



# SpaghettiCode

spaghetti.code.g6@gmail.com

## STUDIO DI FATTIBILITÀ

<b>Versione</b>	v1.0.0
<b>Approvazione</b>	Paparazzo Giorgia
<b>Redazione</b>	Rizzo Stefano Contro Daniel Eduardo Fichera Jacopo Pagotto Manuel
<b>Verifica</b>	Rizzo Stefano Masevski Martin Kostadinov Samuel Pagotto Manuel Paparazzo Giorgia
<b>Uso</b>	Interno
<b>Destinato a</b>	prof. Vardanega Tullio prof. Cardin Riccardo SpaghettiCode

### Descrizione

Questo documento ha lo scopo di descrivere l'analisi dei capitoli d'appalto realizzata dal gruppo al fine di valutarne la fattibilità

## Registro delle modifiche

Versione	Nominativo	Ruolo	Data	Descrizione
v1.0.0	Paparazzo Giorgia	Responsabile	2021-01-01	Approvazione documento
v0.6.0	Rizzo Stefano	Verificatore	2020-12-19	Verifica Capitolato C6 - RGP e Capitolato C7 - SSD
v0.5.0	Masevski Martin	Verificatore	2020-12-19	Verifica Capitolato C5 - PORTACS
v0.4.0	Paparazzo Giorgia	Verificatore	2020-12-19	Verifica Capitolato C3 - GDP
v0.3.0	Pagotto Manuel	Verificatore	2020-12-19	Verifica Capitolato C2 - EmporioLambda
v0.2.0	Rizzo Stefano	Verificatore	2020-12-19	Verifica Capitolato C1 - BlockCOVID
v0.1.0	Kostadinov Samuel	Verificatore	2020-12-17	Verifica Capitolato C4 - HD Viz
v0.0.8	Pagotto Manuel	Analista	2020-12-17	Stesura Capitolato C7 - SSD
v0.0.7	Fichera Jacopo	Analista	2020-12-17	Stesura Capitolato C6 - RGP
v0.0.6	Fichera Jacopo	Analista	2020-12-17	Stesura Capitolato C5 - PORTACS
v0.0.5	Paparazzo Giorgia	Responsabile	2020-12-18	Stesura Capitolato C4 - HD Viz
v0.0.4	Masevski Martin	Amministratore	2020-12-17	Stesura Capitolato C3 - GDP
v0.0.3	Rizzo Stefano	Analista	2020-12-17	Stesura Capitolato C2 - EmporioLambda
v0.0.2	Contro Daniel Eduardo	Analista	2020-12-17	Stesura Capitolato C1 - BlockCOVID
v0.0.1	Pagotto Manuel	Analista	2020-12-12	Creazione del documento



# Indice

<b>1</b>	<b>Introduzione</b>	<b>1</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>2</b>	<b>Capitolato C4 - HD Viz</b>	<b>2</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	2
2.6	Rischi	2
2.7	Conclusioni	3
<b>3</b>	<b>Capitolato C1 - BlockCOVID</b>	<b>4</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>4</b>	<b>Capitolato C2 - EmporioLambda</b>	<b>6</b>
4.1	Informazioni generali	6
4.2	Descrizione	6
4.3	Finalità del progetto	6
4.4	Tecnologie interessate	6
4.5	Aspetti positivi	6
4.6	Rischi	7
4.7	Conclusioni	7
<b>5</b>	<b>Capitolato C3 - GDP</b>	<b>8</b>
5.1	Informazioni generali	8
5.2	Descrizione	8
5.3	Finalità del progetto	8
5.4	Tecnologie interessate	8
5.5	Aspetti positivi	9
5.6	Rischi	9
5.7	Conclusioni	9
<b>6</b>	<b>Capitolato C5 - PORTACS</b>	<b>10</b>
6.1	Informazioni generali	10
6.2	Descrizione	10
6.3	Finalità del progetto	10
6.4	Tecnologie interessate	10
6.5	Aspetti positivi	10
6.6	Rischi	10
6.7	Conclusioni	11



