



Carbon12 – Predire in Grafana

# Studio di Fattibilità

## Informazioni sul documento

<b>Versione</b>	1.0.0
<b>Stato</b>	Approvato
<b>Data di creazione</b>	2019/11/29
<b>Data di approvazione</b>	2020/01/13
<b>Redazione</b>	Giacomo Callegari Manuel De Franceschi Francesco Gobbo Andrea Longo Alessandro Lovo Veronica Pederiva
<b>Verifica</b>	Nicolò Fassina
<b>Approvazione</b>	Giacomo Callegari
<b>Uso</b>	Interno
<b>Destinatari</b>	Carbon12 Prof. Tullio Vardanega Prof. Riccardo Cardin

## Scopo del documento

Studio di fattibilità dei sei capitolati proposti

# Indice

<b><u>1 INTRODUZIONE.....</u></b>	<b><u>1</u></b>
1.1 SCOPO DEL DOCUMENTO.....	1
1.2 GLOSSARIO .....	1
1.3 RIFERIMENTI .....	1
1.3.1 NORMATIVI.....	1
1.3.2 INFORMATIVI .....	1
<b><u>2 CAPITOLATO C1 - AUTONOMOUS HIGHLIGHTS PLATFORM.....</u></b>	<b><u>2</u></b>
2.1 INFORMAZIONI GENERALI .....	2
2.2 DESCRIZIONE .....	2
2.3 FINALITÀ DEL PROGETTO.....	2
2.4 TECNOLOGIE INTERESSATE.....	2
2.5 ASPETTI POSITIVI .....	3
2.6 RISCHI .....	3
2.7 CONCLUSIONI .....	3
<b><u>3 CAPITOLATO C2 - ETHERLESS .....</u></b>	<b><u>4</u></b>
3.1 INFORMAZIONI GENERALI.....	4
3.2 DESCRIZIONE .....	4
3.3 FINALITÀ DEL PROGETTO.....	4
3.4 TECNOLOGIE INTERESSATE.....	4
3.5 ASPETTI POSITIVI.....	5
3.6 RISCHI .....	5
3.7 CONCLUSIONI .....	5
<b><u>4 CAPITOLATO 3 - NATURALAPI.....</u></b>	<b><u>6</u></b>
4.1 INFORMAZIONI GENERALI.....	6
4.2 DESCRIZIONE .....	6
4.3 FINALITÀ DEL PROGETTO.....	6
4.4 TECNOLOGIE INTERESSATE.....	7
4.5 ASPETTI POSITIVI.....	7
4.6 RISCHI .....	7
4.7 CONCLUSIONI .....	7
<b><u>5 CAPITOLATO C4 - PREDIRE IN GRAFANA .....</u></b>	<b><u>8</u></b>
5.1 INFORMAZIONI GENERALI.....	8
5.2 DESCRIZIONE.....	8

<b>5.3 FINALITÀ DEL PROGETTO.....</b>	<b>8</b>
<b>5.4 TECNOLOGIE INTERESSATE.....</b>	<b>8</b>
<b>5.5 ASPETTI POSITIVI.....</b>	<b>9</b>
<b>5.6 RISCHI .....</b>	<b>9</b>
<b>5.7 CONCLUSIONI .....</b>	<b>9</b>
 <b><u>6 CAPITOLATO C5 - STALKER .....</u></b>	 <b><u>10</u></b>
<b>6.1 INFORMAZIONI GENERALI.....</b>	<b>10</b>
<b>6.2 DESCRIZIONE.....</b>	<b>10</b>
<b>6.3 FINALITÀ DEL PROGETTO.....</b>	<b>10</b>
<b>6.4 TECNOLOGIE INTERESSATE.....</b>	<b>10</b>
<b>6.5 ASPETTI POSITIVI.....</b>	<b>11</b>
<b>6.6 RISCHI .....</b>	<b>11</b>
<b>6.7 CONCLUSIONI .....</b>	<b>11</b>
 <b><u>7 CAPITOLATO C6 – THIREMA - THINGS RELATIONSHIP MANAGEMENT .....</u></b>	 <b><u>12</u></b>
<b>7.1 INFORMAZIONI GENERALI.....</b>	<b>12</b>
<b>7.2 DESCRIZIONE.....</b>	<b>12</b>
<b>7.3 FINALITÀ DEL PROGETTO.....</b>	<b>12</b>
<b>7.4 TECNOLOGIE INTERESSATE .....</b>	<b>13</b>
<b>7.5 ASPETTI POSITIVI .....</b>	<b>13</b>
<b>7.6 RISCHI .....</b>	<b>13</b>
<b>7.7 CONCLUSIONI .....</b>	<b>13</b>
 <b><u>REGISTRO DELLE MODIFICHE .....</u></b>	 <b><u>14</u></b>

