

Rapporto finale

Progetto di Ingegneria del Software
Team 10
A.A. 2022/2023

1 Fase preliminare – Sprint 0

1.1 Creazione team

Tramite Trello abbiamo formato il nostro team: Alice Altena, Alice Benatti, Gabriele Buttinoni, Emanuele Campanelli, Denis Gjonaj. Consecutivamente ci siamo organizzati per giocare la partita a Scrumble.

1.2 Partita a Scrumble

Abbiamo svolto la partita a Scrumble il giorno 13 ott. 22. Per prima cosa abbiamo impostato il gioco scegliendo i ruoli: il developer team è formato da Alice Benatti, Gabriele Buttinoni e Denis Gjonaj, lo Scrum Master è Emanuele Campanelli e il Product Owner è Alice Altena. A questo punto lo Scrum Master ha esposto le regole del gioco e il Product Owner ha presentato il prodotto, abbiamo utilizzato le User Stories riguardanti un'applicazione per dietologi e clienti.

Dopodiché, in primo luogo, è stata assegnata la complessità relativa ad ogni singola User Story e in secondo luogo, tenendo conto di quest'ultimo valore, le abbiamo organizzate nella seguente tabella in base alla priorità.

Complexity Priority	1 – XS	2 – S	3 – M	4 – L	5 – XL
High		1 15	2 3		
Avarage		8	6 7	14	13
Low	5 10 11	4 9 12			

Abbiamo dunque iniziato a giocare, la partita si è conclusa in 5 sprint da 9 giorni. Per iniziare il **primo sprint** abbiamo scelto le User Stories #1, #2 e #3, considerando fin da subito quelle con priorità maggiore; infatti, è necessario implementare la #1 e la #2 per poter procedere nello sviluppo di un totale di altri 7 casi d'uso (nello specifico: User Story #1 → #2, #4, #8, #9, #13, #15; User Story #2 → #3, #7). Purtroppo, abbiamo concluso lo sprint riuscendo a completare solamente le prime due User Story, ottenendo così un punteggio di Value uguale a 5. Avremmo terminato con un debito tecnico di 18, dopo esser riusciti a contenerlo nonostante le carte Problem, ma avendo avanzato 7 tasks abbiamo ottenuto un malus che ci ha fatto aumentare di 1 il debito tecnico, portandoci così a 19.

Nel **secondo sprint** abbiamo aggiunto alla User Story già presente (#3), la #15 e #8. Avevamo da completare le tasks relative alla #3 (ovvero 8), e abbiamo poi calcolato le tasks della #15 e la #8, rispettivamente 24 e 24. Durante questo sprint il developer che si occupava della #3 ha terminato le tasks rimanenti e poi si è concentrato sulla riduzione del debito tecnico, chi si è occupato della #15 ha portato a termine il suo obiettivo mentre per la #8 sono avanzate solamente 3 tasks e così ce la siamo ritrovata nel terzo sprint. Grazie a chi ha ridotto il debito tecnico siamo arrivati a 0, non abbiamo ottenuto il malus delle tasks avanzate (essendo meno di 5), ma per via delle carte Daily siamo avanzati fino a 4 (il fattore di debito è rimasto invariato).

Al **terzo sprint** abbiamo pensato di velocizzare il tutto, avendoci preso la mano, e abbiamo scelto di implementare un totale di 4 User Stories. In aggiunta alla #8, abbiamo optato per la #5, #7 e #13. Chi si occupava della #8 ha concluso le tasks rimanenti e poi ha iniziato la #5 che richiedeva solamente un totale di 12 tasks. Mentre la #7 = 36 e la #13 = 60 tasks. Purtroppo, anche questa volta non siamo riusciti a portare a termine tutte le User Stories che ci eravamo prefissati per via della #13. Siamo però riusciti a terminare con un debito tecnico pari a 3 (tenendo già conto del malus (+2 di debito tecnico) per le tasks rimaste (14) e delle carte Daily).

Il **quarto sprint** si è svolto similmente al terzo. Abbiamo terminato la US restante e abbiamo cercato di implementarne altre tre: #6 (36 tasks), #14 (48 tasks) e #10 (12 tasks). Tuttavia, chi si occupava della #10, doveva prima terminare la #13, e non è così riuscito a completare tutte le tasks relative. Il debito tecnico finale è risultato pari a 4.

Finalmente al **quinto** ed ultimo **sprint**, abbiamo rischiato tutto e puntato ad implementare tutte le User Stories che ci avanzavano; riuscendo così a terminare il gioco entro il numero di sprint che ci eravamo prefissati e pure con il Value a 45 (ovvero al massimo). Le User Stories specifiche di questo sprint sono: #10 (3 tasks restanti); #4 (24 tasks), #9 (24 tasks), #11 (12 tasks) e #12 (24 tasks). Il debito tecnico finale è stato 0.

1.3 AUTOVALUTAZIONE:

GOAL	QUESTIONS	EVALUATION	Alice Altena	Alice Benatti	Denis Gjonaj	Gabriele Buttinoni	Emanuele Campanelli
Learn	Q1	1 = no idea of the Scrum roles 5 = perfect knowledge of the roles and their jobs	5	5	4	4	4
	Q2	1 = couldn't repeat the game 5 = could play the game as a Scrum Master by himself	5	5	5	5	5
	Q3	1 = totally lost 5 = leads the game driving the other players	5	5	5	5	5
Practice	Q4	1 = feels the game is unrepeatable 5 = feels the game could be played in any situation	4	4	3	5	4
	Q5	1 = 0 to 3 stories 2 = 4 to 6 3 = 7 to 9 4 = 10 to 12 5 = 13 to 15	PO	5	5	5	SM
	Q6 <u>ONLY DEV TEAM</u>	1 = abnormal difference from the other players 5 = coherent and uniform with the group most of the time	PO	5	5	5	SM
Cooperation	Q7	1 = never speaks with the other players 5 = talks friendly to anyone in every situation	5	5	5	5	5
	Q8	1 = never puts effort in doing something 5 = every time is willing to understand what is going on	5	5	5	5	5
	Q9	1 = never asks for an opinion 5 = wants to discuss about every topic	5	5	5	5	5
Motivation	Q10	1 = not involved by the game 5 = always makes sure everyone is on point	5	5	5	5	5
	Q11 <u>ONLY FOR PO</u>	1 = poor/absent advices 5 = wise and helpful suggestions when is required	5				
	Q12	1 = doesn't express opinions during retrospective 5 = feels the retrospective fundamental to express opinions	5	5	5	5	5
Problem Solving	Q13	On the game board, if the debt pawn is on the lowest stage, the evaluation is 5, for every higher stage it decreases by 1	5	5	5	5	5
	Q14 <u>ONLY DEV TEAM</u>	Calculate the average of tasks left for each sprint: 1 = 21+ 2 = 16-20 3 = 11-15 4 = 6-10 5 = 0-5	PO	4	4	4	SM
	Q15 <u>ONLY FOR PO</u>	Same evaluation as Q14 for the PO	4				

2 Fase di esecuzione – Sprint 1 (19/10 – 02/11)

2.1 Sprint goal

L'obiettivo di questo primo sprint è di impostare il più possibile di tutto ciò che ci servirà per fare il progetto: gli strumenti CAS, l'account da Twitter Developer, il product backlog (che in seguito

subirà modifiche, aggiunte e rimozioni) con le epiche e le User Stories principali; e concludere il tutto con la retrospettiva dello sprint.

2.2 Decisione dei ruoli

Dopo aver giocato a Scrumble ci siamo confrontati e abbiamo deciso di comune accordo di rinominare lo Scrum Master; ruolo per cui si è proposto Gabriele Buttinoni. È così iniziato lo sprint 1, dove abbiamo anche giocato ad Escape The BOOM per vedere se questa nuova organizzazione potesse essere quella definitiva. I ruoli finali dunque sono:

- Alice Altena come Product Owner operativo
- Gabriele Buttinoni come Scrum Master
- Team degli sviluppatori:
 - Alice Benatti
 - Denis Gjonaj
 - Emanuele Campanelli

2.3 Attivazione strumenti CAS

Prima di tutto, abbiamo attivato i nostri account sui i servizi CAS del dipartimento; creando eventuali gruppi e/o progetti per far sì che ognuno potesse lavorare in maniera attiva dal singolarmente. Innanzitutto, abbiamo creato il progetto su Gitlab e poi su Taiga; dove abbiamo creato il nostro Product Backlog, programmato i singoli sprint e sfruttato la wiki per condividere informazioni utili sulle Twitter API. Per lavorare tutti insieme, evitando di doverci vedere di persona, abbiamo sfruttato diverse volte Mattermost.

2.4 Account Twitter Developer

In secondo luogo, abbiamo attivato i nostri account Twitter Developer, per poter avere accesso alle API di Twitter. Una volta scritta una parte di codice per testare il funzionamento delle API, ci siamo resi conto che l'Essential Access che viene attivato inizialmente non ci permetteva di avere accesso a tutte le informazioni di cui avevamo bisogno: abbiamo richiesto quindi l'Elevated Access. Per concludere abbiamo creato i nostri file contenenti le Key ed i Token forniti da Twitter per le API.

2.5 Product Backlog

Per iniziare lo sprint abbiamo creato il nostro Product Backlog su Taiga, iniziando dalle 4 epiche principali, e sviluppando poi le User Stories utili ad esse; oltre alle User Stories riguardanti il team.

EPICHE:

- #1 Raccolta tweet de #leredita per ghigliottina.
Come spettatore de #leredita (RAI 1) **voglio** raccogliere i tweet di chi prova ad indovinare la ghigliottina, **per** visualizzare (in ordine temporale, o su una mappa) tutti coloro che indovinano.
- #2 Raccolta tweet di #reazioneacatena per l'ultima parola.
Come spettatore di #reazioneacatena (RAI 1) **voglio** raccogliere i tweet di chi prova ad indovinare l'ultima parola, **per** visualizzare (in ordine temporale, o su una mappa) tutti coloro che indovinano.
- #3 Sfida a scacchi con mosse scelte a maggioranza.
Come giocatore di scacchi **voglio** sfidare gruppi di persone in rete, **per** giocare le partite le cui mosse verranno scelte a maggioranza.
- #4 Raccolta risposte a mossa scacchi scelta dalla maggioranza.
Come giocatore di scacchi **voglio** raccogliere i tweet che rispondono alla mia mossa, **per** scegliere e visualizzare la mossa scelta dalla maggioranza.

AGGIUNTA IN DATA 01 DIC '22:

- #71 Raccolta tweet riguardanti #FantaCitorio:
Come spettatore di #FantaCitorio (Propaganda Live su La7) **voglio** poter accedere ai dati riguardanti il gioco **per** visualizzare i punteggi e le squadre partecipanti.

