

# Documento dei requisiti

WebMiningProject

Versione: 11072023\_1.0

Autori: Van Sang Ho, Marco Toffoletto

## Sommario

---

<b>1. Requisiti</b>	<b>2</b>
1.1 Introduzione	2
1.2 Requisiti funzionali	2
1.3 Requisiti non funzionali	2
1.4 Requisiti di sistema	3
1.5 Requisiti utente	3
<b>2. Casi d'uso</b>	<b>4</b>
2.1 Caso d'uso UC1: download	4
2.2 Caso d'uso UC2: estrazione	4
2.3 Caso d'uso UC2: download ed estrazione	4
2.4 Caso d'uso UC4: help	5
2.5 Diagramma dei casi d'uso	6
<b>3. Documentazione ed allegati</b>	<b>7</b>
3.1 Diagramma delle classi	7
3.2 Domain Model	7
3.3 Diagramma di sequenza	7
3.4 Diagramma delle attività	7
3.5 Diagramma dei casi d'uso	7
3.6 Manuale d'uso	7
3.7 Software utilizzati	7
<b>4. Appendice</b>	<b>7</b>
4.1 Bibliografia	7
4.2 Sitografia	7

# 1. Requisiti

## 1.1 Introduzione

Il progetto mira a sviluppare un sistema software in Java che sia in grado di scaricare articoli da testate giornalistiche online da diverse fonti, estrarre i termini più "importanti" per ordine di peso all'interno degli articoli e visualizzarli. Il sistema supporta, inoltre, nuove sorgenti e modalità di memorizzazione ed accesso ai dati.

## 1.2 Requisiti funzionali

- **Download degli articoli**  
Il sistema è in grado di scaricare gli articoli dalle testate giornalistiche online fornite come sorgenti, le quali possono essere di due tipi: file CSV o API.
- **Memorizzazione degli articoli**  
Dopo il download, il sistema salva tutti gli articoli scaricati in un singolo file in formato JSON.
- **Supporto a nuove modalità di memorizzazione e accesso agli articoli**  
Il sistema è progettato in modo che sia possibile estenderlo per supportare nuove modalità di memorizzazione ed accesso agli articoli, oltre a quella predefinita. Ciò permette di adattarsi a diverse esigenze e requisiti futuri.
- **Estrazione dei termini più importanti**  
Il sistema è in grado di determinare i 50 termini più importanti per ordine di peso tra quelli estratti dai file degli articoli. L'importanza di un termine è direttamente proporzionale al suo conteggio di frequenza, che rappresenta il numero di volte in cui il termine compare all'interno degli articoli.
- **Memorizzazione dei termini più importanti**  
I 50 termini più importanti vengono memorizzati in un file di testo (.txt). Ogni riga del file viene visualizzata nel formato "termine peso".
- **Supporto per azioni sequenziali**  
L'utente può scegliere se eseguire solo il download, solo l'estrazione dei termini dai file degli articoli o fare entrambe le azioni in sequenza.

## 1.3 Requisiti non funzionali

- **Utilizzo del linguaggio Java**  
Il sistema è sviluppato utilizzando il linguaggio di programmazione Java.
- **Unit testing con JUnit**  
Il codice del sistema è stato sottoposto a unit testing utilizzando il framework JUnit.
- **Commenti e indentazione del codice**  
Il codice sorgente del sistema è stato opportunamente commentato e indentato per garantire la leggibilità e manutenibilità del codice.

## 1.4 Requisiti di sistema

- **Dipendenze software**

Dev'essere installato il pacchetto Java SE 8 oppure Java JDK 8 sul dispositivo utilizzatore.

- **Memoria**

Il sistema richiede una quantità di memoria minima per gestire il caricamento e l'elaborazione dei file degli articoli e dei termini estratti.

- **Spazio di archiviazione**

Il sistema richiede uno spazio di archiviazione minimo per memorizzare i file degli articoli scaricati, i file dei termini estratti e altri dati correlati.

- **Risorse di rete**

Il sistema richiede una connessione ad Internet attiva per il download degli articoli tramite l'API The Guardian e per eventuali altre fonti di articoli online

## 1.5 Requisiti utente

- **Risultati delle elaborazioni**

Il sistema mostra i risultati delle sue elaborazioni visualizzando un messaggio per indicare il completamento dell'operazione richiesta dall'utente, oppure vengono notificate eventuali eccezioni e le relative cause.

