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	*Responsabile*	*Approvazione del documento per RQ*	-
v0.0.2	2021-04-04	Mattia Cocco	*Progettista*	*Stesura § 4*	Andrea Dorigo
v0.0.1	2021-04-02	Alfredo Graziano	*Progettista*	*Stesura § 1*	Igli Mezini

Indice

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$_G$, 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 soddisfando 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$_G$ per i pacchetti di Ubuntu$_G$ siano attive (Universe, Multiverse).

Questo passaggio è richiesto poiché GitHub$_G$ (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 scaricare 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 per ultimare l'aggiunta.

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 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 API Weather Forecast

Si tratta di una Web-API$_G$ che permette la raccolta dati delle previsioni del meteo, sia presenti che future. Viene utilizzata nel modulo Acquisition per associare i dati raccolti dalle webcams con i dati meteo della giornata, così da ottenere più informazioni relative ad una determinata città. Queste previsioni possono essere anche scaricate come file CSV$_G$ o JSON.

- link download: https://openweathermap.org/history .

4.1.3 Kafka

Kafka 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 .

4.1.4 MongoDB

MongoDB 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 non utilizza la classica struttura basata su tabelle, invece si basa sui tipi di documenti JSON, facilitando così l'integrazioni di alcuni tipi di dati.

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

4.1.5 Spring Boot

Spring$_G$ è un framework$_G$ open source$_G$ per lo sviluppo di applicazioni su piattaforma Java. A questo framework$_G$ sono associati altri progetti, in particolare Spring Boot 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 .

4.1.6 Apache Maven

È una tecnologia utilizza per la gestione software$_G$ basati su Java 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 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 .

4.1.7 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, e i relativi programmi, portabili su piattaforme hardware$_G$ diverse.

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

4.1.8 HTML 5

HTML, 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. Nel progetto questo linguaggio viene utilizzato per sviluppare la parte di web-app$_G$, interagendo con anche Java, CSS, Bootstrap e Vue.js.

4.1.9 CSS 3

Il CSS è il principale linguaggio utilizzato per definire la formattazione dei siti e pagine web. L'utilizzo del CSS permette di separare i contenuti della pagina HTML 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$_G$.

4.1.10 Leaflet

Leaflet 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/ .

4.1.11 Vue.js

È un framework$_G$ JavaScript, 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/ .

4.1.12 Node.js

È un runtime system open-source$_G$, orientato ad oggetti, per l'esecuzione di codice JavaScript. Molti moduli di questa tecnologia sono proprio scritti in JavaScript, ed essendo appunto open-source$_G$, programmatori esterni possono crearne ed aggiungerne altri. A differenza di JavaScript 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.

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

4.1.13 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/ .

4.1.14 JSON

Si tratta di un formato testuale necessario per l'esportazione ed importazione dei dati presenti nel *modulo* di salvataggio dati, mediante MongoDB, 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$_g$ multipiattaforma specializzata nella visione artificiale in tempo reale. È stata integrata nel *modulo* adibito alla cattura immagini in tempo reale, in linguaggio python.

- 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 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.

- 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$_G$.

- 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 ed ideato per lavorare assieme a MongoDB da Python.

- 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. Fornisce un grosso supporto a grandi matrici e array multidimensionli, 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. 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 PyUnit

Si tratta di una libreria per Python dedicata a creare e testare unit-test per programmi scritti con tale linguaggio.

- versione utilizzata: segue la stessa versione del linguaggio Python essendo una sua libreria integrata.
- link alla documentazione: https://docs.python.org/3/library/unittest.html .

4.2.9 Checkstyle

Strumento che permette di eseguire l'analisi statica del codice java utilizzato nello sviluppo di un progetto software$_G$.

- versione utilizzata: 2.17;
- link alla repo$_G$: https://github.com/checkstyle/checkstyle .

4.2.10 ESLint

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

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

4.2.11 Prettier

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

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

4.2.12 JUnit

È un semplice framework$_G$ per scrivere unit-test ripetibili.

- versione utilizzata: 4.12;
- link al sito: https://junit.org/junit5/ .

4.2.13 Mockito

Si tratta di un framework$_G$ open-source$_G$ di testing per Java. Consente la creazione di doppi oggetti di test in unit test automatici.

- versione utilizzata: 3.9.0;
- link al sito: https://site.mockito.org/ .

5 Architettura del Prodotto

L'architettura generale del software$_G$ *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 informazione 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 archiviare 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. La web-app$_G$ è composta da due sotto-moduli definiti rispettivamente back-end$_G$ e front-end$_G$, all'interno del primo è presente l'applicazione di Spring, la quale imposta un server per eseguire i servizi per il front end, mentre la seconda è la parte dell'applicazione che mostra i dati all'utente finale. Il back-end$_G$ e front-end$_G$ dialogano attraverso richieste di tipo HTTP, il back-end$_G$ mette a disposizione un server$_G$ con servizi di tipo REST che effettuano solo query al database di tipo GET. Il front-end$_G$ una volta ottenuta l'informazione modifica attraverso metodi javascript$_G$ la mappa che l'utente visualizza insieme alle sue componenti. Nella sezione successiva vengono inseriti i diagrammi di ogni modulo descritto, questo permette di avere una visione globale di ogni modulo nelle sue dipendenze a librerie esterne, nelle sue attività che svolgono e nelle sue componenti. Questi diagrammi sono stati scritti seguendo i principi UML.

