

Relazione conclusiva sul progetto del Corso di Ingegneria del Software

Angelo Galavotti,
Giacomo Ciccone,
Letizia Gorini,
Marco Amerotti,
Matteo Baldazzi

January 24, 2022

Contents

1	Descrizione del prodotto	3
1.1	Scope	3
1.2	Features	3
1.2.1	Ricerca	3
1.2.2	Filtraggio	3
1.2.3	Statistiche	3
1.3	Backlog di prodotto	4
1.4	Dettagli implementativi	5
1.4.1	Design Pattern	6
2	Team e Scrum	7
2.1	Giochi di team-building	7
2.1.1	Would you rather	7
2.1.2	Due verità, una bugia	7
2.1.3	Partita a Scrumble	7
2.2	Divisione dei ruoli	8
2.3	Sprint effettuati	9
2.4	Sprint 1	9
2.4.1	Sprint Backlog	9
2.4.2	Retrospettiva	9
2.4.3	Taiga	10
2.5	Sprint 2	11
2.5.1	Sprint Backlog	11
2.5.2	Retrospettiva	11
2.5.3	Taiga	12
2.6	Sprint 3	13
2.6.1	Sprint Backlog	13
2.6.2	Retrospettiva	13
2.6.3	Taiga	14
2.6.4	SonarQube e Jenkins	14
2.7	Micro-Sprint 4	16
2.7.1	Micro-Sprint Backlog	16
2.7.2	Retrospettiva	16
2.7.3	Taiga	17
3	Test e Definition of Done	17
4	Diagrammi	18
4.1	Diagramma di sequenza	18
4.2	Casi d'uso	19
4.3	Grafo delle dipendenze	20
4.4	Diagramma di deployment	21
5	Conclusioni	21
5.1	Artefatti prodotti	22

1 Descrizione del prodotto

1.1 Scope

Il prodotto realizzato è un client per Twitter che offre funzionalità di ricerca e filtraggio dei tweet, fornisce statistiche sui risultati della ricerca e permette la realizzazione di contest letterari e trivia. La ricerca può essere effettuata anche tramite la funzionalità di streaming, che fornisce risultati pertinenti alla ricerca in tempo reale. Per motivi dipendenti dalla struttura stessa dell'API fornita da Twitter, i risultati appartengono al più all'arco temporale dell'ultima settimana.

1.2 Features

1.2.1 Ricerca

Il client permette di iniziare una ricerca direttamente dalla home page in base ai seguenti criteri:

- keyword: vengono restituiti i tweet contenenti una data parola chiave;
- username: vengono restituiti i tweet pubblicati da un dato utente;
- hashtag: vengono restituiti i tweet per un dato hashtag;
- contest: vengono restituiti i tweet del contest iniziato dall'utente specificato con un determinato hashtag.

L'utente viene quindi portato alla pagina dei risultati, che permette un ulteriore raffinamento della ricerca, così come la possibilità di cominciarne una del tutto nuova. La pagina dei risultati mostra ogni tweet con il suo autore, data di pubblicazione, posizione (se disponibile), numero di retweet, numero di preferiti, link al tweet originale.

1.2.2 Filtraggio

Oltre alle diverse modalità offerte per iniziare una ricerca, il client permette di filtrare i tweet ottenuti come segue:

- per posizione: la mappa interattiva consente di scegliere una posizione e un raggio di ricerca;
- per data: è possibile scegliere gli estremi dell'arco temporale preferito (all'interno dell'ultima settimana);
- per popolarità: è possibile scegliere se restituire solo tweet popolari;
- per lingua italiana: vengono restituiti solo i tweet in lingua italiana.

È inoltre possibile scegliere se attivare lo streaming dei tweet in tempo reale e scegliere il numero di tweet restituiti (questa opzione disabilita tuttavia le statistiche per questioni di performance).

1.2.3 Statistiche

I tweet restituiti vengono elaborati e le seguenti statistiche sono rese disponibili:

- sentiment analysis: ogni tweet riceve un punteggio da 0 a 10 e il punteggio della ricerca è la media di questi (0: negativo, 10: positivo);
- wordcloud: le parole più ricorrenti all'interno della ricerca;
- numero medio di preferiti;
- lingua più comune;
- numero medio di retweet.

1.3 Backlog di prodotto

Fondamentale per il processo di sviluppo verticale è stata la definizione di appropriate User Stories, di seguito elencate con i rispettivi punteggi.

User story	Descrizione	Punteggio
Tema di navigazione scuro (o chiaro)	Come utente abitudinario vorrei poter scegliere il tema di navigazione scuro (o chiaro), per facilitare la leggibilità del testo negli orari serali (o diurni)	6
Informazioni riguardo le funzionalità dell'applicazione	Come persona non molto pratica con la tecnologia vorrei avere accesso alle informazioni riguardo le funzionalità dell'applicazione in ogni momento in modo da essere guidato nell'uso	5
Rimando al tweet su Twitter	Come utente vorrei poter visualizzare i tweet direttamente su twitter in modo da poter facilmente accedere alle funzionalità di questo	10
Responsiveness del sito	Come utente indaffarato vorrei avere la possibilità di usare il sito in modo efficiente e pratico sia da pc che da smartphone (responsiveness)	10
Ricerca in base a filtro singolo	Come utente vorrei poter fare una ricerca secondo un filtro singolo con la quale trovare in modo più immediato i tweet che mi interessano	20
Tweet in base a hashtag e keyword	Come utente vorrei poter cercare i tweet per keyword o in base ad un determinato hashtag per monitorare un evento da me seguito	15
Tweet in base all'utente	Come fan vorrei poter cercare i tweet della mia star preferita in base allo username	12
Mappa	Come utente vorrei poter vedere direttamente sul sito la posizione precisa di alcuni tweet di mio interesse	8
Mappa che mostra tweet con feedback visivo	Come utente vorrei vedere e leggere i tweet graficamente sulla mappa per comodità	12
Tweet Streaming	Come appassionato di un certo evento (denominato da un hashtag o una keyword) o di un certo account vorrei avere a disposizione il "Tweet Streaming" per rimanere aggiornato	25
Selezione di più filtri	Come utente curioso vorrei aver la possibilità di combinare più filtri per avere ricerche più approfondite e più informative	6
Word Cloud	Come analista vorrei sapere quali sono le parole/hashtag/username collegati alla mia ricerca attraverso una Word Cloud per aver in modo grafico e quindi più immediato le informazioni a me utili	15

User story	Descrizione	Punteggio
Filtrare tweet in base alla data	Come analista vorrei filtrare i tweet per data in modo da raccogliere dei dati sui tweet passati	13
Filtrare tweet in base alla locazione	Come analista vorrei filtrare i tweet per locazione in modo da raccogliere dati sui tweet di una determinata area	20
Sentiment Analysis	Come analista vorrei sapere la sentiment analysis per ogni parola/hashtag/username cercato	5
Dashboard/Grafico con statistiche basate sulla ricerca	Come analista vorrei avere una dashboard che racchiuda tutte le statistiche in modo da disperdermi di meno nel sito e guadagnare tempo	20
Concorso letterario	(User story professore) Come organizzatore, voglio bandire un concorso letterario di racconti brevi della durata di due mesi gestito via twitter	15
Sistema raccolta voti	(User story professore) Come scrittore speranzoso, voglio poter far votare il mio libro	8
Sistema per votare	(User story professore) Come lettore, voglio poter disporre di 10 voti per votare il miei libri preferiti	16
Visualizzare risultati contest	(User story professore) Come organizzatore, voglio poter visualizzare i risultati della votazione in tempo reale	17
Organizzazione trivia e contaggio voti	(User story professore) Come organizzatore, voglio fare una domanda (trivia) e contare le risposte, per vedere quanti giocatori ci sono	18
Rispondere al trivia	(User story professore) Come giocatore, voglio rispondere ad una domanda e ottenere punti	18

User story logistiche	Punteggio
Prima progettazione UI	12
Introduzione al metodo Scrum ed inizializzazione progetto	10
Goker deployment	20
Richiesta deployment	6
Scrittura del report finale	20
Rifinire test	10

La stima agile dei requisiti è stata effettuata tramite il metodo del *planning poker*, permettendo ad ogni membro di esprimere una prima opinione non condizionata da quella del resto del team. Nonostante le iniziali, come inevitabili, differenti prospettive sull'argomento, non è risultato arduo trovare un compromesso e convergere su un'unica soluzione.

