Deduplicatore di immagini

1 Introduzione 3

1.1 Informazioni sul progetto 3

1.2 Abstract 3

1.3 Scopo 3

2 Analisi 4

2.1 Analisi del dominio 4

2.2 Analisi e specifica dei requisiti 4

2.3 Use case 5

2.4 Pianificazione 6

2.5 Analisi dei mezzi 7

2.5.1 Software 7

2.5.2 Hardware 7

3 Progettazione 7

3.1 Design delle interfacce 7

3.2 Design procedurale 9

4 Implementazione 10

4.1 Deduplicatore 10

4.2 ImageMenuPanel 16

4.3 MainFrame 19

5 Test 21

5.1 Protocollo di test 21

5.2 Risultati test 22

5.3 Mancanze/limitazioni conosciute 22

6 Consuntivo 23

7 Conclusioni 25

7.1 Sviluppi futuri 25

7.2 Considerazioni personali 25

8 Glossario 26

9 Bibliografia 27

9.1 Sitografia 27

9.2 Indice delle figure 27

10 Allegati 28

# Introduzione

## Informazioni sul progetto

* Allievo: Edoardo Ratti Docente: Geo Petrini
* SAMT sezione infomatica modulo 306
* 09.09.2022 -- 23.12.2022

## Abstract

Con il miglioramento della tecnologia i dispositivi sono e saranno sempre dotati di memorie più grandi ma anche di file più dettagliati e di conseguenza di dimensioni maggiori; dunque, rimane il bisogno di limitare lo spazio utilizzato al minimo possibile evitando gli sprechi.

Io per ovviare al problema ho deciso di sviluppare un applicativo in grado di classificare le immagini in base alla loro similitudine, allo scopo di poter cancellare quelle doppie e quelle molto simili.

Inoltre alla risoluzione di questo problema il programma è anche in grado di darci un’idea delle nostre immagini, permettendo anche di poterle visualizzare senza aprile, dunque risparmiare tempo.

## Scopo

Lo scopo del progetto è quello di separare le immagini simili raggruppandole in base alla loro similitudine, questo può per esempio aiutare l’utente a ottimizzare lo spazio, consigliando quali immagini sono ridondanti.

Inoltre, lo sviluppo del progetto comporta ad approfondimento delle conoscenze nel linguaggio Java e della sua libreria opencv.

Parlando del progetto in generale posso anche dire di aver incrementato le mie capacità inerenti alla progettazione, allo sviluppo di design e tutto l’insieme di file documentativi come documentazione e diari.

# Analisi

## Analisi del dominio

Il mio applicativo funge da riduttore di spazio inutile, è pensato per le per persone con grandi quantità di fotografie, le quali hanno problemi di spazio, non sapendo quali immagini sono realmente simili.

Il programma per risolvere il problema raggruppa tutte le immagini simili tra di loro, allo scopo di avere una visione completa di ciò che si dispone. Il software è adatto a tutti, è molto semplice e intuitivo da utilizzare e non necessità competenze particolari per il suo utilizzo, ma semplicemente il saper utilizzare in compilatore.

## Analisi e specifica dei requisiti

Il prodotto in questione che funge da riduttore di spazio utilizzato è stato commissionato per far fronte al mercato della fotografia, dunque per alcuni mestieri come fotografi e fanartist.

Esso sarà disposto di un’interfaccia molto semplice ed intuitiva per facilitare il lavoro degli utenti, ma anche relativamente spoglia per adattare più facilmente nuove feature.

|  |  |
| --- | --- |
| **ID: REQ-001** | |
| **Nome** | Formati supportati |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | JPG, PNG, WEBP |

|  |  |
| --- | --- |
| **ID: REQ-002** | |
| **Nome** | Scansione ricorsiva |
| **Priorità** | 1 |
| **Versione** | 1.0 |

|  |  |
| --- | --- |
| **ID: REQ-003** | |
| **Nome** | Rilevamento di immagini |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Sotto requisiti** | |
| **001** | Rilevamento immagini identiche |
| **002** | Rilevamento immagini simili |

|  |  |
| --- | --- |
| **ID: REQ-004** | |
| **Nome** | Mostrare lista immagini raggruppate |
| **Priorità** | 2 |
| **Versione** | 1.0 |
| **Note** | Anteprima immagini + nome |

|  |  |
| --- | --- |
| **ID: REQ-005** | |
| **Nome** | Possibilità di salvare i risultati |
| **Priorità** | 2 |
| **Versione** | 1.0 |

|  |  |
| --- | --- |
| **ID: REQ-006** | |
| **Nome** | Gestione scansione |
| **Priorità** | 2 |
| **Versione** | 1.0 |
| **Sotto requisiti** | |
| **001** | Interruzione |
| **002** | Ripresa |

