

Università degli Studi di Padova

DIPARTIMENTO DI MATEMATICA

CORSO DI LAUREA IN INFORMATICA



## Sviluppo di un modulo in Alfresco, sistema di gestione della conoscenza

*Tesi di laurea triennale*

*Relatore*

Prof. Gaggi Ombretta

*Laureando*

Trevisan Davide

---

ANNO ACCADEMICO 2016-2017



Lorem ipsum dolor sit amet, consectetur adipiscing elit.

— Oscar Wilde

Dedicato a ...



# Abstract

Scopo di questa tesi di laurea è esporre il lavoro svolto dal laureando Trevisan Davide durante lo stage di trecentoventi ore presso l'azienda Ennova Research SRL con sede in Venezia-Mestre. Il progetto di stage si è incentrato sullo sviluppo di alcune funzionalità connesse al sistema di gestione della conoscenza che l'azienda utilizza per gestire la documentazione relativa ai progetti assegnati dalle aziende clienti e monitorare l'operato dei dipendenti. Il sistema si basa sulla piattaforma Alfresco, un noto e molto utilizzato [KMS](#), la quale offre una SDK (che verrà descritta estensivamente ed esaurientemente in seguito) lanciabile in maniera autonoma tramite Maven che consente lo sviluppo di moduli per personalizzare e ottimizzare la piattaforma a proprio piacimento. Nella realizzazione del progetto è stato possibile, pertanto, approfondire le potenzialità, i difetti e le caratteristiche della piattaforma Alfresco nonché le fasi che hanno portato alla realizzazione di moduli per aggiungere ad Alfresco nuove funzionalità e una maggiore customizzazione del suo aspetto. Tutte le fasi, le problematiche e quanto prodotto durante il progetto sarà accuratamente esposto nei capitoli che compongono la presente tesi.



*“Life is really simple, but we insist on making it complicated”*

— Confucius

# Ringraziamenti

*Innanzitutto, vorrei esprimere la mia gratitudine al Prof. NomeDelProfessore, relatore della mia tesi, per l'aiuto e il sostegno fornitomi durante la stesura del lavoro.*

*Desidero ringraziare con affetto i miei genitori per il sostegno, il grande aiuto e per essermi stati vicini in ogni momento durante gli anni di studio.*

*Ho desiderio di ringraziare poi i miei amici per tutti i bellissimi anni passati insieme e le mille avventure vissute.*

*Padova, Aprile 2017*

Trevisan Davide





# Indice

<b>1</b>	<b>Introduzione</b>	<b>1</b>
1.1	L'azienda . . . . .	1
1.2	Lo stage . . . . .	1
<b>2</b>	<b>tecnologie e strumenti utilizzati</b>	<b>3</b>
2.1	Tecnologie utilizzate . . . . .	3
2.1.1	Linguaggi di Programmazione . . . . .	3
2.1.2	Formato per l'interscambio di dati . . . . .	3
2.1.3	Tecnica per l'interscambio di dati . . . . .	4
2.1.4	Framework . . . . .	4
2.1.5	Protocolli di servizi . . . . .	5
2.2	Strumenti utilizzati . . . . .	5
<b>3</b>	<b>Descrizione dello stage</b>	<b>7</b>
3.1	Introduzione al progetto . . . . .	7
3.2	Analisi preventiva dei rischi . . . . .	7
3.3	Requisiti e obiettivi . . . . .	7
3.4	Pianificazione . . . . .	7
<b>4</b>	<b>Analisi dei requisiti</b>	<b>9</b>
4.1	Casi d'uso . . . . .	9
4.2	Tracciamento dei requisiti . . . . .	10
<b>5</b>	<b>Progettazione e codifica</b>	<b>13</b>
5.1	Tecnologie e strumenti . . . . .	13
5.2	Ciclo di vita del software . . . . .	13
5.3	Progettazione . . . . .	13
5.4	Design Pattern utilizzati . . . . .	13
5.5	Codifica . . . . .	13
<b>6</b>	<b>Verifica e validazione</b>	<b>15</b>
<b>7</b>	<b>Conclusioni</b>	<b>17</b>
7.1	Consuntivo finale . . . . .	17
7.2	Raggiungimento degli obiettivi . . . . .	17
7.3	Conoscenze acquisite . . . . .	17
7.4	Valutazione personale . . . . .	17
<b>A</b>	<b>Appendice A</b>	<b>19</b>

