

# Face Detection

## 1. Introduzione

- Informazioni sul progetto
- Abstract
- Scopo

## 2. Analisi

- Analisi del dominio
- Analisi dei mezzi
- Analisi e specifica dei requisiti
- Pianificazione

## 3. Progettazione

- Design dell'architettura del sistema
- Design dei dati e database

## 4. Implementazione

## 5. Test

- Protocollo di test
- Risultati test
- Mancanze/limitazioni conosciute

## 6. Consuntivo

## 7. Conclusioni

- Sviluppi futuri
- Considerazioni personali

## 8. Sitografia

## 9. Allegati

## Introduzione

### Informazioni sul progetto

- **Progetto svolto da:** Gionata Battaglioni, Fabio Gola, Gabriele Dominelli, Lucas Previtali
- **Mandanti del progetto:** Luca Muggiasca
- **Docente Responsabile:** Luca Muggiasca
- **Scuola:** Arti e Mestieri Trevano
- **Sezione:** Informatica
- **Classe:** I3AA • **Data d'inizio:** 10.11.2017 •  
**Termine della consegna:** 19.01.2018 •  
**Consegna per Espoprofessioni:** 02.03.2018

### Abstract

At the beginning teachers in charge showed us the project, how it had to be done and what we would have to do to keep an appropriate organization. At the beginning we had some trouble because we didn't know where to start. The main problem was connect the webcam page at the database. Another problem was learning about the raspberry, how it work and how use it. The result was what I expected since the beginning, our approach worked very well except, as said before, we had to invest maybe too much time consulting with various guides.

### Scopo

Lo scopo del progetto è di creare un sistema di riconoscimento facciale tramite una webcam. Questo sistema serve per quantificare il numero di persone che visitano uno stand di Espoprofessioni. Il sistema deve essere in grado di riconoscere le persone che entrano nel campo visivo della webcam.

## Analisi

### Analisi del dominio

Il progetto completo verrà presentato alla prossima edizione di Espoprofessioni, perciò sarà presentato principalmente a un pubblico di ragazzi e ragazze non per forza appassionati di informatica (in generale il gli "utenti" saranno variegati). Il pubblico presente sarà lì soprattutto per vedere diverse professioni e molti saranno studenti. Per capire il funzionamento del sistema non bisogna essere per forza esperti o appassionati di informatica.

### Analisi e specifica dei requisiti

Il committente richiede una pagina web che effettua un riconoscimento facciale. Una volta che la faccia viene riconosciuta dalla pagina viene salvata all'interno di una variabile e a sua volta salvata in un database. In base al numero di persone riconosciute e al tempo che le persone rimangono ferme davanti alla webcam viene redatto un grafico. Mentre un secondo grafico viene redatto in base al numero di persone che sono state riconosciute dalla webcam e alla fascia oraria.

ID	REQ-001
<b>Nome</b>	Face Detection
<b>Priorità</b>	1
<b>Versione</b>	1.0
<b>Note</b>	
<b>Sub-ID</b>	Requisito
<b>001</b>	Creazione macchina virtuale
<b>002</b>	Creazione pagina Web per webcam
<b>003</b>	Creazione pagina Web per grafici
<b>004</b>	Creazione pagina web per gestione dell'admin.
<b>005</b>	Creare una banca dati
<b>006</b>	Ricerca di nuovi volti
<b>007</b>	Salvataggio delle persone su DB
<b>008</b>	Utilizzo del prodotto su RaspBerry
ID	REQ-002
<b>Nome</b>	Creazione macchina virtuale
<b>Priorità</b>	1
<b>Versione</b>	1.0
<b>Note</b>	
<b>Sub-ID</b>	Requisito ID
<b>001</b>	Si deve creare macchina virtuale su "Virtual Box"
<b>002</b>	Si deve installare di XAMPP v7.0.2
<b>003</b>	Si deve creare la struttura in cartelle che ospita i file del progetto.

ID	REQ-003
<b>Nome</b>	Creazione pagina Web per webcam
<b>Priorità</b>	1
<b>Versione</b>	1.0
<b>Note</b>	
<b>Sub-ID</b>	Requisito
<b>001</b>	Si deve implementare una rappresentazione della webcam

