



Manuale Sviluppatore

Jawa Druids

Versione	-
Data approvazione	-
Responsabile	Mattia Cocco
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	Modifica	Verificatore
v1.0.0	-	Mattia Cocco	<i>Responsabile</i>	<i>Approvazione del documento per RQ</i>	-
v0.0.2	2021-04-04	Mattia Cocco	<i>Progettista</i>	<i>Stesura § 4</i>	Andrea Dorigo
v0.0.1	2021-04-02	Alfredo Graziano	<i>Progettista</i>	<i>Stesura § 1</i>	Igli Mezini



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	9
4	Tecnologie coinvolte	10
4.1	Tecnologie	10
4.1.1	Python	10
4.1.2	Kafka	10
4.1.3	MongoDB	10
4.1.4	Spring Boot	11
4.1.5	Apache Maven	11
4.1.6	Java	11
4.1.7	HTML 5	11
4.1.8	CSS 3	12
4.1.9	Leaflet	12
4.1.10	Vue.js	12
4.1.11	Node.js	12
4.1.12	Bootstrap	13
4.1.13	JSON	13
4.2	Librerie di terze parti	13
4.2.1	OpenCV	13
4.2.2	Yolo V3	13
4.2.3	Pandas	14
4.2.4	Scikit-learn	14
4.2.5	Mongoengine	14
4.2.6	NumPy	14



4.2.7	Pylint	14
4.2.8	Checkstyle	15
4.2.9	ESLint	15
4.2.10	Prettier	15
5	Architettura del Prodotto	16
5.1	Architettura modulo Acquisition	16
5.1.1	Diagramma dei Package	16
5.1.2	Diagrammi di attività	17
5.2	Architettura modulo Prediction	17
5.2.1	Diagramma dei package	17
5.2.2	Diagramma di attività	17
5.3	Architettura modulo Web-app	17
5.3.1	Diagrammi dei package	17
5.3.2	Diagrammi delle classi	17
5.3.3	Diagramma di sequenza	17
5.3.4	Diagramma di attività	17
6	Glossario	29



Elenco delle figure

5.1	Diagramma dei package del modulo Acquisition	16
5.2	Diagramma di attività dell'eseguibile Detection	18
5.3	Diagramma di sotto-attività dell'acquisizione delle previsioni meteo	19
5.4	Diagramma di sotto-attività di download e taglio frame	20
5.5	Diagramma di sotto-attività del conta persone	21
5.6	Diagramma di attività di Kafka	22
5.7	Diagramma dei package del modulo Acquisition	23
5.8	Diagramma di attività dell'eseguibile Detection	24
5.9	Diagramma dei package di Spring	25
5.10	Diagramma dei package del modulo Acquisition	26
5.11	Diagramma delle classi di Spring	27
5.12	Diagramma di sequenza di Spring	27
5.13	Diagramma di attività del modulo Web-app	28



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_G, più precisamente Ubuntu_G 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_G è necessario installare `git` e `git-lfs` (*Git Large File Storage*). Su Ubuntu_G 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_G 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 necessario installare le dipendenze necessarie a far eseguire il prodotto software_G adeguatamente. Per fare ciò è sufficiente aprire il terminale all'interno della cartella *gathering-detection-platform_G*, ed eseguire il seguente comando:



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

Una volta conclusa questa operazione con esito positivo, il programma potrà essere eseguito.

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/main/`, e da terminale eseguire il comando:

```
python3 detect.py
```

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

3.3.1 Aggiunta di una webcam

L'aggiunta di una nuova webcam al modulo di acquisizione è possibile attraverso dei pochi semplici passi:

1. Trovare una webcam disponibile all'interno del sito <https://www.whatsupcams.com/>;
2. Inserire il link all'interno del file `webcams.json` seguendo lo schema prestabilito per impostare i parametri della webcam;
3. Salvare il file e l'aggiunta è ultimata.

