

RedRoundRobin



STUDIO DI FATTIBILITÀ

THiReMA PROJECT

www.redroundrobin.site — redroundrobin.site@gmail.com

INFORMAZIONI SUL DOCUMENTO

Versione	1.0.0+b1.0
Uso	Interno
Stato	Approvato
Destinatari	Prof. Tullio Vardanega Prof. Riccardo Cardin Red Round Robin
Redattori	Alessandro Tommasin Giovanni Vidotto Mariano Sciacco
Verificatori	Nicolò Frison Fouad Mouad
Approvazione	Lorenzo Dei Negri

Registro delle modifiche

Versione	Descrizione	Data	Autore	Ruolo
1.0.0+b0.2	Approvazione documento	2019-12-17	Lorenzo Dei Negri	Responsabile
0.1.1+b0.1	Stesura e verifica sezione §4 e spostamento valutazioni C6 in Capitolato Scelto	2019-12-08	Mariano Sciacco e Nicolò Frison	Analista e verificatore
0.1.0+b0.1	Approvazione documento	2019-12-07	Mariano Sciacco	Responsabile
0.0.3+b0.0	Stesura e verifica valutazioni C4, C5, C6	2019-12-06	Giovanni Vidotto e Nicolò Frison	Analista e verificatore
0.0.2+b0.0	Stesura e verifica valutazioni C1, C2, C3	2019-12-03	Alessandro Tommasin e Fouad Mouad	Analista e verificatore
0.0.1+b0.0	Creazione documento, stesura e verifica sezione §1	2019-11-30	Alessandro Tommasin e Nicolò Frison	Analista e verificatore

Indice

1	Introduzione	5
1.1	Scopo del documento	5
1.2	Glossario e documenti esterni	5
1.3	Riferimenti	5
1.3.1	Normativi	5
1.3.2	Informativi	5
2	Valutazione capitolato scelto	6
2.1	Capitolato 6 - ThiReMa	6
2.1.1	Informazioni generali	6
2.1.2	Descrizione capitolato	6
2.1.3	Finalità del progetto	6
2.1.4	Tecnologie interessate	6
2.1.5	Aspetti positivi	7
2.1.6	Criticità	7
2.1.7	Conclusioni	7
3	Valutazione capitolati rimanenti	8
3.1	Capitolato C1 - Autonomous Highlights Platform	8
3.1.1	Informazioni generali	8
3.1.2	Descrizione	8
3.1.3	Finalità del progetto	8
3.1.4	Tecnologie	8
3.1.5	Linguaggi di programmazione	9
3.2	Vincoli del progetto	9
3.2.1	Aspetti positivi	9
3.2.2	Criticità	9
3.2.3	Conclusione	9

3.3	Capitolato C2 - Etherless	10
3.3.1	Informazioni generali	10
3.3.2	Descrizione	10
3.3.3	Finalità del progetto	10
3.3.4	Tecnologie	10
3.3.5	Linguaggi di programmazione	11
3.4	Vincoli del progetto	11
3.4.1	Aspetti positivi	12
3.4.2	Criticità	12
3.4.3	Conclusione	12
3.5	Capitolato C3 - NaturalAPI	13
3.5.1	Informazioni generali	13
3.5.2	Descrizione	13
3.5.3	Finalità del progetto	13
3.5.4	Tecnologie	13
3.5.5	Linguaggi di programmazione	14
3.6	Vincoli del progetto	14
3.6.1	Aspetti positivi	14
3.6.2	Criticità	14
3.6.3	Conclusione	15
3.7	Capitolato C4 - Predire in Grafana	16
3.7.1	Informazioni generali	16
3.7.2	Descrizione capitolato	16
3.7.3	Finalità del progetto	16
3.7.4	Tecnologie interessate	16
3.8	Vincoli del progetto	17
3.8.1	Aspetti positivi	17
3.8.2	Criticità	17

3.8.3	Conclusioni	17
3.9	Capitolato 5 - Stalker	18
3.9.1	Informazioni generali	18
3.9.2	Descrizione capitolato	18
3.9.3	Finalità del progetto	18
3.9.4	Tecnologie interessate	19
3.9.5	Vincoli del progetto	19
3.9.6	Aspetti positivi	20
3.9.7	Criticità	20
3.9.8	Conclusioni	20
4	Conclusioni e motivazioni del capitolato scelto	21

1 Introduzione

1.1 Scopo del documento

Il seguente documento ha l'obiettivo di descrivere brevemente ciò che ogni CAPITOLATO_G ha da proporre, elencando quelli che il nostro gruppo ha considerato come i loro aspetti più interessanti e le loro criticità.

1.2 Glossario e documenti esterni

Per evitare possibili ambiguità relative alle terminologie (che andranno indicate in MAIUSCOLETTO) utilizzate nei vari documenti, verranno utilizzate due simboli:

- una *D* al pedice per indicare il nome di un particolare documento;
- una *G* al pedice per indicare un termine che sarà presente nel GLOSSARIO v1.4.0+B1.0_D.