5.1 Architettura modulo Acquisition

5.1.1 Diagramma dei Package

Il modulo Acquisition utilizza due librerie esterne a Python$_G$, la prima è la libreria di Kafka$_G$ per creare applicativi di tipo consumer e producer e la seconda è MongoEngine$_G$ per creare una connessione ad un database di MongoDB$_G$.

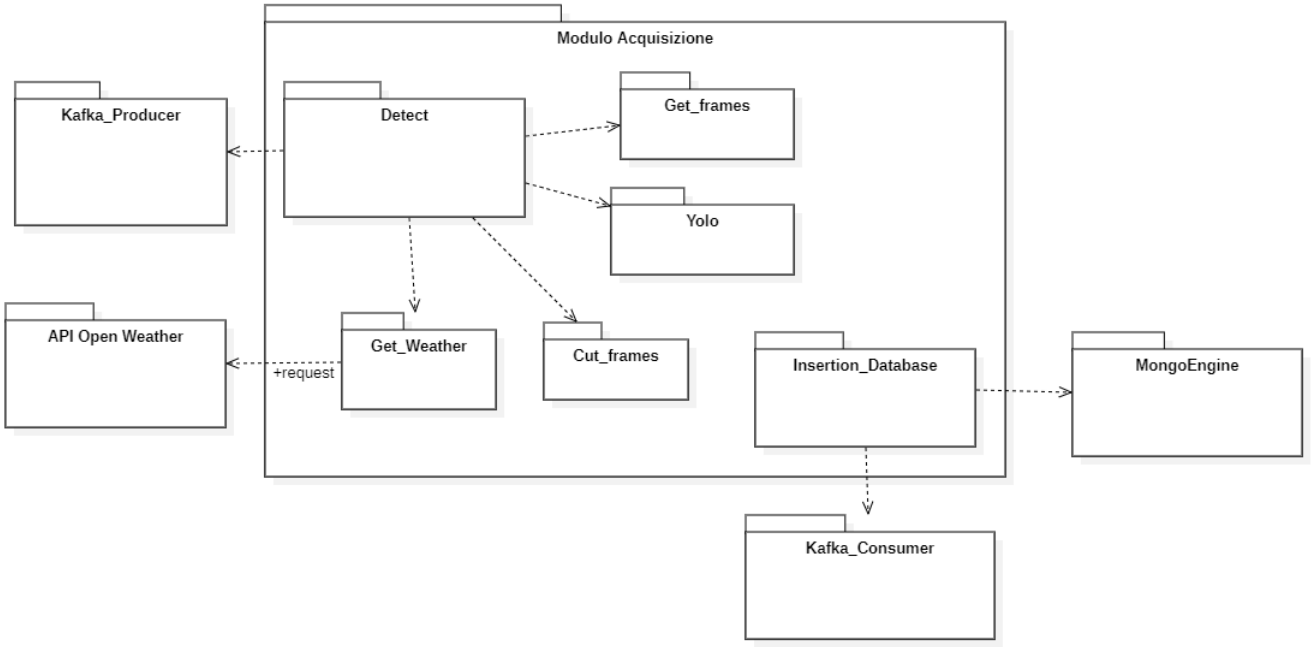

Figura 5.1: Diagramma dei package del modulo Acquisition

5.1.2 Diagrammi di attività

Di seguito vengono descritte le attività più importanti svolte nel modulo Acquisition.

