



Progetto: *HD Viz: visualizzazione di dati multidimensionali*
codebusterswe@gmail.com

Studio Di Fattibilità

Informazioni sul documento

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

Descrizione

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

Registro delle modifiche

Versione	Data	Nominativo	Ruolo	Descrizione
0.3.0	07-12-2020	Sassaro Giacomo	Verificatore	Verifica del documento
0.2.0	02-12-2020	Baldisseri Michele	Verificatore	Verifica del documento
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	Creazione documento Latex e adattamento bozza Latex
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

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;
- Prerequisiti e tecnologie coinvolte;
- Vincoli;
- Aspetti positivi;
- Aspetti critici.

Infine, per ogni capitolato vengono espone le motivazioni e le ragioni per cui il gruppo ha scelto come progetto il capitolato *C4 HD Viz: visualizzazione di dati multidimensionali* a discapito degli altri sei proposti.

I committenti di tutti i capitolati proposti sono Prof. Vardanega Tullio e Prof. Cardin Riccardo

1.2 Glossario

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

2 Riferimenti

2.1 Normativi

- *Studio di Fattibilità 0.0.0.*

2.2 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>

3 C1 - BlockCOVID: supporto digitale al contrasto della pandemia

3.1 Descrizione del capitolo

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

L'azienda proponente è *Imola Informatica*.

3.2 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;
- protocolli asincroni 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.

3.3 Vincoli

Gli obiettivi sono:

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

3.4 Aspetti positivi

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

3.5 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.

3.6 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 complessità, ritenuta più elevata rispetto alle altre proposte.

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

4.1 Descrizione del capitolo

Nel capitolato viene proposto di creare una piattaforma di E-Commerce^G, ovvero la compravendita on-line di prodotti. 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 l'acquisto. La piattaforma dovrà offrire un'interfaccia per i venditori dove possono aggiungere e modificare i prodotti destinati alla vendita.

L'azienda proponente è *Red Babel*.

4.2 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. Questa applicazione deve essere costruita con tecnologia serverless^G con una API^G basata su HTTP^G:

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

4.3 Vincoli

É 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 un ambiente pubblico accessibile dagli utenti. É 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.

E' richiesta la presenza di determinate minime, (ognuna con le proprie funzioni principali):

- Homepage
- Product Listing Page
- Product Detail Pages
- Shopping Cart
- Account

- Checkout
- Merchant dashboard

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

- Amministratore
- Venditore
- Cliente

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

4.4 Aspetti positivi

- E' richiesto l'uso di numerosi servizi forniti da AWS, 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 e JavaScript 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.

4.5 Aspetti critici

Discutendo con il gruppo non abbiamo rilevato aspetti critici rilevanti.

4.6 Conclusioni

In questo caso si utilizzano tecnologie non troppo complesse e, prendendo in considerazione le altre piattaforme di e-commerce esistenti (per esempio Amazon), è relativamente facile progettarela visto che l'e-commerce è ormai la quotidianità. Il gruppo ha però considerato questo capitolato poco appetibile.

5 C3 - GDP - Gathering Detection Platform

5.1 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 e/o deeplearnig in modo da predire le zone potenzialmente a rischio.

L'azienda proponente è *SyncLab*.

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

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

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

Infine, per facilitare la comprensione delle librerie consigliate, l'azienda propone delle guide e qualche altra risorsa online.

5.3 Vincoli

La 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.

É 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*.

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

5.5 Aspetti critici

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

5.6 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.

6 C4 - HD Viz: visualizzazione di dati multidimensionali

6.1 Descrizione del capitolo

Il capitolato C4 ha come obiettivo quello di creare un'applicazione di visualizzazione di grandi quantità di dati con numerose dimensioni in un formato comprensibile dall'occhio umano. A questo scopo è necessario sviluppare algoritmi di intelligenza artificiale che, agendo sulla distanza dei vari punti del grafico, riescano a sviluppare un modello semplificato che ne evidenzia i cluster. L'applicazione dovrà inoltre agire su questi grafici creati evidenziando i dati ottenuti.

L'azienda proponente è *Zucchetti*

6.2 Tecnologie coinvolte

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

- HTML/CSS/Javascript con libreria D3.js per la parte client;
- La parte server e un database (SQL o NOSQL) potranno essere sviluppati in Java con server Tomcat o in Javascript con server Node.js.

6.3 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 ad un database o anche un file in formato CSV preparato in precedenza. HD Viz dovrà presentare le seguenti visualizzazioni:

- Scatter plot Matrix (fino ad un massimo di 5 dimensioni): è la presentazione a riquadri disposti a matrice di tutte le combinazioni di scatter plot;
- Force Field: converte dati in più dimensioni in un grafico a due/tre dimensioni rappresentandoli sulla base della loro distanza ;
- Heat Map: 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: si rappresentano a due dimensioni dati multidimensionali permettendo di spostare gli assi per rendere la visualizzazione più comprensibile.