<b>7</b>	<b>Capitolato C6 - RGP</b>	<b>12</b>
7.1	Informazioni generali . . . . .	12
7.2	Descrizione . . . . .	12
7.3	Finalità del progetto . . . . .	12
7.4	Tecnologie interessate . . . . .	12
7.5	Aspetti positivi . . . . .	12
7.6	Rischi . . . . .	12
7.7	Conclusioni . . . . .	12
<b>8</b>	<b>Capitolato C7 - SSD</b>	<b>13</b>
8.1	Informazioni generali . . . . .	13
8.2	Descrizione . . . . .	13
8.3	Finalità del progetto . . . . .	13
8.4	Tecnologie interessate . . . . .	13
8.5	Aspetti positivi . . . . .	14
8.6	Rischi . . . . .	14
8.7	Conclusioni . . . . .	14



# 1 Introduzione

## 1.1 Scopo del documento

Lo scopo dello STUDIO DI FATTIBILITÀ è quello di fornire le motivazioni che hanno indotto alla scelta del capitolato<sub>C</sub> C4 *HD Viz*, e quindi con l'esclusione degli altri capitolati d'appalto proposti.

## 1.2 Glossario

Alcuni termini all'interno di questo documento possono risultare ambigui a secondo del contesto in cui sono utilizzati. Questi termini sono segnalati con un 'G' a pedice del termine ambiguo; nel documento GLOSSARIO v1.0.0 sono presenti questi termini con il loro significato specifico.

## 1.3 Riferimenti

### 1.3.1 Normativi

- **Norme di progetto:** NORME DI PROGETTO v1.0.0

### 1.3.2 Informativi

- **Capitolato d'appalto C1** - BlockCOVID: supporto digitale al contrasto della pandemia;
- **Capitolato d'appalto C2** - EmporioLambda: piattaforma di e-commerce in stile Serverless;
- **Capitolato d'appalto C3** - GDP: Gathering Detection Platform;
- **Capitolato d'appalto C4** - HD Viz: visualizzazione di dati multidimensionali;
- **Capitolato d'appalto C5** - PORTACS: piattaforma di controllo mobilità autonoma;
- **Capitolato d'appalto C6** - RGP: Realtime Gaming Platform;
- **Capitolato d'appalto C7** - SSD: soluzioni di sincronizzazione desktop.



## 2 Capitolato C4 - HD Viz

### 2.1 Informazioni generali

**Nome:** HD Viz: visualizzazione di dati multidimensionali.

**Proponente:** Zucchetti S.p.A.

**Committente:** prof. Tullio Vardanega e prof. Riccardo Cardin.

### 2.2 Descrizione

Il capitolato richiede lo sviluppo di una web application<sub>G</sub> che abbia come scopo la traduzione di dati con molte dimensioni in grafici che aiutino l'utente a trarre delle interpretazioni e conclusioni. Questi dati dovranno essere inseriti tramite file CSV oppure ottenuti tramite query da un database.

### 2.3 Finalità del progetto

L'azienda richiede che l'applicazione presenti almeno le seguenti visualizzazioni:

- **Scatter plot matrix:** è una delle visualizzazioni facilmente ottenibili con D3.js, ed è la presentazione a riquadri disposti a matrice di tutte le combinazioni di scatter plot;
- **Force field:** è un grafico<sub>G</sub> che interpreta le distanze nello spazio a molte dimensioni in forze di attrazione e repulsione tra i punti proiettati nello spazio bidimensionale o tridimensionale;
- **Heat map:** trasforma la distanza tra i punti i colori più o meno intensi, facendo così capire quali oggetti sono vicini tra loro e quali sono distanti;
- **Proiezione lineare multi-asse:** si occupa di posizionare i punti dello spazio multidimensionale in un piano cartesiano, rappresentando a 2 dimensioni anche dati con molte più dimensioni.