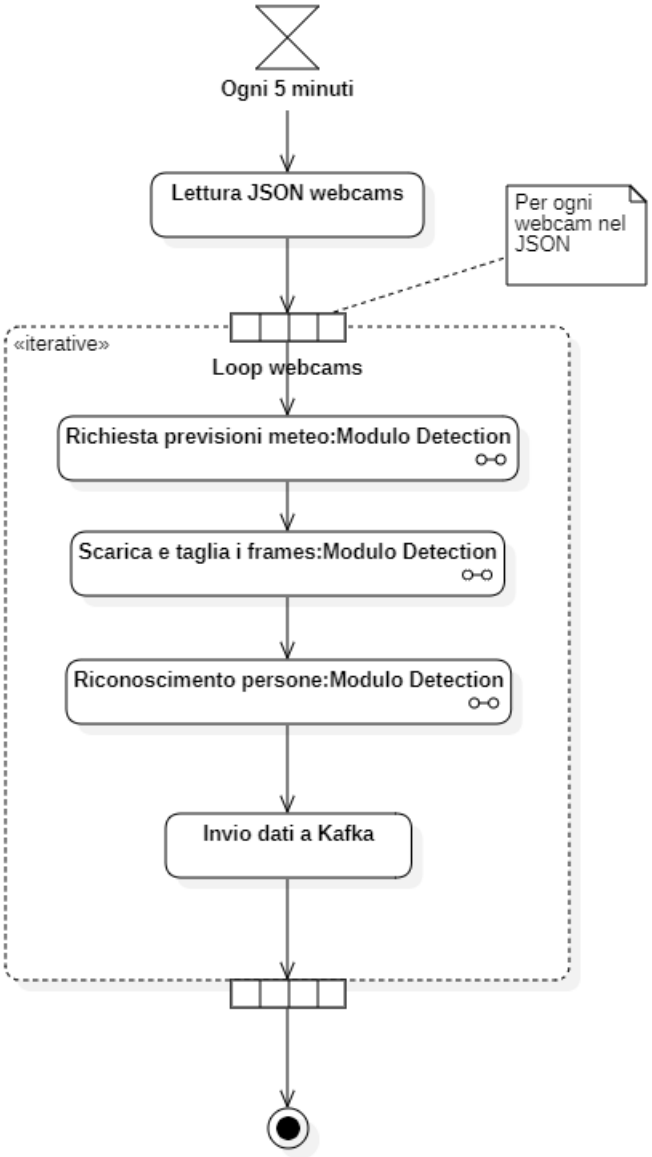

Figura 5.2: Diagramma di attività dell'eseguibile Detection

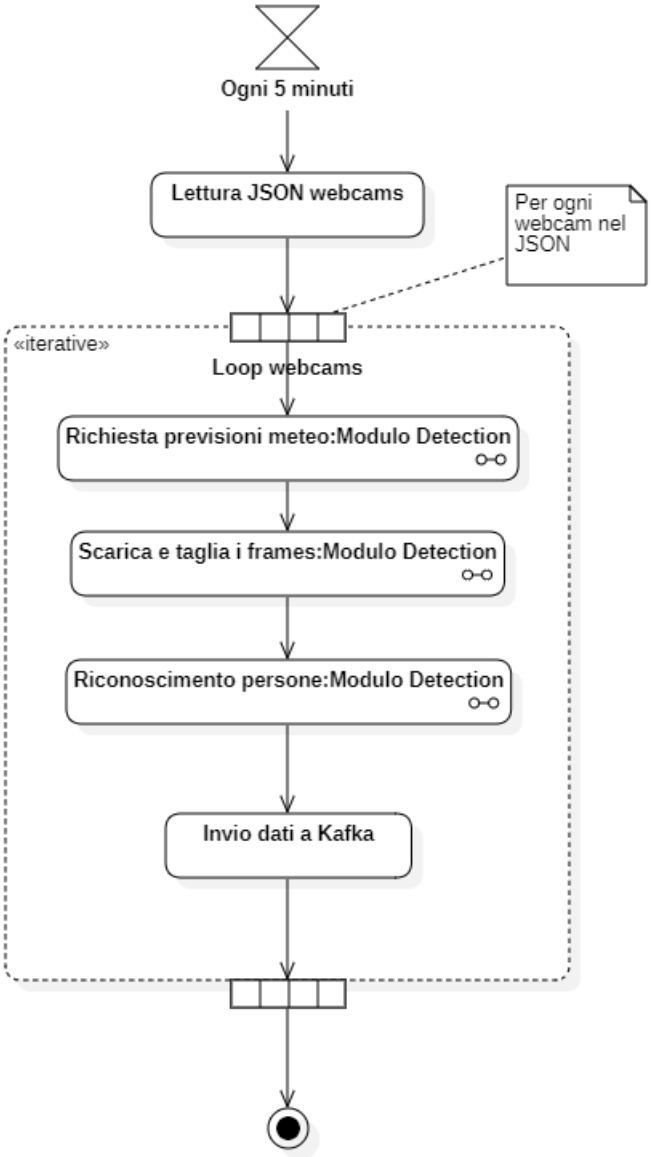

Figura 5.3: Diagramma di sotto-attività dell'acquisizione delle previsioni meteo

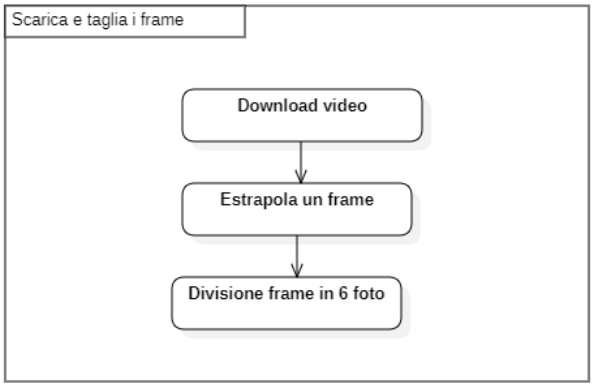

Figura 5.4: Diagramma di sotto-attività di download e taglio frame

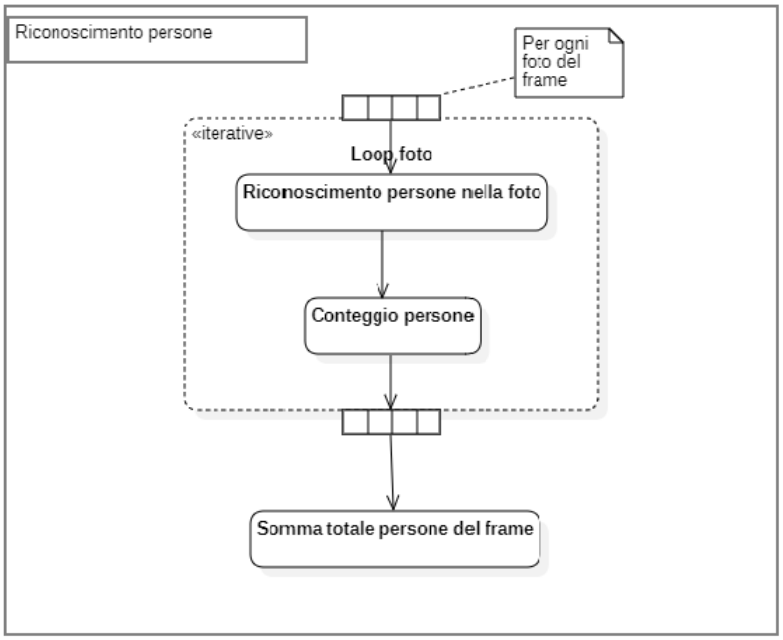

Figura 5.5: Diagramma di sotto-attività del conta persone

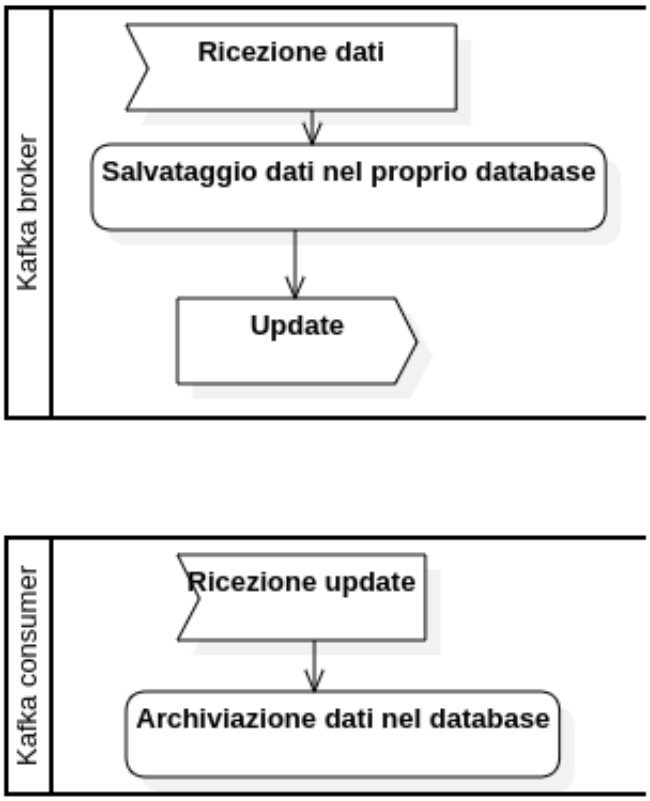

Figura 5.6: Diagramma di attività di Kafka

5.2 Architettura modulo Prediction

5.2.1 Diagramma dei package

Nel modulo Prediction vengono utilizzate le librerie esterne Pandas$_G$, MongoEngine$_G$ e Scikit-Learn$_G$ (abbreviato in Sklearn nella libreria). La libreria più importante è Scikit-Learn$_G$ della quale utilizziamo i metodi per il Preprocessing dei dati, la creazione di modelli con Model_selection e il tipo di modello per generare le predizioni, il Random Forest Regression.

