



---

## Studio di fattibilità

---

techsweave@gmail.com

### Informazioni sul documento

<b>Versione</b>	0.6
<b>Responsabile</b>	
<b>Redattori</b>	Nicolò Giaccone Samuele De Simone
<b>Verificatore</b>	
<b>Destinatari</b>	Prof. Tullio Vardanega Prof. Riccardo Cardin, TechSWEave
<b>Stato</b>	Da approvare
<b>Uso</b>	Interno

## Diario delle modifiche

Modifica	Autore	Ruolo	Data	Versione
Prima stesura documento	Nicolò Giaccone	Analista	11/03/2021	0.1
Revisione struttura Aggiunta capitolati C2, C5	Samuele De Simone	Analista	11/03/2021	0.2
Aggiunta capitolato C6	Tito Scutari	Analista	11/03/2021	0.3
Aggiunta dei capitolati C3 e C4	Nicolò Giaccone	Analista	11/03/2021	0.5
Revisione capitolato C6	Samuele De Simone	Analista	12/03/2021	0.5.1
Aggiunta capitolato C1	Simone Urbani	Analista	12/03/2021	0.6

# Indice

<b>1 Capitolato scelto: C2 - EmporioLambda</b>	<b>3</b>
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 . . . . .	4
1.6 Criticità e fattori di rischio . . . . .	5
1.7 Conclusioni . . . . .	5
<b>2 Capitolato C1 - BlockCOVID</b>	<b>6</b>
2.1 Informazioni generali . . . . .	6
2.2 Descrizione del 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
<b>3 Capitolato C3 - Gathering Detection Platform</b>	<b>8</b>
3.1 Informazioni generali . . . . .	8
3.2 Descrizione del capitolato . . . . .	8
3.3 Finalità del progetto . . . . .	8
3.4 Tecnologie interessate . . . . .	8
3.5 Aspetti positivi . . . . .	8
3.6 Criticità e fattori di rischio . . . . .	8
3.7 Conclusione . . . . .	8
<b>4 Capitolato C4 - HD Viz</b>	<b>9</b>
4.1 Informazioni generali . . . . .	9
4.2 Descrizione capitolato . . . . .	9
4.3 Finalità del progetto . . . . .	9
4.4 Tecnologie . . . . .	9
4.5 Aspetti positivi . . . . .	9
4.6 Aspetti negativi . . . . .	9
4.7 Conclusioni . . . . .	9
<b>5 Capitolato C5 - PORTACS</b>	<b>10</b>
5.1 Informazioni generali . . . . .	10
5.2 Descrizione del capitolato . . . . .	10
5.3 Finalità del progetto . . . . .	10
5.4 Tecnologie interessate . . . . .	10
5.5 Aspetti positivi . . . . .	10
5.6 Criticità e fattori di rischio . . . . .	10
5.7 Conclusioni . . . . .	10
<b>6 Capitolato C6 - RGP: Realtime Gaming Platform</b>	<b>11</b>
6.1 Informazioni . . . . .	11
6.2 Descrizione . . . . .	11
6.3 Finalità progetto . . . . .	11
6.4 Tecnologie impiegate . . . . .	11
6.5 Aspetti positivi . . . . .	11
6.6 Criticità e fattori di rischio . . . . .	12
6.7 Conclusioni . . . . .	12
<b>7 Capitolato C7 - soluzioni di sincronizzazione desktop</b>	<b>13</b>

# 1 Capitolato scelto: C2 - EmporioLambda

## 1.1 Informazioni generali

- **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

EmporioLambda dovrà consistere di 4 moduli:

- **EmporioLambda-frontend:** si occupa di gestire le pagine richieste dall'utente. Il compito di questo modulo è pre-renderizzare le pagine in HTML sul server ad ogni richiesta. Dovrà essere implementato utilizzando Next.js e TypeScript;
- **EmporioLambda-backend:** si occupa di gestire i servizi dell'applicazione e cioè gestire la business-logic, i dati relativi ad utenti, prodotti ed ordini, lo stato del carrello. Dovrà essere implementato utilizzando Serverless Framework, TypeScript e AWS Lambda;
- **EmporioLambda-integration:** rappresenta tutti i servizi di terze parti che interagiscono con EmporioLambda-backend. Dovrà essere implementato utilizzando Serverless Framework e TypeScript;
- **EmporioLambda-monitoring:** rappresenta gli strumenti impiegati dagli admin per monitorare lo stato dell'applicazione. Dovrà essere implementato con Amazon CloudWatch.

