



Studio di fattibilità

Gruppo TeamAFK - Progetto "Predire con Grafana"

gruppoafk15@gmail.com

Informazioni sul documento

Versione	1.0.0
Approvatore	Davide Zilio
Redattori	Alessandro Canesso Victor Dutca Fouad Farid
Verificatori	Simone Federico Bergamin Victor Dutca
Uso	Interno
Distribuzione	Prof. Vardanega Tullio Prof. Cardin Riccardo TeamAFK

Descrizione

Il documento consiste di valutazioni preventive su requisiti e complessità dei capitolati presi in visione. Queste hanno portato a scegliere una delle proposte di progetto disponibili.

Registro delle modifiche

Versione	Data	Descrizione	Nominativo	Ruolo
1.0.0	2020-03-29	Approvazione del documento	Davide Zilio	<i>Responsabile di Progetto</i>
0.2.0	2020-03-28	Seconda verifica documento	Victor Dutca	<i>Verificatore</i>
0.1.2	2020-03-28	Adattamento alle norme	Alessandro Canesso	<i>Analista</i>
0.1.1	2020-03-26	Correzioni §3 e §7	Fouad Farid	<i>Analista</i>
0.1.0	2020-03-26	Prima verifica documento	Simone Federico Bergamin	<i>Verificatore</i>
0.0.6	2020-03-25	Stesura capitolato 6	Victor Dutca	<i>Analista</i>
0.0.5	2020-03-25	Stesura capitolato 5	Fouad Farid	<i>Analista</i>
0.0.4	2020-03-25	Stesura capitolato 4	Alessandro Canesso	<i>Analista</i>
0.0.3	2020-03-24	Stesura capitolato 3	Victor Dutca	<i>Analista</i>
0.0.2	2020-03-24	Stesura capitolato 2	Fouad Farid	<i>Analista</i>
0.0.1	2020-03-24	Stesura §1 e capitolato 1	Davide Zilio	<i>Analista</i>

Indice

1	Introduzione	3
1.1	Scopo del documento	3
1.2	Glossario	3
1.3	Riferimenti	3
2	Capitolato C1 - Autonomous Highlights Platform	4
2.1	Descrizione generale	4
2.2	Obiettivi	4
2.3	Tecnologie utilizzate	4
2.4	Valutazione generale	4
3	Capitolato C2 - Etherless	5
3.1	Descrizione generale	5
3.2	Obiettivi	5
3.3	Tecnologie utilizzate	5
3.4	Valutazione generale	5
4	Capitolato C3 - Natural API	6
4.1	Descrizione generale	6
4.2	Obiettivi	6
4.3	Tecnologie utilizzate	6
4.4	Valutazione generale	7
5	Capitolato C4 - Predire in Grafana	8
5.1	Descrizione generale	8
5.2	Obiettivi	8
5.3	Tecnologie utilizzate	8
5.4	Valutazione generale	8
6	Capitolato C5 - Stalker	10
6.1	Descrizione generale	10
6.2	Obiettivi	10
6.3	Tecnologie utilizzate	10
6.4	Valutazione generale	10
7	Capitolato C6 - ThiReMa	12
7.1	Descrizione generale	12
7.2	Obiettivi	12
7.3	Tecnologie utilizzate	12
7.4	Valutazione generale	12

1 Introduzione

1.1 Scopo del documento

Lo scopo del documento è quello di indicare le motivazioni che hanno portato alla decisione del gruppo di intraprendere il progetto esposto nel capitolato C4. Per ciascun capitolato viene riportato lo studio di fattibilità e le valutazioni riguardo tale progetto.

1.2 Glossario

I termini utilizzati in questo documento potrebbero generare dubbi riguardo al loro significato, richiedendo pertanto una definizione al fine di evitare ambiguità. Tali termini vengono contrassegnati da una G maiuscola finale a pedice della parola. La loro spiegazione è riportata nel *Glossario_v1.0.0*.

1.3 Riferimenti

- Capitolato C1: <https://www.math.unipd.it/~tullio/IS-1/2019/Progetto/C1.pdf>;
- Capitolato C2: <https://www.math.unipd.it/~tullio/IS-1/2019/Progetto/C2.pdf>;
- Capitolato C3: <https://www.math.unipd.it/~tullio/IS-1/2019/Progetto/C3.pdf>;
- Capitolato C4: <https://www.math.unipd.it/~tullio/IS-1/2019/Progetto/C4.pdf>;
- Capitolato C5: <https://www.math.unipd.it/~tullio/IS-1/2019/Progetto/C5.pdf>;
- Capitolato C6: <https://www.math.unipd.it/~tullio/IS-1/2019/Progetto/C6.pdf>.

