



# Allegato Tecnico

## Jawa Druids

<b>Versione</b>	1.0.0
<b>Data approvazione</b>	2021-04-16
<b>Responsabile</b>	Alfredo Graziano
<b>Redattori</b>	Andrea Cecchin Mattia Cocco
<b>Verificatori</b>	Emma Roveroni Margherita Mitillo
<b>Stato</b>	Approvato
<b>Lista distribuzione</b>	Jawa Druids Prof. Tullio Vardanega Prof. Riccardo Cardin
<b>Uso</b>	Esterno

## Sommario

Il presente documento contiene le scelte architetturali che il gruppo *Jawa Druids* ha effettuato ai fini realizzativi del progetto. Contiene i design pattern e i diagrammi di attività, sequenza, classi e package.



# Indice

<b>1</b>	<b>Introduzione</b>	<b>4</b>
1.1	Scopo del documento . . . . .	4
1.2	Scopo del prodotto . . . . .	4
1.3	Glossario . . . . .	4
1.4	Riferimenti . . . . .	4
1.4.1	Riferimenti normativi . . . . .	4
1.4.2	Riferimenti informativi . . . . .	5
<b>2</b>	<b>Architettura del prodotto</b>	<b>6</b>
2.1	Descrizione generale . . . . .	6
2.2	Architettura acquisizione . . . . .	6
2.2.1	Diagramma dei package dell'acquisizione . . . . .	7
2.2.2	Diagramma di attività . . . . .	8
2.2.2.1	Diagramma di sotto attività di richiesta previsioni meteo . . . . .	9
2.2.2.2	Diagramma di sotto attività di scarica e taglia i frame . . . . .	10
2.2.2.3	Diagramma di sotto attività di riconoscimento persone . . . . .	11
2.2.2.4	Diagramma di attività di Kafka . . . . .	12
2.3	Architettura della predizione . . . . .	12
2.3.1	Diagrammi dei package . . . . .	13
2.3.2	Diagrammi di attività . . . . .	14
2.4	Architettura Web-App . . . . .	15
2.4.1	Diagrammi dei package del modulo back-end . . . . .	15
2.4.2	Diagramma delle classi del modulo back-end . . . . .	16
2.4.3	Diagramma di sequenza del modulo back-end . . . . .	16
2.4.4	Diagrammi dei package del modulo front-end . . . . .	17
2.4.5	Diagrammi delle attività . . . . .	18
<b>3</b>	<b>Requisiti soddisfatti</b>	<b>20</b>
3.1	Tabella requisiti funzionali . . . . .	20
3.2	Grafici requisiti funzionali . . . . .	22



## Elenco delle tabelle

3.1	Tabella requisiti funzionali . . . . .	22
-----	--	----



## Elenco delle figure

2.1	Diagramma dei package del modulo acquisition . . . . .	7
2.2	Diagramma di attività del modulo Acquisition . . . . .	8
2.3	Diagramma di sotto attività di richiesta previsioni meteo . . . . .	9
2.4	Diagramma di sotto attività di scarica e taglia i frame . . . . .	10
2.5	Diagramma di sotto attività di riconoscimento persone . . . . .	11
2.6	Diagramma di attività Kafka . . . . .	12
2.7	Diagramma dei package del modulo predizione . . . . .	13
2.8	Diagramma di attività del modulo predizione . . . . .	14
2.9	Diagramma dei package del modulo back-end . . . . .	15
2.10	Diagramma delle classi del modulo back-end . . . . .	16
2.11	Diagramma di sequenza del modulo back-end . . . . .	17
2.12	Diagramma dei package del modulo front-end . . . . .	18
2.13	Diagramma di attività tra modulo front-end e modulo back-end . . . . .	19
3.1	Diagramma a torta di tutti i requisiti funzionali soddisfatti . . . . .	23
3.2	Diagramma a torta di tutti i requisiti funzionali obbligatori soddisfatti . . . . .	24



# 1 Introduzione

## 1.1 Scopo del documento

Lo scopo del documento è quello di elencare e motivare le scelte architetturelle fatte dal gruppo Jawa Druids, per quanto riguarda il progetto GDP: Gathering Detection Platform.

## 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 sia di prevedere assembramenti in intervalli futuri di tempo sia di offrire una storicizzazione dei dati passati.

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

All'interno della documentazione viene fornito un *Glossario v3.0.0*, con l'obiettivo di assistere il lettore specificando il significato e contesto d'utilizzo di alcuni termini strettamente tecnici o ambigui, segnalati con una *G* a pedice.

## 1.4 Riferimenti

### 1.4.1 Riferimenti normativi

- *Norme di Progetto 3.0.0*;
- *Capitolato d'appalto*; <https://www.math.unipd.it/~tullio/IS-1/2020/Progetto/C3.pdf>



## 1.4.2 Riferimenti informativi

- *Dispensa diagrammi delle classi:*  
[https://www.math.unipd.it/~rcardin/swea/2021/Diagrammi%20delle%20Classi\\_4x4.pdf](https://www.math.unipd.it/~rcardin/swea/2021/Diagrammi%20delle%20Classi_4x4.pdf)
- *Dispensa diagrammi dei package:*  
[https://www.math.unipd.it/~rcardin/swea/2021/Diagrammi%20dei%20Package\\_4x4.pdf](https://www.math.unipd.it/~rcardin/swea/2021/Diagrammi%20dei%20Package_4x4.pdf)
- *Dispensa diagrammi di sequenza:*  
[https://www.math.unipd.it/~rcardin/swea/2021/Diagrammi%20di%20Sequenza\\_4x4.pdf](https://www.math.unipd.it/~rcardin/swea/2021/Diagrammi%20di%20Sequenza_4x4.pdf)
- *Dispensa diagrammi di attività:*  
[https://www.math.unipd.it/~rcardin/swea/2021/Diagrammi%20di%20Attivit%c3%a0\\_4x4.pdf](https://www.math.unipd.it/~rcardin/swea/2021/Diagrammi%20di%20Attivit%c3%a0_4x4.pdf)
- *Dispensa principi SOLID:*  
[https://www.math.unipd.it/~rcardin/swea/2021/SOLID%20Principles%20of%20Object-Oriented%20Design\\_4x4.pdf](https://www.math.unipd.it/~rcardin/swea/2021/SOLID%20Principles%20of%20Object-Oriented%20Design_4x4.pdf)
- *Dispensa REST-based API:*  
<https://www.math.unipd.it/~rcardin/web/2021/L03.pdf>



## 2 Architettura del prodotto

### 2.1 Descrizione generale

In fase di progettazione, il gruppo *Jawa Druids* ha deciso di suddividere la modellazione architetture di *Gathering-Detection-Platform* in tre distinti moduli, tutti indipendenti tra loro. Il primo modulo si occupa solamente di leggere, tramite file  $\text{JSON}_G$ , tutte le webcam disponibili per poi effettuare il riconoscimento persone tramite i frame scaricati. Successivamente i dati estrapolati verranno inviati al database. Il secondo modulo, il  $\text{machine-learning}_G$ , si occupa di recuperare questi dati dal database per lavorarli producendo predizioni per le ore future. Infine il terzo modulo, la  $\text{web-app}_G$  vera e propria, si occuperà di rappresentare graficamente i dati all'interno del database mediante una  $\text{heat-map}_G$  e farli visualizzare all'utente.

