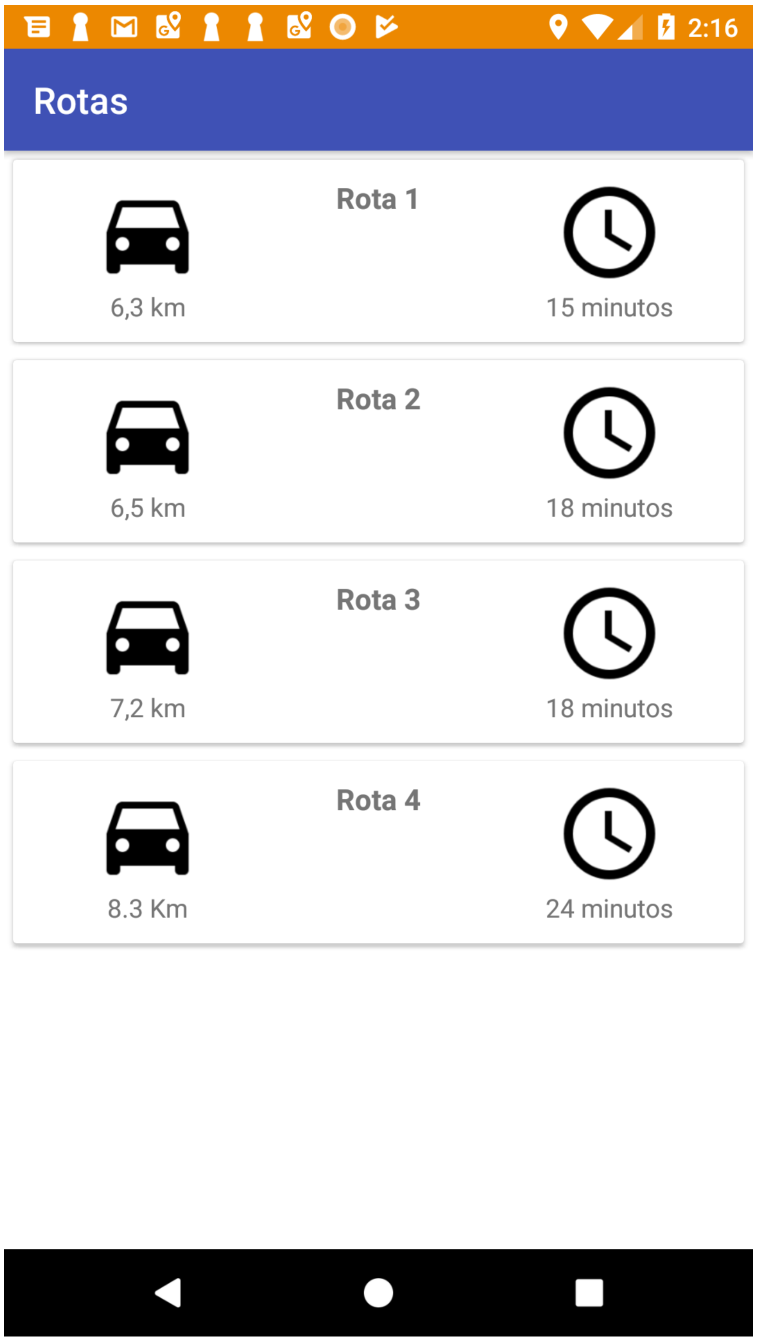


Figure 5: Drawer view.