1.4 Dettagli implementativi

Per usufruire dell'API di Twitter è stato adoperato un account di gruppo per sviluppatori. Le richieste vengono inoltrate e gestite tramite un server Node apposito.

Per la realizzazione del client abbiamo deciso di adottare una variante dello stack MERN:

- React e Redux per il frontend;
- Express e Node per il backend;

Come framework CSS abbiamo optato per Tailwind con Daisy UI per semplificare la gestione dei temi e dei componenti di base.

Seguono per completezza le principali librerie utilizzate.

1. Grafica

- Tailwind CSS e il plugin daisyUI (un kit che usa tailwind per fornire componenti ready to use) consentendo la creazione di un design semplice, pulito ed elegante;

- react-reveal, per effetti grafici interessanti;
- leaflet, react-leaflet, per visualizzare ed usare la mappa.

2. Analisi dei Tweet

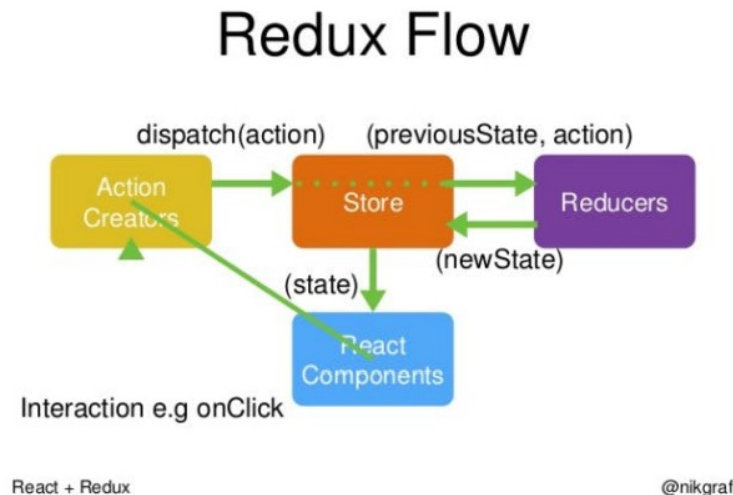
- chart.js, per creare grafici;
- multilang-sentiment, per la sentiment-analysis multilingua;
- stopwords, che permette di eliminare tutte le stopwords in modo da avere token utilizzabili per la wordcloud;
- react-wordcloud, per usare la wordcloud di react che sfrutta d3 cloud e che la rende responsive.

3. Application logic

- react-redux, per la gestione dello stato condiviso;
- Twitter NPM per gestire le chiamate all'api in maniera semplificata per nodejs;
- socket.io-client, per creare un canale socket diretto tra client e server per lo streaming in tempo reale;
- axios, un client HTTP basato sulle promesse per Node.js e per il browser;
- react-hook-form, per gestire e validare i form in react;
- redux-persist, per il salvataggio delle preferenze dell'utente nel browser.

1.4.1 Design Pattern

Il design pattern utilizzato è stato l'*Observer Pattern*, in quanto naturalmente compatibile con il framework React e la libreria Redux. I componenti dell'app fanno riferimento ad uno stato globale che possono modificare tramite determinate *azioni*. Il cambio di stato implica un aggiornamento automatico dei componenti: essi sono infatti gli *observer* e lo stato è il *subject*.



Data la poca compatibilità dei componenti con il paradigma ereditario, non è stato possibile produrre un diagramma delle classi; risulta invece più informativo un grafo delle dipendenze (Fig. 4.3).

2 Team e Scrum

La comprensione e la conoscenza del framework agile Scrum sono state essenziali per lo sviluppo dell'app, portando grandi benefici all'attività di sviluppo e migliorando l'organizzazione del team. L'utilità di Scrum è stata particolarmente evidente per:

- Diventare un team: le attività di team building hanno permesso ai membri di conoscersi meglio sia da un punto di vista personale, sia da una prospettiva lavorativa e hanno contribuito alla divisione dei ruoli e alla comprensione del significato di questi, amalgamando il gruppo e creando un ambiente disteso e privo di conflittualità;
- Sviluppare in modo agile: ad esempio, tramite il pair programming, rivelatosi molto utile;
- Essere trasparenti: durante ogni sprint ogni componente del gruppo era a conoscenza di cosa si stessero occupando gli altri, favorendo l'interazione continua all'interno del team;
- Essere onesti: la valutazione degli sprint effettuati è stata condotta senza nascondere eventuali perplessità, difficoltà e dubbi, concentrandosi anzi su come risolvere tali problematiche in vista degli sprint successivi.

2.1 Giochi di team-building

Come incentivato durante le lezioni, non tutti i membri del gruppo si conoscevano personalmente; per familiarizzare, sono stati condotti dei giochi di team-building:

2.1.1 Would you rather

"*Would you rather*", o "*Preferiresti*": un semplice gioco in cui ogni partecipante chiede agli altri di scegliere tra due ipotetici scenari, di solito ugualmente vantaggiosi o svantaggiosi. Il gioco è servito per rompere il ghiaccio e socializzare nel team.

2.1.2 Due verità, una bugia

Questo gioco è un'alternativa al più comune "*obbligo o verità*", che la modalità da remoto utilizzata per le riunioni non avrebbe permesso di condurre. Ogni membro riporta due fatti sulla sua persona, uno dei quali non vero, e chiede agli altri quale sia l'opzione corretta; è stato utile per conoscere le abitudini e gli impegni degli altri e per capire come organizzare i meeting e il lavoro futuro.

2.1.3 Partita a Scrumble

In una delle prime riunioni del gruppo è stata svolta una partita al gioco di team-building *Scrumble*: per prepararsi è stato necessario capire il funzionamento di un processo Scrum e come utilizzare le carte Essence.

Si è discusso lungamente per scambiare idee sullo svolgimento del gioco, così da fare luce su eventuali dubbi e da poter procedere senza intoppi.

Sono stati quindi decisi i ruoli da assumere durante la partita, quali:

- PO: Angelo Galavotti;
- SM: Matteo Baldazzi;
- Developers: Giacomo Ciccone, Marco Amerotti, Letizia Gorini.

Preparato il necessario per il gioco, si è proceduto a stimare la priorità e la difficoltà di ogni user story attraverso il metodo del *Planning Poker*. Ogni discussione sulla stima di un'User Story ha portato ad una soluzione quasi unanimemente accettata.

Per tutta la partita il debito tecnico è stato molto basso (*No debt*), raramente sfiorando nella zona di *Low debt* e mai oltre. Limitando l'incremento del debito (non sono mai state utilizzate le carte *daily*, che lo aumentavano) si è riusciti a completare quasi tutte le user stories del primo sprint.