### 2.2 Architettura acquisizione

L'architettura riguardante il modulo di acquisizione, ovvero il primo modulo del software, è molto semplice ed intuitiva. Non vi è alcuna classe e si basa su una programmazione procedurale. Nel *detect.py*, ovvero lo script principale del modulo, vengono richiamate le funzioni, in maniera sequenziale, per scaricare e manipolare i dati delle webcam. La scelta dell'utilizzo di un paradigma procedurale risiede nel fatto che la creazione di oggetti e il loro utilizzo risultavano, nell'insieme, più complicati mentre chiamando delle semplici funzioni esterne il programma risultava più leggibile ed efficiente. Gli unici oggetti presenti in *detect.py* sono quelli di tipo *data*, necessari per la giusta esecuzione dello script. Di seguito vengono riportati i diagrammi relativi all'*attività* e di *package*.



## 2.2.1 Diagramma dei package dell'acquisizione

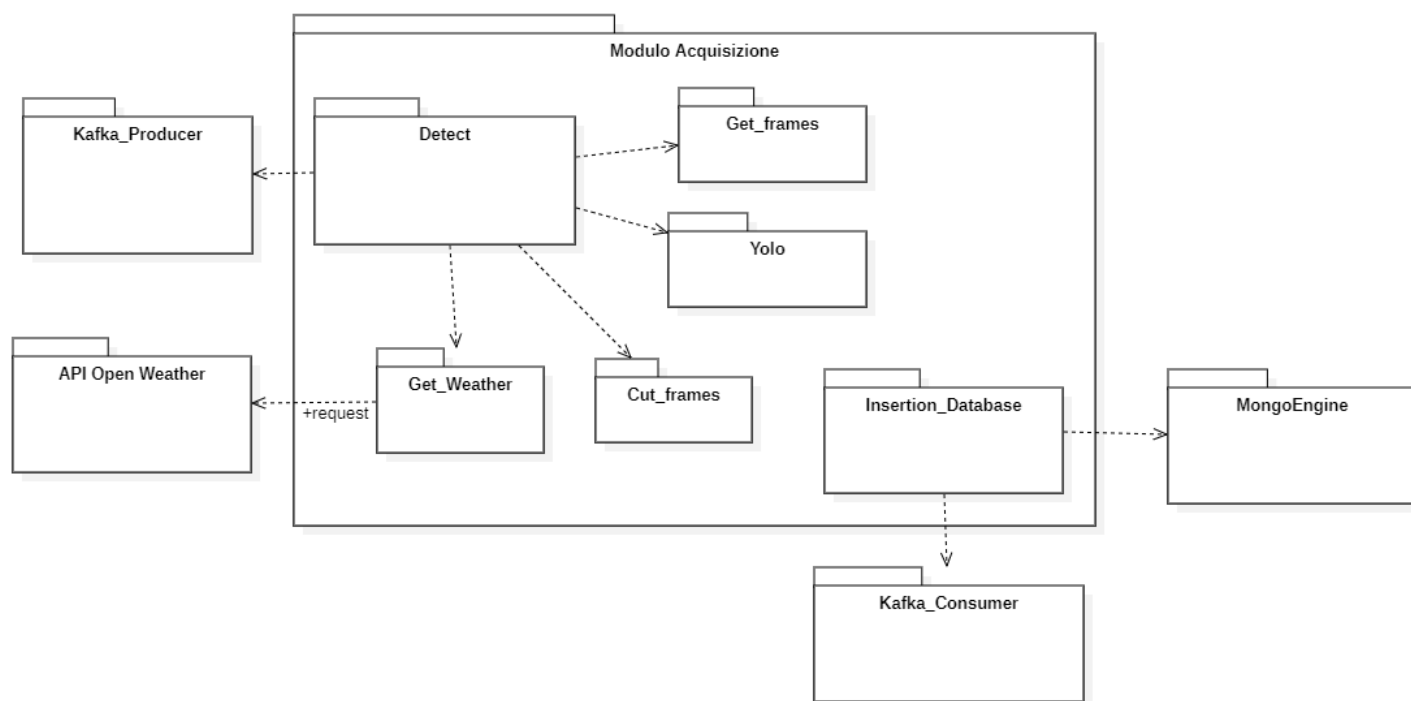


Figura 2.1: Diagramma dei package del modulo acquisition





## 2.2.2 Diagramma di attività

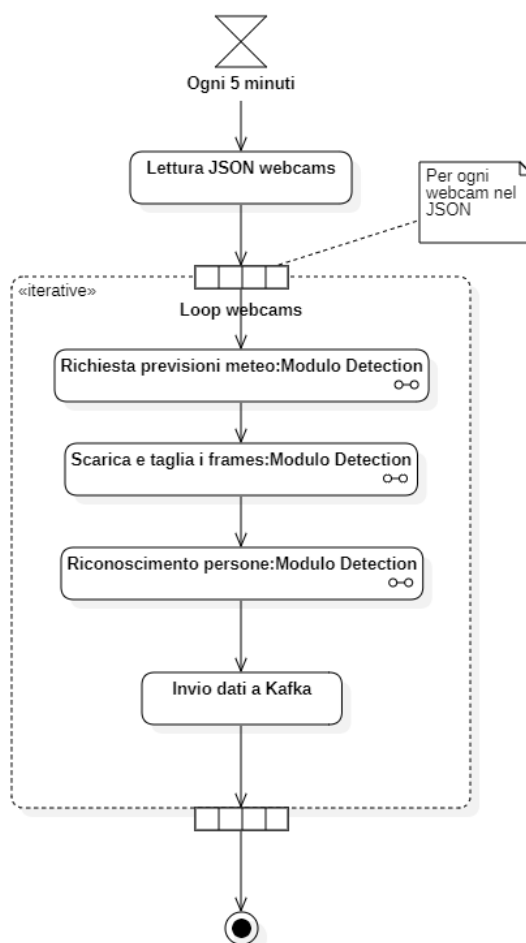


Figura 2.2: Diagramma di attività del modulo Acquisition

### 2.2.2.1 Diagramma di sotto attività di richiesta previsioni meteo

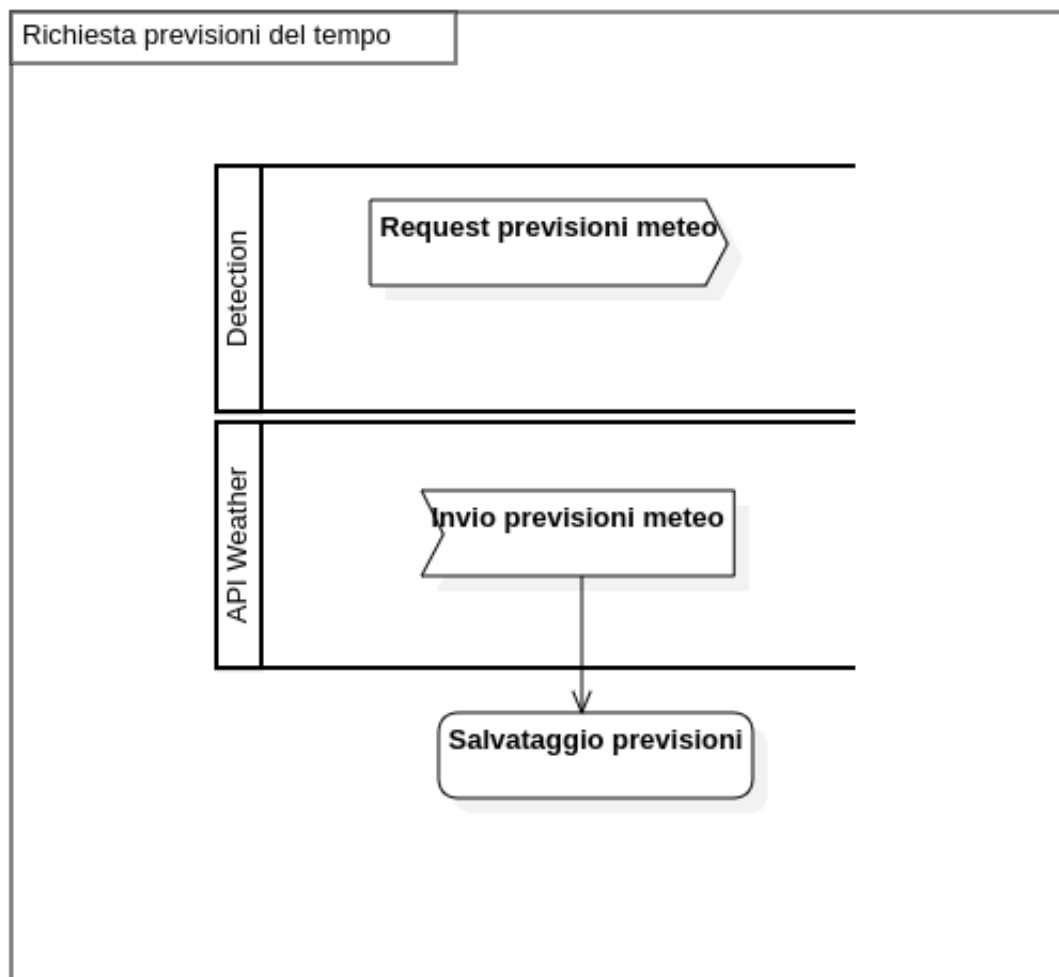


Figura 2.3: Diagramma di sotto attività di richiesta previsioni meteo

### 2.2.2.2 Diagramma di sotto attività di scarica e taglia i frame

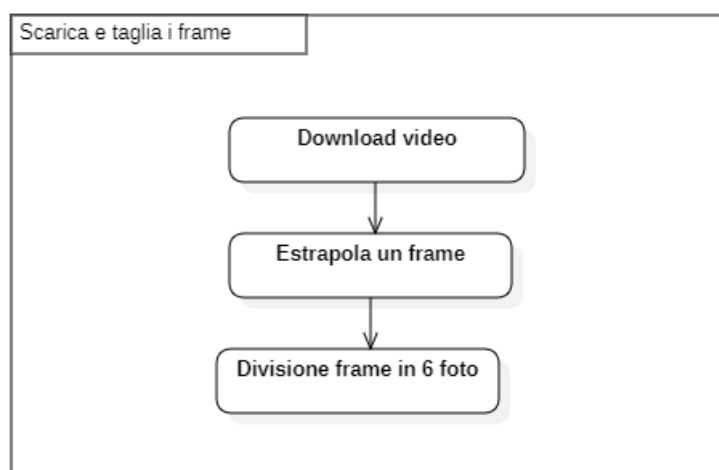


