



>Three Way Milkshake_

Studio di Fattibilità

Three Way Milkshake - Progetto "PORTACS"

threewaymilkshake@gmail.com

Versione	0.1.0
Stato	Non approvato
Uso	Interno
Approvazione	Nome approvatore
Redazione	Nome redattore
Verifica	Tessari Andrea
Destinatari	Three Way Milkshake Prof. Tullio Vardanega Prof. Riccardo Cardin

Descrizione

Questo documento presenta un breve studio dei capitolati proposti, descrivendone caratteristiche ed aspetti principali, per orientare il gruppo ad una scelta.

Registro delle modifiche

Versione	Descrizione	Data	Nominativo	Ruolo
0.7.4	Verifica sezione § 2	2021-01-08	Giada Zuccolo	Verificatore
0.7.3	Verifica sezione § 1 e § 3.2	2021-01-08	Simone De Renzis	Verificatore
0.7.2	Verifica sezione § 4.3	2021-01-07	Tessari Andrea	Verificatore
0.7.1	Correzione struttura sotto sezioni ed aggiunta collegamento a materiale capitolati	2020-12-12	Greggio Nicolò	Redattore
0.7.0	Stesura della sezione § 4.2	2020-12-10	Greggio Nicolò	Redattore
0.6.0	Stesura della sezione § 2	2020-12-05	Tessari Andrea	Redattore
0.5.0	Stesura della sezione § 7	2020-12-02	Tessari Andrea	Redattore
0.4.0	Stesura della sezione § 1	2020-12-01	Tessari Andrea	Redattore
0.3.0	Stesura della sezione § 4.1	2020-11-30	Zuccolo Giada	Redattore
0.2.0	Stesura della sezione § 3.6	2020-11-30	Chiarello Sofia	Redattore
0.1.0	Stesura della sezione § 3.1	2020-11-29	De Renzis Simone	Redattore
0.0.1	Impostazione struttura del documento	2020-11-20	De Renzis Simone	Redattore



Indice

1	Introduzione	4
1.1	Scopo del documento	4
1.2	Riferimenti	4
1.2.1	Normativi	4
1.2.2	Informativi	4
2	Analisi del capitolato scelto	5
2.1	Capitolato C5 - Portacs	5
2.1.1	Informazioni generali	5
2.1.2	Descrizione del capitolato	5
2.1.3	Finalità del progetto	5
2.1.4	Tecnologie interessate	6
2.1.5	Aspetti positivi	6
2.1.6	Criticità e fattori di rischio	6
2.1.7	Conclusioni	6
3	Analisi dei capitolati rimanenti	7
3.1	Capitolato C1 - BlockCOVID	7
3.1.1	Informazioni generali	7
3.1.2	Descrizione del capitolato	7
3.1.3	Finalità del progetto	7
3.1.4	Tecnologie interessate	7
3.1.5	Aspetti positivi	8
3.1.6	Criticità	8
3.1.7	Conclusioni	8
3.2	Capitolato C2 - Emporio Lambda	9
3.2.1	Informazioni generali	9
3.2.2	Descrizione del capitolato	9
3.2.3	Finalità del progetto	9
3.2.4	Tecnologie interessate	9
3.2.5	Aspetti positivi	9
3.2.6	Criticità	10
3.2.7	Conclusioni	10
3.3	Capitolato C3 - GDP - Gathering Detection Platform	11
3.3.1	Informazioni generali	11
3.3.2	Descrizione del capitolato	11
3.3.3	Finalità del progetto	11
3.3.4	Tecnologie interessate	12
3.3.5	Aspetti positivi	12
3.3.6	Criticità	12
3.3.7	Conclusioni	12
3.4	Capitolato C4 - HD Viz, visualizzazione di dati con molte dimensioni	13
3.4.1	Informazioni generali	13
3.4.2	Descrizione del capitolato	13
3.4.3	Finalità del progetto	13
3.4.4	Tecnologie interessate	13
3.4.5	Aspetti positivi	14



