

Università degli Studi di Padova

DIPARTIMENTO DI MATEMATICA "TULLIO LEVI-CIVITA "

CORSO DI LAUREA IN INFORMATICA



**Test di carico in ambiente enterprise: un
effettivo caso di studio**

Tesi di laurea triennale

Relatore

Prof. Francesco Ranzato

Laureando

Gianluca Marraffa

ANNO ACCADEMICO 2017-2018

Sommario

Il presente documento descrive il lavoro svolto durante il periodo di stage, della durata trecentoventi ore, dal laureando Gianluca Marraffa presso l'azienda Infocert S.p.A.

Durante il periodo di tirocinio il laureando è stato inserito all'interno del team Legalmail, responsabile dello sviluppo del prodotto PEC Legalmail dell'azienda.

Lo scopo ultimo dello stage si proponeva come la progettazione e l'installazione di una infrastruttura per test di carico capace di evidenziare e documentare le capacità e i limiti degli applicativi aziendali. In primo luogo si chiedeva un'analisi dello stato dell'arte dei vari strumenti di Load Testing presenti sul mercato volta a sottolineare pro e contro delle varie soluzioni, successivamente si sarebbe scelta la soluzione più adatta alle caratteristiche aziendali e del prodotto in esame.

In secondo luogo in secondo veniva richiesta l'effettiva implementazione dell'infrastruttura concordata, e l'esecuzione di test di carico negli ambienti di test e accettazione.

Per ultimo veniva richiesta la stesura di un documento che esponesse le caratteristiche dell'infrastruttura implementata in modo da estendere il lavoro effettuato, e di un manuale d'utilizzo per il team Legalmail.

“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, Dicembre 2018

Gianluca Marraffa

Indice

1	Il Contesto Aziendale	1
1.1	Profilo Aziendale	1
1.2	Legalmail	2
1.3	Tecnologie Utilizzate	2
1.3.1	Dev	2
1.3.2	Ops	3
1.4	Metodologia Aziendale	4
1.4.1	Metodologia SCRUM	4
1.4.2	Infrastruttura e Applicazioni	5
1.5	Strumenti Utilizzati	6
1.5.1	Jira	6
1.5.2	Confluence	6
1.5.3	Gitlab	6
1.5.4	Jenkins	6
1.5.5	Artifactory	7
1.5.6	Puppet Enterprise	7
1.5.7	Foreman	7
1.5.8	Oracle Secure Global Desktop	7
1.5.9	Ambienti di sviluppo	7
1.6	Propensione all'innovazione	8
2	Interessi e Aspettative	9
2.1	Progetto Test di Carico	9
2.1.1	Descrizione	9
2.1.2	Obiettivi	10
2.1.3	Pianificazione Temporale	10
2.1.4	Prodotti Attesi	10
2.1.5	Vincoli Metodologici	10
2.1.6	Vincoli Tecnologici	10
2.2	Interessi Aziendali	10
2.2.1	Stage nella strategia aziendale	10
2.2.2	I test di carico nel ciclo di vita del software	10
2.3	Aspettative Personali	11
3	Load Test Framework	13
3.1	Analisi delle soluzioni esistenti	13
3.1.1	Strumenti di test di carico	13
3.1.2	Infrastructure as Code	13

3.1.3	Cloud Platform	13
3.1.4	Provisioner	13
3.1.5	Orchestrator	13
3.1.6	La soluzione proposta	13
3.2	Progettazione dell'orchestratore	13
3.2.1	Lo scontro con l'infrastruttura	13
3.2.2	Core	13
3.2.3	Command Line Interface	13
3.2.4	Enterprise Edition	13
3.3	Progettazione dei test	13
3.3.1	Analisi delle capacità dei server	13
3.3.2	Template di scenario	13
3.4	Esecuzione dei test	13
3.5	Documentazione prodotta	13
4	Valutazioni finali	15
4.1	Soddisfacimento Obiettivi	15
4.2	Maturazione Professionale	15
4.3	Considerazioni Personali	15
	Glossario	17
	Bibliografia	19

Elenco delle figure

1.1	Logo di Infocert S.p.A	1
-----	----------------------------------	---

Elenco delle tabelle

Capitolo 1

Il Contesto Aziendale

1.1 Profilo Aziendale

InfoCert è leader del mercato italiano nei servizi di digitalizzazione e dematerializzazione nonché una delle principali Certification Authority a livello europeo per i servizi di Posta Elettronica Certificata, Firma Digitale e Conservazione digitale dei documenti (Conservatore Accreditato [AgID](#)).