## Use case

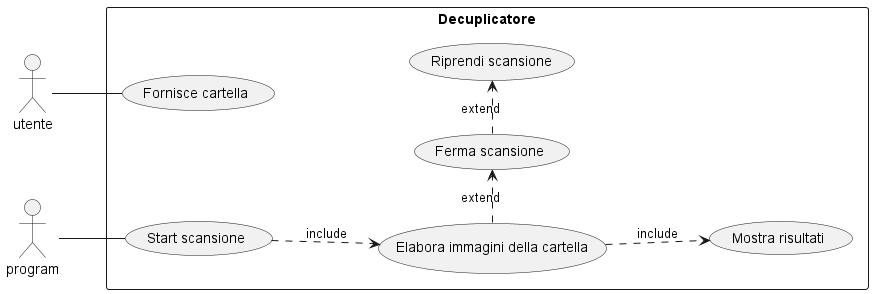


Figura 1 – UseCase

## Pianificazione

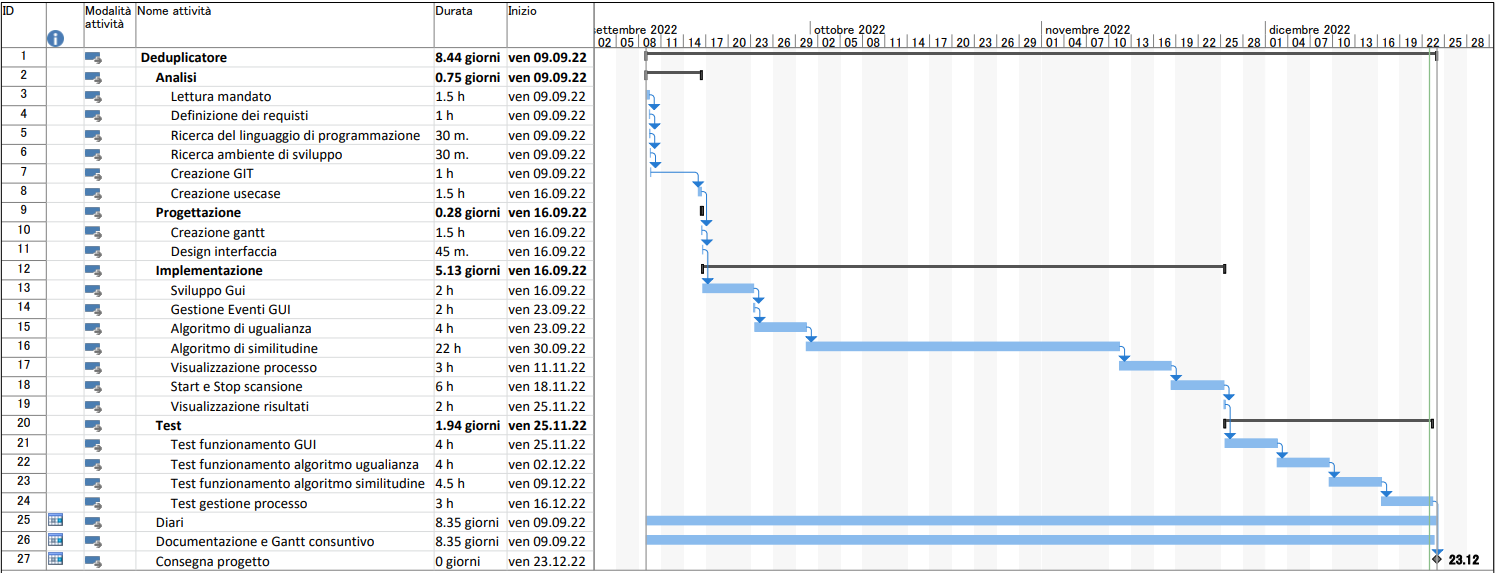


Figura 2 -- Diagramma di Gantt

## Analisi dei mezzi

Durante lo svolgimento del progetto con l’ausilio del computer scolastico sono a andato a ricercare alcuni dati che mi servivano al compimento del codice, in particolare mi serviva una libreria in grado di riconoscere i pattern tra oggetti, tra diversi algoritmi di definizione dei pattern ho scelto SIFT, questo perché molti altri pacchetti erano a pagamento e richiedevano licenze.

### Software

* Apache NetBeans IDE 12.4
* Java JDK 17.0.5
* Opencv 4.5.5
* PlanUML 1.2022.14
* Microsoft Word
* Microsoft Project

### Hardware

* CPU: i7-9700
* RAM: 32GB

# Progettazione

## Design delle interfacce

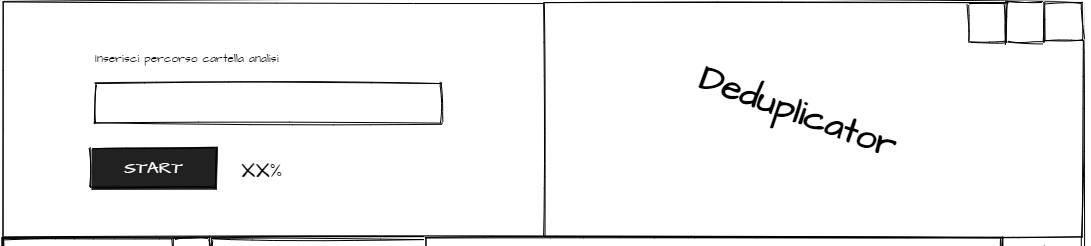


Figura 3 -- Design MainFrame

Interfaccia costituita da una parte che permette la gestione del processo, dove sarà possibile cambiare percorso, far partire e fermare la scansione con tanto di visibilità della percentuale di esecuzione.

