
Twitter Tracker

Presentazione progetto Team 3

Processo e attività

A cura di Paolo Marzolo

Strumenti utilizzati

Nell'ottica di open-source e self-hosting



IVAR JACOBSON
INTERNATIONAL

sonarqube 



NEW -> APPROVE -> VERIFY -> IN PROGRESS -> READY FOR TEST -> DONE

KANBAN PROGETTO SWE 2020 (TEAM 3)

The user stories in this status hidden by default

NEW 1	WAIT APPROVAL 1	WAIT VERIFY 2	READY 2	IN PROGRESS 1	READY FOR TEST 3	DONE 2	ARCHIVED 11	REJECTED
#51 Time tracking for prod metrics Points 18	#91 Background searches Points 20	#92 Notification on threshold Points 15	#78 Make features REALTIME Points 25	#22 Tweet amount chart by time Points 23	#90 Periodic tweet ● Points 30	#146 BIG interface rework Points 31	#64 Map visualization of users tweets Points 21	
			#80 Not assigned #93 Import tweet collection ● Points 16	#16 Word Cloud from stream Points 17	#153 Make coordinate interactive Points 23		#17 Map visualization of hashtags Points 28	
				#182 Filters on data Points 15			#19 Export Points 16	
							#59 Start Building Database ● Points 17	
							#33 Starting Geolocated Filtered Stream ● Points 20	
							#12 API self-study Points 13	
							#9 SonarQube setup ● Points 16	
							Mattia Guazzaloca Mattia Guazzaloca	

NEW -> READY -> IN PROGRESS -> READY FOR REVIEW -> CLOSED -> ARCHIVED

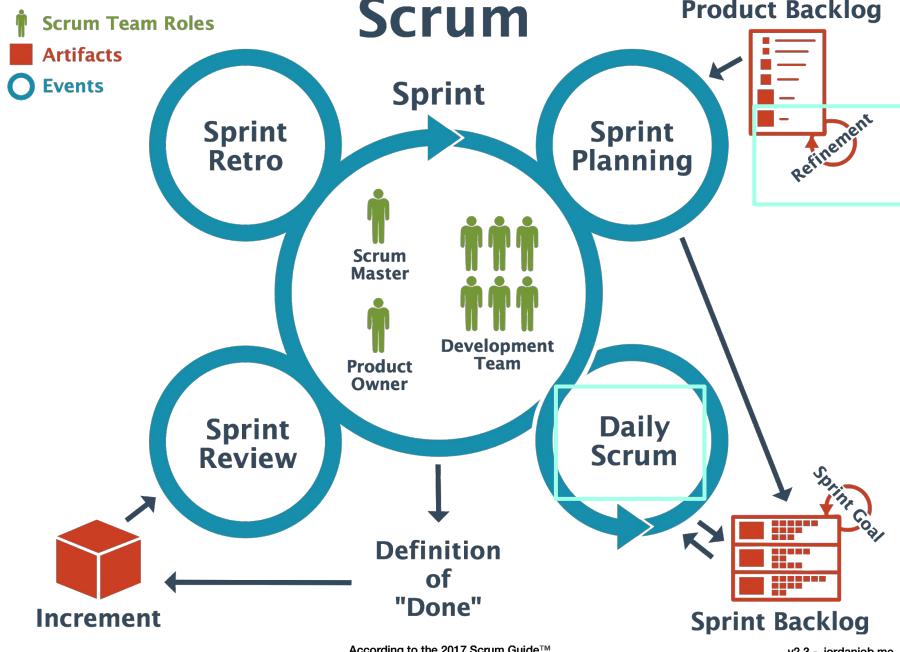
Help

THE PERFECT WORKFLOW IS HERE? PROGETTO SWE 2020 (TEAM 3) 24 NOV 2020-15 DEC 2020

51% ✓ 135 total points 69 completed points | 4 open tasks 32 closed tasks ↗ | 0 cocaine doses

USER STORY	NEW	READY	IN PROGRESS	READY FOR REVIEW	CLOSED
< #92 Notification on threshold + Ready for test 15 points		#117 Add dialog to select threshold 			#115 Add counter for collected tweets #116 Setup mail server
< #51 Time tracking for prod metrics Archived 18 points			#58 Wakatime GitLab Integration 		#53 Opinion on Kimal #52 VS code plugin WAKA
< #21 Import tweet collection + Archived 16 points		#103 Add error boundaries for when the file uploaded is not a JSON or the format of the data is not supported 			#102 Add modular button for import when tweets list is empty or export when there are tweets in the list #113 Handle JSON with wrong syntax
< #16 Word Cloud from stream + Ready for test 17 points			#144 Allow dynamic word amount 		#145 Review style #107 Implement the code by adapting it to the chosen library #106 Search on web the most useful library for the purpose