3.4.6	Criticità e fattori di rischio	14
3.4.7	Conclusioni	14
3.5	Capitolato C6 - RGP: Realtime Gaming Platform	15
3.5.1	Informazioni generali	15
3.5.2	Descrizione del capitolato	15
3.5.3	Finalità del progetto	15
3.5.4	Tecnologie interessate	15
3.5.5	Aspetti positivi	16
3.5.6	Criticità e fattori di rischio	16
3.5.7	Conclusioni	16
3.6	Capitolato C7 - Soluzioni di sincronizzazione Desktop	17
3.6.1	Informazioni generali	17
3.6.2	Descrizione del capitolato	17
3.6.3	Finalità del progetto	17
3.6.4	Tecnologie interessate	17
3.6.5	Aspetti positivi	17
3.6.6	Criticità e fattori di rischio	18
3.6.7	Conclusioni	18



1 Introduzione

1.1 Scopo del documento

In questo documento si riporta l'analisi di ogni capitolato effettuata nel contesto della scelta del capitolato per cui concorrere nella relativa gara d'appalto. Per ogni capitolato, una breve descrizione del progetto proposto e delle sue finalità, le tecnologie da impiegare e le conclusioni del gruppo sugli aspetti positivi e sulle criticità. La scelta è ricaduta sul capitolato C5-PORTACS: piattaforma di mobilità autonoma.

1.2 Riferimenti

1.2.1 Normativi

- NORME DI PROGETTO v1.0.0 : per qualsiasi convenzione sulla nomenclatura degli elementi presenti all'interno del documento;
- Regolamento progetto didattico - slide del corso di Ingegneria del Software:
<https://www.math.unipd.it/~tullio/IS-1/2020/Dispense/P1.pdf>

1.2.2 Informativi

- GLOSSARIO v1.0.0: per la definizione dei termini (pedice G) e degli acronimi (pedice A) evidenziati nel documento;
- Capitolato d'appalto 1: BlockCOVID: supporto digitale al contrasto della pandemia
<https://www.math.unipd.it/~tullio/IS-1/2020/Progetto/C1.pdf>
- Capitolato d'appalto 2: EmporioLambda: piattaforma di e-commerce in stile Serverless
<https://www.math.unipd.it/~tullio/IS-1/2020/Progetto/C2.pdf>
- Capitolato d'appalto 3: GDP: Gathering Detection Platform
<https://www.math.unipd.it/~tullio/IS-1/2020/Progetto/C3.pdf>
- Capitolato d'appalto 4: HD Viz: visualizzazione di dati multidimensionali
<https://www.math.unipd.it/~tullio/IS-1/2020/Progetto/C4.pdf>
- Capitolato d'appalto 5: PORTACS: piattaforma di controllo mobilità autonoma
<https://www.math.unipd.it/~tullio/IS-1/2020/Progetto/C5.pdf>
- Capitolato d'appalto 6: RGP: Realtime Gaming Platform
[Link al capitolato 6](#)
- Capitolato d'appalto 7: SSD: soluzioni di sincronizzazione desktop
<https://www.math.unipd.it/~tullio/IS-1/2020/Progetto/C7.pdf>
- Software Engineering - Iam Sommerville - 10th Edition



2 Analisi del capitolato scelto

2.1 Capitolato C5 - Portacs

2.1.1 Informazioni generali

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

2.1.2 Descrizione del capitolato

Il capitolato C5 propone lo sviluppo di un software per il monitoraggio in tempo reale di varie unità e dei loro spostamenti in uno spazio creato dall'utente.

Ogni unità (rappresentabile da un robot cameriere, da un muletto o da un'auto a guida autonoma) dovrà raggiungere dei "Points Of Interest" (POI). Ognuna di esse avrà:

- Codice identificativo;
- Velocità massima;
- Posizione iniziale;
- Lista ordinata dei POI da attraversare.