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_G si occupa di calcolare, appunto, le predizioni del periodo di tempo futuro, bisogna 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_G, 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
```

Il primo comando inizializza il server di Spring_G che fornisce i servizi per prelevare le informazioni dal database, mentre i comandi successivi installano prima npm_G e successivamente le dipendenze di ogni file necessarie alla web-app_G.



4 Tecnologie coinvolte

In questa sezione vengono elencate le tecnologie, e librerie di terze parti, utilizzate per sviluppare il prodotto software_G *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_G, computazione numerica e sviluppare software_G. 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 Kafka

Kafka_G viene utilizzato, dal modulo acquisizione, come tramite per inviare i dati ad altre applicazioni utilizzando uno standard di comunicazione comune. È stato scelto in quanto acquisisce flussi di dati da diverse fonti e permette a molte applicazioni di scambiarsi dati mediante esso, il suo scopo è quello di "centro smistamento" dei dati.

- versione utilizzata: 2.8.0;
- link download: <https://kafka.apache.org/downloads> .

4.1.3 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, invece si basa sui 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> .



4.1.4 Spring Boot

Spring_G è un framework_G open source_G per lo sviluppo di applicazioni su piattaforma Java_G. A questo framework_G sono associati altri progetti, in particolare Spring Boot_G che permette di creare una applicazione autoconfigurata che avvia un server_G, il quale mette a disposizione i servizi sviluppati attraverso il codice.

