



Analisi dei Requisiti

Jawa Druids

Versione	x.x.x
Data approvazione	xx-xx-xxxx
Responsabile	Nome Cognome
Redattori	Nome Cognome
Verificatori	Nome Cognome Nome Cognome
Stato	Stato
Lista distribuzione	Jawa Druids Nome Professori Sync Lab
Uso	Uso del documento

Sommario

L'*Analisi dei Requisiti* individua tutti i requisiti da implementare nel prodotto da sviluppare.



Registro delle modifiche

Modifica	Autore	Ruolo	Data	Versione
<i>Descrizione delle modifiche</i>	Nome Cognome	<i>Analista</i>	23-11-2020	v0.0.3
<i>Stesura cp 2</i>	Andrea Cecchin	<i>Analista</i>	10-12-2020	v0.0.2
<i>Inizio stesura documento cp 1</i>	Andrea Cecchin	<i>Analista</i>	07-12-2020	v0.0.1



Indice

1	Introduzione	3
1.1	Scopo del documento	3
1.2	Scopo del prodotto	3
1.3	Glossario	3
1.4	Riferimenti	3
1.4.1	Riferimenti normativi	3
1.4.2	Riferimenti informativi	4
2	Descrizione generale	5
2.1	Caratteristiche del prodotto	5
2.2	Funzionalità generali	5
2.3	Caratteristiche utente	5
3	Fasi del progetto	6
3.1	FC1: Acquisizione dati	6
3.1.0.1	FC1.1: Acquisizione con java	6
3.1.0.2	FC1.2: Database	6
3.1.0.3	FC1.3: Apache Kafka _G	6
3.2	FC2: Elaborazione Dati	7
3.2.0.1	FC2.1: Esplorazione Dati	7
3.2.0.2	FC2.2: Preprocessing	7
3.2.0.3	FC2.3: Caso predizione	8
3.2.0.4	FC2.4: Valutazioni e validazione	8
3.3	FC3: Visualizzazione dati	8
3.3.0.1	FC3.1: Front-end	8
3.3.0.2	FC3.2: Back-end	8
4	Requisiti	9
4.1	Requisiti funzionali	10
4.2	Requisiti prestazionali	11
4.3	Requisiti di qualità	11
4.4	Requisiti di vincolo	12
4.5	Tracciamento dei requisiti	13
4.5.1	Requisito-fonte	13
4.5.2	Fonte- requisito	15



Introduzione

1.1 Scopo del documento

Lo scopo del documento è quello di formalizzare i contenuti e le qualità che il prodotto sviluppato dovrà raggiungere. I requisiti sono stati individuati attraverso lo studio del capitolato e incontri con l'azienda proponente *Sync Lab*. Il documento inoltre è necessario a:

- descrivere accuratamente tutti i requisiti proposti dal proponente;
- comprendere da parte del committente quali sono le richieste del cliente;
- definire il formato e contenuto di ogni requisito specifico del software.

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.**

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*, 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 v1.0.0*;



- *Verbale Esterno 17-12-2020;*
- *Capitolato d'appalto C3:*
<https://www.math.unipd.it/~tullio/IS-1/2020/Progetto/C3.pdf>

1.4.2 Riferimenti informativi

- *Presentazione del capitolato:*
<https://www.math.unipd.it/~tullio/IS-1/2020/Progetto/C3.pdf>
- *Materiale didattico relativo all'Analisi dei Requisiti del corso di Ingegneria del Software:*
<https://www.math.unipd.it/~tullio/IS-1/2020/Dispense/L07.pdf>
- *IEEE Recommended Practice for Software Requirements Specifications:*
<https://ieeexplore.ieee.org/document/720574>
- *Seminario per approfondimenti tecnici del capitolato C3:*
<https://www.math.unipd.it/~tullio/IS-1/2020/Progetto/ST1.pdf>



Descrizione generale

2.1 Caratteristiche del prodotto

L'idea del capitolato *GDP* - *Gathering Detection Platform* è di creare una piattaforma che riesca a rappresentare mediante visualizzazione grafica zone potenzialmente a rischio di assembramento e cercare di prevenirle. La piattaforma utilizzerà dati prelevati da sensori (come telecamere, dispositivi contapersone, etc.) o sorgenti dati (come flussi di prenotazioni Uber, le tabelle degli orari di autobus/metro/treno, etc.), i quali mediante la loro elaborazione verranno rappresentati tramite una *heat map*.