Da dicembre 2015 InfoCert è anche gestore accreditato AgID dell'identità digitale di cittadini e imprese, in conformità ai requisiti regolamentari e tecnici dello [SPID](#).

InfoCert si pone sul mercato come un partner altamente specializzato nei servizi di dematerializzazione, capace di garantire ai propri clienti la piena innovazione nei processi di gestione del patrimonio documentale e informativo.

Con sedi a Roma, Milano e Padova, InfoCert rivolge la propria offerta sia alle imprese, pubbliche e private, operanti nel settore Bancario, Assicurativo, Farmaceutico, Manifatturiero, Energy, Utilities, Distribuzione Commerciale, Ambiente, Qualità, Sicurezza, Sanità, Pubblica Amministrazione; sia ad Associazioni di Categoria, Ordini Professionali e Professionisti.



Figura 1.1: Logo di Infocert S.p.A

1.2 Legalmail

La Posta Elettronica Certificata (detta anche posta certificata o PEC) Legalmail di InfoCert è un sistema di comunicazione simile alla posta elettronica standard con in più alcune caratteristiche di sicurezza e di certificazione della trasmissione che rendono i messaggi opponibili a terzi.

La PEC Legalmail consente infatti di inviare/ricevere messaggi di testo e allegati con lo stesso valore legale di una raccomandata con avviso di ricevimento.

Dal punto di vista tecnologico, il prodotto si presenta come un'applicazione web scomposta in diversi [microservizi](#), ognuno responsabile di gestire una particolare funzionalità applicativa. La maggior parte di questi servizi è suddiviso in backend (la logica applicativa) e frontend (l'interfaccia utente) e comunicano con gli altri microservizi attraverso REST API.

Le varie componenti sono tante e non avrebbe senso elencarle tutte, per il progetto di tirocinio le parti da me interessate sono state quelle del backend della dashboard informativa e la gestione delle code di priorità per errori e segnalazioni.

1.3 Tecnologie Utilizzate

Pur essendo molto collaborativi, sviluppatori e operatori utilizzano tecnologie diverse per soddisfare i requisiti richiesti. I primi infatti usano principalmente linguaggi di programmazione per sviluppare le nuove funzionalità, mentre i secondi adoperano anche linguaggi dichiarativi per configurare i vari ambienti di esecuzione.

1.3.1 Dev

Gli sviluppatori seguono la classica divisione in backend developers, responsabili della logica di business dell'applicativo, e frontend developers, responsabili dell'interfacciamento tra utente e applicazione.

Java 8 Enterprise Edition

Lo standard scelto per lo sviluppo della business logic dei vari applicativi è Java¹, linguaggio di programmazione orientato agli oggetti e spesso presenza indiscussa nella maggior parte degli ambienti enterprise per una moltitudine di obiettivi, tra i principali:

- * **Maturità:** Oracle² rilascia la prima versione di Java nel 1995 e conquista col tempo la maggior parte dei software sviluppati, rivelandosi fin da subito un componente robusto e sicuro per lo sviluppo di applicazioni. Le aziende di grosso taglio si sentono quindi al sicuro nell'adottare Java, assicurandosi un prodotto ben consolidato;
- * **Curva d'apprendimento:** nei suoi principi base, Java è un linguaggio molto semplice e, essendo i team di aziende enterprise composti da molte persone, è molto facile formare nuovi sviluppatori all'utilizzo della tecnologia. Inoltre la sua diffusione ha portato negli anni alla creazione di una moltitudine di corsi ben strutturati, conquistando spazi anche nelle università;

¹site:java.

²site:oracle.

- * **Funzionalità:** essendo la community di sviluppatori Java molto diffusa, il mondo open source è pieno di librerie che risolvono problemi comuni a molti casi d'uso, velocizzando i tempi di sviluppo e riducendo l'introduzione di nuovi errori;
- * **Multiplatforma:** un applicativo Java viene scritto una sola volta e, con pochi accorgimenti, può essere distribuito su moltissime piattaforme, permettendo all'azienda di distribuire i propri prodotti senza incappare in "barriere architettoniche";
- * **Scalabilità:** i prodotti di stampo enterprise sono utilizzati da parecchie persone, per questo l'infrastruttura che li ospita deve essere flessibile ad aumenti o diminuzioni di traffico. Java, grazie alle numerose integrazioni esistenti, si presenta come un ottimo candidato per queste necessità.

L'azienda ha sviluppato su di esso un [framework](#) proprietario chiamato BEcon.

Angular JS

Per la parte frontend dei vari prodotti, Infocert ha sviluppato un [framework](#) interno, FEcon, basato su AngularJS³: un [framework](#) strutturale per la creazione di applicazioni web dinamiche.