### 2.4 Tecnologie interessate

Il capitolato prevede quindi lo sviluppo di:

- Una parte di HTML e CSS che dia una struttura e una presentazione all'applicazione;
- Un database che abbia la possibilità di memorizzare i dati in ingresso;
- Una parte di Java con server Tomcat o in JavaScript con server Node.js per comunicare col server;
- Una parte di JavaScript che visualizzi i dati multidimensionali con l'utilizzo di D3.js, una libreria JavaScript;
- Una parte di JavaScript che si occupi di caricare i dati da file CSV;

### 2.5 Aspetti positivi

Gli aspetti positivi che si sono evidenziati sono la possibilità di approfondire la conoscenza di JavaScript tramite l'utilizzo di D3.js; il gruppo ritiene molto utile poter studiare e applicare direttamente uno dei linguaggi più richiesti ed utilizzati nel mondo del lavoro, inoltre sono richiesti alcuni linguaggi che verranno affrontati con Tecnologie Web.

### 2.6 Rischi

Alcuni aspetti negativi invece sono lo studio di algoritmi matematici complessi lontani dalla nostra preparazione accademica, quali ad esempio quelli che trattano le distanze tra i dati. Inoltre, è richiesto molto intuito e capacità interpretative per poter capire quali siano i grafici più adeguati o esplicativi per determinati tipi di dato.



## 2.7 Conclusioni

Il gruppo ha accolto con entusiasmo il progetto proposto dalla *Zucchetti S.p.A.*, soprattutto a seguito del seminario esplicativo che ha stimolato curiosità ed interesse verso l'argomento. C'è stato inoltre molto fervore verso le molteplici possibilità esplorative che offre il progetto nella sua applicazione d'uso.



## 3 Capitolato C1 - BlockCOVID

### 3.1 Informazioni generali

**Nome:** BlockCOVID - supporto digitale al contrasto della pandemia;

**Proponente:** Imola Informatica;

**Committente:** prof. Tullio Vardanega e prof. Riccardo Cardin.

### 3.2 Descrizione

Il capitolato, data la situazione di pandemia globale, nasce dalla necessità di garantire ai fruitori dei luoghi di interesse il rispetto degli standard igienici. Esso ha come obiettivo, a fini di sicurezza e controllo, il tracciamento real-time, immutabile e certificato, di postazioni di lavoro in merito al loro stato di utilizzo e/o di pulizia. Date le responsabilità legali dei datori di lavoro nei confronti dei dipendenti, nel caso gli stessi datori di lavoro non mettano in atto le misure di sicurezza, è necessario che il tracciamento sia opponibile a terzi.

### 3.3 Finalità del progetto

Il prodotto finale dovrebbe essere costituito da due componenti principali:

- **Applicazione mobile:** l'applicazione dovrà essere necessariamente installata nello smartphone (Android o iOS) degli utilizzatori del servizio e permetterà l'individuazione delle postazioni libere e l'eventuale prenotazione di esse; la segnalazione dell'occupazione di una postazione in tempo reale tramite il tag RFID; la segnalazione di avvenuta pulizia di una postazione; la visualizzazione dello storico delle postazioni occupate e dello storico di quelle igienizzate;
- **Server:** ha il compito di garantire la gestione di più stanze e/o postazioni, in particolare deve permettere la visualizzazione dello stato di ciascuna postazione; la visualizzazione delle prenotazioni con le loro relative informazioni e il blocco delle prenotazioni di una specifica stanza in caso di bisogno; deve essere disponibile il tracciamento autenticato di tutti i cambiamenti di stato delle postazioni, nello specifico quelli relativi alla pulizia, di modo che espongano le corrispondenti informazioni. Inoltre, il server deve essere fornito di una UI che permetta l'interazione da parte dell'amministratore con il sistema.