Glossary	21
Acronyms	23
Bibliografia	25

# Elenco delle figure

4.1	Use Case - UC0: Scenario principale . . . . .	9
-----	---	---

# Elenco delle tabelle

4.1	Tabella del tracciamento dei requisiti funzionali . . . . .	11
4.2	Tabella del tracciamento dei requisiti qualitativi . . . . .	11
4.3	Tabella del tracciamento dei requisiti di vincolo . . . . .	11



# Capitolo 1

## Introduzione

In questo capitolo verrà brevemente esposto il contesto in cui si è svolto lo stage, descrivendo le motivazioni che hanno spinto l'azienda a proporre questo stage

### 1.1 L'azienda

Ennova Research srl è un'azienda che opera nel settore [ICT](#) e realizza soluzioni informatiche altamente tecnologiche ed affidabili, che le permettono di agire con successo in settori di business come quello della Pubblica Amministrazione, delle grandi Corporazioni Bancarie, delle Multinazionali [ICT](#) e della Grande Distribuzione. È partner di grandi attori del mercato nazionale e internazionale quali Engineering, Toshiba, EMC, HP, Novell, Nvidia, ed altri . Ennova Research si distingue anche nel campo delle tecnologie open source utilizzate per la realizzazione di soluzioni multimediali avanzate destinate ai mercati [B2C](#) e [B2B](#) e di applicativi software destinati al settore del mobile, ad esempio Slash, che sfrutta le [Application Program Interface](#) di Twitter. L'azienda inoltre investe molto in ricerca e sviluppo ed è sempre pronta ad esplorare nuove tecnologie.

### 1.2 Lo stage

Il progetto di stage svoltosi all'interno dell'azienda Ennova Research è principalmente consistito nello sviluppo di moduli per Alfresco, che è il [KMS](#) che l'azienda ha incominciato da relativamente poco ad utilizzare e a cui intende sviluppare alcune funzionalità, quali ad esempio la gestione dei clienti e dei progetti, che al momento è svolta con l'ausilio di altri applicativi, e la rendicontazione, svolta anch'essa mediante l'utilizzo di altri applicativi. L'azienda tuttavia si pone l'obiettivo di creare non solo un prodotto necessario alle necessità interne, ma anche un prodotto vendibile e che generi profitto attraverso la vendita dello stesso ad aziende clienti. Il progetto è stato denominato Coral Tree e lo stage si è posto l'obiettivo di iniziare a porre le basi per questo progetto, attraverso la creazione di un modulo che introduca il nuovo tema per dare l'aspetto desiderato al prodotto e la creazione dei primi moduli che introducano le funzionalità precedentemente citate.



## Capitolo 2

# tecnologie e strumenti utilizzati

*In questo capitolo saranno esposte le principali tecnologie utilizzate e i principali strumenti che sono stati utilizzati per portare a compimento il progetto assegnato*

### 2.1 Tecnologie utilizzate

#### 2.1.1 Linguaggi di Programmazione

##### **Java**

Java è un linguaggio di programmazione orientato agli oggetti a tipizzazione statica, specificatamente progettato per essere il più possibile indipendente dalla piattaforma di esecuzione. Uno dei principi fondamentali del linguaggio è espresso dal motto *write once, run anywhere*: il codice compilato che viene eseguito su una piattaforma non deve essere ricompilato per essere eseguito su una piattaforma diversa. Il prodotto della compilazione è infatti in un formato chiamato *bytecode* che può essere eseguito da una qualunque implementazione di un processore virtuale detto *Java Virtual Machine*.