2 Capitolato C1 - Autonomous Highlights Platform

2.1 Descrizione generale

La Autonomous Highlights Platform è una piattaforma web con la capacità di creare autonomamente video dei momenti salienti di eventi sportivi, come calcio, F1 o altro.

2.2 Obiettivi

Lo scopo di questo capitolato è la realizzazione di una piattaforma web dotata di un modello di machine learning_G in grado di identificare autonomamente ogni momento importante di un dato evento. Il processo di creazione del video si suddivide in 4 step:

1. caricamento del video;
2. identificazione dei momenti salienti;
3. estrazione delle corrispondenti parti di video;
4. generazione del video di sintesi.

I dati vengono caricati nell'ECS_G, il quale orchestra la collaborazione dei servizi Transcoder_G ed S3_G; successivamente l'ECS si interfaccia con le seguenti tecnologie: Sage Maker_G, DynamoDB_G e Rekognition_G.

2.3 Tecnologie utilizzate

- Elastic Container Service: servizio di gestione di contenitori_G;
- DynamoDB: database_G per immagazzinare i dati;
- AWS Transcode: per convertire ed elaborare i video;
- Sage Maker: servizio di gestione dell'apprendimento automatico_G;
- AWS Rekognition video: servizio cloud_G di Deep Learning_G;
- NodeJS_G: per lo sviluppo di API_G a supporto dell'applicativo;
- Python_G: per lo sviluppo delle componenti di Machine Learning;
- HTML5_G, CSS3_G e Javascript_G: la piattaforma deve essere realizzata come sito web.

2.4 Valutazione generale

Il progetto è stato valutato positivamente per le tecnologie impiegate nella sua realizzazione, come ad esempio NodeJS, e la piattaforma Amazon Web Services_G, entrambe utili in molti ambiti lavorativi. Tuttavia si è scelto di scartare il capitolato in quanto la sua realizzazione richiede una conoscenza di uno spettro troppo ampio di tecnologie e servizi, che il team ha valutato come eccessivamente oneroso in termini di formazione.

3 Capitolato C2 - Etherless

3.1 Descrizione generale

Ethereum è una piattaforma che permette agli utenti di implementare in modo semplificato applicazioni decentralizzate, “Dapps”, che utilizzano la tecnologia blockchain_G.

3.2 Obiettivi

Il capitolato prevede di sviluppare una piattaforma per applicazioni cloud_G Etherless che permette agli sviluppatori di fornire funzioni JS nel cloud e permette ai clienti di pagare solamente per la loro esecuzione, sfruttando la smart contract technology_G di Ethereum_G. L'esecuzione delle funzioni avviene tramite un'interfaccia a linea di comando_G (Etherless-cli). La piattaforma è gestita e mantenuta dagli amministratori; gli utenti, infine, possono eseguire le funzioni fornite pagando una quota decisa dagli sviluppatori.

Le commissioni sono parzialmente trattenute dalla piattaforma come compensazione per il costo dell'esecuzione delle funzioni. A livello industriale una best practice è la suddivisione dello sviluppo software in ambienti differenti: locale, test, staging_G e production (opzionale per il progetto in questione).

Ogni singolo ambiente è capace di sostenere l'intero sistema indipendentemente dagli altri. Lo sviluppatore implementa le funzionalità e le esegue in locale.

Successivamente, verrà implementata una suite di test per verificarne l'efficacia_G. Questi test potranno essere eseguiti localmente oppure potranno far parte di un processo di integrazione continua. Le funzionalità verranno poi distribuite in un ambiente di staging in modo tale che altre persone possano utilizzarle.

Infine, il ciclo di rilascio determinerà quando verrà distribuito il prodotto finale richiesto.

3.3 Tecnologie utilizzate

Etherless è costruita integrando le seguenti tecnologie:

- Ethereum: si occupa dei pagamenti e dell'invocazione delle funzioni;
- Serverless: si occupa dell'esecuzione delle funzioni;
- Etherless-cli (command line interface): tramite tra sviluppatori e utenti;
- Node Package Manager (NPM_G);
- Typescript_G.

3.4 Valutazione generale

Nonostante la mole di lavoro proposta fosse stata considerata ragionevole, il gruppo ha dimostrato meno interesse verso le specifiche tecnologie da impiegare nella realizzazione di questo progetto rispetto ad altri capitoli.

Per tale motivo il capitolato non è stato preso in considerazione come prima scelta.