ID	REQ-003
<b>002</b>	Si deve organizzare e realizzare una interfaccia grafica
ID	REQ-004
<b>Nome</b> Creazione pagina Web per grafici	
<b>Priorità1</b> <b>Versione1.0</b>	
<b>Note</b>	
<b>Sub-</b>	Requisito ID
<b>001</b>	Si deve implementare la lettura dal DB
<b>002</b>	Si devono creare 2 grafici: Numero di persone rilevate in ogni ora del giorno. Tempo medio di una persona di fronte all'obiettivo.
<b>003</b>	Si deve poter utilizzare un form di log in per il REQ-005
ID	REQ-005
<b>Nome</b> Creazione pagina Web per gestione dell'admin	
<b>Priorità2</b> <b>Versione1.0</b>	
<b>Note</b>	
<b>Sub-</b>	Miglioria ID
<b>001</b>	Si deve creare un pannello di gestione per le impostazioni della WebCam.
<b>002</b>	Si deve creare un pannello di gestione per le impostazioni di Tracking.
ID	REQ-006
<b>Nome</b> Creare una banca dati	
<b>Priorità1</b> <b>Versione1.0</b>	
<b>Note</b>	
<b>Sub-</b>	Miglioria ID
<b>001</b>	Si deve creare un DB che ospiterà le persone tramite: Tempo di fronte all'obiettivo e ora del tracking.
ID	REQ-007
<b>Nome</b> Ricerca di nuovi volti	
ID	REQ-007

---

**Priorità1 Versione1.0**

**Note**

**Sub- Requisito ID**

- 001** Si deve scegliere la libreria migliore per il tracking dei volti: openCV, tracking.js
- 002** Si deve implementare il codice JavaScript per l'individuazione dei dati tramite libreria.
- 003** La webcam deve eseguire la ricerca di nuovi volti ogni 15 secondi e se rileva dei volti nelle coordinate vicine a quelle vecchie, non ne terrà conto.
- 

---

ID REQ-008

---

**Nome** Salvataggio delle persone sul DB

**Priorità1 Versione1.0**

**Note**

**Sub- Miglioria ID**

- 001** Si deve gestire il codice PHP del sito web per il corretto invio dei dati al DB.
- 

---

ID REQ-009

---

**Nome** Utilizzo del prodotto su RaspBerry

**Priorità1 Versione1.0**

**Note**

**Sub- Requisito ID**

- 001** Si deve allestire un webserver Linux su RaspBerry
- 002** Si deve trasporre l'intero codice del progetto su RaspBerry
- 003** Si deve poter connettere 1 WebCam e 2 schermi contemporaneamente per input e output.
-

## SAM I3 2017/2018 Progetti Face Detection

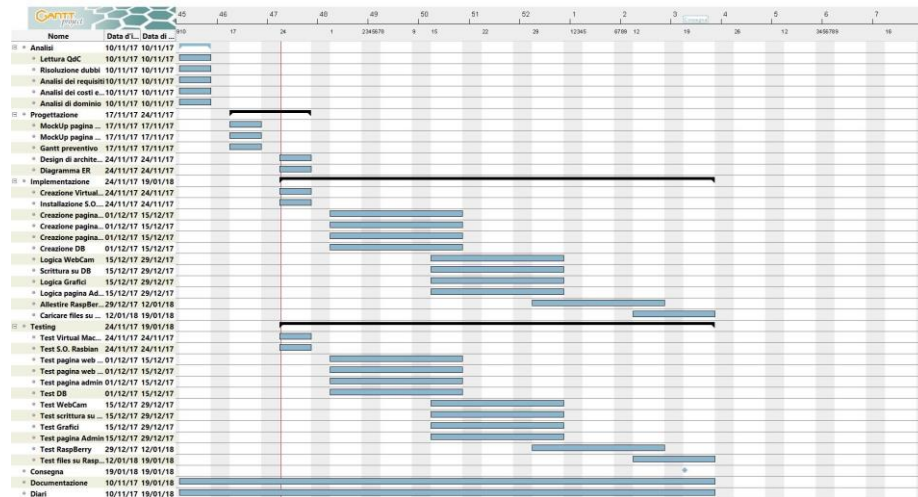


Figure 1: Gantt Preventivo

### Pianificazione

Questo é il Gantt che abbiamo realizzato in base alla lista dei requisiti che abbiamo redatto e al tempo a disposizione.

### Analisi dei mezzi

Come prodotti fisici abbiamo usato i seguenti:

Prodotto	Caratteristiche
Raspberry Pi3	so. raspbian 9.0
tastiera K200	Logitech
Mouse	Lenovo
Monitor HDMI	Asus
Televisore (da definire)	Non usato
Webcam (XXX)	Logitech

Mentre come prodotti virtuali abbiamo usato la libreria tracking.js. Il Raspberry é stato trasformato in un webserver, all'interno abbiamo scaricato le seguenti versioni:

Pacchetto	Versione
Apache	2.4.25
Php	7.0.19
MariaDB	10.1.23

Progetti

6/20

### Analisi dei costi

#### Costo per persona:

È stimato che un apprendista al nostro stesso livello guadagna circa 80 franchi all'ora.

Costo per ora	Ore	persone	Totale
60	48	4	11520 fr.

#### Costo webcam:

Dopo qualche ricerca abbiamo trovato la webcam adatta al nostro progetto e il prezzo è di:

Totale 29  
fr.