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

4.1.5 Apache Maven

È una tecnologia utilizza per la gestione software_G basati su Java_G e build automation_G. Per la gestione si serve di un costrutto denominato POM (Project Object Model)_G, ovvero un file XML_G in cui vengono dichiarate le dipendenze necessarie fra il progetto e le varie librerie utilizzate. Maven_G si occupa di scaricare automaticamente eventuali librerie o plug-in_G mancanti in una cartella predefinita.

- versione utilizzata: 3.6.3;
- link al sito: <https://maven.apache.org/download.cgi> .

4.1.6 Java

Si tratta di una piattaforma che ha come caratteristica principale il fatto di rendere possibile scrittura ed esecuzione di applicazioni indipendenti dall'hardware_G 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_G diverse.

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

4.1.7 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.



4.1.8 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_G, 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_G estetico della web-app_G.

4.1.9 Leaflet

Leaflet_G è una libreria JavaScript_G per sviluppare mappe geografiche, utilizzata nel progetto per realizzare la heat-map_G.

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

4.1.10 Vue.js

È un framework JavaScript_G, configurato come Model-Control-View_G per la creazione di interfacce utente e applicazione single-page_G. 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/> .

4.1.11 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/> .



4.1.12 Bootstrap

Framework_G open-source_G 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/> .

4.1.13 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_G 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_G in linguaggio python_G per il riconoscimento real-time_G 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 modellare dati e manipolarli mediante appositi strumenti. Utilizzata nel *modulo* di machine-learning_G, basata su python_G.

- 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_G. Al suo interno sono presenti numerosi algoritmi, per la manipolazione dati, tra cui quelli di regressione, utilizzati nel *modulo* di machine-learning_G.

- 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_G basato su python_G ed ideato per lavorare assieme a MongoDB_G da python_G.

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

4.2.6 NumPy

Libreria open-source_G, basata sul linguaggio python_G. 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 `javag` utilizzato nello sviluppo di un progetto `softwareg`.

- 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 `JavaScriptg`.

- 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 `JavaScriptg`.

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

5 Architettura del Prodotto

L'architettura generale del software *Gathering-Detection-Platform* è un'architettura monolitica distribuita. Si tratta di due file eseguibili, scritti in Python_g, che sono rispettivamente il modulo Acquisition e il modulo Prediction. Il primo viene utilizzato per acquisire le informazioni estrapolate dalle live webcams delle