### 3.4 Tecnologie interessate

Sebbene il capitolato non ponga vincoli sotto il punto di vista tecnologico, sono state consigliate una o più tecnologie per le varie componenti. Per lo sviluppo del server è stato consigliato di scegliere tra uno dei seguenti linguaggi:

- **Java:** linguaggio di programmazione ad alto livello, orientato agli oggetti<sub>G</sub> e a tipizzazione statica, progettato per rendere il più indipendente possibile il software dalla piattaforma hardware.
- **Python:** linguaggio di programmazione ad alto livello, interpretato, fortemente impiegato nello sviluppo di applicazioni distribuite; supporta diversi paradigmi tra cui quello orientato agli oggetti, quello funzionale e quello riflessivo.
- **Node.js:** runtime di JavaScript open source e multiplatforma, basata sul paradigma ad eventi per eseguire codice JavaScript.

Per le comunicazioni app mobile-server è stato consigliato l'utilizzo di protocolli asincroni: protocolli di trasmissione nei quali la sincronizzazione tra i dispositivi coinvolti avviene utilizzando i dati stessi.

Per le funzionalità esposte dal server è stato consigliato di utilizzare una tra le seguenti architetture:

- **API REST<sub>G</sub>:** particolare architettura di sistemi distribuiti che non prevede il concetto di sessione, bensì si basa sulla trasmissione di dati mediante il protocollo HTTP senza ulteriori livelli;
- **gRPC:** sistema open source per la chiamata di procedure remote che permette di instaurare connessioni client-server cross-platform per diversi linguaggi.





Per garantire l'opponibilità a terzi è stata suggerita la tecnologia Blockchain: struttura dati condivisa ed immutabile, la cui integrità è garantita; una modifica ad essa comporta l'invalidazione della stessa.

- **Kubernetes:** sistema open source di orchestrazione e gestione di container<sub>G</sub>;
- **Openshift:** piattaforma per applicazioni cloud<sub>G</sub> che permette di facilitarne lo sviluppo, la distribuzione e la scalabilità;
- **Rancher:** sistema open source per la gestione di cluster di container multipli.

Per l'identificazione fisica degli utenti è richiesta RFID: tecnologia per l'identificazione e la memorizzazione di informazioni inerenti ad oggetti, basata sull'utilizzo di etichette elettroniche (tag) interrogate a distanza da sistemi ad-hoc mediante radiofrequenze.

### 3.5 Aspetti positivi

Lo sviluppo di un'applicazione per garantire sicurezza nel luogo di lavoro o di studio è stato ritenuto oggettivamente di grande utilità da parte del gruppo. Le tecnologie coinvolte sono largamente utilizzate in ambito lavorativo e anche utili a fini didattici.

### 3.6 Rischi

Per poter effettuare il testing del sistema sarebbe necessario sviluppare un ambiente ad-hoc e ciò comporterebbe un maggiore dispendio di risorse. Il capitolato richiede diverse tecnologie non conosciute dai membri del gruppo e dunque sarebbe necessario un sostanzioso periodo di formazione. Il gruppo non ha trovato di interesse comune l'argomento del capitolato.

### 3.7 Conclusioni

Il gruppo ha optato per alternative ritenute più interessanti e affini alle conoscenze ed abilità pregresse dei vari membri. In luce delle considerazioni presentate.



## 4 Capitolato C2 - EmporioLambda

### 4.1 Informazioni generali

**Nome:** EmporioLambda: piattaforma di e-commerce in stile serverless<sub>G</sub>.

**Proponente:** Red Babel.

**Committente:** prof. Tullio Vardanega e prof. Riccardo Cardin.

### 4.2 Descrizione

EmporioLambda è una piattaforma di e-commerce costruita interamente con tecnologie serverless. Viene richiesto di creare una generica piattaforma di e-commerce che possa essere presentata come prototipo software da vendere a dei commercianti.

### 4.3 Finalità del progetto

Viene richiesta una piattaforma di e-commerce serverless costruita mediante un'architettura a microservizi. Questa piattaforma dovrà soddisfare i seguenti requisiti:

- **Calcolo delle tasse:** paesi diversi possono prevedere accise differenti sullo stesso prodotto;
- **Sistemi di fidelizzazione:** si vuole avere un sistema di punti, sconti e promozioni che fidelizzi il cliente;
- **Presenza di un Inventario:** si vuole tenere traccia dello stato dei prodotti, in maniera tale da automatizzare i rifornimenti ed avvisare il cliente quando un prodotto va out of stock;
- **Soddisfare vincoli geografici e/o legali:** certi prodotti possono essere vietati all'acquisto o la spedizione potrebbe non essere effettuata da/verso certi stati;
- **Interfaccia riusabile:** distinzione tra Model e View, permettendo di scollegare la logica dell'e-commerce dalla sua presentazione.