#### Costo totale:

Facendo una somma dei vari totali arriviamo al costo totale finale di questo progetto.

Totale  
11560 fr.

## Progettazione

### Design dei dati e database

Il database che abbiamo creato è molto basilico. Presenta due tabelle, la tabella Amministratore dove vengono contenuti i seguenti campi.

(NomeUtente(PK),Id\_WebCam(FK),**Password**,Densità\_Bordo,

Conteggio\_Secondi, Dimensione\_step,Scala\_iniziale) Mentre la

seconda tabella chiamata WebCam che contiene:

(Id\_Webcam(PK),Orario\_inizio,Orario\_fine,**Data**)

Le due tabelle sono collegate tramite una relazione molti a uno chiamata “può avere”.

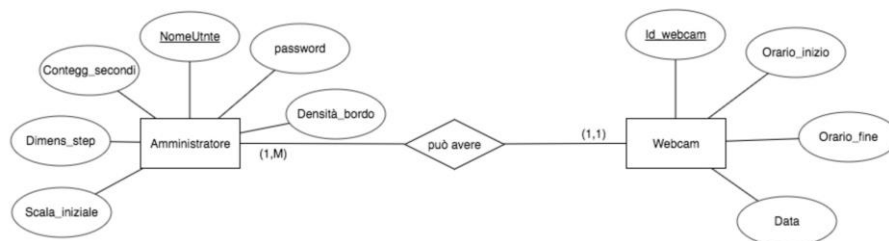


Figure 2: Pagina WebCam  
Progetti

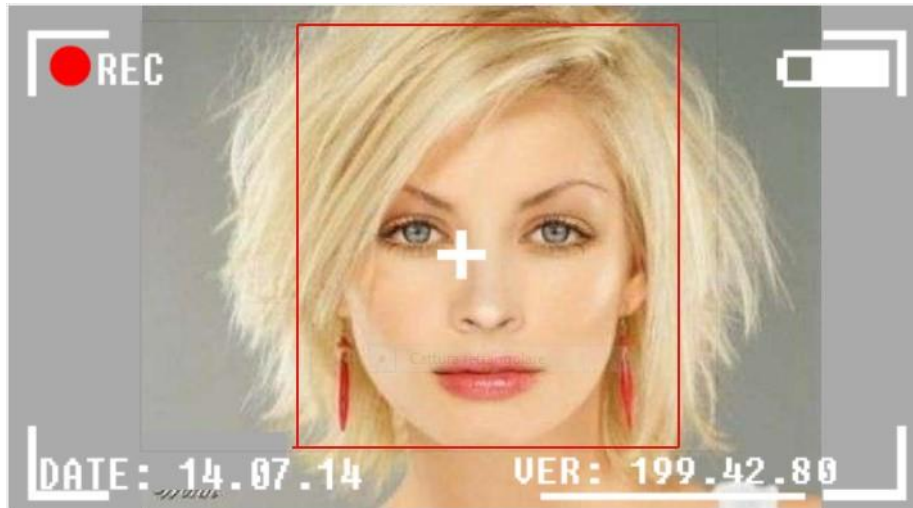


Figure 3: Pagina WebCam

#### Schema E-R, schema logico e descrizione.

Questo é il diagramma ER del database generato per consentire lo scambio dei dati tramite le varie pagine web.

#### Design delle interfacce

Prima di iniziare a scrivere il codice abbiamo scelto assieme al gruppo una struttura base del sito, di come vorremmo che esso diventi. Per la pagina che riguarda la webcam, cioè dove l'utente vede sè stesso abbiamo pensato a un approccio molto minimale.

Mentre per la pagina dove verranno inseriti i grafici abbiamo pensato a un approccio un po' meno minimale ma più adatto alla situazione.

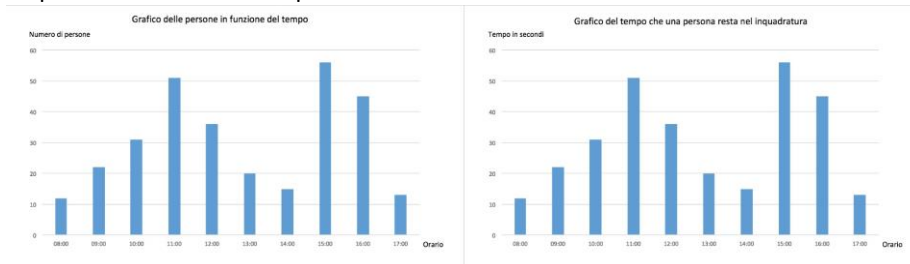


Figure 4: Pagina Grafici

Per la terza e ultima pagina cioè la pagina dove l'amministratore potrà cambiare le impostazioni della webcam o della pagina. Come prima cosa dovrà comparire la pagina di login:



Secondariamente una volta effettuato il login. L' amministratore avrà il diritto di cambiare le impostazioni a suo piacimento.

