



Progetto: *Nome Progetto*

codebusterswe@gmail.com

Studio Di Fattibilità

Informazioni sul documento

Versione	1.0.0
Approvatori	Sassaro Giacomo
Redattori	Pirola Alessandro Zenere Marco Rago Alessandro Safdari Houssaine
Verificatori	Baldisseri Michele Scialpi Paolo
Uso	Interno
Distribuzione	Prof. Vardanega Tullio Prof. Cardin Riccardo Gruppo <i>CodeBusters</i>

Descrizione

Questo documento si occupa di descrivere l'analisi dei capitolati d'appalto realizzata dal gruppo al fine di valutarne la fattibilità

Registro delle modifiche

Versione	Data	Nominativo	Ruolo	Verificatore	Descrizione
0.0.1	2019-11-20	<i>CodeBusters</i>	Analisti	-	Creazione bozza documento, introduzione e paragrafi.

Indice

1	Introduzione	4
1.1	Scopo del Documento	4
1.2	Glossario	4
1.3	Riferimenti	4
1.3.1	Normativi	4
1.3.2	Informativi	4
2	Capitolato C1	6
2.1	Titolo del capitolato	6
2.2	Descrizione del capitolo	6
2.3	Tecnologie coinvolte	6
2.4	Vincoli	7
2.5	Aspetti positivi	7
2.6	Aspetti critici	7
2.7	Conclusioni	7
3	Capitolato C2	8
3.1	Titolo del capitolato	8
3.2	Descrizione del capitolo	8
3.3	Tecnologie coinvolte	8
3.4	Vincoli	9
3.5	Aspetti positivi	9
3.6	Aspetti critici	9
3.7	Conclusioni	9
4	Capitolato C3	10
4.1	Titolo del capitolato	10
4.2	Descrizione del capitolo	10
4.3	Tecnologie coinvolte	10
4.4	Vincoli	11
4.5	Aspetti positivi	11
4.6	Aspetti critici	11
4.7	Conclusioni	11
5	Capitolato C4	12
5.1	Titolo del capitolato	12
5.2	Descrizione del capitolo	12
5.3	Tecnologie coinvolte	12
5.4	Vincoli	13
5.5	Aspetti positivi	13
5.6	Aspetti critici	13
5.7	Conclusioni	13

6	Capitolato C5	14
6.1	Titolo del capitolato	14
6.2	Descrizione del capitolo	14
6.3	Tecnologie coinvolte	14
6.4	Vincoli	15
6.5	Aspetti positivi	15
6.6	Aspetti critici	15
6.7	Conclusioni	15
7	Capitolato C6	16
7.1	Titolo del capitolato	16
7.2	Descrizione del capitolo	16
7.3	Tecnologie coinvolte	16
7.4	Vincoli	17
7.5	Aspetti positivi	17
7.6	Aspetti critici	17
7.7	Conclusioni	17
8	Capitolato C7	18
8.1	Titolo del capitolato	18
8.2	Descrizione del capitolo	18
8.3	Tecnologie coinvolte	18
8.4	Vincoli	19
8.5	Aspetti positivi	19
8.6	Aspetti critici	19
8.7	Conclusioni	19

1 Introduzione

1.1 Scopo del Documento

Questo documento contiene la stesura dello studio di fattibilità riguardante i sei capitolati proposti, dove per ciascuno di essi vengono evidenziati i seguenti aspetti:

- Titolo del capitolato;
- Descrizione generale;
- Prerequisiti e tecnologie coinvolte;
- Vincoli;
- Aspetti positivi;
- Aspetti critici.

Infine, per ogni capitolato vengono espone le motivazioni e le ragioni per cui il gruppo ha scelto come progetto il capitolato CX *Nome Progetto* a discapito degli altri sei proposti.

1.2 Glossario

Al fine di evitare ambiguità fra i termini, e per avere chiare fra tutti gli stakeholder le terminologie utilizzate per la realizzazione del presente documento, il gruppo *CodeBusters* ha redatto un documento denominato *Glossario 1.0.0*. In tale documento, sono presenti tutti i termini tecnici, ambigui, specifici del progetto e scelti dai membri del gruppo con le loro relative definizioni. Un termine presente nel *Glossario 1.0.0* e utilizzato in questo documento viene indicato con un apice ^G alla fine della parola.