città, il secondo invece viene utilizzato per calcolare predizioni su periodi di tempo futuri, utilizzando il machine-learning_g. Entrambi sono collegati ad un database, il quale serve per salvare ed esportare i dati in esso. Il terzo, ed ultimo, modulo riguarda la web-app_g vera e propria, che permette all'utente utilizzatore di visualizzare la heat-map_g relativa alla città.

5.1 Architettura modulo Acquisition

5.1.1 Diagramma dei Package

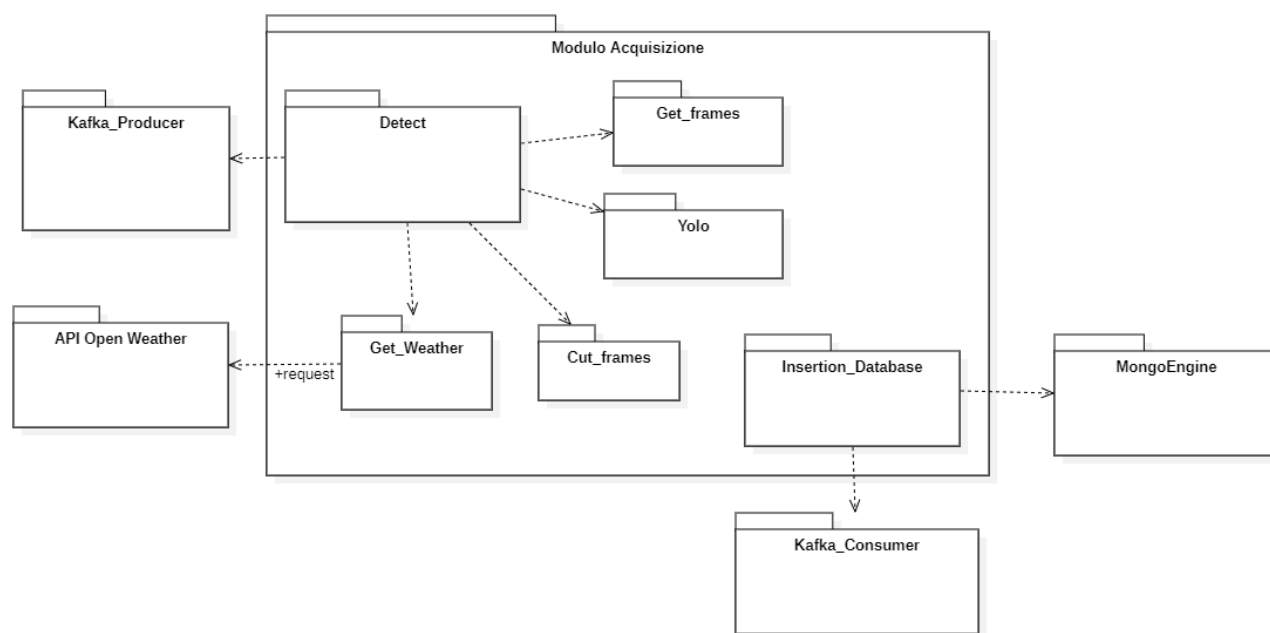


Figura 5.1: Diagramma dei package del modulo Acquisition



5.1.2 Diagrammi di attività

5.2 Architettura modulo Prediction

5.2.1 Diagramma dei package

5.2.2 Diagramma di attività

5.3 Architettura modulo Web-app

5.3.1 Diagrammi dei package

5.3.2 Diagrammi delle classi

5.3.3 Diagramma di sequenza

5.3.4 Diagramma di attività

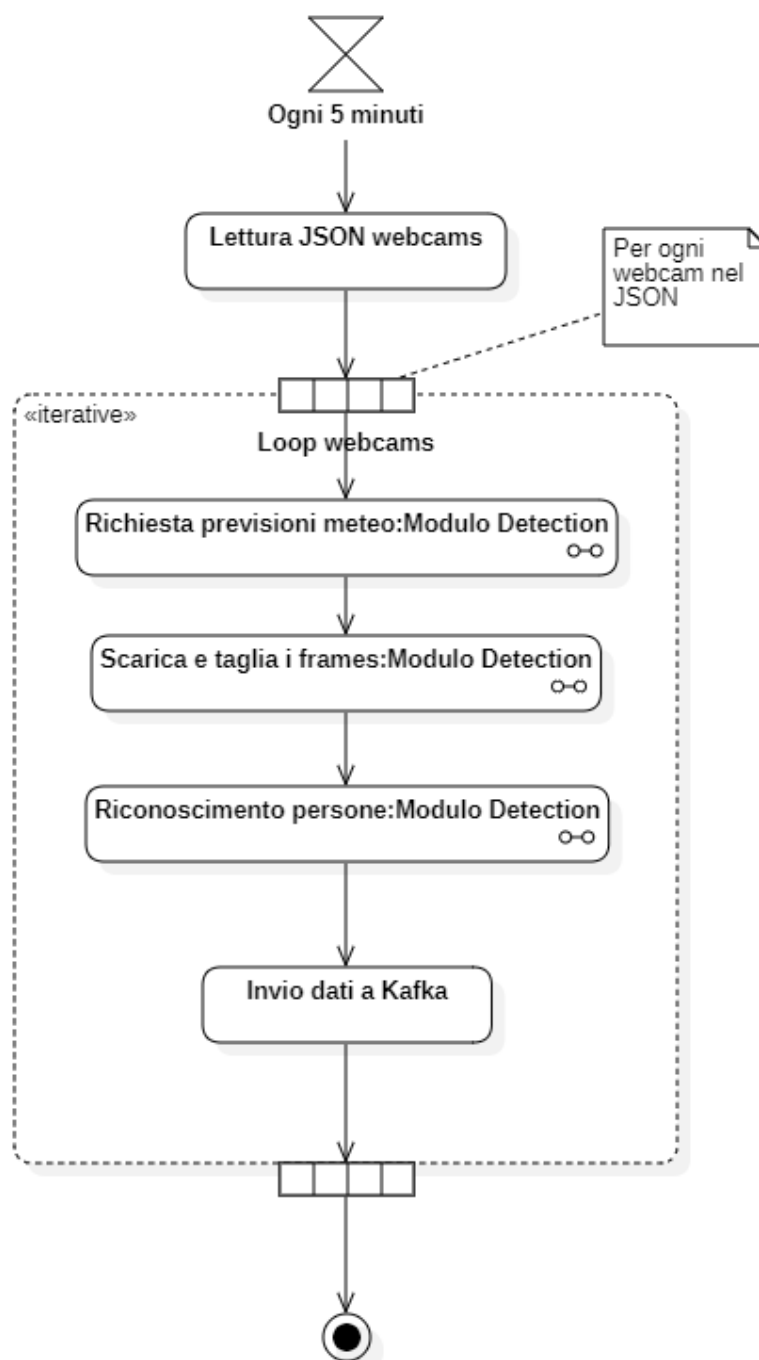


Figura 5.2: Diagramma di attività dell'eseguibile Detection

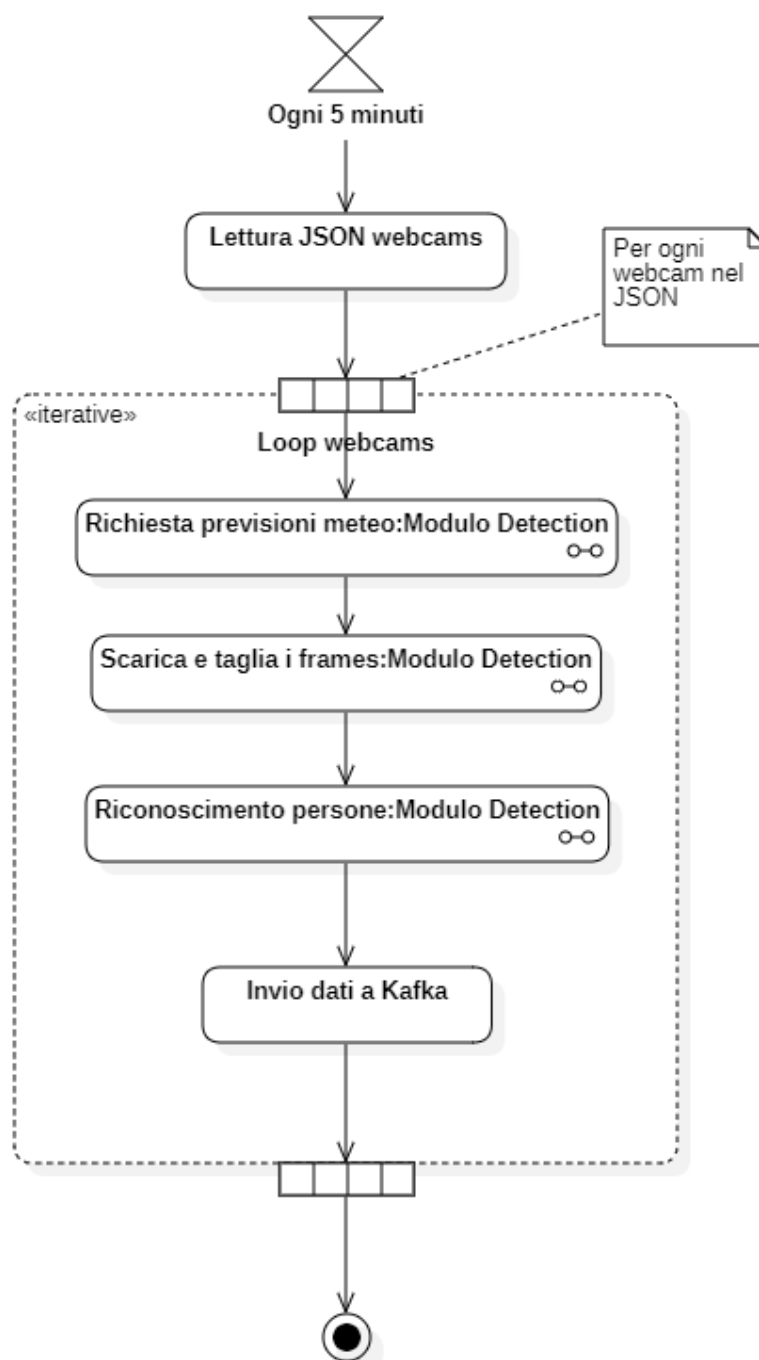


Figura 5.3: Diagramma di sotto-attività dell'acquisizione delle previsioni meteo

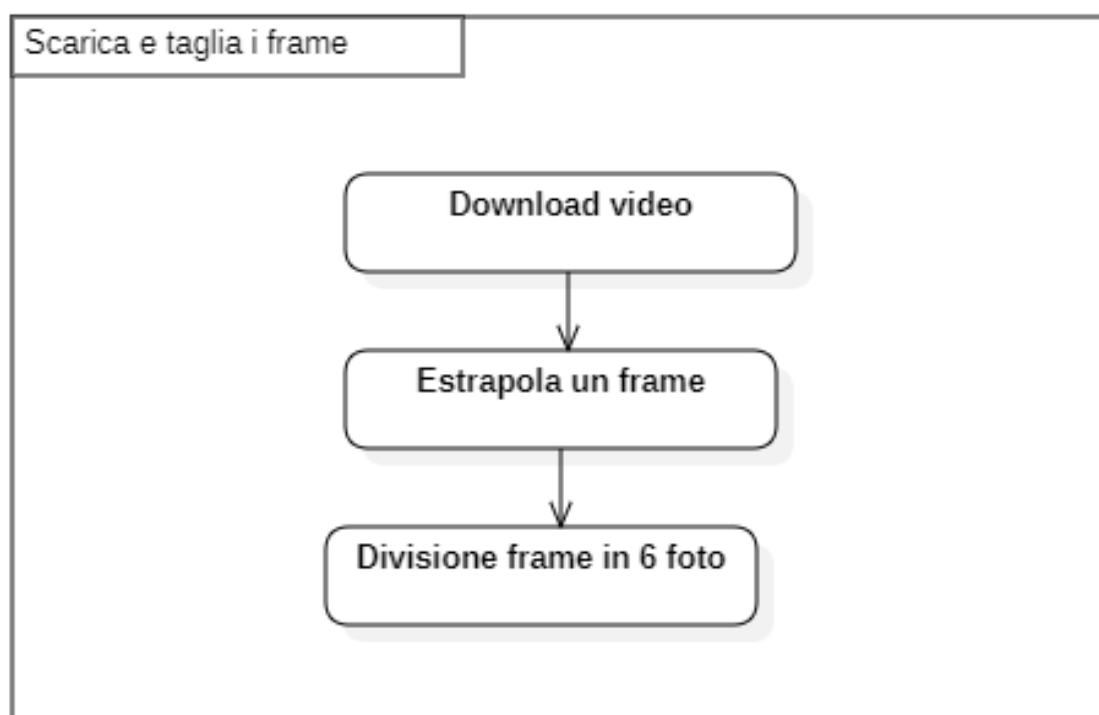


Figura 5.4: Diagramma di sotto-attività di download e taglio frame

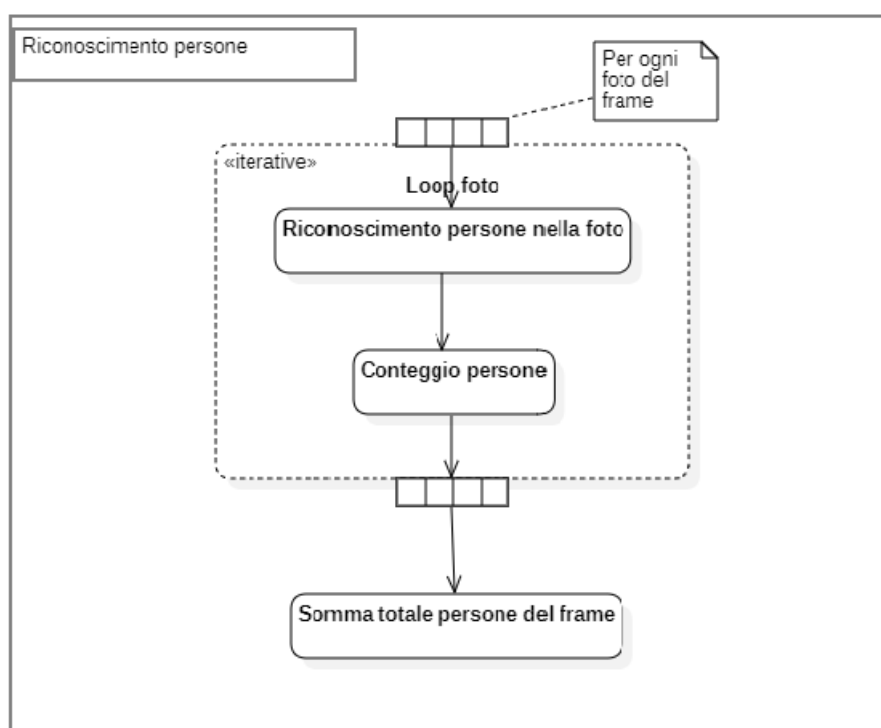


Figura 5.5: Diagramma di sotto-attività del conta persone

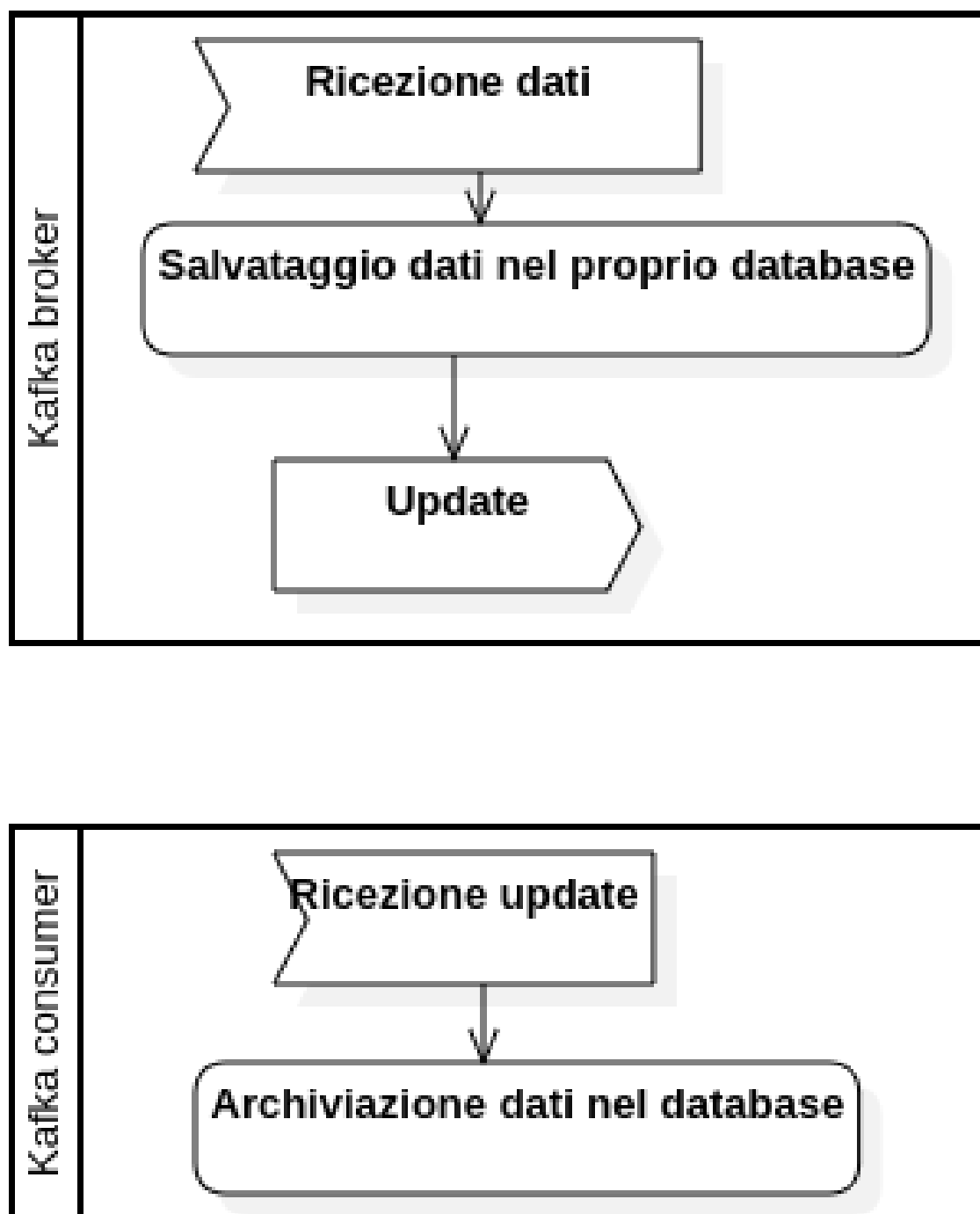


Figura 5.6: Diagramma di attività di Kafka

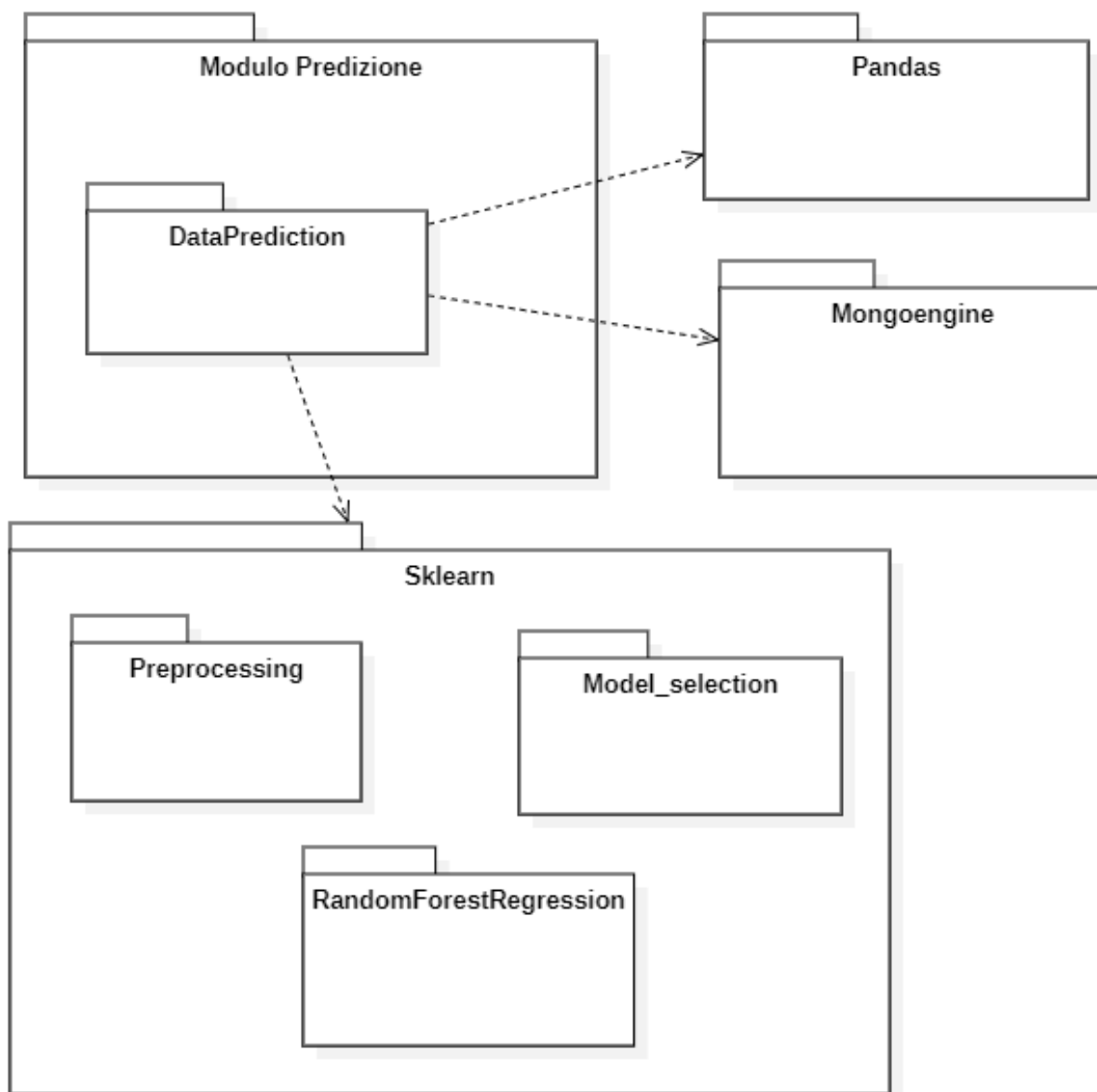
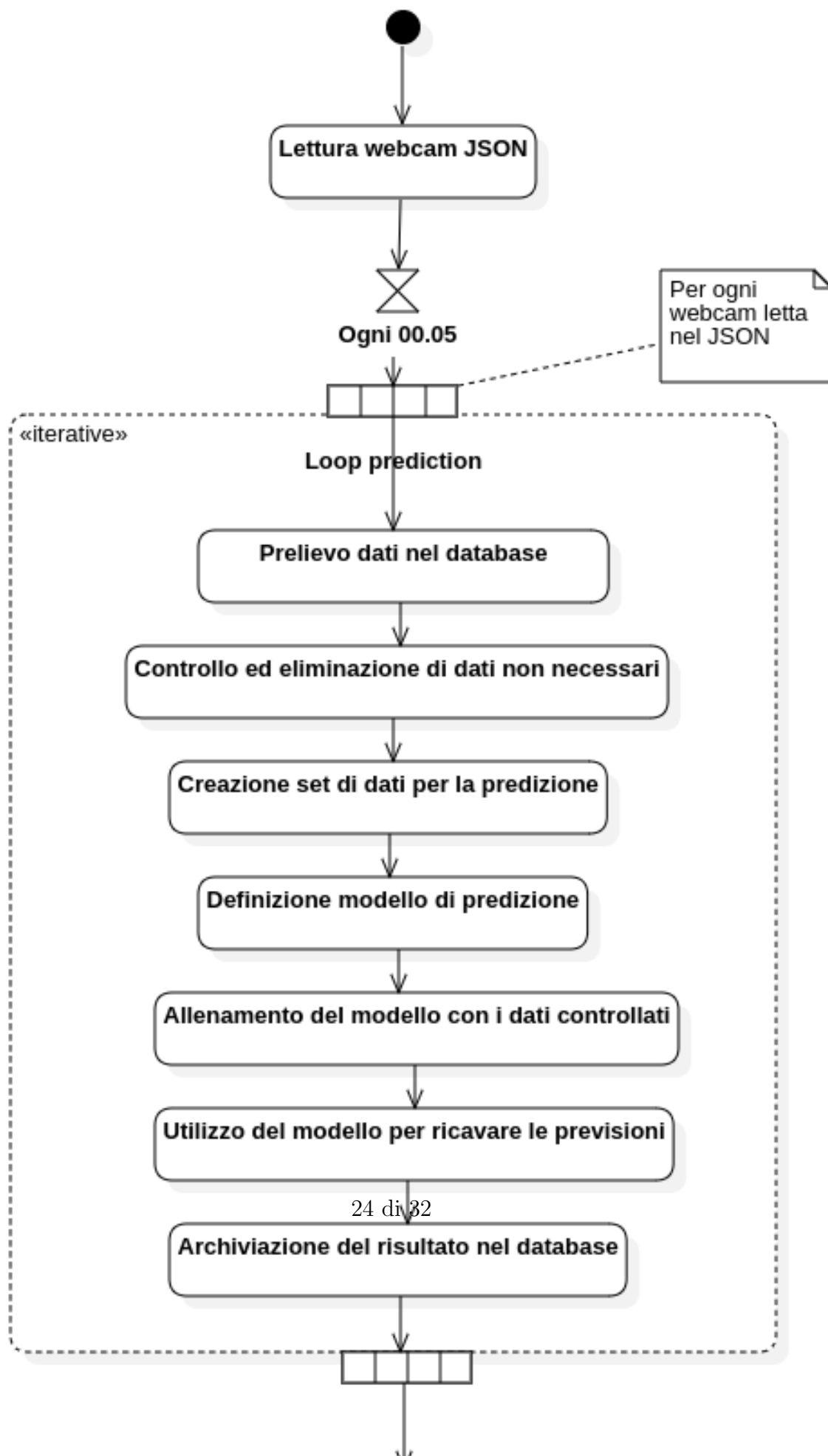


Figura 5.7: Diagramma dei package del modulo Acquisition



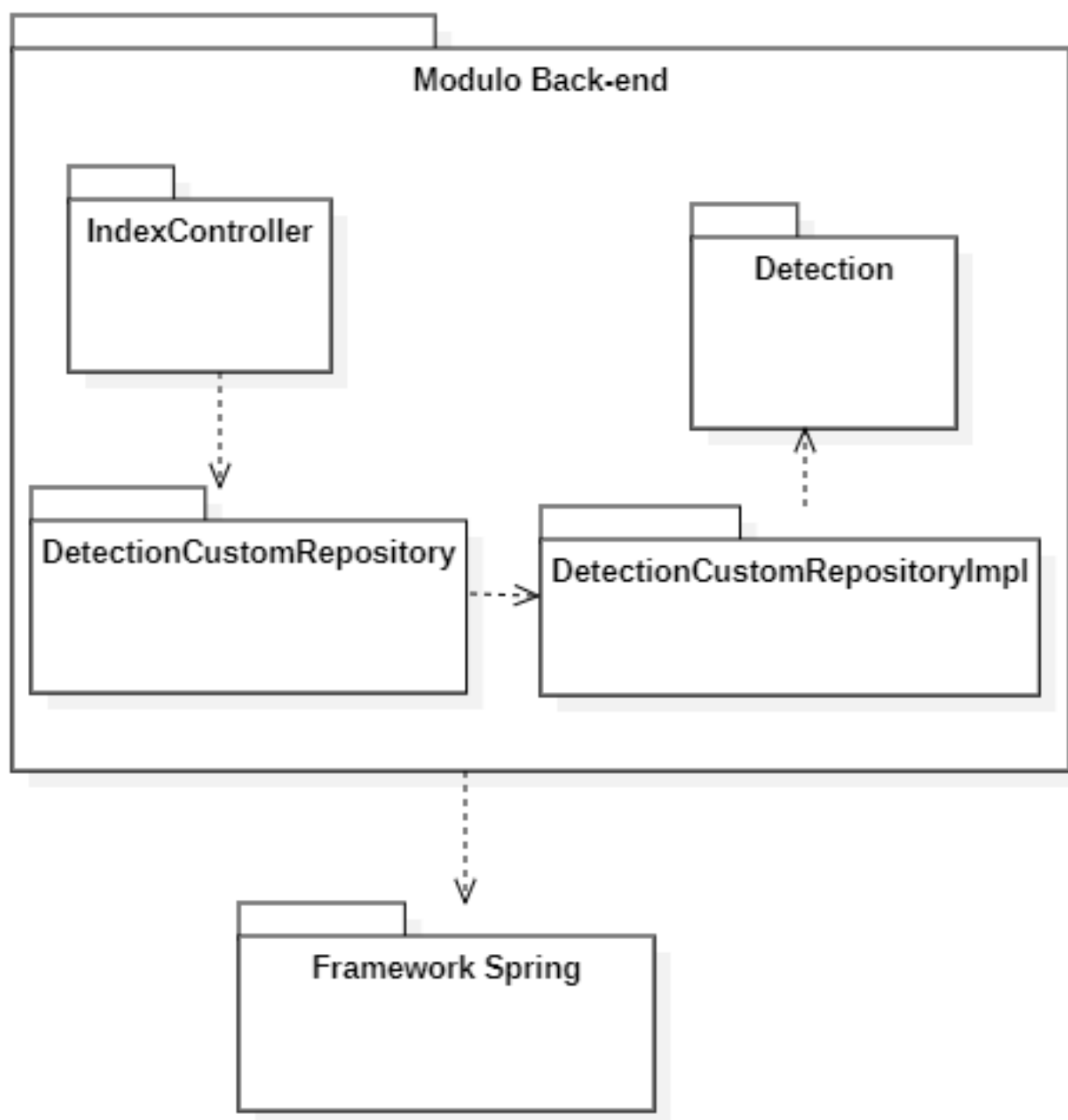


Figura 5.9: Diagramma dei package di Spring

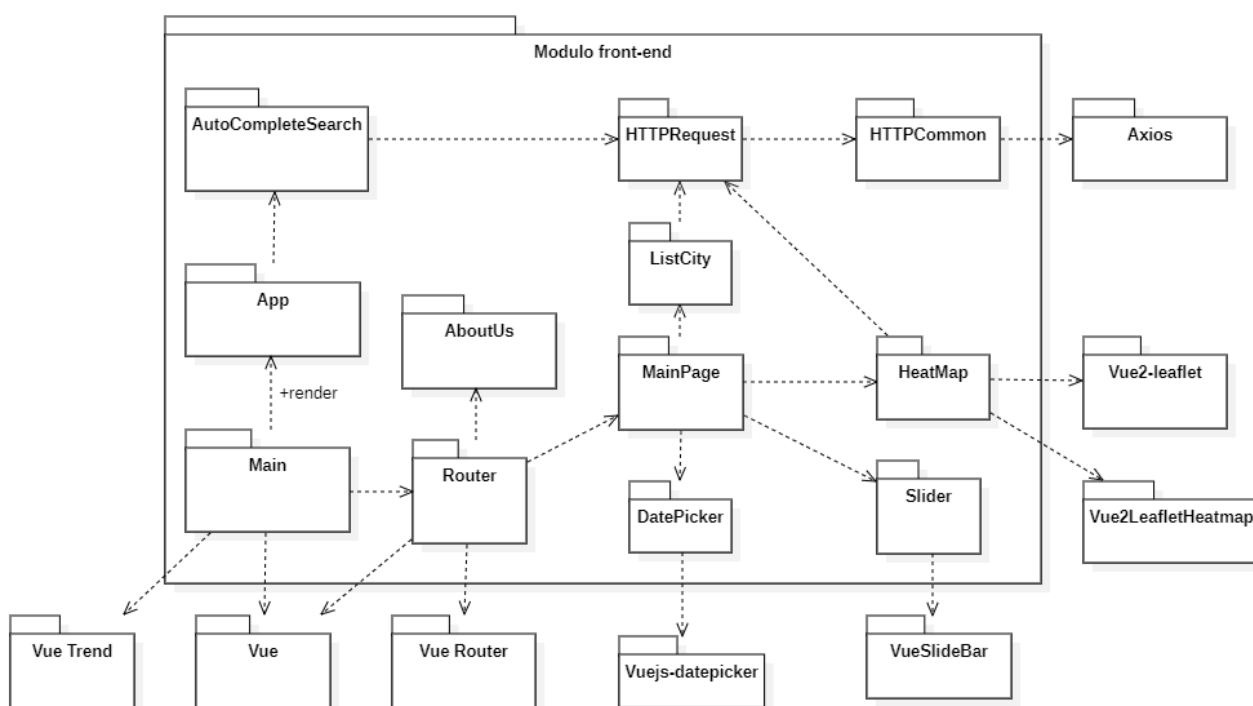