USER STORIES:

- #5 Raccolgere i tweet.
Come spettatore di un gioco televisivo **voglio** avere una raccolta dei tweet del passato, **per** analisi storiche temporali.
'Done': Ottenere un file contenente i tweet (ad esempio .db, .html, ecc.)
- #7 Raccolgere i tweet di #scacchi in risposta alla mia mossa.
Come giocatore di scacchi **voglio** avere una raccolta di tweet di risposta alla mia mossa **per** visualizzarli in modo temporale.
- #9 Sentiment analysis dei tweet.
Come spettatore di un gioco televisivo **voglio** avere un grafico rappresentante le reazioni del pubblico riguardo al gioco **per** analizzare il sentimento generale dei tweet.
'Done': Deve essere mostrata nella GUI qualche forma di rappresentazione dei dati della sentiment analysis fatta sui tweet cercati (grafici, statistiche, ecc.).
Test: In qualsiasi pagina risultante da una ricerca possiamo trovare la sentiment analysis.
- #11 Raccolta tweet per geolocalizzazione. (+ pt. 2 → #83)
Come spettatore di un gioco televisivo **voglio** avere una raccolta di tweet de #leredita e/o #reazioneacatena **per** visualizzarli su una mappa.
CA: L'utente che twitta con #leredita e/o #reazioneacatena deve aver attivato la funzione di condivisione della posizione.
'Done': Tramite l'inserimento di una zona geografica (ad esempio, una città) ritorna i tweet da tale località; mappa in cui visualizzare i mark dei luoghi taggati nei tweet ritornati dalla ricerca; i mark devono contenere: username, luogo taggato e testo del tweet di riferimento.
Test: I tweet sulla mappa sono solamente quelli geolocalizzati.
- #16 Raccolta tweet di #scacchi per sfidare in gruppo.
Come giocatore di scacchi **voglio** poter avere una raccolta di tweet relativi alla mia partita in una tabella/elenco **per** visualizzare le mosse della maggioranza in risposta al tweet.
- #17 Tweet #leredita indovinati
Come spettatore di #leredita **voglio** avere i dati aggiornati all'ultima puntata **per** poter vedere la parola da indovinare e quante persone hanno indovinato.
'Done': Possibilità di vedere le parole del gioco dell'ultima puntata, con la parola da indovinare coperta in modo che si possa scegliere di vederla o meno. Grafico che mostri quante persone hanno indovinato sul totale che ha ci provato.
Test: Nella pagina dedicata posso ottenere tutte le informazioni di cui ho bisogno.
- #20 Tag cloud dei tweet.
Come spettatore di un gioco televisivo **voglio** visualizzare una world cloud delle principali parole **per** vedere quelle più utilizzate nei tweet (riguardanti il mio interesse).
'Done': Visualizzare nel GUI una tag cloud contenente delle parole nei testi dei tweet ricercati.
Test: In qualsiasi pagina risultante da una ricerca possiamo trovare una tag cloud.
- #22 Partita a Scrumble.
Come team **voglio** giocare e completare una partita a Scrumble **per** comprendere e, di conseguenza, assegnare i ruoli.
- #23 Documentazione API Twitter.
Come team **voglio** studiare le API di Twitter **per** poterle utilizzare negli sprint futuri.

- #30 Partita ad Escape The BOOM.
Come team **voglio** giocare e completare una partita a Escape the BOOM **per** mettere alla prova i ruoli e fare ulteriore team building.
- #39 Creare un'interfaccia web minimale.
Come utente **voglio** utilizzare un'interfaccia web minimale **per** poter visualizzare i risultati della mia ricerca.
'Done': Poter vedere (in locale) una GUI web con l'essenziale per gestire le funzionalità della ricerca e filtraggio dei tweet (navigation bar, search bar per i tweet, scegliere il tipo di ricerca, ecc.)
- #41 Creare una partita con un tweet. (+ pt. 2 → #84)
Come giocatore di scacchi **voglio** poter creare una partita di scacchi con un tweet **per** poter sfidare chiunque mi risponda.
- #42 Interfacciare i file output dei tweet col sito web.
Come sviluppatore **voglio** far comunicare ed interagire i file .py con quelli .db/html **per** poter visualizzare in maniera più completa possibile il sito web.
'Done': Rendere la GUI interagibile coi codici .py affinché a schermo siano mostrati i tweet ottenuti dalla ricerca.
Test: Eseguendo una ricerca si dovrebbero ottenere gli stessi tweet che si otterrebbero eseguendola su Twitter.
- #45 Filtrare i tweet precedenti o successivi ad una data x. (+ pt. 2 → #86)
Come utente **voglio** poter scegliere una data limiti (x) **per** poter vedere i tweet (di una persona o riguardanti una determinata parola/ricerca) precedenti o successivi alla data x.
CA: La finestra temporale della raccolta dei tweet ha come limiti di tempo: 10 anni fa rispetto al giorno attuale ed "oggi" stesso.
'Done': Visualizzazione dei tweet partendo dalla scelta della data limite (precedenti o successivi).
Test: Inserendo le date limite si ottengono solo tweet condivisi in quel periodo.
- #46 Rendere interattivi i pulsanti del GUI
Come spettatore **voglio** poter filtrare le mie ricerche tramite dei parametri nella GUI **per** poter scegliere che tweet vengono mostrati dopo la mia ricerca.
'Done': Corretto funzionamento dei tasti relativi al filtraggio.
- #77 Stream tweet. (+ pt. 2 → #85)
Come utente **voglio** poter vedere i tweet in tempo reale cercando una parola (anche hashtag o utente, se precedentemente selezionato il filtro) **per** essere aggiornato sui nuovi tweet.
'Done': Corretto funzionamento della visualizzazione in tempo reale dei tweet riguardanti una determinata ricerca.
Test: Selezionando la funzione stream la pagina dei risultati viene aggiornata in caso di nuovi tweet.
- #100 Pagine dedicate alle epiche.
Come utente **voglio** avere una pagina dedicata ad ogni epica **per** poter visualizzare direttamente i tweet ed i contenuti relativi senza dover fare la ricerca.
'Done': Presenza di pagine dedicate alle epiche che mostrino i contenuti relativi.
Test: Selezionando una pagina dedicata si ottengono informazioni solo riguardo a quell'argomento.
- #106 Pagina dedicata #leredita.
Come spettatore del programma "L'eredità" **voglio** vedere raggruppate tutte le informazioni di #leredita **per** essere sempre aggiornato sui nuovi tweet ad ogni novità e vedere tutte le statistiche.
'Done': Visualizzare tutti i dati dedicati all'epica #1 e le statistiche della User Story #17.
- #114 Pagina dedicata #reazioneacatena.
Come spettatore del programma "Reazione a catena" **voglio** vedere raggruppate tutte le

informazioni di #reazioneacatena **per** essere sempre aggiornato sui nuovi tweet ad ogni novità e vedere tutte le statistiche.

- #121 Giocare a scacchi 1 vs 1.
Come giocatore di scacchi **voglio** poter avviare una partita **per** giocare 1 vs 1.
'Done': Possibilità di dare il via ad una partita 1 contro 1 tramite il nostro sito.
- #139 Ottimizzazione del codice.
Come sviluppatore **voglio** migliorare gli algoritmi, ridurre le chiamate API ed evitare chiamate di funzioni inutili **per** ottimizzare il codice.
'Done': Assenza totale dei tempi di attesa non necessari.
- #180 Grafico a barre con numero di tweet e unità di tempo
Come utente **voglio** vedere su un grafico a barre la distribuzione dei tweet cercati **per** analizzare quando sono più attivi gli utenti per quella ricerca.
'Done': Presenza del grafico con i relativi dati corretti.
Test: Dopo una ricerca posso vedere quanti tweet sono stati condivisi in determinati orari.

AGGIUNTE IN DATA 01 DIC '22:

- #72 Recuperare i punteggi e mostrarne una sintesi.
Come spettatore di #FantaCitorio **voglio** avere una raccolta settimanale dei punteggi **per** poterne visualizzare una sintesi.
'Done': Mostrare statistiche e tabelle (varie) relativa ai singoli turni ed al campionato generale; mostrare i dati opzionali richiesti dal cliente (best climber, best single score, ecc.)
- #73 Visualizzazione classifica #FantaCitorio
Come spettatore di #FantaCitorio **voglio** poter visualizzare una classifica cumulativa dei punteggi **per** vedere i punti relativi ad ogni politico.
'Done': Poter visualizzare la classifica dei politici aggiornata all'ultima giornata; possibilità di modifica della classifica da parte dell'utente.
- #74 Sfogliare le immagini delle squadre registrate
Come spettatore di #FantaCitorio **voglio** sfogliare le immagini delle squadre registrate **per** vedere chi partecipa.
CA: La visualizzazione delle squadre deve essere esposta in maniera ordinata (ad esempio, tramite lo scorrimento di immagini)
'Done': Deve essere possibile, per l'utente, visualizzare a schermo le squadre del #FantaCitorio.
- #75 Trovare squadra da utente.
Come spettatore di #FantaCitorio **voglio** poter cercare un utente **per** vedere se ha una squadra registrata (ed in quel caso mostrarla).
'Done': Dato un nome utente, viene mostrata l'immagine della sua squadra se rispetta i vincoli imposti dal regolamento del #FantaCitorio (condividere la foto su Twitter con l'hashtag #Fantacitorio e fissare il tweet al profilo).
- #115 Pagina dedicata #FantaCitorio.
Come spettatore di #FantaCitorio **voglio**:
 - vedere raggruppate tutte le informazioni relative **per** essere sempre aggiornato sulle classifiche.
 - poter cercare un utente **per** vedere se partecipa al gioco, ed eventualmente visualizzare la sua squadra.
 - poter sfogliare le immagini contenenti le squadre **per** vedere come sono composte.
- #139 Ottimizzazione del codice.
Come sviluppatore **voglio** migliorare gli algoritmi dove possibile e ridurre le chiamate alle API **per** ottimizzare il codice evitando chiamate di funzioni inutili.
'Done': Assenza di tempi di attesa non richiesti/previsti.

2.6 Sprint Backlog

User Story #22 Partita a Scrumble.

Come team **voglio** giocare una partita a Scrumble **per** comprendere e, di conseguenza, assegnare i ruoli.

'Done': Completare una partita ed aver compreso pienamente i ruoli di Scrum.

User Story #23 Documentazione API Twitter

Come team **voglio** studiare le API di Twitter **per** poterle utilizzare negli sprint futuri.

