



VRAM SOFTWARE

Progetto "Predire in Grafana"

Studio di Fattibilità

07 dicembre 2019

Versione	1.0.0
Approvazione	Vittorio Corrizzato
Redazione	Rebecca Schiavon Marco Rampazzo Massimo Toffoletto Marco Dalla Libera
Verifica	Alessandro Spreafico Vittorio Santagiuliana
Stato	Approvato
Uso	Interno
Destinato a	Zucchetti Prof. Tullio Vardanega Prof. Riccardo Cardin
Email di riferimento	vram.software@gmail.com

Descrizione

Studio di Fattibilità dei capitolati proposti e scelta del capitolato.

Registro delle modifiche

Versione	Data	Nominativo	Ruolo	Descrizione
1.0.0	10-12-2019	Vittorio Corrizzato	Responsabile di progetto	Approvazione finale del documento
0.6.0	09-12-2019	Alessandro Spreafico	Verificatore	Verifica dello studio di fattibilità del capitolato 6
0.5.0	08-12-2019	Vittorio Santiagiuliana	Verificatore	Verifica dello studio di fattibilità del capitolato 4
0.4.0	08-12-2019	Vittorio Santiagiuliana	Verificatore	Verifica dello studio di fattibilità del capitolato 2
0.3.0	07-12-2019	Alessandro Spreafico	Verificatore	Verifica dello studio di fattibilità del capitolato 3
0.2.0	07-12-2019	Alessandro Spreafico	Verificatore	Verifica dello studio di fattibilità del capitolato 5
0.1.0	07-12-2019	Vittorio Santiagiuliana	Verificatore	Verifica dello studio di fattibilità del capitolato 1
0.0.9	09-12-2019	Marco Dalla Libera e Alessandro Spreafico	Analista e Verificatore	Correzione e modifica struttura documento
0.0.8	06-12-2019	Marco Dalla Libera	Analista	Stesura dello studio di fattibilità del capitolato 6
0.0.7	05-12-2019	Rebecca SchiavonS	Analista	Stesura dello studio di fattibilità del capitolato 5
0.0.6	05-12-2019	Marco Rampazzo	Analista	Stesura dello studio di fattibilità del capitolato 3
0.0.5	05-12-2019	Rebecca Schiavon	Analista	Stesura dello studio di fattibilità del capitolato 1

0.0.4	04-12-2019	Massimo Toffoletto	Analista	Stesura dello studio di fattibilità del capitolato 4
0.0.3	04-12-2019	Massimo Toffoletto	Analista	Stesura dello studio di fattibilità del capitolato 2
0.0.2	03-12-2019	Marco Rampazzo	Analista	Definita struttura interna del documento
0.0.1	03-12-2019	Marco Dalla Libera	Analista	Creato documento \LaTeX con il titolo

Indice

1	Introduzione	1
1.1	Scopo del documento	1
1.2	Glossario	1
1.3	Riferimenti	1
1.3.1	Normativi	1
1.3.2	Informativi	1
2	Capitolato 1 - Autonomous Highlights Platform	2
2.1	Informazioni generali	2
2.2	Descrizione	2
2.3	Obiettivi del progetto	2
2.4	Requisiti di progetto	2
2.5	Tecnologie interessate	3
2.6	Aspetti positivi	3
2.7	Criticità e fattori di rischio	4
2.8	Conclusioni	4
3	Capitolato 2 - <i>Etherless</i>	5
3.1	Informazioni generali	5
3.2	Descrizione	5
3.3	Obiettivi di progetto	5
3.4	Requisiti di progetto	5
3.5	Tecnologie interessate	6
3.6	Aspetti positivi	7
3.7	Criticità e fattori di rischio	7
3.8	Conclusioni	7
4	Capitolato 3 - NaturalAPI	8
4.1	Informazioni generali	8
4.2	Descrizione	8
4.3	Obiettivi di progetto	8
4.4	Requisiti di progetto	8
4.5	Tecnologie interessate	9
4.6	Aspetti positivi	9
4.7	Criticità e fattori di rischio	10
4.8	Conclusione	10

