

CODEBUSTERS

Progetto: *HD Viz*
codebusterswe@gmail.com

Studio Di Fattibilità

Informazioni sul documento

Versione	1.0.0-1.8
Approvatori	Baldisseri Michele Scialpi Paolo
Redattori	Zenere Marco Rago Alessandro Safdari Hossain Pirolo Alessandro
Verificatori	Baldisseri Michele Sassaro Giacomo
Uso	Interno <i>Zucchetti</i>
Distribuzione	Prof. Vardanega Tullio Prof. Cardin Riccardo Gruppo <i>CodeBusters</i>

Descrizione

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

Registro delle modifiche

Versione	Data	Nominativo	Ruolo	Descrizione
1.0.0	10-12-2020	Baldisseri Michele	Responsabile	Approvazione del documento
0.2.0	05-12-2020	Sassaro Giacomo	Verificatore	Verifica del documento
0.1.5	01-12-2020	Safdari Hossain	Analista	Redazione studio di fattibilità C5
0.1.4	01-12-2020	Pirolo Alessandro	Analista	Redazione studio fattibilità C7
0.1.3	01-12-2020	Rago Alessandro	Analista	Redazione studio di fattibilità C6
0.1.2	01-12-2020	Zenere Marco	Analista	Correzione studio di fattibilità C4
0.1.1	30-11-2020	Zenere Marco	Analista	Redazione studio di fattibilità C4
0.1.0	26-11-2020	Sassaro Giacomo, Baldisseri Michele	Verificatore	Revisione complessiva del documento
0.0.4	19-11-2020	Zenere Marco, Scialpi Paolo	Analista	Redazione studio di fattibilità C2
0.0.3	19-11-2020	Zenere Marco, Scialpi Paolo	Analista	Redazione studio di fattibilità C1
0.0.2	18-11-2020	Pirolo Alessandro	Analista	Redazione studio di fattibilità C3
0.0.1	16-11-2020	<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	C1 - BlockCOVID: supporto digitale al contrasto della pandemia	6
2.1	Descrizione del capitolo	6
2.2	Tecnologie coinvolte	6
2.3	Vincoli	6
2.4	Aspetti positivi	6
2.5	Aspetti critici	7
2.6	Conclusioni	7
3	C2 - EmporioLambda: piattaforma di e-commerce in stile Serverless	8
3.1	Descrizione del capitolo	8
3.2	Tecnologie coinvolte	8
3.3	Vincoli	8
3.4	Aspetti positivi	9
3.5	Aspetti critici	9
3.6	Conclusioni	9
4	C3 - GDP - Gathering Detection Platform	10
4.1	Descrizione del capitolo	10
4.2	Tecnologie coinvolte	10
4.3	Vincoli	10
4.4	Aspetti positivi	11
4.5	Aspetti critici	11
4.6	Conclusioni	11
5	C4 - HD Viz: visualizzazione di dati multidimensionali	12
5.1	Descrizione del capitolo	12
5.2	Tecnologie coinvolte	12
5.3	Vincoli	12
5.4	Aspetti positivi	12
5.5	Aspetti critici	13
5.6	Conclusioni	13
6	C5 - PORTACS - Point Of Interest Oriented Real-Time Anti Collision System	14
6.1	Descrizione del capitolo	14
6.2	Tecnologie coinvolte	14
6.3	Vincoli	14
6.4	Aspetti positivi	14
6.5	Aspetti critici	15

6.6	Conclusioni	15
7	C6 - RGP: Realtime Gaming Platform	16
7.1	Descrizione del capitolo	16
7.2	Tecnologie coinvolte	16
7.3	Vincoli	16
7.4	Aspetti positivi	17
7.5	Aspetti critici	17
7.6	Conclusioni	17
8	C7 - SSD: soluzioni di sincronizzazione desktop	18
8.1	Descrizione del capitolo	18
8.2	Tecnologie coinvolte	18
8.3	Vincoli	18
8.4	Aspetti positivi	18
8.5	Aspetti critici	18
8.6	Conclusioni	18

1 Introduzione

1.1 Scopo del Documento

Questo documento contiene la stesura dello studio di fattibilità riguardante i sette capitolati proposti. Per ciascuno di essi vengono evidenziati i seguenti aspetti:

- **Descrizione generale;**
- **Tecnologie coinvolte;**
- **Vincoli;**
- **Aspetti positivi;**
- **Aspetti critici.**

Per ogni capitolato vengono espone le motivazioni e le ragioni per cui il gruppo ha deciso di non prenderlo in considerazione, scegliendo come progetto il capitolato *HD Viz*.

