

# Studio di Fattibilit

Dream Corp.

05-12-2018



G&B

## Informazioni sul documento

Versione	Versione 0.5.3
Responsabile	Davide Ghiotto
Redattori	Pietro Casotto Michele Clerici
Verificatori	Marco Davanzo Davide Liu
Uso	Interno
Destinatari	Dream Corp. Zucchetti SpA Prof. Tullio Vardanega Prof. Riccardo Cardin



---

Versione	Data	Descrizione	Autore	Ruolo
1.0.0	2018/12/23	Approvazione documento	Davide Ghiotto	Responsabile
0.3.1	2018/11/23	Correzione documento	Marco Davanzo	Verificatore
0.3.0	2018/11/21	C2-C4-C6	Michele Clerici	Analista
0.2.0	2018/11/20	Introduzione e C1-C3-C5	Pietro Casotto	Analista
0.1.0	2018/11/19	Creazione struttura del documento	Michele Clerici	Analista



# 1 Introduzione

## 1.1 Obiettivo del documento

Lo scopo del documento è di motivare la scelta del capitolato<sub>G</sub>C3 "G&B", e di presentare le considerazioni che ci hanno portato a scartare gli altri.

## 1.2 Glossario

Allo scopo di evitare ambiguità a lettori esterni al gruppo, si specifica che all'interno del documento verranno inseriti dei termini con un carattere G come pedice, questo significa che il significato inteso in quella situazione è stato inserito nel Glossario;

## 1.3 Riferimenti

### 1.3.1 Normativi

1. *Norme di progetto*

### 1.3.2 Informativi

1. **Capitolato 1 : Butterfly** - monitor per processi CI/CD;  
<https://www.math.unipd.it/~tullio/IS-1/2018/Progetto/C1.pdf>
2. **Capitolato 2 : Colletta** - piattaforma raccolta dati di analisi di test;  
<https://www.math.unipd.it/~tullio/IS-1/2018/Progetto/C2.pdf>
3. **Capitolato 3 : G&B** - monitoraggio intelligente di processi DevOps<sub>G</sub>;  
<https://www.math.unipd.it/~tullio/IS-1/2018/Progetto/C3.pdf>
4. **Capitolato 4 : MegAlexa** - Arrichitore di skill di Amazon Alexa;  
<https://www.math.unipd.it/~tullio/IS-1/2018/Progetto/C4.pdf>
5. **Capitolato 5 : P2PCS** - Piattaforma di peer-to-peer car sharing;  
<https://www.math.unipd.it/~tullio/IS-1/2018/Progetto/C5.pdf>
6. **Capitolato 6 : Soldino** - Piattaforma Ethereum per pagamenti IVA;  
<https://www.math.unipd.it/~tullio/IS-1/2018/Progetto/C6.pdf>



## 2 Capitolato scelto C3

### 2.1 Descrizione

Il progetto G&B consiste nel sviluppare un plug-in<sub>G</sub> di Grafana<sub>G</sub>, un software per il monitoraggio di sistemi, al fine di non solo monitorare i livelli di risorse utilizzate dai suddetti sistemi, ma anche di dare direttive sugli interventi da eseguire.

### 2.2 Finalità

Gli obiettivi principali del progetto sono: implementare un plug-in di Grafana che legge una rete Bayesiana e che a intervalli regolari o in continuità applica il ricalcolo delle probabilità della rete e fornisce i nuovi dati alla dashboard<sub>G</sub> attraverso grafi.

### 2.3 Tecnologie utilizzate

- **Javascript<sub>G</sub>** : linguaggio di scripting orientato agli oggetti ed agli eventi;
- **Reti Bayesiane<sub>G</sub>** : modello grafico probabilistico che rappresenta un insieme di variabili stocastiche con le loro dipendenze condizionali (Consigliata la libreria jsbayes : <https://github.com/vangj/jsbayes>);
- **Grafana** : software open source per il monitoraggio di sistemi <https://grafana.com/>.