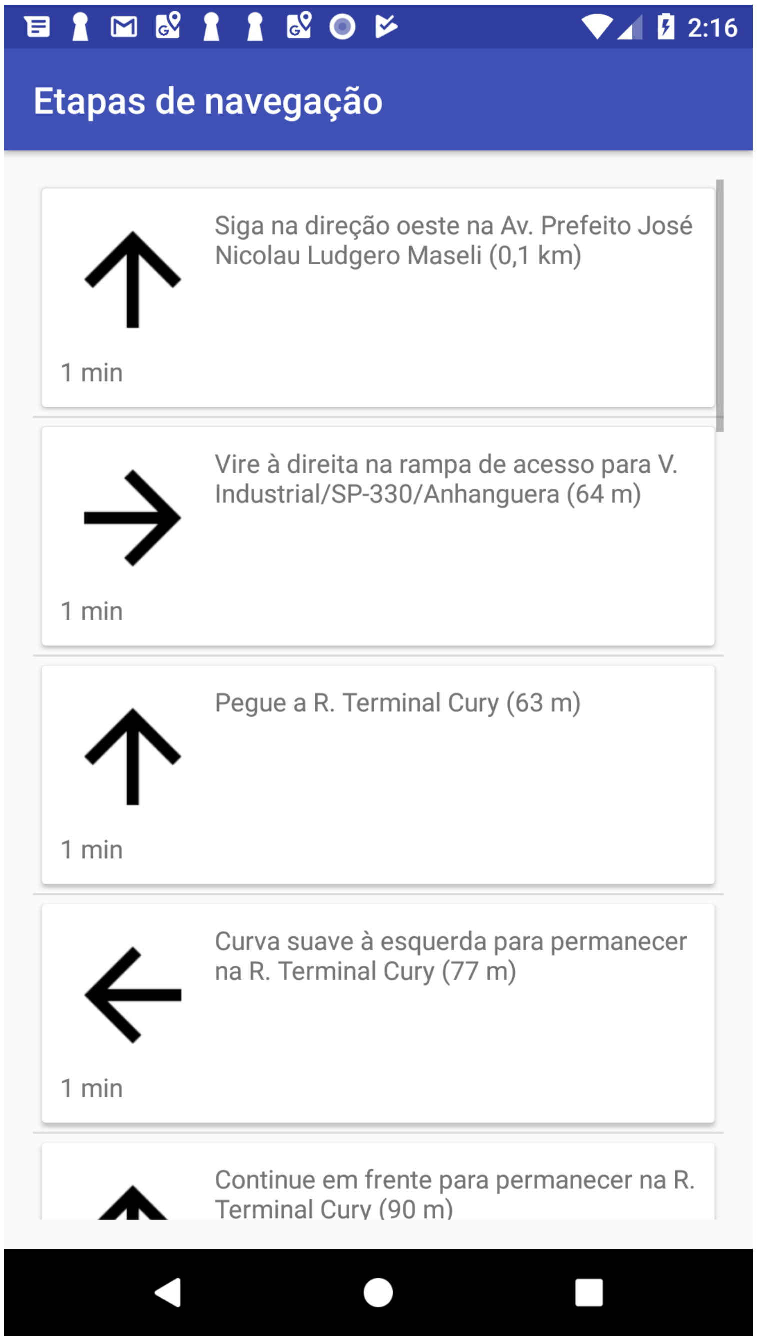
The first option is the routes options. This shows a list with usually 4 routes. The user can click which one to choose. If the safe route is already calculated, it will be the last one on this list. There is also an option on the drawer to choose directly the safe route from there.



6,3 km	Rota 1	15 minutos
6,5 km	Rota 2	18 minutos
7,2 km	Rota 3	18 minutos
8.3 Km	Rota 4	24 minutos

Figure 6: Routes option list.

The second options is the steps list. This shows every step for the navigation of the selected route. There is also an option on the drawer to use Google Maps navigation using those steps.



Etapas de navegação

Siga na direção oeste na Av. Prefeito José Nicolau Ludgero Maseli (0,1 km)

1 min

Vire à direita na rampa de acesso para V. Industrial/SP-330/Anhanguera (64 m)

1 min

Pegue a R. Terminal Cury (63 m)

1 min

Curva suave à esquerda para permanecer na R. Terminal Cury (77 m)

1 min

Continue em frente para permanecer na R. Terminal Cury (90 m)

Figure 7: Steps list.

The interface also has a legend for the levels of danger showed on the map. The legend shows shades of red which goes from lighter to darker colors, which represent an increase in peril.

