



ProApes

proapes11@gmail.com

Studio di Fattibilità

Versione	1.0.0-1.10
Data approvazione	2020-01-14
Responsabile	Fiammetta Cannavò
Redattori	Francesco Bari Federico Carboni
Verificatori	Valentina Signor Giacomo Piran
Stato	Approvato
Lista distribuzione	<i>ProApes</i> <i>Prof. Tullio Vardanega</i> <i>Prof. Riccardo Cardin</i>
Uso	Interno

Sommario

Studio di Fattibilità dei capitoli proposti.

Diario delle Modifiche

Versione	Data	Modifica	Autore	Ruolo
v1.0.0	2020-01-14	<i>Approvazione del documento per RR</i>	Fiammetta Cannavò	<i>Responsabile di Progetto</i>
v0.4.0	2019-12-24	<i>Revisione complessiva di coerenza e coesione (Verificatore: Giacomo Piran)</i>	Francesco Bari	<i>Analista</i>
v0.3.1	2019-12-23	<i>Corretti capitoli C2, C3 (Verificatore: Giacomo Piran)</i>	Francesco Bari	<i>Analista</i>
v0.3.0	2019-12-22	<i>Revisione complessiva di coerenza e coesione (Verificatore: Giacomo Piran)</i>	Francesco Bari	<i>Analista</i>
v0.2.1	2019-12-20	<i>Corretti capitoli C5, C6</i>	Federico Carboni	<i>Analista</i>
v0.2.0	2019-12-19	<i>Revisione complessiva di coerenza e coesione (Verificatore: Valentina Signor)</i>	Francesco Bari	<i>Analista</i>
v0.1.3	2019-12-18	<i>Aggiunti capitoli C2, C3 (Verificatore: Valentina Signor)</i>	Francesco Bari	<i>Analista</i>
v0.1.2	2019-12-17	<i>Aggiunti capitoli C5, C6 (Verificatore: Giacomo Piran)</i>	Federico Carboni	<i>Analista</i>
v0.1.1	2019-12-17	<i>Corretti capitoli C4, C1 e Introduzione (Verificatore: Giacomo Piran)</i>	Francesco Bari	<i>Analista</i>

v0.1.0	2019-12-16	<i>Revisione complessiva di coerenza e coesione (Verificatore: Valentina Signor)</i>	Francesco Bari	<i>Analista</i>
v0.0.3	2019-12-15	<i>Aggiunto capitolato C4, C1 (Verificatore: Valentina Signor)</i>	Francesco Bari	<i>Analista</i>
v0.0.2	2019-12-15	<i>Aggiunta introduzione (Verificatore: Giacomo Piran)</i>	Federico Carboni	<i>Analista</i>
v0.0.1	2019-12-13	<i>Prima stesura del documento (Verificatore: Giacomo Piran)</i>	Francesco Bari	<i>Analista</i>

Indice

1	Introduzione	5
1.1	Scopo del documento	5
1.2	Glossario	5
1.3	Riferimenti	5
1.3.1	Riferimenti normativi	5
1.3.2	Riferimenti informativi	5
2	Capitolato Scelto: C4 - Predire in Grafana	6
2.1	Informazioni generali	6
2.2	Descrizione del capitolato	6
2.3	Finalità del progetto	6
2.4	Tecnologie interessate	6
2.5	Aspetti positivi	7
2.6	Criticità e fattori di rischio	7
2.7	Conclusioni	7
3	Capitolato C1 - Autonomous Highlights Platform	9
3.1	Informazioni generali	9
3.2	Descrizione del capitolato	9
3.3	Finalità del progetto	9
3.4	Tecnologie interessate	9
3.5	Aspetti positivi	10
3.6	Criticità e fattori di rischio	10
3.7	Conclusioni	10
4	Capitolato C2 - Etherless	11
4.1	Informazioni generali	11
4.2	Descrizione del capitolato	11
4.3	Finalità del progetto	11
4.4	Tecnologie interessate	11
4.5	Aspetti positivi	12
4.6	Criticità e fattori di rischio	12
4.7	Conclusioni	12
5	Capitolato C3 - Natural API	13
5.1	Informazioni generali	13
5.2	Descrizione del capitolato	13
5.3	Finalità del progetto	13
5.4	Tecnologie interessate	13
5.5	Aspetti positivi	13
5.6	Criticità e fattori di rischio	14
5.7	Conclusioni	14
6	Capitolato C5 - Stalker	15
6.1	Informazioni generali	15
6.2	Descrizione del capitolato	15
6.3	Finalità del progetto	15
6.4	Tecnologie interessate	15
6.5	Aspetti positivi	15

