

Interfacce di gestione web-based

Prof. Viglietti Francesco

Www.in4matika.altervista.org

Motivazione

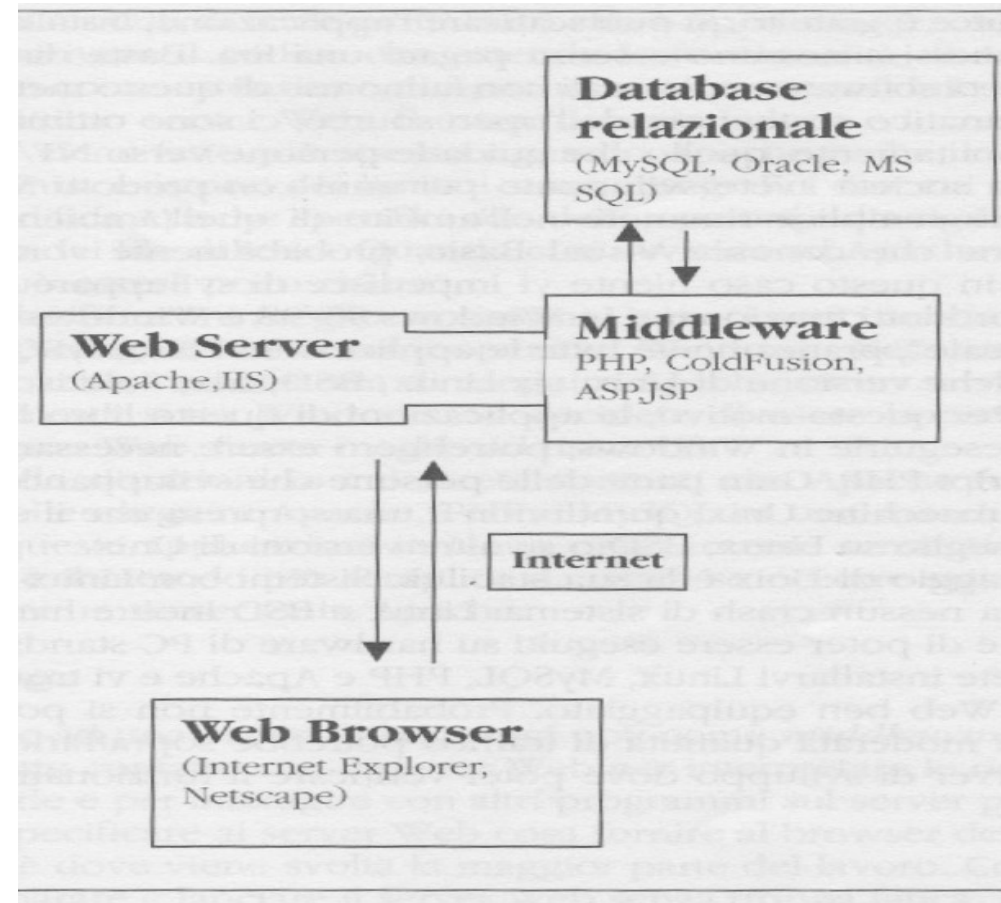
L'utilizzo di interfacce web per l'interrogazione di database rappresenta da una parte una competenza estremamente professionalizzante, dall'altra un forte incentivo all'approfondimento della conoscenza dei database.

Per la didattica dei database si tratta quindi di uno strumento eccellente in grado di incrementare la produttività della classe al costo di un minimo overhead legato all'apprendimento dei rudimenti dei linguaggi php e html.

Architettura

L'architettura delle applicazioni che fruiscono di database tramite interfaccia grafica web based è di tipo client/server.

Oltre al DBMS, cui accedere tramite il linguaggio SQL (sia per le query e l'inserimento dei dati che per la creazione dello schema del DB), sono necessari un'applicazione che svolga la funzionalità di server web, un middleware che abbia la funzione di interfaccia e un client rappresentato da un semplice browser web, che può essere locale o più probabilmente remoto



Componenti del framework: client

Le applicazioni che si possono sviluppare con MySql e PHP fanno uso di un singolo client: *il browser Web*. Questa non è l'unica possibilità per le applicazioni basate su Internet. Per applicazioni molto sofisticate che richiedono una maggiore elaborazione sul lato client o che devono mantenere uno stato può essere necessario un *applet java*. Ma a meno che non si debba realizzare qualcosa come un programma di chat in tempo reale, l'uso di Java sul lato client non è assolutamente necessario ed attualmente è deprecato dagli attuali browser.

