

## **CAPITOLO 3 - L'applicazione web Image-Compare.**

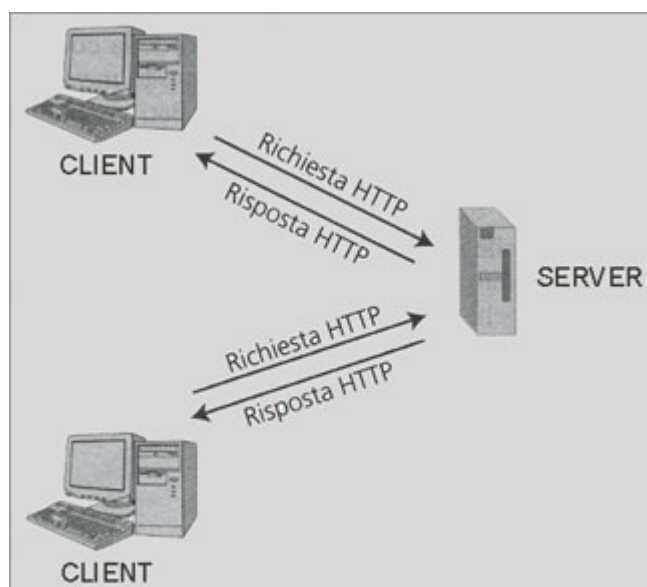
Nei precedenti capitoli sono state poste le basi per realizzare un'applicazione che sia in grado di espletare le operazioni di archiviazione e ricerca di immagini per contenuto cromatico attraverso l'utilizzo di un database. Siamo partiti illustrando gli scenari e i motivi per i quali può risultare utile un'applicazione di questo genere ed abbiamo quindi introdotto tutti i concetti e gli elementi necessari per poterla definire. Possiamo quindi procedere nel descrivere le funzionalità dell'applicazione e il modo in cui questa può essere utilizzata. A tal fine procederemo separando gli argomenti: per prima cosa verranno esposte quelle che sono le funzioni basilari che permettono all'applicazione di raggiungere lo scopo dichiarato, mentre successivamente parleremo delle funzionalità offerte dall'interfaccia, mediante la quale un generico utente utilizza l'applicazione per i fini preposti. Prima di entrare nei dettagli di questi argomenti è però utile fare una breve panoramica sul tipo di architettura sulla quale l'applicazione è costruita e sui vari linguaggi utilizzati per la sua realizzazione. Questa descrizione è opportuno che preceda gli altri argomenti in quanto, nel seguito della trattazione, per una buona comprensione della stessa, è fondamentale aver ben presenti queste informazioni.

### **3.1 - Architettura e linguaggi.**

L'architettura dell'applicazione, dovendo essere questa di tipo web, è ovviamente costruita sul modello client-server. Dobbiamo però evidenziare da subito un aspetto importante: il fatto che si vuole realizzare un'applicazione web non è l'unico motivo per cui l'architettura deve essere di tipo client-server, infatti, anche PostgreSQL è un database che opera a sua volta nella modalità client-server. A tal riguardo è giusto dire che per il funzionamento di PostgreSQL è sempre necessaria la presenza di un processo, postmaster, residente sul server, il cui scopo è appunto quello di stare in ascolto delle richieste di connessione al database che, una qualsiasi applicazione lato client, può inoltrare. Il processo in questione viene attivato dal sistema operativo come servizio dell'utente postgres; sappiamo che postgres è il super user

creato di default da PostgreSQL ad ogni sua installazione. Generalmente, se non diversamente specificato, postmaster sta in ascolto sulla porta 5432. Aver ben chiaro da subito questo aspetto è importante in considerazione del fatto che, l'applicazione web, sarà realizzata in modo tale da essere completamente residente sulla stessa macchina fisica, che verrà identificata di conseguenza sia come server che come client, ragion per cui dobbiamo sempre tener presente, quando parliamo di richieste fatte al database, che esse avvengono tramite la seguente coppia indirizzo IP, porta: 127.0.0.1:5432 (oppure localhost:5432), poiché, questa, è di fatto la socket utilizzata per la connessione al database server PostgreSQL nel caso in cui, sia il database server, che l'applicazione client, siano fisicamente residenti sullo stesso host.

Fatta questa doverosa precisazione torniamo a discutere dell'architettura generale dell'applicazione. Come ogni applicazione web, abbiamo già detto che essa si basa sul modello client-server. In figura 3.1 è schematizzata una generica architettura client-server, con host client e server distinti, sulla quale gira un'applicazione di tipo web.



**Figura 3.1 - Modello Client-Server**

Ritornando alla nostra applicazione web, questa dovrà quindi essere strutturata in

modo tale da poter ricevere da un client le richieste di archiviazione e ricerca delle immagini, inoltrando a sua volta, in ogni caso necessario, ulteriori richieste al database server per poter soddisfare le richieste pervenute dal client. A tale scopo, un linguaggio di programmazione che consente di realizzare un'applicazione che possa effettuare queste operazioni è PHP. La sigla PHP sta per Hypertext Preprocessor. PHP è un linguaggio di scripting server-side che permette di generare pagine web in maniera dinamica, consentendo di interfacciarsi ad un database per generare i contenuti delle pagine stesse. Nel modello client-server, per rispondere alle funzionalità che l'applicazione deve offrire, possiamo quindi ritenere indispensabile la presenza dei seguenti elementi: il browser, il server HTTP, l'interprete PHP, detto anche motore, ed il DBMS. Tenendo presente quanto detto in precedenza, cioè che il client ed il server sono residenti sulla stessa macchina, possiamo procedere ad illustrare brevemente il funzionamento di ognuno degli elementi appena identificati.