Figura 2.4: Diagramma di sotto attività di scarica e taglia i frame

### 2.2.2.3 Diagramma di sotto attività di riconoscimento persone

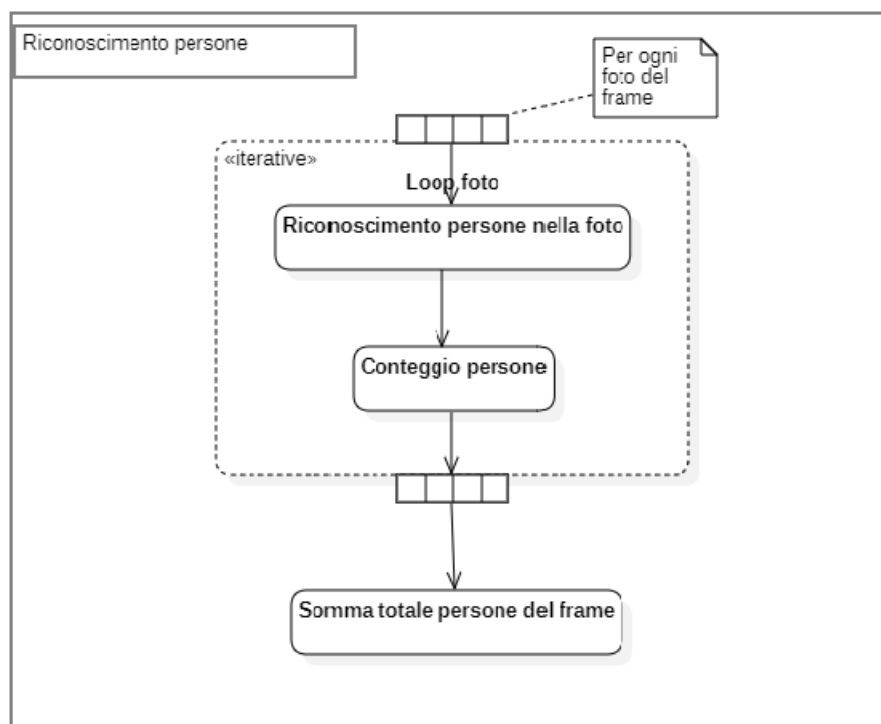


Figura 2.5: Diagramma di sotto attività di riconoscimento persone

#### 2.2.2.4 Diagramma di attività di Kafka

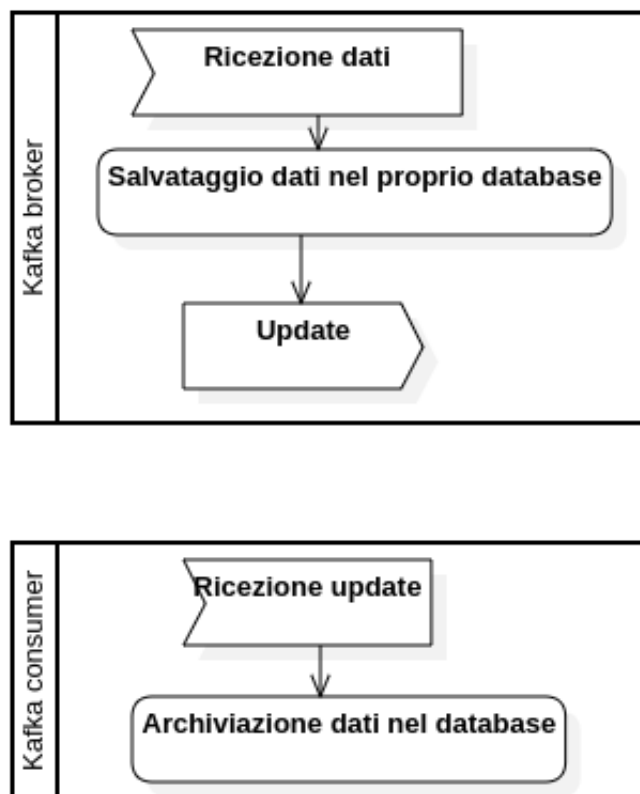


Figura 2.6: Diagramma di attività Kafka

## 2.3 Architettura della predizione

L'architettura del modulo del machine-learning<sub>6</sub> si può semplificare ad un modulo unico con all'interno i metodi necessari per prelevare dati dal database per generare delle predizioni ed archiviarle nel database. Non necessita classi interne in quanto svolge esclusivamente operazioni procedurali.

