



ByteOps.swe@gmail.com

Verbale Interno · Data: 26/02/2024

Informazioni documento

Luogo	Discord
Orario	09:30 - 11:00
Redattore	R. Smanio
Verificatore	E. Hysa
Amministratore	F. Pozza
Destinatari	T. Vardanega R. Cardin
Partecipanti	A. Barutta E. Hysa R. Smanio D. Diotto F. Pozza L. Skenderi N. Preto

Il responsabile: L. Skenderi

Registro delle modifiche

Versione	Data	Autore	Verificatore	Dettaglio
0.0.1	26/02/2024	R. Smanio	E. Hysa	Redazione documento

Indice

ByteOps

Febbraio 26, 2024

Contents

1	Revisione del periodo precedente	4
2	Ordine del giorno	4
2.1	Calcolo punteggio di salute	4
2.2	Progettazione e sviluppo database	5
2.3	Progettazione e sviluppo dashboards Grafana	5
2.4	Progettazione e sviluppo test	5
3	Attività da svolgere	6

1 Revisione del periodo precedente

La seconda fase della revisione RTB_G con il professor Vardanega, svolta il 20/02/2024, è stata completata con esito positivo. Successivamente, sono state apportate le correzioni indicate durante la revisione ai documenti, e sono state avviate le prime $attività_G$ in vista della Product Baseline.

È stata condotta un'analisi approfondita per pianificare in modo adeguato le $attività_G$ relative alla progettazione e al testing. Di conseguenza, è stata elaborata la struttura dei nuovi documenti, tra cui la Specifica Tecnica e il Manuale Utente, dei quali sono state redatte le sezioni introduttive.

Durante lo svolgimento delle $attività_G$ del periodo precedente non sono state riscontrate difficoltà significative che richiedano un cambiamento nell'approccio di lavoro, in quanto sia le aspettative che le tempistiche sono state rispettate.

2 Ordine del giorno

2.1 Calcolo punteggio di salute

Si è discusso sulle direttive da seguire per lo sviluppo della funzionalità del calcolo del punteggio di salute. Uno $script_G$ Faust è incaricato di estrarre i dati da $Kafka_G$ e di inviarli al modello per il calcolo del punteggio di salute. Il modello, attraverso un algoritmo dedicato, elabora i dati e genera il punteggio di salute corrispondente. Tale punteggio viene successivamente reinserito in $Kafka_G$ e memorizzato in $Clickhouse_G$.

Il modello calcola il punteggio di salute per ciascuna cella urbana seguendo una procedura dettagliata. In primo luogo, per ogni misurazione effettuata, viene calcolata la distanza rispetto ad un valore ideale corrispondente. Successivamente, per ogni tipologia di misurazione, viene calcolata una media delle distanze. In particolare, viene calcolata la media delle distanze per la temperatura, per le polveri sottili e per l'umidità. Infine, queste medie vengono sommate per ottenere il punteggio di salute complessivo della cella. È importante notare che se una cella non dispone di determinati tipi di sensori, il punteggio di salute non viene incrementato per quei parametri.

In questo modo, è possibile ottenere il punteggio di salute dell'intera città sommando i punteggi di salute delle varie celle urbane o di una o più specifiche celle.

2.2 Progettazione e sviluppo database

In risposta alla richiesta della *proponente_G* di semplificare la struttura del *database_G* per evitare sovra-ingegnerizzazione e migliorare la manutenibilità, è stata presa la decisione di adottare un approccio più semplice. Questo ha comportato la creazione di una singola tabella contenente le varie misurazioni provenienti dai sensori, con eventuali aggregazioni dei dati effettuate al momento tramite *query_G*. Questa scelta è stata motivata dal fatto che, nel precedente approccio basato su materialized view e projection, venivano create tabelle separate per memorizzare gli aggregati delle misurazioni. Tuttavia, la *proponente_G* ha contestato tale approccio, sottolineando che nel contesto specifico non vi è la necessità di utilizzare tali funzionalità. Al contrario, l'adozione di tale approccio avrebbe comportato la creazione di nuove tabelle dedicate alle misurazioni aggregate soggette a costi di manutenzione potenzialmente elevati.

2.3 Progettazione e sviluppo dashboards Grafana

Durante la riunione, è stata definita la struttura finale della *dashboard_G Grafana_G*. Oltre alla *dashboard_G* principale, è stata presa la decisione di creare una *dashboard_G* personalizzata che offre un'analisi dettagliata delle misurazioni di una tipologia specifica. Questa *dashboard_G* consente l'applicazione di diversi filtri all'interno di un ambiente dedicato, permettendo così un'analisi approfondita.

Inoltre, si è optato per l'impiego di un plugin che consenta di posizionare le variabili accanto ai grafici, agevolando la selezione di sensori o celle specifiche per visualizzare le relative misurazioni in modo intuitivo e semplice. In seguito a ciò sono stati aggiornati alcuni dei casi d'uso presenti nel documento Analisi dei Requisiti.

2.4 Progettazione e sviluppo test

Si è deciso di realizzare *test_G* di unità relativi ai simulatori dei sensori. Inoltre, si è pianificato di eseguire *test_G* di *integrazione_G* per verificare l'integrità delle misurazioni che passano attraverso *Kafka_G*, vengono eventualmente elaborate dallo *script_G* Faust, memorizzate in *Clickhouse_G* e visualizzate nella *dashboard_G Grafana_G*. A tal proposito, è stato deciso di sviluppare una suite di *test_G* automatizzati per garantire che il flusso di dati rimanga integro durante la trasmissione tra i vari componenti dell'*architettura_G*, partendo dai sensori fino a *Clickhouse_G*.

Infine, è stato deciso di sviluppare stress *test_G* per valutare le prestazioni del *sistema_G* e verificarne la resilienza in condizioni di carico elevato.

3 Attività da svolgere

Titolo	# Issue	Verificatore
Modificare struttura <i>repository_G</i>	93	L. Skenderi
Redazione verbale interno 26/02/2024	94	E. Hysa
Stesura periodo 9 - PdP	98	N. Preto
Progettazione <i>dashboard_G</i> <i>Grafana_G</i>	104	E. Hysa
Implementazione calcolo punteggio salute	105	L. Skenderi
Progettazione <i>database_G</i>	106	N. Preto
Sviluppo <i>database_G</i>	107	A. Barutta