### 2.4 Conclusioni

L'argomento è stato trovato molto interessante, soprattutto per quanto riguarda l'uso delle reti baiesiane. Questo progetto ci permette di interfacciarci con nuove funzionalità e tecnologie a noi sconosciute e che potremmo utilizzare nel corso della nostra carriera. Il progetto si presenta molto impegnativo in termini di studio personale per quanto riguarda le funzionalità di Grafana. Inoltre la disponibilità dell'azienda ha fatto preferire questo capitolato agli altri.



## 3 Capiolalo C1

### 3.1 Descrizione

Il progetto Butterfly prevede lo sviluppo di applicativi per la raccolta di messaggi provenienti da diversi servizi, il raggruppamento di essi all'interno di topic, ed infine l'invio ai sistemi di messaggistica degli utenti desiderati.

### 3.2 Finalità

Butterfly si articola in quattro diversi tipi di applicativi :

- **Producers** : componenti che hanno il compito di recuperare le segnalazioni e pubblicarle sotto forma di messaggi all'interno dei topic adeguati, per il progetto viene richiesto lo sviluppo di almeno due componenti;
- **Broker** : strumento per gestire i topic, implementato attraverso un DockerFile;
- **Consumers** : componenti che si abboneranno ai topic adeguati e invieranno i messaggi ai destinatari finali, per il progetto vengono richiesti almeno due di questi componenti;
- **Gestore Personale** : un componente custom specifico che attraverso il sistema di Consumer/Producer reindirizza i messaggi alla persona più appropriata secondo i canali di comunicazione desiderati.

### 3.3 Tecnologie utilizzate

- **Linguaggi consigliati**
  - **Java** : linguaggio di programmazione ad alto livello orientato agli oggetti, a tipizzazione statica;
  - **Python** : linguaggio ad alto livello, orientato agli oggetti, adatto a scripting, applicazioni distribuite e system testing;
  - **Node.js** : piattaforma Open source event-driven per l'esecuzione di codice JavaScript Server-side.
- **Apache Kafka** : piattaforma open source di stream processing;
- **Docker** : progetto open-source che automatizza il deployment di applicazioni all'interno di contenitori software.



### 3.4 Conclusioni

Questo progetto è stimolante perchè ci introduce all'apprendimento di tecnologie non ancora viste che sono molto utilizzate in ambito lavorativo che riteniamo fondamentali da avere in un curriculum. Non conoscendo la complessità degli argomenti da studiare abbiamo scartato questo capitolato.



## 4 Capitolato C2

### 4.1 Descrizione

Il progetto Colletta prevede lo sviluppo di un servizio web o un'applicazione mobile per la raccolta di dati indiretta. Questi verranno utilizzati per istruire un'intelligenza artificiale.

### 4.2 Finalità

L'obiettivo del progetto è quello di predisporre degli esercizi di grammatica (ad esempio analisi grammaticale) che potranno essere svolti da utenti e docenti registrati al servizio. In seguito allo svolgimento degli esercizi, i dati raccolti attraverso quest'ultimi saranno utilizzati per istruire un'intelligenza artificiale tramite tecniche di apprendimento automatico supervisionato, in modo che essa possa eseguirli.

### 4.3 Tecnologie utilizzate

Viene dato ampio margine alla scelta delle tecnologie, anche se ne vengono suggerite alcune:

- **Firebase** : piattaforma di sviluppo mobile e web, consigliato per la raccolta dati;
- **Hunpos** : software opensource per il pos-tagging;
- **FreeLing** : altro software per il pos-tagging.

### 4.4 Conclusioni

L'ampia libertà di scelta delle tecnologie è stata molto apprezzata. Anche il dover interfacciarsi con l'intelligenza artificiale risulta molto interessante, ma la richiesta in sé non lo è. L'assenza di profondità del capitolato ci ha portati a scartarlo.



## 5 Capitolato C4

### 5.1 Descrizione

Il progetto MegAlexa prevede di sviluppare un sistema Workflow(shortcut) per la creazione di skill custom.

