



Manuale Sviluppatore

Jawa Druids

Versione	-
Data approvazione	-
Responsabile	-
Redattori	Alfredo Graziano Igli Mezini
Verificatori	-
Stato	Approvato
Lista distribuzione	Jawa Druids Prof. Tullio Vardanega Prof. Riccardo Cardin
Uso	Esterno

Sommario

Il documento ha lo scopo di presentare le tecnologie e l'architettura del sistema agli sviluppatori interessati al software *GDP - Gathering Detection Platform*.



Registro delle modifiche

Versione	Data	Autore	Ruolo	Verificatore	Modifica
v0.0.2	04-04-2021	Mattia Cocco	<i>Progettista</i>	Andrea Dorigo	<i>Stesura § 4</i>
v0.0.1	02-04-2021	Alfredo Graziano	<i>Progettista</i>	Igli Mezini	<i>Stesura § 1</i>



Indice

1	Introduzione	4
1.1	Scopo del documento	4
1.2	Scopo del prodotto	4
1.3	Glossario	4
2	Requisiti di sistema	5
3	Procedure di installazione	6
3.1	Download repo	6
3.2	Installazione dipendenze	6
3.3	Lancio del software	6
4	Tecnologie coinvolte	7
4.1	Tecnologie	7
4.1.1	Python	7
4.1.2	MongoDB	7
4.1.3	HTML 5	7
4.1.4	CSS 3	8
4.1.5	Java	8
4.1.6	Vue.js	8
4.1.7	Node.js	8
4.1.8	Bootstrap	9
4.1.9	JSON	9
4.2	Librerie di terze parti	9
4.2.1	OpenCV	9
4.2.2	Yolo V3	9
4.2.3	Pandas	10
4.2.4	Scikit-learn	10
4.2.5	Mongoengine	10
4.2.6	NumPy	10
4.2.7	Pylint	10
4.2.8	Checkstyle	11
4.2.9	ESLint	11
4.2.10	Prettier	11
5	Architettura del Prodotto	12



Elenco delle figure



1 Introduzione

1.1 Scopo del documento

Il documento si propone come guida introduttiva del software *GDP: Gathering Detection Platform*, indirizzata agli sviluppatori che ci lavoreranno. Nello specifico è presentata l'architettura del prodotto e l'organizzazione del codice sorgente ed inoltre sono indicate la procedura di installazione in locale e le tecnologie coinvolte.

1.2 Scopo del prodotto

In seguito alla pandemia del virus COVID-19 è nata l'esigenza di limitare il più possibile i contatti fra le persone, specialmente evitando la formazione di assembramenti. Il progetto *GDP: Gathering Detection Platform* di *Sync Lab* ha pertanto l'obiettivo di **creare una piattaforma in grado di rappresentare graficamente le zone potenzialmente a rischio di assembramento, al fine di prevenirlo**. Il prodotto finale è rivolto specificatamente agli organi amministrativi delle singole città, cosicché possano gestire al meglio i punti sensibili di affollamento, come piazze o siti turistici. Lo scopo che il software intende raggiungere non è solo quello della rappresentazione grafica real-time ma anche quella di poter riuscire a prevedere assembramenti in intervalli futuri di tempo.

Al tal fine il gruppo *Jawa Druids* si prefigge di sviluppare un prototipo software in grado di acquisire, monitorare ed analizzare i molteplici dati provenienti dai diversi sistemi e dispositivi, a scopo di identificare i possibili eventi che concorrono all'insorgere di variazioni di flussi di utenti. Il gruppo prevede inoltre lo sviluppo di un'applicazione web da interporre fra i dati elaborati e l'utente, per favorirne la consultazione.

1.3 Glossario

Allo scopo di evitare ambiguità a lettori esterni si aggiunge in appendice un glossario dei termini ambigui o specifici utilizzati nel presente documento che verranno segnalati con una *G* a pedice.



2 Requisiti di sistema



3 Procedure di installazione

Questa sezione terrà conto delle procedure di installazione all'interno del sistema operativo Linux, più precisamente Ubuntu, in quanto utilizzato anche per lo sviluppo del software stesso.