5	Capitolato 4 - Predire in Grafana	11
5.1	Informazioni generali	11
5.2	Descrizione	11
5.3	Obiettivi di progetto	11
5.4	Requisiti	11
5.5	Tecnologie interessate	12
5.6	Aspetti positivi	12
5.7	Criticità e fattori di rischio	13
5.8	Conclusione	13
6	Capitolato 5 - Stalker	14
6.1	Informazioni	14
6.2	Descrizione	14
6.3	Obiettivi	14
6.4	Requisiti	14
6.5	Tecnologie interessate	15
6.6	Aspetti positivi	16
6.7	Criticità e fattori di rischio	16
6.8	Conclusioni	16
7	Capitolato 6 - ThiReMa	17
7.1	Informazioni generali	17
7.2	Descrizione	17
7.3	Obiettivi del progetto	17
7.4	Requisiti di progetto	17
7.5	Tecnologie interessate	18
7.6	Aspetti positivi	19
7.7	Criticità e fattori di rischio	19
7.8	Conclusioni	19

1 Introduzione

1.1 Scopo del documento

Nel seguente documento verranno analizzati i capitolati_G presentati nell'ambito del progetto di Ingegneria del Software. Saranno esposte le ragioni che hanno portato il gruppo alla scelta del capitolato 4 (*Predire in Grafana*) e alla conseguente esclusione delle altre proposte.

1.2 Glossario

Questo documento verrà corredato da un *Glossario v.1.0.0* dove saranno illustrati i termini tecnici o altamente specifici per evitare ambiguità in essi. Le voci interessate saranno identificate da una 'G' a pedice.

1.3 Riferimenti

1.3.1 Normativi

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

1.3.2 Informativi

- **Capitolato_G d'appalto C1 - Autonomous Highlights 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:** <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 1 - Autonomous Highlights Platform

2.1 Informazioni generali

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

2.2 Descrizione

Autonomous Highlights Platform prevede lo sviluppo di una piattaforma web capace di ricevere in input dei video di eventi sportivi e fornire in output un video contenente i momenti salienti.

2.3 Obiettivi del progetto

Il progetto prevede la creazione di un software che riceve in input dei video di eventi sportivi come una partita di calcio, Formula1, Motogp o altri sport e riesca a creare in autonomia un video di massimo 5 minuti con i soli suoi momenti salienti (highlights) da fornire in output. A questo fine la piattaforma dovrà essere dotata di un modello di machine learning_G in grado di identificare ogni momento importante dell'evento. In particolare il flusso di generazione del video sarà così organizzato:

- Caricamento del video dell'evento desiderato;
- Individuazione degli highlights;
- Estrazione delle parti individuate dal video originario;
- Generazione del video dei momenti salienti.

2.4 Requisiti di progetto

Al fine del corretto svolgimento del progetto sarà necessario rispettare i seguenti vincoli forniti dall'azienda proponente:

- Utilizzo di Sage Maker_G per la costruzione del modello di intelligenza artificiale_G;
- Strutturazione di un'architettura a multiservizi_G;

- Permettere il caricamento dei video tramite riga di comando;
- Creazione di un'interfaccia web per l'analisi e il controllo dello stato di elaborazione del video.

2.5 Tecnologie interessate

Le tecnologie previste per la realizzazione del capitolato sono:

- **Amazon web services_G**, in particolare:
 - **AWS Elastic Container Service_G** come servizio per la gestione dei contenitori ad alte prestazioni;
 - **AWS Dynamo DB_G**, database non relazionale in NoSQL_G per il supporto e la gestione dei tag;
 - **AWS Elastic Transcoder_G** per la conversione e la rielaborazione dei video;
 - **AWS Sage Maker_G** per la creazione di un modello per il riconoscimento degli eventi salienti se si sceglie di implementare un addestramento supervisionato dell'intelligenza artificiale;
 - **AWS Rekognition_G** da usare nel caso si scelga, invece, l'apprendimento non supervisionato e quindi l'utilizzo di modelli già esistenti.
- **NodeJS_G** per lo sviluppo di API Restful JSON_G al fine di garantire una scalabilità ottimale;
- **Python_G** per lo sviluppo delle componenti necessarie di machine learning;
- **HTML5_G**, **CSS3_G** e **Javascript_G** per lo sviluppo dell'interfaccia web.

