



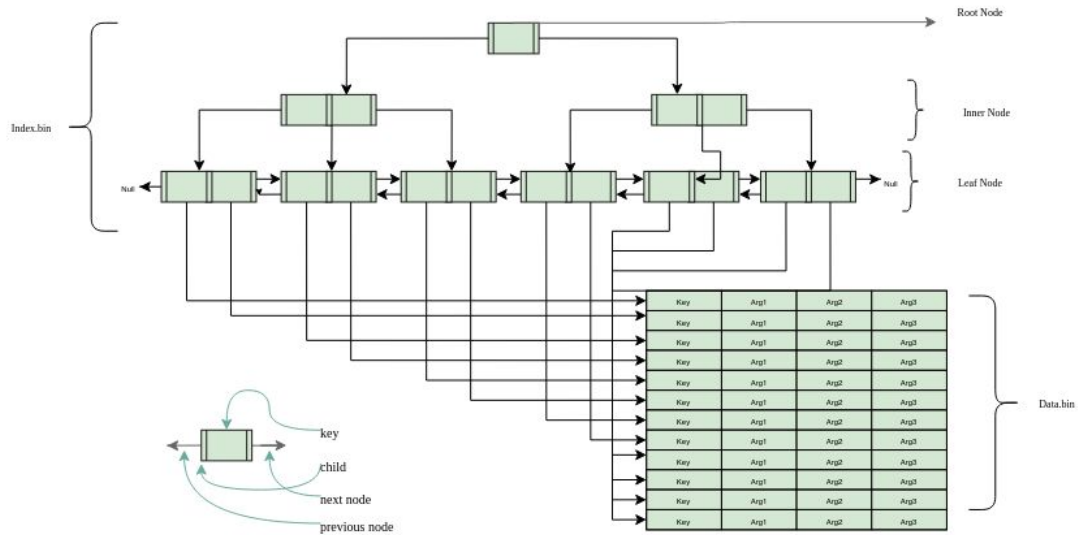
# B+Tree & Static Hashing

Integrantes:

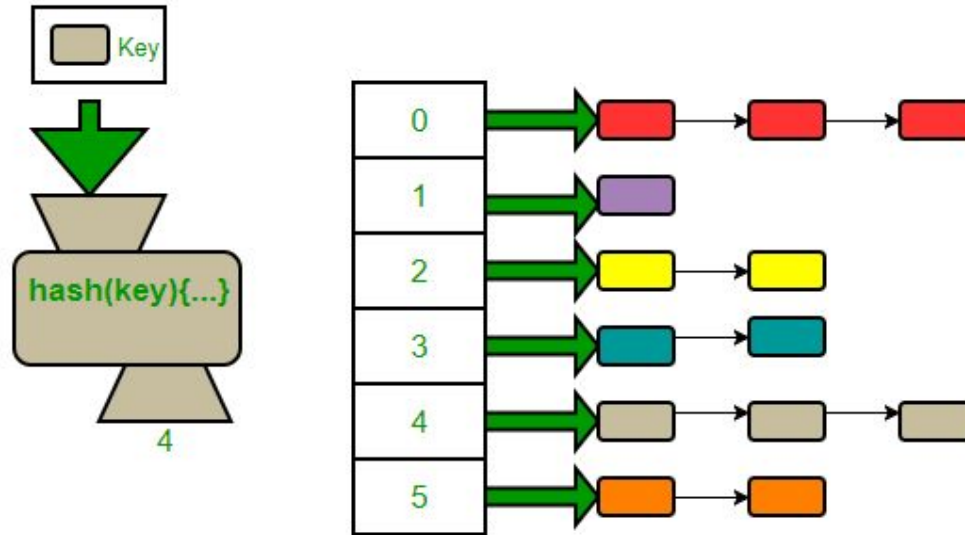
- Giordano Alvitez
- Juan Vargas
- Roosevelt Ubaldo

# B+Tree

B+Tree Structure



# Static Hashing



Using B+Tree Index				
Iteration	Time (ms)			
	Insert 1k	Insert 10k	Insert 100k	Insert 1M
1	40	506	5754	62000
2	38	544	5442	63556
3	42	592	5248	58902
4	46	486	4986	56574
5	42	469	4745	58336
Average	41,6	519,4	5235	59873,6

Static Hashing				
Iteration	Time (ms)			
	Insert 1k	Insert 10k	Insert 100k	Insert 1M
1	10	97	1542	30363
2	10	103	1128	39649
3	9	113	1207	53470
4	19	80	1308	88453
5	10	100	1466	71911
Average	11,6	98,6	1330,2	56769,2