## 1.4 Tecnologie interessate

Il proponente richiede l'utilizzo delle seguenti tecnologie:

- **Architettura serverless:** Un'architettura serverless è un'applicazione di progettazione che include servizi "Backend as a ServiceG" (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 CoudWatch:** 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.

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

### 2.1 Informazioni generali

- **Titolo:** *BlockCOVID*;
- **Proponente:** *Imola Informatica*;
- **Commitenti:** *Prof. Tullio Vardanega, Prof. Riccardo Cardin*;

### 2.2 Descrizione del capitolato

Il capitolato richiede di realizzare un'applicazione in grado di monitorare le postazioni di più stanze in modo da permetterne un'efficace tracciamento della sanificazione delle stesse in maniera immutabile e certificata. Deve essere quindi possibile tracciare in tempo reale l'occupazione delle postazioni e la loro prenotazione, considerando che le postazioni da sanificare non possono essere utilizzate. Deve essere possibile anche tracciare la sanificazione di tali postazioni, sia che siano fatte da aziende terze specializzate, sia che siano fatte manualmente dall'utilizzatore della postazione.

### 2.3 Finalità del progetto

La finalità del progetto è quella di creare un server completo di *UI* per la gestione delle stanze e delle postazioni e un'applicazione mobile per Android e iOS. Il server deve poter esporre queste funzionalità:

- Prevedere un sistema di autenticazione e ruoli.
- Creare e modificare stanze e postazioni.
- Monitorare lo stato delle postazioni: occupate, prenotate da pulire e pulite.
- Esportazione di un report delle pulizie per ogni singola postazione o per stanza.

L'applicazione deve permettere le seguenti operazioni:

- Visualizzare le postazioni libere ed eventualmente prenotarne una.
- Tracciamento in tempo reale della postazione utilizzata.
- Sanificazione di una postazione.
- Storico delle postazioni occupate e di quelle igienizzate.

La comunicazione tra server e applicazione mobile avviene nel momento in cui lo smartphone dell'utente viene a contatto con il tag RFID della postazione. In questo modo è possibile sapere che la postazione sta venendo usata da un utente, e quindi sarà da sanificare una volta liberata.

### 2.4 Tecnologie interessate

La scelta delle tecnologie da utilizzare è in parte data dal proponente, in parte lasciata a discrezione agli sviluppatori del progetto. È comunque possibile utilizzare tecnologie diverse da quelle obbligatorie discutendone preventivamente con il proponente.

- Tecniche per raggiungere i requisiti minimi:
  - Possibilità di usare l'applicativo attraverso delle API Rest esposte dal server. In alternativa, è possibile usare gRPC a Rest.
  - Lettore RFID integrato nei dispositivi mobili, solamente per il tempo strettamente necessario.
  - Test unitari e d'integrazione per le componenti applicative, con test end-to-end sull'intero sistema.
- Tecniche suggerite, ma non vincolanti:
  - Java (versione 8 o superiori), Python o Node.js per lo sviluppo del server back-end.
  - Utilizzo di protocolli asincroni per la comunicazione tra app e server.
  - Un sistema blockchain per salvare con opponibilità a terzi i dati di sanificazione.
  - AAS Kubernetes o di un PAAS, Openshift o Rancher, per il rilascio delle componenti del server e la gestione della scalabilità orizzontale.

## **2.5 Aspetti positivi**

- Utilizzo di svariate tecnologie recenti e all'avanguardia.
- Integrazione tra dispositivi mobili e postazioni fisse.
- Possibilità di realizzare un software di grande utilità ad aziende e personale messe in difficoltà dal Covid-19.
- Possibilità di realizzare un progetto per un'azienda con molta esperienza e interessata alle tecnologie open-source.

