Documentazione

[1 Introduzione 3](#_Toc70588107)

[1.1 Informazioni sul progetto 3](#_Toc70588108)

[1.2 Abstract 3](#_Toc70588109)

[1.3 Scopo 3](#_Toc70588110)

[2 Analisi 4](#_Toc70588111)

[2.1 Analisi del dominio 4](#_Toc70588112)

[2.2 Analisi e specifica dei requisiti 4](#_Toc70588113)

[2.3 Use case 8](#_Toc70588114)

[2.4 Pianificazione 9](#_Toc70588115)

[2.5 Analisi dei mezzi 10](#_Toc70588116)

[2.5.1 Software 10](#_Toc70588117)

[2.5.2 Hardware 11](#_Toc70588118)

[3 Progettazione 11](#_Toc70588119)

[3.1 Design dell’architettura del sistema 11](#_Toc70588120)

[3.2 Design dei dati e database 11](#_Toc70588121)

[3.3 Design delle interfacce 11](#_Toc70588122)

[3.4 Design procedurale 11](#_Toc70588123)

[4 Implementazione 13](#_Toc70588124)

[5 Test 13](#_Toc70588125)

[5.1 Protocollo di test 13](#_Toc70588126)

[5.1.1 Test funzionali 13](#_Toc70588127)

[5.1.2 Test non funzionali 16](#_Toc70588128)

[5.2 Risultati test 17](#_Toc70588129)

[5.3 Mancanze/limitazioni conosciute 17](#_Toc70588130)

[6 Consuntivo 17](#_Toc70588131)

[7 Conclusioni 17](#_Toc70588132)

[7.1 Sviluppi futuri 17](#_Toc70588133)

[7.2 Considerazioni personali 17](#_Toc70588134)

[8 Bibliografia 17](#_Toc70588135)

[8.1 Bibliografia per articoli di riviste: 17](#_Toc70588136)

[8.2 Bibliografia per libri 18](#_Toc70588137)

[8.3 Sitografia 18](#_Toc70588138)

[9 Allegati 18](#_Toc70588139)

# Introduzione

## Informazioni sul progetto

|  |  |
| --- | --- |
| **Responsabile progetto** | Geo Petrini |
| **Autori** | Thaisa De Torre, Viktorija Tilevska |
| **Inizio** | 14.01.2021 |
| **Consegna** | 20.05.2021 |

## Abstract

E’ 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 di creare un OCR Command Line Interface. L’OCR CLI è uno strumento a linea di comando in grado di estrare il testo dalle immagini e creare un file in formato mettendoci dentro il testo dell’immagine. Inoltre, è possibile vedere le statistiche per ogni scansione, come il tempo impiegato della scansione e il numero delle parole contenute nel file txt.

# Analisi

## Analisi del dominio

Questo applicativo è orientato verso persone che hanno bisogno di avere il contenuto di un’immagine in testo facile da modificare. Visto che l’applicativo si utilizza da linea di comando, gli utenti devono avere una conoscenza base di come funziona la linea di comando.

## Analisi e specifica dei requisiti

|  |  |
| --- | --- |
| **ID: REQ-001** | |
| **Nome** | Acquisizione di immagini in formato PNG o JPG/JPEG |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | L’OCR deve accettare solo immagini RNG e JPG/JPEG |

|  |  |
| --- | --- |
| **ID: REQ-002** | |
| **Nome** | Rilevamento del testo dall’immagine con un algoritmo OCR |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | L'OCR deve rilevare il testo correttamente dall'immagine |

|  |  |
| --- | --- |
| **ID: REQ-003** | |
| **Nome** | Output del testo come TXT |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | Ci deve essere un file TXT che contiene il testo rilevato |

|  |  |
| --- | --- |
| **ID: REQ-004** | |
| **Nome** | Dev’essere bilingue |
| **Priorità** | 2 |
| **Versione** | 1.0 |
| **Note** | Il contenuto può essere interpretato in 2 lingue: Italiano e Inglese |
| **ID: REQ-005** | |
| **Nome** | Upload file multipli con output singolo |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | L’elaborato di più file di input viene ritornato come output singolo.  file.png 🡪 file1.txt file.png file.jpg 🡪 file.txt  Anche con mask: \*.\* 🡪 file.txt |