Lo spazio sarà definito da una mappa inserita dall'utente e avrà la struttura di una scacchiera con dei vincoli sulla viabilità (percorrenze in parallelo e/o sensi unici) e ostacoli in base alla topologia fornita. Il software inoltre dovrà occuparsi di prevedere situazioni critiche, come il traffico ed eventuali collisioni esterne.

2.1.3 Finalità del progetto

Il progetto PORTACS si pone come obiettivo finale di dimostrare la fattibilità dello sviluppo di un software che permetta il coordinamento di unità che si muovono in uno spazio per raggiungere una lista ordinata di punti d'interesse. Questo implica che bisognerà sviluppare un progetto negli ambiti del real time monitoring e analysis, e del predictivity e real time decision making. Inoltre, esso affronta le problematiche del mondo della logistica e dell'ottimizzazione delle performance di consegna. Verranno quindi consegnati i seguenti documenti:

- Codice sorgente di quanto realizzato
- Docker file con la componente applicativa, rappresentante il motore di calcolo;
- Docker file con la componente applicativa, rappresentante il visualizzatore/monitor real time;
- Docker file, da istanziare N volte, rappresentante la singola unità;
- Docker file, da istanziare N volte, rappresentante il singolo pedone (facoltativo).



2.1.4 Tecnologie interessate

Le tecnologie interessate sono:

- **Diagrammi UML;**
- **Github o Bitbucket;**
- **Docker.**

2.1.5 Aspetti positivi

L'azienda mette a disposizione figure di diverso livello in modo tale da poter coprire nella maniera più appropriata le esigenze degli studenti. Inoltre il codice prodotto sarà reso disponibile al pubblico con licenza libera su Github o BitBucket alla fine del progetto.

2.1.6 Criticità e fattori di rischio

Per lo sviluppo di questo progetto, si andrà a lavorare con il multithreading e sarà necessario prendere delle decisioni per il paradigma di programmazione che si sceglierà di usare. Tutto questo, sommato al fatto che si deve anche individuare un algoritmo che risolvi il problema del Path Finding, potrebbe non essere immediato e portare al riscontro di diverse difficoltà.

2.1.7 Conclusioni

La disponibilità dell'azienda assieme alla curiosità scaturita dopo il seminario offerto hanno favorito la scelta di questo capitolato, già presa in considerazione visto l'argomento di interesse comune del gruppo riguardo gli argomenti di sviluppo proposti. Inoltre questo progetto aiuta a mettere in pratica le proprie conoscenze di ricerca operativa ed è infine molto utile acquisire competenze nel mondo Docker e nell'ambito del real time monitoring e analysis, e del predictivity e real time decision making.



3 Analisi dei capitolati rimanenti

3.1 Capitolato C1 - BlockCOVID

3.1.1 Informazioni generali

- **Nome:** BlockCOVID
- **Proponente:** Imola Informatica
- **Committente:** Prof. Tullio Vardanega, Prof. Riccardo Cardin

3.1.2 Descrizione del capitolato

Il software BlockCOVID si pone l'obiettivo di fornire un'infrastruttura che consenta il tracciamento del personale e della pulizia delle postazioni di lavoro all'interno di un'azienda. Mira infatti a favorire l'attuazione e amministrazione delle misure necessarie per la tutela della salute dei lavoratori nel contesto della pandemia da COVID-19.

3.1.3 Finalità del progetto

Il prodotto finale prevede due modalità di tracciamento:

- **Tracciamento delle presenze:** registrazione e monitoraggio in tempo reale delle presenze all'interno di un ambiente di lavoro (laboratorio informatico). Gli utenti devono avere la possibilità, in base alle disponibilità di postazioni, di effettuare prenotazioni tramite un applicativo per dispositivi mobili Android e IOS. Tramite lo stesso dispositivo segnaleranno la propria presenza una volta occupata la postazione ed eventualmente la sanificazione della stessa dopo l'uso. L'amministratore deve poter creare e gestire la struttura dell'ambiente, e visualizzarne in tempo reale lo stato di occupazione e sanificazione delle postazioni.
- **Tracciamento della pulizia delle postazioni:** dedicato agli addetti alle pulizie, il prodotto deve consentire di visualizzare le postazioni non sanificate e segnalare le postazioni sanificate in seguito alla pulizia delle stesse.