Evoluzione del processo



Final Activities

Sprint Review



Sprint Retrospective

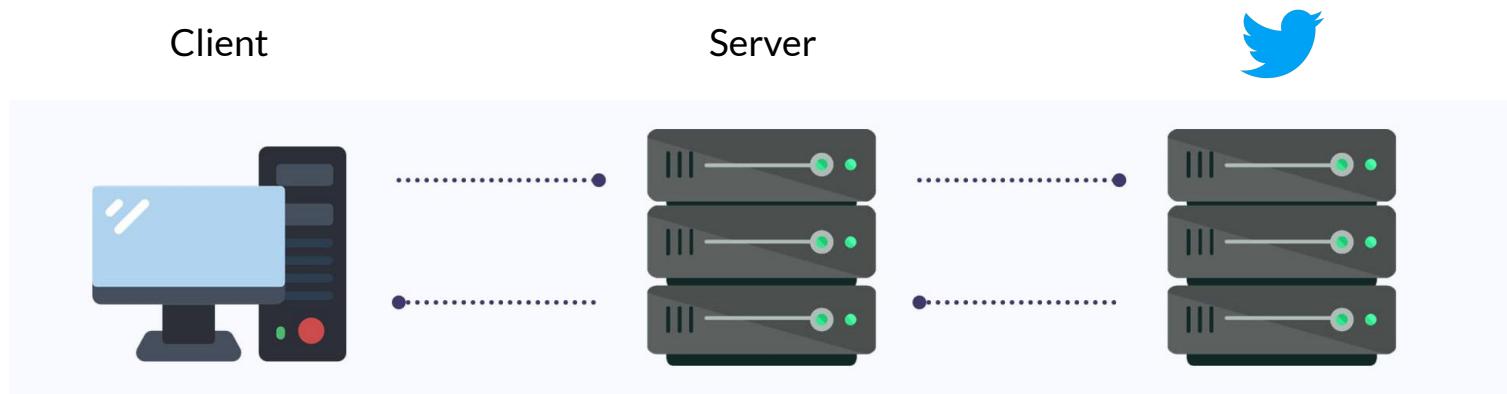


Il Prodotto

A cura di Mattia Guazzaloca



Architettura Generale

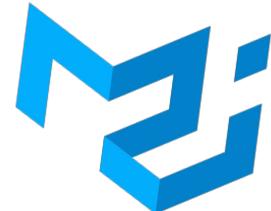
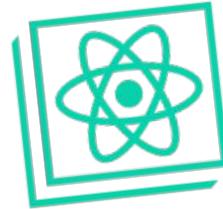
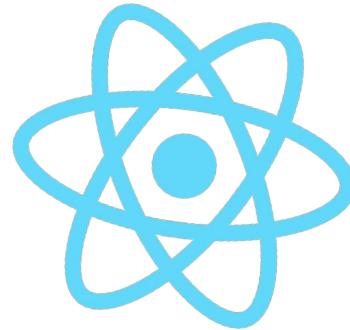


Client

Il front-end dell'applicazione è scritto interamente nel linguaggio Javascript attraverso il framework [React.js](#)

Il setup iniziale dell'applicazione è stato creato tramite il tool [create-react-app](#)

Anche l'interfaccia grafica è stata sviluppata in JavaScript grazie alla libreria [Material-UI](#) che implementa in JavaScript le [Material Design](#) guidelines di Google

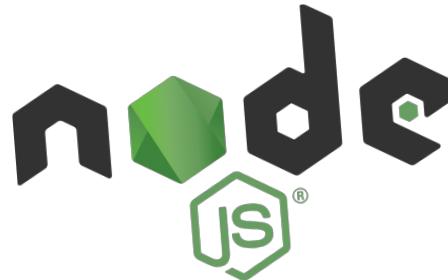


Server

Anche il back-end dell'applicazione è scritto nel linguaggio JavaScript ed implementa un semplice server [NodeJS](#).

Per semplificare la gestione delle richieste dei client e fornire una comoda API REST abbiamo utilizzato il framework [Express.js](#).

Per la parte real-time dell'applicazione si è invece deciso di fare uso dei WebSocket, in modo da inviare i tweet raccolti direttamente al client, grazie alla libreria [Socket.io](#).