Premendo sullo start il pulsante cambierà funzione diventando “STOP” e facendolo un'altra volta “RESTART”.

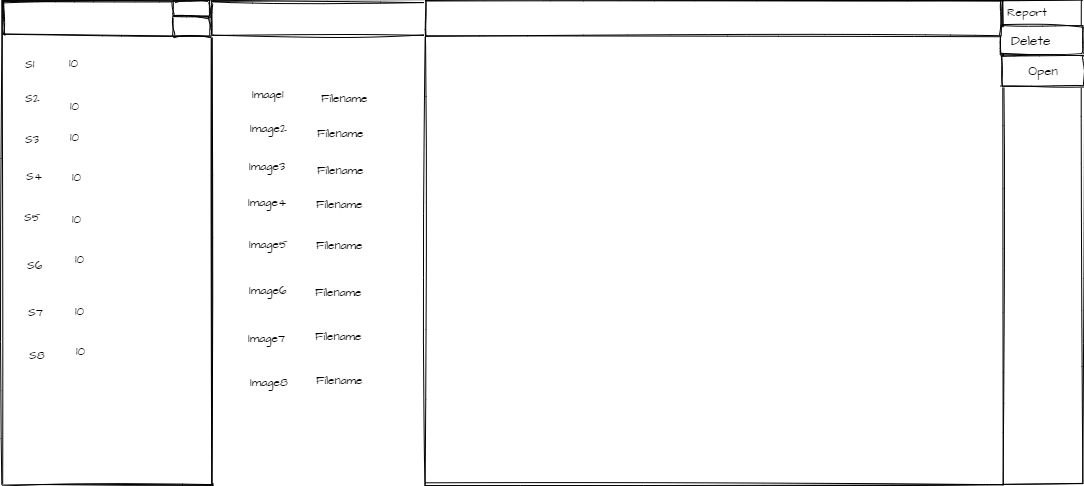


Figura 4 -- ImageMenuPanel

Nella seconda parte è possibile selezionare le singole serie da visualizzare tramite l’input predisposto, tale comportamento porta alla visualizzazione delle immagini contenute in una serie specifica, inoltre sono presenti dei pulsanti per il report della serie, cancellazione immagine e download immagine.

## Design procedurale



Figura 5 -- Diagramma UML delle classi

# Implementazione

Installare opencv e javacv con le varie JAR e DLL nel progetto.

## Deduplicatore

Caricare opencv nella classe



Figura 6 -- Caricamento della libreria

Creare una costante stringa contenente tutte le estensioni possibili desiderate.



Figura 7 -- Array esensioni

Creare il filtro con l’ausilio del metodo *accept* che ci permette di stabilire il filtro sulle estensioni scelte.

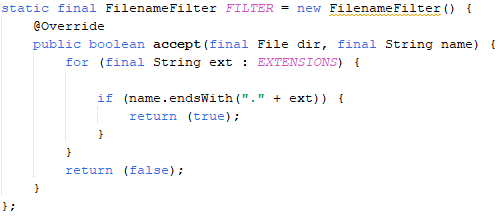


Figura 8 -- Filtro estensioni

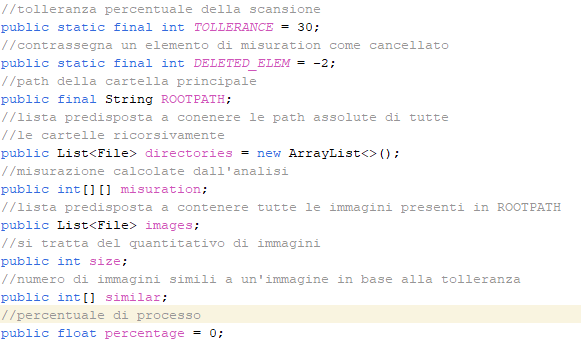


Figura 9 -- Attributi della classe Deduplicatore

Questo metodo ricorsivo permette di ottenere un’alberatura completa in base alla root path selezionata in modo ricorsivo.

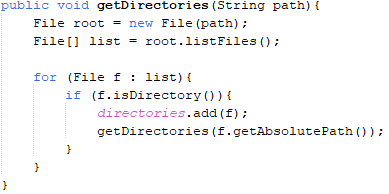


Figura 10 -- getDirectories

Questo metodo invece permette di ottenere tutte le immagini contenute in una cartella in base ad un filtro.

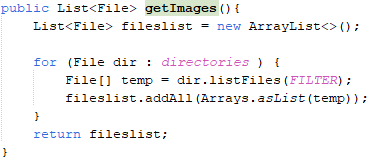


Figura 11 -- getImages

Metodo che permette di rimuove le serie ridondanti, ovvero con dati identici ma in posizione dell’array *misuration* differenti.

