

Guest Data Logger

Titolo del progetto: Documentazione Guest Data Logger
Alunno/a: Gioele Cavallo
Classe: Info 4AC
Anno scolastico: 2022/2023
Docente responsabile: Luca Muggiasca

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	7
2.4	Pianificazione	8
2.5	Analisi dei mezzi.....	9
2.5.1	Software	9
2.5.2	Hardware.....	9
3	Progettazione	10
3.1	Design dell'architettura del sistema	10
3.2	Design dei dati e database.....	11
3.2.1	UTENTE.....	11
3.3	Design delle interfacce	11
3.3.1	Interfaccia grafico	11
3.3.2	Interfaccia Login.....	12
3.3.3	Interfaccia Impostazioni admin	12
3.4	Design procedurale	13
4	Implementazione	14
4.1	Applicazione web	14
4.1.1	Controllers.....	14
4.2	Applicazione python	16
4.2.1	args_parser.....	16
4.2.2	check_args.....	16
4.2.3	test_input.....	17
4.2.4	detect_from_image	17
4.2.5	get_boxes.....	19
4.2.6	print_in_file.....	19
4.2.7	get_input	19
4.2.8	draw_detection.....	20
4.2.9	start_detection	21
5	Test.....	22
5.1	Protocollo di test.....	22
5.2	Risultati test.....	25
5.3	Mancanze/limitazioni conosciute.....	26
6	Consuntivo.....	27
7	Conclusioni	29
7.1	Sviluppi futuri.....	29
7.2	Considerazioni personali	29
8	Bibliografia	30
8.1	Sitografia	30
8.2	Glossario	31
8.3	Indice delle figure	31
9	Allegati.....	31

1 Introduzione

1.1 Informazioni sul progetto

Questo progetto è supervisionato dal docente Luca Muggiasca e sviluppato dall'allievo Gioele Cavallo. Il progetto è svolto nella scuola SAMT di Canobbio nelle ore adibite per i progetti dal 29.08.2022 aa 07.12.2022.

1.2 Abstract

Situazione iniziale

Mi è stato chiesto di realizzare un programma che dato come input una videocamera, riuscisse a rilevare le persone e a contarle. I risultati devono essere poi salvati in un file così da poter essere letti e visualizzati tramite un grafico all'interno di un sito web.

Il sito web dispone di un'area riservata agli admin che possono cambiare i parametri del grafico: file dei dati, intervalli ai quali prendere i dati, tempo minimo e massimo nel quale considerare i dati.

Attuazione

Per questo progetto sono necessarie conoscenze della programmazione web per il sito e bisogna saper utilizzare delle librerie già esistenti per la rivelazione delle persone.

Il sito web deve essere dinamico, quindi bisogna creare un'api che ritorna i dati necessari per poterli aggiornare senza ricaricare l'intera pagina.

Risultati

Questo progetto è realizzato con il framework in php Laravel, per il sito web. Per l'applicazione che rivela le persone è stata utilizzata la libreria OpenCV in python.

1.3 Scopo

- Didattici:
 - Saper creare una progettazione e rispettarla con le dovute tempistiche.
 - Saper documentare il lavoro, redigere un report giornaliero.
- Operativi:
 - Saper utilizzare delle librerie.
 - Incrementare la conoscenza sulla creazione dei siti web e dei framework più utilizzati.

Guardare il QdC in allegato.

2 Analisi

2.1 Analisi del dominio

Questo progetto inizia da zero. Mi è stato assegnato il compito di portare a termine lo sviluppo di questo progetto che permetterà il rilevamento fisico, rilevamento emotivo e conteggio delle persone dentro ad uno stand. I dati raccolti devono poi poter essere consultabili tramite un grafico in una pagina web.

I dati vengono salvati in un file nella cartella **public\data** del sito MVC.

Il programma di rilevamento parte da linea di comando, in quanto non serve un'interfaccia per la gestione dell'avvio e blocco dell'esecuzione.

Le conoscenze consigliate per questo progetto sono: basi di python, buon livello JavaScript, conoscenza base di Laravel.

2.2 Analisi e specifica dei requisiti

ID: REQ-01	
Nome	Rilevamento persone
Priorità	1
Versione	1.0
Note	L'applicativo deve distinguere le persone.

ID: REQ-02	
Nome	Contare le persone
Priorità	1
Versione	1.0
Note	L'applicativo deve contare le persone.
Sotto requisiti	
001	Bisogna rilevare le persone

ID: REQ-03	
Nome	Statistiche presenza persone
Priorità	1
Versione	1.0
Note	Devono essere create delle statistiche sulle persone rilevate.
Sotto requisiti	
001	Bisogna poter contare le persone

ID: REQ-04	
Nome	Periodo statistiche
Priorità	1
Versione	1.0
Note	Bisogna poter visualizzare le statistiche in periodi o in giornate
Sotto requisiti	
001	Bisogna poter creare le statistiche

ID: REQ-05	
Nome	Funzionamento in assenza di internet
Priorità	1
Versione	1.0
Note	Il sistema deve funzionare anche senza connessione ad internet

ID: REQ-06	
Nome	Funzionamento login admin
Priorità	1
Versione	1.0
Note	Bisogna possedere una schermata di login per gli admin funzionante.

ID: REQ-07	
Nome	Presenza pagina amministrativa
Priorità	1
Versione	1.0
Note	Bisogna possedere una pagina per gli admin, la quale permette di modificare e parametrizzare le funzionalità dell'applicativo e dei grafici.
Sotto requisiti	
001	Bisogna avere il login funzionante

ID: REQ-08	
Nome	Informazioni sulle persone
Priorità	2
Versione	1.0
Note	Bisogna poter visualizzare le informazioni delle persone sul video (età, sesso, stato d'animo, ...)
001	Bisogna rilevare le persone
Sotto requisiti	
001	Bisogna rilevare le persone

