

Università degli Studi di Padova

DIPARTIMENTO DI MATEMATICA "TULLIO LEVI-CIVITA"

CORSO DI LAUREA IN INFORMATICA



Progettazione e sviluppo dell'applicazione  
per la gestione amministrativa di VIC SRL

*Tesi di laurea triennale*

*Relatore*

Dott. Silvia Crafa

*Laureando*

Luca Allegro

---

ANNO ACCADEMICO 2017-2018



*Be pitiful, for every man is fighting a hard battle. .*

---

– Ian Maclaren , *The British Weekly*, 1897

Dedicato ad Alessio e nonno Lorenzo



# Sommario

---

Il presente documento descrive il lavoro svolto durante il periodo di stage dal laureando Luca Allegro presso l'azienda VIC s.r.l.

Il tirocinio si è tenuto presso il Dipartimento di Ricerca e Sviluppo (R&D) durante i mesi di luglio e agosto 2018 e ha avuto la durata di 310 ore.

L'obiettivo dello stage è stato lo sviluppo di un'applicazione web per la gestione amministrativa dell'azienda che comprende il tenere traccia dei movimenti bancari e la gestione delle fatture attive e delle fatture passive.

L'elaborato ha lo scopo di illustrare:

- il contesto aziendale dove è stato svolto lo stage ([Capitolo 1](#));
- la descrizione del progetto sviluppato ([Capitolo 2](#));
- le attività svolte durante esso ([Capitolo 3](#));
- una valutazione finale sull'esperienza e le competenze acquisite([Capitolo 4](#)).



# Ringraziamenti

---

*Innanzitutto, vorrei esprimere la mia gratitudine alla Professoressa Silvia Crafa, relatrice della mia tesi, per la l'attenzione e l'aiuto fornitomi durante la stesura del lavoro.*

*Un sincero grazie a Lorenzo e Luca per avermi accompagnato in modo impeccabile nell'inserimento in azienda e aver reso lo stage un'esperienza di crescita.*

*Ringrazio i compagni di studio per aver reso indimenticabile questo percorso condividendo giornate (di studio e non), difficoltà e successi come una famiglia.*

*Ringrazio l'Associazione Italiana Arbitri per tutte le persone che mi ha permesso di conoscere e per le infinite occasioni di crescita e confronto che mi offre.*

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

*Desidero ringraziare con affetto la mia famiglia per essermi stata vicina e per avermi dato fiducia durante tutti gli anni di studio.*

*Ringrazio miei genitori per gli insegnamenti di vita che ogni giorno mi danno, il miglior esempio da seguire.*

*Ringrazio tutte le altre persone a me care che conosco e ho conosciuto e che hanno contribuito, anche solo per un momento, a farmi diventare la persona che sono oggi.*

*Padova, Dicembre 2018*

Luca Allegro





# Indice

---

<b>1</b>	<b>L'Azienda</b>	<b>1</b>
1.1	Chi è VIC . . . . .	1
1.2	Il Dipartimento di Ricerca e Sviluppo . . . . .	2
1.3	Opportunità di Stage . . . . .	2
<b>2</b>	<b>Il Progetto</b>	<b>3</b>
2.1	L'applicativo . . . . .	3
2.1.1	Banche . . . . .	3
2.1.2	Fatture attive . . . . .	4
2.1.3	Fatture passive . . . . .	4
2.2	Obbiettivi . . . . .	5
<b>3</b>	<b>Svolgimento</b>	<b>7</b>
3.1	Pianificazione . . . . .	7
3.2	Tecnologie adottate . . . . .	8
3.2.1	Django . . . . .	8
3.3	Progettazione . . . . .	8
3.4	Realizzazione . . . . .	9
3.5	Verifica e Validazione . . . . .	9
3.5.1	Analisi Statica . . . . .	9
3.5.2	Analisi dinamica . . . . .	9
3.6	Risultato finale . . . . .	9
3.6.1	Modifiche Migliorative . . . . .	9
<b>4</b>	<b>Analisi retrospettiva</b>	<b>11</b>
4.1	Bilancio formativo . . . . .	11
4.2	Competenze Tecnologiche . . . . .	11
4.3	Esperienza . . . . .	12
<b>A</b>	<b>API Rest</b>	<b>13</b>
A.1	Introduzione . . . . .	13
A.2	REST . . . . .	13
A.2.1	Identificazione delle risorse . . . . .	13
A.2.2	Utilizzo esplicito di metodi Hypertext Transfer Protocol (HTTP) . . . . .	13
A.2.3	Risorse autodescrittive . . . . .	14
A.2.4	Collegamenti tra risorse . . . . .	14
A.2.5	Comunicazioni prive di stato . . . . .	14

