

T4-reportfinale

Progetto anno 2021/2022 del team: t4

Descrizione del prodotto

Il prodotto viene presentato con il nome di Heimdallt4 e si presenta come una web-app.

Il problema che vogliamo risolvere con il nostro software riguarda il monitoraggio del traffico presso i web server apache, il cui scopo principale è analizzare il traffico in modo da ottimizzare le risorse del server, e in secondo luogo, trarre valutazioni a fini commerciali.

La nostra soluzione si basa sull'analisi dei file di log che vengono analizzati dal Backend per creare una distinzione tra 2 tipi di connessioni: Malevole, benevole.

La distinzione crea 2 entità che verranno immesse in un database e una entità che verrà riformattata, da qui il Frontend ha la possibilità di mostrare i dati che vengono raccolti per una migliore interpretazione.

I dati mostrati indicano che tipo di connessione viene fatta, ed eventualmente la sua negatività evidenziandola, inoltre dopo un accesso alla pagina web si possono vedere i dati nel dettaglio, analizzando a livello temporale cosa è successo al server.

Il database creato mantiene le seguenti informazioni:

- record dei file di log
- record delle connessioni

La nostra web-App risiederà all'interno di un server Linode, essa sarà resa raggiungibile tramite un URL che verrà generato ad ogni nuova connessione, il metodo di connessione si basa sulla creazione di un tunnel.

Il tunnel locale che viene generato espone la porta 3000 utilizzata dalla nostra interfaccia frontend e la rende raggiungibile dall'esterno.

Installazione ed esecuzione

Per installare il nostro prodotto su di un server bisogna prima di tutto scaricare la cartella git:

```
git clone git@dw.gnet.it:2022T4/2022t4.git
```

Successivamente bisogna avviare il backend spostandosi nella cartella code, poi in Heimdall dove risiede il codice sorgente del backend ed avviare spring:

```
./mvnw spring-boot:run
```

Una volta avviato il backend si può procedere con l'avvio del frontend:

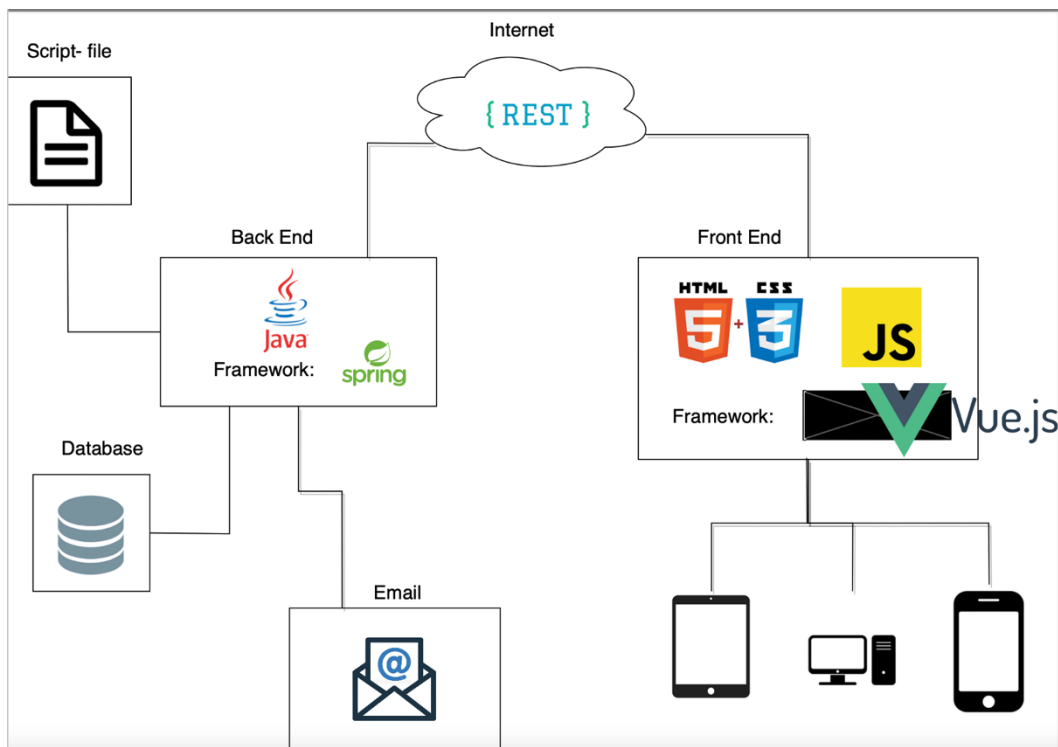
```
npm run dev
```

Successivamente per rendere accessibile da esterni il sistema appena avviato si deve avviare il processo di tunnelling sulla porta 3000, installando prima il sistema di tunneling:

```
sudo npm install -g localtunnel  
lt --port 3000
```

Architettura

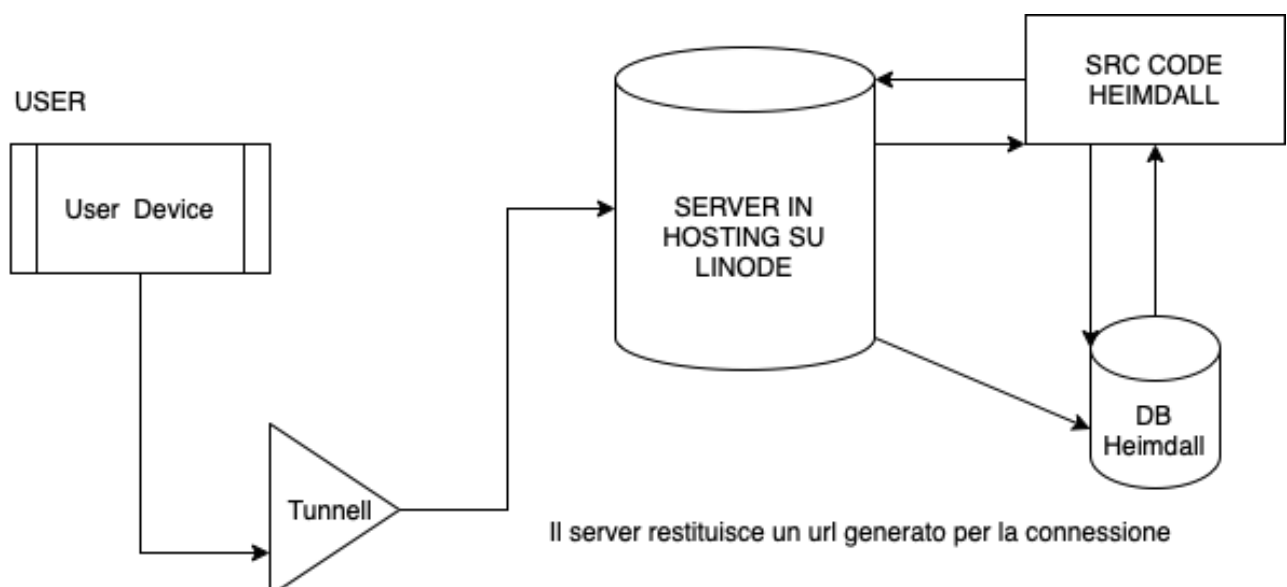
Al fine di illustrare con più precisione la nostra soluzione abbiamo deciso di riportare anche lo schema architetturale della nostra soluzione.



Vogliamo evidenziare come l'unica variazione architetturale sia derivata dalla scelta del framework poiché i membri del team che hanno lavorato al frontend hanno deciso di passare da react a vue.js in quanto più preparati con il secondo.

Per il deploy della nostra web-App abbiamo deciso di utilizzare un server Linode in cui si trova il nostro codice e il database, da questo server viene esposta la porta 3000 agli utenti che attraverso un URL generato ad ogni nuovo utilizzo si possono connettere.

Riportiamo in seguito lo schema per il deploy:



A seguire riportiamo i casi d'uso che ci hanno guidato nella realizzazione del prodotto e nell'analisi delle funzionalità da implementare.

