



INGEGNERIA DEL SOFTWARE
a.a. 2018/2019

Capitolato C4 - MegAlexa

Studio di Fattibilità

Componenti:

Sonia MENON
Alberto MIOLA
Andrea PAVIN
Alessandro PEGORARO
Matteo PELLANDA
Pardeep SINGH
Luca STOCCO

Destinatari:

Prof. Tullio VARDANEGA
Prof. Riccardo CARDIN

Informazioni sul documento

<i>Responsabile</i>	Sonia MENON
<i>Verifica</i>	Alberto MIOLA, Luca STOCCO
<i>Redazione</i>	Alessandro PEGORARO, Andrea PAVIN Pardeep SINGH, Matteo PELLANDA
<i>Uso</i>	Interno
<i>Stato</i>	Approvato
<i>Email</i>	duckware.swe@gmail.com
<i>Riferimento</i>	Capitolato C4 - MegAlexa

Descrizione

Il presente documento contiene l'analisi dei capitolati d'appalto, per valutare la fattibilità e i punti critici dei progetti software in oggetto.

Versione 1.0.0 del
6 Dicembre 2018

Indice

Registro delle Modifiche

1	Introduzione	1
1.1	Scopo del documento	1
1.2	Scopo del prodotto	1
1.3	Glossario	1
1.4	Riferimenti	1
1.4.1	Riferimenti normativi	1
1.4.2	Riferimenti informativi	1
2	Capitolato C1 - Butterfly	3
2.1	Descrizione	3
2.2	Dominio Applicativo	3
2.3	Dominio Tecnologico	4
2.4	Valutazione Finale	4
3	Capitolato C2 - Miviq	5
3.1	Descrizione	5
3.2	Dominio Applicativo	5
3.3	Dominio Tecnologico	5
3.4	Valutazione Finale	6
4	Capitolato C3 - Grafana & Bayes	7
4.1	Descrizione	7
4.2	Dominio Applicativo	7
4.3	Dominio Tecnologico	7
4.4	Valutazione Finale	7
5	Capitolato C4 - MegAlexa	9
5.1	Descrizione	9
5.2	Dominio Applicativo	9
5.3	Dominio Tecnologico	9
5.4	Valutazione Finale	9
6	Capitolato C5 - GaiaGo	10
6.1	Descrizione	10

6.2	Dominio Applicativo	10
6.3	Dominio Tecnologico	10
6.4	Valutazione Finale	10
7	Capitolato C6 - Soldino	12
7.1	Descrizione	12
7.2	Dominio Applicativo	12
7.3	Dominio Tecnologico	12
7.4	Valutazione Finale	13

Registro delle Modifiche

Ver.	Data	Autore	Ruolo	Descrizione
1.0.0	2018-12-06	Sonia MENON	Responsabile	Verifica documento
0.2.0	2019-01-06	Luca STOCCO	Verificatore	Aggiornamento documento ed Adattamento al nuovo template
0.1.1	2018-12-05	Alberto MIOLA	Verificatore	Verifica capitoli C1, C2 e C3
0.1.0	2018-12-04	Luca STOCCO	Verificatore	Verifica capitoli C4, C5 e C6
0.0.7	2018-12-03	Alberto MIOLA	Analista	Analisi capitolo C4
0.0.6	2018-12-02	Alessandro PEGORARC	Analista	Analisi capitolo C5
0.0.5	2018-12-02	Matteo PELLANDA	Analista	Analisi capitolo C2
0.0.4	2018-12-01	Andrea PAVIN	Analista	Analisi capitolo C6
0.0.3	2018-12-01	Pardeep SINGH	Analista	Analisi capitolo C1
0.0.2	2018-11-30	Luca STOCCO	Analista	Analisi capitolo C3
0.0.1	2018-11-30	Luca STOCCO	Programmatore	Creazione del documento

Tabella 1: Registro delle Modifiche

1 Introduzione

1.1 Scopo del documento

Lo scopo del documento è quello di riportare lo studio fatto dal gruppo riguardante i vari capitoli, mostrando nel dettaglio i motivi che lo hanno portato a scegliere il progetto C4 e a scartare gli altri.