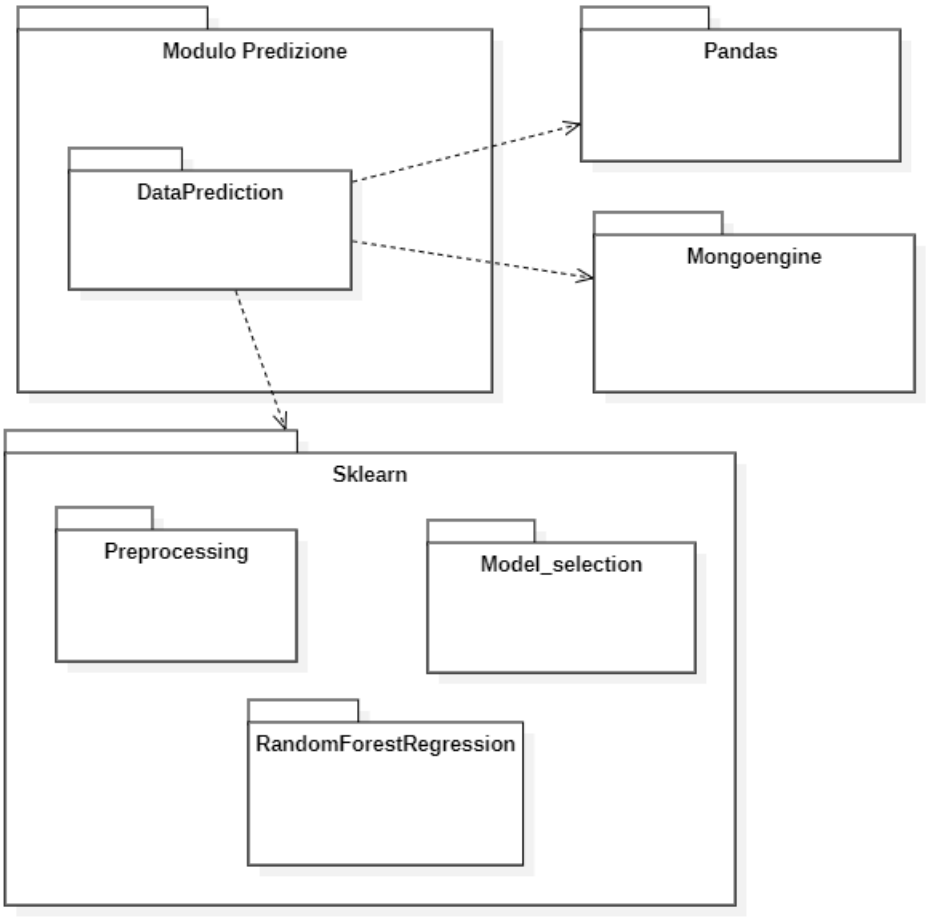

Figura 5.7: Diagramma dei package del modulo Acquisition

5.2.2 Diagramma di attività

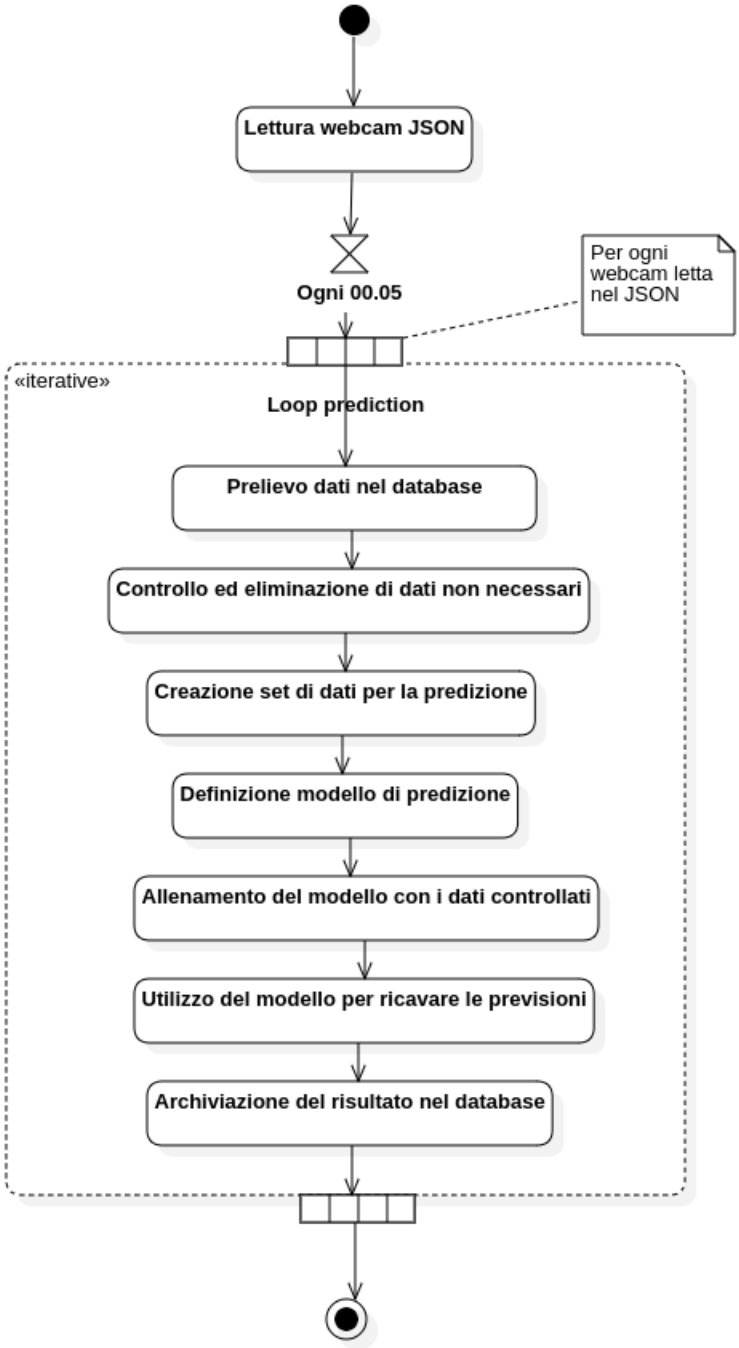

Figura 5.8: Diagramma di attività dell'eseguibile Detection

5.3 Architettura modulo Web-app

5.3.1 Diagrammi dei package

Il modulo della web-app$_G$, come descritto in precedenza, è diviso in due sotto-moduli rispettivamente per il back-end$_G$ e front-end$_G$. Di seguito vengono visualizzate le dipendenze dei due sotto-moduli, per il back-end$_G$ è solo necessaria la libreria del framework$_G$ Spring$_G$, mentre per il front-end$_G$ sono necessarie varie librerie per ogni componente della web-app$_G$.