## **2.6 Criticità e fattori di rischio**

- Il capitolato non presenta posti disponibili.
- Nessun membro del gruppo ha abbastanza conoscenze riguardanti le tecnologie da utilizzare.
- L'immutabilità e la certificazione del tracciamento sono questioni molto delicate, in quanto sono previste anche sanzioni penali per un datore di lavoro che non possa dimostrare di aver garantito la salubrità dell'ambiente di lavoro.

## **2.7 Conclusioni**

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

### 3 Capitolato C3 - Gathering Detection Platform

#### 3.1 Informazioni generali

- **Titolo:** *Gathering Detection Platform*;
- **Proponente:** *Sync Lab*;
- **Commitenti:** *Prof. Tullio Vardanega, Prof. Riccardo Cardin*;

#### 3.2 Descrizione del capitolato

Il capitolato richiede un software che ricevendo dati da dispositivi installati ed operativi in specifiche zone, vada ad identificare il possibile formarsi di flussi di utenti, con l'intento di evitare assembramenti per scongiurare il diffondersi di Covid-19. L'interfaccia grafica dell'applicazione sarà caratterizzata da una presenza di *heat-map* per indicare l'eventuale presenza di assembramenti in una determinata area.

#### 3.3 Finalità del progetto

Il progetto consiste nella realizzazione di una web-app collegata ad un server dotato di gui. All'interno di questo server verranno salvati tutti i dati ricevuti dai dispositivi analizzati (il numero di persone presenti all'interno di un mezzo di trasporto pubblico, l'affluenza media in un centro commerciale in un determinato orario, flussi prenotazioni Uber, contapersone etc...). Con questi dati attraverso algoritmi di machine learning e tecniche di *Predictive Analytics* verranno create delle stime da fornire poi agli utenti dell'applicazione. Queste stime potranno essere visualizzate dagli utenti attraverso heat-map, grafici e previsioni che useranno per evitare le zone di maggiore rischio.

#### 3.4 Tecnologie interessate

- **Java e Angular:** per sviluppare sia le parti di Back-end che di Front-end della Web Application.
- **Leaflet:** framework per la gestione delle mappe (heatmao ecc.).
- **Protocolli asincroni** per le comunicazioni tra i componenti.
- **Pattern Publisher/Subscriber.**
- **MQTT** (Message Queue Telemetry Transport): protocollo di messaggistica open, di facile interpretazione e leggero, molto diffuso in applicazioni M2M e IoT.

#### 3.5 Aspetti positivi

- L'argomento proposto è importante per la situazione attuale nella lotta alla pandemia globale.
- Il gruppo ritiene che sarebbe stato interessante sviluppare e studiare tecnologie di machine learning

#### 3.6 Criticità e fattori di rischio

- Alcune tecnologie proposte dal capitolato non hanno riscosso grande interesse nella maggior parte del gruppo.
- Il lavoro da svolgere per completare i requisiti è sembrato molto di più rispetto ad altri capitolati studiati.
- Il capitolato era già stato scelto dal numero massimo di gruppi e quindi non risultava più disponibile.

#### 3.7 Conclusione

Il capitolato, in quanto membri del secondo lotto, non presenta più posti disponibili, inoltre non suscitava l'interesse del gruppo in quanto presenta tecnologie non conosciute, quali machine learning e kafka. Inoltre il capitolato ha funzionalità molto estese che avrebbero aumentato il lavoro da svolgere per il gruppo. Tutto questo ha fatto desistere il gruppo dal scegliere il capitolato anche se l'argomento trattato rappresentava un grande interesse in quanto ha il proposito di aiutare a contrastare la pandemia Covid-19.

## 4 Capitolato C4 - HD Viz

### 4.1 Informazioni generali

- Nome : HD Viz;
- Proponente : Zucchetti S.p.A;
- Committente : *Prof Tullio Vardanega e Prof Riccardo Cardin.*

### 4.2 Descrizione capitolato

Il capitolato richiede di realizzare un'applicazione per la visualizzazione di dati con molte dimensioni a supporto della fase esplorativa dell'analisi dei dati. L'obiettivo attraverso l'uso di tale applicazione è di ottenere visualizzazioni più semplici e comprensibili per l'utente finale.

### 4.3 Finalità del progetto

Allo scopo di sviluppare modelli semplificati che evidenzino delle anomalie in alcuni casi analizzati sarà necessario l'utilizzo di algoritmi di AI che agiscono sui concetti che variano dal grafico preso in analisi.