2.6 Aspetti positivi

- Le tecnologie proposte risultano innovative e utili per la loro larga diffusione nel mondo del lavoro;
- La documentazione delle tecnologie reperibile è ben approfondita;
- Le specifiche del capitolato sono state fornite in modo chiaro e preciso.

2.7 Criticità e fattori di rischio

- La maggior parte delle tecnologie proposte, seppure molto interessanti, non è prevista dal nostro corso di laurea. Dunque lo svolgimento di questo capitolato richiederebbe quindi un numero di ore di apprendimento difficilmente quantificabile;
- Si è ritenuto non semplice e dispendioso in termini di tempo il reperimento e l'analisi dei video per l'apprendimento della macchina.

2.8 Conclusioni

Lo scopo del capitolato è risultato accattivante, tuttavia la quantità di nuove tecnologie da apprendere ha demotivato il gruppo a scegliere questo progetto.

3 Capitolato 2 - *Etherless*

3.1 Informazioni generali

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

3.2 Descrizione

Etherless è una *Cloud Application Platform_G* che permette agli sviluppatori che la utilizzano di caricare nel cloud_G delle funzioni *JavaScript_G*. Tali procedure possono poi essere acquistate da terzi tramite l'impiego della criptovaluta_G *Ethereum_G* implementata grazie alla tecnologia della *blockchain_G*.

3.3 Obiettivi di progetto

Il progetto si propone di aiutare gli sviluppatori fornendo:

- un framework_G *Serverless_G* che gestisca il costo computazionale della funzione;
- un servizio di *Smart Contracts_G* che sovrintenda il processo di pagamento.

Quest'ultima tecnologia sarà di supporto anche per l'acquirente in quanto assicurerà il completamento della transazione solo a lavoro verificato e completato.

3.4 Requisiti di progetto

È richiesto di dividere lo sviluppo in tre fasi:

- **Local:** utilizzo dell'applicativo in ambiente locale, in cui può essere utilizzato *Ethereum testrpc_G* di *Truffle* per l'emulazione_G della blockchain_G;
- **Test:** utilizzo dell'applicativo in ambiente di testing, in cui può essere usata la soluzione proposta al punto precedente;
- **Staging:** utilizzo dell'applicativo in un ambiente pubblico, in tal caso si potrà usufruire di *Ropsten Ethereum_G* come rete di testing.

Inoltre è obbligatorio separare il software in tre parti:

- **etherless-cli**: modulo attraverso il quale lo sviluppatore interagisce con Etherless;
- **etherless-smart**: modulo per l'interazione tra *etherless-cli* e la parte server;
- **etherless-server**: modulo che ascolta gli eventi emessi da *etherless-smart* per attivare le rispettive funzioni lambda_G .

Ognuna di queste operazioni dovrà poi essere caricata e versionata tramite GitHub_G o GitLab_G .

3.5 Tecnologie interessate

Per l'implementazione delle varie funzionalità vengono date dall'azienda delle linee guida sulle tecnologie da utilizzare:

- **Typescript 3.6_G**: linguaggio di programmazione da impiegare, tramite l'approccio Promise_G o async-await_G , nello sviluppo della piattaforma *Etherless*;
- **Solidity_G**: linguaggio per la creazione e la gestione degli *Smart Contracts_G*;
- **AWS Lambda_G**: piattaforma computazione serverless fornita da *Amazon* per la coordinazione degli eventi;
- **Serverless Framework_G**: framework_G Web per la creazione di applicazioni su *AWS Lambda_G*;
- **typescript-eslint_G**: strumento di analisi statica_G del codice *TypeScript_G*.