Dopo svariate ore dedicate al gioco, era ormai abbastanza chiaro come funzionasse un processo Scrum; la partita è stata gestita in modo soddisfacente e ha permesso successivamente di mettere in pratica quanto imparato.

Conclusioni: le attività svolte sono state utili sia per amalgamare il team, sia per comprendere le diverse fasi del metodo Scrum (*Sprint planning, Daily Scrum, Sprint Review e Sprint Retrospective*), in modo da prendere confidenza con questa metodologia. Di seguito è riportato il questionario completato a fine partita.

GOAL	QUESTION	Angelo PO	Matteo SM	Giacomo DEV	Marco DEV	Letizia DEV
Learn	Do team members understand the Scrum roles?	3	4	5	4	4
	Do team members feel they learned the process?	4	5	5	5	5
	Does everyone keep up with the other players?	4	4	5	5	5
Practice	Are the game mechanics linear and repeatable?	5	5	5	5	4
	Do team success in completing the game?	2	2	2	2	2
	Do team members efficiently estimate during sprint planning?	x	x	5	5	4
Cooperation	Do team members know each other better?	5	4	5	4	4
	Does the game let all players cooperate?	4	5	4	5	5
	Do team member consult each other about a topic?	4	4	5	4	4
Motivation	Do team members encourage colleagues in need?	5	5	5	5	5
	Does PO help the team?	4	x	x	x	x
	Does the team come up with good ideas?	3	5	5	5	4
Problem Solving	Do team members behave well when facing a problem?	4	4	4	4	5
	Does team organize their tasks properly?	x	x	5	5	5
	Does PO plan efficiently the Sprint Backlog?	5	x	x	x	x

Questionario di fine partita.

2.2 Divisione dei ruoli

La divisione dei ruoli è derivata in parte dal gioco di team-building Scrumble, dove è stato necessario assegnare dei ruoli provvisori per iniziare a comprendere cosa fosse un processo Scrum. La generale soddisfazione alla fine dell'attività ha contribuito a consolidare la scelta dei ruoli anche per l'effettivo sviluppo del progetto, constatando l'adeguatezza degli stessi per i vari componenti del gruppo e le esigenze di ognuno.

2.3 Sprint effettuati

Il processo di sviluppo si è articolato in tre sprint principali, integrati da un minore sprint aggiuntivo: il tempo a disposizione è stato, nel complesso, adeguato e sufficiente al completamento dell'app. Le User Stories del secondo sprint hanno necessitato più impegno del previsto e non è stato possibile completarle tutte in tempo; la situazione è però rientrata con lo sprint successivo e il precedente ha prevenuto il ripresentarsi della questione.

Alla fine di ogni sprint abbiamo valutato il lavoro svolto con una retrospettiva, guidati dalle carte Essence. Questo comporta l'individuazione di tre categorie, Glad, Sad, Mad, nelle quali inserire rispettivamente cosa ci ha lasciato soddisfatti, cosa ha suscitato perplessità e cosa ci è risultato ostico. Questa metodologia è stata di grande aiuto per ripensare il lavoro svolto e dipanare le problematiche sorte.

2.4 Sprint 1

Il primo sprint ha consentito di entrare nell'ottica del progetto, di comprendere cosa implicasse sviluppare agilmente secondo i principi Scrum e di familiarizzare con i membri del team. Non avendo compreso qualora fosse necessario cominciare l'implementazione, abbiamo preferito dedicarci allo studio della consegna del progetto, ipotizzando possibili soluzioni e prendendo confidenza con i tools forniti dall'università, quali GitLab, Taiga, Mattermost, CAS, SonarQube e Jenkins. Buona parte dello sprint è stata impegnata dallo studio dell'API di Twitter da alcuni primi esperimenti con essa. Abbiamo successivamente appreso come fosse richiesta una demo delle features complete fino a quel momento: avendo precedentemente discusso ampiamente dei dettagli implementativi, non è stato difficile realizzarle e presentare una prima versione base del client.

2.4.1 Sprint Backlog

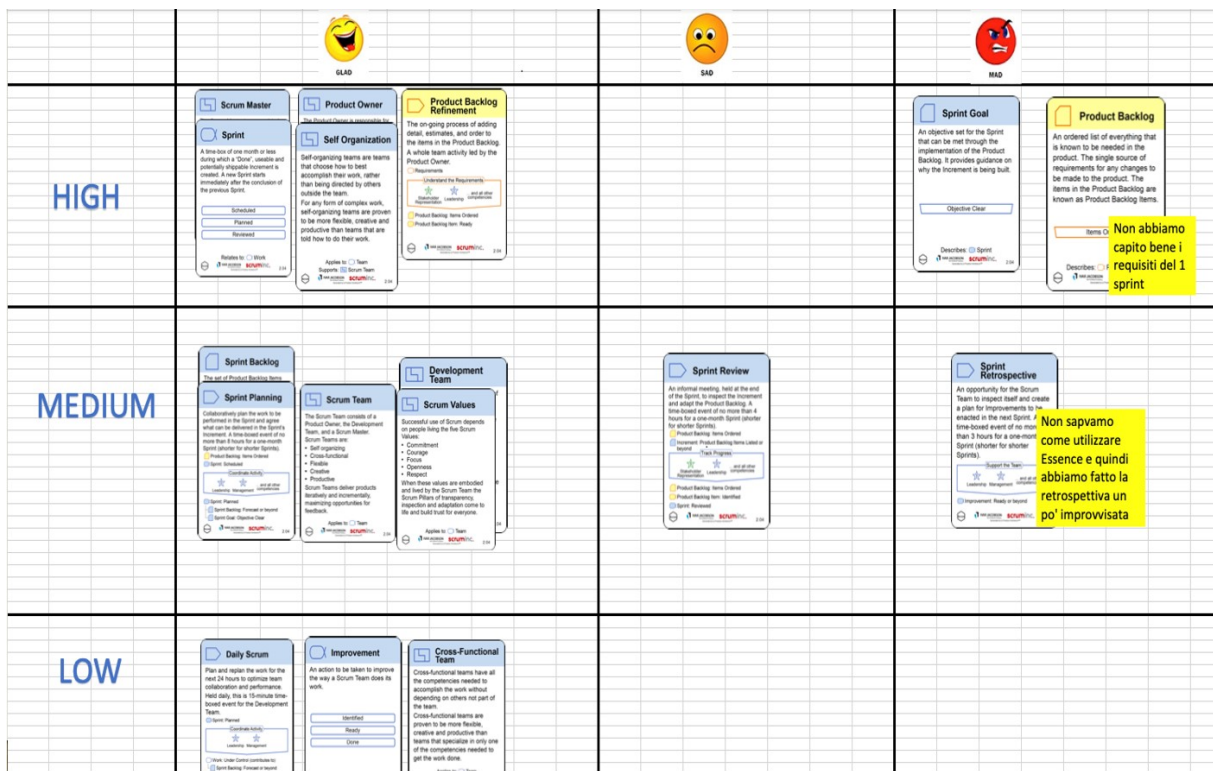
Questo sprint comprendeva le seguenti User Stories:

User Story	Punteggio
Introduzione al metodo Scrum ed inizializzazione progetto	10
Tema di navigazione scuro (o chiaro)	6
Prima progettazione UI	12

2.4.2 Retrospettiva

Essendo il primo sprint, vi erano ancora dubbi sulle modalità di lavoro e sui requisiti di questa prima fase; tali circostanze hanno influito sulla performance del gruppo e sul prodotto. Siamo stati contenti dell'organizzazione trovata, nonostante le difficoltà; è stato invece meno incoraggiante il giudizio finale dello sprint complessivo da parte del cliente, contrariamente alle nostre speranze.