|  |  |
| --- | --- |
| **ID: REQ-006** | |
| **Nome** | Statistiche scansioni |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | Per ogni esecuzione dev’essere prodotta una statistica con: la quantità di parole rilevate e il tempo di elaborazione |

|  |  |
| --- | --- |
| **ID: REQ-007** | |
| **Nome** | Visualizzazione dati statistici tramite parametro opzionale |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | I dati statistici vengono visualizzati solamente se abilitati tramite parametro |

|  |  |
| --- | --- |
| **ID: REQ-008** | |
| **Nome** | Guida utilizzo |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | Visualizzare una guida di utilizzo tramite parametro (--help). La guida viene visualizzata automaticamente in caso di parametri assenti |

|  |  |
| --- | --- |
| **ID: REQ-009** | |
| **Nome** | Utilità (applicazione) |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | Il lavoro deve poter essere utilizzato in produzione nell’azienda |

|  |  |
| --- | --- |
| **ID: REQ-010** | |
| **Nome** | Gestione degli errori |
| **Priorità** | 2 |
| **Versione** | 1.0 |
| **Note** | eventuali errori sono identificati e gestiti tramite i mezzi adeguati |

|  |  |
| --- | --- |
| **ID: REQ-011** | |
| **Nome** | Registro eventi/Logging |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | L'elaborazione dei dati corretta o errata viene registrata in un file di registro. Al fine di permettere un’interpretazione corretta del registro, i dati pertinenti devono essere memorizzati nel formato appropriato e raggruppati in modo utile. |

|  |  |
| --- | --- |
| **ID: REQ-012** | |
| **Nome** | Tutte le dipendenze del software devono essere incluse |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | Eventuali librerie esterne devono essere incluse e facilmente installabili |

|  |  |
| --- | --- |
| **ID: REQ-013** | |
| **Nome** | Gestione delle eccezioni (batch) |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | Il programma deve essere in grado di funzionare senza supervisioni e di reagire alle potenziali situazioni secondo dei processi predefiniti. In caso d’errore, il programma effettua le azioni definite conformemente alla specifica, scrive i dati nel file di log in modo da permettere un’analisi della situazione che ha portato all’interruzione. |

|  |  |
| --- | --- |
| **ID: REQ-014** | |
| **Nome** | Attendibilità dei dati inseriti dall’utente |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | I campi di immissione sono contrassegnati in modo chiaro e vengono rivisti. Plausibilità: In caso di errore di digitazione, l'utente viene aiutato da indicazioni concrete e il campo corrispondente viene attivato/ evidenziato. |

|  |  |
| --- | --- |
| **ID: REQ-015** | |
| **Nome** | Organizzazione del programma |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | Il programma è stato strutturato in maniera intelligente senza procedure e funzioni e sotto-procedure/funzioni ridondanti. |

|  |  |
| --- | --- |
| **ID: REQ-016** | |
| **Nome** | Utilizzo di diagrammi di flusso |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | I diagrammi di flusso rappresentano lo svolgimento completo del programma. È rappresentato in modo chiaro, facilmente leggibile e contiene anche dei commenti pertinenti. |

## Use case

I casi d’uso rappresentano l’interazione tra i vari attori e le funzionalità del prodotto.

