Video processing cluster

1 Introduzione 3

1.1 Informazioni sul progetto 3

1.2 Abstract 3

Situazione iniziale 3

Attuazione 3

Risultati 3

1.3 Scopo 3

2 Analisi 4

2.1 Analisi del dominio 4

2.2 Analisi e specifica dei requisiti 4

Analisi 4

specifica dei requisiti 5

2.3 Use case 8

2.4 Pianificazione 9

2.5 Analisi dei mezzi 10

2.5.1 Software 10

2.5.2 Hardware 10

3 Progettazione 11

3.1 Design dell’architettura del sistema 11

3.2 Design dei dati e database 11

3.3 Design delle interfacce 11

3.4 Design procedurale 11

4 Implementazione 12

5 Test 12

5.1 Protocollo di test 12

5.2 Risultati test 16

5.3 Mancanze/limitazioni conosciute 16

6 Consuntivo 16

7 Conclusioni 16

7.1 Sviluppi futuri 16

7.2 Considerazioni personali 16

8 Bibliografia 16

8.1 Bibliografia per articoli di riviste: 16

8.2 Bibliografia per libri 16

8.3 Sitografia 17

9 Allegati 17

# Introduzione

## Informazioni sul progetto

Questo progetto è supervisionato dal docente Geo Petrini e sviluppato dagli allievi: Samuele Abbà, Damian Campesi e Gioele Cavallo.

Il progetto è svolto nella scuola SAMT di Canobbio nelle ore adibite per i progetti dal 27.01.2022 al 05.05.2022.

## Abstract

## Situazione iniziale

All’inizio di questo progetto ci è stato chiesto di realizzare un sistema che permettesse di caricare dei filmati mp4 (di massimo 500MB) in un sito. Il sito permette di ottenere delle statistiche, con relativo grafico, sulla sua composizione (quantità di frame I, B, P) e poterne scaricare la versione: con motion vector o con solo frame I / B / P o tutti i frame I / B / P. In breve: questo progetto è incentrato sull’elaborazione di un portale web che possa gestire la possibilità di caricare un file video e riceverne tutte le sue informazioni comodamente.

## Attuazione

Per questo progetto, saranno necessarie buone conoscenze nella programmazione web e nella sistemistica, poiché complesso nella sua struttura di rete e nelle sue funzionalità che il portale web dovrà offrire. Il client si connettere al load balancer, quest’ultimo reindirizzerà il client ad uno dei due server meno carico di lavoro. Dopo il server mostrerà una pagina dalla quale sarà possibile caricare un file video mp4 di 500MB, dopo che il client effettuerà l’upload del file, il server ne verificherà tutte le caratteristiche e creerà la pagina con le statistiche, inoltre la cartella associata al client contenente il file video e tutti gli altri file necessari per la le statistiche, infine sempre dalla pagina web sarà possibile creare e scaricare la versione del video: con motion vector o con solo frame I / B / P o tutti i frame I / B / P.

## Risultati

Questo progetto è realizzato da zero con i principali linguaggi web (HTML, CSS, PHP) per il sito, con Linux server, Memcached e apache per i due server e con Nginx per il load balancer. Abbiamo quindi avuto possibilità di lavorare su argomenti già consolidati e argomenti non ancora consolidati, questo ci ha permesso di ampliare il nostro bagaglio delle conoscenze durante il progetto.

Uno dei punti fondamentali tenuti in considerazione durante il progetto riguarda lo studio per raggiungere la facilità di utilizzo per un utente medio. Questo portale web è difatti utilizzabile anche da un’utenza che non possiede conoscenze informatiche. In conclusione, questo progetto è ora l’unico sistema che consente di caricare un filmato mp4 (di massimo 500MB) in un sito ed ottenerne le statistiche con la possibilità di scaricarne delle informazioni, tutto ciò è gestito da un load balancer e due server che condividono la memoria.