##### **Javascript**

JavaScript è un linguaggio di scripting orientato agli oggetti e agli eventi, comunemente utilizzato nella programmazione Web lato client per la creazione, in siti web e applicazioni web, di effetti dinamici interattivi, tramite funzioni di script invocate da eventi innescati, a loro volta, in vari modi dall'utente sulla pagina web in uso. Tali funzioni di script possono essere opportunamente inserite in file HTML, in pagine JSP o in appositi file separati con estensione .js, poi richiamati nella logica di business.

#### 2.1.2 Formato per l'interscambio di dati

##### **JSON**

JSON (JavaScript Object Notation) è un formato di scambio dati leggero e facile da leggere e scrivere sia per macchine che per umani, si basa su un sottoinsieme del linguaggio di programmazione JavaScript (Standard ECMA-262 3rd Edition - Dicembre 1999), inoltre è un formato di testo completamente indipendente da qualsiasi linguaggio.

Per queste caratteristiche è un linguaggio di scambio dati ideale. JSON è costruito su due strutture:

- \* Una collezione di coppie nome/valore, spesso realizzato come un array associativo;
- \* Un elenco ordinato di valori, spesso realizzato come una lista.

### 2.1.3 Tecnica per l'interscambio di dati

#### AJAX

AJAX, acronimo di Asynchronous Javascript And XML, è una tecnica di sviluppo software per la realizzazione di applicazioni web interattive. Lo sviluppo di applicazioni HTML con AJAX si basa su uno scambio di dati in background fra web browser e server, che consente l'aggiornamento dinamico di una pagina web senza esplicito ricaricamento da parte dell'utente. AJAX è asincrono, nel senso che i dati extra sono richiesti al server e caricati in background senza interferire con il comportamento della pagina esistente. Normalmente le funzioni richiamate sono scritte con il linguaggio JavaScript. Tuttavia, e a dispetto del nome, l'uso di JavaScript e di XML non è obbligatorio, come non è detto che le richieste di caricamento debbano essere necessariamente asincrone.

### 2.1.4 Framework

#### Spring

Spring è un framework per realizzare applicazioni web basate sul Modello MVC avendo come punti di forza l'inversion of control (tramite dependency injection) e la aspect oriented programming. Esso si occupa di mappare metodi e classi Java con determinati url, di gestire differenti tipologie di "viste" restituite al client, di realizzare applicazioni internazionalizzate e di gestire i cosiddetti temi per personalizzare al massimo l'esperienza utente. Questo framework è strutturato a livelli, e permette di scegliere quale dei suoi componenti usare, fornendo nello stesso momento un framework coesivo per lo sviluppo di applicazioni distribuite. Questa architettura a livelli consiste in diversi moduli (o componenti) ben definiti, ognuno dei quali può rimanere da solo o essere implementato con altri.

#### Jquery

JQuery è un framework nato con il preciso intento di rendere il codice più sintetico e di limitare al minimo l'estensione degli oggetti globali per ottenere la massima compatibilità con altre librerie. Grazie a questo principio jquery è in grado di offrire un'ampia gamma di funzionalità, che vanno dalla manipolazione degli stili CSS e degli elementi HTML, agli effetti grafici, per passare a comodi metodi per chiamate AJAX cross-browser.

#### Alfresco Surf

Alfresco Surf è un framework messo a disposizione della piattaforma Alfresco con cui è possibile creare interfacce grafiche, modelli e componenti sfruttando scripts server-side e templates.



