ESTRUCTURA DE DATOS 1
Código ST0245

Laboratory practice No. 5: Graphs

Miguel Ángel Correa Manrique
Universidad Eafit
Medellín, Colombia
macorream@eafit.edu.co

Pablo Buitrago Jaramillo
Universidad Eafit
Medellín, Colombia
pbuitragoj@eafit.edu.co

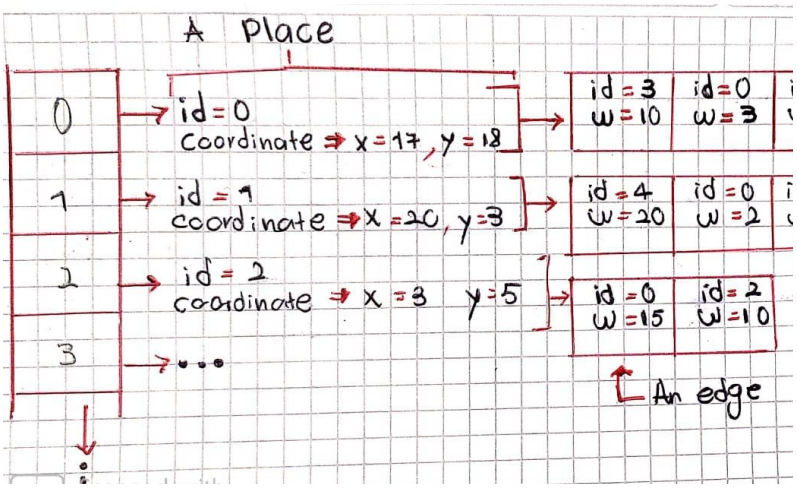
3) Practice for final project defense presentation

3.1

- **Python Implementation:**

The python code reads has to read files using urllib.request. The structure used in this code is based in dict() -dictionaries-, one dict for all the vertex hat the user wants to add given the ID (key), x coordinate, y coordinate and the name of the vertex (value); another dict() for the arcs between the vertex that has as key the arc name, and the initial and final vertex, and its length as value. We used dicts because its easy and comfortable implementation, its time complexity is O(n) for access and O(1) to add

- **C++ Implementation:**



By the time that each place is read, these will have redefine another id, starting with 0 the first one read and finishing in n, for the last one.

As it is known later the connections are given, a temporary hash map will be storing as key the id that they had before, and as its value the new id assigned while reading; this is done because of the fact that when a connection is read it will be stored as an edge to each corresponding place in the cityMap data structure.

PhD. Mauricio Toro Bermúdez
Professor | School of Engineering | Informatics and Systems
Email: mtorobe@eafit.edu.co | Office: Building 19 – 627
Phone: (+57) (4) 261 95 00 Ext. 9473

All that was constructed in that way because is required to save memory as possible, it is going to be explained as follows:

- It was not used adjacent matrices since the graph is not that dense, a city does not have connections from each place to everywhere.

- It was necessary to create a place object because if we were to save each coordinate on each edge (these designating the coordinates of that place where is heading) it would be necessary to do this on each connection, which is not feasible, also if future developers have to use the coordinates while they are iterating over the connections of a place, they just have to access the id that is referred.

The CityMap data structure is defined as in the image, having an array of places that will be placed matching each id already purified.

Note: Sorry for so many lines on C++ implementation description.

3.2 More than adjacent list implementation, would be the size in bytes of each Edge, represented as 'S' (relative as the way it is implemented), in this case is told we have 'N' = 300000, would be N^2 times S.

3.3 As we explained in 3.1 with a temporary hash map table.

Every time that a place is read it will be assigned a new value for the id, starting with 0 to solved the problem until n (the number of places given) as it does that a temporary hash map table will store for every place a key as the previous id, and for its value the new id, then while reading the connections the algorithm will know which id belong to which place.