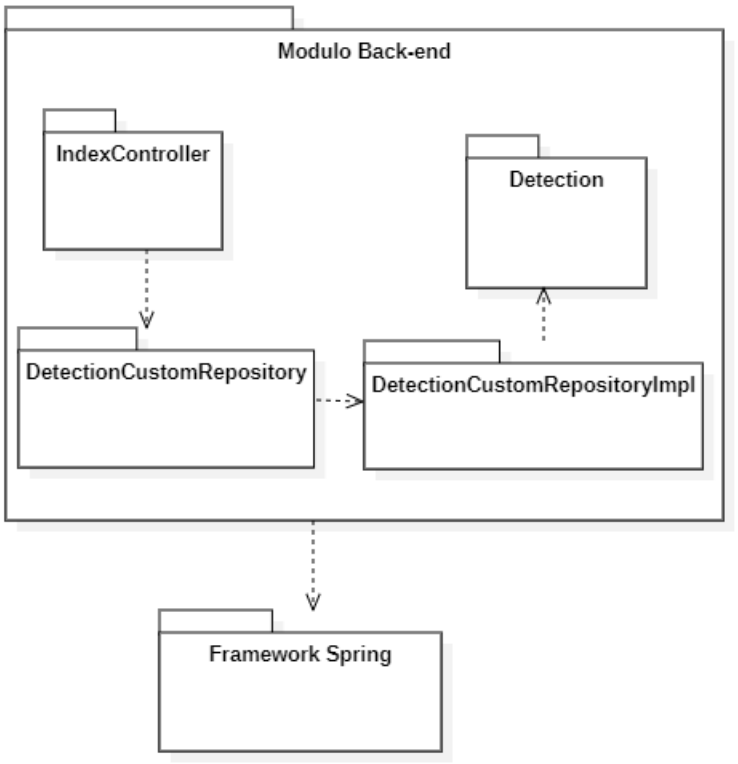

Figura 5.9: Diagramma dei package di Spring

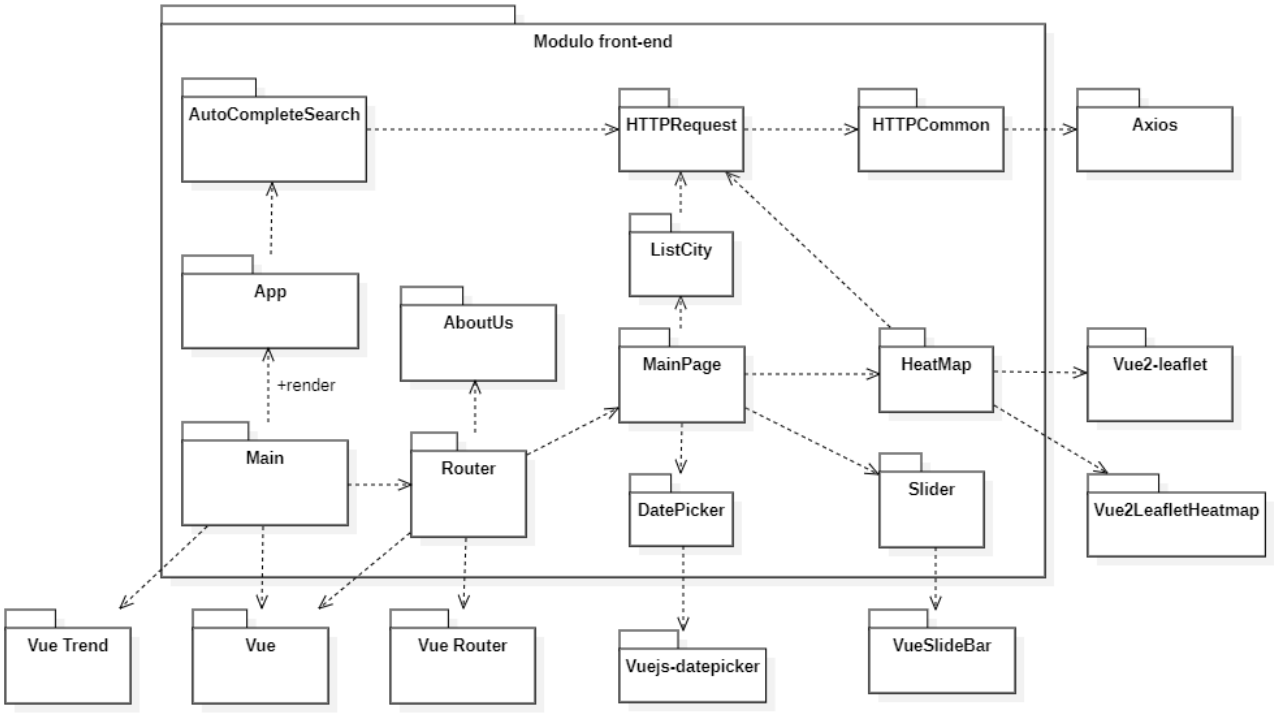

Figura 5.10: Diagramma dei package del modulo Acquisition

5.3.2 Diagrammi delle classi

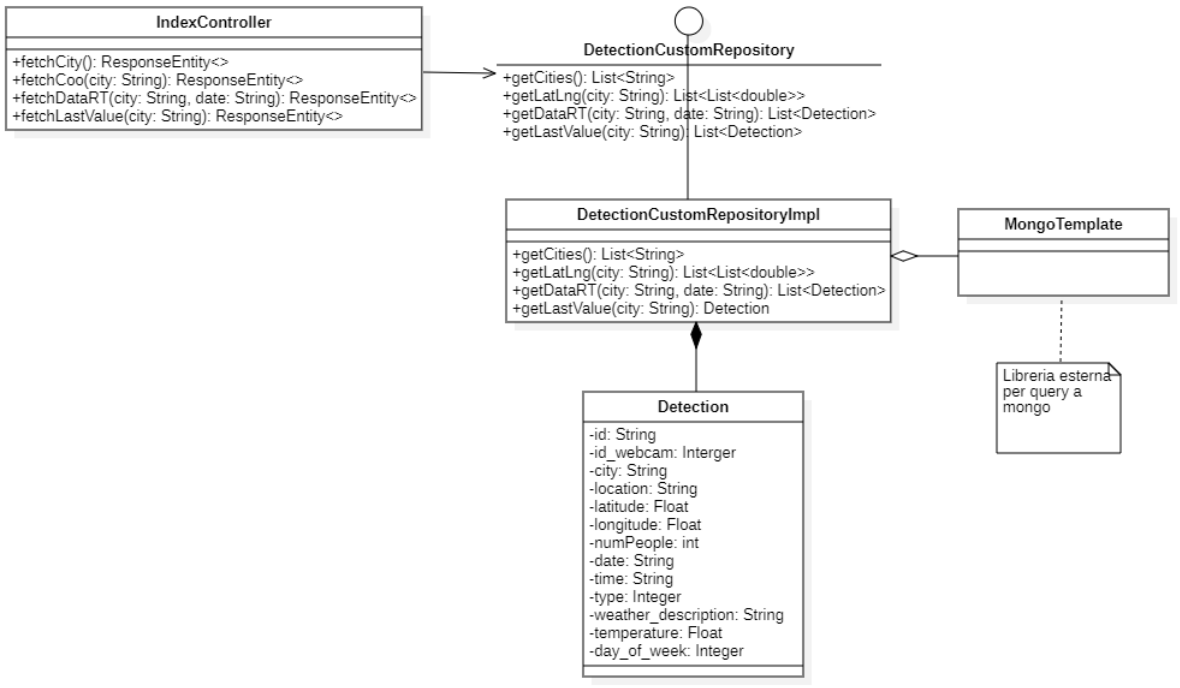

Figura 5.11: Diagramma delle classi di Spring

5.3.3 Diagramma di sequenza

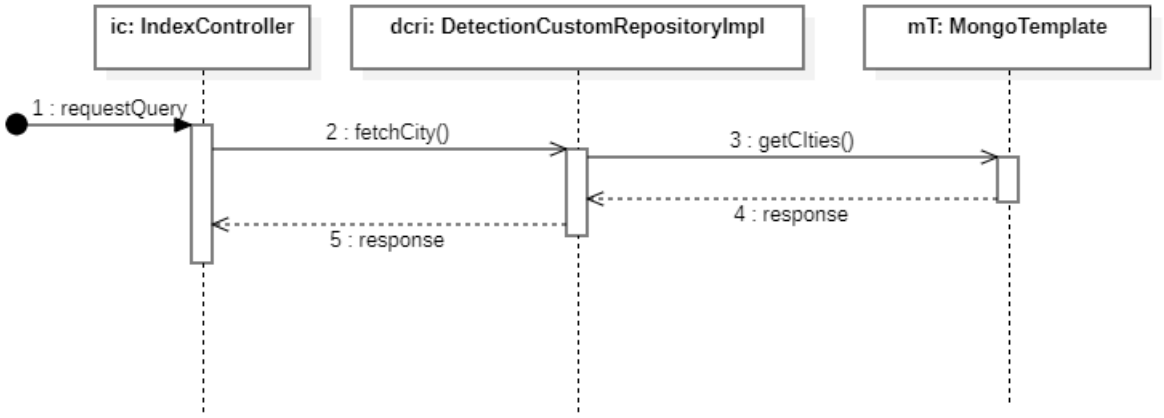

Figura 5.12: Diagramma di sequenza di Spring

5.3.4 Diagramma di attività

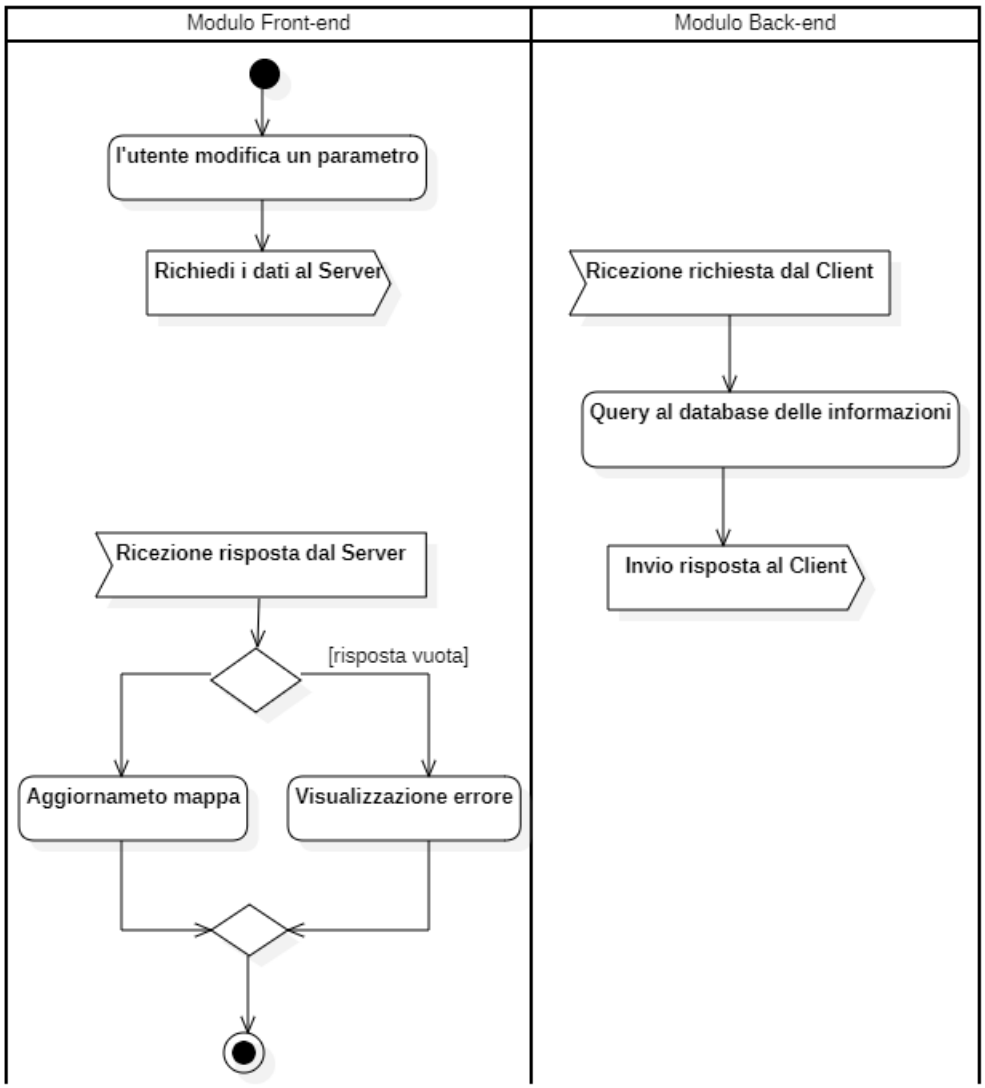

Figura 5.13: Diagramma di attività del modulo Web-app

6 Glossario

A

Application client-server
Indica un'applicazione basata su un'architettura di rete nella quale, generalmente, un computer si collega ad un server$_G$ per le fruizione di und eterminato servizio.

Applicazioni single-page
S'intende una web-app$_G$ o un sito internet che può essere usato, o consultato, tramite una singola pagina, fornendo così un'esperienza più fluida e intuitiva all'utente.

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.

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

C

Client
In ambito informatico si intendo i dispositivi collegati ad un server ed in grado di scambiarvici informazioni.

D

Database
Insieme strutturati, ovvero omogenei per contenuti e formato, rappresentanti, digitalmente, un archivio dati.

Document-object manager
Abbreviato in "DOM", in italiano è tradotto letteralmente modello a oggetti del documento, è una forma di rappresentazione dei documenti strutturati come modello orientato agli oggetti.

F

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

G

Git