1.3.2 Ops

Gli operatori non hanno distinzioni interne e utilizzano Puppet per la configurazione delle macchine e Java e/o Python per la realizzazione dei test d'accettazione e sviluppo delle sonde di monitoraggio.

Le sonde di monitoraggio sono piccoli applicativi che verificano lo stato dei server nei vari ambienti d'esecuzione; in caso di macchine non operative avvisano i relativi responsabili.

Puppet

Puppet è uno strumento di [configuration management](#) che permette di definire lo stato delle macchine tramite un linguaggio dichiarativo dedicato, garantendo il versionamento dell'infrastruttura e la replica delle configurazioni su più macchine con il minimo sforzo. Operando le modifiche ai server esclusivamente tramite Puppet è infatti possibile mantenere consistente lo stato dell'infrastruttura, rendendo più rapida la propagazione delle modifiche in flotte composte da molti server.

Un altro punto di forza di Puppet è quello di definire delle classi tramite il linguaggio proprietario dell'applicativo, permettendo ai meno esperti di utilizzarle senza conoscerne l'implementazione interna (proprio come una classe vera e propria, con metodi privati e pubblici) e rendendo più accessibile la gestione della configurazione. Queste classi infatti possono essere configurate tramite file YAML, rendendone possibile l'utilizzo anche a chi non è formato per utilizzare il linguaggio interno di Puppet.

Python

Python è un linguaggio di programmazione non tipizzato che da tempo sta conquistando sempre più aree dello sviluppo software. Grazie alla sua bassa curva d'apprendimento e alle innumerevoli librerie open-source presenti sul mercato, risulta la scelta ideale

³site:angular.

per molte soluzioni software, tra le quali troviamo le utility di scripting, il calcolo distribuito, il data mining e lo sviluppo dei test di sistema.

La forza di python sta nel permettere la realizzazione di compiti complessi tramite la scrittura poche righe di codice, mantenendo alta la leggibilità dello stesso.

1.4 Metodologia Aziendale

1.4.1 Metodologia SCRUM

La metodologia utilizzata dal team Legalmail (e dalla maggior parte dei restanti team in Infocert) è quella Agile, implementata tramite il [framework](#) SCRUM. Questa metodologia impone un'approccio semplice ma efficace per la gestione del ciclo di vita del software e fa leva sulla capacità auto organizzativa del team di sviluppo.

In un team SCRUM infatti sono presenti solo 3 categorie di elementi e sono pensate per permettere agli sviluppatori di concentrarsi principalmente sul lavoro da effettuare piuttosto che all'interazione con gli [stakeholders](#).

- * **Product Owner:** il ruolo del Product Owner è cruciale in quanto rappresenta il ponte tra team di sviluppo e stakeholders. Il PO ha infatti il compito di convertire i requisiti del cliente in items usabili dal team di sviluppo, oltre ha comunicare eventuali variazioni delle tempistiche agli stakeholders;
- * **Scrum Master:** il ruolo dello Scrum Master è essenziale per permettere il corretto svolgimento dello sprint, questo infatti s'impegna a rimuovere ostacoli nello sviluppo (come improvvisa mancanza di personale, assegnamento di maggiori risorse su un particolare compito etc.) e a variare gli obiettivi fissati nello sprint planning in base al feedback offerto dagli sviluppatori;
- * **Team di Sviluppo:** Il team di sviluppo rappresenta tutti i membri responsabili delle varie fasi del ciclo di vita del software ed è completamente autonomo: al suo interno ci sono tutte le competenze necessarie per iniziare e finire il progetto. I componenti del gruppo sono in grado di organizzarsi da soli e assegnarsi le attività che ritengono più appropriate in modo completamente indipendente. Avendo l'azienda abbracciato la filosofia [DevOps](#) i team di sviluppo si dividono internamente in:
 - **Dev:** responsabili di progettazione e sviluppo delle funzionalità applicative;
 - **Ops:** responsabili della fase di testing e distribuzione delle nuove [release](#), monitoraggio della produzione, verifica e rispetto degli [SLA](#) contrattualizzati, assolvimento delle procedure normative e automazione dei compiti ripetitivi.

I momenti formali che prevede questa metodologia sono chiamati eventi e sono *Sprint*, *Sprint Planning*, *Daily Scrum*, *Sprint Review* e *Sprint retrospective*.

