

21 dicembre 2015

StarWare

Quizzipedia: software per la gestione di questionari



Studio di Fattibilità

Informazioni sul documento

Nome Documento	Studio di Fattibilità
Versione	1.2.0
Stato	<i>Formale</i>
Uso	<i>Interno</i>
Data Creazione	30 Novembre 2015
Data Ultima Modifica	21 dicembre 2015
Redazione	Igor Baylyak Thomas Pigarelli Andrea Venier
Verifica	Nicola De Cao Alessio Vitella
Approvazione	Walter Sandon
Lista Distribuzione	StarWare Prof. Tullio Vardanega Prof. Riccardo Cardin Il proponente Zucchetti S.p.a.

Registro delle modifiche

Versione	Autore	Data	Descrizione
v1.2.0	Walter Sandon	2015-12-21	Correzioni minori ed approvazione
v1.1.1	Alessio Vitella	2015-12-21	Revisione: correzioni sintattiche
v1.1.0	Nicola De Cao	2015-12-20	Prima revisione: correzioni lessicali, grammaticali e di stile
v1.0.7	Thomas Pigarelli	2015-12-18	Aggiunta introduzione
v1.0.6	Thomas Pigarelli	2015-12-18	Normalizzate sezioni capitoli
v1.0.5	Andrea Venier	2015-12-08	Aggiunto capitolo 2 e 6
v1.0.4	Igor Baylyak	2015-12-08	Aggiunto capitolo 4
v1.0.3	Igor Baylyak	2015-12-08	Aggiunto capitolo 3
v1.0.2	Thomas Pigarelli	2015-12-07	Aggiunto capitolo 5
v1.0.1	Thomas Pigarelli	2015-12-06	Aggiunto capitolo 1
v1.0.0	Nicola De Cao	2015-11-30	Creazione documento

Tabella 1: Versionamento del documento

Indice

1	Introduzione	3
1.1	Scopo	3
1.2	Descrizione	3
1.3	Riferimenti	3
1.3.1	Informativi	3
2	Capitolato C5 - Quizzpedia	4
2.1	Descrizione	4
2.1.1	Dominio applicativo	4
2.1.2	Dominio tecnologico	4
2.2	Valutazione	4
2.2.1	Aspetti positivi	4
2.2.2	Potenziati criticità	5
2.3	Conclusione	5
3	Altri Capitolati	6
3.1	Capitolato C1 Actorbase	6
3.1.1	Descrizione	6
3.1.2	Aspetti positivi	6
3.1.3	Fattori di rischio	6
3.2	Capitolato C2 CLIPS	7
3.2.1	Descrizione	7
3.2.2	Aspetti positivi	7
3.2.3	Fattori di rischio	7
3.3	Capitolato C3 UMAP	8
3.3.1	Descrizione	8
3.3.2	Aspetti positivi	8
3.3.3	Fattori di rischio	8
3.4	Capitolato C4 MaaS	9
3.4.1	Descrizione	9
3.4.2	Aspetti positivi	9
3.4.3	Fattori di rischio	9
3.5	Capitolato C6 MIVOQ	10
3.5.1	Descrizione	10
3.5.2	Aspetti positivi	10
3.5.3	Fattori di rischio	10

Elenco delle tabelle

1	Versionamento del documento	1
---	---------------------------------------	---

1 Introduzione

1.1 Scopo

In questo documento vengono analizzati i requisiti, aspetti positivi e fattori di rischio di ogni capitolato proposto per evidenziare i punti di forza e gli aspetti negativi che hanno portato a scegliere o scartare un capitolato.

1.2 Descrizione

Per il capitolato scelto dal gruppo ovvero Quizzpedia_G viene presentata una descrizione e una valutazione approfondita sugli aspetti positivi, negativi e i fattori di rischio che sono stati valutati e hanno determinato la scelta di questo capitolato. Gli altri capitolati vengono presentati con una breve descrizione. Per ognuno di essi vengono inoltre elencati aspetti positivi tenuti in considerazione e fattori di rischio che hanno spinto il gruppo ad escludere i capitolati in esame.

