



Zeus Code - Progetto "P2PCS"

## Glossario

<b>Versione</b>	1.0.0
<b>Approvazione</b>	Riccardo Dario
<b>Redazione</b>	Andrea Pigatto Irina Hornoiu
<b>Verifica</b>	Marco dalla B�
<b>Stato</b>	Approvato
<b>Uso</b>	Esterno
<b>Destinato a</b>	Zeus Code Red Babel Prof. Tullio Vardanega Prof. Riccardo Cardin

### Descrizione

Questo documento ha lo scopo di chiarire le possibili ambiguit  tra i termini utilizzati all'interno dei documenti scritti e redatti dal team.

zeuscode17@gmail.com

## Diario delle modifiche

Versione	Data	Nominativo	Ruolo	Descrizione
1.0.0	2019-01-12	Matteo Santinon	<i>Responsabile di Progetto</i>	Approvazione del documento per RR.
0.1.0	2019-01-11	Federico Biciato	<i>Verificatore</i>	Verifica documento.
0.0.2	2019-01-10	Mattia Bolzonella, Samuele Giuliano Piazzetta	<i>Analista, Analista</i>	Inserimento dei termini di glossario attraverso lo script in Python.
0.0.1	2018-11-25	Mattia Bolzonella	<i>Analista</i>	Creato documento latex e creata pagina del titolo.

## Indice

A . . . . .	3
B . . . . .	4
C . . . . .	5
D . . . . .	6
E . . . . .	7
F . . . . .	8
G . . . . .	9
H . . . . .	10
I . . . . .	11
J . . . . .	12
K . . . . .	13
L . . . . .	14
M . . . . .	15
N . . . . .	16
O . . . . .	17
P . . . . .	18
Q . . . . .	19
R . . . . .	20
S . . . . .	22
T . . . . .	23
U . . . . .	24
V . . . . .	25
W . . . . .	26
X . . . . .	27
Y . . . . .	28
Z . . . . .	29

## A

### **Add-on**

Parte di software aggiuntiva che estende le funzionalità e/o caratteristiche del software di base. Di solito è eseguito insieme al software in cui è aggiunto.

### **Agile**

Modello di sviluppo software caratterizzato da: team di piccole dimensioni che comunicano informalmente, importanza del software rispetto ai documenti, pianificazione adattiva, coinvolgimento del cliente nel processo di sviluppo.

### **Apache Kafka**

Piattaforma open source di streaming distribuito, usata per stream processing, monitoraggio di dati e traffico web, messaggistica e varie operazioni su file di log.

### **API**

Con Application Programming Interface si intende un insieme di procedure e funzioni offerte ai programmatori per facilitare lo sviluppo. Le API espongono blocchi di codice delle librerie di cui fanno parte, per permetterne il riuso.

### **Area di staging**

L'area di staging è un'area di immagazzinamento intermedia usata per i dati in lavorazione durante l'estrazione, la trasformazione e il caricamento di un processo. Su Git è usata come step precedente al processo di commit.

### **Artefatti**

Un artefatto è un sottoprodotto che viene realizzato durante lo sviluppo software. Sono artefatti i casi d'uso, i diagrammi delle classi, i modelli UML<sub>G</sub>, il codice sorgente e la documentazione varia.

### **Attore**

Entità che interagisce con il sistema per svolgere delle attività. Può essere umana o un'altro sistema.

### **AWS**

Amazon Web Services è un insieme di servizi di cloud computing che compongono la piattaforma on demand offerta dall'azienda Amazon

---

 $B$ 

---

**Back end**

Back end è un termine largamente utilizzato per caratterizzare le interfacce che hanno come destinatario un programma. Una applicazione back end è un programma con il quale l'utente interagisce indirettamente, in generale attraverso l'utilizzo di una applicazione front end.

**Blockchain**

Struttura dati condivisa ed immutabile, composta da un registro di blocchi concatenati in ordine cronologico. I blocchi una volta inseriti non sono più modificabili. Questa tecnologia, assimilabile ad un database distribuito, è gestito da una rete di nodi, ognuno dei quali possiede una copia dei dati. Ogni transazione è regolata da un protocollo che ne verifica l'autenticità e la approva o meno. Una transazione approvata non è più modificabile e viene aggiunta alla blockchain. Con questo sistema non è necessaria la presenza di un'autorità esterna alla transazione che faccia da garante.