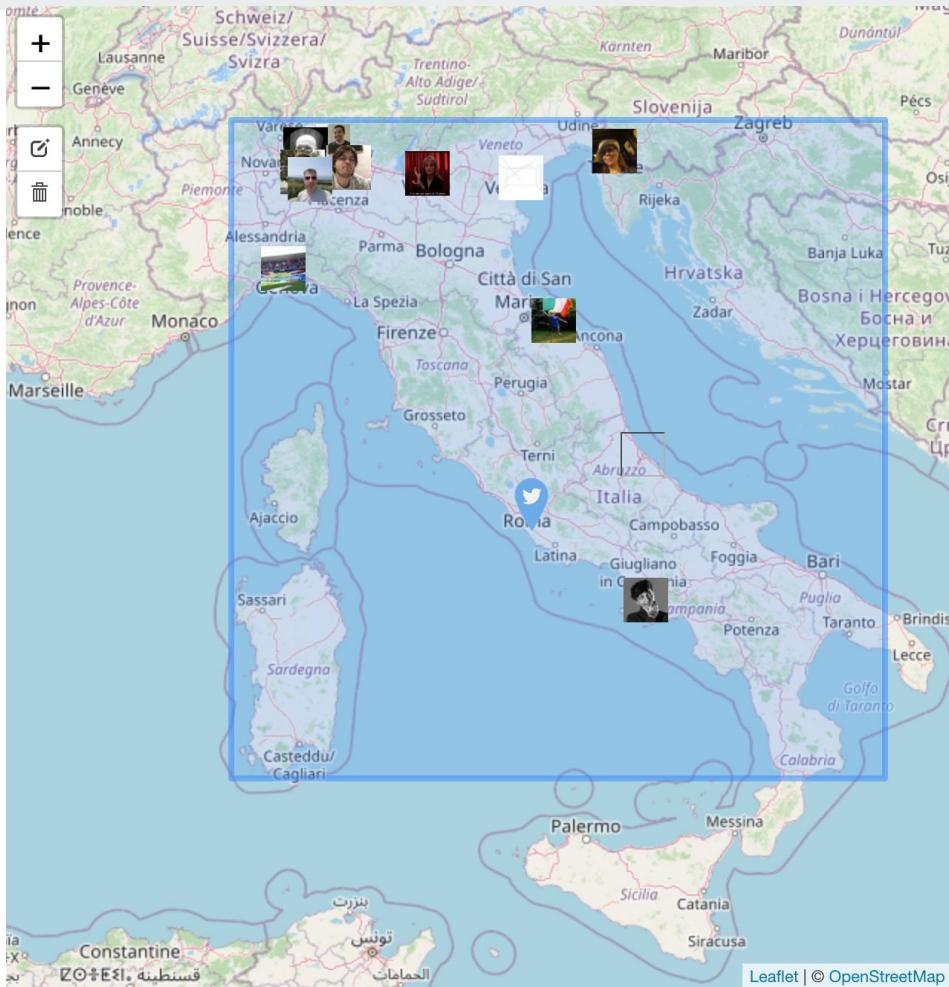
express



OpenStreetMap

La mappa sfrutta le API pubbliche, open source e gratuite di OpenStreetMap grazie alla libreria [Leaflet.js](#) e al corrispondente binding per React, [react-leaflet](#).

La possibilità di disegnare i bounding-box direttamente sulla mappa non è una feature built-in di react-leaflet. Pertanto è stato necessario integrare la libreria con un plugin apposito.



Twitter API

A cura di Lorenzo Liso



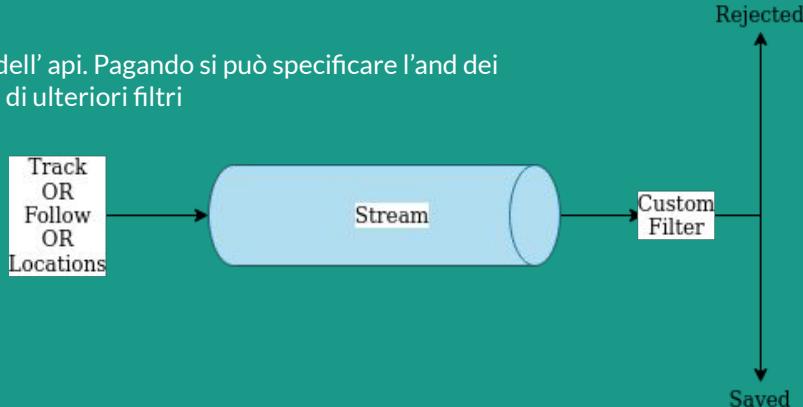
Utilizzo

- **Libreria** : twitter-lite disponibile su npm
- **Versione** : 1.1
- **Autenticazione** :
 - **User** : streams **Endpoints** : statuses/filter
 - **App** : username -> id **Endpoints** : users/lookup
 - **3-Legged-OAuth** : postare un tweet **Endpoints** : media/upload, statuses/update + funzioni di libreria getRequestToken e getAccessToken

