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

2.2.1 Spiegazione elementi tabella dei requisiti: 5

2.3 Use case 5

2.4 Pianificazione 5

2.5 Analisi dei mezzi 5

2.5.1 Software 6

2.5.2 Hardware 6

3 Progettazione 6

3.1 Design dell’architettura del sistema 6

3.2 Design dei dati e database 6

3.3 Design delle interfacce 6

3.4 Design procedurale 6

4 Implementazione 7

5 Test 7

5.1 Protocollo di test 7

5.2 Risultati test 8

5.3 Mancanze/limitazioni conosciute 8

6 Consuntivo 8

7 Conclusioni 8

7.1 Sviluppi futuri 8

7.2 Considerazioni personali 8

8 Glossario 8

9 Bibliografia 9

9.1 Bibliografia per articoli di riviste: 9

9.2 Bibliografia per libri 9

9.3 Sitografia 9

10 Allegati 9

# Introduzione

## Informazioni sul progetto

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

## Abstract

È una breve e accurata rappresentazione dei contenuti di un documento, senza notazioni critiche o valutazioni. Lo scopo di un abstract efficace dovrebbe essere quello di far conoscere all’utente il contenuto di base di un documento e metterlo nella condizione di decidere se risponde ai suoi interessi e se è opportuno il ricorso al documento originale.

Può contenere alcuni o tutti gli elementi seguenti:

* **Background/Situazione iniziale**
* **Descrizione del problema e motivazione**: Che problema ho cercato di risolvere? Questa sezione dovrebbe includere l'importanza del vostro lavoro, la difficoltà dell'area e l'effetto che potrebbe avere se portato a termine con successo.
* **Approccio/Metodi**: Come ho ottenuto dei progressi? Come ho risolto il problema (tecniche…)? Quale è stata l’entità del mio lavoro? Che fattori importanti controllo, ignoro o misuro?
* **Risultati**: Quale è la risposta? Quali sono i risultati? Quanto è più veloce, più sicuro, più economico o in qualche altro aspetto migliore di altri prodotti/soluzioni?

Esempio di abstract:

*As the size and complexity of today’s most modern computer chips increase, new techniques must be developed to effectively design and create Very Large-Scale Integration chips quickly. For this project, a new type of hardware compiler is created. This hardware compiler will read a C++ program, and physically design a suitable microprocessor intended for running that specific program. With this new and powerful compiler, it is possible to design anything from a small adder, to a microprocessor with millions of transistors. Designing new computer chips, such as the Pentium 4, can require dozens of engineers and months of time. With the help of this compiler, a single person could design such a large-scale microprocessor in just weeks.*

## Scopo

Lo scopo del progetto è quello di separare le immagini simili raggruppandoli in base alla 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

Questo capitolo dovrebbe descrivere il contesto in cui il prodotto verrà utilizzato, da questa analisi dovrebbero scaturire le risposte a quesiti quali ad esempio:

* Background/Situazione iniziale
* Quale è e come è organizzato il contesto in cui il prodotto dovrà funzionare?
* Come viene risolto attualmente il problema? Esiste già un prodotto simile?
* Chi sono gli utenti? Che bisogni hanno? Come e dove lavorano?
* Che competenze/conoscenze/cultura posseggono gli utenti in relazione con il problema?
* Esistono convenzioni/standard applicati nel dominio?
* Che conoscenze teoriche bisogna avere/acquisire per poter operare efficacemente nel dominio?
* …

## Analisi e specifica dei requisiti

Il progettista, dopo aver ricevuto il mandato, in collaborazione con il committente redige una lista di requisiti. Durante questi incontri, tramite interviste (da inserire nei diari), il progettista deve cercare di rispondere alle seguenti domande:

* Quali sono i bisogni del committente?
* Quali funzioni deve svolgere il prodotto?
* Come devono essere implementate?
* L’utente, come vorrebbe/dovrebbe interagire con il prodotto?
* Come verrà utilizzato il prodotto?
* Che tipo di interfaccia si immagina?
* Che prestazioni minime deve fornire il prodotto?
* Che grado di sicurezza deve avere il prodotto?
* …

In base alla lista dei requisiti e all’analisi degli stessi, il progettista redige una *specifica dei requisiti* in cui elenca e descrive in modo dettagliato quali sono le funzionalità che il prodotto fornirà. La specifica dovrebbe essere abbastanza dettagliata da poter essere utilizzata come base per lo sviluppo, ma non troppo; ad esempio non dovrebbe contenere dettagli di implementazione, o definizioni dettagliate dell’interfaccia grafica a meno che questi non siano considerati cruciali. Non si deve scordare che i requisiti non rappresentano delle attività bensì delle caratteristiche che il prodotto dovrà possedere.

|  |  |
| --- | --- |
| **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 |

### Spiegazione elementi tabella dei requisiti:

**ID**: identificativo univoco del requisito

**Nome**: breve descrizione del requisito

**Priorità**: indica l’importanza di un requisito nell’insieme del progetto, definita assieme al committente. Ad esempio, poter disporre di report con colonne di colori diversi ha priorità minore rispetto al fatto di avere un database con gli elementi al suo interno. Solitamente si definiscono al massimo di 2-3 livelli di priorità.