1.2 Scopo del prodotto

Lo scopo del prodotto è di creare una $skill_G$ utilizzabile con l'assistente $Alexa_G$ che consista nell'avviare una routine personalizzata. Un utente potrà quindi creare un suo $workflow_G$ collegando micro-funzioni riconosciute singolarmente da $Alexa_G$, per poterle utilizzare in sequenza con un solo comando da lui scelto.

1.3 Glossario

Nel redigere il documento sono stati usati termini ed acronimi che possono risultare ambigui o poco chiari. Tutte queste parole saranno scritte in corsivo e marcate con una G pedice, anch'essa in corsivo, per indicare che il loro significato è disambiguato nel documento *Glossario*.

1.4 Riferimenti

1.4.1 Riferimenti normativi

- *Norme di progetto v 1.0.0.*

1.4.2 Riferimenti informativi

- **Capitolato scelto - C4**
MegAlexa: arricchitore di skill di Amazon_G Alexa_G
<https://www.math.unipd.it/tullio/IS-1/2018/Progetto/C4.pdf>
- **Capitolato C1**
Butterfly: monitor per processi_G CI/CD
<https://www.math.unipd.it/tullio/IS-1/2018/Progetto/C1.pdf>
- **Capitolato C2**
Colletta: piattaforma raccolta dati di analisi di testo
<https://www.math.unipd.it/tullio/IS-1/2018/Progetto/C2.pdf>
- **Capitolato C3**
G & B: monitoraggio intelligente di processi_G DevOps
<https://www.math.unipd.it/tullio/IS-1/2018/Progetto/C3.pdf>

- **Capitolato C5**

P2PCS: piattaforma di peer-to-peer car sharing

<https://www.math.unipd.it/tullio/IS-1/2018/Progetto/C5.pdf>

- **Capitolato C6**

Soldino: piattaforma Ethereum per pagamenti IVA

<https://www.math.unipd.it/tullio/IS-1/2018/Progetto/C6.pdf>

2 Capitolato C1 - Butterfly

2.1 Descrizione

L'obiettivo del progetto Butterfly è di semplificare i *processi_G* di Continuous Integration e Continuous Delivery che vengono attuati nelle realtà aziendali. Questi *processi_G* coinvolgono vari strumenti *software_G* tra i quali troviamo:

Git_G: Per il controllo di versione distribuito (*DVCS_G*)

SonarQube: piattaforma che consente di capire la qualità del codice attraverso la sua analisi statica

Redmine: Permette di gestire uno o più progetti mantenendo sotto controllo le varie attività che ne fanno parte

Jenkins: Fornisce una serie di servizi che permettono la Continuous Integration durante lo sviluppo del *software_G*

Questi strumenti forniscono una propria interfaccia per la segnalazione dei messaggi verso l'esterno. Questa discrepanza di meccanismi di interazione con gli strumenti complica di molto i *processi_G* di Continuous Integration/Delivery. Infatti per conoscere lo stato di un particolare strumento bisogna accedere ad una specifica dashboard; inoltre l'interfaccia offerta da questi strumenti potrebbe anche soffrire delle limitazioni di visibilità di rete, con la conseguenza di non essere raggiungibili in tutte le situazioni.

2.2 Dominio Applicativo

Creare una piattaforma *software_G* che accentri le segnalazioni provenienti dai vari strumenti *software_G*. Questa piattaforma *software_G* deve seguire il *design pattern_G* Publisher/Subscriber che consenta la comunicazione asincrona fra i vari strumenti. Si richiede, quindi, di sviluppare una serie di componenti che si interfaccino con i vari strumenti, e provvedano a riportare le segnalazioni intercettate all'utente finale. La piattaforma sarà composta da quattro tipologie di componenti:

Producers: Sono i componenti con il compito di recuperare le segnalazioni degli strumenti, e pubblicare queste ultime all'interno di adeguati Topic. Queste segnalazioni devono essere pubblicate sotto forma di messaggi. E' richiesta la creazione di un componente per ciascun strumento.

Broker: Strumento che permetta l'istanziamento e la gestione dei vari Topic.

Consumers: Questi componenti sono quelli che si interfacciano con l'utente finale.