## Implementazione

### Supporto

Inizialmente abbiamo optato per utilizzare un raspberry dove avremmo caricato tutti i file inerenti al progetto. Così abbiamo recuperato in raspberry e lo abbiamo allestito come webserver. Una volta eseguiti i test su raspberry abbiamo notato che non era abbastanza potente per poter ospitare un progetto come il nostro. Quindi abbiamo spostato tutto su un computer portatile offertoci dalla scuola. Sul computer abbiamo scaricato un programma che lo rende webserver chiamato USB\_SERVER versione 8.5. Una volta fatto ciò abbiamo caricato.

### Creazione pagina WebCam

La pagina della WebCam è la pagina che dirige tutte le operazioni. Prima di tutto abbiamo scelto di utilizzare la libreria di tracking.js per gestire il riconoscimento facciale. Da una piccola struttura di base trovata sul sito principale è ora necessario aggiungere una qualche riga di codice in più necessaria alla corretta registrazione dei volti sul DataBase: 1. Id univoco per ogni persona rilevata dalla WebCam. 2. Orario nel quale una persona viene specchiata. 3. Orario di fine del tracking 4. Giorno in cui è stato eseguito il tracking

**Ognuna di queste informazioni saranno poi fondamentali per la creazione dei grafici di statistica!**

Così è come la pagina mostrala sua forma:

---

Il diagramma illustra la struttura di una pagina di login. Al centro c'è un rettangolo blu con il testo "FOTO" in bianco. Sotto di esso, il testo "Nome utente :" è seguito da un rettangolo blu vuoto per l'input. Ancora sotto, il testo "Password :" è seguito da un rettangolo blu vuoto per l'input.

Figure 5: Pagina Grafici  
Progetti

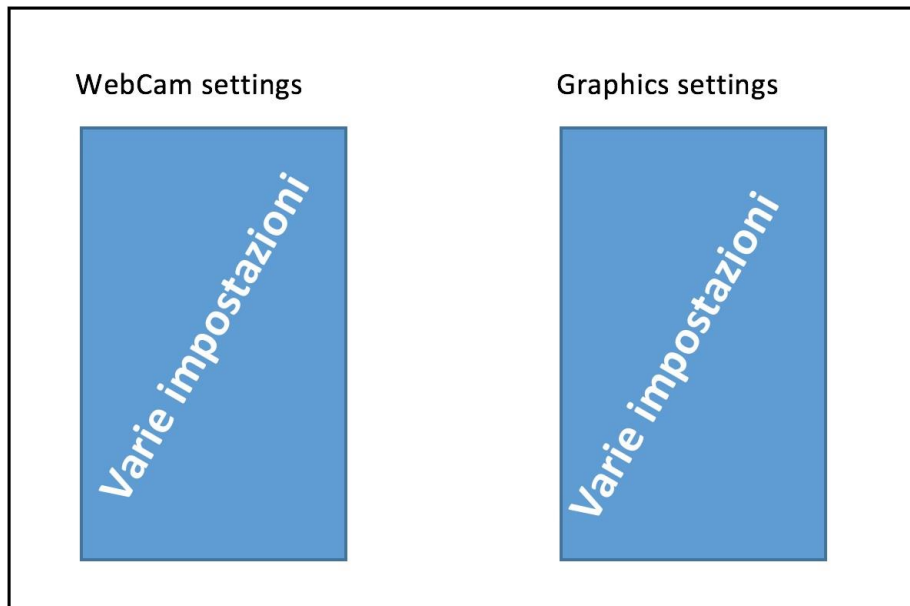


Figure 6: Pagina Grafici



Figure 7: SO WebCam

Con il corretto rilevamento di ogni faccia sullo schermo (contenuto in un oggetto "rect"), viene eseguito un codice JavaScript. Per questa operazione si utilizza un canvas su HTML (qui chiamato "context") che permette di disegnare sullo schermo figure in maniera semplice. Il codice qui riportato permette di definire i colori del rettangolo (bordo e testo), disegnarlo e scrivere del testo di informazioni (come l'ID attribuito alle facce e le coordinate che lo localizzano).

```
context.strokeStyle = 'red';  
context.strokeRect(rect.x, rect.y, rect.width, rect.height); context.font =  
'13px Helvetica'; context.fillStyle = "#fff";  
context.fillText('x: ' + rect.x + 'px', rect.x + rect.width + 5, rect.y + 11); context.fillText('y: ' + rect.y + 'px', rect.x +  
rect.width + 5, rect.y + 22); context.fillText('id: ' + ID, rect.x + rect.width + 5, rect.y + 33);
```

