

**Università degli Studi “Roma Tre”**

Dipartimento di Informatica ed automazione

**Facoltà di Ingegneria**

Corso di Laurea Magistrale in Ingegneria Informatica

Tesi di Laurea

**Sviluppo di un framework per la sperimentazione di Architetture Software**

Relatore Candidato

**Prof. L. Cabibbo** **Francesco Paris** Università di Roma Tre matricola 453908

Anno Accademico 2014 – 2015

*Ai miei genitori, Giuseppe e Monica,*

*per il costante e prezioso*

*supporto in questi anni di studio*

**INTRODUZIONE**

Il mercato attuale richiede lo sviluppo di applicazioni che siano distribuite, transazionali, portabili, sicure, veloci e affidabili. Tali sistemi devono essere sviluppati nel minor tempo possibile per permettere ad una azienda di restare competitiva in un mercato costantemente soggetto a variazioni. La realizzazione di tali applicazioni rappresenta una sfida formidabile per i team di sviluppo che si devono districare in una ragnatela di tecnologie costantemente variabili e con peculiarità diverse. Per rispondere alle esigenze del cliente, sia a livello di funzionalità che di qualità, uno sviluppatore deve possedere un forte background metodologico e tecnologico. A valle di tali riflessioni si denota immediatamente l’importanza di un corso di Architetture Software erogato in ambito universitario. Tale insegnamento ha infatti l’obiettivo di presentare ad un discente aspetti sia metodologici sia tecnologici relativi alle architetture software, che hanno un ruolo fondamentale nel raggiungimento degli obiettivi di qualità dei sistemi software. Tali conoscenze, per uno studente, rappresentano un bagaglio culturale immediatamente spendibile nel mondo lavorativo. Nel corso vengono affrontate diverse tecnologie. Alcune sono state ampiamente usate nello sviluppo del software, altre rappresentano le tendenze attuali e le possibili evoluzioni future. Tuttavia per avere una vera e propria conoscenza è necessario che lo studente sperimenti con mano tali tecnologie. Una sperimentazione efficace, che prenda in considerazione e mostri il loro reale uso, può essere difficoltosa. Lo scopo di tale tesi è quello di creare una infrastruttura che proponga delle idee per permettere allo studente di utilizzare tecnologie ormai consolidate nello sviluppo del software ma anche di entrare in serio contatto con tecnologie all’avanguardia.

Per ottemperare a tale obiettivo, una prima fase del lavoro di tesi è stata impiegata per rintracciare una serie di strumenti che ben si prestassero ad essere introdotti in un corso come Architetture Software per sperimentare quanto appreso durante le lezioni. Gli strumenti identificati hanno la caratteristica di essere semplici da usare e con una curva di apprendimento bassa in modo da non costituire un aggravio per lo studente. Successivamente l’attenzione è stata rivolta ad individuare una applicazione d’esempio che ben si prestasse ad essere impiegata per una realizzazione che sfruttasse gli strumenti identificati, le tecnologie e le metodologie affrontate durante il corso di Architetture Software. Tale applicazione è stata quindi implementata in una pluralità di modi pensando ad una possibile erogazione come homework da destinare agli studenti del corso.

Per quanto concerne gli strumenti, è stato proposto l’utilizzo di Maven il cui scopo è quello di aiutare lo sviluppatore nella gestione di progetti Java e fornire funzioni di build automation. Nel corso di Architetture Software vengono trattati argomenti, sia di carattere tecnico che metodologico, che spesso hanno a che fare non con sistemi standalone ma distribuiti. Per far simulare allo studente un ambiente distribuito è stato usato Vagrant, in combinazione con Virtual Box, con lo scopo di effettuare in modo automatizzato il provisioning di macchine virtuali. Attraverso questi strumenti lo studente, con uno sforzo di configurazione assai ridotto, potrà creare una rete virtuale in cui testare applicazioni client-server distribuite. Seguendo una tendenza attuale del mercato, è stato inoltre utilizzato Spring Framework, celebre framework Java lighweight. Nel lavoro di tesi, Spring è stato usato in maniera intensa per la realizzazione dell’interfaccia web e sono state anche testate le capacità del modulo Spring-Integration. Visto l’ampio utilizzo di tale framework si è cercato di fare un analisi architetturale di Spring ed un confronto con Java Platform Enteprise Edition (Java EE). Come strumento di supporto per la gestione delle configurazioni software è stato proposto GitHub, celebre piattaforma di social-coding che permette con estrema facilità a gruppi di studenti di collaborare suddivisi in team. Un discente, lavorando in piccoli gruppi eterogenei, ha la possibilità di accrescere le proprie competenze come sancito dallo psicologo Vygotskij nell’ambito della teoria del “Cooperative Learning” .