1.3 Riferimenti

1.3.1 Normativi

- NORME DI PROGETTO v2.3.0+B1.0_D.

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 Valutazione capitolato scelto

2.1 Capitolato 6 - ThiReMa

2.1.1 Informazioni generali

- **nome:** ThiReMa - Things Relationship Management;
- **proponente:** Sanmarco Informatica;
- **committente:** Prof. Tullio Vardanega e Prof. Riccardo Cardin.

2.1.2 Descrizione capitolato

Sviluppo di un software che, dopo aver ricevuto misurazioni da sensori eterogenei, li accumula in un database centralizzato. Questa applicazione viene poi completata da un servizio di `DISPATCHINGG` per inoltrare in modo tempestivo le informazioni utili per gestire le azioni urgenti. I dati messi a disposizione dal database centralizzato dovranno essere suddivisi in due macro-categorie: dati operativi e fattori influenzanti.

2.1.3 Finalità del progetto

Creare una `WEB-APPLICATIONG` che permetta di valutare la correlazione tra dati operativi (misure) e i fattori influenzanti. Tale applicazione si potrà focalizzare sulla definizione di uno o più algoritmi per la successiva analisi dei dati al fine di essere in grado di effettuare delle previsioni sull'andamento dei dati stessi ed offrire, ad esempio, dei servizi di `MANUTENZIONE PREDITTIVAG`. Per ogni tipologia di informazioni rilevate dovrà anche essere possibile assegnare il monitoraggio ad un particolare ente. Analizzando un determinato sensore, in base ai dati ricevuti, si può prevedere un deterioramento complessivo tale da generare una necessaria azione di manutenzione preventiva. La `WEB-APPLICATIONG` dovrà essere suddivisa in 3 macro-sezioni:

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

2.1.4 Tecnologie interessate

- **APACHE KAFKA_G:** il progetto mira a creare una piattaforma a bassa latenza ed alta velocità per la gestione di feed dati in tempo reale;
- **API PRODUCER_G, CONSUMER_G, CONNECT_G e STREAMS_G:** API_G consigliate per la produzione di componenti custom per KAFKA_G;
- **POSTGRESQL_G, TIMESCALEDB_G, ClickHouse:** implementazioni database suggerite per contenere i dati relativi alle misurazioni, agli utenti e le loro informazioni di autorizzazione;

- **DOCKER_G**: tecnologia di CONTAINERIZZAZIONE_G che consente la creazione e l'utilizzo di container. Nel progetto risulterebbe utile per l'istanziamento di tutti i componenti;
- **JAVA_G**: linguaggio di programmazione ad alto livello orientato agli oggetti e a tipizzazione statica;
- **BOOTSTRAP_G**: raccolta di strumenti OPEN SOURCE_G per la creazione di siti e applicazioni per il Web.

2.1.5 Aspetti positivi

- tecnologie già in parte conosciute dal gruppo con la possibilità di ampliarne le conoscenze;
- l'azienda mette a disposizione figure di diverso livello per rispondere alle varie esigenze del gruppo e per facilitare la creazione di ambienti di sviluppo e test;
- consistente set di dati su cui testare l'applicativo;
- interfacciarsi con l'hardware tramite API.

2.1.6 Criticità

- protocolli proprietari, la documentazione su di essi potrebbe essere limitata;
- il capitolato richiede un'analisi dei dati più avanzata rispetto agli altri.

2.1.7 Conclusioni

Il capitolato ha suscitato l'interesse del gruppo, dando la possibilità di ampliare la conoscenza di tecnologie già parzialmente note ed al contempo molto attuali, quali IoT_G e BIG DATA_G. C'è stato inoltre molto entusiasmo per la tipologia di WEB-APPLICATION_G da sviluppare. Nell'insieme questo capitolato è stato accolto con forte interesse da tutti i componenti del gruppo.

3 Valutazione capitolati rimanenti

3.1 Capitolato C1 - Autonomous Highlights Platform

3.1.1 Informazioni generali

- **nome:** Autonomous Highlights Platform;
- **proponente:** Zero12;
- **committente:** Prof. Tullio Vardanega e Prof. Riccardo Cardin.

3.1.2 Descrizione

L'obiettivo di questo CAPITOLATO_G è creare una PIATTAFORMA WEB_G che è capace di ricevere in input dei video di eventi sportivi, come una partita di calcio o di tennis, e che riesca a creare autonomamente un video di massimo 5 minuti contenente soltanto i suoi momenti chiave (HIGHLIGHTS_G).