## Pianificazione

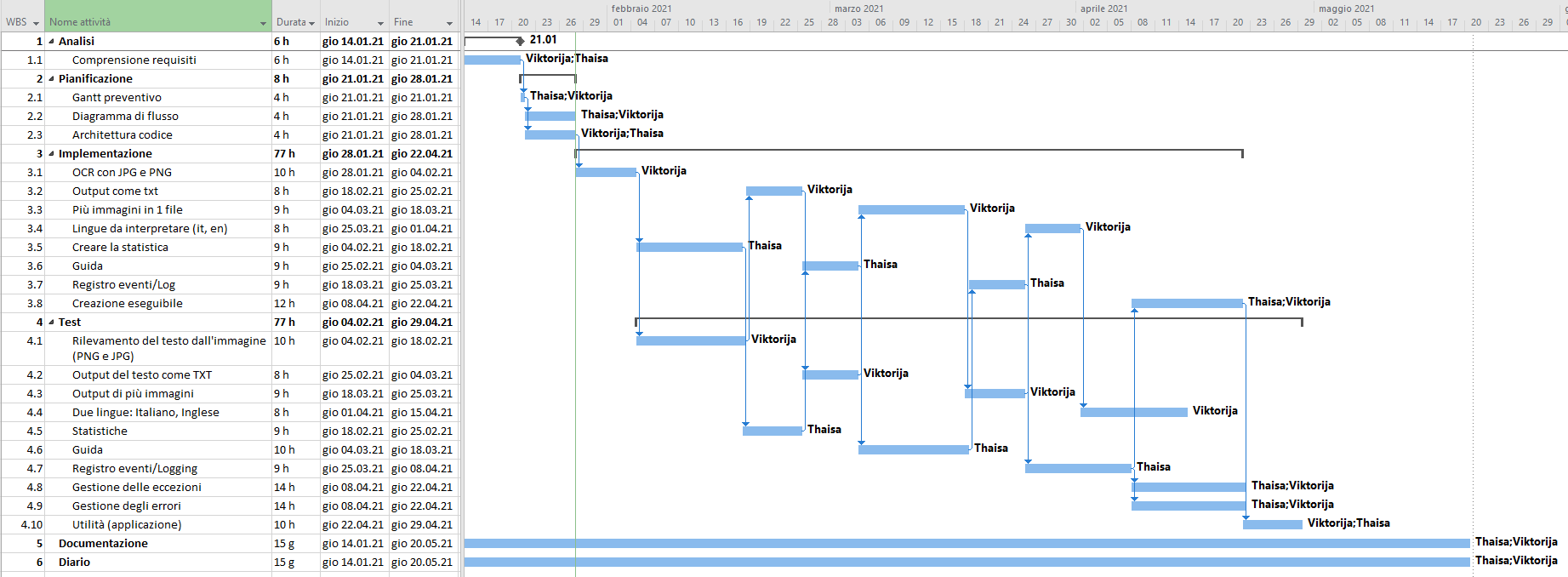
Nella foto sotto è rappresentato il nostro Gantt preventivo:

Figura 1 - Gantt preventivo

Il modello di sviluppo che abbiamo pensato di utilizzare era il modello a cascata (waterfall). C'è lo sviluppo a fasi sequenziali. Come si può vedere dal Gantt, la fine di una fase e l'inizio dell'altra fase.

Abbiamo scelto questo modello perché secondo noi era il modello più efficacie per il progetto. Abbiamo organizzato le fasi in modo che con la fine di una fase si può iniziare l'altra fase. Volevamo lavorare in un modo semplice dove dedichiamo tempo a ogni fase e dopo averla finita fare il test di solo quella fase e così via. In questo modo pensavamo di poter svolgere il lavoro in un modo organizzato senza avere tanti errori e senza dover tornare indietro e fare tutti i test di nuovo.

Figura 2 - Gantt preventivo

Per fare questo progetto avevamo a disposizione 5 mesi (dal 14 gennaio 2021 fino al 20 maggio 2021). Il lavoro l'abbiamo diviso nella fase di analisi, pianificazione, implementazione, test, e alla fine il tempo per la documentazione e il diario.

Abbiamo pensato di dedicare 1 settimana alle analisi, 1 settimana alla pianificazione, 77 ore all'implementazione, altre 77 ore ai test e alla fine un po’ di tempo ogni settimana per il diario e la documentazione.

Nella fase di analisi c'è l'attività di comprensione dei requisiti. Cresta attività consiste nel guardare e rileggere il quaderno e costruire i requisiti necessari per lo svolgimento del progetto. La seguente fase è la fase della pianificazione. In questa fase pensavamo di fare il gantt preventivo, i diagrammi di flusso e di pianificare l'architettura del codice.

La prossima fase è quella dell'implementazione composta da tante attività necessarie per svolgere il progetto. Dopo l'implementazione c'è la fase dei test dove vengono fatti tutti i test. Nel nostro caso il fare d'implementazione e la fase di test vengono svolti contemporaneamente, nel senso che dopo un'attività d'implementazione viene eseguito il suo ripetitivo test. Oltre alle fasi principali abbiamo messo anche la documentazione e il diario come fasi perché abbiamo deciso che ogni settimana dedicheremmo del tempo che per scrivere il diario e la documentazione.