Il proponente lascia ampia libertà di esplorazione dell'argomento e accetta altri requisiti opzionali proposti dal fornitore.

6.4 Aspetti positivi

- L'analisi dei Big Data 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/CSS e Javascript sono già stati visti nel corso di Tecnologie Web;
- La libreria D3 è ben documentata e largamente fornita di esempi sulle visualizzazioni richieste;
- L'azienda ha dato disponibilità per ulteriori chiarimenti in corso d'opera.

6.5 Aspetti critici

- Il capitolato potrebbe risultare difficoltoso nella realizzazione degli algoritmi e nel testarli;
- Altri capitolati hanno per argomento oggetti o situazioni più reali in confronto all'analisi ed elaborazione dei dati.

6.6 Conclusioni

Il capitolato C4 è tra i preferiti dal gruppo che infine ha deciso di (...)

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

7.1 Descrizione del capitolo

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

Ogni unità (che può essere un robot, un muletto o un'auto a guida automatica) ha un punto di partenza nella griglia e una lista di POI (*"Point Of Interest"*) che deve raggiungere. Il sistema dovrà indicare ad ognuna di esse la prossima mossa che dovrà fare in base alla posizione, direzione e velocità delle altre unità. Ognuna dovrà inviare costantemente questi parametri al sistema, in modo da rendere possibile pilotare e coordinare tutte le unità evitando incidenti ed ingorghi. L'azienda proponente è *Sanmarco Informatica*

7.2 Tecnologie coinvolte

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

- Nodejs per la parte di Back-end e Reactjs o Next.js per lo sviluppo di Front-end che permettono di creare una SPA (*"Single Page Application"*) ;
- Socket: è una libreria che consente di sviluppare applicazioni real-time e risulta facile da integrare con Nodejs;

7.3 Vincoli

L'azienda proponente pone i seguenti obiettivi da raggiungere:

- il sistema dovrà avere una visualizzazione in real-time della mappa e relativa posizione delle singole unità;
- l'interfaccia utente che rappresenterà ogni singola unità dovrà prevedere 4 frecce direzionali (si *"accenderà"* quella suggerita dal sistema), il pulsante start/stop e l'indicatore della velocità attuale.

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

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

7.4 Aspetti positivi

- Il proponente non obbliga l'implementazione di algoritmi di ricerca operativa per l'ottimizzazione dei percorsi;
- Non si richiede la geolocalizzazione interna o esterna: questa verrà simulata, riducendo il tempo dedicato alla parte di codifica;
- L'utilizzo di tecnologie per la gestione real-time è parso un argomento molto interessante e ha suscitato curiosità nel gruppo;
- L'azienda proponente ha dimostrato disponibilità per ulteriori chiarimenti e/o informazioni.

7.5 Aspetti critici

L'azienda proponente non suggerisce tecnologie per lo sviluppo della piattaforma apprese in ambito universitario o conosciute dai membri del gruppo; inoltre l'ambito risulta avere una complessità elevata rispetto agli altri capitolati e ciò potrebbe comportare una dilatazione eccessiva del tempo dedicato allo sviluppo.

7.6 Conclusioni

Tale 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 e molto utile per il futuro. 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.

8 C6 - RGP: Realtime Gaming Platform

8.1 Descrizione del capitolo

L'azienda proponente è *Zero12*.

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 si basa sullo 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.

8.2 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 realtime (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 NodeJS^G;
- DynamoDB^G come database NoSQL 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 (preferenza del proponente) o Android come sistema operativo dove sviluppare l'interfaccia mobile.
A seconda di quest'ultima scelta i linguaggi da utilizzare sono:
 - Swift^G e framework SwitUI^G (engine SceneKit^G/SpriteKit^G) con target minimo iOS 13;
 - Kotlin^G con target minimo Android 8.

8.3 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 sviluppate in NodeJS;
- 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 o Android per testare tutti i micro-servizi e le dinamiche gaming in realtime.

8.4 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.

8.5 Aspetti critici

- Il capitolato presenta un progetto molto specifico rispetto ad altri. L'ambito di sviluppo è quello gaming e perchè si lavori al meglio deve essere di 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.

8.6 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.

9 C7 - SSD: soluzioni di sincronizzazione desktop

9.1 Descrizione del capitolo

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

9.2 Tecnologie coinvolte

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

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

9.3 Vincoli

La 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.

9.4 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. Riteniamo, quindi, che sia di grande aiuto apprenderlo;
- Il capitolato propone l'utilizzo del framework Qt^G, che è già familiare a tutto il gruppo di progetto.

9.5 Aspetti critici

- Lo sviluppo di un algoritmo di sincronizzazione a partire da zero;
- Il framework Qt è già conosciuto, questo rende il capitolato meno interessante, in quanto questo progetto non viene visto solo come un'occasione per fare esperienza nel campo del Software Engineering, ma anche come un'opportunità per apprendere tecnologie che non si studiano durante il percorso di studi.

9.6 Conclusioni

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