roundabout

Roundabout - Etherless

Studio di fattibilitÃă

Versione	1.0.0
Approvazione	Marco Positello
Redazione	Egon Galvani
	Antonio Zlatkovski
Verifica	Alessandro Sgreva
	Feim Jakupi
Stato	Approvato
Uso	Interno
Destinato a	Roundabout
	Prof. Tullio Vardanega
	Prof. Riccardo Cardin

Descrizione
Studio di fattibilitÃă dei capitolati$_G$ proposti.

team.roundabout.13@gmail.com

Registro delle modifiche

Versione	Data	Nominativo	Ruolo	Descrizione
1.0.0	2020-04-10	Marco Positello	*Responsabile*	Approvazione del documento.
0.2.0	2020-03-17	Feim Jakupi	*Verificatore*	Revisione e modifica del documento.
0.1.3	2020-03-13	Antonio Zlatkovski	*Analista*	Stesura dello studio di fattibilità del capitolato$_G$ C6.
0.1.2	2020-03-13	Antonio Zlatkovski	*Analista*	Stesura dello studio di fattibilità del capitolato$_G$ C5.
0.1.1	2020-03-13	Antonio Zlatkovski	*Analista*	Stesura dello studio di fattibilità del capitolato$_G$ C4.
0.1.0	2020-03-13	Alessandro Sgreva	*Verificatore*	Revisione e modifica del documento.
0.0.4	2020-03-13	Egon Galvani	*Analista*	Stesura dello studio di fattibilità del capitolato$_G$ C3.
0.0.3	2020-03-12	Egon Galvani	*Analista*	Stesura dello studio di fattibilità del capitolato$_G$ C1.
0.0.2	2020-03-12	Egon Galvani	*Analista*	Stesura dello studio di fattibilità del capitolato$_G$ C2.
0.0.1	2020-03-12	Egon Galvani	*Analista*	Creazione documento LaTeX$_G$ e struttura generale.

Indice

1 Introduzione

1.1 Scopo del documento:

Elencare le motivazioni, i punti positivi e negativi considerati, che hanno portato alla scelta del capitolato$_G$ C2 denominato *Etherless*, e alla consecutiva esclusione degli altri capitolati$_G$ disponibili.

1.2 Glossario

I termini tecnici utilizzati in questo documento sono contrassegnati da una 'G' a pedice ed ulteriormente approfonditi nel Glossario denominato "Glossario Interno 1.0.0", disponibile in allegato al presente documento.

1.3 Riferimenti

1.3.1 Riferimenti normativi

- **Norme di Progetto**: *Norme di Progetto v1.0.0.*

1.3.2 Riferimenti informativi

- **Capitolato$_G$ d'appalto C1: Autonomous Highlights$_G$ Platform**
  https://www.math.unipd.it/~tullio/IS-1/2019/Progetto/C1.pdf;
- **Capitolato$_G$ d'appalto C2: Etherless**
  https://www.math.unipd.it/~tullio/IS-1/2019/Progetto/C2.pdf;
- **Capitolato$_G$ d'appalto C3: NaturalAPI**
  https://www.math.unipd.it/~tullio/IS-1/2019/Progetto/C3.pdf;
- **Capitolato$_G$ d'appalto C4: Predire in Grafana$_G$**
  https://www.math.unipd.it/~tullio/IS-1/2019/Progetto/C4.pdf;
- **Capitolato$_G$ d'appalto C5: Stalker**
  https://www.math.unipd.it/~tullio/IS-1/2019/Progetto/C5.pdf;
- **Capitolato$_G$ d'appalto C6: ThiReMa - Things Relationship Management**
  https://www.math.unipd.it/~tullio/IS-1/2019/Progetto/C6.pdf.

2 Capitolato scelto C2 - *Etherless*

2.1 Informazioni generali

- **Nome:** *Etherless*;
- **Proponente$_G$:** *Red Babel*;
- **Committente$_G$:** Prof. Tullio Vardanega e Prof. Riccardo Cardin.

