

Università degli Studi di Padova

DIPARTIMENTO DI MATEMATICA "TULLIO LEVI-CIVITA"

CORSO DI LAUREA IN INFORMATICA



Advertising management system:  
Creazione e gestione di contenuti pubblicitari

*Tesi di laurea*

*Relatore*

Prof. Claudio Enrico Palazzi

*Laureando*

Alessandro Discalzi

---

ANNO ACCADEMICO 2019-2020



# Sommario

Il presente documento descrive il lavoro svolto durante il periodo di stage dal laureando Alessandro Discalzipresso l'azienda SCAI ITEC. Lo stage è stato svolto al termine del percorso di studi della laurea triennale in informatica e la sua durata è stata di 312 ore. L'obiettivo dello stage è stato di implementare un applicativo per la creazione e per la gestione di contenuti pubblicitari. Il presente documento vuole illustrare il contesto aziendale dove si è svolto lo stage, le attività svolte e una valutazione sul lavoro effettuato e su quanto appreso.



“Nessuno ha mai ottenuto nulla con le lacrime.”

— Brucaliffo

# Ringraziamenti

*Innanzitutto, vorrei esprimere la mia gratitudine al Prof. Claudio Enrico Palazzi, relatore della mia tesi, per l'aiuto e il sostegno che mi ha fornito durante la stesura del lavoro.*

*Desidero ringraziare con affetto la mia famiglia, e in particolare i miei genitori per il sostegno e per il grande aiuto che mi hanno dato durante gli anni di studio.*

*Desidero inoltre ringraziare il mio tutor aziendale, Dott. Bledar Gogaj, e il suo collega, Dott. Marco Lionello per l'enorme aiuto che mi hanno dato durante il periodo di stage.*

*Un ringraziamento infine, ai miei amici per tutti i bei momenti passati insieme e per avermi sopportato tutti questi anni.*

*Padova, Settembre 2020*

Alessandro Discalzi



# Indice

<b>1</b>	<b>Introduzione</b>	<b>1</b>
1.1	L'azienda . . . . .	1
1.2	Scopo dello stage . . . . .	2
1.3	Tecnologie utilizzate . . . . .	2
1.3.1	JHipster . . . . .	2
1.3.2	Java Enterprise . . . . .	3
1.3.3	Spring . . . . .	3
1.3.4	Hibernate . . . . .	4
1.3.5	REST <sup>1</sup> . . . . .	4
1.3.6	Oracle . . . . .	5
1.3.7	Liquibase . . . . .	6
1.3.8	Angular . . . . .	6
1.4	Strumenti di sviluppo . . . . .	7
1.4.1	Eclipse . . . . .	7
1.4.2	Maven . . . . .	7
1.4.3	Git . . . . .	7
1.4.4	SQL Developer . . . . .	8
1.5	Organizzazione del testo . . . . .	8
<b>2</b>	<b>Obbiettivi e pianificazione</b>	<b>9</b>
2.1	Obbiettivi . . . . .	9
2.1.1	Obbiettivi obbligatori . . . . .	9
2.1.2	Obbiettivi desiderabili . . . . .	9
2.1.3	Obbiettivi opzionali . . . . .	9
2.2	Pianificazione . . . . .	10
<b>3</b>	<b>Metodologia di sviluppo e composizione del team</b>	<b>11</b>
<b>4</b>	<b>Prodotto software</b>	<b>13</b>
<b>5</b>	<b>Documentazione e Test</b>	<b>15</b>
<b>6</b>	<b>Considerazioni finali</b>	<b>17</b>
	<b>Bibliografia</b>	<b>21</b>

---

<sup>1</sup>REST: acronimo di Representational State Transfer.