# 1 Introduzione

## 1.1 Scopo del documento

Il presente documento è stato redatto per descrivere le motivazioni che hanno guidato il team di Carbon12 alla scelta di candidarsi per la realizzazione del capitolato C4 – Predire in Grafana escludendo di conseguenza gli altri capitolati proposti.

## 1.2 Glossario

In allegato al presente documento viene fornito il *Glossario interno v.1.0.0* nel quale vengono riportati e definiti in modo inequivocabile i termini qui citati che potrebbero essere mal interpretati dai membri del team. I termini riportati nel glossario presentano una GI a pedice.

## 1.3 Riferimenti

### 1.3.1 Normativi

- Norme di Progetto: *Norme di progetto v.1.0.0*

### 1.3.2 Informativi

- Capitolato C1 – Autonomous Highlights Platform  
<https://www.math.unipd.it/~tullio/IS-1/2019/Progetto/C1.pdf>
- Capitolato C2 – Etherless  
<https://www.math.unipd.it/~tullio/IS-1/2019/Progetto/C2.pdf>
- Capitolato C3 – NaturalAPI  
<https://www.math.unipd.it/~tullio/IS-1/2019/Progetto/C3.pdf>
- Capitolato C4 – Predire in Grafana  
<https://www.math.unipd.it/~tullio/IS-1/2019/Progetto/C4.pdf>
- Capitolato C5 – Stalker  
<https://www.math.unipd.it/~tullio/IS-1/2019/Progetto/C5.pdf>
- Capitolato C6 – ThiReMa  
<https://www.math.unipd.it/~tullio/IS-1/2019/Progetto/C6.pdf>

## 2 Capitolato C1 - 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 nasce dall'esigenza di automatizzare il processo di realizzazione degli highlights di un determinato evento sportivo, come ad esempio una partita di calcio o di basket, una gara di Formula1 o MotoGP. L'idea alla base del progetto è quella di utilizzare i moderni modelli di MACHINE LEARNING per ricostruire un video contenente solo i momenti salienti di una competizione sfruttando la potenza computazionale dei calcolatori e riducendo così il lavoro che dovrebbe essere fatto da un operatore umano.

### 2.3 Finalità del progetto

L'obiettivo è di concentrarsi su una sola tipologia di sport in modo da realizzare una piattaforma web dotata di un modello di MACHINE LEARNING che venga addestrato a riconoscere i momenti più significativi dello sport scelto. Il risultato della computazione dovrà essere un video di massimo 5 minuti che contenga gli highlights della gara analizzata.

La struttura di funzionamento della piattaforma si può riassumere come segue:

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

### 2.4 Tecnologie interessate

- **Amazon Web Services (AWS):** è una piattaforma di cloud computing che consiste nella distribuzione on-demand delle risorse IT tramite Internet, con una tariffazione basata sul consumo;
- **Elastic Container Service o Elastic Kubernetes Service:** è un servizio di orchestrazione 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:** è un 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:** è un 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:** ideale per lo sviluppo di API RESTFUL JSON a supporto dell'applicativo;

- **Python:** linguaggio di programmazione ideale per lo sviluppo delle componenti di MACHINE LEARNINGGI;
- **HTML5, CSS3 e Javascript:** linguaggi di programmazione per lo sviluppo dell'interfaccia web di gestione dei workflow.

## 2.5 Aspetti Positivi

Il presente capitolato è interessante in quanto tratta dell'applicazione delle tecnologie di MACHINE LEARNINGGI al mondo sportivo.

I proponenti sono disponibili a fornire attività di formazione riguardo le principali tecnologie AWS e WIREFRAMEGI dell'interfaccia della console web di analisi e controllo dello stato di elaborazione del video, le quali sarebbero molto utili data l'inesperienza del gruppo nel mondo del MACHINE LEARNINGGI.

Il capitolato è inoltre molto chiaro per quel che riguarda le aspettative sul prodotto, le tecnologie da utilizzare, delle quali viene fornita anche la documentazione, e i materiali da condividere con il proponente.