3.1.4 Tecnologie interessate

Java (versione 8 o superiori), Python o nodejs

- **Java, Python, Node.js:** linguaggi suggeriti dal proponente per lo sviluppo del server back-end che amministra l'infrastruttura di tracciamento;
- **Protocolli asincroni:** da utilizzare per gestire la comunicazione tra le componenti del sistema: i dispositivi mobili degli utenti, dell'amministratore e degli addetti alle pulizie con il server centrale;
- **Blockchain:** sistema che assicura la registrazione dei dati relativi alle sanificazioni conferendone valore legale;
- **Kubernetes IAAS, PAAS, Openshift, Rancher:** piattaforme suggerite dal proponente per il rilascio delle componenti del server e la gestione della scalabilità orizzontale.

Inoltre il proponente specifica i seguenti requisiti minimi:

- Fornire delle API che utilizzino tecnologie Rest o gRPC nell'ambito della comunicazione tra server e dispositivi;
- Studiare la tecnologia RFID o eventuali alternative da utilizzare per certificare la presenza di persone nelle postazioni, analizzandone il consumo energetico;
- Effettuare test di integrazione e unità (copertura $\geq 80\%$) delle componenti applicative;
- Assicurare l'integrità del sistema tramite test end-to-end.

3.1.5 Aspetti positivi

Il progetto si pone in un contesto quanto mai attuale e di grande interesse generale. Trova infatti soluzione a problematiche diffuse e richieste da molte realtà, fornendo un caso di studio molto interessante per quanto riguarda i sistemi distribuiti e real-time. Le tecnologie impiegate e richieste sono valide, e il loro studio consentirebbe ai membri del gruppo di acquisire competenze facilmente spendibili nel mondo del lavoro.

3.1.6 Criticità

I requisiti del proponente sono dettagliati e appaiono abbastanza esigenti; le tecnologie da impiegare, per quanto interessanti, configurano un carico di lavoro consistente.

In generale, il capitolato ha riscosso un discreto interesse da parte del gruppo ma non si è configurato come prima scelta a favore di capitolati riguardanti temi considerati più interessanti, tecnologie più stimolanti e carichi di lavoro più sostenibili.

3.1.7 Conclusioni

Nonostante l'elevato interesse e attenzione che questi sistemi hanno attirato, il gruppo ha deciso di procedere a favore di altri capitolati. Per quanto attuale, il progetto si presuppone essere completato ed eventualmente utilizzato in un periodo in cui le problematiche legate alla pandemia dovrebbero – sperabilmente – risultare fortemente mitigate dall'avvento delle campagne vaccinali. Non è comunque da escludere che, visto l'enorme impatto di questa emergenza, forme di tracciamento non debbano continuare ad essere applicate anche in condizioni di normalità.



3.2 Capitolato C2 - Emporio Lambda

3.2.1 Informazioni generali

- **Nome:** Emporio Lambda
- **Proponente:** Red Babel
- **Committente:** Prof. Vardanega Tullio, Prof. Cardin Riccardo

3.2.2 Descrizione del capitolato

L'idea alla base di Emporio Lambda è quella di costruire una piattaforma di e-commerce che si basi interamente su tecnologie serverless.

3.2.3 Finalità del progetto

Il prodotto in questione dovrà fornire due insiemi di funzionalità principali:

- una orientata ai clienti (pagina principale, lista e descrizione prodotti, carrello degli acquisti, pagamento, gestione account);
- l'altra alle funzioni di back office (tutto ciò che può servire agli impiegati, ossia gestione di contabilità, inventario, ordini, giacenza, distribuzione, spedizioni...).

