



Università degli Studi di Padova

Laurea: Informatica

Corso: Ingegneria del Software

Anno Accademico: 2025/2026



Gruppo RubberDuck

email: GroupRubberDuck@gmail.com

Verbale Riunione con M31

2025-10-23

Vers.	Data	Autore	Verificatore	Descrizione
1.0.0	2025-10-29	Cristian Pirlog	Cristian Pirlog	Approvazione
0.1.0	2025-10-28	Davide Testolin	Filippo Guerra	Prima stesura

Indice

Informazioni generali	1
Ordine del giorno	2
Domande	3
Estensioni future	3
Segregazione tenant	3
Configurazione sensori	3
App di terze parti	3
Failover automatico	4
Scalabilità	4
Documentazione codice	4
Stato dell'Arte	4
Incontri e comunicazione	5
Test di sicurezza	5
Simulatore gateway	5
Protocollo di comunicazione	6
Persistenza dei dati	6
Approvazione esterna	7

Informazioni generali

- **Tipo di riunione** : Esterno
- **data**: 2025-10-23
- **luogo**: Riunione su Zoom
- **Ora inizio**: 15:00
- **Ora fine**: 15:45
- **Scriba**: Davide Testolin
- **Partecipanti**:
 - Davide Testolin
 - Filippo Guerra
 - Aldo Bettega
 - Cristian Pirlog (M31)

Ordine del giorno

- Presentazione e analisi con l'azienda M31 delle 13 domande elaborate dal gruppo durante la fase preparatoria.

Domande

Le domande poste sono state le seguenti:

Estensioni future

Nel documento viene detto “Il progetto deve prestarsi a molteplici estensioni future, tra cui l’integrazione con sistemi di analisi avanzata e algoritmi predittivi”. Quali sono questi sistemi di analisi avanzata e algoritmi predittivi? In che cosa devono essere predittivi?

Risposta 1

Ci potrebbe essere l’integrazione con sistemi di analisi avanzata e algoritmi predittivi ma si tratta di un “**nice to have**”. I dati potrebbero passare attraverso una serie di stadi di “processing” prima di arrivare al cliente finale.

Segregazione tenant

Nella pratica in che modo e quali strumenti vengono utilizzati per realizzare una segregazione tra tenant?

Risposta 2

Sostanzialmente ogni tenant è un cliente. Tenere i dati in un singolo DB potrebbe essere problematico se si hanno ad esempio centinaia di migliaia di sensori che generano dati contemporaneamente. Quindi sia a livello logico di codice che a livello fisico di DB separati (oppure vedi la pagina Row Security Policy di Postgres).

Configurazione sensori

La configurazione di nuovi sensori o gateway deve essere disponibile anche a lato client? Se sì perché?

Risposta 3

Direi che questi dettagli potremo definirli insieme più avanti, poiché in uno dei primi incontri sarà necessario anche discutere e concordare i requisiti elencati. Tali requisiti non sono scolpiti nella pietra. Sta a voi valutarli e segnare tempestivamente eventuali difficoltà nel realizzarli. In sostanza, ci aspettiamo di avere una dashboard_G dove possiamo vedere i dati trasmessi dai sensori attivi in real time, magari con qualche filtro, per il tenant corrente.

App di terze parti

Nei requisiti opzionali si parla di integrazione di third-party-app, in che punto del sw ci si aspetta che vengano integrate? Ci sono consigli su quali applicare?

Risposta 4

A livello di API, alla fine di tutta la catena. Una volta che il dato è stato filtrato e processato, il cliente potrà accedervi attraverso le API del cloud.

Failover automatico

Nei requisiti opzionali di sicurezza si parla di resilienza agli attacchi tramite failover automatico. Questo implica che il sw deve essere dotato di un sistema di riserva? Se sì quali parti del sistema devo duplicare e in che modo?

Risposta 5

Dovete studiare voi una strategia per mitigare questi attacchi.

Scalabilità

Nei requisiti generici si parla di scalabilità, vorremmo saperne di più.

Risposta 6

Sì, il sistema deve essere pensato per essere scalabile. I servizi devono essere “stateless” e i dati letti/scritti dal DB. Tutti i servizi devono poter accedere allo stesso stato.

Documentazione codice

Nella parte di documentazione si parla di “documentazione del codice”, nell’atto pratico che cosa ci si aspetta che contenga? Una spiegazione dettagliata del funzionamento del codice per chi dovrà guardarlo? Una specie di guida pratica?