### 2.1.5 Protocolli di servizi

#### LDAP

LDAP è un acronimo che sta per LIGHTWEIGHT DIRECTORY ACCESS PROTOCOL. Come suggerisce il nome stesso, è un protocollo leggero per accedere ai servizi di directory, basati sul protocollo X.500. LDAP opera su TCP/IP o su altre connessioni orientate ai servizi di trasferimento. LDAP nasce per sostituire DAP in quanto molto oneroso dal punto di vista dell'impiego delle risorse. LDAP è basato sul modello client-server: un client LDAP invia una richiesta ad un server LDAP, che processa la richiesta ricevuta, accede eventualmente ad un directory database e ritorna dei risultati al client. Il modello di informazioni di LDAP è basato sulle entry. Un'entry è una collezione di attributi aventi un unico nome globale: il Distinguished Name (DN). Il DN è usato per riferirsi ad una particolare entry, senza avere ambiguità. Ogni attributo dell'entry ha un tipo ed uno o più valori. In LDAP, le entry di una directory sono strutturate come in una struttura gerarchica di un albero.

## 2.2 Strumenti utilizzati

### Strumenti di supporto all'attività di Analisi

**Astah** Astah è un software molto diffuso per la produzione di diagrammi UML. Le sue potenzialità vanno in realtà oltre alla semplice produzione grafica di diagrammi, ma in questo progetto è stato utilizzato esclusivamente per la realizzazione dei diagrammi dei casi d'uso inseriti nell'analisi dei requisiti.

### Strumenti di supporto all'attività di Codifica

**Eclipse** Eclipse è un ambiente di sviluppo integrato multi-linguaggio e multipiattaforma. Ideato da un consorzio di grandi società quali Ericsson, HP, IBM, Intel, MontaVista Software, QNX, SAP e Serena Software, chiamato Eclipse Foundation. Eclipse è un software libero distribuito sotto i termini della Eclipse Public License ed è lo strumento che viene imposto dall'azienda ai suoi dipendenti, in quanto software libero, quindi la sua adozione per questo progetto è stata scontata. Questo IDE è stato utilizzato per l'importazione di un progetto allo scopo di estendere la piattaforma di Alfresco. Le funzionalità presentate da questo IDE in termini di aiuti al programmatore e la sua integrazione con Maven hanno permesso uno sviluppo rapido ed efficiente dei plugin.

### Strumenti per il versionamento

**Git** Il sistema di versioning adottato è Git, che negli ultimi anni si è affermato come uno dei migliori sistemi di controllo di versione. Le caratteristiche per cui si è distinto dagli altri software sono:

- \* L'architettura, progettata per essere completamente distribuita, in modo da rendere possibile il lavoro e il versionamento offline;
- \* Le potenti funzionalità con cui è possibile lavorare alle ramificazioni branching della storia del codice sorgente;
- \* Le performance generalmente migliori.

Ennova Research si appoggia inoltre su GitLab, un servizio di hosting per progetti su Git.

### **Strumenti per *build automation***

**Apache Maven** Maven è un software usato principalmente per la gestione di progetti Java e *build automation*. Per funzionalità è simile ad Apache Ant, ma basato su concetti differenti. Può essere usato anche in progetti scritti in C#, Ruby, Scala e altri linguaggi. Il progetto Maven è ospitato da Apache Software Foundation, dove faceva parte dell'ex progetto Jakarta. Maven usa un costrutto conosciuto come Project Object Model (POM); un file XML che descrive le dipendenze fra il progetto e le varie versioni di librerie necessarie nonché le dipendenze fra di esse. In questo modo si separano le librerie dalla directory di progetto utilizzando questo file descrittivo per definirne le relazioni. Maven effettua automaticamente il download di librerie Java e plug-in Maven dai vari repository definiti scaricandoli in locale o in un repository centralizzato lato sviluppo. Questo permette di recuperare in modo uniforme i vari file JAR e di poter spostare il progetto indipendentemente da un ambiente all'altro avendo la sicurezza di utilizzare sempre le stesse versioni delle librerie. Per automatizzare la build e il lancio dell'SDK, Alfresco si basa appunto su Maven.

## Capitolo 3

# Descrizione dello stage

*Breve introduzione al capitolo*

### 3.1 Introduzione al progetto

### 3.2 Analisi preventiva dei rischi

Durante la fase di analisi iniziale sono stati individuati alcuni possibili rischi a cui si potrà andare incontro. Si è quindi proceduto a elaborare delle possibili soluzioni per far fronte a tali rischi.