3.4 A Graph data structure implemented with adjacent matrices, each row represents the places that I have while the columns the places I can go to, places represented with numbers, every place is called 'node'.

As it was told that every node had a way to travel to whichever node, it was thought to build an algorithm to go through every node, starting in one of those given, once it starts the algorithm will paint the node it is on, and painting with the contrary color all the ones connected to that node, if it was already painted but the color the algorithm is going to paint in that moment is not the same is not bipartite.

3.5 The time complexity of this algorithm is $O(n^2)$, since the data structure is implemented with adjacent matrices.

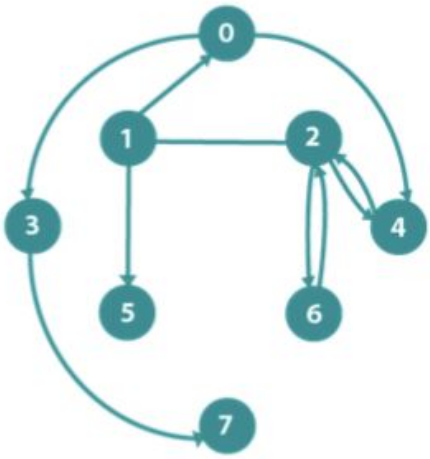
3.6

Variable	Description
n	Number of vertices

PhD. Mauricio Toro Bermúdez
Professor | School of Engineering | Informatics and Systems
Email: mtorobe@eafit.edu.co | Office: Building 19 – 627
Phone: (+57) (4) 261 95 00 Ext. 9473

4) Practice for midterms

4.1



	0	1	2	3	4	5	6	7
0				1	1			
1	✓		✓			✓		
2					✓		✓	
3								✓
4			✓					
5								
6			✓					
7								

4.2

0 ->[3,4]
1 ->[0, 2, 5]
2 ->[4, 6]
3 ->[7]
4 ->[2]
5 ->[ ]
6 ->[2]
7 ->[ ]