3.1.3 Finalità del progetto

Il prodotto finale, ovvero la PIATTAFORMA WEB_G, dovrà essere dotata di un MODELLO DI MACHINE LEARNING_G in grado di identificare i momenti più importanti dell'evento sportivo che le è stato inviato. Un fattore importante è che andrà scelto uno sport sul quale focalizzare la propria attenzione e sul quale verrà addestrato il modello di apprendimento atto all'identificazione automatica dei momenti salienti di un evento dello sport. Il flusso di generazione del suddetto highlight dovrà avere la seguente struttura:

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

3.1.4 Tecnologie

Le tecnologie consigliate dall'azienda riguardano la tecnologia di AMAZON WEB SERVICES_G ed in particolare:

- **ELASTIC CONTAINER SERVICE_G o ELASTIC KUBERNETES SERVICE_G:** è un servizio che permette la gestione di contenitori, altamente dimensionabile e ad elevate prestazioni;
- **DYNAMODB_G:** è un database non relazionale per applicazioni che necessitano di prestazioni elevate su qualsiasi scala;
- **AWS TRANSCODE_G:** è un servizio di transcodifica di contenuti multimediali nel CLOUD_G;

- **SAGE MAKER_G**: è un servizio completamente gestito che permette a sviluppatori e DATA SCIENTIST_G di creare, addestrare e distribuire modelli di apprendimento automatico;
- **AWS REKOGNITION VIDEO_G**: è un servizio di analisi video basato su APPRENDIMENTO APPROFONDITO_G.

3.1.5 Linguaggi di programmazione

- **NODE.JS_G**: linguaggio ideale per sviluppare API RESTFUL_G JSON_G a supporto dell'applicativo;
- **PYTHON_G**: linguaggio ideale per lo sviluppo delle componenti di MACHINE LEARNING_G;
- **HTML5_G, CSS3_G, JAVASCRIPT_G**: linguaggi per la realizzazione dell'interfaccia web di gestione del flusso di lavoro, utilizzando un FRAMEWORK_G responsive come Twitter.

3.2 Vincoli del progetto

- utilizzo di SAGE MAKER_G;
- l'architettura dovrà essere basata su MICRO-SERVIZI_G, suddividendo il progetto in tante funzioni di base denominate servizi; questi ultimi dovranno essere indipendenti fra loro;
- caricamento dei video da elaborare tramite riga di comando;
- console web di analisi e controllo degli stati di elaborazione dei video.

3.2.1 Aspetti positivi

- per quanto riguarda le tecnologie interessate è presente molta documentazione;
- il tema principale del progetto è stato accolto con molto interesse dal gruppo, che si è dimostrato incuriosito ad approfondire l'argomento;
- Il proponente fornisce attività di formazione sulle principali tecnologie AWS e WIREFRAME_G dell'interfaccia della console web di analisi e controllo dello stato di elaborazione dei video.

3.2.2 Criticità

- il proponente non fornisce nessun DATA-SET_G per effettuare il TRAINING_G dell'algoritmo;
- buona parte del tempo verrebbe utilizzato per lavori ripetitivi, come la creazione di un data-set, e che non stimolerebbero il gruppo.

3.2.3 Conclusione

Dopo aver valutato gli aspetti positivi e le criticità, si è deciso che questo capitolato, nonostante tocchi delle tecnologie piuttosto interessanti, richiede, secondo il gruppo, un'eccessiva mole di lavoro ripetitivo.

3.3 Capitolato C2 - Etherless

3.3.1 Informazioni generali

- **nome:** Etherless;
- **proponente:** Red Babel;
- **committente:** Prof. Tullio Vardanega e Prof. Riccardo Cardin.

3.3.2 Descrizione

Etherless è una piattaforma $CLOUD_G$ che permette, agli sviluppatori di fare il $DEPLOY_G$ di funzioni $JAVASCRIPT_G$, mentre agli utenti finali di pagare per l'esecuzione delle suddette funzioni (si basa su $CAAS_G$, ovvero Computation-as-a-Service). Una parte di ciò che viene pagato dagli utenti verrà trattenuta dal piattaforma stessa come compenso per l'effettiva esecuzione della funzione.

3.3.3 Finalità del progetto

Il progetto finale si propone quindi di fornire un servizio di $CAAS_G$ utilizzando la seguente struttura:

- **Etherless-cli:** è il modulo con il quale gli sviluppatori interagiscono con Etherless. Deve supportare diversi comandi, ovvero: configurare il proprio $ACCOUNT_G$, eseguire il $DEPLOY_G$ delle funzioni, elencare le funzioni già presenti, eseguire una funzione e visualizzare i $LOGS_G$ riguardanti una specifica funzione;
- **Etherless-smart:** consiste in un set di $CONTRATTI_SMART_G$ che gestiscono la comunicazione e il trasferimento di denaro (detto ETH) tra Etherless-cli ed Etherless-server;
- **Etherless-server:** è il modulo che esegue le funzioni e che tramite Etherless-smart comunica con Etherless-cli. Nel momento in cui viene restituito il valore di una funzione o viene lanciata una eccezione, Etherless-server emetterà un evento nella $BLOCKCHAIN_G$ che verrà ricevuto dalla Etherless-cli che mostrerà infine il risultato all'utente.

