



Code of Duty - Progetto *HD VIZ*

Studio di fattibilità

Versione	v0.0.1
Approvazione	
Redazione	
Verifica	
Stato	In lavorazione
Uso	Interno
Destinato a	

Descrizione

`info@codeofduty.it`

Diario delle modifiche

Versione	Data	Nominativo	Ruolo	Descrizione
v0.0.1	02/12/2020	Damiano Zanardo	Analista	Creazione Template del documento
v0.0.2	02/12/2020	Diego Piola	Analista	Aggiunto capitolato scelto C4
v0.0.3	02/12/2020	Damiano Zanardo	Analista	Aggiunto capitolato C1
v0.0.4	02/12/2020	Alessandro Flori	Analista	Aggiunto capitolato C2
v0.0.5	02/12/2020	Andrea Breggion	Analista	Aggiunto capitolato C3
v0.0.6	02/12/2020	Andrea Mascari	Analista	Aggiunto capitolato C5
v0.0.7	02/12/2020	Andrea Signori	Analista	Aggiunto capitolato C6
v0.0.8	02/12/2020	Matteo Falsetti	Analista	Aggiunto capitolato C7

Indice

1	CAPITOLATO SCELTO: Capitolato 4 - HD Viz	4
1.1	Informazioni generali	4
1.2	Descrizione capitolato	4
1.3	Finalità del progetto	4
1.4	Tecnologie interessate	4
1.5	Aspetti positivi	5
1.6	Criticità e fattori di rischio	5
1.7	Conclusioni	5
2	Capitolato 1 - BlockCOVID	6
2.1	Informazioni generali	6
2.2	Descrizione capitolato	6
2.3	Finalità del progetto	6
2.4	Tecnologie interessate	6
2.5	Aspetti positivi	7
2.6	Criticità e fattori di rischio	7
2.7	Conclusioni	7
3	Capitolato 2 - EmporioLambda	8
3.1	Informazioni generali	8
3.2	Descrizione capitolato	8
3.3	Finalità del progetto	8
3.4	Tecnologie interessate	8
3.5	Aspetti positivi	9
3.6	Criticità e fattori di rischio	9
3.7	Conclusioni	9
4	Capitolato 3 - GDP	10
4.1	Informazioni generali	10
4.2	Descrizione capitolato	10
4.3	Finalità del progetto	10
4.4	Tecnologie interessate	10
4.5	Aspetti positivi	10
4.6	Criticità e fattori di rischio	11
4.7	Conclusioni	11
5	Capitolato 5 - PORTACS	12
5.1	Informazioni generali	12
5.2	Descrizione capitolato	12
5.3	Finalità del progetto	12
5.4	Tecnologie interessate	12
5.5	Aspetti positivi	12
5.6	Criticità e fattori di rischio	12
5.7	Conclusioni	13

6	Capitolato 6 - RGP	14
6.1	Informazioni generali	14
6.2	Descrizione capitolato	14
6.3	Finalità del progetto	14
6.4	Tecnologie interessate	14
6.5	Aspetti positivi	14
6.6	Criticità e fattori di rischio	15
6.7	Conclusioni	15
7	Capitolato 7 - SSD	16
7.1	Informazioni generali	16
7.2	Descrizione capitolato	16
7.3	Finalità del progetto	16
7.4	Tecnologie interessate	16
7.5	Aspetti positivi	17
7.6	Criticità e fattori di rischio	17
7.7	Conclusioni	17

1 CAPITOLATO SCELTO: Capitolato 4 - HD Viz

1.1 Informazioni generali

- **Nome:** *HD Viz*: Visualizzazione di dati con molte dimensioni
- **Proponente:** *Zucchetti S.p.A.*
- **Committente:** *Prof. Tullio Vardanega e Prof. Riccardo Cardin*

1.2 Descrizione capitolato

Il proponente richiede l'implementazione di un'applicazione di visualizzazione di dati con molte dimensioni. Tale applicazione verrà sviluppata tramite le tecnologie HTML/CSS/JavaScript, con supporto della libreria D3.js. È inoltre richiesta una parte server di supporto alla presentazione nel browser e per interfacciarsi ad un database, la quale può essere implementata in Java con server Tomcat oppure in Javascript con server node.js.