Gli eventi a cui ho avuto modo di partecipare attivamente sono stati lo Scrum Daily e lo Sprint. Il primo è una breve riunione di massimo 15 minuti effettuata nel primo mattino atta ad esporre ai colleghi le attività lavorative svolte nella precedente giornata e ad organizzare le prossime 24 ore, durante la riunione vengono esposti anche eventuali problemi sorti durante lo sviluppo in modo da aggiornare i colleghi sullo stato del lavoro che si sta svolgendo. Durante questa breve riunione viene aggiornata la SCRUM board con le informazioni rilevate, permettendo allo SCRUM master di adeguare la visione

generale dello stato del progetto. La SCRUM board è una lavagna che tiene traccia dei task che i Dev e gli Ops devono svolgere durante lo sprint e permettono allo Scrum master di avere una buona visione d'insieme del suo andamento. La struttura della SCRUM board è a discrezione delle esigenze del team, per quanto riguarda Legalmail ogni riga rappresenta un componente del team mentre le colonne sono divise in:

- * **Todo:** I compiti assegnati al componente ma non ancora presi in carico dallo stesso;
- * **Progress:** I compiti assegnati che il componente ha iniziato a svolgere;
- * **Impediment:** I compiti che non possono proseguire perché bloccati da impedimenti al di fuori della responsabilità del componente;
- * **Developed:** I compiti sviluppati dal componente che devono essere verificati;
- * **Testing:** I compiti in fase di verifica;
- * **Failed:** I compiti che non hanno soddisfatto i requisiti richiesti;
- * **Done:** I compiti che sono terminati ed hanno ricevuto l'approvazione da parte dello SCRUM master.

Dopo poche partecipazioni ho assorbito l'importanza di questo momento: il fatto di esporre a tutto il team il problema riscontrato permette di giungere più facilmente ad una soluzione. Può essere infatti che un membro esterno alla vicenda (il compito da svolgere) si sia imbattuto in un problema simile nel passato e abbia trovato una soluzione, ma il componente colpito dal problema non essendone a conoscenza non può confrontarsi con il diretto interessato. Grazie a questi 15 minuti di condivisione invece, il collega può prendere visione del problema e proporre la sua soluzione, permettendo al team di risparmiare tempo e procedere velocemente verso la conclusione del task assegnato. Lo sprint è invece l'effettiva finestra temporale nella quale si prefissano degli obiettivi che dovranno essere completati alla fine della stessa. All'interno del team Legalmail la durata di uno sprint è stata fissata a 14 giorni (10 giorni lavorativi). Ogni informazione aggiuntiva può essere ricavata visitando il sito ufficiale della metodologia SCRUM.

1.4.2 Infrastruttura e Applicazioni

Infocert è un'azienda che sviluppa prodotti di stampo Enterprise: soluzioni dalla grande distribuzione rivolte ad aziende molto grandi e importanti. Per garantire qualità in termini di prodotto ai suoi clienti, l'azienda ha negli anni maturato diverse certificazioni necessarie per lo sviluppo di certi tipi di applicativi, queste certificazioni si portano l'onere di rispettare imposizioni particolari, imponendo una determinata [way of working](#) aziendale. I server utilizzati per conservare determinati tipi di dati, ad esempio, devono rispettare caratteristiche di sicurezza non banali, impedendo ai team dei vari prodotti di gestire in totale libertà la loro infrastruttura.

L'azienda ha quindi predisposto un team, *Infrastructure* per l'appunto, dedito alla definizione e implementazione di standard per la creazione delle macchine virtuali utilizzate per ospitare le applicazioni, lasciando come unica libertà agli *ops* quella di configurare, nei limiti definiti, le macchine richieste.

Questa specificazione non è atta in alcun modo a screditare le modalità operative aziendali, ma è stata introdotta per giustificare alcune scelte adoperate per portare

a termine il progetto di stage che, come descritto nei capitoli successivi, ha dovuto accettare dei compromessi per essere integrato nell'infrastruttura aziendale.

1.5 Strumenti Utilizzati

1.5.1 Jira

Jira⁴ è uno strumento rilasciato da Atlassian per gestire tutta la componente organizzativa del ciclo di vita del software, è ideato per supportare i modelli agili ed è lo strumento più utilizzato dalle aziende che ne adottano le metodologie.

Offre funzionalità di *Project Managment*, *Issue Tracking*, *Agile Reporting* e *Roadmap Planning*. All'interno dell'azienda viene ad esempio utilizzato per la gestione delle SCRUM board e del release planning, oltre che a tutti gli aspetti della gestione di progetto.

1.5.2 Confluence

Confluence⁵ è uno strumento rilasciato sempre da Atlassian per la gestione della parte documentale non per forza legata alla documentazione del software: descrizioni di procedure, guide per installazioni, inventario indirizzi macchine virtuali e tutte le informazioni necessarie per distribuire la conoscenza aziendale a tutti i dipendenti.

