8 dicembre 2015

StarWare

Quizzipedia: software per la gestione di questionari

Template documenti per SWE

Informazioni sul documento

Nome Documento	Template documenti per SWE
Versione	1.0.1
Stato	*Informale* // *Formale*
Uso	*Interno* // *Esterno*
Data Creazione	30 Novembre 2015
Data Ultima Modifica	8 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
v1.0.0	Nicola De Cao	30 Novembre 2015	Prima revisione
v1.0.1	Thomas Pigarelli	6 Dicembre 2015	Aggiunto capitolato 1
v1.0.2	Thomas Pigarelli	7 Dicembre 2015	Aggiunto capitolato 5
v1.0.2	Igor Baylyak	8 Dicembre 2015	Aggiunto capitolato 3
v1.0.2	Igor Baylyak	8 Dicembre 2015	Aggiunto capitolato 4

Tabella 1: Versionamento del documento

Indice

1 Introduzione

1.1 Scopo del Documento

1.2 Scopo del Prodotto

1.3 Glossario

1.4 Riferimenti

1.4.1 Normativi

1.4.2 Informativi

2 Capitolato C5 - Quizzpedia

2.1 Descrizione

Il capitolato propone la creazione di un sito web per la creazione e somministrazione di questionari scelti per argomento.

2.1.1 Dominio applicativo

Le domande verranno fornite attraverso uno specifico *Quiz Markup Language* (QML), la cui creazine è oggetto principale del capitolato. L'interfaccia amministrativa dovrà supportare dispositivi desktop, mentre per i candidati viene richiesto il supporto di dispositivi desktop e mobili.

Si dovranno considerare tre tipi di profili:

- Amministratore (crea profili, gestisce questionari e domande);
- Insegnante (crea domande e questionari e analizza i risultati);
- Studente (completa i questionari).

con i relativi permessi.

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 o 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;
- utilizzando Javascript bisogna tenere a mente che la semplicità e la dinamicità del linguaggio sono armi a doppio taglio;
- sia Tomcat che Node.js sono sconosciuti a tutti i membri del gruppo;
- le tecnologie utilizzate sono per lo più già conosciute e non contribuiscono molto al curriculum individuale.

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 alcune tecnologie siano sconosciute o presentino aspetti negativi, il capitolato è abbastanza flessibile da permettere di limitare tali rischi scegliendo tecnologie più sicure o strumenti che rendano più difficile commettere errori.

Il dominio applicativo risulta chiaro e più in linea con il tipo di progetti a cui siamo stati sottoposti da altri corsi. Per cui crediamo che gli obiettivi e le tecnologie utilizzate ci diano una maggiore sicurezza, al costo dell'esposizione a tecnologie e modelli meno stimolanti.

3 Altri Capitolati

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

Il gruppo ha rilevato i seguenti 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 maggiorparte del gruppo;
- possibilità di scegliere Java al posto di Scala, considerando che tutto il gruppo ha asperienza con il primo;
- il proponente interno potrebbe avere una maggiore disponibilità.

3.1.3 Fattori di rischio

Questo capitolato non è stato scelto per i seguenti motivi:

- 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 dispozione.

3.2 Capitolato C2 CLIPS

3.2.1 Valutazione Generale

Per questo capitolato è richiesta l'implementazione di un'applicazione per Smartphone. L'applicazione deve permettere di gestire la microlocalizzazione di luoghi fisici, dove è presente la tecnologia BLE Beacon, e veicolare al utente finale informazioni contestuali a questi luoghi. Il gruppo ha trovato questo capitolato indubbiamente interessante perchè posto nell'ambito della ricerca,permettendo lo studio di tecnologie recenti, e anche perchè viene data la possibilità di proporre e sviluppare una propria idea originale. Tuttavia a causa una serie di limiti legati alla tecnologia Beacon BLE, il team ha deciso di propendere verso un altro capitolato.

Si è scelto di non accettare questo capitolato per i seguenti motivi:

- La teconologia dei Beacon BLE e le sue applicazioni sono ancora immature.
- Abbiamo ritenuto che un costo medio di 25 euro per Beacon fosse eccessivamente 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 Beacon, risultando in un consumo notevole della batteria.
- L'impossibilità di interagire in alcun modo con i Beacon senza un applicazione dedicata.

3.2.2 Fattori di Rischio

- Difficoltà nel proporre un idea non banale e funzionale considerando l'imprecisione e il tempo di latenza nel rilevamento dei Beacon da parte si 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.

3.3.2 Requisiti obbligatori

Sviluppo di :

1. Frontend;
2. Business Logic API;
3. Engine predittivo;
4. Broker MQTT.

3.3.3 Aspetti Positivi:

1. 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.

3.3.4 Fattori di Rischio:

1. Lo studio e l'apprendimento delle tecnologie necessarie sicuramente risulterebbe in uno sforzo notevole da parte di ogni componente del gruppo;
2. Un altra difficolta risiede nella realizzazione del motore predittivo dato che al gruppo mancano totalmente le idee e le conoscenze per concepire l'elaborazione di dati eterogenei, considerando anche il fatto che durante la presentazione non sono stati dati spunti a riguardo;
3. Il sistema da realizzare sarebbe incredibilmente amplio.

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 account web che dia la possibilità di eliminare l'onere all'utente di eseguire la preparazione e l'installazione di Maap.

3.4.2 Requisiti obbligatori

1. Le tecnologie richieste includono:
   (a) Node.js per il backend. MaaS deve essere in Long Term Support [LTS] versione Argon;
   (b) MongoDB, versione maggiore di 3.x come database per l'applicazione;
2. Il Sistema deve essere costruito usando un framework di alto livello per Node.js. Noi fortemente consigliamo di usare IBM loopback;
3. Il sistema deve essere schierabile con Heroku;
4. Il codice sorgente deve essere pubblicato e versionato usando github oppure bitbucket.

3.4.3 Aspetti Positivi:

1. Potenzialmente la mole di lavoro e la complessità dello stesso potrebbero essere esigue dato che il capitolato si basa già su un prodotto esistente;
2. Alcune delle tecnologie a supporto del capitolato potrebbero essere , anche se in modo parziale, già note al gruppo (html, css, un servizio di database, java).

3.4.4 Fattori di Rischio:

1. L'argomento risulta essere poco stimolante, dato che la parte più interessante sarebbe stata la creazione di Maap che è già stato implementato;
2. 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 MIVOQ

3.5.1 Valutazione Generale

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 HTTP. Si è deciso di non accettare il capitolato per i seguenti motivi:

- Scarso interesse da parte dei membri del gruppo di lavoro nella teconologia TTS.
- Difficoltà nel trovare un idea per un applicazione che fosse innovativa e utile.
- Le Tecnologie necessarie alla realizzazione del capitolato dovrebbero essere tutte studiate da zero con il rischio di non riuscire ad avere una stima approssimativa della mole di lavoro da svolgere.

3.5.2 Fattori di Rischio

- Mancanza di conoscenze nell'implementazione di meccanismi atti a risolvere le problematiche legate all'utilizzo di un servizio remoto.
- L'obbiettivo 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 per l'installazione e della interazione tra applicazioni e funzionalità del sistema operativo.