- GLAD: l'organizzazione dello sprint, del backlog, e del team in generale;
- SAD: il feedback del cliente (*scilicet* il professore) è stato inaspettatamente non positivo;
- MAD: la mancata comprensione dei requisiti e dell'obiettivo dello sprint. Non era inoltre inizialmente chiaro come usare adeguatamente le carte Essence.



Retrospectiva dello Sprint 1 con carte Essence.

2.4.3 Taiga

Avendo finito tutte le user stories nel burndown chart non vi sono punti residui.
Totale punti del primo sprint: 28.
Punti rimanenti: 0.



Burndown chart dello Sprint 1.

2.5 Sprint 2

Chiariti i dubbi del primo sprint, il secondo si è focalizzato sul rendere il client effettivamente fruibile, implementando i primi filtri e la ricerca, insieme alla mappa interattiva. Particolare attenzione è stata posta alla realizzazione di un prodotto migliore dello sprint precedente, data la valutazione negativa di quest'ultimo da parte del cliente; ciò ha tuttavia involontariamente contribuito alla sovrastima delle User Stories implementabili, non permettendo di completarle per tempo. L'esperienza è stata comunque utile per comprendere meglio il carico di lavoro per User Story e valutare meglio gli sprint successivi, oltre che per familiarizzare con lo sviluppo agile e di gruppo. Concluso questo sprint, avevamo a disposizione un *minimum viable product*.

2.5.1 Sprint Backlog

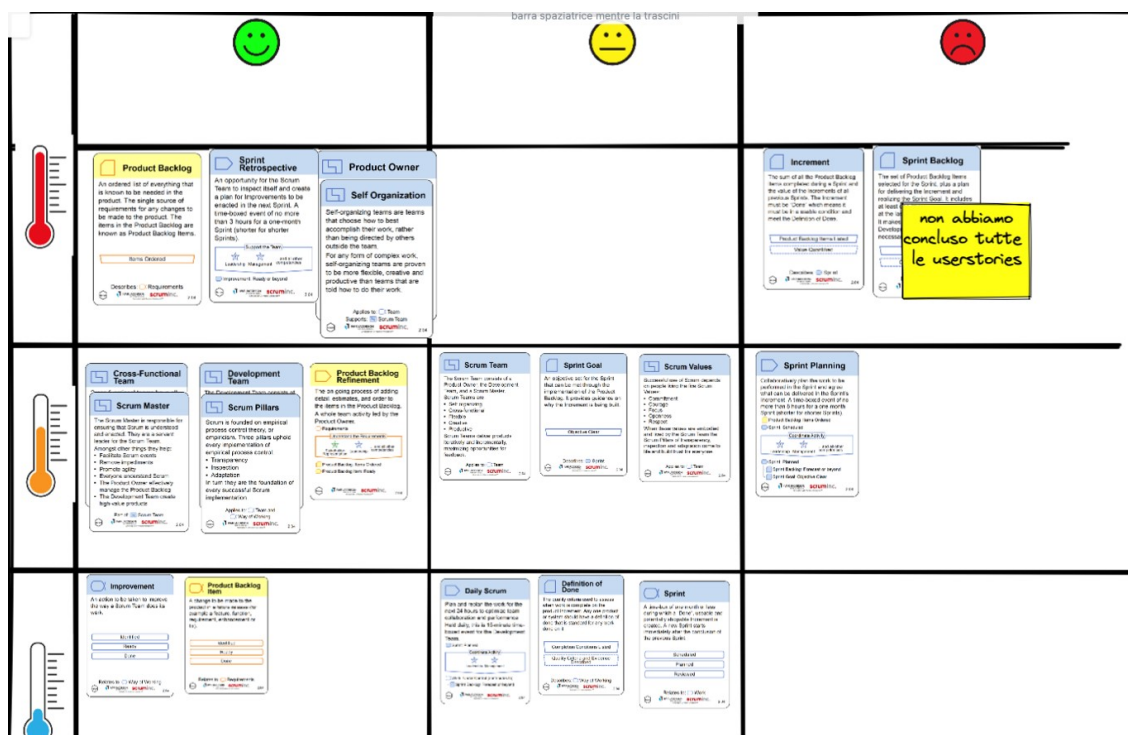
Questo sprint comprendeva le seguenti User Stories:

User Story	Punteggio
Barra di ricerca	15
Rimando al tweet su Twitter	10
Responsiveness del sito	10
Ricerca in base a filtro singolo	20
Mappa	8

2.5.2 Retrospettiva

Il secondo sprint ha mostrato un miglioramento dal punto di vista della comprensione dei requisiti e dei principi dello sviluppo agile, cosa che si è poi ripercossa positivamente sull'organizzazione e la produttività del team. Più difficile è stato mettere in atto tali principi a causa del minore tempo che è stato possibile dedicare allo sprint da parte dei membri del gruppo. Nell'ottica di recuperare il lavoro negativamente valutato del primo sprint, il carico delle User Stories è stato sottostimato e ciò non ne ha consentito il completamento.

- GLAD: organizzazione personale e del team, comprensione di Scrum;
- SAD: messa in atto dei principi di Scrum e tempo dedicato allo sprint;
- MAD: sprint planning e mancato completamento delle User Stories.



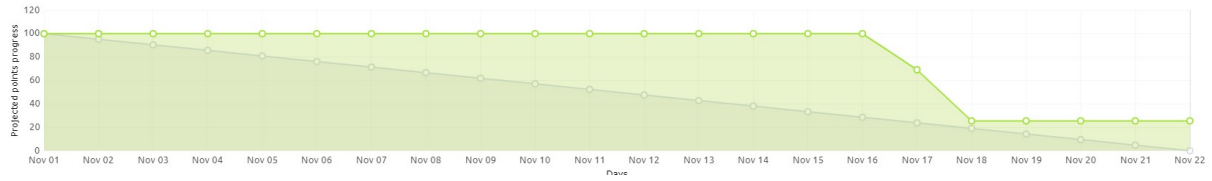
Retrospettiva dello Sprint 2 con carte Essence.

2.5.3 Taiga

Non avendo finito tutte le user stories, rimangono nel burndown chart punti residui, che saranno poi assegnati nel terzo sprint.

Totale punti del primo sprint: 100.

Punti rimanenti: 25.5.



Burndown chart dello Sprint 2.

2.6 Sprint 3

Il terzo sprint è quello che più ha contribuito alla forma attuale del prodotto: sono state implementate funzioni di ricerca avanzate, così come di analisi dei risultati, quali Sentiment Analysis e Word Cloud. La mappa è stata resa completamente interattiva e sincronizzata con i risultati. Nonostante la quantità di User Stories scelte, l'andamento di questo sprint è stato in generale migliore dei precedenti: sono state completate le User Stories mancanti dello sprint precedente e buona parte dello sviluppo è avvenuta in sessioni di pair programming. Questo miglioramento è da attribuirsi all'ormai consolidata comprensione sia dei principi di Scrum, sia dei requisiti del progetto, dalla quale la produttività generale ha evidentemente tratto beneficio. Sono state aggiunte alcune user stories, data la conclusione preventiva di quelle programmate: non tutte sono state ultimate, ma si è così anticipata parte del lavoro del quarto sprint.