6.6	Criticità e fattori di rischio	16
6.7	Conclusioni	16
7	Capitolato C6 - ThiReMa	17
7.1	Informazioni generali	17
7.2	Descrizione del capitolato	17
7.3	Finalità del progetto	17
7.4	Tecnologie interessate	17
7.5	Aspetti positivi	17
7.6	Criticità e fattori di rischio	17
7.7	Conclusioni	17

1 Introduzione

1.1 Scopo del documento

Il seguente documento ha come scopo la descrizione delle motivazioni che hanno portato alla scelta del capitolato_G C4, ovvero *Predire in Grafana*, da parte del gruppo di lavoro *ProApes*. Sono inoltre spiegate le cause della conseguente esclusione degli altri progetti proposti.

1.2 Glossario

All'interno del documento sono presenti termini che possono presentare significati ambigui o incongruenti a seconda del contesto. Al fine quindi di evitare l'insorgere d'incomprensioni viene fornito un glossario individuabile nel file *Glossario v1.0.0* contenente i suddetti termini e la loro spiegazione.

Nella seguente documentazione per favorire maggiore chiarezza ed evitare inutili ridondanze tali parole vengono indicate mettendo una "G" a pedice di ogni prima occorrenza del termine che si incontri ad ogni inizio di sezione.

1.3 Riferimenti

1.3.1 Riferimenti normativi

- *Norme di Progetto v1.0.0.*

1.3.2 Riferimenti informativi

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

2 Capitolato Scelto: C4 - Predire in Grafana

Il capitolato_G C4 è stato presentato dalla *Zucchetti S.p.A.*, software house italiana di rilevanza internazionale, che da 41 anni realizza procedure gestionali nell'ambito delle Tecnologie Web. Tali applicazioni vengono installate nella server farm dell'azienda stessa o in data center privati da parte dei clienti attraverso cloud pubblici.

2.1 Informazioni generali

- **Nome:** *Predire in Grafana* - monitoraggio predittivo per DevOps_G;
- **Proponente_G:** *Zucchetti S.p.A.*;
- **Committente_G:** *Prof. Tullio Vardanega e Prof. Riccardo Cardin.*

2.2 Descrizione del capitolato

L'azienda richiede l'implementazione di funzionalità predittive sulla base di dati raccolti fino a quel momento nei propri sistemi. Per il monitoraggio, *Zucchetti S.p.A.* sfrutta *Grafana_G*, un software open-source_G che offre un sistema di presentazione dei dati ottenuti tramite l'impiego di dashboard_G operative e sistemi di allarme, predisposti per inviare una segnalazione al raggiungimento di soglie prestabilite per tali dati. *Grafana* è inoltre estendibile attraverso diversi plug-in_G scritti in *JavaScript_G*: sarà questo il linguaggio da usare per lo sviluppo delle funzionalità richieste.

2.3 Finalità del progetto

Con i sistemi predittivi richiesti, l'azienda conta di sfruttare al massimo le potenzialità dell'approccio DevOps (Development e Operations, sviluppo e applicazione), non solo aumentando l'accuratezza in tempo reale dei resoconti del sistema, ma anche acquisendo la capacità di orientare in maniera più precisa gli interventi migliorativi alla linea di produzione del software. Il plug-in richiesto dovrà:

- leggere da un file in formato *JSON_G* la definizione dei calcoli da applicare (*SVM_G*, *Support Vector Machine*, o *RL_G*, *Regressione Lineare*) per poi associarli ad alcuni nodi della rete del flusso del monitoraggio;
- eseguire i calcoli previsti e produrre dei valori che potranno essere aggiunti al flusso del monitoraggio come se fossero stati rilevati dal campo;
- presentare i nuovi dati ottenuti tramite grafici e dashboard.

Sarà necessario un periodo di addestramento della *SVM* e della *RL*: questo potrà essere eseguito in alcuni casi all'interno della stessa piattaforma di *Grafana*, altrimenti dovranno essere condotti in un modulo apposito.