3.1 Download repo

Come prima fase bisogna clonare la repo relativa al progetto *Gathering-Detection-Platform* al seguente sito:

<https://github.com/Andrea-Dorigo/gathering-detection-platform>

3.2 Installazione dipendenze

Dopo aver clonato la repo nella propria macchina, bisogna accedervi. Ci sono due modi per farlo:

- mediante il terminale: `cd Path/To/File;`
- mediante lo spostamento all'interno delle cartelle tramite mouse.

Una volta nella cartella del software basterà eseguire il file da terminale

dipendenze.sh

Successivamente bisognerà aspettare che vengano installate tutte le dipendenze necessarie per poter permettere al software di funzionare adeguatamente.

3.3 Lancio del software

Finita l'installazione non rimane che eseguire effettivamente il software completo. Per fare ciò basterà eseguire il file

start.sh

Ciò farà iniziare la corretta esecuzione del prodotto.



4 Tecnologie coinvolte

In questa sezione vengono elencate le tecnologie, e librerie di terze parti, utilizzate per sviluppare il prodotto software *Gathering-Detection-Platform*.

4.1 Tecnologie

4.1.1 Python

Si tratta di un linguaggio di programmazione definito "ad alto livello" rispetto alla maggior parte di essi. Si tratta di un linguaggio orientato ad oggetti, utile a sviluppare script, computazione numerica e sviluppare software. Nel progetto *Gathering-Detection-Platform*, Python_G è il linguaggio su cui si basa tutto il backend_G , compreso il modulo del machine-learning $_G$.

- versione utilizzata: 3.8.x;
- link download: <https://www.python.org/downloads/> .

4.1.2 MongoDB

MongoDB_G è stato scelto come database_G nel quale salvare i dati ottenuti dal modulo di acquisizione e dal modulo di machine-learning $_G$. Si tratta di un database_G non relazionale e orientato ai documenti. Classificato come tipo NoSQL_G , MongoDB_G non utilizza la classica struttura basata su tabelle ma invece si basa su tipi di documenti JSON_G , facilitando così l'integrazioni di alcuni tipi di dati.

- versione utilizzata: 4.4.4;
- link al sito: <https://www.mongodb.com/it>.

4.1.3 HTML 5

HTML_G , acronimo di HyperText Markup Language, è un linguaggio di mark up per siti web. Era stato ideato per la formattazione e impaginazione di pagine ipertestuali sul web. Oggi giorno viene utilizzato soprattutto per gestire la separazione tra la struttura logica della pagina web e la sua rappresentazione, gestita dal CSS_G . Nel progetto questo linguaggio viene utilizzato per sviluppare la parte di web-app_G , interagendo con anche Java_G , CSS_G , Bootstrap_G e Vue.js_G .



4.1.4 CSS 3

Il CSS_G è il principale linguaggio utilizzato per definire la formattazione dei siti e pagine web. L'utilizzo del CSS_G permette di separare i contenuti della pagina HTML_G dal proprio layout ma anche di rendere la programmazione più chiara e facile da utilizzare, garantendo il riutilizzo di codice e facilitando la manutenzione. Nel progetto viene utilizzato per formattare il layout estetico della web-app_G.

4.1.5 Java

Si tratta di una piattaforma che ha come caratteristica principale il fatto di rendere possibile scrittura ed esecuzione di applicazioni indipendenti dall'hardware di esecuzione. Il risultato è una virtualizzazione dalla piattaforma stessa, che rende così il linguaggio Java_G, e i relativi programmi, portabili su piattaforme hardware diverse.

- versione utilizzata: 11.x;
- link download: <https://www.java.com/it/download/>.

4.1.6 Vue.js

È un framework JavaScript_G, configurato come Model-Control-View per la creazione di interfacce utente e applicazione single-page. Supporta molte funzionalità, anche avanzate, grazie ad una serie di librerie di supporto dedicate che sono ufficialmente mantenute.

- versione utilizzata: 4.5.x;
- link al sito: <https://vuejs.org/>.

4.1.7 Node.js

È un runtime system open-source_G, orientato ad oggetti, per l'esecuzione di codice JavaScript_G. Molti moduli di questa tecnologia sono proprio scritti in JavaScript_G, ed essendo appunto open-source_G, programmatori esterni possono crearne ed aggiungerne altri. A differenza di JavaScript_G che in origine era lato client_G, Node.js_G viene utilizzato lato server_G, ad esempio per produzioni di pagine dinamiche. Implementa il paradigma "JavaScript everywhere" in modo da unificare lo sviluppo di applicazioni web intorno ad un unico linguaggio di programmazione, JavaScript_G.