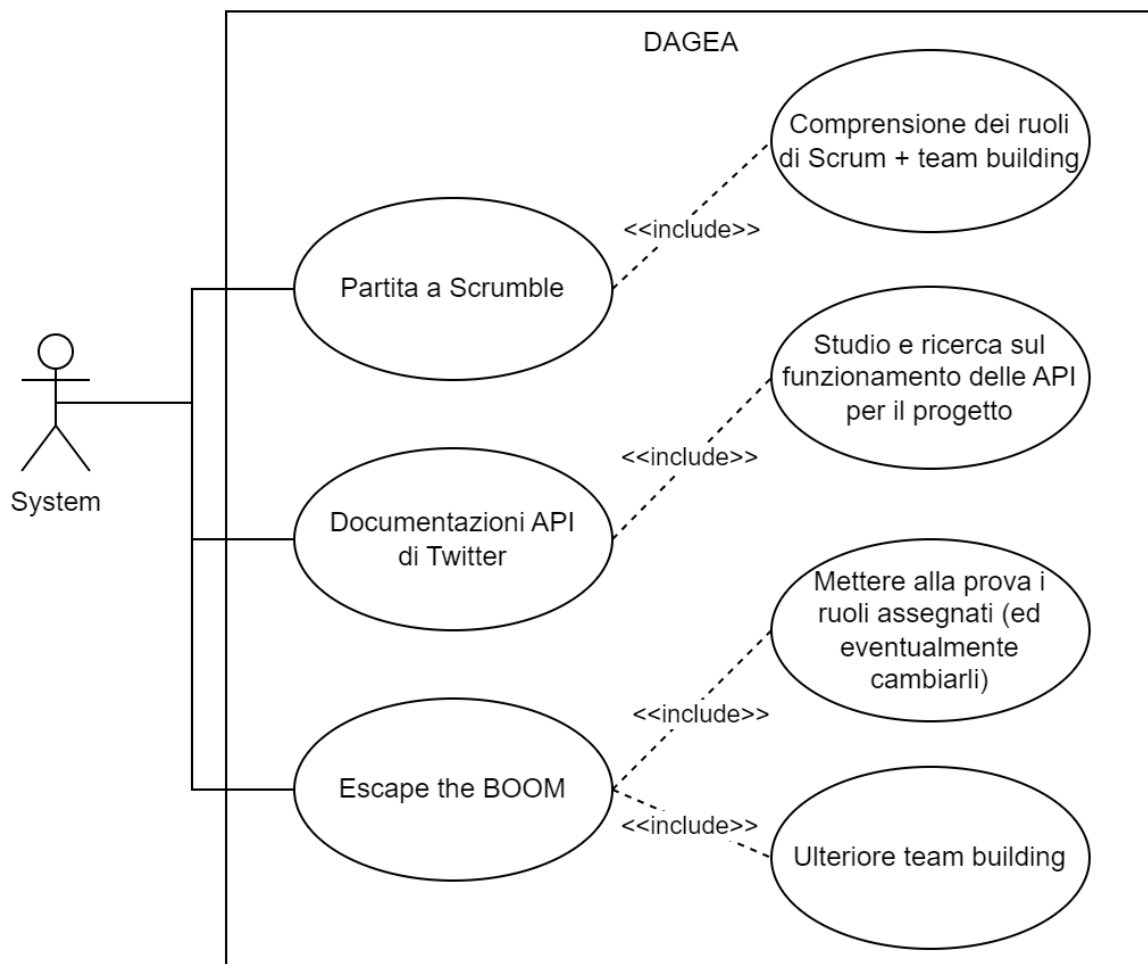
'Done': Iniziazione account Twitter Developer e comprensione documentazione di base.

User Story #30 Escape The BOOM

Come team **voglio** giocare una partita a Escape the BOOM **per** mettere alla prova i ruoli e fare ulteriore team building.

'Done': Completare una partita e confermare il corretto assegnamento dei ruoli.

2.7 Diagramma UML



2.8 Escape The BOOM

In data 02 nov. 22 abbiamo giocato ad Escape The BOOM, gioco alternativo a Scrumble proposto dalla tesista Francesca Chiriaco. Abbiamo giocato online e lo abbiamo trovato molto comodo poiché

il Product Owner ha potuto silenziare il Developer Team, sentendo così solamente le indicazioni riportate dallo Scrum Master (che si è occupato di intermediare).

Per cominciare il PO ha installato l'app sul proprio dispositivo, e al Developer Team è stato reso disponibile il manuale. Sapendo che i livelli erano 4 ed il tempo massimo per la soluzione di ognuno 5 min, il PO ha stimato un totale di 20 min per completare il gioco. A questo punto abbiamo dato il via al **livello 1**, il quale ci ha richiesto solamente 29 secondi per essere risolto. Siamo subito passati al **livello 2** ovvero l'unico che ci ha dato problemi: abbiamo fatto esplodere la bomba due volte; la prima dopo 1 minuto di tentativi, la seconda dopo solamente 3 secondi. Abbiamo quindi tentato per una terza volta e ci abbiamo impiegato 2 minuti e 57 secondi; per un totale di 4 minuti per il livello. La risoluzione del **livello 3** è quella che è durata maggiormente: 4 minuti e 21 secondi. Infine, per il **livello 4** ci abbiamo messo 4 minuti e 16 secondi.

LIVELLO 1	LIVELLO 2	LIVELLO 3	LIVELLO 4	TOTALE
29 sec.	4 min.	4 min. 21 sec.	4 min. 16 sec.	13 min. 06 sec.

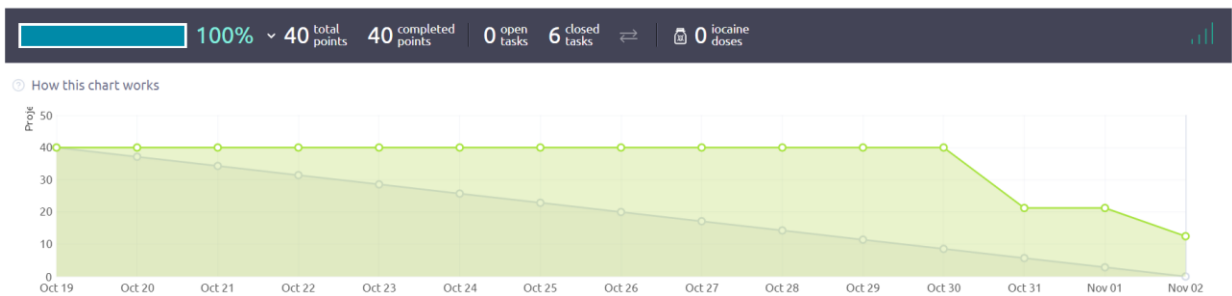
Siamo quindi rimasti entro il tempo massimo stimato dal Product Owner prima di iniziare il gioco, ci abbiamo anzi messo 6 minuti e 54 secondi in meno.

2.9 AUTOVALUTAZIONE:

GOAL	QUESTIONS	EVALUATION	Alice Altena	Gabriele Buttinoni	Denis Gjonai	Alice Benatti	Emanuele Campanelli
Apprendimento	Q1	1 = non hanno capito i ruoli di Scrum 5 = conoscenza perfetta del proprio ruolo	5	5	5	5	5
	Q2	1 = non si riesce a replicare il gioco 5 = può giocare prendendo il ruolo di Scrum Master	5	5	5	5	5
	Q3	1 = totalmente perso 5 = guida il gioco anche per gli altri membri	5	5	5	5	5
Pratica	Q4	1 = crede che il gioco non possa essere replicato 5 = crede che il gioco possa essere eseguito anche in altre situazioni	5	5	5	5	5
	Q5	1 = nessun livello superato 5 = tutti i livelli superati	5	5	5	5	5
	Q6 SOLO PO	1 = la stima è completamente diversa dal risultato finale 5 = risultato finale corrisponde alla stima effettuata	5				
Cooperazione	Q7	1 = non parla con gli altri giocatori 5 = parla con tutti	5	5	5	5	5
	Q8	1 = non fa nessuno sforzo per collaborare 5 = sempre pronto a capire cosa sta succedendo	5	5	5	5	5
	Q9	1 = non ascolta le opinioni degli altri 5 = usa ogni argomento per creare una discussione	5	5	5	5	5
Motivazione	Q10	1 = non è interessato al gioco 5 = controlla sempre che ognuno dica la sua	5	5	5	5	5
	Q11 SOLO PO	1 = spiegazioni confuso 5 = spiegazioni dettagliate e coerenti	4				
	Q12	1 = non esprime pareri su come vincere 5 = le proprie idee portano ad una soluzione vincente	5	5	5	5	5
Problem Solving	Q13	Se il numero di volte in cui la bomba è esplosa è > 4, allora mettere 1, se la bomba non è mai esplosa allora mettere 5	3	3	3	3	3
	Q14 SOLO DEV TEAM	Se per ogni livello la bomba è esplosa più di 2 volte mettere 1, se non è mai esplosa mettere 5	PO	SM	4	4	4
	Q15 SOLO PO	Stessa valutazione di Q14 ma fatta solo dal PO	4				

2.10 Burndown

Sprint 1 19 Oct 2022 to 02 Nov 2022



2.11 Retrospettiva



3 Fase di esecuzione – Sprint 2 (03/11 – 16/11)

3.1 Sprint goal

Ci siamo posti l'obiettivo di iniziare a lavorare con i dati forniti da Twitter per poi renderli visibili in un'interfaccia web minimale in modo tale da verificarne il corretto funzionamento. Inoltre, vogliamo assolutamente attivare l'ultimo strumento CAS mancante, ovvero SonarQube.

3.2 Sprint Backlog

User Story #5 Raccogliere i Tweet per temporalità.

Come spettatore di un gioco televisivo **voglio** avere una raccolta di tweet del passato **per** analisi storiche temporali.

Done: Ottenere un file contenente i tweet (ad esempio .db, .html, ecc.)

User Story #39 Creare un'interfaccia web minimale.

Come utente **voglio** utilizzare un'interfaccia web minimale **per** poter visualizzare i risultati della mia ricerca.

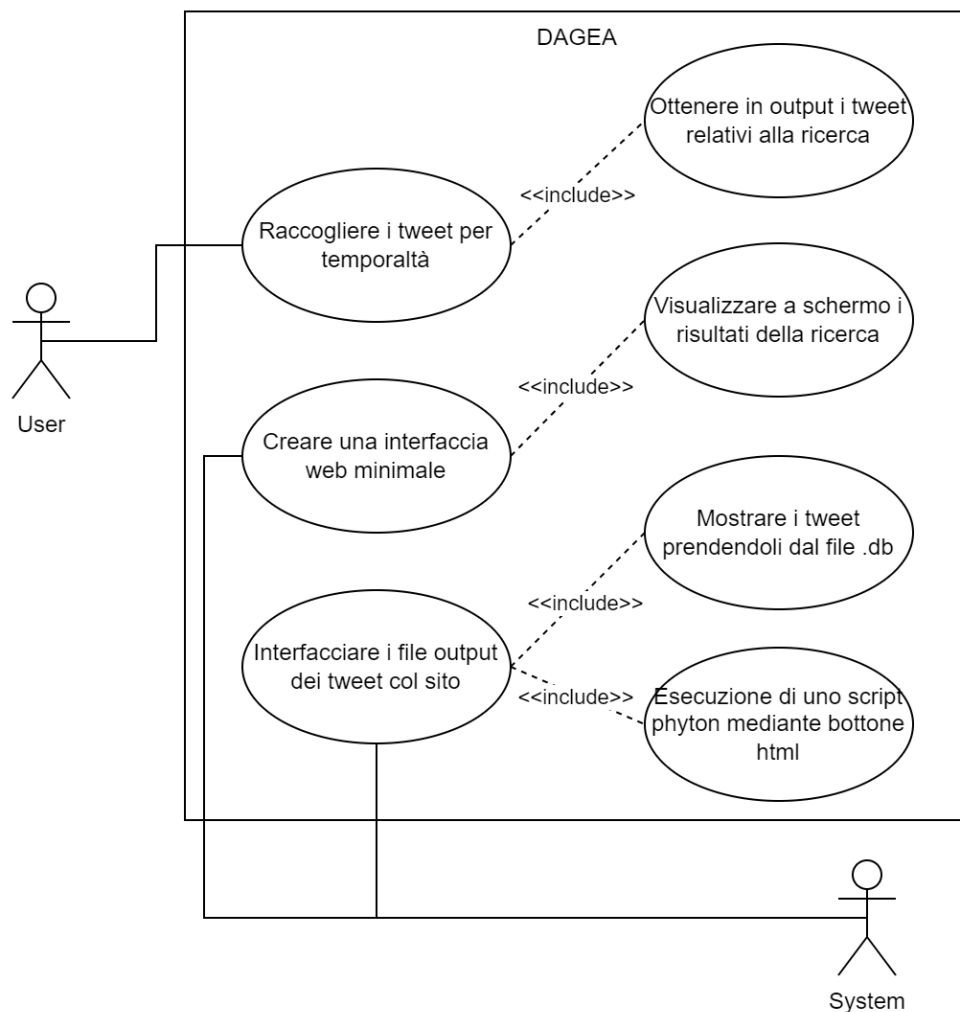
‘Done’: Poter vedere (in locale) una GUI web con l’essenziale per gestire le funzionalità della ricerca e il filtraggio dei tweet (navigation bar, search bar per i tweet, scegliere il tipo di ricerca, ecc.)