1.3 Finalità del progetto

Tramite la realizzazione di questo software l'azienda può visualizzare in maniera più efficiente grosse quantità di dati. Un esempio pratico può essere il controllo dei cedolini degli stipendi dei dipendenti. Il programma dovrà:

- **I dati da visualizzare** dovranno poter avere almeno fino a dimensioni.
- I dati devono poter essere forniti al sistema di visualizzazione, sia con query ad un database che da file in formato CSV.

Inoltre HD Viz dovrà presentare almeno le seguenti visualizzazioni:

1. Scatter plot Matrix (fino a 5 dimensioni)
2. Force Field
3. Heat Map
4. Proiezione Lineare Multi Asse

Riguardo al grafico Heat Map HD Viz dovrà ordinare i punti nel grafico per evidenziare i cluster presenti nei dati.

1.4 Tecnologie interessate

Il software richiesto dovrà utilizzare le seguenti tecnologie:

- **JavaScript/HTML/CSS:** linguaggi da impiegare per lo sviluppo dell'applicazione di visualizzazione.
- **Libreria D3.js:** libreria che contiene tipi di visualizzazione predefiniti.
- **Java o Javascript:** linguaggi da utilizzare per la presentazione nel browser e per l'interfacciamento a database.
- **Tomcat o Node.js:** web server per verificare il corretto funzionamento dell'applicativo

1.5 Aspetti positivi

- Come primo aspetto, sicuramente, la fama dell'azienda ha trasmesso al gruppo una certa sicurezza. Lavorare con esperti è sempre un onore.
- L'Azienda si è dimostrata molto disponibile a chiarimenti di tipo tecnico.
- L'utilizzo di JavaScript il quale interessa alla totalità del gruppo.
- La chiarezza con il quale il proponente ha esposto il dominio del problema nel seminario.

1.6 Criticità e fattori di rischio

- L'apprendimento delle tecnologie richieste potrebbe risultare lento per quei membri del gruppo che non le hanno mai utilizzate.

1.7 Conclusioni

Inizialmente il progetto sembrava complicato per via della parte di manipolazione dei dati, tuttavia l'incontro con la proponente ha chiarito le idee. La parte di modellazione e rappresentazione dei dati rimane una sfida difficile ma che ha catturato l'interesse del gruppo.

2 Capitolato 1 - BlockCOVID

2.1 Informazioni generali

- **Nome:** *BlockCOVID*: supporto digitale al contrasto della pandemia
- **Proponente:** *Imola informatica*
- **Committente:** *Prof. Tullio Vardanega e Prof. Riccardo Cardin*

2.2 Descrizione capitolato

L'obiettivo del capitolato è quello di creare una piattaforma web e un'app che segnalano ad un server il tracciamento delle presenze in tempo reale nel luogo di lavoro e della pulizia delle postazioni di lavoro. Il tracciamento deve essere immutabile e certificato e quindi è necessario utilizzare un sistema di *blockchain*, in questo caso basato su *ethereum*. La piattaforma web è dedicata all'amministratore mentre l'app è dedicata agli utenti (dipendenti e addetto alle pulizie).