## 2.6 Rischi

Il capitolato C1 prevede l'utilizzo di numerose tecnologie nuove per i membri del team, sia dal punto di vista teorico che pratico, le quali richiedono un tempo di approfondimento e studio personale che potrebbe avere un costo importante sul progetto.

Nonostante la disponibilità dimostrata dai proponenti, il team concorda che un ulteriore fattore di rischio riguarda l'aspetto di addestramento di un modello di MACHINE LEARNINGGI la cui realizzazione potrebbe richiedere competenze complesse.

## 2.7 Conclusioni

Dopo un'attenta valutazione, nonostante l'interesse di alcuni membri del gruppo al capitolato C1, si è scelto di non proporsi per la sua realizzazione in quanto i rischi connessi al suo svolgimento non fanno sentire a proprio agio i membri del team.

## 3 Capitolato C2 - Etherless

### 3.1 Informazioni generali

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

### 3.2 Descrizione

Etherless è una piattaforma cloud che consente agli sviluppatori di distribuire funzioni Javascript nel cloud che gli utenti possono eseguire pagando la commissione stabilita dallo sviluppatore stesso sfruttando la tecnologia SMART CONTRACT<sub>GI</sub> di ETHEREUM<sub>GI</sub>. Etherless integra le tecnologie ETHEREUM<sub>GI</sub>, per gestire i pagamenti e innescare l'invocazione delle funzioni, e SERVERLESS<sub>GI</sub>, per eseguire le funzioni.

### 3.3 Finalità del progetto

L'obiettivo del progetto è la creazione di un sistema per il deploy e l'esecuzione di funzioni Javascript con pagamenti gestiti tramite ETHEREUM<sub>GI</sub>.

L'architettura di Etherless sarà composta da 3 moduli principali:

- **Etherless-cli**: è il modulo con cui gli sviluppatori interagiranno con Etherless. Deve permettere:
  - la configurazione di un account ETHEREUM<sub>GI</sub>;
  - la possibilità di fare il deploy di funzioni sul server;
  - di visualizzare le funzioni presenti sul server;
  - di eseguire le funzioni presenti sul server;
  - di visualizzare log riguardo l'esecuzione di queste funzioni.
- **Etherless-smart**: è un insieme di SMART CONTRACT<sub>GI</sub> che gestisce la comunicazione tra etherless-cli e etherless-server tramite l'ascolto e l'emissione di eventi. Inoltre, deve gestire i pagamenti necessari all'esecuzione delle funzioni presenti sul server.
- **Etherless-server**: questo componente rimane in ascolto di eventi da etherless-smart per effettuare il deploy o per eseguire funzioni; una volta che una funzione è stata eseguita questo componente deve emettere un evento che contiene il valore restituito dalla funzione oppure un'eccezione nel caso si sia verificata.

### 3.4 Tecnologie interessate

- **Ethereum blockchain**: la BLOCKCHAIN<sub>GI</sub> per approvare ed archiviare le transazioni che avvengono sulla piattaforma;
- **Ethereum Virtual Machine (EVM)**: macchina virtuale distribuita sulla rete ETHEREUM<sub>GI</sub>, verrà utilizzata per eseguire le DApps;
- **Smart contract**: utilizzati per amministrare i contratti e/o transazioni tra i vari attori.
- **Solidity**: il linguaggio di programmazione usato per gli SMART CONTRACT<sub>GI</sub>;
- **DApps**: un'applicazione distribuita realizzata con gli smart contract di ETHEREUM<sub>GI</sub>;
- **Serverless framework**: un FRAMEWORK<sub>GI</sub> scritto in Node.js che permette la realizzazione di applicazioni SERVERLESS<sub>GI</sub>;

### **3.5 Aspetti positivi**

Il capitolato è interessante perché utilizza tecnologie innovative come la BLOCKCHAIN<sup>GI</sup> e Serverless Framework.

I proponenti sono apparsi particolarmente disponibili a guidare il gruppo soprattutto nel periodo iniziale di progetto.

Inoltre, non appare complicato comprendere dal capitolato il funzionamento logico e le specifiche del prodotto atteso.