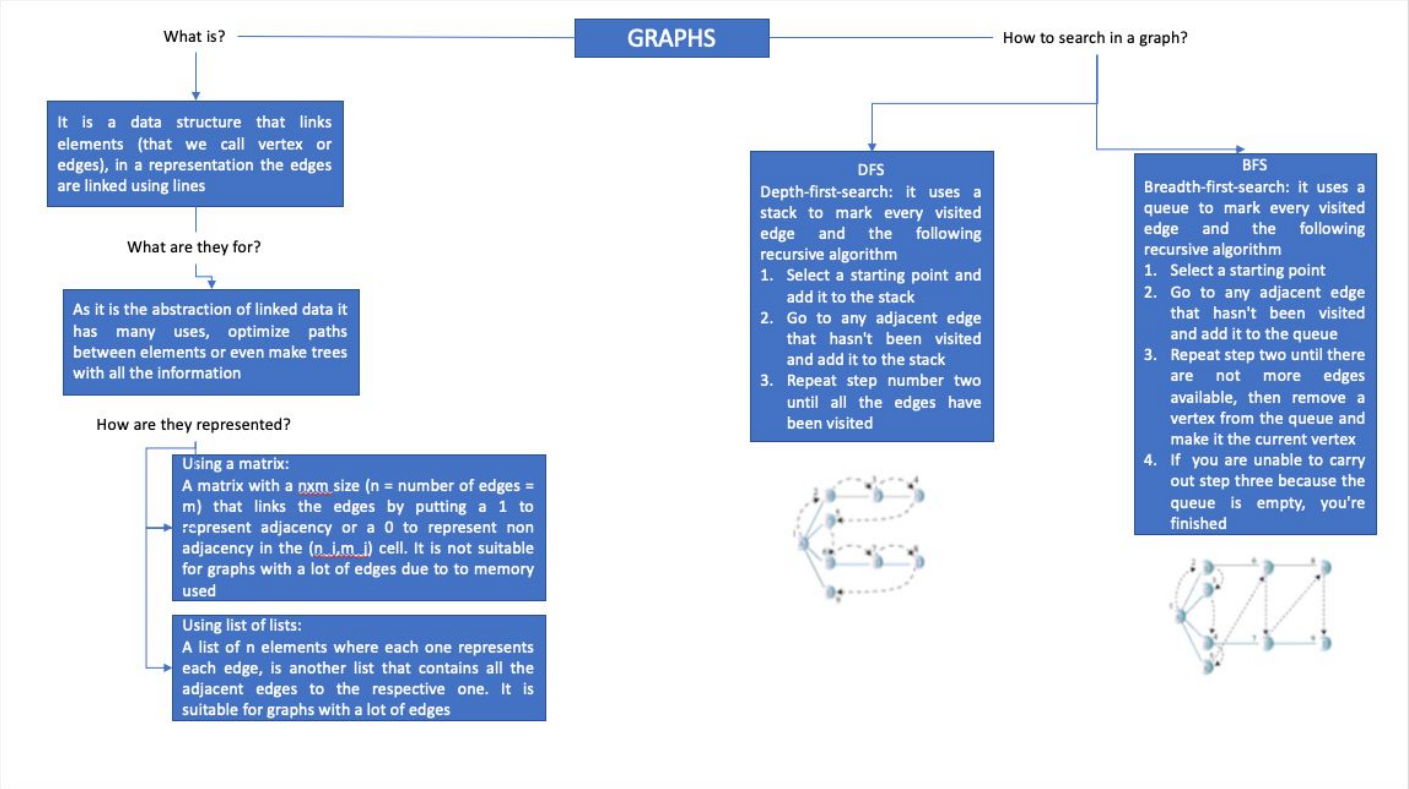
4.3 b

4.4

***4.4.1** ii*
***4.4.2** i*

5) Recommended reading (optional)

PhD. Mauricio Toro Bermúdez
Professor | School of Engineering | Informatics and Systems
Email: mtorobe@eafit.edu.co | Office: Building 19 – 627
Phone: (+57) (4) 261 95 00 Ext. 9473



Mapa conceptual

6) Team work and gradual progress (optional)

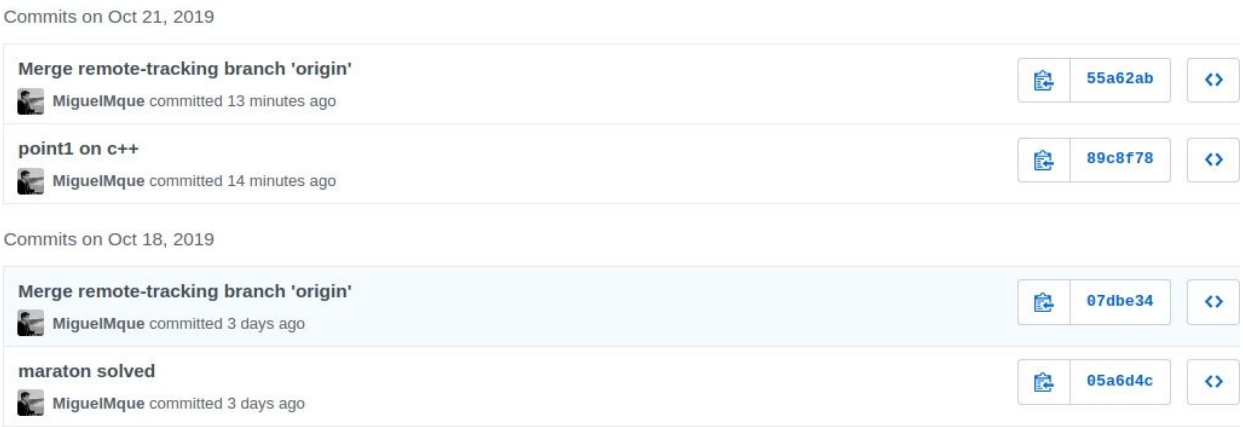
6.1 Meeting minutes

Member	Date	Done	Doing	To do
Pablo	2019/10/15	Worksheet point 5	Worksheet point 1	Worksheet point 1, 3, 4, 6
Pablo	2019/10/15	Point 1 Python version	Worksheet point 4	
Miguel	2019/10/18	Worksheet point 2		Worksheet point 1, 3, 6