3.3.4 Tecnologie

Il progetto si basa sull'unione di due tecnologie, ovvero $ETHEREUM_G$ e $SERVERLESS_G$; più precisamente utilizza:

- **Ethereum:** è una piattaforma che permette ai suoi utenti di scrivere applicazioni decentralizzate (dette $DAPPS_G$) che usano la tecnologia $BLOCKCHAIN_G$;
- **Ethereum Virtual Machine (EVM):** è una $MACCHINA_VIRTUALE_DECENTRALIZZATA_G$ che esegue $SCRIPT_G$ usando un $NETWORK_G$ internazionale di nodi pubblici;
- **Blockchain:** è una struttura dati condivisa e immutabile; tramite questa struttura è possibile tenere traccia dei pagamenti effettuati in Ethereum;

- **Smart Contract:** sono utilizzati per le interazioni tra attori e possono contenere denaro (ETHER_G o ETH), dati o una combinazione di entrambi;
- **Gas:** è il carburante che permette alla EVM_G di eseguire un programma;
- **MAINNET_G, ROPSTEN_G:** sono reti sulle quali vengono eseguiti i protocolli di Ethereum;
- **Eventi Ethereum:** sono una delle parti fondamentali di Ethereum in quanto possono essere: il valore di ritorno di uno smart contract, un innesco asincrono contenente dati oppure una forma di immagazzinamento più economica rispetto ad uno smart contract;
- **Architettura SERVERLESS_G:** è un metodo di creazione ed esecuzione di applicazioni e servizi che non richiede la gestione di un'infrastruttura (idea di BAAS_G, BACK END_G-as-a-Service);
- **AWS LAMBDA_G:** è un servizio che permette di eseguire codice in risposta ad eventi, senza effettuare il provisioning né gestire server;
- **Serverless Framework:** è un FRAMEWORK_G OPEN SOURCE_G che permette di sviluppare e distribuire applicazioni SERVERLESS_G;
- **CloudFormation:** è uno strumento che permette di gestire risorse e per fare il deploy di infrastrutture;
- **API GATEWAY_G, AWS DYNAMODB_G, AWS S3_G:** componenti che possono servire da supporto alle applicazioni SERVERLESS_G;
- **Truffle:** è un ambiente di sviluppo per Ethereum, consigliato per creare un web server locale per gestire la parte FRONT END_G del progetto;
- **Node Package Manager (NPM):** è un PACKAGE MANAGER_G consigliato per installare etherless-cli;
- **ESLint:** è uno strumento di ANALISI DEL CODICE STATICO_G per identificare schemi problematici nel codice JAVASCRIPT_G.

3.3.5 Linguaggi di programmazione

- **YAML_G, JSON_G:** sono due linguaggi che permettono di dare una struttura ai dati che poi verrà creata tramite CLOUD FORMATION_G;
- **SOLIDITY_G:** è un linguaggio tipato staticamente che supporta l'ereditarietà, librerie e tipi definiti dall'utente; viene utilizzato per scrivere smart contracts;
- **JAVASCRIPT_G:** linguaggio utilizzato per svolgere differenti compiti all'interno del progetto;
- **TYPESCRIPT_G:** linguaggio basato su JAVASCRIPT_G che permette di definire la tipologia dei dati. Ne è consigliata la versione 3.6.

3.4 Vincoli del progetto

- Etherless-cli deve poter essere installato con il comando bash: `npm install -g etherless-cli;`

- dopo essere stato installato uno sviluppatore deve essere in grado di eseguire diversi comandi associati ai seguenti compiti: creare o entrare in un `ACCOUNTG` `ETHEREUMG`, rilasciare una funzione, eseguire una funzione già presente, eliminare una funzione caricata;
- deve essere possibile fare degli upgrade agli smart contracts;
- Etherless deve essere sviluppato usando `TYPESCRIPTG` 3.6 usando un `APPROCCIO PROMISE/ASYNC-AWAITG`;
- deve essere usato `typescript-eslint` insieme a `ESLint` durante la fase di sviluppo;
- `Etherless-server` deve essere implementato usando `Serverless Framework`.

3.4.1 Aspetti positivi

- dato l'interesse riscosso dalla tecnologia `BLOCKCHAING`, questo capitolato potrebbe essere un notevole arricchimento del bagaglio di conoscenze del gruppo.

3.4.2 Criticità

- l'azienda ha sede all'estero e potrebbe fornire un supporto inferiore rispetto alle aziende che si trovano nel territorio nazionale;
- il capitolato dopo un'attenta analisi non ha riscosso molto interesse.