2.6.1 Sprint Backlog

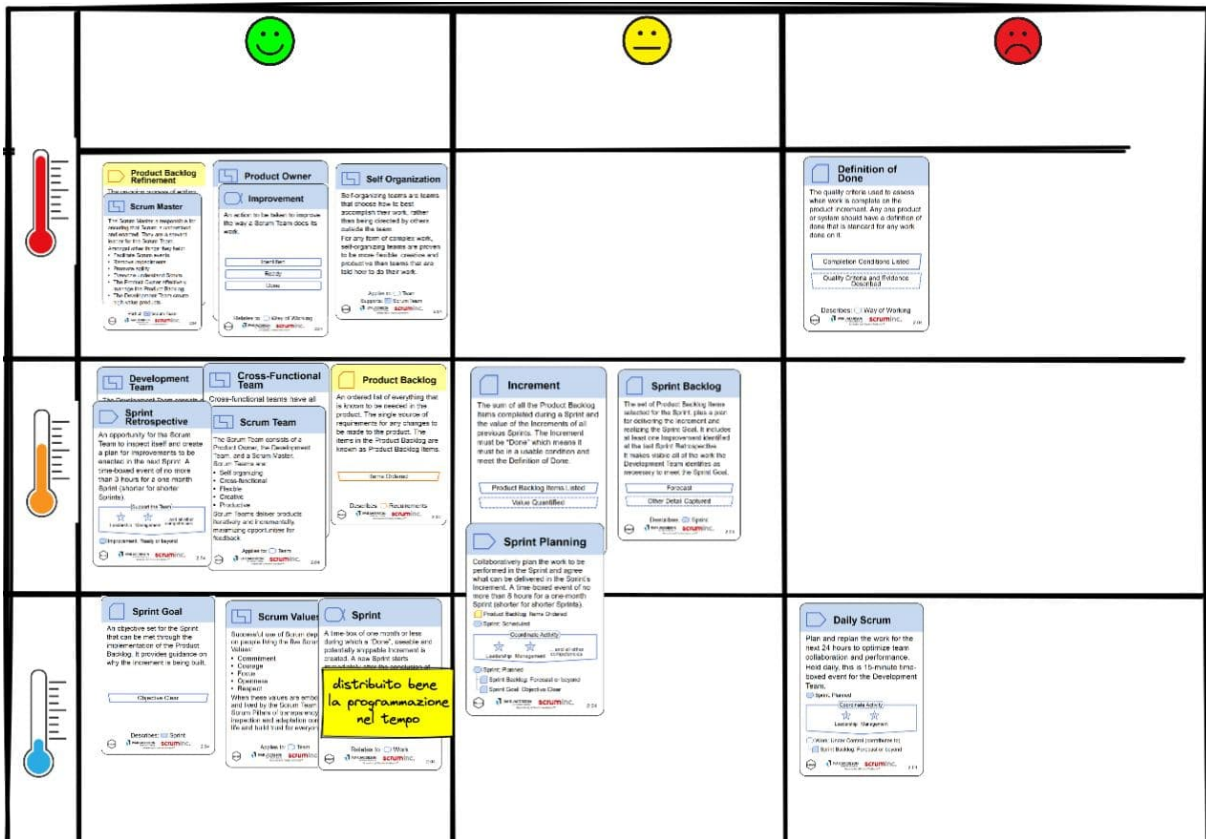
Questo sprint prevedeva le seguenti User Stories:

User Story	Punteggio
Tweet in base all'utente	12
Selezione di più filtri	6
Filtrare tweet in base alla locazione	20
Mappa che mostra tweet con feedback visivo	12
Dashboard/Grafico con statistiche basate sulla ricerca	20
Sentiment Analysis	5
Word Cloud	15
Filtrare tweet in base alla data	13
Tweet Streaming	25
Tweet in base a hashtag e keyword	15
Informazioni riguardo le funzionalità dell'applicazione	5
Concorso letterario	15
Visualizzare risultati contest	17
Rispondere al trivia	18

2.6.2 Retrospettiva

In questo sprint è emersa la migliorata comprensione di Scrum e delle pratiche di sviluppo agili, che sono state utilizzate maggiormente. Con il senno di poi, si sarebbero potute aggiungere user stories a questo sprint e il lavoro potrebbe essere stato programmato diversamente, evitando di aggiungere user stories verso la fine dello sprint. Ancora carenti il Daily Scrum e la completezza e affidabilità del prodotto in questo momento.

- GLAD: organizzazione, comprensione e messa in atto di Scrum, progresso generale del progetto;
- SAD: Sprint planning e Sprint backlog
- MAD: Definition of Done e Daily Scrum



Retrospettiva dello Sprint 3 con carte Essence.

2.6.3 Taiga

Vi sono alcuni punti non assegnati: questi sono quelli delle user stories successivamente aggiunte a quelle inizialmente programmate.

Totale punti del primo sprint: 165.

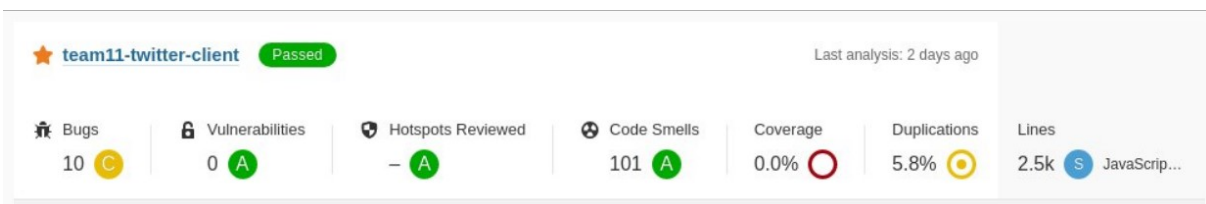
Punti rimanenti: 75.



Burndown chart dello Sprint 3.

2.6.4 SonarQube e Jenkins

Durante lo sprint 3 la situazione su SonarQube era la seguente:



Stato di SonarQube allo Sprint 3.

Inoltre abbiamo qui una delle primissime build fatte con Jenkins, ancora sulla versione su server pubblico:

 **Build #23 (Dec 3, 2021 2:09:51 PM)**

 No changes.

 Started by user [Angelo Galavotti](#)

 **Revision:** d1cc771a4e955605ac1327938cd859bb21dc81b3
Repository: <https://aminsep.disi.unibo.it/gitlab/uniboswe-team-11/twitter-client.git>

- refs/remotes/origin/master

Esempio build fatta con Jenkins.

2.7 Micro-Sprint 4

Avendo quasi del tutto ultimato le user stories nel terzo sprint, questo è durato meno dei precedenti. Il focus è stato sul perfezionamento del client e su altre task principalmente logistiche per consegnare il progetto.