Figura 5.10: Diagramma dei package del modulo Acquisition

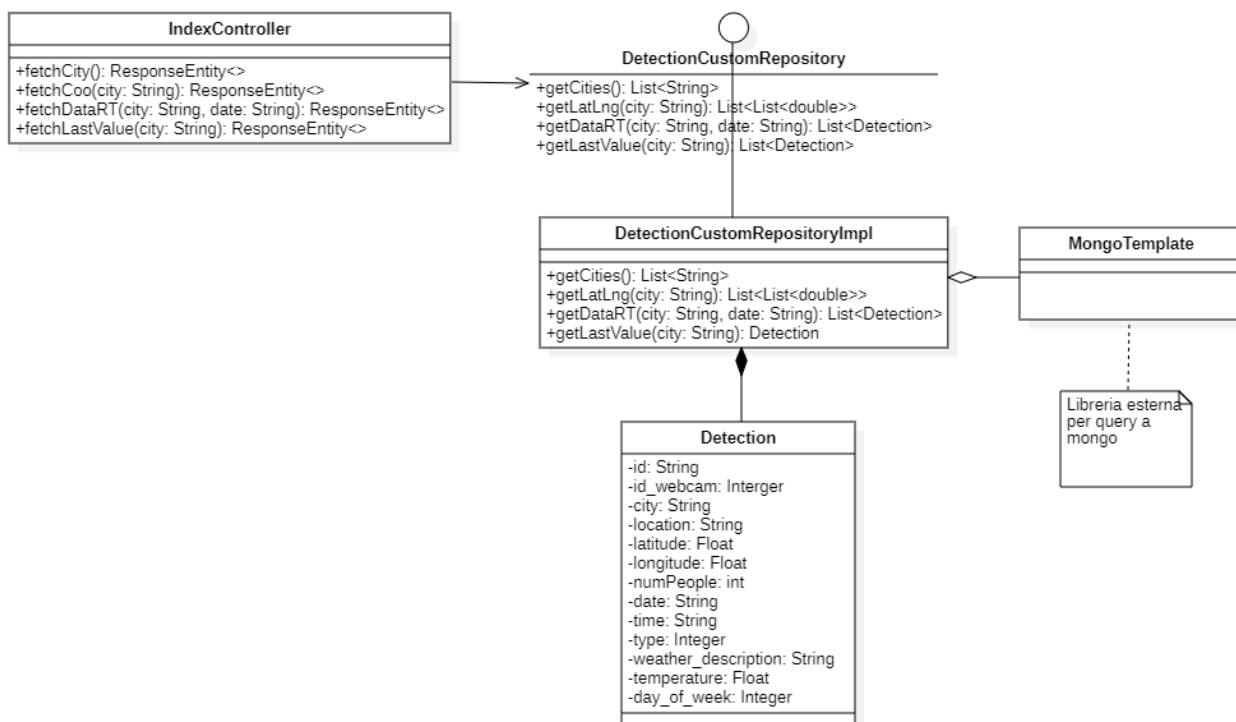


Figura 5.11: Diagramma delle classi di Spring

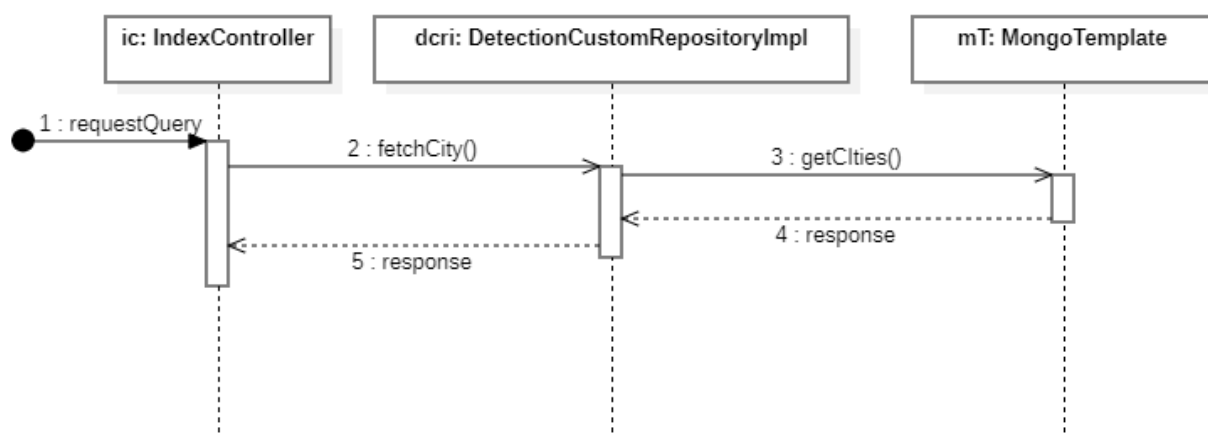


Figura 5.12: Diagramma di sequenza di Spring

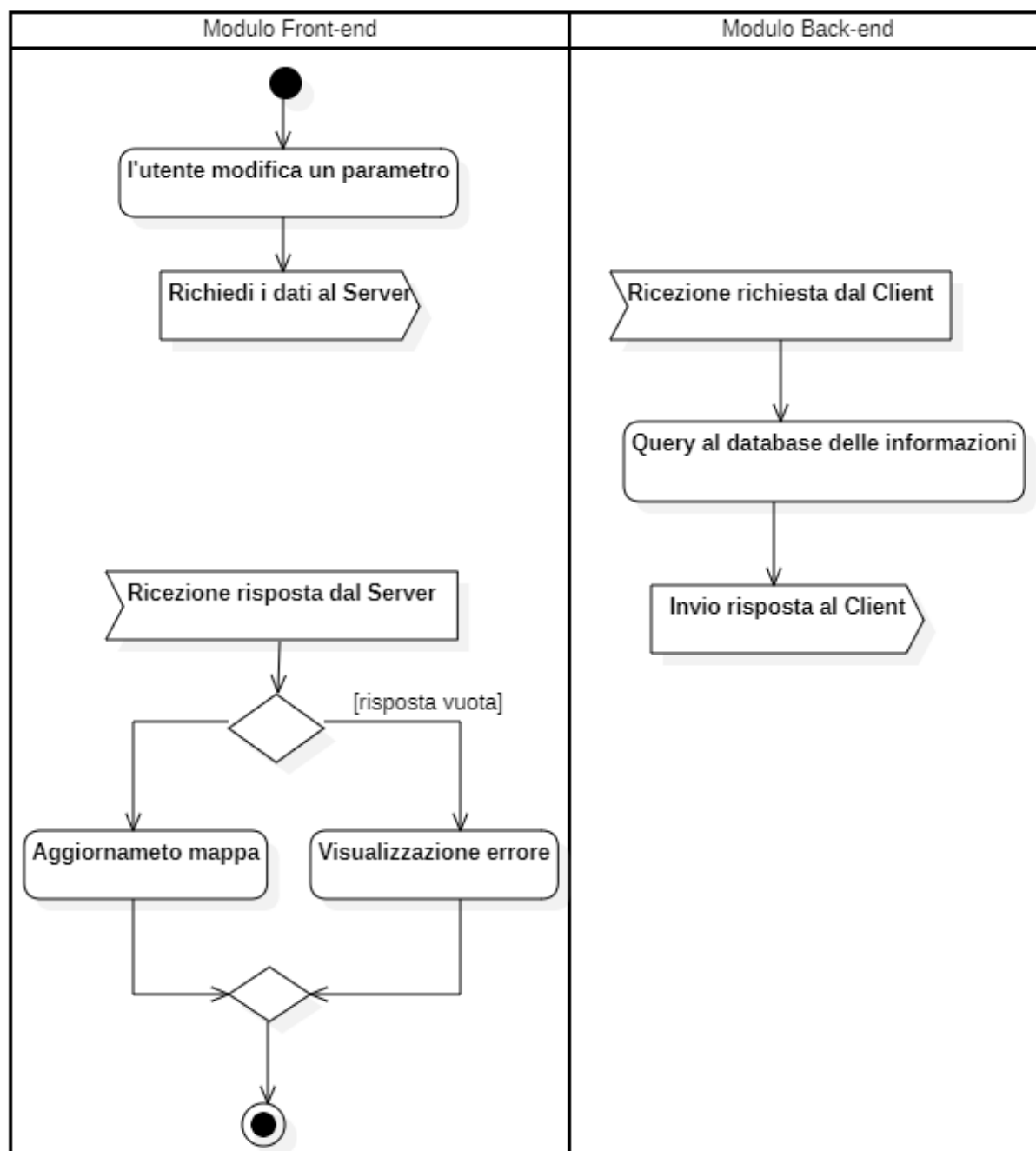


Figura 5.13: Diagramma di attività del modulo Web-app



6 Glossario

A

Application client-server

Applicazioni single-page

B

Back-end

Interfaccia con la quale il gestore di un sito web dinamico ne gestisce i contenuti e le funzionalità. A differenza del frontend_G, l'accesso al backend è riservato agli amministratori del sito che possono accedere dopo essersi autenticati.

Bootstrap

Build automation

In informatica è l'atto di scrivere o automatizzare un'ampia varietà di compiti che gli sviluppatori software fanno nelle loro attività quotidiane di sviluppo.

C

Client

CSS

Linguaggio utilizzato per la formattazione di documenti HTML. Serve per separare i contenuti delle pagine HTML dalla loro formattazione o layout, inoltre permette una programmazione più chiara e una più facile manutenibilità del codice stesso.

D

Database

Document-object manager

Duplicare

F

Framework

Utilizzato per descrivere la struttura operativa nella quale viene elaborato un dato software. Un framework, in generale, include software di supporto, librerie, un linguaggio per gli script e altri software che possono aiutare a mettere insieme le varie componenti di un progetto.



H

Heat-map

Rappresentazione grafica dei dati dove i singoli valori contenuti in una matrice sono rappresentati da colori.

J

Java