Stream

- Stream creato con parametri standard di twitter in OR tra loro(hashtag/parole, utenti, coordinate) *
- Ad ogni tweet ritornato viene applicato un filtro specificato alla creazione. Es : Se si vuole raccogliere i tweet con #apple localizzati a bologna, si passano le coordinate di bologna allo stream e si controlla se i tweet ritornati contengono #apple

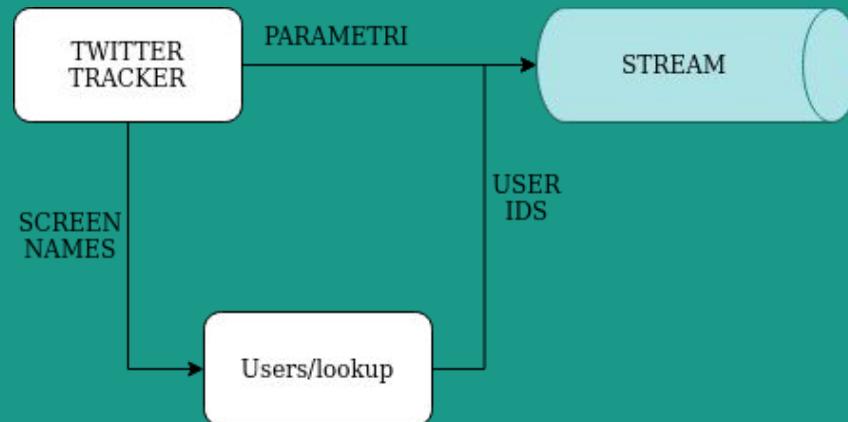
* limite della versione free dell' api. Pagando si può specificare l'and dei valori evitando la necessità di ulteriori filtri



Gestione Username

Gli streams accettano come parametro follows una stringa di ids separati da virgolette.

Bisogna tradurre gli username inseriti in ids, questo viene fatto tramite l'endpoint `users/lookup`.

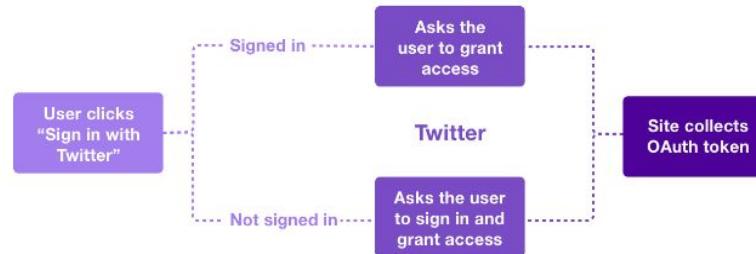


Autenticazione a Tre Vie

Fase 1 : getRequestToken(callbackUrl) registra l'url a cui l'utente verrà re-indirizzato una volta autenticato, la funzione ritorna un token di autenticazione

Fase 2 : reindirizzamento al seguente link dove l'utente inserisce le credenziali di twitter, redirect a callbackurl con parametri delle credenziali temporanee oauthVerifier, oauthToken

Fase 3 : getAccessToken(oauthVerifier, oauthToken) per ottenere credenziali definitive
L'utente ora può eseguire dei tweet usando twitter-tracker



Postare dei Tweet

Un tweet ha del testo e può avere dei media allegati.

Per allegare dei media bisogna caricare prima il contenuto usando l'endpoint *media/upload* che restituisce un id della risorsa

Per postare un tweet si utilizza l' endpoint *statuses/update* indicando il testo (status) e i media_ids dei contenuti da allegare