## Analisi dei mezzi

### Software

|  |  |  |  |
| --- | --- | --- | --- |
| **Nome del software** | **Versione** | **Utilizzo** **nel** **progetto** | **Collegamento** |
| Oracle VM VirtualBox | 6.1.16 | Software usato per creare le macchine virtuali | https://www.virtualbox.org/ |
| GitHub | 2.17.1 | Piattaforma che abbiamo utilizzato per gestire il progetto e sincronizzare il lavoro | https://github.com/ |
| GitHub Desktop | 2.6.3 | Interfaccia grafica che semplifica la gestione dei repository di GitHub | https://desktop.github.com/ |
| VisualStudio Code [extention Python] | 1.52.1 | Editor di testo utilizzato per sviluppare il programma in Python | https://code.visualstudio.com/ |
| Python | 3.9.0 | Linguaggio di programmazione | https://www.python.org/ |
| Pytesseract | 0.3.7 | È uno strumento OCR per Python | https://pypi.org/project/pytesseract/ |
| Tesseract OCR | 4.1.1 | Lo strumento OCR usato nel progetto | https://tesseract-ocr.github.io/ |
| Pillow | 8.1.0 | Libreria di Python per l’apertura, la manipolazione e il salvataggio immagini. | https://pillow.readthedocs.io/en/stable/ |
| Draw.io | 14.4.9 | Software per creare dei diagrammi di flusso. | https://app.diagrams.net/ |
| Microsoft Project Professional 2016 | 2016 | Software per creare il diagramma di Gantt. | https://www.microsoft.com/en-us/microsoft-365/project/project-management-software |
| PowerPoint 2016 | 2016 | Software per creare la presentazione. | https://www.microsoft.com/en-us/microsoft-365/powerpoint?ms.officeurl=powerpoint&rtc=1 |

### 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

Descrizione delle interfacce interne ed esterne del sistema e dell’interfaccia utente. La progettazione delle interfacce è basata sulle informazioni ricavate durante la fase di analisi e realizzata tramite mockups.

## 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

In questo capitolo dovrà essere mostrato come è stato realizzato il lavoro. Questa parte può differenziarsi dalla progettazione in quanto il risultato ottenuto non per forza può essere come era stato progettato.

Sulla base di queste informazioni il lavoro svolto dovrà essere riproducibile.

In questa parte è richiesto l’inserimento di codice sorgente/print screen di maschere solamente per quei passaggi particolarmente significativi e/o critici.

Inoltre dovranno essere descritte eventuali varianti di soluzione o scelte di prodotti con motivazione delle scelte.

Non deve apparire nessuna forma di guida d’uso di librerie o di componenti utilizzati. Eventualmente questa va allegata.

Per eventuali dettagli si possono inserire riferimenti ai diari.

# Test

## Protocollo di test

Definire in modo accurato tutti i test che devono essere realizzati per garantire l’adempimento delle richieste formulate nei requisiti. I test fungono da garanzia di qualità del prodotto. Ogni test deve essere ripetibile alle stesse condizioni.