L'esecuzione del sistema avviene su architetture serverless che possono incorporare BaaS di terze parti, o che contengono codice proprietario eseguito su container effimeri (che hanno una durata limitata ad una singola invocazione) su piattaforme FaaS. SPA è l'approccio da adottare per la realizzazione dell'applicazione web. Tramite queste caratteristiche, il sistema nel suo complesso dovrebbe beneficiare di una riduzione in termini di costi di operazioni, complessità e tempi di consegna.

3.2.4 Tecnologie interessate

Il linguaggio principale che dev'essere adottato è Typescript. Segue un elenco di altre tecnologie a supporto, obbligatorie o consigliate dal proponente:

- **AWS Lambda:** piattaforma di calcolo serverless basata su eventi, parte della suite di servizi web forniti da Amazon;
- **CloudFormation:** strumento consigliato per il rilascio di infrastrutture AWS;
- **Serverless Framework:** framework Node.js da utilizzare per la parte back end;
- **Next.js:** framework da usare per la parte front end;
- **CloudWatch o Datadog:** per implementare il sistema di monitoring;
- **Contentful:** CMS suggerito per l'implementazione di una parte opzionale.

3.2.5 Aspetti positivi

Emporio Lambda rappresenta sicuramente una realtà familiare a tutti, in quanto sono sempre più numerose e diffuse le piattaforme per gli acquisti online. Inoltre offre l'opportunità di lavorare con tecnologie nuove e all'avanguardia, adottando paradigmi diversi a quelli tradizionali e fornendo delle ottime opportunità di apprendimento.



3.2.6 Criticità

I requisiti imposti sembrano essere molto vincolanti, e le possibilità di scelte libere attuabili in fase di sviluppo appaiono limitate. Le piattaforme di e-commerce sono un dominio estremamente diffuso e conosciuto e, almeno dal lato dell'utente, non rappresentano una novità per i membri del team: altri capitolati sono stati giudicati più stimolanti.

3.2.7 Conclusioni

Questo capitolato non ha suscitato fin dall'inizio interesse nel gruppo. Non essendoci inoltre stato un seminario di approfondimento, non è stato possibile rivedere in chiave diversa le tematiche coinvolte, dunque l'opinione generale interna è rimasta invariata. Il gruppo si orienta verso un'altra scelta.



3.3 Capitolato C3 - GDP - Gathering Detection Platform

3.3.1 Informazioni generali

- **Nome:** GDP - Gathering Detection Platform
- **Proponente:** SyncLab
- **Committente:** Prof. Vardanega Tullio, Prof. Cardin Riccardo

3.3.2 Descrizione del capitolato

In seguito al lockdown generato a causa del virus COVID-19, si sono venute a creare situazioni, lungo l'arco della vita quotidiana, in cui le persone entrano in contatto tra di loro, generando assembramento.

L'informatica può aiutare a governare meglio questa situazione.

Il software GDP (Gathering Detection Platform) è costituito da una piattaforma che rappresenta, mediante visualizzazione grafica, le zone potenzialmente a rischio di assembramento col fine di prevenirne di nuove, attraverso sensori e varie sorgenti di dati.

3.3.3 Finalità del progetto

Il software finale prevede l'acquisizione di informazione da sensoristica e da altre sorgenti:

- **Sensori:**
 - telecamere
 - dispositivi contapersone
 - etc.
- **Sorgenti varie e eterogenee:**
 - flussi di prenotazioni Uber
 - orari dei mezzi di trasporto con capienze medie per corsia (autobus, metro, treno)
 - etc.

Gli utilizzatori del software potranno vedere una rappresentazione delle zone a rischio (attuali o possibilmente in futuro), attraverso una heat-map.

In questo modo potranno accedere alla situazione globale dei vari flussi:

- in tempo reale, con bassa latenza
- flussi previsti in futuro.
- flussi vecchi raccolti e storicizzati.