### 5.2 Finalità

L'obiettivo del progetto è quello di creare una skill per Amazon Alexa in grado di avviare workflow creati dagli utenti attraverso un'interfaccia web o mobile. L'utente, all'interno della piattaforma, potrà inserire delle microfunzioni all'interno del workflow, eseguite tramite controllo vocale.

### 5.3 Tecnologie

- **Amazon Web Services**
  - **API Gateway** : servizio che semplifica le attività svolte su API su qualsiasi scala;
  - **Lambda** : permette esecuzione di codice senza badare al server;
  - **DynamoDB** : database non relazionale che fornisce prestazioni affidabili su qualsiasi scala.
- **Node.js** : piattaforma Open source event-driven per l'esecuzione di codice JavaScript;
- **HTML, CSS3, Javascript** : linguaggi per il servizio web;
- **Bootstrap** : framework responsive;
- **Swift o Kotlin** : linguaggi di programmazione utilizzate per sviluppare rispettivamente applicazioni per iOS e Android.

### 5.4 Conclusioni

Il progetto utilizza tecnologie attuali ed interessanti. Pensiamo che possano essere anche molto utili per il futuro. Tuttavia non è stato motivo di ispirazione per il gruppo.





## 6 Capitolato: C5

### 6.1 Descrizione

Il capitolato "P2PCS" prevede l'arricchimento delle funzionalità dell'applicazione che gestisce un car sharing condominiale.

### 6.2 Finalità

La finalità del progetto è quella di arricchire il supporto nella piattaforma android, creare un servizio specifico di car sharing peer to peer e sfruttare almeno 5 dei core drive del framework di Octalysis.

### 6.3 Tecnologie utilizzate

- **NodeJS** : linguaggi di scripting comunemente utilizzati per la creazione di contenuti web;
- **GoogleCloud** : registro digitale aperto e distribuito, in grado di memorizzare record di dati (solitamente, denominati "transazioni") in modo sicuro, verificabile e permanente.

### 6.4 Conclusioni

Il progetto è troppo vago e non si capisce esattamente cosa ci è richiesto. Nemmeno l'esposizione delle tecnologie da utilizzare sono molto chiare. Non condividiamo l'idea di base dell'applicazione.



## 7 Capitolato: C6

### 7.1 Descrizione

Il capitolato "Soldino" prevede la realizzazione di una piattaforma web per la gestione automatizzata delle VAT.

### 7.2 Finalità

L'obiettivo è la creazione di una serie di DApps che lavorano sulla EVM per la gestione automatica della VAT. Viene richiesto lo sviluppo di un'applicazione web o di un programma ad interfaccia grafica composta da una o più sezioni per le funzionalità del Governo, delle imprese, e dei cittadini. Inoltre vengono richieste delle sezioni dove le varie entità possono eseguire transazioni tra di loro.

### 7.3 Tecnologie utilizzate

- **HTML, CSS, Javascript** : linguaggi di programmazione comunemente utilizzati per la creazione di contenuti web;
- **Blockchain** : registro digitale aperto e distribuito, in grado di memorizzare record di dati (solitamente, denominati "transazioni") in modo sicuro, verificabile e permanente;
- **Ethereum** : piattaforma creata per permettere agli utenti di creare in modo semplice applicazioni decentralizzate (applicazioni distribuite dove ogni parte è in grado di svolgere il proprio compito senza dipendere dalle altre) che usano la tecnologia blockchain;
- **Truffle** : ambiente di sviluppo, framework di test e pipeline di asset per blockchain che utilizzano Ethereum Virtual Machine (EVM);
- **React/Redux** : framework che aiuta a scrivere applicazioni Javascript che si comportino in modo consistente, funzionino in diversi ambienti e siano facili da testare.

### 7.4 Conclusioni

La tecnologia Blockchain è risultata a tutti interessante e innovativa, in quanto costituisce il futuro delle transazioni, ma prevede uno studio dedicato a cui nessun membro del team è particolarmente interessato. Seguendo il trend degli altri capitolati, non andremo ad usare



direttamente la tecnologia in questione ma dovremo solo capire come "interfacciarsi" con quest'ultima, che per quanto interessante non è fonte di ispirazione per il nostro lavoro.