**Body of knowledge**

Insieme di conoscenze di uno stesso dominio. Le conoscenze fanno parte del dominio e allo stesso tempo lo costituiscono.

**Boilerplate**

Pratica di riuso del codice in cui un frammento di codice viene replicato in più punti di un file, o in più file nello stesso punto. Solitamente rappresenta la base da cui cominciare per la stesura di un'applicazione.

## C

### **CamelCase**

Pratica di scrivere una parola unendone delle altre lasciando le loro iniziali maiuscole.

### **Capitolato**

Il capitolato è un documento tecnico, in genere allegato ad un contratto di appalto e parte di esso. Serve a definire le specifiche tecniche delle opere che andranno ad eseguirsi per effetto del contratto stesso.

### **Changelog**

Tabella delle modifiche di un documento. Indica la versione del documento, la data di modifica, chi ha prodotto la modifica e una descrizione della stessa.

### **Coesione**

Stretta unione di parti che concorrono alla stessa funzionalità, allo stesso obiettivo: le parti coese sono tutto il necessario e nulla di superfluo.

### **Comportamento emergente**

Nuovo comportamento del software nato dall'integrazione delle parti, spesso non previsto dagli sviluppatori delle singole parti.

### **Continuous delivery**

Definisce la metodologia attraverso la quale si hanno brevi cicli di rilascio. In questo modo è possibile rilasciare in modo affidabile il software in qualsiasi momento.

### **Continuous integration**

E' una pratica di sviluppo che consiste nell'integrare frequentemente le modifiche locali degli sviluppatori al ramo condiviso principale. Questa pratica permette di minimizzare problemi di integrazione corposa.

### **Copertura**

Nell'analisi dei requisiti, è la quantità di requisiti soddisfatti. Nei test del codice, è la quantità di righe percorse durante il test. Si ricerca copertura massima.

## D

### **DApp**

Anche scritta ÐApp, è un'applicazione decentralizzata che gestisce gli smart contract<sub>G</sub>, arricchendo le transazioni con regole user-defined, garantendo maggiore sicurezza e controllo. Solitamente una ÐApp è open-source e gestisce operazioni con dati criptati.

### **Dashboard**

È una postazione o a una pagina web basata su una tecnologia che visualizza informazioni relative a un business, raccolte in tempo reale da varie fonti nel settore. I dati vengono visualizzati in tempo reale nella pagina usando grafici, riepiloghi e liste.

### **Design pattern**

Soluzione progettuale generale ad un problema ricorrente. Una descrizione o un modello da applicare per risolvere un problema che può presentarsi in diverse situazioni durante la progettazione e lo sviluppo del software.

### **DevOps**

Metodologia di sviluppo software, che combina lo sviluppo software con la tecnologia delle informazioni. Nasce dalla sinergia tra cultura aziendale, pratiche e strumenti, e fornisce a un'organizzazione l'abilità di sviluppare applicazioni e servizi con un breve ciclo di vita del software e in allineamento con gli obiettivi aziendali.

### **Dilazione**

L'atto di rimandare ad un secondo momento l'esecuzione di un'operazione, posticipando il termine normalmente fissato.

### **Docker**

Software per la creazione, sviluppo ed esecuzione di applicazione attraverso l'uso di container. I container permettono di incorporare un'applicazione e tutte le parti necessarie ad essa in un unico pacchetto.

### **Documento esterno**

Documento a uso di team, committente e proponente (RedBabel).

### **Documento interno**

Documento a uso del team di sviluppo e del committente (Prof. Vardanega Tullio e Prof. Cardin Riccardo).

### **Driver**

Nel test di unità, il driver funge da main e permette l'esecuzione di unità chiamate dal main.

*E***ESlint**

Lint<sub>er</sub><sub>G</sub> per l'analisi statica del codice e per l'identificazione di pattern in JavaScript. Si usa per ottenere codice corretto, comprensibile, conforme a delle regole date.



---

$$F$$
**Firestore Storage**

Piattaforma offerta da Google che permette a siti web ed applicazioni di salvare i dati dell'utente in diversi formati. Consente upload e download sicuri, anche con una connessione di scarsa qualità.

**Framework**

Un framework è un'astrazione software, in cui è scritto del codice che fornisce funzionalità generiche e atte a essere estese, scrivendo così del software specifico. Può includere vari componenti, come librerie, compilatori ed  $API_G$ , tutte atte a migliorare il processo di sviluppo software.