### 4.4 Tecnologie interessate

Il capitolato richiede l'utilizzo obbligatorio di alcune tecnologie, tra le quali:

- **Amazon Web Services<sub>G</sub>:** EmporioLambda dovrà necessariamente funzionare senza la necessità di installazioni su server dedicati. L'utilizzo di questa architettura permette di ottenere un sistema scalabile, ridurre i costi di installazione e di manutenzione. Amazon AWS è la suite<sub>G</sub> consigliata che permette di adempiere allo scopo;
- **TypeScript:** variante di JavaScript tipizzato. Dovrà essere il linguaggio adottato per lo sviluppo;
- **Next.js:** framework<sub>G</sub> JavaScript da adottare per lo sviluppo dell'interfaccia grafica;
- **Auth0:** identity manager consigliato per la gestione delle credenziali;
- **isStripe:** provider obbligatorio per la gestione del pagamento.

### 4.5 Aspetti positivi

Uno degli aspetti più interessanti di questo capitolato è lo sviluppo mediante microservizi. L'approccio serverless ed i microservizi sono tecnologie che si stanno affermando in maniera sempre più decisa nel mondo dell'informatica negli ultimi anni. Approfondire questi argomenti potrebbe essere un buon tornaconto personale. Inoltre, il proponente Red Babel sembra essere stata chiaro nell'esposizione dei requisiti.



## 4.6 Rischi

Il rischio<sub>G</sub> maggiore è che questo capitolato richiede molto lavoro di integrazione di tecnologie diverse, che sembra essere un timore diffuso all'interno del gruppo, inoltre l'utilizzo massiccio di JavaScript potrebbe essere un ulteriore ostacolo.

## 4.7 Conclusioni

Per i motivi sopracitati, è stato scelto di non considerare questo capitolato come prima scelta. Visto comunque il potenziale apporto positivo alla crescita personale e l'interesse da parte del gruppo, il capitolato è stato classificato come terza scelta.



## 5 Capitolato C3 - GDP

### 5.1 Informazioni generali

**Nome:** GDP: Gathering Detection Platform.

**Proponente:** Sync Lab.

**Committente:** prof. Tullio Vardanega e prof. Riccardo Cardin.

### 5.2 Descrizione

Il capitolato richiede di realizzare una piattaforma web nella quale si raccolgono e mostrano a video dei flussi di dati proveniente da varie fonti, come ad esempio i flussi dei passeggeri sui mezzi pubblici. Nasce come idea di prevenzione di assembramenti, dato dall'attuale periodo in cui stiamo vivendo a causa del COVID-19. Oltre a mostrare dati in tempo reale, si chiede che la piattaforma visualizzi anche una storicizzazione dei dati del passato e tramite algoritmi di machine learning mostri delle previsioni per il futuro.

### 5.3 Finalità del progetto

L'azienda proponente chiede in particolare di realizzare tre moduli principali:

- **Flussi di persone all'interno dei mezzi pubblici e/o alle fermate:** questa parte si divide in 2 sotto moduli:
  1. la prima parte riguarda le previsioni sui flussi, infatti, si chiede una stima del numero di persone che ci potrebbero essere ad una certa fermata/linea autobus in un particolare orario di un determinato giorno;
  2. la seconda parte è opzionale e chiede la realizzazione di un software che, tramite video, conti quante persone ci sono all'interno di un mezzo o un locale. Questo software deve essere costruito tramite librerie già esistenti/open source<sub>G</sub>.
- **Affollamento esercizi commerciali:** si chiede di sfruttare i dati forniti da Google Maps che indicano se e quando un luogo è affollato, allo scopo di determinare la presenza di persone, e sfruttare questi dati per allenare gli algoritmi di machine learning<sub>G</sub> nella predizione degli affollamenti;
- **Uber:** il terzo ed ultimo modulo proposto riguarda le corse effettuate dai mezzi Uber, infatti si vogliono sfruttare i dati delle partenze e degli arrivi per poter stimare quante persone si troveranno in un esatto luogo.

