



## Glossario

### Informazioni sul documento

<b>Versione</b>	4.0.0
<b>Redazione</b>	Francesco Agostini
<b>Verifica</b>	Michael Munaro
<b>Approvazione</b>	Davide Trevisan
<b>Uso</b>	Esterno
<b>Lista di Distribuzione</b>	ScalateKids Prof. Tullio Vardanega Prof. Riccardo Cardin



## Diario delle modifiche

Versione	Autore	Ruolo	Data	Descrizione
4.0.0	Davide Trevisan	Responsabile	2016-06-15	Approvazione documento
3.1.0	Michael Munaro	Verificatore	2016-06-10	Verifica definizioni aggiunte
3.0.1	Francesco Agostini	Analista	2016-06-09	Incremento definizioni
3.0.0	Francesco Agostini	Responsabile	2016-05-16	Approvazione documento
2.1.0	Giacomo Vanin	Verificatore	2016-05-15	Verifica definizioni aggiunte
2.0.2	Andrea Giacomo Baldan	Analista	2016-05-15	Incremento definizioni e aggiunta indice
2.0.1	Andrea Giacomo Baldan	Analista	2016-04-30	Incremento definizioni
2.0.0	Giacomo Vanin	Responsabile	2016-04-08	Approvazione documento
1.4.0	Andrea Giacomo Baldan	Verificatore	2016-04-07	Verifica definizioni aggiunte
1.3.1	Michael Munaro	Analista	2016-04-05	Incremento definizioni
1.3.0	Marco Boseggia	Verificatore	2016-03-23	Verifica definizioni
1.2.1	Alberto De Agostini	Analista	2016-03-21	Aggiunte definizioni
1.2.0	Michael Munaro	Verificatore	2016-03-08	Verifica ultime definizioni aggiunte
1.1.2	Andrea Giacomo Baldan	Analista	2016-03-07	Aggiunte definizioni
1.1.1	Alberto De Agostini	Analista	2016-02-28	Aggiunte definizioni
1.1.0	Davide Trevisan	Verificatore	2016-02-19	Verifica ultime modifiche
1.0.1	Michael Munaro	Analista	2016-02-18	Cancellazione sezione 1 e tolto titolo sezione 2
1.0.0	Andrea Giacomo Baldan	Responsabile	2016-01-20	Approvazione documento
0.3.0	Francesco Agostini	Verificatore	2016-01-19	Verifica definizioni
0.2.3	Michael Munaro	Analista	2016-01-18	Aggiunte definizioni
0.2.2	Marco Boseggia	Analista	2016-01-16	Aggiunte definizioni
0.2.1	Giacomo Vanin	Analista	2016-01-10	Aggiunte definizioni
0.2.0	Alberto De Agostini	Verificatore	2016-01-03	Verifica definizioni
0.1.2	Davide Trevisan	Analista	2016-01-02	Aggiunte definizioni
0.1.1	Michael Munaro	Analista	2015-12-28	Aggiunte definizioni
0.1.0	Francesco Agostini	Verificatore	2015-12-26	Verifica definizioni
0.0.3	Giacomo Vanin	Analista	2015-12-23	Aggiunte definizioni
0.0.2	Michael Munaro	Analista	2015-12-16	Prima stesura voci riscontrate
0.0.1	Andrea Giacomo Baldan	Amministratore	2015-12-15	Creazione scheletro del documento



## Indice

<b>A</b>	<b>3</b>
Account . . . . .	3
Ack (diminutivo di acknowledgment) . . . . .	3
Actormodel . . . . .	3
Akka . . . . .	3
Amazon DynamoDB . . . . .	3
Ant . . . . .	3
Architettura . . . . .	3
Array . . . . .	3
Attività . . . . .	3
Attore . . . . .	3
<b>B</b>	<b>4</b>
Backup . . . . .	4
Banner . . . . .	4
Bash . . . . .	4
Beacon . . . . .	4
Bit . . . . .	4
Brainstorming . . . . .	4
Branch . . . . .	4
BSON(Binary javaScript Object Notation) . . . . .	4
Bug . . . . .	4
Build . . . . .	5
Bundled Software . . . . .	5
Business Logic . . . . .	5
BV (Budget Variance) . . . . .	5
Byte . . . . .	5
Bytecode, Java . . . . .	5
<b>C</b>	<b>6</b>
CamelCase (Notazione a Cammello) . . . . .	6
CAP Theorem . . . . .	6
Ciclo di Deming . . . . .	6
CLI (Command Line Interface) . . . . .	6
Client . . . . .	6
Clientactor . . . . .	6
Cluster . . . . .	6
Codice . . . . .	6
Collaboratore . . . . .	6
Collezione . . . . .	7
Command Pattern . . . . .	7
Commit . . . . .	7



Concorrenza . . . . .	7
Container docker . . . . .	7
Controller . . . . .	7
Console . . . . .	7
CR (Carriage Return) . . . . .	7
Crash . . . . .	7
Cursore . . . . .	7
<b>D</b>	<b>8</b>
Database . . . . .	8
Database Management System (DBMS) . . . . .	8
DBMS (DataBase Management System) . . . . .	8
Deserializzazione . . . . .	8
Design Pattern . . . . .	8
Dizionario (o Array Associativo) . . . . .	8
Diagramma dei package . . . . .	8
Diagramma delle classi . . . . .	8
Diagramma di attività . . . . .	8
Diagramma di Gantt . . . . .	8
Diagramma di sequenza . . . . .	9
Docker . . . . .	9
Domain Specific Language (DSL) . . . . .	9
Drive, Google . . . . .	9
Driver . . . . .	9
DSL (Domain-Specific Language) . . . . .	9
<b>E</b>	<b>10</b>
<b>F</b>	<b>11</b>
Façade . . . . .	11
File . . . . .	11
Flag . . . . .	11
Form . . . . .	11
Forum (internet) . . . . .	11
Framework . . . . .	11
Funzionale, programmazione . . . . .	11
<b>G</b>	<b>12</b>
Git . . . . .	12
GitHub . . . . .	12
<b>H</b>	<b>13</b>
<b>I</b>	<b>14</b>
IDE (Integrated Development Environment) . . . . .	14
IEC (International Electrotechnical Commission) . . . . .	14



Instant messaging . . . . .	14
Integrazione Continua . . . . .	14
Invoker . . . . .	14
IoT (Internet of Things) . . . . .	14
ISO (International Organization for Standardization) . . . . .	14
Item . . . . .	14
<b>J</b>	<b>15</b>
JSON (JavaScript Object Notation) . . . . .	15
JVM . . . . .	15
<b>K</b>	<b>16</b>
Key-value Database . . . . .	16
<b>L</b>	<b>17</b>
Leader Election . . . . .	17
LF (Line Feed) . . . . .	17
Link . . . . .	17
Load Balancing . . . . .	17
lowerCamelCase . . . . .	17
<b>M</b>	<b>18</b>
Macro . . . . .	18
Mailing List . . . . .	18
Main Actor . . . . .	18
Manager . . . . .	18
Major Release . . . . .	18
Merge . . . . .	18
Model . . . . .	18
Modello ad attori . . . . .	18
Modello relazionale . . . . .	18
MVC (Model-View-Controller) . . . . .	18
<b>N</b>	<b>20</b>
Ninja . . . . .	20
Nodo . . . . .	20
NoSQL . . . . .	20
<b>O</b>	<b>21</b>
Observer . . . . .	21
OOP (Object Oriented Programming) . . . . .	21
Organigramma . . . . .	21
<b>P</b>	<b>22</b>
Package . . . . .	22
Parsing . . . . .	22
Path . . . . .	22