#### 1. Performance del simulatore hardware

**Descrizione:** le performance del simulatore hardware e la comunicazione con questo potrebbero risultare lenti o non abbastanza buoni da causare il fallimento dei test.

**Soluzione:** coinvolgimento del responsabile a capo del progetto relativo il simulatore hardware.

### 3.3 Requisiti e obiettivi

### 3.4 Pianificazione



## Capitolo 4

# Analisi dei requisiti

*Breve introduzione al capitolo*

### 4.1 Casi d'uso

Per lo studio dei casi di utilizzo del prodotto sono stati creati dei diagrammi. I diagrammi dei casi d'uso (in inglese *Use Case Diagram*) sono diagrammi di tipo uml dedicati alla descrizione delle funzioni o servizi offerti da un sistema, così come sono percepiti e utilizzati dagli attori che interagiscono col sistema stesso. Essendo il progetto finalizzato alla creazione di un tool per l'automazione di un processo, le interazioni da parte dell'utilizzatore devono essere ovviamente ridotte allo stretto necessario. Per questo motivo i diagrammi d'uso risultano semplici e in numero ridotto.

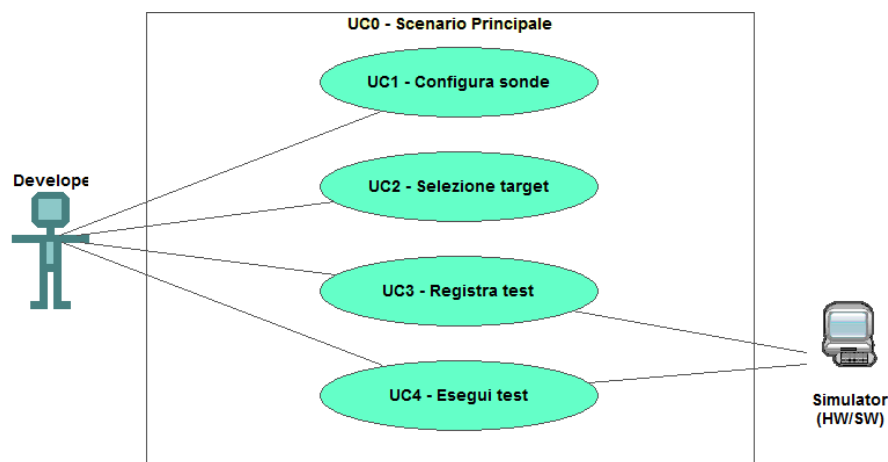


figura 4.1: Use Case - UC0: Scenario principale

#### UC0: Scenario principale

**Attori Principali:** Sviluppatore applicativi.

**Precondizioni:** Lo sviluppatore è entrato nel plug-in di simulazione all'interno dell'I-

DE.

**Descrizione:** La finestra di simulazione mette a disposizione i comandi per configurare, registrare o eseguire un test.

**Postcondizioni:** Il sistema è pronto per permettere una nuova interazione.

## 4.2 Tracciamento dei requisiti

Da un'attenta analisi dei requisiti e degli use case effettuata sul progetto è stata stilata la tabella che traccia i requisiti in rapporto agli use case.

Sono stati individuati diversi tipi di requisiti e si è quindi fatto utilizzo di un codice identificativo per distinguerli.

Il codice dei requisiti è così strutturato  $R(F/Q/V)(N/D/O)$  dove:

R = requisito

F = funzionale

Q = qualitativo

V = di vincolo

N = obbligatorio (necessario)

D = desiderabile

Z = opzionale

Nelle tabelle [4.1](#), [4.2](#) e [4.3](#) sono riassunti i requisiti e il loro tracciamento con gli use case delineati in fase di analisi.