PhD. Mauricio Toro Bermúdez
Professor | School of Engineering | Informatics and Systems
Email: mtorobe@eafit.edu.co | Office: Building 19 – 627
Phone: (+57) (4) 261 95 00 Ext. 9473

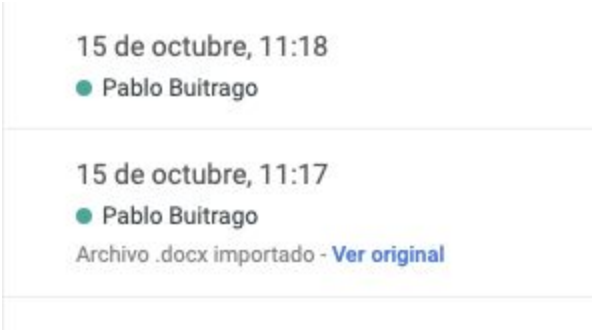
Miguel	2019/10/21		Worksheet point 1	Worksheet point 1, 3, 6
Miguel	2019/10/21	Point 1 C++ version	Point 3	Worksheet point
Pablo	2019/10/21	Worksheet point 4	Worksheet point 3	Worksheet point 6
Pablo	2019/10/21	Worksheet point 3	Worksheet point 6	

6.2 History of changes of the code

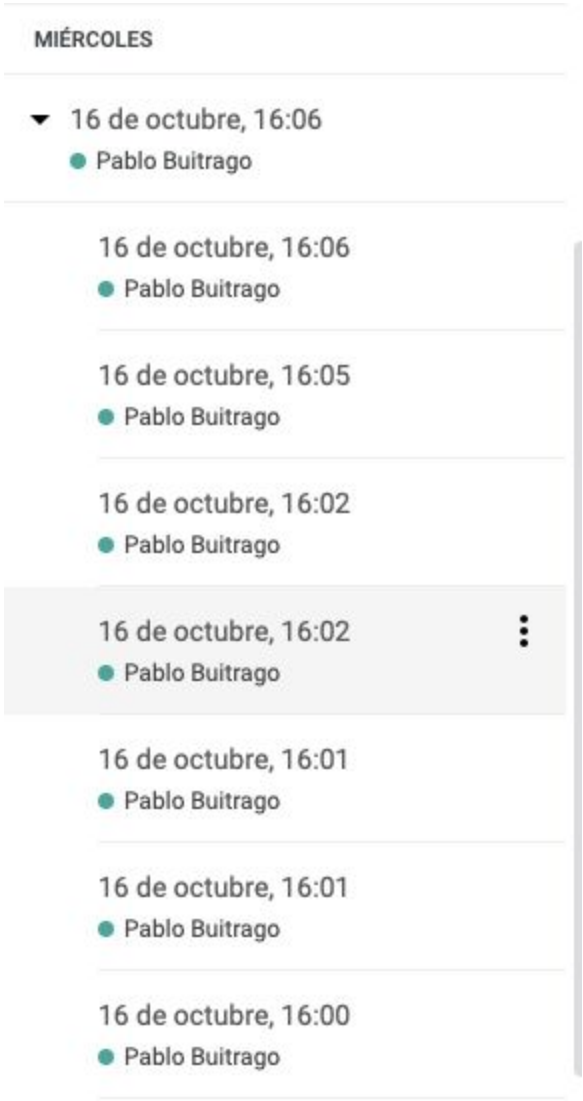


PhD. Mauricio Toro Bermúdez
Professor | School of Engineering | Informatics and Systems
Email: mtorobe@eafit.edu.co | Office: Building 19 – 627
Phone: (+57) (4) 261 95 00 Ext. 9473

