



Studio di fattibilità

techsweave@gmail.com

Informazioni sul documento

Versione	0.2
Responsabile	
Redattori	Nicolò Giaccone Samuele De Simone
Verificatore	
Destinatari	Prof. Tullio Vardanega Prof. Riccardo Cardin, TechSWEave
Stato	Da approvare
Uso	Interno

Diario delle modifiche

- *Prima stesura del documento*, Nicolò Giaccone,(ruolo da definire), v0.1;

Indice

1	Capitolato scelto: C2 - EmporioLambda	3
1.1	Informazioni generali	3
1.2	Descrizione del capitolato	3
1.3	Finalità del progetto	3
1.4	Tecnologie interessate	3
1.5	Aspetti positivi	5
1.6	Criticità e fattori di rischio	5
1.7	Conclusioni	5
2	Capitolato C1 - BlockCOVID	6
3	Capitolato C3 - Gathering Detection Platform	6
4	Capitolato C4 - HD Viz	6
5	Capitolato C5 - PORTACS	6
5.1	Informazioni generali	6
5.2	Descrizione del capitolato	6
5.3	Finalità del progetto	6
5.4	Tecnologie interessate	6
5.5	Aspetti positivi	6
5.6	Criticità e fattori di rischio	7
5.7	Conclusioni	7
6	Capitolato C6 - Realtime Gaming Platform	7
7	Capitolato C7 - soluzioni di sincronizzazione desktop	7

1 Capitolato scelto: C2 - EmporioLambda

1.1 Informaizoni genarli

- **Titolo:** *EmporioLambda*;
- **Proponente:** *RedBabel*;
- **Commitenti:** *Prof. Tullio Vardanega, Prof. Riccardo Cardin*;

1.2 Descrizione del capitolato

Il capitolato richiede la realizzazione di una piattaforma e-commerce basata interamente sulle tecnologie serverless che possa essere utilizzata dal proponente come prototipo da mostrare ad altre aziende. EmporioLambda deve poter essere utilizzato da un ipotetico commerciante con il minimo quantitativo di configurazione manuale tramite account AWS Merchant. Il capitolato prevede che vengano fornite alcune funzioni irrinunciabili per tutte le categorie di utenti che ne faranno uso: clienti, commercianti, admin.

1.3 Finalità del progetto

La finalità del progetto è quella di realizzare una applicazione in grado di gestire, lo spostamento di un unità all'interno di un area controllata, attraverso la consocenza di alcuni dati e con l'obbiettivo di minimizzare costi e tempi di viaggio e di evitare eventuali collisioni

1.4 Tecnologie interessate

- **Architettura serverless:** Un'architettura serverless è un'applicazione di progettazione che include servizi "Backend as aServiceG" (BaaS) di terze parti, codice personalizzato e container temporanei su una piattaforma di esecuzione "Function as a ServiceG" (FaaS). Le architetture Serverless potrebbero trarre beneficio dalla riduzione dei costi delle operazioni, complessità e tempi di consegna. Essi sono sistemi basati su cloud di eventi, dove lo sviluppo di applicazioni è basato esclusivamente su una combinazione di servizi di terze parti, logica lato client e chiamate a procedure presenti nel cloud.
- **AWS Lambda:** WS Lambda è una piattaforma di elaborazione serverless basata su eventi fornita da Amazon come parte di Amazon Web Services. È un servizio di elaborazione che esegue il codice in risposta agli eventi e automaticamente gestisce le risorse di calcolo necessarie per l'esecuzione di quel codice.
- **Serverless Framework:** Serverless Framework è un framework Web gratuito e open source scritto utilizzando Node.js. Serverless è il primo framework sviluppato per la creazione di applicazioni su AWS Lambda.
- **Next.js:** Next.js è un framework web di sviluppo front-end React open source che abilita funzionalità come il rendering lato server e la generazione di siti web statici per applicazioni web basate su React. Fornisce possibilità di integrazione con Typescript.
- **Typescript:** TypeScript è un Super-set di JavaScript. Il linguaggio estende la sintassi di JavaScript in modo che qualunque programma scritto in JavaScript sia anche in grado di funzionare con TypeScript senza nessuna modifica.