### 2.3.1 Diagrammi dei package

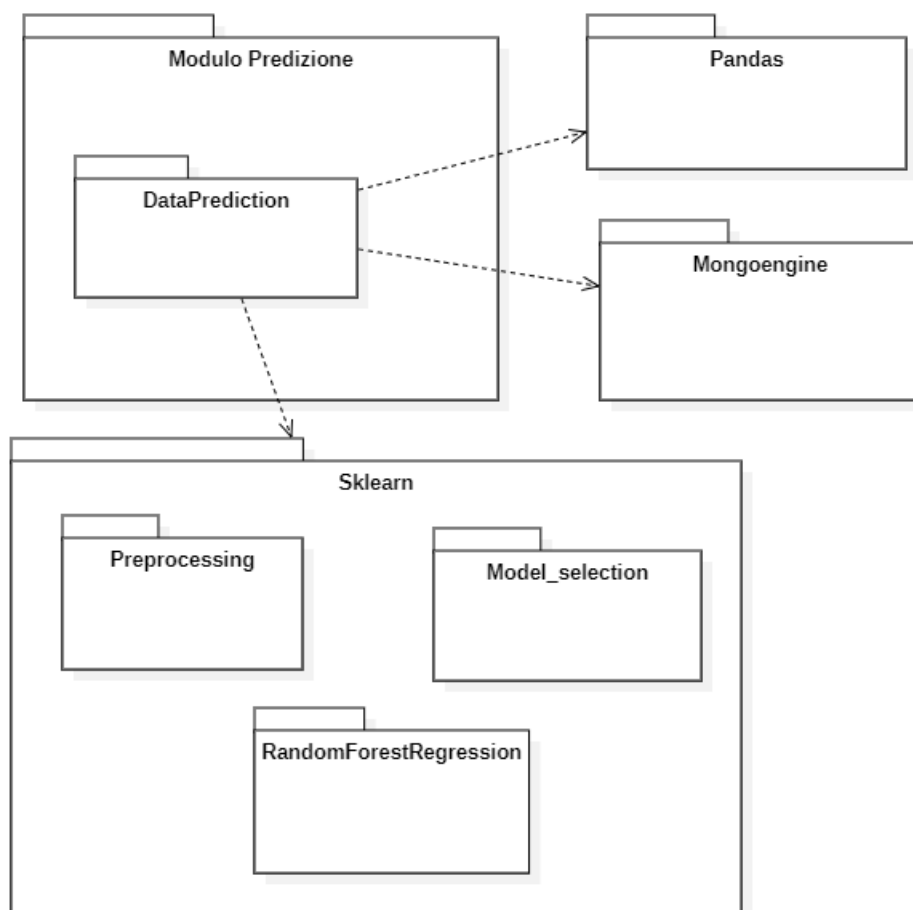


Figura 2.7: Diagramma dei package del modulo predizione

## 2.3.2 Diagrammi di attività

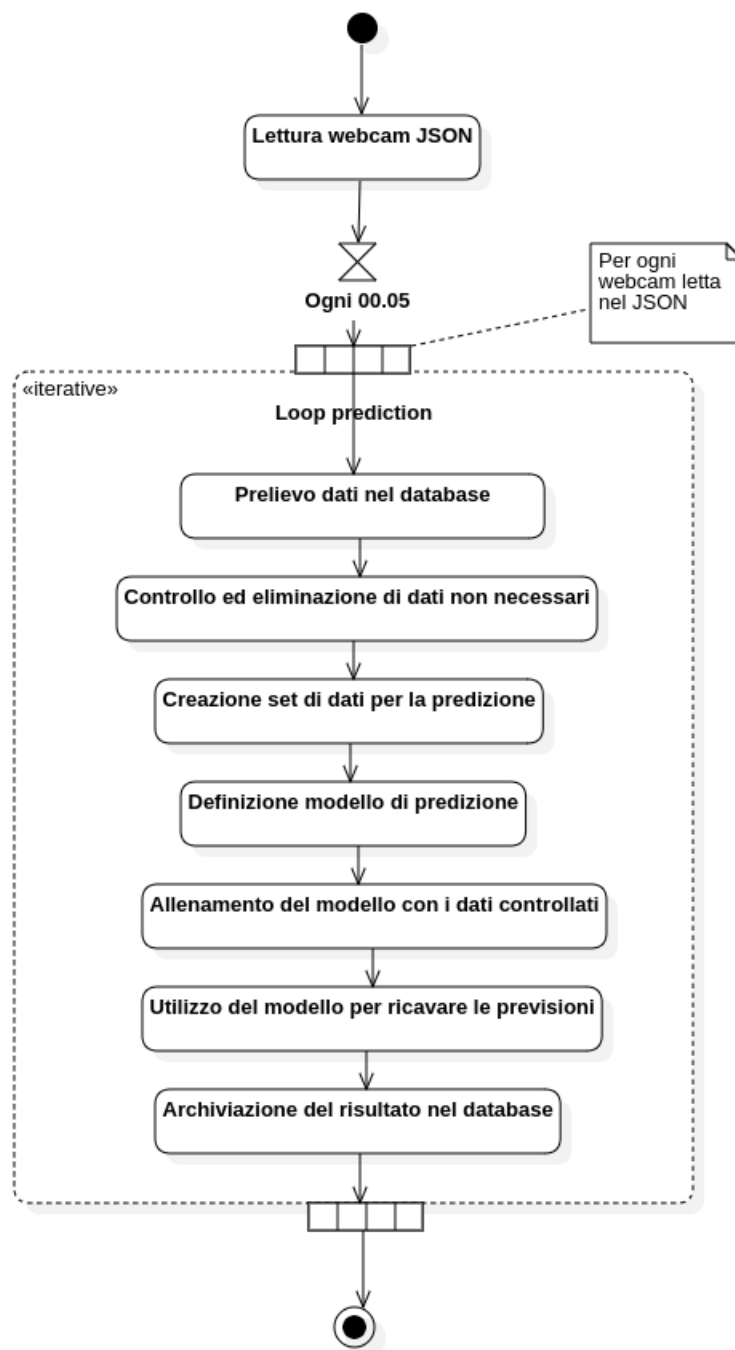


Figura 2.8: Diagramma di attività del modulo predizione

## 2.4 Architettura Web-App

Lo scambio di dati tra  $\text{front-end}_G$  e  $\text{back-end}_G$  avviene attraverso il design pattern $_G$  architetturale  $\text{REST}_G$ . Si ha optato per questo pattern in modo da avere un oggetto unico strutturalmente di scambio tra le due parti e quindi non vincolato dalla struttura presente nel database. Questo permette di aumentare la portabilità dell'applicazione web potendo applicare il  $\text{back-end}_G$ , possibilmente, a diversi  $\text{front-end}_G$ . Le richieste che il  $\text{front-end}_G$  effettua al  $\text{back-end}_G$  sono HTTP Request  $\text{GET}_G$ , quindi sono sempre richieste di visione di informazioni per poi essere utilizzate per aggiornare la parte grafica visibile dal client.

### 2.4.1 Diagrammi dei package del modulo back-end

Il diagramma dei package del modulo  $\text{back-end}_G$  espone le classi all'interno del modulo e l'uso della libreria esterna framework di Spring $_G$  per generare il servizio  $\text{REST}_G$  con l'autoconfigurazione messa a disposizione da Spring Boot $_G$ .