1.3 Riferimenti

1.3.1 Normativi

- *Norme di Progetto 1.0.0*.

1.3.2 Informativi

- **Capitolato d'appalto C1 - BLOCKCOVID:**
<https://www.math.unipd.it/~tullio/IS-1/2020/Progetto/C1.pdf>
- **Capitolato d'appalto C2 - EmporioLambda:**
<https://www.math.unipd.it/~tullio/IS-1/2020/Progetto/C2.pdf>
- **Capitolato d'appalto C3 - GDP:**
<https://www.math.unipd.it/~tullio/IS-1/2020/Progetto/C3.pdf>

- **Capitolato d'appalto C4 - HD Viz:**
<https://www.math.unipd.it/~tullio/IS-1/2020/Progetto/C4.pdf>
- **Capitolato d'appalto C5 - PORTACS:**
<https://www.math.unipd.it/~tullio/IS-1/2020/Progetto/C5.pdf>
- **Capitolato d'appalto C6 - RGP:**
<https://www.math.unipd.it/~tullio/IS-1/2020/Progetto/C6.pdf>
- **Capitolato d'appalto C7 - SSD:**
<https://www.math.unipd.it/~tullio/IS-1/2020/Progetto/C6.pdf>

2 Capitolato C1

2.1 Titolo del capitolato

Il capitolato in questione si chiama "*BlockCOVID: supporto digitale al contrasto della pandemia*", il proponente è l'azienda *imola informatica* e i committenti sono Prof. Vardanega Tullio e Prof. Cardin Riccardo.

2.2 Descrizione del capitolo

Il capitolato uno propone di sviluppare un'applicazione per registrare le presenze in uffici in modo da permettere la pulizia delle postazioni ed evitare che la stessa sia utilizzata da più persone. L'obiettivo finale è la creazione di un'applicazione mobile con due interfacce. La prima destinata agli amministratori per la gestione degli uffici e delle postazioni mentre la seconda per due categorie di utenti: il dipendente che registra la sua presenza e l'addetto alle pulizie, il quale riceve la lista delle postazioni da disinfettare.

2.3 Tecnologie coinvolte

((Sebbene l'azienda non impone tecnologie specifiche per lo sviluppo del server o della UI, vi sono comunque delle scelte preferenziali da considerare nello svolgimento del progetto))(?):

- Java (versione 8 o superiori), Python o nodejs per lo sviluppo del server back-end;
- protocolli asincroni per le comunicazioni app mobile-server;
- un sistema blockchain per salvare con opponibilità a terzi i dati di sanificazione;
- IAAS Kubernetes o di un PAAS, Openshift o Rancher, per il rilascio delle componenti del server e la gestione della scalabilità orizzontale.
- Avere il server che esponga delle API Rest attraverso le quali sia possibile utilizzare l'applicativo. In alternativa è possibile utilizzare gRPC come soluzione alternativa al Rest.

2.4 Vincoli

DA FARE

2.5 Aspetti positivi

DA FARE

2.6 Aspetti critici

DA FARE

2.7 Conclusioni

Visto il contesto attuale di pandemia un'applicazione su questo campo è quantomai utile e dunque stimola interesse. La valutazione è positiva anche per quanto riguarda le tecnologie che si dovranno esplorare per portare a termine questo capitolato. Il lato negativo risiede nella complessità dello sviluppo dell'applicazione mobile nonché del lato server.

3 Capitolato C2

3.1 Titolo del capitolato

Il capitolato in questione si chiama ” *EmporioLambda: piattaforma di e-commerce in stile Serverless*”, il proponente è l’azienda *Red Babel* e i committenti sono Prof. Vardanega Tullio e Prof. Cardin Riccardo.

3.2 Descrizione del capitolo

Nel capitolato due viene proposto di creare una piattaforma di E-Commerce, ovvero la compravendita on-line di prodotti. Lo scopo finale del capitolato è avere una piattaforma dove i clienti possano registrarsi, ricercare i prodotti, aggiungerli al carrello (ovvero una pagina che permette di rivedere e considerare i prodotti e l’ammontare complessivo dei prezzi) e l’acquisto. La piattaforma dovrà offrire un’interfaccia per i venditori dove possono aggiungere e modificare i prodotti destinati alla vendita.