- **ESLint:** Strumento che permette di verificare che il codice TypeScript scritto rispetti correttamente la sintassi di linguaggio.
- **HTML:** Html, acronimo di hyper-text mark-up language, permette di disporre gli elementi all'interno di una pagina Web, ma è sempre più utilizzato anche per la realizzazione di contenuti e applicazioni mobile.
- **CSS:** CSS, acronimo di cascade style sheet, è un linguaggio usato per definire la formattazione di documenti HTML, XHTML e XML, ad esempio i siti web e relative pagine web.
- **Amazon CloudWatch:** Amazon CloudWatch è un servizio di monitoraggio e osservabilità creato per ingegneri, sviluppatori, ingegneri responsabili dell'affidabilità del sito e manager IT DevOps.
- **Identity manager:** Sistema in grado di consentire alle organizzazioni di facilitare - e al tempo stesso controllare - gli accessi degli utenti ad applicazioni e dati critici, proteggendo contestualmente i dati personali da accessi non autorizzati.
- **AWS Amazon Cognito:** Amazon Cognito permette di aggiungere strumenti di registrazione degli utenti, accesso e controllo degli accessi alle app Web e per dispositivi mobili.
- **Payment service:** Offre servizi online ad enti, negozi e commercianti per accettare pagamenti elettronici con una varietà di metodi di pagamento, tra cui carta di credito, pagamenti basati su banca come addebito diretto, trasferimento bancario e trasferimento bancario in tempo reale basato sul modello di Banca online.
- **Stripe:** Fornisce un'infrastruttura che permette a privati e aziende di inviare e ricevere pagamenti via internet.

Inoltre nel capitolato sono citate altre tecnologie che il proponente consiglia o indica come interessanti, ma di cui non impone l'utilizzo:

- **AWS S3:** AWS S3, acronimo di Amazon Web Services Amazon Simple Storage, è un servizio Web di memorizzazione offerto da Amazon Web Services;
- **Auth0:** Piattaforma di autenticazione e autorizzazione;
- **Datadog:** Datadog è un servizio di monitoraggio per applicazioni su scala cloud, che fornisce il monitoraggio di server, database, strumenti e servizi, attraverso una piattaforma di analisi dei dati basata su SaaS;
- **AWS CloudFormation:** AWS CloudFormation offre un modo semplice per modellare una raccolta di risorse AWS e di terze parti, effettuarne il provisioning in modo rapido e coerente e gestirle in tutto il loro ciclo di vita, trattando l'infrastruttura come codice;
- **AWS API Gateway:** Amazon API Gateway è un servizio che semplifica per gli sviluppatori la creazione, la pubblicazione, la manutenzione, il monitoraggio e la protezione delle API su qualsiasi scala;
- **AWS DynamoDB:** Amazon DynamoDB è un database che supporta i modelli di dati di tipo documento e di tipo chiave-valore che offre prestazioni di pochi millisecondi a qualsiasi scala. Si tratta di un database durevole, multiregione, multiattivo e completamente gestito che offre sicurezza, backup e ripristino integrati e memorizzazione nella cache in memoria per applicazioni Internet;

- **Content management system:** Un content management system, in acronimo CMS, è uno strumento software, installato su un server web, il cui compito è facilitare la gestione dei contenuti di siti web, svincolando il webmaster da conoscenze tecniche specifiche di programmazione Web.

1.5 Aspetti positivi

- L'argomento del capitolato è di interesse per tutti i membri del gruppo.
- Le tecnologie coinvolte nello sviluppo del progetto sono innovative e di interesse per tutti i membri del gruppo.
- Sia l'azienda che la sua proposta appaiono moderne e meritevoli di attenzioni, padroneggiare le tecnologie richieste dal capitolato significherebbe investire in formazione concreta e non in un progetto didattico fine a sé stesso.
- La chiarezza dell'esposizione del capitolato, riscontrati sia nel documento che nel video di presentazione.