2.3 Finalità del progetto

Il proponente descrive i seguenti casi d'uso possibili della piattaforma e app:

- Amministratore
 - Creare, modificare o eliminare postazioni e stanze
 - Creare, modificare o eliminare credenziali per utenti e personale specializzato
 - Monitorare numero di dipendenti in stanze e postazioni
 - Ricercare e effettuare report sui dati di tracciamento
- Utente - dipendente
 - Effettuare prenotazioni di postazioni
 - Posizionare il telefono sul tag RFID in postazioni per ricevere informazioni sullo stato della postazione e/o per segnalare la propria presenza
 - Segnalare l'igienizzazione autonoma della postazione
 - Ricevere il nome di una postazione all'interno di una stanza
- Utente - addetto alle pulizie
 - Ricevere elenco delle stanze (e postazioni) da igienizzare
 - Segnalare una stanza come igienizzata

2.4 Tecnologie interessate

Il proponente lascia la possibilità di scegliere le tecnologie da utilizzare ma consiglia:

- **Java, Python o NodeJS:** linguaggi per parte di server back-end per la creazione di API Rest
- Sistema di **blockchain Ethereum:** piattaforma decentralizzata per la creazione di applicazioni basate su smart-contract

- **IAAS Kubernetes** o di un **PAAS, Openshift** o **Rancher**: strumenti per il rilascio delle componenti server del progetto, utilizzabili anche per gestire la scalabilità delle operazioni
- **Tag RFID**: etichette elettroniche in grado di memorizzare informazioni, in questo caso il codice della postazione; sono leggibili da lettori particolari oppure da molti modelli di smartphone tramite o standard NFC
- **Java** o **Kotlin**: linguaggi per sviluppare l'applicazione per dispositivi Android

2.5 Aspetti positivi

- Il proponente ha specificato in modo dettagliato il capitolato spiegando cosa si aspetta di ottenere.
- La parte di codifica non sembra troppo complessa (però piuttosto lunga) fatta eccezione per l'utilizzo di blockchain.

2.6 Criticità e fattori di rischio

- Le tecnologie da utilizzare sono nuove per quasi tutti i membri del gruppo, in particolare i sistemi di blockchain sono nuovi per tutto il gruppo e sembrano abbastanza complessi da implementare.

2.7 Conclusioni

Il capitolato ha catturato l'interesse della maggior parte dei membri del gruppo, le tecnologie coinvolte e il problema da affrontare sono state valutate stimolanti e formative. Dopo aver partecipato ai numerosi seminari tecnologici ed esserci riuniti in più occasioni si è deciso però di rigettare il capitolato in favore del C4.

3 Capitolato 2 - EmporioLambda

3.1 Informazioni generali

- **Nome:** *EmporioLambda*: piattaforma di e-commerce in stile Serverless
- **Proponente:** *RedBabel*
- **Committente:** *Prof. Tullio Vardanega e Prof. Riccardo Cardin*

3.2 Descrizione capitolato

Lo scopo del capitolato è creare una piattaforma di E-commerce generica, utilizzabile come dimostrazione, vendibile ai commercianti e facilmente configurabile dagli stessi. L'intera piattaforma deve essere costruita nella sua interezza utilizzando tecnologie serverless. EmporioLambda è basato su un'architettura a microservizi, e quindi suddiviso in moduli, gestiti da un BFF (Backend For Frontend). Per la realizzazione del progetto è richiesta l'integrazione di diversi servizi di terze parti, in particolare un payment-provider.

3.3 Finalità del progetto

È richiesta l'implementazione di tutti i moduli facenti parte dell'architettura:

- **Frontend:** si occupa di costruire la pagina HTML così come è stata richiesta dall'utente;
- **Backend:** che implementa la business logic;
- **Integration:** rappresenta tutti i servizi di terze parti utilizzati dal backend;
- **Monitoring** un set di strumenti utilizzati dall'admin per monitorare lo stato dell'applicazione.

Sono inoltre richiesti una serie di requisiti funzionali per ognuna delle parti di cui è composto EmporioLambda, e opzionali.

3.4 Tecnologie interessate

Il proponente richiede l'utilizzo di architetture serverless, in particolare i servizi offerti da Amazon tramite AWS Lambda.

