

**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 l'individuazione di un'infrastruttura ideale per l'esecuzione di test di carico. 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 si chiedeva di instanziare un'infrastruttura atta all'esecuzione dei test in ambiente di sviluppo e accettazione con la relativa esecuzione e analisi dei risultati prodotti.

Per ultimo veniva richiesta la stesura di un documento che dimostrasse le scelte effettuate e permettesse al team di proseguire con lo sviluppo dell'infrastruttura realizzata.



*“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

<b>1</b>	<b>Il Contesto Aziendale</b>	<b>1</b>
1.1	Profilo Aziendale . . . . .	1
1.2	Legalmail . . . . .	2
1.3	Tecnologie Utilizzate . . . . .	2
1.3.1	Java 8 Enterprise Edition . . . . .	2
1.3.2	Angular JS . . . . .	2
1.3.3	Python . . . . .	2
1.3.4	Puppet . . . . .	2
1.3.5	YAML . . . . .	2
1.4	Metodologia Aziendale . . . . .	2
1.4.1	Metodologia SCRUM . . . . .	2
1.4.2	Smart Working . . . . .	4
1.4.3	Infrastruttura e Applicazioni . . . . .	4
1.5	Strumenti Utilizzati . . . . .	4
1.5.1	Jira . . . . .	4
1.5.2	Confluence . . . . .	5
1.5.3	Gitlab . . . . .	5
1.5.4	Jenkins . . . . .	5
1.5.5	Artifactory . . . . .	5
1.5.6	Puppet Enterprise . . . . .	6
1.5.7	Foreman . . . . .	6
1.5.8	Oracle Secure Global Desktop . . . . .	6
1.5.9	Ambienti di sviluppo . . . . .	6
1.6	Propensione all'innovazione . . . . .	6
<b>2</b>	<b>Interessi e Aspettative</b>	<b>7</b>
2.1	Progetto Test di Carico . . . . .	7
2.1.1	Descrizione . . . . .	7
2.1.2	Obiettivi . . . . .	7
2.1.3	Pianificazione Temporale . . . . .	7
2.1.4	Risultati Attesi . . . . .	7
2.1.5	Vincoli Metodologici . . . . .	7
2.1.6	Vincoli Tecnologici . . . . .	7
2.2	Interessi Aziendali . . . . .	7
2.2.1	Stage nella strategia aziendale . . . . .	7
2.2.2	I test di carico nel ciclo di vita del software . . . . .	7
2.3	Aspettative Personali . . . . .	8

<b>3</b>	<b>Load Test Framework</b>	<b>9</b>
3.1	Analisi delle soluzioni esistenti . . . . .	9
3.1.1	Strumenti di test di carico . . . . .	9
3.1.2	Strumenti di orchestrazione . . . . .	9
3.2	Progettazione e Sviluppo dei test . . . . .	9
3.3	Orchestrazione dell'infrastruttura . . . . .	9
3.4	Esecuzione e reportistica . . . . .	9
3.5	Documentazione prodotta . . . . .	9
<b>4</b>	<b>Valutazioni finali</b>	<b>11</b>
4.1	Soddisfamento Obiettivi . . . . .	11
4.2	Maturazione Professionale . . . . .	11
4.3	Considerazioni Personali . . . . .	11
	<b>Glossario</b>	<b>13</b>
	<b>Bibliografia</b>	<b>15</b>



# 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 SpA è una società di proprietà al 99,9% di Tecnoinvestimenti SPA. Inoltre, InfoCert è azionista al 80% di Sixtema, il partner tecnologico del mondo [CNA](#).

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

### 1.3.1 Java 8 Enterprise Edition

### 1.3.2 Angular JS

### 1.3.3 Python

### 1.3.4 Puppet

### 1.3.5 YAML

## 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 di approvvigionamento e monitoraggio dell'infrastruttura, oltre che dell'automazione dei compiti ripetitivi.

I momenti formali che prevede questa metodologia sono chiamati eventi e sono *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 gli sviluppatori 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 si divide nelle seguenti colonne:

- \* **Todo::**
- \* **Progress::**
- \* **Impediment::**
- \* **Developed::**
- \* **Done::**

La cosa che più mi ha colpito di queste riunioni è il carattere informale all'interno di esse, che permette di concentrarsi maggiormente sulle decisioni da prendere a discapito del formalismo.