3.3 Tecnologie coinvolte

((Sebbene l’azienda non impone tecnologie specifiche per lo sviluppo del server o della UI, vi sono comunque delle scelte preferenziali da considerare nello svolgimento del progetto))(?):

- Questa applicazione deve essere costruita con la tecnologia serverless con una API basata su HTTP. Per questo scopo si utilizza AWS Lambda e altre componenti di supporto (AWS API Gateway, AWS DynamoDB, AWS S3)
- CloudFormation per la gestione delle risorse sopraelencate
- Serverless Framework per l’implementazione di applicazioni serverless facilitandone alcune difficili strutture (come permessi, sottoscrizioni o accesso)
- Il BFF (Back end for Front end) richiede l’uso di Next.js

3.4 Vincoli

DA FARE

3.5 Aspetti positivi

DA FARE

3.6 Aspetti critici

DA FARE

3.7 Conclusioni

er il capitolato due si utilizzano tecnologie alla portata di tutti e, prendendo in considerazione le altre piattaforme di e-commerce esistenti, è relativamente facile progettarela. Proprio perché ci sono già moltissime piattaforme di e-commerce esistenti rende questo capitolato poco appetibile.

4 Capitolato C3

4.1 Titolo del capitolato

Il capitolato in questione si chiama "*GDP - Gathering Detection Platform*", il proponente è l'azienda *SyncLab* e i committenti sono Prof. Vardanega Tullio e Prof. Cardin Riccardo.

4.2 Descrizione del capitolo

Il capitolato presentato si pone come obiettivo finale la realizzazione di una piattaforma in grado di dare all'utilizzatore finale una rappresentazione grafica delle informazioni sulla probabilità di assembramento nelle zone potenzialmente a rischio. Questo viene fatto sulla base di alcuni dati che vengono individuati, raccolti ed elaborati.

La piattaforma deve sfruttare tecniche di machine learning e/o deeplearning in modo da predire le zone potenzialmente a rischio.

4.3 Tecnologie coinvolte

Sebbene l'azienda non impone tecnologie specifiche per lo sviluppo del server o della UI, vi sono comunque delle scelte preferenziali da considerare nello svolgimento del progetto:

- utilizzo di Java^G e di Angular^G per lo sviluppo della parte di Back-end e di Front-end della componente Web Application del sistema;
- per la gestione delle mappe (heatmap ecc.) il framework Leaflet^G (<https://leafletjs.com>);
- utilizzo di protocolli asincroni per le comunicazioni tra le diverse componenti;
- utilizzo del pattern Publisher/Subscriber, e adozione del protocollo MQTT ('MQ Telemetry Transport or Message Queue Telemetry Transport'), caratterizzato per essere open, di facile implementazione e ampia diffusione in applicazioni M2M (MachineToMachine) e IoT (InternetOfThings).

Per la parte di Machine Learning, l'azienda dà ampia libertà, sebbene ci siano delle tecnologie consigliate:

- Python^G come linguaggio di programmazione;
- TensorFlow^G, Pytorch^G, Keras^G e Scikit-learn^G come libreria per l'apprendimento automatico;

Infine, per facilitare la comprensione delle librerie consigliate, l'azienda propone delle guide come cartacee come:

- Hands-on Machine Learning with Stick-Learn, Keras & TensorFlow;
- Phyton Machine Learning,

e qualche altra risorsa online come Kaggle, Aurelien Geron's Github e Google scholar.

4.4 Vincoli

La proponente suggerisce una architettura in grado di implementare quello che si definisce essere un Sistema Reattivo, in grado cioè di soddisfare le seguenti caratteristiche:

- responsive: la richiesta di un servizio deve sempre avere una risposta, anche quando si verifica un guasto;
- resilient: i servizi devono poter essere ripristinati a seguito di guasti;
- elastic: i servizi devono poter essere scalati in base alla effettiva domanda;
- message-driven: i servizi devono rispondere al mondo, non tentare di controllare ciò che fa.

Ed è richiesto che tutte le componenti applicative siano correlate da test unitari e d'integrazione. Inoltre, è richiesto che il sistema venga testato nella sua interezza tramite test *end-to-end*.