L'azienda ha adottato Confluence per racchiudere in un punto centrale tutte queste informazioni, permettendo anche di commentare i documenti scritti da altre persone in modo da migliorare attivamente la qualità della documentazione.

Una parte del mio progetto consisteva nello scrivere una guida su Confluence per l'istanziamento dell'infrastruttura di test ed il lancio dello stesso.

1.5.3 Gitlab

Il versionamento degli applicativi viene fatto tramite Git⁶ e i repository sono ospitati su Gitlab⁷: una piattaforma che permette la gestione centralizzata dei repository Git, permettendo l'amministrazione dei permessi d'accesso tramite una semplice interfaccia grafica. Tutti gli applicativi e le configurazioni dell'infrastruttura vengono ospitati sulla piattaforma e il versionamento segue le regole del Git-Flow: le funzionalità nuove vengono sviluppate su un branch dedicato, per poi essere spostate successivamente nei [branch](#) di sviluppo, accettazione e infine produzione.

Il rispetto del Git-Flow non ha solo scopo formale: la definizione corretta dei branch scatena tutto un processo che dall'esecuzione dei test effettua il [deploy](#) dell'applicazione su Artifactory.

Questo procedimento prende il nome di [Integrazione Continua](#) e viene gestito da Jenkins.

1.5.4 Jenkins

Jenkins⁸ è uno strumento open source sviluppato dalla Jenkins CI community che permette l'implementazione di sistemi di [Integrazione Continua](#) su web-server Java.

⁴[site:jira](#).

⁵[site:confluence](#).

⁶[site:git](#).

⁷[site:gitlab](#).

⁸[site:jenkins](#).

Jenkins è molto flessibile, permette la sua esecuzione in modo automatizzato e/o manuale ed è predisposto per essere esteso tramite plugin scritti dalla community. All'interno dell'azienda la sua esecuzione è innescata dai commit su gitlab e, in base al [branch](#) di appartenenza del commit, scatena una serie di operazioni volute ad automatizzare la verifica, validazione e distribuzione su Artifactory del software.

1.5.5 Artifactory

Artifactory⁹ è uno strumento di repository per pacchetti applicativi (binaries) sviluppato da JFrog, questo permette il versionamento delle [build](#) e una conseguente diminuzione dei tempi di [deploy](#) dell'applicativo. All'interno dell'azienda, Jenkins effettua la [build](#) dei codici sorgenti e li carica su Artifactory; questi pacchetti, stabili e versionati, vengono poi indirizzati dai sistemi di [configuration management](#) per essere installati sui server operativi.

1.5.6 Puppet Enterprise

Puppet Enterprise¹⁰ è la versione Enterprise dello strumento di [configuration management](#) Puppet, questa piattaforma offre una semplice gestione della configurazione per grandi flotte di server in modo completamente automatizzato e parallelo, con la possibilità di gestire la propagazione delle modifiche tramite un'intuitiva interfaccia grafica.

All'interno dell'azienda lo strumento viene utilizzato per applicare le modifiche ai server in modo rapido e affidabile.

1.5.7 Foreman

Foreman¹¹ è uno strumento per la completa gestione del ciclo di vita delle macchine fisiche e/o virtuali. Si presenta come un'interfaccia grafica e permette l'approvvigionamento dei server sia su [datacenter](#) privati che soluzioni nel cloud.

All'interno dell'azienda il team infrastrutture crea dei modelli di macchine interfacciandosi al [datacenter](#) privato, mentre i team di sviluppo personalizzano questi modelli per creare le macchine necessarie al funzionamento dell'applicativo desiderato.

1.5.8 Oracle Secure Global Desktop

Oracle secure global desktop¹² è uno strumento sviluppato da Oracle per l'accesso sicuro alle macchine virtuali da remoto.

Nel contesto aziendale viene utilizzato per centralizzare la definizione dei permessi di accesso alla flotta dei server del [datacenter](#) privato.

1.5.9 Ambienti di sviluppo

L'azienda non pone limitazioni per quanto riguarda gli [IDE](#), tuttavia mostra una forte preferenza per i software sviluppati da JetBrains¹³, in particolare *IntelliJ IDEA*, per Java, *PyCharm* per Python e *Webstorm* per Javascript.