1.3 Riferimenti

1.3.1 Informativi

- [Capitolato C1](#)
- [Capitolato C2](#)
- [Capitolato C3](#)
- [Capitolato C4](#)
- [Capitolato C5](#)
- [Capitolato C6](#)

2 Capitolato C5 - Quizzpedia

2.1 Descrizione

Il capitolato propone lo sviluppo di un sito web per la creazione e somministrazione di questionari.

2.1.1 Dominio applicativo

Le domande verranno fornite attraverso uno specifico *Quiz Markup Language* (QML), la cui creazione è uno degli oggetti principali del capitolato. L'interfaccia amministrativa dovrà supportare dispositivi *desktop*, mentre per i candidati viene richiesto il supporto di dispositivi *desktop* e *mobile*. Si dovranno considerare tre tipi di profili:

- **Amministratore:** crea profili di tipo docente e ha il completo controllo e accesso alla piattaforma;
- **Docente:** crea questionari, classi di studenti e analizza i risultati delle risposte;
- **Studente:** si iscrive a classi e completa i questionari.

2.1.2 Dominio tecnologico

Per quanto riguarda il *server* è richiesto l'utilizzo di una delle seguenti opzioni:

- Tomcat (Java);
- Node.js (JavaScript).

Per l'archiviazione sono proposte le seguenti tecnologie:

- PostgreSQL;
- MongoDB.

Per il *client* viene richiesto l'utilizzo di HTML5, CSS3 e il supporto *desktop* per l'intera applicazione e *mobile* solo per la parte destinata allo studente.

Per la creazione del linguaggio QML il gruppo ha considerato l'utilizzo di XML, di Markdown o di altri linguaggi di markup estendibili.

2.2 Valutazione

2.2.1 Aspetti positivi

Sono stati evidenziati i seguenti aspetti positivi che hanno contribuito alla scelta di questo capitolato:

- semplicità e chiarezza degli obiettivi;
- la gestione dell'interfaccia *responsive* può essere risolto attraverso i numerosi *framework* (tra cui [Bootstrap](#) e [Foundation](#));

- PostgreSQL e il sistema relazionale è propedeutico per questo corso quindi è conosciuto dall'intero gruppo;
- HTML, CSS e Javascript sono trattate nel corso di Tecnologie Web che tutti i membri del gruppo stanno o hanno seguito;
- il cuore applicativo è semplice da implementare ed è possibile utilizzare un approccio incrementale per lo sviluppo delle varie funzioni, sia richieste che opzionali.

2.2.2 Potenziali criticità

Sono state inoltre rilevate possibili punti critici nello sviluppo applicativo e/o nelle tecnologie utilizzate che dovranno essere tenuti sotto controllo:

- nel caso di utilizzo di MongoDB, la tecnologia il paradigma (NoSQL) non sono affatto conosciuti dai membri del gruppo;
- utilizzando Javascript bisogna tenere a mente che la semplicità e la dinamicità del linguaggio sono armi a doppio taglio;
- le librerie Tomcat e Node.js sono sconosciute a tutti i membri del gruppo;

Per lo sviluppo Javascript, vengono consigliati strumenti per un maggiore controllo della correttezza del codice (come [Flow](#)) oppure altri linguaggi tipati che compilati producono Javascript (come [CoffeeScript](#)).

2.3 Conclusione

Nonostante la scelta delle tecnologie da utilizzare verrà affrontata più avanti, le possibili scelte fornite dal capitolato sono tecnologie molto utilizzate nelle aziende per cui il loro utilizzo ci permetterebbe di apprendere nuove conoscenze e competenze spendibili nel mondo lavorativo.