- **CloudFormation:** servizio suggerito per la gestione delle risorse AWS e di terze parti utilizzate
- **Serverless Framework** strumento che nasce per gestire il lifecycle di applicazioni serverless
- **Typescript:** linguaggio di programmazione
- **NodeJS:** framework per la realizzazione del BFF, e per renderizzare e servire l'interfaccia al browser
- **Amazon CloudWatch** o **Datadog:** strumenti che permettono il monitoraggio di applicazioni

- **AWS Cognito Identity:** servizio per la gestione dei diversi tipi di utenza (utente, commerciante, amministratore)
- **Github o Gitlab:** servizi di versionamento
- **Stripe:** modulo di terze parti che agisce come payment-provider.

3.5 Aspetti positivi

- Il proponente ha presentato un capitolato d'appalto completo e difficilmente si verificheranno imprevisti durante lo sviluppo del progetto, inoltre si dimostra molto disponibile, mettendo a disposizione diversi canali di comunicazione, come email e Slack.
- Il fornitore incita al confronto e all'esplorazione per quanto riguarda le aree non trattate con precisione nel documento.
- È richiesto l'utilizzo di tecnologie rilevanti nel panorama odierno, in particolare i servizi AWS e la programmazione funzionale.

3.6 Criticità e fattori di rischio

- Il capitolato descrive una grande quantità di tecnologie, il rischio principale è che i costi di formazione non siano sostenibili
- Nessun componente del gruppo di lavoro ha conoscenze specifiche vantaggiose ai fini della realizzazione del progetto.

3.7 Conclusioni

Sebbene sia stata ritenuta stimolante, la proposta di RedBabel è stata comunque scartata, per via della mole di formazione necessaria, ritenuta proibitiva.

4 Capitolato 3 - GDP

4.1 Informazioni generali

- **Nome:** *GDP: Gathering Detection Platform*
- **Proponente:** *Sync Lab S.r.l.*
- **Committente:** *Prof. Tullio Vardanega e Prof. Riccardo Cardin*

4.2 Descrizione capitolato

Lo scopo del progetto è lo sviluppo di una piattaforma interattiva in grado di raccogliere i dati riguardanti il flusso di persone da varie fonti e provvedere alla costruzione di un modello tramite **Machine Learning** capace di effettuare previsioni sulla affluenza nei vari luoghi in modo da mostrare tramite *heatmap* la situazione di tali flussi.

4.3 Finalità del progetto

Il prodotto dovrà offrire la possibilità di raccogliere, analizzare e flussi di dati riguardante lo spostamento di persone, in particolare:

- Vengono raccolti i dati grezzi da sorgenti eterogenee, e processati
- Questi vengono poi archiviati in un database
- Eventualmente vengono usati per l'online learning di un modello precedentemente costruito tramite machine learning
- Questi vengono mostrati con bassa latenza su una heatmap, o nel caso venga richiesta una previsione, vengono forniti al modello
- L'utilizzatore prende una decisione basandosi sull'output fornito

4.4 Tecnologie interessate

- **Web development** L'azienda preferisce l'utilizzo di java per lo sviluppo back-end e Angular per il front-end.
- **Leaflet:** Framework javascript per la gestione di mappe.
- **Machine Learning:** L'azienda non suggerisce particolare restrizione alle librerie da utilizzare o agli algoritmi da adottare.
- **Apache Kafka:** Piattaforma open-source per il processo di stream di dati in tempo reale.

4.5 Aspetti positivi

- Viene lasciata agli sviluppatori la scelta riguardo l'implementazione degli algoritmi di machine learning
- La conoscenza di alcune librerie di machine learning e delle tecnologie proposte nel progetto potrebbe rivelarsi utile per un futuro nel mondo del lavoro
- Il capitolato suggerisce le tecnologie e i pattern da adottare

4.6 Criticità e fattori di rischio

- Alta quantità di tecnologie nuove, che richiedono una quantità di studio non indifferente
- Richiesti alti standard di qualità

4.7 Conclusioni

Il capitolato non riesce a catturare l'attenzione del gruppo, sia per il tema trattato, sia per la mole di tecnologie sconosciute, e il conseguente costo in termini temporali volto all'assimilazione delle stesse troppo grande.