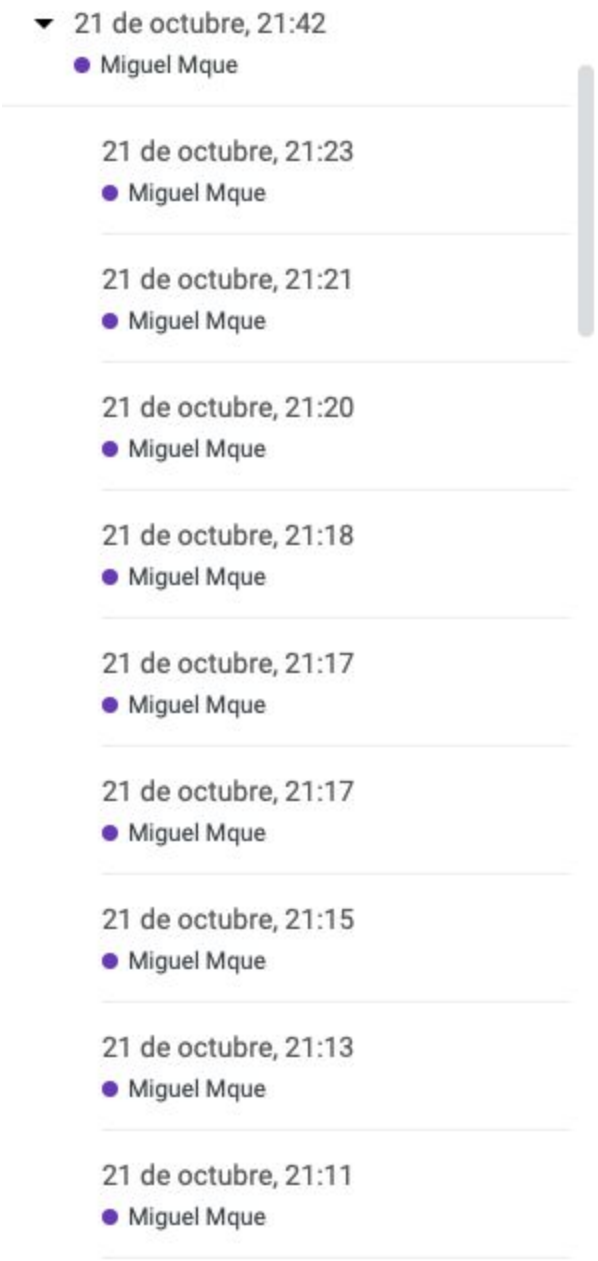
6.3 History of changes of the report



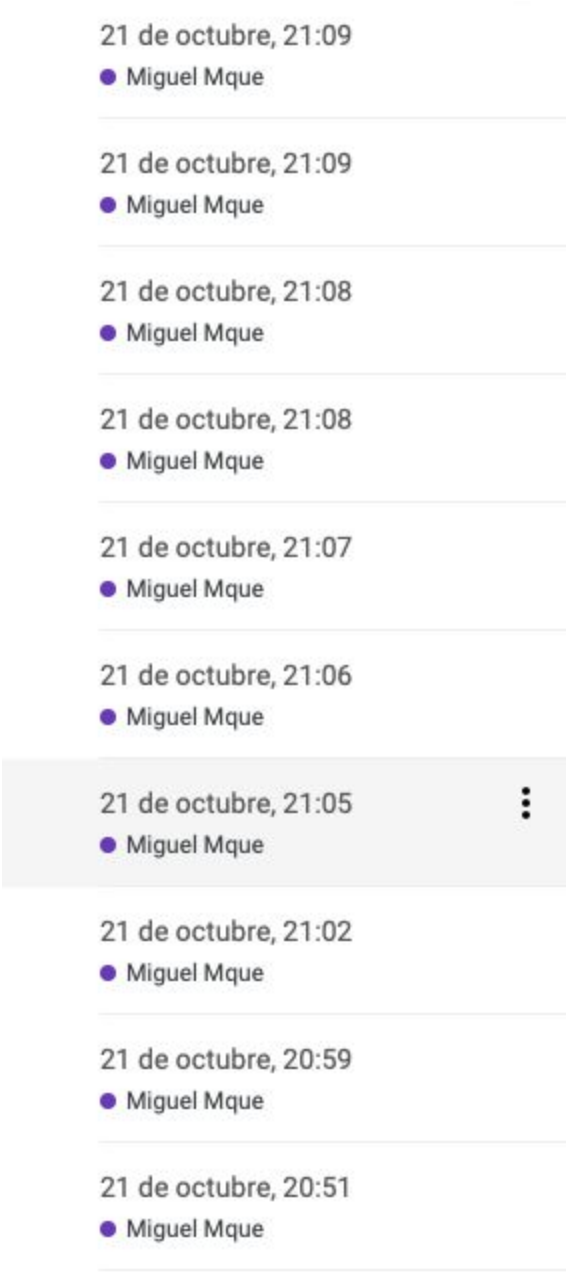
PhD. Mauricio Toro Bermúdez
Professor | School of Engineering | Informatics and Systems
Email: mtorobe@eafit.edu.co | Office: Building 19 – 627