3.4.3 Conclusione

Dopo aver valutato attentamente il capitolato, il gruppo ha deciso di non considerarlo come prima scelta.

3.5 Capitolato C3 - NaturalAPI

3.5.1 Informazioni generali

- **nome:** NaturalAPI;
- **proponente:** teal.blue;
- **committente:** Prof. Tullio Vardanega e Prof. Riccardo Cardin.

3.5.2 Descrizione

Il progetto NaturalAPI nasce dall'idea che ogni persona in base al contesto in cui si trova nel proprio lavoro sviluppa ed utilizza un linguaggio specifico, utilizzando termini diversi per descrivere gli stessi problemi e/o soluzioni. Tuttavia le soluzioni correnti usano uno strato di collegamento tra linguaggio naturale (inglese, italiano, etc.) e il codice stesso, strato che viene creato in maniera arbitraria da chi è il responsabile della progettazione delle API_G.

3.5.3 Finalità del progetto

Il progetto finale si propone dunque di fornire una (PROOF-OF-CONCEPT_G) serie di strumenti, chiamata NaturalAPI, che possano ridurre il divario tra specifiche di progetto (espresse in inglese) e le API stesse. I tre strumenti che andranno a comporre questo TOOLKIT_G saranno:

- **NaturalAPI Discover:** un estrattore del linguaggio specifico del dominio trattato, che estrarrà potenziali entità (nomi/oggetti), processi (azioni/verbi) o una combinazione di questi ultimi da documenti non strutturati (che possono appartenere a diverse tipologie come manuali, WIKI_G o documenti veri e propri) legati al dominio in questione. Prima della fase successiva gli STAKEHOLDER_G del progetto selezioneranno un sottoinsieme delle entità/processi prodotti dal NaturalAPI Discover. Questo strumento produrrà dei file con estensione .bdl (business domain language);
- **NaturalAPI Design:** responsabile della creazione di un'API specifica del dominio, utilizzando dei documenti GHERKING_G e dei documenti .bdl legati allo stesso dominio (prodotti dal NaturalAPI Discover); questo strumento produrrà dei file con estensione .bal (business application language);
- **NaturalAPI Develop:** sarà responsabile della conversione e dell'esportazione dei file .bal (prodotti dal NaturalAPI Design) in test automatici ed API in uno dei linguaggi di programmazione/Framework_G scelti, supportando sia la creazione di una nuova REPOSITORY_G che l'aggiornamento di una già esistente. Da notare che prima dell'esportazione finale deve essere possibile definire dei dettagli specifici legati al linguaggio scelto dallo sviluppatore delle API; questi dettagli andranno immagazzinati in file con estensione .pla (programming language adapter).

3.5.4 Tecnologie

Il progetto sfrutterà le seguenti tecnologie:

- **Gherkin:** è un formato per scrivere test in CUCUMBER_G, utilizzando delle KEYWORD_G (come Given, When, Then) molto simili al linguaggio naturale. Viene proposto per scrivere dei casi d'uso nella fase di creazione dei file .bal (di cui è responsabile il NaturalAPI Design);
- **OpenAPI Specification:** definisce una descrizione dell'interfaccia standard, indipendente dal linguaggio di programmazione usato, per REST API_G. Ne viene consigliato l'uso nella parte di generazione finale delle API, il cui compito è affidato al NaturalAPI Develop.

3.5.5 Linguaggi di programmazione

Non c'è alcun vincolo sul linguaggio di programmazione/Framework_G da utilizzare.

3.6 Vincoli del progetto

- ogni strumento facente parte di NaturalAPI deve poter essere accessibile almeno tramite due tra le seguenti interfacce:
 - interfaccia con linea di comando;
 - interfaccia grafica minimale;
 - interfaccia WEB REST_G;
- naturalAPI deve essere disponibile in almeno una tra le seguenti piattaforme desktop (Ubuntu, macOS, Windows o tramite browser);
- gli strati logici devono essere rilasciati in una delle seguenti modalità:
 - come una libreria (statica o dinamica);
 - come parte degli stessi eseguibili delle modalità di rilascio scelte;
 - come un processo/servizio indipendente (locale o remoto);
- le modalità di rilascio devono condividere gli stessi LAYERS LOGICI_G, che non devono avere nessuna dipendenza dalla modalità di rilascio;
- tutte le risorse in input/output devono seguire la codifica UTF-8_G e interruzioni di riga UNIX_G.

3.6.1 Aspetti positivi

- il capitolato affronta un problema molto importante, quale il confronto tra persone abituate ad esprimere gli stessi concetti in modi diversi, cercando di trovare una soluzione.

3.6.2 Criticità

- il capitolato ha riscosso scarso interesse nella maggior parte dei membri del gruppo;
- il progetto implicato dal capitolato non sembra particolarmente concreto.