## Scopo

Scopi

* Didattici:
* Saper creare una progettazione e rispettarla, saper documentare il lavoro, creare i diari e lavorare in team.
* Operativi:
* Saper creare un load balancer, saper creare due server in cluster, saper creare una Gui Web complessa dinamica, saper utilizzare i log, saper fare uno schema di rete, saper utilizzare Linux server e saper utilizzare la sicurezza opportuna.
* Vedi QdC in allegato

# Analisi

## Analisi del dominio

Per questo progetto ci è stato chiesto di creare un sistema in cluster per l’elaborazione di filmati ed estrazione dei vari dati. Oggi per visualizzare le statistiche dei video bisogna passare attraverso molti programmi, ed è tutto meno che immediato. Con il cluster di server ci si assicura che il servizio sia sempre online e grazie alla GUI web il processo diventa user friendly ed intuitivo. Con questo progetto sarà poi possibile visualizzare le statistiche e scaricare il video con i motion vector, solo i frame I/B/P o tutti le immagini che compongono il video in pochi click, senza dover utilizzare svariati programmi.

## Analisi e specifica dei requisiti

## Analisi

Portale web che permetta di caricare e quindi di visualizzare le statistiche e scaricare un video con i motion vector, solo i frame I/B/P o tutti le immagini che compongono il video. Questo servizio deve sempre essere disponibile anche se uno dei due server ha qualche problema, inoltre i dati non devono essere condivisi e non cancellati subito ma dopo 1 ora dalla creazione, questo per aggiungere della ridondanza ai dati e quindi consistenza al servizio. Tutto questo servizio deve lavorare su un sistema di tipo Linux server in modo da utilizzare meno risorse possibili, mentre quando un client si connette ad uno dei due server ottiene una pagina web con la quale può interagire. Questo servizio potrà essere utilizzato da chiunque si possa connettere ad internet.

## specifica dei requisiti

|  |  |
| --- | --- |
| **ID: REQ-01** | |
| **Nome** | Funzionamento load balancer basato sul carico |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | Il load balancer deve reindirizzare le richieste in base al carico dei server |
| **Sotto requisiti** | |
| **01** | Avere il cluster di server |
| **02** | Avere il load balancer configurato |

|  |  |
| --- | --- |
| **ID: REQ-02** | |
| **Nome** | Funzionamento delle sessioni |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | Il server deve gestire le sessioni |
| **Sotto requisiti** | |
| **01** | Avere il server |

|  |  |
| --- | --- |
| **ID: REQ-03** | |
| **Nome** | Funzionamento elaborazione in background dei video caricati |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | Il server deve elaborare i video in background |
| **Sotto requisiti** | |
| **01** | Avere il server |

|  |  |
| --- | --- |
| **ID: REQ-04** | |
| **Nome** | Funzionamento upload di un filmato (massimo 500MB) |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | Il server deve poter ricevere un filmato di massimo 500MB |
| **Sotto requisiti** | |
| **01** | Avere il server |

|  |  |
| --- | --- |
| **ID: REQ-05** | |
| **Nome** | Produrre statistiche sul video |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | Il server deve generare:   * Statistiche sui frame * Grafico sui frame |
| **Sotto requisiti** | |
| **01** | Avere il server |
| **02** | REQ-04 |

|  |  |
| --- | --- |
| **ID: REQ-06** | |
| **Nome** | Possibilità di scaricare differenti contenuti |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | Il server deve generare:   * Video con motion vector * Video con frame I/B/P * Immagini con tutti i frame I/B/P |
| **Sotto requisiti** | |
| **01** | Avere il server |
| **02** | REQ-04 |

|  |  |
| --- | --- |
| **ID: REQ-07** | |
| **Nome** | Cancellazione dati sul db e sul server |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | Il server e il db devono cancellare i dati dopo 1 ora dalla loro creazione |
| **Sotto requisiti** | |
| **01** | Avere il server |
| **02** | Avere il db |