<b>Glossario</b>	<b>15</b>
<b>Acronimi</b>	<b>17</b>
<b>Bibliografia</b>	<b>19</b>

# Elenco delle figure

---

1.1	Il logo di VIC . . . . .	1
3.1	Metologia Agile Fonte: <a href="https://www.cleart.com/what-is-agile-software-development.html">https://www.cleart.com/what-is-agile-software-development.html</a> . . . . .	7

# Elenco delle tabelle

---



# 1 L'Azienda

---

*Questo capitolo descrive la realtà aziendale e il suo ambito di interesse*

## 1.1 Chi è VIC

VIC s.r.l.[1] offre servizi di ispezione delle merci in tutto il mondo per individuare eventuali non conformità delle merci al fine di tutelare i propri clienti e ridurre al minimo le perdite.

VIC nasce a Venezia nel 2006 ed ora è una delle più grandi compagnie del mercato globale in quanto è presente con 22 diverse aziende e più di 50 laboratori in tutti i più grandi snodi di scambio del mondo.

La chiave del successo di VIC s.r.l. è l'attenzione che pone alla riduzione del tempo tra ispezione e reporting al cliente. Questo è stato possibile grazie all'automazione e alla semplificazione di molte attività svolte dai dipendenti mediante moderni sistemi informatici e allo sviluppo di diversi servizi web ed applicazioni mobile con l'obiettivo di fornire più informazioni possibili (inclusi foto e video) ai clienti.



**Figura 1.1:** Il logo di VIC

## 1.2 Il Dipartimento di Ricerca e Sviluppo

Il Dipartimento di Ricerca e Sviluppo (abbreviato: R&D) ha sede a Padova e si occupa di realizzare e mantenere tutti gli strumenti che l'azienda offre, ad esempio:

- il portale *VIC Online* per i clienti, mediante cui è possibile controllare lo stato dei propri ordini, ottenere informazioni utili, immagini e video riguardanti la merce ordinata;
- le applicazioni rivolte agli ispettori per raccogliere i dati sulla merce e realizzare dei resoconti sulle analisi effettuate e lo stato della merce.

Il dipartimento R&D è alla continua ricerca di metodi di innovazione per velocizzare, automatizzare e standardizzare le attività svolte dall'azienda nell'ambito del controllo qualità e quantità delle merci.

Tutti i servizi sviluppati dal dipartimento R&D sono composti dalle componenti front-end (client) e back-end (server). La comunicazione tra queste due componenti avviene utilizzando le quattro operazioni CRUD (Create, Read, Update, Delete) di HTTP, in modo da essere conformi ai principi [REST](#) (REpresentational State Transfer).

## 1.3 Opportunità di Stage

VIC s.r.l. è favorevole ad ospitare laureandi per lo stage perché crede che questa attività porti molteplici benefici all'azienda, nello specifico:

- il confronto con persone nuove e la loro esperienza può offrire spunti di miglioramento per i sistemi e le metodologie;
- per lo studio di nuove tecnologie e sul loro possibile utilizzo in progetti aziendali;
- la conoscenza di possibili candidati per l'assunzione.

# 2 Il Progetto

---

*Descrizione del progetto di stage, motivazioni, obiettivi.*

## 2.1 L'applicativo

L'azienda utilizza un software gestionale ricco di funzionalità, ma molto datato. Esso infatti è stato sviluppato interamente in [PHP](#)[2] ed è basato sul framework *QCubed*[3] che sta per concludere il suo ciclo di vita in quanto non è supportato più supportato e non è compatibile con le nuove versioni di [PHP](#).