Using B+Tree Index				
Iteration	Disk Access			
	Find 1k	Find 10k	Find 100k	Find 1M
1	10	13	17	20
2	10	13	17	20
3	10	13	17	20
4	10	13	17	20
5	10	13	17	20
Average	10	13	17	20

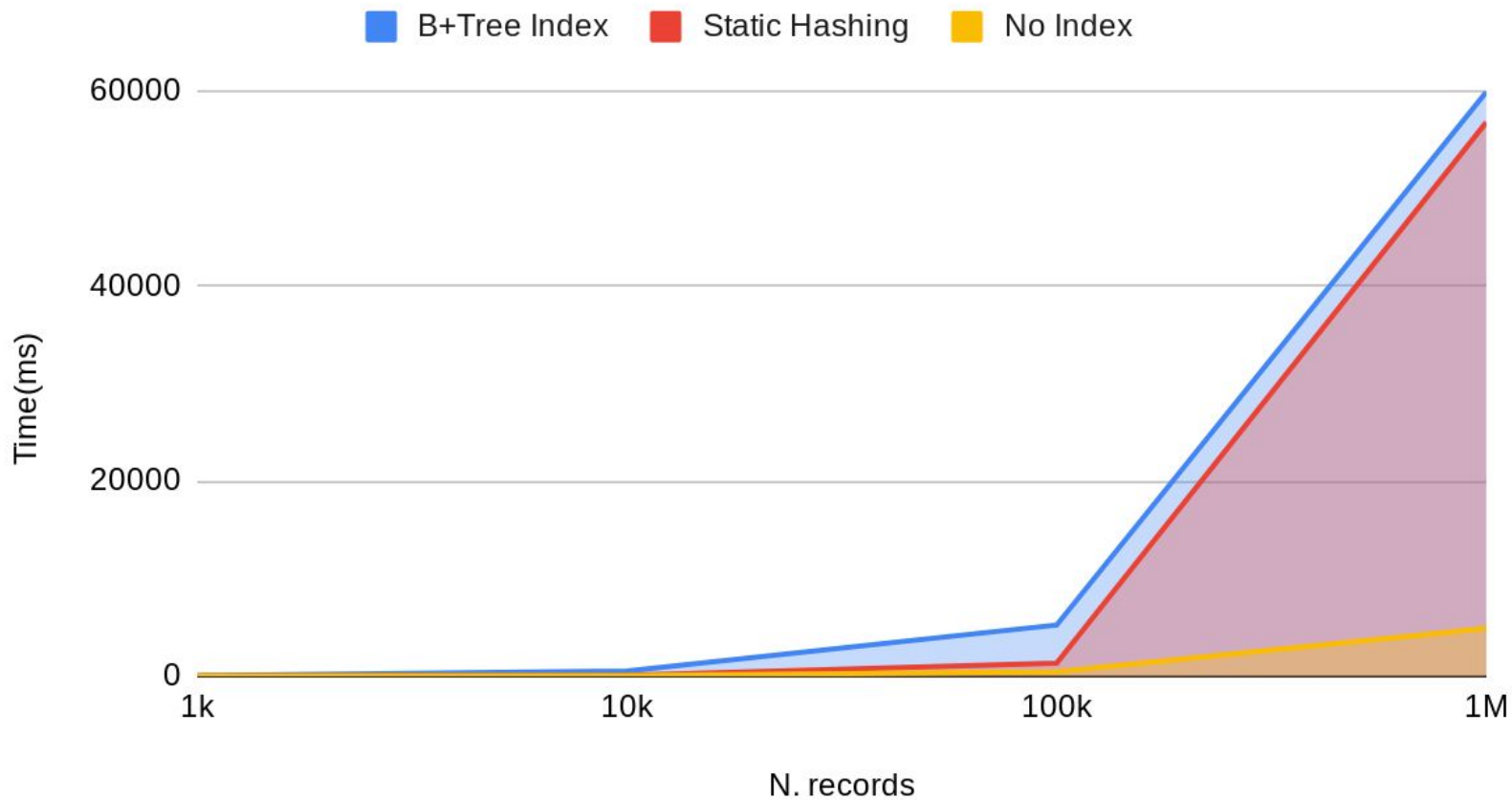
Static Hashing				
Iteration	Disk Access			
	Find 1k	Find 10k	Find 100k	Find 1M
1	2	2	2	4
2	2	2	2	4
3	2	2	2	4
4	2	2	2	4
5	2	2	2	4
Average	2	2	2	4