Il browser, che svolge la funzione di client dell'applicazione web, consente di inoltrare delle richieste al server HTTP (Apache 2.2.3). L'oggetto di ogni richiesta è rappresentato da un URL (Uniform Resource Locator) di una pagina web, la quale può essere di due tipi: statica o dinamica.

Nel caso statico, la pagina web richiesta è una pagina scritta totalmente in HTML o XHTML, cioè in uno dei linguaggi preposti alla scrittura di pagine web, le quali possono essere interpretate e visualizzate da un browser. Per la realizzazione dell'applicazione oggetto di questo lavoro di tesi è stato utilizzato XHTML.

Nel caso dinamico, invece, il browser richiede una pagina che al suo interno presenta uno script, cioè una pagina web dove, oltre al codice XHTML, è presente del codice dinamico scritto in un linguaggio di scripting server-side, come ad esempio l'appena citato PHP. In questo secondo caso, il server HTTP, non può restituire immediatamente la pagina web richiesta dal browser, poiché non è in grado di elaborare il codice dinamico. Dopo aver appreso dall'estensione .php della pagina richiesta, che si tratta di una pagina dinamica, il server HTTP deve provvedere al parsing della pagina, inoltrando il codice dinamico ad un motore PHP

(Zend), ogni volta che all'interno della pagina lo incontra. Il motore provvede quindi all'interpretazione di questo codice e all'esecuzione delle azioni richieste, al fine di produrre i risultati necessari al server HTTP per generare la pagina web, che successivamente verrà restituita al browser, il quale, come abbiamo già detto, ha la sola facoltà di poter leggere codice HTML o XHTML. Generalmente, alla richiesta di una pagina web dinamica, può essere abbinata la trasmissione dal browser al server HTTP di alcuni parametri, che a seconda dei valori assunti, notificano delle diverse richieste fatte dal client, a seguito delle quali lo stesso, in linea di principio, si aspetta di volta in volta dei risultati distinti, determinati appunto dall'elaborazione dei diversi valori inviati.

Il database entra in gioco quando nel codice dinamico sono presenti delle richieste di connessione verso di esso, al fine di poter inoltrare delle query ed ottenere dei risultati. In questo contesto, per rendere meglio l'idea, possiamo vedere il server HTTP come elemento bivalente del modello client-server: si comporta da server nei confronti del browser e da client verso il database. Questa visione dell'architettura potrebbe destare alcune perplessità, poiché di fatto non è il server web a comunicare con il database server, ma bensì il motore PHP, che riesce a farlo tramite l'abilitazione di un'opportuna estensione. Essendo più che lecita questa perplessità, ritengo che, questo modo di vedere l'architettura possa essere compreso con maggior cognizione di causa facendo un dovuto approfondimento sul fatto che il server HTTP può avere due distinte modalità di interazione con l'interprete PHP, dipende da come quest'ultimo è stato installato sull'host server. Gli approcci possono essere due: il primo consiste nell'installazione di PHP sotto forma di applicazione CGI; il secondo nell'installazione sotto forma di modulo del server web.

Nell'installazione di tipo CGI, l'interprete PHP verrà invocato dal server web come una qualsiasi altra applicazione in grado di comunicare con esso attraverso la Common Gateway Interface. L'interprete PHP, dunque, sarà un programma esterno al server web. Questo tipo di installazione è il più semplice, ma comporta alcuni svantaggi, sia in termini di prestazioni che in termini di funzionalità. Le prestazioni

vengono influenzate negativamente dalla necessità di creare un nuovo processo ogni volta che viene invocato uno script PHP, con un considerevole aggravio del carico di lavoro della macchina server. Inoltre, alcune funzionalità del linguaggio, in particolare quelle che richiedono un'interazione più profonda con il server web, sono disponibili solo quando PHP viene utilizzato come modulo.

Il secondo approccio, cioè l'installazione dell'interprete PHP come modulo del web server Apache, può avvenire seguendo due strade d'integrazione: sotto forma di modulo statico o dinamico. Nel primo caso (modulo statico) sarà necessario effettuare la ricompilazione del server web ed il risultato ottenuto sarà un nuovo eseguibile, completo di supporto PHP. Nel secondo caso (modulo dinamico) l'interprete PHP verrà compilato come libreria a caricamento dinamico: DSO (Dynamic Shared Object) su piattaforme Unix e DLL (Dynamic Link Library) su piattaforme Windows. La più diretta conseguenza dell'integrazione di PHP all'interno del server web è rappresentata dal fatto che l'esecuzione di uno script PHP non richiede l'invocazione di un programma esterno (e quindi la creazione di un nuovo processo, come avveniva nell'ipotesi di installazione CGI) in quanto viene effettuata all'interno dello stesso server web. L'installazione di PHP sotto forma di modulo del server web è quella che fornisce le prestazioni più elevate, peraltro senza rinunce in termini di funzionalità disponibili; tale approccio sarà quindi da preferire in contesti applicativi, come quello nel quale ci troviamo adesso, per la realizzazione dell'applicazione web di nostro interesse.