2.2 Descrizione

Etherless è una piattaforma cloud$_G$ che permette agli sviluppatori di effettuare il deploy$_G$ di funzioni Javascript$_G$ al suo interno e far pagare gli utenti finali per la loro esecuzione da remoto, tramite l'utilizzo di smart-contract$_G$. Etherless si basa sull'integrazione di due tecnologie: Serverless$_G$ e Ethereum$_G$. Ethereum$_G$ viene utilizzato per la gestione dei pagamenti e delle richieste di esecuzione di funzioni. Serverless$_G$, invece, per l'effettiva esecuzione di tali funzioni. La comunicazione tra queste due tecnologie è resa possibile tramite l'ascolto e l'emissione di eventi$_G$ Ethereum$_G$.

2.3 Obiettivo finale

La struttura richiesta per Etherless si compone di 3 moduli$_G$ principali:

- **Etherless-cli$_G$:** è il modulo$_G$ con cui gli sviluppatori interagiscono con Etherless. Supporta diversi comandi, tramite i quali lo sviluppatore può:
  - configurare il proprio account;
  - eseguire il deploy$_G$ di funzioni Javascript$_G$;
  - visualizzare la lista delle funzioni di cui ha già svolto il deploy$_G$;
  - richiedere l'esecuzione di una funzione e visualizzarne il risultato;
  - visualizzare i log$_G$ relativi all'esecuzione di una specifica funzione;

  A seconda del comando, *Etherless-cli$_G$* si occuperà di emettere o rimanere in attesa di un determinato evento$_G$ Ethereum$_G$.
- **Etherless-smart:** un insieme di Ethereum$_G$ smart-contract$_G$ che gestiscono la comunicazione tra *Etherless-cli$_G$* e *Etherless-server*, ed i pagamenti in ETH$_G$ necessari per l'esecuzione delle funzioni;
- **Etherless-server:** si occupa di ascoltare gli eventi$_G$ trasmessi da *Etherless-smart* e di avviare l'esecuzione delle funzioni così richieste. I risultati ottenuti vengono inviati tramite un ulteriore evento$_G$ nella blockchain$_G$ e mostrati all'utente attraverso *Etherless-cli$_G$*;

2.4 Tecnologie coinvolte

- **Ethereum$_G$:** piattaforma che permette ai suoi utenti di scrivere applicazioni decentralizzate (dette ĐApps$_G$) che usano la tecnologia blockchain$_G$;
- **Ethereum$_G$ Virtual Machine$_G$ (EVM$_G$):** macchina virtuale distribuita sulla rete Ethereum$_G$;

- **Solidity$_G$:** linguaggio usato per la codifica degli smart-contract$_G$;
- **Serverless$_G$ Framework$_G$:** framework$_G$ per la realizzazione di applicazioni che vengono eseguite in architetture serverless$_G$ (come AWS$_G$ Lambda, Google Cloud$_G$ Functions, Microsoft Azure Functions);
- **Typescript$_G$ 3.6:** Linguaggio che supporta Javascript$_G$ ES6, introducendo alcune funzionalità, tra cui la tipizzazione;
- **ESLint$_G$:** tool di analisi statica del codice e syntax checking;
- **AWS$_G$ Lambda:** consente di eseguire codice in risposta a determinati eventi$_G$, senza dover effettuare il provisioning né gestire server.

2.5 Aspetti positivi

- La blockchain$_G$ e le tecnologie serverless$_G$ sono argomenti che hanno suscitato un notevole interesse generale negli ultimi anni; per questo la relativa conoscenza potrebbe essere positiva da un punto di vista curricolare;
- la conoscenza di Typescript$_G$ e Javascript$_G$ è molto richiesta, oltre ad essere la base di svariate librerie e framework$_G$;
- il tema principale del progetto è stato accolto con molto interesse dal gruppo, che si è dimostrato disponibile ad approfondire l'argomento.

