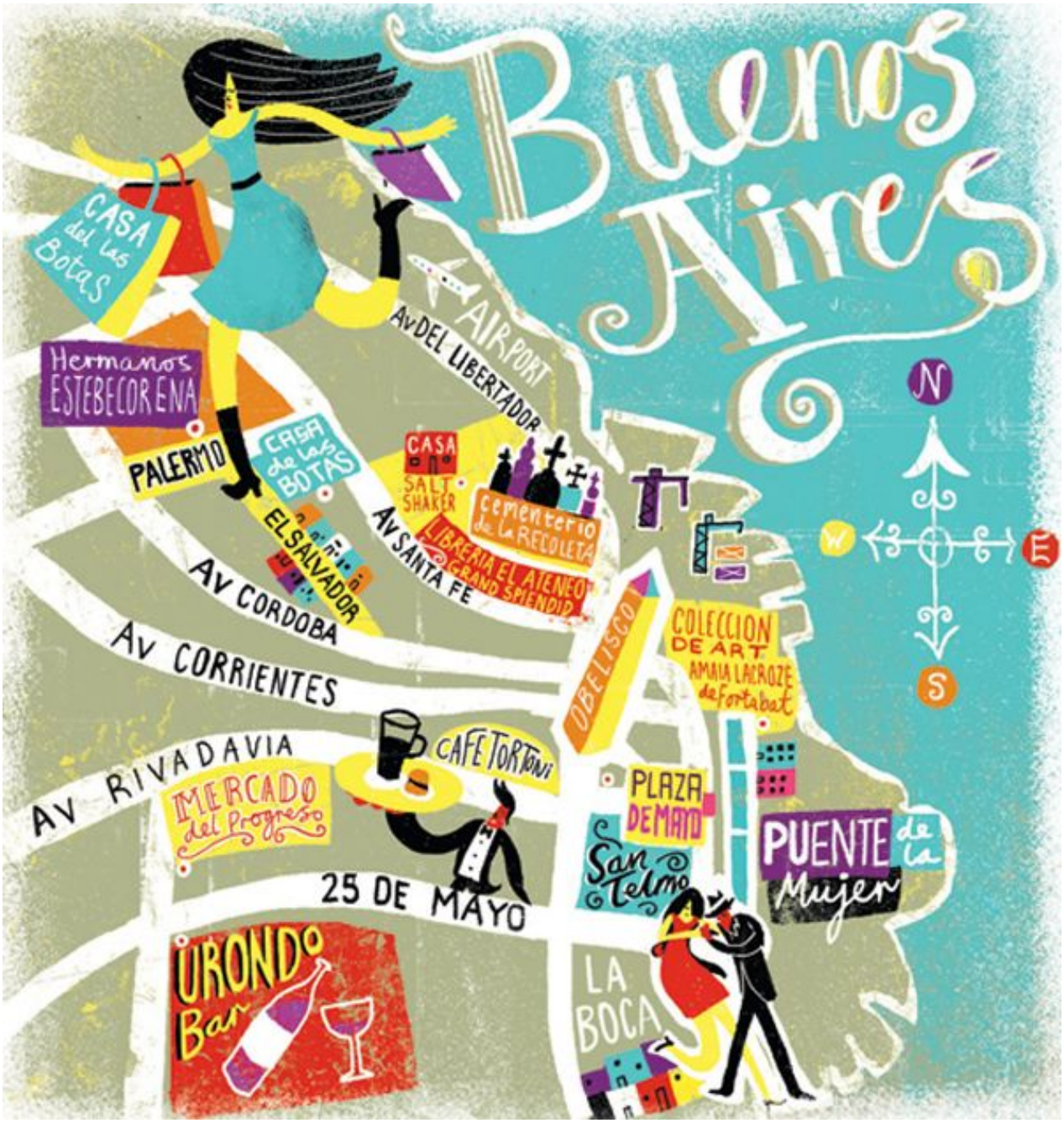
LAB Buenos Aires Office - Technical test



Introduction

Given the following list of restaurants stored in JSON format:

```
{
    "restaurants": [
        {
            "id": 1,
            "name": "La nueva panaderia",
            "rating": 5.0,
```

```
            "coor_x": 1.0,
            "coor_y": 1.0
        },
        {
            "id": 2,
            "name": "Mi Buenos Aires Parrilla",
            "rating": 2.5,
            "coor_x": -5.0,
            "coor_y": 2.0
        },
        {
            "id": 3,
            "name": "Sabores propios",
            "rating": 1.5,
            "coor_x": 1.0,
            "coor_y": 5.0
        },
        {
            "id": 4,
            "name": "Cafe del mar",
            "rating": 4.5,
            "coor_x": -2.0,
            "coor_y": 3.0
        },
        {
            "id": 5,
            "name": "Tony & Company",
            "rating": 3.5,
            "coor_x": 4.0,
            "coor_y": -5.0
        },
        {
            "id": 6,
            "name": "Green life",
            "rating": 5.0,
            "coor_x": -5.0,
            "coor_y": -4.0
        }
    ]
}
```


The following table shows the distribution in a two-dimension plane:

						..						
						0;5	R3					
						0;4						
	R2			R4		**0;3**						
						0;2						
						0;1	**R1**					
-6;0	**-5;0**	**-4;0**	**-3;0**	**-2;0**	**-1;0**	**0;0**	**1;0**	**2;0**	**3;0**	**4;0**	**5;0**	**6;0**
						0;-1						
						0;-2						
						0;-3						
	R6					**0;-4**						
						0;-5				R5		
						..						