In relazione a quanto esposto è quindi assolutamente lecito considerare il server web come l'applicazione client che inoltra delle richieste, mediante il proprio modulo PHP integrato, e resta in attesa dei risultati forniti dal database server per generare la pagina richiesta. Nei fatti, lo script PHP che genera la richiesta di connessione verso il database viene eseguito internamente al web server, non esiste quindi un processo nel sistema, riconducibile allo script in esecuzione, dal quale può essere identificata una richiesta di connessione al database, ed è per questo che si può vedere il server web come elemento bivalente del modello client-server. Detto ciò si può ritenere ben definita, forse anche oltre il necessario, l'architettura di

base dell'applicazione web. Prima di procedere però alla descrizione delle funzioni, devono essere fatti due ulteriori cenni riguardo ai linguaggi utilizzati per realizzare l'applicazione.

Fino a questo punto è stato appurato l'utilizzo di PHP come linguaggio di programmazione server-side e di XHTML come linguaggio di marcatura per la strutturazione delle pagine web. Alla realizzazione globale dell'applicazione concorrono anche altre due realtà: JavaScript e CSS. Il primo è un linguaggio di scripting client-side che viene processato sul client (quindi dal browser) e serve per effettuare dei controlli sulle operazioni fatte dall'utente, prima di inviare le richieste al server HTTP. Nel nostro caso JavaScript è anche utilizzato per la creazione del menù dinamico presente nell'interfaccia. In verità, senza voler comunque affrontare questioni che esulano dal contesto, il suddetto menù è realizzato con un connubio tra PHP e JavaScript, poiché esso deve essere dinamico sia nella forma che nei contenuti. CSS è invece l'acronimo di Cascading Style Sheet ed è il linguaggio in cui vengono scritti i fogli di stile per il web. Un foglio di stile serve per separare la definizione delle regole che determinano il layout delle pagine dai tag di marcatura propri dell'XHTML. Mentre i tag sono i componenti che definiscono la struttura della pagina, e sono effettivamente gli elementi che la costituiscono, le regole scritte in un foglio di stile determinano come la pagina deve essere rappresentata, ed è giusto che queste regole siano dichiarate separatamente dalla struttura della pagina, poiché i linguaggi di marcatura, quali l'HTML prima e l'XHTML dopo, non sono nati per stabilire come rappresentare una pagina web, ma solo per definirne la struttura. L'applicazione utilizza a tal riguardo un foglio di stile esterno (screen.css), collegato alle varie pagine web tramite la direttiva `@import`.

### **3.2 - Le funzioni alla base dell'applicazione.**

In questo paragrafo, l'intento sarà quello di fornire una panoramica sulle funzioni PHP basilari, che permettono all'applicazione web di effettuare le operazioni per le quali è stata progettata. L'intenzione non è comunque quella di effettuare un'analisi

dettagliata del codice delle singole funzioni, poiché questa sarebbe un'operazione inutile per gli scopi di questa tesi. L'obiettivo sarà, invece, quello di fornire al lettore una cognizione sufficiente per orientarsi in tutto quello che sta “dietro all'applicazione”, ovvero tutto ciò che non è possibile intuire con il solo utilizzo del front-end della stessa. Non parleremo quindi delle funzionalità presenti nel menù dell'interfaccia, poiché queste rappresentano di fatto delle macro azioni, che saranno descritte nel paragrafo successivo.

In primo luogo, possiamo iniziare dicendo, che l'applicazione web può essere richiamata tramite un qualsiasi browser, digitando nella sua barra degli indirizzi il seguente URL: `http://localhost/www.image-compare.net`. Se tutti i software sono stati installati e configurati correttamente, dopo aver atteso alcuni secondi, dovrebbe apparire sul browser l'interfaccia grafica dell'applicazione, alla quale è stato attribuito il nome Image-Compare. La stringa `www.image-compare.net` rappresenta, come di consueto, il nome attribuito alla directory, collocata nella document root del server web Apache, dove sono presenti tutti gli script PHP necessari al funzionamento dell'applicazione.

Nella sotto directory *includes* troviamo due file di enorme importanza: *setting.php* e *common.php*. Il primo è il file di configurazione dell'applicazione, dove sono definiti i parametri di accesso al database, le estensioni dei file supportati, i valori HSI definiti per generare la mappa dei colori RGB e alcuni parametri necessari per l'upload e l'elaborazione delle immagini. Poche righe sopra, abbiamo detto, che l'applicazione impiega alcuni secondi prima di rendere disponibile sul browser la propria interfaccia grafica, vediamo subito per quale motivo.