2.4 Tecnologie interessate

Il software richiesto dovrà utilizzare le seguenti tecnologie:

- **SVM:** *Support Vector Machine* utilizzata ai fini predittivi nella classificazione;
- **RL:** *Regressione Lineare* utilizzata per la previsione di valori numerici;
- **Grafana:** software dedicato all'analisi temporale e all'organizzazione di dashboard di monitoraggio;
- **JSON:** formato del file contenente i dati di addestramento con i parametri per le previsioni ottenute usando la *SVM* o la *Regressione Lineare*;
- **JavaScript:** linguaggio da impiegare per lo sviluppo del plug-in *Grafana* richiesto. Esso supporta nativamente il parsing *JSON*;

- **Java JMeter**: sistema utilizzato per effettuare test di carico;
- **InfluxDB**: database apposito finalizzato alla gestione di grandi moli di dati;
- **Java JMX**: standard per il monitoraggio della JVM;
- **NodeJS**: linguaggio per lo sviluppo di *API_G Restful JSON* a supporto dell'applicativo.

2.5 Aspetti positivi

- L'azienda proponente ha una lunga storia alle spalle ed è una delle maggiori esponenti nel proprio settore: questo ha dato sicurezza al gruppo e lo ha incoraggiato a collaborare con un organo così esperto e consolidato;
- Il proponente si è dimostrato aperto al confronto, disponibile al chiarimento e all'esposizione di concetti nuovi, facendo intendere di voler curare la formazione dei membri del gruppo sugli argomenti che non sono ancora stati trattati durante il nostro percorso universitario;
- L'uso di un approccio DevOps, innovativo all'interno di un'azienda storica nell'ambito dell'informatica;
- La presenza di una forte componente matematica nelle funzionalità da implementare, che ha suscitato l'interesse di diversi membri del gruppo;
- La possibilità di poter contribuire ad un progetto rilasciato con licenza open-source;
- La facilità di integrazione del prodotto in altri sistemi, per la natura stessa del plug-in e per l'assenza di vincoli intrinseci con il sistema per cui sarà sviluppato (al netto della fase di addestramento);
- La trasversalità del prodotto sviluppato, grazie alla flessibilità di addestramento;
- La chiarezza e completezza dell'esposizione del capitolato, riscontrati sia nel documento del capitolato che nella presentazione;
- La prossimità geografica dell'azienda alle sedi universitarie, fattore che renderebbe agevoli eventuali incontri di presenza;
- Interesse collettivo da parte dei membri del gruppo sulle tematiche di Machine Learning_G previste dal capitolato.

2.6 Criticità e fattori di rischio

- Non è emerso chiaramente come (con quali dati) andranno testati i prodotti una volta sviluppati;
- Ci sono vincoli molto stretti sulle tecnologie su cui lavorare (*Grafana* e di conseguenza *JavaScript*), le quali non sono familiari a tutti i membri del gruppo;
- L'apprendimento delle tecnologie o delle strumentazioni previste potrebbe risultare lento per quei membri del gruppo che non le hanno mai utilizzate in precedenza.

2.7 Conclusioni

Il capitolato in questione ha attirato sin da subito l'interesse del gruppo per la chiarezza e linearità della sua esposizione. La presenza di un numero relativamente ristretto di tecnologie da imparare e utilizzare permetterebbe inoltre di concentrare i propri sforzi al meglio su di esse, in modo da apprenderle profondamente per poterle poi adottare con maggior consapevolezza. Un altro fattore decisivo è stato dato dall'esplicita disponibilità del proponente a fornire supporto per la comprensione del Machine Learning, settore che attrae tutti i componenti del team, non essendo affrontato nei corsi della LT in Informatica. Anche la presenza di applicazioni matematiche a contesti reali ha contribuito al successo riscosso

dalla proposta della *Zucchetti S.p.A.*. A seguito di attente valutazioni, il gruppo *ProApes* ha eletto questo capitolato come propria prima scelta, fiducioso che i punti non del tutto chiari e le difficoltà che potrebbero insorgere durante il percorso potranno essere risolte in sinergia col proponente. Questo ovviamente verrà accompagnato dalla forte determinazione personale volta all'apprendimento delle tecniche predittive e delle strumentazioni richieste a tali fini.