Account:

- Come utente voglio accedere nella webapp per usare l'applicazione.

Ricerca:

- Come utente voglio definire una parola chiave per creare un filtro di ricerca testuale
- Come utente voglio definire un filtro complesso di ricerca usando più filtri singoli
- Come utente voglio isolare il traffico usando un filtro per una distinzione degli accessi
- Come utente voglio inserire una (sottodirectory del sito) per vederne il traffico.
- Come utente voglio definire un arco di tempo per vederne il traffico.

Visualizzare:

- Come utente voglio vedere in un'unica schermata tutto il log corrente, per una visione generale
- Come utente voglio vedere la sorgente di provenienza del traffico per analizzarlo.
- Come utente voglio definire un'area sulla mappa per vederne il traffico.
- Come utente voglio analizzare dati per prendere decisioni commerciali
- Come utente voglio analizzare dati per prendere decisioni amministrative (es. espandersi in Russia)
- Come utente vorrei poter analizzare quotidianamente le statistiche per un'analisi temporale più dettagliata

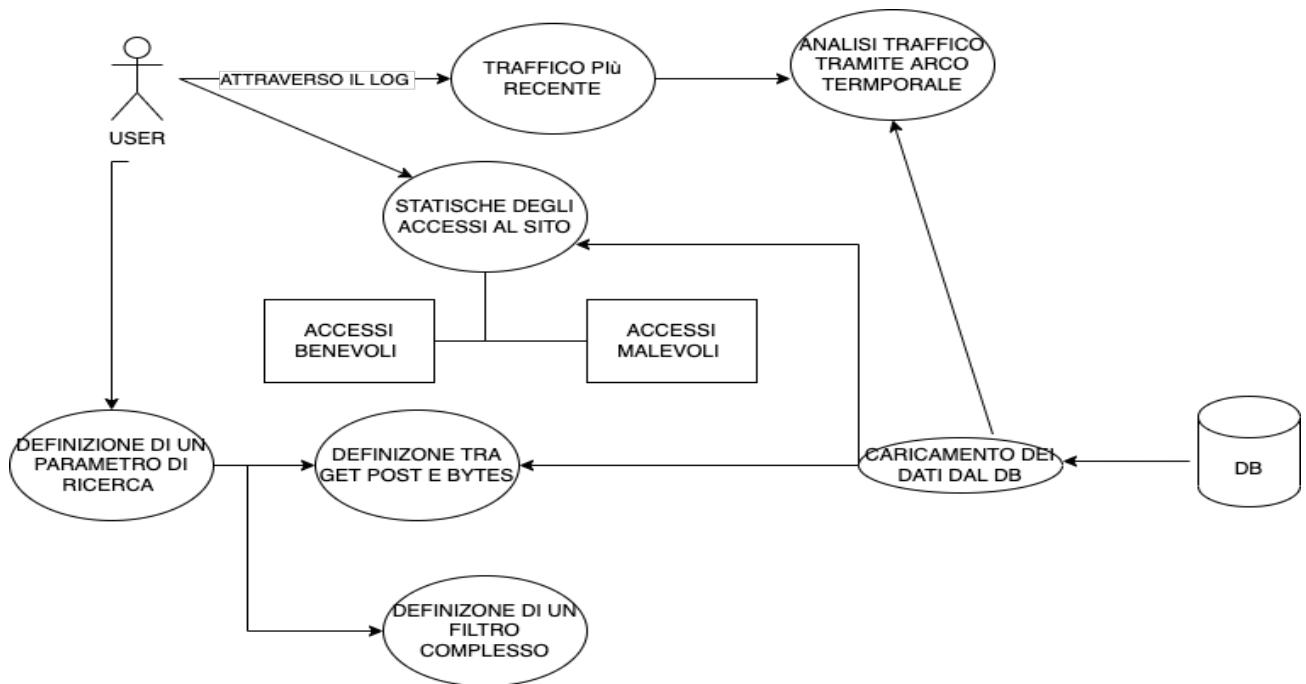
Eseguire:

- Come Superutente voglio bloccare il traffico da un indirizzo malevole per una maggior sicurezza.
- Come Sistema voglio analizzare il traffico per risolvere conflitti
- Come Sistema voglio fornire dati all'utente per poterli visualizzare
- Come Sistema voglio intervenire con un firewall per bloccare l'accesso in caso di necessità
- Come Sistema voglio allertare un utente per avvisarlo in caso di interventi
- Come sistema voglio allertare un utente per avvisarlo in caso di malfunzionamenti
- Come Sistema voglio salvare i dati in un database per poterli accedere in seguito

Account:

- Come Amministratore voglio creare utenti per farli accedere alla web-app
- Come Amministratore voglio eliminare utenti per evitare che possano accedere in caso non fossero più autorizzati
- Come Amministratore voglio creare un manuale d'uso per installare il backend
- Come Amministratore voglio creare una guida per l'uso del frontend.
- Come Amministratore voglio mantenere il sistema per renderlo utilizzabile

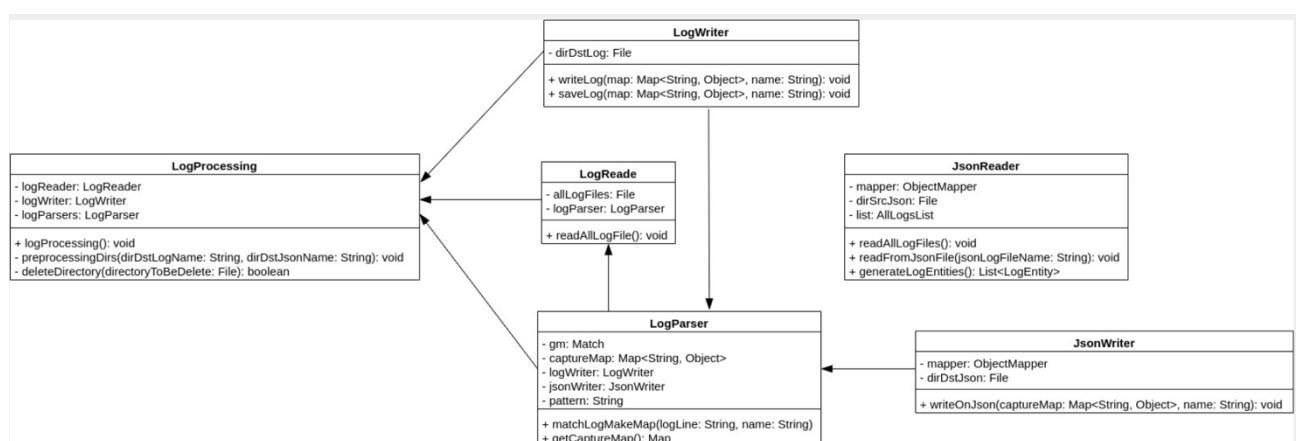
Riportiamo in seguito il diagramma dei casi d'uso:



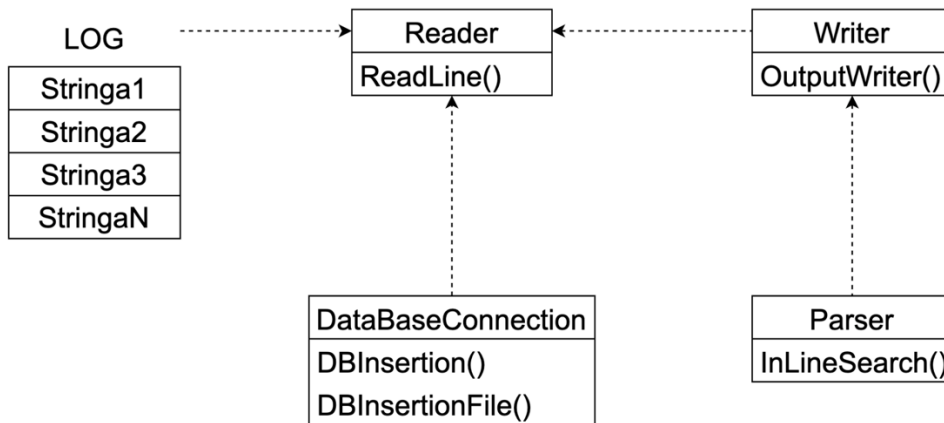
Dopo una fase iniziale di progettazione con il proseguimento degli sprint e della progettazione abbiamo rivisitato i casi d’uso del nostro progetto eliminando i casi che richiedevano troppo tempo per la realizzazione, procedendo quindi secondo metodologia agile ad uno snellimento dei casi d’uso.