5 Capitolato 5 - PORTACS

5.1 Informazioni generali

- **Nome:** *PORTACS*: piattaforma di controllo mobilità autonoma
- **Proponente:** *Sanmarco Informatica*
- **Committente:** *Prof. Tullio Vardanega e Prof. Riccardo Cardin*

5.2 Descrizione capitolato

Il capitolato chiede lo sviluppo di un software che visualizza in tempo reale delle unità generiche all'interno di una griglia il cui scopo è dirigersi verso dei POI (Point of Interest). Questa rappresentazione astratta trova applicazione in vari ambiti pratici, per esempio, nella geolocalizzazione di magazzino, di trasporti o dei robot camerieri.

5.3 Finalità del progetto

- Ogni unità principale contiene un identificativo, una velocità massima, una posizione iniziale e una lista di POI che deve attraversare.
- La griglia deve contenere i POI e definire al suo interno delle definizioni di percorrenza, come ad esempio dei sensi unici, o percorsi non attraversabili.
- Il sistema centrale dovrà muovere le unità facendole raggiungere i POI assegnati rispettando i vincoli di percorrenza della griglia ed evitando collisioni tra di loro.
- Il software può prevedere la presenza di pedoni all'interno della griglia, le unità devono essere in grado di evitare collisioni anche con i pedoni.

5.4 Tecnologie interessate

Nel capitolato non vengono specificate particolari tecnologie da utilizzare per la realizzazione del software. Le tecnologie elencate di seguito sono state individuate dai membri del gruppo.

- Libreria Pathfinding per **Python**: utile per trovare i percorsi all'interno della griglia
- **React**: libreria JavaScript per lo sviluppo della real-time user interface

5.5 Aspetti positivi

- L'obiettivo presentato dal proponente è interessante e il suo sviluppo una sfida stimolante.
- L'ottimizzazione dei percorsi non è una richiesta obbligatoria, ciò semplifica considerevolmente l'implementazione degli algoritmi di path finding.

5.6 Criticità e fattori di rischio

- Lo sviluppo del progetto richiede abilità di problem-solving.
- Difficoltà nel prevedere i tempi di realizzazione.

5.7 Conclusioni

Sebbene il capitolato abbia destato interesse in alcuni membri del gruppo, non è stato scelto per mancanza di unanimità nel giudicare la difficoltà dello sviluppo di un sistema real-time, in particolare, il problema di definire in modo abbastanza rigoroso i costi che avrebbe richiesto in termini di tempo hanno convinto il gruppo a focalizzarsi su altri capitolati.

6 Capitolato 6 - RGP

6.1 Informazioni generali

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

6.2 Descrizione capitolato

Il capitolato prevede la realizzazione di un videogioco a scorrimento verticale in stile Aero Fighters per dispositivi mobile, deve implementare sia una modalità di gioco in singleplayer e una in multiplayer. Il proponente richiede che venga tenuta in considerazione la scalabilità dell'applicazione, soprattutto lato server. Richiede che venga sviluppata l'applicazione mobile per almeno uno dei principali sistemi operativi per smartphone consigliando iOS (Swift/SwiftUI) fornendo anche un certificato per effettuare i test sui dispositivi. Il proponente richiede di effettuare un'analisi preliminare su quale tecnologia AWS sia meglio utilizzare, schema dell'architettura cloud, una documentazione dettagliata delle API e un piano di test di unità. Il tutto dovrà poi essere motivato e consegnato al proponente.

6.3 Finalità del progetto

La modalità multiplayer del gioco deve essere sviluppata su tecnologia Amazon Web Service (AWS) e permetterà all'utente di monitorare i progressi degli altri giocatori in tempo reale, vince l'ultimo giocatore a non aver esaurito le vite. La modalità single player invece sarà una modalità infinita, terminerà solo con l'esaurimento delle vite.