Lo scopo dello stage è stato quello di sviluppare un nuovo gestionale che si interfacci con lo stesso database del precedente e con le stesse funzionalità, ma basato su tecnologie moderne, prestando cura anche al rinnovamento dell'interfaccia grafica.

Nello specifico il software da realizzare rappresenta il Modulo Amministrativo: esso è uno dei moduli principali in cui è diviso il software di gestione in uso attualmente presso l'azienda ed si occupa di gestire tutte le fatture e tenere sotto controllo la gestione contabile.

Esso si divide in tre sottomoduli: Banche, Fatture Attive e Fatture Passive.

### 2.1.1 Banche

Il modulo banche si occupa di tenere sotto controllo la situazione contabile dei vari conti correnti aziendali.

Dovrà offrire le seguenti funzionalità:

- visualizzazione della lista delle banche associata alla sede corrente mostrandone per ognuna:
  - il saldo reale (inserito manualmente dagli operatori) di ogni banca;
  - la data di inserimento del saldo reale;
  - il saldo calcolato dal sistema in base ai movimenti bancari inseriti (che dovrebbe coincidere con il saldo reale, se aggiornato);
  - il fido di cassa, cioè la somma che la banca mette a disposizione dell'azienda;

- il saldo disponibile, ottenuto sommando il saldo reale al, se presente, il fido di cassa;
- visione di tutti i movimenti di un conto corrente;
- aggiornamento del saldo di un conto corrente;
- caricamento e visione di estratti conti giornalieri e mensili;
- inserimento, modifica e rimozione di movimenti bancari;
- generazione di file nei formati *PDF* e *xls* della lista dei movimenti bancari.

### 2.1.2 Fatture attive

Il modulo delle fatture attive si occupa di tenere traccia di tutti gli ordini completati e in attesa di fatturazione, della creazione della fattura stessa con la possibilità di specificare i parametri richiesti (data fattura, scadenza, tipo fattura. . . ), della possibilità di inviare le fatture al cliente e la gestione dello scadenziario delle fatture scadute e non ancora incassate.

Dovrà offrire le seguenti funzionalità:

- visualizzazione degli ordini completati e non ancora fatturati di cui è possibile creare la fattura;
- ricerca di fatture attraverso il numero di ordine o il nome della nave;
- creazione, modifica e rimozione di una fattura;
- visualizzazione della lista di fatture non ancora inviate al cliente;
- generazione di file in formato *PDF* corrispondenti alle fatture;
- invio di fatture al cliente tramite email;
- scadenziario fatture emesse da incassare
  - già scadute;
  - in scadenza.

### 2.1.3 Fatture passive

Il modulo delle fatture passive si occupa di tenere traccia di tutte le passività relative all'attività. Si dividono in due gruppi: dirette e indirette. Le fatture passive dirette rappresentano delle passività direttamente riconducibile ad un preciso ordine (fornitori, agenzie, laboratori, spedizionieri), mentre le indirette rappresentano dei costi generici di gestione (elettricità, bollette telefoniche, affitti, . . . ). Le fatture passive indirette si suddividono a propria volta in due categorie: una tantum e cicliche.

Dovrà offrire le seguenti funzionalità:

- visualizzazione lista passive dirette non pagate raggruppate per fornitore;
- visualizzazione lista passive indirette (divise tra cicliche e una tantum);
- visualizzazione lista fatture per ogni fornitore;



- visione dettaglio singola fattura;
- modifica fattura;
- eliminazione fattura;
- generazione del movimento bancario associato (con riempimento automatico dei campi).

## 2.2 Obiettivi

Sulla base della durata massima di 320 ore prevista per lo stage, il tutor e lo stagista hanno stilato un piano di lavoro e coerentemente hanno concordato gli obiettivi minimi e massimi che si aspetta di veder raggiunti al termine del rapporto lavorativo.