**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.

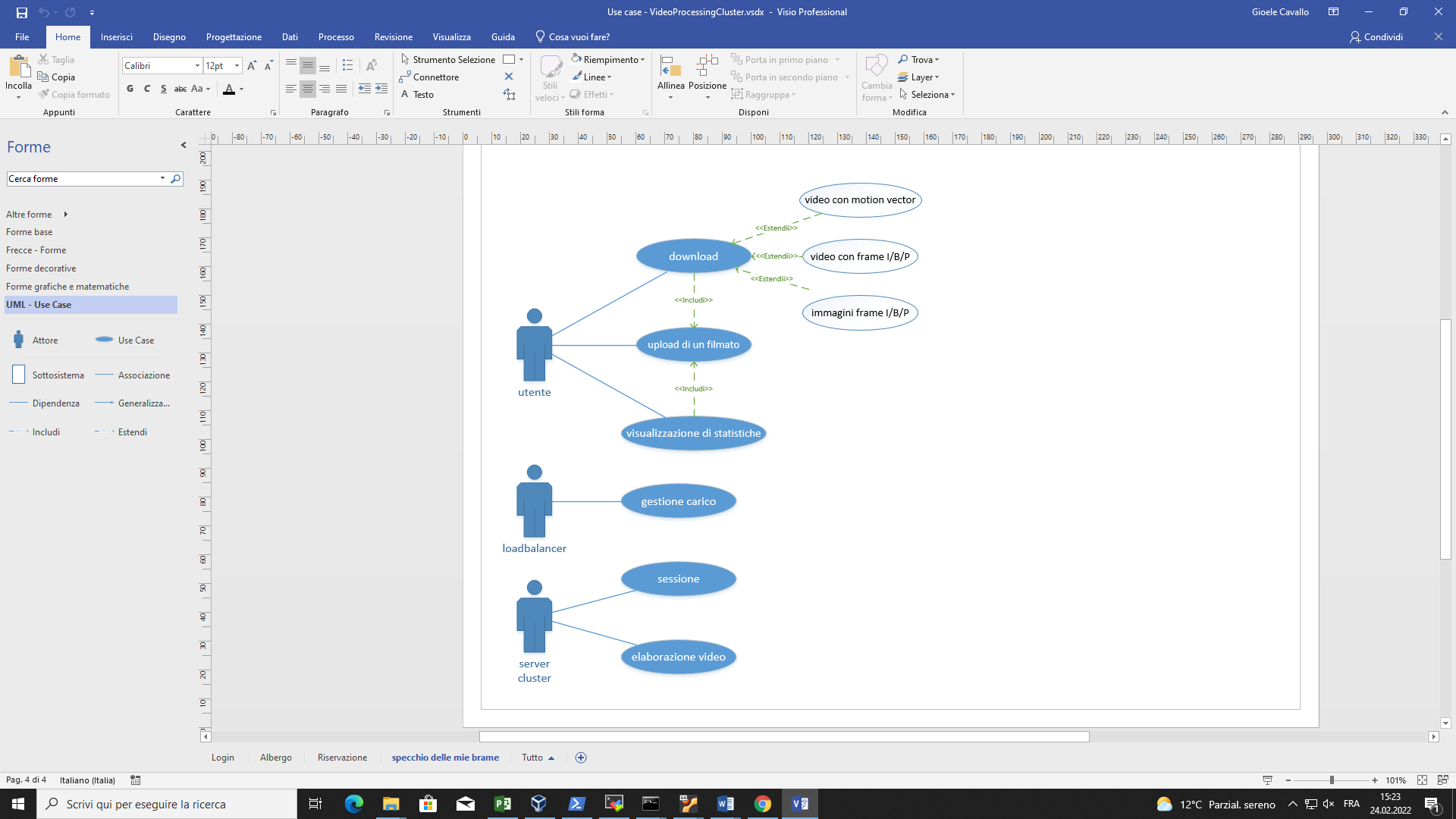
**Versione**: indica la versione del requisito. Ogni modifica del requisito avrà una versione aggiornata.

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.

