



ByteOps.swe@gmail.com

Verbale Interno · Data: 05/03/2024

Informazioni documento

Luogo	Discord
Orario	17:30 - 18:30
Redattore	N. Preto
Verificatore	A. Barutta
Amministratore	E. Hysa
Destinatari	T. Vardanega R. Cardin
Partecipanti interni	A. Barutta E. Hysa R. Smanio D. Diotto F. Pozza L. Skenderi N. Preto

Il responsabile: R. Smanio

Registro delle modifiche

Versione	Data	Autore	Verificatore	Dettaglio
0.0.1	05/03/2024	N. Preto	A. Barutta	Redazione documento

Indice

ByteOps

Marzo 05, 2024

Contents

1	Revisione del periodo precedente	4
2	Ordine del giorno	4
2.1	Notifiche Grafana su Discord	4
2.2	Time-To-Live ClickHouse	4
2.3	Continuous Integration	5
3	Attività da svolgere	6

1 Revisione del periodo precedente

Dopo l'ultimo incontro, il gruppo ha proseguito con il lavoro sulle *attività_G* pianificate. In particolare, sono stati sviluppati nuovi *test_G* di unità per i sensori e si è proceduto con lo sviluppo dei *test_G* di *integrazione_G* tra *Kafka_G*, *Python_G* e *ClickHouse_G*, come previsto durante l'incontro precedente.

Inoltre, per quanto riguarda la redazione dei documenti, è stata completata la redazione della sezione relativa al decimo periodo del *Piano di Progetto*, e si è proseguito con la stesura del *Manuale Utente*.

In conclusione, le *attività_G* pianificate nell'ultimo incontro sono attualmente in esecuzione secondo i tempi previsti in quanto non sono emersi dubbi o difficoltà durante il loro svolgimento.

2 Ordine del giorno

L'incontro odierno è stato convocato in via straordinaria al fine di coordinare le ultime *attività_G* prima della conclusione del progetto.

Data l'importanza di questo momento cruciale, è stato ritenuto necessario convocare una riunione al fine di garantire un'efficace sincronizzazione delle azioni rimanenti e garantire il completamento del progetto conformemente agli *standard_G* di qualità definiti e nei tempi previsti.

2.1 Notifiche Grafana su Discord

Durante la riunione, il gruppo ha esaminato la possibilità di inviare le notifiche da *Grafana_G* su *Discord_G* al fine di informare tempestivamente gli utenti su eventuali problemi o anomalie rilevate dal *sistema_G*.

Dopo aver valutato la fattibilità di questa funzionalità e aver confermato la possibilità di implementarla, è stata presa la decisione di procedere con la sua realizzazione.

Nel corso dell'incontro, sono stati definiti inoltre i dettagli tecnici e le modalità di implementazione di questa funzionalità. In particolare, si è definito il formato delle notifiche e le modalità di ricezione delle stesse tramite il server *Discord_G* dedicato.

2.2 Time-To-Live ClickHouse

Si è discusso sulla possibilità di implementare la funzionalità Time-To-Live (TTL) per gestire in modo efficiente i dati memorizzati nel *database_G* *ClickHouse_G*.

La decisione di implementare tale funzionalità è stata presa per gestire in modo efficiente grandi volumi di dati e garantire la pulizia automatica di informazioni obsolete o non più rilevanti.

Sfruttando il motore Rollup, è stato possibile configurare l'eliminazione automatica dei dati meno recenti, specificando l'intervallo temporale a partire dal quale questi vengono rimossi. Ciò assicura l'efficienza delle operazioni nel *database_G* e favorisce una maggiore manutenibilità. Tuttavia, al fine di preservare la capacità di analizzare le misurazioni meno recenti, prima di procedere alla loro rimozione, viene effettuata un'operazione di aggregazione che consiste nel calcolare la media oraria delle misurazioni e memorizzare nel *database_G* solamente un singolo dato per ogni ora.

Questa strategia mira a ottimizzare le prestazioni complessive del *sistema_G* e a gestire in modo efficiente i grandi volumi di dati accumulati nel tempo.

2.3 Continuous Integration

Durante l'incontro, si sono analizzati i vantaggi derivanti dall'implementazione della Continuous Integration (CI).

La decisione di adottare la CI è stata motivata dalla necessità di garantire un flusso di sviluppo fluido e affidabile, soprattutto considerando l'*integrazione_G* dei *test_G* automatizzati.

La CI offre la possibilità di integrare regolarmente il codice nel *repository_G* condiviso e di eseguire automaticamente una serie di *test_G*, tra cui *test_G* di unità, *test_G* di *integrazione_G* e *test_G* end-to-end. Questo approccio ci consente di individuare e risolvere tempestivamente eventuali problemi, riducendo al minimo il rischio di errori e garantendo che il codice rimanga sempre funzionante.

Inoltre, l'automazione dei *test_G* all'interno del processo di CI contribuisce a migliorare la qualità del *software_G*, poiché consente di identificare e correggere rapidamente eventuali difetti introdotti dalle nuove modifiche. Questa pratica non solo garantisce la correttezza e la qualità del *software_G*, ma favorisce anche una maggiore collaborazione tra i membri del team, promuovendo una cultura di sviluppo condiviso e feedback costante.

In definitiva, la scelta di utilizzare la CI è stata guidata dalla volontà di garantire un processo di sviluppo efficiente, affidabile e di alta qualità.

3 Attività da svolgere

Titolo	# Issue	Verificatore
Implementazione <i>sistema_G</i> di notifiche <i>Grafana_G</i> su <i>Discord_G</i>	112	A. Barutta
Implementazione TTL in <i>ClickHouse_G</i>	111	A. Barutta
Implementazione della Continuous Integration	113	R. Smanio
Implementazione calcolo punteggio di salute	105	R. Smanio