### **3.6 Rischi**

Il team ha poche conoscenze nel campo della BLOCKCHAIN<sup>GI</sup>, dei servizi SERVERLESS<sup>GI</sup> e in particolare della tecnologia ETHEREUM<sup>GI</sup>, sia dal punto di vista teorico che pratico. Lo svolgimento del progetto richiederebbe quindi uno studio approfondito per comprendere adeguatamente tali tecnologie e la loro natura fa apparire ai membri del team l'attività di auto-formazione particolarmente complessa.

### **3.7 Conclusioni**

Nonostante si tratti di un progetto interessante per alcuni membri del gruppo, si è deciso di non proporsi per lo svolgimento del capitolato, sia per il rischio legato alle tempistiche di studio e approfondimento delle tecnologie richieste, sia per la preferenza del gruppo verso altri capitolati.



## 4 Capitolato 3 - NaturalAPI

### 4.1 Informazioni generali

- Nome: NaturalAPI
- Proponente: teal.blue
- Committente: Prof. Tullio Vardanega e Prof. Riccardo Cardin

### 4.2 Descrizione

Uno dei principali problemi legati alla realizzazione di un progetto software riguarda le difficoltà comunicative tra STAKEHOLDER<sub>GI</sub>, le quali possono causare fraintendimenti e conseguenti ritardi o aumenti dei costi del progetto. Per esempio, gli sviluppatori possono descrivere le proprie soluzioni mediante il codice che, nella maggior parte dei casi, sarà però di difficile interpretazione per il cliente. Il BEHAVIOR-DRIVEN DEVELOPMENT (BDD)<sub>GI</sub> ha come scopo lo sfruttamento della potenza del linguaggio naturale, noto a tutti, per parificare la comunicazione tra stakeholder mediante l'utilizzo di modelli di linguaggio standardizzati che rendano facile e priva di fraintendimenti la comunicazione. Gherkin è lo standard de facto che permette di descrivere scenari in linguaggio naturale secondo un modello standard che può essere poi collegato al codice dell'applicazione. Attualmente però il linguaggio naturale e il codice sono ancora collegati dall'intervento dello sviluppatore, che stabilisce la firma del metodo dell'API a cui viene passato il parametro rilevato dallo scenario scritto in linguaggio naturale.

### 4.3 Finalità del progetto

L'obiettivo di teal.blue è quello di creare un toolkit PROOF-OF-CONCEPT<sub>GI</sub> chiamato NaturalAPI che permetta di ridurre il divario tra le specifiche del progetto in linguaggio naturale e le API. Il fine sarà quello di consentire agli sviluppatori futuri di scrivere API più coerenti, più prevedibili e più gestibili a partire direttamente dai requisiti del progetto. In questo modo gli sviluppatori potrebbero concentrarsi sullo sviluppo di funzionalità senza dover replicare il lavoro di analisi già svolto per produrre i casi d'uso o altre forme di modellazione del progetto.

NaturalAPI dovrà quindi essere composto di 3 elementi:

- **NaturalAPI Discover:** un BUSINESS DOMAIN LANGUAGE (BDL)<sub>GI</sub> extractor, che sia in grado di individuare i termini più utilizzati dentro un certo tipo di business e le relazioni fra essi (entità, verbi e predicati);
- **NaturalAPI Design:** un parser di scenari e funzionalità che crei un'API BUSINESS APPLICATION LANGUAGE (BAL)<sub>GI</sub> a partire dai documenti Gherkin e un business domain language (BDL);
- **NaturalAPI Develop:** un esportatore di linguaggio che converta il BAL in casi test e API in uno dei linguaggi e FRAMEWORK<sub>GI</sub> disponibili per l'applicazione supportando la creazione di nuovi REPOSITORY<sub>GI</sub> di codice e l'aggiornamento di quelli esistenti.

#### 4.4 Tecnologie interessate

- **Natural language processing:** insieme delle tecnologie per automatizzare l'analisi della struttura di un testo;
- **Stanford Parser:** un parser open source;
- **Behavioural Driven Development**
- **Gherkin:** uno standard per produrre user case in formato BDD;
- **Cucumber o Heaptest:** strumenti software per supportare il BDD;
- **OpenApi:** un formato per rappresentare le API;
- **Swagger:** software FRAMEWORKGI per creare API, basato sul formato OpenAPI.