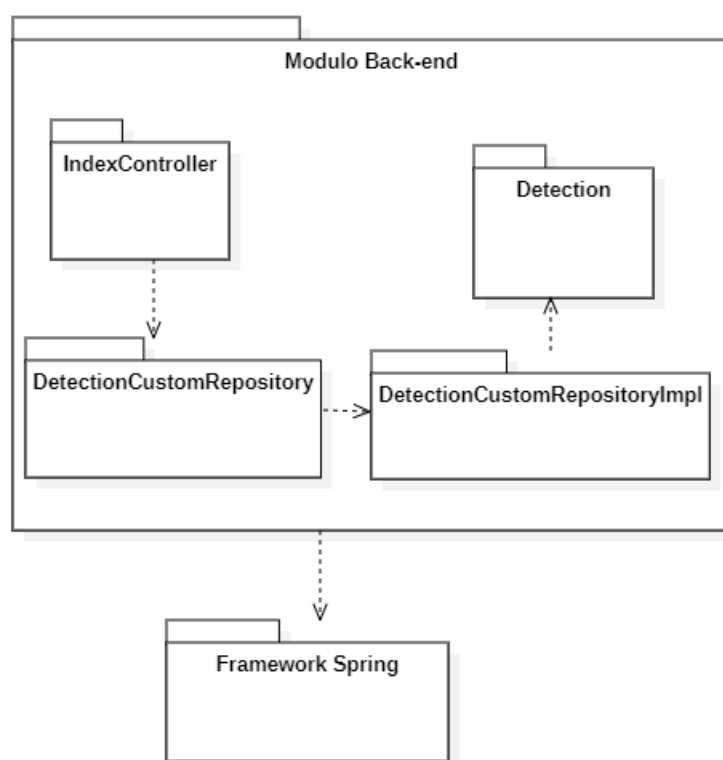


Figura 2.9: Diagramma dei package del modulo back-end



## 2.4.2 Diagramma delle classi del modulo back-end

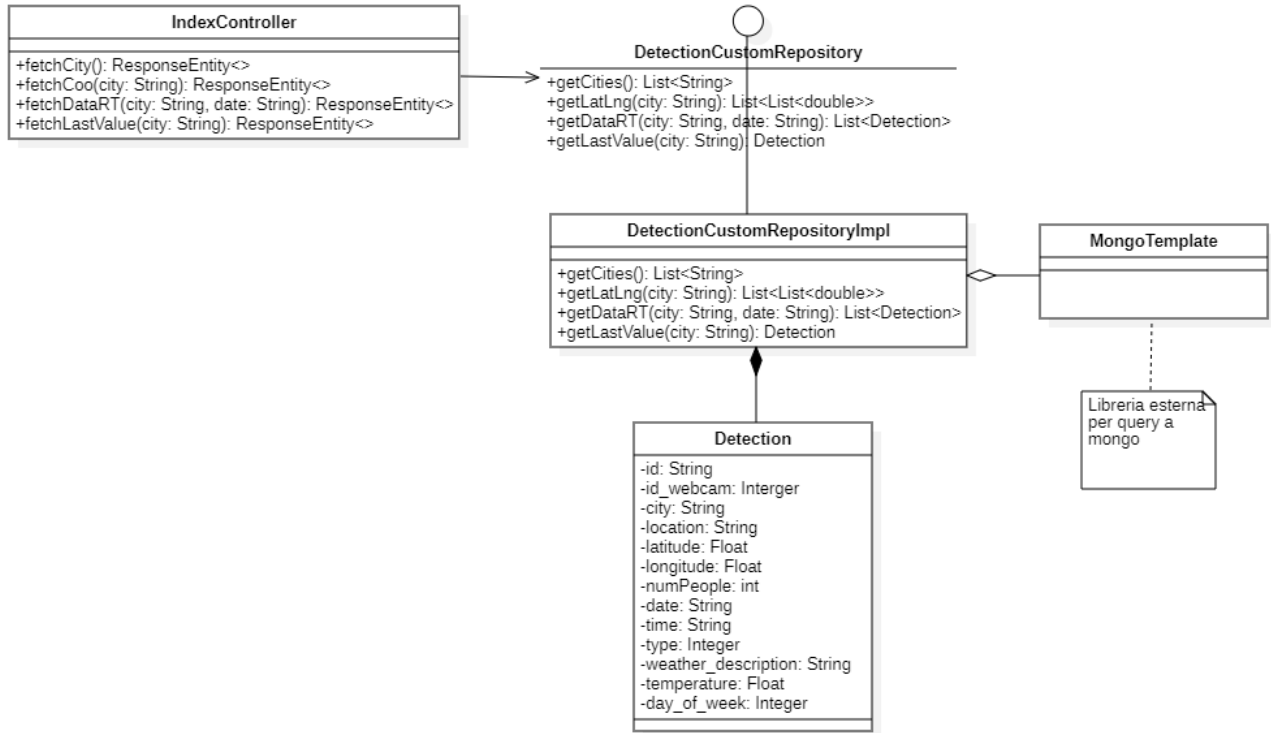


Figura 2.10: Diagramma delle classi del modulo back-end

## 2.4.3 Diagramma di sequenza del modulo back-end

Il diagramma di sequenza illustrato descrive la sequenza di azioni dopo una richiesta di visualizzazione della lista delle città presenti nel database. Questo schema omette tutte le componenti create autonomamente da Spring Boot<sub>c</sub>. Tutte le altre richieste hanno uno schema analogo, ma con una query<sub>c</sub> diversa al database.

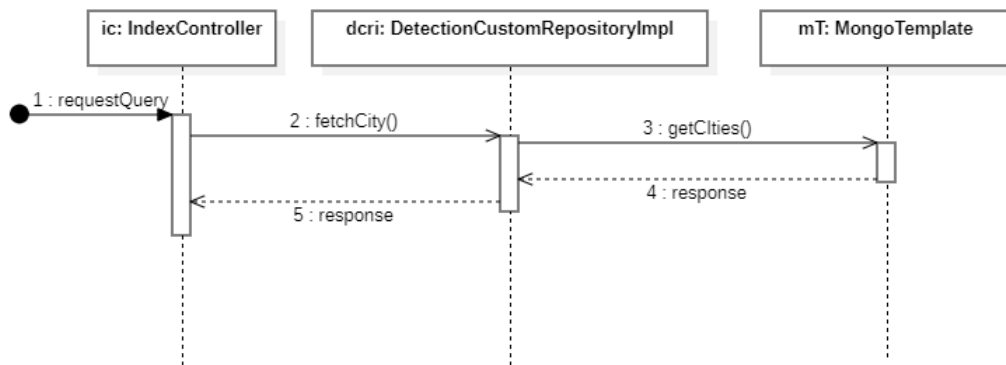


Figura 2.11: Diagramma di sequenza del modulo back-end

#### 2.4.4 Diagrammi dei package del modulo front-end

Il diagramma dei package del modulo front-end<sub>*e*</sub> espone i collegamenti tra i vari file e componenti dell'applicazione sviluppata in Vue<sub>*e*</sub>.