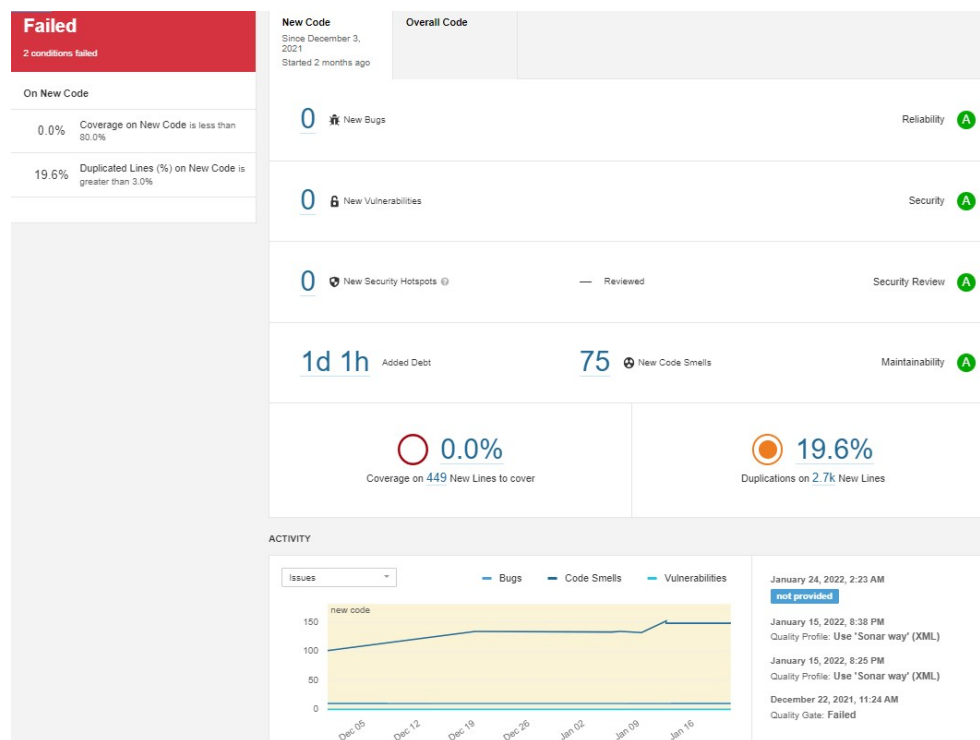
2.7.1 Micro-Sprint Backlog

User Story	Punteggio
Sistema raccolta voti	8
Organizzazione triviale e conteggio voti	18
Sistema per votare	16
Goker deployment	20
Richiesta deployment	6
Scrittura del report finale	20
Rifinire test	10

2.7.2 Retrospettiva

Non essendoci molte user stories da implementare, avendo quindi una limitata quantità di codice da scrivere, non è sembrato necessario fare una retrospettiva. Evidenziamo tuttavia le task prettamente logistiche portate avanti:

- Jenkins: in aggiunta al server Jenkins fornito dall'Università, si è adoperato un server privato ([server Jenkins](#)) per questioni di sicurezza, preferendo invece utilizzare il primo come backup. Entrambi i server sono operativi e garantiscono l'integrazione continua;
- nonostante numerosi tentativi e il ricorso sia alla documentazione, sia alla [community](#), non vi è stato modo di aggiornare correttamente la code coverage, che dunque rimane a zero.



finale SonarQube.

Infine, riportiamo quanto mostrato dal Git Inspector:



Git inspector finale.

2.7.3 Taiga

Avendo completato tutte le User Stories, non vi sono punti residui nello sprint backlog.
Totale punti del primo sprint: 98.
Punti rimanenti: 0.



Burndown chart dello Sprint 4.

3 Test e Definition of Done

Durante lo sviluppo del prodotto, ciascuna user story è stata definita come "done" solo dopo il superamento di determinati test manuali *ad hoc*.

Verso la fine del secondo sprint, sono stati progettati i primi unit test per alcuni componenti, utilizzando la libreria **JEST**; la UI è stata invece testata manualmente.

Alla fine dello sviluppo, sono stati progettati 26 unit test in totale.

4 Diagrammi

Di seguito riportiamo alcuni diagrammi che aiutano a comprendere la struttura ed il funzionamento del client.

4.1 Diagramma di sequenza

Il client opera su due server distinti, rispettivamente frontend e backend. L'utente imposta e raffina la ricerca tramite il primo; la query ottenuta viene dunque gestita dal server backend, che la trasforma in una richiesta valida per l'API di Twitter; quindi, una volta ricevuti i risultati, il server backend genera le statistiche e seleziona le informazioni necessarie, che vengono mandate al frontend per essere fruibili dall'utente.

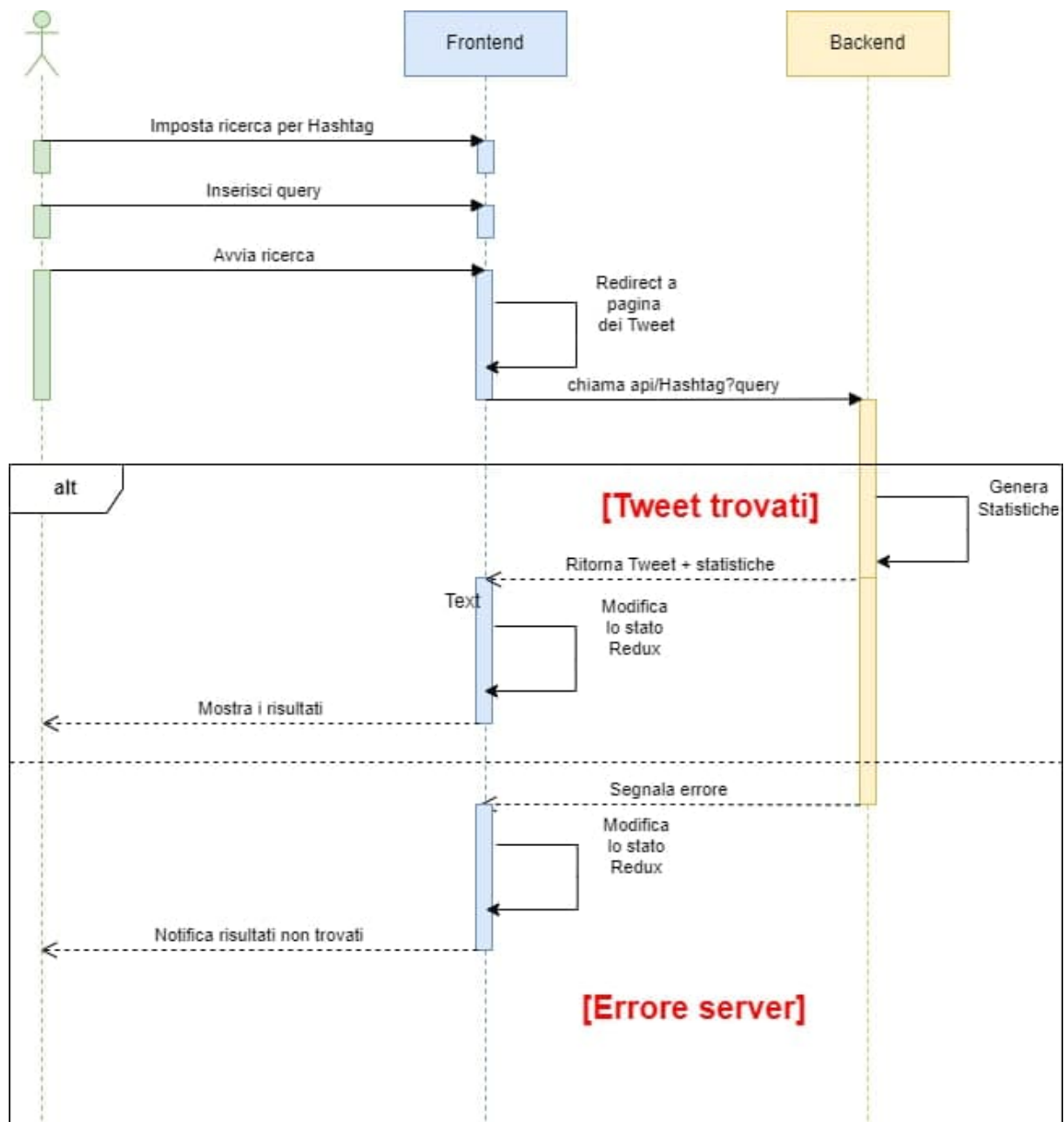


Diagramma di sequenza.

