



## Verbale Esterno 2016-01-19

### Informazioni sul documento

<b>Versione</b>	1.0.0
<b>Redazione</b>	Giacomo Vanin
<b>Verifica</b>	Marco Boseggia
<b>Approvazione</b>	Andrea Giacomo Baldan
<b>Uso</b>	Esterno
<b>Lista di Distribuzione</b>	ScalateKids Prof. Tullio Vardanega Prof. Riccardo Cardin



## Diario delle modifiche

Versione	Autore	Ruolo	Data	Descrizione
1.0.0	Andrea Giacomo Baldan	Responsabile	2016-01-21	Validazione documento
0.1.1	Marco Boseggia	Verificatore	2016-01-20	Verifica documento
0.1.0	Giacomo Vanin	Analista	2016-01-20	Prima stesura documento
0.0.1	Alberto De Agostini	Analista	2016-01-12	Creazione scheletro del documento



## Indice

<b>1</b>	<b>Informazioni sulla riunione</b>	<b>3</b>
<b>2</b>	<b>Domande e risposte</b>	<b>4</b>



## 1 Informazioni sulla riunione

- **Data:** 2015-01-19
- **Luogo:** Torre Archimede aula 1C150
- **Orario d'inizio:** 13:15
- **Durata:** 15'
- **Partecipanti interni:** *ScalateKids*
  - Andrea Giacomo Baldan
  - Alberto De Agostini
  - Davide Trevisan
  - Giacomo Vanin
  - Marco Boseggia
  - Michael Munaro
- **Partecipanti esterni:**
  - Riccardo Cardin



## 2 Domande e risposte

In questo capitolo sono elencate le domande fatte dal gruppo *ScalateKids* in grassetto e le risposte del proponente *Riccardo Cardin* in corsivo.

### **E' necessario fornire un sistema di autenticazione per il database?**

*Quando ci si connette a un database è ragionevole fornire una serie di credenziali. Nel capitolato non è espressamente scritto ma è una caratteristica che hanno tutti i database quindi è fortemente consigliato.*

### **Il prodotto finale sarà un client e server allo stesso tempo in modo da garantire scalabilità orizzontale?**

*Il prodotto finale sarà un'applicazione distribuita ma saranno tutti peer<sub>c</sub>, cioè parti che interoperano tra di loro per dare a un client finale il risultato di un'operazione.*

### **E' ragionevole quindi che con il prodotto finale forniamo un eseguibile con due istanze, una per la modalità server e una per la modalità client?**

*Dovrà esserci un eseguibile che avrà due componenti: il server e l'interfaccia da riga di comando. Quest'ultima sarà il vostro client.*

### **Nei casi d'uso come dovrà essere rappresentata l'interazione tra i vari "server" e il client?**

*Dovranno essere rappresentati come attori secondari.*

### **Quanto dovranno andare nel dettaglio i casi d'uso? Dovrà esserci la shell che parla con il DBMS<sub>c</sub>, il DBMS<sub>c</sub> che parla con la base di dati effettiva?**

*Basterà considerarlo come un'unico blocco. Per questa fase è sufficiente fermarsi a un livello di dettaglio in cui avete i casi d'uso riguardanti la shell e il driver. Dovrete inoltre differenziare gli errori possibili. Potrebbe succedere, ad esempio, che il sistema vada in timeout. Pensate bene alle operazioni che volete fornire al vostro database. La definizione di DBMS<sub>c</sub> tuttavia è a livello architetturale ed esula dall'ambito dei casi d'uso attuali, quindi ci penserete meglio durante le attività architetturali.*

### **Abbiamo visto il funzionamento di Amazon Dynamo per prendere spunto. Dobbiamo puntare a qualcosa del genere? Cioè dobbiamo avere uno schema per ogni chiave o diamo per scontato che sia una stringa?**

*Scegliete voi il modello da seguire in base alle vostre esigenze. L'importante è che il sistema funzioni. Ovviamente fare una copia di un sistema già esistente sarebbe poco sfidante, non sarebbe un'opportunità.*