Payload	22
PDCA	22
PDF	22
Peer-to-peer	22
Persistence Actor	22
Persistenza	22
Pickle	22
Plugin	23
Programamzione Funzionale	23
Prompt	23
Prototipo	23
Pull	23
Push	23
Push Model	23
<b>Q</b>	<b>24</b>
Query	24
<b>R</b>	<b>25</b>
Range	25
Reactive	25
Receiver	25
Refactoring	25
REPL (read-eval-print loop)	25
Repository (Software Repository)	25
RESP (REdis Protocol SPecification)	25
Round Robin	25
Routing	25
<b>S</b>	<b>26</b>
SBT (Scala Build Tool)	26
Scala	26
Scalabilità Orizzontale (Scale Out)	26
Scaladoc	26
Scale Out	26
Script	26
Section	26
Serializzazione	26
Server	26
Shell	26
Single Responsibility Principle	26
Singleton	27
Sistema di Ticketing	27
Smartphone	27
Socket	27
Spring	27



Spring Shell . . . . .	27
SQL (Structured Query Language) . . . . .	27
Stack . . . . .	27
Storefinder . . . . .	27
Storekeeper . . . . .	28
Strategy, pattern . . . . .	28
SV (Schedule Variance) . . . . .	28
SVN . . . . .	28
<b>T</b>	<b>29</b>
Task . . . . .	29
TCP (Transmission Control Protocol) . . . . .	29
tcpserver . . . . .	29
Template . . . . .	29
Thread-Safe . . . . .	29
Topic . . . . .	29
Top-Down . . . . .	29
Travis-CI . . . . .	29
<b>U</b>	<b>30</b>
UI (User Interface) . . . . .	30
UML (Unified Modeling Language) . . . . .	30
Unicode . . . . .	30
UpperCamelCase . . . . .	30
URL (Uniform Resource Locator) . . . . .	30
Userkeeper . . . . .	30
Username(Nome Utente) . . . . .	30
UTF-8 (Unicode Transformation Format, 8 bit) . . . . .	30
<b>V</b>	<b>31</b>
Versionamento . . . . .	31
View . . . . .	31
Virtual Machine . . . . .	31
Visibilità . . . . .	31
VoIP (Voice over IP) . . . . .	31
<b>W</b>	<b>32</b>
Warehouseman . . . . .	32
Whiteboarding . . . . .	32
Wiki . . . . .	32
<b>X</b>	<b>33</b>
<b>Y</b>	<b>34</b>
<b>Z</b>	<b>35</b>



## A

**Account:** Insieme di funzionalità, strumenti e contenuti attribuiti ad un nome utente in determinati contesti operativi, spesso in siti web o per usufruire di determinati servizi su Internet.

**Ack (diminutivo di acknowledgment):** Termine che viene usato nell'informatica per indicare un riscontro positivo.

**Actormodel:** Modello matematico per la gestione della concorrenza computazionale proposto nel 1973 da Carl Hewitt. All'interno del modello, gli "attori" vengono trattati come primitive universali e possono scambiarsi messaggi tra di loro; in risposta ad un messaggio ricevuto, un attore può prendere decisioni, creare nuovi attori, inviare altri messaggi e determinare come rispondere al messaggio successivo.

**Akka:** È uno strumento per la semplificazione e costruzione di applicazioni concorrenti e distribuite sulla JVM (Java Virtual Machine). Akka supporta diversi modelli per la concorrenza prediligendo, però, l'uso del modello ad attori.

**Amazon DynamoDB:** Database proprietario di tipo NoSQL e completamente gestito da Amazon.com all'interno del suo portfolio 'Amazon Web Services'.

**Ant:** Apache Ant è un software per l'automazione del processo di build. È simile a make, ma scritto in Java ed è principalmente orientato allo sviluppo in Java. Ant è un progetto Apache, open source, ed è distribuito sotto licenza Apache.

**Architettura:** Per architettura informatica si intende l'insieme dei criteri e regole in base ai quali è progettato e realizzato un sistema informatico oppure un dispositivo facente parte di esso.

**Array:** Un array o vettore, in informatica, indica una struttura dati complessa, statica e omogenea. Si può immaginare un array come una sorta di contenitore, le cui caselle sono dette celle dell'array stesso. Ciascuna delle celle si comporta come una variabile tradizionale che rappresenta un elemento dell'array; tutte le celle sono variabili di uno stesso tipo preesistente, detto tipo base dell'array.

**Attività:** Aggregazione di *task* (vedi sotto).

**Attore:** vedi *Actormodel*





## B

---

**Backup:** In informatica con il termine backup si indica la replicazione, su un qualunque supporto di memorizzazione, di materiale informativo archiviato nella memoria di massa dei computer, siano essi personal computer, workstation o server, ma pure home computer e smartphone, al fine di prevenire la perdita definitiva dei dati in caso di eventi malevoli accidentali o intenzionali. Si tratta dunque di una misura di ridondanza fisica dei dati, tipica delle procedure di disaster recovery.

**Banner:** Banner è un termine in lingua inglese che significa bandiera, vessillo o striscione. Un esempio classico di banner è la striscia che solitamente compare all'inizio di una pagina web e che riporta il nome del sito.

**Bash:** è una shell testuale e un linguaggio di comando del progetto GNU usata nei sistemi operativi Unix e Unix-like, specialmente in GNU/Linux, ma disponibile anche per sistemi Microsoft Windows

**Beacon:** il beacon è un piccolo dispositivo BLE (acronimo per Bluetooth Low Energy, cioè Bluetooth a basso consumo energetico) che trasmette piccole quantità di dati ogni secondo o quasi. A livello di hardware, i beacon sono dispositivi BLE-compatibile che trasmettono dati sotto il protocollo "beacon". A livello software, i beacon sono messaggi inviati da questi dispositivi trasmettenti, che sono poi rilevati e processati da un dispositivo ricevente, come un'app mobile. Nel caso degli iBeacon vengono trasmessi i valori Major e Minor che insieme all'UUID (acronimo di Universally Unique Identifier), permettono l'identificazione di un iBeacon da parte dell'applicazione. (riferirsi a <https://it.wikipedia.org/wiki/iBeacon> e <http://www.beaconitaly.it/cose-ibeacon/>);

**Bit:** cifra binaria, ovvero uno dei due simboli del sistema numerico binario, classicamente chiamati zero (0) e uno (1).

**Brainstorming:** Metodo decisionale in cui la ricerca della soluzione di un problema è effettuata mediante sedute intensive di dibattito e confronto delle idee e delle proposte espresse dai partecipanti.

**Branch:** In computer science, il branch ('salto' in inglese) è un punto in cui in un processo viene alterato il sequenziale flusso delle istruzioni.