Sistema di controllo gratuito a versione distribuita progettato per tenere traccia del lavoro svolto durante l'intero periodo di sviluppo del software. Utilizzato anche per tenere traccia di tutte le modifiche fatte nei file. I suoi punti di forza sono l'integrità dei dati e il supporto per flussi di lavoro distribuiti e non lineari.

GitHub
GitHub$_G$ è un servizio di hosting per progetti software. Il nome deriva dal fatto che esso è una implementazione dello strumento di controllo versione distribuito Git$_G$. **H**

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

L

Layout
In informatica si intende la disposizione degli elementi che costituiscono una pagina internet.

Linux
Si tratta di una famiglia di sistemi operativi open-source$_G$ pubblicati poi in varie distribuzioni.

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.

MVC
È un pattern architetturale, l'acronimo MVC$_G$ sta per *Model-View-Controller*. Il *model* si occupa della rappresentazione dei dati in oggetti. Il *view* gestisce la rappresentazione grafica di

essi. Infine il *controller* si occupa delle interazioni degli utenti.

N

NoSQL
È un movimento che promuove sistemi software dove la persistenza dei dati è in generale caratterizzata dal fatto di non utilizzare il modello relazionale, di solito usato dalle basi di dati tradizionali.

NPM
Gestore di pacchetti per il linguaggio di programmazione JavaScript$_G$, consiste in un client$_G$ da linea di comando, chiamato anch'esso npm, e un database$_G$ online di pacchetti pubblici e privati, chiamato npm registry.

O

Open source
Un software open-source$_G$ 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
Acronimo di Project Object Model è l'unità fondamentale di lavoro di Maven al cui interno sono presenti le configurazioni e le informazioni riferite al progetto.

R
Real-time