AWS Lambda_G può essere corredata con altri componenti quali: *AWS API Gateway_G* (per eventi HTTP_G), *AWS DynamoDB_G* (database non relazionale_G) o *AWS S3_G* (servizio di memorizzazione). Tutto questo poi può essere gestito tramite *AWS CloudFormation_G* (piattaforma per l'organizzazione delle risorse AWS).

3.6 Aspetti positivi

- Ethereum_G e la tecnologia della blockchain_G più in generale sono tematiche molto attuali e innovative che suscitano interesse tra i componenti del gruppo anche per le possibili applicazioni future;
- L'azienda ha avallato delle richieste chiare e precise, aspetto valutato positivamente dal gruppo.

3.7 Criticità e fattori di rischio

- L'azienda ha sede in Olanda, il rapporto con i proponenti potrebbe essere quindi meno efficace;
- Seppur siano proposte tematiche interessanti il gruppo ha presentato dei dubbi riguardanti la difficoltà e le tempistiche per l'approfondimento delle tecnologie presentate.

3.8 Conclusioni

Il capitolato è apparso stimolante per quanto riguarda le tecnologie utilizzate e convincente nella sua esposizione. Tuttavia la distanza geografica e la mole di lavoro prevista sono risultate un ostacolo nell'effettiva realizzazione del software. Per questo motivi il gruppo ha deciso di vertere la sua scelta su un altro progetto.

4 Capitolato 3 - NaturalAPI

4.1 Informazioni generali

- Nome: NaturalAPI;
- Proponente: Teal Blue;
- Committente: Prof. Tullio Vardanega e Prof. Riccardo Cardin;

4.2 Descrizione

L'obiettivo di questo capitolato è quello di far parlare a tutti gli stakeholder_G un linguaggio comune, in modo da velocizzare ed evitare confusione durante la progettazione. Teal Blue vuole utilizzare le specifiche e i requisiti di un software, scritti in linguaggio naturale (Inglese, Italiano, ecc.), per generare API_G.

4.3 Obiettivi di progetto

Il prodotto da sviluppare è **NaturalAPI** un toolkit_G che dovrà generare API_G complete e test automatici a partire da un linguaggio naturale. Il workflow_G di questo prodotto consiste nel ricevere in input del testo in linguaggio naturale. Da questo testo verrà generato un BDL_G (Business Domain Language). Dal BDL_G dovrà essere sviluppato un BAL_G (Business Application Language) che, in seguito, verrà convertito in una API_G.

4.4 Requisiti di progetto

Per la realizzazione di questo prodotto è richiesto lo sviluppo di tre PoC_G o feature_G.

1. **NaturalAPI Discover:** *estrattore di BDL_G*. L'estrattore *Discover* ha lo scopo di estrarre entità, processi e combinazioni tra essi da un documento testuale di business;
2. **NaturalAPI Design:** *parser_G di scenari e caratteristiche*; Questa PoC_G dovrà creare una BAL_G API_G in tempo reale a partire dai documenti di Gherkin_G e da un BDL_G;
3. **NaturalAPI Develop:** *esportatore di linguaggio*. Questa feature_G dovrà convertire un BAL_G in casi di test e API_G nel linguaggio di programmazione scelto, supportando la creazione e l'aggiornamento di nuove repository_G.

Ogni $feature_G$ sopraindicata dovrà essere accessibile attraverso almeno due dei seguenti modi: interfaccia da linea di comando, GUI_G minimale o un'interfaccia web $REST_G$. La parte logica del prodotto finale dovrà invece essere esportata in una delle seguenti modalità: come una libreria, come parte di un eseguibile o come un processo indipendente locale o remoto. Per quanto riguarda l'input e output bisognerà sottostare allo standard UTF-8_G e indentatura tipo UNIX_G.

