



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	5
1.1	Scopo del documento	5
1.2	Scopo del prodotto	5
1.3	Glossario	5
2	Requisiti di sistema	6
3	Procedure di installazione	7
3.1	Download della repository	7
3.2	Installazione delle dipendenze	7
3.3	Inizializzazione del modulo di acquisizione	8
3.3.1	Aggiunta di una webcam	8
3.4	Inizializzazione modulo Prediction	8
3.5	Inizializzazione modulo Web-App	8
4	Tecnologie coinvolte	10
4.1	Tecnologie	10
4.1.1	Python	10
4.1.2	MongoDB	10
4.1.3	Spring Boot	10
4.1.4	Java	11
4.1.5	HTML 5	11
4.1.6	CSS 3	11
4.1.7	Leaflet	11
4.1.8	Vue.js	11
4.1.9	Node.js	12
4.1.10	Bootstrap	12
4.1.11	JSON	12
4.2	Librerie di terze parti	12
4.2.1	OpenCV	12
4.2.2	Yolo V3	13
4.2.3	Pandas	13
4.2.4	Scikit-learn	13
4.2.5	Mongoengine	13
4.2.6	NumPy	13
4.2.7	Pylint	14
4.2.8	Checkstyle	14



4.2.9	ESLint	14
4.2.10	Prettier	14

5	Architettura del Prodotto	15
----------	----------------------------------	-----------



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 esporrà le procedure di installazione all'interno del sistema operativo Linux, più precisamente Ubuntu 20.04 LTS, in quanto utilizzato anche per lo sviluppo del software stesso. Rimane comunque possibile installare il software su altri sistemi operativi soddisfacendo le dipendenze necessarie, ma non verrà qui esplicitato.

3.1 Download della repository

Per scaricare correttamente i contenuti della repository è necessario installare `git` e `git-lfs` (*Git Large File Storage*). Su Ubuntu 20.04, questo è possibile eseguendo il comando:

```
sudo apt install git git-lfs
```

assumendo che le principali repository per i pacchetti di Ubuntu siano attive (Universe, Multi-verse).

Questo passaggio è richiesto poiché GitHub (il sito che ospita la repository del progetto) consente l'upload di file con dimensioni massime fino a 100MB. L'utilizzo di *Git Large File Storage* permette l'upload e il download di file che superano questo limite, ed in particolare permette l'upload e download dei pesi necessari all'algoritmo YOLOv3 per il rilevamento di oggetti (più precisamente per il rilevamento delle persone in un'immagine), il quale ha una dimensione maggiore di 200MB.

Maggiori informazioni riguardo Git Large File Storage sono reperibili all'indirizzo: <https://git-lfs.github.com>. È dunque possibile clonare correttamente la repository relativa al progetto *Gathering-Detection-Platform* con il seguente comando:

```
git clone https://github.com/Andrea-Dorigo/gathering-detection-platform.git
```

3.2 Installazione delle dipendenze

Dopo aver eseguito il passo sopra descritto, è obbligatorio installare le dipendenze necessarie a far eseguire il prodotto software adeguatamente. Per fare ciò sarà necessario aprire il terminale, all'interno della cartella *gathering-detection-platform*, e copiare il seguente comando:

```
sudo apt install python3-opencv python3-pip mongo maven npm && pip3 install mongoe
```




Una volta conclusa questa operazione il programma potrà essere eseguito senza problemi.

3.3 Inizializzazione del modulo di acquisizione

Per eseguire il modulo di acquisizione, in modo da iniziare a raccogliere i dati dalle webcam salvate, basterà posizionarsi all'interno della cartella acquisition/, dopodiché nella cartella "main" e da terminale eseguire il comando:

```
python3 detect.py
```

Se i passi precedenti sono stati eseguiti correttamente allora si vedranno comparire sul terminale i vari passi che svolge il modulo.

3.3.1 Aggiunta di una webcam

L'aggiunta di una webcam al modulo d'acquisizione è relativamente semplice. È sufficiente accedere al file *webcams.json* ed inserire la webcam prendendo spunto dallo schema delle altre già inserite. Attualmente per una questione di codifica, il link della webcam dev'essere conforme a quelle già presenti, ovvero provenire da <https://www.whatsupcams.com/>.

3.4 Inizializzazione modulo Prediction

Per eseguire il modulo di predizione, che tramite il machine-learning si occupa di calcolare, appunto, le predizioni per lassi di tempo futuri, bisognerà posizionarsi all'interno della cartella "prediction" ed eseguire il seguente comando da terminale:

```
python3 DataPrediction.py
```

In tal modo verrà attivato il modulo per le predizioni sui dati.

3.5 Inizializzazione modulo Web-App

Per avviare la web-app, e le sue funzioni, si devono eseguire alcuni comandi, sempre da terminale, a partire dalla cartella webapp.

1. posizionarsi all'interno della cartella "webapp" ed eseguire:



```
mvn spring-boot:run
```

2. posizionarsi all'interno della cartella "vue-js-client-crud" ed eseguire:

```
npm install
```

3. infine, all'interno della stessa cartella bisogna eseguire:

```
npm run
```



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 Spring Boot

Spring è un framework open source per lo sviluppo di applicazioni su piattaforma Java $_G$. A questo framework sono associati altri progetti, in particolare Spring Boot che permette di creare una applicazione avente un metodo main che lancia l'intera applicazione web, compreso il web server integrato.

- versione utilizzata: 2.4.4;
- link al sito: <https://spring.io/projects/spring-boot>.



4.1.4 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/>.

4.1.5 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 .

4.1.6 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 .

4.1.7 Leaflet

Leaflet è una libreria JavaScript per sviluppare mappe geografiche, utilizzata nel progetto per realizzare la heat-map.

- versione utilizzata: 1.7.1;
- link al sito: <https://leafletjs.com/>.

4.1.8 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: 2.6.12;
- link al sito: <https://vuejs.org/>.



4.1.9 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/>.

4.1.10 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/>.

4.1.11 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_c 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_g. 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 java_g utilizzato nello sviluppo di un progetto software.

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

4.2.9 ESLint

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

- 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 JavaScript_g.

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



5 Architettura del Prodotto