UI/UX

A cura di Enea Guidi



UI

Sincronizzazione non in corso

TWITTER TRACKER

124 tweets collected

SEARCH

START

WORDCLOUD

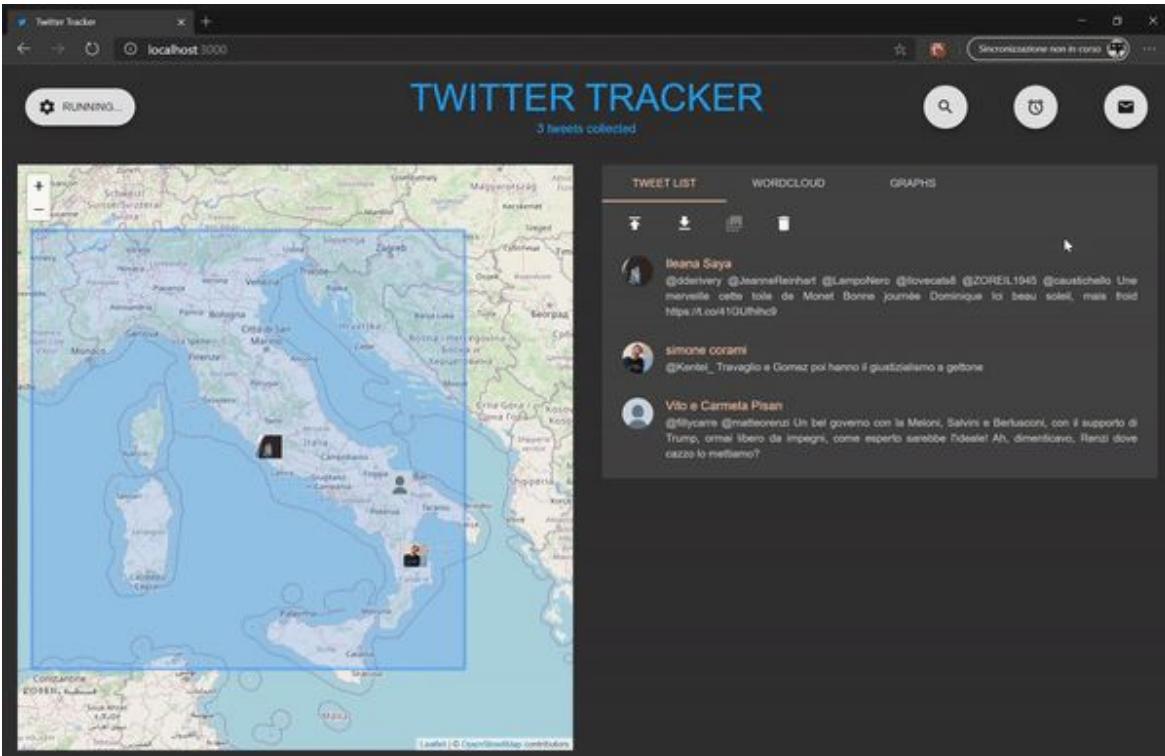
GRAPH

Map showing tweet locations across Europe, with many points clustered in Italy and surrounding areas.

LIST OF TWEETS:

- un'immagine un po' curiosa
@AnnamariaBello Normale
- Ivana Saya
@doluccio16 Lasciate bionerare.
- sarra_befazi
#Lavoro_dici_ogni https://t.co/gPjwka9BC
- Traven Alekza si
@StevenghVince Pe lo ni resultat Sternovit 154 pje 10 je. Tokko dozvati po em drugi testovi, ki kaže stopnjo nadobrošči !
- Graziella Pezzettamus #vaccinaneH24
@comitg @Emmanu26648690 @La_menina_ @madisonrci @italiaViva Ma cara molto diaformata o prevenuta, non so, l'articolo 18 non è MAI stato abolito, questa bufala nemmeno vi sfiorate di verificartela, per il resto fa uno sforzo e scopri cosa ha fatto, mettiti comoda e una lunga lettura.
- Tommaso Fattani
@FBlasin E non hanno mai sentito la canzone del Quartetto Cetra "Didi Vai Pelo"
- Portail Azur Langues • GND Interlangues Ac-Nice
⚠ Ouverture des inscriptions pour les stages de perfectionnement linguistique 🔍 17 janvier.
- Francesca Ragazzino
@andrea70136118 Auguri
- Lorenzo Guerini Rocco
@venere_dilimmi Asilo inaslucca 😊

Email notification



Scheduled tweet

The screenshot shows a web browser window titled "Twitter Tracker" with the URL "its-hmny/TwitterTracker:Awesome". The main interface features a map of Northern Italy with various towns labeled. Overlaid on the map is a white rectangular dialog box containing the following content:

Scheduled tweet

By authenticating here you can set an interval upon which a tweet will be published on your account with the given text and an image of the map and/or the picture of the wordcloud

Interval between tweets: STATUS: stopped

how many hours should we wait?