2.6 Criticità

- L'azienda ha sede all'estero, quindi la comunicazione con i referenti sarà meno agevole rispetto ad aziende che si trovano nel territorio nazionale;
- il capitolato$_G$ richiede l'utilizzo di tecnologie recenti, la cui documentazione potrebbe non essere pienamente esaustiva e il cui apprendimento richiederà una mole di studio non indifferente.

2.7 Esito

Il progetto è stato subito accolto caldamente dai componenti del gruppo, le sfide tecnologiche sono stimolanti e gli argomenti trattati sono di interesse. Con questo progetto tutti i componenti del team avranno l'opportunità di studiare un campo dell'informatica che all'università è poco trattato, potendo aggiungere al proprio bagaglio curricolare delle voci particolarmente interessanti.

3 Valutazione capitolati rimanenti

3.1 Capitolato C1 - Autonomous Highlights Platform

3.1.1 Informazioni generali

- **Nome:** Autonomous Highlights$_G$ Platform;
- **Proponente$_G$**: Zero12;
- **Committente$_G$**: Prof. Tullio Vardanega e Prof. Riccardo Cardin.

3.1.2 Descrizione

Creare una piattaforma Web che riceve in input dei video di eventi sportivi, come una partita di calcio o una gara di F1/Moto GP, e crea in autonomia un video di massimo 5 minuti contenente i suoi momenti salienti.

3.1.3 Obiettivo finale

Viene richiesto che la piattaforma sia dotata di un modello$_G$ di machine learning$_G$ in grado di identificare ogni momento importante dell'evento. Il flusso di generazione di tale video deve essere così strutturato:

- caricamento del video;
- identificazione dei momenti salienti;
- estrazione delle corrispondenti parti di video;
- generazione del video di sintesi.

3.1.4 Tecnologie coinvolte

- **Elastic Container Service$_G$ o Elastic Kubernetes Service:** è un servizio di orchestrazione di contenitori altamente dimensionabile ad elevate prestazioni;
- **DynamoDB:** database NoSQL$_G$ dalle alte performance, ideale per la conservazione di tag o altre informazioni a supporto dell'applicativo;
- **AWS$_G$ Transcode:** servizio gestito per la conversione ed elaborazione di diversi formati video;
- **Sage Maker$_G$:** servizio completamente gestito che copre l'intero flusso di lavoro dell'apprendimento automatico$_G$: etichettazione e preparazione dei dati, scelta di un algoritmo, formazione del modello$_G$, ottimizzazione di quest'ultimo per la distribuzione, generazione di previsioni ed avvio di azioni;
- **AWS$_G$ Rekognition video:** servizio di analisi video basato su apprendimento approfondito$_G$; è in grado di riconoscere i movimenti delle persone in un fotogramma e di riconoscere soggetti, volti, oggetti, celebrità e contenuti inappropriati.

3.1.5 Aspetti positivi

- Tutte le tecnologie coinvolte sono ben documentate;
- il Proponente$_G$ Zero12 fornisce attività di formazione sulle principali tecnologie AWS$_G$ e Wireframe$_G$, utilizzate dall'interfaccia della console Web di analisi e controllo dello stato di elaborazione dei video, per la generazione del video di highlights$_G$.

3.1.6 Criticità

- Il Proponente$_G$ non fornisce alcun data-set$_G$ per effettuare il training$_G$ del modello$_G$ di machine learning$_G$;
- l'identificazione manuale di tutti i momenti principali di un evento sportivo è un'azione ripetitiva e costosa in termini di tempo.

3.1.7 Esito

Il capitolato$_G$ non è risultato molto stimolante, in quanto lo sviluppo di alcune componenti sembra caratterizzato da attività ripetitive. Si è quindi deciso di non considerare questa proposta.