4 Capitolato C3 - Natural API

4.1 Descrizione generale

"What if I try to withdraw 20 Euros from the ATM could semi-automatically be converted to a function with a signature like withdraw (user, amount, source)?"

Data l'ambiguità del linguaggio naturale che spesso comporta fraintendimenti tra stakeholders_G, "Natural API" propone un collegamento univoco tra il linguaggio umano_G e linguaggio di programmazione_G, permettendo in futuro lo sviluppo di librerie_G più consistenti, prevedibili e manutenibili_G.

4.2 Obiettivi

L'obiettivo di questo progetto è fornire uno strumento Proof-of-Concept_G che riduca il divario (gap) tra specifiche di progetto e API_G, generando librerie complete e facilmente manutenibili e i relativi test di unità e di integrazione, ricoprendo i linguaggi di programmazione e i frameworks_G più popolari. NaturalAPI dovrebbe inoltre permettere agli sviluppatori di concentrarsi sull'implementazione dei metodi piuttosto che sulla modellazione dell'intero sistema, attraverso:

1. ricerca di combinazioni tra nomi e verbi (Gherkin);
2. normalizzazione e conversione di predicati in funzioni e metodi;
3. ricerca di argomenti di funzioni ricorrenti al fine di generare oggetti e proprietà.

Ogni Natural API deve essere accessibile in almeno due tra le seguenti modalità:

- interfaccia web REST_G ;
- un'interfaccia grafica minimale;
- una command line interface_G .

4.3 Tecnologie utilizzate

- Per la generazione di API e DLS:
 - Swagger: strumento per la generazione di codice basato in OpenAPI;
 - OWL v2: formato per la rappresentazione ontologica;
 - OpenAPI.
- Behaviour Driven Development (BDD_G):
 - Hiptest: è una piattaforma cloud di gestione dei test che supporta la continuous delivery_G ;
 - Gherkin: parser_G di linguaggio di Cucumber;
 - Cucumber: è uno strumento software che supporta il BDD.

4.4 Valutazione generale

L'idea è sembrata molto interessante e all'avanguardia, utile ai team di sviluppo per essere più efficienti, riducendo comunicazioni superflue o possibili fraintendimenti.

Nonostante ciò, il progetto è stato valutato come troppo ambizioso in quanto potrebbe comportare un probabile sforamento delle scadenze.

È stato perciò deciso di non considerarlo come prima scelta.

5 Capitolato C4 - Predire in Grafana

5.1 Descrizione generale

La fabbrica del software e gli operatori che erogano il servizio sono a stretto contatto tra loro e la costante collaborazione tra i due attori porta ad un notevole miglioramento della qualità. Perché la collaborazione sia efficace è necessario un pieno monitoraggio dei sistemi, in modo che gli erogatori del servizio possano segnalare alla fabbrica i punti in cui è possibile migliorare la procedura.

Per eseguire il monitoraggio dei propri sistemi la *Zucchetti SPA* ha scelto Grafana_G, un prodotto OpenSource, estendibile con plug-in_G in linguaggio JavaScript.

5.2 Obiettivi

Il sistema che il progetto richiede sono due plug-in di Grafana, scritti in linguaggio JavaScript, che leggeranno da un file Json_G la definizione dei calcoli da applicare (SVM_G o RL_G) e quindi permetteranno di associarli ad alcuni nodi della rete del flusso del monitoraggio. I plug-in quindi eseguiranno i calcoli previsti, producendo dei valori che potranno essere aggiunti al flusso del monitoraggio come se fossero stati rilevati dal campo.

Il software richiesto dovrà svolgere i seguenti compiti:

1. produrre un file json dai dati di addestramento con i parametri per le previsioni con Support Vector Machine (SVM) per le classificazioni o la Regressione Lineare;
2. leggere la definizione del predittore dal file in formato json;
3. associare i predittori letti dal file json al flusso di dati presente in Grafana;
4. applicare la previsione e fornire i nuovi dati ottenuti dalla previsione al sistema di Grafana;
5. rendere disponibili i dati al sistema di creazione di grafici e dashboard per la loro visualizzazione.

5.3 Tecnologie utilizzate

- Grafana: interfaccia per il monitoraggio delle applicazioni;
- JavaScript / Json;
- svmjs: libreria js per classificare i valori di addestramento;
- Regression-js: libreria js per attribuire un valore numerico come stima di un osservazione partendo da dei valori di riferimento;
- Orange Canvas: strumento per l'analisi dei dati.