Per rendere più comprensibile al cliente e al personale tecnico il nostro prodotto riportiamo anche gli schemi UML e gli schemi delle classi per una migliore comprensione del nostro lavoro.

UML del Preprocessing:



Questo schema è stato utilizzato dal team del backend per lo sviluppo delle funzionalità di analisi dei file di log, il cui parsing avviene attraverso l’utilizzo di una libreria open-source: Grok . Riportiamo i blocchi logici del backend per illustrare come abbiamo interpretato la nostra soluzione, ad alto livello:



Team

T4

La scelta del gruppo su Trello è avvenuta cercando di unire persone con qualità e abilità differenti che potessero ricoprire tutti i ruoli all'interno del team, considerando che i partecipanti non si conoscevano prima abbiamo subito un periodo di aggiustamento riguardante le capacità e competenze tecniche richieste per lo svolgimento dell'incarico.

La strutturazione del gruppo è ritenuta soddisfacente in quanto ha promosso il lavoro in sottogruppi dediti alla risoluzione di un problema, senza alcun dubbio bisogna migliorare la interoperatività dei sottogruppi che spesso hanno subito ritardi nel momento in cui dovevano comunicare.

Il lavoro del team ha cercato di svilupparsi il più possibile rispettando le richieste e i dettami della programmazione agile ma non sempre siamo riusciti a rispettarli in quanto alle volte la natura stessa del compito da svolgere non richiedeva un'assiduità di rispetto delle regole della programmazione agile.

Il gruppo ha sempre cercato di mantenere un buon livello di comunicazione e rispettare le consegne e le loro scadenze che venivano considerate come punto focale del lavoro di ogni sprint, la definition of done è sempre stata rispettata per quanto riguarda la consegna del materiale ed è sempre stata tenuta in considerazione durante i momenti di sviluppo del progetto.

Sicuramente l'introduzione tardiva della fase di testing ci ha impedito di avere un ciclo di sviluppo agile fin dall'inizio ma ne abbiamo sopperito con l'introduzione di test appena in grado, sicuramente per un futuro ci sarebbe piaciuto introdurre una fase di testing dai primi momenti del progetto e riuscire a creare un sistema di testing automatizzato ma le nostre competenze e le restrizioni temporali non ci hanno permesso di fare diversamente.

In tutta la durata del progetto abbiamo saputo correggere la traiettoria del progetto in base alle restrizioni temporali eliminando tutte quelle funzionalità non necessarie e che avrebbero altrimenti rallentato ulteriormente il lavoro, quindi si può dire che per noi la variabile tempo ha sempre avuto un ruolo importante e decisivo.

I membri con le loro funzioni

Massimiliano Villa matr:136435

Ruolo: full-stack developer

Jacopo Vecchi matr: 119362

Ruolo: Project owner, developer tester, developer backend

Lorenzo Lucchina matr 135609

Ruolo: Scrum master, sysadmin, backend Developer

Simone Baroni matr: 122586

Ruolo: Developer tester

Filippo Reggiani matr: 148084

Ruolo: Developer backend

Janath Uthayakumar matr: 145610

Ruolo: Developer backend, Database architect, developer frontend

Al fine di completare il nostro progetto i ruoli nel team non sono stati considerati lapidari e immutabili ma si è preferito mettere l'obiettivo al primo posto il che ci ha portato ad una configurazione del team liquida, in cui i membri non si fermavano al loro solo scopo ma si preferiva cercare di completare gli incarichi aiutando chi ne aveva bisogno indipendentemente dal ruolo.

L'organizzazione strutturale del nostro gruppo ha permesso anche di sopperire ai problemi di assenza per qualche giorno di alcuni membri a causa di ragioni famigliari o di salute, riconfigurando velocemente il team e proseguendo con lo sviluppo del nostro prodotto.

Descrizione degli sprint

Il progetto è durato 4 sprint (s0, s1, s2, s3) iniziando il 04 aprile 2022 e concludendosi il 28 maggio 2022.

La progettazione degli sprint ha ricevuto particolare attenzione dal gruppo in quanto fin dallo sprint0 abbiamo dedicato almeno una seduta a sprint di diverse ore in presenza per il confronto e la definizione dei punti fondamentali su cui lavorare.

Lavorando in questo abbiamo notato come una buona progettazione e definizione dei punti focali di ogni sprint ci permettesse di lavorare in modo più fluido e preciso.

Sprint0

In questo sprint abbiamo lavorato all'analisi e identificazione del problema da risolvere andando a dare molta importanza alle problematiche di backend e alla soluzione tecnica.

Sprint1

Nello sprint1 abbiamo cercato di presentare una soluzione di backend funzionante, concentrandoci molto sulla sua strutturazione in quanto fin dal principio abbiamo ritenuto le funzionalità di backend lo scheletro portante del nostro progetto, mentre abbiamo iniziato a lavorare ad una prima bozza del frontend.

Sprint2

Nello sprint2 abbiamo dato importanza all'implementazione tra backend e frontend e abbiamo dato importanza alla presentazione di un prodotto funzionante secondo le richieste basilari del nostro cliente sia a livello backend che a livello del frontend.

Sprint3

Durante l'ultimo sprint abbiamo dato priorità al deploy online della nostra soluzione vagliando varie soluzioni, e abbiamo cercato di finire di implementare tutte le funzionalità opzionali.

Daily Scrum

I daily scrum si sono dimostrati fondamentali per aggiornare i membri e tenerci aggiornati sulle fasi di sviluppo del progetto, all'inizio hanno scandito con una non trascurabile regolarità il nostro lavoro

avvenendo giornalmente fino allo sprint 2, successivamente abbiamo deciso di abbandonare la pratica giornaliera per aggiornarci ogni 2 giorni, così facendo abbiamo avuto daily scrum più intensi e precisi nella comunicazione dei problemi.

Considerando l'andamento generale durante il progetto i daily scrum ci hanno aiutato nella fase iniziale a lavorare con costanza e nella fase finale nel aggiornarci tra i diversi membri per la risoluzione di problemi dandoci lo spazio di risolvere i problemi e comunicare in modo efficace.

Risultato dello Scrumble iniziale

Il risultato dello Scrumble iniziale ha evidenziato la necessità del gruppo di avere una prima fase di documentazione al fine di comprendere meglio gli aspetti tecnici della soluzione.

Molto è stato indagato non solo a livello di utilizzo dei diversi framework ma anche a livello tecnico rispetto ai sistemi web server apache e Docker, questa attività è stata ritenuta necessaria da tutti i componenti del gruppo dopo i primi incontri e a seguito dell'inizio della fase di progettazione.

Partendo da una fase iniziale di documentazione siamo però riusciti a portare tutti i partecipanti sullo stesso livello, almeno nelle prime fasi del progetto, sia con gli ambienti di sviluppo che con la comprensione ed analisi della soluzione proposta, proseguendo in modo più snello e deciso una volta che tutti avevano compreso la roadmap del progetto.