# Elenco delle figure

1.1	Logo SCAI ITEC . . . . .	1
1.2	Logo JHipster . . . . .	3
1.3	Logo Java EE . . . . .	3
1.4	Logo Spring . . . . .	4
1.5	Logo Hibernate . . . . .	4
1.6	Logo REST . . . . .	5
1.7	Logo Oracle . . . . .	5
1.8	Logo Liquibase . . . . .	6
1.9	Logo Angular . . . . .	6
1.10	Logo Eclipse . . . . .	7
1.11	Logo Maven . . . . .	7
1.12	Logo Git . . . . .	7
1.13	Logo SQL Developer . . . . .	8

# Elenco delle tabelle



# Capitolo 1

## Introduzione

### 1.1 L'azienda

**SCAI ITEC**<sup>1</sup> è un'azienda italiana appartenente al **gruppo SCAI**. ITEC si occupa di consulenza, System Integration ed Application management in ambito ICT. L'azienda opera principalmente in settore bancario, assicurativo, industriale e di pubblica amministrazione e servizi.

Gli elementi chiave del successo e della crescita di SCAI ITEC sono:

- vasta e profonda conoscenza delle tecnologie;
- grande attenzione per la soddisfazione del cliente;
- molta esperienza, maturata nel corso del tempo.

ITEC è oggi una delle maggiori realtà nel nord-est del paese in ambito ICT e si propone come partner per qualsiasi tipo di applicazione, progetto e servizio modellato sulle specifiche necessità del cliente.

Grazie alla consolidata esperienza nel ruolo di **System Integrator**<sup>[g]</sup> ed alle soluzioni leader di mercato proposte, ITEC è in grado di garantire ai propri clienti risposte rapide, concrete e qualificate in base alle specifiche esigenze di tipo gestionale e applicativo.

Ultimo, ma non meno importante, dei motivi per cui l'azienda è all'avanguardia rispetto le nuove tecnologie è il grande investimento di ITEC in ricerca, sviluppo e formazione del personale.



**Figura 1.1:** Logo SCAI ITEC

---

<sup>1</sup>SCAI ITEC abbrev. ITEC.

## 1.2 Scopo dello stage

L'obiettivo principale dello stage è stato quello di inserire lo studente all'interno di una nuova progettualità, con un particolare focus sulle tematiche legate alle tecnologie multimediali e alla loro distribuzione. Lo studente, affiancato da un IT Architect, ha avuto la possibilità di partecipare al disegno, alla progettazione e realizzazione di una nuova progettualità. L'obiettivo è stato apprendere le tecnologie e le best practice utilizzate in azienda nel ciclo di vita di un applicativo. La progettualità vista è volta a creare un software per la gestione di [contenuti informativi](#)<sup>[8]</sup>, per la loro creazione, modifica e distribuzione nei vari canali di vendita. Grazie a questa nuova applicazione si potrà:

- creare, modificare eliminare e clonare dei contenuti informativi;
- raggruppare i contenuti informativi su segmenti di mercato e distribuirli;
- pianificare l'esecuzione e l'aggiornamento dei contenuti informativi sui vari canali di distribuzione;
- seguire un processo di Authoring (paradigma Editore, Redattore, Supervisore) nella fase di creazione e distribuzione.

In quanto l'intero progetto è di dimensione non indifferente l'obiettivo dello stage è stato di sviluppare le funzionalità relative al ruolo di Editore, più nello specifico:

- creazione di un contenuto;
- modifica di un contenuto;
- eliminazione di un contenuto;
- preparazione di un contenuto per la distribuzione;
- auditing delle azioni effettuate dagli utenti;
- documentazione delle funzionalità implementate.

## 1.3 Tecnologie utilizzate

### 1.3.1 JHipster

**JHipster** è una piattaforma di sviluppo, con uno stack tecnologico ben definito, utilizzata per generare, sviluppare e rilasciare, applicazioni e web services all'avanguardia. Supporta molteplici tecnologie per il frontend, tra le quali Angular, React e Vue. Fornisce inoltre supporto per le applicazioni per dispositivi mobili utilizzando Ionic e React Native. Per quanto riguarda il backend, JHipster supporta spring Boot (con l'ausilio di Java o Kotlin), Micronaut, Quarkus, NodeJS e .NET. Per il rilascio sono adottati i principi di [Cloud nativo](#)<sup>[8]</sup>. Il rilascio è inoltre supportato su AWS, Azure, Cloud Foundry, Google Cloud Platform, Heroku ed OpenShift.