2.3 Use case

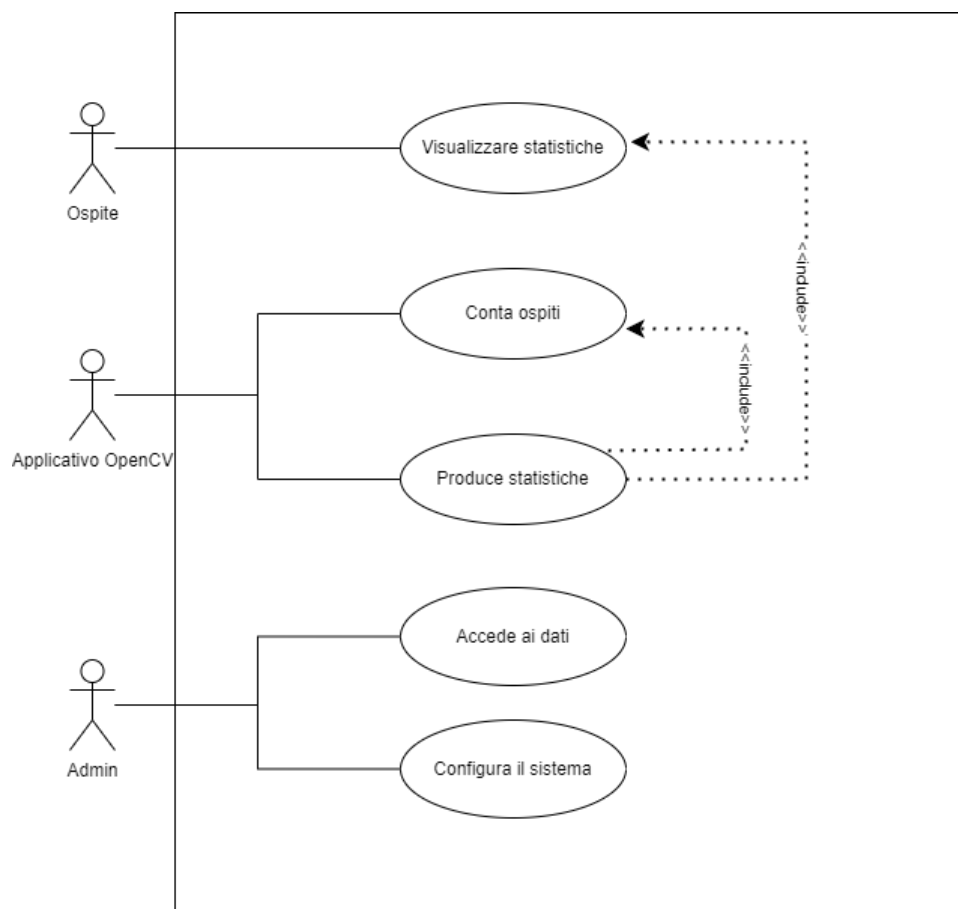


Figura 1 Use case progetto

Per questo progetto sono stati definiti 3 attori:

- Ospite: questo attore è il più statico di tutti, in quanto può solo visualizzare le statistiche tramite il sito web.
- Applicativo OpenCV: questo attore è il cuore del progetto. Produce le statistiche riguardanti il rilevamento degli ospiti.
- Admin: questo attore può accedere ai dati e configurare i parametri per la visualizzazione del grafico.

2.4 Pianificazione

Di seguito è rappresentata la pianificazione preventiva del progetto. si può notare come la maggior parte del tempo sia dedicato all'implementazione. La metodologia utilizzata per la pianificazione è quella Waterfall, a motivo della facilità nell'implementarla.

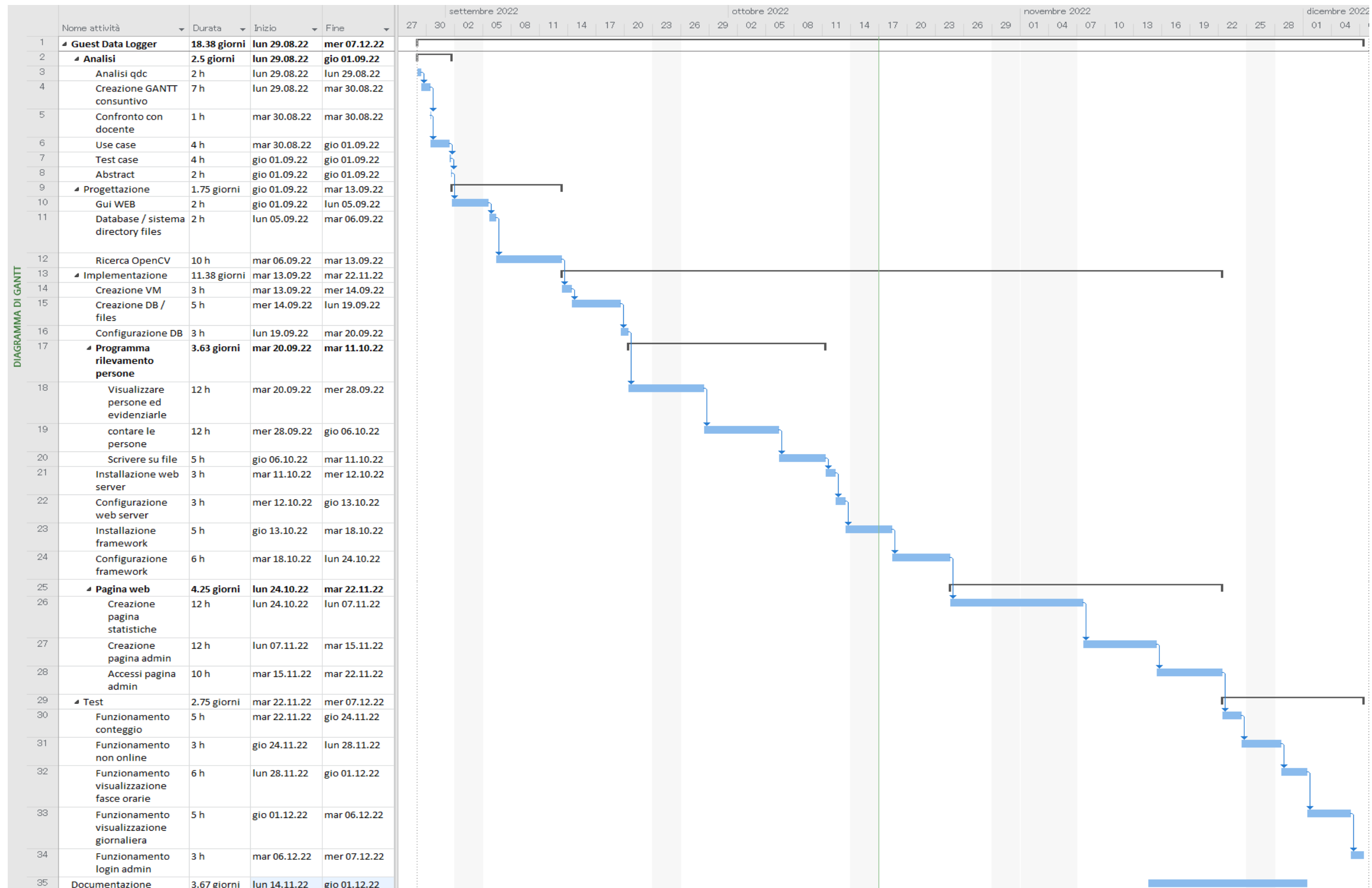


Figura 2 GANTT preventivo

2.5 Analisi dei mezzi

Per questo progetto mi è stato fornito un computer con Windows 10 Enterprise.

2.5.1 Software

- OpenCv – 4.6.0
- Python3
- HTML, CSS, PHP, JS
- Chart.js - 3.5
- Laravel

2.5.2 Hardware

- HDD per memorizzare le VM
- un Computer con:
 - Intel Xeon CPU E3-1240
 - RAM 16 GB
- Una camera LOGITECH C920 HD Pro

3 Progettazione

3.1 Design dell'architettura del sistema

Per questo progetto ho deciso di utilizzare la struttura MVC (Model View Controller).

