

Relazione Progetto

Fernando H. Gavezzotti

Indice

1 API NASA	2
2 Architettura	2
2.1 Database	2
2.2 Backend	3
2.3 Frontend	3
3 Query principali	3
4 Insight ottenuti	4
5 Riflessione Personale e Contestuale	4
6 Proposte di miglioramento	5

1 API NASA

Per la realizzazione della dashboard l'API NASA scelta è stata **Asteroids-NeWs**, che fornisce dati sugli asteroidi vicini alla Terra; in particolare, la scelta è ricaduta su questa API per la possibilità ampia di visualizzazione dei dati, che si presta bene alla realizzazione di una dashboard. In particolare i dati forniti da questa API che sono stati utilizzati sono:

- ID dell'asteroide
- Nome dell'asteroide
- Nasa JPL URL (link alla pagina dell'asteroide sul sito della NASA)
- Magnitudine assoluta dell'asteroide
- Diametro stimato dell'asteroide
- Se l'asteroide è potenzialmente pericoloso o meno
- Se l'asteroide è una sentinella o meno
- Data di avvicinamento alla terra
- Velocità dell'asteroide
- Distanza dell'asteroide dalla terra

2 Architettura

L'architettura del progetto è stata progettata per essere modulare, scalabile e sicura. Il progetto è stato suddiviso in tre componenti principali:

- Database
- Backend
- Frontend

2.1 Database

Per il database è stato scelto il sistema di gestione dei database relazionali PostgreSQL.



Figura 1: Diagramma del database

Per la realizzazione del database è stata utilizzata una tecnica di normalizzazione separando in tabelle distinte le informazioni sugli asteroidi e le informazioni sugli avvicinamenti alla terra. In questo modo è stato possibile evitare la ridondanza dei dati

e garantire l'integrità referenziale tra le tabelle. Inoltre è stato creato un index in SQL tra `idx_close_approach_date` e `close_approach_date` per migliorare le performance delle query che filtrano per data di avvicinamento.

2.2 Backend

Per il backend è stato utilizzato Node JS con il framework Express. Il backend si occupa di:

- Popolare il database con i dati forniti dall'API NASA, che avviene in due modi: al primo avvio del server vengono inserite le informazioni a partire da una data di inizio fino al giorno precedente a quello attuale, e successivamente viene eseguito un processo schedulato ogni giorno per inserire le informazioni del giorno precedente.
- Fornire un'API interna al frontend per recuperare i dati dal database, in modo da non esporre direttamente l'API NASA o il database al frontend.

2.3 Frontend

Il frontend è stato realizzato usando JavaScript, HTML e CSS per creare una dashboard moderna e leggibile. Per la visualizzazione dei dati è stato utilizzato ApexCharts, una libreria JavaScript per la creazione di grafici interattivi.

3 Query principali

Di seguito sono riportate le query principali utilizzate per recuperare i dati dal database:

sql

```

1 -- Query per recuperare tutti gli asteroidi con i loro avvicinamenti alla
2   terra ordinati per data di avvicinamento. Viene utilizzata dall'API per
3   interna per recuperare i dati da visualizzare nella dashboard.
4
5 SELECT
6   a.id,
7   a.name,
8   a.nasa_jpl_url,
9   a.is_potentially_hazardous,
10  a.is_sentry_object,
11  a.absolute_magnitude_h,
12  a.estimated_diameter_min_m,
13  a.estimated_diameter_max_m,
14  TO_CHAR(c.close_approach_date, 'YYYY-MM-DD') as close_approach_date,
15  c.miss_distance_km,
16  c.relative_velocity_kmh
17 FROM close_approaches c
18 JOIN asteroids a ON c.asteroid_id = a.id
19 ORDER BY c.close_approach_date ASC

```

```

1 -- Query per inserire gli asteroidi recuperati dall'API NASA nel database.
I campi $1, $2, ... $8 rappresentano i valori da inserire, che vengono
passati come parametri alla query, in modo da prevenire attacchi di SQL
injection. La clausola ON CONFLICT (id) DO NOTHING garantisce che se un
asteroide con lo stesso ID esiste già nel database, l'inserimento viene
ignorato, evitando così duplicati o errori.
2 INSERT INTO asteroids (
3   id, name, nasa_jpl_url, absolute_magnitude_h,
4   estimated_diameter_min_m, estimated_diameter_max_m,
5   is_potentially_hazardous, is_sentry_object
6 ) VALUES ($1, $2, $3, $4, $5, $6, $7, $8)
7 ON CONFLICT (id) DO NOTHING;                                sql

1 -- Query per inserire gli avvicinamenti alla terra recuperati dall'API
NASA nel database.
2 INSERT INTO close_approaches (
3   asteroid_id, close_approach_date, epoch_date_close_approach,
4   relative_velocity_kmh, miss_distance_km, orbiting_body
5 ) VALUES ($1, $2, $3, $4, $5, $6)
6 ON CONFLICT (asteroid_id, epoch_date_close_approach) DO NOTHING;

```

4 Insight ottenuti

Osservando i dati raccolti e visualizzati tramite grafici, è possibile ottenere alcune informazioni interessanti.

1. Guardando il grafico «Hazard Assessment» è possibile notare che la maggior parte degli asteroidi vicini alla Terra non sono potenzialmente pericolosi, ma ci sono comunque alcuni asteroidi che lo sono, e che quindi è importante monitorarli attentamente.
2. L'affermazione del punto precedente si conferma guardando anche il grafico «Daily Approach Count», che mostra che per ciascun giorno la maggior parte degli avvicinamenti alla Terra sono di asteroidi non pericolosi.
3. Dal grafico 3D è possibile notare che gli asteroidi più vicini alla Terra tendono ad essere anche quelli classificati come potenzialmente pericolosi.
4. Alcuni asteroidi classificati come non pericolosi sono monitorati costantemente dalla NASA (sentry object) perché potrebbero rappresentare una minaccia in futuro.

5 Riflessione Personale e Contestuale

Lavorare con dati esterni e API pubbliche è una sfida che ho affrontato durante lo sviluppo di [aw-cli](#), un'interfaccia a riga di comando (CLI) che interagisce con l'API GraphQL di [AniList](#). Questa esperienza mi ha insegnato che le API pubbliche possono essere imprevedibili, soggette a cambiamenti e possono presentare limitazioni di rate limit. Per questo motivo, nel progetto della dashboard, ho implementato un debouncing delle API interne per evitare di sovraccaricare il server con richieste frequenti. Per lo stesso motivo è presente un database che viene popolato con i dati dell'API NASA, in modo da non dover fare richieste all'API esterna ogni volta che il frontend richiede i dati.

6 Proposte di miglioramento

Il progetto potrebbe essere espanso e migliorato in diversi modi:

1. Quando vengono presi i dati giornalieri dall'API NASA, se sono presenti asteroidi pericolosi o molto vicini alla Terra, potrebbe essere inviata una notifica via email, Telegram Bot o altro canale di comunicazione, in modo da avvisare gli utenti senza dover accedere alla dashboard.
2. Potrebbe essere integrata anche l'API CAD, che fornisce i dati sugli asteroidi che si avvicineranno e che si sono già avvicinati alla Terra. In questo modo sarebbe possibile visualizzare il percorso che un asteroide ha seguito nel tempo e prevederne il percorso futuro, in modo da avere una visione più completa della situazione.
3. Potrebbe essere implementata una generazione di report periodici da un sistema di intelligenza artificiale, che analizza i dati raccolti e fornisce un'analisi approfondita della situazione degli asteroidi vicini alla Terra, evidenziando eventuali trend o anomalie nei dati.