4.2 Casi d'uso

Sono stati considerati diversi possibili utenti, *in primis* l'utente medio e l'utente curioso. L'utente medio svolge operazioni base di ricerca dei tweet secondo i semplici criteri di ricerca per username, hashtag e keyword. L'utente curioso aggiunge a ciò il filtraggio dei tweet per popolarità, lingua, data e posizione e l'analisi dei tweet tramite statistiche. Successivamente è stato considerato il caso dell'utente che volesse creare contest e trivie tramite Twitter e le funzionalità di condivisione e gradimento dei Tweet. Questo utente può creare o cercare un contest, votare, visualizzarne i voti e i vincitori.

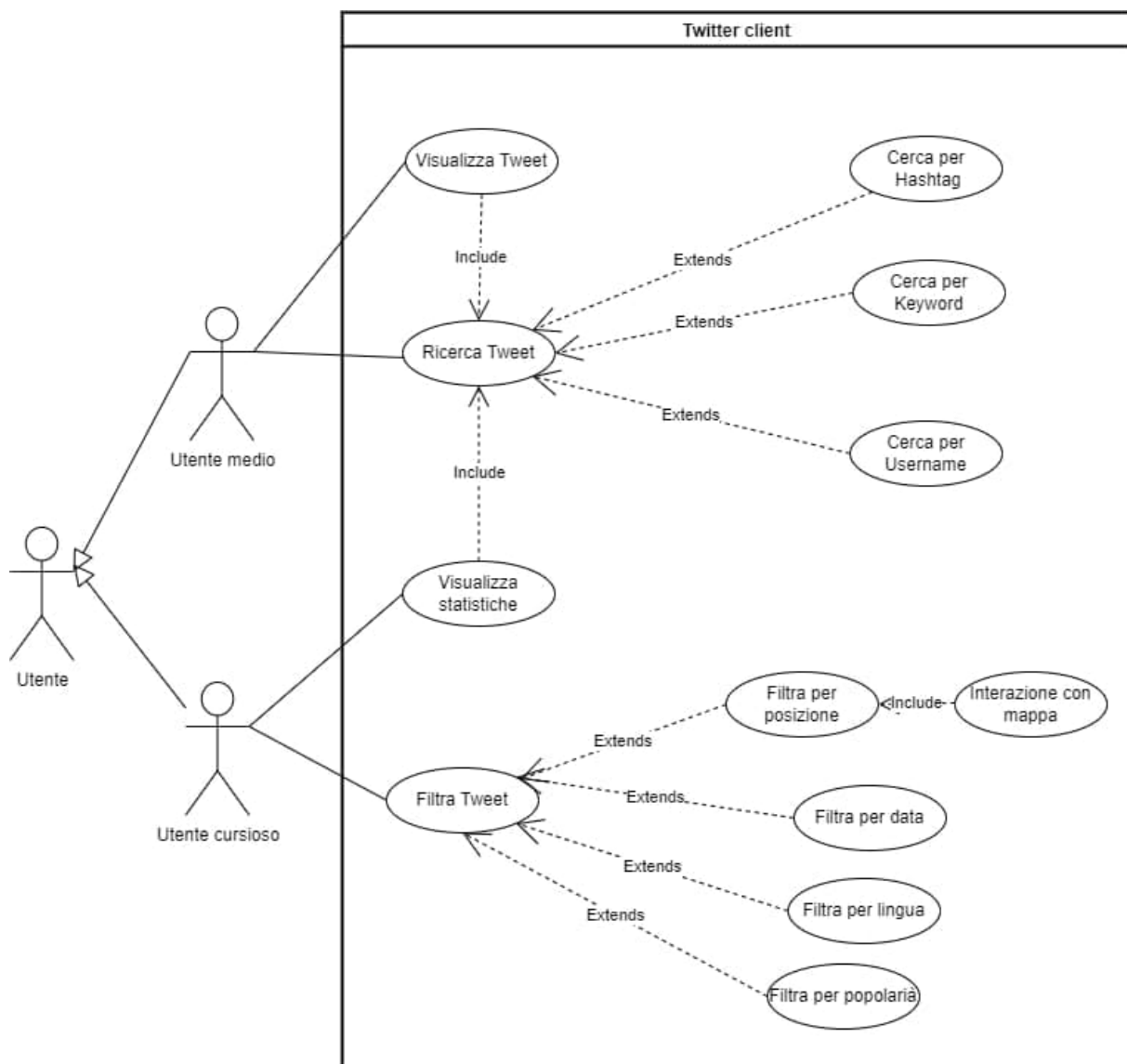
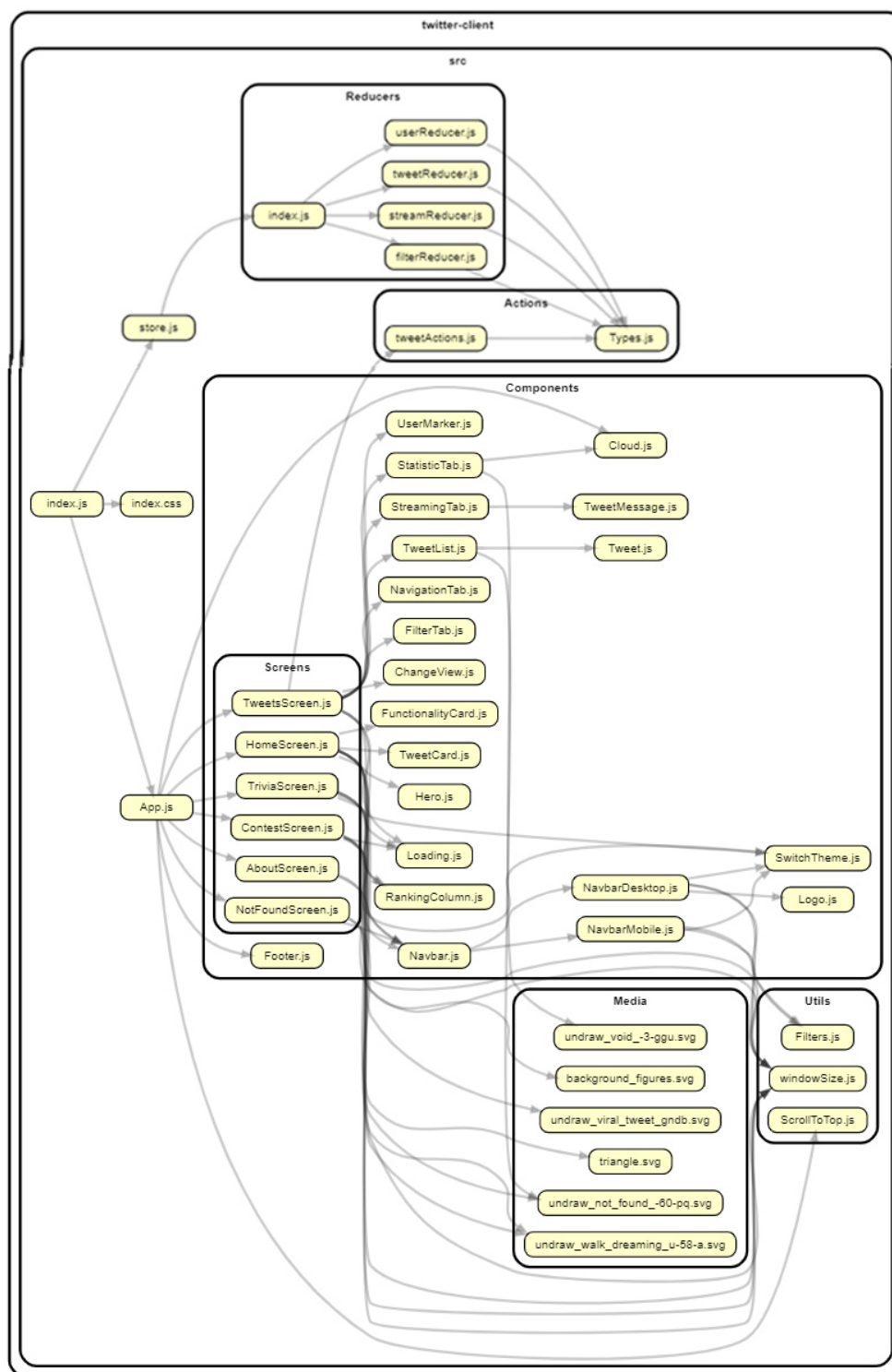


Diagramma dei casi d'uso.