Summary of Insertion			
Size of Data	B+Tree Index	Static Hashing	No Index
1k	41,6	11,6	4,2
10k	519,4	98,6	45,8
100k	5235	1330,2	450,4
1M	59873,6	56769,2	4909

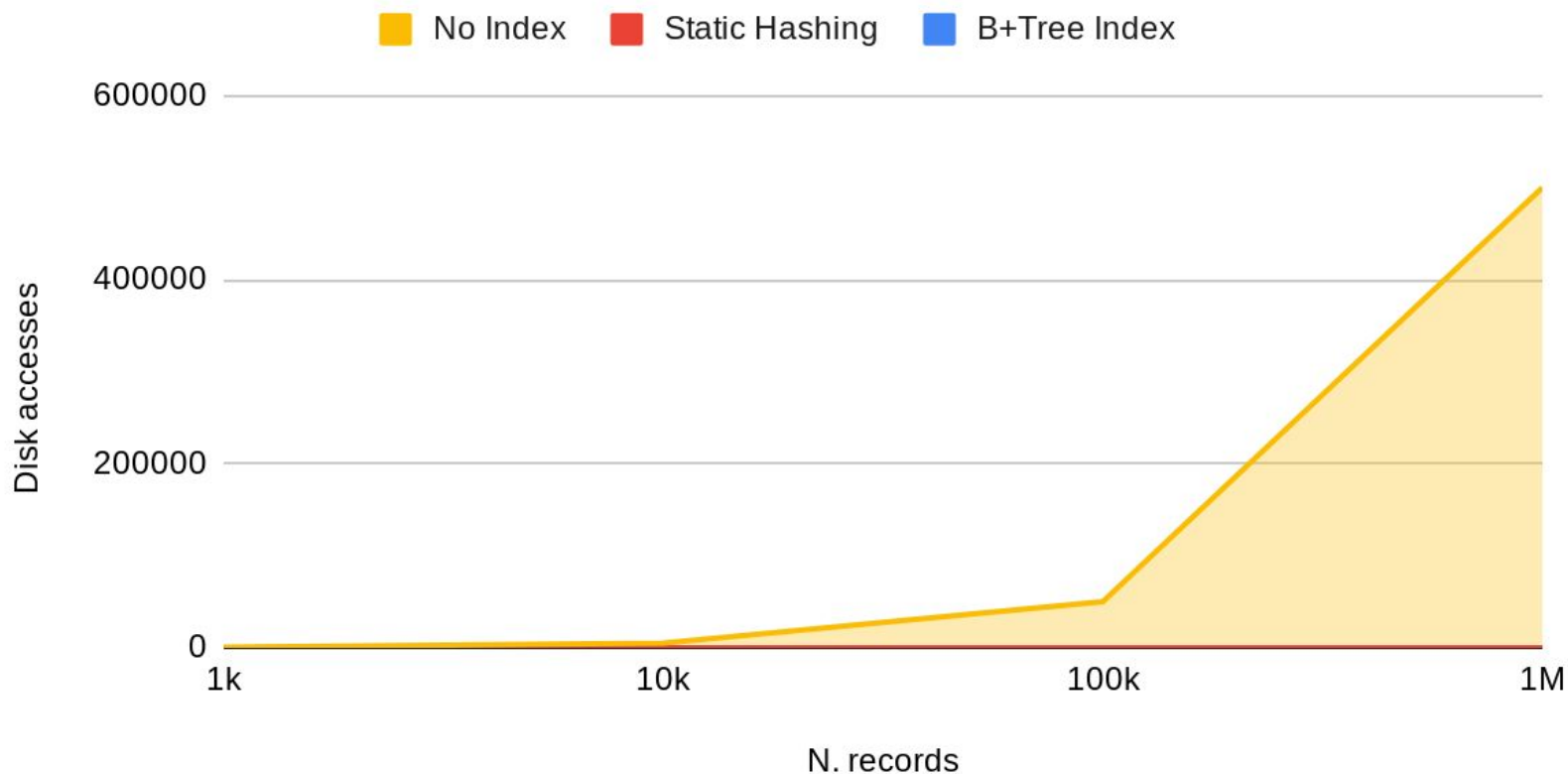
Summary of find time			
N records	B+Tree Index	Static Hashing	No Index
1k	0	0	1,8
10k	1	0	14,4
100k	0	0	133,6
1M	0	0	1538,2

