



# IES Project - Trendit

Filipe Gonçalves, 98083

Pedro Lopes, 97827

Vicente Costa, 98515

João Borges, 98155



## The team



Vicente Samuel - Architect



Filipe Gonçalves - Team Manager



Pedro Lopes - DevOps Manager



João Borges - Product Owner



## The concept - A filtered twitter

- A bot which acquires information from twitter according to the interests of the user
- Possibility of creating an account where the user can set interests, so that the main page only presents the respective points of interest
- It can filter tweets around a topic or a location
- Shows statistics about certain topic, such as popularity, amount of tweets done in a period of time, etc





## User Stories - Anabela Flor

An outgoing person, extremely interested in fashion, but with little time to spend

1. Creates an account to cope up with the latest trends
2. While creating an account, puts the topic “fashion” as topics of interests
3. She can search for different topics out of the ones in her main page
4. She can search what is trending on fashion in another location, like australia



## User Stories - Joshua Jo

A leader, free spirit person; leads a company to help children

1. Creates an account for enterprise use, so that he can cope up with what is now popular and what would be the best choice of product to choose
2. Chooses the topics that his company has interest in
3. Can check the statistics of a trending topic, like the amount of tweets throughout time



# User Stories - Alonso Boda

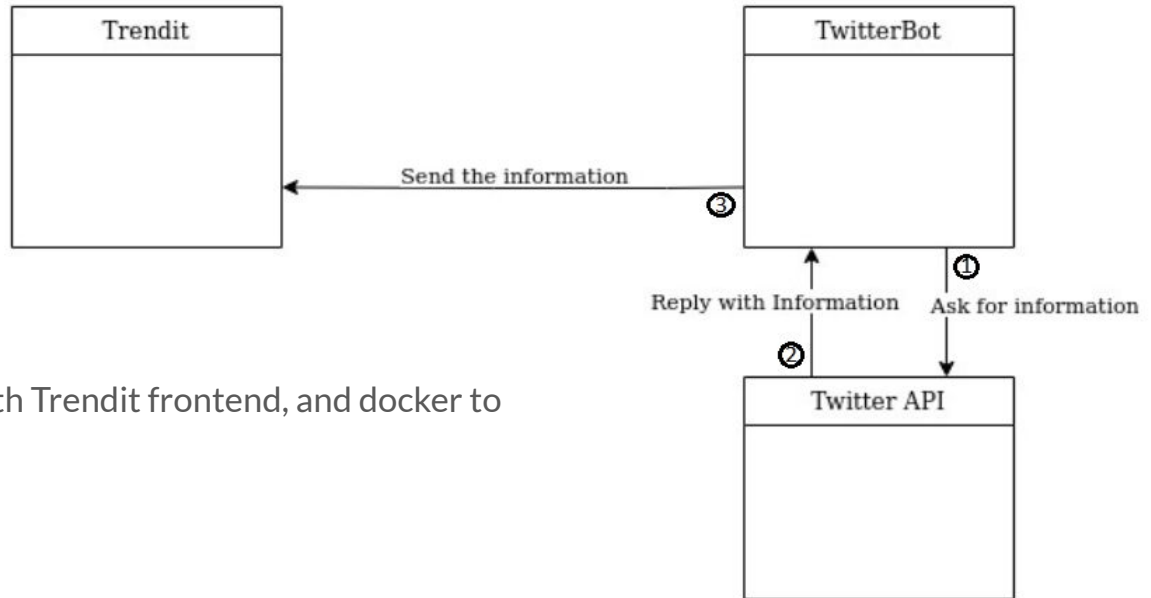
A shy person, but interested in everything

1. To always have an interesting topic to talk with new people, alonso creates an account on TrendIt.
2. Alonso selects all the topics in the register page, so that he is updated with everything that is trending on the world.
3. He can also search what is trending on another countries other than his home country