Per ciò che riguarda il passaggio dei dati al database ci siamo dovuti collegare al database tramite php ed abbiamo preso i dati ricevuti dalla pagina e li abbiamo inseriti nell'apposita tabella del database.

```
$sql = "INSERT INTO webcam (Orario_inizio, Orario_fine, Data)
      VALUES (".$Orario_inizio.", " .$.Orario_fine.", " .$.Data.");"; if($conn->query($sql) == FALSE) {
    echo "invio non riuscito!";
  }
```

### Creazione pagina Grafici

La pagina grafici esegue una continua sincronizzazione sul DataBase affinché tutti i dati siano sempre aggiornati. La sua funzione è quella di mostrare 2 grafici: 1. Mostrare la quantità di persone specchiati nella WebCam per ogni fascia oraria. 2. Mostrare il tempo medio di tracking per ogni fascia oraria.

Tramite una ricerca su Internet siamo venuti a conoscenza di una libreria specializzata nel display di grafici non troppo dispendiosa per quanto riguarda le nostre singole conoscenze personali: Chart.js.

Per creare i grafici mi sono dapprima connesso al database ed ho preso ed inserito dentro degli array i dati della data attuale, dell'inizio e della fine di sessione.

```
$sql = "SELECT Orario_inizio, Orario_fine, Data FROM webcam"; $result =
$conn->query($sql);
```

```
$Orario_inizio=array();
```

```
$Orario_fine=array();
```

```
$Data=array(); $arrayIndex = 0;
```

```
if ($result->num_rows > 0) { while($row = $result->fetch_assoc()) {
    array_push($Orario_inizio, $row["Orario_inizio"]);
    array_push($Orario_fine, $row["Orario_fine"]); array_push($Data,
    $row["Data"]);
  }
} else { echo "0 results";
}
```

In seguito ho fatto un counter che permettesse di calcolare quante persone utilizzano l'applicazione in vari fasce d'orario. Inoltre ho utilizzato una tecnica simile anche per calcolare la media della durata di sessione per ogni utente. Dopo aver eseguito tutti i calcoli ed aver inserito i risultati negli array, ho creato i grafici ed inserito i dati. Per inserirli ho utilizzato più volte una tecnica di programmazione che permette l'utilizzo delle variabili presenti nella porzione di codice php e l'utilizzo diretto nella parte javascript, e l'ho inserita all'interno della parte "data" della creazione del grafico.

```
var dataNumeroVisite = {
```

labels: ["09:00", "10:00", "11:00", "12:00", "13:00", "14:00", "15:00", "16:00", "17:00",  
datasets: [{ label: "Numero di visite",



Figure 8: Pagina Admin

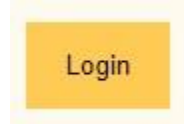


Figure 9: Pulsante Login

```
backgroundColor: "rgba(255,99,132,0.2)", borderColor:
"rgba(255,99,132,1)", borderWidth: 2,
hoverBackgroundColor: "rgba(255,99,132,0.4)",
hoverBorderColor: "rgba(255,99,132,1)",
data: [<?php echo $countUsers[0]; ?>, <?php echo $countUsers[1]; ?>, <?php e
}};
```

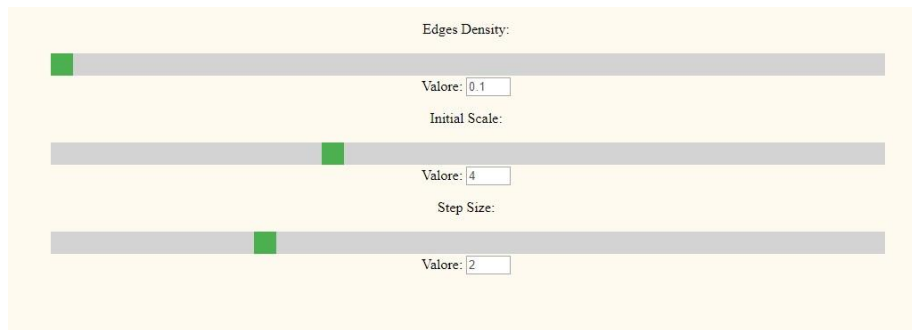
Questo è il risultato ottenuto:

Da questa pagina è anche possibile per l'admin eseguire l'accesso. L'accesso avviene tramite un pulsante (che mostra la scritta "Login")

dopo aver premuto il pulsante appare un form che permette di inserire i dati. I dati richiesti sono semplicemente username e password.

Per verificare i dati è richiesto di premere il pulsante login. Se i dati inseriti sono validi verrà caricata in una nuova scheda la pagina dell'admin.

Figure 10: form login



The screenshot shows three sliders on a light yellow background. Each slider has a green handle and a numerical value displayed below it. The first slider is labeled 'Edges Density:' and has a value of 0.1. The second slider is labeled 'Initial Scale:' and has a value of 4. The third slider is labeled 'Step Size:' and has a value of 2.