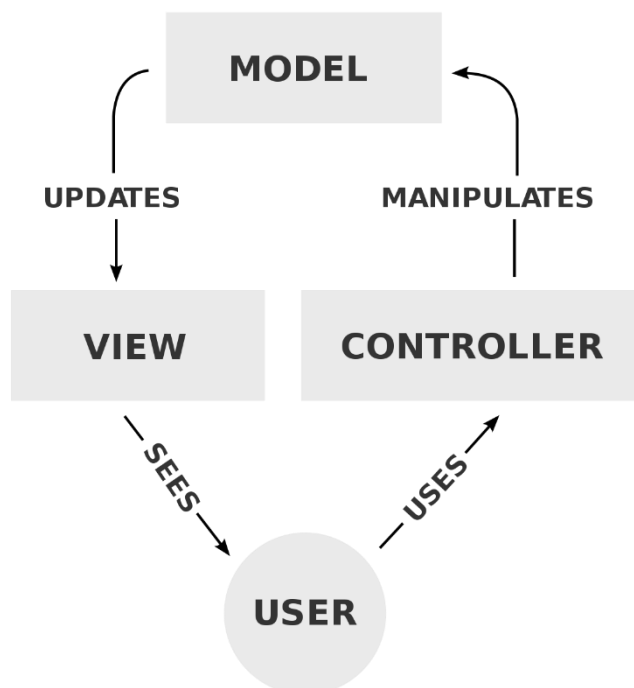


Figura 3 flusso funzionamento MVC

L'utente manda delle richieste http ai *controllers*, che possono essere di qualsiasi tipologia (GET, POST, HEAD, ...). Le azioni dell'utente vengono interpretate poi dai *models* che posseggono le funzioni per eseguire le operazioni.

Una volta aver interpretato le richieste dell'utente, viene ritornata una view che è la parte grafica visualizzabile dall'utente.

Ho usato il framework Laravel che usa un MVC proprio con all'interno le configurazioni necessarie per la sicurezza, database, richieste, routing, ed altro.

3.2 Design dei dati e database

Questo progetto possiede un database con un'unica tabella per gli utenti.

3.2.1 UTENTE

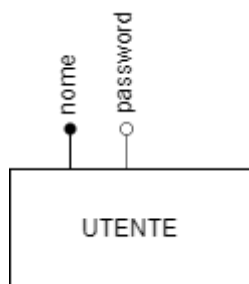


Figura 4 tabella utente

La tabella utente ha due attributi: nome e password. Il nome serve per identificare gli utenti, e la password per autenticarli. Per salvare la password viene utilizzato l'algoritmo bcrypt.

3.3 Design delle interfacce

3.3.1 Interfaccia grafico

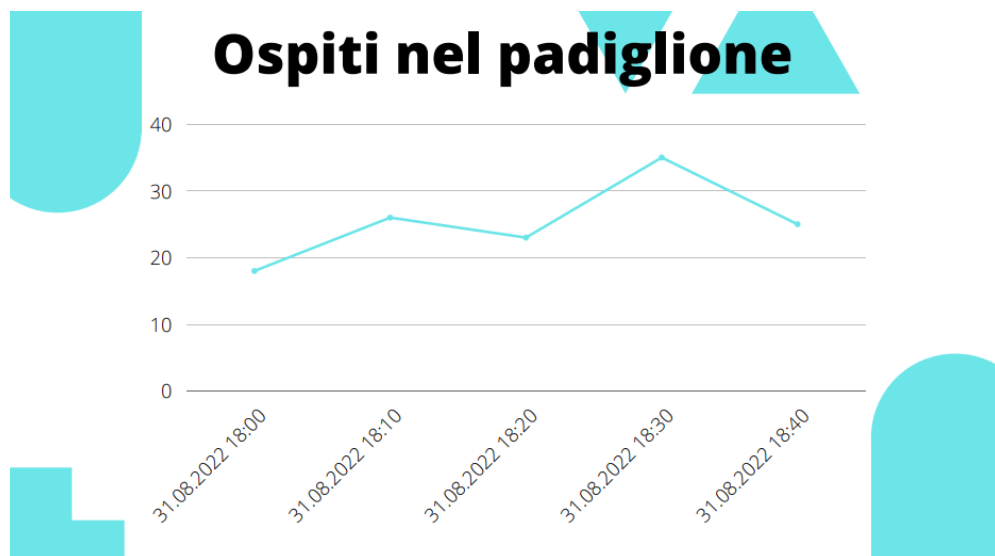


Figura 5 Interfaccia grafico

Questa è la progettazione della pagina web dove verrà visualizzato il grafico. La parte del grafico viene aggiornata dinamicamente.

3.3.2 Interfaccia Login

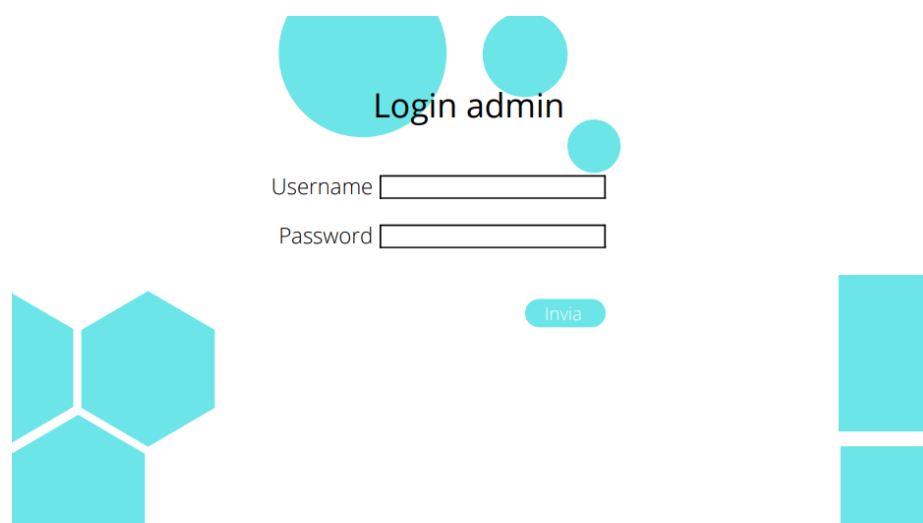


Figura 6 Interfaccia Login

Questa è la progettazione della pagina web del login. Da qui è possibile accedere all'area riservata agli admin.

3.3.3 Interfaccia Impostazioni admin

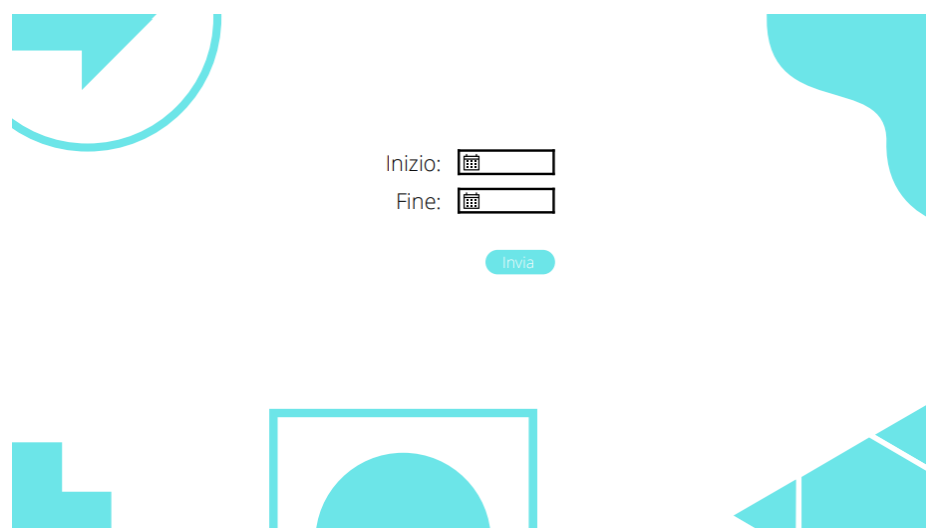


Figura 7 design interfaccia pagina impostazioni

Questa è la progettazione della pagina delle impostazioni dove solo gli admin hanno accesso. Da qui è possibile modificare e parametrizzare la visualizzazione del grafico.