2.2 Funzionalità generali

Il capitolato *GDP* individua tre principali funzionalità da sviluppare:

- Acquisizione di dati: l'acquisizione avverrà attraverso sistemi di monitoraggio e motori software "contapersone" applicati ad immagini/stream delle videocamere;
- Elaborazione di dati: i dati verranno elaborati per generare valore aggiunto agli stessi e confrontare flussi diversi di informazioni;
- Rappresentazione di dati: attraverso un sito web i dati elaborati verranno visualizzati a video mediante una *heat map*.

2.3 Caratteristiche utente

Il progetto è rivolto principalmente ad utenti di tipo amministrativi, cioè i quali devono visualizzare l'intera mappa di una regione per motivi lavorativi.

Le conoscenze dell'utente per l'utilizzo del software sono:

- Conoscenza base nell'utilizzo del motore di ricerca;
- Padronanza nella lettura della *heat map*.



Fasi del progetto

In questo capitolo verranno illustrate le fasi del progetto identificate dal capitolato d'appalto *GDP-Gathering Detection Platform*. Il capitolo viene diviso nelle tre fasi generali del progetto: acquisizione, elaborazione e visualizzazione dei dati. Secondo lo standard 830-1998 verranno spiegati tutti i punti da sviluppare e nel capitolo successivo i requisiti obbligatori da implementare per la creazione del prodotto richiesto da *Sync Lab*. La descrizione delle fasi è stata inserita in quanto ritenuta necessaria per il chiarimento della necessità dei requisiti individuati.

3.1 FC1: Acquisizione dati

In questa sezione vengono descritte le fasi di acquisizione dei dati.

3.1.0.1 FC1.1: Acquisizione con java

- **Descrizione:** attraverso il linguaggio Java si creerà un programma che preleva informazioni da sorgenti esterne e le invia al server.
- **Linguaggio di programmazione:** Java.
- **Input:** i dati forniti saranno prelevati da siti con live-feed di webcam di Roma e simulatori di spostamenti di persone.
- **Output:** i dati resteranno immutati.
- **Risposta ad errori:** nel caso di mancanza di risposta dai siti con live-feed il programma si bloccherà ed invierà un segnale di errore al server.

3.1.0.2 FC1.2: Database

- **Descrizione:** creazione del database e archiviazione dei dati in esso;
- **Linguaggio:** mySQL.

3.1.0.3 FC1.3: Apache Kafka_G

- **Descrizione:** impostazione di una piattaforma di data streaming_G che consente di gestire e trasferire grandi volumi di dati in tempo reale, abbassando notevolmente i tempi di latenza;
- **Input:** flussi di dati dall'acquisizione con Java;



- **Output:** il flusso di dati rimane immutato.

3.2 FC2: Elaborazione Dati

Ottenuti i dati, essi verranno elaborati attraverso librerie di sci-kit e tensorflow con il linguaggio di programmazione python. Di seguito vengono individuate le fasi da seguire per l'elaborazione dei dati.

3.2.0.1 FC2.1: Esplorazione Dati

- **Descrizione:** si discriminano elementi all'interno del dataset che portano a predizioni errate del modello.
- **Input:** i dati vengono prelevati dal database.
- **Output:** i dati controllati vengono aggiunti in appositi spazi per individuare la loro correttezza.
- **Processo:** si controlla se c'è presenza di valori mancanti, dataset non bilanciati, outliers, livello di rumore dei dati e correlazione dei dati.