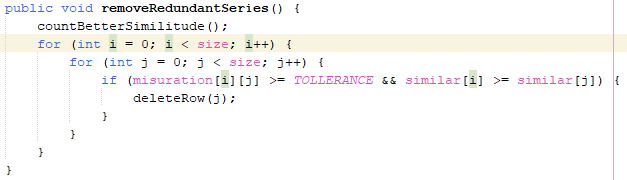


Figura 12 -- removeRedundantSeries

Questo seguente particolare metodo ha il compito di calcolare quante immagini simili abbia effettivamente ogni immagine e quanto lo siano.

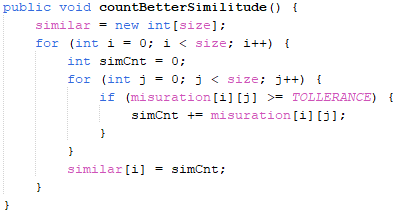


Figura 13 -- countBetterSimilitude

Il sottostante metodo serve a eliminare assegnando il valore di *DELETED\_ELEM* a tutte le immagini che sono già state assegnate a una serie e quelle con un valore minore di *TOLLERANCE*.

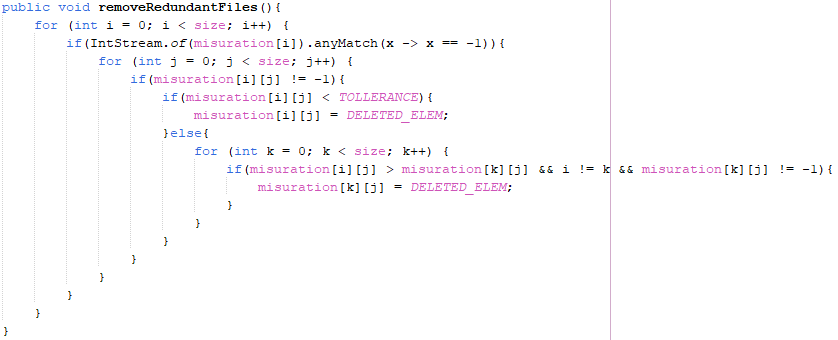


Figura 14 -- removeRedundantFiles

Si tratta del metodo più importante dell’intero programma, esso è proprio quello che va a calcolare la similitudine tra le immagini; infatti, esso possiede due parametri stringa contenenti le paths delle immagini che permetteranno di ottenere degli oggetti *Mat* appartenenti alla libreria installata.

Nelle due variabili descriptor1 e 2 verranno immagazzinati i numeri di descrittori delle immagini, ovvero il numero di punti simili trovati nella scansione che poi verranno rapportati per ottenere una percentuale di similitudine attendibile.

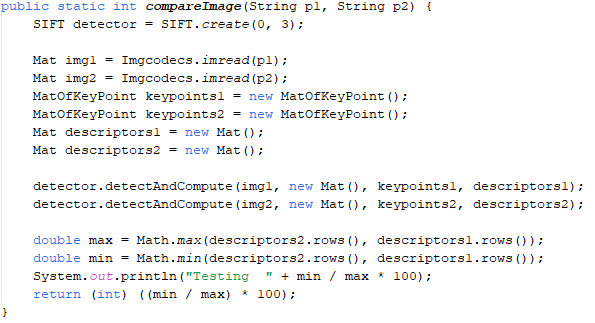


Figura 15 -- compareImage

Il metodo *analyseImage* è quello che mette assieme tutti gli altri metodi visti in precedenza allo scopo di paragonare tutte le immagini con se tutte le altre e già rimuovendo in gran parte i calcoli di dati ridondanti grazie alla consapevolezza che i dati sono specchiati in diagonale e che questa fornirà solo valori del 100%.

Alla fine del metodo c’è la stampa dell’array bidimensionale *misuration*, per poter contemplare i dati ottenuti dall’analisi.



Figura 16 -- analyseImage

Il seguente metodo serve a notificare alla classe genitore del cambiamento della percentuale di esecuzione che cambia ciclicamente all’analisi di due immagini.

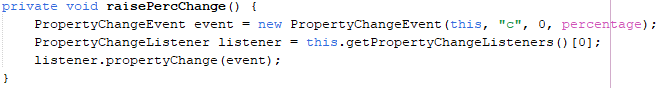


Figura 17 -- raisePercChange

Codesto svolge la funzione di offrire al client l’analisi dei dati anche su un file di testo oltre che nell’interfaccia durante l’esecuzione.