Gli obiettivi minimi concordati sono:

1. comprensione del software e del database esistenti;
2. studio delle tecnologie;
3. studio degli strumenti di sviluppo;
4. definire, progettare, codificare e verificare le funzionalità che riguardano i moduli:
  - Banche;
  - Fatture Attive.

Gli obiettivi massimi concordati sono:

1. definire, progettare, codificare e verificare le funzionalità che riguardano il modulo:
  - Fatture Passive.



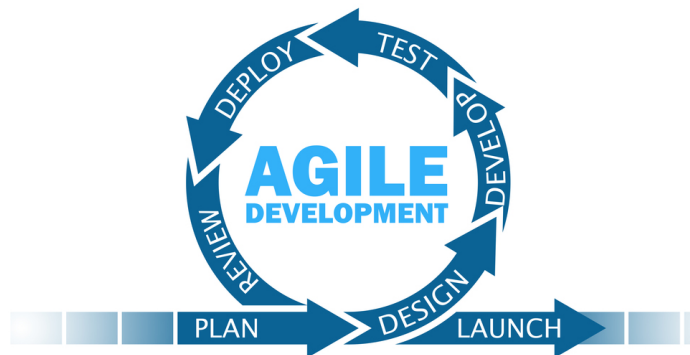
## 3 Svolgimento

---

*Il presente capitolo descrive le varie fasi dello sviluppo dell'applicazione*

### 3.1 Pianificazione

Come modello per la gestione del progetto ho adottato la metodologia Agile. Tale modello mi ha permesso di adottare una certa flessibilità e velocità nel rispondere alle esigenze e ai feedback del Dipartimento di Amministrazione.



**Figura 3.1:** Metodologia Agile

Fonte: <https://www.cleart.com/what-is-agile-software-development.html>

Il modello Agile prevede iterazioni continue, della durata massima di due settimane, entro le quali si susseguono attività di:

1. analisi dei requisiti che emergono dall'interazione con il cliente, in questo caso il tutor aziendale e il Dipartimento di Amministrazione;
2. pianificazione delle funzionalità da includere nello sprint, suddivisione in task e loro assegnazione ai membri del team;

3. progettazione delle funzionalità da implementare nell'iterazione corrente;
4. codifica e sviluppo delle funzionalità previste;
5. verifica;
6. rilascio;
7. monitoraggio continuo e verifica dello stato dell'applicazione.

Il punto di partenza di ogni ciclo è il risultato raggiunto con il precedente e il feedback ricevuto tramite la presentazione della soluzione raggiunta fin'ora agli stakeholder.

La durata dello stage è stata di 8 settimane quindi ho pianificato 4 sprint della durata di due settimane:

1. studio delle tecnologie, degli strumenti di sviluppo e del software gestionale esistente;
2. progettazione, codifica e testing del modulo Banche;
3. progettazione, codifica e testing del modulo Fatture Attive;
4. progettazione, codifica e testing del modulo Fatture Passive.

## 3.2 Tecnologie adottate

### 3.2.1 Django

Per creare il back-end delle applicazioni rivolte ai clienti, il dipartimento utilizza Django, un framework di alto livello gratuito e open source scritto in Python, che incoraggia lo sviluppo veloce e pulito di applicazioni web.

## 3.3 Progettazione

L'architettura del prodotto si suddivide in due macro componenti:

- **Client:** rappresenta l'applicativo desktop attraverso cui gli utenti interagiscono con il front-end dell'applicazione. Il progetto si focalizza sullo sviluppo di questa componente.
- **Server:** rappresenta il sistema remoto che si occupa di calcolare e rispondere alle richieste del Client, viene implementato tramite il back-end. Lo sviluppo di questa componente non è pertinente al progetto ed è assegnata ad altri dipendenti dell'azienda.

Il Client interagirà con il Server tramite delle API messe pubbliche, che sono conformi ai principi REST.