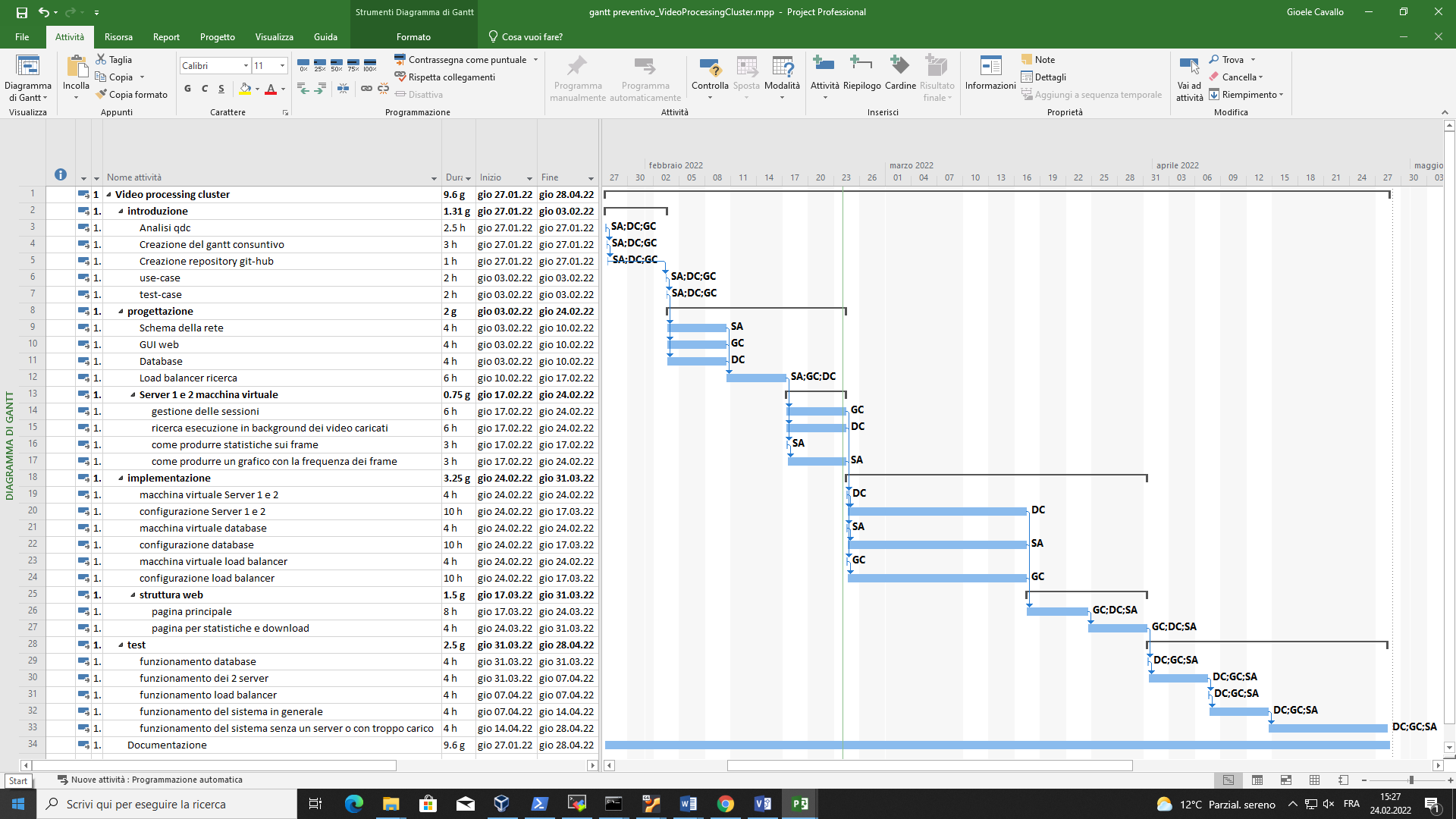
## Use case



All’interno dell’applicativo abbiamo inserito 3 tipi diversi di attori, più specificatamente:

* Utente: questi attori potranno interagendo con la pagina web servita dal server: caricare un filmato, vedere una pagina web con le relative statistiche e scaricare un video composto dai motion vector, un video con i frame I/B/P e immagini con i frame I/B/P.
* Load balancer: questo attore è singolo e il suo compito è quello di gestire il carico, indirizzando le richieste aduno dei due server in base a dei criteri definiti.
* Server cluster: questi attori sono due e lavorano in cluster ovvero condividono delle risorse, come la gestione delle sessioni e l’elaborazione dei video.

## Pianificazione



Le annotazioni a fianco di ogni attività sono i nomi delle persone le quali devono occuparsi di tale: GC = Gioele Cavallo, DC = Damian Campesi,   
SA = Samuele Abbà

.

## Analisi dei mezzi

Per questo progetto mi sarà fornito 1 PC della scuola, sul quale ci sarà un sistema operativo Windows10Pro.

### Software

I software utilizzati per questo progetto sono Virtualbox per la creazione delle macchine virtuali Linux, HTML, CSS, PHP e apache per il sito applicativo, Memcached per mettere i due server in cluster e infine Nginx per il load balancer.

### Hardware

L’hardware coinvolto nel progetto è un PC fornito dalla scuola, esso ha ottime prestazioni.

Per questo progetto utilizzeremo inoltre una serie di supporti esterni dove salvare il nostro lavoro come: una serie di SSD e chiavette.