3.4 Design procedurale

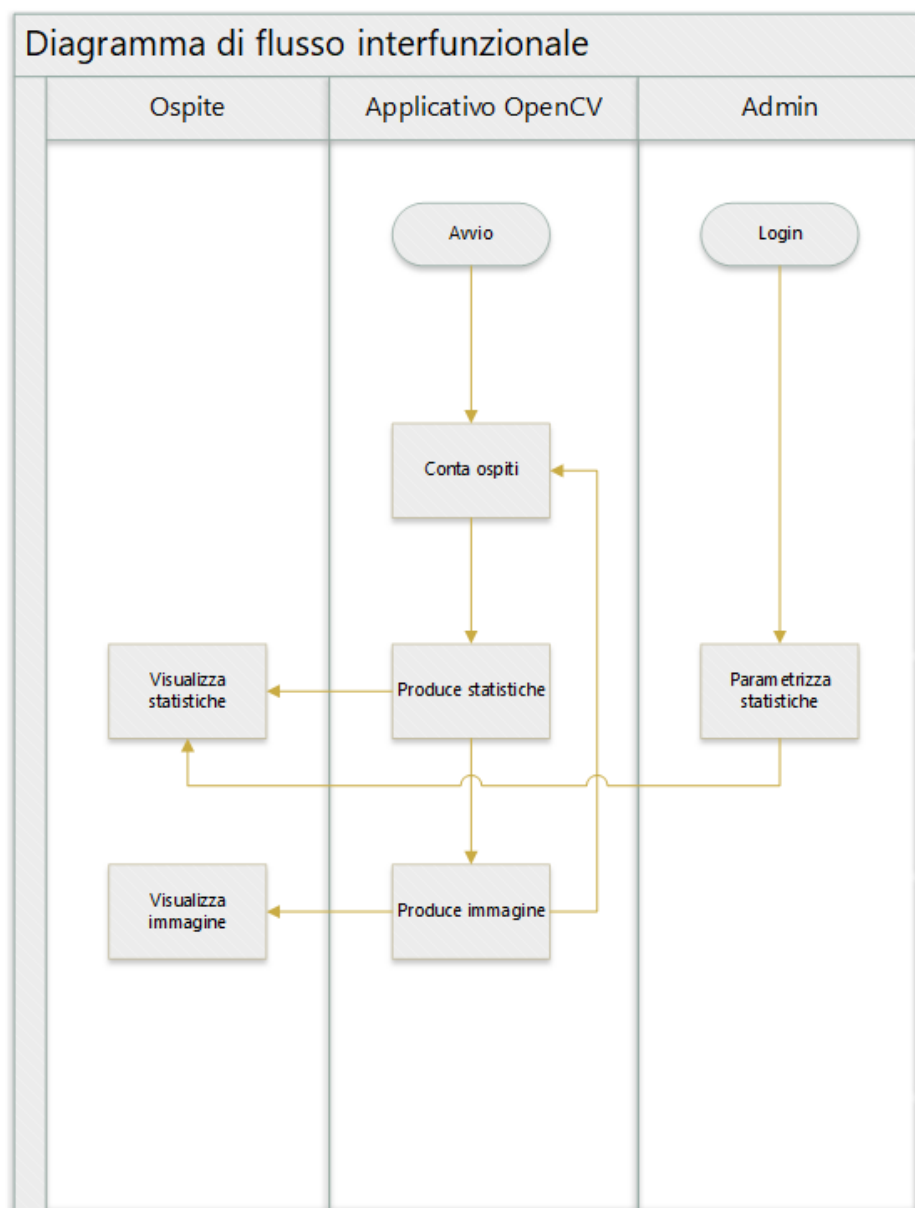


Figura 8 Diagramma di flusso progetto

Questo è il diagramma di flusso *Swimlane* ideato per il progetto con tre attori: ospite, applicativo OpenCV e admin.

Iniziando dall'avvio dell'applicativo OpenCV, vengono contati gli ospiti e su quel conteggio vengono eseguite le statistiche. Una volta prodotte le statistiche si possono visualizzare, e l'applicativo mostrerà l'immagine del video con le persone rilevate. Qui il processo si ripete tornando al conteggio degli ospiti.

Gli ospiti possono visualizzare le statistiche prodotte e anche le immagini prodotte dall'applicativo. Le statistiche e la loro visualizzazione sono parametrizzate dagli admin, una volta che eseguono il login.

4 Implementazione

4.1 Applicazione web

L'applicativo web permette di interfacciarsi con la produzione di statistiche dell'applicativo del rilevamento delle persone. Risponde alle richieste GET e POST per la visualizzazione del grafico e le relative pagine per la gestione.

4.1.1 Controllers

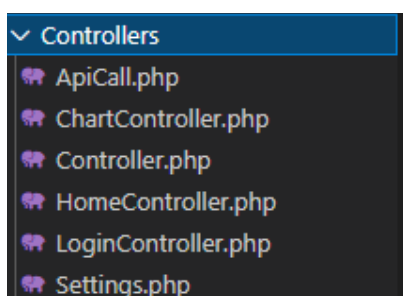


Figura 9 elenco *controllers*

Ci sono sei *controllers* che gestiscono eventi e pagine distinte tra di loro.

4.1.1.1 ApiCall

Questo controller contiene le funzionalità per poter ricevere i dati da visualizzare nella corretta formattazione:

- `getData`: Questa funzione pubblica ritorna i dati presi dal file, settato nel file di configurazione, filtrati e compattati seguendo i parametri inseriti nel file di configurazione.
- `compactData`: Questa funzione privata permette di ricevere i dati passati filtrandoli con il parametro "intervals" e compattandoli con il parametro "pick_up_time".
- `statistics`: Questa funzione privata dato un array, contenente il conteggio delle persone contante come parametro, ritorna la statistica delle persone contante.
- `fetchData`: La funzione privata è viene usata per ritornare i dati grezzi del file nel quale sono salvati.
- `readRawData`: Questa funzione privata ritorna il contenuto del file presente in `public/data` passatogli come parametro.

4.1.1.2 ChartController

Questo controller contiene una sola funzione (`index`) che ritorna la view per la visualizzazione del grafico.

4.1.1.3 HomeController

Questo controller contiene una sola funzione (`index`) che ritorna la view della pagina principale.

4.1.1.4 LoginController

Questo controller contiene 3 metodi inerenti al login degli admin.

- index: Questa funzione pubblica ritorna la view del login e permette quindi la visualizzazione della pagina di login.
- rightLogin: Questa funzione privata permette di verificare che l'utente e la password (non criptata) passati coincidano con un'utente registrato nel database (quindi admin).
- checkLogin: Questa funzione pubblica viene chiamata all'invio dei dati quando si esegue il login. Verifica che i dati siano giusti e permette di visualizzare la pagina degli admin.