3.6.3 Conclusione

Nonostante il capitolato C3 cerchi di risolvere un problema molto sentito, la mancanza di concretezza del progetto stesso e lo scarso interesse mostrato per quest'ultimo ha portato il gruppo ad orientarsi verso un'altra scelta.

3.7 Capitolato C4 - Predire in Grafana

3.7.1 Informazioni generali

- **nome:** Predire in Grafana - Monitoraggio predittivo per DevOps;
- **proponente:** Zucchetti SPA;
- **committente:** Prof. Tullio Vardanega e Prof. Riccardo Cardin.

3.7.2 Descrizione capitolato

La società Zucchetti, per eseguire il monitoraggio dei propri sistemi, ha scelto GRAFANA_G , un prodotto OPEN SOURCE_G estendibile tramite PLUG-IN_G . Il capitolato chiede la realizzazione di alcuni PLUG-IN_G , da integrare in GRAFANA_G , che effettuino delle previsioni sul flusso dei dati raccolti al fine di monitorare il corretto funzionamento del sistema e consigliare eventuali interventi alla linea di produzione del software tramite allarmi o segnalazioni.

3.7.3 Finalità del progetto

È richiesto lo sviluppo di due PLUG-IN_G per Grafana che siano in grado di predire un possibile stato dell'imminente futuro della macchina. Le predizioni devono essere effettuate tramite $\text{REGRESSIONE LINEARE}_G$ (RL) e/o $\text{SUPPORT VECTOR MACHINE}_G$ (SVM). La Regressione Lineare consente di analizzare dati uniformi ed è molto utile se i dati hanno un dominio continuo e una dispersione uniforme (OMOSCHEDASTICI_G) con andamento lineare. Per avere una predizione valida si deve guardare la dispersione del grafo dei residui, nel quale una dispersione uniforme è indice di una RL corretta ed attendibile. Gli SVM sono algoritmi di classificazione per dati con dominio discreto e sono scalabili su spazi di dati molto grandi. Essi consistono nell'effettuare trasformazioni dello spazio dati per trovare una predizione che sia in grado di separare i dati in varie classi. L'addestramento degli algoritmi di $\text{MACHINE LEARNING}_G$ viene effettuato in un ambiente separato ed isolato da GRAFANA_G tramite SUITE_G di generazione dati come JMETER_G che consente di creare dati compatibili. A fine addestramento i predittori necessitano di essere salvati in file JSON_G per poi essere associati al flusso dati di GRAFANA_G . Una volta ottenute le previsioni sui dati essi dovranno essere resi disponibili al sistema di creazione di grafici di Grafana ed essere visualizzati sulla DASHBOARD_G .

Come obiettivi opzionali viene richiesto di dare la possibilità di definire "alert" in base a soglie raggiunte dalle previsioni; di dare la possibilità di effettuare trasformazioni sulle misure in modo da ottenere regressioni non lineari (es. logaritmiche o esponenziali); di consentire l'apprendimento costante tramite i dati raccolti dall'agente JMX. Viene richiesto, inoltre, di fornire anche i dati di bontà dei predittori. Per gli SVM sono "Precision", che consiste nel rapporto tra i dati veri predetti veri e i dati falsi predetti veri (veri trovati falsi positivi); e "Recall", che è il rapporto tra i veri valutati veri ed i veri non valutati tali; invece per RL si fa riferimento al coefficiente " R^2 " che rappresenta il rapporto tra gli errori rispetto alla retta e gli errori rispetto alla media del codominio dei dati.

3.7.4 Tecnologie interessate

Il progetto sfrutterà le seguenti tecnologie:

- **SUPPORT VECTOR MACHINES_G (SVM)**: modello di classificazione basata su trasformazioni dello spazio dei dati;
- **REGRESSIONE LINEARE_G (RL)**: modello di predizione basata sulla differenza dei quadrati che genera una predizione lineare del tipo $y = ax + b$;
- **JMX_G**: agente installato sulle macchine che raccoglie i dati necessari alle predizioni;
- **JMETER_G**: SUITE_G di generazione di dati virtuali per l'addestramento;
- **JAVASCRIPT_G**: linguaggio non tipizzato orientato ad oggetti sia CLIENT SIDE_G (JAVASCRIPT_G, JQUERY_G) che SERVER SIDE_G (AJAX_G, NODE.JS_G).

3.8 Vincoli del progetto

- il plug-in per Grafana deve essere realizzato utilizzando il linguaggio JavaScript.

3.8.1 Aspetti positivi

- l'azienda, consapevole del fatto che gli algoritmi di MACHINE LEARNING_G non fanno parte del corso di studi della laurea triennale, è disponibile alla formazione ed alla fornitura di questi tipi di algoritmi;
- i requisiti obbligatori del capitolato sono basilari e danno la possibilità di ampliare notevolmente il progetto con i requisiti opzionali;
- uso di un linguaggio flessibile, all'avanguardia e richiesto sul mercato.