Risposta 7

Un README con la struttura del repository, organizzazione, principi base, istruzioni per compilare il codice e test. Inoltre la documentazione nel codice: i commenti (es. Javadoc).

Stato dell’Arte

È possibile avere delle dritte per il requisito di analisi dello stato dell’arte? Esempi di piattaforme da analizzare? Quale dovrebbe essere la struttura approssimativa?

Risposta 8

Sicuramente, all’occorrenza, possiamo darvi delle indicazioni al momento opportuno. Anche se avete già a disposizione un esempio di quella che è l’architettura a grandi linee.

Incontri e comunicazione

Sarebbe possibile organizzare degli incontri con i referenti tecnici del progetto? E se sì, con quale frequenza? Con quale sistema ci teniamo in contatto per eventuali aggiornamenti (Email, telegram, ecc.)?

Risposta 9

Per la frequenza due tipi di incontri in giorni settimanali fissi:

- incontro “formale” bisettimanale di stato avanzamento lavori e allineamento;
- incontro bisettimanale breve (nelle settimane in cui non c'è l'incontro “formale”) per risolvere i problemi insieme, preferibilmente inviando qualche giorno prima le domande.

Test di sicurezza

Per quanto riguarda i test di sicurezza elencati a pagina 12 avete delle preferenze riguardo alla tipologia di penetration test sulla quale dovremo concentrarci maggiormente (es cloud testing, API testing)?

Risposta 10

Solo nei punti più critici dell'applicazione, che dovrete individuare voi.

Simulatore gateway

Capitolo 2, verso la fine è scritto: “Nel contesto del progetto, gli studenti non realizzeranno un gateway fisico ma svilupperanno un simulatore di gateway, in grado di riprodurre in maniera verosimile il comportamento di un nodo BLE-WiFi. Tale simulatore sarà utilizzato per validare l'infrastruttura cloud e per testare i flussi di comunicazione previsti”. Per quanto riguarda il simulatore del gateway BLE-WiFi:

- Saranno forniti esempi o specifiche dei flussi di comunicazione tra il gateway (Edge Layer) e il cloud (Control & Management Layer)?
- Quale livello di complessità è richiesto per il simulatore? Deve anche gestire scenari di errore (es. perdita di connessione con i sensori o con il cloud), oppure è sufficiente simulare un comportamento ideale per la validazione del sistema?

Risposta 11

- No, perchè è uno degli ottetti del capitolato, ma se necessario si può venire incontro per definire il flusso al momento opportuno.
- Il grado di complessità richiesto è il minimo indispensabile. Ci aspettiamo, oltre al caso ideale, almeno di gestire almeno qualche caso di errore come il caso di interruzione della comunicazione tra Gateway e Cloud (es. salta la corrente in casa e il gateway continua a raccogliere dati grazie alla propria batteria interna, per poi inviarli quando sarà stata ristabilita la connessione al cloud). Tutto ciò che comprende la connessione tra sensori e gateway è fuori scope. Il simulatore deve essere configurabile (es. attraverso un file YAML/JSON), per poter indicare i tipi di sensori connessi da simulare e generare dati verosimili.

Tipi di sensori possono essere quelli per misurare i parametri vitali di una persona (es. battito cardiaco, ECG, pressione sanguigna, saturazione ossigeno, livelli di glicemia, peso, ecc.).

Protocollo di comunicazione

Avete una preferenza sul protocollo di comunicazione tra gateway e cloud (HTTPS o MQTT)? Meglio solo uno o entrambi?

Risposta 12

Sì, MQTT, purché non produca ritardi, in quel caso meglio HTTPS.

Persistenza dei dati

Nel capitolo 3.3 è scritto: “Include un sistema di persistenza temporanea dei dati, con possibilità di estensione a database a lungo termine.” Quindi nell’MVP è richiesto che l’applicazione salvi i dati in un database a lungo termine?

Risposta 13

Sì, è richiesto che nell’MVP l’applicazione salvi i dati in un database a lungo termine.

Approvazione esterna

Il confronto ha permesso di chiarire alcuni dei nostri dubbi, rappresentando un'importante occasione di condivisione e collaborazione.

Con la seguente firma, il proponente esterno conferma e valida il presente documento.