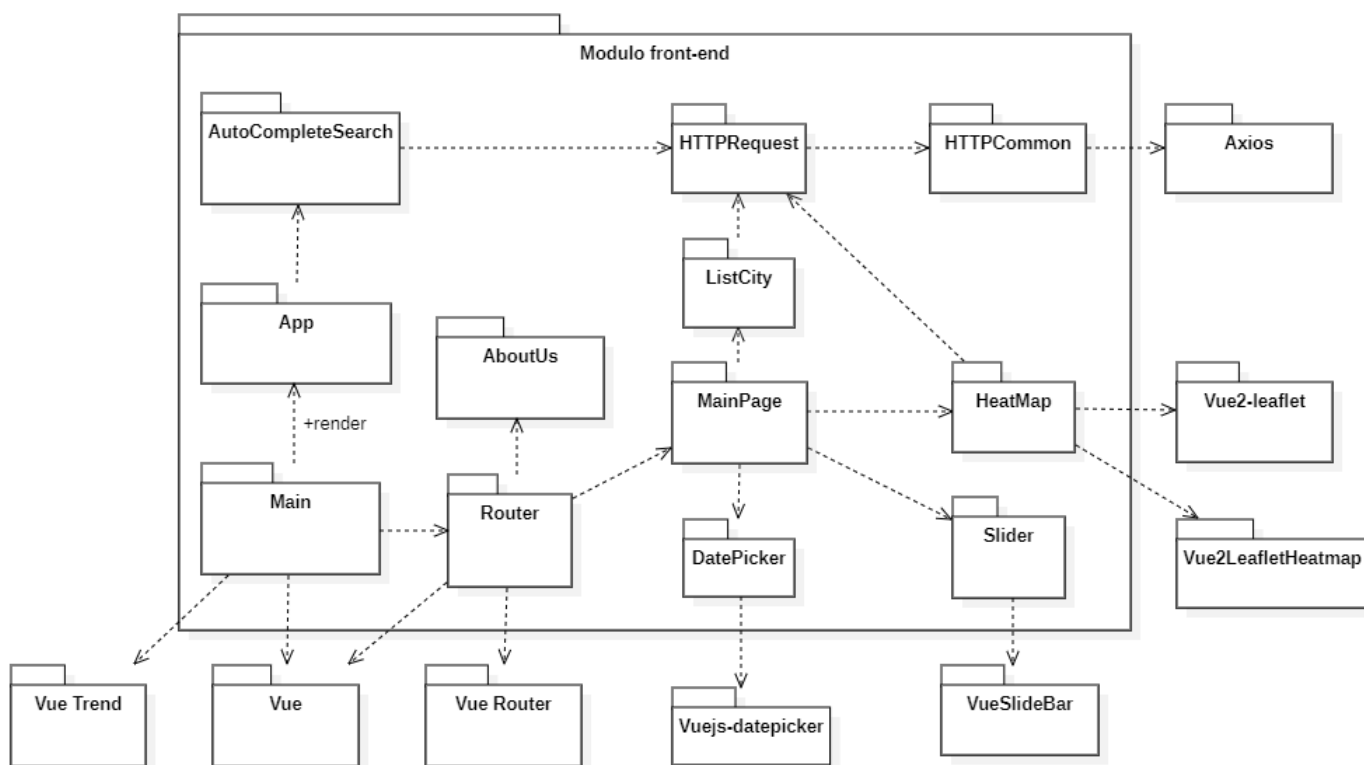


Figura 2.12: Diagramma dei package del modulo front-end

### 2.4.5 Diagrammi delle attività

In questo diagramma di attività viene illustrata la sequenza di operazioni dell'aggiornamento della  $heat-map_c$  a seguito di un cambiamento dei parametri della web-app da parte dell'utente.

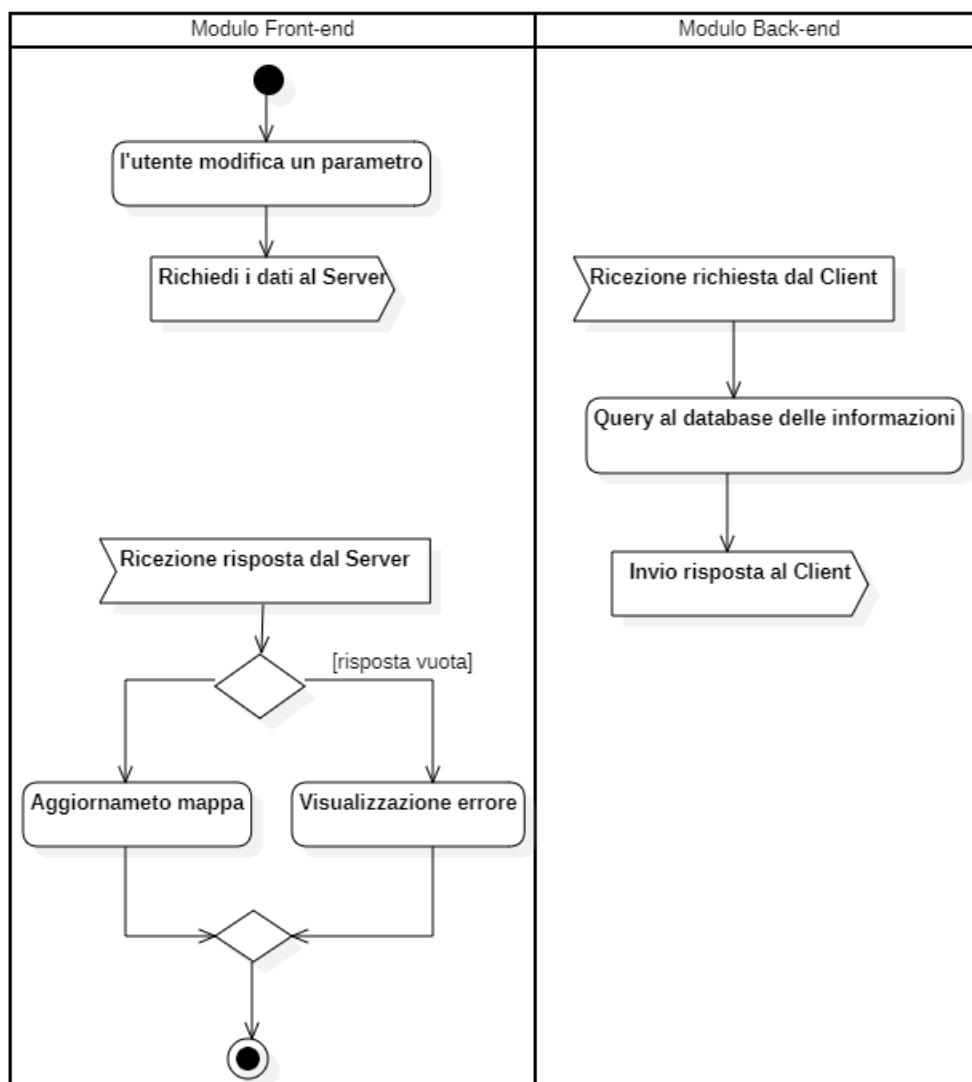


Figura 2.13: Diagramma di attività tra modulo front-end e modulo back-end



## 3 Requisiti soddisfatti

In questo capitolo vengono illustrati attraverso grafici a torta e tabelle i requisiti funzionali che sono stati implementati all'interno della demo sviluppata per la *Revisione di qualifica*. Utilizzando la codifica descritta all'interno delle *Norme 3.0.0*.

### 3.1 Tabella requisiti funzionali

Codice Requisito	Soddisfatto
RSFO1	Soddisfatto
RSFF2	Non soddisfatto
RSFO3	Soddisfatto
RSFO4	Soddisfatto
RSFO4.1	Soddisfatto
RSFO4.2	Soddisfatto
RSFO5	Soddisfatto
RSFD5.1	Non soddisfatto
RSFD6	Non soddisfatto
RSFO7	Soddisfatto
RSFO8	Soddisfatto
RSFO9	Soddisfatto
RSFO10	Soddisfatto
RSFO11	Soddisfatto
RSFF12	Non soddisfatto
RSFD13	Non soddisfatto
RSFD14	Non soddisfatto
RSFF15	Non soddisfatto
RSFF16	Non soddisfatto
RSFO17	Soddisfatto