Parameter	Value
Edges Density	0.1
Initial Scale	4
Step Size	2

Figure 11: Pagina Admin

### Creazione pagina admin

La pagina dell'admin serve per permettere di modificare alcune impostazioni relative alla pagina della webcam. Queste impostazioni sono: 1. Edges Density 2. Initial Scale 3. Step size 4. Giorni arretrati 5. Refresh automatico della pagina dei grafici. La prima impostazione permette di cambiare la densità del rettangolo che si crea attorno alle facce individuate, la seconda serve per definire la larghezza del rettangolo, la terza permette di modificare la frequenza con la quale aggiornare il frame della pagina. I Giorni arretrati e il refresh automatico sono invece impostazioni relative alla pagina dei grafici. I giorni arretrati indicano per quanti giorni i dati sono salvati all'interno dei due grafici, il refresh indica ogni quanti secondi la pagina verrà ricaricata in modo automatico.

per immettere i valori abbiamo creato la pagina (per quanto riguarda la parte html) in modo dinamico, così facendo si evita di inserire ogni volta del codice per aggiungere eventuali impostazioni. Per farlo la pagina si connette al database e conta il numero di righe nella tabella configurazioni (che corrispondono al numero di impostazioni che l'admin ha a disposizione). Dentro a un ciclo vengono generati tanti input quante sono le impostazioni.

```
$i= 0;
$result=mysqli_query($conn,"SELECT count(*) as total from configurazione");
$data=mysqli_fetch_assoc($result);
while($i < $data['total']){
echo "<p>$Testo[$i]</p>";
<input type='range' min='1' max='50' value='' class='slider' id='$i'
      name='$i' onchange='nuovoValore()'>
Valore: <input type='text' class='tbx' id='value.$i'>";
$i++;
}
```

Questa pagina è stata ideata per fare in modo che le impostazioni venissero passate al database. Dal database le nuove impostazioni devono essere mandate alla pagina della webcam per aggiornarla.

▼	facedetection	48.0 KiB
	amministratore	32.0 KiB
	webcam	16.0 KiB

Figure 12: SO Caricamento

#	Nome	Tipo di dati	Lunghezza/set	Senza segno	Permetti NULL	Riempi con z...	Predefinito
1	Id_webcam	INT	11	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	AUTO_INCREMENT
2	Orario_inizio	TIME		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL
3	Orario_fine	TIME		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL
4	Data	DATE		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL

Figure 13: SO Caricamento

### Creazione Database

Per far sì che le pagine comunicano e si scambino i dati tra di loro è stato necessario creare un database. Per crearlo abbiamo usato Heidi versione 9.4.0.5125. Heidi è un programma che permette di creare database tramite un'interfaccia grafica. Ecco come si presenta il database su Heidi.

La tabella Webcam si presenta così:

Mentre la tabella amministratore si presenta in questo modo:

Ora abbiamo un database in grado di comunicare i propri dati.

#	Nome	Tipo di dati	Lunghezza/set	Senza segno	Permetti NULL	Riempi con z...	Predefinito
1	Nome_Utente	VARCHAR	20	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Nessun
2	Id_webcam	INT	11	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Nessun
3	Password	VARCHAR	50	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Nessun
4	Densità_bordo	DOUBLE		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Nessun
5	Scala_iniziale	DOUBLE		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Nessun
6	Dimens_step	DOUBLE		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Nessun
7	Contegg_seco...	INT	11	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Nessun

Figure 14: SO Caricamento

### Test

#### Protocollo di test

Le tabelle sottostanti rappresentano i test che abbiamo svolto in base ai requisiti che abbiamo scelto e creato.

Test Case	TC-001
Nome	Creazione macchina virtuale
Riferimento	REQ-002
Descrizione	Creazione macchina virtuale per gestire le cartelle su raspberry

<b>Prerequisiti</b>	
<b>Procedura</b>	- Installare un programma per gestire le macchine virtuali, noi abbiamo usato VirtualBox. - Creare una macchina virtuale linux basata su raspbian, in questo caso abbiamo usato debian
<b>Risultati attesi</b>	Avere la stessa struttura di cartelle che é presente su raspbian.

Test Case	TC-002
<b>Nome</b>	Creazione pagina Web per webcam
<b>Riferimento</b>	REQ-003
<b>Descrizione</b>	Gestire il riconoscimento facciale
<b>Prerequisiti</b>	-
<b>Procedura</b>	- Scaricare la libreria traking.js - Modificare la libreria con il linguaggio JavaScript.
<b>Risultati attesi</b>	Avere la pagina web che riconosce le facce e manda le informazioni al database

Test Case	TC-003
<b>Nome</b>	Creazione pagina Web per grafici
<b>Riferimento</b>	REQ-004
<b>Descrizione</b>	Gestire i dati mandati dalla pagina web della webcam tramite dei grafici
<b>Prerequisiti</b>	-
<b>Procedura</b>	- Creare i grafici - Prendere i dati dal database e inserirli all'interno dei grafici.
<b>Risultati attesi</b>	I grafici vengono mostrati correttamente in base ai dati presi dal database.