**JSON(Binary JavaScript Object Notation):** è un formato informatico di scambio dati utilizzato principalmente come magazzino dati e formato di trasferimento di rete nel database MongoDB. Si tratta di un formato binario per rappresentare strutture dati semplici e array associativi

**Bug:** in informatica, identifica un errore nella scrittura di un programma software, impedendone il corretto funzionamento e provocandone un comportamento imprevisto o comunque diverso da quello specificato dal



produttore. Meno comunemente, il termine bug può indicare un difetto di progettazione in un componente hardware. L'origine del nome (che in inglese significa "insetto") deriva da un malfunzionamento del computer Mark II nel 1947 dovuto ad una falena incastrata tra i circuiti.

**Build:** trasformazione del codice sorgente in software eseguibile (tramite compilazione e linking). Per estensione, si dice anche del prodotto finale della trasformazione.

**Bundled Software:** Si tratta di software distribuito con un altro prodotto come parte di un hardware o come parte di un gruppo di pacchetti software venduti assieme.

**Business Logic:** L'espressione logica di business (in inglese business logic) si riferisce a tutta quella logica applicativa che rende operativa un'applicazione cioè la parte o nucleo (core) di elaborazione. Con tale nome ci si riferisce quindi all'algoritmica che gestisce lo scambio di informazioni tra una sorgente dati (generalmente una base dati) deputata gestione della persistenza dei dati stessi da una parte e l'interfaccia utente attraverso la logica di presentazione e le elaborazioni intermedie sui dati estratti dall'altra. È un termine largamente utilizzato nella progettazione del software per individuare un componente software, un layer (o tier, cioè livello) di una architettura software, ecc.

**BV (Budget Variance):** la differenza tra il budget previsto in entrata per una determinata baseline e l'effettivo budget attualmente disponibile.

**Byte:** è una sequenza di bit (vedi sopra), il cui numero dipende dall'implementazione fisica della macchina sottostante.

**Bytecode, Java:** Il Java Bytecode è l'istruzione set della JVM (Java Virtual Machine). Ogni bytecode è composto da un byte (in certi casi due) che rappresenta l'istruzione e zero o più byte per il passaggio dei parametri.



## C

---

**CamelCase (Notazione a Cammello):** è la pratica nata durante gli anni settanta di scrivere parole composte o frasi unendo tutte le parole tra loro, ma lasciando le loro iniziali maiuscole. La prima lettera può essere sia maiuscola che minuscola. Esistono diverse varianti della CamelCase tra le quali ricordiamo la **UpperCamel-Case**, ovvero una notazione a cammello in cui la prima lettera è obbligatoriamente maiuscola, e la **lowerCamel-Case** dove la prima lettera è sempre minuscola.

**CAP Theorem:** In informatica teorica, il teorema CAP, noto anche come teorema di Brewer, afferma che è impossibile per un sistema informatico distribuito fornire simultaneamente tutte e tre le seguenti garanzie:[]

- *Coerenza*, tutti i nodi vedono gli stessi dati nello stesso momento);
- *Disponibilità*, la garanzia che ogni richiesta riceva una risposta su ciò che sia riuscito o fallito;
- *Tolleranza di partizione*, il sistema continua a funzionare nonostante arbitrarie perdite di messaggi.

Secondo il teorema, un sistema distribuito è in grado di soddisfare al massimo due di queste garanzie allo stesso tempo, ma non tutte e tre.

**Ciclo di Deming:** vedi PDCA.

**CLI (Command Line Interface):** detta semplicemente *riga di comando*, indica una tipologia di interfaccia utente caratterizzata da un'interazione di tipo testuale tra utente ed elaboratore.

**Client:** in informatica, indica una componente che accede ai servizi o alle risorse di un'altra componente detta server. In questo contesto si può quindi parlare di client riferendosi all'hardware oppure al software. Esso fa parte dunque dell'architettura logica di rete detta client-server.

**Clientactor:** nel contesto del capitolato da noi considerato, con Clientactor si intende un tipo di attore dedicato che regola la connessione da parte di un client esterno.

**Cluster:** In informatica un computer cluster, o più semplicemente un cluster (dall'inglese grappolo), è un insieme di computer connessi tra loro tramite una rete telematica. Lo scopo di un cluster è quello di distribuire una elaborazione molto complessa tra i vari computer componenti il cluster.

**Codice:** Testo di un algoritmo di un programma scritto in un linguaggio di programmazione da un programmatore.

**Collaboratore:** Nell'ambito di Actorbase, con collaboratore si intende un utente (autenticato) che ha visione in collezioni non create da lui stesso



**Collezione:** Nell'ambito di Actorbase, una collezione è una struttura dati di tipo chiave-valore

**Command Pattern:** Il Command pattern è uno dei design pattern che permette di isolare la porzione di codice che effettua un'azione (eventualmente molto complessa) dal codice che ne richiede l'esecuzione; l'azione è incapsulata nell'oggetto Command. L'obiettivo è rendere variabile l'azione del client senza però conoscere i dettagli dell'operazione stessa. Altro aspetto importante è che il destinatario della richiesta può non essere deciso staticamente all'atto dell'istanziamento del command ma ricavato a tempo di esecuzione.

**Commit:** Nel contesto di computer science e gestione di dati, commit si riferisce all'idea di rendere permanenti i cambiamenti effettuati nel repository locale.

**Concorrenza:** in informatica, il concetto di concorrenza, contrapposto a quello di sequenzialità, indica una caratteristica dei sistemi di elaborazione nei quali può verificarsi che un insieme di processi o sottoprocessi (thread) computazionali siano in esecuzione nello stesso istante. L'esecuzione contemporanea può condurre a interazione tra processi quando viene coinvolta una risorsa condivisa.

**Container docker:** I container docker impacchettano un pezzo di software in un filesystem completo, che contiene tutto ciò che è necessario per farlo funzionare: codice, strumenti di sistema, librerie di sistema e tutto ciò che può essere installato su un server. Questo garantisce che il software funzionerà sempre allo stesso modo, indipendentemente dall'ambiente in cui si troverà.

**Controller:** vedi *MVC (Model-View-Controller)*.

**Console:** vedi *CLI*.

**CR (Carriage Return):** Carattere di controllo (o meccanismo) utilizzato per riposizionare il device all'inizio della linea di testo.

**Crash:** Nel gergo informatico indica il blocco o la terminazione improvvisa, non richiesta e inaspettata di un programma in esecuzione, oppure il blocco completo dell'intero computer. La parola è un prestito dalla lingua inglese, nella quale ha il significato principale di grave incidente, collisione.

**Cursore:** Nell'ambito dell'interfaccia utente in informatica, è un simbolo che appare sullo schermo con lo scopo di indicare la posizione in cui viene inserito il testo digitato sulla tastiera o su cui ha effetto la pressione dei pulsanti del mouse o di qualunque altra periferica che possa muovere il cursore stesso.



## D

---

**Database:** In informatica, il termine database (in italiano *base di dati*) indica un insieme organizzato di dati. Le informazioni contenute in un database sono strutturate e collegate tra loro secondo un particolare modello logico scelto dal progettista del database (ad esempio relazionale, gerarchico, key-value, ...). Gli utenti si interfacciano con i database attraverso i cosiddetti *query language* e grazie a particolari applicazioni software dedicate.