User Story #42 Interfacciare i file output dei tweet col sito web.

Come sviluppatore **voglio** far comunicare ed interagire i file .py con quelli .db/html **per** poter visualizzare in maniera più completa possibile il sito web.

‘Done’: Rendere la GUI interagibile coi codici .py affinché a schermo siano mostrati i tweet ottenuti dalla ricerca.

3.3 Diagramma UML



3.4 SonarQube

In questo sprint (in data 15 nov. 22) abbiamo concluso l’attivazione di tutti gli strumenti CAS impostando SonarQube per valutare lo sviluppo del nostro progetto. Avendo ancora poche linee di codice nel branch “main” (ovvero quello principale), questo è ciò che abbiamo potuto vedere.

Team10

Passed

Last analysis: 21 minutes ago

PRIVATE

Bugs

0

A

Vulnerabilities

0

A

Hotspots Reviewed

—

A

Code Smells

2

A

Coverage

0.0%

Duplications

0.0%

Lines

36

XS

Python

3.5 Burndown

Sprint 2 03 Nov 2022 to 16 Nov 2022



3.6 Retrospettiva



4 Fase di esecuzione – Sprint 3 (17/11 – 01/12)

4.1 Sprint goal

In questo sprint vogliamo implementare diverse funzionalità che si basano su altre già attive, ad esempio vorremo aggiungere alla visualizzazione base dei tweet che abbiamo già quella di poterli vedere in continuo aggiornamento, o di poter selezionare il periodo di tempo dei tweet che vogliamo visualizzare.

4.2 Sprint Backlog

User Story #11 Raccolta Tweet per geolocalizzazione.

Come spettatore di un gioco televisivo **voglio** avere una raccolta di tweet di #leredita e #reazioneacatena **per** visualizzarli su una mappa.

CA: L'utente che twitta con #leredita e/o #reazioneacatena deve aver attivato la funzione di condivisione della posizione.

'Done': Tramite l'inserimento di una zona geografica (ad esempio: una città), ritorna i tweet provenienti da tale area.

User Story #41 Creare una partita con un tweet.

Come giocatore di scacchi **voglio** poter creare una partita di scacchi con un tweet **per** poter sfidare

chiunque mi risponda.

'Done': Possibilità di giocare una partita a scacchi a partire da un tweet.

User Story #45 Filtrare i tweet precedenti o successivi ad una data x.

Come spettatore **voglio** poter scegliere una data limite (x) **per** poter vedere i tweet (di una persona/riguardanti una determinata parola/ricerca) precedenti o successivi alla data x.

CA: La finestra temporale della raccolta dei tweet ha come limiti dell'intervallo di tempo 10 anni fa (rispetto al giorno corrente) ed il giorno corrente stesso.

'Done': Visualizzazione dei tweet partendo dalla scelta della data limite (precedenti/successivi).

User Story #46 Rendere interattivi i pulsanti della GUI.

Come spettatore **voglio** poter filtrare le mie ricerche tramite dei parametri nella GUI **per** poter scegliere che tweet vengono mostrati dopo la mia ricerca.

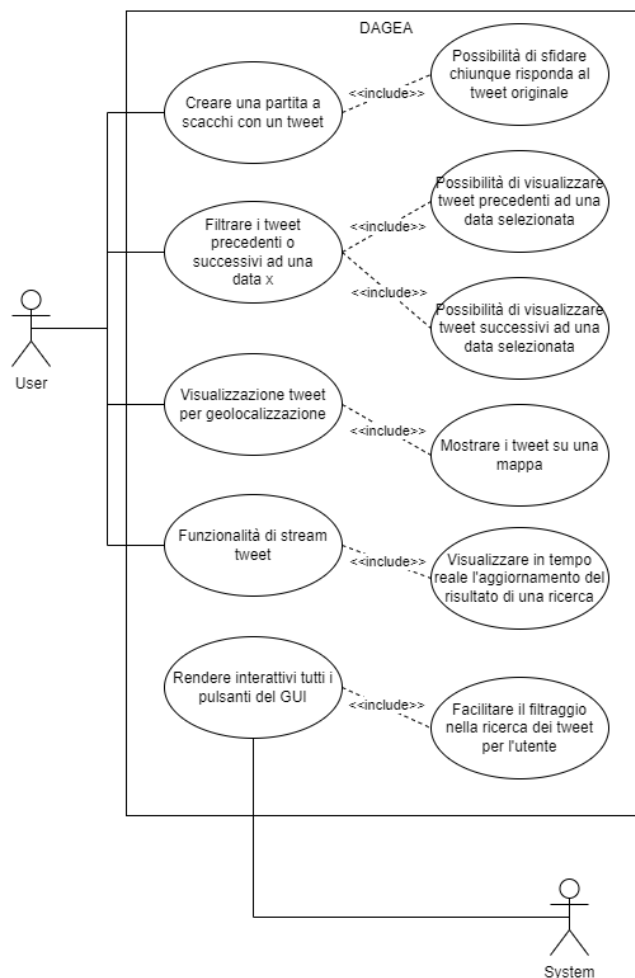
'Done': Corretto funzionamento dei tasti relativi al filtraggio.

User Story #77 Stream tweet.

Come utente **voglio** poter vedere i tweet in tempo reale cercando una parola **per** essere aggiornato sui tweet appena creati.

'Done': Corretto funzionamento della visualizzazione in tempo reale dei tweet riguardanti una determinata ricerca.


4.3 Diagramma UML




4.4 Mattermost + Taiga integrations


In data 21 nov. '22 abbiamo attivato il servizio di notifiche di Mattermost per GitLab. Tramite i webhook di Mattermost e le integrazioni di GitLab abbiamo impostato le notifiche automatiche riguardanti i push, gli issues e le merge request che avvengono su GitLab. Abbiamo precedentemente impostato diversi canali per suddividere le notifiche.

November 21


 **alice.altena BOT** 11:05 AM
[Team10 / Team10] Issue #5 Prova per Taiga Webhook opened by Alice Altena (alice.altena)

#5 Prova per Taiga Webhook
Niente di importante, sto solo provando a vedere se funziona il collegamento agli issues di Taiga.

 1





 **alice.altena BOT** 9:48 AM
Alice Benatti (alice.benatti4) unapproved merge request [!5 Html basic main page](#) in [Team10 / Team10](#)

November 21

 **alice.altena BOT** 9:39 AM
Alice Altena pushed to branch [main](#) of [Team10 / Team10](#) (Compare changes)


[011ce1a1](#): Update sonar-project.properties - Gabriele Buttinoni
[1867a189](#): Merge branch 'html-basic-main-page' into 'main' - Alice Altena
[5a491434](#): Merge branch 'main' into 'html-basic-main-page' - Alice Benatti


Abbiamo inoltre attivato l'integrazione one-way tra Taiga e GitLab, la quale permette la creazione automatica di un issue su Taiga una volta che questo viene creato su GitLab.


   #62 Prova per Taiga Webhook Closed ▾ 21 Nov 2022 


#62 Prova per Taiga Webhook < >

ISSUE

[Add tag +](#) Created by GitLab 21 Nov 2022 11:05 

 **GitLab** 21 Nov 2022 11:06
Issue closed by [alice.altena](#) from [GitLab#5](#).

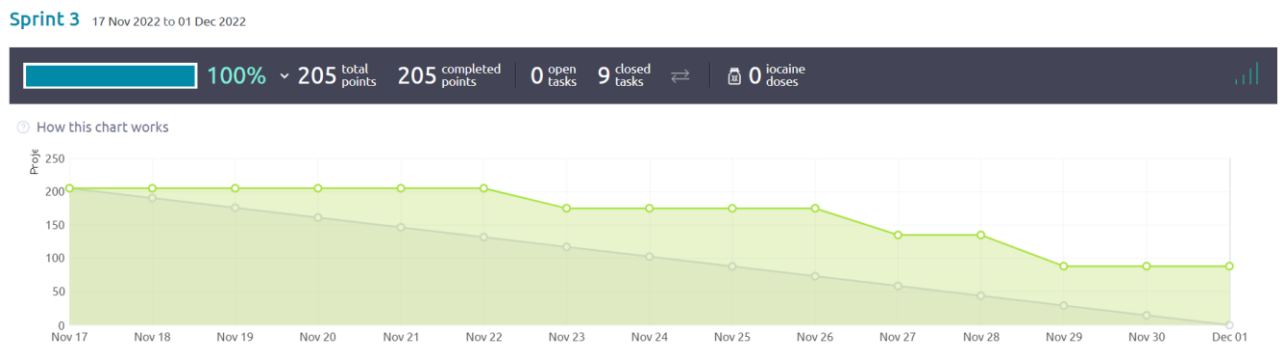
 **GitLab** 21 Nov 2022 11:06
[alice.altena](#) says in [GitLab#5](#):
"Aggiornamento: funziona!"

 **GitLab** 21 Nov 2022 11:05
Issue created by [alice.altena](#) from [GitLab#5](#).

4.5 Aspetto GUI



4.6 Burndown



4.7 Retrospettiva



4.8 Aggiunta epica FantaCitorio

In data 01 dic. '22 ci è stata richiesta con priorità di una nuova epica riguardante il FantaCitorio (gioco di Propaganda Live, in onda su La7), essendo l'ultimo giorno dello sprint 3 abbiamo deciso di iniziare a lavorarci ufficialmente dallo sprint 4. Abbiamo dunque aggiornato il product backlog aggiungendo:

EPICA:

- #71 Raccolta tweet riguardanti #FantaCitorio:
Come spettatore di #FantaCitorio (Propaganda Live su La7) **voglio** poter accedere ai dati riguardanti il gioco **per** visualizzare i punteggi e le squadre partecipanti.

