



>Three Way Milkshake_

Studio di Fattibilità

Three Way Milkshake - Progetto "PORTACS"

threewaymilkshake@gmail.com

Versione	1.0.0
Stato	Approvato
Uso	Interno
Approvazione	Greggio Nicolò
Redazione	Three Way Milkshake
Verifica	Three Way Milkshake
Destinatari	Three Way Milkshake Prof. Vardanega Tullio Prof. Cardin Riccardo

Descrizione

Questo documento presenta un breve studio dei [capitolati](#)_G proposti, descrivendone caratteristiche ed aspetti principali, per orientare il gruppo ad una scelta.

Registro delle modifiche

Versione	Descrizione	Data	Nominativo	Ruolo
1.0.0	Approvazione documento	2021-01-10	Greggio Nicolò	Responsabile
0.9.5	Verifica sezione § 3.4 e 3.6	2021-01-09	Crivellari Alberto	Verificatore
0.9.4	Verifica sezione § 3.5	2021-01-09	Greggio Nicolò	Verificatore
0.9.3	Verifica sezione § 3.1	2021-01-09	Chiarello Sofia	Verificatore
0.9.2	Verifica sezione § 2	2021-01-08	Zuccolo Giada	Verificatore
0.9.1	Verifica sezione § 1 e § 3.2	2021-01-08	De Renzis Simone	Verificatore
0.9.0	Verifica sezione § 3.3	2021-01-07	Tessari Andrea	Verificatore
0.8.0	Stesura della sezione § 2	2020-12-06	Tessari Andrea	Redattore
0.7.0	Stesura della sezione § 3.3	2020-12-05	Crivellari Alberto	Redattore
0.6.1	Correzione struttura sotto sezioni ed aggiunta collegamento a materiale capitolati_G	2020-12-05	Greggio Nicolò	Redattore
0.6.0	Stesura della sezione § 3.4	2020-12-04	Greggio Nicolò	Redattore
0.5.0	Stesura della sezione § 3.5	2020-12-03	Tessari Andrea	Redattore
0.4.0	Stesura della sezione § 1	2020-12-01	Tessari Andrea	Redattore
0.3.0	Stesura della sezione § 3.2	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	2020-11-28	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_G 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_G 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	15
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
Lista degli Acronimi		19
Glossario dei Termini		19



1 Introduzione

1.1 Scopo del documento

In questo documento si riporta l'analisi di ogni [capitolato_G](#) effettuata nel contesto della scelta del [capitolato_G](#) per cui concorrere nella relativa gara d'appalto. Per ogni [capitolato_G](#), una breve descrizione del [progetto_G](#) 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_G](#) C5-PORTACS: piattaforma di mobilità autonoma.

1.2 Riferimenti

1.2.1 Normativi

- NORME DI [PROGETTO_G](#) v1.0.0 : per qualsiasi convenzione sulla nomenclatura degli elementi presenti all'interno del documento;
- [Regolamento progetto_G didattico - slide del corso di Ingegneria del Software](#)

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](#)
- [Capitolato d'appalto 2: EmporioLambda: piattaforma di e-commerce in stile Serverless](#)
- [Capitolato d'appalto 3: GDP: Gathering Detection Platform](#)
- [Capitolato d'appalto 4: HD Viz: visualizzazione di dati multidimensionali](#)
- [Capitolato d'appalto 5: PORTACS_A: piattaforma di controllo mobilità autonoma](#)
- [Capitolato d'appalto 6: RGP: Realtime Gaming Platform](#)
- [Capitolato d'appalto 7: SSD: soluzioni di sincronizzazione desktop](#)
- Software Engineering - Iam Sommerville - 10th Edition

2 Analisi del **capitolato_G** scelto

2.1 Capitolato C5 - Portacs

2.1.1 Informazioni generali

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

2.1.2 Descrizione del capitolato

Il **capitolato_G** 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_A** 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_G** **PORTACS_A** 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_G** 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_G](#).

2.1.6 Criticità e fattori di rischio

Per lo sviluppo di questo [progetto_G](#), 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 risolve 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_G](#), già presa in considerazione visto l'argomento di interesse comune del gruppo riguardo gli argomenti di sviluppo proposti. Inoltre questo [progetto_G](#) 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**_G rimanenti

3.1 Capitolato C1 - BlockCOVID

3.1.1 Informazioni generali

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

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 *Node.js*.

- **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_G** minimi:

- Fornire delle **API_A** 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_G** 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_G** 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_G** ha riscosso un discreto interesse da parte del gruppo ma non si è configurato come prima scelta a favore di **capitolati_G** 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_G**. Per quanto attuale, il **progetto_G** si presuppone essere completato ed eventualmente utilizzato in un **periodo_G** in cui le problematiche legate alla pandemia dovrebbero, auspicabilmente, 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_A** di terze parti, o che contengono codice proprietario eseguito su container effimeri (che hanno una durata limitata ad una singola invocazione) su piattaforme **FaaS_A**. **SPA_A** 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_A** 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](#)_G imposti sembrano essere molto vincolanti, e le possibilità di scelte libere attuabili in [fase](#)_G 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](#)_G sono stati giudicati più stimolanti.

3.2.7 Conclusioni

Questo [capitolato](#)_G 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_G** 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_G** è definito un Sistema Reattivo.

3.3.5 Aspetti positivi

Il **progetto_G** 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_G** 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_G**.

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_G**.