A seguito si riporta il risultato dello Scrumble iniziale:

Nome criticità	Impatto	Difficoltà	Rischio
Come utente voglio vedere a colpo d'occhio le sorgenti malevole	5	3	15
Come utente voglio vedere subito che cosa sta succedendo nel sito	5	3	15
Come utente voglio vedere se ci sono stati attacchi o problemi di sovraccarico	4	4	16
Come utente voglio vedere in un'unica schermata tutto il log corrente, per una visione generale	5	5	25
Come utente voglio definire una parola chiave per un filtro di ricerca testuale.	4	5	20
Come utente voglio isolare il traffico usando un filtro per una distinzione degli accessi.	2	5	10
Come utente voglio isolare il traffico usando un filtro per una distinzione degli accessi.	5	5	25
Come utente voglio definire un arco di tempo per vederne il traffico.	3	5	15
Come utente voglio vedere il traffico sulle varie sezioni del sito specificandole .	1	4	4
Come utente voglio vedere la sorgente di provenienza del traffico per poterlo analizzare .	5	4	20
Come utente voglio definire un'area sulla mappa per vederne il traffico .	5	5	25
Come utente voglio bloccare il traffico da un indirizzo malevolo .	2	5	10
Come utente voglio analizzare i dati per prendere decisioni commerciali .	5	5	25
Come utente voglio analizzare i dati per prendere decisioni amministrative .	1	5	5

Come utente vorrei poter analizzare quotidianamente le statistiche per un'analisi temporale dettagliata.	4	5	20
Come utente voglio usare la web app anche se cambio provider come utente voglio navigare nei log in avanti e indietro cronologicamente.	5	3	15
Come utente voglio inserire nel log dei marcatori per salvarli.	5	5	25

Retrospettiva finale

Con la conclusione dell'ultimo sprint (s3) ci siamo trovati davanti alla decisione di consegnare un prodotto che nel corso del tempo era mutato, a causa principalmente delle richieste del cliente e della variabile tempo, nonostante ciò, il gruppo è stato soddisfatto del lavoro fatto.

Cosa è andato bene:

La comunicazione nell'ultimo sprint è stata molto più fluida precisa ed esaustiva rispetto agli sprint precedenti, il focus dello sprint e la definition of done sono state considerate le bussole con cui abbiamo lavorato per il completamento del prodotto, il che ci ha resi efficienti.

Cosa è andato male:

Sicuramente il prodotto non ha raggiunto un livello di raffinatezza che ci eravamo imposti all'inizio del progetto ma le tempistiche ci hanno portato a fare modifiche e tagli alle volte anche sostanziali.

Cosa possiamo migliorare:

Allo sprint conclusivo abbiamo notato che possiamo migliorare tramite strumenti di CI/CD l'aggiornamento delle versioni del nostro prodotto per non incappare in inutili errori di compatibilità di versioni del prodotto.

Riportiamo ora la lavagna che abbiamo usato per l'ultima retrospettiva.

😊 GOOD	😞 BAD	🤔 IMPROVABLE ?
<p>DEDICAZIONE PROGETTO</p> <ul style="list-style-type: none"> • OTTIMA COMUNICAZIONE • B-E = F-E • OTTIMA COOPERAZIONE • APPRENDIMENTO NUOVE SKILLS 	<p>PESSIMA COMUNICAZIONE DEI SINGOLI</p> <ul style="list-style-type: none"> • MANCATI DAILY SCRUM X2 • MANCATO UTILIZZO DOCKER • GESTIONE DEL TEMPO • METODO SCRUM • SEPARAZIONE RUOLI • MANCANZA DI FUNZIONALITÀ SECONDARIE • conclusione precoce del project 	<p>BISOGNA MIGLIORARE LA COMUNICAZIONE DEI SINGOLI</p> <ul style="list-style-type: none"> • PROGETTUALITÀ DA MIGLIORARE • Lavorare fino alla fine

Storia degli sprint

S1

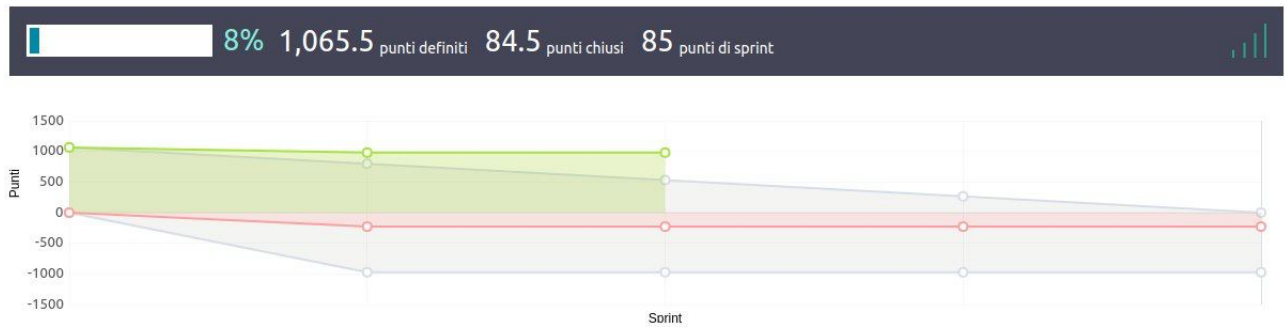
Date: 30/04

- Goal:

Presentazione di uno scheletro di backend funzionante insieme ad uno scheletro di frontend.

Backlog:

Scrum



Backlog

17 storie utente

+ STORIA UTENTE

+

Filtri

soggetto o riferimento

Etichette

Previsione di velocità

STORIA UTENTE

STATO

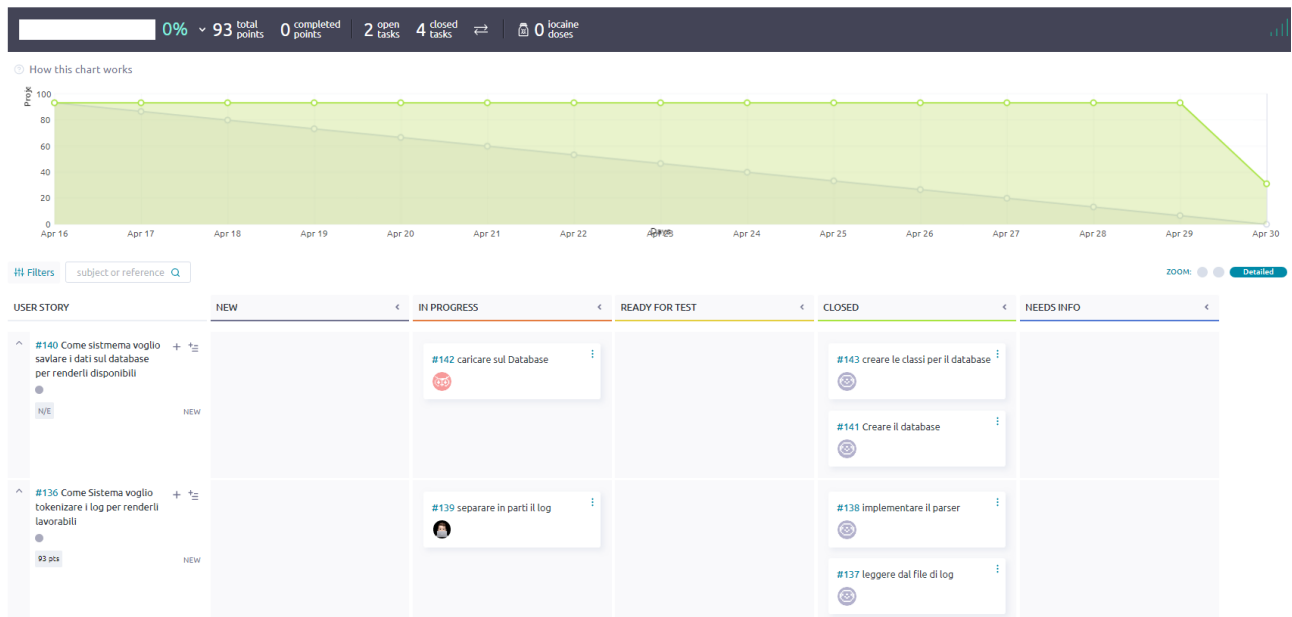
PUNTI

<input type="checkbox"/> #105	Come utente voglio inserire nel log dei marcatori per salvarli da parte	 	Nuovo	76	
<input type="checkbox"/> #57	Come utente voglio vedere in modo semplice e comprensibile il log corrente per capirne i dati		Nuovo	88	
<input type="checkbox"/> #111	avviso email tramite sistema smtp		Nuovo	?	
<input type="checkbox"/> #106	Come utente voglio ricevere delle mail che mi notificano in caso di attacco per essere informato	 	Nuovo	40	