Tweet Text:
This is an automated tweet sent by my new webapp Twitter Tracker check it out here: <https://github.com/its-hmny/TwitterTracker>

122/140

Map Word cloud CHANGE INTERVAL SEND NOW! CONFIRM

Leaflet | © OpenStreetMap contributors

Diagramma delle classi e diagramma dei casi d'uso

A cura di Matteo Lorenzoni

Diagramma delle classi

Utilità

- Componenti
- Organizzazione e relazioni
- Tracciabilità
- Dimensione applicazione

Difficoltà

- Applicazione su javascript (react)
- Aggiornamento

Link

- [Demo interattiva](#)
- [Diagramma completo HQ](#)
- [4° sprint](#)
- [3° sprint](#)
- [2° sprint](#)

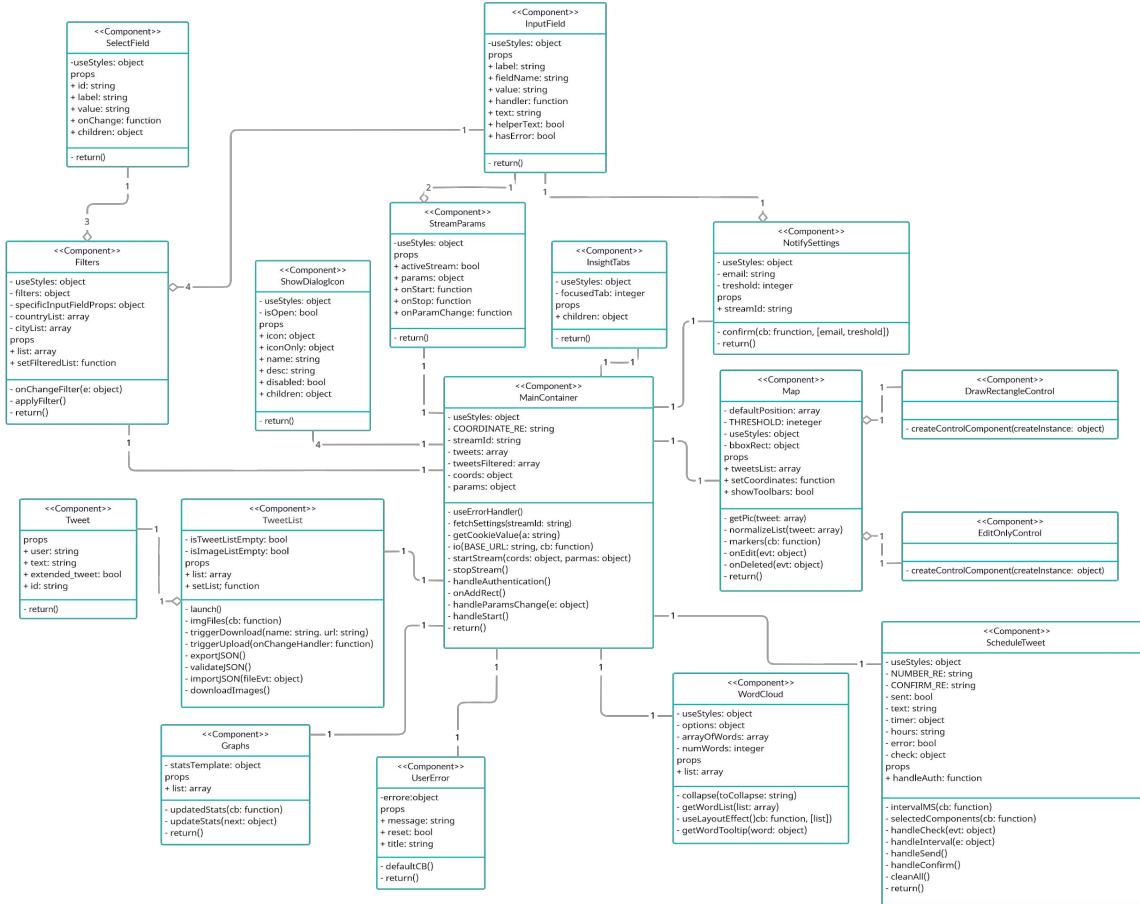
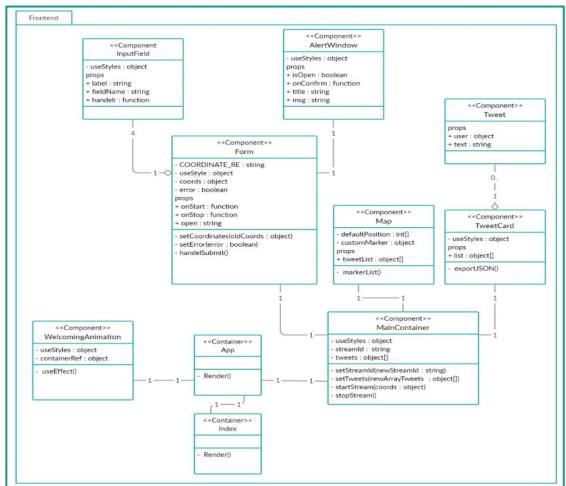