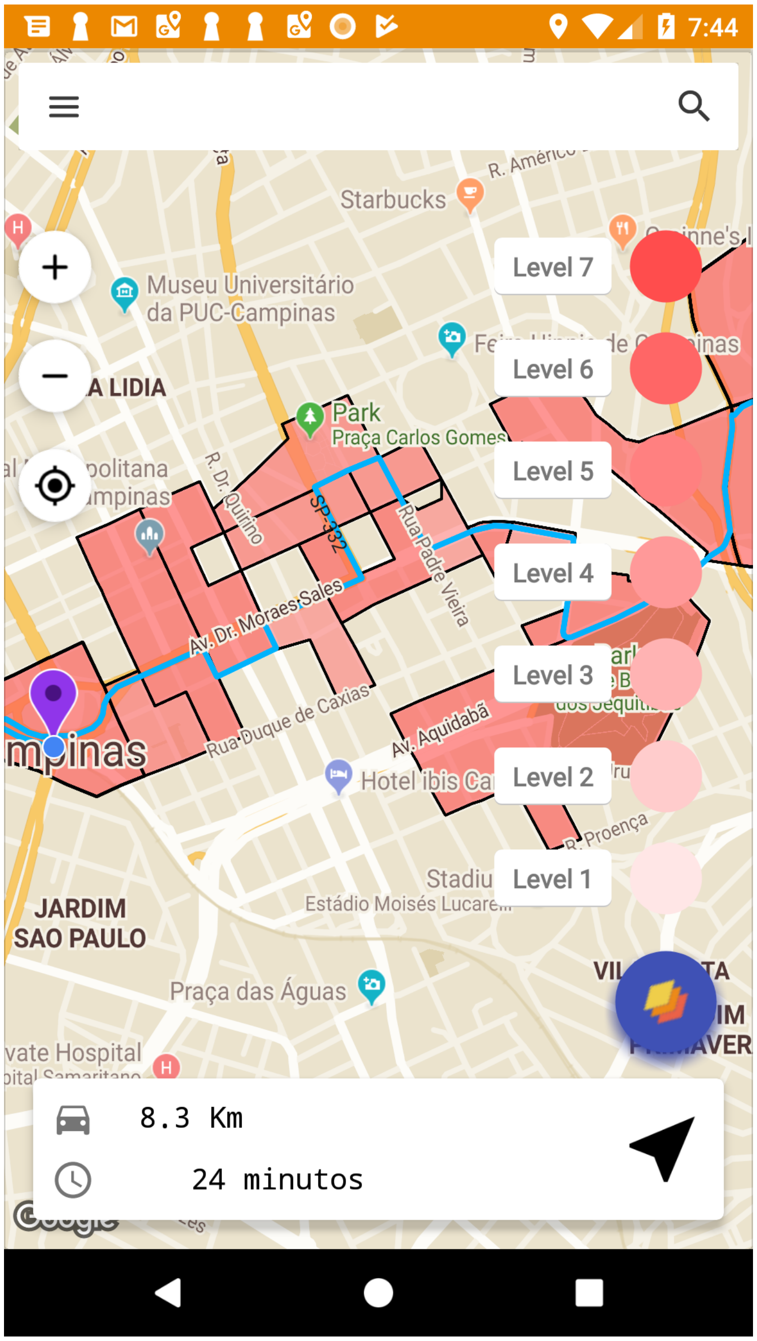


Figure 8: Main screen.

The application contains more than one type of map layer available for visualization. The user can choose between default, satellite and terrain, and can also choose to display traffic data.