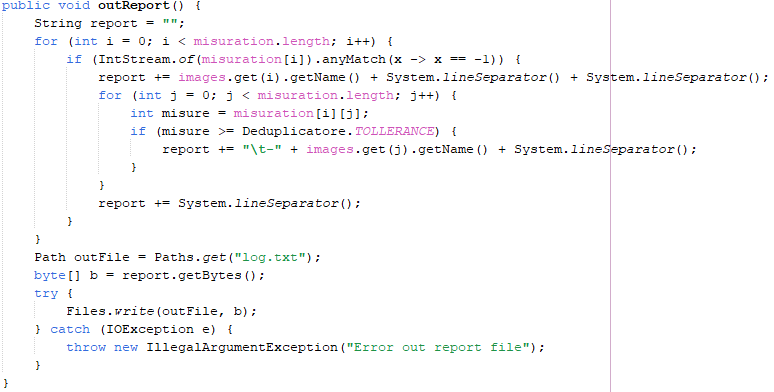


Figura 18 -- outReport

## ImageMenuPanel

Proprietà utilizzate in questa classe

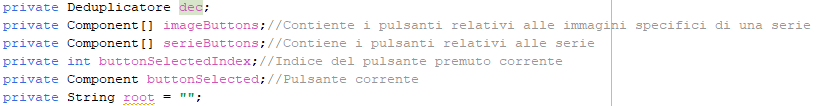


Figura 19 -- Attributi della classe ImageMenuPanel

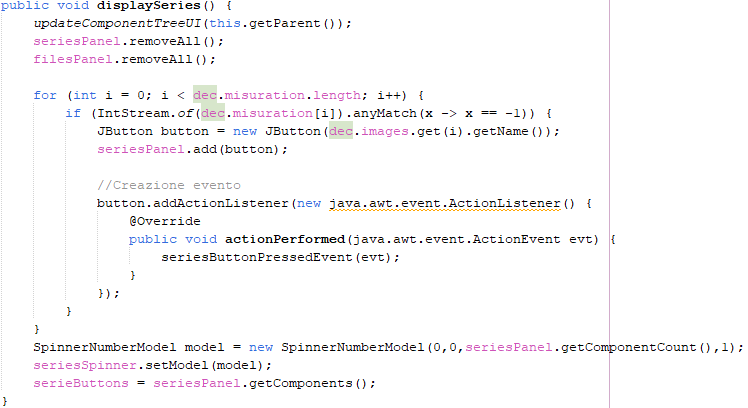
Metodo che mostra le serie sull’interfaccia e assegna ad ogni pulsante inerente ad una serie l’evento che permetterà la selezione di quest’ultimo. Alla fine, è presente un pezzo di codice che stabilisce le dimensioni dello spinner presente nell’interfaccia in base alla quantità di serie disponibili. 

Figura 20 -- displaySeries

Proprio come il metodo spiegato sopra questo secondo serve a svolgere il medesimo compito, ma la sostanziale differenza è che questo mostra invece le immagini al posto delle serie.

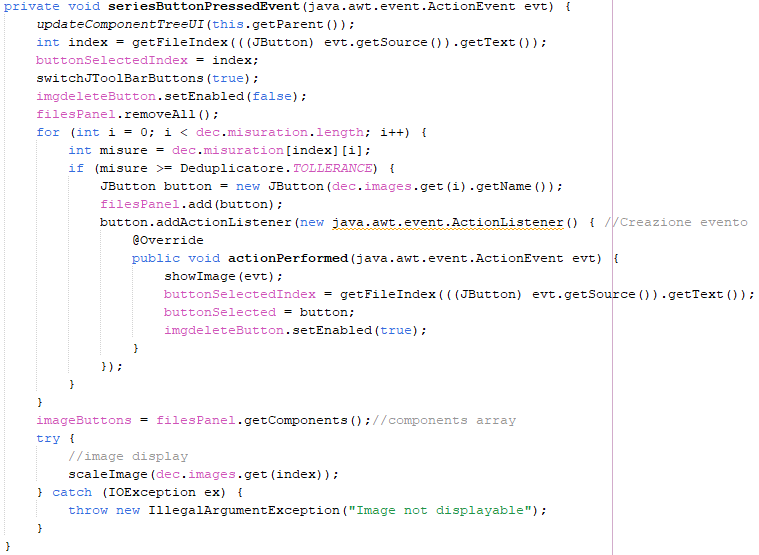


Figura 21 -- seriesButtonPressedEvent

I sottostanti due metodi hanno la funzione di filtrare la ricerca all’interno dell’interfaccia il primo filtra in base alla presenza della sottostrigna scritta nel campo di testo nelle immagini disponibili, mentre nel secondo caso tramite lo spinner sarà possibile selezionare una serie in base al numero di indice. Il numero zero è quello che mostra tutte le serie.

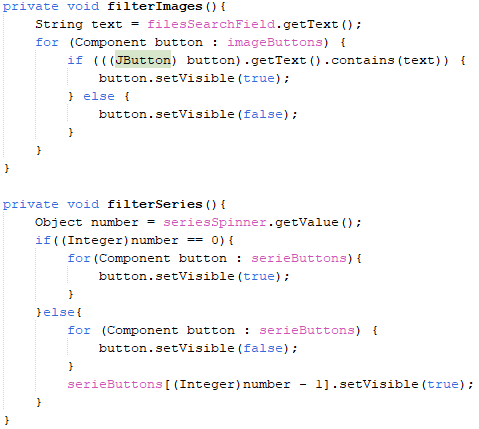


Figura 22 -- Metodi per filtrare i record

Semplicemente serve a cambiare lo stato dei tre pulsanti alla destra dell’interfaccia tra abilitato e non abilitato, allo scopo di agevolare il lavoro di quei tre.

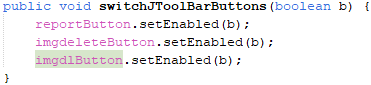


Figura 23 -- switchJToolBarButtons

*ScaleImage,* si occupa di fare in modo che l’immagine che deve comparire sia mostrata in scala.