Sync Lab rimane aperta ad altre proposte da parte nostra per poter stimare al meglio le previsioni e creare delle heatmap più precise.

### 5.4 Tecnologie interessate

Per lo sviluppo del software del capitolato viene consigliato l'uso delle seguenti tecnologie:

- **Kafka:** accentratore di dati, piattaforma open-source di stream processing;
- **NumPy, Keras, TensorFlow, pytorch , scikit-learn:** vari algoritmi di machine learning per le predizioni;
- **TypeScript:** superset del linguaggio ECMAScript 6 (ES6);
- **Angular:** web framework basato su TypeScript per lo sviluppo front-end<sub>G</sub>;
- **Java spring:** framework basato sul linguaggio Java per lo sviluppo back-end;
- **Librerie open source:** motori software "contapersone" attraverso stream/immagini
- **Leaflet:** framework per la gestione delle heatmap.



## 5.5 Aspetti positivi

L'aspetto che più ci è sembrato interessante di questo progetto è la possibilità di poter contribuire ad un software che possa aiutare nella situazione attuale, qualora il progetto venisse poi sviluppato e adattato al mondo reale. Un altro aspetto che ci è parso molto interessante è la possibilità di vedere da vicino come funzionano alcune tecnologie che durante gli anni della triennale non vengono affrontate durante le lezioni, ad esempio algoritmi di machine learning, Angular, TypeScript, Java spring.

## 5.6 Rischi

La mancanza di esperienza, è un grosso rischio, poiché l'unica formazione che avremo su questi nuovi argomenti sarà lo studio autodidatta. Un'altra difficoltà, da non sottovalutare è il fatto che non ci verranno forniti dei database già pronti dai quali prendere i dati e poi elaborarli, ma sarà compito nostro, come richiesto dal capitolato, sviluppare un sistema che simuli tutti questi flussi in maniera più realistica possibile. Quindi non andremo a lavorare con dati reali, ma dovremo produrli noi.

## 5.7 Conclusioni

Nonostante fosse uno dei capitolati che più ci interessava dal punto di vista tecnologico, lo abbiamo scartato per via dei rischi. Non ci ha convinto la parte di simulazione dei dati, e abbiamo pensato che avremmo impiegato troppe risorse solo per simulare i dati in modo realistico e avremmo avuto meno tempo per lavorare effettivamente con la parte che più ci interessava, ovvero l'uso degli algoritmi di machine learning per la predizione degli affollamenti.



## 6 Capitolato C5 - PORTACS

### 6.1 Informazioni generali

**Nome:** PORTACS: piattaforma di controllo mobilità autonoma.

**Proponente:** Sanmarco Informatica S.p.A.

**Committente:** prof. Tullio Vardanega e prof. Riccardo Cardin.

### 6.2 Descrizione

Si vuole realizzare un sistema per la gestione dei percorsi di unità robotiche. Ogni entità nel sistema si sposta su una scacchiera virtuale da una posizione verso un punto d'interesse. In viaggio ogni singola unità deve evitare di creare rallentamenti o di scontrarsi con altre unità.

### 6.3 Finalità del progetto

Il proponente chiede che vengano sviluppati:

- Un sistema centralizzato con il quale le entità comunichino costantemente per tracciare il percorso e la posizione. Il sistema si prende carico di dare un corretto indirizzamento delle entità sulla scacchiera. La scacchiera presenta corsie, ognuna di esse è dotata di una capacità limitata di entità che possono attraversarla contemporaneamente. L'indirizzamento deve evitare che si formino code o avvengano collisioni. Non è richiesto che i percorsi siano ottimi.
- Un'interfaccia utente che mostri le entità presenti nel sistema e i loro spostamenti in real time, per ognuna di essa, inoltre, deve essere anche visualizzata la velocità di crociera e il percorso scelto. Si deve avere la possibilità di interrompere la navigazione di un'entità o di farla riprendere.
- Un database per la memorizzazione delle griglie in cui vengono definiti ostacoli, corsie eventualmente direzionate, e le entità registrate per ogni mappa.

