AlphaSix

Studio di Fattibilità

Informazioni sul documento

Nome Documento	StudioDiFattibilità_v1_0_0.pdf
Data di Creazione	22 novembre 2018
Data ultima modifica	07 Gennaio 2016
Stato	Approvato
Redazione	Laura Cameran Timoty Granziero Ciprian Voinea Samuele Gardin Nicola Carlesso Matteo Marchiori
Verifica	Laura Cameran Timoty Granziero
Approvazione	Ciprian Voinea
Uso	Interno
Distribuzione	AlphaSix
Destinato a	Prof. Tullio Vardanega, Prof. Riccardo Cardin, Imola Informatica
Email di riferimento	alpha.six.unipd@gmail.com

Descrizione

Questo documento analizza tutti i capitolati proposti definendo gli aspetti positivi e negativi di ognuno, motivando la scelta del capitolato C1, Butterfly.

Registro delle modifiche

Versione	Descrizione	Autore	Ruolo	Data
0.0.4	Aggiunta sezione per C4	Timoty Granziero	Redattore	27-11-2018
0.0.2	Struttura di base del documento	Laura Cameran	Redattore	23-11-2018
0.0.1	Creazione template	Timoty Granziero	Redattore	22-11-2018

Indice

1 Introduzione **1**
1.1 Scopo del documento . . . 1
1.2 Glossario . . . 1
1.3 Riferimenti . . . 1
1.3.1 Riferimenti normativi . . . 1
1.3.2 Riferimenti informativi . . . 1

2 Butterfly - C1 **2**
2.1 Descrizione generale . . . 2
2.2 Obiettivo finale . . . 2
2.3 Tecnologie coinvolte . . . 2
2.4 Valutazione conclusiva . . . 3

3 Colletta - C2 **4**
3.1 Descrizione generale . . . 4
3.2 Obiettivo finale . . . 4
3.3 Tecnologie coinvolte . . . 4
3.4 Valutazione conclusiva . . . 4

4 G&B - C3 **5**
4.1 Descrizione generale . . . 5
4.2 Obiettivo finale . . . 5
4.3 Tecnologie coinvolte . . . 5
4.4 Valutazione conclusiva . . . 5

5 MegAlexa - C4 **6**
5.1 Descrizione generale . . . 6
5.2 Obiettivo finale . . . 6
5.3 Tecnologie coinvolte . . . 6
5.4 Valutazione conclusiva . . . 6

6 P2PCS - C5 **7**
6.1 Descrizione generale . . . 7
6.2 Obiettivo finale . . . 7
6.3 Tecnologie coinvolte . . . 7
6.4 Valutazione conclusiva . . . 7

7 Soldino - C6 **8**
7.1 Descrizione generale . . . 8
7.2 Obiettivo finale . . . 8
7.3 Tecnologie coinvolte . . . 8
7.4 Valutazione conclusiva . . . 8

1 Introduzione

1.1 Scopo del documento

Il documento corrente ha lo scopo di descrivere brevemente ma con adeguata precisione ognuno dei sei capitolati proposti, in modo da rendere ben visibili i motivi per cui è stato scelto il capitolato C1 *Butterfly*.

1.2 Glossario

Tutti i termini qui presenti che richiedono una spiegazione più dettagliata, per evitare ambiguità, sono riconoscibili dal pedice G (e.g. Alexa$_G$) e possono essere trovati, insieme alla loro definizione, in un documento allegato denominato *Glossario v1.0.0*.

1.3 Riferimenti

1.3.1 Riferimenti normativi

- **Norme di progetto**: *Norme di Progetto v1.0.0*.

1.3.2 Riferimenti informativi

- Capitolato d'appalto C1, **Butterfly**[1]: monitor per processi CI/CD;
- Capitolato d'appalto C2, **Colletta**[2]: piattaforma raccolta dati di analisi di testo;
- Capitolato d'appalto C3, **G&B**[3]: monitoraggio intelligente di processi DevOps;
- Capitolato d'appalto C4, **MegAlexa**[4]: arricchitore di skill di Amazon Alexa;
- Capitolato d'appalto C5, **P2PCS**[5]: piattaforma di peer-to-peer car sharing;
- Capitolato d'appalto C6, **Soldino**[6]: piattaforma Ethereum per pagamenti IVA.

