



Chimica, Materiali e Biotecnologie  
Grafica e Comunicazione  
Trasporti e Logistica



Istituto Tecnico Tecnologico Statale "Alessandro Volta"  
Scuol@2.0

Via Assisana, 40/E - Loc. Piscille - 06135 Perugia  
Centralino 075.31045 fax 075.31046 C.F. 80005450541  
www.avolta.pg.it  
PGTF010005@istruzione.it PGTF010005@pec.istruzione.it



Meccanica, Meccatronica ed Energia  
Elettronica ed Elettrotecnica  
Informatica e Telecomunicazioni

**Corso di**  
**Tecnologie e progettazione di sistemi informatici e di**  
**telecomunicazioni**

**Classi 5AInf, 5BInf, 5CInf**  
**a.s. 2020/2021**

**Relazione progetto**  
**“Gestione Dati”**

**Studente: Francesco Alunni**

**Docente/i: Monica Ciuchetti**

# **Documento di Specifica dei Requisiti (Software Requirements Specification)**

## **1. Introduzione e formulazione del problema**

L'applicazione risolve il problema della gestione di file di tipo XML e JSON, permettendo la lettura di un file formattato in tale modo, l'elaborazione con votazione della canzone e consecutiva scrittura della risposta su file. Il prodotto verrà creato in Java e interfacerà l'utente via testuale.

## **2. Descrizione dell'architettura dell'applicazione (componenti hardware e software)**

### **2.1 Attori**

- Utente: Entità che rappresenta l'utente che sceglie il tipo di file da gestire
- ThreadLettore: Thread che legge il file.
- ThreadElaboratore: Thread che elabora i dati ottenuti dalla lettura ed effettua la votazione della canzone.
- ThreadScrittore: Thread che scrive su file.
- GestoreFile: Entità che gestisce il file (lettura, scrittura, parsing)

### **2.2 Casi d'uso**

- Utente:
  - Inserisce estensione file: Inserisce in input l'estensione del file da gestire.
  - Vota canzone: Effettua la votazione della sua canzone preferita.
- GestoreFile:
  - Legge file: Legge il file che gestisce
    - Parse da XML: fa il parsing del contenuto da XML
    - Parse da JSON: fa il parsing del contenuto da JSON
  - Scrivi su file: Scrivi su file il contenuto passato.
    - Scrivi file XML: Scrive il file formattato in XML

-ThreadElaboratore:

-Elabora dati: Elabora i dati ottenuti dalla lettura per fare la votazione.

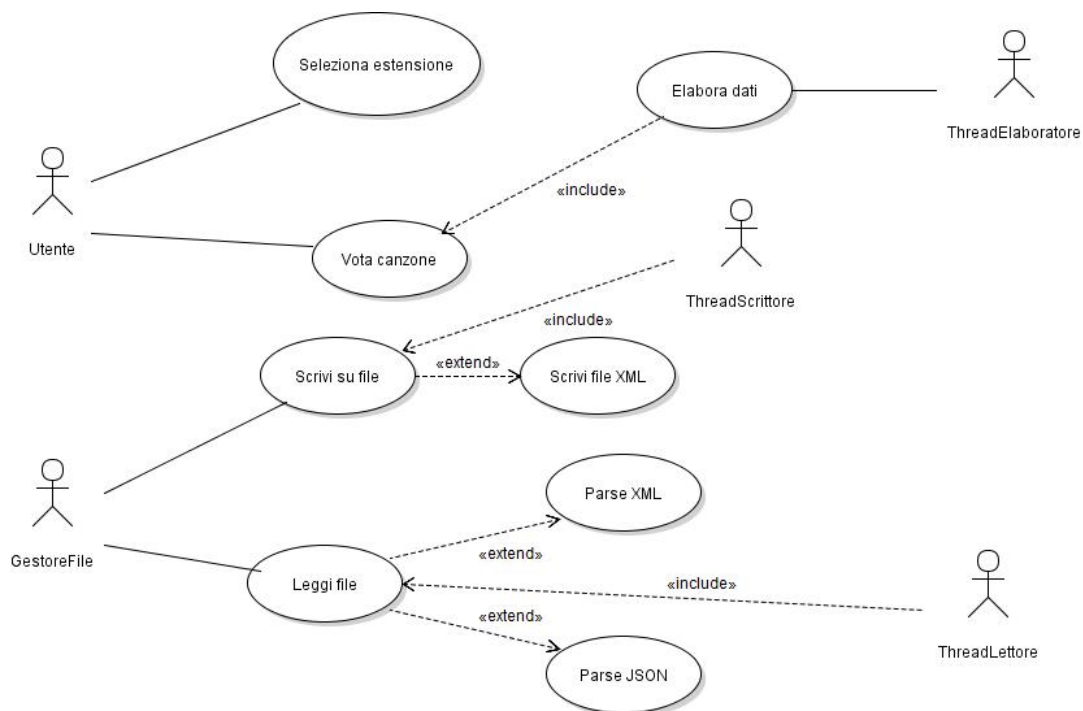
-ThreadLettore:

-Chiama il caso d'uso "Leggi file", ma sincronizzando l'accesso.