1.6 Criticità e fattori di rischio

- L'apprendimento delle tecnologie coinvolte, dato il loro numero e complessità, potrebbe richiedere molto tempo in quanto nessuno dei membri del gruppo dispone di esperienza pregressa con la maggior parte di esse.
- Occorre tenere in considerazione la limitatezza delle risorse di cui il gruppo dispone, in particolar modo del tempo materiale, e la costante penalizzazione sul piano organizzativo e gestionale che comporta la situazione pandemica in essere.
- Il proponente ha residenza all'estero e la comunicazione potrebbe rivelarsi più problematica rispetto a quella con un'azienda locale.

1.7 Conclusioni

Le tecnologie richieste per la realizzazione della proposta di RedBabel hanno fin da subito suscitato l'interesse dei membri del gruppo. Nel dettaglio, la possibilità di approfondire paradigmi e tecnologie che esulano o che non vengono approfonditi dal programma di studi della Laurea Triennale, come la programmazione funzionale asincrona e l'architettura serverless, ha ricoperto un ruolo fondamentale nel designare questo capitolato come uno dei preferibili. Sarebbe inoltre inopportuno trascurare il fatto che, facendo parte dei gruppi del secondo lotto del corso didattico di Ingegneria del Software, il gruppo TechSWEave ha avuto possibilità limitate nella scelta del capitolato. Alla luce di queste considerazioni e nonostante le difficoltà che potrebbero presentarsi e la complessità percepita per lo sviluppo del software richiesto, il gruppo TechSWEave ha deciso di optare per la realizzazione di questo capitolato, fiducioso del fatto che i punti non ancora completamente chiari saranno adeguatamente sviluppati e compresi in sinergia con il proponente. Questo dovrà ovviamente andare di pari passo con la consapevolezza, da parte di ciascun membro del gruppo, della necessità di impegnarsi al proprio meglio al fine di apprendere e padroneggiare l'uso delle tecnologie che questo progetto impone.

2 Capitolato C1 - BlockCOVID

3 Capitolato C3 - Gathering Detection Platform

4 Capitolato C4 - HD Viz

5 Capitolato C5 - PORTACS

5.1 Informaizoni genarli

- **Titolo:** *PORTACS*;
- **Proponente:** *Sanmarco Informatica*;
- **Commitenti:** *Prof. Tullio Vardanega, Prof. Riccardo Cardin*;

5.2 Descrizione del capitolato

Il capitolato richiede un software in grado di gestire, in real-time, una serie di unità con dei punti di interesse da raggiungere evitando le collisioni con altre unità del sistema. Presenta anche una *UI* che rappresenterà una mappa in cui sono indicate tutte le unità controllate con un'interfaccia che indica la prossima direzione consigliata all'unità di cui saranno indicati: id, un pulsante stop/start, velocità attuale.

5.3 Finalità del progetto

La finalità del progetto è quella di realizzare una web-app che ricevuti degli input, possa creare una mappa, all'interno della quale censire le varie unità operative (id, velocità massima e di crociera, posizione iniziale, lista dei POI da raggiungere,) e di consigliare alle unità la prossima mossa da eseguire tenendo conto delle dimensioni delle unità e di eventuali collisioni con altre unità presenti all'interno del sistema

5.4 Tecnologie interessate

- La scelta delle tecnologie da utilizzare è stata lasciata agli sviluppatori del progetto, sono comunque stati indicati alcune tecnologie :
 - A seconda dell'approccio scelto per il real-time sono stati consigliati determinati linguaggi come derivati del C, o linguaggi multi-thread
 - Per la ui essendo una web-app, sono stati consigliati framework come React ed Angular.
- **Docker:** progetto open-source che automatizza il deployment di applicazioni all'interno di container.

5.5 Aspetti positivi

- L'obiettivo del progetto è in linea con le problematiche tecnologiche attuali;
- Ha una grande valenza formativa per i componenti del gruppo

5.6 Criticità e fattori di rischio

- Il capitolato non presentava più posti disponibili;

5.7 Conclusioni

Il capitolato per quanto di forte interesse e utilità non presenta posti disponibili e non è stato quindi possibile sceglierlo.

6 Capitolato C6 - Realtime Gaming Platform

7 Capitolato C7 - soluzioni di sincronizzazione desktop