



Verbale Esterno 2016-01-12

Informazioni sul documento

Versione	1.0.0
Redazione	Alberto De Agostini
Verifica	Marco Boseggia
Approvazione	Andrea Giacomo Baldan
Uso	Esterno
Lista di Distribuzione	ScalateKids Prof. Tullio Vardanega Prof. Riccardo Cardin



Diario delle modifiche

Versione	Autore	Ruolo	Data	Descrizione
1.0.0	Andrea Giacomo Baldan	Responsabile	2016-01-13	Validazione documento
0.1.1	Marco Boseggia	Verificatore	2016-01-12	Verifica documento
0.1.0	Alberto De Agostini	Analista	2016-01-12	Prima stesura documento
0.0.1	Alberto De Agostini	Analista	2016-01-12	Creazione scheletro del documento



Indice

1 Informazioni sulla riunione	3
2 Domande e risposte	4



1 Informazioni sulla riunione

- **Data:** 2015-01-12
- **Luogo:** Torre Archimede aula 1C150
- **Orario d'inizio:** 13:15
- **Durata:** 25'
- **Partecipanti interni:** *ScalateKids*
 - Andrea Giacomo Baldan
 - Alberto De Agostini
 - Francesco Agostini
 - 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.

Nel capitolato non c'è una chiara divisione tra requisiti desiderabili e requisiti opzionali, c'è semplicemente un elenco di requisiti opzionali. Vi è una preferenza da parte sua tra questi?

Avendoli lasciati tutti opzionali nel capitolato, rimando a voi la scelta su quali pensate siano più importanti, cercando di implementare prima gli opzionali che al meglio completano quelli obbligatori.

Dovremo istanziare un terminale isolato? (come ad esempio una shell_c MySql)

No, la CLI_c da creare sarà composta da un gruppo di semplici comandi da usare direttamente da terminale_c, il quale non sarà altro che (uno dei) client che si connetteranno al DB al fine di effettuare operazioni. Ciò naturalmente non implica che il prodotto sia da sviluppare esclusivamente per ambienti linux, terminale_c è inteso come ui_c comandi generica(es bash_c).

Nel capitolato, come requisiti obbligatori, sono presenti solo i comandi per inserimento, cancellazione e modifica dati. Un database generalmente consente più operazioni di queste, alcune delle quali riteniamo essere essenziali (es. query_c). Sono queste da considerarsi requisiti opzionali?

Sicuramente dev'essere possibile interrogare il database, gli altri comandi basilari sono dei requisiti obbligatori impliciti. La lista delle funzionalità tuttavia è una cosa che voi dovete decidere e scrivere nei vostri documenti.

Nel capitolato c'è scritto che gli attori_c di tipo *Manager* sono responsabili della gestione del numero massimo di coppie chiave / valore contenute in un'istanza di un attore *storekeeper_c*. Con che criterio deve essere scelto questo numero massimo?

Può essere scelto in diversi modi; il più facile di tutti è senz'altro scegliere un numero prefissato da impostare, altrimenti si potrebbe scegliere di cambiare questo numero dinamicamente in base al carico. Tuttavia questo non è un compito semplice, quindi anche in questo caso la decisione spetta a voi.

Abbiamo deciso di utilizzare il linguaggio *Scala* per sviluppare il prodotto. Le ultime versioni di questo linguaggio non sono compatibili tra loro, quale versione ci consiglia di utilizzare? (2.10 o 2.11)

Suggerirei di utilizzare le ultime versioni, soprattutto dal punto di vista di compatibilità, utilizzare la major_c più recente vi darebbe qualche garanzia in più nella compatibilità con le librerie chiave del



progetto (akka_c). La JVM_c di riferimento dovrebbe essere dunque la versione 8.

Cosa dovrà arrivare all'utente finale visto come utilizzatore? Le librerie akka_c dovrebbero figurare come dipendenze da installare o dovrebbero essere fornite assieme al prodotto?

Dovrete consegnare un eseguibile comprensivo di librerie bundled_c per semplificare l'installazione all'utente. Tuttavia è da tenere presente che il prodotto dovrà essere al 100% scalabile orizzontalmente_c, il suo ambito di utilizzo quindi dovrà spaziare dal singolo privato, a grandi reti di server, all'occorrenza.

L'attore warehousemen_c dovrà occuparsi della persistenza su disco dei dati inseriti nelle mappe. Quale approccio consiglia?

Sicuramente dovrà essere fornita un interfaccia da parte dell'attore che si occupa della persistenza, come effettivamente venga implementato il salvataggio dati su disco è lasciato a voi, l'unico requisito importante è l'estendibilità (e questo si applica a tutto il sistema), deve essere possibile estendere l'attore che si occupa di persistenza senza difficoltà, nel caso in cui un utente decida di implementare il proprio warehousemen_c e sostituirlo alla version "base"