1.2 Glossario

Il gruppo *CodeBusters* ha redatto un documento denominato *Glossario 1.0.0* così da evitare ambiguità fra i termini presenti durante la lettura. In tale documento, sono presenti tutti i termini tecnici, ambigui e specifici del progetto con le loro relative definizioni. Un termine presente nel *Glossario 1.0.0* e utilizzato in questo documento viene indicato con la lettera "G" ad apice 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/C7.pdf>.

2 C1 - BlockCOVID: supporto digitale al contrasto della pandemia

2.1 Descrizione del capitolato

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

L'azienda proponente è *Imola Informatica*.

2.2 Tecnologie coinvolte

L'azienda non impone tecnologie specifiche per lo sviluppo, ma consiglia l'utilizzo di:

- Java^G (versione 8 o superiori), Python^G o Node.js^G per lo sviluppo del server back-end^G;
- Protocolli asincroni^G per le comunicazioni app mobile-server;
- Un sistema blockchain^G per salvare con opponibilità a terzi i dati di sanificazione;
- Kubernetes^G o di un PAAS^G, Openshift^G o Rancher^G, per il rilascio delle componenti del server e la gestione della scalabilità orizzontale.

2.3 Vincoli

Gli obiettivi sono:

- Realizzare un server e un'applicazione mobile, dove è prevista la presenza di due tipologie di utilizzatori:
 - **Amministratore di sistema:** ha i permessi di gestire e monitorare postazioni, stanze e utenti;
 - **Utente:** può essere:
 - * **Dipendente:** scansiona il tag presente alla sua postazione e segnala la sua presenza, con la possibilità di prenotarla;
 - * **Addetto alle pulizie:** può gestire e monitorare le stanze da igienizzare;
- È richiesto il report dei test effettuati;
- È richiesta una documentazione sulle scelte, le loro relative motivazioni ed eventuali problemi incontrati con le soluzioni attuate.

2.4 Aspetti positivi

- Progetto con fine attuale ed utile alla società;
- Stimola molto interesse per via dell'uso di molti servizi;
- L'azienda risulta essere molto disponibile per seguire il gruppo nello sviluppo del progetto.

2.5 Aspetti critici

- Per questo capitolato è presente molta concorrenza;
- Richiede l'apprendimento di molti servizi e linguaggi che i membri del gruppo non conoscono, andando ad incidere molto sul tempo che andrebbe dedicato per il loro apprendimento.

2.6 Conclusioni

Visto il contesto attuale di pandemia un'applicazione su questo campo risulta essere molto utile; anche le tecnologie coinvolte hanno suscitato un forte interesse. Però, dopo un'attenta valutazione, il capitolato in questione è stato escluso a causa della sua complessità, ritenuta più elevata rispetto alle altre proposte.

3 C2 - EmporioLambda: piattaforma di e-commerce in stile Serverless

3.1 Descrizione del capitolo

Nel capitolato viene proposto di creare una piattaforma di e-commerce^G. Lo scopo finale 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 acquistarli. La piattaforma deve offrire un'interfaccia per i venditori, dove essi possono aggiungere e modificare i prodotti destinati alla vendita.

L'azienda proponente è *Red Babel*.

3.2 Tecnologie coinvolte

Il progetto prevede l'adozione di una bozza iniziale dell'architettura dettata dal proponente e di alcune scelte tecnologiche obbligatorie; il fornitore è libero e altamente incoraggiato ad esplorare diverse scelte. L'applicazione deve essere costruita con tecnologia serverless^G con una API^G basata su HTTP^G:

- Si utilizzano AWS Lambda^G e altre componenti di supporto (AWS API Gateway^G, AWS DynamoDB^G, AWS S3^G);
- CloudFormation^G per la gestione delle risorse sopraelencate;
- Serverless Framework^G per l'implementazione di applicazioni serverless^G, facilitandone alcune difficili strutture (come permessi, sottoscrizioni o accesso);
- Il BFF^G (Back end For Front end) sarà implementato in Next.js^G e richiede l'uso di Typescript^G come linguaggio principale;
- il codice sorgente dovrebbe essere pubblicato e aggiornato utilizzando GitHub^G o GitLab^G.