Ad ogni nuova apertura di sessione di Image-Compare, l'applicazione, prima di rendersi disponibile per l'archiviazione e/o la ricerca di immagini, o per qualsiasi altra macro azione accessibile dal menù dell'interfaccia, deve effettuare alcune operazioni. La prima operazione è quella di stabilire una connessione con PostgreSQL, in modo tale da poter accedere al database che abbiamo descritto nel secondo capitolo. Questo accesso avviene mediante la chiamata della funzione PHP *pg\_connect()* inserita nello script *common.php*. Possiamo dire che in questo script

sono chiamate e definite le funzioni più importanti dell'applicazione; questo script è infatti incluso sistematicamente in tutti gli script che costituiscono l'applicazione. Dopo aver stabilito la connessione, l'applicazione effettua un'altra importantissima operazione, della quale avevamo già dato cenno nel primo capitolo: procede alla definizione e alla creazione della mappa dei colori. Partendo dalle componenti HSI definite nel file di configurazione, l'applicazione calcola i colori equivalenti nello spazio RGB (mantenendo le componenti di ogni singolo colore in una variabile di sessione) e immediatamente dopo provvede a calcolare i coefficienti della matrice di similarità, relativi alla mappa dei colori appena generata. Questa modalità operativa consente all'applicazione di essere aggiornata in tempo reale su qualsiasi eventuale cambiamento, che può essere fatto nel file di configurazione, riguardo alle componenti HSI scelte. Lo stesso procedimento, definito “Rigenera Mappa Colori”, può essere avviato anche manualmente, tramite lo script *wiew\_color.php*, raggiungibile dal menù dell'interfaccia dall'area “Help”, nella sezione “Colori della mappa”. A questo procedimento concorrono varie funzioni, tra le quali, le più importanti, possono essere individuate senza dubbio nelle funzioni *convert\_HSI2RGB(\$tinta,\$satur,\$intens)* e *MatricePesi()*. La prima riceve in ingresso le componenti HSI di un colore e restituisce le componenti equivalenti di quel colore nello spazio RGB, implementando di fatto le formule esposte nel prospetto 1.4 del paragrafo 1.5. La seconda, nota le componenti RGB di tutti i colori presenti nella mappa, che ricordiamo essere mantenute da una variabile di sessione, calcola i coefficienti della matrice di similarità in conformità a quanto descritto nel paragrafo 1.4. Calcolati tutti i coefficienti, come anticipato nel secondo capitolo, questi vengono memorizzati in un'apposita tabella del database, per i motivi sui quali abbiamo già fornito ampie spiegazioni nel paragrafo 2.4. terminate tutte queste operazioni, l'applicazione è ora disponibile per essere utilizzata. Solitamente, dal momento in cui è richiamata al momento in cui si rende disponibile, l'applicazione impiega circa 15 secondi per espletare tutte le operazioni appena descritte. Il tempo indicato è riferito a prove di utilizzo effettuate su una macchina con le caratteristiche hardware riportate nella sezione A.1 dell'appendice A. Tutti i



riferimenti successivi ai vari tempi di elaborazione sono stati rilevati nelle stesse condizioni operative.

Siamo quindi arrivati nella situazione di poter descrivere le funzioni che permettono di effettuare le operazioni principali: l'inserimento e la ricerca delle immagini. Per poter inserire un'immagine è necessario aver creato una galleria, poiché l'applicazione richiede che, al momento dell'inserimento di ogni immagine, venga preventivamente specificato in quale galleria deve essere inserita. La creazione, la modifica e l'eliminazione di una galleria vengono effettuate dagli script *build\_gallery.php*, *alter\_gallery.php* e *remove\_gallery.php*, all'interno dei quali non sono utilizzate funzioni di particolare interesse. L'unica eccezione può essere fatta per la funzione *rmdirtree(\$dirname)* che viene chiamata per eliminare fisicamente dal disco tutti i file delle immagini contenute nella galleria e la directory che le contiene. Il path fisico della directory da eliminare è infatti l'unico parametro della funzione. Sappiamo dai capitoli precedenti, che per ogni immagine inserita, vengono memorizzati sul disco quattro distinti file, e che l'eliminazione di una galleria comporta l'eliminazione in cascata di tutte le risorse riconducibili alle immagini contenute, siano esse risorse sul disco o righe di tabelle del database.

Sicuramente è di maggior interesse l'inserimento di un'immagine. Se l'immagine è di un formato accettabile (.gif, .jpg, .jpeg o png) e rispetta i limiti di dimensione impostati nel file di configurazione, l'applicazione effettua il suo upload sul disco. Siamo già a conoscenza del fatto che in fase d'inserimento, oltre all'upload del file, avvengono altre operazioni di importanza cruciale per il funzionamento dell'applicazione: la mappatura dell'immagine e la creazione del suo istogramma di colore. Alla base di queste due operazioni agiscono due funzioni fondamentali: *MakeMapped(\$src,\$dst,\$image\_ext)* e *Color(\$point,\$max\_dist,\$min\_dist)*. La prima riceve in ingresso tre parametri: l'immagine inserita da mappare, la destinazione dove memorizzare l'immagine mappata e l'estensione dell'immagine. A questo punto, grazie all'abilitazione dell'estensione di PHP alle librerie grafiche GD2, PHP dispone di funzioni che possono manipolare l'immagine per estrarre il contenuto di colore, espresso in coordinate RGB, di ogni suo singolo pixel. Quindi,