**Database Management System (DBMS):** è un sistema software progettato per consentire la creazione, la manipolazione, da parte di un amministratore DBA (Database Admin) e l'interrogazione efficiente (da parte di uno o più utenti client) di database, per questo detto anche "gestore o motore del database", e ospitato su architettura hardware dedicata (server) oppure su semplice computer.

**DBMS (DataBase Management System):** è un sistema software progettato per consentire la creazione, la manipolazione (da parte di un amministratore) e l'interrogazione efficiente di un database. È ospitato su architettura hardware dedicata (server) oppure su semplice computer.

**Deserializzazione:** In informatica, nella programmazione orientata agli oggetti, deserializzare un oggetto significa ricostruirlo a partire dalla sua rappresentazione binaria ottenuta da un file o da un canale di rete.

**Design Pattern:** In ingegneria del software, un design pattern descrive una soluzione generale a un problema di progettazione ricorrente, gli attribuisce un nome, astrae e identifica gli aspetti principali della struttura utilizzata per la soluzione del problema, identifica le classi e le istanze partecipanti e descrive quando e come può essere applicato. In breve definisce un problema, i contesti tipici in cui si trova e la soluzione ottimale allo stato dell'arte.

**Dizionario (o Array Associativo):** Struttura dati che contiene una collezione di oggetti (o records). Questi oggetti contengono diversi campi, ognuno dei quali contiene dei dati. È possibile accedere, gestire e salvare i diversi records tramite una chiave che li identifica in maniera univoca.

**Diagramma dei package:** Un diagramma dei package è un diagramma che illustra come gli elementi di modellazione sono organizzati in package e le relazioni (dipendenze) tra i package.

**Diagramma delle classi:** Un diagramma delle classi descrive il tipo degli oggetti facenti parte di un sistema e le varie tipologie di relazioni statiche tra di essi. I diagrammi delle classi mostrano le proprietà e le operazioni di una classe e i vincoli che si applicano alla classe e alle relazioni tra classi.

**Diagramma di attività:** Descrive la sequenza delle attività e supporta un comportamento condizionale e parallelo. Particolarmente utile per descrivere *workflow process* e *parallel process*.



**Diagramma di Gantt:** È uno strumento di supporto alla gestione dei progetti. Si tratta, sostanzialmente, di un diagramma costruito a partire da un asse orizzontale, a rappresentazione dell'arco temporale totale (o parziale) del progetto, e da un asse verticale che rappresenta le mansioni o attività che costituiscono il progetto. Lo scopo principale del diagramma di Gantt è di dislocare temporalmente le attività per poterne rappresentare la durata, la sequenzialità e il parallelismo e poter confrontare le stime con i progressi.

**Diagramma di sequenza:** Diagramma previsto dall'UML utilizzato per descrivere uno scenario. Uno scenario è una determinata sequenza di azioni in cui tutte le scelte sono state già effettuate; in pratica nel diagramma non compaiono scelte, né flussi alternativi. Normalmente da ogni diagramma di attività (vedi sopra) sono derivati uno o più diagrammi di sequenza.

**Docker:** è un progetto open-source che automatizza il deployment di applicazioni all'interno di container software, fornendo un'astrazione addizionale grazie alla virtualizzazione a livello di sistema operativo di Linux.

**Domain Specific Language (DSL):** È un linguaggio di programmazione o un linguaggio di specifica dedicato a particolari problemi di un dominio, a una particolare tecnica di rappresentazione e/o a una particolare soluzione tecnica.

**Drive, Google:** Google Drive è un servizio, in ambiente cloud computing, di memorizzazione e sincronizzazione online introdotto da Google il 24 aprile 2012. Il servizio comprende il file hosting, il file sharing e la modifica collaborativa di documenti

**Driver:** componente attiva fittizia per pilotare i test.

**DSL (Domain-Specific Language):** nello sviluppo software e nell'ingegneria di dominio, il DSL è un linguaggio di programmazione o un linguaggio di specifica dedicato a particolari problemi di un dominio, a una particolare tecnica di rappresentazione e/o a una particolare soluzione tecnica.



## E

---



## F

---

**Façade:** Letteralmente façade significa "facciata", ed infatti nella programmazione ad oggetti indica un oggetto che permette, attraverso un'interfaccia più semplice, l'accesso a sottosistemi che espongono interfacce complesse e molto diverse tra loro, nonché a blocchi di codice complessi. L'utilizzo del pattern façade (qui realizzato attraverso la classe Facade) permette di nascondere la complessità dell'operazione.

**File:** In informatica, viene utilizzato per riferirsi a un contenitore di informazioni/dati in formato digitale, tipicamente presenti su un supporto digitale di memorizzazione opportunamente formattato in un determinato file system. Le informazioni scritte/codificate al suo interno sono leggibili solo tramite uno specifico software in grado di effettuare l'operazione.

**Flag:** In informatica, un flag (parola inglese che significa bandiera) è una variabile (solitamente booleana) che può assumere solo due stati ("vero" o "falso", "on" e "off", "1" e "0", "acceso" e "spento") e che segnala, con il suo valore, se un dato evento si è verificato oppure no, o se il sistema è in un certo stato oppure no

**Form:** In informatica, un form è un termine usato per indicare l'interfaccia di un'applicazione che consente all'utente client di inserire e inviare al web server uno o più dati liberamente digitati dallo stesso.

**Forum (internet):** Il termine "forum", nell'ambito dell'informatica, è utilizzato per indicare l'insieme delle sezioni di discussione in una piattaforma informatica dove gli utenti intrattengono conversazioni postando messaggi. Un forum può essere di tipo gerarchico e con struttura ad albero in quanto possono esservi al suo interno diversi subforum, ognuno dei quali con un proprio topic (argomento di discussione).

**Framework:** Un framework, termine della lingua inglese che può essere tradotto come intelaiatura o struttura, 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. Un framework è definito da un insieme di classi astratte e dalle relazioni tra esse. Istanziare un framework significa fornire un'implementazione delle classi astratte.

**Funzionale, programmazione:** In informatica la programmazione funzionale è un paradigma di programmazione in cui il flusso di esecuzione del programma assume la forma di una serie di valutazioni di funzioni matematiche. Il punto di forza principale di questo paradigma è la mancanza di effetti collaterali (side-effect) delle funzioni, il che comporta una più facile verifica della correttezza e della mancanza di bug del programma e la possibilità di una maggiore ottimizzazione dello stesso. La programmazione funzionale pone, dunque, una maggior attenzione sulla definizione di funzioni, rispetto ai paradigmi procedurali e imperativi, che invece prediligono la specifica di una sequenza di comandi da eseguire.





## G

---

**Git:** Git è un sistema software di controllo di versione distribuito, creato da Linus Torvalds nel 2005. Esso permette, dunque, di tenere traccia delle modifiche e delle versioni apportate al codice sorgente di un software, senza la necessità di dover utilizzare un server centrale. Con questo sistema gli sviluppatori possono collaborare individualmente e parallelamente offline su di un proprio ramo (*branch*) di sviluppo, registrare le proprie modifiche (*commit*) ed in seguito condividerle o unirle (*merge*) a quelle di altri, il tutto, come detto in precedenza, senza bisogno del supporto di un server.