STORIA UTENTE	STATO	PUNTI
#104 Come utente voglio navigare nel log, cronologicamente, in avanti e indietro per scorrerli	Nuovo	49
#102 Come utente vorrei analizzare un repilogo quotidiano per rendere facile l'analisi	Nuovo	48
#101 Come utente voglio analizzare dati per prendere decisioni amministrative	Nuovo	15
#100 Come utente voglio analizzare dati per prendere decisioni commerciali	Nuovo	36
#97 Come utente voglio vedere la sorgente di provenienza del traffico per poterlo analizzare.	Nuovo	33
#99 Come utente voglio bloccare il traffico da un indirizzo malevole per una maggior sicurezza	Nuovo	61
#96 Come utente voglio vedere il traffico sulle varie sezioni del sito per vederne l'afflusso	Nuovo	26
#89 Come utente voglio definire un arco di tempo: settimanale, giornaliero, orario per vederne il traffico	Nuovo	55
#80 Come utente voglio isolare il traffico usando un filtro per una distinzione degli accessi	Nuovo	79
#70 Come utente voglio definire un filtro complesso di ricerca usando più filtri singoli	Nuovo	66
#98 Come utente voglio definire un'area sulla mappa per vederne il traffico.	Nuovo	51
#63 Come utente voglio definire una parola chiave per creare un filtro di ricerca testuale	Nuovo	96
#48 Come utente voglio vedere se ci sono stati attacchi o problemi di sovraccarico senza dover esaminare il log	Nuovo	69

Burndown chart :

Sprint 1 16 Apr 2022 to 30 Apr 2022



Definition of Done:

La nostra definition of done in questo sprint era definita dalla capacità di effettuare una prima analisi dei file di log.

S2

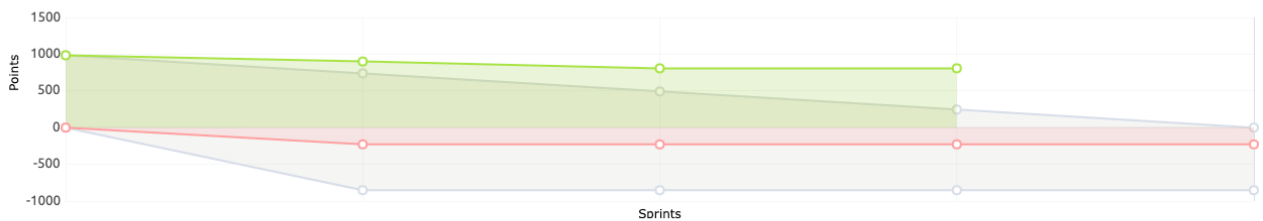
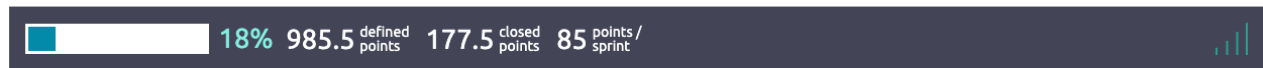
Date: 14/05

- Goal:

Avere un software funzionante che raccolga i dati dentro ad un DB e li renda disponibili ad un client Frontend che a sua volta li mostra all'utente.

Backlog:

Scrum



Backlog 13 user stories

Filters subject or reference Tags

+ USER STORY

Velocity forecasting

USER STORY	STATUS	POINTS
#105 Come utente voglio inserire nel log dei marcatori per salvarli da parte	In progress	76
#57 Come utente voglio vedere in modo semplice e comprensibile il log corrente per capirne i dati	Ready for test	88
#111 avviso email tramite sistema smtp	New	?
#106 Come utente voglio ricevere delle mail che mi notificano in caso di attacco per essere informato	New	40
#104 Come utente voglio navigare nel log, cronologicamente, in avanti e indietro per scorrerli	New	49
#102 Come utente vorrei analizzare un repilogo quotidiano per rendere facile l'analisi	New	48

Backlog 13 user stories

+ USER STORY

Filters

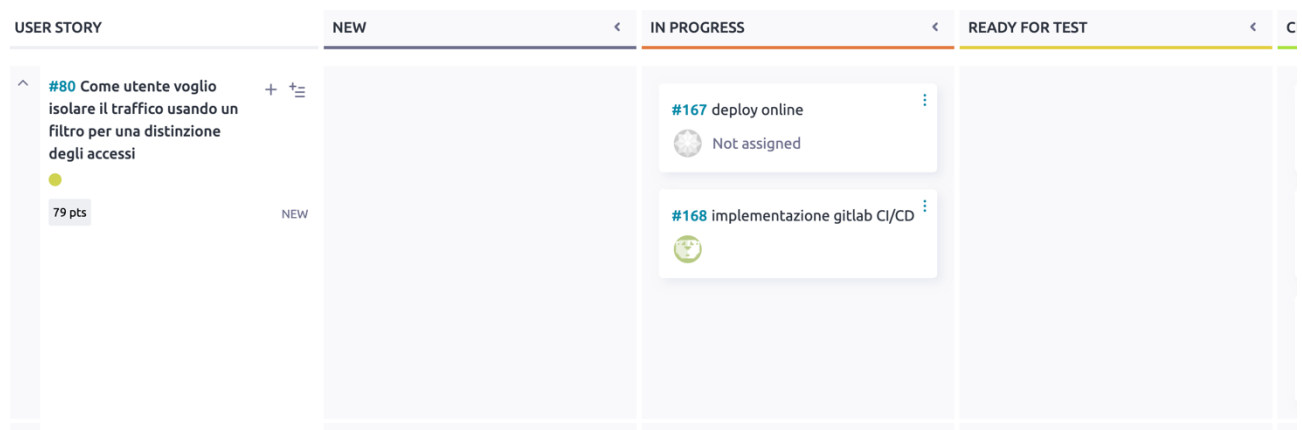
subject or reference

Tags

Velocity forecasting

USER STORY	STATUS	POINTS
#103 Come utente voglio inserire nel log dei marker per salvarli da parte	In progress	70
#57 Come utente voglio vedere in modo semplice e comprensibile il log corrente per capirne i dati	Ready for test	88
#111 avviso email tramite sistema smtp	New	?
#106 Come utente voglio ricevere delle mail che mi notificano in caso di attacco per essere informato	New	40
#104 Come utente voglio navigare nel log, cronologicamente, in avanti e indietro per scorrerli	New	49
#102 Come utente vorrei analizzare un repilogo quotidiano per rendere facile l'analisi	New	48
#97 Come utente voglio vedere la sorgente di provenienza del traffico per poterlo analizzare.	New	33
#99 Come utente voglio bloccare il traffico da un indirizzo malevole per una maggior sicurezza	New	61
#96 Come utente voglio vedere il traffico sulle varie sezioni del sito pre vederne l'afflusso	New	26
#89 Come utente voglio definire un arco di tempo: settimanale, giornaliero, orario per vederne il traffico	In progress	55
#70 Come utente voglio definire un filtro complesso di ricerca usando più filtri singoli	In progress	66
#98 Come utente voglio definire un'area sulla mappa per vederne il traffico.	New	51
#63 Come utente voglio definire una parola chiave per creare un filtro di ricerca testuale	New	96

Burndown chart:



Definition of Done:

Dopo che una funzionalità è testata oltre al 40% ed esegue il suo compito essa viene ritenuta completata.