Test Case	TC-004
<b>Nome</b>	Creazione pagina Web per gestione dell'admin
<b>Riferimento</b>	REQ-005
<b>Descrizione</b>	Creazione di una pagina inserente all'accesso admin per la modifica dei parametri della webcam
<b>Prerequisiti</b>	Aver creato la pagina web per la Webcam
<b>Procedura</b>	- Creare un login con l'uso di JavaScript - Gestire la pagina Amministratore inserendo i parametri di modifica.
<b>Risultati attesi</b>	L'utente admin riesce a accedere alla pagina e a modificare i parametri.

Test Case	TC-005
<b>Nome</b>	Creare una banca dati

<b>Riferimento</b>	REQ-006
<b>Descrizione</b>	Creazione di una banca dati che contiene i dati raccolti dalla pagina web della Webcam
<b>Prerequisiti</b>	Aver creato la pagina della webcam
<b>Procedura</b>	- Scaricare un programma per creare il database, noi abbiamo utilizzato Heidi - Creare il database con gli stessi parametri della libreria presa e del codice scritto nella pagina web della Webcam.
<b>Risultati attesi</b>	Il database riesce a prendere i dati delle pagine prescritte

---

Test Case	TC-006
<b>Nome</b>	Ricerca di nuovi volti
<b>Riferimento</b>	REQ-007
<b>Descrizione</b>	I volti vengono trovati dalla pagina della Webcam
<b>Prerequisiti</b>	Download libreria tracking.js
<b>Procedura</b>	- Aver installato la libreria tracking.js - Gestire il riconoscimento facciale tramite JavaScript
<b>Risultati attesi</b>	La pagina della Webcam é in grado di riconoscere i volti

---

Test Case	TC-007
<b>Nome</b>	Salvataggio delle persone sul DB
<b>Riferimento</b>	REQ-008
Test Case	TC-007
<b>Descrizione</b>	Questo test serve per verificare il corretto funzionamento dell'immissione dei dati all'interno del database
<b>Prerequisiti</b>	aver creato il database
<b>Procedura</b>	- Collegare il database alle pagine tramite php
<b>Risultati attesi</b>	I dati presi dalla webcam sono presenti all'interno del database

---

Test Case	TC-008
<b>Nome</b>	Utilizzo del prodotto su Raspberry
<b>Riferimento</b>	REQ-009
<b>Descrizione</b>	Installazione del sistema operativo su Raspberry e installazione webserver
<b>Prerequisiti</b>	avere un Raspberry
<b>Procedura</b>	- Caricare l'immagine del sistema operativo sul raspberry - Installare un webserver tramite in comandi opportuni
<b>Risultati attesi</b>	Raspberry é effettivamente un webserver.

---



## Risultati test

I risultati dei test non sono male ma neanche eccelsi. I test TC-001, TC-003, TC-004, TC-005, TC-006, TC-008 sono passati. Ma il problema principale non è stato tanto la creazione delle pagine ma il collegamento tra loro mediante il database. Un altro problema è stato il Raspberry, quando abbiamo eseguito i test della prestazione ci siamo accorti che effettivamente il nostro raspberry era troppo “debole” per poter reggere il nostro programma. La webcam andava a scatti e il raspberry di impallava. Per rimediare abbiamo utilizzato un computer prestatoci dalla scuola.

## Mancanze e limitazioni conosciute

Per noi è stata difficile la partenza, cioè suddividerci il lavoro e imparare a lavorare come un team. Una volta capito il vero funzionamento del lavoro di squadra e una volta che abbiamo suddiviso i compiti siamo riusciti a arrivare ad avere un vero e proprio team organizzato. Per quanto riguarda le competenze informatiche abbiamo avuto qualche difficoltà con l'utilizzo di programmi o linguaggi che non abbiamo mai usati. Come per esempio raspberry, abbiamo dovuto installare il suo sistema operativo (raspbian) e installare un webserver su esso.

Il raspberry presentava però un problema, ovvero che non era in grado di reggere la pagina della webcam, infatti molte volte ci sono state interruzioni della pagina o anche crash del sistema. Per questo abbiamo deciso di trasferire il tutto su un muletto.