⁹[site:artifactory.](#)

¹⁰[site:puppet.](#)

¹¹[site:foreman.](#)

¹²[site:osgd.](#)

¹³[site:jetbrains.](#)

Necessitando il mio progetto di diverse piattaforme d'esecuzione, ho optato per *Visual Studio Code*¹⁴, IDE sviluppato da Microsoft¹⁵ e consigliato per lavorare per progetti multi linguaggio, in quanto estensibile tramite [plugin](#) per supportare la maggior parte dei linguaggi di programmazione.

1.6 Propensione all'innovazione

Infocert pone l'innovazione alla base della crescita dell'azienda, non solo infatti la compagnia ha già ottenuto diverse certificazioni di qualità, ma una buona parte degli investimenti viene riposta in ricerca e sviluppo, partendo dall'acquisizione di società innovative fino alla collaborazione con le università per gli stage accademici, che consentono di concentrare risorse unicamente sulla ricerca e lo sviluppo di nuove ed efficaci soluzioni.

Il personale lavorativo inoltre è spesso coinvolto in processi di formazione aziendale, iniziando dai corsi individuali per arrivare alla partecipazione a workshop e conferenze per rimanere aggiornati sulle ultime novità in ambito tecnologico.

Grazie ad una concreta [Corporate Social Responsibility](#) e ad un robusto codice etico, Infocert sostiene in modo fervido l'azienda ecosostenibile, dalla formazione del singolo fino all'organizzazione dei processi lavorativi.

¹⁴<site:vscode>.

¹⁵<site:microsoft>.

Capitolo 2

Interessi e Aspettative

2.1 Progetto Test di Carico

2.1.1 Descrizione

Il progetto Test di Carico, nella sua accezione più semplice, si traduce in un effettivo caso di studio dello stato dell'arte del mondo del **Performance Testing**.

Questo processo è una categoria di test di sistema essenziale per applicazioni esposte a grandi flussi di utenza e con vincoli di qualità stringenti riguardanti le performance del sistema. Lo scopo finale di questi test è infatti quello di misurare le prestazioni di un applicativo a svariati livelli di utilizzo permettendo di determinarne eventuali pregi e difetti, e attestare la qualità del software prima del rilascio.

Applicativi rilasciati senza includere i Performance Test nella fase di validazione spesso presentano problemi di reattività, peggiorando l'esperienza dell'utente e causando cattiva reputazione del servizio.

Esistono svariate declinazioni dei performance test, tutte concentrate sui vari aspetti dei possibili problemi di prestazioni e incentrate su determinati tipi di metrica.

Lo stage si è concentrato su due particolari categorie di questi test, i **Test di carico** e gli **Stress Test**.

Test di carico

I test di carico si concentrano sul verificare le abilità del sistema a rispondere a flussi di carico predeterminati, ricavati da aspettative di popolarità che l'applicativo dovrà sostenere. In questo particolare tipo di test infatti si applica il traffico desiderato al sistema, verificando che i tempi di risposta (e quindi l'esperienza utente) siano coerenti con i valori indicati negli [SLA](#).

Dai risultati prodotti da questo test si deduce se l'infrastruttura a supporto dell'applicativo sia adeguata a sostenere il flusso d'utenza che gli studi di mercato predicono.

Stress Test

Gli stress test tendono invece ad individuare il limite superiore delle performance di un sistema, verificando la robustezza di un applicativo all'insorgere di flussi d'utenza non previsti inizialmente.

Questo processo quindi viene fatto aumentando gradualmente le richieste al sistema

fino all'insorgere di risposte non previste per arrivare eventualmente alla rottura del sistema, ovvero quando il software smette completamente di rispondere.

La prima parte dello studio si è quindi concentrata sulla ricerca delle soluzioni software presenti sul mercato che permettano di simulare il comportamento di un grande flusso di utenti, software racchiusi nella categoria dei **Load testing tools**.

Essendo Legalmail un prodotto enterprise il traffico d'utenza usuale è molto alto, richiedendo un grande sforzo computazionale per simularne il volume; per questo la seconda parte dello studio è stata rivolta all'analisi delle soluzioni nel cloud che permettano la simulazione di questi carichi in caso l'infrastruttura interna dell'azienda non ne fosse in grado.

Una volta ottenute le basi teoriche sopraelencate si sarebbe progettata ed implementata un'infrastruttura che permettesse l'esecuzione di questi test in modo *semplice, flessibile, ripetibile*, ed *automatizzabile*.

2.1.2 Obiettivi

2.1.3 Pianificazione Temporale

2.1.4 Prodotti Attesi

2.1.5 Vincoli Metodologici

La realizzazione dello studio e la progettazione dell'infrastruttura non hanno subito limitazioni durante il loro svolgimento, ogni obiettivo è stato perseguito in piena libertà, ponendo come unico obbligo la documentazione adeguata di ogni scelta progettuale e metodologica effettuata.