### 6.4 Tecnologie interessate

- Java Servlet oppure Node.js, per la realizzazione del back-end.
- HTML, CSS, JavaScript (Angular, etc..) per la realizzazione del front-end.
- Un database per la memorizzazione dei dati.
- Docker<sub>G</sub> per la realizzazione delle componenti in modo modulare.
- Ambiente di simulazione per poter eseguire testing runtime.

### 6.5 Aspetti positivi

Non è richiesto l'ottimizzazione dei percorsi, tuttavia, essendo stato svolto il corso di Ricerca Operativa da parte di tutti i membri del gruppo, si hanno le competenze necessarie per considerare l'opzione senza dover generare troppe difficoltà nella realizzazione del prodotto. L'uso di Docker suggerisce modularità del progetto, di conseguenza potrebbe essere semplice da dividere tra i membri del gruppo e collaudare diverse le macro-componenti.

### 6.6 Rischi

La struttura del prodotto richiesto è complessa in quanto richiede numerose componenti: un server centralizzato che opera logica non banale, un database per tutte le componenti e la realizzazione di una applicazione responsive<sub>G</sub> per la gestione e il monitoraggio delle unità. Non conoscendo la tecnologia Docker si rischia di usarla in modo improprio e di non sfruttare a pieno i vantaggi che essa comporta.



## 6.7 Conclusioni

L'apparente complessità della richiesta non è stato considerato un aspetto negativo in quanto è un'idea accattivante, tuttavia, nonostante un grande apprezzamento del gruppo è stato deciso di rinunciare il progetto per via della forte affluenza da parte degli altri.



## 7 Capitolato C6 - RGP

### 7.1 Informazioni generali

**Nome:** RGP: Realtime Gaming Platform.

**Proponente:** Zero12.

**Committente:** prof. Tullio Vardanega e prof. Riccardo Cardin.

### 7.2 Descrizione

Si vuole realizzare un gioco a scorrimento verticale di navicelle viste dall'alto in cui il giocatore deve eliminare nemici ed evitare proiettili. Viene richiesto che il gioco possa essere giocato sia in modalità a giocatore singolo che multiplo. In una sessione a più giocatori non è possibile interagire direttamente con gli altri partecipanti che saranno visualizzati come "fantasmi" però sarà condivisa la plancia di gioco.

### 7.3 Finalità del progetto

Il proponente chiede che venga effettuato:

- Scouting di una tecnologia AWS per trovare quella che si presta meglio ad un gioco in real time fornendo motivazioni che supportano la scelta;
- Sviluppo di Architettura<sub>G</sub> Server-Cloud based su tecnologia Amazon Web Services scalabile per poter effettuare partite multigiocatore per un massimo di 6 device che metta a disposizione a tutti i giocatori lo stesso ambiente di gioco con il quale interagire e i movimenti e azioni degli altri giocatori;
- Implementazione del gioco con una visualizzazione dell'intera applicazione per iOS ed eventualmente, a discrezione del gruppo per Android.

### 7.4 Tecnologie interessate

- Viene richiesto l'uso di Node.js se possibile;
- Sviluppo nativo parte client: Kotlin per Android e Swift per iOS.

### 7.5 Aspetti positivi

Si tratta di una tipologia di giochi diffusa e si potrebbe prendere spunto da progetti già esistenti. L'azienda provvede a fornire gli asset per la realizzazione della grafica, quindi il risultato finale potrebbe risultare gradevole alla vista in quanto realizzata da persone di competenza.