## 2. Casi d'uso

### 2.1 Caso d'uso UC1: download

**Obbiettivo:** Scaricare gli articoli dalle sorgenti fornite e salvarli in una cartella dedicata.

**Attore:** Utente.

**Precondizioni:** L'utente ha letto il manuale.

**Post-condizioni:**

1. Download terminato con successo.
2. Salvataggio di tutti gli articoli in un file in formato json.

**Basic flow:**

1. L'utente scrive da riga di comando '-d' seguito dall'argomento di suo interesse all'interno di due doppie virgolette e preme invio.
2. Gli articoli vengono scaricati con successo ed immagazzinati nella cartella dedicata.

**Frequenza di utilizzo:** Può essere anche continua.

### 2.2 Caso d'uso UC2: estrazione

**Obbiettivo:** Estrarre i termini dal file contenente gli articoli serializzati.

**Attore:** Utente.

**Precondizioni:**

1. L'utente ha letto il manuale.
2. La cartella dedicata deve contenere il file degli articoli scaricati in precedenza.

**Post-condizioni:**

1. Estrazione terminata con successo.
2. Salvataggio dei termini più importanti per ordine di peso in un file di testo.

**Basic flow:**

1. L'utente scrive da riga di comando '-e' e preme invio.
2. I termini vengono estratti con successo ed immagazzinati nella cartella dedicata

**Frequenza di utilizzo:** Può essere anche continua.

### 2.3 Caso d'uso UC3: download ed estrazione

**Obbiettivo:** Scaricare gli articoli ed estrarne i termini più importanti per ordine di peso.

**Attore:** Utente.

**Precondizioni:**

1. L'utente ha letto il manuale.

**Post-condizioni:**

1. Download terminato con successo.
2. Salvataggio di tutti gli articoli in un file in formato json.
3. Estrazione terminata con successo.
4. Salvataggio dei termini più importanti per ordine di peso in un file di testo.

**Basic flow:**

1. L'utente scrive da riga di comando '-d' seguito dall'argomento di suo interesse all'interno di due doppie virgolette, a sua volta seguito dal comando '-e' e preme invio.
2. Gli articoli vengono scaricati con successo ed immagazzinati nella cartella dedicata.
3. I termini vengono estratti con successo ed immagazzinati nella cartella dedicata