L'azienda ha inoltre posto forte attenzione sulla mia partecipazione agli eventi SCRUM affrontati nel periodo di stage, in modo da ampliare la mia formazione sulle metodologie Agile.

2.1.6 Vincoli Tecnologici

L'azienda non ha imposto veri e propri vincoli sulle tecnologie da utilizzare per portare a termine gli obiettivi. Tuttavia, sia per la parte infrastrutturale che per quella applicativa, sarebbe stato preferibile proporre una soluzione che utilizzasse linguaggi di programmazione già adottate all'interno del team Legalmail, in modo da abbattere l'eventuale [debito tecnico](#) d'ingresso.

Eventuali prodotti che interagissero con l'infrastruttura esistente avrebbero comunque dovuto rispettare i vincoli tecnologici della stessa.

2.2 Interessi Aziendali

2.2.1 Stage nella strategia aziendale

2.2.2 I test di carico nel ciclo di vita del software

Infocert tramite questo progetto formativo mira ad impostare le basi per l'inserimento in modo costante dei test di carico nel ciclo di vita del software da loro rilasciato.

Il team Legalmail già in passato ha effettuato test di questo tipo ma, a causa di stretti vincoli temporali e dall'impossibilità di posizionare risorse esclusivamente su questo progetto, non ha avuto modo di creare un'infrastruttura che rendesse questo processo ripetibile e semplice da implementare. Il gruppo infatti ha da tempo abbracciato la filosofia [DevOps](#), sviluppando [framework](#) interni ottimizzati per aderire alla pratica dell'Integrazione Continua, proprio per questo motivo il team ha estremo interesse nello sviluppare un framework facilmente integrabile nella loro piattaforma anche per l'esecuzione dei test di carico.

Per l'azienda quindi, l'inserimento di una risorsa che si occupi di questo progetto a tempo pieno, risulta una scelta molto efficace.

L'esecuzione dei test di carico, inoltre, presenta vantaggi non indifferenti e ha valenza informativa non solo per l'area tecnica ma anche per gli [stakeholders](#): la prima infatti può analizzare i dati ottenuti per pianificare l'ampliamento o la riduzione della flotta dei server, mentre i secondi possono utilizzare i dati come certificazione della qualità del prodotto.

2.3 Aspettative Personali

Capitolo 3

Load Test Framework

3.1 Analisi delle soluzioni esistenti

3.1.1 Strumenti di test di carico

3.1.2 Infrastructure as Code

3.1.3 Cloud Platform

3.1.4 Provisioner

3.1.5 Orchestrator

3.1.6 La soluzione proposta

3.2 Progettazione dell'orchestratore

3.2.1 Lo scontro con l'infrastruttura

3.2.2 Core

3.2.3 Command Line Interface

3.2.4 Enterprise Edition

3.3 Progettazione dei test

3.3.1 Analisi delle capacità dei server

3.3.2 Template di scenario

3.4 Esecuzione dei test

3.5 Documentazione prodotta

Capitolo 4

Valutazioni finali

4.1 Soddisfacimento Obiettivi

4.2 Maturazione Professionale

conoscenza mondo enterprise
metodologia SCRUM
conoscenza teorica IaC
conoscenza teorica vari load test tool
competenze jmeter
competenze terraform
competenze aws
competenze puppet (+o-)
miglioramenti python & bash

4.3 Considerazioni Personali

Glossario

AgID L'Agenzia per l'Italia digitale (abbreviato AgID) è una agenzia pubblica italiana istituita dal governo Monti. L'Agenzia è sottoposta ai poteri di indirizzo e vigilanza del presidente del Consiglio dei ministri o del ministro da lui delegato. Svolge le funzioni ed i compiti ad essa attribuiti dalla legge al fine di perseguire il massimo livello di innovazione tecnologica nell'organizzazione e nello sviluppo della pubblica amministrazione e al servizio dei cittadini e delle imprese, nel rispetto dei principi di legalità, imparzialità e trasparenza e secondo criteri di efficienza, economicità ed efficacia. [1](#), [13](#)

Architettura a microservizi L'architettura a microservizi è uno stile architetturale per lo sviluppo di una singola applicazione come un insieme di microservizi, questi sono dei servizi piccoli e autonomi, eseguiti come processi distinti, che lavorano insieme comunicando mediante meccanismi leggeri. Ogni microservizio si occupa di una sola specifica unità applicativa.. [2](#), [14](#)