- versione utilizzata: 14.16.x;
- link download: <https://nodejs.org/it/download/>.



4.1.8 Bootstrap

Framework open-source che, mediante le proprie librerie, viene utilizzato per uniformare i vari componenti che compongono un'interfaccia web, oltre che per crearli.

- versione utilizzata: 4.3.1;
- link download: <https://getbootstrap.com/docs/5.0/getting-started/download/>.

4.1.9 JSON

Si tratta di un formato testuale necessario per l'esportazione ed importazione dei dati presenti nel *modulo* di salvataggio dati, mediante MongoDB_G, ed esterni al database_G. È un formato dati diffuso per lo scambio di essi in applicazioni client-server_G. Basato su oggetti, ovvero coppie chiave/valore, e supporta una moltitudine di dati diversi. Infine è di facile lettura per l'utente e non necessita particolari procedure per modificarlo.

4.2 Librerie di terze parti

Insieme alle tecnologie sopra citate, sono state anche integrate delle librerie di terze parti.

4.2.1 OpenCV

OpenCV è una libreria software multiplatforma specializzata nella visione artificiale in tempo reale. È stata integrata nel *modulo* adibito alla cattura immagini in tempo reale, in linguaggio python_G.

- versione utilizzata: 4.2.0;
- link al sito: <https://opencv.org/>.

4.2.2 Yolo V3

Si tratta di uno script in linguaggio python_G per il riconoscimento real-time di oggetti in una foto. Viene utilizzato nel *modulo* di acquisizione dati per il riconoscimento e conteggio delle persone presenti in un singolo frame.

- link al sito: <https://pjreddie.com/darknet/yolo/>.



4.2.3 Pandas

È una libreria veloce, potente e flessibile creata appositamente per analizzare dati e manipolarli mediante appositi strumenti. Utilizzata nel *modulo* di machine-learning_c, basata su python_c.

- versione utilizzata: 1.2.1;
- link all'installazione: https://pandas.pydata.org/getting_started.html.

4.2.4 Scikit-learn

È una libreria open source di apprendimento automatico per il linguaggio di programmazione Python. Al suo interno sono presenti numerosi algoritmi, per la manipolazione dati, tra cui quelli di regressione, utilizzati nel *modulo* di machine-learning_c.

- versione utilizzata: 0.24.1;
- link all'installazione: <https://scikit-learn.org/stable/install.html>.

4.2.5 Mongoengine

Si tratta di un document-object mapper basato su python_c ed ideato per lavorare assieme a MongoDB_c da python_c.

- versione utilizzata: 0.24.1;
- link alla repo_c: <https://github.com/MongoEngine/mongoengine>.

4.2.6 NumPy

Libreria open-source_c, basata sul linguaggio python_c. Fornisce un grosso supporto a grandi matrici e array multidimensionali, inoltre integra molte funzioni matematiche adatte a lavorare su tali strutture dati.

- versione utilizzata: 1.20.1;
- link all'installazione: <https://numpy.org/install/>.

4.2.7 Pylint

Strumento utilizzato per l'analisi statica del codice sorgente Python_c. Viene quindi adottato per controllare la presenza di errori nel codice, con l'obiettivo di applicare uno standard codifica e di promuovere buone prassi di scrittura del codice

- versione utilizzata: 2.7.3;
- link al sito: <https://pypi.org/project/pylint/>.



4.2.8 Checkstyle

Strumento che permette di eseguire l'analisi statica del codice `javac` utilizzato nello sviluppo di un progetto software.

- versione utilizzata: 2.17;
- link alla repo_c: <https://github.com/checkstyle/checkstyle>.

4.2.9 ESLint

Strumento di analisi del codice statico per identificare i le problematiche trovate nel codice `JavaScriptc`.

- versione utilizzata: 2.1.19;
- link alla repo: <https://www.npmjs.com/package/eslint>.

4.2.10 Prettier

Strumento per il controllo automatico della formattazione del codice scritto in linguaggio `JavaScriptc`.

- versione utilizzata: 6.3.2;
- link al sito: <https://prettier.io/>.



5 Architettura del Prodotto