# Progettazione

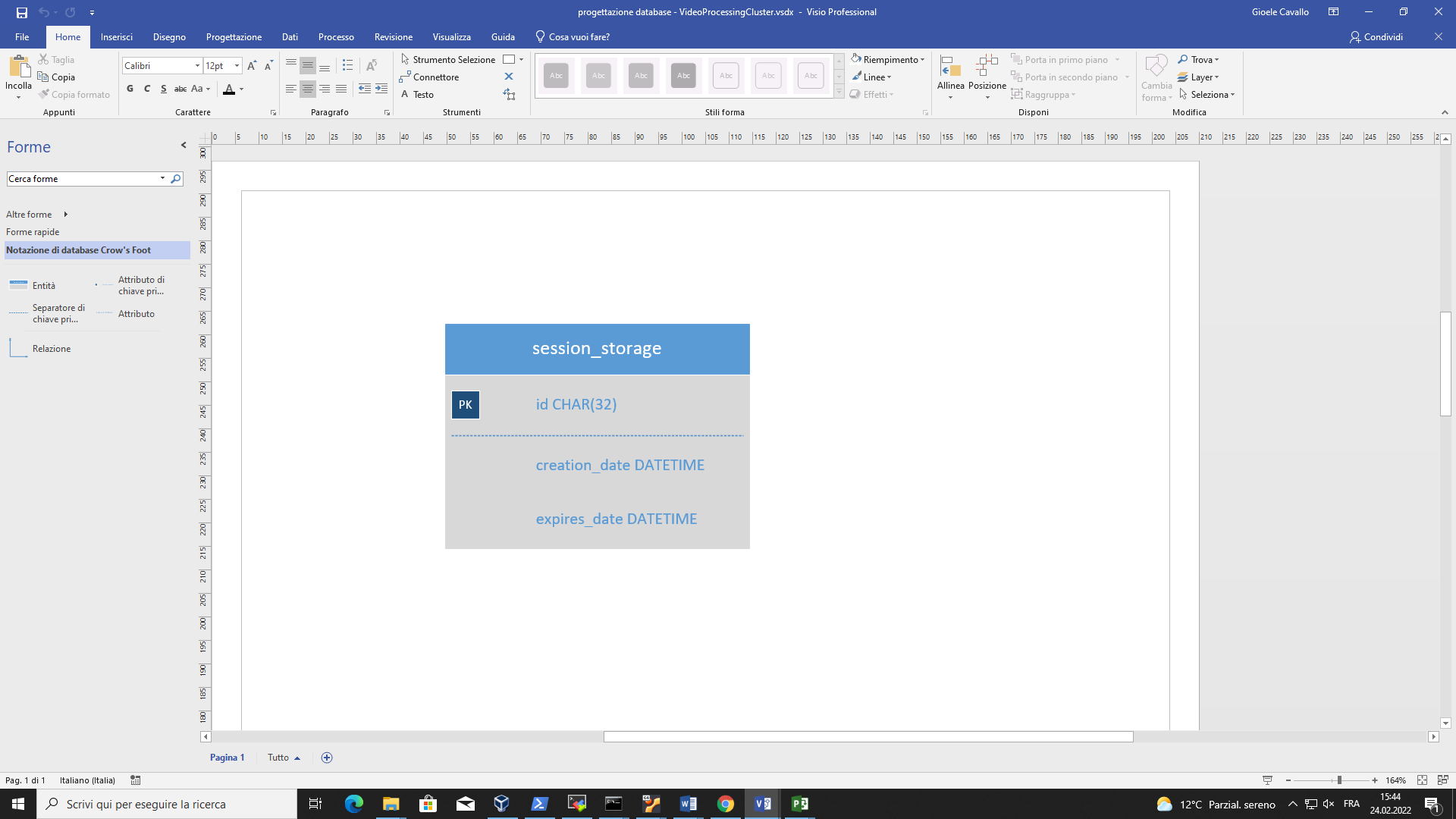
Questo capitolo descrive esaustivamente come deve essere realizzato il prodotto fin nei suoi dettagli. Una buona progettazione permette all’esecutore di evitare fraintendimenti e imprecisioni nell’implementazione del prodotto.

## 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



Questa à l’architettura del database per questo progetto. È costituito da un'unica tabella la quale ci permette di immagazzinare un id, usato per memorizzare l’utente che si connette così da mostrare solo a lui le statistiche sul video caricato. Poi ci sono altri 2 attributi: **creation\_date** che è la data nel quale è stata creata l’istanza ed **expires\_date** che è la data nella quale verrà eliminata l’istanza.