3.3 Vincoli

È richiesto l'uso di un numero minimo di ambienti di lavoro: uno locale di ogni sviluppatore, uno per i test (accessibile a tutti gli sviluppatori) e uno pubblico (accessibile agli utenti). È richiesta l'implementazione di tutti i moduli di alto livello:

- **EmporioLambda-frontend** (EML-FE): modulo che serve le pagine web richieste dal cliente;
- **EmporioLambda-backend** (EML-BE): modulo che espone i servizi dell'applicazione;
- **EmporioLambda-integration** (EML-I): rappresenta tutti i servizi di terze parti;
- **EmporioLambda-monitoring** (EML-MON): è il set di strumenti usati per monitorare lo stato dell'applicazione.

È richiesta la presenza delle seguenti pagine (con relative principali funzionalità):

- Homepage;
- Lista dei prodotti;

- Dettaglio prodotto;
- Carrello;
- Account;
- Checkout;
- Controllo del venditore.

Il sito deve implementare i seguenti ruoli utilizzando AWS Cognito Identity^G:

- Amministratore;
- Venditore;
- Cliente.

L'unica integrazione obbligatoria richiesta è il provider di pagamento Stripe^G.

3.4 Aspetti positivi

- È richiesto l'uso di numerosi servizi forniti da AWS^G, sia comuni che meno, i quali possono risultare utili al nostro futuro;
- Nel capitolato viene richiesto il pagamento elettronico, argomento attualmente di nicchia;
- Linguaggi come TypeScript^G e JavaScript^G sono molto utili anche in ambito lavorativo;
- Nel 2020 ormai gli e-commerce sono alla portata di tutti e fanno parte della vita quotidiana delle persone, perciò è un argomento familiare ai membri del gruppo.

3.5 Aspetti critici

Discutendo con il gruppo non abbiamo rilevato aspetti critici rilevanti.

3.6 Conclusioni

In questo caso si utilizzano tecnologie non troppo complesse e, prendendo in considerazione le altre piattaforme di e-commerce esistenti, al momento non risultano problemi evidenti che si potrebbero riscontrare nella fase di sviluppo. Tuttavia, il gruppo ha considerato questo capitolato poco appetibile.

4 C3 - GDP - Gathering Detection Platform

4.1 Descrizione del capitolo

Il capitolato presentato si pone come obiettivo finale la realizzazione di una piattaforma in grado di dare all'utilizzatore 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^G e/o deep learning^G.

L'azienda proponente è *SyncLab*.

4.2 Tecnologie coinvolte

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

- Utilizzo di Java^G e Angular^G per lo sviluppo della parte di back-end^G e di front-end^G della componente web del sistema;
- Il framework^G Leaflet^G per la gestione delle mappe (heatmap ecc.) (<https://leafletjs.com>);
- Utilizzo di protocolli asincroni^G per le comunicazioni tra le diverse componenti;
- Utilizzo del pattern^G Publisher/Subscriber, e adozione del protocollo MQTT^G ("MQ Telemetry Transport" o "Message Queue Telemetry Transport") di facile implementazione e ampia diffusione in applicazioni M2M^G (MachineToMachine) e IoT^G (InternetOfThings).

Per la parte di machine learning^G 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.

Per facilitare la comprensione delle librerie consigliate, l'azienda propone delle guide e delle risorse online.

4.3 Vincoli

L'azienda proponente suggerisce un'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 all'effettiva domanda;
- **Message-driven:** i servizi devono rispondere a quello che accade, non tentare di controllare ciò che fa.

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

4.4 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, molto interessante e che potrebbe risultare utile in futuro;
- L'azienda sembra essere molto disponibile per seguire il gruppo nel percorso di sviluppo del progetto.

4.5 Aspetti critici

- Il capitolato richiede l'apprendimento di molti servizi, sconosciuti dai membri del gruppo, basati su argomenti non trattati nel percorso universitario. Gran parte del tempo andrebbe quindi dedicato allo studio di questi concetti e strumenti, per poi applicare le conoscenze acquisite;
- 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.

4.6 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 C4 - HD Viz: visualizzazione di dati multidimensionali

5.1 Descrizione del capitolo

Il capitolato C4 ha come obiettivo quello di creare un'applicazione di visualizzazione di grandi quantità di dati, con numerose dimensioni e in un formato comprensibile. A questo scopo è necessario utilizzare algoritmi di intelligenza artificiale che, agendo su concetti che variano dal grafico preso in considerazione, riescano a sviluppare un modello semplificato che ne evidenzi i cluster^G. L'applicazione dovrà permettere l'interazione dell'utente con i grafici ottenuti.