**Freeling**

Software specializzato nel Part of Speech (PoS) tagging $_G$ . Il software sfrutta delle tecniche di apprendimento automatico supervisionato per tale classificazione.

**Front end**

Il front end è la parte di un'applicazione con la quale l'utente interagisce direttamente, responsabile dell'acquisizione dei dati di ingresso e per la loro elaborazione. Tali dati sono poi utilizzabili dal back end $_G$ .

---

 $G$ **Gamification**

Riutilizzo di concetti ed elementi tipici dei giochi in applicazioni di diverso contesto. Utilizzando questi principi si ottengono sia maggiore coinvolgimento degli utenti dell'applicazione, sia maggiore produttività ed organizzazione nel lavoro.

**Gantt**

Il diagramma di Gantt è uno strumento di supporto alla gestione dei progetti. E' usato principalmente nelle attività di project management. L'asse orizzontale rappresenta l'arco temporale totale del progetto, suddiviso in fasi incrementali; l'asse verticale rappresenta le mansioni o attività che costituiscono il progetto.

**Gigacore**

Boilerplate<sub>G</sub> per lo sviluppo di applicazioni che fanno uso di React<sub>G</sub>, Redux<sub>G</sub> e Sass<sub>G</sub>.

**GitLab**

Sistema di gestione repository online, che integra servizi quali issue tracking e l'integrazione di pipeline volte a continuous delivery<sub>G</sub> e continuous integration<sub>G</sub>.

$H$ **Hunpos**

Software specializzato in Part of Speech (PoS) tagging<sub>G</sub>. Il software sfrutta delle tecniche di apprendimento automatico supervisionato per tale classificazione.

*I*

*J*

## **JSON**

JavaScript Object Notation. E' una sintassi per il salvataggio e scambio di dati.

$K$

$L$ **Lint**

Programma che analizza il codice sorgente per segnalare errori, bug, costrutti sospetti e/o mancato adempimento a uno stile.

**Linter**

Sinonimo: vedi  $\text{Lint}_G$ .

---

*M***Milestone**

Indica il raggiungimento di obiettivi stabiliti con importanza strategica.

**Modello a V**

Modello di sviluppo software esteso dal modello a cascata. Pone la scrittura del test del software nelle fasi iniziali dello sviluppo e prima della codifica.

**Modello di sviluppo**

Detto anche modello di sviluppo software o modello dei processi software. Si tratta di un sistema di processi cooperativi e atti ad aiutare gli sviluppatori a realizzare il software. I processi sono scelti in base al modo di operare del team di sviluppo e al tipo di progetto che si intraprende.



$N$ **Node.js**

Ambiente open-source per l'esecuzione di codice Javascript a run-time, al di fuori dei browser. Ciò permette l'esecuzione di codice Javascript server-side.

*O***Open-Source**

Software non protetto da copyright, che segue la filosofia della produzione collaborativa e dell'accesso pubblico al codice sorgente.

---

 $P$ **Partizione di equivalenza**

Nei test case, insieme di valori che sono trattati uniformemente: valori validi, valori di contorno della partizione o valore non validi.

**Peer-to-peer**

Nelle reti telematiche, architettura in cui tutti i computer connessi svolgono la funzione sia di client che di server.

**Plug-in**

Componente aggiuntivo che può essere aggiunto a un'altro software per ampliarne le funzionalità. Di solito può essere eseguito in modo indipendente.

**PoS tagging**

Interpretazione di un testo etichettando ciascuna parola con il suo significato grammaticale.

**Product Baseline**

Baseline del progetto di ingegneria del software in cui il prodotto è stato realizzato e funziona, ma non è ancora pronto per il rilascio.

**Proof of Concept**

È un prototipo software che dimostra che il progetto è fattibile conformemente alle richieste.

**Pull request**

Funzionalità di GitHub. Permette di comunicare agli altri membri del team i cambiamenti fatti, i quali sono stati pushati sulla repository<sub>G</sub> di GitHub.

**Python**

Linguaggio di programmazione general purpose, interpretato, ad alto livello. Gode di ampia diffusione per la sua semplicità e l'ampia collezione di librerie che lo supportano.