### 7.6 Rischi

Viene richiesto di sviluppare con i servizi forniti da Amazon, tra i quali si deve scegliere quelli più adatti. Individuarli potrebbe non essere banale ma potrebbe semplificare l'architettura. Si dovrà garantire una bassa latenza: il gioco dovrà essere ben ottimizzato. Lo sviluppo dei giochi, tuttavia, non è considerato semplice.

### 7.7 Conclusioni

Nonostante le premesse sembrino incoraggianti e vista la semplicità del gioco, non essendo il tema del capitolato di interesse per il gruppo è stato deciso di scartarlo.





## 8 Capitolato C7 - SSD

### 8.1 Informazioni generali

**Nome:** SSD: soluzioni di sincronizzazione desktop.

**Proponente:** Zextras.

**Committente:** prof. Tullio Vardanega e prof. Riccardo Cardin.

### 8.2 Descrizione

L'utente professionista si aspetta di avere accesso dinamico a tutti i contenuti, creati o modificati con il loro dispositivo principale, con tutti i propri dispositivi (es. mobile o web). Viene richiesto di sviluppare un algoritmo<sub>G</sub> di sincronizzazione desktop, il quale si appoggerà alla piattaforma Zextras Drive, e un'interfaccia multiplatforma (Linux, MacOS, Windows). Il prodotto finale dovrà aderire al design pattern<sub>G</sub> MVC.

### 8.3 Finalità del progetto

Il progetto consiste nello sviluppare un'applicazione di sincronizzazione Desktop strutturata nel seguente modo:

- **Algoritmo di sincronizzazione:** un algoritmo solido ed efficiente in grado di garantire il salvataggio in cloud del lavoro e contemporaneamente la sincronizzazione dei cambiamenti presenti in cloud;
- **Interfaccia multiplatforma:** interfaccia pensata per l'uso dell'algoritmo. Deve essere sviluppata per i tre sistemi operativi più popolari (Linux, MacOS, Windows);
- **Integrazione con Zextras Drive:** entrambi i punti sviluppati sopra, dovranno essere integrate con Zextras Drive e quindi poter fruire dei contenuti di questo prodotto.

L'algoritmo e l'interfaccia sviluppati dovranno poter essere utilizzabili senza richiedere all'utente l'installazione manuale di ulteriori prodotti.

Le seguenti funzionalità base dovrebbero essere integrate:

- Configurazione e autenticazione dell'utente;
- Sincronizzazione costante dei contenuti (locali o remoti);
- Gestione dei file da sincronizzare e da ignorare nelle cartelle locali e in quelle cloud, con la possibilità di modificare in qualsiasi momento le preferenze di sincronizzazione;
- Sistema di notifica l'utente dei cambiamenti relativi ai file;
- Altre funzionalità avanzate presente nei principali competitor:
  - Gestione delle condivisioni;
  - Integrazione con il protocollo MAPI;
  - Integrazione con il prodotto web.

### 8.4 Tecnologie interessate

Per questo capitolato viene consigliato di utilizzare le seguenti tecnologie:

- **Qt Framework:** basato su C++, è consigliato per lo sviluppo dell'interfaccia grafica e del controller d'architettura;
- **Python:** linguaggio di programmazione, consigliato per lo sviluppo della Business Logic (algoritmo di sincronizzazione). Include le chiamate API<sub>G</sub> verso i sistemi di Zextras Drive, facilitandone l'integrazione.

Tecnologie necessaria alla realizzazione del progetto:

- **Zextras Drive:** sistema di gestione file in cloud;
- **Zimbra:** software applicativo di gruppo per la gestione della posta elettronica.



## 8.5 Aspetti positivi

Il capitolato risulta interessante per la scelta delle tecnologie di sviluppo consigliate, python infatti non è presente nel nostro percorso di studi e potrebbe essere una buona occasione per il suo apprendimento.

## 8.6 Rischi

Il gruppo ha poca familiarità con lo sviluppo delle applicazioni desktop. Inoltre, la documentazione delle API Zextras Drive sembra poco accurata.

## 8.7 Conclusioni

In seguito alla valutazione da parte del gruppo, è stato deciso di non prendere in considerazione il capitolato come prima scelta.