Come accennato in precedenza, il sistema basato su componenti del framework React non si presta bene all'ereditarietà. Risulta invece più informativo il grafo delle dipendenze di seguito riportato.



4.4 Diagramma di deployment

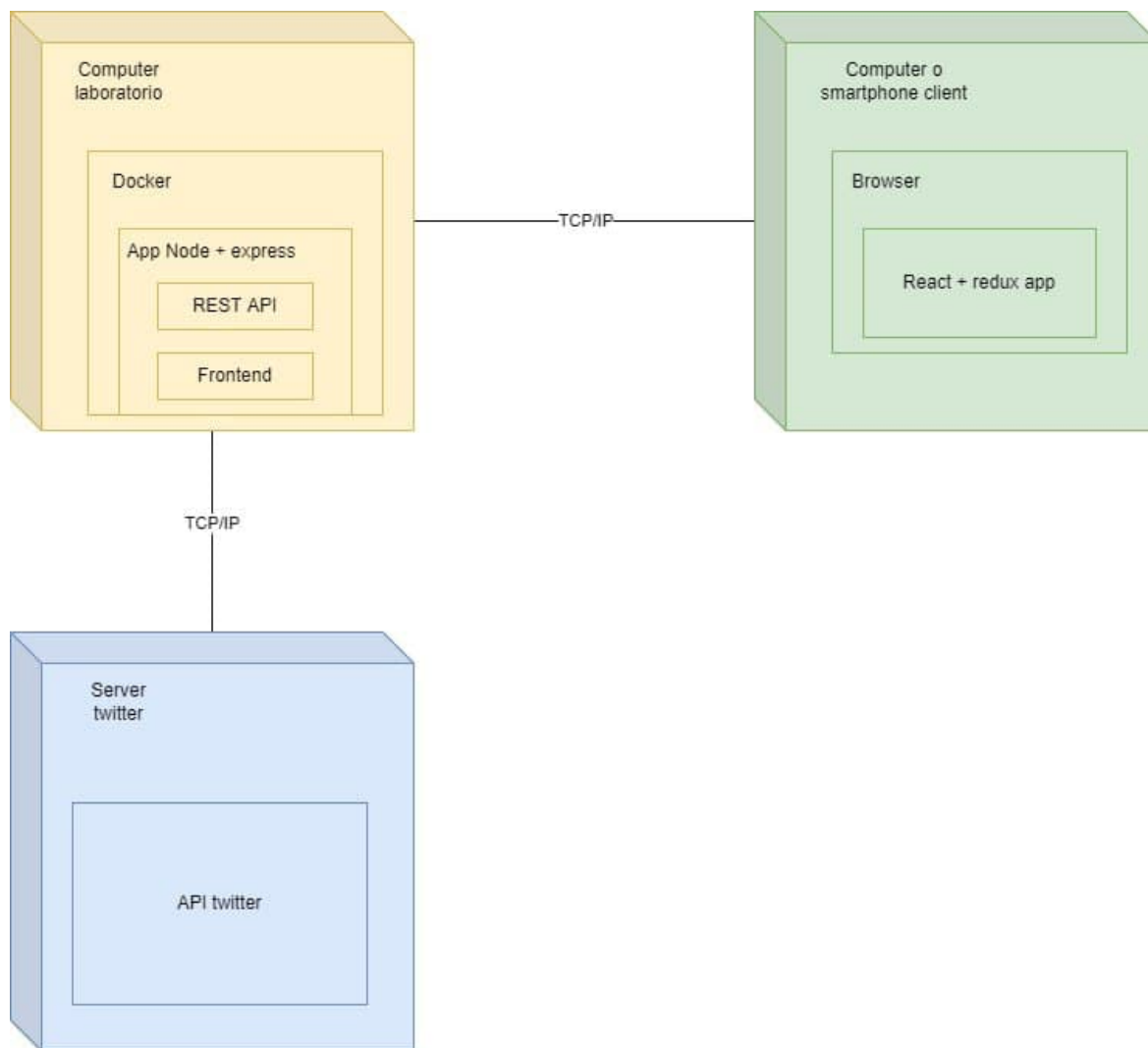


Diagramma di deployment.

5 Conclusioni

Il lavoro svolto si è rivelato profondamente utile su molteplici fronti.

Le attività di gruppo sono state essenziali per la coesione e stima reciproca all'interno del team, che ha permesso a questo progetto di andare avanti pacificamente e in un clima sereno e produttivo.

Tale coesione ha favorito l'adozione del metodo di sviluppo agile, garantendo l'interazione costante tra i membri del team e incentivando la partecipazione di ognuno. Scrum si è mostrato essere un paradigma eccellente per lavorare in un team e avere un'organizzazione precisa e ordinata delle tasks e dello stato del software; essendo entrati in contatto per la prima volta con questa metodologia, non possiamo fare a meno di segnalare come sarebbe stato certamente utile apprendere all'inizio dei nostri studi per affrontare le consegne passate.

Infine l'introduzione ai tool forniti dall'Università, per quanto non del tutto inizialmente lineare, ha aiutato considerevolmente lo sviluppo e ha fornito un'idea di quello che vuol dire sviluppare software, in contrapposizione con la precedente e naturale predisposizione dello studente alla semplice programmazione.

Grazie a questo lavoro risulta ora evidente come "fare" del software coinvolga numerose attività e competenze che trascendono la semplice implementazione, ma che anzi spaziano dalla progettazione al mantenimento, al lavoro di squadra, alla pianificazione, al testing e molto altro.

5.1 Artefatti prodotti

- questo documento;
- Prodotto finale: <https://site202148.tw.cs.unibo.it/>;
- Video demo: <https://www.youtube.com/watch?v=v2LiRsR4ejs>;
- Sorgente: <https://aminsep.disi.unibo.it/gitlab/uniboswe-team-11/twitter-client.git>;
- Sprint: <https://aminsep.disi.unibo.it/project/admin-progetto-2021-team-11/timeline>;
- Testing: <https://aminsep.disi.unibo.it/sonarqube/dashboard?id=team11-twitter-client>;
- Continuous integration: <http://angelinojenkins.ddns.net:3401>.

Sprint Taiga:

- Sprint 1: <https://aminsep.disi.unibo.it/project/admin-progetto-2021-team-11/taskboard/sprint-0-2>;
- Sprint 2: <https://aminsep.disi.unibo.it/project/admin-progetto-2021-team-11/taskboard/sprint-2>;
- Sprint 3: <https://aminsep.disi.unibo.it/project/admin-progetto-2021-team-11/taskboard/sprint-3-3>;
- Sprint 4: <https://aminsep.disi.unibo.it/project/admin-progetto-2021-team-11/taskboard/microsprint-4>.