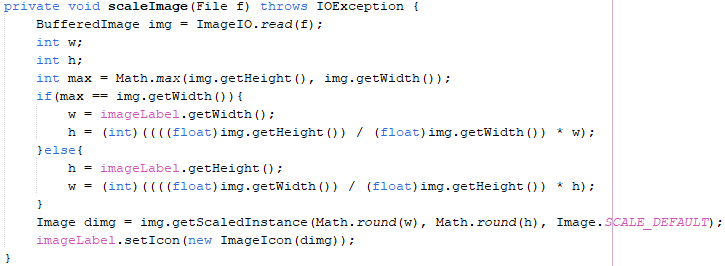


Figura 24 -- scaleImage

## MainFrame

Per cominciare ho creato una piccola GUI contenente anche imageMenuPanel che permette di interagire in maniera ottimale con il programma.

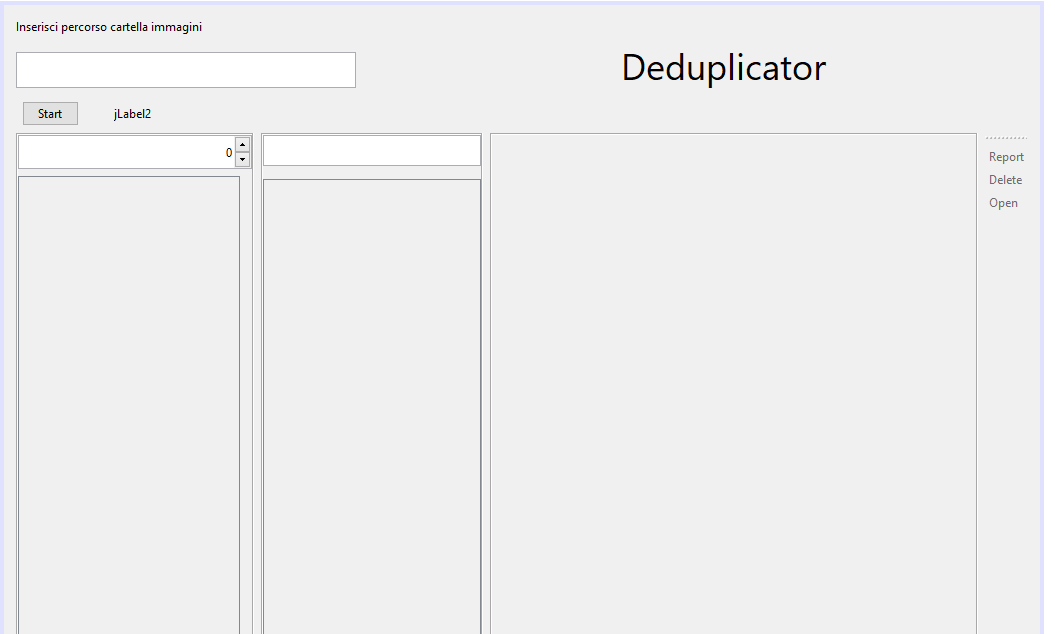


Figura 25 -- GUI



Figura 26 -- Attributi della classe MainFrame

Metodo che permette di scriver la percentuale di esecuzione all’interno della GUI, purtroppo il metodo non è funzionante, per qualche motivo non fa la stampa nell’etichetta, però è comunque in grado di stampare il numero corretto nel terminale.

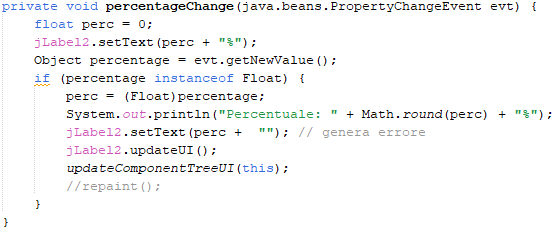


Figura 27 -- percentageChange

# Test

## Protocollo di test

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-001  REQ-001 | **Nome:** | Rilevamento formato immagini |
| **Descrizione:** | Controllo se l’immagini hanno un formato supportato | | |
| **Prerequisiti:** |  | | |
| **Procedura:** | 1. Cominciare una scansione con formati supportati e non supportati | | |
| **Risultati attesi:** | Nel listing vengono mostrate solo le immagini con formati supportati | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-002  REQ-002 | **Nome:** | Test di ricorsività |
| **Descrizione:** | Controllo se il programma raggiunge tutte le immagini | | |
| **Prerequisiti:** | TC-001 | | |
| **Procedura:** | 1. Mettere nella root delle cartelle immagini e cartelle con a loro volta altre immagini | | |
| **Risultati attesi:** | Tutte le immagini sono presenti nel listing | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-003  REQ-003 | **Nome:** | Test di comparazione immagini |
| **Descrizione:** | Controllo se l’immagine uguale o simile | | |
| **Prerequisiti:** | TC-001, TC-002 | | |
| **Procedura:** | 1. Cominciare una scansione con immagini simili, uguali e differenti 2. Controllare I risultati | | |
| **Risultati attesi:** | Immagine identica: =100% - Immagine simile: ≠ 100%, >0% | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-004  REQ-004 | **Nome:** | Mostra lista files |
| **Descrizione:** | La lista compare | | |
| **Prerequisiti:** | TC-003 | | |
| **Procedura:** | 1. Cominciare una scansione 2. Selezionare una serie | | |
| **Risultati attesi:** | Il pannello cambia mostrando i file della serie selezionata | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-005  REQ-005 | **Nome:** | Apertura immagini |
| **Descrizione:** | È possibile aprire le immagini | | |
| **Prerequisiti:** | TC-004 | | |
| **Procedura:** | 1. Cominciare una scansione 2. Premere sul pulsante “apri” | | |
| **Risultati attesi:** | Sarà aperta l’immagine | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-006  REQ-006 | **Nome:** | Interruzione e restart scansione |
| **Descrizione:** | Possibilità di gestire la scansione | | |
| **Prerequisiti:** | TC-004 | | |
| **Procedura:** | 1. Cominciare una scansione 2. Interrompere la scansione 3. Ricominciare la scansione | | |
| **Risultati attesi:** | Ogni volta è cambiato lo stato della scansione in maniera corretta | | |