L'azienda proponente è *Zucchetti*.

5.2 Tecnologie coinvolte

L'applicazione "HD Viz" dovrà utilizzare le seguenti tecnologie:

- HTML^G/CSS^G/Javascript^G con libreria D3.js^G per la parte client;
- Il database^G (SQL^G o NOSQL^G) potrà essere sviluppato in Java^G con server Tomcat^G o in JavaScript^G con server Node.js^G.

5.3 Vincoli

I vincoli del capitolato C4 riguardano, in prima istanza, il numero di dimensioni dei dati da visualizzare. L'applicazione deve riuscire a visualizzare dati con almeno 15 dimensioni. Il formato dei dati da fornire a HD Viz deve essere una query^G ad un database^G o anche un file in formato CSV^G preparato in precedenza. HD Viz dovrà presentare le seguenti visualizzazioni:

- **Scatter plot Matrix^G** (fino ad un massimo di 5 dimensioni): è la presentazione a riquadri disposti a matrice di tutte le combinazioni di scatter plot^G;
- **Force Field^G**: converte dati in più dimensioni in un grafico a due/tre dimensioni rappresentandoli sulla base della loro distanza;
- **Heat Map^G**: rappresenta la distanza tra i punti con colori più o meno intensi. In questa visualizzazione è necessario ordinare i dati per evidenziarli;
- **Proiezione Lineare Multi Asse^G**: si rappresentano a due dimensioni dati multidimensionali, permettendo di spostare gli assi per rendere la visualizzazione più comprensibile.

È necessario che l'utente non percepisca le operazioni di riduzione dimensionale; inoltre può scegliere in quali dimensioni vuole visualizzare il grafico. Il proponente lascia ampia libertà di esplorazione dell'argomento e accetta altri requisiti opzionali proposti dal fornitore come, ad esempio, la creazione di algoritmi per ridurre le dimensioni.

5.4 Aspetti positivi

- L'analisi dei Big Data^G è una problematica più che mai attuale e che potrebbe risultare molto utile in ambito professionale;
- Nel capitolato si tratta l'intelligenza artificiale, altro argomento molto interessante in un futuro ambito lavorativo;

- Gli strumenti HTML^G/CSS^G/JavaScript^G sono già stati visti nel corso di Tecnologie Web;
- La libreria D3.js è ben documentata e largamente fornita di esempi sulle visualizzazioni richieste;
- L'azienda ha dato disponibilità per ulteriori chiarimenti in corso d'opera;
- L'azienda fornirà dei set di dati per testare il software.

5.5 Aspetti critici

- Il capitolato potrebbe risultare difficoltoso nel testare gli algoritmi;
- Altri capitolati hanno per argomento situazioni più concrete in confronto all'analisi ed elaborazione dei dati;
- La libreria D3.js^G è molto potente e può richiedere diverso tempo per imparare a padroneggiarla.

5.6 Conclusioni

La possibilità di applicare conoscenze del settore matematico a contesti reali ha riscosso fin da subito un notevole interesse da parte di tutti i membri.

In seguito ad alcuni chiarimenti forniti dall'azienda in questione, rivelatasi molto chiara e disponibile, il gruppo ha percepito l'importanza di approfondire il settore dei Big Data^G. Esplorando questo campo, al giorno d'oggi molto florido nell'utilizzo e in continuo sviluppo, tutti i membri hanno l'opportunità di aggiungere al proprio bagaglio curricolare un'esperienza importante.

L'impiego della libreria D3.js^G, una delle più usate in questo ambito, permette di conoscere strumenti moderni e potenti, nonostante richiedano un notevole impegno per padroneggiarli. Tuttavia l'utilizzare HTML^G, Javascript^G e CSS^G, linguaggi già visti nel corso di Tecnologie Web, agevolerà e permetterà di dedicare molto più tempo per approfondirli.

In seguito ad un'attenta valutazione, anche rispetto alle altre proposte, il gruppo ha scelto questo capitolato come prima preferenza, fiducioso nell'instaurare un buon rapporto con l'azienda.