S3

Date: 28/05

- Goal:

Deploy online della web-app co tutte le sue funzionalità, testing finale del prodotto.

Backlog:

▼ **Sprint 3**

14 May 2022-28 May 2022

0 closed

339 total

<div></div>	
#57 Come utente voglio vedere in modo semplice e comprensibile il log corrente per capirne i dati ●	88
#89 Come utente voglio definire un arco di tempo: settimanale, giornaliero, orario per vederne il traffico ●	55
#97 Come utente voglio vedere il luogo di provenienza del traffico per poterlo analizzare. ●	33
#96 Come utente voglio vedere il traffico sulle varie sezioni del sito pre vederne l'aflusso ●	26
#104 Come utente voglio navigare nel log, cronologicamente, in avanti e indietro per scorrerli ●	49
#102 Come utente vorrei analizzare un repilogo quotidiano per rendere facile l'analisi ●	48
#106 Come utente voglio ricevere delle mail che mi notificano in caso di attacco per essere informato ● ●	40

Definition of Done:

Le funzionalità vengono considerate terminate dopo una positiva implementazione ed esecuzione all'interno del server usato per il deploy.

Riportiamo in seguito il chart finale dello sprint 3 con le attività concluse.

SER STORY	NEW	IN PROGRESS	READY FOR T...	CLOSED
<div>#57 Come utente voglio vedere in modo semplice e comprensibile il log corrente per capirne i dati</div> <div>88 pts</div> <div>READY FOR TEST</div>				<div>#173</div> <div>#124</div>
<div>#89 Come utente voglio definire un arco di tempo: settimanale, giornaliero, orario per vederne il traffico</div> <div>55 pts</div> <div>IN PROGRESS</div>				<div>#174</div> <div>#171</div> <div>#172</div> <div>#178</div>

SER STORY	NEW	IN PROGRESS	READY FOR T...	CLOSED
<div>#97 Come utente voglio vedere il luogo di provenienza del traffico per poterlo analizzare.</div> <div>33 pts</div> <div>NEW</div>				<div>#179</div> <div>#180</div>
<div>#96 Come utente voglio vedere il traffico sulle varie sezioni del sito pre vederne l'aflusso</div> <div>26 pts</div> <div>NEW</div>				<div>#177</div> <div>#176</div>
<div>#104 Come utente voglio navigare nel log, cronologicamente, in avanti e indietro per scorrerli</div>				<div>#117</div>

SER STORY	NEW	IN PROGRESS	READY FOR T...	CLOSED
<div>#102 Come utente vorrei analizzare un repilogo quotidiano per rendere facile l'analisi</div> <div>48 pts</div> <div>NEW</div>				<div>#123</div> <div>#187</div> <div>#183</div> <div>#184</div> <div>#188</div> <div>#185</div>

Link ai prodotti utilizzati

Taiga: <https://tree.taiga.io/project/lucchi-web-app-apache-log-analyzer/timeline>

Gitlab: <https://dw.gnet.it/2022T4/2022t4>

Mattermost: <http://212.71.250.11:8065>

Per la realizzazione del progetto abbiamo scelto di usare il più possibile gli strumenti floss, usando come SO per lo sviluppo ubuntu e debian, come editor vscode e IDE intellij, abbiamo usato per il deploy e come librerie esterne solo prodotti opensource.

- grok: <https://github.com/oracle/opengrok/wiki/Features>
- spring framework: <https://github.com/spring-projects>
- localtunnel: <https://github.com/localtunnel/localtunnel>

La demo del prodotto si trova al seguente link:

<https://flat-streets-marry-212-71-250-11.locat.it>

Report ore di lavoro complessive

Riportiamo in seguito le ore complessive e il link al foglio di lavoro utilizzato per la registrazione delle ore totali di lavoro con anche le attività svolte:

https://docs.google.com/spreadsheets/d/1Os-4cWbGvn2PUOWA2VI5hk_w1UynI31BhIVyn8JeWgg/edit#gid=0

Massimiliano Villa matr:136435

81 ore complessive sul progetto

70 commit

Jacopo Vecchi matr: 119362

127 ore complessive sul progetto

51 commit

Lorenzo Lucchina matr 135609

110 ore complessive sul progetto

90 commit

Simone Baroni matr: 122586

60 ore complessive sul progetto

32 commit

Filippo Reggiani matr: 148084

61,5 ore complessive sul progetto

41 commit

Janath Uthayakumar matr: 145610

90,5 ore complessive sul progetto

69 commit

TOTALE DELLE ORE COMPLESSIVO: 530 ore totali

Alcune ore di lavoro all'interno del progetto non sono state segnate in quanto non erano significative o ricadevano in una fase iniziale del progetto in cui non abbiamo tenuto traccia del lavoro svolto.

Il totale dei commit complessivi nel progetto avvenuti da tutti gli account utilizzati dai partecipanti è di 387 commit.




Riportiamo ora l'interezza dei commit eseguiti dal più recente al primo:

```
Merge remote-tracking branch 'origin/main'
main
" commenti e aggiunte LogRepo.java"
Delete old page
Update dashboard
Add 'dettagli logs' link
Add API for geolocation
" log err commentato"
" :sparkle: database errori caricato"
" :sparkle: nuovo LogError.java e AllLogListsErr.java"
Merge remote-tracking branch 'origin/main'
Implementazione inserimento sul database dei log di errore
" :sparkle: nuovo LogError.java"
Merge remote-tracking branch 'origin/main'
Implementazione inserimento sul database dei log di errore
" :sparkle: log minimo implementato cosi carichiamo meno roba nel frontend e approfondiamo
con il see more di max"
" :bug: bugfix "
" :bug: bugfix "
" :bug: bugfix "
"commit commenti javadoc"
"commit /alldb/all/{search_by_id}"
" correzione index sballati"
"implementata /api/alldb/all"
"implementata /api/smart/alldb"
commit prova
development
"commit logrepo search by id"
"commit giusto"
Inseriti nuovi test.
Delete Test.java
Inseriti dei nuovi test.
Delete Test.java
Fixed typo error
Add SonarQube project's properties
Set filter and layout
Modify styles
Removed search component
Correzione del diagramma UML del pre processing.
Correzione del diagramma UML del pre processing.
Delete Diagramma_UML_PreProcessing.jpg
Delete Diagramma_UML_PreProcessing
" smart view"
"dannati mac"
"ho probabilmente rotto i TEST my bad , credo di aver mergeato male scusate"
Merge remote-tracking branch 'origin/development' into rework
Merge remote-tracking branch 'origin/main' into rework
Merge branch 'main' into rework
Merge remote-tracking branch 'origin/rework' into rework
"databse popolato"
Merge remote-tracking branch 'origin/development' into development
```