RSFO18	Soddisfatto
RSFO18.1	Soddisfatto
RSFO19	Soddisfatto
RSFO20	Soddisfatto
RSFO21	Soddisfatto
RSFO22	Soddisfatto
RSFO22.1	Soddisfatto
RSFO22.2	Soddisfatto
RSFF23	Soddisfatto
RSFO24	Soddisfatto
RSFO25	Soddisfatto
RSFO26	Soddisfatto
RSFO27	Soddisfatto
RSFO28	Soddisfatto
RSFD29	Non soddisfatto
RSFO30	Soddisfatto
RSFF31	Non soddisfatto
RSFO32	Soddisfatto
RSFO32.1	Soddisfatto
RSFO32.1.1	Soddisfatto
RSFO32.1.2	Soddisfatto
RSFO32.1.3	Soddisfatto
RSFO32.2	Soddisfatto
RSFD33	Non soddisfatto
RSFD33.1	Soddisfatto
RSFD33.2	Non soddisfatto
RSFD34	Non soddisfatto



RSFD35	Non soddisfatto
RSFD36	Non soddisfatto
RSFD36.1	Non soddisfatto
RSFD36.2	Non soddisfatto
RSFD37	Non soddisfatto
RSFD37.1	Non soddisfatto
RSFD38	Non soddisfatto
RSFD39	Non soddisfatto
RSFD40	Non soddisfatto
RSFD41	Soddisfatto

Tabella 3.1: Requisiti funzionali soddisfatti

## 3.2 Grafici requisiti funzionali

Nel grafico seguente vengono visualizzati tutti i requisiti funzionali soddisfatti.

## Requisiti Funzionali

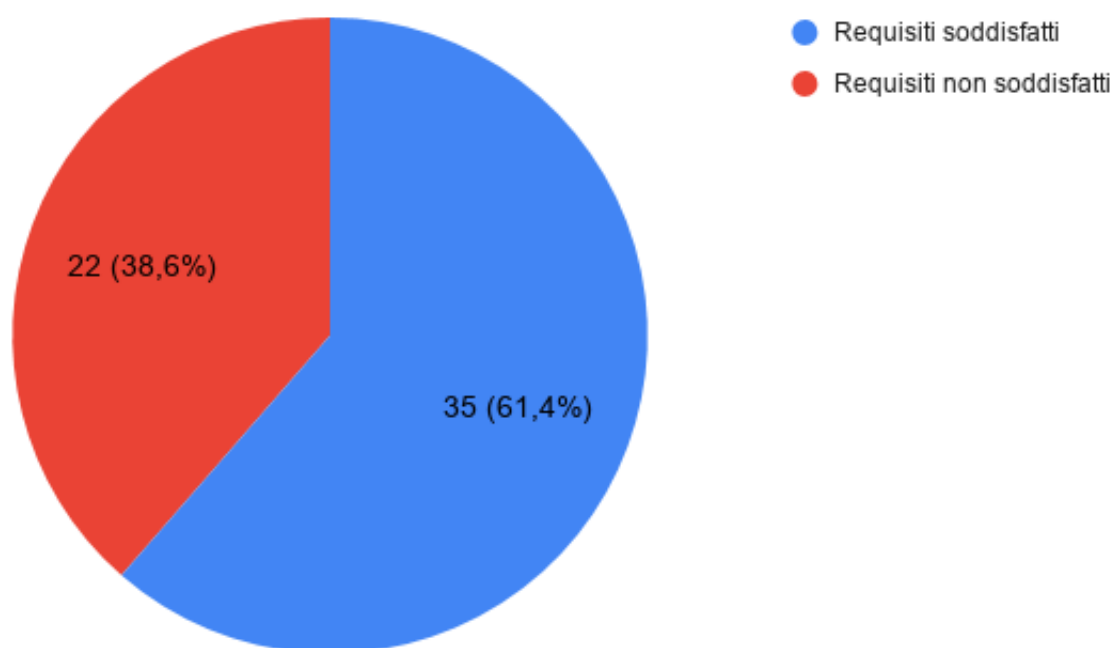


Figura 3.1: Diagramma a torta di tutti i requisiti funzionali soddisfatti

Nel grafico seguente vengono visualizzati tutti i requisiti funzionali obbligatori soddisfatti.





## Requisiti Funzionali Obbligatori

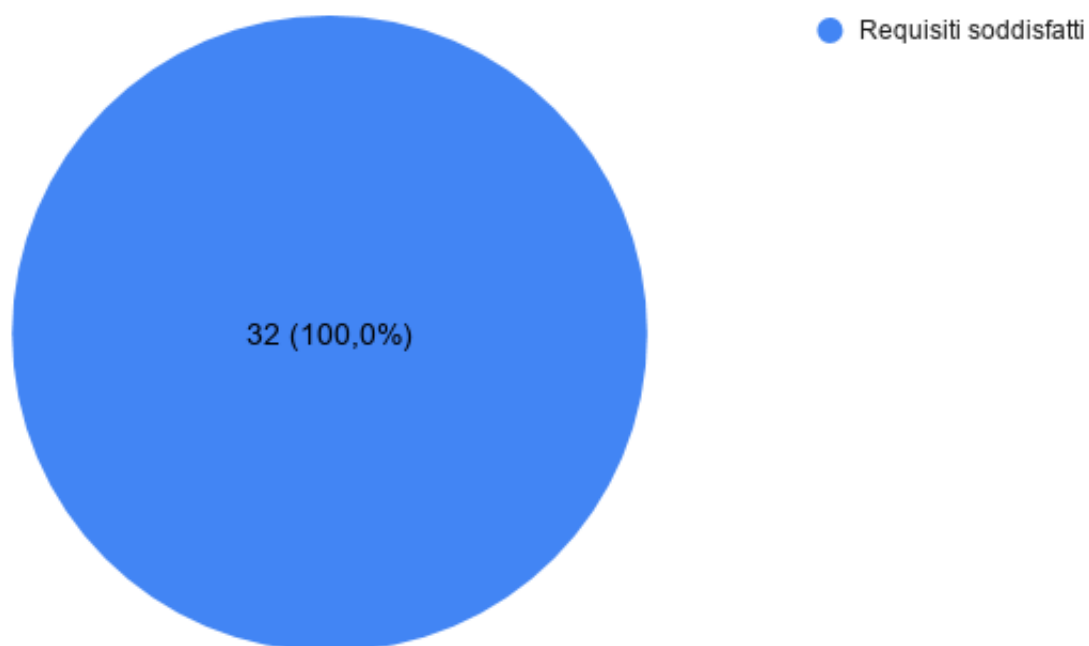


Figura 3.2: Diagramma a torta di tutti i requisiti funzionali obbligatori soddisfatti