4.1.1.5 Settings

Questo controller si occupa di tutto ciò che concerne le impostazioni e parametrizzazione della visualizzazione del grafico.

- writeSettings: Questa funzione pubblica risponde alla richiesta post eseguita quando si salvano le impostazioni.
- sanizeDate: Questa funzione privata si occupa di trasformare le date in un formato riutilizzabile nel programma.
- deleteConf: Questa funzione privata permette di eliminare la configurazione passata come parametro dal file di configurazione.
- writeConf: Questa funzione privata si occupa di sovrascrivere una configurazione esistente.
- GetSettings: Questa funzione pubblica si occupa di ritornare tutte le configurazioni presenti nel file.
- getDirFiles: Questa funzione pubblica si occupa di leggere i file presenti in public/data/ e ritornarne l'elenco.
- getConf: Questa funzione privata ritorna il valore della configurazione passatagli.
- getFilePath: Questa funzione pubblica ritorna la posizione relativa a public/data/ del file da dove vengono presi i dati.
- getIntervals: Questa funzione pubblica ritorna l'intervallo dei dati che vengono presi configurato nel file di configurazione.
- getPickUpTime: Questa funzione pubblica ritorna il tempo per il quale vengono raggruppati i dati.

4.2 Applicazione python

Questa applicazione si occupa di rilevare le persone da un video / camera e visualizzarne il risultato del rilevamento conservando in un file i dati creati.

4.2.1 args_parser

Questa funzione ritorna gli argomenti con il quale è stato eseguito il programma sottoforma di array indicizzato.

```
def args_parser():
    arg_parse = argparse.ArgumentParser()
    arg_parse.add_argument("-o", "--output", default=None, type=str,
        help="Output file for statistics")
    arg_parse.add_argument("-c", "--camera", default=False, help="Set to true for camera")
    arg_parse.add_argument("-v", "--video", default=None, help="The video path")
    args = vars(arg_parse.parse_args())
    return args
```

4.2.2 check_args

Questa funzione controlla la validità degli argomenti passatigli e ne ritorna il risultato del controllo.

```
def check_args(args):
    if args["camera"] == False and not args["video"]:
        print("Set an input file/video with -c True or -v <path>")
        return False
    elif args["camera"] == True and args["video"] != "":
        print("Set only one input file/video")
        return False
    else:
        return True
```


4.2.3 test_input

Questa funzione controlla se l'input del video (file o camera) sia valido, e ne ritorna il risultato del controllo.

```
def test_input(input_detection):
    """
    This function return the existence of the camera.
    """

    if isinstance(input_detection, str):
        if not exists(input_detection):
            print("Given file doesn't exists")
        return exists(input_detection)

    video = cv2.VideoCapture(input_detection, cv2.CAP_DSHOW)
    if video.isOpened():
        return True
    print("Unable to open camera")
    return False
```

4.2.4 detect_from_image

Questa funzione data un'immagine, un modello, dalla quale estrarre le coordinate dei volti delle persone.

La prima parte di codice serve per preparare le variabili necessarie quando si andrà ad iterare i risultati e a rilevare le persone dall'immagine.

Nelle prime righe di codice viene eseguito il ridimensionamento dell'immagine e la trasformazione a blob. Poi viene passato il blob come input per il riconoscimento dei volti.

```
def detect_from_image(image, model, layer_name, personidz=0):
    (H, W) = image.shape[:2]

    blob = cv2.dnn.blobFromImage(image, 1 / 255.0, (416, 416), swapRB=True, crop=False)

    model.setInput(blob)
    layerOutputs = model.forward(layer_name)

    boxes = []

    centroids = []

    confidences = []
```

```

for output in layerOutputs:
    for detection in output:
        scores = detection[5:]

        classID = np.argmax(scores)
        confidence = scores[classID]

        if classID == personidz and confidence > MIN_CONFIDENCE:

            box = detection[0:4] * np.array([W, H, W, H])
            (centerX, centerY, width, height) = box.astype("int")

            x = int(centerX - (width / 2))
            y = int(centerY - (height / 2))

            boxes.append([x, y, int(width), int(height)])
            centroids.append((centerX, centerY))
            confidences.append(float(confidence))

detection_no_boxes = cv2.dnn.NMSBoxes(boxes, confidences, MIN_CONFIDENCE, NMS_THRESHOLD)

results = []
if len(detection_no_boxes) > 0:
    results = get_boxes(detection_no_boxes, boxes, confidences, centroids)

return results

```

In questa parte di codice si iniziano le interazioni ai risultati ritornati dai metodi di CV2. Vengono estratti i dati essenziali quali: confidenza, coordinate faccia, e il centro dei box.

4.2.5 get_boxes

Questa funzione ritorna le coordinate delle facce, la confidenza e il centro dei box.

```
def get_boxes(detection_no_boxes, boxes, confidences, centroids):
    results = []
    for i in detection_no_boxes.flatten():
        (x, y) = (boxes[i][0], boxes[i][1])
        (w, h) = (boxes[i][2], boxes[i][3])

        res = (confidences[i], (x, y, x + w, y + h), centroids[i])
        results.append(res)
    return results
```

4.2.6 print_in_file

Questa funzione scrive nel file passatogli come parametro il contenuto passatogli in modalità *append*.

```
def print_in_file(file, text):
    file_obj = open(file, 'a')
    if(file_obj.writable()):
        file_obj.write(text)
    file_obj.close()
```

4.2.7 get_input

Questa funzione prende un solo input video (nel caso se ne fossero settati di più) e ne ritorna il valore.

```
def get_input(args):

    if args["camera"] == "True" :
        input_detection = 0
    else :
        input_detection = args["video"]

    if test_input(input_detection) == False:
        exit()
```

4.2.8 draw_detection

Questa funzione prende come input il network delle emozioni, il modello delle facce, i nomi dei livelli, i nomi degli oggetti disponibili e l'input del video.

Dopodiché inizia a passare l'immagine ai metodi che ritornano le coordinate delle facce e l'emozioni delle persone. In fine mostra il risultato del rilevamento.

```
image = imutils.resize(image, width=700)
results = detect_from_image(image, model, layer_name,
                             personidz=LABELS.index("person"))

for res in results:
    persons += 1
    emotion_detected = True
    image_copy = Image.fromarray(image)
    image_cropped = image_copy.crop(
        (res[1][0], res[1][1], res[1][2], res[1][3]))
    try:
        face_emotion = emotion(np.array(image_cropped), net_emotion)

        font = cv2.FONT_HERSHEY_SIMPLEX
        org = (res[1][0], res[1][1])
        fontScale = 1
        color = (255, 0, 0)
        thickness = 2

        cv2.putText(image, face_emotion,
                     org, font, fontScale, color, thickness, cv2.LINE_AA)
    except:
        emotion_detected = False
        cv2.rectangle(image, (res[1][0], res[1][1]),
                      (res[1][2], res[1][3]), (0, 255, 0), 2)

    if emotion_detected :
        cv2.rectangle(image, (res[1][0], res[1][1]),
                      res[1][2], res[1][3]), (0, 255, 0), 2)
```

Questa è la parte di codice più importante in draw_detection, dove prima vengono eseguite delle operazioni sull'immagine catturata per rendere più veloce il rilevamento della persona. Una volta eseguite queste alterazioni vengono iterati tutti i risultati.