**Frequenza di utilizzo:** Può essere anche continua.

## **2.4 Caso d'uso UC4: help**

**Obbiettivo:** Fornire supporto all'utente che non ricorda i comandi.

**Attore:** Utente.

**Precondizioni:** L'utente ha letto il manuale.

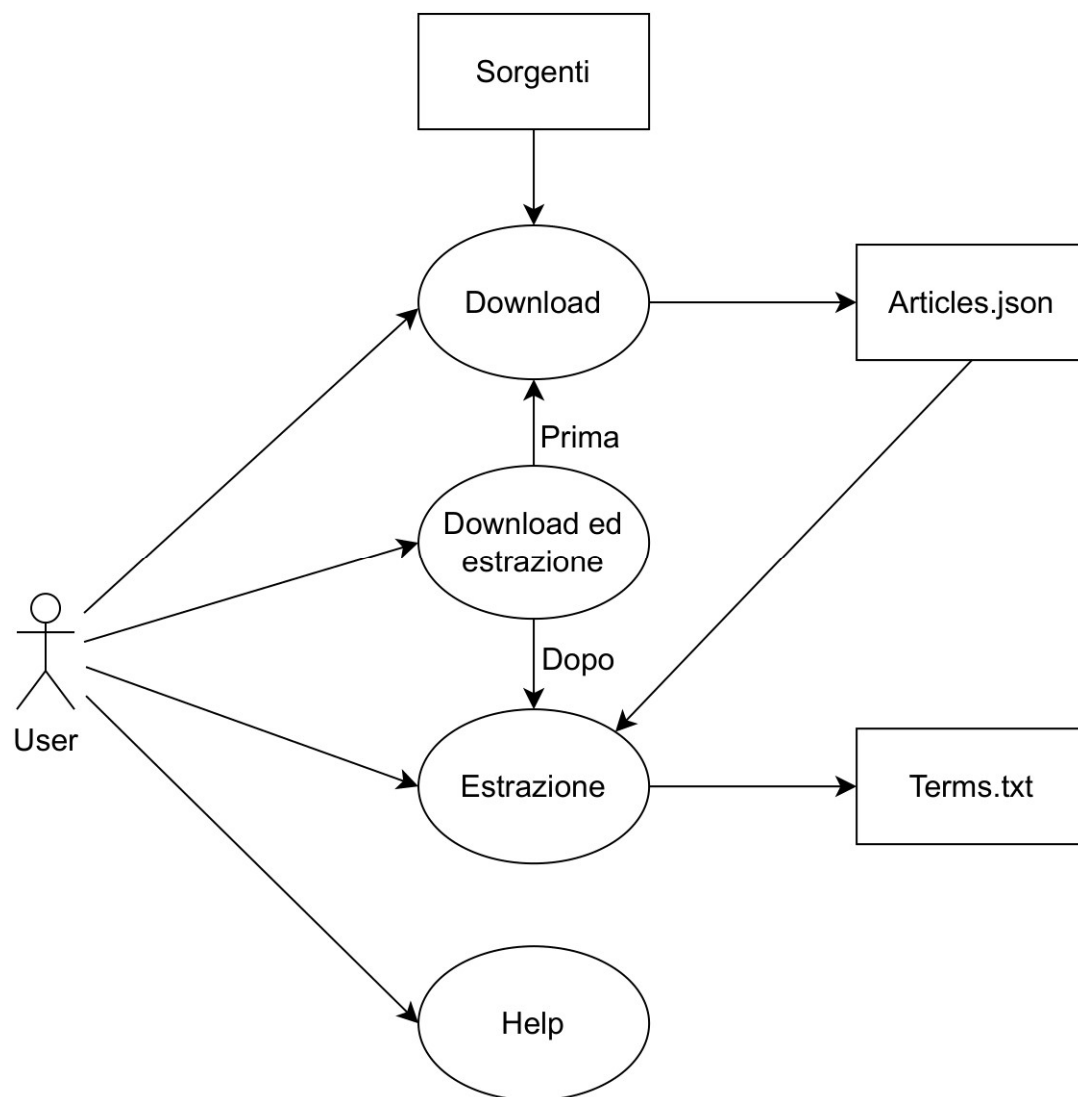
**Post-condizioni:** Apertura del menù di supporto avvenuta con successo.

**Basic flow:**

1. L'utente scrive da riga di comando '-h' e preme invio.
2. L'apertura del menù di supporto avviene con successo.

**Frequenza di utilizzo:** Può essere anche continua.

## 2.5 Diagramma dei casi d'uso



## **3. Documentazione ed allegati**

### **3.1 Diagramma dei casi d'uso**

Il documento "DiagrammaCasiUso.pdf" in allegato (e riportato sopra) descrive le relazioni fra le funzionalità specifiche che compongono il caso d'uso preso in esame.

### **3.2 Domain model**

Il documento "DomainModel.pdf" in allegato descrive schematicamente le associazioni e cardinalità tra le diverse classi.

### **3.3 Diagramma delle classi**

Il documento "DiagrammaDelleClassi.pdf" in allegato descrive nel dettaglio le strutture interne (metodi, attributi) delle classi e le associazioni tra di esse.

### **3.4 Diagramma di sequenza**

Il documento "DiagrammaDiSequenza.pdf" in allegato descrive l'ordine temporale dell'esecuzione dei vari metodi e gli scambi di dati tra le diverse classi.

### **3.5 Diagramma delle attività**

Il documento "DiagrammaDelleAttivita.pdf" in allegato descrive il flusso delle attività che compongono il programma.

### **3.6 Manuale d'uso**

Il documento "Manuale.pdf" in allegato fornisce una panoramica concisa e ad alto livello del progetto, soffermandosi su come installare ed eseguire il software. Vi sono inoltre indicazioni su quali funzioni sono state riutilizzate da librerie esistenti, specificandone la versione.

### **3.7 Software utilizzati**

PlantUML - <https://plantuml.com/>

Draw.io - <https://app.diagrams.net/>

IntelliJ - <https://www.jetbrains.com/idea/>

## 4. Appendice

### 4.1 Bibliografia

Slide delle lezioni presenti nel corso moodle di Elementi di Ingegneria del Software dell'Università di Padova

### 4.2 Sitografia

Corso di Elementi di Ingegneria del Software - <https://stem.elearning.unipd.it/course/view.php?id=4742>

Documentazione Java - <https://docs.oracle.com/en/java/>

Documentazione The Guardian - <https://open-platform.theguardian.com/documentation/>

Content API The Guardian - <https://github.com/matarrese/content-api-the-guardian>