4.5 Aspetti positivi

- Il capitolato ha come obiettivo la creazione di una piattaforma di grande aiuto a fronte della situazione che stiamo vivendo;
- Nel capitolato si tratta l'argomento del machine learning^G, un argomento molto interessante e che potrebbe risultare molto utile in altri ambiti e in futuro;
- L'azienda sembra essere molto disponibile per seguire il gruppo nel percorso di sviluppo del progetto.

4.6 Aspetti critici

- Il capitolato richiede l'apprendimento di molti servizi, sconosciuti dai membri del gruppo, che si basano su argomenti non conosciuti (come il machine learning^G) e non affrontati nel corso di laurea triennale. Quindi gran parte del tempo andrebbe dedicato allo studio di questi concetti e strumenti, ed infine di applicare queste conoscenze;
- Oltre al tempo impiegato per l'apprendimento di questi strumenti c'è da tenere in considerazione il tempo per addestrare la parte di machine learning^G.

4.7 Conclusioni

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

5 Capitolato C4

5.1 Titolo del capitolato

Il capitolato in questione si chiama "*GDP - Gathering Detection Platform*", il proponente è l'azienda *SyncLab* e i committenti sono Prof. Vardanega Tullio e Prof. Cardin Riccardo.

5.2 Descrizione del capitolo

Il capitolato presentato si pone come obiettivo finale la realizzazione di una piattaforma in grado di dare all'utilizzatore finale una rappresentazione grafica delle informazioni sulla probabilità di assembramento nelle zone potenzialmente a rischio. Questo viene fatto sulla base di alcuni dati che vengono individuati, raccolti ed elaborati.

La piattaforma deve sfruttare tecniche di machine learning e/o deeplearnig in modo da predire le zone potenzialmente a rischio.

5.3 Tecnologie coinvolte

Sebbene l'azienda non impone tecnologie specifiche per lo sviluppo del server o della UI, vi sono comunque delle scelte preferenziali da considerare nello svolgimento del progetto:

- utilizzo di Java^G e di Angular^G per lo sviluppo della parte di Back-end e di Front-end della componente Web Application del sistema;
- per la gestione delle mappe (heatmap ecc.) il framework Leaflet^G (<https://leafletjs.com>);
- utilizzo di protocolli asincroni per le comunicazioni tra le diverse componenti;
- utilizzo del pattern Publisher/Subscriber, e adozione del protocollo MQTT ('MQ Telemetry Transport or Message Queue Telemetry Transport'), caratterizzato per essere open, di facile implementazione e ampia diffusione in applicazioni M2M (MachineToMachine) e IoT (InternetOfThings).

Per la parte di Machine Learning, l'azienda dà ampia libertà, sebbene ci siano delle tecnologie consigliate:

- Python^G come linguaggio di programmazione;
- TensorFlow^G, Pytorch^G, Keras^G e Scikit-learn^G come libreria per l'apprendimento automatico;

Infine, per facilitare la comprensione delle librerie consigliate, l'azienda propone delle guide come cartacee come:

- Hands-on Machine Learning with Stick-Learn, Keras & TensorFlow;
- Phyton Machine Learning,

e qualche altra risorsa online come Kaggle, Aurelien Geron's Github e Google scholar.

5.4 Vincoli

La proponente suggerisce una architettura in grado di implementare quello che si definisce essere un Sistema Reattivo, in grado cioè di soddisfare le seguenti caratteristiche:

- responsive: la richiesta di un servizio deve sempre avere una risposta, anche quando si verifica un guasto;
- resilient: i servizi devono poter essere ripristinati a seguito di guasti;
- elastic: i servizi devono poter essere scalati in base alla effettiva domanda;
- message-driven: i servizi devono rispondere al mondo, non tentare di controllare ciò che fa.

Ed è richiesto che tutte le componenti applicative siano correlate da test unitari e d'integrazione. Inoltre, è richiesto che il sistema venga testato nella sua interezza tramite test *end-to-end*.