**GitHub:** Servizio Web di hosting per lo sviluppo di progetti software, che usa il sistema di controllo versione Git.



# H

---



# I

---

**IDE (Integrated Development Environment):** Un *ambiente di sviluppo integrato* è un software che, in fase di programmazione, aiuta i programmatori nello sviluppo del codice sorgente di un programma. Spesso l'IDE aiuta lo sviluppatore segnalando errori di sintassi del codice direttamente in fase di scrittura, oltre a tutta una serie di strumenti e funzionalità di supporto alla fase di sviluppo e debugging.

**IEC (International Electrotechnical Commission):** È un'organizzazione internazionale per la definizione di standard in materia di elettricità, elettronica e tecnologie correlate, fondata a Londra nel 1906.

**Instant messaging:** È una categoria di sistemi di comunicazione in tempo reale in rete, tipicamente Internet o una rete locale, che permette ai suoi utilizzatori lo scambio di brevi messaggi.

**Integrazione Continua:** Nell'ingegneria del software, l'integrazione continua (*continuous integration* in inglese, spesso abbreviato in CI) è una pratica che si applica in contesti in cui lo sviluppo del software avviene attraverso un sistema di versioning. Consiste nell'allineamento frequente dagli ambienti di lavoro degli sviluppatori verso l'ambiente condiviso (*mainline*).

**Invoker:** Uno dei termini associati al *Command Pattern* (vedi sopra); indica un oggetto che sa come eseguire un comando e opzionalmente può tenere traccia del numero di volte in cui il comando è stato eseguito.

**IoT (Internet of Things):** è un neologismo riferito all'estensione di Internet al mondo degli oggetti e dei luoghi concreti. L' *Internet delle cose* è vista come una possibile evoluzione dell'uso della Rete: gli oggetti si rendono riconoscibili e acquisiscono intelligenza grazie al fatto di poter comunicare dati su se stessi e accedere ad informazioni aggregate da parte di altri.

**ISO (International Organization for Standardization):** È la più importante organizzazione a livello mondiale per la definizione di norme tecniche fondata nel 1947 a Ginevra, in Svizzera. L'ISO coopera strettamente con l'IEC (vedi sopra), responsabile per la standardizzazione degli equipaggiamenti elettrici. Le norme ISO sono numerate e hanno un formato del tipo ISO nnnn:yyyy - titolo, dove nnnn è il numero della norma, yyyy l'anno di pubblicazione e titolo è una breve descrizione della norma.

**Item:** con item si intende una entry di una *collezione* (vedi sopra)



## J

---

**JSON (JavaScript Object Notation):** è un formato adatto all'interscambio di dati fra applicazioni client-server.

**JVM:** La Java Virtual Machine è il componente della piattaforma Java che esegue i programmi tradotti in byte-code dopo una prima compilazione.



## K

---

**Key-value Database:** È un paradigma di archiviazione di dati progettato per archiviare, recuperare e gestire array associativi (vedi *Dizionario*).



# L

---

**Leader Election:** L'algoritmo "Leader Election" viene eseguito nei sistemi distribuiti per stabilire il coordinatore a cui i processi fanno riferimento per un servizio. Se il coordinatore si guasta, l'intero sistema si blocca. Per risolvere il problema, occorre eseguire un algoritmo di elezione in cui i processi ancora attivi eleggono un nuovo coordinatore.

**LF (Line Feed):** carattere di controllo utilizzato per terminare le linee di testo nei sistemi Mac OS X e versioni successive, Unix e le loro varianti.

**Link:** speciale rimando contenuto all'interno di un ipertesto.

**Load Balancing:** tecnica informatica utilizzata nell'ambito dei sistemi informatici che consiste nel distribuire il carico di elaborazione di uno specifico servizio, ad esempio la fornitura di un sito web, tra più server. Si aumentano in questo modo la scalabilità e l'affidabilità dell'architettura nel suo complesso.

**lowerCamelCase:** vedi *CamelCase*.



# M

---

**Macro:** in informatica indica una procedura, ovvero un insieme di comandi o istruzioni, tipicamente ricorrente durante l'esecuzione di un programma.

**Mailing List:** è un servizio/strumento offribile da una rete di computer verso vari utenti e costituito da un sistema organizzato per la partecipazione di più persone ad una discussione asincrona o per la distribuzione di informazioni utili agli interessati/iscritti attraverso l'invio di email ad una lista di indirizzi di posta elettronica di utenti iscritti.

**Main Actor:** nel contesto del capitolato da noi considerato, con main si intende uno storefinder principale che funge da punto di accesso al database.

**Manager:** nel contesto del capitolato da noi considerato, con manager si intende un particolare tipo di attore che riceve le richieste di gestione degli attori di tipo storekeeper. In particolare, questi attori sono responsabili della gestione del numero massimo di coppie chiave/valore contenute in un'istanza di storekeeper.

**Major Release:** con major release di un software si intende una release che non è puramente una revisione o una release atta a correggere dei bug, ma che comunque contiene dei cambiamenti sostanziali.

**Merge:** mescolare, unire.

**Model:** vedi *MVC (Model-View-Controller)*.

**Modello ad attori:** vedi *Actormodel*.

**Modello relazionale:** modello logico di rappresentazione o strutturazione dei dati di un database implementato su sistemi di 'gestione di basi di dati' (DBMS). Si basa sulla teoria degli insiemi e sulla logica del primo ordine ed è strutturato intorno al concetto matematico di relazione (detta anche tabella). Per il suo trattamento ci si avvale di strumenti quali il calcolo relazionale e l'algebra relazionale.

**MVC (Model-View-Controller):** in informatica, è un pattern architetturale molto diffuso nello sviluppo di sistemi software, in particolare nell'ambito della programmazione orientata agli oggetti. Il pattern è basato sulla separazione dei compiti fra i componenti software che interpretano tre ruoli principali:

- il *model* fornisce i metodi per accedere ai dati utili all'applicazione;
- il *view* visualizza i dati contenuti nel model e si occupa dell'interazione con utenti e agenti;



- il *controller* riceve i comandi dell'utente (in genere attraverso il view) e li attua modificando lo stato degli altri due componenti.

Questo schema, fra l'altro, implica anche la tradizionale separazione fra la logica applicativa (in questo contesto spesso chiamata "logica di business"), a carico del controller e del model, e l'interfaccia utente a carico del view.





## N

---

**Ninja:** nel contesto del capitolato da noi considerato, con storefinder si intende un particolare tipo di attore che permette di mantenere consistente il database anche in caso di caduta di uno storekeeper. Essi si trovano su nodi differenti dai rispettivi storekeeper e sono in continuo aggiornamento con loro.

**Nodo:** In informatica e telecomunicazioni un nodo è un qualsiasi dispositivo del sistema in grado di comunicare con gli altri dispositivi che fanno parte della rete.