Q

---

## *R*

### **React**

Libreria open source per JavaScript per la creazione di interfacce grafiche e la gestione delle interazioni in ambito web.

### **Redmine**

Piattaforma open source che permette la gestione di più progetti contemporaneamente, offrendo funzionalità di project management e issue tracking.

### **Redux**

Libreria open source JavaScript per la gestione degli stati di  $\text{React}_G$ .

### **Repository**

In generale, locazione di salvataggio dei dati. Nei sistemi di versionamento è una struttura dati più complessa, contenente metadati e operazioni per maneggiarla.

### **REST**

Representational State Transfer, stile architetturale atto alla creazione di servizi web. Si espongono operazioni per manipolare i servizi, appoggiandosi a protocolli web esistenti, tipicamente HTTP.

### **Rete Bayesiana**

Modello grafico probabilistico che rappresenta un insieme di variabili stocastiche con le loro dipendenze condizionali.

---

## *S*

### **Sass**

Sigla per "Syntactically Awesome Style Sheets", è un linguaggio interpretato o compilato in CSS dotato di una sintassi che semplifica ed estende le funzionalità offerte da CSS. Supporta anche la nuova sintassi SCSS<sub>G</sub>.

### **SCSS**

Sintassi per l'estensione del linguaggio CSS, che ne aumenta le funzionalità e l'espressività.

### **Server-side**

Lato server: fa riferimento a operazioni compiute dal server in un ambito client-server.

### **Shortcut**

Scorciatoia: una maniera alternativa e più veloce per svolgere la stessa attività.

### **Slack**

Sistema di messaggistica cloud-based orientato al mondo lavorativo. Permette la creazione di diverse aree di lavoro, suddivise in canali che trattano un particolare topic.

### **Smart contract**

Protocolli per facilitare, attuare e verificare la negoziazione di un contratto in versione digitale. Permettono di ottenere lo stesso valore di un contratto reale senza l'ausilio di un garante esterno. Le transazioni che avvengono con questo protocollo sono tracciabili e irreversibili. Uno smart contract rappresenta del codice che può essere eseguito.

### **Snake case**

Convenzione per i nomi dei file. Utilizza "underscore" come carattere separatore. Nell'accezione del team, le lettere sono tutte minuscole e le preposizioni non si omettono.

### **Solidity**

Linguaggio di programmazione utilizzato per lo sviluppo di smart contracts<sub>G</sub> eseguibili in diverse blockchain<sub>G</sub>.

### **SonarQube**

Piattaforma open source che permette di svolgere una continua revisione del codice caricato in una repository, svolgendo analisi statica per individuare possibili bug e vulnerabilità in sicurezza.

### **Staging**

Fase successiva alla fase di sviluppo del software che consiste nell'assemblare tutte le componenti e testarle in un server svolgendo verifiche e test. Se il software possiede il comportamento desiderato allora può passare alla fase di produzione.

**Stand-up**

Riunioni giornaliere di breve durata in cui le persone che partecipano ad un progetto espongono i progressi raggiunti e le attività pianificate per la giornata. Sono uno degli elementi principali dei metodi di sviluppo Agile<sub>G</sub>.

**Stub**

Nel test di unità, lo stub è un'unità fantoccio, usata a supporto l'interfaccia dell'unità da testare. Ad esempio, se l'unità da testare fa una chiamata a una funzione non ancora definita, lo stub funge funzione chiamata e ritorna un risultato conforme senza eseguire calcoli.

**Surge.sh**

Web server che offre il servizio di hosting per siti web statici.

$T$

**Technology Baseline**

Definisce le tecnologie, i framework e le librerie selezionate per lo sviluppo del prodotto.

**Telegram**

Servizio di messaggistica istantanea e VoIP cloud-based.

**Time Series**

Insieme di variabili casuali ordinate rispetto al tempo, esprimendo quindi la dinamica di un certo fenomeno nel tempo.

**Twelve-Factor App**

Metodologia per la creazione di software come applicazioni di servizio. Queste best practice sono progettate per consentire alle applicazioni di essere costruite con portabilità e resilienza una volta implementate sul Web.



U**UML**

Unified Modeling Language è un linguaggio standard per specificare, visualizzare, costruire e documentare gli artefatti di un sistema software.

V

W**Workflow**

Modello composto da un insieme di regole che tutti rispettano per lavorare con fluidità.

X

Y

Z