Tradotto in italiano: in tempo reale.

Repository-Repo
Ambiente di un sistema informativo in cui vengono conservati e gestiti file, documenti e metadati relativi ad un'attività di progetto.

Run-time system
Termine utilizzato per indicare un software che fornisce i servizi necessari all'esecuzione di un programma.

S
Script

File contenente codice eseguibile.

Server
È un componente, o sotto sistema, adibito all'elaborazione e gestione del traffico dati fornito da servizi verso altre componenti.

Software
È l'insieme delle procedure e delle istruzioni in un sistema di elaborazione dati.

Spring
In informatica Spring è un framework$_G$ open-source$_G$ per lo sviluppo di applicazioni su piattaforma Java$_G$.

Streaming
Identifica un flusso di dati audio/video trasmessi da una sorgente a una o più destinazioni tramite una rete telematica. Questi dati vengono riprodotti man mano che arrivano a destinazione.

T
Tomcat
(Apache) è un server web open-source$_G$.

U
Ubuntu
Sistema operativo basato su Linux$_G$, più precisamentesu Debian.

V
W
Web API
Un'API Web è un'interfaccia di programmazione dell'applicazione per un server Web o un browser Web. In questo caso è un sito sul quale si appoggia il software$_G$ per acquisire delle informazioni.

Web-app/Applicazioni web
In ambito informatico si intende un'applicazione web, ovvero applicazioni fruibili mediante via web, come un sito internet che offre determinati servizi al client$_G$.

X
XML
È un formato di file appertenente a script$_G$ scritti linguaggio col medesimo nome. Si tratta di un linguaggio di markup ossia basato su un meccanismo sintattico che consente di definire e

controllare il significato degli elementi contenuti in un documento o in un testo.