3.2.0.2 FC2.2: Preprocessing

- **Descrizione:** preparazione dei dati grezzi e renderli adatti ad un modello di machine learning.
- **Input:** i dati controllati.
- **Output:** dati pronti per l'elaborazione nel modello machine learning.
- **Processo:**
 1. Cleaning: eliminazione o correzione di dati con valori invalidi o corrotti.
 2. Trasformazione dei dati: i dati vengono normalizzati, discretizzati, aggregati, si calcolano nuove variabili etc.
 3. Feature extraction: si ricavano attraverso i dati trasformati valori derivati, i quali sono più informativi e non ridondanti, facilitano le fasi successive di apprendimento e generalizzazione.
 4. Filtraggio dei dati: attraverso appositi filtri eliminare i dati ridondanti e irrilevanti al training del modello.
 5. Train / Test set splitting: si dividono i dati in due gruppi uno per il training e uno per il testing.



3.2.0.3 FC2.3: Caso predizione

- **Descrizione:** in questa fase si effettua una scelta sull'algoritmo più adeguato da utilizzare per il training di dati.
- **Input:** dati controllati nella fase di preprocessing per il training.
- **Output:** modello di Machine Learning_G allenato sui dati di input.
- **Tipi di algoritmi:** si dividono per classificazione e regressione.

3.2.0.4 FC2.4: Valutazioni e validazione

- **Descrizione:** attraverso varie metriche si valuta quanto valido è il modello nella predizione dei casi.
- **Input:** risposta del modello Machine Learning_G dai dati di test, dati ricavati dalle sorgenti esterne effettivi.
- **Output:** dati che superano la validazione.

3.3 FC3: Visualizzazione dati

In questa sezione verranno illustrate le fasi di sviluppo della parte visiva della web-app.

3.3.0.1 FC3.1: Front-end

- **Descrizione:** sviluppo di una pagina web semplice ed intuitiva.
- **Strumenti:** si utilizzerà Angular e Spring, due librerie per framework di javascript.
- **Vincolo:** la web app dovrà essere costruita sia desktop che mobile friendly.
- **Struttura:** la pagina sarà principalmente rivolta alla visione della mappa per la visualizzazione di aree a rischio assembramenti.

3.3.0.2 FC3.2: Back-end

- **Descrizione:** sviluppo della parte di comunicazione di informazioni tra server/database e front-end.
- **Strumenti:** si utilizzerà Java.



Requisiti

In questa sezione vengono illustrati attraverso una tabella tutti i requisiti individuati dal proponente e il gruppo Jawa Druids. Ogni requisito viene individuato da un codice identificativo, una sua descrizione, la tipologia di requisito e codice della fase di riferimento, la spiegazione di ogni parte è descritta nel documento Norme del Progetto.



4.1 Requisiti funzionali

Codice RS	Descrizione	Tipo di requisito	Fonte
RSFO1	Realizzazione di motori software ‘contapersone’	Obbligatorio	Capitolato Verbale esterno 17-12-2020
RSFF2	Realizzazione di simulatori di altre sorgenti dati sia dei dati storici/in monitoraggio che dati previsionali	Facoltativo	Capitolato
RSFO3	Il sistema deve visualizzare un messaggio d’errore se il flusso di dati esterno viene a mancare	Obbligatorio	Interno FC1.1
RSFO4.1	Archiviazione di tutti i dati acquisiti nel database	Obbligatorio	Capitolato
RSFO4.2	Archiviazione di tutti i dati elaborati nel database	Obbligatorio	Capitolato
RSFO5	Elaborazione in tempo reale dei dati acquisiti da flussi esterni	Obbligatorio	Capitolato
RSFO5.1	Identificazione di eventi che portano alla variazione del flusso di utenti	Obbligatorio	Capitolato
RSFD6	Previsione dell’insorgenza futura di variazioni significative di flussi di persone	Desiderabile	Capitolato
RSFO7	Visualizzazione dei dati elaborati attraverso heat map	Obbligatorio	Capitolato
RSFO8	Apache Kafka _c deve poter comunicare con il database, l’applicazione web e il modello di machine learning	Obbligatorio	Interno