## Risultati test

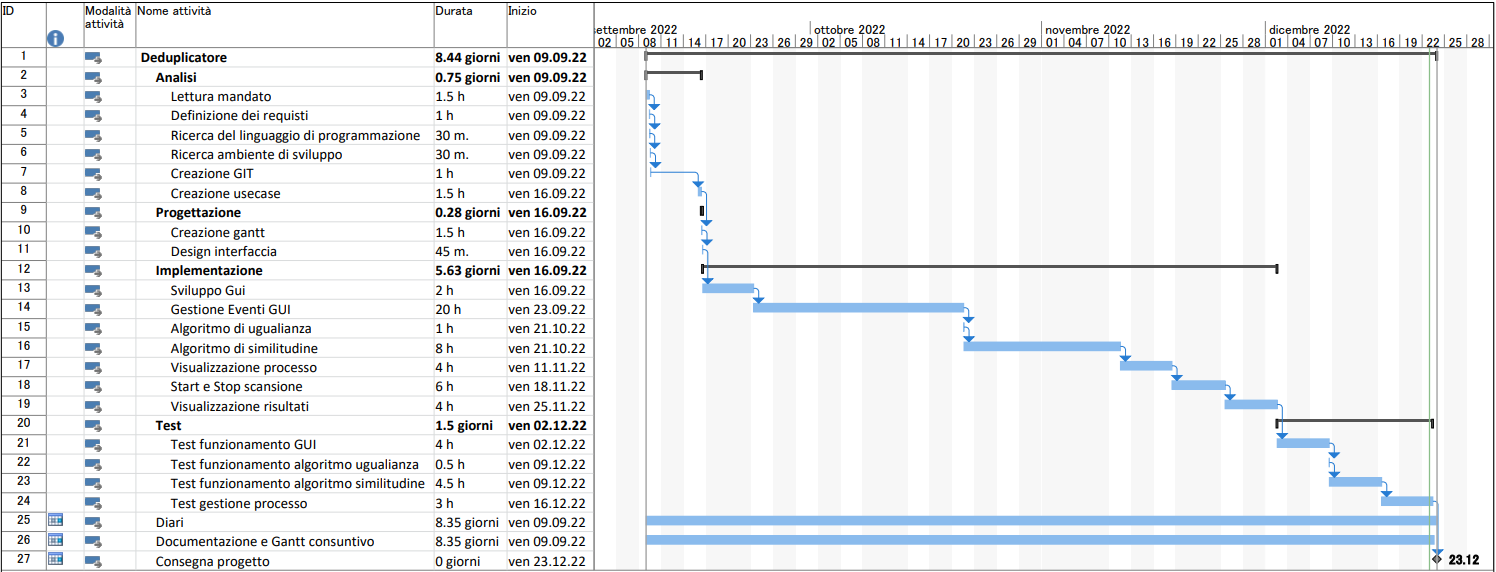
|  |  |
| --- | --- |
| Test Case: | Risultato: |
| TC-001 | Passato |
| TC-002 | Passato |
| TC-003 | Passato |
| TC-004 | Passato |
| TC-005 | Passato |
| TC-006 | Non passato |

Per quanto riguarda il TC-006, unico test non passato, ho pensato a una miglioria applicabile, ovvero un migliore utilizzo delle funzioni di threading, relativo al cambiamento di posizione della creazione del thread.

## Mancanze/limitazioni conosciute

Durante l’implementazione delle funzionalità ho avuto un problema con l’utilizzo delle thread, non sono riuscito ad ottenere risultato atteso nel TC-006, e di conseguenza nemmeno a far comparire la percentuale di esecuzione a schermo.

# Consuntivo

  
Figura 28 -- Gantt Consuntivo

Il gantt consuntivo ha alcuni cambiamenti, ma uno è stato quello più grande. Durante la pianificazione ero convinto che la parte di creazione dell’algoritmo di similitudine avrebbe richiesto l’intervento maggiore di tempo nel progetto, ma infine non è stato così infatti una parte che pensavo fosse molto semplice, ovvero la parte riguardante la gestione eventi della GUI è stata quella che mi ha fatto occupare la maggior parte del tempo, infatti era quella dove ho riscontrato più problemi, avendo il numero maggiore di attività da svolgere, anche se piccole richiedevano una breve ricerca di informazioni non indifferente.