Il dominio applicativo risulta chiaro e più in linea con il tipo di progetti a cui siamo stati sottoposti da altri corsi. Inoltre il capitolato è abbastanza flessibile da permettere di limitare alcuni rischi scegliendo strumenti più sicuri e familiari e di applicare direttamente conoscenze apprese dal corso di ingegneria del software quali un approccio incrementale per le varie funzionalità dell'applicazione.

Per questi motivi crediamo che gli obiettivi del capitolato e le tecnologie utilizzate siano in linea con gli obiettivi formativi e le competenze che vogliamo acquisire.

3 Altri Capitoli

3.1 Capitolato C1 Actorbase

3.1.1 Descrizione

Il capitolato richiede l'implementazione di un *database* NoSQL di tipo *key-value* basato sull'architettura ad attori. E' inoltre richiesta la definizione di un *domain-specific-language* (DSL) per poter interagire con il *database* da linea da comando.

3.1.2 Aspetti positivi

- Scala risulta un linguaggio interessante e conciso;
- l'architettura proposta non è di difficile comprensione;
- il modello *actor-model* è presentato anche da Programmazione Concorrente e Distribuita di quest'anno, corso che è seguito dalla maggior parte del gruppo;
- possibilità di scegliere Java al posto di Scala, considerando che tutto il gruppo ha esperienza con il primo;
- il proponente interno potrebbe avere una maggiore disponibilità.

3.1.3 Fattori di rischio

- la libreria Akka è per lo più sconosciuta ai membri del gruppo;
- nel caso si scelga Java, l'interazione con la libreria Akka risulterebbe di gran lunga più verbosa in quanto essa è stata pensata per Scala;
- poca conoscenza delle caratteristiche dei sistemi distribuiti;
- lo sviluppo di un DSL appropriato potrebbe richiedere conoscenze relative al *parsing* e analisi lessicale;
- se Scala è scelto come linguaggio implementativo, sarà necessario imparare il linguaggio e l'ecosistema di strumenti e librerie a sua disposizione.

3.2 Capitolato C2 CLIPS

3.2.1 Descrizione

Per questo capitolato è richiesta l'implementazione di un'applicazione per *smartphone*. L'applicazione deve permettere di gestire la micro-localizzazione dove è presente la tecnologia BLE Beacon, e veicolare all'utente finale informazioni contestuali a questi luoghi.

3.2.2 Aspetti positivi

- il gruppo ha trovato questo capitolato indubbiamente interessante perché posto nell'ambito della ricerca;
- permetterebbe lo studio di tecnologie recenti;
- la possibilità di proporre e sviluppare una propria idea originale.

3.2.3 Fattori di rischio

- la tecnologia dei Beacons BLE e le sue applicazioni sono ancora immature;
- abbiamo ritenuto che un costo medio di 25 euro per Beacon fosse eccessivo mettendolo in relazione con le funzionalità offerte;
- l'obbligo da parte del sistema operativo Android o IOS di tenere attivo il sistema di localizzazione dello *smartphone* per poter localizzare i Beacons risulta in un consumo notevole della batteria;
- l'impossibilità di interagire in alcun modo con i Beacons senza un'applicazione dedicata.
- difficoltà nel proporre un'idea non banale e funzionale considerando l'imprecisione e il tempo di latenza nel rilevamento dei Beacons da parte di uno *smartphone*.

3.3 Capitolato C3 UMAP

3.3.1 Descrizione

Questo capitolato richiede l'implementazione di un sistema per la raccolta e la elaborazione di dati eterogenei, dando inoltre la possibilità all'utente finale di leggere i dati e l'analisi fatta sugli stessi da un *engine* predittivo.

Il capitolato richiede lo sviluppo di:

- frontend;
- business Logic API;
- *engine* predittivo;
- broker MQTT.

3.3.2 Aspetti positivi

- le tecnologie necessarie per svolgere il capitolato sono svariate (Java, MongoDB, AWS , HTML e molte altre) il che permetterebbe al gruppo di venire in contatto con strumenti nuovi ed estremamente utili in ambito lavorativo.