4.2 Requisiti prestazionali

Codice RS	Descrizione	Tipo di requisito	Fonte
RSPO1	Capacità di acquisizione continuativa nel tempo dei dati da flussi esterni	Obbligatorio	Capitolato
RSPO2	Modalità a bassa latenza dell'acquisizione di informazioni	Obbligatorio	Capitolato

4.3 Requisiti di qualità

Codice RS	Descrizione	Tipo di requisito	Fonte
RSQO1	La progettazione e la codifica dei requisiti devono rispettare le norme e le metriche definite nel documento <i>Norme di Progetto v1.0.0</i>	Obbligatorio	Interno
RSQF2	Il codice sorgente del software deve essere disponibile in una repository _G pubblica su Github _G	Facoltativo	Interno
RSQF3	Deve essere sviluppato e fornito un documento con lo schema della base di dati relazionale	Facoltativo	Interno
RSQF4	Deve essere realizzato un documento contenente tutti gli errori risolti durante la realizzazione del software	Facoltativo	Interno
RSQO5	Test che dimostrino il corretto funzionamento dei servizi e delle funzionalità previste	Obbligatorio	Capitolato



4.4 Requisiti di vincolo

Codice RS	Descrizione	Tipo di requisito	Fonte
RSVO1	I dati acquisiti da telecamere in tempo reale devono avere data di riferimento associato	Obbligatorio	Interni
RSVO1.1	I dati acquisiti da telecamere in tempo reale devono avere un orario di riferimento associato	Obbligatorio	Interni
RSVO1.2	I dati acquisiti da telecamere in tempo reale devono avere un luogo di riferimento associato	Obbligatorio	Interni
RSVO2	Il front-end del prodotto viene sviluppato utilizzando tecnologie web	Obbligatorio	Capitolato FC3.1
RSVF2.1	Utilizzo di leaflet.js per la creazione di heat-map	Facoltativo	Capitolato FC3.1
RSVO2.2	Utilizzo di angular.js per la creazione della wep-app	Obbligatorio	Capitolato FC3.1
RSVO3	Il sistema deve far uso dell'ecosistema Apache Kafka _g	Obbligatorio	Capitolato FC1.3



4.5 Tracciamento dei requisiti

4.5.1 Requisito - fonte

Codice RS	Fonte
RSFO1	Capitolato Verbale esterno 17-12-2020
RSFF2	Capitolato
RSFO3	Interno FC1.1
RSFO4.1	Capitolato
RSFO4.2	Capitolato
RSFO5	Capitolato
RSFO5.1	Capitolato
RSFD6	Capitolato
RSFO7	Capitolato
RSFO8	Interno
RSPO1	Capitolato
RSPO2	Capitolato
RSQO1	Interno
RSQF2	Interno
RSQF3	Interno
RSQF4	Interno
RSQO5	Capitolato
RSVO1	Interno
RSVO1.1	Interno
RSVO1.2	Interno
RSVO2	Capitolato FC3.1
RSVF2.1	Capitolato FC3.1



Codice RS	Fonte
RSVO2.2	Capitolato FC3.1
RSVO3	Capitolato FC1.3



4.5.2 Fonte - requisito

Fonte	Codice RS
Capitolato	RSFO1 RSFF2 RSFO4.1 RSFO4.2 RSFO5 RSFO5.1 RSFD6 RSFO7 RSPO1 RSPO2 RSQO5 RSVO2 RSVF2.1 RSVO2.2 RSVO3
Interno	RSFO3 RSFO8 RSQO1 RSQF2 RSQF3 RSQF4 RSVO1 RSVO1.1 RSVO1.2
Verbale esterno 17-12-2020	RSFO1
FC1.1	RSFO3
FC1.3	RSVO3
FC3.1	RSVO2 RSVF2.1 RSVO2.2