Quindi l'unico client di cui preoccuparsi è il browser Web. Le applicazioni dovranno essere presentate nel browser. Il principale linguaggio dei browser è l'HTML (Hypertext Markup Language).

Naturalmente molti browser accettano anche altro oltre a HTML. Ci sono tutti i tipi di plug-in, inclusi RealPlayer, Flash e Shockwave. Tutti i browser attuali supportano anche **Javascript**, e possono lavorare con XML.

Noi useremo un approccio del tipo minimo comun denominatore, creando applicazioni che possano essere lette da ogni browser.

Componenti del framework: server web

Il server Web svolge quello che può sembrare un lavoro abbastanza semplice. Mentre è in esecuzione nel vostro OS, attende le richieste fatte da qualcuno sul Web, vi risponde e fornisce le appropriate pagine Web. In realtà, le cose sono più complicate di così, e per la natura del Web, la stabilità di un server Web è un elemento fondamentale.

Ci sono molti server Web tra cui scegliere, ma due tipi dominano il mercato. Sono Apache e IIS (Internet Information Server) di Microsoft. Quest'ultimo è strettamente connesso all'ambiente Windows ed è un componente chiave di ASP (Active Server Pages) di Microsoft. Per vari motivi la nostra scelta ricade sul suo principale concorrente

Componenti del framework: server web

Apache

Il server Web Apache è il server più diffuso. È, come Linux, PHP e MySQL, un progetto open source. Non è una sorpresa che Apache funzioni meglio in ambienti Unix, ma funziona bene anche sotto Windows. Apache utilizza moduli di terze parti. Dato che è open source, chiunque ne abbia le competenze può scrivere il codice che estenda le funzionalità di Apache. PHP viene spesso eseguito come estensione di Apache, nota come modulo di Apache.

Apache è un ottimo server Web. E' estremamente veloce e incredibilmente stabile. L'appunto che viene fatto più spesso riguardo Apache è che, come per molti altri software per Unix, esistono dei limitati strumenti grafici con cui manipolare l'applicazione. Si modifica Apache specificando delle opzioni sulla riga di comando oppure modificando dei file di testo. Quando ci si avvicina ad Apache per la prima volta, tutto questo può essere poco chiaro.

Componenti del framework: middleware

PHP appartiene ad una classe di linguaggi noti come middleware. Questi linguaggi lavorano a stretto contatto con il server Web per interpretare le richieste fatte da Internet, elaborarle e per interagire con altri programmi sul server, per soddisfare le richieste, e per specificare al server Web cosa fornire al browser del client.

Oltre a PHP ci sono molti altri linguaggi che eseguono funzioni simili. Alcune delle scelte più diffuse sono JSP, ASP, Perl e ColdFusion.

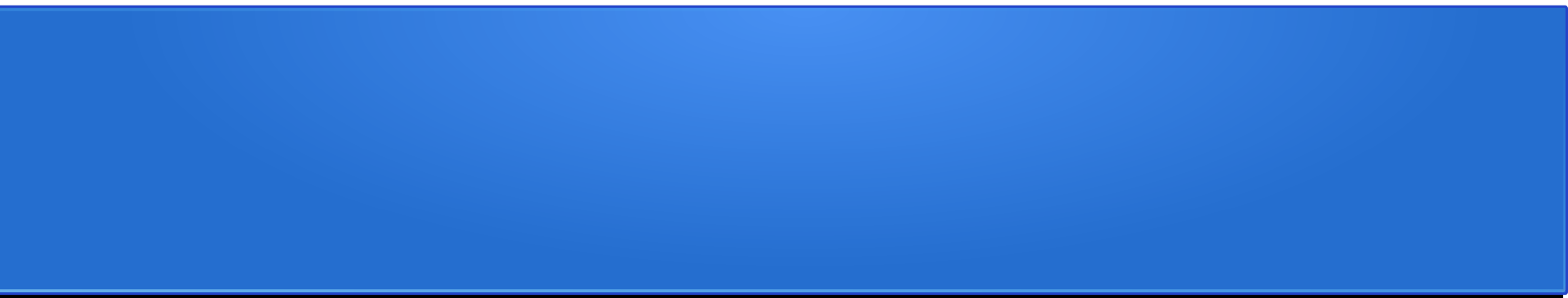
PHP nasce nel 1994 la cui funzione era quella di facilitare ai programmatori l'amministrazione delle homepage personali: da qui trae origine il suo nome, che allora significava appunto Personal Home Page. Oggi, invece, è conosciuto come PHP: Hypertext Pre Processor, ed è un linguaggio completo di scripting, sofisticato e flessibile, che può girare praticamente su qualsiasi server Web, e su qualsiasi sistema operativo (Windows o Unix/Linux, ma anche MacOS, AS/400, Novell, OS/2 e altri), e consente di interagire praticamente con qualsiasi tipo di database (MySQL, PostgreSQL, Sql Server, Oracle, SyBase, Access e altri). Si può utilizzare per i più svariati tipi di progetti, dalla semplice home page dinamica fino al grande portale o al sito di e-commerce.