**NoSQL:** è un movimento che promuove sistemi software dove la persistenza dei dati è caratterizzata dal fatto di non utilizzare il *modello relazionale* (vedi sopra), solitamente utilizzato dai database tradizionali. L'espressione NoSQL fa riferimento al linguaggio SQL, il più comune tra i linguaggi per l'interrogazione dei database relazionali, qui preso a simbolo dell'intero paradigma relazionale. Molto spesso nei linguaggi NoSQL viene sacrificato in favore delle prestazioni il paradigma ACID (atomicità, consistenza, isolamento, durabilità) in quanto, per il teorema CAP, noto anche come teorema di Brewer, è impossibile per un sistema informatico distribuito, fornire simultaneamente tutte e tre le seguenti garanzie: Coerenza (tutti i nodi vedono gli stessi dati nello stesso momento) Disponibilità (la garanzia che ogni richiesta riceva una risposta su ciò che sia riuscito o fallito) Tolleranza di partizione (il sistema continua a funzionare nonostante arbitrarie perdite di messaggi) Secondo il teorema, un sistema distribuito è in grado di soddisfare al massimo due di queste garanzie allo stesso tempo, ma non tutte e tre.



# O

---

**Observer:** L'Observer pattern è un design pattern utilizzato per tenere sotto controllo lo stato di diversi oggetti. È un pattern intuitivamente utilizzato come base architetture di molti sistemi di gestione di eventi. Si basa su uno o più oggetti, chiamati osservatori o observer, che vengono registrati per gestire un evento che potrebbe essere generato dall'oggetto "osservato".

**OOP (Object Oriented Programming):** paradigma di programmazione che prevede di raggruppare in una zona circoscritta del codice sorgente (chiamata classe) la dichiarazione delle strutture dati e delle procedure che operano su di esse. Le classi costituiscono dei modelli astratti, che a tempo di esecuzione vengono invocate per istanziare o creare oggetti software relativi alla classe invocata.

**Organigramma:** in un'organizzazione aziendale, con il termine organigramma si intende una formalizzazione di quali sono gli organi e le loro relazioni gerarchiche o funzionali all'interno di un ambiente di lavoro.



## P

---

**Package:** un pacchetto (o pacchetto di software), in informatica è una serie di programmi che si distribuiscono congiuntamente.

**Parsing:** Processo che analizza un flusso continuo di dati in ingresso (input) (letti per esempio da un file o una tastiera) in modo da determinare la sua struttura grazie ad una data grammatica formale. Un parser è un programma che esegue questo compito.

**Path:** in informatica indica la posizione di file e directory presenti all'interno della gerarchia di un file system. Il concetto di percorso si basa sulla struttura gerarchica (ad albero) del file system

**Payload:** In informatica, indica la parte di un flusso di dati che rappresenta il contenuto informativo.

**PDCA:** il ciclo di Deming (chiamato anche ciclo di PDCA - "plan-do-check-act") è un modello studiato per il miglioramento continuo della qualità in un'ottica a lungo raggio. Serve per promuovere una cultura della qualità che è tesa al miglioramento continuo dei processi e all'utilizzo ottimale delle risorse. Questo strumento parte dall'assunto che per il raggiungimento del massimo della qualità sia necessaria la costante interazione tra ricerca, progettazione, test, produzione e vendita. Per migliorare la qualità e soddisfare il cliente, le quattro fasi devono ruotare costantemente, tenendo come criterio principale la qualità.

**PDF:** il Portable Document Format, comunemente abbreviato in PDF, è un formato di file basato su un linguaggio di descrizione di pagina sviluppato da Adobe Systems nel 1993 per rappresentare documenti in modo indipendente dall'hardware e dal software utilizzati per generarli o per visualizzarli.

**Peer-to-peer:** peer-to-peer (P2P) o rete paritaria, in informatica, è un'espressione che indica un'architettura logica di rete informatica in cui i nodi non sono gerarchizzati unicamente sotto forma di client o server fissi, ma sotto forma di nodi equivalenti o paritari (in inglese peer) che possono cioè fungere sia da cliente che da server verso gli altri nodi terminali (host) della rete. Essa dunque è un caso particolare dell'architettura logica di rete client-server.

**Persistence Actor:** attore persistente.

**Persistenza:** si riferisce alla caratteristica dei dati di sopravvivere all'esecuzione del programma che li ha creati. Nella programmazione informatica, la persistenza si riferisce in particolare alla possibilità di far sopravvivere delle strutture dati all'esecuzione di un singolo programma. Questa possibilità è raggiunta salvando i dati in uno storage non volatile, come su un file system o su un database (es. applicazioni web).



**Pickle:** *Scala Pickling* è un framework per la serializzazione automatica per Scala. Permette agli utenti di utilizzare diversi formati di serializzazione (come JSON, o formati binari) o di definire il loro formato personale per la serializzazione.

**Plugin:** il plugin in campo informatico è un programma non autonomo che interagisce con un altro programma per ampliarne o estenderne le funzionalità originarie. A seconda dei programmi e delle piattaforme software, i plugin possono esser chiamati con sinonimi diversi: add-in, add-on e estensione.

**Programamzione Funzionale:** vedi *Funzionale*, *programmazione*.

**Prompt:** in una interfaccia a riga di comando, il prompt è il cursore lampeggiante o evidenziato che indica la posizione nella schermata dove verrà eventualmente scritto il comando impartito dall'utente.

**Prototipo:** il modello realizzato nell'ultima fase della progettazione e sperimentazione, e destinato a divenire il punto di partenza della produzione in serie

**Pull:** comando di git per aggiornare il repository locale alla commit più recente.

**Push:** comando di git per inviare le modifiche sulla copia locale al repository remoto.

**Push Model:** Particolare modello del pattern MVC (vedi sopra) chiamato così poichè prevede che le azioni dell'utente vengano interpretate dal *controller*, che genera quindi i dati e li instrada (push) alla *view*.



## Q

---

**Query:** termine che viene usato in informatica per indicare l'interrogazione di un database da parte di un utente.



# R

---

**Range:** estensione, ampiezza.

**Reactive:** un paradigma di programmazione orientato al flusso di dati e al loro continuo cambiamento. Questo significa poter esprimere con facilità, nel linguaggio di programmazione utilizzato, sia flussi di dati statici che dinamici. Inoltre, il modello di esecuzione sottostante propagherà automaticamente i cambiamenti nei flussi di dati.

**Receiver:** Uno dei termini associati al *Command Pattern* (vedi sopra); indica un oggetto che conosce come portare a termine la richiesta del comando concreto.

**Refactoring:** in ingegneria del software, il refactoring (o code refactoring) è una tecnica per modificare la struttura interna di porzioni di codice senza modificarne il comportamento esterno applicata per migliorare alcune caratteristiche non funzionali del software. I vantaggi che il refactoring persegue riguardano in genere un miglioramento della leggibilità della manutenibilità, della riusabilità e dell'estendibilità del codice e la riduzione della sua complessità, eventualmente attraverso l'introduzione a posteriori di design pattern.

**REPL (read-eval-print loop):** è un semplice ambiente interattivo di programmazione informatico che prende singoli input utente(cioè singole espressioni), li valuta, e ritorna il risultato all'utente; un programma scritto in ambiente REPL viene eseguito a tratti.