3.3.4 Tecnologie interessate

- **Java, Angular:** linguaggi suggeriti dal proponente per lo sviluppo del server back-end e della componente Web Application del sistema;
- **framework Leaflet:** framework da utilizzare per la gestione delle mappe, ad esempio heatmap;
- **protocolli asincroni:** per la comunicazione tra le varie componenti;
- **pattern Publisher/Subscriber e protocollo MQTT:** (Message Queue Telemetry Transport), consigliato per essere open, di facile implementazione, di ampia diffusione in applicazioni IoT e M2M.

Inoltre il proponente specifica i seguenti requisiti minimi:

- **responsive** deve sempre rispondere a una richiesta di servizio, anche in caso di guasto;
- **resilient** I servizi devono essere sempre ripristinabili a seguito di guasti;
- **elastic** I servizi devono scalare in base alla domanda;
- **message-driven** I servizi devono rispondere al mondo, senza controllarlo.

Un sistema che soddisfa questi requisiti è definito un Sistema Reattivo.

3.3.5 Aspetti positivi

Il progetto si pone in un contesto molto tecnologico, di grande interesse per l'attuale mercato, molto sensibile e maturo su questa problematica.

Anche le tecnologie impiegate sono attuali e valide, il cui studio porterebbe all'acquisizione di competenze molto utili in ambito lavorativo.

3.3.6 Criticità

Nonostante il capitolato abbia riscosso un buon interesse da parte del gruppo, le tecnologie da impiegare, in quanto alcune nuove e diverse da qualsiasi altro linguaggio imparato durante il percorso di studi, gravano molto sulla facilità di realizzazione del progetto.

In particolare le criticità principali sono:

- **Leaflet:** Tecnologia relativa alle heat-map e framework Leaflet;
- **Linguaggio Angular:** Linguaggio di programmazione che viene eseguito su web browser;
- **Protocolli asincroni, Pattern Publisher/Subscriber e protocollo MQTT:** Varie tecnologie, alcune sconosciute, che servono alla realizzazione del progetto.

3.3.7 Conclusioni

Nonostante l'elevata attenzione che questi sistemi hanno attirato, il gruppo non ha preso in considerazione questo capitolato in quanto molti altri gruppi di lavoro avevano già segnalato un alto interesse per questo progetto.



3.4 Capitolato C4 - HD Viz, visualizzazione di dati con molte dimensioni

3.4.1 Informazioni generali

- **Nome:** *HD Viz, visualizzazione di dati con molte dimensioni;*
- **Proponente:** *Zucchetti;*
- **Committente:** *Prof. Tullio Vardanega e Prof. Riccardo Cardin;*

3.4.2 Descrizione del capitolato

Il capitolato propone la realizzazione di un'applicazione chiamata "HD Viz", utile per l'interpretazione dell'analisi di dati a molte dimensioni, raccolti sfruttando tecniche di statistica, machine learning e intelligenza artificiale. Richiede quindi lo sviluppo di un'applicazione per la visualizzazione di dati con molte dimensioni (ovvero i dati da visualizzare dovranno poter avere almeno fino a 15 dimensioni e deve però essere possibile visualizzare anche dati con meno dimensioni) a supporto della fase esplorativa di questi dati da interpretare, forniti al sistema di visualizzazione sia con query che da file in formato CSV. La visualizzazione di dati con il software "HD Viz" deve essere presentata almeno con i grafici "Scatter plot Matrix", "Force Field", "Heat Map" e "Proiezione Lineare Multi Asse". In particolare dovrà ordinare i punti nel grafico "Heat Map" per evidenziare i "cluster" presenti nei dati. Inoltre sono proposti altri requisiti valutati come opzionali.

3.4.3 Finalità del progetto

L'obiettivo è sviluppare un'applicazione per la visualizzazione di dati, di molte dimensioni, raccolti con la finalità di conseguire poi con un'analisi dei dati. Il software "HD Viz" dovrà presentare almeno le visualizzazioni con:

- Scatter plot Matrix (fino ad un massimo di 5 dimensioni);
- Force Field;
- Heat Map (obbligatoriamente dovrà organizzare i punti per evidenziare i "cluster" presenti nei dati);
- Proiezione Lineare Multi Asse;

3.4.4 Tecnologie interessate

L'azienda indica lo sviluppo dell'applicazione "HD Viz" con:

- **HTML/CSS/JavaScript** utilizzando la **libreria D3.js**;
- **Java con server Tomcat** o **JavaScript con server Node.js** per la parte server di supporto alla presentazione nel browser e alle query ad un database SQL o NoSQL.