USER STORIES:

- #72 Recuperare i punteggi e mostrarne una sintesi.
Come spettatore di #FantaCitorio **voglio** avere una raccolta settimanale dei punteggi **per** poterne visualizzare una sintesi.
- #73 Visualizzazione classifica #FantaCitorio
Come spettatore di #FantaCitorio **voglio** poter visualizzare una classifica cumulativa dei punteggi **per** vedere i punti relativi ad ogni politico.
- #74 Sfogliare le immagini delle squadre registrate
Come spettatore di #FantaCitorio **voglio** sfogliare le immagini delle squadre registrate **per** vedere chi partecipa.
- #75 Trovare squadra da utente.
Come spettatore di #FantaCitorio **voglio** poter cercare un utente **per** vedere se ha una squadra registrata (ed in quel caso mostrarla).

5 Fase di esecuzione – Sprint 4 (02/12 – 16/12)

5.1 Sprint goal

Il goal è terminare le tasks e le user stories restanti dallo sprint precedente, inoltre abbiamo deciso di creare una pagina dedicata ad ogni epica per rendere più semplice la raccolta delle informazioni per l'utente interessato ad un determinato argomento. Nello specifico abbiamo iniziato dalla pagina dedicata all'eredità.

5.2 Sprint Backlog

In questo sprint abbiamo duplicato delle User Stories dello sprint precedente (aggiungendo un "pt. 2" in fondo al nome della US), poiché erano caratterizzate da un numero elevato di tasks che erano state portate a termine solo in parte. Dunque, per non far risultare come nullo il lavoro svolto nello Sprint 3 abbiamo pensato di lasciare come fatte le User Stories con le relative tasks completate, e di spostare nello Sprint 4 solo le tasks non concluse.

User Story #86 Filtrare i tweet precedenti o successivi ad una data x (pt. 2).

Come spettatore **voglio** poter scegliere una data limite (x) **per** poter vedere i tweet (di una persona/riguardanti una determinata parola/ricerca) precedenti o successivi alla data x.

CA: La finestra temporale massima è vincolata dal range delle API usate, in questo caso si ha una finestra temporale degli ultimi 7 giorni.

'Done': Filtrare i tweet che sono stati creati nel range temporale inserito dall'utente.

User Story #85 Stream tweet (pt. 2).

Come utente **voglio** poter vedere i tweet in tempo reale cercando una parola (anche hashtag o utente, se precedentemente selezionato il filtro) **per** essere aggiornato sui tweet appena creati.

‘Done’: Corretto funzionamento della visualizzazione in tempo reale dei tweet riguardanti una determinata ricerca.

User Story #100 Pagine dedicate alle epiche.

Come utente **voglio** avere una pagina dedicata ad ogni epica **per** poter visualizzare direttamente i tweet ed i contenuti relativi senza dover fare la ricerca.

‘Done’: Presenza di pagine dedicate alle epiche che mostrino i contenuti relativi.

User Story #106 Pagina dedicata #leredita.

Come spettatore del programma “L’eredità” **voglio** vedere raggruppate tutte le informazioni di #leredita **per** essere sempre aggiornato sui nuovi tweet ad ogni novità e vedere tutte le statistiche.

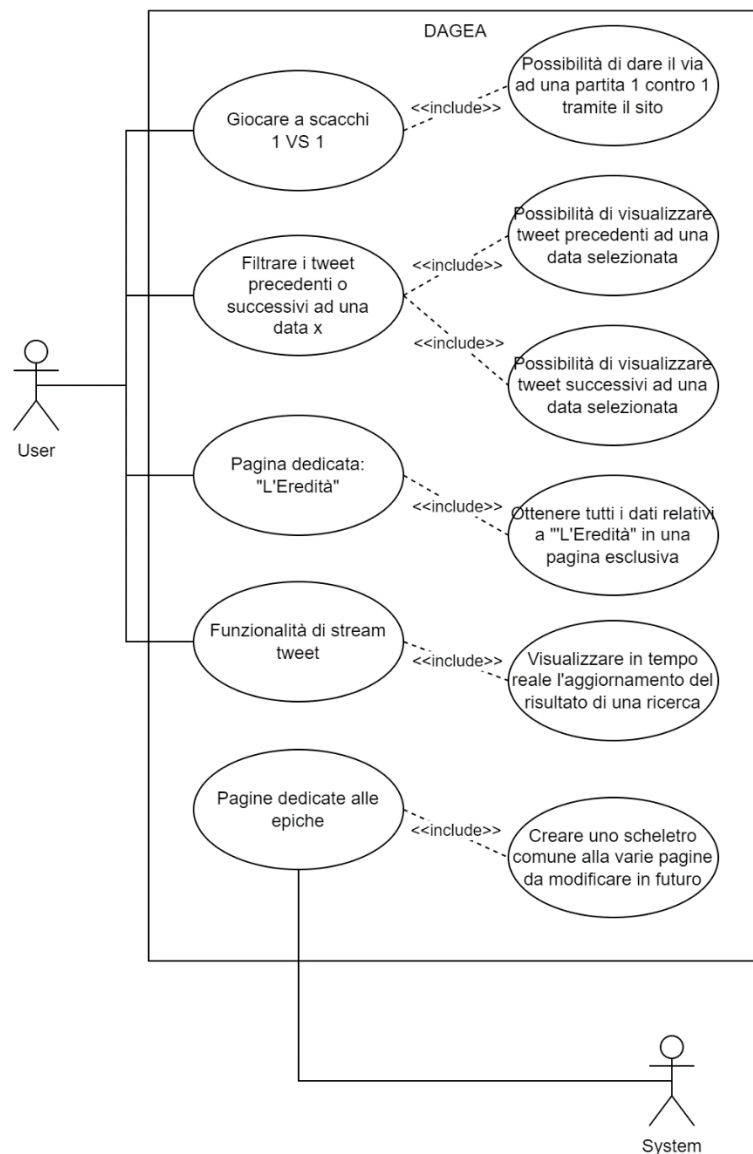
‘Done’: Visualizzare tutti i dati dedicati all’epica #1 e le statistiche della User Story #17.

User Story #121 Giocare a scacchi 1 vs 1

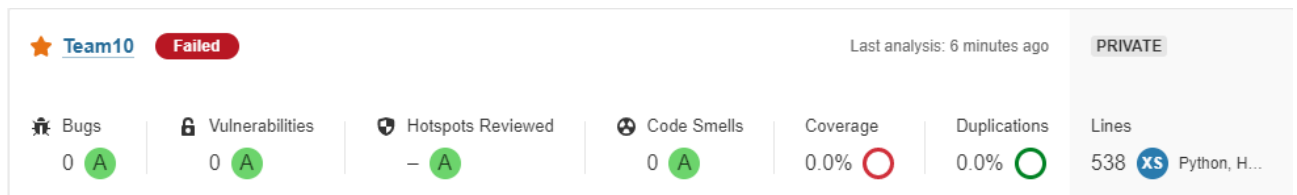
Come giocatore di scacchi **voglio** poter avviare una partita **per** giocare 1 vs 1.

‘Done’: Possibilità di dare il via ad una partita 1 contro 1 tramite il nostro sito.

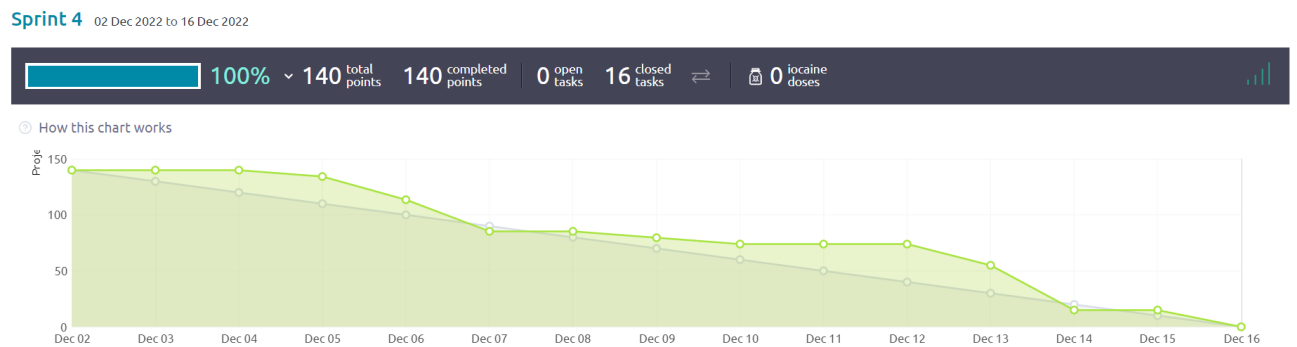
5.3 Diagramma UML



5.4 SonarQube (al 05/12)



5.5 Burndown



5.6 Retrospettiva



6 Fase di esecuzione – Sprint 5 (17/12 – 04/01)

6.1 Sprint goal

In questo sprint l'obiettivo principale era quello di implementare tutte le User Story relative al "Fantacitorio" in modo da terminare l'epica a riguardo. Abbiamo inoltre puntato a terminare l'unica User Story di cui erano rimaste delle tasks non completate, ovvero quella riguardo alla geolocalizzazione dei tweet. E infine abbiamo modificato e reso uniche le pagine dedicate al Fantacitorio ed a "Reazione a catena". Questo sprint è stato rallentato per via del periodo di Natale

(in cui tutti per qualche giorno hanno lavorato meno), e dall'imminente inizio della sessione di esami (cosa che verrà riscontata anche nei prossimi due sprint); nonostante ciò, siamo riusciti a terminare quanto prefissato.

6.2 Sprint Backlog

User Story #72 Recuperare i punteggi e mostrarne una sintesi.

Come spettatore di #FantaCitorio **voglio** avere una raccolta settimanale dei punteggi **per** poterne visualizzare una sintesi.

'Done': Mostrare statistiche e tabelle (varie) relative ai singoli turni ed al campionato generale; mostrare i dati opzionali richiesti dal cliente (best climber, best single score, ecc.).

User Story #73 Visualizzazione classifica #FantaCitorio

Come spettatore di #FantaCitorio **voglio** poter visualizzare una classifica cumulativa dei punteggi **per** vedere i punti relativi ad ogni politico.

'Done': Poter visualizzare la classifica dei politici aggiornata all'ultima giornata; possibilità di modifica della classifica da parte dell'utente.

User Story #74 Sfogliare le immagini delle squadre registrate.

Come spettatore di #FantaCitorio **voglio** sfogliare le immagini delle squadre registrate **per** vedere chi partecipa.

CA: La visualizzazione delle squadre deve essere esposta in maniera ordinata (ad esempio, tramite lo scorrimento di immagini).

'Done': Deve essere possibile, per l'utente, visualizzare a schermo le squadre del #FantaCitorio.

User Story #75 Trovare squadra da utente.

Come spettatore di #FantaCitorio **voglio** poter cercare un utente **per** vedere se ha una squadra registrata (ed in quel caso mostrarla).

'Done': Dato un nome utente, viene mostrata l'immagine della sua squadra se rispetta i vincoli imposti dal regolamento del #FantaCitorio (condividere la foto su Twitter con l'hashtag #Fantacitorio e fissare il tweet al profilo).

User Story #83 Raccolta tweet per geolocalizzazione pt. 2

Come spettatore di un gioco televisivo **voglio** avere una raccolta di tweet de #leredita e/o #reazioneacatena **per** visualizzarli su una mappa.

CA: L'utente che twitta con #leredita e/o #reazioneacatena deve aver attivato la funzione di condivisione della posizione.