Phone: (+57) (4) 261 95 00 Ext. 9473



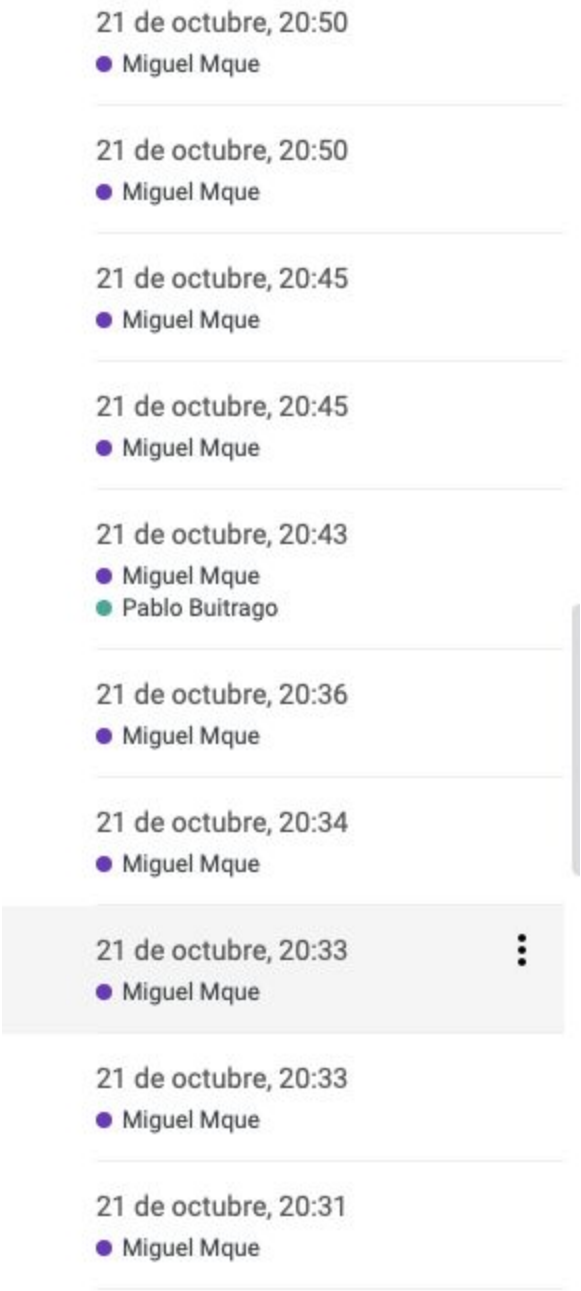
PhD. Mauricio Toro Bermúdez
Professor | School of Engineering | Informatics and Systems
Email: mtorobe@eafit.edu.co | Office: Building 19 – 627
Phone: (+57) (4) 261 95 00 Ext. 9473