5.5 Aspetti positivi

- Il capitolato ha come obiettivo la creazione di una piattaforma di grande aiuto a fronte della situazione che stiamo vivendo;
- Nel capitolato si tratta l'argomento del machine learning^G, un argomento molto interessante e che potrebbe risultare molto utile in altri ambiti e in futuro;
- L'azienda sembra essere molto disponibile per seguire il gruppo nel percorso di sviluppo del progetto.

5.6 Aspetti critici

- Il capitolato richiede l'apprendimento di molti servizi, sconosciuti dai membri del gruppo, che si basano su argomenti non conosciuti (come il machine learning^G) e non affrontati nel corso di laurea triennale. Quindi gran parte del tempo andrebbe dedicato allo studio di questi concetti e strumenti, ed infine di applicare queste conoscenze;
- Oltre al tempo impiegato per l'apprendimento di questi strumenti c'è da tenere in considerazione il tempo per addestrare la parte di machine learning^G.

5.7 Conclusioni

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

6 Capitolato C5

6.1 Titolo del capitolato

Il capitolato in questione si chiama "*GDP - Gathering Detection Platform*", il proponente è l'azienda *SyncLab* e i committenti sono Prof. Vardanega Tullio e Prof. Cardin Riccardo.

6.2 Descrizione del capitolo

Il capitolato presentato si pone come obiettivo finale la realizzazione di una piattaforma in grado di dare all'utilizzatore finale una rappresentazione grafica delle informazioni sulla probabilità di assembramento nelle zone potenzialmente a rischio. Questo viene fatto sulla base di alcuni dati che vengono individuati, raccolti ed elaborati.

La piattaforma deve sfruttare tecniche di machine learning e/o deeplearning in modo da predire le zone potenzialmente a rischio.

6.3 Tecnologie coinvolte

Sebbene l'azienda non impone tecnologie specifiche per lo sviluppo del server o della UI, vi sono comunque delle scelte preferenziali da considerare nello svolgimento del progetto:

- utilizzo di Java^G e di Angular^G per lo sviluppo della parte di Back-end e di Front-end della componente Web Application del sistema;
- per la gestione delle mappe (heatmap ecc.) il framework Leaflet^G (<https://leafletjs.com>);
- utilizzo di protocolli asincroni per le comunicazioni tra le diverse componenti;
- utilizzo del pattern Publisher/Subscriber, e adozione del protocollo MQTT ('MQ Telemetry Transport or Message Queue Telemetry Transport'), caratterizzato per essere open, di facile implementazione e ampia diffusione in applicazioni M2M (MachineToMachine) e IoT (InternetOfThings).

Per la parte di Machine Learning, l'azienda dà ampia libertà, sebbene ci siano delle tecnologie consigliate:

- Python^G come linguaggio di programmazione;
- TensorFlow^G, Pytorch^G, Keras^G e Scikit-learn^G come libreria per l'apprendimento automatico;

Infine, per facilitare la comprensione delle librerie consigliate, l'azienda propone delle guide come cartacee come:

- Hands-on Machine Learning with Stick-Learn, Keras & TensorFlow;
- Phyton Machine Learning,

e qualche altra risorsa online come Kaggle, Aurelien Geron's Github e Google scholar.

6.4 Vincoli

La proponente suggerisce una architettura in grado di implementare quello che si definisce essere un Sistema Reattivo, in grado cioè di soddisfare le seguenti caratteristiche:

- responsive: la richiesta di un servizio deve sempre avere una risposta, anche quando si verifica un guasto;
- resilient: i servizi devono poter essere ripristinati a seguito di guasti;
- elastic: i servizi devono poter essere scalati in base alla effettiva domanda;
- message-driven: i servizi devono rispondere al mondo, non tentare di controllare ciò che fa.

Ed è richiesto che tutte le componenti applicative siano correlate da test unitari e d'integrazione. Inoltre, è richiesto che il sistema venga testato nella sua interezza tramite test *end-to-end*.

6.5 Aspetti positivi

- Il capitolato ha come obiettivo la creazione di una piattaforma di grande aiuto a fronte della situazione che stiamo vivendo;
- Nel capitolato si tratta l'argomento del machine learning^G, un argomento molto interessante e che potrebbe risultare molto utile in altri ambiti e in futuro;
- L'azienda sembra essere molto disponibile per seguire il gruppo nel percorso di sviluppo del progetto.