Come scenario di caso d’uso è stata ideata un’applicazione clone di Car2Go, noto servizio di car-sharing. Lo scopo di tale applicazione è quello di gestire il noleggio e la prenotazione di un parco auto all’interno del quadrante del Grande Raccordo Anulare di Roma. Un utente può registrarsi al sistema ed effettuare prenotazioni di veicoli. Una volta che un veicolo è stato prenotato, verrà simulata la guida ottenendo dati sulla distanza e sul tempo necessario a completare il percorso secondo il traffico attuale grazie all’interazione con i servizi messi a disposizione da Google Maps. Nell’applicazione è presente un amministratore che ha il compito di gestire le utenze nonché decretare la necessità di effettuare i rifornimenti dei veicoli, pulizia interna o riparazione dell’auto. Un utente può interagire con l’applicazione mediante un’interfaccia web. Parte del lavoro di tesi prevedeva quindi la realizzazione delle funzionalità dell’applicazione in vari modi mantenendo però fissa la parte di presentazione realizzata mediante delle pagine web. Al fine di strutturare l’applicazione in modo consono è stato seguito lo stile architetturale “Ports and Adapters”. In questo modo è possibile utilizzare servizi/funzionalità di natura diversa senza dover cambiare il resto dell’applicazione. Le varie funzionalità sono state realizzate prima come semplici managed-bean di Spring, poi mediante RMI, Enterprise Java Beans, Web Service (Soap e Rest). E’ stato anche previsto l’erogazione di un caso d’uso che sfrutti JMS.

Nel lavoro di tesi le principali difficoltà hanno riguardato la configurazione di ambienti virtuali che fanno uso di application server Java EE come Glassfish e il far colloquiare tecnologie estremamente diverse.

Nella realizzazione della presente tesi, in ogni capitolo, si è cercato di inquadrare una possibile problematica e proporre una soluzione andandone ad evidenziare quali sono i punti di forza ed, eventualmente, le debolezze. In totale sono stati redatti X capitoli.

Nel primo capitolo è stato motivato l’uso di Maven andando ad illustrare le principali azioni da adottare per facilitare il lavoro di sviluppo.

Nel secondo capitolo è stato affrontato Vagrant con un’ampia discussione su Chef e Puppet, linguaggi alternativi per effettuare il provisiong di macchine virtuali.

Nel terzo capitolo è presente l’analisi architetturale di Spring, spiegando in modo dettagliato quali sono i punti di forza che hanno decretato il successo di tale framework nella comunità degli sviluppatori Java.

Nel quarto capitolo vengono affrontate le problematiche relative alla gestione dei cambiamenti ed alla necessità di utilizzare un sistema per il controllo di versione. Come strumento viene suggerito l’uso di GitHub.

Nel quinto capitolo viene presentato lo studio di caso. Si affronta pertanto l’analisi dell’applicazione clone di Car2Go andando ad analizzare i requisiti funzionali e di qualità.

Nel sesto capitolo viene illustrato il pattern “Ports and Adapters – Archiettura Esagonale” utilizzato per strutturare l’applicazione clone di Car2Go.

Nel settimo capitolo vengono descritte le scelte progettuali effettuate per l’implementazione dello strato di persistenza e le tecnologie utilizzate.

Scopo dell’ottavo capitolo è quello di spiegare le decisioni assunte nell’implementazione della logica di business del sistema.

Il nono capitolo illustra le scelte per l’implementazione dell’interfaccia web, a livello di decisioni progettuali e di tecnologie adottate.

Nel decimo capitolo sono motivate le scelte effettuate per la realizzazione di una versione dell’applicazione basata su RMI, mentre nell’undicesimo capitolo è spiegato l’uso di JMS.

Nel dodicesimo capitolo è illustrata una versione dell’applicazione basata su EJB.

Nel tredicesimo capitolo viene descritto come sia possibile usare i WebService per erogare le funzionalità dell’applicazione.