**Repository (Software Repository):** è uno spazio di archiviazione da dove è possibile recuperare e scaricare dei pacchetti software.

**RESP (REdis Protocol SPecification):** Protocollo utilizzato per la comunicazione tra un client Redis ed un server Redis. Anche se nasce in ambito Redis, può essere utilizzato anche per altri progetti software di tipo client-server.

**Round Robin:** in informatica, è uno degli algoritmi utilizzati dallo scheduler di processi. In particolare, il tempo disponibile viene diviso in diverse porzioni assegnate a turno ai singoli processi in egual misura, senza distinzioni di priorità e in modo circolare. Si tratta di un algoritmo semplice da implementare e evita il rischio di starvation.

**Routing:** Il routing è l'instradamento effettuato a livello di rete; con instradamento si intende decidere su quale porta o interfaccia inviare un elemento di comunicazione ricevuto (conversazione telefonica, pacchetto dati, cella, flusso di dati).



## S

---

**SBT (Scala Build Tool):** è un software open source per il build di progetti in Scala e Java.

**Scala:** È un linguaggio di programmazione di tipo general-purpose studiato per integrare le caratteristiche e funzionalità dei linguaggi orientati agli oggetti e dei linguaggi funzionali. La compilazione di codice sorgente Scala produce Java bytecode per l'esecuzione su una JVM.

**Scalabilità Orizzontale (Scale Out):** il termine scalabilità, nelle telecomunicazioni, nell'ingegneria del software e in altre discipline, si riferisce, in termini generali, alla capacità di un sistema di "crescere" o diminuire di scala in funzione delle necessità e delle disponibilità. Un sistema che gode di questa proprietà viene detto scalabile. In particolare la scalabilità orizzontale fa riferimento all'aggiunta di nuove unità messe tra loro in parallelo perché funzionino come una sola, unica unità.

**Scaladoc:** è un applicativo incluso nel Scala Development Kit, utilizzato per la generazione automatica della documentazione del codice sorgente scritto in linguaggio Scala.

**Scale Out:** vedi *Scalabilità Orizzontale*

**Script:** il termine script, in informatica, designa un tipo particolare di programma, scritto in una particolare classe di linguaggi di programmazione, detti linguaggi di scripting. La distinzione tra un programma normale ed uno script non è netta ma solitamente gli script si distinguono per la loro complessità relativamente bassa, la mancanza di una propria interfaccia grafica e il fatto di essere utilizzati per svolgere mansioni accessorie e molto specifiche.

**Section:** comando di  $\LaTeX$  che indica l'inizio di una sezione.

**Serializzazione:** Processo per salvare un oggetto in un supporto di memorizzazione lineare (ad esempio, un file o un'area di memoria), o per trasmetterlo su una connessione di rete. La serializzazione può essere in forma binaria o può utilizzare codifiche testuali (ad esempio il formato XML) direttamente leggibili dagli esseri umani.

**Server:** in informatica è un componente o sottosistema informatico di elaborazione e gestione del traffico di informazioni che fornisce, a livello logico e fisico, un qualunque tipo di servizio ad altre componenti (tipicamente chiamate clients, cioè clienti) che ne fanno richiesta attraverso una rete di computer, all'interno di un sistema informatico o anche direttamente in locale su un computer.

**Shell:** in informatica, è la parte di un sistema operativo che permette agli utenti di interagire con il sistema stesso, impartendo comandi e richiedendo l'avvio di altri programmi. La shell può essere di tipo *testuale* (vedi *terminale*) o *grafico*.



**Single Responsibility Principle:** il principio di singola responsabilità (single responsibility principle, abbreviato con SRP) afferma che ogni elemento di un programma (classe, metodo, variabile) deve avere una sola responsabilità, e che tale responsabilità deve essere interamente incapsulata dall'elemento stesso. Tutti i servizi offerti dall'elemento dovrebbero essere strettamente allineati a tale responsabilità.

**Singleton:** Il singleton è un design pattern creazionale che ha lo scopo di garantire che di una determinata classe venga creata una e una sola istanza, e di fornire un punto di accesso globale a tale istanza.

**Sistema di Ticketing:** un sistema di ticketing è uno strumento web, appoggiato a un database accessibile dagli utenti, che permette una facile introduzione e descrizione della propria necessità attraverso l'apertura di un ticket.

**Smartphone:** lo smartphone ("telefono intelligente") è un telefono cellulare con capacità di calcolo, di memoria e di connessione dati molto più avanzate rispetto ai normali telefoni cellulari, basato su un sistema operativo per dispositivi mobili.

**Socket:** Un'astrazione software progettata per poter utilizzare delle API standard e condivise per la trasmissione e la ricezione di dati attraverso una rete oppure come meccanismo di IPC.

**Spring:** In informatica Spring è un framework open source per lo sviluppo di applicazioni su piattaforma Java.

**Spring Shell:** Shell interattiva che permette di aggiungere dei comandi personalizzati usando un modello di programmazione basato su *Spring*.

**SQL (Structured Query Language):** è un linguaggio standardizzato per database basati sul modello relazionale progettato per:

- creare e modificare schemi di database (DDL - Data Definition Language);
- inserire, modificare e gestire dati memorizzati (DML - Data Manipulation Language);
- interrogare i dati memorizzati (DQL - Data Query Language);
- creare e gestire strumenti di controllo ed accesso ai dati (DCL - Data Control Language).

**Stack:** in informatica, indica un tipo di dato astratto che viene usato in diversi contesti per riferirsi a strutture dati, le cui modalità d'accesso ai dati in essa contenuti seguono una modalità LIFO (Last In First Out), ovvero tale per cui i dati vengono estratti (letti) in ordine rigorosamente inverso rispetto a quello in cui sono stati inseriti (scritti).

**Storefinder:** nel contesto del capitolato da noi considerato, con storefinder si intende un particolare tipo di attore che si occupa di ricevere le richieste dall'esterno e instradarle ai rispettivi storekeeper, in modo da soddisfarle.





**Storekeeper:** nel contesto del capitolato da noi considerato, con storekeeper si intende un particolare tipo di attore che mantiene fisicamente al proprio interno le coppie chiave-valore.

**Strategy, pattern:** Lo strategy pattern è uno dei pattern comportamentali. L'obiettivo di questa architettura è isolare un algoritmo all'interno di un oggetto. Il pattern strategy è utile in quelle situazioni dove sia necessario modificare dinamicamente gli algoritmi utilizzati da un'applicazione. Questo pattern prevede che gli algoritmi siano intercambiabili tra loro (in base ad una qualche condizione) in modo trasparente al client che ne fa uso.

**SV (Schedule Variance):** indica quanto sia in anticipo o in ritardo il progetto rispetto ad una schedulazione fatta a priori

**SVN:** SubVersioN (noto anche come svn, che è il nome del suo client a riga di comando) è un sistema di controllo versione per software distribuito come free software sotto licenza Apache. Gli sviluppatori software utilizzano Subversion per mantenere le versioni correnti e storiche di file quali codice sorgente, pagine web e documentazione.



# T

---

**Task:** compito assegnabile ad un singolo componente del gruppo di lavoro che dovrebbe essere ultimato entro un periodo di tempo prefissato o entro una deadline.