#### 4.5 Aspetti positivi

Il progetto proposto da teal.blue è particolarmente interessante per la finalità che esso ha. La comunicazione è infatti un aspetto molto importante e uno strumento che sia in grado di standardizzare il linguaggio usato dagli stakeholder avrà sicuramente un impatto positivo in termini di efficacia. La tecnologia del natural language processing è proiettata in un futuro in cui il fulcro della realizzazione di un prodotto sono le funzionalità che esso mette a disposizione e la logica implementativa di queste ultime, mentre lo sviluppo del codice diventerà sempre più automatico. Il presente capitolato è quindi molto stimolante per quel che riguarda le finalità del prodotto proposto e permetterebbe di conoscere nuove tecnologie importanti quali BDDGI.

#### 4.6 Rischi

Il rischio principale riguarda la mancata conoscenza da parte del team di progetto delle tecnologie Natural language processing, BDDGI e conseguentemente Gherkin e Cucumber. Sarebbe quindi necessario un approfondito studio preliminare affinché il team sia in grado di padroneggiare i concetti richiesti e possa metterli in pratica.

#### 4.7 Conclusioni

Il progetto risulta particolarmente interessante e stimolante, tuttavia i membri del team concordano sul fatto che lo studio preliminare dei concetti e delle tecnologie coinvolte potrebbero avere un impatto troppo importante sulle tempistiche di progetto. Si decide quindi di non proporsi per la realizzazione del presente capitolato a causa della mancata garanzia di rientrare nei tempi realizzativi richiesti.

## 5 Capitolato C4 - Predire in Grafana

### 5.1 Informazioni generali

- Nome Capitolato: Predire in Grafana
- Proponente: Zucchetti
- Committente: Prof. Tullio Vardanega e Prof. Riccardo Cardin

### 5.2 Descrizione

GRAFANAGI è un prodotto open-source che i proponenti utilizzano per monitorare le prestazioni dei propri sistemi. Attualmente la DASHBOARDGI di Grafana presenta lo stato dei sistemi basato sui dati raccolti dagli stessi evidenziando eventuali situazioni di allarme. Il proponente vorrebbe estendere le funzionalità del prodotto realizzando un PLUG-INGI che analizzi il flusso di dati raccolto per sviluppare delle previsioni che permettano di prevenire stati critici nella linea di produzione software.

### 5.3 Finalità del progetto

Nello specifico, si richiede di realizzare un PLUG-INGI di GRAFANAGI che sia in grado di svolgere un'analisi predittiva sul flusso di dati con l'uso di SUPPORT VECTOR MACHINEGI e REGRESSIONE LINEAREGI. Il fine è quello di attivare allarmi o segnalatori e visualizzare grafici riassuntivi che possano comunicare agli operatori quali problemi potrebbero insorgere e quali siano i punti critici della linea di produzione su cui conviene intervenire.

Il software richiesto dovrà essere in grado di:

1. Produrre un file JSONGI dai dati di addestramento con i parametri per le previsioni con Support Vector Machine (SVM) per le classificazioni o la Regressione Lineare (RL o LR da Linear Regression in inglese);
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

### 5.4 Tecnologie interessate

- **JavaScript:** linguaggio di scripting orientato agli oggetti e agli eventi, utilizzato per lo sviluppo delle componenti di GRAFANAGI e delle librerie consigliate per le SVMGI, RLGI e RETI NEURALIGI;
- **JSON:** formato adatto all'interscambio di dati fra applicazioni client/server. Richiesto per l'addestramento delle SVM/RL e per la definizione del predittore;
- **Grafana:** prodotto di monitoraggio di dati;
- **Orange Canvas:** strumento di analisi per la comprensione delle SVMGI e RLGI.
- **github.com o repository pubblici alternativi:** strumenti per condividere alla community la soluzione in conformità con i relativi requisiti di natura open-source;

## 5.5 Aspetti positivi

Il gruppo ritiene molto interessante il tema dell'analisi dei dati e l'utilizzo di modelli di apprendimento automatico.

Inoltre, si è compresa l'importanza dell'analisi preventiva, in quanto la si reputa molto utile per il lavoro quotidiano, nell'ottica di perseguire un approccio del tipo QUALITY ASSURANCE<sup>GI</sup>, sia per gli operatori di servizi Cloud che per la produzione quotidiana del Software.

Un altro aspetto importante è che il proponente si è reso disponibile a fornire la formazione necessaria e le librerie per la realizzazione degli algoritmi di MACHINE LEARNING<sup>GI</sup>.