-ThreadScrittore:

-Chiama il caso d'uso "Scrivi su file", ma sincronizzando l'accesso.

## 2.3 Diagramma dei casi d'uso



## 2.4 Vincoli e tecnologie usate

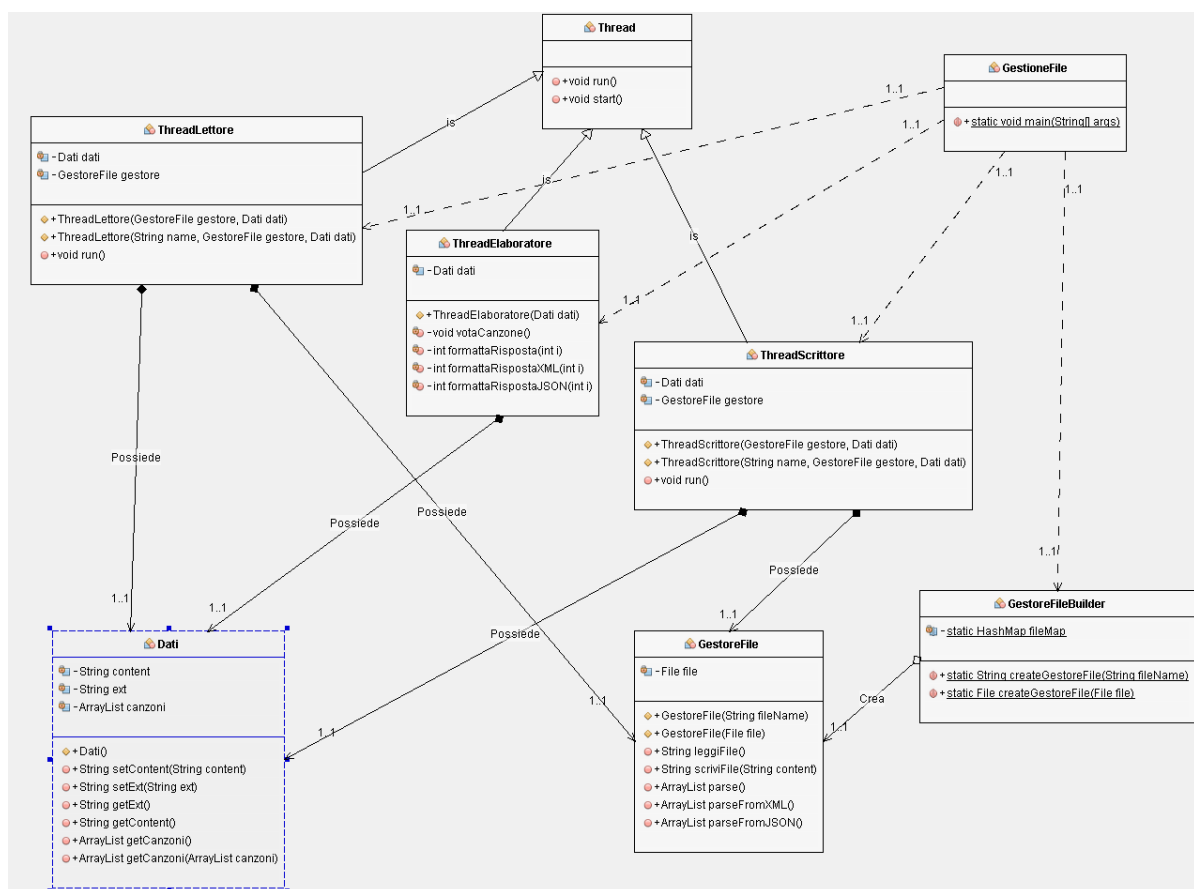
I vincoli e tecnologie usate sono: l'uso dell'IDE Netbeans 8.2 per lo sviluppo dell'applicazione e, di conseguenza, il JDK 1.8 e la JVM per permetterne lo sviluppo e l'esecuzione (sia di Netbeans che dell'applicazione).

## 2.5 Prototipo dell'interfaccia

L'utente deve inserire in maniera testuale prima l'estensione dei file da gestire e poi, quando viene richiesto, la votazione della canzone preferita. Se l'utente inserisce un'estensione non consentita o un indice di canzone non esistente, verrà ripetuto l'inserimento visualizzando un messaggio d'errore.