# Architecture

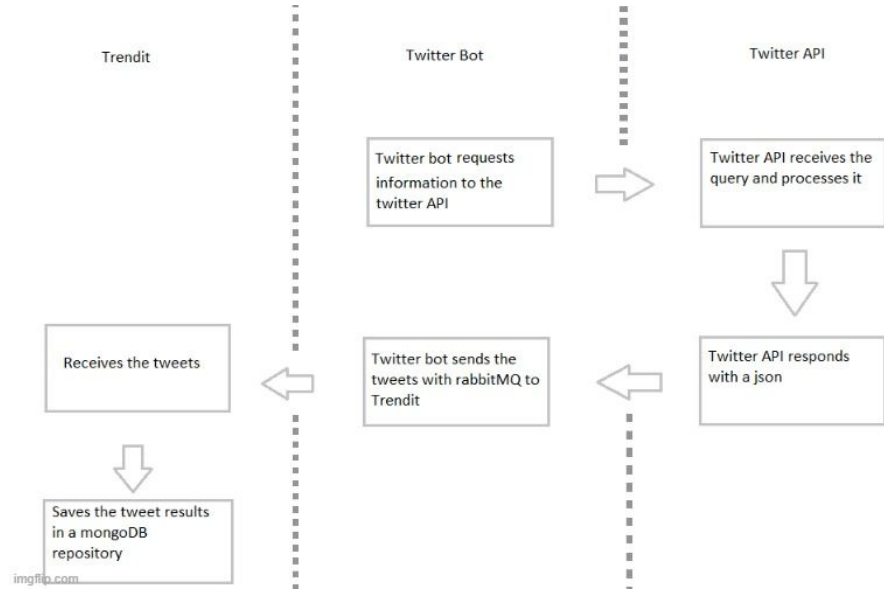
Our project uses Maven so it is easy to initialize our application, and also used MongoDB to store our data.

We also use RabbitMQ to communicate with Trendit frontend, and docker to containerize every service.



# Backend - Twitter Bot & Twitter API

- Twitter Bot - Sends curl requests to the Twitter API
- TwitterAPI - responds with tweets in json format, which will later be converted to java objects with the help of retrofit
- RabbitMQ- Sends the tweets to the Trendit frontend







## Database

Made in mongoDB, saves the tweets searched with a certain query to show in the interface

It also saves trends and a tweet counter for each trend, as well as all the data from each user

We can manipulate the database by using the CRUD methods in the REST Controller on the main application: `/TrendIt/all_users` will post all the users, `/TrendIt/all_tweets` will post all the tweets, etc.





# Containers

We used docker to containerize mongodb and rabbitmq services, and also to containerize the front end and the back end.

It is crucial to have the docker services so that our product works everytime and everywhere, without it there would be a huge amount of time and effort spent in vain.