4.5 Tecnologie interessate

- NLP_G o Natural Language Processing cioè un trattamento informatico del linguaggio naturale, che si occupa della realizzazione di sistemi in grado di comprendere il linguaggio naturale;
- $Dependency\ Parser_G$ cioè un parser che si occupa di analizzare la struttura grammaticale di una frase in linguaggio naturale per identificare le relazioni tra parole chiavi e parole che le modificano;
- BDD_G Behaviour Driven Development che è un processo di sviluppo software agile che ha, alla sua base, una continua comunicazione tra tutti gli $stakeholder_G$ di un progetto informatico;
- $Hiptest_G$ e $Cucumber_G$ che sono degli strumenti software che supportano il processo BDD_G ; il primo serve per eseguire test automatici, mentre il secondo legge le specifiche software in linguaggio naturale, scritte con alcune regole di sintassi ($Gherkin_G$), e controlla che il software rispetti i requisiti;
- Generazione API_G e DLS_G utilizzando:
 - $OpenAPI_G$: uno standard per descrivere API_G ;
 - $Swagger_G$: un $framework_G$ che aiuta a sviluppare servizi web $REST_G$;
 - OWL_G : un linguaggio web semantico.
- Un qualsiasi $framework_G$ a scelta come Qt_G , $React_G$, ecc.

4.6 Aspetti positivi

- Il proponente non impone vincoli sui linguaggi di programmazione da utilizzare, questo permette al gruppo di sviluppare un prodotto con un linguaggio o un $framework_G$ considerato più interessante;

- I requisiti, le tecnologie e il modo di esportare il prodotto finale sono spiegati in modo preciso e esauriente;
- L'azienda Teal Blue si dimostra disponibile a incontri ed a una comunicazione aperta con il fornitore;

4.7 Criticità e fattori di rischio

- Questo capitolato non ha suscitato molto interesse nel gruppo a causa dell'eccessiva astrattezza del capitolato e per il fatto che i concetti su cui prepararsi non erano, per il gruppo VRAM Software, stimolanti;
- Le tecnologie da imparare sono molte ed è arduo quantificare il tempo necessario per raggiungere una preparazione sufficiente per gestire in modo produttivo le tecnologie elencate.

4.8 Conclusione

Il gruppo ha trovato interessante l'idea di Teal Blue di rendere la comunicazione tra tutti stakeholder_G chiara e veloce, tuttavia abbiamo deciso di orientarci verso un progetto meno astratto.

5 Capitolato 4 - Predire in Grafana

5.1 Informazioni generali

- Nome: Predire in Grafana_G: Monitoraggio predittivo per DevOps_G;
- Proponente: Zucchetti;
- Committente: Prof. Tullio Vardanega e Prof. Riccardo Cardin;

5.2 Descrizione

Con questo capitolato, Zucchetti mira a realizzare due plug-in_G per lo strumento di monitoraggio Grafana_G. Essi devono applicare gli algoritmi di Support Vector Machine(SVM)_G e Regressione Lineare(RL)_G al flusso di dati al fine di prevedere livelli al di sopra di una certa soglia e quindi generare un allarme, oppure permettere agli operatori del servizio Cloud_G di generare segnalazioni dei punti critici che l'erogazione mette in evidenza. In tal modo la linea di produzione del software può intervenire con cognizione di causa sul sistema.

5.3 Obiettivi di progetto

Il sistema richiede di implementare due plug-in_G di Grafana_G scritti in linguaggio Javascript_G che eseguiranno i calcoli(Support Vector Machine SVM o Regressione Lineare RL) letti da un file json_G e produrranno valori tali da essere aggiunti al flusso di monitoraggio.

5.4 Requisiti

- Produrre un file json_G dai dati di addestramento con i parametri per le previsioni attraverso SVM_G per le classificazioni o RL_G;
- Leggere la definizione del predittore_G dal file in formato json_G;
- Associare i predittori_G letti dal file json_G al flusso di dati presente in Grafana_G;
- Applicare la previsione e fornire i nuovi dati ottenuti dalla previsione al sistema di Grafana_G;
- Rendere disponibili i dati al sistema di creazione di grafici e dashboard_G per la loro visualizzazione;

Opzionalmente è richiesto di implementare le seguenti funzionalità:

- Offrire la possibilità di definire degli "alert"_G in base a livelli di soglia raggiunti dai nodi collegati alle previsioni;
- Fornire i dati che definiscono bontà dei modelli di previsione utilizzati;
- Offrire la possibilità di applicare delle trasformazioni alle misure lette dal campo per ottenere delle regressioni_G esponenziali o logaritmiche;
- Offrire la possibilità di addestrare SVM_G o RL_G direttamente in Grafana_G;
- Implementare dei meccanismi di apprendimento di flusso, per poter disporre di sistemi di previsione che si adattano costantemente ai dati rilevati sul campo;
- Utilizzare altri metodi di previsione che possano dare benefici alla previsione dei dati, tra cui la versione delle SVM_G adattate alla regressione_G o piccole reti neurali_G per la classificazione;

5.5 Tecnologie interessate

- **Javascript**: linguaggio di programmazione richiesto per costruire i plug-in_G di Grafana_G ed eseguire i calcoli tramite SVM_G e RL_G per fare le previsioni;
- **Grafana**: sistema di monitoraggio open-source_G estendibile con plug-in_G javascript_G;
- **Librerie Javascript per reti neurali**: librerie che permettono di implementare reti neurali_G per migliorare le previsioni, utili a soddisfare uno dei requisiti opzionali;

5.6 Aspetti positivi

- Svolgere questo capitolato permette di acquisire competenze in ambito di analisi e previsione dei dati, molto richieste in ambito lavorativo;
- La presentazione del problema è stata molto chiara e l'azienda è apparsa molto disponibile;
- Gli algoritmi di SVM_G e RL_G verranno forniti dall'azienda;
- L'azienda si rende disponibile a impartire delle basi di machine learning_G che non sono fornite nel nostro corso di laurea.

5.7 Criticità e fattori di rischio

- Gli algoritmi da utilizzare sono basati su concetti matematici e valutati di non facile apprendimento e che porteranno ad una fase di studio non indifferente.

5.8 Conclusione

Il gruppo ha trovato molto chiaro e definito il problema, inoltre l'azienda, forse anche perché di grandi dimensioni, si è mostrata molto disponibile a dedicare risorse per i gruppi che sceglieranno il suo capitolato.

6 Capitolato 5 - Stalker

6.1 Informazioni

- **Nome:** Stalker
- **Proponente:** Imola Informatica
- **Committente:** Prof. Tullio Vardanega

6.2 Descrizione

Il progetto Stalker nasce dalla necessità di tracciare il numero di persone presenti all'interno dei locali, richiesto dalle normative, utile in vari ambiti: dal controllare l'affluenza in luoghi di interesse o ad eventi, al monitorare le presenze del personale nel luogo di lavoro. La soluzione da loro proposta è un'un'applicazione mobile che invii la posizione dei dispositivi degli utenti ad un server, il quale si occuperà dell'estrazione e memorizzazione delle informazioni utili.

6.3 Obiettivi

La soluzione software proposta da Imola Informatica prevede lo sviluppo di un'applicazione mobile e un'infrastruttura server a supporto, controllabile da un'interfaccia web di amministrazione. Lo scenario di utilizzo prevede che tutti gli utenti siano muniti di uno smartphone e abbiano la possibilità di installare l'apposita applicazione; i luoghi da monitorare possono essere corredati con access point_G o beacon Bluetooth_G per facilitare il tracciamento, in particolare nei luoghi chiusi. Lo scopo del progetto è quello di monitorare il numero di presenze, in forma anonima o autenticata, in uno o più luoghi circoscritti e fornire statistiche a riguardo.

6.4 Requisiti

Un'applicazione mobile con i seguenti requisiti:

- Recupero lista organizzazioni;
- Possibilità di effettuare login tramite LDAP_G per organizzazioni che lo richiedono;
- Storico accessi;

- Visualizzazione in tempo reale della propria presenza all'interno di un luogo monitorato e cronometro del tempo trascorso al suo interno.