6.4 Tecnologie interessate

- **NodeJS**
- **Swift/SwiftUI**
- **Kotlin**
- **Python**
- **Bootstrap:** framework il quale richiede conoscenze di HTML5, CSS3 e JavaScript
- **AWS:** in particolare bisogna scegliere tra AWS GameLift o AWS Appsync

6.5 Aspetti positivi

- Usufruire dei servizi AWS permettendoci di studiare e approfondire le conoscenze dei servizi AWS
- Sviluppo mobile iOS/Android
- Il proponente sembra essere molto disponibile e interessato allo sviluppo del progetto, fornendo servizi e supporto al team.

6.6 Criticità e fattori di rischio

- Richiede lo sviluppo nativo sia iOS e/o Android. Negando di fatto la possibilità di usare motori grafici come Unity o UE che permetterebbero di ridurre l'ammontare di ore necessario allo sviluppo in quanto permettono la compilazione per entrambi i SO mobile.

6.7 Conclusioni

Il progetto ha catturato l'interesse solo della minoranza dei membri del gruppo, il capitolato è stato rigettato.

7 Capitolato 7 - SSD

7.1 Informazioni generali

- **Nome:** *SSD*: soluzioni di sincronizzazione desktop
- **Proponente:** *Zextras*
- **Committente:** *Prof. Tullio Vardanega e Prof. Riccardo Cardin*

7.2 Descrizione capitolato

Il Capitolato propone una soluzione dei problemi di sincronizzazione desktop per utenti professionali, quindi di sviluppo di una piattaforma dove gli utenti possano lavorare dal loro dispositivo ma mantenere una copia del lavoro salvata su cloud. Gli obbiettivi in particolare sono 3:

- Lo sviluppo di un algoritmo solido ed efficiente in grado di garantire il salvataggio in cloud ed un lavoro contemporaneo sincronizzato dai cambiamenti
- Lo sviluppo di un'interfaccia multipiattaforma (MacOS, Windows, Linux)
- L'utilizzo dell'algoritmo in sincronizzazione verso il prodotto già in commercio del proponente Zextras Drive.

7.3 Finalità del progetto

Il proponente richiede le seguenti 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

Inoltre come funzionalità avanzate propone:

- Gestione delle condivisioni, integrazione protocollo MAPI e integrazione con il prodotto web

7.4 Tecnologie interessate

Il proponente lascia al gruppo la decisione delle tecnologie usate, ma consiglia l'uso delle seguenti tecnologie:

- **Qt Framework:** framework basato su C++. Consigliato per lo sviluppo dell'interfaccia data la sua ottima performance e la possibilità di creare widget.
- **Python:** linguaggio per lo sviluppo della Buisness Logic e del Beckend. In particolare la Standard Library di Python.

7.5 Aspetti positivi

- Il framework consigliato Qt è stato già visto e usato da tutti i membri del gruppo, come anche l'architettura del progetto, e quindi necessita di un approfondimento minore

7.6 Criticità e fattori di rischio

- La tempistiche di apprendimento degli algoritmi richiesti e dello sviluppo degli stessi possono sfociare facilmente in un prolungamento eccessivo del progetto.
- Il proponente risulta molto vago nelle specifiche delle problematiche di contesto
- Non esiste una classificazione concreta di "utente professionista" che dovrebbe utilizzare il prodotto

7.7 Conclusioni

Lo scopo del capitolato e le tecnologie annesse non hanno suscitato interesse nella grande maggioranza dei componenti del gruppo. Inoltre le problematiche di tempo richieste hanno spostato l'attenzione dei membri verso altri capitolati che rispecchiano l'interesse dei membri del gruppo, e che assicurano uno svolgimento più efficace e di maggior crescita per gli stessi.