## 5.6 Rischi

Le poche conoscenze nell'ambito dei modelli di apprendimento automatico richiedono uno studio preliminare di queste per comprendere meglio l'ambiente di applicazione delle stesse.

## 5.7 Conclusioni

La scelta di candidarsi per la realizzazione del presente capitolato è stata stimolata dall'interesse nell'impiego di modelli di apprendimento per l'analisi predittiva dei dati e per l'impatto positivo e costruttivo che l'analisi preventiva può portare in un ambiente lavorativo importante che richiede monitoraggio.

Inoltre, il gruppo condivide la scelta del proponente di rendere pubblico lo sviluppo e la realizzazione di tale progetto al fine di rispettare le politiche open-source delle tecnologie utilizzate, così che se ci fossero aziende interessate allo sviluppo di PLUG-IN<sup>GI</sup> simili ne possano usufruire liberamente.

Gli argomenti trattati in questo capitolato incuriosiscono e interessano i membri del team in una prospettiva di incremento delle nostre conoscenze personali e in un'ottica orientata al mondo del lavoro in quanto l'utilizzo di algoritmi di MACHINE LEARNING<sup>GI</sup> è sempre più richiesta (come si può notare anche dal suo utilizzo in altri capitolati).

Sebbene le tecnologie richieste non siano molto conosciute dai membri del gruppo, viene fatto affidamento sulla disponibilità del proponente ad offrire ore di formazione, qualora lo studio autonomo per colmare le lacune presenti non sia sufficiente.

## 6 Capitolato C5 - Stalker

### 6.1 Informazioni generali

- Nome: Stalker
- Proponente: Imola informatica
- Committente: Prof. Tullio Vardanega e Prof. Riccardo Cardin

### 6.2 Descrizione

La normativa vigente in materia di sicurezza comporta la necessità di regolamentare e gestire le presenze nei locali pubblici, al fine di garantire la maggior sicurezza possibile in caso di emergenze o più semplicemente di attestare il rispetto della capienza massima di un locale. Tale approccio potrebbe inoltre risultare di interesse in ambito lavorativo per segnalare la propria presenza sul posto di lavoro. Per fare ciò diventa essenziale poter tracciare il numero di persone presenti all'interno dei luoghi che si ha interesse a monitorare.

### 6.3 Finalità del progetto

Gli obiettivi del progetto prevedono:

- Server, completo di UI, con la principale finalità di consentire agli amministratori di
  - Loggarsi con un'utenza personale;
  - Creare, modificare ed eliminare organizzazioni (soggetto che ha interesse a tracciare le presenze delle persone all'interno dei propri luoghi);
  - Creare, modificare ed eliminare luoghi (spazio fisico riconducibile ad una organizzazione)
  - Monitorare le presenze in qualsiasi momento;
  - Effettuare ricerche per specifico dipendente;
  - Fornire report e statistiche di interesse.
- Applicazione mobile (IOS o Android) che permetta essenzialmente agli utenti di scaricare le liste delle organizzazioni ed autenticarsi in una di queste qualora previsto o, in alternativa, di risultare presente in maniera anonima.
- TEST DI CARICOGI che dimostrino il corretto funzionamento in situazioni normali, di carico e di sovraccarico.
- Copertura di test  $\geq 80\%$  correlata di report.
- Report dei test effettuati relativamente all'ottimizzazione della precisione dell'applicazione rispetto al consumo della batteria dei cellulari.
- Documentazione riguardo:
  - Scelte implementative e progettuali effettuate e relative motivazioni;
  - Problemi aperti e eventuali soluzioni proposte da esplorare.

### 6.4 Tecnologie interessate

- **Java** (versione 8 o superiori), **Python/Nodejs** (per server back-end)
- **Protocolli asincroni** (app mobile-server)
- **Pattern di Publisher/Subscriber**
- **IAAS Kubernetes o PAAS (Openshift o Reacher)**: per il rilascio delle componenti del Server nonché per la gestione della scalabilità orizzontale;

## **6.5 Aspetti positivi**

La finalità del progetto è sicuramente un aspetto positivo in quanto si vuole tentare di fornire una valida soluzione al problema della gestione delle presenze in luoghi pubblici, molto importante al giorno d'oggi. Fornisce infatti uno strumento utile a garantire il rispetto delle normative di sicurezza, ma trova una giustificata applicazione anche in ambito lavorativo per un rapido e preciso controllo degli orari lavorativi dei dipendenti. L'applicazione inoltre, considerati i numerosi scenari in cui potrebbe essere impiegata, potrebbe raggiungere un importante bacino di utenti utilizzatori.