# Conclusioni

## Sviluppi futuri

Come future migliorie io punto sull’aumento di features relative alle serie e alle immagini, come il report e l’apertura immagine, l’aumento di scansioni simultanee, utilizzo di un esplora risorse al posto di un campo di testo per la selezione del percorso, aumento delle informazioni relative ai record.

## Considerazioni personali

Dopo questo periodo semestrale dove dedicavo una parte del mio tempo a questo progetto posso dire di essere contento del mio risultato. Penso che il mio progetto anche se non sia di fondamentale utilità, o meglio non si tratti di un applicativo che si usi moltissimo possa essere un cambiamento, ha cambiato in parte il mio modo di pensare, programmare, ma sicuramente mi ha aperto gli occhi.

Ha fatto approfondire le mie conoscenze nel linguaggio java: ho imparato ad installare librerie e sfruttare meglio le possibilità che offre, come l’utilizzo di liste, thread ed eventi. Adesso sono in grado di sviluppare un’interfaccia grafica e di implementare al suo interno controlli e funzioni.

Sicuramente la mia soluzione ha dei difetti, ma io sono felice di portali dietro, sono arrivato fino al risultato che ho ottenuto solo grazie agli sforzi che ho dato, dunque agli errori che ho portato avanti allo scopo di risolverli. Credo di aver ottenuto un buon risultato, anche se non sono riuscito a completarlo con tutte le caratteristiche che volevo implementare comunque sono arrivato ad un prodotto funzionante.

’

# Glossario

|  |  |
| --- | --- |
| **Termine** | **Descrizione** |
| Fanartist | Professione, creatore di disegni e immagini stilizzati |
| Spinner | Controllo input che permette di scalare i numeri in base a due pulsanti |

# Bibliografia

## Sitografia

* <https://wireframepro.mockflow.com>, Wireframepro
* <https://github.com/bytedeco/javacv>, Github
* <https://sourceforge.net/projects/opencvlibrary/files/4.5.5/opencv-4.5.5-vc14_vc15.exe>, Souce Forge
* <https://docs.opencv.org/4.x/d5/dde/tutorial_feature_description.html>, Opencv
* <https://docs.opencv.org/4.5.5/d7/d60/classcv_1_1SIFT.html>, Opencv
* <https://docs.opencv.org/3.4/db/d39/classcv_1_1DescriptorMatcher.html>, Opencv
* <https://docs.opencv.org/4.5.5/d2/d6e/classcv_1_1StereoMatcher.html>, Opencv

## Indice delle figure

[Figura 1 -- UseCase 5](#_Toc122694127)

[Figura 2 -- Diagramma di Gantt 6](#_Toc122694128)

[Figura 3 -- Design MainFrame 7](#_Toc122694129)

[Figura 4 -- ImageMenuPanel 8](#_Toc122694130)

[Figura 5 -- Diagramma UML delle classi 9](#_Toc122694131)

[Figura 6 -- Caricamento della libreria 10](#_Toc122694132)

[Figura 7 -- Array esensioni 10](#_Toc122694133)

[Figura 8 -- Filtro estensioni 10](#_Toc122694134)

[Figura 9 -- Attributi della classe Deduplicatore 11](#_Toc122694135)

[Figura 10 -- getDirectories 11](#_Toc122694136)

[Figura 11 -- getImages 11](#_Toc122694137)

[Figura 12 -- removeRedundantSeries 12](#_Toc122694138)

[Figura 13 -- countBetterSimilitude 12](#_Toc122694139)

[Figura 14 -- removeRedundantFiles 12](#_Toc122694140)

[Figura 15 -- compareImage 13](#_Toc122694141)

[Figura 16 -- analyseImage 14](#_Toc122694142)

[Figura 17 -- raisePercChange 14](#_Toc122694143)

[Figura 18 -- outReport 15](#_Toc122694144)

[Figura 19 -- Attributi della classe ImageMenuPanel 16](#_Toc122694145)

[Figura 20 -- displaySeries 16](#_Toc122694146)

[Figura 21 -- seriesButtonPressedEvent 17](#_Toc122694147)

[Figura 22 -- Metodi per filtrare i record 18](#_Toc122694148)

[Figura 23 -- switchJToolBarButtons 18](#_Toc122694149)

[Figura 24 -- scaleImage 19](#_Toc122694150)

[Figura 25 -- GUI 19](#_Toc122694151)

[Figura 26 -- Attributi della classe MainFrame 19](#_Toc122694152)

[Figura 27 -- percentageChange 20](#_Toc122694153)

[Figura 28 -- Gantt Consuntivo 23](#_Toc122694154)

# Allegati

Elenco degli allegati, esempio:

* Diari di lavoro
* Deduplicatore.java, MainFrame.java, ImageMenuPanel.java
* Mandato e QdC
* Prodotto