per ogni singolo pixel, dopo aver estratto le componenti di colore, viene chiamata la seconda funzione: *Color(\$point,\$max\_dist,\$min\_dist)*. Anche questa riceve in ingresso tre parametri: un vettore che contiene le componenti RGB del pixel da mappare e la massima e la minima distanza (euclidea) che intercorre tra due colori presenti nella mappa. Queste due distanze sono calcolate precedentemente alla chiamata tramite le funzioni *Distanza(\$point\_1,\$point\_2)* e *MinDistMap()*. La funzione (*Color*) restituisce in uscita il colore della mappa che si trova a distanza minima dal colore del pixel, per il quale la funzione è stata invocata. Il colore restituito è ovviamente quello attribuito al pixel di pari coordinate spaziali nell'immagine mappata. Il tempo necessario a mappare un pixel può variare molto a seconda del fatto che il colore a distanza minima venga individuato tra i primi o tra gli ultimi colori della mappa. Il tempo necessario a mappare un'immagine, poiché è dato dalla somma dei singoli tempi impiegati a mappare ogni pixel, a parità di dimensioni, può variare molto da un'immagine all'altra mettendo a confronto diverse immagini di tipo monocolori, mentre è abbastanza costante per immagini distinte che raffigurano situazioni realistiche. Il motivo dovrebbe essere facilmente intuibile: immagini che raffigurano situazioni reali hanno una distribuzione di colori abbastanza ampia, quindi, statisticamente, c'è la stessa probabilità di incontrare pixel che vengono mappati con colori collocati sia nelle prime che nelle ultime posizioni della mappa. Attualmente l'applicazione non accetta immagini che eccedono i 250.000 pixel e mediamente la mappatura di un'immagine di queste dimensioni può richiedere circa 4 minuti. Ulteriori considerazioni, sulle situazioni limite che determinano sostanziali differenze nei tempi di elaborazione, saranno fatte nell'ultimo capitolo.

Durante il processo di mappatura viene anche creato l'istogramma di colore dell'immagine, direttamente nella funzione *MakeMapped(\$src,\$dst,\$image\_ext)*. Il risultato della funzione, infatti, è proprio un vettore che rappresenta l'istogramma dell'immagine mappata, il quale viene inserito nel database insieme a tutti i dati che caratterizzano l'immagine. L'inserimento nel database, al pari di qualunque altra query SQL utilizzata nell'applicazione, avviene mediante una chiamata alla

funzione PHP *pg\_query(\$db\_connect, \$sql)*, dove la variabile *\$sql* è una stringa che rappresenta di volta in volta la query che deve essere eseguita e *\$db\_connect* è una variabile che identifica la connessione risultante dalla funzione *pg\_connect()*, menzionata ad inizio paragrafo.

Le ulteriori operazioni, sempre previste in fase d'inserimento di un'immagine, sono la creazione e la memorizzazione sul disco dei thumbnail dell'immagine originale e dell'immagine mappata. Queste operazioni avvengono mediante due distinte chiamate alla solita funzione: *MakeThumb(\$src,\$dst,\$dstx,\$dsty,\$image\_ext)*. I cinque parametri indicano in sequenza: il path dell'immagine da ridimensionare, il path di destinazione del thumbnail, la sua dimensione orizzontale e verticale, ed il formato, che deve essere lo stesso dell'immagine a dimensione reale. La dimensione dei thumbnail può essere modificata dal file di configurazione, benché questa operazione sia altamente sconsigliata. I thumbnail sono creati per fornire un preview delle immagini contenute in una galleria e l'interfaccia è progettata con un layout adatto alla loro attuale dimensione, stabilita in 128x128 pixel. Sebbene questa scelta possa sembrare inadeguata, cioè vincolare la dimensione dei thumbnail a scelte di impaginazione fatte a priori, posso affermare che progettare dei layout che si adattano alle dimensioni del loro contenuto, i così detti layout “liquidi”, è tutt'oggi operazione molto difficile e talvolta infattibile; questo avviene a causa della diversa interpretazione che i browser spesso danno alle regole dichiarate nei fogli di stile; tale fatto comunque è stato esposto solo a scopo informativo poiché non è rilevante per gli interessi dell'applicazione.

Resta adesso da parlare delle operazioni di ricerca, delle quali sappiamo già che la maggior mole di lavoro è a carico di PostgreSQL. Come è infatti sempre stato detto, le ricerche avvengono grazie alla funzione “confronto”, esposta nel capitolo 2, richiamata direttamente nella clausola WHERE di una query SQL. Gli script PHP che consentono di eseguire dei confronti sono sette: *compare\_image.php*, *compare\_all.php*, *compare\_gallery.php*, *compare\_onlyone.php*, *choose\_image.php*, *choose\_second\_image.php* e *choose\_gallery*. I tipi di confronti fruibili tramite l'interfaccia sono tre (come verrà dettagliato nel prossimo paragrafo) e, a seconda

del confronto richiesto, viene eseguita un'opportuna query SQL, sempre mediante la funzione `pg_query($db_connect, $sql)`, dove nella stringa `$sql` è contenuta la query nella quale è inserita la chiamata alla funzione “confronto” scritta in PL/pgSQL.

Le uniche operazioni che espletano questi script PHP sono, quindi, quelle di selezionare i dati con i quali effettuare la query. Questa operazione avviene tramite dei form XHTML che, da script in script, inviano in modalità POST i valori selezionati dall'utente, fino al momento in cui non si hanno tutte le informazioni per poter effettuare la ricerca richiesta. In sostanza, ai fini delle ricerche e dei confronti, questi script non fanno altro che raccogliere i dati necessari per formulare la query da inviare a PostgreSQL e mostrare i risultati ottenuti. Può sembrare strano, ma questo è proprio quello che abbiamo sempre voluto fin dall'inizio!