5.4 Valutazione generale

Sin da subito il capitolato ha suscitato il nostro interesse, data la presenza di elementi statistici e matematici di previsione sui dati. Un punto a favore dell'offerta proposta sta nell'uso di

metodologie come la regressione lineare, in quanto quest'ultima ha dimostrato di essere uno degli strumenti più affidabili e collaudati per l'analisi statistica. L'approfondimento di tali tecnologie e la curiosità nata per la gestione del flusso di dati si sono rivelati di interesse comune tra i membri del gruppo.

E' stato quindi deciso di scegliere la proposta di progetto del capitolato proposto da Gregorio Piccoli - *Zucchetti SPA*.

6 Capitolato C5 - Stalker

6.1 Descrizione generale

La normativa vigente in materia di sicurezza regola la gestione delle presenze nei locali pubblici ed aperti al pubblico prevedendo una serie di precauzioni ed adempimenti volti a garantire uno sfollamento sicuro in caso di emergenze e bisogno.

In questo senso diventa necessario tracciare il numero di persone presenti all'interno dei locali.

6.2 Obiettivi

L'obiettivo del progetto è quello di sviluppare un'applicazione mobile_G OpenSource_G (Android_G o IOS_G) in grado di segnalare ad un server dedicato sia l'ingresso che l'uscita dell'utilizzatore dalle aree di interesse.

È prevista la presenza di due tipologie di attori: amministratore e utente, il quale può essere un utilizzatore anonimo (tramite un'apposita funzionalità) o un dipendente autenticato.

Nel server deve essere possibile gestire più organizzazioni, ed ognuna di esse deve poter gestire molteplici luoghi e definire se prevedere una tracciatura autenticata (LDAP) e l'eventuale procedura di autenticazione. L'applicazione deve permettere:

- il recupero della lista delle organizzazioni;
- il login con eventuale autenticazione;
- lo storico degli accessi;
- la visualizzazione in real-time_G della propria presenza o meno.

Il progetto inoltre necessita di test di carico e di sovraccarico, di copertura, report e documentazione sulle scelte implementative e progettuali e/o eventuali problemi e soluzioni proposte.

6.3 Tecnologie utilizzate

- Docker: container_G per l'istanziamento di tutti i componenti;;
- Github: sistema di versionamento_G ;
- NodeJS/Python/Java: per lo sviluppo del server back-end_G ;
- RESTful API/gRPC: frameworks per fornire le funzionalità di autenticazione e per utilizzare l'applicativo.

6.4 Valutazione generale

Il progetto risulta molto interessante in riferimento alle tecnologie da utilizzare, ritenute attuali e d'impatto sulla vita di tutti i giorni, toccando problematiche sensibili alla quotidianità. Rispetto ai capitoli analizzati precedentemente, è il primo servizio che richiede la realizzazione di un'applicazione mobile, il che garantisce un punto a favore nel considerare tale

progetto come un possibile candidato. Nonostante l'interesse mostrato, il gruppo ha dovuto scartarlo in quanto la disponibilità offerta dall'azienda proponente è esaurita.

7 Capitolato C6 - ThiReMa

7.1 Descrizione generale

La continua necessità delle aziende di avere soluzioni sempre più versatili dovuta dalla gestione di grandi quantità di dati richiede un software che indirizzi nel modo giusto le informazioni agli stakeholders.

7.2 Obiettivi

Il progetto richiede un software, che verrà realizzato attraverso una web-application, in grado di ricevere misurazioni da vari sensori (temperatura, tensione di corrente, etc) e di accumularli efficientemente e in maniera affidabile in un database centralizzato che contiene due macro-categorie: dati operativi e fattori influenzanti.

La web-application dovrà esser suddivisa in tre macro-sezioni:

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

Gli utenti avranno a disposizione un'interfaccia grafica che gli permetterà di seguire in tempo reale l'andamento dei sensori e visualizzare dati di interesse.

7.3 Tecnologie utilizzate

- Kafka: cluster_G distribuito per lo streaming_G di dati in tempo reale;
- PostgreSQL, TimescaleDB, ClickHouse: database_G suggeriti per la raccolta dei dati;
- BootStrap: per il front-end_G styling;
- Docker: container per l'istanziamento di tutti i componenti;
- Java: sviluppo della business logic_G.

7.4 Valutazione generale

Mettendo a confronto questo progetto con il quarto capitolato proposto da *Zucchetti SPA* - "*Predire in Grafana*", è stata notata una notevole somiglianza sullo scopo del progetto, incentrato sulla raccolta e la gestione del flusso dei dati.

Dopo averli esaminati e aver valutato aspetti positivi e negativi di entrambi, *ThiReMa* è stato preso in considerazione come possibile candidato.

Purtroppo però tale scelta non è stata possibile data la disponibilità terminata del proponente.