L'obiettivo di JHipster è generare applicazioni web o microservizi all'avanguardia, unendo:

- uno stack lato server robusto, ad alte prestazioni e coperto da test;

- un interfaccia utente accattivante, moderna e mobile-first usando Angular, React o Vue e Bootstrap per il CSS;
- un workflow ben definito per fare la build dell'applicazione con Maven o Gradle;
- un architettura a microservizi resiliente, utilizzando i principi di [Cloud nativo](#);
- infrastruttura definita come codice, in modo da rendere la distribuzione su cloud veloce.

Nel corso dello stage JHipster è stato utilizzato per generare l'applicazione di base, utilizzata come punto di partenza per lo sviluppo.



**Figura 1.2:** Logo JHipster

### 1.3.2 Java Enterprise

**Java Enterprise**, conosciuto anche come Java EE è un insieme di specifiche che mirano ad estendere Java 8, aggiungendo funzionalità enterprise come elaborazione distribuita e servizi web. Le applicazioni Java EE possono essere eseguite sia come [Microservizi](#)<sup>[g]</sup> che su [Application server](#)<sup>[g]</sup>. In entrambi i casi vengono gestite: transazionalità, scalabilità, sicurezza e concorrenza. Nel corso dello stage Java EE è stato utilizzato per la programmazione lato backend.



**Figura 1.3:** Logo Java EE

### 1.3.3 Spring

**Spring** è un framework applicativo open source e un container per l'[inversione di controllo](#)<sup>[g]</sup> utilizzato dalla piattaforma Java. Le funzionalità di base possono essere

usate da una qualsiasi applicazione Java, mentre quelle più avanzate sono disponibili solamente per Java Enterprise. Il framework Spring ha a se associati vari moduli, nel corso del progetto sono stati utilizzati principalmente:

- Spring Boot: utilizzato per creare applicazioni basate su spring eseguibili senza la necessità di configurare un web server;
- Spring Data: il cui obiettivo è di facilitare la gestione e l'interazione di applicazioni Java con un database.



**Figura 1.4:** Logo Spring

### 1.3.4 Hibernate

**Hibernate** è un framework per lo sviluppo di applicazioni in Java, utilizzato per gestire e mantenere su un database relazionale un insieme di oggetti Java. Le sue funzionalità principali sono:

- mappare oggetti Java come tabelle su database;
- convertire i campi dati Java a quelli del **DBMS**<sup>[g]</sup> utilizzato;
- generare chiamate SQL e convertire la risposta ottenuta in un oggetto Java.

Tali funzionalità sono state largamente utilizzate nel corso dello stage.



**Figura 1.5:** Logo Hibernate

### 1.3.5 REST<sup>2</sup>

**REST** è un modello architetturale per i sistemi distribuiti. I sistemi rest si basano su HTTP e prevedono una struttura degli URL ben definita, che identifichi univocamente le risorse secondo la convenzione del modello stesso<sup>3</sup>. In REST per il trasferimento di dati vengono utilizzati i metodi HTTP, più nello specifico:

- GET: per il recupero di informazioni;
- POST, PUT, PATCH: per l'inserimento di informazioni;
- DELETE: per l'eliminazione di informazioni.

---

<sup>2</sup>**REST**: acronimo di **Representational State Transfer**.

<sup>3</sup>**Resource Naming**: <https://restfulapi.net/resource-naming/>.