Quindi i componenti consumer dovranno recuperare i messaggi dai Topic adeguati e procedere al loro invio verso l'utente finale. L'utente finale può vedere i messaggi recuperati dai Topic nei seguenti modi:

- Telegram

2 Capitolato C1 - Butterfly

- $Slack_G$
- Email

Perciò è richiesta la creazione di componenti consumer che permettano, rispettivamente, di recapitare i messaggi su Telegram, $Slack_G$ o invio di Email. Viene inoltre chiesto di creare un applicativo Gestore Personale che recuperi i messaggi da un Topic specifico, e in base alla tipologia del messaggio (deducibile da alcuni metadati) lo inoltri verso l'utente più appropriato.

2.3 Dominio Tecnologico

Il capitolato non mette vincoli rigidi sull'utilizzo delle tecnologie, tuttavia viene consigliato:

- L'uso di un linguaggio di programmazione tra $Java_G$ (versione 8 o superiore), Python o NodeJs per lo sviluppo dei componenti applicativi
- L'uso del $software_G$ Apache Kafka per la realizzazione del componente Broker.

Invece tra le richieste difficilmente contrattabili troviamo:

- il $software_G$ sviluppato deve rispettare i 12 fattori esposti dal documento ["The Twelve-Factor App"](#)
- uso di Docker per la creazione di vari container
- i vari componenti devono mettere a disposizione delle API_G Rest
- tutte le componenti applicative devono essere corredate da test unitari e d'integrazione

2.4 Valutazione Finale

Il gruppo ha mostrato discreto interesse per questo capitolato in quanto risulta ben strutturato e dettagliato (riguardo anche le aspettative minime del prodotto $software_G$ finale). Tuttavia si è deciso di mantenere questo capitolato come terza scelta del gruppo per i seguenti motivi:

- le tecnologie presentate hanno generato uno scarso stimolo positivo da parte dei componenti del gruppo.
- è richiesto uno sforzo abbastanza grande per quanto riguarda lo studio dell'interfaciamento con i singoli componenti.

3 Capitolato C2 - Miviq

3.1 Descrizione

Lo scopo del capitolato C2, è quello di sviluppare una piattaforma web di tipo collaborativo di raccolta dati, in cui gli utenti possano predisporre e/o svolgere esercizi di grammatica. Grazie a questa piattaforma i dati raccolti verranno utilizzati da sviluppatori al fine di insegnare ad un elaboratore a svolgere esercizi mediante tecniche di apprendimento automatico.

3.2 Dominio Applicativo

L'obiettivo finale del capitolato è quello di realizzare una piattaforma che raccolga dati da tre diverse tipologie di utenti: insegnanti, allievi, sviluppatori. Per gli insegnanti la piattaforma deve poter:

- Predisporre di esercizi di analisi grammaticale
- Inserire nuove frasi nel sistema della piattaforma e poterle etichettare in modo automatico con informazioni relative all'analisi
- Correggere gli svolgimenti

Al fine di agevolare il lavoro di preparazione viene consigliato l'utilizzo di *software_G* di terze parti. Poiché tali *software_G* sono soggetti ad errori l'insegnante dovrà poter correggere i risultati prodotti da esso. Per gli allievi la piattaforma deve poter:

- Svolgere gli esercizi proposti e/o predisposti e ricevere una valutazione immediata

Per gli sviluppatori la piattaforma deve poter:

- Prelevare ed accedere ai dati che sono raccolti nella piattaforma e poter visionare più versioni di analisi di ogni frase svolta al fine di dedurre quale sia l'annotazione più corretta
- Conoscere lo storico dei dati
- Accedere ai modelli realizzati

3.3 Dominio Tecnologico

MaryTTS e FreeLing: Sono piattaforme di sintesi vocale multilingua e open-source. Questi *software_G* implementano il part-of-speech tagger, cioè etichettare le parti del discorso con tag riferiti alle classi grammaticali.

Firebase: è una piattaforma di sviluppo di applicazioni web e mobili sviluppata da Google nel 2014. Offre funzionalità che ruotano attorno ai servizi cloud, consentendo agli utenti di salvare e recuperare i dati ai quali potranno accedere da qualsiasi dispositivo o browser. Nel caso in questione il suo utilizzo è quello di analizzare e immagazzinare al meglio i dati estratti dalla piattaforma richiesta.