3.3.3 Fattori di rischio

- lo studio e l'apprendimento delle tecnologie necessarie sicuramente risulterebbe in uno sforzo notevole da parte di ogni componente del gruppo;
- un'altra difficoltà risiede nella realizzazione del *engine* predittivo dato che al gruppo mancano totalmente le idee e le conoscenze per concepire l'elaborazione di dati eterogenei;
- il sistema da realizzare sarebbe incredibilmente ampio.

3.4 Capitolato C4 MaaS

3.4.1 Descrizione

Maas è basato su un prodotto di nome Maap, che è un database NoSql per il trattamento di dati *business* attraverso un interfaccia grafica permettendone l'utilizzo a persone che non sono esperte di *database*. Il committente richiede che questo prodotto diventi un servizio gestito attraverso un sito web che dia la possibilità di eliminare l'onere all'utente di eseguire la preparazione e l'installazione di Maap.

Sono stati rilevati i seguenti requisiti:

- le tecnologie richieste includono:
 - Node.js per il backend, MaaS deve essere in Long Term Support [LTS] versione Argon;
 - mongoDB versione maggiore di 3.x come database per l'applicazione;
- il Sistema deve essere costruito usando un *framework* di alto livello per Node.js (noi fortemente consigliamo di usare IBM loopback);
- il sistema deve essere schierabile_G con Heroku;
- il codice sorgente deve essere pubblicato e versionato usando GitHub oppure Bitbucket.

3.4.2 Aspetti positivi

- potenzialmente la mole di lavoro e la complessità dello stesso potrebbero essere esigue dato che il capitolato si basa già su un prodotto esistente che deve essere prima compreso e studiato in tutte le sue parti;
- alcune delle tecnologie a supporto del capitolato potrebbero essere , anche se in modo parziale, già note al gruppo (HTML, CSS, DBMS, Java).

3.4.3 Fattori di rischio

- l'argomento risulta essere poco stimolante, dato che la parte più interessante sarebbe stata la creazione di Maap che è già stato implementato;
- essendo il committente una *startup* con risorse di tempo, personale e finanze limitate è possibile che sia difficile aiutare e seguire in gruppo di lavoro in caso di problemi o dubbi inerenti al progetto.

3.5 Capitolato C6 MIVQQ

3.5.1 Descrizione

Il Capitolato richiede che il gruppo di lavoro proponga un'idea e realizzi applicazione mobile che sfrutti le funzionalità di *text-to-speech* fornite attraverso delle API via HTTP.

3.5.2 Aspetti positivi

- le tecnologie necessarie per la creazione dell'applicazione mobile sono: su Android Java e la libreria android-SDK, su IOS Objective-C, su WindowsPhone .NET. Tutti questi linguaggi e *framework* sono largamente documentati e usati, elemento che faciliterebbe sicuramente il lavoro del gruppo;
- il capitolato può essere molto stimolante grazie al fatto che il proponete da la possibilità al gruppo di sviluppare un'idea originale per l'applicazione.

3.5.3 Fattori di rischio

- scarso interesse da parte dei membri del gruppo di lavoro nella tecnologia TTS;
- difficoltà nel trovare un'idea per un'applicazione che fosse innovativa e utile;
- le tecnologie necessarie alla realizzazione del capitolato non sono conosciute dai membri del gruppo e c'è il rischio di non riuscire ad avere una stima approssimativa della mole di lavoro da svolgere;
- mancanza di conoscenze nell'implementazione di meccanismi atti a risolvere le problematiche legate all'utilizzo di un servizio remoto;
- l'obiettivo di rendere disponibile la sintesi vocale a tutte le applicazioni e fornire un'interfaccia di configurazione estesa comporta conoscenze specifiche e uno studio particolarmente approfondito sulle modalità di interazione tra applicazioni e funzionalità del sistema operativo.