Tasks

You will create a repository using your personal Github account to store your work. When you finished the tasks, just add `@pciruzzi` and `@agdiaz` as collaborators and we will check your proposed solution. We suggest to develop each task in a branch, then create the pull request and finally merge to master. That will help to review your solution.

Very important: **Due date: One week after you read this message.**

Configure your REST API

We want to create a Node.js application which will start a REST API. Feel free to use a library such as Express.js or Hapi.js. You can validate that everything is alright using Google Chrome (or other web browser), Postman or writing Curl commands in your terminal.

After you set up the REST API we will expect to make requests to:

```
GET http://localhost:<port>/api/
```

And receive a message saying "Everything works fine. There are restaurants!". Based in the given example the message should be:

```
Everything works fine. There are 6 restaurants!
```

Getting restaurants near me

We want to start using your application to help users to find where go to dine out!

Now, your API should be able to respond messages with the next format:

```
GET http://localhost:<port>/api/search?coor_x=<number>&coor_y=
<number>&results=<number>
```

As you see, there are three parameters:

1. `coor_x`: it's the position in the "x" axis
2. `coor_y`: it's the position in the "y" axis
3. `results`: it's the count of restaurants that the user wants to get

We suggest to measure distances between restaurants using the Euclidean distance formula.

For instance, executing this request:

```
GET  http://localhost:<port>/api/search?coor_x=-4&coor_y=3&results=2
```

Should retrieve a JSON response like the next one:

```
{
    "search_result": {
        "criteria": "",
        "count": 2,
        "restaurants": [
          {
              "id": 2,
              "name": "Mi Buenos Aires Parrilla",
              "rating": 2.5,
              "coor_x": -5.0,
              "coor_y": 2.0
          },
          {
              "id": 4,
              "name": "Cafe del mar",
              "rating": 4.5,
              "coor_x": -2.0,
              "coor_y": 3.0
          }
      ]
    }
}
```

						..						
						0;5	R3					
						0;4						
	R2	X		R4		0;3						
						0;2						
						0;1	R1					
-6;0	-5;0	-4;0	-3;0	-2;0	-1;0	0;0	1;0	2;0	3;0	4;0	5;0	6;0
						0;-1						
						0;-2						
						0;-3						
	R6					0;-4						
						0;-5				R5		
						..						

Be more helpful!

Now, we want to apply some orders in the results near the client before retrieve them. We suggest two criteria: alphabetical order and sorted by rating:

Alphabetical order:

```
GET http://localhost:<port>/api/search?criteria=name&coor_x=<number>&coor_y=
<number>&results=<number>
```

```
{
    "search_result": {
        "criteria": "name",
        "count": 2,
        "restaurants": [
          {
              "id": 4,
              "name": "Cafe del mar",
              "rating": 4.5,
              "coor_x": -2.0,
              "coor_y": 3.0
          },
          {
              "id": 2,
              "name": "Mi Buenos Aires Parrilla",
              "rating": 2.5,
              "coor_x": -5.0,
              "coor_y": 2.0
          }
      ]
    }
}
```

Rating order:

```
GET http://localhost:<port>/api/search?criteria=rating&coor_x=<number>&coor_y=<number>&results=<number>
```

```
{
    "search_result": {
        "criteria": "rating",
        "count": 2,
        "restaurants": [
          {
              "id": 4,
              "name": "Cafe del mar",
              "rating": 4.5,
              "coor_x": -2.0,
              "coor_y": 3.0
          },
          {
              "id": 2,
              "name": "Mi Buenos Aires Parrilla",
              "rating": 2.5,
              "coor_x": -5.0,
              "coor_y": 2.0
          }
      ]
    }
}
```

It's important to mention that the endpoint must work without the criteria parameter working as the previous task.

Read from a file

Instead of having the data inside the project itself, it'd be helpful to have it in a separate file which can be read in order to change the restaurant distrubution without having to change the code.

Final thoughts

Document!

Please while you're working on the exercise, don't hesitate to document how to get it running, what choices did you make in terms of design or architecture, and maybe some examples in the README of the project :)

About the complexity of the tasks

This exercise is meant to be as deep as you want. If you consider it is beyond your knowledge or you don't find how to do anything, feel free to leave some pending points, as the idea is to try to make as much as possible, but they are not all mandatory. If you do so, then again try to justify what's going on in that case.

If you have any doubt or anything, you can contact Adrián or Pablo, at addiaz@thefork.com and pciruzzi@thefork.com respectively, and we will try to help you as much as possible. You can also make any assumption you want, as far as it is well documented.

Good luck! 😄