3.2 Capitolato C3 - NaturalAPI

3.2.1 Informazioni generali

- **Nome:** NaturalAPI;
- **Proponente**$_G$: teal.blue;
- **Committente**$_G$: Prof. Tullio Vardanega e Prof. Riccardo Cardin.

3.2.2 Descrizione

Creare un toolkit$_G$ per restringere il divario presente tra le specifiche di progetto e le API$_G$, permettendo così ai programmatori di dedicarsi principalmente allo sviluppo di nuove funzionalità piuttosto che provare a replicare il modello aziendale.

3.2.3 Obiettivo finale

Il toolkit$_G$ richiesto si dovrà comporre di tre parti:

- **NaturalAPI Discover:** si occupa di identificare ed estrarre da documenti di testo potenziali entità di business (oggetti/nomi), processi (azioni/verbi) e predicati. In questo modo viene creato il Business Domain Language$_G$ (BDL$_G$);
- **NaturalAPI Design:** si occupa di creare un Business Application Language$_G$ (BAL$_G$) a partire dai documenti in Gherkin$_G$;
- **NaturalAPI Develop:** converte il BAL$_G$ in API$_G$ scritte in uno dei linguaggi di programmazione e framework$_G$ supportati; il tutto gestendo sia la creazione di nuove repository$_G$ per il codice, sia la modifica di quelle già esistenti.

3.2.4 Tecnologie coinvolte

- **Natural language processing:** il toolkit$_G$ utilizza tecniche di elaborazione del linguaggio naturale$_G$;
- **Gherkin$_G$:** lo standard principale per scrivere in linguaggio naturale$_G$ specifiche di funzionalità;
- **Cucumber$_G$:** framework$_G$ per eseguire test automatici per BDD$_G$ (Behavior-driven development$_G$);
- **Swagger$_G$:** tool di generazione di codice che si basa su OpenAPI$_G$.

3.2.5 Aspetti positivi

- Non viene richiesta un'interfaccia grafica, lasciando quindi più tempo al team di dedicarsi allo sviluppo delle funzionalità richieste;
- il capitolato$_G$ è molto dettagliato e descrive in maniera precisa il comportamento richiesto dal toolkit$_G$.

3.2.6 Criticità

- Il capitolato$_G$ non ha suscitato molto interesse nel gruppo;
- gli argomenti trattati sono stati considerati complessi e di difficile comprensione.

3.2.7 Esito

Lo scopo del capitolato$_G$ non è risultato molto stimolante. Si è quindi deciso di non considerare questa proposta.

3.3 Capitolato d'appalto C4: Predire in Grafana

3.3.1 Introduzione al capitolato

- **Nome:** Predire in Grafana$_G$;
- **Proponente$_G$:** Zucchetti S.p.A;
- **Committente$_G$:** Prof. Tullio Vardanega, Prof. Riccardo Cardin.

3.3.2 Descrizione

Attualmente il Proponente$_G$ fa ampio uso del progetto open-source$_G$ di nome Grafana$_G$ per monitorare la "liveliness" dei propri sistemi, ovvero per avere una visione generale sullo stato di operatività di essi in tempo reale. Il Proponente$_G$ desidera estendere tali capacità aggiungendo un complesso sistema di previsioni che, basandosi sui dati raccolti, riescano a segnalare potenziali irregolarità in arrivo sui singoli componenti o quanto meno identificare le singole zone di intervento.