6.6 Aspetti critici

- Il capitolato richiede l'apprendimento di molti servizi, sconosciuti dai membri del gruppo, che si basano su argomenti non conosciuti (come il machine learning^G) e non affrontati nel corso di laurea triennale. Quindi gran parte del tempo andrebbe dedicato allo studio di questi concetti e strumenti, ed infine di applicare queste conoscenze;
- Oltre al tempo impiegato per l'apprendimento di questi strumenti c'è da tenere in considerazione il tempo per addestrare la parte di machine learning^G.

6.7 Conclusioni

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

7 Capitolato C6

7.1 Titolo del capitolato

Il capitolato in questione si chiama "*GDP - Gathering Detection Platform*", il proponente è l'azienda *SyncLab* e i committenti sono Prof. Vardanega Tullio e Prof. Cardin Riccardo.

7.2 Descrizione del capitolo

Il capitolato presentato si pone come obiettivo finale la realizzazione di una piattaforma in grado di dare all'utilizzatore finale una rappresentazione grafica delle informazioni sulla probabilità di assembramento nelle zone potenzialmente a rischio. Questo viene fatto sulla base di alcuni dati che vengono individuati, raccolti ed elaborati.

La piattaforma deve sfruttare tecniche di machine learning e/o deeplearnig in modo da predire le zone potenzialmente a rischio.

7.3 Tecnologie coinvolte

Sebbene l'azienda non impone tecnologie specifiche per lo sviluppo del server o della UI, vi sono comunque delle scelte preferenziali da considerare nello svolgimento del progetto:

- utilizzo di Java^G e di Angular^G per lo sviluppo della parte di Back-end e di Front-end della componente Web Application del sistema;
- per la gestione delle mappe (heatmap ecc.) il framework Leaflet^G (<https://leafletjs.com>);
- utilizzo di protocolli asincroni per le comunicazioni tra le diverse componenti;
- utilizzo del pattern Publisher/Subscriber, e adozione del protocollo MQTT ('MQ Telemetry Transport or Message Queue Telemetry Transport'), caratterizzato per essere open, di facile implementazione e ampia diffusione in applicazioni M2M (MachineToMachine) e IoT (InternetOfThings).

Per la parte di Machine Learning, l'azienda dà ampia libertà, sebbene ci siano delle tecnologie consigliate:

- Python^G come linguaggio di programmazione;
- TensorFlow^G, Pytorch^G, Keras^G e Scikit-learn^G come libreria per l'apprendimento automatico;

Infine, per facilitare la comprensione delle librerie consigliate, l'azienda propone delle guide come cartacee come:

- Hands-on Machine Learning with Stick-Learn, Keras & TensorFlow;
- Phyton Machine Learning,

e qualche altra risorsa online come Kaggle, Aurelien Geron's Github e Google scholar.

7.4 Vincoli

La proponente suggerisce una architettura in grado di implementare quello che si definisce essere un Sistema Reattivo, in grado cioè di soddisfare le seguenti caratteristiche:

- responsive: la richiesta di un servizio deve sempre avere una risposta, anche quando si verifica un guasto;
- resilient: i servizi devono poter essere ripristinati a seguito di guasti;
- elastic: i servizi devono poter essere scalati in base alla effettiva domanda;
- message-driven: i servizi devono rispondere al mondo, non tentare di controllare ciò che fa.

Ed è richiesto che tutte le componenti applicative siano correlate da test unitari e d'integrazione. Inoltre, è richiesto che il sistema venga testato nella sua interezza tramite test *end-to-end*.

7.5 Aspetti positivi

- Il capitolato ha come obiettivo la creazione di una piattaforma di grande aiuto a fronte della situazione che stiamo vivendo;
- Nel capitolato si tratta l'argomento del machine learning^G, un argomento molto interessante e che potrebbe risultare molto utile in altri ambiti e in futuro;
- L'azienda sembra essere molto disponibile per seguire il gruppo nel percorso di sviluppo del progetto.

7.6 Aspetti critici

- Il capitolato richiede l'apprendimento di molti servizi, sconosciuti dai membri del gruppo, che si basano su argomenti non conosciuti (come il machine learning^G) e non affrontati nel corso di laurea triennale. Quindi gran parte del tempo andrebbe dedicato allo studio di questi concetti e strumenti, ed infine di applicare queste conoscenze;
- Oltre al tempo impiegato per l'apprendimento di questi strumenti c'è da tenere in considerazione il tempo per addestrare la parte di machine learning^G.