3 Capitolato C2 - Miviq

Per l'obiettivo richiesto il gruppo 3 preventiva l'utilizzo di tecnologie web come:

Html e CSS: Sono linguaggi che hanno applicazione in pagine/siti web, nati per la formattazione e impaginazione di documenti ipertestuali. Tramite questi linguaggi sarà possibile realizzare un'interfaccia, la più smart e friendly possibile, fra l'utente e la piattaforma.

Javascript: E' un linguaggio di scripting orientato agli oggetti e agli eventi, comunemente utilizzato nella programmazione lato *client*_G per la creazione di applicazioni e piattaforme web. Il suo ambito in questo capitolato sarà quello di interfacciarsi con le piattaforme comunicateci (MaryTTS o FreeLing e Firebase) per poter implementare le features richieste.

3.4 Valutazione Finale

Il gruppo Duckware ha espresso pareri positivi riguardo la richiesta di utilizzo della piattaforma Firebase, tuttavia il capitolato presentato non ha richiamato l'attenzione di tutti i partecipanti del nostro gruppo, i quali non lo hanno trovato appetibile rispetto alle tendenze tecnologiche odierne e poco stimolante nell'auto-apprendimento. La presentazione è stata interessante anche se a primo impatto risultava vaga sulla richiesta del committente, solo dopo un'ulteriore analisi infatti si è inquadrato a pieno l'obiettivo richiesto. Infine dopo aver analizzato tutti i capitoli disponibili e le specifiche per essi richieste, il team di Duckware ha deciso di non considerare questo progetto in quanto non presenta tecnologie di interesse, che siano sufficientemente innovative e forniscano nuove capacità e conoscenze.

4 Capitolato C3 - Grafana & Bayes

4.1 Descrizione

Il capitolato C3 si pone l'obiettivo di sviluppare un plug-in per la piattaforma Grafana, che gestisca e monitori il flusso di dati proveniente dal reparto DevOps, tramite l'implementazione di una o più reti Bayesiane. L'applicativo, attraverso il modello probabilistico, dovrebbe essere in grado di evidenziare le criticità nei *processi_G* di DevOps, aiutando quindi le *software_G houses* a capire dove intervenire per migliorare l'erogazione del servizio.

4.2 Dominio Applicativo

Costruire un plug-in di Grafana che applichi la definizione di una o più reti bayesiane al flusso di dati provenienti dal campo, per identificare i problemi che gli operatori di erogazione possono segnalare alla fabbrica del *software_G* al fine di migliorare la qualità del servizio.

4.3 Dominio Tecnologico

Grafana: Piattaforma open source di analisi e monitoraggio di flussi di dati in real-time. Fortemente modulare, consente agli sviluppatori di creare plug-in personalizzati sfruttando la duttilità di javascript. Grafana archivia i dati in InfluxDB, un database specializzato per serie temporali.

Link: <https://grafana.com/>

Javascript: Linguaggio di scripting orientato agli oggetti e agli eventi, comunemente utilizzato nella programmazione web lato *client_G* per la creazione, in siti e applicazioni web, di effetti dinamici interattivi tramite funzioni di script invocate da eventi. Ultimamente il suo campo di utilizzo è stato esteso alle cosiddette Hybrid App (app ibride), con le quali è possibile creare app per più sistemi operativi utilizzando un unico codice sorgente.

Link: <https://www.javascript.com/>

Reti Bayesiane: Modello grafico probabilistico che rappresenta un insieme di variabili stocastiche con le loro dipendenze condizionali attraverso l'uso di un grafo aciclico, in cui i nodi rappresentano le variabili, mentre gli archi rappresentano le relazioni di dipendenza statistica tra le variabili e le distribuzioni locali di probabilità dei nodi figlio rispetto ai valori dei nodi padre.

Libreria utilizzata: <https://github.com/vangj/jsbayes>

4.4 Valutazione Finale

Questo capitolato è stato accolto discretamente bene dal gruppo, forse per merito dell'ottima presentazione e dall'ambizione dei singoli di interfacciarsi con una grossa azienda del settore, tanto da essersi guadagnato il terzo posto tra le preferenze generali. Tuttavia, in seguito ad un'attenta analisi dei progetti disponibili e delle specifiche richieste, il team

4 Capitolato C3 - Grafana & Bayes

di Duckware ha deciso di scartare questo progetto in quanto ritenuto troppo circoscritto, oltre a fornire un apporto cognitivo inferiore rispetto alle altre proposte.