3.3.3 Obiettivo Finale

Il prodotto$_G$ finale richiesto è composto da due plug-in$_G$ scritti in Javascript$_G$ che, oltre a gestire Grafana$_G$, applicano Support Vector Machine$_G$ (SVM$_G$) e/o Regressione Lineare$_G$ (RL) al flusso dei dati ricevuti per allarmi o segnalazioni tra gli operatori del servizio Cloud$_G$ e la linea di produzione del software. Tali plug-in$_G$ devono essere in grado di applicare gli algoritmi sopra citati a nodi della rete specifici in base ai dati che ricevono in input. Le previsioni ottenute saranno successivamente inserite all'interno del sistema di monitoraggio di Grafana$_G$ come se fossero dati realmente registrati. Sarà di conseguenza possibile visualizzare i grafici e la dashboard$_G$ aggiornate con le previsioni del sistema. Il prodotto$_G$ finale può inoltre avere alcune funzionalità opzionali, tra le quali essere in grado di scatenare degli alert se i nodi collegati alle previsioni raggiungono o superano certi livelli di soglia oppure applicare delle trasformazioni alle misure lette dal campo per ottenere delle regressioni esponenziali o logaritmiche oltre a quelle lineari.

3.4 Tecnologie Coinvolte

- **Javascript$_G$:** linguaggio di programmazione lato client con il quale sono stati sviluppati Grafana$_G$ e alcune delle librerie consigliate per gli algoritmi SVM$_G$ e RL$_G$.
- **Grafana$_G$:** progetto open-source$_G$ che permette il monitoraggio dei dati in tempo reale.
- **Github$_G$:** piattaforma ospitante di progetti e librerie open-source$_G$.
- **Orange Canvas$_G$:** strumento che permette di comprendere meglio la SVM$_G$ e la RL$_G$.

3.4.1 Aspetti Positivi

- La collaborazione con un Proponente$_G$ di importanza mondiale, quale la Zucchetti S.p.A., è un'esperienza che offre la possibilità di crescita sia a livello personale che professionale, oltre a rappresentare un primo passo importante verso il mondo del lavoro;
- il progetto offre varie opportunità di apprendimento di nuovi concetti informatici e matematici, potenzialmente riutilizzabili in un secondo momento, vista la politica di "maximum reliability" dei servizi Cloud$_G$ sempre più numerosi;

- apprezziamo inoltre la decisione del Proponente$_G$ di condividere il prodotto$_G$ finale come progetto open-source$_G$, in modo tale da renderlo disponibile ad un numero sempre maggiore di utenti.

3.4.2 Criticità

Il progetto richiede la conoscenza di argomenti il cui approfondimento può impiegare più tempo di quello disponibile. In particolare nessuno dei membri del team ha avuto l'opportunità di affrontare argomenti di tale natura.

3.4.3 Esito

Vista la scarsa preparazione dei membri del team nell'ambito dei modelli di apprendimento automatico$_G$, la relativa complessità degli argomenti trattati e le alte aspettative da parte del Proponente$_G$ si è deciso di non proporsi per la realizzazione del capitolato$_G$.

3.5 Capitolato d'appalto C5: Stalker

3.5.1 Introduzione al capitolato

- **Nome:** Stalker;
- **Proponente$_G$:** Imola Informatica;
- **Committente$_G$:** Prof. Tullio Vardanega, Prof. Riccardo Cardin.

3.5.2 Descrizione

La normativa vigente in materia di sicurezza regolamenta la gestione delle presenze nei locali pubblici ed aperti al pubblico prevedendo una serie di precauzioni ed adempimenti volti a garantire uno sfollamento sicuro in caso di emergenze e bisogno. La verifica dell'applicazione di tali vincoli è attualmente compito di persone fisiche ed è conseguentemente soggetta a errori di natura umana. Si vuole automatizzare tale processo creando una rete di smartphone appartenenti ai visitatori del locale in questione. Basandosi sullo stato di tale rete e/o sulla posizione dei dispositivi recuperata tramite GPS, sarà possibile risalire al numero totale di individui presenti in una certa area geografica. Sarà inoltre possibile, per le reti adeguatamente configurate, identificare univocamente ogni client collegato alla rete, una funzionalità fondamentale per verificare la presenza dei dipendenti nel posto di lavoro.