Per velocizzare il rilevamento delle emozioni poi, viene ritagliata dall'immagine originale la persona trovata e questa immagine viene passata al metodo che ne estrae l'emozione. In questo modo si è sicuri che ciò che viene rilevato è relativo alla persona in questione e non alle altre nell' immagine.

Una volta finite queste operazioni viene aggiunta all'immagine iniziale il riquadro che evidenzia la persona e, se esistente, anche l'emozione.

4.2.9 start_detection

Questa funzione inizia il rilevamento dei volti dal video.

```
def start_detection(args):

    input_detection = get_input(args)

    labelsPath = "coco.names"
    LABELS = open(labelsPath).read().strip().split("\n")

    weights_path = "yolov4-tiny.weights"
    config_path = "yolov4-tiny.cfg"

    model_faces = cv2.dnn.readNetFromDarknet(config_path, weights_path)

    layer_name = model_faces.getLayerNames()
    layer_name = [layer_name[i - 1] for i in
                    model_faces.getUnconnectedOutLayers()]

    model_emotion = 'Model/emotion-ferplus-8.onnx'
    net_emotion = cv2.dnn.readNetFromONNX(model_emotion)

    cap = cv2.VideoCapture(input_detection)
    print(f"input: {input_detection}")

    draw_detection(net_emotion, model_faces, layer_name, LABELS, cap)
```

In questa funzione vengono istanziati gli oggetti utili durante il rilevamento, come i modelli e le reti neurali già allenate.

5 Test

5.1 Protocollo di test

Test Case:	TC-001	Nome:	Rilevamento persone
Riferimento:	REQ-001		
Descrizione:	L'applicativo deve riuscire a rilevare le persone presenti nel video della webcam.		
Prerequisiti:	<ul style="list-style-type: none"> Bisogna avere la telecamera collegata al computer e funzionante. 		
Procedura:	<ol style="list-style-type: none"> Attivare il virtual environment con il comando ". \env\Scripts\activate". Una volta entrati nel <i>virtual enviroment</i> far partire il programma con il comando "python final.py -c True" 		
Risultati attesi:	Quando si entra nel campo visivo, nel video bisogna vedere un quadrato che rileva la persona. Quando si esce dal campo visivo invece non deve essere più visibile il quadrato.		

Test Case:	TC-002	Nome:	Conteggio persone
Riferimento:	REQ-002		
Descrizione:	L'applicativo deve riuscire a contare le persone presenti nel video della webcam.		
Prerequisiti:	<ul style="list-style-type: none"> Bisogna avere la telecamera collegata al computer e funzionante. L'applicativo deve essere funzionante e deve rilevare le persone. 		
Procedura:	<ol style="list-style-type: none"> Attivare il virtual environment con il comando ". \env\Scripts\activate". Una volta entrati nel virtual enviroment far partire il programma con il comando "python final.py -c True" 		
Risultati attesi:	Quando si entra nel campo visivo il conteggio deve incrementare, mentre quando si esce il conteggio deve decrementare.		

Test Case:	TC-003	Nome:	Statistiche sulle persone
Riferimento:	REQ-003		
Descrizione:	Deve esserci una pagina web consultabile per la visualizzazione delle statistiche.		
Prerequisiti:	<ul style="list-style-type: none"> Il programma di riconoscimento delle persone deve funzionare integralmente. Bisogna avere un web server funzionante e operativo. 		
Procedura:	<ol style="list-style-type: none"> Entrare all'interno del sito web tramite "localhost:8000/Chart". 		
Risultati attesi:	Bisogna visualizzare un grafico inerente ai dati selezionati nella sezione degli admin.		

Test Case:	TC-004	Nome:	Visualizzazione grafico
Riferimento:	REQ-04		
Descrizione:	Le statistiche devono poter essere visualizzate in giornate o in periodi di tempo.		
Prerequisiti:	<ul style="list-style-type: none"> • L'applicativo deve rilevare le persone e contarle • Bisogna poter visualizzare il grafico 		
Procedura:	<ol style="list-style-type: none"> 1. Utilizzare un set di dati conosciuto 2. Cambiare la visualizzazione da giornata a periodo 3. Controllare che il grafico sia pertinente ai dati 		
Risultati attesi:	I dati conosciuti devono essere visualizzati correttamente in tutti e due i modi (periodo e giornaliero).		

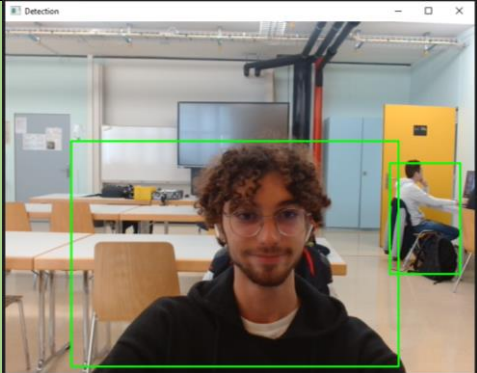
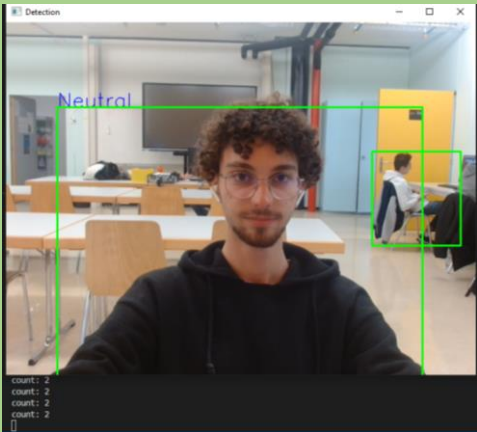
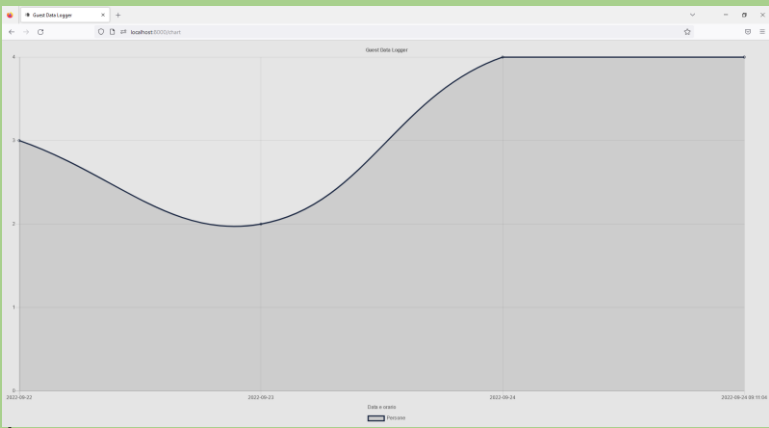
Test Case:	TC-005	Nome:	Funzionamento in assenza di internet
Riferimento:	REQ-05		
Descrizione:	l'intero sistema deve funzionare senza la connessione internet.		
Prerequisiti:	<ul style="list-style-type: none"> • Il sistema deve funzionare completamente 		
Procedura:	<ol style="list-style-type: none"> 1. Staccare il computer / VM da internet 2. Testare tutte le funzionalità 		
Risultati attesi:	Dopo aver staccato il computer / VM da internet ed eseguito tutte le operazioni consentite non devono essere riscontrati problemi o malfunzioni.		