5 Capitolato C4 - MegAlexa

5.1 Descrizione

Il capitolato C4 mira a creare un'applicazione per dispositivi mobile o un'interfaccia web che sia in grado di creare delle routine personalizzabili dagli utenti. Queste ultime dovranno essere gestibili utilizzando *Alexa_G*, l'assistente personale intelligente creato da *Amazon_G*. Ogni utente potrà quindi creare una propria sequenza di azioni preferite scegliendo fra quelle già disponibili ed interagendo con *Alexa_G*.

5.2 Dominio Applicativo

Creare una skill di *Alexa_G* che, tramite sito web o applicazione mobile, possa aiutare l'utente a creare un *workflow_G* nel quale vengono specificate una serie di azioni da compiere. Sarà disponibile una serie di *connettori_G*, ovvero delle azioni predefinite che l'utente sarà in grado di inserire nel suo *workflow_G*.

5.3 Dominio Tecnologico

Java_G: Linguaggio di programmazione ad alto livello, orientato agli oggetti e a tipizzazione statica, specificamente progettato per essere il più possibile indipendente dalla piattaforma di esecuzione.

Link: <https://docs.oracle.com/javase/8/>

Bootstrap: Framework per HTML, CSS, Javascript per creare siti web responsivi e mobile-first

Link: <https://www.javascript.com/>

PHP: Linguaggio di scripting che consente la creazione di pagine web dinamiche ed interazioni con i database

Amazon Web Services_G: Servizio di computazione su cloud fornito da *Amazon_G* che propone diverse soluzioni fra le quali: *AWS_G Lambda* per l'esecuzione in cloud di servizi, *AWS_G API Gateway* per la creazione di *API_G* (potenzialmente per infrastrutture REST) e *AWS_G DynamoDB* per il supporto ai database non relazionali

Link: <https://aws.amazon.com/it/lambda/>

Link: <https://aws.amazon.com/it/api-gateway/>

Link: <https://aws.amazon.com/it/dynamodb/>

5.4 Valutazione Finale

Questo capitolato è sembrato molto stimolante al gruppo in quanto propone una vasta gamma di tecnologie da utilizzare e consente di approcciarsi ai dispositivi personali intelligenti più recenti. Si è ritenuto fosse particolarmente interessante poter gestire da vicino il dispositivo utilizzando prima un'app sul proprio telefono e poi inviando comandi vocali. Un fattore importante nella scelta di questo capitolato è stato l'uso dei servizi di *Amazon_G* poichè *AWS_G* è uno dei leader mondiali del *cloud computing_G* e si sta affermando sempre di più sul mercato.

6 Capitolato C5 - GaiaGo

6.1 Descrizione

Il capitolato C5 ha lo scopo di creare un'applicazione di Car Sharing per dispositivi *Android_G*. L'app deve permettere agli utenti di definire un calendario per segnare i giorni in cui è disposto a prestare il proprio veicolo. Quest'ultimo verrà munito del dispositivo GaiaBox con cui il proprietario potrà geolocalizzarlo e controllare i percorsi effettuati dalla macchina. Ogni singolo utente potrà quindi chiedere in prestito tutte le macchine registrate nelle sue vicinanze e solo nei giorni concordati.

6.2 Dominio Applicativo

Creare un supporto applicativo che possa permettere all'utente di creare un calendario nel quale vengono specificati i giorni in cui pone a disposizione il proprio veicolo. Creare un servizio specifico Peer To Peer che organizzi il calendario e calcoli l'effettiva disponibilità di ogni macchina per ogni utente. Implementare nell'applicativo almeno 5 dei core drive del *framework_G* di Octalysis per incentivare gli utenti ad usare GaiaGo.

6.3 Dominio Tecnologico

Non vengono imposti vincoli nella scelta delle tecnologie per implementare il servizio Peer To Peer e la creazione dell'applicazione *Android_G*, mentre per la Gamification viene richiesto l'utilizzo del *framework_G* Octalysis.

