



Piattaforma di gestione dei lavori di un Team Scrum

Un'applicazione serverless per l'organizzazione dei
compiti relativi a progetti in via di sviluppo
all'interno di un gruppo di lavoratori che utilizza il
framework agile Scrum

STUDENTESSA: ILENIA ROCCA

CORSO DI STUDI IN INGEGNERIA INFORMATICA

UNIVERSITÀ DEGLI STUDI DI ROMA "TOR VERGATA"

MATRICOLA: 0285151

Cos'è Scrum

È un framework agile che prevede l'organizzazione del lavoro in blocchi sequenziali e iterativi, chiamati “Sprint”.

A ciascuno Sprint viene assegnata una lista di funzionalità, chiamate anche “story”, che sono a loro volta suddivise in task dal team di sviluppo.



L'applicazione

Accesso alla piattaforma tramite login con email e password, previa registrazione.

Entità coinvolte:

- **Progetti** (id, titolo, scadenza, descrizione, commenti)
- **Story** (id, titolo, scadenza, descrizione, progetto di appartenenza, punteggio, stato, priorità)
- **Task** (id, titolo, story di appartenenza, stato)

Possibilità di inserimento, modifica ed eliminazione per ciascuna entità.

Ogni task appartiene ad una sola story, ogni story appartiene ad un solo progetto.

Realizzazione delle API: AWS Lambda e API Gateway

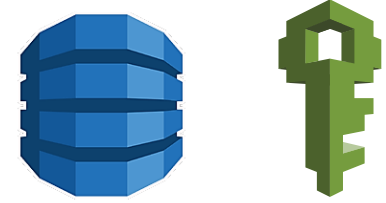


AWS Lambda consente la creazione di un'applicazione serverless tramite la scrittura di funzioni lambda nel linguaggio di programmazione Python.

Sono state realizzate 6 API con il servizio di Amazon API Gateway ed in totale vengono utilizzati 16 metodi REST (POST, PUT, PATCH, DELETE, GET).

Nella API relativa ai Task si usano sia il metodo patch che il metodo put al fine di effettuare due diversi tipi di modifiche sull'oggetto. Nel primo caso, viene semplicemente modificato il titolo del task a piacimento dell'utente. Nel secondo caso, viene aggiornato lo stato del task da “da fare” a “completato” in maniera automatica al click del bottone relativo al task.

Storage: DynamoDB e IAM



Database NoSQL di tipo chiave-valore.

Creazione tabelle *Task*, *Story*, *Project* ciascuna con *id* come chiave primaria.

Ruolo *LambdaBoard* creato su IAM con permessi per le operazioni di scan, put, get, update, delete, tramite l'ARN delle rispettive tabelle.

Cognito



Sign in with your email and password

Email

Password

[Forgot your password?](#)

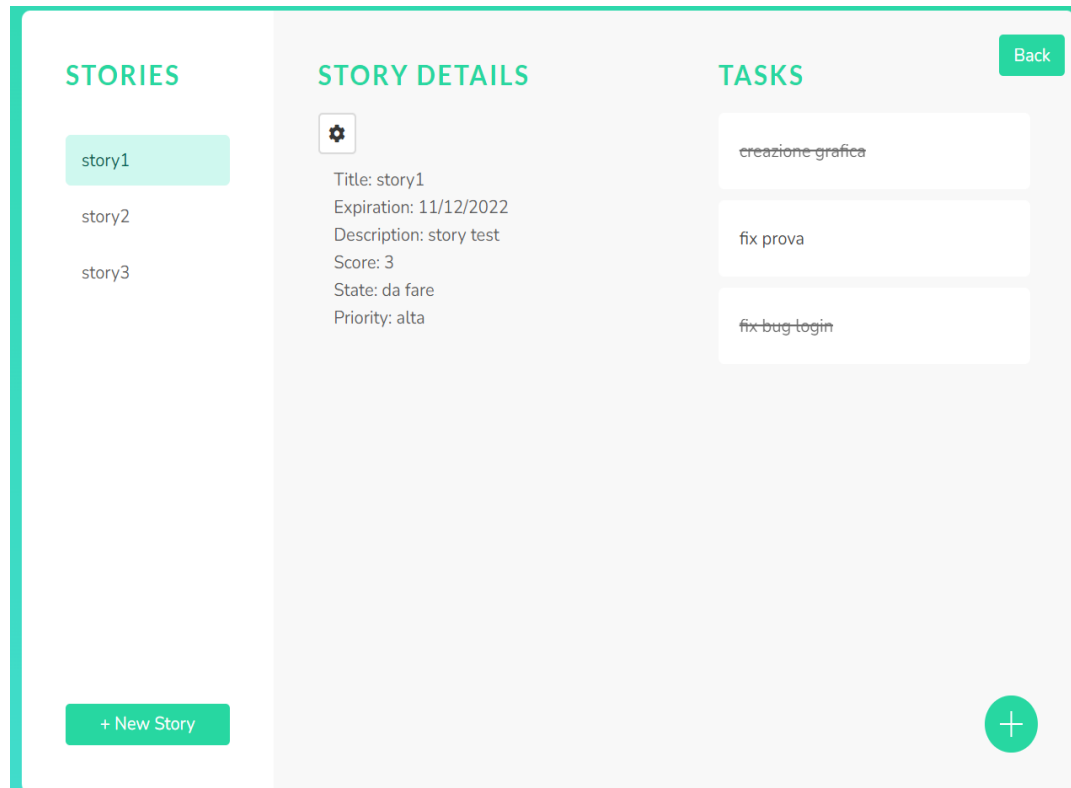
Sign in

Need an account? [Sign up](#)

Servizio di AWS che gestisce la registrazione degli utenti ed il mantenimento delle informazioni da questi inserite.

Permette di effettuare il login nella piattaforma tramite le credenziali precedentemente registrate.

Frontend in Angular



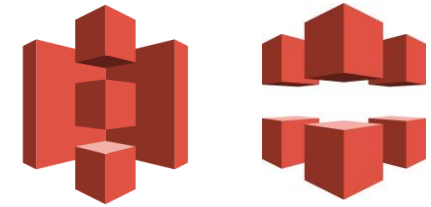
Una delle parti più “dispendiose” della realizzazione in termini di tempo, poiché prima esperienza con lo sviluppo lato frontend.

Utilizzo di Visual Studio.

Applicazione composta da vari component, circa uno per ogni pagina.

Passaggio dei dati alle api tramite strutture di tipo *json*.

S3 e CloudFront



Caricamento della build del progetto Angular all'interno del bucket S3 *serverlessboard* in modalità hosting.

Problema: il bucket offriva un indirizzo con l'uso del protocollo http e questo non rendeva possibile il reindirizzamento dall'host relativo al servizio di login Cognito. Era necessario un url con protocollo sicuro https.

Risoluzione: introduzione del servizio AWS CloudFront che offre una nuova distribuzione con https di supporto al proprio url.

Utilizzando questa distribuzione si viene correttamente reindirizzati alla dashboard dell'applicazione ed è possibile utilizzare la stessa.

Conclusioni

Una piattaforma di questo tipo potrebbe sicuramente essere ampliata e migliorata per avere un ruolo concreto all'interno di un team di sviluppo software, migliorando ed automatizzando il lavoro del gruppo.





Grazie per l'attenzione!