### Test funzionali

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-001  REQ-001 | **Nome:** | Acquisizione di immagini in formato PNG o JPG/JPEG | |
| **Descrizione:** | Verificare che l'OCR prenda solo PNG e JPG/JPEG | | | |
| **Prerequisiti:** | Avere installato l’OCR e avere un’immagine di un testo in formato PNG, JPG/JPEG e un altro formato | | | |
| **Procedura:** | 1. Utilizzare l’OCR con un’immagine JPEG, JPG, PNG e un altro formato   > ocr.py img/engtxtjpeg.jpeg  > ocr.py img/engtxtpng.png  > ocr.py img/itatxtjpg.jpg  > ocr.py img/error.tiff   1. Guardare il file di log per capire quali sono i formati accettati | | | |
| **Risultati attesi:** | L’OCR dovrebbe accettare solo le immagini PNG e JPG/JPEG e per i file con un formato diverso stampa il messaggio d’errore. | | | |
| **Dato** | | | **Risultato** |
| Con PNG | | | OK |
| Con JPG | | | OK |
|  | Con JPEG | | | OK |
|  | Altro formato | | | FAILED |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-002  REQ-002 REQ-003 | **Nome:** | Rilevamento del testo dall’immagine con un algoritmo OCR | |
| **Descrizione:** | L'OCR deve rilevare il testo correttamente dall'immagine | | | |
| **Prerequisiti:** | Avere installato l’OCR e avere un’immagine di un testo in formato JPG/JPEG o PNG. | | | |
| **Procedura:** | 1. Utilizzare l’OCR con un’immagine che rispetta le specifiche che si possono trovare nella documentazione 2. Guardare l’immagine e il file di output e vedere se il testo è uguale | | | |
| **Risultati attesi:** | Il testo rilevato dall’immagine deve coincidere con il testo dell’immagine | | | |
| **Dato** | | | **Risultato** |
| Con PNG o JPG/JPEG | | | Output testo dall’immagine su file TXT |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-003  REQ-004 | **Nome:** | Rilevamento del testo in inglese, italiano e qualsiasi altra lingua | |
| **Descrizione:** | L'OCR deve rilevare il testo correttamente dall'immagine | | | |
| **Prerequisiti:** | Aver installato l’OCR e avere un’immagine di un testo in formato JPG/JPEG o PNG aggiungendo l’argomento –lang | | | |
| **Procedura:** | 1. Utilizzare l’OCR con un’immagine aggiungendo il parametro –l o –lang   > ocr.py img/itatxtjpg.jpg -l eng  > ocr.py img/itatxtjpg.jpg -l ita  > ocr.py img/itatxtjpg.jpg -l fra   1. Guardare il file di log per capire se la lingua inserita è accettata | | | |
| **Risultati attesi:** | L’OCR dovrebbe accettare solo le lingue Italiano e Inglese e per le altre lingue stampa un messaggio d’errore. | | | |
| **Dato** | | | **Risultato** |
| Con PNG/JPG/JPEG in inglese | | | OK |
| Con PNG/JPG/JPEG in italiano | | | OK |
| Altra lingua | | | FAILED |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-004  REQ-005 | **Nome:** | Inserire multipli file di input e ricevere un file di output singolo | |
| **Descrizione:** | Fare il rilevamento di più immagini e scrivere l’output in un singolo file TXT | | | |
| **Prerequisiti:** | Avere un’immagine in formato PNG o JPG/JPEG | | | |
| **Procedura:** | 1. Utilizzare l’OCR con più immagini   > ocr.py img/engtxtjpeg.jpeg img/engtxtpng.png img/itatxtjpg.jpg  > ocr.py img/\*  > ocr.py \*.\*   1. Controllare se nel file di output ci sono i contenuti di tutte le immagini | | | |
| **Risultati attesi:** | Si ottiene un singolo file di output | | | |
| **Dato** | | | **Risultato** |
| Input: img1.png img2.png, img3.jpeg | | | Un file di output .TXT che contiene il contenuto di tutte le immagini inserite |
| Input: img.jpg img/ img1/ img.png | | |
|  | Input: \*.\* | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-005  REQ-006  REQ-007 | **Nome:** | Statistiche scansioni | |
| **Descrizione:** | Controllare che dopo l’esecuzione dell’OCR vengano calcolate le statistiche e controllare che i dati statistici vengano visualizzati tramite il parametro opzionale | | | |
| **Prerequisiti:** | Avere un’immagine in formato PNG o JPG/JPEG | | | |
| **Procedura:** | 1. Utilizzare l’OCR   > ocr.py img/engtxtpng.png –stats | | | |
| **Risultati attesi:** | Alla fine dell’esecuzione a terminale dev’essere stampata la statistica del file.  Le statistiche devono coincidere con il file di output dell’immagine | | | |
| **Dato** | | | **Risultato** |
|  | Con il parametro –stats | | | Stampa le statistiche al terminale |
|  | Senza parametri | | | Esegue l’OCR normalmente senza mostrare le statistiche |
|  | Input: img.txt | | | Le statistiche stampate al terminale coincidono con le statistiche del file di output dell’immagine |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-006  REQ-008 | **Nome:** | Guida utilizzo | |
| **Descrizione:** | Con il parametro –h/--help viene stampata a terminale la guida di utilizzo | | | |
| **Prerequisiti:** | Utilizzare il programma ocr.py | | | |
| **Procedura:** | 1. Utilizzare il comando OCR con il parametro –h o -help   > ocr.py –help | | | |
| **Risultati attesi:** | Viene stampata a terminale la guida di utilizzo | | | |
| **Dato** | | | **Risultato** |
| Con il parametro –h / --h / --help | | | La guida di utilizzo |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-007  REQ-012 | **Nome:** | Dipendenze SW incluse |
| **Descrizione:** | Eventuali librerie esterne devono essere incluse e facilmente installabili | | |
| **Prerequisiti:** | Con l’exe devono essere installate e incluse tutte le dipendenze | | |
| **Procedura:** | 1. Eseguire il .exe dell’OCR 2. Controllare che tutte le dipendenze siano installate | | |
| **Risultati attesi:** | Tutte le dipendenze devono essere incluse e installate con l’exe | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-008  REQ-013 | **Nome:** | Attendibilità dei dati inseriti dall’utente | |
| **Descrizione:** | I campi di immissione sono contrassegnati in modo chiaro e vengono rivisti | | | |
| **Prerequisiti:** | Avere il programma funzionante | | | |
| **Procedura:** |  | | | |
| **Risultati attesi:** |  | | | |
| **Dato** | | | **Risultato** |
| -dest / -d | | | OK |
| -lang / -l | | | OK |
|  | -prefix / -p | | | OK |
|  | Qualsiasi altro parametro | | | FAILED |