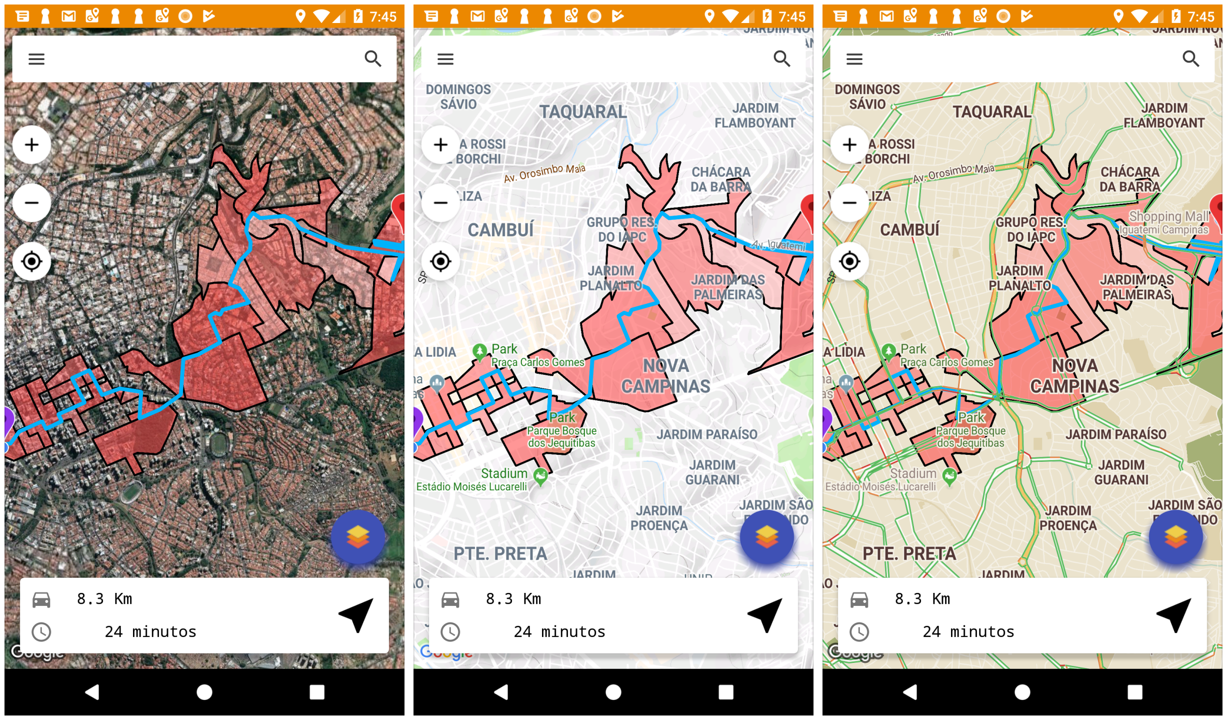


Figure 9: Maps Layers from left to right: Satellite, Terrain, Traffic Data

4.3 Future Work

The application can be improved. For example, in terms of user experience, the app could have voice navigation built in itself so the user wouldn't need to leave the app for this. The limitation for this at the time of the development was that Google Maps API did not provide its navigation features outside Google Maps app, so it would be necessary this part from scratch, which was not the focus of this work. Another way of improving the current work is to offer directions not just for cars, but also for walking, cycling or by transit.

It is also worth to consider that this application was developed with criminal data just from Campinas. Future works could expand it to other big cities of Brazil.

Regarding the algorithmic calculation of the safest route, it could be enhanced considering a more macroscopic detour of the normal route. In the current state of the application, some diversions generated are so small that they don't make a significant difference in the safety of the route.

5 Conclusion

The goal of this work was to address the problem of generating not only fast routes, but routes that are safe. We were able to create this version of the application by collaborating with Senso Criminal for getting information about criminal rate of almost two thousand sectors on the city of Campinas. The safe routes generated had some detours that did not result in significant safety improvements, but the general results obtained were satisfactory.

Utilizing Dijkstra Algorithm for shortest paths, together with the information cited above, and Google Maps Directions API routes it was able to show the user routes that were safe, fast and more adapted to the reality of the major cities of Brazil.

References

[1] Google Maps API. Online:
`https://developers.google.com/maps/documentation/android-sdk/intro?hl=pt-br.`. Access in : June 17th 2018.

[2] Ray-casting Algorithm. Online:
`https://rosettacode.org/wiki/Ray-casting_algorithm`. Access in : June 17th 2018.

[3] Souza, Allan M. de et al. Por Aqui é Mais Seguro: Melhorando a Mobilidade e a Segurança nas Vias Urbanas. Simpósio Brasileiro de Redes de Computadores (SBRC), [S.l.], v. 36, may 2018. Available in:
`http://portaldeconteudo.sbc.org.br/index.php/sbrc/article/view/2475`. Access in : June 17th 2018.

[4] Shah, S., Bao, F., Lu, C.-T., and Chen, I.-R. (2011). Crowdsafe: Crowd sourcing of crime incidents and safe routing on mobile devices. In Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, GIS '11, pages 521–524, New York, NY, USA. ACM.

[5] Galbrun, E., Pelechrinis, K., and Terzi, E. (2016). Urban navigation beyond shortest route. Inf. Syst., 57(C):160–171.