[1] http://www.math.unipd.it/~tullio/IS-1/2018/Progetto/C1.pdf
[2] http://www.math.unipd.it/~tullio/IS-1/2018/Progetto/C2.pdf
[3] http://www.math.unipd.it/~tullio/IS-1/2018/Progetto/C3.pdf
[4] http://www.math.unipd.it/~tullio/IS-1/2018/Progetto/C4.pdf
[5] http://www.math.unipd.it/~tullio/IS-1/2018/Progetto/C5.pdf
[6] http://www.math.unipd.it/~tullio/IS-1/2018/Progetto/C6.pdf

2 Butterfly - C1

2.1 Descrizione generale

Il capitolato C1 prevede lo sviluppo di Butterfly, un applicativo di supporto alle figure coinvolte nella produzione di un prodotto software che utilizzano strumenti con interfacce individuali che non comunicano fra di loro. Questo progetto nasce dalla necessità di accentrare e standardizzare queste segnalazioni oltre a permetterne una gestione automatizzata e personalizzabile.

2.2 Obiettivo finale

Butterfly si presenta come un insieme di componenti che si interfacciano con gli strumenti di sviluppo in modo da recuperare o intercettare le segnalazioni che questi mandano e riportarle all'utente nella forma che quest'ultimo sceglie (telegram, mail, slack).
Quest'applicativo deve essere in grado di incanalare le notifiche di ciascuno strumento coinvolto nella pipeline di produzione in un singolo broker che gestisce l'inoltro verso un gestore interno all'azienda oppure alla persona interessata tramite applicazioni di messaggistica o email in base al topic con cui sono queste notifiche contrassegnate.

2.3 Tecnologie coinvolte

Si richiede l'utilizzo di un pattern Publisher / Subscriber per la gestione dei messaggi:

- Publisher: strumenti che generano i messaggi
  - Redmine: applicazione web di project managment
  - GitLab: servizio di hosting per progetti software che implementa lo strumento di controllo di sviluppo git
  - SonarQube: applicazione di Continuous Inspection che permette di ispezionare il codice ed analizzare il codice in maniera statica e dinamica da remoto
- Broker: interfaccia che raccoglie i messaggi e li incanala verso l'utente
  - Apache Kafka: piattaforma online per la gestione dei feed di dati in tempo reale
- Subscriber: mezzi su cui i messaggi arrivano all'utente
  - Telegram
  - Slack
  - E-mail

I linguaggi indicati nel capitolato in cui l'applicativo può essere sviluppato sono: Java, Python, NodeJS.
Sono inoltre richeste:

- API REST per interfacciarsi con i componenti dell'applicativo
- Dockerfile contenente le configurazioni necessarie per il container sul quale si andrà ad eseguire l'applicativo

2.4 Valutazione conclusiva

La scelta di sviluppare il progetto presentato in questo capitolato permette al team di lavorare con un ampio ventaglio di tecnologie richieste sul mercato come Docker e Apache Kafka e questo ha motivato il gruppo a prenderne parte.
Ha colpito molto la possibilità di creare un prodotto concreto e impiegato giornalmente dai developer che possa facilitarne il lavoro e automatizzare una parte del processo di sviluppo.
Ha inoltre attirato l'attenzione del gruppo la consegna chiara e i vincoli precisi posti dall'azienda.
D'altra parte l'impiego di numerose tecnologie richiede un impegno non indifferente e potrebbe risultare complesso.

3 Colletta - C2

3.1 Descrizione generale

Il progetto Colletta proposto dall'azienda Mivoq prevede la realizzazione di una piattaforma di collezione dati inerenti piccoli esercizi di grammatica. In essa si identificano tre attori principali: insegnanti, allievi e sviluppatori. L'insegnante fornisce gli esercizi di grammatica che vengono svolti dagli allievi. I dati derivanti da queste interazioni tra insegnati e allievi possono in seguito essere utilizzate dagli sviluppatori per migliorare il sistema di riconoscimento delle frasi.