'Done': Tramite l'inserimento di una zona geografica (ad esempio, una città) ritorna i tweet da tale località; mappa in cui visualizzare i mark dei luoghi taggati nei tweet ritornati dalla ricerca; i mark devono contenere: username, luogo taggato e testo del tweet di riferimento.

User Story #115 Pagina dedicata #FantaCitorio.

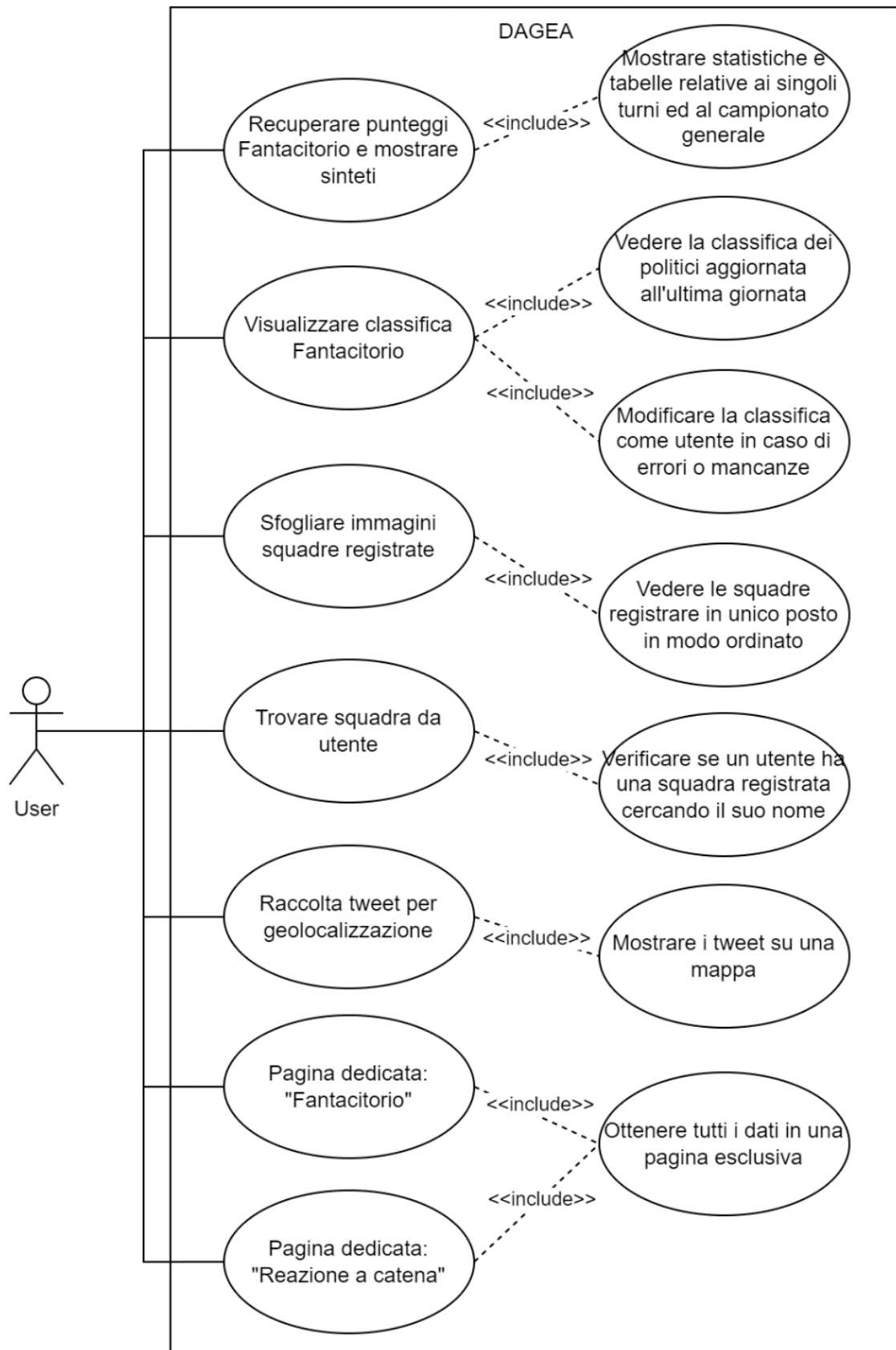
Come spettatore di #FantaCitorio **voglio**:

- vedere raggruppate tutte le informazioni relative **per** essere sempre aggiornato sulle classifiche.
- poter cercare un utente **per** vedere se partecipa al gioco, ed eventualmente visualizzare la sua squadra.
- poter sfogliare le immagini contenenti le squadre **per** vedere come sono composte.

User Story #114 Pagina dedicata #reazioneacatena.

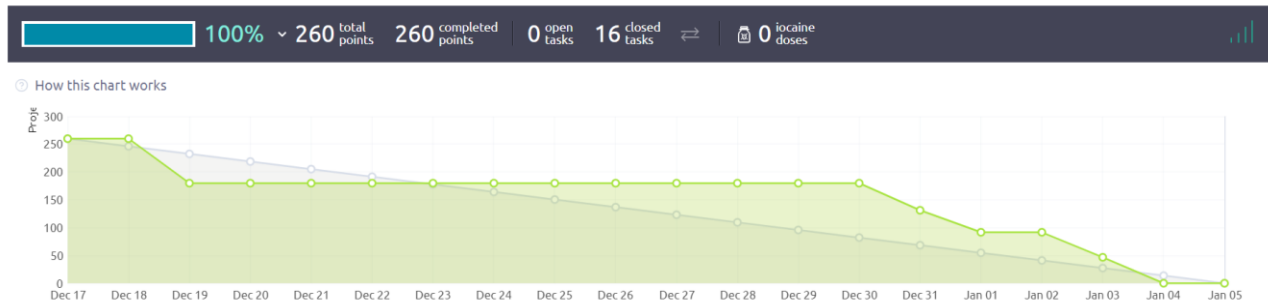
Come spettatore del programma "Reazione a catena" **voglio** vedere raggruppate tutte le informazioni di #reazioneacatena **per** essere sempre aggiornato sui nuovi tweet ad ogni novità e vedere tutte le statistiche.

6.3 Diagramma UML



6.4 Burndown

Sprint 5 17 Dec 2022 to 05 Jan 2023



6.5 Retrospettiva



7 Fase di esecuzione – Sprint 6 (05/01 – 20/01)

7.1 Sprint goal

Obiettivo dello sprint: terminare l'implementazione delle funzionalità utili a mostrare dati riguardanti le ricerche, come la sentiment analysis dei tweet e la tag cloud contenente i termini più utilizzati nei testi dei tweet ricercati. Infine, ottimizzare il codice per aver tempi di attesa più brevi. Dal burndown possiamo notare un periodo di fermo dal 09/01 fino alla fine dello sprint dato da User Stories aggiunte e su cui non si è lavorato che sono state spostate nello sprint successivo, causa di ciò l'esserci prolungati fino a questo periodo per concludere il progetto ed aver quindi riscontrato dei rallentamenti (come accennato precedentemente).

7.2 Sprint Backlog

User Story #9 Sentiment analysis dei tweet.

Come spettatore di un gioco televisivo **voglio** avere un grafico rappresentante le reazioni del pubblico riguardo al gioco **per** analizzare il sentimento generale dei tweet.

'Done': Deve essere mostrata nella GUI qualche forma di rappresentazione dei dati della sentiment analysis fatta sui tweet cercati (grafici, statistiche, ecc.).

User Story #20 Tag cloud dei Tweet

Come spettatore di un gioco televisivo **voglio** visualizzare una world cloud delle principali parole **per** vedere quelle più utilizzate nei tweet (riguardanti il mio interesse).

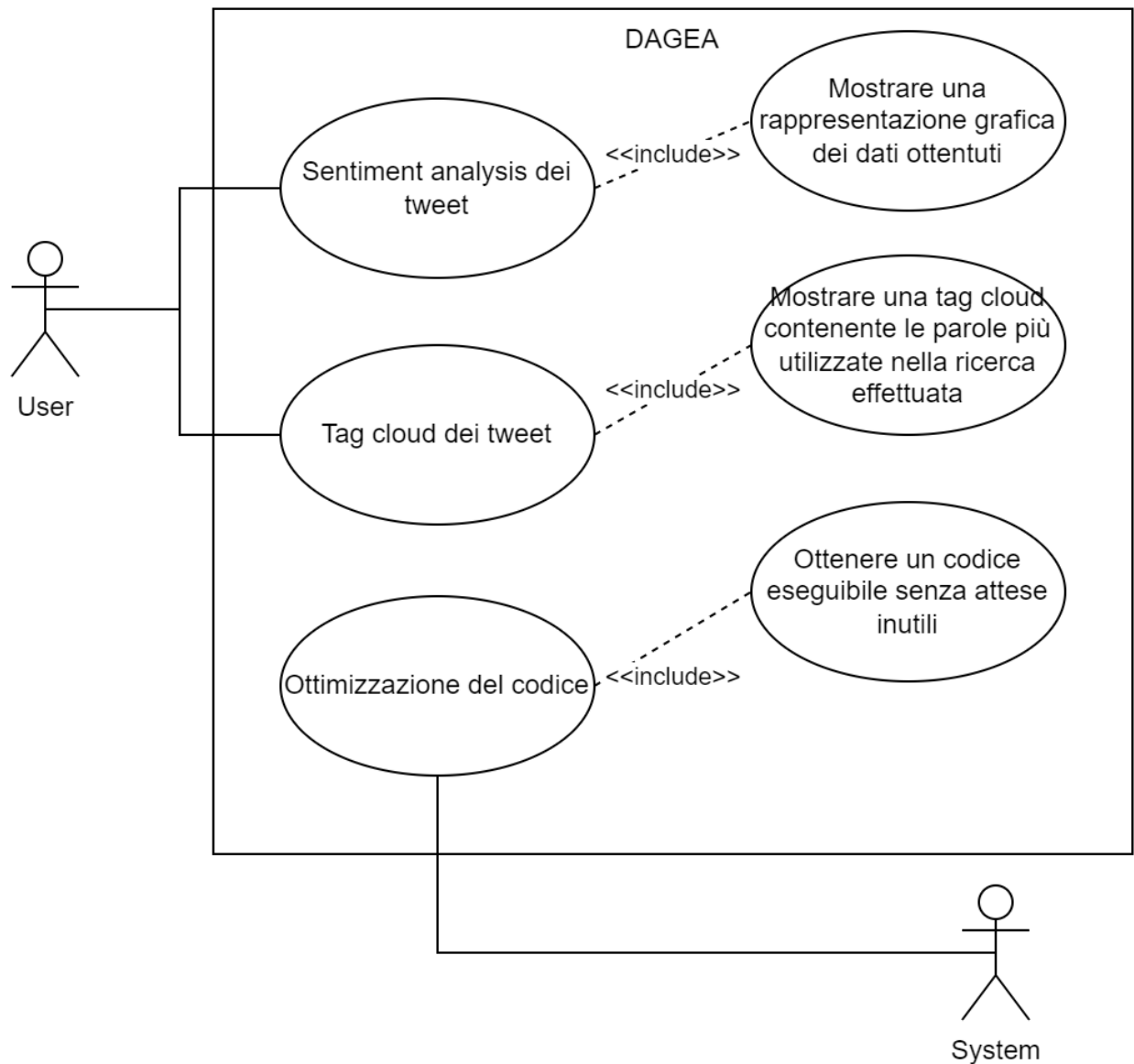
'Done': Visualizzare nel GUI una tag cloud contenente le parole più utilizzare nei testi dei tweet ricercati.