**TCP (Transmission Control Protocol):** Protocollo di rete a pacchetto di livello di trasporto, appartenente alla suite di protocolli Internet, che si occupa di controllo di trasmissione ovvero rendere affidabile la comunicazione dati in rete tra mittente e destinatario.

**tcpserver:** Nel contesto del capitolato da noi considerato, con tcpserver si intende un particolare tipo di attore che si mette in ascolto su una porta e delega le connessioni entranti al clientactor (vedi sopra).

**Template:** trattasi di un modello che ha il compito di fornire la struttura di un oggetto (documento, sito web, ...) in cui si trovano spazi temporaneamente "bianchi" da riempire successivamente.

**Thread-Safe:** Il concetto di thread safety (dall'inglese sicurezza dei thread) viene utilizzato, nell'ambito del multithreading, per indicare la caratteristica di una porzione di codice che si comporta in modo corretto nel caso di esecuzioni multiple da parte di più threads. In particolare è importante che i vari threads possano avere accesso alle stesse informazioni condivise, ma che queste siano accessibili solo da un thread alla volta.

**Topic:** tema di discussione, argomento.

**Top-Down:** strategia di elaborazione dell'informazione e di gestione delle conoscenze, riguardante principalmente il software. Nel modello top-down si formula inizialmente una visione generale del sistema ovvero se ne descrive la finalità principale senza scendere nel dettaglio delle sue parti. Ogni parte del sistema è successivamente rifinita aggiungendo maggiori dettagli della progettazione. Ogni nuova parte così ottenuta può quindi essere nuovamente rifinita, specificando ulteriori dettagli, finché la specifica completa è sufficientemente dettagliata da validare il modello.

**Travis-CI:** servizio distribuito per l'integrazione continua usato per testare i progetti hostati su GitHub.



## U

---

**UI (User Interface):** L'interfaccia utente è ciò che si frappone tra una macchina e un utente, consentendo l'interazione tra i due. In ambito informatico sono tipicamente utilizzati i seguenti tipi di interfaccia utente:

- Interfaccia a riga di comando (CLI);
- Interfaccia grafica (GUI).

**UML (Unified Modeling Language):** In ingegneria del software, UML è un linguaggio di modellizzazione e specifica basato sul paradigma orientato agli oggetti. L'UML svolge un'importantissima funzione di "lingua franca" nella comunità della progettazione e programmazione a oggetti. Gran parte della letteratura di settore usa UML per descrivere soluzioni analitiche e progettuali in modo sintetico e comprensibile a un vasto pubblico.

**Unicode:** sistema di codifica che assegna un numero univoco ad ogni carattere usato per la scrittura di testi, in maniera indipendente dalla lingua, dalla piattaforma informatica e dal programma utilizzato

**UpperCamelCase:** vedi *CamelCase*.

**URL (Uniform Resource Locator):** nella terminologia delle telecomunicazioni e dell'informatica è una sequenza di caratteri che identifica univocamente l'indirizzo di una risorsa in Internet, tipicamente presente su un host server, come ad esempio un documento, un'immagine, un video, rendendola accessibile ad un client che ne faccia richiesta attraverso l'utilizzo di un web browser.

**Userkeeper:** nel contesto del capitolato da noi considerato, con userkeeper si intende un particolare tipo di attore dedicato alla gestione del profilo utente. Contiene una lista di collezioni e permessi annessi.

**Username(Nome Utente):** in informatica definisce il nome con il quale l'utente viene riconosciuto da un computer, da un programma o da un server.

**UTF-8 (Unicode Transformation Format, 8 bit):** È una codifica dei caratteri *Unicode* (vedi sopra) in sequenze di lunghezza variabile di byte, creata da Rob Pike e Ken Thompson. UTF-8 usa gruppi di byte per rappresentare i caratteri Unicode, ed è particolarmente utile per il trasferimento tramite sistemi di posta elettronica a 8-bit. In particolare, UTF-8 usa da 1 a 4 byte per rappresentare un carattere Unicode.



## V

---

**Versionamento:** Il versionamento (versioning), in informatica, è la gestione di versioni multiple di un insieme di informazioni. Gli strumenti software per il controllo versione sono ritenuti molto spesso necessari per la maggior parte dei progetti di sviluppo software.

**View:** Le viste sono un elemento utilizzato dalla maggior parte dei DBMS (DataBase Management System). Si tratta, come suggerisce il nome, di "modi di vedere i dati". Una vista è rappresentata da una query, il cui risultato può essere utilizzato come se fosse una tabella (o collezione).

**Virtual Machine:** Il termine macchina virtuale (VM) indica un software che, attraverso un processo di virtualizzazione, crea un ambiente virtuale che emula tipicamente il comportamento di una macchina fisica grazie all'assegnazione di risorse hardware (porzioni di disco rigido, RAM e risorse di processamento) ed in cui alcune applicazioni possono essere eseguite come se interagissero con tale macchina. Tra i vantaggi vi è il fatto di poter offrire contemporaneamente ed efficientemente a più utenti diversi ambienti operativi separati, ciascuno attivabile su effettiva richiesta, senza sporcare il sistema fisico reale con il partizionamento del disco rigido oppure fornire ambienti clusterizzati (connessi tra loro) su sistemi server.

**Visibilità:** Nell'ambito di Actorbase, gli utenti registrati hanno la possibilità di avere diversi gradi di visibilità sulla base di dati, in base al fatto che questi siano amministratori o semplici utenti. La visibilità garantisce privilegi sia in lettura che in scrittura; un amministratore ha visibilità sull'intero sistema mentre ogni utente generico solo sulle proprie collezioni e/o su collezioni di proprietà altrui su cui è in collaborazione.

**VoIP (Voice over IP):** tecnologia che rende possibile effettuare una conversazione telefonica sfruttando una connessione Internet o una qualsiasi altra rete dedicata a commutazione di pacchetto che utilizzi il protocollo IP senza connessione per il trasporto dati.



## W

---

**Warehouseman:** Nel contesto del capitolato da noi considerato, con warehouseman si intende un particolare tipo di attore che si interfaccia con gli *storekeeper* (vedi sopra) e persiste su disco le rispettive mappe.

**Whiteboarding:** È una tecnica di condivisione di un notebook o di una whiteboard (Lavagna) virtuali nei quali più persone possono lavorare contemporaneamente per scambiarsi idee, bozze di grafici e schemi. Tutto questo viene fatto online così da permettere ai singoli utenti di vedere in tempo reale le modifiche fatte al documento.

**Wiki:** Il wiki è un sito web, costruito appoggiandosi su una piattaforma o software collaborativo detto software wiki o semplicemente Wiki, che permette ai propri utenti di aggiungere, modificare o cancellare contenuti apertamente attraverso un browser web, in genere utilizzando un linguaggio di markup semplificato o un editor di testo online. Si tratta in altre parole di una raccolta di documenti ipertestuali che viene aggiornata dai suoi stessi utilizzatori e i cui contenuti sono sviluppati in collaborazione da tutti coloro che vi hanno accesso (contenuto generato dagli utenti), memorizzati normalmente su un database o un repository.



ScalateKids

ScalateKids

*INDICE*

**X**



## Y

---



## Z

---