3 Capitolato C1 - Autonomous Highlights Platform

3.1 Informazioni generali

- **Nome:** *Autonomous Highlights Platform*;
- **Proponente_G:** *Zero12*;
- **Committente_G:** *Prof. Tullio Vardanega e Prof. Riccardo Cardin.*

3.2 Descrizione del capitolato

L'obiettivo del capitolato è la creazione di una piattaforma web che, ricevuto in input il video di un dato evento sportivo, sia in grado di produrre un video contenente i momenti salienti (highlights) di tale evento. Poiché la sintesi del video di output dovrà essere automatica, la piattaforma dovrà includere un modello di *Machine Learning_G*, sul quale saranno necessarie delle sessioni di addestramento. Per la realizzazione del progetto è richiesto di focalizzarsi su una sola disciplina sportiva e focalizzare l'addestramento del software su tale sport.

3.3 Finalità del progetto

L'azienda propone il seguente flusso di operazioni per l'attività della piattaforma:

- caricamento del video sulla piattaforma da parte dell'utente tramite linea di comando;
- individuazione degli highlights;
- estrazione degli highlights individuati;
- sintesi del video riassuntivo mediante l'assemblamento di tali parti.

3.4 Tecnologie interessate

Il proponente raccomanda l'uso delle tecnologie di *Amazon Web Services_G* (AWS) per lo sviluppo del prodotto. In particolare, viene consigliato:

- **Elastic Container Service/Elastic Kubernetes Service:** servizio per la gestione di contenitori altamente dimensionabile ad elevate prestazioni;
- **DynamoDB:** database NoSQL dalle alte performance, ideale per la conservazione di tag o altre informazioni a supporto dell'applicativo;
- **AWS Transcode:** servizio gestito per la conversione ed elaborazione di diversi formati video;
- **Sage Maker:** servizio completamente gestito che copre l'intero flusso di lavoro dell'apprendimento automatico per etichettare e preparare i dati, scegliere un algoritmo, formare il modello, ottimizzarlo per la distribuzione, effettuare previsioni e intraprendere azioni;
- **AWS Rekognition video:** servizio di analisi video basato su apprendimento approfondito; è in grado di riconoscere i movimenti delle persone in un fotogramma e di riconoscere soggetti, volti, oggetti, celebrità e contenuti inappropriati;
- **NodeJS:** linguaggio per lo sviluppo di *API_G Restful JSON* a supporto dell'applicativo;
- **Python:** linguaggio per lo sviluppo delle componenti di *Machine Learning*.

3.5 Aspetti positivi

- Il gruppo era interessato all'apprendimento dell'ambiente *Amazon Web Services*, in quanto realtà che offre un'ampia gamma di servizi ancora estranei a tutti i membri;
- Realizzazione di un prodotto che permette di creare contenuti che quotidianamente vediamo in televisione (i video di highlights sono ampiamente utilizzati nei programmi sportivi);
- L'approccio agile e la sua componente dinamica volta alla costante produzione di software implementabile ha incuriosito i membri del gruppo.

3.6 Criticità e fattori di rischio

- La parte di addestramento del software, anche se fatta per una sola disciplina sportiva, è apparsa, in prima analisi, molto articolata e complessa;
- L'addestramento fatto su un singolo sport sembra limitante dal punto di vista dell'applicabilità del prodotto; d'altra parte un addestramento per più sport richiederebbe un'ingente mole di lavoro, data dal fatto che ogni disciplina sportiva ha i propri "momenti salienti", spesso senza alcun legame con quelli delle altre.

3.7 Conclusioni

Nonostante tale capitolato abbia destato particolare interesse all'interno del gruppo, dato soprattutto dalla possibilità di utilizzare tecnologie innovative, il team di lavoro ha valutato la complessità di tale progetto come molto elevata e ha preferito orientarsi verso un'altra alternativa altrettanto stimolante.