Tecnologie adottate  
Strumenti

## 3.4 Realizzazione

## 3.5 Verifica e Validazione

Durante l'intera attività di sviluppo ho verificato che quanto svolto fosse conforme alle mie aspettative e a quelle del cliente. Per far ciò ho attuato molte prove pratiche sulle funzionalità implementate, per assicurarmi che funzionassero in modo adeguato. L'azienda ha preferito utilizzare le prove pratiche piuttosto che quelle automatizzate, in modo che potessi concentrarmi sull'adeguare il prodotto secondo le loro esigenze. Grazie al lavoro svolto in fase di configurazione dell'ambiente di lavoro, ho potuto testare velocemente l'applicazione in locale con la base di dati contenente i dati reali.

### 3.5.1 Analisi Statica

Per l'attività di verifica durante tutto l'arco di sviluppo del progetto ho utilizzato TSHint tramite il pacchetto Atom-jshint creato appositamente per l'editor Atom. JSHint è uno strumento di analisi statica del codice JavaScript sviluppato a partire da un fork di JSLint, in quanto l'originale risultava essere troppo meticoloso e non personalizzabile. JSHint mi ha permesso di controllare che il codice fosse uniforme e scritto secondo dei buoni canoni, grazie a degli appositi warning ed errori.

### 3.5.2 Analisi dinamica

Per l'analisi del prodotto software tramite la sua esecuzione mi sono avvalso di Genymotion, un'applicazione che consente l'emulazione di dispositivi mobile. Ho dapprima richiesto al tutor aziendale quali fossero le configurazioni tipo su cui l'applicazione sarebbe stata eseguita, così da impostare le macchine virtuali su Genymotion. Successivamente, mi sono assicurato di eseguire dei test mirati a controllare il funzionamento delle specifiche funzionalità dell'applicazione, utilizzando i dati reali contenuti nella base di dati copia fornitami dall'azienda.

Ho individuato le seguenti funzionalità principali che nel complesso formano l'applicazione: .....

## 3.6 Risultato finale

Descrizione dell'applicazione con screenshot

### 3.6.1 Modifiche Migliorative



## 4 Analisi retrospettiva

---

*Considerazioni sull'esperienza maturata durante lo stage e, più in generale, attraverso l'intero percorso formativo.*

### 4.1 Bilancio formativo

Dal punto di vista formativo l'attività di stage è stata estremamente positiva. Ha arricchito il mio bagaglio personale di competenze professionali non banali.

Nonostante lo stage in questione abbia richiesto una considerevole mole di studio ed autoformazione, gli argomenti trattati e di ricerca sono risultati piacevoli, interessanti, utili

mondo aziendale diverso da mondo universitario: Esperienza di integrazione in una realtà aziendale con i vari aspetti positivi (collegli, tutor che aiuta), ma anche negativi l'ambiente di lavoro non ideale secondo i principi dell'ingegneria del software (imprevisiti, urgenze, scadenze, non conoscenza del dominio in cui , il confronto con altre persone, la carenza di documentazione e best practice da seguire)

Durante tutto lo stage ho dovuto lavorare da sola, visto che il progetto era stato appositamente calibrato in questa modalità. Questo mi ha permesso di raggiungere un grado di autonomia che fin'ora non possedevo. Ho comunque avuto la possibilità di confrontarmi con personale qualificato in ogni momento di difficoltà. Solitamente quando sono bloccata su un problema, preferisco cercare di risolverlo da sola, piuttosto che chiedere aiuto a qualcuno. Durante lo stage ho appreso che questa mia metodologia non è del tutto corretta: ho imparato a confrontarmi nei momenti di necessità per raggiungere in tempi più

Questo insegnamento mi sarà sicuramente utile se in futuro lavorerò in un team di sviluppo.

### 4.2 Competenze Tecnologiche

Lo stage studiare in modo più approfondito le tecnologie e usarle in ambiente lavorativo mi ha permesso di farne una valutazione più raffinata, comprenderne pregi e difetti e imparare a farne un uso più consapevole. Nello specifico ho particolarmente migliorato le mie conoscenze riguardanti i seguenti argomenti:

- **la metodologia Agile:** ne ho apprezzato la flessibilità e leggerezza, ma ho constatato che in alcune circostanze la
- **lo stile architetturale REST:** ;
- **il linguaggio Python:** un linguaggio molto intuitivo, semplice e conciso; per contro è un linguaggio a tipizzazione dinamica per cui necessaria l'esecuzione del codice, molto tempo perso per il debugging. A causa questo difetto credo che lungo periodo e in progetti impegnativi sia più conveniente utilizzare altri linguaggi altrettanto ricchi di framework ad esempio Java
- **Django:** supporto a ottima documentazione, query in modo semplice, ottima pagina di debugging
- **Typescript:** sovrainsieme del già conosciuto Javascript,
- **Angular:** ordinato, comoda separazione tra visualizzazione e comportamento, e tra servizi e pagine.
- **l'utilizzo di IntelliJ**(e le sue derivate PyCharm e WebStorm): eccezionale, integrazione con visual environment, permette configurazione del sistema di building in modo semplice per integrazione con Git, sistema di navigazione e ricerca, sistema di versionamento integrato (per ogni modifica), integrazione con accesso a database, Rest client, integra il controllo del codice e di sintassi

### 4.3 Esperienza

La seconda parte che va a formare il bagaglio culturale creato con l'attività di stage è sicuramente l'esperienza aziendale, parte che tra le due ritengo la più importante. La collaborazione con i colleghi, il rapporto con il datore di lavoro e gli stakeholder, il rispetto degli orari, della sicurezza e della regolamentazione costituiscono un'esperienza fondamentale per uno studente di un indirizzo orientato al mondo del lavoro. Poter usufruire di un servizio universitario, che garantisce un'esperienza lavorativa all'interno del percorso di studi, permette a tutti gli studenti che entrano per la prima volta nel mondo del lavoro di conoscere già le dinamiche generali aziendali così da poter fin da subito adattarvisi. Ritengo dunque il bilancio formativo davvero positivo e posso affermare che lo stage rappresenta una delle più importanti attività svolte nel mio personale percorso universitario.

Consiglio pertanto l'attività dello stage anche a tutti quelli studenti di altri corsi di studi che non la prevedono come obbligatoria, in quanto solo tramite il suo svolgimento 'è possibile apprezzare nozioni che durante il normale corso di studi non 'è possibile apprendere.



# A API Rest

---

## A.1 Introduzione

La prima definizione di [API REST](#) si ha grazie alla tesi "*Architectural Styles and the Design of Network-based Software Architectures*" dello studente **Roy Fielding** negli anni 2000, che espone un nuovo tipo di architettura per sistemi di ipertesto distribuiti, come ad esempio il *World Wide Web*.

## A.2 REST

[REST](#) detta un insieme di linee guida riassumibili in cinque principi:

1. identificazione delle risorse;
2. utilizzo esplicito di metodi [HTTP](#);
3. risorse autodescrittive;
4. collegamenti tra risorse;
5. comunicazioni prive di stato.

### A.2.1 Identificazione delle risorse

Per risorsa si intende un qualsiasi elemento oggetto di elaborazione. Le risorse rappresentano un concetto fondamentale nelle [API](#) di tipo [REST](#). Queste risorse vengono richieste tipicamente da una macchina denominata *client* verso una macchina denominata *server*. Le richieste vengono solitamente effettuate tramite il protocollo [HTTP](#), ma è possibile utilizzare un tipo di protocollo diverso. È possibile stabilire anche il formato delle risorse.

### A.2.2 Utilizzo esplicito di metodi [HTTP](#)

Le richieste se effettuate tramite l'utilizzo del protocollo [HTTP](#) possono essere eseguite utilizzando i metodi CRUD. Il protocollo [HTTP](#) mappa i metodi [Create Read Update Delete](#) (CRUD) nella seguente maniera:

- **create** è associato a richieste di tipo PUT o POST;

- **read** è associato a richieste di tipo GET;
- **update** è associato a richieste di tipo PUT o POST;
- **delete** è associato a richieste di tipo DELETE.

### A.2.3 Risorse autodescrittive

I principi [REST](#) non impongono vincoli sulle modalità di rappresentazione di una risorsa. L'utilizzo di formati standard però è consigliato, in maniera tale da avere una migliore comunicazione con i client. Per ottenere una rappresentazione standard è possibile rappresentare le risorse con l'adeguato tipo mime, per contro, un client ha la possibilità di richiedere la risorsa secondo uno specifico formato: in questa maniera, il server può fornire più modi di rappresentare per una risorsa.