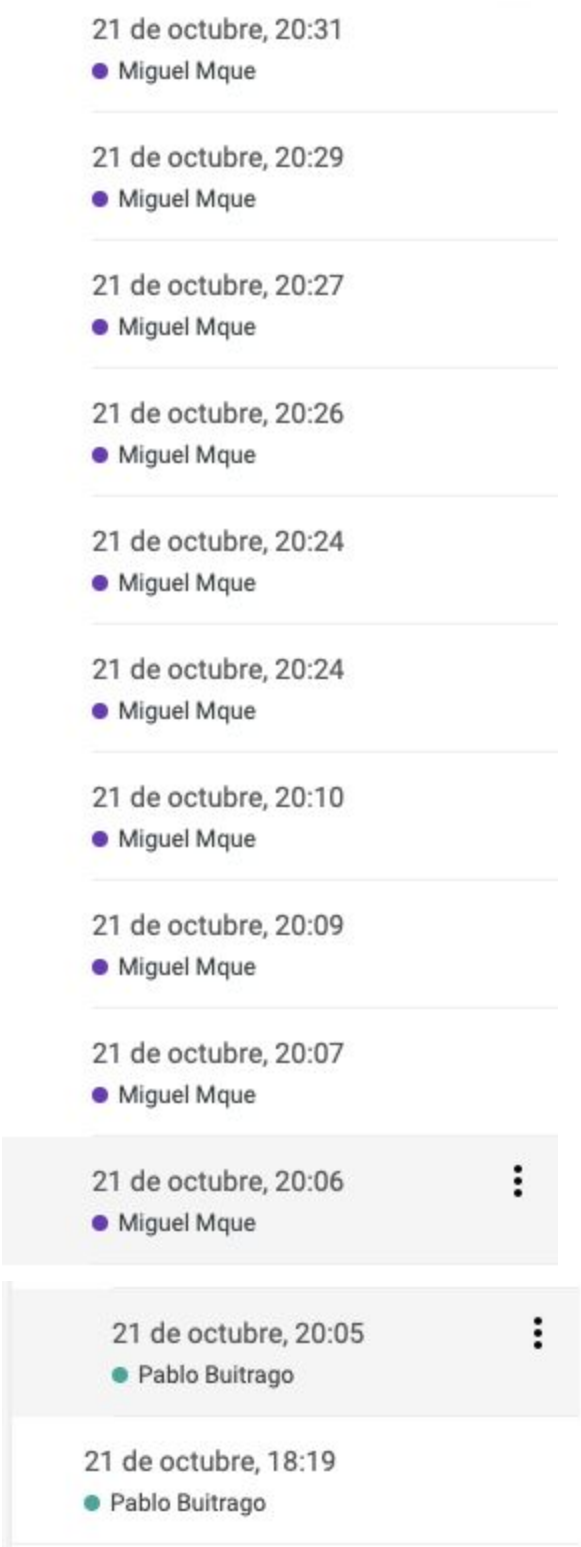
PhD. Mauricio Toro Bermúdez
Professor | School of Engineering | Informatics and Systems
Email: mtorobe@eafit.edu.co | Office: Building 19 – 627
Phone: (+57) (4) 261 95 00 Ext. 9473