3.5.3 Obiettivo Finale

Il prodotto$_G$ finale prevede un server dotato di UI e un applicazione sviluppata per Android oppure iOS.
Per comprendere meglio il problema definiamo i seguenti concetti:

- **Organizzazione:** un soggetto che ha interesse a tracciare le presenze delle persone all'interno dei propri luoghi, in maniera anonima o autenticata.
- **Luogo:** spazio fisico identificato da un insieme di coordinate geografiche. Ciascun luogo è riconducibile ad una organizzazione.
- **Tracciatura:** rilevamento della presenza all'interno di un luogo. Può essere:
  - **anonima:** quando il soggetto non è autenticato nell'organizzazione di riferimento;
  - **autenticata:** quando il soggetto è autenticato nell'organizzazione di riferimento.

Il server deve svolgere i seguenti compiti:

- permettere la registrazione delle organizzazioni e la definizione della loro struttura tramite una mappa interattiva, permettendo l'inserimento e la gestione dei luoghi ad essa appartenenti;
- prevedere una procedura di autenticazione specifica per l'organizzazione, nel caso in cui l'organizzazione decida di implementare una tracciatura autenticata;
- fornire un tipo di autenticazione che permetta di gestire le autorizzazioni degli utenti. In base al livello di autorizzazione del singolo utente sarà possibile fare ricerche e analisi statistiche sui dati registrati dall'organizzazione.

L'applicazione per smartphone riguarda maggiormente gli individui tracciati e deve avere le seguenti funzionalità:

- consultare la lista delle organizzazioni disponibili;
- fare il LogIn nella propria organizzazione, anche come utente anonimo se la propria organizzazione prevede la tracciatura autenticata;
- visualizzare il tempo reale della propria presenza e lo storico degli accessi.

Il sistema inoltre deve essere facilmente scalabile secondo la politica della scalabilità orizzontale per permettere la tracciatura anche in luoghi molto affollati, come le fiere. Per garantire tale funzionalità è necessario implementare dei test di carico che dimostrino il corretto funzionamento in situazioni normali, di carico e di sovraccarico.
Sono inoltre richiesti test di sistema con copertura di test >= 80% correlata di report, e la completa documentazione per quanto riguarda:

- scelte implementative e progettuali effettuate e relative motivazioni;
- problemi aperti e eventuali soluzioni proposte da esplorare.

3.6 Tecnologie Coinvolte

- **Java$_G$:** linguaggio di programmazione per applicazioni Android(versione 8 o superiori).
- **Swift$_G$:** linguaggio di programmazione per applicazioni iOS.
- **Python$_G$ o nodejs$_G$:** sviluppo del server back-end$_G$.
- **Javascript$_G$:** linguaggio di programmazione lato client.
- **Android Studio oppure XCode:** IDE$_G$ di sviluppo per applicazioni per Android e iOS.
- **Protocolli asincroni:** per le comunicazioni mobile-server.
- **Pattern di Publisher/Subscriber:** utilizzo dell'IAAS Kubernetes o di un PAAS, Openshift o Rancher, per il rilascio delle componenti del Server nonché per la gestione della scalabilità orizzontale.
- **API Rest$_G$:** API$_G$ attraverso le quali sia possibile utilizzare l'applicativo. In alternativa è possibile utilizzare **gRPC3**.

3.6.1 Aspetti Positivi

- Apprezziamo l'approccio semplice con cui si tenta di risolvere un problema di grande calibro particolarmente importante nella società di oggi. Stalker, oltre ad eliminare potenziali errori umani, rafforza e migliora i sistemi di sicurezza già presenti in modo elegante e non intrusivo;
- il prodotto$_G$ aiuta l'avanzamento tecnologico eliminando la necessità di registri cartacei per quanto riguarda la gestione della presenza nel posto di lavoro e il conteggio delle ore lavorative svolte;
- sono svariate le casistiche in cui questo progetto può dare valore aggiuntivo all'organizzazione ricevente, dalle piccole aziende alle fiere di dimensione notevoli. Di conseguenza, visto il numero elevato di potenziali utenti, pensiamo che questo progetto sia un ottimo modo per investire il nostro tempo.