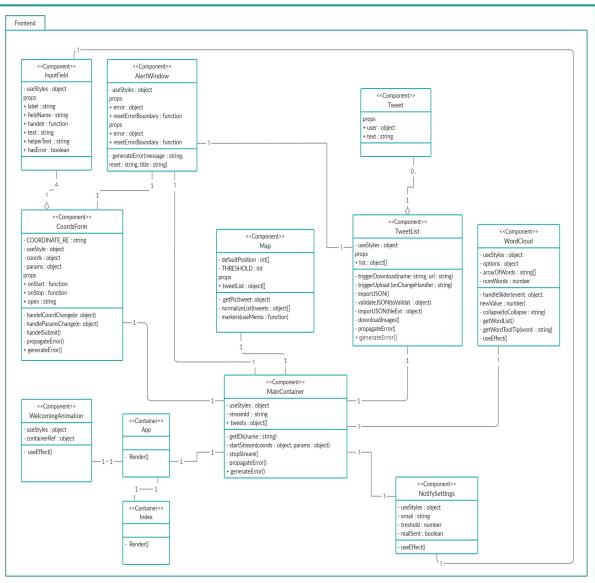
Diagramma (parziale) delle classi finale

Evoluzione del diagramma delle classi

2° SPRINT



3° SPRINT



4° SPRINT

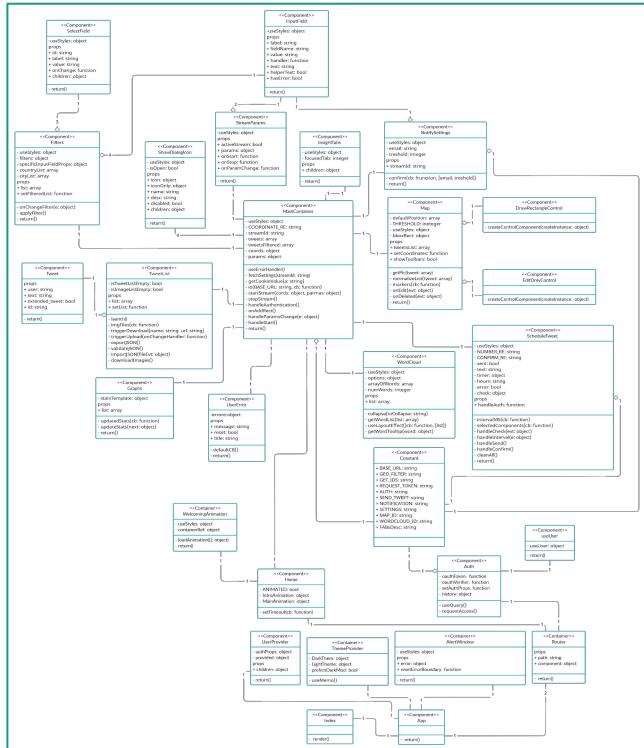


Diagramma dei casi d'uso

Utilità

- Realizzazione completa
- Realizzazione precisa
- Organizzazione e relazioni
- Dimensione

Difficoltà

- Reperimento e trascrizione richieste
- Standard scenario
- Aggiornamento

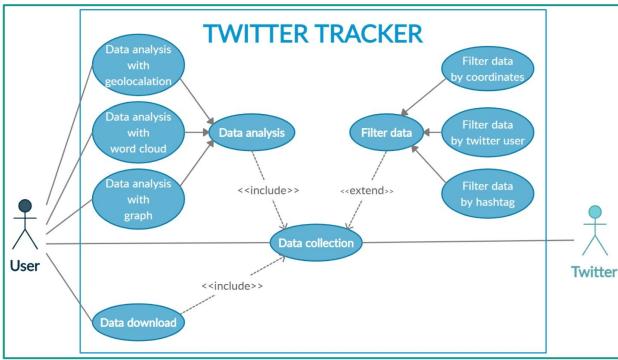
Link

- [4° sprint](#)
- [3° sprint](#)
- [2° sprint](#)