3.4.5 Aspetti positivi

Questo capitolato propone lo sviluppo un progetto che si inserisce in un ambito relativamente nuovo, cioè la raccolta, la gestione, la memorizzazione e la visualizzazione di grandi quantità di dati, i Big Data, al giorno d'oggi utilizzabili in tutti i vari campi. Lo sviluppo di questo progetto avrebbe permesso l'approfondimento di questo nuovo ma più complesso ambito, utilizzando delle tecnologie non totalmente nuove alla gran parte dei membri del gruppo.

3.4.6 Criticità e fattori di rischio

Sebbene le tecnologie interessate per lo sviluppo dell'applicazione non siano totalmente nuove alla gran parte dei membri del gruppo, applicarle per il soddisfacimento delle richieste obbligatorie e dei requisiti opzionali, potrebbe portare ad una situazione svantaggiosa in quanto, sia le richieste obbligatorie sia i requisiti opzionali, siano più complessi da sviluppare e nel complesso anche numerose.

3.4.7 Conclusioni

Malgrado l'ambito abbastanza stimolante, il capitolato ha riscosso un misurato interesse da parte del gruppo. Per questo motivo si è scelto di selezionare altri capitoli che riscontravano maggiore curiosità nel gruppo e che si avvalgono di tecnologie nuove.

3.5 Capitolato C6 - RGP: Realtime Gaming Platform

3.5.1 Informazioni generali

- **Nome:** *RGP: Realtime Gaming Platform;*
- **Proponente:** *Zero12 s.r.l.;*
- **Committente:** *Prof. Tullio Vardanega e Prof. Riccardo Cardin;*

3.5.2 Descrizione del capitolato

Il capitolato proposto prevede la realizzazione di un videogioco a scorrimento verticale, fruibile da dispositivi mobile, con la possibilità di giocare in real time multiplayer. La modalità del gioco è simile ad Aero Fighters, mentre la grafica è scelta del gruppo di lavoro o fornita da Zero12. Le modalità di gioco sono:

- singleplayer
- multiplayer

La sfida tra più giocatori rappresenta il cuore del progetto ed è anche la componente di sviluppo principale. Il gioco è ad eliminazione, l'ultimo giocatore non eliminato vince. Durante la partita deve essere possibile vedere, in tempo reale, i movimenti del rivale ed è necessario sincronizzare eventuali nemici e power-up in modo tale che la sfida sia la medesima. L'interazione tra giocatori diversi è puramente visiva. La modalità singleplayer consiste in una serie infinita di livelli a difficoltà crescente. Il gioco termina quando l'utente ha concluso le vite oppure se non ha raccolto power-up sufficienti a mantenere il proprio oggetto attivo.

3.5.3 Finalità del progetto

Il progetto è finalizzato allo sviluppo di un'applicazione mobile superando dei vincoli quali:

- Ricerca delle tecnologie AWS per capire quale si può adattare meglio ad un gioco con requisiti di realtime, raccogliendo le motivazioni che supportano la scelta di una tecnologia rispetto ad un'altra.
- Implementazione della componente server-side.
- Implementazione del gioco per piattaforma mobile.

3.5.4 Tecnologie interessate

Le tecnologie interessate sono:

- **AWS** GameLift, Appsync oppure altre architetture serverless;
- **NodeJs**;
- **Swift/SwiftUI** oppure **Kotlin**;
- **GIT**;



3.5.5 Aspetti positivi

Questo progetto aiuta sia a familiarizzare con lo sviluppo di applicazioni per Android e per Ios che a creare connessioni in real time tra due o più dispositivi attraverso i servizi di Amazon AWS. L'azienda inoltre prevede corsi di formazioni sui vari servizi offerti dall'AWS.