**Versione**: indica la versione del requisito. Ogni modifica del requisito avrà una versione aggiornata.

Sulla documentazione apparirà solamente l’ultima versione, mentre le vecchie dovranno essere inserite nei diari.

**Note**: eventuali osservazioni importanti o riferimenti ad altri requisiti.

**Sotto requisiti**: elementi che compongono il requisito.

## Use case

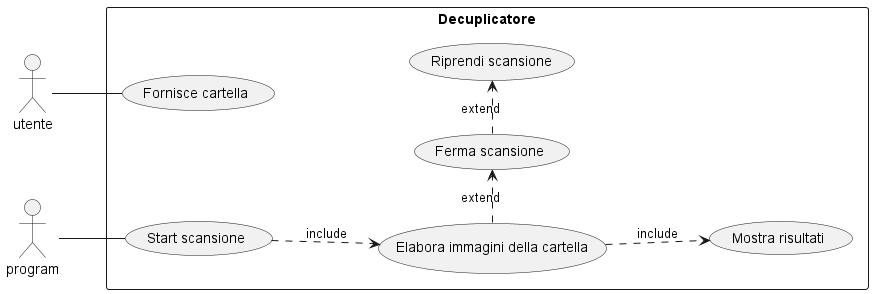
Diagramma UseCase del progetto

Figura – UseCase

## Pianificazione

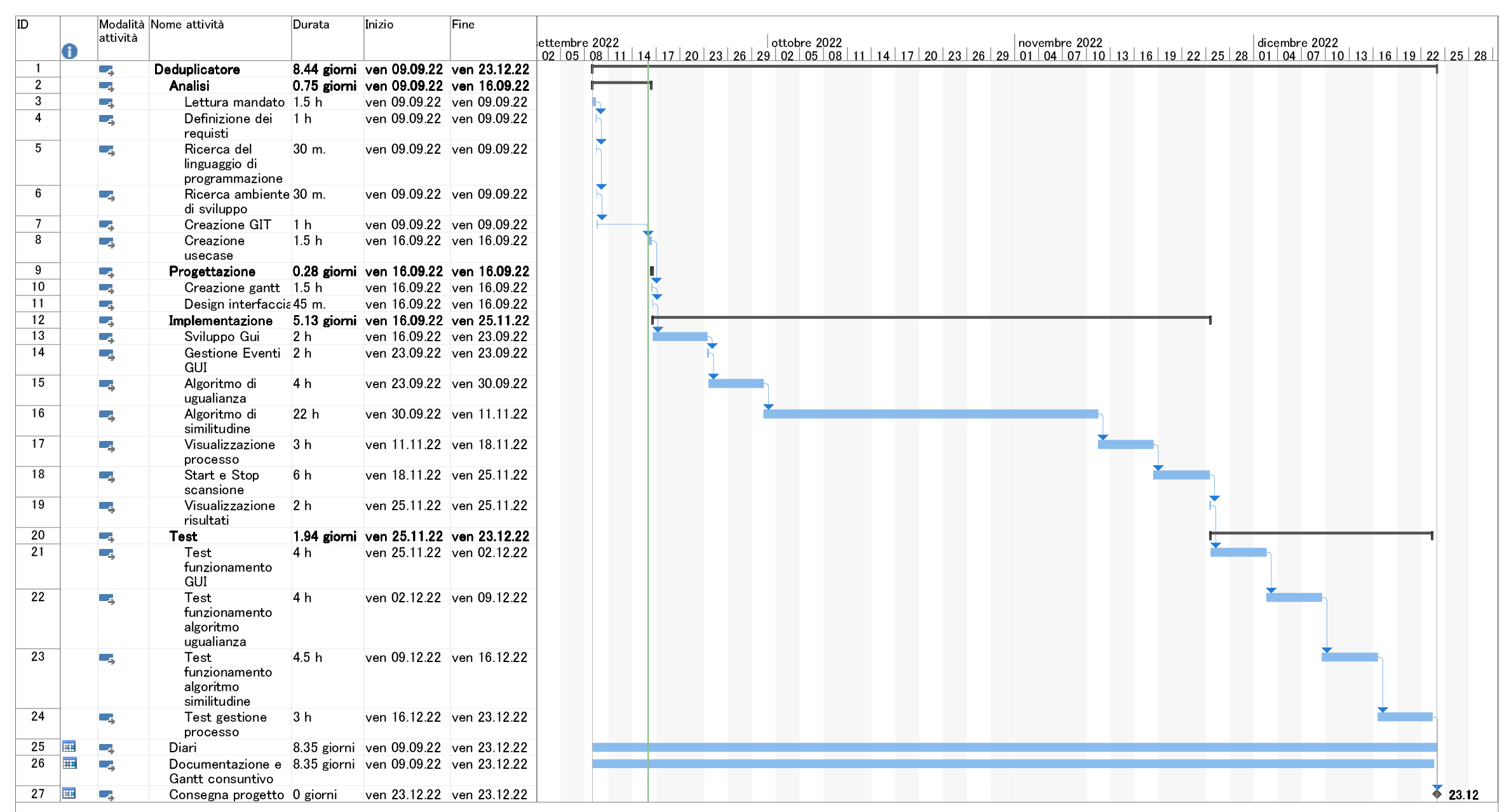


Figura -- Diagramma di Gant

## Analisi dei mezzi

Elencare e descrivere i mezzi disponibili per la realizzazione del progetto. Ricordarsi di sempre descrivere nel dettaglio le versioni e il modello di riferimento.

### Software

SDK, librerie, tools utilizzati per la realizzazione del progetto e eventuali dipendenze.