Linguaggio di programmazione ad alto livello, orientato agli oggetti e a tipizzazione statica, che si appoggia sull'omonima piattaforma software di esecuzione, specificamente progettato per essere il più possibile indipendente dalla piattaforma hardware di esecuzione. Le principali caratteristiche di Java sono la portabilità, cioè il codice sorgente è compilato in bytecode e può essere eseguito su ogni PC che ha JVM (Java Virtual Machine), e la robustezza.

Javascript

Linguaggio di programmazione orientato ad oggetti ed eventi. Utilizzato specificatamente per la programmazione Web lato client. Aggiunge al sito effetti dinamici tramite funzioni invocate da un'azione eseguita sulla pagine Web(es. click del mouse, movimento del mouse, caricamento pagina).

L

Layout

Leaflet

Libreria open-source di JavaScript che permette la realizzazione di mappe interattive che funzionano efficientemente sia su desktop che su mobile.

Linux

M

Machine-learnig

Metodo di analisi dati che automatizza la costruzione di modelli analitici. È una branca dell'Intelligenza Artificiale e si basa sull'idea che i sistemi possono imparare dai dati, identificare modelli autonomamente e prendere decisioni con un intervento umano ridotto al minimo.

Maven

(Apache) Maven è uno strumento di gestione progetti software basati su Java e Build Automation. Supporta anche altri linguaggi quali C#, Ruby e Scala. Mediante il file POM.xml vengono descritte le dipendenze fra il progetto e le varie versioni di librerie necessarie nonché le dipendenze fra di esse. Maven effettua automaticamente il download delle librerie necessarie tra i vari repository definiti scaricandoli in locale o in un repository centralizzato lato sviluppo.