Branch L'architettura a microservizi è uno stile architetturale per lo sviluppo di una singola applicazione come un insieme di microservizi, questi sono dei servizi piccoli e autonomi, eseguiti come processi distinti, che lavorano insieme comunicando mediante meccanismi leggeri. Ogni microservizio si occupa di una sola specifica unità applicativa.. [5](#), [13](#)

Build L'architettura a microservizi è uno stile architetturale per lo sviluppo di una singola applicazione come un insieme di microservizi, questi sono dei servizi piccoli e autonomi, eseguiti come processi distinti, che lavorano insieme comunicando mediante meccanismi leggeri. Ogni microservizio si occupa di una sola specifica unità applicativa.. [5](#), [6](#), [13](#)

CNA La CNA, Confederazione Nazionale dell'Artigianato e della Piccola e Media Impresa, dal 1946 rappresenta e tutela gli interessi delle micro, piccole e medie imprese, operanti nei settori della manifattura, costruzioni, servizi, trasporto, commercio e turismo, delle piccole e medie industrie, ed in generale del mondo dell'impresa e delle relative forme associate, con particolare riferimento al settore dell'artigianato; degli artigiani, del lavoro autonomo, dei professionisti nelle sue diverse espressioni, delle imprenditrici e degli imprenditori e dei pensionati. [1](#), [13](#)

Configuration Management L'architettura a microservizi è uno stile architetturale per lo sviluppo di una singola applicazione come un insieme di microservizi, questi sono dei servizi piccoli e autonomi, eseguiti come processi distinti, che lavorano insieme comunicando mediante meccanismi leggeri. Ogni microservizio si occupa di una sola specifica unità applicativa.. [6](#)

Datacenter L'architettura a microservizi è uno stile architetturale per lo sviluppo di una singola applicazione come un insieme di microservizi, questi sono dei servizi piccoli e autonomi, eseguiti come processi distinti, che lavorano insieme comunicando mediante meccanismi leggeri. Ogni microservizio si occupa di una sola specifica unità applicativa.. [6](#), [13](#)

Deploy L'architettura a microservizi è uno stile architetturale per lo sviluppo di una singola applicazione come un insieme di microservizi, questi sono dei servizi piccoli e autonomi, eseguiti come processi distinti, che lavorano insieme comunicando mediante meccanismi leggeri. Ogni microservizio si occupa di una sola specifica unità applicativa.. [5](#), [6](#), [13](#)

DevOps DevOps (contrazione dei termini "development" e "operations") è una cultura/pratica dell'ingegneria del software che mira a unificare lo sviluppo del software e le operazioni effettuate per gestirlo. La caratteristica principale del movimento è il forte orientamento verso l'automazione e il monitoraggio di tutti gli step della costruzione del software, partendo dalla stesura della prima riga di codice fino alla gestione dell'infrastruttura. [3](#), [7](#), [14](#)

Framework Un framework, in informatica e specificatamente nello sviluppo software, è un'architettura logica di supporto (spesso un'implementazione logica di un particolare design pattern) su cui un software può essere progettato e realizzato, spesso facilitandone lo sviluppo da parte del programmatore. [2](#), [7](#), [14](#)

CI L'architettura a microservizi è uno stile architetturale per lo sviluppo di una singola applicazione come un insieme di microservizi, questi sono dei servizi piccoli e autonomi, eseguiti come processi distinti, che lavorano insieme comunicando mediante meccanismi leggeri. Ogni microservizio si occupa di una sola specifica unità applicativa.. [5](#), [14](#)

SPID SPID è il sistema di autenticazione che permette a cittadini ed imprese di accedere ai servizi online della pubblica amministrazione e dei privati aderenti con un'identità digitale unica. L'identità SPID è costituita da credenziali (nome utente e password) che vengono rilasciate all'utente e che permettono l'accesso a tutti i servizi online. [1](#), [14](#)

Stakeholders Tutti i soggetti, individui od organizzazioni, attivamente coinvolti in un'iniziativa economica (progetto, azienda), il cui interesse è negativamente o positivamente influenzato dal risultato dell'esecuzione, o dall'andamento, dell'iniziativa e la cui azione o reazione a sua volta influenza le fasi o il completamento di un progetto o il destino di un'organizzazione. [2](#), [8](#), [14](#)

Way Of Working L'architettura a microservizi è uno stile architetturale per lo sviluppo di una singola applicazione come un insieme di microservizi, questi sono dei servizi piccoli e autonomi, eseguiti come processi distinti, che lavorano insieme comunicando mediante meccanismi leggeri. Ogni microservizio si occupa di una sola specifica unità applicativa.. [4](#), [14](#)

Bibliografia