Javascript: Linguaggio di scripting orientato agli oggetti e agli eventi, comunemente utilizzato nella programmazione web lato *client_G* per la creazione di applicazioni. Il suo ambito in questo capitolato sarà quello di interfacciarsi tra l'applicazione e il *server_G* contenente il calendario.

Link: <https://www.javascript.com/>

PHP: Linguaggio di scripting utilizzato per le interazioni con il database.

Link: <http://www.php.net/>

Framework Octalysis: Servizio di Gamification ideato da Yu-kai Chou; la Gamification viene intesa come spinta motivazionale e agonistica che mantiene l'utente fedele ad un dato servizio. Il *framework_G* si divide in otto core principali e ci viene chiesto di sviluppare il progetto implementandone almeno 5.

Link: <https://yukaichou.com/gamification-examples/octalysis-complete-gamification-framework/>

Link: <https://octalysisgroup.com/>

Link: <https://www.yukaichou.com/octalysis-tool/>

6.4 Valutazione Finale

Questo capitolato inizialmente si è presentato come il più appetibile per il gruppo date le conoscenze personali già affermate per la maggior parte dei componenti di Duckware,

ma un'analisi più attenta ha messo in luce una limitata possibilità di aumentare il proprio bagaglio conoscitivo e di competenze rispetto ad altri progetti. Il capitolato non è più stato la scelta predominante del gruppo ma ha guadagnato il secondo posto come progetto da poter sviluppare.

7 Capitolato C6 - Soldino

7.1 Descrizione

Il capitolato C6, Soldino, richiede di creare un sistema di gestione dell'imposta sul valore aggiunto (IVA), su piattaforma decentralizzata Ethereum. Gli utenti sono composti da:

- Produttori
- Grossisti
- Dettaglianti
- Consumatori finali.

Le transazioni tra questi saranno eseguite attraverso Ethereum e tramite lo stesso verrà versato allo stato l'incasso netto dell'IVA accumulato nelle transazioni tra le prime tre tipologie di utenti elencati.

7.2 Dominio Applicativo

Creare un sistema di gestione dell'IVA che attraverso un sito web permetta di automatizzare transazioni e versamenti appoggiandosi alla blockchain Ethereum.

7.3 Dominio Tecnologico

Smart contract: Procedure che facilitano, verificano, o fanno rispettare la negoziazione o l'esecuzione di un contratto. Sono definiti dal loro creatore, ma la loro esecuzione e, di conseguenza, il servizio che offrono, sono dettati dal network Ethereum. Una volta messi in atto esistono finchè esiste l'intero network, scompaiono solo se programmati per autodistruggersi.

Link: <https://www.investopedia.com/terms/s/smart-contracts.asp>

Solidity: Linguaggio contract-oriented che consente l'implementazione di smart contracts su Ethereum Virtual Machine (EVM).

Link: <https://solidity.readthedocs.io/en/v0.4.25/>

Ropsten: Rete di test utilizzando gli stessi protocolli di Ethereum utile a testare dApp(Ethereum enabled distributed applications).

Link: <https://ethereum.stackexchange.com/questions/13534/what-is-actually-ropsten-what-is-a-new-network>

React: Libreria JavaScript per la creazione di interfacce utente riutilizzabili.

Link: <https://reactjs.org/>

Surge: Servizio che permette la messa in uso di web-app.

Link: <https://surge.sh/>

MetaMask: Estensione per browser che permette di allacciarsi a servizi presenti su blockchain Ethereum (Ethereum enabled distributed applications).

Link: <https://metamask.zendesk.com/hc/en-us>

Raiden: Blockchain complementare a Ethereum che permette pagamenti istantanei.

Link: <https://raiden.network/>

7.4 Valutazione Finale

Il capitolato propone tecnologie moderne e pone un obiettivo sfumato ma avveniristico. Lascia un ampio margine di libertà nell'implementazione: le direttive ci sono sembrate sfocate ed è difficile immaginare la forma del prodotto finito. La presentazione è stata vaga sulla richiesta del committente e non è stato ancora capito a fondo l'obiettivo richiesto. Il team di Duckware ha quindi espresso poca preferenza verso questo capitolato.