6 C5 - PORTACS - Point Of Interest Oriented Real-Time Anti Collision System

6.1 Descrizione del capitolo

Il capitolato propone la realizzazione di un sistema per il monitoraggio e la gestione di unità presenti in una mappa.

Ogni unità (che può essere un robot, un muletto o un'auto a guida automatica) ha un punto di partenza nella mappa e una lista di POI (Point Of Interest) che deve raggiungere. Il sistema dovrà indicare ad ogni unità la prossima mossa che dovrà fare in base alla posizione, direzione e velocità di tutte le altre. Ognuna dovrà inviare costantemente questi parametri al sistema, in modo da evitare incidenti e ingorghi.

L'azienda proponente è *Sanmarco Informatica*

6.2 Tecnologie coinvolte

Il proponente non impone specifiche tecnologie per lo sviluppo della piattaforma ma consiglia l'utilizzo di :

- Node.js^G per la parte di back-end^G;
- React.js^G o Next.js^G per lo sviluppo di front-end^G;
- Socket^G, ossia una libreria per sviluppare applicazioni real-time^G, facilmente integrabile con Node.js^G.

6.3 Vincoli

L'azienda proponente pone i seguenti obiettivi da raggiungere:

- Il sistema deve avere una visualizzazione in real-time^G della mappa e della posizione delle singole unità;
- L'interfaccia utente (presente in ogni unità) deve prevedere quattro frecce direzionali, un pulsante start/stop e un indicatore della velocità attuale.

La versione finale del software deve essere in grado di accettare i seguenti input:

- Scacchiera oppure mappa con percorsi predefiniti e relativi vincoli (sensi unici, numero massimo di unità contemporanee);
- Definizione dei POI (aree di carico/scarico e sosta).

6.4 Aspetti positivi

- Il proponente non obbliga l'implementazione di algoritmi di ricerca operativa per l'ottimizzazione dei percorsi;
- Non si richiede la geo-localizzazione interna o esterna. Questa verrà simulata, riducendo il tempo dedicato alla parte di codifica;

- L'utilizzo di tecnologie per la gestione real-time^G è parso un argomento molto interessante e ha suscitato curiosità nel gruppo;
- L'azienda proponente ha dimostrato disponibilità per ulteriori chiarimenti.

6.5 Aspetti critici

- L'azienda proponente non suggerisce tecnologie apprese in ambito universitario o già conosciute dai membri del gruppo;
- Complessità elevata rispetto ad altri capitolati.

6.6 Conclusioni

Il capitolato ha attirato l'attenzione e stimolato l'interesse di tutti i membri del gruppo, in particolar modo per il contesto applicativo che risulta essere un argomento attuale, seppur molto complesso. Per via del livello di difficoltà e dell'impegno stimato, il gruppo ha ritenuto troppo elevata la mole di lavoro necessaria e ha deciso di orientarsi verso altri capitolati.

7 C6 - RGP: Realtime Gaming Platform

7.1 Descrizione del capitolo

L'obiettivo del capitolato è la realizzazione di un videogioco a scorrimento verticale, fruibile da cellulare, con due modalità di gioco:

- Single player;
- Multiplayer.

Il fulcro del progetto è lo sviluppo della parte multiplayer, affinché si possa giocare in sessioni da 2 a 6 giocatori in modalità fantasma, ossia potendo vedere i movimenti degli avversari ma senza poter interagire con essi.

Riguardo il single player, è necessario creare una modalità "infinita", con livelli di difficoltà crescente, senza interruzioni tra un livello e il successivo.

L'azienda proponente è *Zero12*.

7.2 Tecnologie coinvolte

L'azienda suggerisce l'utilizzo di tecnologie differenti a seconda della componente da implementare:

- Tecnologie AWS^G per la parte server-side, lasciando al fornitore l'analisi sul servizio più adatto per la realizzazione di giochi real-time^G (per esempio AWS GameLift^G, AWS Appsync^G).
Nel caso dell'utilizzo di servizi che richiedano lo sviluppo di codice è vincolato l'uso del linguaggio Node.js^G;
- DynamoDB^G come database^G NoSQL^G per il salvataggio delle classifiche e informazioni a supporto dell'applicativo;
- Tecnologie native per la realizzazione della piattaforma mobile.
Il fornitore ha la possibilità di scegliere tra iOS^G (preferenza del proponente) o Android^G come sistema operativo dove sviluppare l'interfaccia.
A seconda del sistema operativo i linguaggi da utilizzare sono:
 - Swift^G e framework SwiftUI^G (engine SceneKit^G/SpriteKit^G) con target minimo iOS 13;
 - Kotlin^G con target minimo Android 8.