Created search filters
Merge remote-tracking branch 'origin/rework' into rework
update gitignore
" :map: we got the geo location BOYZ PART 3 "
" :map: we got the geo location BOYZ PART 3 "
update gitignore
" :map: we got the geo location BOYZ PART 3 "
" :map: we got the geo location BOYZ PART 2 "
Update fetching method
Upload New File
Add new directory
Replace casoduso.pdf
Merge branch 'docker_branch' into development
commentis2
" :ambulance: funzioni AllLogsList.Crea_data_bella e AllLogsList.Tipo_di_richiesta
funzionano e salvano sul database "
" :bug: help me filippo "
docker images 18/05
" :sparkle: Help me please Lore , non salva il Json nel database "
riolti conflitti docker-compose
vecchie modifiche docker-compose
" :sparkle: git ignore "
cambiamento DB
Upload New File
Upload New File
docker-compose semi funzionante
mysql on container
Aggiunte ore di lavoro recenti e pregresse di Villa
Update file
new docker branch
Merge branch 'development' of https://dw.gnet.it/2022T4/2022t4 into development
aggiunta cartelle con file di log e log errore
aggiornamento ore lavoro lorenzo
merge branch 'development' of https://dw.gnet.it/2022T4/2022t4 into development
Create LineChart component
Create BarChart component
Update login page
Create page for charts visualisation
":sparkles: aggiunta fusione count bytes"
Add dependencies
update jar
Add chart's dependencies
Update nuxt.config.js: add @nuxt/proxy
Add environment variables
Merge branch 'development' of https://dw.gnet.it/2022T4/2022t4 into development
Add charts.vue page for stats
Add stats img
Modified logsvew
Upload New File
docker-compose funzionante, update README.md
Merge branch 'development' of dw.gnet.it:2022T4/2022t4 into development
backend.dockerfile with mvnw

```
test
Merge branch 'dockerizing' into development
modifiche LogReader
backend.dockerfile e docker-compose.yml
Merge remote-tracking branch 'origin/development' into development
":sparkles: aggiunto LogErrRepo.java per gestire input nel database"
backend.dockerfile e docker-compose.yml
f
frontend.dockerfile working
t
Merge remote-tracking branch 'origin/development' into development
":sparkles: aggiunto controller per ricerca in base IP e ricerca in base IP MESE GIORNO
ANNO"
Merge remote-tracking branch 'origin/dockerizing' into dockerizing
backend.dockerfile e docker-compose.yml
Update logview
Update README.md
Update logview
Add @nuxt/http module
Add @nuxt/http module
Update navbar
Deleted json for test
Merge branch 'development' of https://dw.gnet.it/2022T4/2022t4 into development
Deleted Logo
Aggiornamento del file dei test: aggiunto nuovo test.
Cancellazione del file per uno più aggiornato.
dockerizzazione completata
dockerfile
Upload New File
":sparkle: aggiunte le chiamate nel controller e service , time to sleep boys "
Upload New File
Upload New File
Add new directory
sistemazione main
":test_tube: logrepo pronta ad essere testata "
File contenente tutti i test del gruppo testing formato da Jacopo Vecchi e Simone Baroni.
added jar maven plugins
implementing testSpammer LogController
Merge remote-tracking branch 'origin/development' into development
Conversione in Json dello Spammer
Delete wrong folder
Create retrieve template for json
Add dashboard link
Add @nuxt/content module
":sparkles: aggiunto il primo Test basato su quale ip ha fatto tante richieste"
Create content/ folder
Add logs-view page
Create a simple navbar
Add @nuxt/content for fetching data
Modified nuxt.config.js and package.json
postprocessing json field referer BUG-Fixed
Merge branch 'postprocessing' into development
```

postprocessing json field referer
rimozione elementi dalla cartella Frontend
":sparkles: aggiunto il primo Test basato su quale ip ha fatto tante richieste"
commit prova
Merge remote-tracking branch 'origin/development' into development
commit prova
":wrench: messo apposto il main commentando la parte di parsing,
commit prova
Merge branch 'Prova_codice' into development
modifica campo referer
":white_check_mark: Quarries funzionanti con JPA la soluzione era mettere una doppia lista
di stringhe come tipo di ritorno "
"ripulito main , spring worka"
Merge remote-tracking branch 'origin/development' into development
"database jpa communication "
Aggiornato Heimdall con log_errore in classi
Merge remote-tracking branch 'origin/development' into development
testing matchmakerlog risolto e funzionante
script di prova sql
fixed bug for Log Error json writing
testing nullo
aggiunta todo logparser e commenti main
Aggiungo modifiche parser errori
"aggiunta entita LogError.java , pronto per testing"
modifiche al JsonWriter.java per renderlo più adatto al framework di frontend
testing writer non funzionante
testing writer
Bug fixes for input parameters
ore di lavoro aggiornate
testing dentro spring
testing backedn con test che non funzionano riguardo ai valori nulli
testing delle funzioni delle backend
Delete Domande_da_fare.txt
inizio implementazioni del testing
commit readme test
Diagramma UML pre processing in formato draw.io per eventuali modifiche.
Diagramma UML del pre processing.
Cancellazione Diagrammi_UML_Backand.jpg in quanto inutile.
Cancellazione Diagramma_UML_Backand in quanto sbagliato.
Merge branch 'development'
Aggiunta cartella file json (ignored)
Diagramma UML del Backand in formato draw.io.
Diagramma UML del Backand in formato immagine.
aggiunto dump del databse
Upload New File
Add new directory
Delete Ore_di_lavoro.xlsx
commentato il salvataggio delle entità nel database, definitivamente popolato
lettura plenaria, creazione oggetti log, popolamento db
lettura plenaria, creazione oggetti log, popolamento db
"Versione pulita databse funzionante"
LogEntityJson.java e reading LogEntityJson e creazione ArrayList nobug

LogEntityJson.java e reading LogEntityJson e creazione ArrayList no bug
LogEntityJson.java e reading LogEntityJson e creazione ArrayList
"aggiornamento HeimdallApplication"
LogEntityJson.java e reading LogEntityJson
JsonReader.java
rimozione dipendenze per sicurezza
Merge remote-tracking branch 'origin/development' into development
"db insert funziona"
"db insert funziona"
Merge remote-tracking branch 'origin/development' into development
heimdall funzionante
"commit 
"commit 
"commit 
"commit prova"
Merge remote-tracking branch 'origin/development' into development
"commit finale super finale"
"commit finale"
Upload frontend.mp4
Deleted old README.md
Modified README.md
Add login component
Merge branch 'development' of https://dw.gnet.it/2022T4/2022t4 into development
Login page created
Modified typo error
Modified main.css
"spring che worka"
spring versione pulita
Merge branch 'development' of https://dw.gnet.it/2022T4/2022t4 into development
Set up BootstrapVue Nuxt.js module
refactoring processing
Merge branch 'development' of https://dw.gnet.it/2022T4/2022t4 into development
Set up main font from Google Font
Add :root pseudo-class
" :loop: Rework log entity 2"
" :loop: Rework log entity"
Delete error.vue
Merge branch 'development' of https://dw.gnet.it/2022T4/2022t4 into development
Add error 404 page
Added layouts folder
Merge remote-tracking branch 'origin/development' into development
first classes for json deserialization
set up the main css file
Add css folder
commento Retrospettiva
Merge branch 'main' of https://dw.gnet.it/2022T4/2022t4
commento sprint 1
Upload New File
Merge branch 'development' of https://dw.gnet.it/2022T4/2022t4 into development
change dependencies
r
Readme leggermente sistemato

Screenshot di Taiga sul backlog dello sprint 1 (parte 3).
Screenshot di Taiga sul backlog dello sprint 1 (parte 2).
Screenshot di Taiga sul backlog dello sprint 1 (parte 1).
improvement json writer
json writing
Deleted Tutorial.vue
Initialising a NUXTJS project
Deleted wrong version
Added NUXTJS project
modificato README.md
modificato README.md
secure configuration
Commento provvisorio sprint 1
inserimento ore di lavoro
Burndown chart sprint 1
v2.0
final push for sprint 1
v0.2
aggiunta documentazione Service
readme e service doc
Upload README.md for frontend.
aggiunta documentazione
Upload New File
Upload New File
inserimento Frontend in code
Add new directory for Frontend development
aggiunti commenti
inclusione dependency geoIp
aggiunta richieste Service
piccola sistemazione LogService.java e LogController.java
sistemazione LogController e LogServicer.java
write all files
sistemazione metodo readAllLogFile
Merge remote-tracking branch 'origin/development' into development
rimosso l'array di file nel writer
"aggiunta funzione readAllLogFile() , in attesa del writer per il testing "
commenti
aggiunto commenti
"piccole rifiniture"
spostata la logica in service
Creazione packages repo e service
refactoring
Merge branch 'origin/main'
Merge branch 'lollo_branch'
rimozione grokdemo e LoggerPDS
modifica albero project
merge di Filippo_branch su lollo_branch
creazione controller e logica RPW
Aggiunto modifiche progetto Heimdall
"piccole rifiniture"
commitato per sbaglio in test correggo la cartella
creati scheletri delle classi MYsql