I principi guida di REST sono:

- client-server: separazione dei problemi della UI da quelli di storage dei dati;
- statelessness: ogni richiesta deve avere tutte le informazioni necessarie per il suo processamento;
- cacheable: i client possono memorizzare in cache le risposte, queste devono essere definite esplicitamente o implicitamente cacheable, in modo da evitare il riutilizzo di dati errati;
- uniform interface: utilizzo di un'interfaccia di comunicazione omogenea tra client e server in modo da disaccoppiare e semplificare l'architettura per poterla modificare a blocchi;
- layered system: la struttura del sistema può essere composta da strati gerarchici. In questo caso ogni componente non può "vedere" oltre lo strato con cui sta interagendo;
- code on demand (opzionale): il codice lato client può essere esteso scaricando ed eseguendo applet o script.

Nella definizione di API se queste rispettano tutti i vincoli imposti dall'architettura REST allora possono essere definite RESTful.



**Figura 1.6:** Logo REST

### 1.3.6 Oracle

Oracle database è un DBMS di tipo relazionale prodotto da Oracle corporation.

I database oracle sono noti per offrire performance, scalabilità, affidabilità e sicurezza oltre a poter essere utilizzati sia on premise che nel cloud.



**Figura 1.7:** Logo Oracle

### 1.3.7 Liquibase

**Liquibase** è una libreria open source indipendente dal **DBMS** utilizzato. Durante il periodo di stage è stata utilizzata per tracciare, gestire e applicare le modifiche allo schema del database.



**Figura 1.8:** Logo Liquibase

### 1.3.8 Angular

**Angular** è un framework open source per la progettazione e lo sviluppo di applicazioni web. Le applicazioni angular vengono eseguite interamente a lato client ma grazie alla moltitudine di moduli presenti è possibile integrare un sistema di backend più complesso eseguito lato server. Nel corso dello stage, il frontend, è stato scritto interamente in Angular.



**Figura 1.9:** Logo Angular

## 1.4 Strumenti di sviluppo

### 1.4.1 Eclipse

L'IDE utilizzato per lo sviluppo, previo consiglio del tutor aziendale, è stato **Eclipse**. Eclipse è un ambiente di sviluppo integrato multiplatforma e rientra nella categoria di software libero, distribuito secondo i termini della **Eclipse Public License**.



**Figura 1.10:** Logo Eclipse

### 1.4.2 Maven

**Maven** è uno strumento di gestione di progetti software, è basato su un Project Object Model (POM) e può gestire la build, il reporting e la documentazione di un progetto. Nel corso dello stage Maven è stato utilizzato principalmente per automatizzare la build del progetto, sia in sviluppo che in produzione.



**Figura 1.11:** Logo Maven

### 1.4.3 Git

**Git** è un sistema di controllo di versione distribuito, gratuito ed open source, progettato per gestire progetti di qualsiasi tipo. Nel corso dello stage l'utilizzo di Git è stato affiancato a quello di GitLab, una piattaforma web per la gestione di repository Git.



**Figura 1.12:** Logo Git

#### 1.4.4 SQL Developer

**SQL Developer** è un ambiente di sviluppo integrato per lavorare con SQL nei database Oracle. Nel corso dello stage è stato utilizzato per gestire e testare il database Oracle usato in produzione.



**Figura 1.13:** Logo SQL Developer

### 1.5 Organizzazione del testo

**Il secondo capitolo** descrive gli obbiettivi dello stage, la pianificazione del lavoro effettuata a monte e le aspettative personali riguardanti lo stage;

**Il terzo capitolo** approfondisce la metodologia di lavoro adottata e i ruoli adottati dal team di sviluppo;

**Il quarto capitolo** descrive dettagliatamente le funzionalità del software prodotto e le soluzioni adottate durante la codifica;

**Il quinto capitolo** descrive la documentazione prodotta e i test eseguiti;

**Il sesto capitolo** contiene le considerazioni finali riguardanti lo stage e una valutazione personale sul lavoro svolto;

Riguardo la stesura del testo, relativamente al documento sono state adottate le seguenti convenzioni tipografiche:

- gli acronimi, le abbreviazioni e i termini ambigui o di uso non comune menzionati vengono definiti nel glossario, situato alla fine del presente documento;
- per la prima occorrenza dei termini riportati nel glossario viene utilizzata la seguente nomenclatura: *parola*<sup>[g]</sup>;
- i termini in lingua straniera o facenti parti del gergo tecnico sono evidenziati con il carattere *corsivo*.