### A.2.4 Collegamenti tra risorse

È possibile che siano presenti relazioni tra le risorse. Un metodo consigliato per rappresentarle e distinguerle è tramite l'utilizzo di un identificatore ogni volta diverso. In questa maniera, è possibile avere diversi collegamenti tra risorse tramite un [Uniform Resource Locator \(URL\)](#) univoco.

### A.2.5 Comunicazioni prive di stato

Questa caratteristica è possibile trovarla proprio nel protocollo [HTTP](#), dove non si mantiene un collegamento attivo, non esiste il concetto di "sessione" e ciascuna richiesta non ha alcuna relazione con le richieste precedenti o successive. Questo concetto è stato portato anche nei servizi di tipo [REST](#). In questa maniera è il client che si deve occupare di mantenere lo stato dell'applicazione, permettendo di avere un'alta scalabilità lato *server*, in quanto non sono presenti problemi di sincronizzazione delle varie sessioni o mantenimenti di stato tra le varie macchine.

# A Glossario

---

**Agile** Insieme di principi per lo sviluppo software sotto i quali requisiti e soluzioni evolvono tramite gli sforzi collettivi del gruppo auto-organizzato polifunzionale. L'obiettivo di questo modello è soddisfare il cliente, rilasciando software in maniera continua e in periodi brevi, in modo che possa osservare l'andamento dello sviluppo e i risultati ottenuti. Di conseguenza è anche lecito accogliere eventuali cambiamenti dei requisiti durante l'intero ciclo di sviluppo.. [7](#), [15](#)

**api** Insieme di servizi (procedure, funzioni, strutture dati) che un sistema espone in qualche modo ai propri utilizzatori.. [13](#), [15](#)

**PHP** Linguaggio di scripting interpretato, con licenza open source e libera, originariamente concepito per la realizzazione di pagine web dinamiche.. [3](#)

**REST** Si tratta di un tipo di architettura software per i sistemi di ipertesto distribuiti come il World Wide Web. Un concetto importante in REST è l'esistenza di risorse (fonti di informazioni), a cui si può accedere tramite un identificatore globale (un URI). Per utilizzare le risorse, le componenti di una rete (componenti client e server) comunicano attraverso una interfaccia standard (ad es. HTTP) e si scambiano rappresentazioni di queste risorse.. [2](#), [13](#), [14](#)



# A Acronimi

---

CRUD Create Read Update Delete. [13](#)

HTTP Hypertext Transfer Protocol. [ix](#), [13](#), [14](#)

PHP [PHP: Hypertext Preprocessor](#). [15](#)

REST [REpresentational State Transfer](#). [15](#)

URL Uniform Resource Locator. [14](#)



# A Bibliografia

---

## Riferimenti bibliografici

- [4] Daniel T. Jones James P. Womack. *Lean Thinking, Second Editon*. Simon & Schuster, Inc., 2010.
- [5] Ian Sommerville. *Software Engineering, 10th Edition*. Pearson, 2015.

## Siti web consultati

- [1] VIC s.r.l. URL: <http://vicworldwide.com/> (cit. a p. 1).
- [2] PHP. URL: <http://www.php.net/> (cit. a p. 3).
- [3] QCubed. URL: <http://qcubed.github.io/> (cit. a p. 3).
- [6] Manifesto Agile. URL: <http://agilemanifesto.org/iso/it/>.
- [7] Python Virtual Environment. URL: <http://docs.python-guide.org/en/latest/dev/virtualenvs/>.
- [8] Django. URL: <https://www.djangoproject.com/>.
- [9] Django REST Framework. URL: <http://www.django-rest-framework.org/>.
- [10] JSON. URL: <https://www.json.org/json-it.html>.
- [11] JSON Web Tokens. URL: <https://jwt.io/>.
- [12] Angular. URL: <https://angular.io/>.