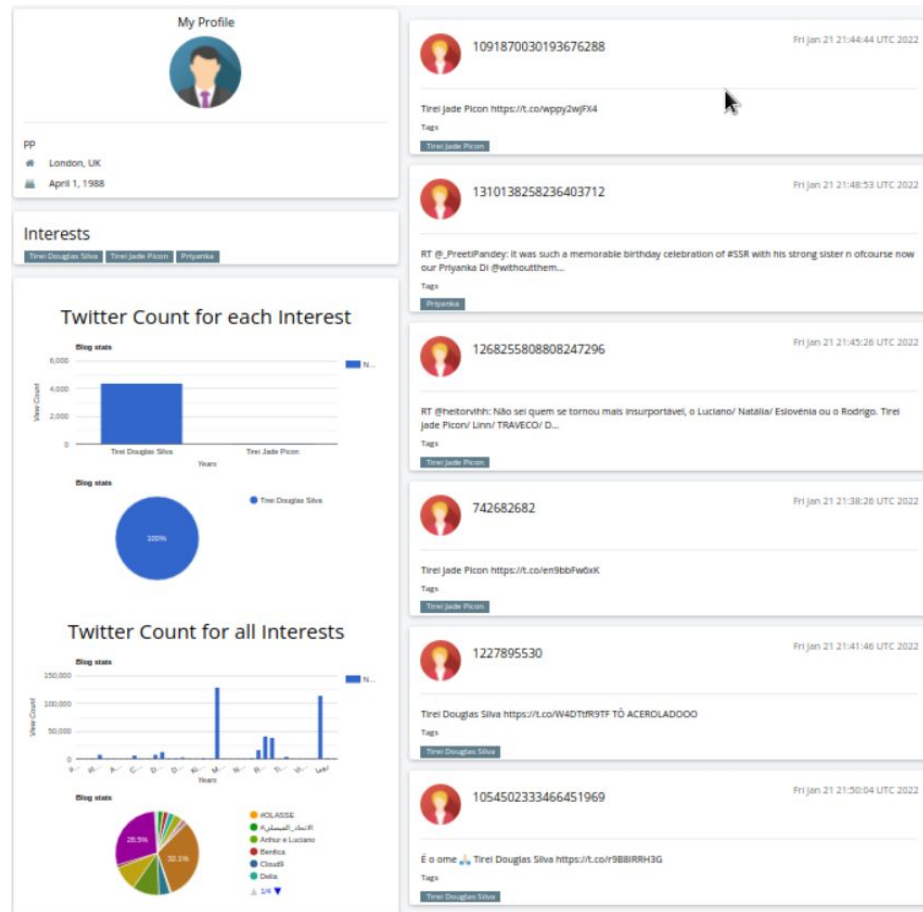
## RabbitMQ

The RabbitMQ is a message broker who serves the purpose of establishing communication between the Trendit and the twitter bot. The twitter API will send the response data in json format; the twitter bot will receive it and proceed to send it with the rabbitMQ service.

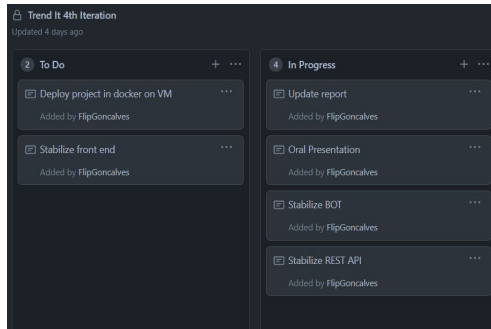


# Frontend - Trendit

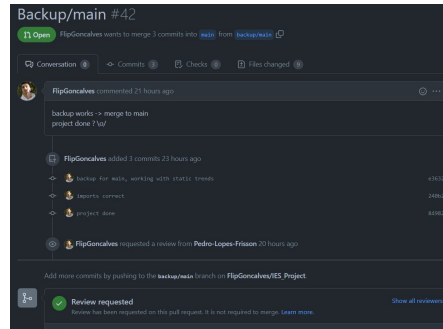
- REST and View Controllers - Allow access to the database to obtain the tweets and users
- Index page
- Register and login
- Home page with: Statistical data on the left, and filtered tweets on the right



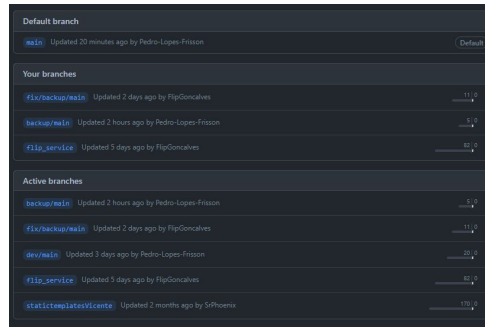
# Github Organization



Scrum Board



Pull Requests, approved by all team



Branching



Webhooks

# Demo

---

## Future Implementations

- Possibility of searching topics
- Better graphs and charts for statistical purposes
- Interest selection inside the home page
- Account settings, so the user can update his stats
- A way to update user trends
- Favourite section for tweets and trends, so the user can save important data