### 4.4 Tecnologie

Le tecnologie richieste per la realizzazione del prodotto sono

- HTML/CSS/JavaScript utilizzando la libreria D3.js per la realizzazione del lato Front-End dell'applicazione;
- Tomcat o Node.js per lo sviluppo di un server database(SQL o NoSQL), in supporto al lato browser

### 4.5 Aspetti positivi

- Tecnologie come HTML/CSS/JAVAScript sono state già utilizzate in corsi e progetti precedenti.
- La libreria D3.js fornisce già alcune delle visualizzazioni richieste.
- il capitolato tratta l'implementazione di un'intelligenza artificiale aspetto molto interessante.
- Lo sviluppo di applicazioni per l'analisi dei Big Data è un problema molto comune e affrontato in molti settori oggi giorno e può risultare utile in ambito professionale futuro.

### 4.6 Aspetti negativi

- La libreria D3.js è molto ampia e per imparare a padroneggiarla può essere richiesto molto tempo.
- Lavorando con dati grandi dimensioni comporta si potranno riscontrare difficoltà nei test degli algoritmi.
- Altri capitoli presentano scenari di lavoro più concreti rispetto all'analisi ed elaborazione dei dati.

### 4.7 Conclusioni

Il capitolato non presenta più posti disponibili, ma il gruppo è comunque interessato a lavorare su scenari differenti, non sarebbe quindi stata una delle prime scelte.

## 5 Capitolato C5 - PORTACS

### 5.1 Informazioni generali

- **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 deploymentGdi 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 - RGP: Realtime Gaming Platform

### 6.1 Informazioni

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

### 6.2 Descrizione

Il progetto prevede la realizzazione di un gioco ascorrimento verticale giocabile due modalità: single player e multiplayer. Il gioco tuttavia non prevede interazione fra gli attori dei giocatori ma semplicemente una sincronizzazione dei nemici power up e altre componenti in modo che la sfida sia la medesima.

### 6.3 Finalità progetto

Il progetto richiede:

- Ricerca della soluzione AWS più adatta;
- Sviluppo della componente server e della sincronizzazione per la modalità multiplayer;
- Sviluppo del gioco per mobile, scegliendo una piattaforma tra iOS e android.

Sono inoltre richiesti i seguenti materiali prima di procedere allo sviluppo:

- Scelta servizio AWS;
- Schema architettura cloud;
- Documentazione dell'API;
- Piano di test unità;

Infine i seguenti materiali sono da consegnare al termine dello sviluppo

- Report sulle scelte della tecnologia AWS;
- Configurazione dell'architettura cloud (con eventualmente codice sorgente);
- codice sorgente dell'app.

Le parti di codice sono da consegnare mediante repository GIT.

### 6.4 Tecnologie impiegate

- **AWS**, soluzioni dedicate all'hosting per l'architettura serverless:
  - **GameLift**, apposito per il gaming
  - **Appsync**, alternativa generica
- **NodeJS**, environment per eseguire Javascript fuori dal browser, da utilizzare se il servizio AWS impiegato richiede scrittura di codice.
- Linguaggi di programmazione per lo sviluppo mobile:
  - **Swift** per iOS
  - **Kotlin** per android

### 6.5 Aspetti positivi

- Le tecnologie impiegate sono moderne e molto diffuse;
- Lo sviluppo mobile è attualmente molto rilevante.

## **6.6 Criticità e fattori di rischio**

- Scelta obbligata fra android e iOS;
- Poca dimestichezza del team con lo sviluppo mobile;
- Alcuni aspetti dello sviluppo del gioco, come parti grafiche, regole e attori potrebbero essere necessari per svolgere il testing ma dispendiosi da implementare e poco rilevanti agli scopi del progetto.

## **6.7 Conclusioni**

Il capitolato è stato scartato perché la parte mobile e dello sviluppo del gioco in se è parsa poco stimolante e potenzialmente insidiosa. Inoltre per quanto riguarda le tecnologia AWS altri capitolati ne proponevano un uso più approfondito e interessante.

## **7 Capitolato C7 - soluzioni di sincronizzazione desktop**