3.8.2 Criticità

- integrare un sistema già esistente con conseguente innalzamento delle ore di documentazione sul sistema, oltre che sulle tecnologie;
- le tecnologie di MACHINE LEARNING_G non sono ancora ben chiare al gruppo;
- necessita uno studio approfondito della documentazione di varie SUITE_G e programmi per poter avere un'idea più concreta del progetto.

3.8.3 Conclusioni

Il capitolato ha stimolato un discreto interesse nel gruppo dal momento che si usano tecnologie importanti e consente di acquisire conoscenze spendibili nel mondo reale. L'idea di integrare un sistema già esistente però ha convinto il gruppo ad orientarsi su un altro capitolato.

3.9 Capitolato 5 - Stalker

3.9.1 Informazioni generali

- **nome:** Stalker;
- **proponente:** Imola Informatica;
- **committente:** Prof. Tullio Vardanega e Prof. Riccardo Cardin.

3.9.2 Descrizione capitolato

Il proponente chiede la realizzazione di una MOBILE-APPLICATION_G al fine di poter tracciare, in forma anonima e non, il numero esatto di persone presenti all'interno di uno spazio fisico identificato da un insieme di coordinate geografiche.

3.9.3 Finalità del progetto

L'obiettivo è sviluppare un'applicazione in grado di segnalare, ad un server dedicato, l'ingresso e l'uscita dell'utilizzatore dalle aree d'interesse (basandosi sulla posizione attuale dell'utilizzatore del dispositivo) in due modalità, autenticata o anonima, a seconda delle esigenze. L'applicazione deve permettere le seguenti operazioni:

- recupero lista organizzazioni (refresh manuale e/o temporizzato);
- LOGIN_G nell'organizzazione con eventuale autenticazione;
- storico degli accessi;
- visualizzazione in tempo reale della propria presenza o meno all'interno di un luogo monitorato e cronometro del tempo trascorso al suo interno;
- predisposizione di un pulsante "anonimo" che permetta di risultare presente in maniera anonima all'interno dell'organizzazione.

Le comunicazioni tra applicazione cellulare e server dovranno avvenire solo nel momento d'ingresso ed uscita dai luoghi designati. Il rilevamento della posizione può essere effettuato in due modi distinti:

- **DEAD RECKONING_G:** dato un punto di partenza, la velocità, la direzione del movimento, il tempo trascorso e la distanza percorsa si può comprendere il punto di arrivo;
- **PROXIMITY SENSING_G:** la posizione del punto mobile è ricavata dalle coordinate di determinate stazioni che tracciano il segnale che viene trasmesso da esse (CELL ID_G). Ogni stazione ha un suo PATTERN_G di segnale.

In genere con il GPS_G si usa la trilaterazione, prendendo la posizione nota di due o più punti di riferimento e la distanza misurata tra il punto mobile ciascun punto di riferimento. La triangolazione permette di calcolare la posizione sulla base di angoli di arrivo (AOA) tra punto mobile e punti di

riferimento e la distanza stessa tra i punti di riferimento. Ovviamente una precisione perfetta é difficile da raggiungere, perciò l'obbiettivo é un'approssimazione abbastanza precisa e dimostrabile della posizione. Oltre all'ottimizzazione del rilevamento viene richiesto di limitare il consumo energetico che il sistema può utilizzare in modo da estendere la durata di eventuali batterie presenti nel sistema.

3.9.4 Tecnologie interessate

Per lo sviluppo del server BACK END_G sono consigliate le seguenti tecnologie:

- **JAVA_G**: linguaggio di programmazione ad alto livello orientato agli oggetti e a tipizzazione statica;
- **PYTHON_G**: linguaggio di programmazione ad alto livello orientato agli oggetti adatto, tra gli altri usi, a sviluppare applicazioni distribuite, SCRIPTING_G, computazione numerica e SYSTEM TESTING_G;
- **NODE.JS_G**: è una RUNTIME_G di JAVASCRIPT_G OPEN SOURCE_G multiplatforma orientato agli eventi per l'esecuzione di codice JAVASCRIPT_G;
- **CONTINUOUS INTEGRATION_G**: continua disponibilità del codice prodotto tra gli sviluppatori;
- **CONTINUOUS DELIVERY_G**: gestione degli artefatti creati in produzione con relativi versionamenti;
- **CONTINUOUS TESTING_G**: il codice sviluppato viene testato, tramite test unitari, prima di essere rilasciato, per fornire una base solida per lo sviluppo da parte di altri sviluppatori interni, e consente di capire cosa é funzionante e cosa no. In generale si vuole una copertura del 85-90%.