3.3.7 Conclusioni

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

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. Vardanega Tullio e Prof. Cardin Riccardo;

3.4.2 Descrizione del capitolato

Il [capitolato_G](#) 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_G](#) 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_G](#) 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_G](#) propone lo sviluppo un [progetto_G](#) 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_G](#) 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_G](#) opzionali, potrebbe portare ad una situazione svantaggiosa in quanto, sia le richieste obbligatorie sia i [requisiti_G](#) opzionali, siano più complessi da sviluppare e nel complesso anche numerose.

3.4.7 Conclusioni

Malgrado l'ambito abbastanza stimolante, il [capitolato_G](#) ha riscosso un misurato interesse da parte del gruppo. Per questo motivo si è scelto di selezionare altri [capitolati_G](#) 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_G** proposto prevede la realizzazione di un videogioco a scorrimento verticale, fruibile da dispositivi mobile, con la possibilità di giocare in multiplayer real time. La modalità di gioco, per quanto riguarda il gameplay, sarà simile ad Aero Fighters¹, mentre la grafica potrà essere scelta liberamente dal gruppo o fornita dal team di Zero12.

3.5.3 Finalità del progetto

Sono previste le modalità di gioco single e multiplayer, ma il cuore del **progetto_G** sono le sfide tra più giocatori: il gioco è ad eliminazione, l'ultimo rimasto vince. Durante la partita deve essere possibile vedere, in tempo reale, i movimenti del rivale e per garantire l'equità bisognerà prevedere la sincronizzazione di eventuali nemici e powerup. La modalità single player invece 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. Il **progetto_G** 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_G** di realtime, raccogliendo le motivazioni a supporto della scelta;
- realizzazione della componente server-side;
- implementazione del gioco per piattaforma mobile.

3.5.4 Tecnologie interessate

Le tecnologie interessate sono:

- **AWS:** per la realizzazione dei servizi per la gestione dei giochi;
- **NodeJs:** per lo sviluppo di codice server side;
- **Swift o Kotlin:** per l'implementazione delle applicazioni mobile, rispettivamente per iOS e Android;
- **Git:** per il versionamento del codice sviluppato.

3.5.5 Aspetti positivi

Il **capitolato_G** permette di entrare nel mondo dello sviluppo mobile nativo e della gestione di comunicazioni real time fra più dispositivi. Inoltre consente anche di familiarizzare con le tecnologie serverless anche tramite i servizi di Amazon AWS, per i quali l'azienda proponente prevede corsi di formazione.

¹Esempio Aero Fighters su youtube: <https://youtu.be/5Phj-735p30?t=28>



3.5.6 Criticità e fattori di rischio

L'ambiente AWS è molto vasto ed è critica la scelta iniziale del servizio sul quale basarsi non avendo alcuna esperienza pregressa: una scelta errata può portare ad una realizzazione non ottimale, rischiando così di non soddisfare pienamente i [requisiti](#)_G real time della parte multiplayer del videogioco. Inoltre lo sviluppo di applicazioni mobile native è già un compito arduo di per sé, per cui aggiungere comunicazioni real time e sincronizzazioni tra dispositivi diversi aumenta ulteriormente il grado di difficoltà e le possibilità d'errore.

3.5.7 Conclusioni

I corsi di formazione offerti dall'azienda potrebbero essere interessanti, ma valutata l'elevata difficoltà e lo scarso interesse da parte del gruppo, la scelta non si pone qui.



3.6 Capitolato C7 - Soluzioni di sincronizzazione Desktop

3.6.1 Informazioni generali

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

3.6.2 Descrizione del capitolato

Il **capitolato_G** 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_G** 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](#)_G poiché questo non ha suscitato grande interesse per quanto riguarda la tematica che affronta e il settore interessato.



Lista degli Acronimi

API Application Programming Interface. [8](#)

BaaS Backend as a Service. [9](#)

CMS Content Management System. [9](#)

FaaS Functions as a Service. [9](#)

POI Point Of Interest. [5](#), [19](#)

PORTACS [POI](#) Oriented Real Time Anti Collision System. [4](#), [5](#)

SPA Single Page Application. [9](#)

Glossario dei Termini

capitolato Documento tecnico redatto dal cliente in cui vengono specificati i vincoli contrattuali (prezzo e scadenze) per lo sviluppo di un determinato prodotto software. Viene presentato in un bando d'appalto per trovare qualcuno che possa svolgere il lavoro richiesto. [1](#), [2](#), [4–8](#), [10](#), [12–15](#), [17](#), [18](#)

fase Segmento temporale contiguo che presuppone uno stazionamento in uno stato o in una transizione di ciclo di vita. [10](#), [13](#)

periodo Nel contesto del documento, l'intervallo di tempo che intercorre tra due revisioni successive. [8](#)

progetto Insieme di attività che devono raggiungere determinati obiettivi a partire da determinate specifiche; hanno una data d'inizio e una data di fine fissate, dispongono di risorse limitate e consumano tali risorse nel loro svolgersi. [4–6](#), [8](#), [12](#), [14](#), [15](#), [17](#)

requisito Esistono due interpretazioni principali di un requisito

- dal lato del bisogno (ovvero il cliente/utente) è la capacità necessaria a un utente per risolvere un problema o raggiungere un obiettivo
- dal lato della soluzione (ovvero lo sviluppatore) è la capacità che deve essere posseduta (o condizione che deve essere soddisfatta) da un sistema per adempiere a un obbligo

. [8](#), [10](#), [12–16](#)