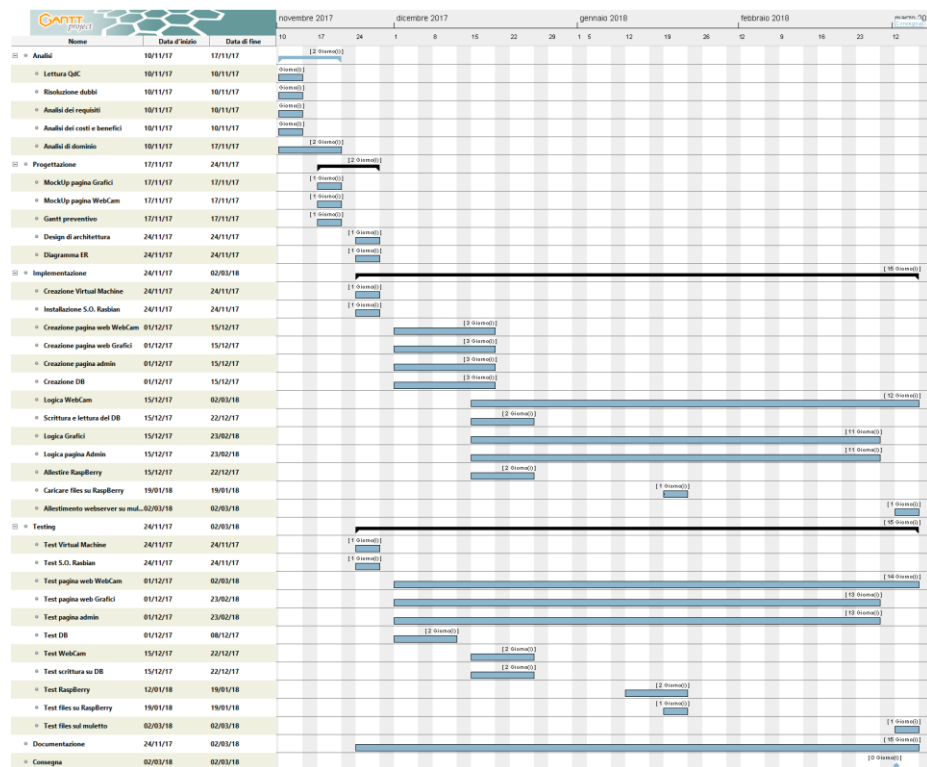


Figure 15: Gantt Preventivo

## **Consuntivo**

Ecco come sono andate le ore di lavoro, in base a quelle che abbiamo programmato prima dell'inizio del progetto.

## **Conclusioni**

La soluzione che abbiamo portato ci soddisfa ma non al 100%. Il nostro programma avrà sicuramente un impatto positivo con le persone che lo proveranno, porterà molto divertimento. Noi non pensiamo che il nostro programma cambierà il mondo ma siamo sicuri che porterà una piccola svolta, sicuramente più per noi stessi che per gli altri. Possiamo definire questo progetto un successo importante, più che successo questo progetto ci porta un forte orgoglio personale soprattutto perchè verrà utilizzato da gente esterna alla nostra scuola. Questo progetto é una grande aggiunta alla nostra crescita professionale, ci ha dato molto questo progetto sia come competenze lavorative che come competenze sociali.

## **Sviluppi futuri**

Come miglioria potrebbe essere implementato in una scala molto più grande rispetto che un semplice schermo con una webcam. Sarebbe bello poter collegare il nostro progetto su delle videocamere reali in modo da riuscire a riconoscere una quantità maggiore di volti. Mentre come sviluppo futuro vorremmo riuscire a migliorare il nostro prodotto, in modo da avere un programma che funzioni alla perfezione.

## **Considerazioni personali**

Con questo progetto abbiamo imparato cosa vuol dire lavoro di squadra, di quanto esso sia estremamente importante e di come con dei collaboratori sia più facile e eccitante lavorare.

## Sitografia

- <https://trackingjs.com/>, \*Tracking.js La libreria per implementare il riconoscimento facciale.
- <https://www.wikihow.it/Creare-un-Web-Server-su-Raspberry-Pi>, \*Manuale web-server linux Abbiamo usato questo sito per creare un web server su raspberry.
- <https://www.raspberrypi.org/downloads/raspbian/>, \*Sistema operativo Raspbian Abbiamo scaricato il sistema operativo direttamente dal sito del produttore di raspberry.
- <http://www.vemp.org/raspberrypi/preparare-una-card-sd-con-raspbian/>, \*Programma per caricare il .img di raspbian su raspberry.
- <https://www.w3schools.com/js/>, \*Guida JavaScript Abbiamo utilizzato questa guida per eventuali errori o mancanze delle nostre competenze sul linguaggio.
- <https://www.w3schools.com/php/>, \*Guida Php Abbiamo utilizzato questa guida in caso di mancanze o scarse competenze.
- <http://www.chartjs.org/>, \*pagina per i grafici

## Allegati

Elenco degli allegati, esempio:

- Diari di lavoro
- Guida utente / Manuale di utilizzo