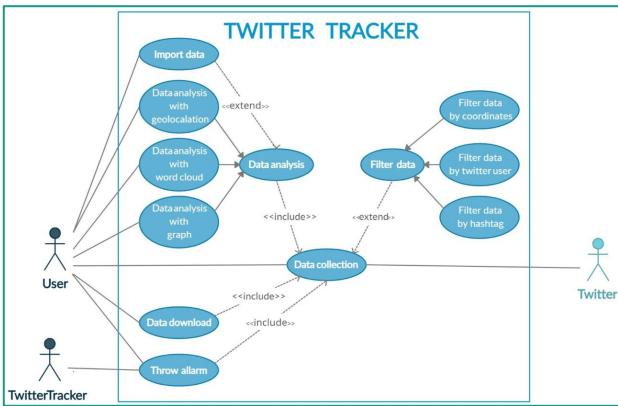
USE CASE: TWTR_TRCKR03	DATA DOWNLOAD	Date: 05/01/2021 Version: 0.01.001
Description:	It allows the user to download the collected data	
Priority:	Medium	
Duration:	Seconds	
Primary actor:	User His interest is to download the data that he had just collected and that he wants to save them, so he can see them again in the future	
Secondary actors:	(Empty) (Empty)	
Precondition:	User has already started and finished the data collection phase and he did not receive an empty collection, so he has some material to work on	
Failure guarantees:	No data is downloaded, the button used for this task cannot be clicked	
Success guarantees	All collected data are downloaded in a json file	
Start:	The data download button is pressed	
Main scenario		
	USER	SYSTEM
1.		It check if tweet list contains at least one tweet, if the verification is confirmed then it makes the relative button available
2.	He presses the download button	
3.		It download the collected data to the user's personal computer
Error scenario		
Tweet list is empty (no tweet available to download)		
	USER	SYSTEM
1.1		It check if tweet list contains at least one tweet, if the verification is not confirmed, the relative button is still not available
2.1	He can't press the download button	
-		END

Scenario del caso d'uso "Data download"

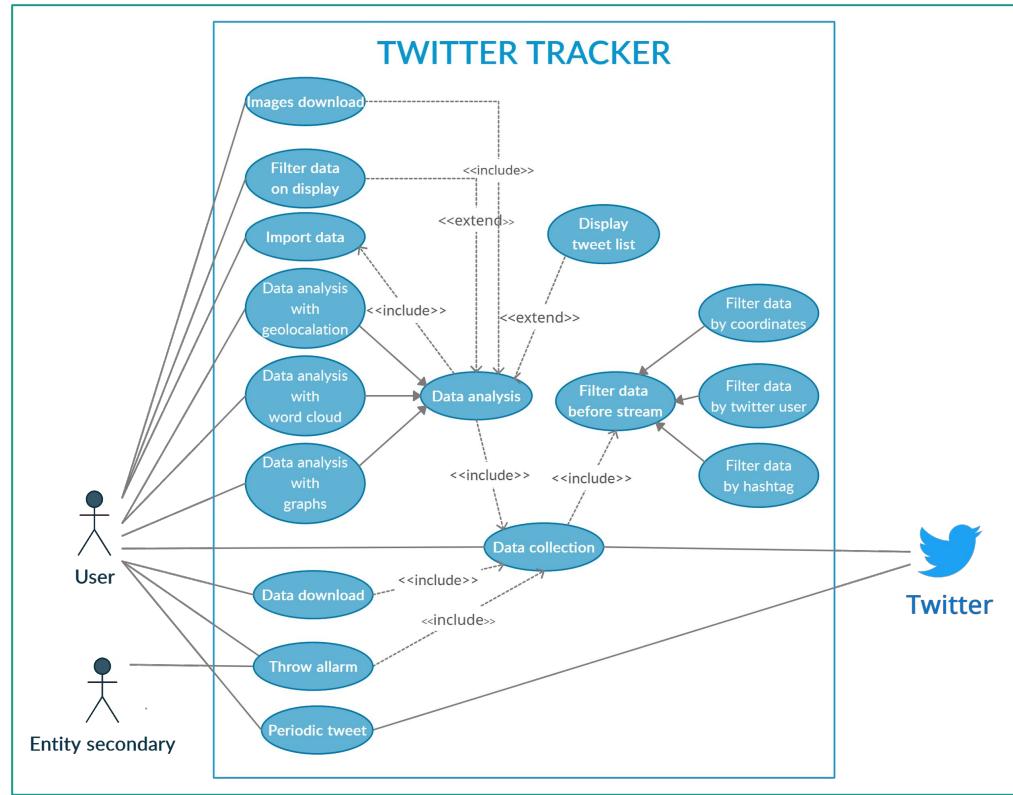
Evoluzione del diagramma delle classi



2° SPRINT



3° SPRINT



4° SPRINT