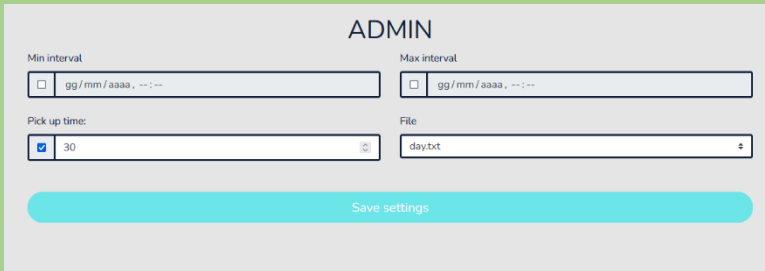
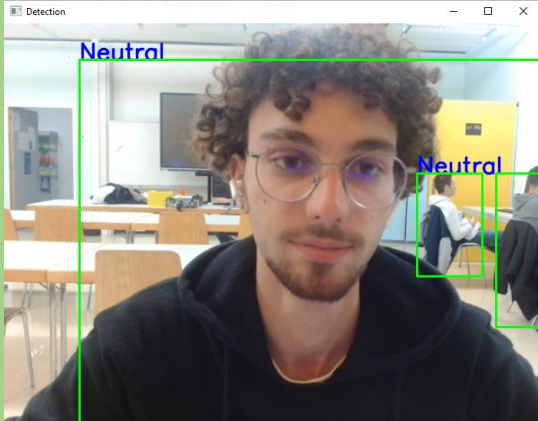
Test Case:	TC-006	Nome:	Login per gli admin
Riferimento:	REQ-06		
Descrizione:	La pagina di login deve funzionare correttamente.		
Prerequisiti:	<ul style="list-style-type: none"> • Il database con gli utenti deve essere funzionante 		
Procedura:	<ol style="list-style-type: none"> 1. Accedere alla pagina di login 2. Eseguire il login con un utente valido e con la password corretta 3. Eseguire nuovamente il login con un'utente non valido ma con una password esistente 4. Eseguire il login con un utente valido ed utilizzando una password non valida 		
Risultati attesi:	La prima volta che si esegue il login (con utente e password corretti) bisogna poter accedere alla schermata degli amministratori. Quando invece si usa la combinazione utente-password sbagliata non deve essere possibile accedere alla pagina degli admin.		

Test Case:	TC-007	Nome:	Pagina admin
Riferimento:	REQ-07		
Descrizione:	Bisogna possedere una pagina web per gli admin, dalla quale possono modificare e parametrizzare l'applicativo e i grafici.		
Prerequisiti:	<ul style="list-style-type: none"> Bisogna avere il login funzionante 		
Procedura:	<ol style="list-style-type: none"> Accedere alla pagina amministrativa Cambiare la visualizzazione del grafico e le varie opzioni 		
Risultati attesi:	Controllare che funzioni tutto quanto e che le modifiche vengano applicate.		

Test Case:	TC-008	Nome:	Informazioni sulle persone
Riferimento:	REQ-08		
Descrizione:	Bisogna visualizzare le informazioni delle persone sul video della camera.		
Prerequisiti:	<ul style="list-style-type: none"> Bisogna avere il programma funzionante 		
Procedura:	<ol style="list-style-type: none"> Entrare nel campo visivo della camera 		
Risultati attesi:	Bisogna vedere le informazioni inerenti alla persona entrata nel campo visivo all'interno del video.		

5.2 Risultati test

Test case	Esito	Risultati	Data
TC-001	OK	 <p>Il programma evidenzia le persone in modo corretto.</p>	09.11.2022
TC-002	OK	 <p>Facendo partire il programma sulla console viene mostrato il conteggio in tempo reale.</p>	16.11.2022
TC-003	OK	 <p>È presente una pagina web consultabile per il grafico.</p>	09.11.2022
TC-004	OK	È possibile mostrare il grafico in diverse formattazioni.	09.11.2022
TC-005	OK	Togliendo internet al computer tutto continua ad essere funzionante.	21.11.2022
TC-006	OK	È possibile effettuare il login correttamente.	09.11.2022

TC-007	OK	 <p>È presente una pagina accessibile dagli admin così da modificare i parametri della visualizzazione del grafico.</p>	09.11.2022
TC-008	OK	 <p>Il programma rivela lo stato d'animo della persona</p>	09.11.2022

5.3 Mancanze/limitazioni conosciute

Per questo progetto tutti i requisiti sono stati soddisfatti con buoni risultati nei test, quindi non ci sono limitazioni finali o mancanze di cui parlare.

6 Consuntivo

Di seguito è rappresentata la progettazione consuntiva del progetto.