## Capitolo 2

# Obbiettivi e pianificazione

*In questo capitolo sono descritti gli obiettivi dello stage, la pianificazione del lavoro e le aspettative personali.*

### 2.1 Obbiettivi

L'obiettivo di questo stage è la realizzazione di una Proof of Concept dell'applicazione, in cui vengono rese disponibili le funzionalità descritte in §1.2. Per aumentare la produttività e permettere allo studente di conoscere nuove tecnologie viene utilizzato JHipster. Lo studente verrà inserito in un gruppo di lavoro composto da 4 persone in modo da favorire, oltre alla comprensione delle nuove tecnologie, la capacità di lavorare in un team. Al termine del periodo di stage verrà effettuata una presentazione del prodotto alla direzione dell'azienda.

Nel piano di lavoro, documento la cui stesura è avvenuta prima dell'inizio dello stage, sono stati individuati i seguenti obiettivi suddivisi in obbligatori, desiderabili e opzionali.

#### 2.1.1 Obbiettivi obbligatori

- Ob1: Conoscenza del framework Spring e in particolare Spring MVC REST;
- Ob2: Interazione e gestione database Oracle;
- Ob3: Realizzazione delle funzionalità backend del progetto.

#### 2.1.2 Obbiettivi desiderabili

- D1: Realizzazione delle funzionalità frontend del progetto;
- D2: Conoscenza base sviluppo applicazioni frontend Angular;
- D3: Grado di autonomia nel processo di analisi/sviluppo.

#### 2.1.3 Obbiettivi opzionali

- Op1: Conoscenza base dei strumenti per CI/CD.

## 2.2 Pianificazione

Lo stage prevede una durata di 312 ore complessive corrispondenti a 8 ore di lavoro giornaliero per un periodo di circa 8 settimane. L'orario di lavoro è dal Lunedì al Venerdì, dalle ore 9:00 alle 13:00 e dalle ore 14:00 alle 18:00. L'ora tra le 13:00 e le 14:00 è dedicata alla pausa pranzo.

La pianificazione redatta per il periodo di stage, divisa per obiettivi settimanali, è la seguente:

- **prima settimana:**
  - studio strumenti di sviluppo (Eclipse, Maven, Git);
  - analisi dei requisiti.
- **seconda settimana:**
  - creazione struttura del database;
  - gestione database Oracle.
- **terza settimana:**
  - studio e utilizzo del framework Spring;
  - studio e utilizzo di Hibernate;
  - realizzazione dell'object-relational mapping.
- **quarta settimana:**
  - utilizzo spring MVC e Jackson;
  - realizzazione dei servizi REST.
- **quinta settimana:**
  - studio e utilizzo di elementi avanzati di Oracle;
  - gestione changeset Liquibase;
  - sviluppo di altri servizi REST.
- **sesta settimana:**
  - studio di Angular;
  - sviluppo frontend;
- **settima settimana:**
  - continuous integration e continuous delivery;
  - utilizzo di Sonarqube;
  - controllo qualità del codice.
- **ottava settimana:**
  - gestione cache dell'applicazione;
  - ottimizzazione;

## Capitolo 3

# Metodologia di sviluppo e composizione del team

*Introduzione*



## Capitolo 4

# Prodotto software

*Introduzione*



## Capitolo 5

# Documentazione e Test

*Introduzione*





## Capitolo 6

# Considerazioni finali

*Introduzione*







# Bibliografia