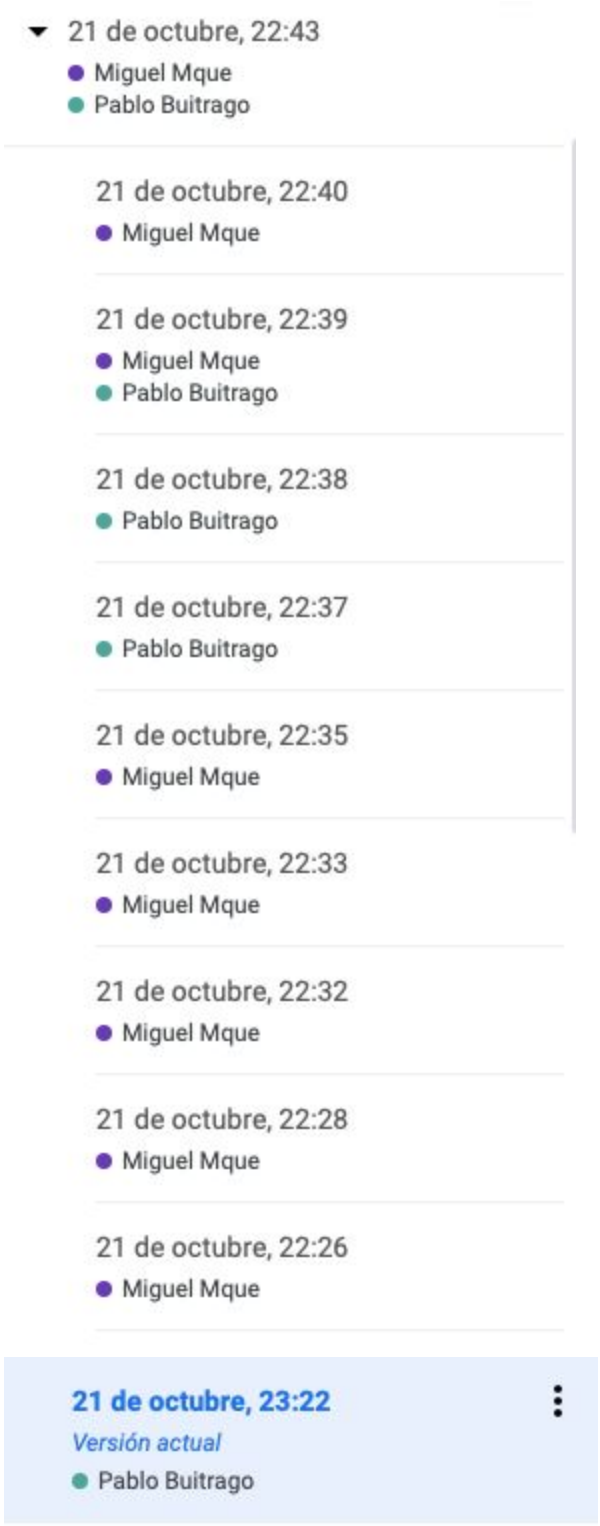
PhD. Mauricio Toro Bermúdez
Professor | School of Engineering | Informatics and Systems
Email: mtorobe@eafit.edu.co | Office: Building 19 – 627
Phone: (+57) (4) 261 95 00 Ext. 9473



PhD. Mauricio Toro Bermúdez
Professor | School of Engineering | Informatics and Systems
Email: mtorobe@eafit.edu.co | Office: Building 19 – 627
Phone: (+57) (4) 261 95 00 Ext. 9473



PhD. Mauricio Toro Bermúdez
Professor | School of Engineering | Informatics and Systems
Email: mtorobe@eafit.edu.co | Office: Building 19 – 627
Phone: (+57) (4) 261 95 00 Ext. 9473



PhD. Mauricio Toro Bermúdez
Professor | School of Engineering | Informatics and Systems
Email: mtorobe@eafit.edu.co | Office: Building 19 – 627
Phone: (+57) (4) 261 95 00 Ext. 9473