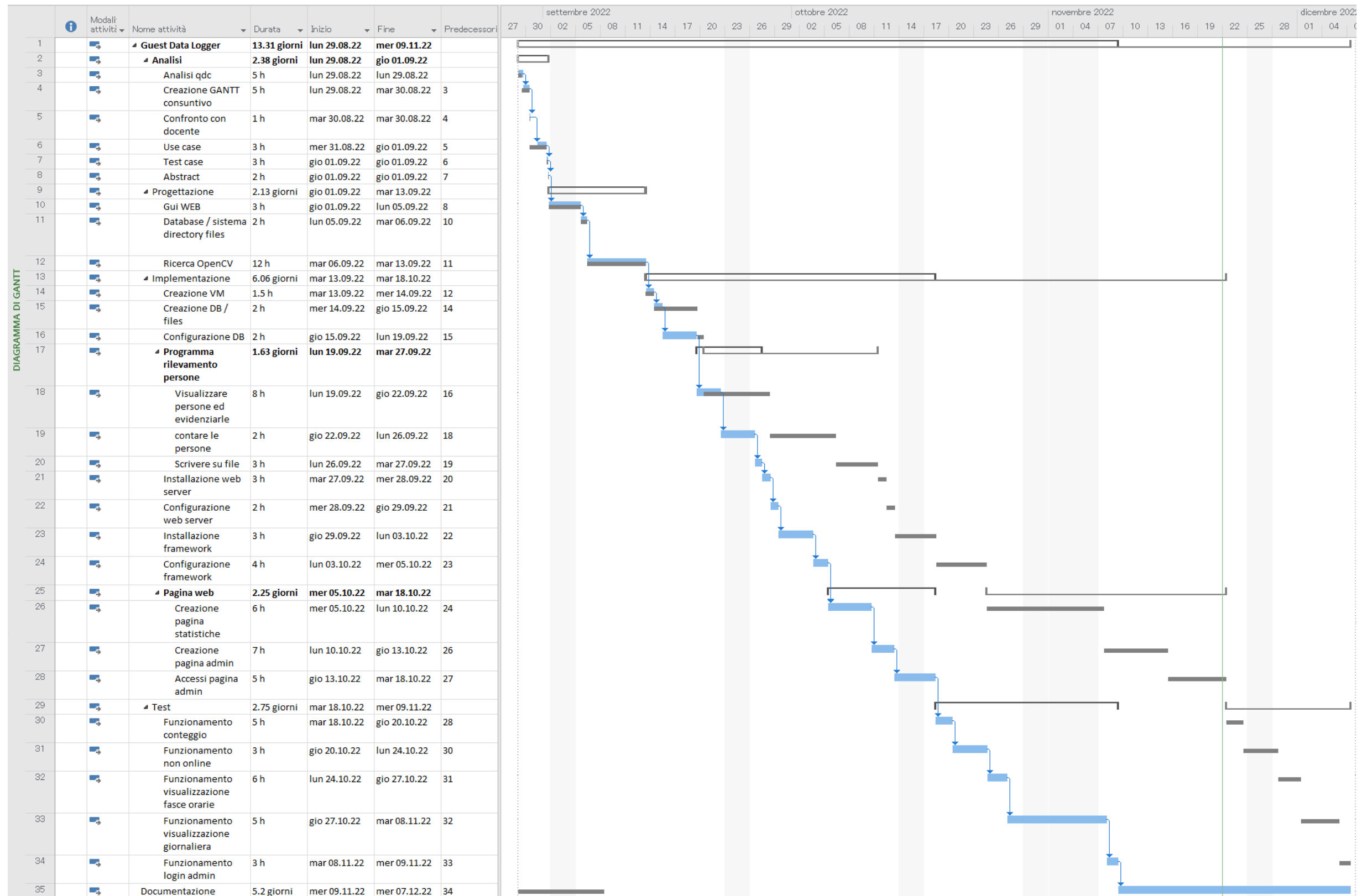


Figura 10 GANTT consuntivo

Come si può notare dal GANTT consuntivo i tempi sono diversi da quelli preventivati. Infatti iniziata l'implementazione le cose sono andate più velocemente del previsto. Ciò ha causato uno squilibrio nella separazione tempistiche dei diversi processi quali: implementazione Test e documentazione (alla quale è stata dedicata più tempo).

Questo è dovuto alla mole di informazioni che sono riuscito a trovare online con solide soluzioni per il mio progetto.

7 Conclusioni

Questo progetto è nato per essere esibito al pubblico in posti come espoprofessioni. Quindi può essere utilizzato come presentazione della scuola ai futuri scolari della CPT informatica.

7.1 Sviluppi futuri

Il progetto ha molti sbocchi, dall'essere come vetrina per la scuola, a l'implementazione per i sistemi di sicurezza o anche per dei giochi (ovviamene con le dovute modifiche). Si potrebbe implementare anche il rilevamento dell'età, sesso e origini della persona.

Per la parte del sito si può migliorare il design e implementare funzionalità, come lo scegliere la finestra temporale direttamente sul grafico rendendolo quindi interattivo oltre che a dinamico.

7.2 Considerazioni personali

Questo progetto mi ha permesso di migliorare le mie capacità nell'utilizzo di laravel e ajax, ma soprattutto mi ha aiutato a migliorare le capacità nella ricerca. La parte di *detection*, che era la maggior parte del progetto, ha richiesto molta ricerca e documentazione sulla libreria OpenCV, alternative e svariati forum. Queste attività mi hanno fatto comprendere quanto sia fondamentale partire con una solida base teorica prima di mettere mano al codice, in questo modo si scrive il codice in prospettiva del risultato finale e teoriche implementazioni future.

8 Bibliografia

8.1 Sitografia

1. <https://pyimagesearch.com/2015/11/16/hog-detectmultiscale-parameters-explained/>, *HOG detectMultiScale parameters explained*, 06-09-2022
2. <https://www.codeproject.com/Articles/5270244/Training-a-Custom-Model-with-OpenCV-and-ImageAI/>, *Training a Custom Model with OpenCV and ImageAI*, 08-09-2022
3. <https://www.codeproject.com/Articles/5270240/Using-Pretrained-Models-to-Detect-People-With-Open/>, *Using Pretrained Models to Detect People With OpenCV and ImageAI*, 12-09-2022
4. <https://stackoverflow.com/questions/36218385/parameters-of-detectmultiscale-in-opencv-using-python/>, *Parameters of detectMultiScale in OpenCV using Python*, 08-09-2022
5. <https://medium.com/analytics-vidhya/person-pedestrian-detection-in-real-time-and-recorded-videos-in-python-windows-and-macos-4c81142f5f59/>, *Person / Pedestrian Detection in Real-Time and Recorded Videos in Python — Windows and macOS*, 14-09-2022
6. <https://data-flair.training/blogs/pedestrian-detection-python-opencv/>, *Real-time Pedestrian Detection using Python & OpenCV*, 15-09-2022
7. <https://bleedai.com/facial-expression-recognition-emotion-recognition-with-opencv/>, *Emotion / Facial Expression Recognition with OpenCV*, 17-10-2022
8. <https://stackoverflow.com/questions/70122322/error-in-installing-dlib-library-in-python-3-10>, *error in installing dlib library in python 3.10*, 24-10-2022
9. <https://www.edlitera.com/blog/posts/emotion-detection-in-images#:~:text=This%20means%20that%20emotion%20recognition,to%20detect%20emotions%20in%20images>, *How to Detect Emotions in Images Using Python*, 27-10-2022
10. <https://sefiks.com/2020/08/25/deep-face-detection-with-opencv-in-python/>, *Deep Face Detection with OpenCV in Python*, 27-10-2022

8.2 Glossario

Termine	Descrizione
Chart.js	Libreria JavaScript per la visualizzazione di grafici e dati.
Diagramma di flusso <i>Swimlane</i>	Tipologia di diagramma di flusso che suddivide i compiti in sotto-processi collegati fra di loro.
Framework	Architettura logica di supporto sulla quale un software è progettato.
GANTT	È un diagramma che aiuta la gestione dei compiti, risorse e tempo di un progetto.
Laravel	Framework open source di tipo MVC scritto in PHP.
MVC	Struttura per creazione di applicazioni orientate ad oggetti.
Network	Struttura informatica che riproduce i comportamenti del cervello umano.
OpenCV	Libreria software per la visione artificiale in tempo reale.
Python	Linguaggio di programmazione orientato ad oggetti.
Virtual enviroment	Strumento in python utile per gestire le dipendenze degli applicativi e permette l'installazione di pacchetti locali in cartelle separate dal sistema.
VM	<i>Virtual Machine</i> computer virtualizzato su un host.

8.3 Indice delle figure

Figura 1 Use case progetto	7
Figura 2 GANTT preventivo	8
Figura 3 flusso funzionamento MVC	10
Figura 4 tabella utente	11
Figura 5 Interfaccia grafico	11
Figura 6 Interfaccia Login	12
Figura 7 design interfaccia pagina impostazioni	12
Figura 8 Diagramma di flusso progetto	13
Figura 9 elenco <i>controllers</i>	14
Figura 10 GANTT consuntivo	27

9 Allegati

Elenco degli allegati:

- Diari di lavoro
- Codici sorgente
- Istruzioni di installazione del prodotto (README.md)
- Manuale per utenti di utilizzo (README.md)
- Qdc