### Test non funzionali

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-09  REQ-009 | **Nome:** | Utilità (applicazione) |
| **Descrizione:** | Controllo del funzionamento del programma | | |
| **Prerequisiti:** |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-010  REQ-010 | **Nome:** | Gestione degli errori |
| **Descrizione:** | Verifica se tutti gli errori sono gestiti correttamente | | |
| **Prerequisiti:** |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-011  REQ-011 | **Nome:** | Registro eventi/Logging |
| **Descrizione:** | Verifica se i dati corretti o errati registrati in un file di log;  Verifica se i dati sono salvati nel formato appropriato e raggruppati in modo utile | | |
| **Prerequisiti:** | Utilizzare l’OCR | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-012  REQ-013 | **Nome:** | Gestione delle eccezioni (batch) |
| **Descrizione:** | Controllo se tutti gli errori sono gestiti correttamente | | |
| **Prerequisiti:** |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-013  REQ-015 | **Nome:** | Organizzazione del programma |
| **Descrizione:** | Se il programma è strutturato in metodi ed è commentato | | |
| **Prerequisiti:** |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-014  REQ-016 | **Nome:** | Utilizzo di diagrammi di flusso |
| **Descrizione:** | Se ci sono i diagrammi di flusso | | |
| **Prerequisiti:** |  | | |

## 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.

**Legenda**: test riuscito test fallito

|  |  |  |
| --- | --- | --- |
| ID Test Case | Nome | Risultato |
| TC-001 | Acquisizione di immagini in formato PNG o JPG/JPEG |  |
| TC-002 | Rilevamento del testo dall’immagine con un algoritmo OCR |  |
| TC-003 | Rilevamento del testo in inglese, italiano e qualsiasi altra lingua |  |
| TC-004 | Inserire multipli file di input e ricevere un file di output singolo |  |
| TC-005 | Statistiche scansioni |  |
| TC-006 | Guida utilizzo |  |
| TC-007 | Dipendenze SW incluse |  |
| TC-008 | Attendibilità dei dati inseriti dall’utente |  |
| TC-009 | Utilità (applicazione) |  |
| TC-010 | Gestione degli errori |  |
| TC-011 | Registro eventi/Logging |  |
| TC-012 | Gestione delle eccezioni (batch) |  |
| TC-013 | Organizzazione del programma |  |
| TC-014 | Utilizzo di diagrammi di flusso |  |

## 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 Gannt 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

# 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
* …