

Progetto #2 - Prova in itinere DSBD aa2025-2026

Obiettivo

Dato il sistema sviluppato, estendere le funzionalità introducendo:

- Tutte le operazioni verso Open Sky Network devono essere protette da un Circuit Breaker.
- Un meccanismo asincrono di notifica che permetta all'utente di ricevere un messaggio (posta elettronica) ogni volta che il numero di voli in arrivo e/o in partenza da un dato aeroporto superano una certa soglia.
- Un API gateway (NGINX) come punto di ingresso del sistema.
- Un message broker (Kafka).

Modifiche sulle richieste dell'utente

L'utente potrà fornire - oltre all'aeroporto di interesse in modo indipendente - due parametri aggiuntivi, high-value e low-value.

Tali parametri potranno essere successivamente aggiunti/aggiornati.

La funzione di aggiornamento dell'utente dovrà essere modificata opportunamente per tenere conto dei nuovi parametri.

NOTE:

- Il database deve essere opportunamente modificato per supportare i nuovi parametri.
- Può essere presente anche un solo valore tra high-value e low-value.
- Se sono presenti entrambi, high-value deve essere maggiore di low-value.

Modifiche al DataCollector

Dopo aver recuperato gli ultimi valori degli aeroporti di interesse e aggiornato il database, il DataCollector invierà un messaggio (producer) su un topic Kafka* **to-alert-system** notificando i consumer che la fase di aggiornamento dei valori è stata completata e inviando gli ultimi dati raccolti.

Componente aggiuntivo: AlertSystem

Un servizio indipendente che, alla ricezione di un messaggio nel topic Kafka **to-alert-system** verifica per ogni profilo se il valore del numero di voli per un dato aeroporto è o maggiore di high-value o minore di low-value (se dati), invia un messaggio (kafka producer) sul topic Kafka **to-notifier** contenente i parametri <utente, aeroporto, condizione di superamento soglia>.

Componente aggiuntivo: AlertNotifierSystem

Un servizio indipendente che, alla ricezione di un messaggio sul topic Kafka **to-notifier** (kafka consumer), invia una email** con parametri "to: email, subject: aeroporto" e body: condizione di superamento soglia (specificando se è stata superata la soglia superiore o inferiore).

(*Inserire un broker Kafka come componente aggiuntivo)

(** potrebbe essere anche un tipo di messaggio diverso - e.g, Telegram. In questo caso bisogna usare un ID diverso per gestire la registrazione utente.)

Nuovi Requisiti Tecnici:

Circuit Breaker, Broker kafka e API gateway.

Il testo del presente homework mette in evidenza solo i nuovi punti richiesti, non è da considerarsi esaustivo in termini di architettura e di interazioni tra i componenti, infatti lo studente è libero di modificare l'architettura aggiungendo qualsiasi elemento ritenga necessario e di modificare le interazioni tra i componenti.

Note di progetto

Le scelte progettuali (database utilizzati, decomposizione in microservizi, eventuali pattern di comunicazione) sono oggetto di valutazione e devono essere opportunamente motivate. Ogni gruppo può liberamente personalizzare il progetto, considerando che la valutazione delle scelte progettuali prevale sull'implementazione del codice. **NON** si richiede alcuna interfaccia grafica o qualsiasi forma di frontend e non incide in alcun modo sulla valutazione.

Modalità e tempi di consegna

Il progetto completo va consegnato improrogabilmente entro il 19/12/2025 (hard deadline).

Si consiglia di inviare una email con oggetto “Consegna Homework#2 - Nome e Cognome di ogni studente del gruppo”.

Lo studente (ovvero il gruppo) deve esporre il progetto su GitHub e mandare il link tramite email al gruppo di docenti. (antonella.distefano@unict.it, giovanni.morana@unict.it, tutor alessandro.genovese@phd.unict.it).

È necessario differenziare il codice tra primo e secondo homework. Si consiglia di utilizzare una branch differente su github.

Assieme al progetto va consegnata:

- una relazione che illustri le scelte progettuali (e.g., descrizione dell'applicazione, possibili variazioni della schema architettonico - (micro)servizi e relative comunicazioni - lista delle API implementate). La relazione deve includere:
 - un diagramma architettonico che mostri i micro-servizi coinvolti;
 - un diagramma che mostri il dettaglio delle interazioni, sia tra componenti dell'applicazione che tra l'applicazione e il mondo esterno.
- un documento con le informazioni utili (se presenti) per build & deploy, da consegnare in formato Markdown o PDF.

Il progetto può essere presentato anche se incompleto.