3.5.6 Criticità e fattori di rischio

L'ambiente AWS è molto vasto ed è critica la scelta iniziale su quale servizio basarsi. Una scelta errata può portare ad una pessima connessione tra dispositivi e, di conseguenza, ad un non soddisfacimento del requisito di "real time" della parte multiplayer del videogioco.

3.5.7 Conclusioni

Anche se i corsi di formazioni proposti risultano allettanti, la maggior parte del gruppo di lavoro non è entusiasmato all'idea di realizzare un videogioco.



3.6 Capitolato C7 - Soluzioni di sincronizzazione Desktop

3.6.1 Informazioni generali

- **Nome:** *Soluzioni di sincronizzazione Desktop;*
- **Proponente:** *Zextras;*
- **Committente:** *Prof. Tullio Vardanega e Prof. Riccardo Cardin;*

3.6.2 Descrizione del capitolato

Il capitolato richiede di sviluppare un algoritmo di sincronizzazione Desktop solido ed efficiente in grado di garantire il salvataggio in cloud del lavoro, in modo da poter raggiungere in qualsiasi momento e da qualsiasi dispositivo il proprio lavoro. Inoltre deve essere sviluppata un'interfaccia multiplatforma per l'uso dell'algoritmo nei più importanti sistemi operativi esistenti (MacOs, Windows, Linux) senza richiedere all'utente l'installazione di ulteriori prodotti per il funzionamento.

3.6.3 Finalità del progetto

L'obiettivo è quello di creare questo algoritmo di sincronizzazione che funzioni in più piattaforme in grado di garantire il salvataggio in cloud del lavoro e contemporaneamente la sincronizzazione dei cambiamenti presenti in cloud. Inoltre viene richiesto l'utilizzo dell'algoritmo sviluppato per richiedere e fornire i cambiamenti ai contenuti in sincronizzazione verso il prodotto Zextras Drive. L'algoritmo deve avere le seguenti principali funzionalità:

- Configurazione ed autenticazione dell'utente;
- Gestione di cosa sincronizzare e di cosa ignorare nelle cartelle cloud;
- Gestione di cosa sincronizzare e di cosa ignorare nelle cartelle locali;
- Sincronizzazione costante dei cambiamenti, siano essi locali o remoti;
- Possibilità di modifica delle preferenze a posteriori;
- Sistema di notifica utente dei cambiamenti.

3.6.4 Tecnologie interessate

L'azienda esprime la necessità di sviluppare l'algoritmo per i più importanti sistemi operativi esistenti (MacOs, Windows, Linux) e consiglia l'utilizzo:

- **Qt Framework:** strumento basato sul linguaggio C++ orientato ad oggetti, da utilizzare per creare l'interfaccia poiché fortemente supportato e documentato;
- **Python:** strumento consigliato per lo sviluppo della Business Logic, linguaggio ad alto livello adatto allo sviluppo di applicazioni distribuite.

3.6.5 Aspetti positivi

Il progetto si pone in un contesto molto utilizzato sia da utenti privati che da aziende. Inoltre le tecnologie consigliate fanno già parte delle conoscenze di gran parte dei membri del gruppo e ci sono molti esempi, anch'essi ben conosciuti, a cui ispirarsi per la creazione dell'algoritmo (Dropbox, Google Drive, ect).



3.6.6 Criticità e fattori di rischio

Il confronto con tecnologie già esistenti di questo tipo può essere svantaggioso poiché si potrebbe creare un prodotto molto più inefficiente. Inoltre le richieste sono numerose e abbastanza complesse.

3.6.7 Conclusioni

Si è deciso di puntare su altri capitolati poiché questo non ha suscitato grande interesse per quanto riguarda la tematica che affronta e il settore interessato.



>Three Way
Milkshake_

Studio di Fattibilità