Commento allo sprint 0
config db
aggiunto heimdall nuovo
riorganizzazione
prova
nuovo progetto heimdall
versione Heimadal con classi
Aggiungo cartella Heimdall
sistemazione spring project
creazione progetto spring
aggiornato grokdemo
scrittura del grok cion reader che utilizza la classe parser con grok e classe writer
sistemato grok funzionante e spostato il progetto nella sua directory
aggiungo demo grok
working maven project w/ grok
aggiunta commento
sistemato metodo cercatoreGrok
inclusione di grok : RIUSCITAAAAAA
sistema errore sintattico
sistemato le stringhe contenti i pattern
classe parser di jACOPO senza sistema di scrittura
aggiunto pdf del metodo parsing , ancora da implementare
sviluppo del secondo metodo costruttore conoscendo il pattern
aggiunto metodo costruttore che prende un pattern , WorkInProgress
aggiunti i commenti
credo che le regex siano la strada giusta ma sono troppo inesperto per farle
creazione branch + cartella DBMS
Ho modificato la classe DBConnection
Blocchi backend in versione PDF.
Blocchi backend in formato draw.io.
aggiornato file
prova commit filippo
Delete Prova.odt
Lui è tornato.
File che serve per dei test.
Il file prova non serve più.
Delete Prova.odt
Questo è un file di prova che non serve a nulla.
versione finale della classe Reader pronta per implementazione con Grok
modifica del main
Update .gitlab-ci.yml file
Merge branch 'main' of <https://dw.gnet.it/2022T4/2022t4> into main
prima struttura backend con abbozzo di Logger e sistema per lettura dei file di log
snapshots e log
v0.1
correzione documentazione doc_rischi
Documento dei rischi e delle priorità con la documentazione relativa alla partita di
scrubbe più il commento finale.
sottodirectory di doc
Caricato il file che contiene la tabella dei rischi e delle priorità.
Final mockup
aggiunto file casiduso

```
Update Domande_da_fare.txt
finito readme per sprint 0
finito readme per sprint 0
commentiMissiroi12apr
File che piega il file "Mockup progetto.pdf"
Un ipotetico mockup della pagina utente (non ancora specificato se utente "normale" o
"amministratore").
Ho aggiunto la mia matricola
presentazione progetto in breve
Upload New File
Upload New File
Update Domande_da_fare.txt
Delete Ore_di_lavoro.xlsx
Upload New File where we can put our hours of work
Update README.md
Add new file
Upload New File with the user stories
una directory per le cose vecchie che non ci servono più
Upload New File
Upload File with the question
Delete UserStories.txt
Upload architectural diagram in drawio format
Delete Diagramma_architetturale.drawio.xml
Upload architectural structure in pdf format
User Stories provvisorie da riguardare prima di aggiungerle su taiga
Upload architectural diagram
Delete Diagramma_architetturale.drawio
Upload file architectural diagram
Delete photo5942639149179844470.jpg
Upload New File
aggiunta matr jaco
rimozione cartella
inserimento matricola lorenzo
mockup pagina2b
mockup pagina2a
mockup pagina1b
mockup pagina1a
mockup pagina0
cartella dei mockup
Add new directory
Add new directory
Add new directory
Add new directory
Aggiornato README.md
Initial commit
```

Descrizione Artefatti

Processo

Definition of Done:

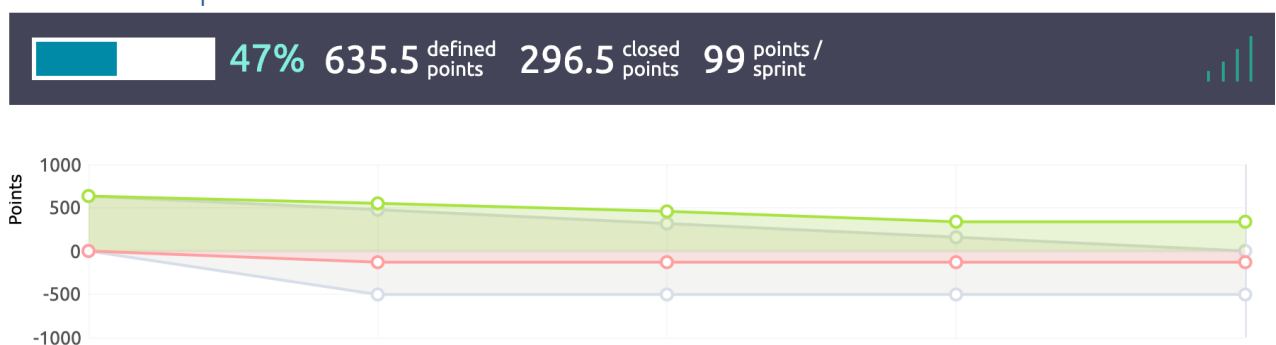
Durante il nostro processo produttivo la definition of Done è variata in base alle richieste del cliente e alle nostre competenze tecniche che a mano a mano andavano ad aumentare.

Generalmente al termine di ogni sprint il team si riuniva per almeno un paio di ore e si discuteva di cosa dovesse essere migliorato e sistemato a livello tecnico, nel codice, ma anche e soprattutto a livello produttivo.

Ciò ha permesso di adeguare ad ogni sprint il nostro prodotto con le richieste del cliente dando priorità agli aspetti ritenuti fondamentali.

I test manuali ci hanno iniziato ad accompagnare dal secondo sprint, e proprio grazie all'attività di testing abbiamo trovato e risolto una serie di bug che avrebbero severamente compromesso il progetto, il testing ha poi proseguito fino all'ultimo sprint cambiando focus se prima solo sulla parte di backend successivamente anche sul frontend, testando le interazioni con la UI per il cliente.

Burndown complessivo



Come si evince dalla analisi complessiva del lavoro il gruppo ha sempre cercato di modificare in modo agile obiettivi e il progetto per raggiungere il massimo livello di fedeltà alla richiesta del cliente.

Prodotto

Gli artefatti del prodotto sono due, che si trovano in esecuzione sul server remoto:

- Un eseguibile .jar funzionante su macchina Linux o su qualsiasi macchina con java aggiornato alla versione corretta.
- Database mysql.
- Uno script di node.js realizzato con framework vue.js che funziona su ogni tipo di macchina partendo dal presupposto che le versioni corrette di npm e node siano installate.
- Il prodotto si rende raggiungibile tramite browser.

Conclusioni

La realizzazione di questo progetto ci ha posto davanti alla grande sfida del lavoro di gruppo mai realizzato prima e sicuramente è stata una grande occasione di crescita per ogni membro del gruppo, sia a livello tecnico che a livello personale.

Sicuramente abbiamo imparato come la programmazione agile possa aiutare nella realizzazione di progetti di grandi dimensioni e come il codice ha la sua importanza ma la sua parte progettuale e l'organizzazione dei compiti ha una rilevanza da non sottovalutare, nella preparazione prima della fase di coding il progetto prende forma e un buon gruppo è soprattutto un gruppo organizzato a livello di strumenti e obiettivi comuni.

Dopo questa esperienza ci siamo accorti della rilevanza della fase progettuale e della necessità di micro e macro managing delle risorse, soprattutto temporali, che è il fattore decisivo sulla riuscita di un progetto più che il codice o la preparazione stessa dei membri a livello tecnico.

Consigli per gli esami futuri

Considerando il potenziale dell'esame i pochi crediti che rappresenta sono una inezia in confronto allo sforzo che gli studenti devono fare, nello specifico aumentare a 9 crediti potrebbe giustificare lo sforzo che molti fanno per completare l'esame soprattutto se fatto in modalità di presentazione del progetto.

A livello di materiale didattico ci ha messo fortemente in difficoltà l'assenza di una documentazione, spiegazioni o almeno un indice per la documentazione in cui riferire gli aspetti tecnici che richiederebbero l'assistenza o la guida di figure esperte per usare efficacemente certe nuove tecnologie.