Con quanto detto, dovrebbe essere adesso più fattibile comprendere come l'applicazione esegue le macro operazioni proposte nel menù dell'interfaccia. Possiamo quindi procedere nella descrizione delle modalità con cui un utente può utilizzare il front-end dell'applicazione per effettuare queste operazioni.

### **3.3 - Le funzionalità eseguibili dall'interfaccia.**

L'interfaccia si presenta proponendo nella sua prima pagina (`index.php`), quella caricata di default ad ogni avvio di sessione dell'applicazione, denominata comunemente “Home Page”, un menù generale dove possono essere individuate quattro macro aree: confronta, immagini, gallerie ed help. Le solite quattro aree sono raggiungibili anche dal menù orizzontale collocato sotto l'intestazione (header) dell'interfaccia. In pratica, quest'ultimo è un menù ad accesso veloce, in quanto è presente in ogni sezione dell'interfaccia e talvolta permette anche di eliminare alcune selezioni, necessarie invece in certi casi per accedere alle sezioni d'interesse utilizzando il menù in Home Page. Come era stato detto nel primo paragrafo, questo è un menù dinamico realizzato in JavaScript, dove in alcune circostanze viene fatto uso del PHP per includere in tempo reale nelle voci di menù alcuni valori presenti nel database. In particolare mi riferisco ai submenù presenti nell'area “gallerie”.

Possiamo procedere adesso nel descrivere le funzionalità offerte da ogni singola area che abbiamo appena elencato. Questa descrizione è consultabile anche dall'interfaccia nell'area “Help”, alla voce “Documentazione”, poiché è di fatto l'esposizione delle modalità con le quali l'interfaccia deve essere utilizzata.

### 3.3.1 - Confronta

Da questa area è possibile selezionare tre distinte tipologie di confronto tra le immagini archiviate, ai fini di effettuare delle ricerche di similarità cromatica.

- *Con tutte le immagini:* questa funzionalità consente di effettuare il confronto tra un'immagine di riferimento e tutte le immagini archiviate. Questo confronto è articolato in tre distinte operazioni, elencate di seguito:

[1] Viene selezionata la galleria dalla quale si vuole scegliere l'immagine di riferimento per effettuare il confronto.

[2] Va impostato il valore di tolleranza entro il quale l'applicazione deve ritenere simili due immagini confrontate. Di default questo valore è impostato a 0.50, che di fatto è una tolleranza abbastanza elevata. In caso di modifica di questo valore, si deve prestare attenzione a separare i numeri decimali dagli interi facendo uso del punto e non della virgola. Vengono quindi visualizzate le immagini presenti nella galleria selezionata al punto 1 e viene richiesto di scegliere l'immagine desiderata. Passando con il mouse sopra alle immagini è possibile vedere le immagini mappate corrispondenti. Di default, è sempre selezionata per il confronto la prima immagine della galleria.

[3] Viene effettuato il confronto e vengono visualizzati i parametri e l'immagine di riferimento scelta, comprensiva di dettagli. Le immagini confrontate, che hanno soddisfatto il confronto entro la tolleranza indicata al punto 2, sono visualizzate a seguire. Le immagini sono ordinate in base alla loro distanza dall'immagine di riferimento. Per ogni immagine, oltre alla

distanza, sono riportati il nome e la galleria di provenienza. Inoltre, passando con il mouse sopra all'immagine, è sempre possibile vedere l'immagine mappata corrispondente. Se nessuna immagine dovesse soddisfare il confronto, l'applicazione notifica il risultato e rimanda all'Home Page. Infine, per ogni immagine che ha soddisfatto il confronto, può essere inoltrata una nuova richiesta di confronto, senza dover ripetere la procedura. Il nuovo confronto sarà effettuato utilizzando la tolleranza indicata nei parametri esposti relativi al confronto eseguito. Tale valore può anche essere modificato, se si desidera effettuare il nuovo confronto con una tolleranza diversa da quella precedentemente impostata.

- *Con una galleria:* questa funzionalità consente di effettuare il confronto tra un'immagine di riferimento e tutte le immagini presenti in una galleria scelta. Questo confronto è articolato in quattro distinte operazioni:

[1] Vedi punto 1 della funzionalità precedente.

[2] Vedi punto 2 della funzionalità precedente.

[3] Viene selezionata la galleria di immagini che devono essere confrontate con l'immagine di riferimento scelta al punto 2. La galleria può essere sia quella alla quale appartiene l'immagine di riferimento scelta, sia una galleria diversa.

[4] Vedi punto 3 della funzionalità precedente. L'unica variazione, rispetto al caso precedente, è rappresentata dal fatto che in questo caso, richiedendo di effettuare un nuovo confronto scegliendo una delle immagini che hanno soddisfatto quello appena eseguito, oltre alla tolleranza, può essere variata anche la galleria con la quale effettuare il nuovo confronto richiesto.

- *Con una sola immagine:* questa funzionalità consente di effettuare il confronto tra un'immagine di riferimento ed un'altra immagine scelta. Questo confronto è articolato in cinque distinte operazioni, elencate di seguito:

[1] Vedi punto 1 della funzionalità precedente.