7.7 Conclusioni

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

8 Capitolato C7

8.1 Titolo del capitolato

Il capitolato in questione si chiama "*GDP - Gathering Detection Platform*", il proponente è l'azienda *SyncLab* e i committenti sono Prof. Vardanega Tullio e Prof. Cardin Riccardo.

8.2 Descrizione del capitolo

Il capitolato presentato si pone come obiettivo finale la realizzazione di una piattaforma in grado di dare all'utilizzatore finale una rappresentazione grafica delle informazioni sulla probabilità di assembramento nelle zone potenzialmente a rischio. Questo viene fatto sulla base di alcuni dati che vengono individuati, raccolti ed elaborati.

La piattaforma deve sfruttare tecniche di machine learning e/o deeplearnig in modo da predire le zone potenzialmente a rischio.

8.3 Tecnologie coinvolte

Sebbene l'azienda non impone tecnologie specifiche per lo sviluppo del server o della UI, vi sono comunque delle scelte preferenziali da considerare nello svolgimento del progetto:

- utilizzo di Java^G e di Angular^G per lo sviluppo della parte di Back-end e di Front-end della componente Web Application del sistema;
- per la gestione delle mappe (heatmap ecc.) il framework Leaflet^G (<https://leafletjs.com>);
- utilizzo di protocolli asincroni per le comunicazioni tra le diverse componenti;
- utilizzo del pattern Publisher/Subscriber, e adozione del protocollo MQTT ('MQ Telemetry Transport or Message Queue Telemetry Transport'), caratterizzato per essere open, di facile implementazione e ampia diffusione in applicazioni M2M (MachineToMachine) e IoT (InternetOfThings).

Per la parte di Machine Learning, l'azienda dà ampia libertà, sebbene ci siano delle tecnologie consigliate:

- Python^G come linguaggio di programmazione;
- TensorFlow^G, Pytorch^G, Keras^G e Scikit-learn^G come libreria per l'apprendimento automatico;

Infine, per facilitare la comprensione delle librerie consigliate, l'azienda propone delle guide come cartacee come:

- Hands-on Machine Learning with Stick-Learn, Keras & TensorFlow;
- Phyton Machine Learning,

e qualche altra risorsa online come Kaggle, Aurelien Geron's Github e Google scholar.

8.4 Vincoli

La proponente suggerisce una architettura in grado di implementare quello che si definisce essere un Sistema Reattivo, in grado cioè di soddisfare le seguenti caratteristiche:

- responsive: la richiesta di un servizio deve sempre avere una risposta, anche quando si verifica un guasto;
- resilient: i servizi devono poter essere ripristinati a seguito di guasti;
- elastic: i servizi devono poter essere scalati in base alla effettiva domanda;
- message-driven: i servizi devono rispondere al mondo, non tentare di controllare ciò che fa.

Ed è richiesto che tutte le componenti applicative siano correlate da test unitari e d'integrazione. Inoltre, è richiesto che il sistema venga testato nella sua interezza tramite test *end-to-end*.

8.5 Aspetti positivi

- Il capitolato ha come obiettivo la creazione di una piattaforma di grande aiuto a fronte della situazione che stiamo vivendo;
- Nel capitolato si tratta l'argomento del machine learning^G, un argomento molto interessante e che potrebbe risultare molto utile in altri ambiti e in futuro;
- L'azienda sembra essere molto disponibile per seguire il gruppo nel percorso di sviluppo del progetto.

8.6 Aspetti critici

- Il capitolato richiede l'apprendimento di molti servizi, sconosciuti dai membri del gruppo, che si basano su argomenti non conosciuti (come il machine learning^G) e non affrontati nel corso di laurea triennale. Quindi gran parte del tempo andrebbe dedicato allo studio di questi concetti e strumenti, ed infine di applicare queste conoscenze;
- Oltre al tempo impiegato per l'apprendimento di questi strumenti c'è da tenere in considerazione il tempo per addestrare la parte di machine learning^G.

8.7 Conclusioni

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