### Hardware

Su quale piattaforma dovrà essere eseguito il prodotto? Che hardware particolare è coinvolto nel progetto? Che particolarità e limitazioni presenta? Che HW sarà disponibile durante lo sviluppo?

# Progettazione

Questo capitolo descrive esaustivamente come deve essere realizzato il prodotto fin nei suoi dettagli. Una buona progettazione permette all’esecutore di evitare fraintendimenti e imprecisioni nell’implementazione del prodotto.

## Design dell’architettura del sistema

Descrive:

* La struttura del programma/sistema lo schema di rete...
* Gli oggetti/moduli/componenti che lo compongono.
* I flussi di informazione in ingresso ed in uscita e le relative elaborazioni. Può utilizzare *diagrammi di flusso dei dati* (DFD).
* Eventuale sitemap

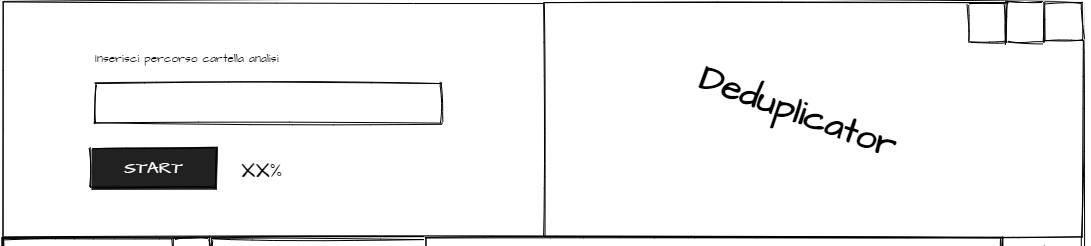
## Design dei dati e database

Descrizione delle strutture di dati utilizzate dal programma in base agli attributi e le relazioni degli oggetti in uso.

Schema E-R, schema logico e descrizione.