7.3 Vincoli

I vincoli imposti dall'azienda proponente sono:

- Effettuare un'analisi preliminare circa la tecnologia da utilizzare per la componente server. Questa analisi deve essere documentata, giustificando la scelta finale. Il team di *Zero12* si offre in aiuto al fornitore in caso di perplessità a riguardo;
- API^G sviluppate in Node.js^G;
- Architettura di progetto basata a micro-servizi, ossia a tante funzioni base indipendenti, utilizzabili singolarmente senza compromettere la funzionalità delle altre;
- Sviluppo della mobile app in ambiente iOS^G o Android^G per testare tutti i micro-servizi e le dinamiche gaming in real-time^G.

7.4 Aspetti positivi

- Utilizzo di tecnologie innovative e molto usate nell'ambito lavorativo;
- Possibilità di osservare i risultati di test dinamici direttamente nell'interfaccia di gioco.

7.5 Aspetti critici

- Il capitolato presenta un progetto molto specifico rispetto ad altri. L'ambito di sviluppo è quello gaming e affinché si lavori al meglio deve essere d'interesse a tutti i membri del gruppo;
- Le tecnologie consigliate sono nuove a tutti i membri del gruppo e richiedono quindi uno studio preliminare approfondito per essere utilizzate correttamente;
- I vincoli imposti risultano essere per lo più generali e se da un lato potrebbe essere un bene per la creatività concessa al gruppo, dall'altro risulta essere anche un rischio durante lo sviluppo.

7.6 Conclusioni

Per la principale causa del generale disinteresse riguardo il game development, è stato deciso di orientarsi verso un altro capitolato che potesse creare stimoli e curiosità a tutti i membri del gruppo.

8 C7 - SSD: soluzioni di sincronizzazione desktop

8.1 Descrizione del capitolo

Il capitolato presentato pone come obiettivo finale la realizzazione di un algoritmo di sincronizzazione desktop in grado di garantire il salvataggio in cloud^G di qualsiasi lavoro, sincronizzando contemporaneamente i cambiamenti. In aggiunta a ciò, lo sviluppo di un'interfaccia multiplatforma^G per l'uso dell'algoritmo e l'utilizzo di quest'ultimo con il prodotto dell'azienda *Zextras Drive*.

L'azienda proponente è *Zextras*.

8.2 Tecnologie coinvolte

L'azienda non impone tecnologie specifiche per lo sviluppo della parte back-end^G o della UI^G, ma vi sono comunque delle scelte preferenziali da considerare nello svolgimento del progetto:

- Linguaggio Python^G per lo sviluppo dell'algoritmo di sincronizzazione;
- Framework^G Qt^G per lo sviluppo dell'interfaccia e del controller d'architettura.

8.3 Vincoli

Il proponente richiede di rispettare alcuni vincoli sull'architettura del progetto:

- L'algoritmo di sincronizzazione e l'interfaccia utente devono essere utilizzabili nei tre principali sistemi operativi desktop (Mac, Windows, Linux) senza richiedere all'utente l'installazione manuale di ulteriori prodotti per il funzionamento;
- Forte distinzione tra interfaccia e algoritmo di sincronizzazione;
- Utilizzare il pattern architetturale MVC^G.

8.4 Aspetti positivi

- Il linguaggio proposto, Python^G, è in grande ascesa negli ultimi anni e sono molte le tecnologie che lo stanno implementando o che lo hanno già fatto. Il gruppo ritiene che sia di grande aiuto apprenderlo;
- Il capitolato propone l'utilizzo del framework^G Qt^G, che è già familiare a tutto il gruppo di progetto.

8.5 Aspetti critici

- Lo sviluppo di un algoritmo di sincronizzazione a partire da zero;
- Il framework^G Qt^G è già conosciuto. Questo rende il capitolato meno interessante per quanto riguarda l'opportunità di apprendere nuove tecnologie.

8.6 Conclusioni

Il capitolato non ha destato molto la curiosità del gruppo; l'utilizzo del linguaggio Python^G è l'unica nota d'interesse sollevata dal gruppo, ma questo è proposto anche in altri capitolati che hanno catturato maggiormente l'attenzione.