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 Smart Working

Essendo Infocert un'azienda distribuita in più sedi sul territorio italiano, è molto probabile che i componenti di un team non lavorino nello stesso ufficio fisico, all'interno del team Legalmail infatti quattro componenti erano distribuiti tra le sedi di Milano e Roma.

Questa particolare caratteristica però non può influire con l'agilità del processo lavorativo, l'azienda ha quindi predisposto un sistema per gestire al meglio le comunicazioni virtuali: per organizzare le riunioni con i componenti da remoto (come ad esempio il Daily SCRUM) ogni team deve prenotare una stanza virtuale alla quale è possibile collegarsi tramite le opportune autorizzazioni e, di conseguenza, rendere agevoli gli incontri virtuali.

Per garantire l'agilità di questo processo all'interno degli uffici sono presenti stanze fisiche predisposte per le riunioni virtuali: telecamere grandangolari, schermi molto grandi e microfoni d'ambiente.

Questa particolare predisposizione al lavoro remoto ha innescato il fenomeno dello smart working: tutti i dipendenti aziendali hanno diritto ad un giorno alla settimana per poter lavorare da casa, in modo da garantire un adeguato bilanciamento tra vita personale e professionale.

È bene precisare che, per quanto flessibile, lo smart working è una pratica ben organizzata, i giorni disponibili sono infatti fissati e gestiti in modo da garantire la presenza della maggior parte del team in ufficio.

### 1.4.3 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, infrastrutture 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

Jira<sup>1</sup> 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.

---

<sup>1</sup>[site:jira](https://www.atlassian.com/jira).

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

Confluence<sup>2</sup> 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'esecuzione dei test di carico.

### 1.5.3 Gitlab

Il versionamento degli applicativi viene fatto tramite Git<sup>3</sup> e i repository sono ospitati su Gitlab<sup>4</sup>: una piattaforma che permette la gestione centralizzate 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

Jenkins<sup>5</sup> è uno strumento open source sviluppato dalla Jenkins CI community che permette l'implementazione di sistemi di [Integrazione Continua](#) su web-server Java.

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

Artifactory<sup>6</sup> è uno strumento di repository per pacchetti applicativi (binaries) sviluppato da JFrog, questo permette il versionamento delle [build](#) e una conseguente diminuzione

---

<sup>2</sup>[site:confluence.](#)

<sup>3</sup>[site:git.](#)

<sup>4</sup>[site:gitlab.](#)

<sup>5</sup>[site:jenkins.](#)

<sup>6</sup>[site:artifactory.](#)

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 Enterprise<sup>7</sup> 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

Foreman<sup>8</sup> 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 desktop<sup>9</sup> 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

## 1.6 Propensione all'innovazione

---

<sup>7</sup>[site:puppet.](#)

<sup>8</sup>[site:foreman.](#)

<sup>9</sup>[site:osg.](#)



## Capitolo 2

# Interessi e Aspettative

*In questa sezione verranno presentati gli obiettivi del progetto, la pianificazione temporale del periodo di stage, gli interessi aziendali verso i progetti di stage e verso questo progetto nello specifico.*

*In conclusione verrà effettuata una valutazione delle mie aspettative personali.*

### 2.1 Progetto Test di Carico

#### 2.1.1 Descrizione

#### 2.1.2 Obiettivi

#### 2.1.3 Pianificazione Temporale

#### 2.1.4 Risultati Attesi

#### 2.1.5 Vincoli Metodologici

#### 2.1.6 Vincoli Tecnologici

### 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 Strumenti di orchestrazione

### 3.2 Progettazione e Sviluppo dei test

### 3.3 Orchestrazione dell'infrastruttura

### 3.4 Esecuzione e reportistica

### 3.5 Documentazione prodotta



## Capitolo 4

# Valutazioni finali

- 4.1 Soddisfacimento Obiettivi
- 4.2 Maturazione Professionale
- 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](#)

**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](#)

**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](#)

**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](#)

. [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](#)

**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](#)

**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](#), [13](#)

**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](#), [13](#)

**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](#)

**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](#), [13](#)

**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](#), [13](#)

**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](#)



# Bibliografia