[2] Vedi punto 2 della funzionalità precedente.

[3] Viene selezionata la galleria dalla quale scegliere l'immagine da

confrontare con l'immagine di riferimento scelta al punto precedente.

[4] Come al punto 2 vengono visualizzate le immagini presenti nella galleria selezionata e viene richiesto di scegliere un'immagine. In questo caso non è ovviamente richiesto di indicare nessun valore di tolleranza, poiché già impostato al punto 2.

[5] Le due immagini vengono confrontate e viene fornito il risultato del confronto. Se le due immagini non sono valutate simili entro la tolleranza indicata al punto 2, l'applicazione notifica il risultato e rimanda all'Home Page. In questo caso è superfluo offrire la possibilità di ripetere il confronto, poiché in nessun caso sussiste la convenienza di ripetere questo tipo di confronto senza reimpostare i dati.

### 3.3.2 - Immagini

Da questa area è possibile effettuare due distinte operazioni: l'inserimento e l'eliminazione dall'archivio di immagini. Da notare che è stato scelto di non poter accedere da questa sezione alla visualizzazione diretta di un'immagine. Questa scelta è dovuta al fatto che, tutte le immagini archiviate sono organizzate in gallerie tematiche, ed è stato quindi ritenuto più efficiente procedere prima alla visualizzazione di tutte le immagini presenti in una galleria e da lì poter accedere alla visualizzazione diretta di una singola immagine nelle sue dimensioni reali.

- *Inserisci:* tramite questa funzione è possibile archiviare un'immagine. Le immagini che l'applicazione accetta di memorizzare sono le sole supportate in ambito web: gif, jpg, jpeg e png. Inoltre, l'immagine che si vuole archiviare deve soddisfare un'altra serie di requisiti: almeno una delle sue dimensioni deve superare i 128 pixel; i pixel totali dell'immagine devono essere inferiori a 250.000; l'immagine non può eccedere i 250 KB; non possono essere archiviate due immagini dello stesso formato con lo stesso nome. Queste limitazioni vengono imposte per ottenere delle performance

soddisfacenti dall'applicazione e non risultano comunque essere limitative per gli scopi dell'applicazione stessa. Questi vincoli d'inserimento possono sempre essere visualizzati ponendo il mouse sopra al punto interrogativo collocato alla destra della maschera d'inserimento. Per ogni immagine inserita deve essere obbligatoriamente selezionata la galleria di destinazione, mentre sono opzionali l'inserimento di una dicitura didascalica e di un testo inerente le eventuali caratteristiche che l'immagine presenta. Se esistono delle gallerie complete, cioè contenenti il numero massimo di immagini stabilito per esse, queste non vengono visualizzate tra quelle selezionabili, poiché non possono accettare di archiviare ulteriori immagini. Dopo ogni inserimento, l'applicazione notifica l'avvenuta archiviazione dell'immagine, restituendo anche il tempo impiegato. Deve essere sempre ricordato che durante la fase d'inserimento di un'immagine avviene anche la sua elaborazione, necessaria alla creazione dell'immagine mappata corrispondente, dell'istogramma di colore e dei relativi thumbnail, ragion per cui, talvolta, questa fase può richiedere un tempo abbastanza elevato per il suo completamento (ovviamente variabile in funzione delle caratteristiche della macchina che svolge la funzione di server).

- *Elimina:* tramite questa funzione è possibile eliminare un'immagine dall'archivio. Viene per prima cosa richiesto di selezionare la galleria dalla quale si vuole eliminare una o più immagini. Inoltrata la selezione, l'applicazione procede a visualizzare le immagine presenti nella galleria indicata; ad ognuna è associato un checkbox che consente di selezionarla per l'eventuale eliminazione. Eliminare un'immagine comporta la cancellazione di tutti i file presenti sul server che si riferiscono a quell'immagine; vengono inoltre cancellati dal database tutte le informazioni ad essa relative. Selezionando più checkbox, l'applicazione esegue in una sola volta l'eliminazione di tutte le immagini selezionate. Questa operazione richiede sempre una conferma preventiva, prima di procedere alla cancellazione.



### 3.3.3 - Gallerie

Da questa area possono essere effettuate quattro distinte operazioni: creare una nuova galleria, modificare una galleria esistente, eliminare una galleria e visualizzare le immagini contenute in una galleria. Come già anticipato, dalla visualizzazione delle immagini presenti in una galleria sarà possibile accedere alla visualizzazione di una singola immagine.