## **6.6 Rischi**

Il principale problema riscontrato è legato alla necessità di fornire con sufficiente precisione la posizione degli utenti dell'applicazione, in modo da attestare con certezza la loro presenza all'interno dei luoghi da monitorare. In particolare, tale operazione richiede di tenere conto delle risorse utilizzate dalla geolocalizzazione sui dispositivi mobili per ottimizzarne il più possibile l'autonomia. Anche l'utilizzo di una soluzione ibrida network-GPS pone l'attenzione proprio sulla precisione che entrambi questi sistemi possono fornire per raggiungere lo scopo prefissato.

## **6.7 Conclusioni**

Il progetto è risultato molto interessante per la possibilità che offre di approfondire tecnologie non affrontate nel corso di studi e per l'obiettivo di grande utilità che si prefissa di raggiungere. Tuttavia, valutati i rischi del progetto e le problematiche in gioco per la loro risoluzione, oltre che le singole conoscenze da raggiungere per una piena padronanza delle scelte progettuali da effettuare, si è deciso di non far ricadere la scelta su questo capitolato.

## 7 Capitolato C6 – ThiReMa - Things Relationship Management

### 7.1 Informazioni generali

- Nome: ThiReMa (Things Relationship Management)
- Proponente: Sanmarco Informatica
- Committente: Prof. Tullio Vardanega e Prof. Riccardo Cardin

### 7.2 Descrizione

ThiReMa - Things Relationship Management - si inserisce nel contesto dei Big Data, IOT e analisi predittiva. Al giorno d'oggi infatti sono sempre più i prodotti intelligenti presenti sul mercato dotati di sensori che producono un'ingente quantità di dati, salvata nel cloud, il cui studio può fornire informazioni utili per il miglioramento e la manutenzione del prodotto stesso. I dati possono provenire da una gamma di prodotti intelligenti molto diversi tra loro: si parla sia di elettrodomestici (come frigoriferi e macchine da caffè) che di wearable devices (fitband e smartwatch), ma anche di robot e macchine industriali. Ne consegue che sia necessario un software in grado di gestire l'eterogeneità delle misurazioni fatte dai diversi sensori e accumulare i dati rilevati in maniera EFFICIENTE e affidabile in un database centralizzato affinché siano disponibili per una successiva attività di analisi predittiva.

### 7.3 Finalità del progetto

Lo scopo del progetto è quello di realizzare una web-application che sia in grado di gestire efficientemente la memorizzazione in un database centralizzato dei dati provenienti da sensori eterogenei installati nei dispositivi presenti sul mercato. I dati raccolti dovranno quindi essere elaborati ed analizzati col fine di valutare la correlazione tra dati operativi misurati e i fattori influenzanti. L'analisi dei dati dovrà essere in grado di fornire semplici previsioni sull'andamento dei dati stessi ed offrire soluzioni in termini di manutenzione predittiva. Inoltre, deve essere previsto un sistema di dispatching che permetta di inoltrare tempestivamente le informazioni urgenti per il buon governo dei dispositivi.

La web-application sarà suddivisa in tre macro-sezioni:

1. Censimento dei sensori e dei relativi dati;
2. Modulo di analisi di correlazione;
3. Modulo di monitoraggio per ente.

L'ambiente in cui si inserisce questo sistema software sarà pertanto strutturato in:

- **EDGE DATA POINT:** i dispositivi che effettuano misurazioni di diversi parametri e le inviano ad un database;
- **Database:** utile per la raccolta e la memorizzazione dei dati provenienti dai diversi sensori ed i relativi METADATI. Inoltre, deve registrare anche i dati che riguardano gli enti utilizzatori del servizio, con le relative informazioni di autorizzazione;
- **Interfaccia di gestione:** un'applicazione web che deve fornire una gestione minimale degli utenti e degli enti. Inoltre, deve consentire di definire quali dati può inviare ogni EDGE DATA POINT, chi può visualizzarli e impostare le soglie massime/minime per innescare l'invio di notifiche. Deve possedere anche una sezione di visualizzazione dei dati in tempo reale tramite grafici e una serie di previsioni effettuate sulla base delle relazioni tra i vari dati;

## 7.4 Tecnologie interessate

- **Kafka:** è una piattaforma software open-source che fornisce la gestione di un flusso di dati in tempo reale, garantendo una bassa latenza ed un'elevata velocità;
- **Java 8:** è supportato nativamente da Kafka, quindi è il linguaggio di programmazione consigliato per lo sviluppo della logica del sistema e delle varie componenti Kafka;
- **Time series database (TSDB):** è un database ottimizzato per dati che hanno una stretta relazione cronologica. Sono molto comuni negli ambiti in cui è necessaria una raccolta di dati che devono essere aggregati in base al tempo.