3.6.2 Criticità

Le tecnologie coinvolte sono numerose e maggiormente sconosciute ai membri del team. Lo studio di tali tecnologie e lo svolgimento del lavoro richiesto può facilmente impiegare più tempo di quello disponibile.

3.6.3 Esito

Senz'altro trattasi di uno dei progetti più interessanti con svariate possibilità di applicazione. Purtroppo però, non essendo più tra i capitolati$_G$ disponibili, non è stato possibile candidarsi.

3.7 Capitolato d'appalto C6: ThiReMa - Things Relationship Management

3.7.1 Introduzione al capitolato

- **Nome:** ThiReMa - Things Relationship Management;
- **Proponente$_G$:** Sanmarco Informatica;
- **Committente$_G$:** Prof. Tullio Vardanega, Prof. Riccardo Cardin.

3.7.2 Descrizione

Si vuole applicare l'analisi predittiva su un insieme di dati eterogenei provenienti da un complesso di dispositivi IoT$_G$. Tali dispositivi, potenzialmente di natura distinta, comunicano con un database centralizzato dove verranno memorizzati ed elaborati i valori raccolti allo scopo di fornire un servizio di manutenzione preventiva.

3.7.3 Obiettivo Finale

Il prodotto$_G$ finale rappresenta una web-app che permetta di valutare la correlazione tra dati operativi, ovvero le misure, e i fattori influenzanti.
Tale applicazione deve permettere di memorizzare ed elaborare una grande quantità di misure eterogenee raccolte da numerosi dispositivi collegati alla rete.

- La memorizzazione deve essere veloce ed economica in termini di performance, vista la numerosità di dispositivi che si vogliono gestire;
- l'analisi dei dati deve poter fornire semplici previsioni sull'andamento dei dati. Viene richiesto inoltre un sistema automatizzato che segnali in anticipo la necessità di manutenzione di un certo dispositivo.

È richiesta la suddivisione della web-app in 3 macro-sezioni:

- censimento dei sensori e dei relativi dati;
- modulo$_G$ di analisi di correlazione;
- modulo$_G$ di monitoraggio per ente.

L'interfaccia utente è suddivisa in 2 parti:

- **interfaccia di gestione**: consente agli utenti autorizzati il monitoraggio remoto del cluster$_G$, inclusa la possibilità di attivare e disattivare singoli nodi, il censimento dei sensori, e la definizione degli utenti e delle loro autorizzazioni;
- **interfaccia di interrogazione** permette di seguire graficamente in tempo reale l'andamento di uno o più sensori, eventualmente aggregati, e visualizzare alcuni dati di interesse, ad es. numero di errori/giorno ecc.

3.8 Tecnologie Coinvolte

- **Java**$_G$**:** linguaggio di programmazione consigliato per lo sviluppo delle componenti Kafka;
- **Apache Kafka**$_G$**:** piattaforma a bassa latenza ed alta velocità per la gestione di feed dati in tempo reale;
- **Time-Series Database:** database in cui l'ordine temporale della raccolta dei dati è fondamentale. I database consigliati sono:
  - **PostgreSQL**;
  - **TimescaleDB5**;
  - **ClickHouse6**.
- **Bootstrap**$_G$**:** sviluppo Front-end$_G$.

3.8.1 Aspetti Positivi

- Il progetto rappresenta un'opportunità di imparare molto sul mondo dell'IoT$_G$, una tecnologia sempre più utilizzata;
- offre inoltre l'opportunità di migliorare le proprie conoscenze su argomenti parzialmente conosciuti dal team, come lo sviluppo server-side.

3.8.2 Criticità

- Scarso interesse da parte dei membri del team;
- poca conoscenza del mondo dell'IoT$_G$ e tecnologie coinvolte.

3.8.3 Esito

Vista la non disponibilità del capitolato$_G$, non è stato possibile candidarsi.