**tabella 4.1:** Tabella del tracciamento dei requisiti funzionali

Requisito	Descrizione	Use Case
RFN-1	L'interfaccia permette di configurare il tipo di sonde del test	UC1

**tabella 4.2:** Tabella del tracciamento dei requisiti qualitativi

Requisito	Descrizione	Use Case
RQD-1	Le prestazioni del simulatore hardware deve garantire la giusta esecuzione dei test e non la generazione di falsi negativi	-

**tabella 4.3:** Tabella del tracciamento dei requisiti di vincolo

Requisito	Descrizione	Use Case
RVO-1	La libreria per l'esecuzione dei test automatici deve essere riutilizzabile	-





## Capitolo 5

# Progettazione e codifica

*Breve introduzione al capitolo*

### 5.1 Tecnologie e strumenti

Di seguito viene data una panoramica delle tecnologie e strumenti utilizzati.

#### **Tecnologia 1**

Descrizione Tecnologia 1.

#### **Tecnologia 2**

Descrizione Tecnologia 2

### 5.2 Ciclo di vita del software

### 5.3 Progettazione

#### **Namespace 1**

Descrizione namespace 1.

**Classe 1:** Descrizione classe 1

**Classe 2:** Descrizione classe 2

### 5.4 Design Pattern utilizzati

### 5.5 Codifica



## Capitolo 6

# Verifica e validazione



## Capitolo 7

# Conclusioni

7.1 Consuntivo finale

7.2 Raggiungimento degli obiettivi

7.3 Conoscenze acquisite

7.4 Valutazione personale



Appendice A

Appendice A

Citazione

---

Autore della citazione





# Glossario

**API** in informatica con il termine *Application Programming Interface API* (ing. interfaccia di programmazione di un'applicazione) si indica ogni insieme di procedure disponibili al programmatore, di solito raggruppate a formare un set di strumenti specifici per l'espletamento di un determinato compito all'interno di un certo programma. La finalità è ottenere un'astrazione, di solito tra l'hardware e il programmatore o tra software a basso e quello ad alto livello semplificando così il lavoro di programmazione. [1](#)

**B2B** Business-to-business, spesso indicato con l'acronimo *B2B*, in italiano commercio interaziendale, è una locuzione utilizzata per descrivere le transazioni commerciali elettroniche tra imprese, distinguendole da quelle che intercorrono tra le imprese e altri gruppi, come quelle oppure quelle tra una impresa e il governo. [1](#)

**B2C** Con Business to Consumer, spesso abbreviato in *B2C*, si indicano le relazioni che un'impresa commerciale detiene con i suoi clienti per le attività di vendita e/o di assistenza. [1](#)

**ICT** Information and communication technology:  
Le tecnologie dell'informazione e della comunicazione (in inglese Information and Communications Technology, in acronimo ICT), sono l'insieme dei metodi e delle tecnologie che realizzano i sistemi di trasmissione, ricezione ed elaborazione di informazioni (tecnologie digitali comprese). [1](#)

**KMS** Knowledge management system:  
I Knowledge management system sono sistemi software che supportano le fasi del ciclo dell'informazione e la comunicazione all'interno di una comunità di pratica (ad esempio un'azienda) o di apprendimento (ad esempio una classe "virtuale") anche disperse nello spazio. Dovrebbero assistere le persone ad esplicitare la conoscenza tacita, a reperirla, a condividerla, supportando in particolare le seguenti funzioni:

- \* Cattura delle competenze collettive
- \* Controllo per realizzare obiettivi comuni
- \* Integrazione delle conoscenze frammentate

. [v](#), [1](#)



# Acronimi

**API** [Application Program Interface.](#) 21

**B2B** [Business-to-business.](#) 21

**B2C** [Business-to-Business.](#) 21

**ICT** [Information and Communication Tecnology.](#) 21

**KMS** [Knowledge Managment System.](#) 21



# Bibliografia

## Riferimenti bibliografici

James P. Womack, Daniel T. Jones. *Lean Thinking, Second Editon*. Simon & Schuster, Inc., 2010.

## Siti Web consultati

*Manifesto Agile*. URL: <http://agilemanifesto.org/iso/it/>.