Un'interfaccia web per l'amministrazione con le seguenti funzionalità:

- Funzionalità di login;
- Creazione/modifica/eliminazione di organizzazioni;
- Aggiunta/modifica/rimozione di luoghi, definiti da coordinate geografiche;
- Configurazione collegamento server LDAP_G;
- Invio di notifiche push_G alle applicazioni per segnalare l'aggiornamento delle liste di organizzazioni e luoghi;
- Monitoraggio del numero di utenti presenti nei luoghi dell'organizzazione in tempo reale;
- Ricerca sugli accessi dei dipendenti e creazione di report sulla frequentazione dei luoghi;
- Gestione delle autorizzazioni per gli utenti dell'interfaccia web.

L'infrastruttura su cui si baserà l'applicazione dovrà sottostare ai seguenti vincoli:

- Comunicazioni solo all'entrata od uscita dall'area interessata;
- Cifratura di tutte le comunicazioni tra app e server;
- Architettura scalabile_G (verticalmente e orizzontalmente) e tollerante a picchi di traffico per il lato server;
- Test di carico che dimostrino il funzionamento in varie situazioni.

6.5 Tecnologie interessate

- Java_Go Swift_G(applicazione mobile);
- NodeJS_Go Python_G(back-end_G);
- Utilizzo di protocolli asincroni_G per le comunicazioni tra app e server;
- HTML5_G, CSS3_Ge Javascript_G(interfaccia web lato server);

- Utilizzo del pattern di Publisher/Subscriber_G, ovvero mittenti e destinatari dimessaggi dialogano attraverso un tramite(dispatcher);
- Utilizzo dell'IAAS Kubernetes o di un PAAS, Openshift o Rancher, per il rilascio delle componenti server;
- API REST_G esposte dal server, o gRPC_G in alternativa;
- GPS_G/sistemi ibridi di geolocalizzazione;
- LDAP_G(Lightweight Directory Access Protocol);
- Test unitari_G e d'integrazione per tutte le componenti applicative.

6.6 Aspetti positivi

- Lo sviluppo di applicazioni mobili è una conoscenza richiesta;
- Argomento interessante, anche grazie alla potenziale utilità nell'ambito della sicurezza;

6.7 Criticità e fattori di rischio

- I requisiti sono numerosi e richiedono conoscenze non banali;
- Lo sviluppo lato server potrebbe portare complicazioni, viste le numerose tecnologie coinvolte e i requisiti in scalabilità;
- La precisione ottenibile con le tecnologie attuali non è sufficiente per rendere l'applicazione usabile nella realtà;
- Nella presentazione è stato discusso il funzionamento del sistema GPS ma completamente tralasciata la funzionalità dei backend di geolocalizzazione presenti nei moderni sistemi mobile (ad esempio i Google Mobile Services per Android).

6.8 Conclusioni

Nonostante il nobile obiettivo, la difficile fattibilità di un prodotto concreto e la possibilità di complicazioni scoraggia il gruppo nella decisione d'intraprendere questo capitolato.

7 Capitolato 6 - ThiReMa

7.1 Informazioni generali

- Nome: ThiReMa;
- Proponente: Sanmarco Informatica;
- Committente: Prof. Tullio Vardanega e Prof. Riccardo Cardin;

7.2 Descrizione

Lo scopo di questo progetto è di creare un'applicazione web che permetta di valutare la correlazione fra dati operativi_G e fattori influenzanti_G in ambito IoT_G. I dati saranno ricavati da dispositivi IoT_G eterogenei tramite piattaforma Apache Kafka_G, quindi il software si occuperà di raccolta, storicizzazione, monitoraggio, analisi e presentazione dei dati. L'obiettivo finale del progetto è di rendere i processi aziendali intelligenti trasformando i dati in informazioni, così da permettere ad esempio la pianificazione di manutenzioni predittive calcolate su informazioni reali, in automatico.

7.3 Obiettivi del progetto

Realizzare il prodotto **ThiReMa**, una web application che raccoglierà dati da dispositivi IoT_G eterogenei tramite piattaforma Apache Kafka, li monitorerà e storicizzerà in un time series database_G e li presenterà elaborati all'utente tramite interfaccia web. Inoltre dovrà inviare notifiche tempestive agli operatori tramite un servizio basato su Telegram_G e permettere, sempre tramite tale servizio, di comandare i dispositivi IoT_G stessi.