I principali TSDB consigliati sono:

- **Timescale**
- **Clikhouse**
- **Bootstrap:** è un FRAMEWORKGI per lo sviluppo del front-end di applicazioni web; contiene diversi template CSS e Javascript ottimizzati per rendere l'interfaccia responsive;
- **GRAFANA:** per la visualizzazione dei dati raccolti sotto forma di grafici;
- **Docker:** sistema utile per la gestione di container che eseguono processi in ambienti isolati, che verrà utilizzato per l'istanziamento di tutti i componenti.

## 7.5 Aspetti Positivi

Il capitolato è un'intersezione tra tecnologie già note, come Java8 e lo sviluppo front-end di applicazioni web, e tecnologie innovative che non sono mai state incontrate durante il corso di studi, tra cui i time series database e la piattaforma Apache Kafka.

Inoltre, il presente capitolato permette di affrontare argomenti interessanti quali l'Internet of Things e la correlazione di dati eterogenei per lo svolgimento di analisi predittive e preventive.

## 7.6 Rischi

Nonostante i software da utilizzare siano ampiamente diffusi e ben documentati, è necessario prestare attenzione alla loro configurazione, sia come unità che come sistema.

Inoltre, poiché l'interazione con l'utente è fondamentale, deve essere garantita una continua consistenza del sistema.

Altri fattori non meno importanti sono la scarsa esperienza del team nella gestione server side e la mancanza di conoscenze riguardanti l'analisi e la correlazione di grandi quantità di dati eterogenei. Per evitare che questi fattori di rischio compromettano la qualità del prodotto finale, il gruppo dovrebbe quindi affrontare un periodo di studio importante per comprendere adeguatamente le tecnologie coinvolte.

## 7.7 Conclusioni

Il capitolato ha da subito interessato molti componenti del gruppo. Si è fatta quindi un'attenta analisi degli aspetti positivi e dei rischi correlati al presente progetto a cui è poi seguita una fase di comparazione con gli altri capitolati e si è infine deciso di non candidarsi per la realizzazione di ThiReMa.



## Registro delle modifiche

Versione	Data	Descrizione	Nominativo	Ruolo
1.0.0	2020/01/13	Approvazione del documento	Giacomo Callegari	Responsabile
0.3.0	2020/01/11	Revisione del documento	Nicolò Fassina	Verificatore
0.2.1	2019/12/15	Correzioni e integrazioni minori al documento	Veronica Pederiva	Analista
0.2.0	2019/12/14	Revisione del documento	Nicolò Fassina	Verificatore
0.1.2	2019/12/13	Redazione dello studio di fattibilità del capitolato C5	Alessandro Lovo	Analista
0.1.1	2019/12/13	Redazione dello studio di fattibilità del capitolato C1	Giacomo Callegari	Analista
0.1.0	2019/12/12	Verifica degli studi di fattibilità svolti; correzioni e integrazioni agli studi di fattibilità di C2 e C4	Nicolò Fassina Francesco Gobbo, Manuel De Franceschi	Verificatore, Analisti
0.0.6	2019/12/11	Redazione dello studio di fattibilità del capitolato C4	Francesco Gobbo	Analista
0.0.5	2019/12/11	Redazione dello studio di fattibilità del capitolato C3	Veronica Pederiva	Analista
0.0.4	2019/12/11	Redazione dello studio di fattibilità del capitolato C6	Andrea Longo	Analista
0.0.3	2019/12/10	Redazione dell'introduzione e dello studio di fattibilità del capitolato C2	Manuel De Franceschi	Analista
0.0.2	2019/12/04	Redazione struttura documento	Francesco Gobbo	Amministratore
0.0.1	2019/11/29	Creazione del documento	Andrea Longo	Responsabile