3.2 Obiettivo finale

L'obiettivo del progetto è creare un'applicazione (web o mobile) con una struttura come quella descritta precedentemente, in cui la raccolta di dati avviene in modo implicito tramite il solo utilizzo da parte degli utenti. Questi dati possono essere impiegati successivamente per la produzione di servizi utili basati sull'apprendimento automatico.

3.3 Tecnologie coinvolte

La scelta delle tecnologie viene lasciata molto libera ma vengono consigliate:

- **Firebase** o altri servizi esistenti per l'immagazzinamento dei dati;
- Software open-source per lo svolgimento degli esercizi:
  - **Hunpos**
  - **Freeling**

3.4 Valutazione conclusiva

Il capitolato appena descritto non è stato scelto dal gruppo perchè tocca solo marginalmente l'ambito del machine learning. Tratta solamente di una piattaforma per la raccolta dati, motivo per cui non risulta particolarmente accattivante. La mancanza di vincoli non da un'idea precisa di come sarebbe possibile operare per sviluppare il progetto.

4 G&B - C3

4.1 Descrizione generale

Il capitolato C3 parte dall'esigenza di dover monitorare i sistemi che nel prossimo futuro di occuperanno della fatture emesse in formato digitale. Data la grande mole di dati emessi, tali sistemi devono essere soggetti ad un continuo controllo. Per fare ciò si ricerca una stretta collaborazione tra "Development" e "Operation", detta anche "DevOps", grazie ad un efficiente sistema di notifica di eventuali problemi nei sistemi utilizzati.

4.2 Obiettivo finale

L'azienda proponente, come strumento di monitoraggio per questo tipo di sistemi, si affida a Grafana. Purtroppo Grafana si limita a segnalare eventuali problematiche principalmente nel momento in cui i parametri osservati superano un valore soglia. L'azienda vorrebbe dunque creare un plug-in per Grafana che sfrutti l'intelligenza artificiale attraverso una rete Bayesiana affinché il monitoraggio sia più efficace.

4.3 Tecnologie coinvolte

L'azienda consiglia / richiede di utilizzare:

- **JavaScript**
- **Grafana**
- **Rete Bayesiana**: attraverso la libreria "jsbayes".
- **GitHub**

4.4 Valutazione conclusiva

Il capitolato richiede lo studio di strumenti inerenti l'intelligenza artificiale, un argomento ritenuto molto interessante dal gruppo. Purtroppo il dover creare un plug-in per un software già ben strutturato, quale Grafana, ha spinto il team a pensare che le tecnologie utilizzate fossero toccate in modo marginale.

5 MegAlexa - C4

5.1 Descrizione generale

Il capitolato$_G$ propone lo sviluppo di una piattaforma dedita alla creazione di una routine$_G$ in grado di eseguire una sequenza di skill$_G$ per Alexa$_G$, l'assistente vocale di Amazon.

5.2 Obiettivo finale

Nello specifico, è richiesto lo sviluppo di una piattaforma multilingua web e mobile (iOS o Android) che sia in grado di creare workflow$_G$ personalizzati per Alexa creati dagli utenti. Ogni utente deve aver la possibilità di poter nominare i propri workflow senza collidere con quelli di altri utenti (e.g. un workflow chiamato "Buongiorno" dell'utente A è diverso dal workflow "Buongiorno" dell'utente B).

5.3 Tecnologie coinvolte

L'azienda consiglia di utilizzare:

- Linguaggi per la piattaforma web:
  - **HTML5**$_G$, **CSS3**$_G$(**Bootstrap**$_G$), **Javscript**$_G$ per il frontend;
  - **Node.js**$_G$ per il backend;
- ***Kotlin***$_G$/***Swift***$_G$: linguaggi per lo sviluppo delle app rispettivamente per Android e iOS;
- ***Amazon Web Services***$_G$(AWS): servizio scelto per l'hosting del software e del database. Nello specifico, saranno utilizzati:
  - ***API Gateway***$_G$
  - ***Lambda***$_G$
  - ***Aurora Serverless***$_G$

5.4 Valutazione conclusiva

Dall'analisi fatta dal team è stato deciso di non sviluppare questo capitolato per l'elevata mole di lavoro prevista. Non è semplice inoltre gestire le varie routine per gli utenti oltre che sviluppare un'applicazione web e mobile nativa multilingua. Nonostante questo le tecnologie di AWS e di workflow di assistenti vocali risultano interessanti.