7.4 Requisiti di progetto

- Lettura dati da gateway e sensori IoT_G tramite Apache Kafka_G;
- Salvataggio dei dati su time series database_G come TimescaleDB_G, ClickHouse_G tramite Apache Kafka_G;
- Creazione di una web application per presentare i dati raccolti;
- Gestione di utenti, ruoli e gruppi tramite web application;
- Possibilità di configurare da quali dispositivi i dati devono essere raccolti;

- Possibilità di mettere in correlazione i dati tra loro, scegliendo quali visualizzare;
- Possibilità di inviare comandi di configurazione o azione ai gateway tramite web application o Telegram_G;
- Deve prevedere dei meccanismi di avviso e notifica per ente.

Opzionalmente è richiesto di implementare le seguenti funzionalità:

- Possibilità di impostare una frequenza di interrogazione dei dispositivi ed invio dati;
- Possibilità di specificare delle funzioni di accumulo sui dati;
- Rappresentare i dati raccolti e le correlazioni tramite grafici;
- Possibilità di scegliere fra più algoritmi di correlazione;
- Sviluppare ed istanziare le feature_G tramite uso della tecnologia di containerizzazione_G Docker_G.

7.5 Tecnologie interessate

- Kafka_G : piattaforma di stream-processing_G distribuito, permette di raccogliere e monitorare flussi di record di dati come se fossero una coda di messaggi. Permette anche la storicizzazione e l'elaborazione di questi stream di dati;
- Time series database_G : sono database molto efficienti nella storicizzazione di dati IoT_G, in quanto occupano poca memoria pur mantenendo le informazioni basilari necessarie. A questi database è possibile affiancare altri database per contenere i metadati dei dispositivi IoT_G ed i dati degli utenti. Alcuni possibili database sono PostgreSQL_G, TimescaleDB_G, ClickHouse_G;
- JAVA_G : popolare linguaggio di programmazione interpretato ed orientato agli oggetti. È consigliato il suo uso per realizzare la business logic del programma tramite piattaforma Kafka_G;
- Bootstrap_G : popolare framework_G CSS utilizzato per realizzare l'interfaccia grafica di siti ed applicazioni web. È consigliato il suo uso per la realizzazione dell'interfaccia web dell'applicazione;

- Docker_G : è una tecnologia di containerizzazione_G che permette di eseguire in modo efficiente più applicativi software in ambienti dedicati ed isolati risiedenti in un'unica macchina fisica. Tali ambienti sono detti container e permettono di eseguire più servizi in un'unica macchina fisica rendendoli indipendenti gli uni dagli altri, è quindi possibile fermare un container lasciando gli altri regolarmente attivi.

7.6 Aspetti positivi

- Sviluppo di competenze nell'ambito IoT_G, molto richieste dal mondo del lavoro ed interessanti per i membri del gruppo;
- Sviluppo di conoscenze in ambito Big Data_G ed analisi dei dati, molto richieste attualmente dal mondo del lavoro e probabilmente anche in futuro;
- Utilizzo di Java, un linguaggio di programmazione molto popolare e previsto da un insegnamento del nostro corso di studi;
- Architettura del sistema ben definita e componenti ben chiare;
- L'ambito del capitolato è interessante per tutti i membri del gruppo.

7.7 Criticità e fattori di rischio

- Numero elevato di componenti da realizzare ed integrare che comportano di conseguenza molte tecnologie da apprendere;
- Probabile necessità di effettuare test in sede aziendale per avere accesso ai dispositivi fisici.

7.8 Conclusioni

Il gruppo ha trovato l'ambito di applicazione del problema interessante ed ha apprezzato la chiarezza con cui sono definite le varie parti del sistema. Proprio l'elevato numero di componenti da realizzare ha però fatto calare la volontà dei membri del gruppo a svolgere questo progetto.