Saranno inoltre necessari:

- protocolli asincroni per le comunicazioni APP MOBILE-SERVER_G;
- **PATTERN DI PUBLISHER/SUBSCRIBER_G**: in questo PATTERN_G, mittenti e destinatari di messaggi dialogano attraverso un tramite, che può essere detto DISPATCHER_G o BROKER_G. Il mittente di un messaggio (detto PUBLISHER_G) non deve essere consapevole dell'identità dei destinatari (detti SUBSCRIBER_G); esso si limita a "pubblicare" il proprio messaggio al DISPATCHER_G. I destinatari si rivolgono a loro volta al DISPATCHER_G "abbonandosi" alla ricezione di messaggi. Il DISPATCHER_G quindi inoltra ogni messaggio inviato da un PUBLISHER_G a tutti i SUBSCRIBER_G interessati a quel messaggio;
- utilizzo dell'IAAS_G KUBERNETES_G o di un PAAS_G, OPENSIFT_G o RANCHER_G, per il rilascio delle componenti del server nonché per la gestione della scalabilità orizzontale.

3.9.5 Vincoli del progetto

Non sono stati espressi vincoli di sviluppo, ma è stata espressa l'esigenza di avere un rilevamento di posizione precisa e testata.

3.9.6 Aspetti positivi

- possibilità di ampliare il bagaglio di tecnologie conosciute per lo sviluppo di applicazioni mobile;
- il proponente non impone tecnologie specifiche per lo sviluppo del server o della UI_G.

3.9.7 Criticità

- progetto lungo da sviluppare;
- documentazione GPS complicata da interpretare;
- difficile determinare con massima precisione la posizione dell'utilizzatore.

3.9.8 Conclusioni

Il capitolato ha stimolato l'interesse del gruppo visto le tipologie di tecnologie che verranno utilizzate e per l'applicazione che l'azienda intende farne. Il proponente chiede di fare diversi test per ottenere una stima il più precisa possibile sull'utilizzatore, dimostrando i risultati ottenuti. Dopo un'accurata riflessione abbiamo constatato che la dimostrazione dei test poteva essere molto dispendiosa a livello di tempistiche. Nonostante l'interesse il gruppo ha deciso, dopo un conteggio delle preferenze, di non scegliere questo capitolato.

4 Conclusioni e motivazioni del capitolato scelto

Dopo una attenta analisi di tutti i seminari di approfondimento effettuati dai relativi proponenti, si è deciso di optare per un argomento che racchiudesse il maggiore grado di interesse di tutto il gruppo. Basandosi sulle proprie conoscenze, inoltre, si è cercato di determinare un capitolato adeguato che potesse produrre una applicazione concreta nella materia di interesse proposta, con l'obiettivo di raggiungere una buona qualità per il prodotto finale da consegnare.

Inizialmente, abbiamo preso in ampia considerazione il capitolato C4 (*Predire in Grafana*), poiché richiede una analisi dei dati reperiti dalla `SERVER FARMG` facendo uso di algoritmi di predizione e, soprattutto, si appoggia su un software di monitoraggio molto utilizzato e `OPEN SOURCEG`. Tuttavia, la poca competenza nella parte di `MACHINE LEARNINGG` e l'idea di dover sviluppare essenzialmente una piccola parte di un sistema molto più grande e complesso con un unico linguaggio di programmazione, ci ha fatto convergere verso un argomento correlato che riprende l'analisi dei dati, ma in modo più avvincente.

Infatti, si è voluto optare per il capitolato C6 (*ThiReMa*) che racchiude lo sviluppo di una web application in diretto contatto con il mondo `IoTG`, attraversando le più moderne tecnologie (quali `DOCKERG` e i database non relazionali) per giungere alla gestione remota di dispositivi fisici su larga scala.

L'idea che ha menzionato l'azienda proponente per questo capitolato, inoltre, ci ha particolarmente colpito: *nascondere la complessità e ricavare l'informazione dai dati (grezzi)*. Da questa idea si è stimolato il nostro interesse di poter approfondire queste pratiche di sviluppo per la gestione e interpretazione dei dati, tali per cui l'uso di un sistema automatizzato e ampiamente connesso permetterebbe di semplificare di gran lunga il lavoro di reperimento dell'informazione.

Chiaramente, la nostra sfida si baserà sull'apprendere il più possibile queste nuove tecnologie (come `KAFKAG`, `DOCKERG` e i `TIMESERIES DBG`) così da realizzare un prodotto valido che soddisfi le aspettative e i requisiti richiesti dal capitolato, nonché presentare tutta la documentazione aggiuntiva richiesta dal proponente prima e dopo lo sviluppo del software.

Concludendo, con la scelta questo capitolato si vuole ampliare il proprio bagaglio di conoscenza in questa materia, facendo luce sui modi di integrare le principali funzionalità del sistema, usando tecnologie moderne e di grande interesse per tutto il gruppo, senza trascurare gli aspetti formali del progetto.