User Story #139 Ottimizzazione del codice.

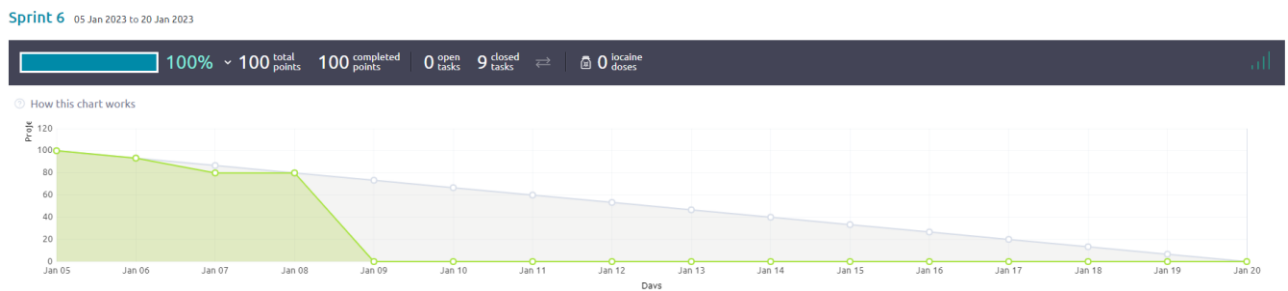
Come sviluppatore **voglio** migliorare gli algoritmi dove possibile e ridurre le chiamate alle API **per** ottimizzare il codice evitando chiamate di funzioni inutili.

'Done': Assenza di tempi di attesa non richiesti/previsti.

7.3 Diagramma UML



7.4 Burndown



7.5 Retrospettiva



8 Fase di esecuzione – Sprint 7 (21/01 – 14/02)

8.1 Sprint goal

Implementare le parti mancanti del progetto in vista della consegna, scrivere gli ultimi dettagli nella relazione e registrare e montare il video di presentazione.

8.2 Sprint Backlog

User Story #7 Raccogliere i tweet di scacchi in risposta alla mia mossa.

Come giocatore di scacchi **voglio** avere una raccolta di tweet di risposta alla mia mossa **per** visualizzarli in modo temporale.

User Story #16 Raccolta tweet di scacchi per sfidare in gruppo.

Come giocatore di scacchi **voglio** poter avere una raccolta di tweet relativi alla mia partita in una tabella/elenco **per** visualizzare le mosse della maggioranza in risposta al tweet.

User Story #17 Tweet #leredita indovinati.

Come spettatore di #leredita **voglio** avere i dati aggiornati all'ultima puntata **per** poter vedere la parola da indovinare e quante persone hanno indovinato.

'Done': Possibilità di vedere le parole del gioco dell'ultima puntata, con la parola da indovinare coperta in modo che si possa scegliere di vederla o meno. Grafico che mostri quante persone hanno indovinato sul totale che ha ci provato.

User Story #84 Creare una partita con un tweet (pt. 2).

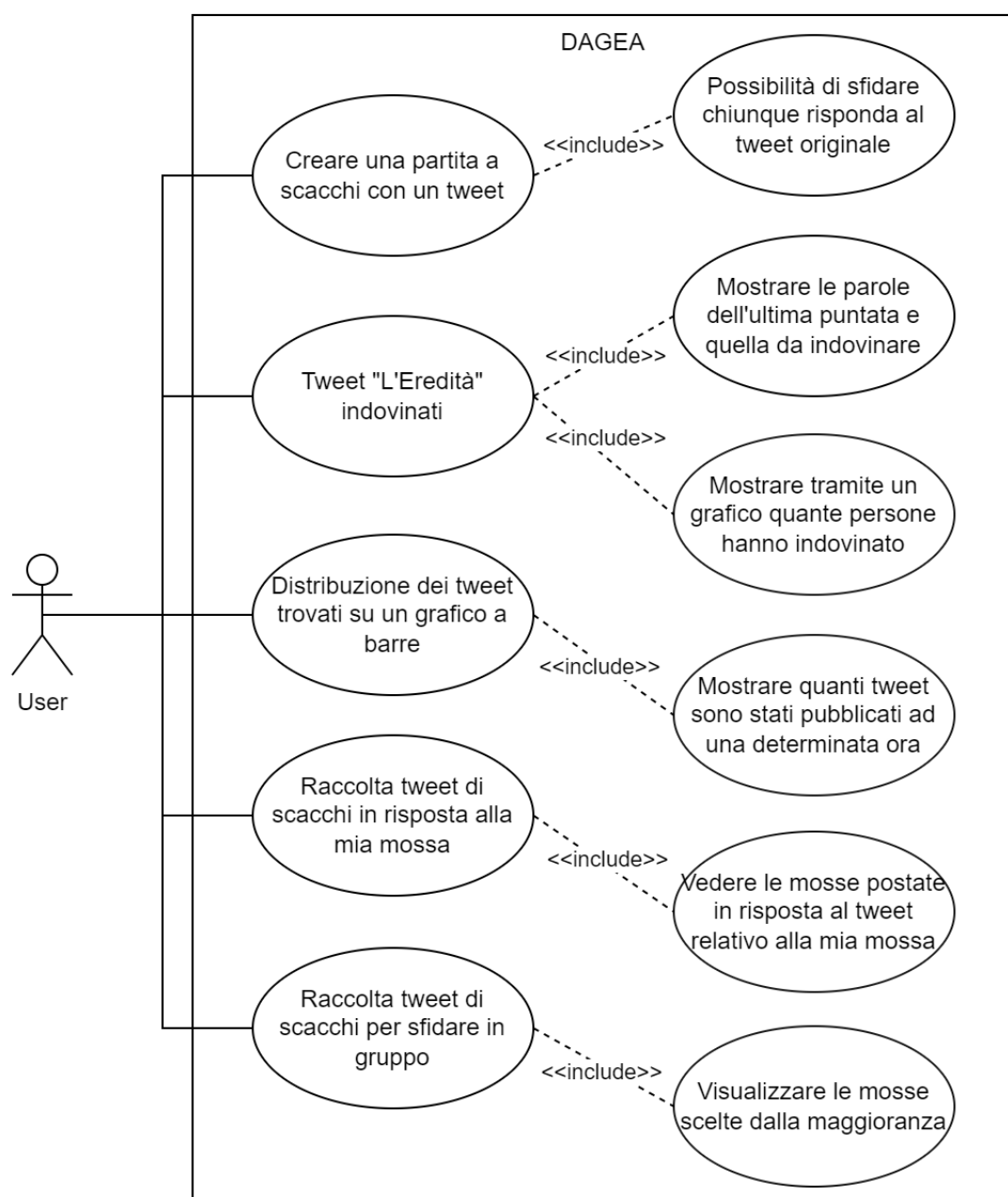
Come giocatore di scacchi **voglio** poter creare una partita di scacchi con un tweet **per** poter sfidare chiunque mi risponda.

User Story #180 Distribuire su un grafico a barre i tweet trovati.

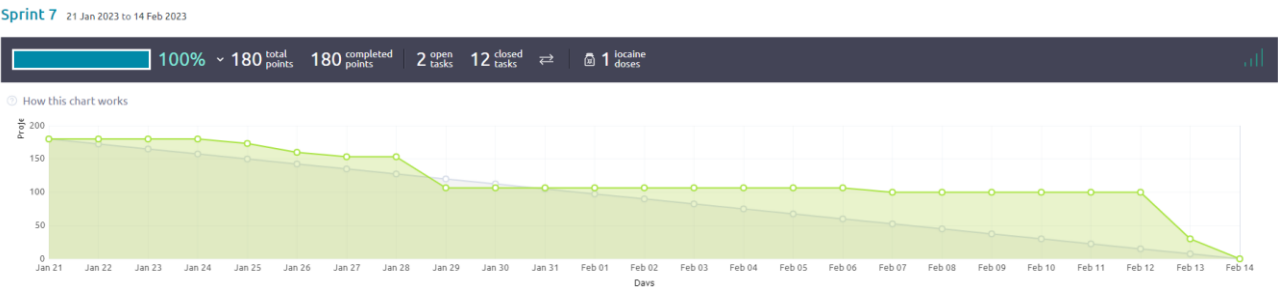
Come utente **voglio** vedere su un grafico a barre la distribuzione dei tweet cercati **per** analizzare quando sono più attivi gli utenti per quella ricerca.

'Done': Presenza del grafico con i relativi dati corretti.