Ciò permette un maggior controllo in caso si debba andare a cercare una determinata libreria.

MongoDB

MVC

N

Node.js

NoSQL

O

Open source

Un software open-source è reso tale per mezzo di una licenza attraverso cui i detentori dei diritti favoriscono la modifica, lo studio, l'utilizzo e la redistribuzione del codice sorgente.

P

Pom

Python

Python è un linguaggio di programmazione ad alto livello, rilasciato pubblicamente per la prima volta nel 1991 dal suo creatore Guido van Rossum, supporta diversi paradigmi di programmazione, come quello orientato agli oggetti (con supporto all'ereditarietà multipla), quello imperativo e quello funzionale, ed offre una tipizzazione dinamica forte. Python è un linguaggio pseudocompilato: un interprete si occupa di analizzare il codice sorgente e, se sintatticamente corretto, di eseguirlo. Questa caratteristica rende Python un linguaggio portabile. Una volta scritto un sorgente, esso può essere interpretato ed eseguito sulla gran parte delle piattaforme attualmente utilizzate, semplicemente basta la presenza della versione corretta dell'interprete.

R

Real-time

Repository

Ambiente di un sistema informativo in cui vengono conservati e gestiti file, documenti e meta-dati relativi ad un'attività_G di progetto.

S

Script

Server

Software

Spring

In informatica Spring è un framework_G open-source_G per lo sviluppo di applicazioni su piatta-



forma Java_G.

Spring Boot

U

Ubuntu

V

Vue.js

Framework_G open-source_G per lo sviluppo di applicazioni web, interfacce utente e applicazioni a singola pagina.

W

Web-app

X

XML