6 P2PCS - C5

6.1 Descrizione generale

Il quinto capitolato propone la creazione di un'applicazione *Android*$_G$ in grado di gestire una piattaforma di *car sharing*$_G$ *peer to peer*$_G$.

6.2 Obiettivo finale

È richiesto lo sviluppo di un sistema software che consenta ad un utente la condivisione della propria auto. Tale condivisione si potrà avere solo una volta che l'operatore avrà inserito nel calendario i giorni in cui non utilizzerà il mezzo. In questo modo si lascia così la possibilità di poter dare le chiavi della propria autovettura in mano ad un'altra figura.

6.3 Tecnologie coinvolte

L'azienda consiglia di utilizzare:

- ***Google Maps***$_G$
- ***Google Cloud Platform***$_G$
- ***Henshin - movens platform***$_G$
- ***Octalysis***$_G$
- ***Node.js***$_G$

6.4 Valutazione conclusiva

Fin da subito, l'idea di "condividere la propria macchina con altre persone amiche o meno" (cit da capitolato) non ha suscitato interesse e motivazione al gruppo. Inoltre la tecnologia *Octalysis*$_G$ sembra vincolare troppo l'andamento del progetto, richiedendone un uso stringente e diventa solamente un'applicazione che, secondo il parere del gruppo, sfrutta il *gamification*$_G$ per convincere gli utenti. Dopo queste considerazioni il team ha deciso di spostare i propri interessi verso altri capitolati.

7 Soldino - C6

7.1 Descrizione generale

Redbabel Studio propone lo sviluppo di un sistema che consenta l'amministrazione delle tassazioni (V.A.T.) imposte su beni venduti tramite ecommerce.

7.2 Obiettivo finale

L'obiettivo è quello di ottenere un prodotto che coinvolga l'intera gestione delle tasse, dal pagamento del cliente per il singolo bene fino alla ricezione e gestione da parte del governo, passando per il fornitore. L'applicativo è composto da una parte che si occupa di gestire gli smart contracts e da una parte Web che permetta l'accesso alla EVM (Ethereum Virtual Machine).

7.3 Tecnologie coinvolte

L'azienda consiglia:

- **Ethereum**: piattaforma che consente la scrittura facilitata di applicazioni distribuite che usano Blockchain.
- **Blockchain**: database distribuito che consente la tracciabilità di transazioni riguardanti cryptocurrency.
- **Smart Contracts**: contratti gestiti dalla EVM tramite codice.
- **EIP712**: standard per la firma dei contratti su Ethereum.
- **Gas**: costo dovuto alla computazione perché la transazione sia effettiva.
- **ERC20**: standard per i gettoni di Ethereum (beni digitali).
- **Reti Ethereum**: reti per lo scambio di cryptocurrency, ad esempio MainNet. Richiedono tempi lunghi per le transazioni.
- **Reti Raiden**: reti alternative, consentono tempi istantanei per il completamento delle transazioni.
- **Javascript ES8**: ultima versione di javascript, da usare con l'approccio promise centric.
- **React/Redux**: framework per lo sviluppo frontend di applicazioni basato su Javascript.
- **SCSS**: linguaggio compilato per sviluppo di css.

7.4 Valutazione conclusiva

Nonostante l'uso di tecnologie innovative come Blockchain, smart contracts e gestione delle tasse in modo automatico, il gruppo non ha scelto il progetto perché richiede molto tempo da dedicare allo studio approfondito di tutti gli argomenti coinvolti. Essendo queste tecnologie nuove il team era indeciso se scegliere questo capitolato o optare su strumenti più consolidati.