- *Crea nuova*: questa funzione consente di creare una nuova galleria. La creazione di una galleria richiede obbligatoriamente di specificare un nome ed il numero massimo di immagini che l'applicazione può memorizzare in tale galleria. Non possono essere create due gallerie distinte alle quali viene attribuito lo stesso nome; ogni nome assegnato ad una galleria è case insensitive. Sono invece opzionali l'inserimento di una breve descrizione della galleria, comunque utile, unitamente al nome, per qualificarla, e le sue caratteristiche, dove per esse si sottintende una descrizione delle caratteristiche generali possedute dalle immagini che verranno inserite in essa. Il numero massimo di immagini di default è impostato a 30, poiché 30 è un numero ritenuto sufficiente per popolare una galleria d'immagini a tema. L'imposizione di questo limite è stata introdotta, benché questa applicazione non sia al momento destinata ad essere utilizzabile in rete, per evitare che un potenziale utente o un qualsiasi script effettuino inserimenti di immagini a ripetizione, con la finalità di sovraccaricare il server e metterlo fuori servizio (attacco DOS). Per completezza, deve essere detto che, insieme a questo accorgimento per garantire la sicurezza dell'applicazione, dovrebbero essere implementate anche delle politiche di controllo dei permessi, per differenziare le funzionalità alle quali ogni singolo utente ha accesso. Questa operazione non è stata fatta poiché, come già detto, attualmente l'applicazione non è progettata per operare su macchine server inserite in reti aperte, come internet. Ogni creazione di una nuova galleria determina la

creazione di una directory sul server, dove vengono memorizzati tutti i file relativi ad ogni singola immagine inserita nella galleria.

- *Modifica*: con questa funzione è possibile modificare i parametri con cui è stata creata una galleria, compresi il nome ed il numero massimo di immagini. Anche in questo caso non sarà accettata la richiesta di modificare il nome di una galleria con un nome già assegnato ad un'altra. Inoltre l'applicazione non consente di modificare il numero massimo di immagini specificando un valore inferiore al numero di immagini attualmente presenti nella galleria. Posizionando il mouse sopra al punto interrogativo, collocato a fianco del box dove è indicato il numero massimo di immagini inseribili nella galleria, verrà visualizzato questo avvertimento.
- *Elimina*: questa funzione consente di eliminare una o più gallerie. L'applicazione presenta la lista di tutte le gallerie e ad ognuna è associato un checkbox selezionabile per l'eventuale eliminazione. L'eliminazione di una galleria determina la sua cancellazione dal database e la conseguente cancellazione di tutte le immagini contenute in essa. Le immagini saranno cancellate sia dal database che dal disco, nello stesso modo se venissero cancellate dalla sezione “Elimina” dell'area “Immagini”. Infine, sarà anche cancellata la directory che conteneva i file relativi alle immagini presenti nella galleria eliminata. Selezionando più checkbox, l'applicazione esegue in una sola volta l'eliminazione di tutte le immagini selezionate. Questa operazione richiede sempre una conferma preventiva prima di procedere alla cancellazione.
- *Visualizza*: questa funzione permette di visualizzare tutte le immagini presenti in una galleria. Ad inizio pagina vengono sempre visualizzate le informazioni disponibili sulla galleria scelta. Per ogni immagine viene mostrato un thumbnail di dimensione pari a 128x128 pixel o inferiore in una delle due dimensioni, qualora l'aspect ratio dell'immagine originale non consenta di effettuare il resize quadrato. Passando con il cursore del mouse sopra ad ogni immagine sarà possibile vedere l'immagine mappata

corrispondente. Cliccando invece sul bottone "View Full Size" si accede alla visualizzazione dell'immagine nelle sue dimensioni reali. Per ogni immagine vengono visualizzate le caratteristiche principali e passando con il mouse sull'immagine originale è possibile vedere la corrispondente immagine mappata, anch'essa nelle reali dimensioni dell'immagine originale.

### 3.3.4 - Help

In questa area, oltre alla sezione di documentazione, che è equivalente nei contenuti al presente paragrafo 3.3, sono contenute due ulteriori pagine di aiuto, le cui funzionalità sono documentate a seguire.

- *Colori della mappa:* da questa pagina è possibile acquisire alcune importanti informazioni relative ai colori presenti nella mappa e al processo di mappatura di un'immagine. Si può accedere inoltre ad alcune funzionalità che possono risultare utili alla miglior comprensione del funzionamento di questa applicazione. In questa pagina ci si propone di fornire un'idea indicativa su cosa siano i "Colori della mappa" e su cosa sia il processo di mappatura di un'immagine. In particolare, vengono esplicitamente riportati tutti i colori che attualmente costituiscono la mappa dei colori. Sempre da questa pagina è possibile visualizzare una rappresentazione degli spazi di colore HSI e RGB. Inoltre, è presente un link ad una rappresentazione tridimensionale nel cubo RGB della distribuzione dei colori appartenenti alla mappa. L'ultima funzionalità offerta è la possibilità di rigenerare la mappa dei colori. Tale mappa viene generata ad ogni apertura di sessione dell'applicazione ed una sua rigenerazione, tramite il bottone "Rigenera Mappa Colori" presente in questa sezione, può determinare dei cambiamenti solo nel caso in cui siano state variate, dal file di configurazione, le componenti HSI scelte per generare i colori. Questa considerazione vale in maniera del tutto analoga per la conseguente ricreazione dei coefficienti della

matrice di similarità: se i colori non sono variati, la rigenerazione ricrea esattamente gli stessi coefficienti.

- *Documentazione*: pagina di documentazione sull'utilizzo dell'interfaccia dell'applicazione web Image-Compare.
- *Credits*: in questa pagina sono elencati i software e le librerie utilizzate per realizzare l'applicazione Image-Compare. Per ogni singola voce è riportato il logo identificativo ed il link al sito ufficiale del software indicato. Nei siti web è possibile reperire tutta la documentazione necessaria per eventuali approfondimenti sull'installazione e la configurazione degli ambienti di sviluppo utilizzati.