8.3 Diagramma UML



8.4 Burndown



8.5 Retrospettiva



9 Fase conclusiva

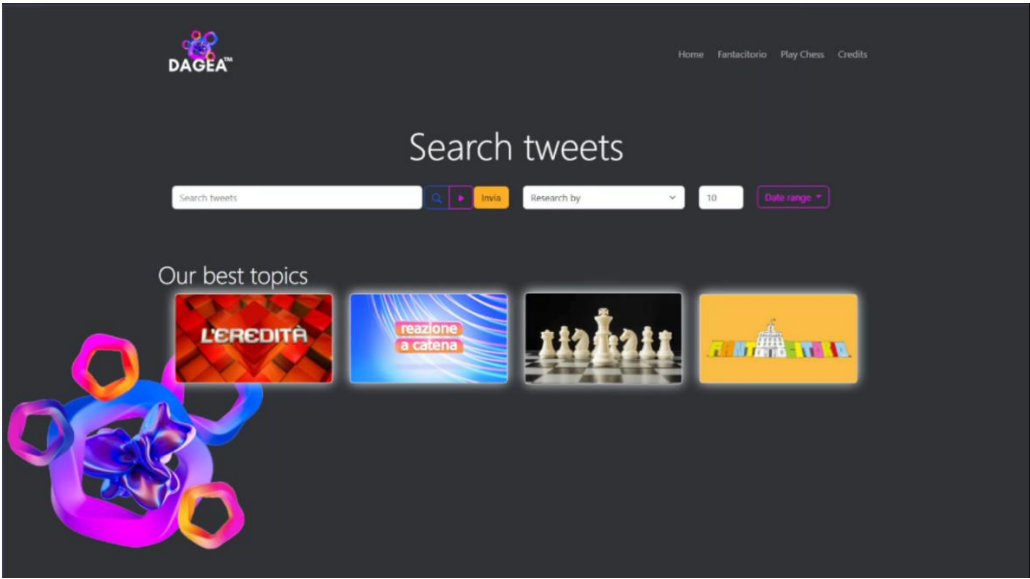
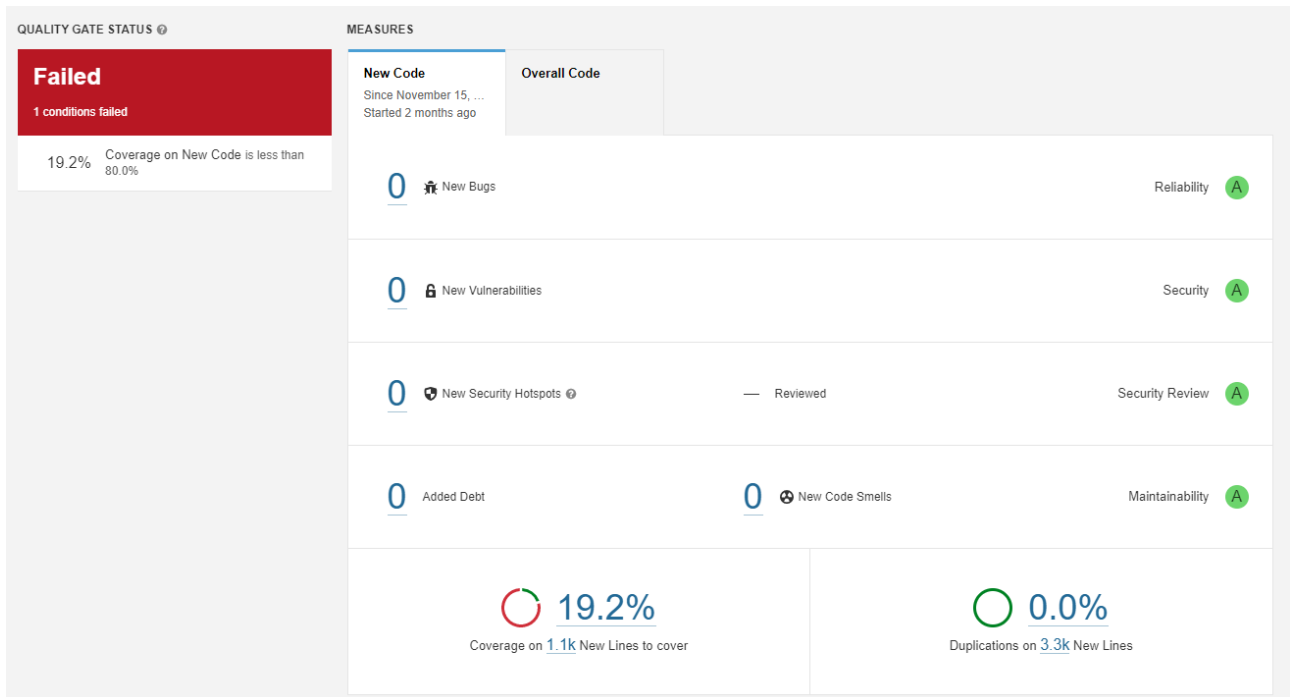


Figura 1: Schermata principale del prodotto

9.1 SonarQube

Possiamo vedere come nella schermata finale di SonarQube a progetto terminato l'unica cosa che produce "Failed" sia la bassa percentuale di coverage, ovvero codice coperto dai test; tuttavia ciò avviene poiché SonarQube non riconosce pienamente tutti i test, infatti di seguito avremo anche una schermata che prova che invece funzionano correttamente.



Coverage report: 42%

coverage.py v6.5.0, created at 2023-02-13 19:53 +0100

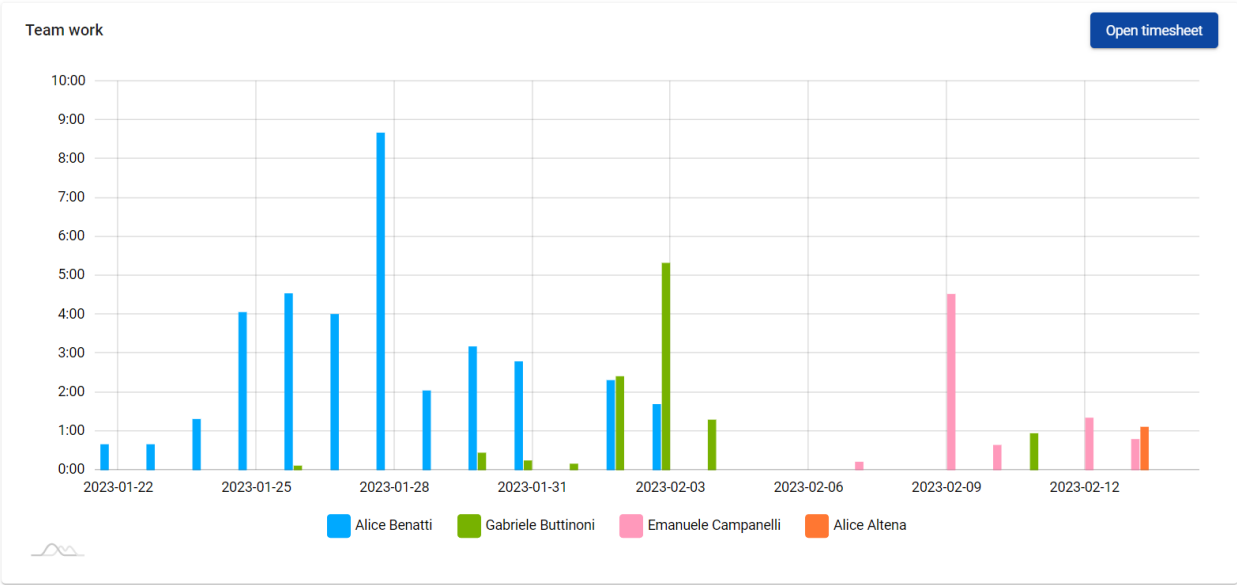
Module ↓	statements	missing	excluded	coverage
Sheet.py	49	1	0	98%
Sheet_test.py	40	3	0	92%
FantacitorioAnalysis.py	167	33	0	80%
FantacitorioAnalysis_test.py	27	1	0	96%
Fantacitorio.py	85	2	0	98%
Fantacitorio_test.py	22	1	0	95%
/usr/lib/python3/dist-packages/cffi/model.py	389	251	0	35%
/usr/lib/python3/dist-packages/cffi/lock.py	10	6	0	40%
/usr/lib/python3/dist-packages/cffi/error.py	19	8	0	58%
/usr/lib/python3/dist-packages/cffi/api.py	544	481	0	12%
/usr/lib/python3/dist-packages/cffi/_init_.py	7	0	0	100%
Total	1359	787	0	42%

Per alcuni servizi (ad esempio: l'eredità, la stream dei tweet, il grafico temporale, ecc.) i test non risultano funzionanti in quanto ci si appoggiava a tweet non più reperibili a causa del disservizio di Twitter.

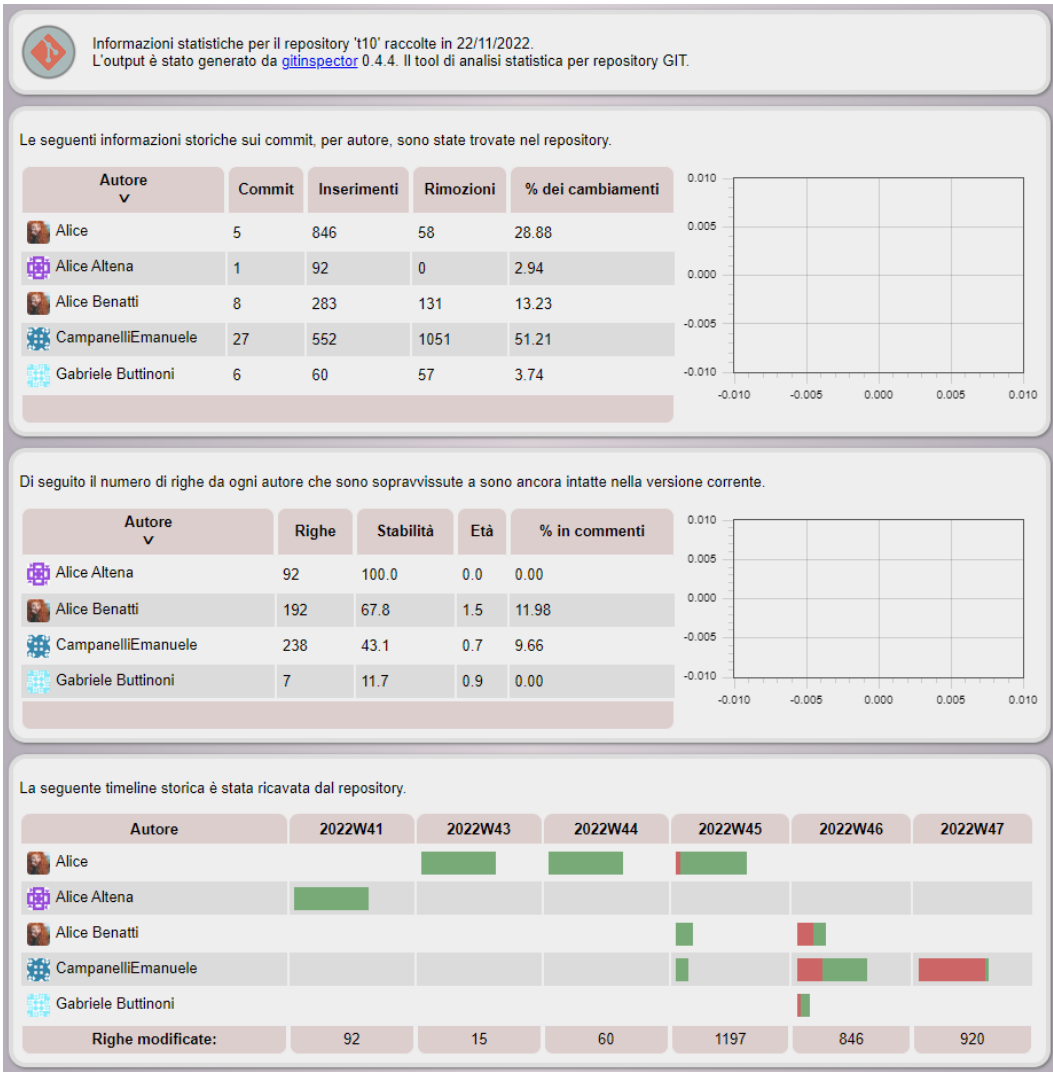
9.2 Logger

Non funzionando il logger consigliato abbiamo deciso di attivarne uno diverso in data 24/11/2022, tuttavia anche con questo logger abbiamo riscontrato problemi (soprattutto alla fine). Di seguito

infatti verranno riportati solo parte dei dati poiché non risultano segnate tutte le ore di lavoro effettivamente fatte. Inoltre, come si può notare facilmente, non risultano ore di lavoro di un componente che sono invece state effettuate.



9.3 Gitinspector



9.4 Retrospettiva finale



9.5 Artefatti

Gitlab: <https://git.hjkl.gq/team10/dagea>

Taiga: <https://taiga.hjkl.gq/project/dageatm/timeline>

SonarQube: https://qube.hjkl.gq/dashboard?id=team10_dagea_AYQZW1D8XMVnFi3f5j-Y

Mattermost: <https://chat.hikl.gq/dagea/channels/town-square>

Video presentazione: [Google Drive](#)

Canva (usato per fare le retrospettive): <https://www.canva.com/folder/FAFQs89iYMQ>

Sito: <http://87.3.48.86:5000/>

Per tutti gli strumenti CAS sono stati invitati entrambi i proff. all'interno del team.