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

Registro delle modifiche

Versione	Descrizione	Autore e Ruolo	Data
0.0.2	Struttura di base del documento	Laura Cameran Analista	2018-23-11
0.0.1	Creazione template	Timoty Granziero Analista	2018-22-11

Indice

1 Introduzione

1.1 Scopo del documento

Riempire

1.2 Glossario

Riempire

1.3 Riferimenti

Riempire

1.3.1 Riferimenti normativi

Riempire

1.3.2 Riferimenti informativi

Riempire

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.
Queste componenti si collegano fra loro in un pattern Pubblisher / Subscriber e andranno ad essere configurate tramite un'interfaccia web che permette all'utente di impostare

2.3 Tecnologie coinvolte

Come scritto precedentemente si richiede l'utilizzo di un pattern Pubblisher / 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
- test unitari d'integrazione per ciascun componente in modo singolo e sull'intero sistema

2.4 Valutazione conclusiva

La scelta di sviluppare il progetto presentato in questo capitolato sta nell'utilizzo di tecnolgie richeste sul mercato come Docker e Apache Kafka.
La possibilità di creare un prodotto concreto e utilizzabile giornalmente dai developer.
Ha inoltre attirato l'attenzione del gruppo la consegna chiara e i vincoli precisi posti dall'azienda.

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) 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** (piattaforma mobile di Google di aiuto per lo sviluppo veloce di app di alta qualità), **AWS** (piattaforma Amazon che offre servizi di cloud) o altri servizi esistenti per l'immagazzinamento dei dati;
- **Hunpos** (software open-source della proponente scritto in OCaml per l'analisi di parti del discorso), **Freeling** (libreria C++ che fornisce delle funzionalità per l'analisi del linguaggio) o altri software open-source per lo svolgimento degli esercizi;

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.

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", e 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 efficacie.

4.3 Tecnologie coinvolte

- **JavaScript**
- **Grafana**
- **Rete Bayesiana** (attraverso la libreria "jsbayes")
- **GitHub**

4.4 Valutazione conclusiva

Il capitolato richiede lo studio di tecnologie inierenti all'intelligenza artificiale, un argomento ritenuto molto interessante. Purtroppo, per quanto concerne agli interessi del gruppo, il dover creare un plug-in per una tecnologia già ben strutturata, quale è Grafana, ha fatto ritenere al team che le tecnologie affrontate fossero poche e toccate in modo marginale.

5 MegaAlexa - C4

5.1 Descrizione generale

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

5.2 Obiettivo finale

Nello specifico, è richiesto lo sviluppo di una piattaforma web e mobile, in cui sarà possibile scegliere iOS o Android, che sia in grado di creare workflow custom creati dagli utenti.

5.3 Tecnologie coinvolte

- ***HTML5***$_G$, ***CSS3***$_G$, ***Javscript***$_G$:
- ***Node.js***$_G$:
- ***Kotlin***$_G$/***Swift***$_G$:
- ***Amazon Web Services***$_G$
  - ***API Gateway***$_G$;
  - ***Lambda***$_G$;
  - ***Aurora Serverless***$_G$.

5.4 Valutazione conclusiva

Riempire

6 P2PCS - C5

6.1 Descrizione generale

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

6.2 Obiettivo finale

È richiesto lo sviluppo di un sistema software che consente ad un utente la condivisione della propria auto. Tale condivisione si potrà avere una volta che l'operatore avrà inserito nel calendario i giorni in cui non utilizzerà il mezzo, lasciando così la possibilità di poterlo usufruire ad un'altra figura.

6.3 Tecnologie coinvolte

- **Google Maps**: servizio di mappe e navigazione satellitare.
- **Google Cloud Platform**: Piattaforma che permette agli sviluppatori di costruire, testare e distribuire applicazioni.
- **Henshin - movens platform**: piattaforma Open Source per la mobilità.
- **Octalysis**: È un progetto che cerca di ottimizzare le scelte umane servendosi del coinvolgimento, degli interessi e della motivazione.
- **Node.js**: piattaforma per il motore Javascript V8.
- **Java EE**: insieme di specifiche indicate per la programmazione in Java?

6.4 Valutazione conclusiva

Fin da subito, l'idea di "condividere la propria macchina con altre persone, amici o meno" non ha suscitato interesse e motivazione al gruppo. Inoltre la tecnologia Octalysis vincolava troppo l'andamento del progetto, richiedendone un uso stringente. 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 in un ecommerce.

7.2 Obiettivo finale

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

7.3 Tecnologie coinvolte

- **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.
- **Reti Ethereum**: reti per lo scambio di cryptocurrency, ad esempio MainNet.
- **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, degli smart contracts e della 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.