## 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 Case:**  **Riferimento**: | TC-01  REQ-01 | **Nome:** | Funzionamento load balancer basato sul carico |
| **Descrizione:** | Verificare il corretto funzionamento del load balancer | | |
| **Prerequisiti:** | Avere il cluster di server  Avere il load balancer configurato | | |
| **Procedura:** | 1. Saturare la potenza di calcolo del primo server 2. Collegarsi al load balancer 3. Verificare che si è collegati al secondo server 4. Ricollegarsi al load balancer 5. Verificare che si è collegati nuovamente al secondo server | | |
| **Risultati attesi:** | Si aspetta che la procedura venga soddisfatta e che alla fine si è collegati al secondo server ovvero quello con meno carico. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-02  REQ-02 | **Nome:** | Funzionamento delle sessioni |
| **Descrizione:** | Verificare il corretto funzionamento della gestione delle sessioni | | |
| **Prerequisiti:** | Avere il cluster di server | | |
| **Procedura:** | 1. Collegarsi al primo server 2. Iniziare un’iterazione 3. Collegarsi al secondo server 4. Verificare che siano apparse le statistiche sul file caricato in precedenza | | |
| **Risultati attesi:** | Si aspetta che le statistiche visualizzate siano riguardanti quelle del file caricato. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-03  REQ-03 | **Nome:** | Funzionamento elaborazione in background dei video caricati |
| **Descrizione:** | Verificare il funzionamento dell’elaborazione in background | | |
| **Prerequisiti:** | Avere un server funzionante | | |
| **Procedura:** | 1. Collegarsi al server 2. Caricare un file nella GUI 3. Far avviare i processi 4. Chiudere la GUI 5. Riaprire la GUI dopo un po’ di tempo 6. Verificare che i processi siano continuati | | |
| **Risultati attesi:** | Si aspetta che vengano visualizzate le statistiche una volta riaperto il browser. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-04  REQ-04 | **Nome:** | Funzionamento upload di un filmato |
| **Descrizione:** | Verificare il funzionamento dell’upload di un filmato di massimo 500MB | | |
| **Prerequisiti:** | Avere un server funzionante | | |
| **Procedura:** | 1. Collegarsi al server 2. Caricare un file nella GUI di dimensioni maggiori di 500MB 3. Verificare che venga mostrato un errore concernente la dimensione troppo grande del filmato caricato. 4. Ricaricare un file nella GUI di dimensioni minori di 500MB 5. Verificare che non vengano dati errori | | |
| **Risultati attesi:** | Si aspetta che inizialmente le procedure del server non vengano eseguite, mentre dopo il secondo upload esse vengano eseguite e appaiano le statistiche. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-05  REQ-05 | **Nome:** | Produrre statistiche sul video |
| **Descrizione:** | Verificare che vengano create le statistiche sul video | | |
| **Prerequisiti:** | Avere un server funzionante | | |
| **Procedura:** | 1. Collegarsi al server 2. Caricare un file nella GUI 3. Verificare che vengano mostrate le statistiche inerenti ad esso | | |
| **Risultati attesi:** | Si aspetta che vengano visualizzate le statistiche. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-06  REQ-06 | **Nome:** | Possibilità di scaricare differenti contenuti |
| **Descrizione:** | Verificare che ci sia la possibilità di scaricare i file nelle diverse modalità | | |
| **Prerequisiti:** | Avere un server funzionante | | |
| **Procedura:** | 1. Collegarsi al server 2. Caricare un file nella GUI 3. Verificare che vengano creati i file seguenti:    1. Video con motion vector    2. Video con frame I/B/P    3. Immagini con tutti i frame I/B/P | | |
| **Risultati attesi:** | Si aspetta che vengano creati correttamente i file sopra citati. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-07  REQ-07 | **Nome:** | Cancellazione dati sul db e sul server |
| **Descrizione:** | Verificare che i dati sul db e sul server vengano cancellati dopo 1 ora | | |
| **Prerequisiti:** | Avere un server funzionante  Avere un db funzionante | | |
| **Procedura:** | 1. Collegarsi al server 2. Caricare un file 3. Attendere 1 ora 4. Verificare che si sia svuotata la cache del server 5. Collegarsi al db 6. Verificare che si sia svuotato | | |
| **Risultati attesi:** | Si aspetta che i dati più vecchi di 1 ora non siano più memorizzati. | | |

## 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.

# 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
* …