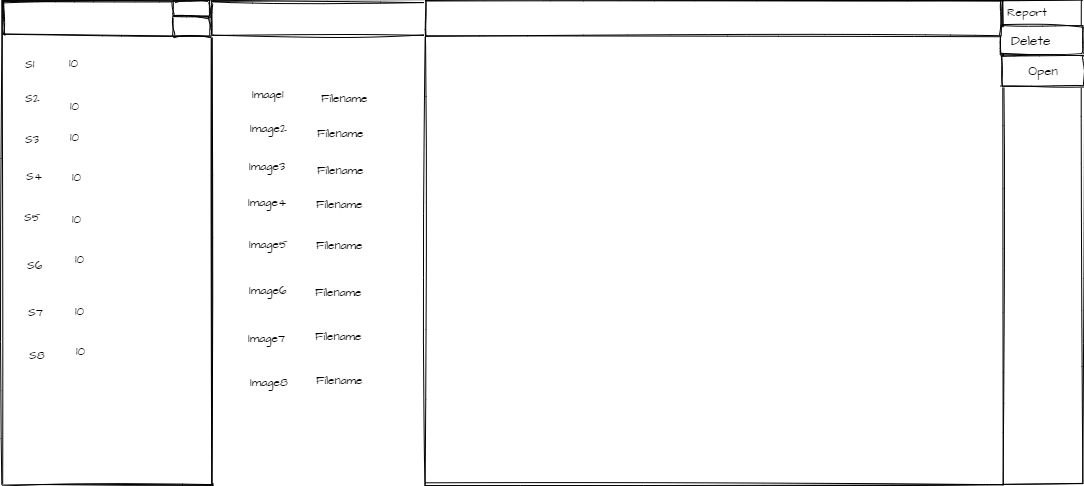
Se il diagramma E-R viene modificato, sulla doc dovrà apparire l’ultima versione, mentre le vecchie saranno sui diari.

## Design delle interfacce



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”.



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

Descrive i concetti dettagliati dell’architettura/sviluppo utilizzando ad esempio:

* Diagrammi di flusso e Nassi.
* Tabelle.
* Classi e metodi.
* Tabelle di routing
* Diritti di accesso a condivisioni …

Questi documenti permetteranno di rappresentare i dettagli procedurali per la realizzazione del prodotto. 

# Implementazione

Installare opencv e javacv con le varie JAR e DLL

## Deduplicatore

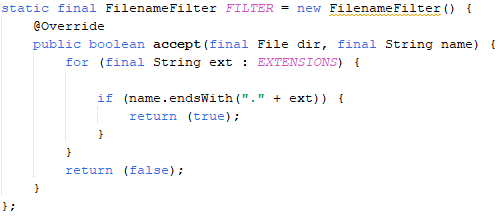
Caricare opencv nella classe



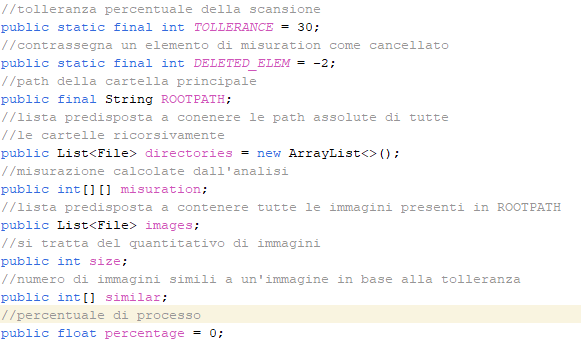
Creare una costante stringa contenente tutte le estensioni che c.



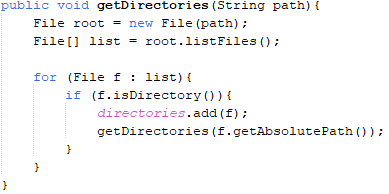
Creare il filtro con l’ausilio del metodo *accept* che ci permette di stabilire il filtro sulle estensioni scelte.



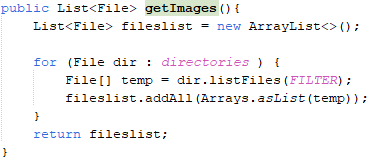
Ecco le proprietà che ho utilizzato durante l’implementazione della classe.



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



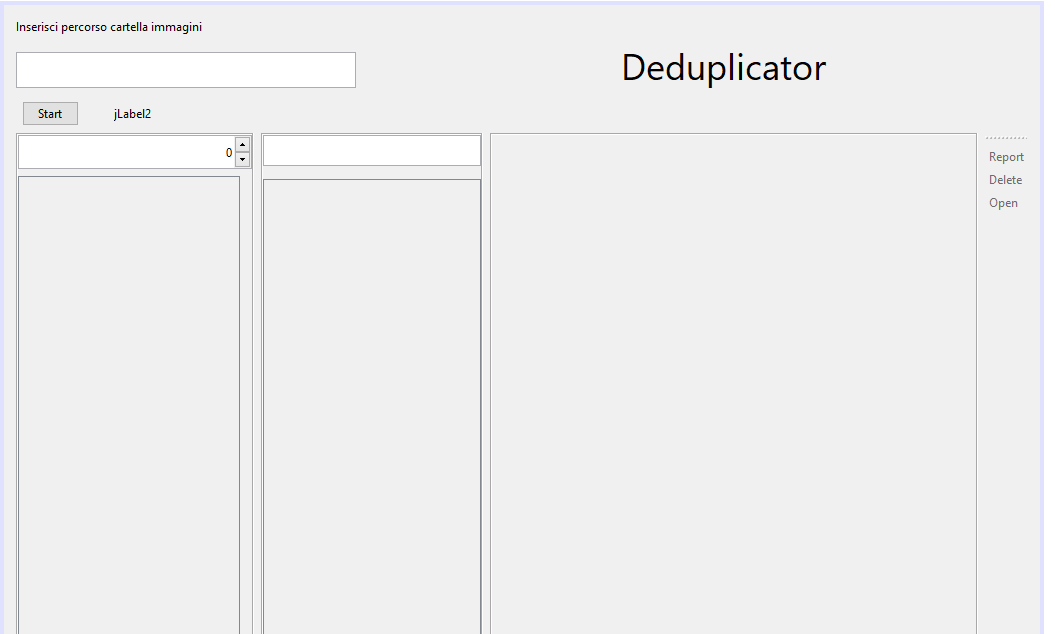
Questo metodo invece permette di ottenere tutte le immagini contenute in una cartella in base ad un filtro.