4 Capitolato C2 - Etherless

4.1 Informazioni generali

- **Nome:** *Etherless*;
- **Proponente_G:** *RedBabel*;
- **Committente_G:** *Prof. Tullio Vardanega e Prof. Riccardo Cardin.*

4.2 Descrizione del capitolato

Il capitolato richiede di realizzare una Cloud Application Platform_G che permetta agli sviluppatori software di caricare funzioni scritte in *JavaScript_G* nel cloud distribuito ed essere pagati dagli utenti finali, anch'essi sviluppatori software, per la loro esecuzione. Il tutto avverrà secondo il modello *Computation-as-a-Service* (CaaS). Il pagamento sarà effettuato tramite *Ethereum_G*, grazie al rapporto di indebitamento garantito dagli *Smart Contracts*.

4.3 Finalità del progetto

L'azienda propone quindi di realizzare una architettura basata su tre moduli principali:

1. **CLI (Command Line Interface):** permetterà agli sviluppatori di interagire con la piattaforma. Saranno garantite quindi le operazioni di configurazione dell'account, rilascio al cliente del codice eseguibile, visualizzazione di una lista delle funzioni già caricate dallo sviluppatore nel sistema e comandi per l'esecuzione di una singola funzione con il ritorno del risultato cercato. L'interfaccia in base al comando potrà quindi emettere un evento in *Ethereum*, eseguendo *smart contracts*, oppure attendere eventi emessi dall'*Ethereum* stesso (modulo Server).
2. **Smart:** costituito da un set di *smart contracts* che si occuperanno della comunicazione tra il modulo CLI e il modulo Server, oltre che alla riscossione di commissioni in Ether per garantire questa comunicazione e il pagamento finale.
3. **Server:** questo modulo rimarrà in attesa di eventi emessi dal modulo Smart per il rilascio e l'esecuzione di funzioni caricate sull'applicazione. Una volta che tali funzioni ritorneranno un risultato, il modulo Server emetterà un evento nella *blockchain_G* contenente l'informazione corrispondente a tale valore. L'evento verrà poi ascoltato dal modulo CLI, che mostrerà il risultato allo sviluppatore richiedente.

4.4 Tecnologie interessate

Il software richiesto dal capitolato dovrà fare uso delle seguenti tecnologie:

- **Ethereum:** piattaforma decentralizzata sul web per la creazione e pubblicazione peer-to-peer di *smart contracts*;
- **Smart Contacts:** protocolli che verificano e fanno rispettare la negoziazione o l'esecuzione di un contratto. Sono eseguibili sulla *Ethereum Virtual Machine*;
- **Solidity:** linguaggio di programmazione object-oriented per la scrittura di *smart contracts*;
- **TypeScript:** linguaggio di programmazione che estende *JavaScript* aggiungendo la tipizzazione statica;
- **MainNet:** *blockchain* pubblica per *Ethereum*;
- **AWS Lambda:** piattaforma di calcolo serverless guidata dagli eventi. È un servizio di calcolo che esegue codice in risposta ad eventi e automaticamente ne gestisce le risorse;

- **ESLint**: strumento di analisi del codice statico per identificare schemi problematici trovati nel codice *JavaScript*;
- **CloudFormation**: servizio completamente gestito che consente di usare linguaggi di programmazione per modellare ed effettuare il provisioning;
- **Serverless Freamwork**: framework sviluppato per la creazione di applicazioni su *AWS Lambda*;
- **JSON**: formato adatto all'interscambio di dati fra client e server, basato su *JavaScript*;
- **NodeJS**: linguaggio per lo sviluppo di *API_G Restful JSON* a supporto dell'applicativo.

4.5 Aspetti positivi

- Le tecnologie coinvolte nello sviluppo del prodotto sono innovative e accattivanti, opportunità di formazione per l'intero gruppo;
- Nel complesso, sia l'azienda che la sua proposta appaiono molto moderne e all'avanguardia.

4.6 Criticità e fattori di rischio

- Sia il documento del capitolato che la successiva presentazione sono parsi dispersivi e poco chiari;
- L'azienda proponente ha sede all'estero, quindi la comunicazione con i referenti sarà meno agevole rispetto ai rapporti con un'azienda che ha sede nel territorio nazionale.
- L'apprendimento delle tecnologie coinvolte, dato il loro numero e la loro complessità, richiederebbe molto tempo.

4.7 Conclusioni

Dopo un'attenta valutazione, il capitolato è stato escluso a seguito della elevata complessità percepita per lo sviluppo del software, nonostante abbia suscitato interesse per le tecnologie in esso coinvolte.