Summary of find disk accesses			
Size of Data	B+Tree Index	Static Hashing	No Index
1k	10	2	500
10k	13	2	5000
100k	17	2	50000
1M	20	2	500000

# Inserting records with and without indexes

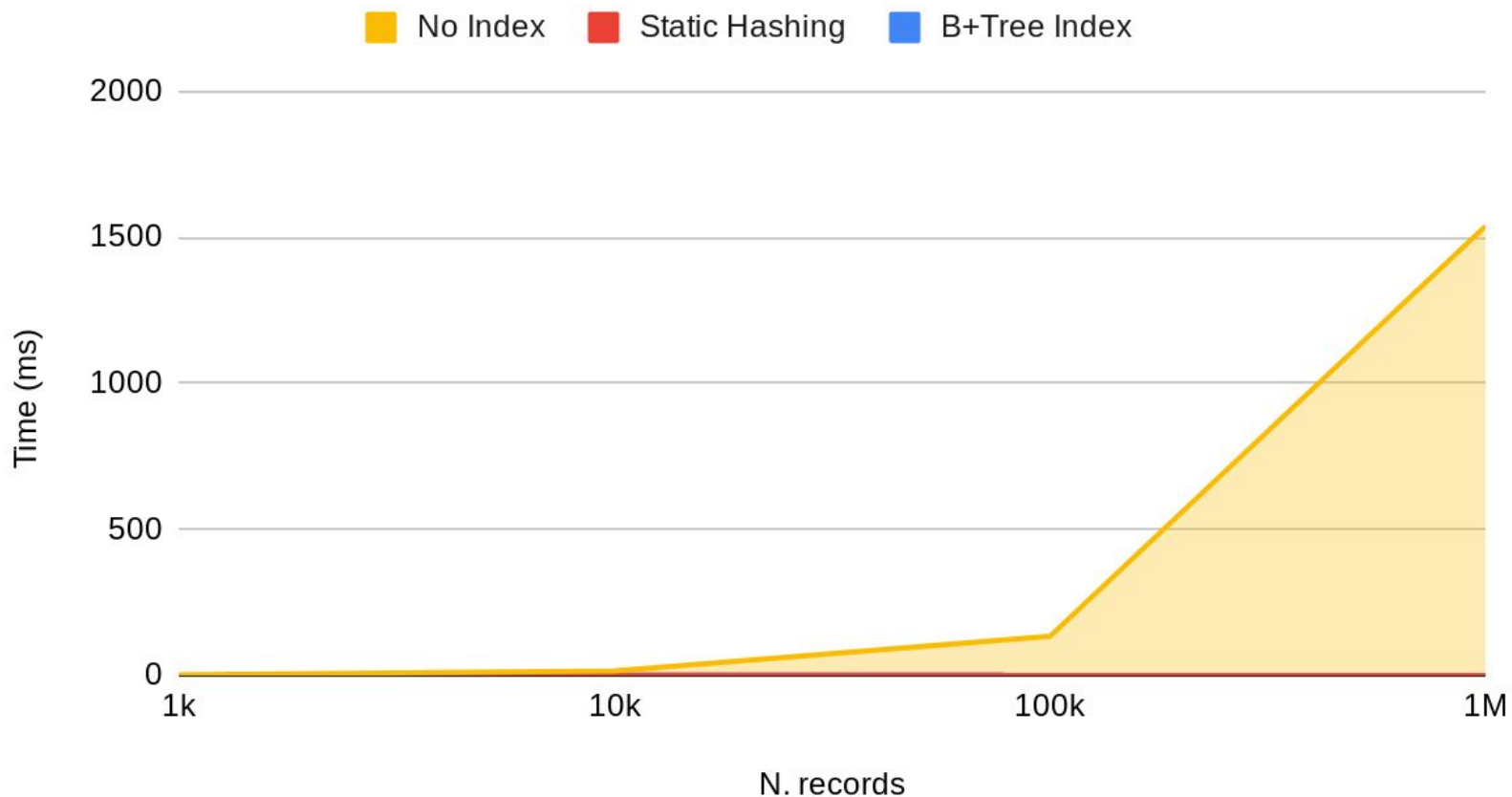


# Disk access on find a record by key value with and without indexes





## Find a record by key value with and without indexes



## Insert 1k, Insert 10k, Insert 100k y Insert 1M