## Documento di progettazione ed implementazione dell'applicazione

### 1. Diagramma delle classi

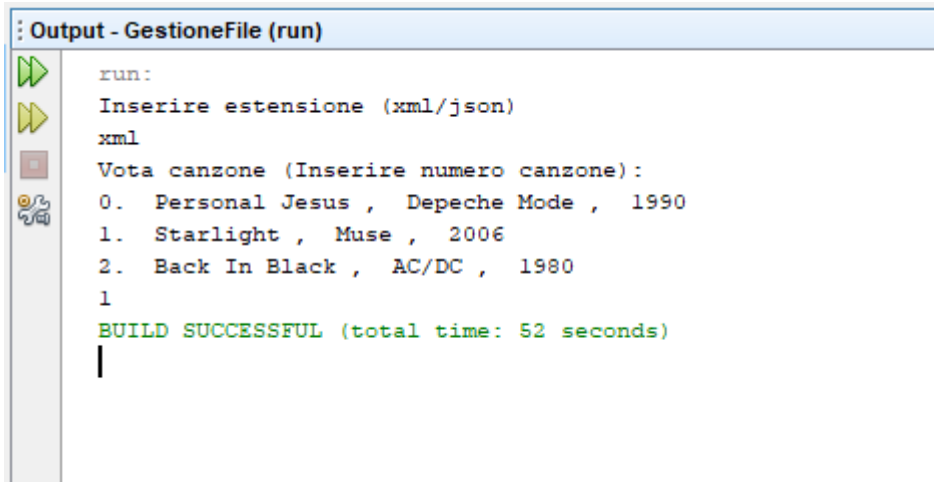


### 5. Commento del codice

Codice commentato con implementazione del Javadoc.

## 7. Test dell'applicazione

Programma in esecuzione:




```
Output - GestioneFile (run)

run:
Inserire estensione (xml/json)
xml
Vota canzone (Inserire numero canzone):
0. Personal Jesus , Depeche Mode , 1990
1. Starlight , Muse , 2006
2. Back In Black , AC/DC , 1980
1
BUILD SUCCESSFUL (total time: 52 seconds)
```

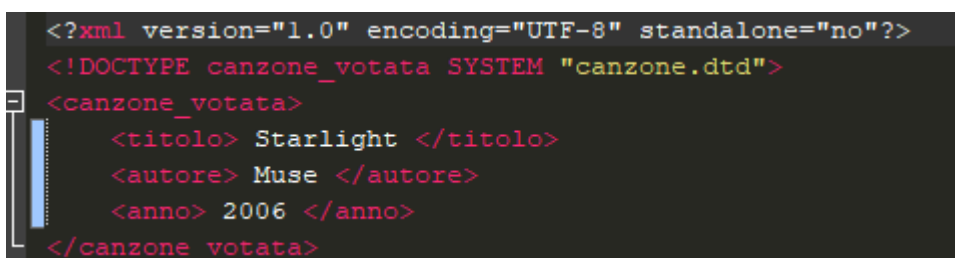
Uso dell'XML:

File con tutte le canzoni iniziali:



```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<playlist>
  <nome> Brani preferiti </nome>
  <canzone id="1">
    <titolo> Personal Jesus </titolo>
    <autore> Depeche Mode </autore>
    <anno> 1990 </anno>
  </canzone>
  <canzone id="2">
    <titolo> Starlight </titolo>
    <autore> Muse </autore>
    <anno> 2006 </anno>
  </canzone>
  <canzone id="3">
    <titolo> Back In Black </titolo>
    <autore> AC/DC </autore>
    <anno> 1980 </anno>
  </canzone>
</playlist>
```

Risultato dopo la votazione:



```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!DOCTYPE canzone_votata SYSTEM "canzone.dtd">
<canzone_votata>
  <titolo> Starlight </titolo>
  <autore> Muse </autore>
  <anno> 2006 </anno>
</canzone_votata>
```

## Uso del JSON

File con tutte le canzoni iniziali:

```
{
  "playlist" : [
    {
      "titolo" : "Personal Jesus",
      "autore" : "Depeche Mode",
      "anno" : "1990"
    },
    {
      "titolo" : "Starlight",
      "autore" : "Muse",
      "anno" : "2006"
    },
    {
      "titolo" : "Back In Black",
      "autore" : "AC/DC",
      "anno" : "1980"
    }
  ]
}
```

Risultato dopo la votazione:

```
{
  "canzone_votata": {
    "titolo": "Back In Black",
    "anno": "1980",
    "autore": "AC/DC"
  }
}
```

## Licenza e manuale d'uso

### 1. Licenza d'uso

Versione programma 1.0

### 2. Manuale d'uso

Dopo l'esecuzione del programma, l'unica cosa da fare per l'utente è inserire un'estensione accettata e poi votare la canzone che preferiscono, dopodiché il programma penserà a preparare il file con il risultato formattato del tipo precedentemente richiesto.