5 Capitolato C3 - Natural API

5.1 Informazioni generali

- **Nome:** *Natural API* - form specs to code, smoothly;
- **Proponente_G:** *teal.blue*;
- **Committente_G:** *Prof. Tullio Vardanega e Prof. Riccardo Cardin.*

5.2 Descrizione del capitolato

Il capitolato richiede di realizzare un insieme di strumenti software prototipizzati, attui a colmare il divario tra le specifiche di progettazione espresse in linguaggio naturale e le APIs_G utilizzate per implementarle. Grazie infatti alla generazione automatica di codice e all'elaborazione di linguaggio, il capitolato richiede di creare un software che, senza passare per il programmatore, generi le funzioni utili a svolgere mansioni precedentemente dichiarate in linguaggio naturale.

5.3 Finalità del progetto

Il progetto richiede dunque lo sviluppo di tre principali strumenti:

1. **NaturalAPI Discover:** si occuperà, a partire da un testo espresso in linguaggio naturale, di estrarre possibili entità, processi o predicati, utili alla realizzazione di una qualche procedura;
2. **NaturalAPI Design:** utilizzando i dati ricevuti dal modulo Discover, si occuperà di creare al momento, tramite *parsing_G* dei dati stessi, un *Business Application Language*;
3. **Natural API Develop:** un esportatore di linguaggi, convertirà il Business Application Language in una serie di funzioni/segnature, scritte in uno dei linguaggi di programmazione disponibili.

5.4 Tecnologie interessate

Il software richiesto dovrà utilizzare le seguenti tecnologie:

- **Cucumber:** framework utile alla lettura e alla scrittura di test case_G facilmente interpretabili;
- **Gherkin:** linguaggio utilizzato da *Cucumber* per scrivere test case automatici;
- **Hiptest:** piattaforma online per la scrittura di funzionalità BDD con inoltre funzionalità di generazione automatica del codice;
- **Swagger:** strumento per la generazione di codice;
- **OWL:** linguaggio di markup che rappresenta esplicitamente significato e semantica dei termini.

5.5 Aspetti positivi

- La possibilità di creare un collegamento tra il linguaggio naturale e quello di programmazione ha suscitato l'interesse di alcuni membri del gruppo;
- Mira a risolvere un problema non banale di interazione all'interno di un progetto in cui gli stakeholder_G appartengono a categorie molto diverse.

5.6 Criticità e fattori di rischio

- Il gruppo ritiene difficile realizzare un prodotto veramente efficace;
- La maggior parte del gruppo considera il lavoro troppo astratto;
- I membri del gruppo trovano troppo restrittivo l'applicazione del servizio alla sola lingua inglese, e dall'altro lato molto complicata l'applicazione a più linguaggi naturali (analogo al problema della disciplina sportiva unica nel software sviluppato nel capitolato C1, descritto nella sezione §3).

5.7 Conclusioni

Lo scopo del capitolato non ha attirato le preferenze della maggior parte dei membri del gruppo, che si sono quindi orientati verso alternative maggiormente stimolanti.

6 Capitolato C5 - Stalker

6.1 Informazioni generali

- **Nome:** *Stalker*;
- **Proponente_G:** *Imola Informatica*;
- **Committente_G:** *Prof. Tullio Vardanega, Prof. Riccardo Cardin*.

6.2 Descrizione del capitolato

Lo scopo del progetto è la raccolta di informazioni riguardanti le persone presenti all'interno di un locale per motivi di sicurezza e di controllo. I due casi d'uso presi in considerazione sono:

- monitoraggio di singole presenze dei dipendenti nei differenti luoghi di lavoro;
- monitoraggio delle persone presenti in una grande struttura fieristica, sia globalmente che nei singoli padiglioni.

6.3 Finalità del progetto

L'obiettivo è quello di realizzare un'applicazione che dovrà essere necessariamente installata sui dispositivi degli utenti, in grado di segnalare ad un server dedicato l'ingresso e l'uscita dell'utilizzatore dalle aree d'interesse (in modalità anonima o meno, a seconda delle esigenze). Il server deve essere in grado di gestire più organizzazioni, per ognuna delle quali deve:

- poter gestire diversi luoghi;
- decidere se effettuare o meno l'autenticazione.