## ImageMenuPanel

## MainFrame

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



# 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:** | Salvataggio immagini |
| **Descrizione:** | È possibile salvare le immagini | | |
| **Prerequisiti:** | TC-004 | | |
| **Procedura:** | 1. Cominciare una scansione 2. Premere sul pulsante “download” | | |
| **Risultati attesi:** | Sarà effettuato un download dell’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

Tabella riassuntiva in cui si inseriscono i test riusciti e non del prodotto finale. Se un test non riesce e viene corretto l’errore, questo dovrà risultare nel documento finale come riuscito (la procedura della correzione apparirà nel diario), altrimenti dovrà essere descritto l’errore con eventuali ipotesi di correzione.

## Mancanze/limitazioni conosciute

Descrizione con motivazione di eventuali elementi mancanti o non completamente implementati, al di fuori dei test case. Non devono essere riportati gli errori e i problemi riscontrati e poi risolti durante il progetto.

# Consuntivo

Consuntivo del tempo di lavoro effettivo e considerazioni riguardo le differenze rispetto alla pianificazione (cap. 1.7) (ad esempio Gantt consuntivo).

# Conclusioni

Quali sono le implicazioni della mia soluzione? Che impatto avrà? Cambierà il mondo? È un successo importante? È solo un’aggiunta marginale o è semplicemente servita per scoprire che questo percorso è stato una perdita di tempo? I risultati ottenuti sono generali, facilmente generalizzabili o sono specifici di un caso particolare? ecc.

## Sviluppi futuri

Migliorie o estensioni che possono essere sviluppate sul prodotto.

## Considerazioni personali

Cosa ho imparato in questo progetto? ecc.

# Glossario

Inserite una semplice tabella con due colonne che spieghi i termini specifici del progetto (lista dei termini in ordine alfabetico A-Z)

Esempio:

|  |  |
| --- | --- |
| **Termine** | **Descrizione** |
| GUI | **Graphic User Interface:** Interfaccia grafica di un programma con lo scopo di facilitare il contatto pc user. |
| CSS | **Cascading Style Sheets**: linguaggio che permette di definire il layout e la grafica di una pagina web. |

# Bibliografia

## Bibliografia per articoli di riviste:

1. Cognome e nome (o iniziali) dell’autore o degli autori, o nome dell’organizzazione,
2. Titolo dell’articolo (tra virgolette),
3. Titolo della rivista (in italico),
4. Anno e numero
5. Pagina iniziale dell’articolo.

## Bibliografia per libri

1. Cognome e nome (o iniziali) dell’autore o degli autori, o nome dell’organizzazione,
2. Titolo del libro (in italico),
3. ev. Numero di edizione,
4. Nome dell’editore,
5. Anno di pubblicazione,
6. ISBN.

## Sitografia

1. URL del sito (se troppo lungo solo dominio, evt completo nel diario),
2. Eventuale titolo della pagina (in italico),
3. Data di consultazione (GG-MM-AAAA).

**Esempio:**

* http://standards.ieee.org/guides/style/section7.html, *IEEE Standards Style Manual*, 07-06-2008.

# Allegati

Elenco degli allegati, esempio:

* Diari di lavoro
* Codici sorgente/documentazione macchine virtuali
* Istruzioni di installazione del prodotto (con credenziali di accesso) e/o di eventuali prodotti terzi
* Documentazione di prodotti di terzi
* Eventuali guide utente / Manuali di utilizzo
* Mandato e/o QdC
* Prodotto
* …