6.4 Tecnologie interessate

L'azienda lascia la possibilità di scegliere se sviluppare l'applicativo per sistemi operativi *Android* o *iOS*, e consiglia:

- **Java 8 o superiore:** linguaggio di programmazione ad alto livello orientato agli oggetti;
- **Python:** linguaggio di programmazione ad alto livello, adatto a sviluppare applicazioni distribuite;
- **NodeJS:** linguaggio per lo sviluppo di API Restful JSON_G a supporto dell'applicativo;
- **Protocolli asincroni:** protocolli che si sincronizzano utilizzando i dati stessi. Utili per le comunicazioni app-server;
- **Kubernetes:** sistema open-source_G di orchestrazione e gestione dei container;
- **Openshift:** piattaforma applicativa per gestione dei container;
- **Rancher:** piattaforma open-source di orchestrazione multi-cluster.

6.5 Aspetti positivi

- Lo sviluppo di software in funzione della sicurezza dei luoghi pubblici è sembrato al gruppo un obiettivo valido e utile;
- Diversi dipendenti dell'azienda proponente_G saranno a disposizione del gruppo per supportare il lavoro e la formazione riguardo le nuove tecnologie;
- L'uso di un approccio DevOps, innovativo nell'ambito dell'informatica.

6.6 Criticità e fattori di rischio

- Durante il seminario dedicato al capitolato_G, la spiegazione del suo aspetto più peculiare, cioè l'implementazione della geolocalizzazione, è sembrata poco chiara;
- Il fatto che i dati restituiti dall'applicazione non saranno precisi ma approssimati fa supporre che lo sviluppo e il testing risulteranno abbastanza incerti;
- Se gli utenti del servizio desiderassero, per esempio, un elevato grado di precisione della funzionalità di geolocalizzazione, sarebbe necessario la presenza di supporti hardware specifici, talvolta molto costosi;
- Dare per scontato che un utente (non amministratore) si rechi sul posto di lavoro sempre con lo smartphone (strumento essenziale per il corretto funzionamento del prodotto) sembra un presupposto poco realistico.

6.7 Conclusioni

Nonostante il gruppo fosse inizialmente interessato a questo capitolato, esso è stato accantonato a causa della remota prospettiva di vedere una realizzazione concreta del prodotto richiesto.

7 Capitolato C6 - ThiReMa

7.1 Informazioni generali

- **Nome:** *ThiReMa*;
- **Proponente_G:** *Sanmarco Informatica*;
- **Committente_G:** *Prof. Tullio Vardanega, Prof. Riccardo Cardin*.

7.2 Descrizione del capitolato

Il capitolato_G richiede di sviluppare un software in grado di ricevere misurazioni da sensori eterogenei dislocati geograficamente e di accumularli in un database centralizzato. Questa applicazione viene poi completata da un servizio di notifica, basato su Telegram_G, per inoltrare in modo tempestivo le informazioni per la gestione di azioni urgenti.

7.3 Finalità del progetto

L'obiettivo finale è la creazione di una web-application in grado di valutare la correlazione tra misure e fattori influenzanti, che saranno successivamente utilizzati da uno o più algoritmi per l'analisi dei dati e una previsione dell'andamento. Dovrà inoltre anche essere possibile assegnare il monitoraggio dei dati, ad enti differenti prescelti in base alla tipologia di informazione rilevata.

7.4 Tecnologie interessate

- **Kafka:** piattaforma open-source_G di stream processing;
- **Java 8:** linguaggio di programmazione ad alto livello orientato agli oggetti;
- **Bootstrap:** raccolta di strumenti liberi per la creazione di siti e applicazioni per il Web;
- **Docker:** progetto open-source che automatizza il deployment_G di applicazioni all'interno di container.

7.5 Aspetti positivi

- È applicabile in molti ambiti diversi.

7.6 Criticità e fattori di rischio

- Non ha attirato l'interesse del gruppo;
- Il gruppo ha scarse conoscenze riguardo le tecnologie *Docker_G* e *Kafka_G*.

7.7 Conclusioni

Nonostante il capitolato in oggetto abbia alcuni tratti in comune con quello *Predire in Grafana* scelto, esso è stato scartato a causa del suo campo di applicazione troppo ampio, e per preferenze del gruppo riguardo le tecnologie utilizzate nel capitolato C4 §2.4.