Il framework chiavi in mano: XAMPP o WAMP

Uno strumento di interesse anche professionale ma che sicuramente risulta estremamente valida per la didattica dei database è la suite open source XAMPP. Questo strumento integra i software MySQL e Apache + modulo PHP integrato, con tanto di installazione sotto MS Windows.

È sufficiente installare e lanciare XAMPP, installare le proprie pagine html nella directory www della dir di installazione del software, ed è subito possibile consultarle connettendosi da un qualunque browser web all'indirizzo IP del computer, se in remoto, o a 127.0.0.1(localhost) se in locale per consultarle

Esistono anche altre suite complete che integrano i sopracitati moduli ad es. WAMP



Framework per Applicazioni Client-Server (DB con Interfaccia Web-based)

Prima applicazione

Per incentivare immediatamente l'apprendimento è utile partire da subito con una prima applicazione completa.

Probabilmente risulta difficile generalizzare immediatamente la comprensione di tutti gli strumenti/linguaggi che compongono l'applicazione, ma maneggiare immediatamente una applicazione funzionante verificandone la ridottissima complessità può fortemente incentivare lo sforzo necessario per approfondire le conoscenze.

L'applicazione rappresenta un semplice libro degli ospiti (guestbook) con la possibilità di inserire i propri dati da semplice interfaccia web e di visualizzare i dati di tutti gli ospiti presenti.

Prima applicazione

Vediamo le specifiche dettagliate:

1. E' richiesta la realizzazione di un DB con interfaccia web che tenga conto dei dati di tutti i visitatori che si registrano nel sito. I dati richiesti sono i seguenti: name, location, email, homepage URL, comments, questo campo prevede la raccolta di eventuali commenti da parte degli ospiti.
2. Vanno definiti 3 script SQL: uno per la creazione dello schema del db, che non prevede interfaccia WEB, e altri 2, con interfaccia WEB, per:
 - Inserimento dati (Sign my guestbook)
 - Visualizzazione dei dati di tutti gli ospiti presenti (View my guestbook)

Introduzione

L'applicazione verrà creata nella directory guestbook nella cartella www della installazione di EasyPHP. Oppure nella cartella htdocs di XAMPP. Affinchè possa funzionare è necessario anzitutto creare il database chiamato guestbook.

Una volta avviato EasyPHP è quindi possibile utilizzare l'applicazione connettendosi all'indirizzo 127.0.0.1/guestbook.

Spieghiamo i vari componenti dell'applicazione nelle slide seguenti.

Lo script guestbook.sql

```
create database guestbook;  
use guestbook;  
drop table if exists guestbook;  
create table guestbook(  
    Id int auto_increment primary key,  
    name varchar(40) null,  
    location varchar(40) null,  
    email varchar(40) null,  
    url varchar(40) null,  
    comments text null);
```

Questo script sql va eseguito dalla linea di comando di mysql o dall'opportuno form di phpMyAdmin (sezione SQL).

Provvede a creare lo schema del DB, consistente in una sola tabella.

A questo punto si ha un database chiamato guestbook e una tabella all'interno del database chiamata guestbook

La pagina index.html

```
<html><head></head><body>  
<a href="sign.php">Sign my guestbook!</a>  
<br><a href="view.php">Read my guestbook!</a>  
</body></html>
```

Questo codice HTML corrisponde alla pagina index.html, che rappresenta la pagina iniziale dell'applicazione.

Senza entrare nei dettagli dell'html, descrive 2 linee le quali rappresentano 2 link corrispondenti ciascuno ad uno script correlato a una delle 2 interazioni (inserimento dati e visualizzazione dati presenti) con il database richieste dalle specifiche.

Generalmente questi script hanno una estensione .php, anche se è possibile fare interpretare a PHP qualsiasi cosa, compresi i file html. Gli script e il file index.html saranno contenuti all'interno della cartella raggiungibile dalla pagina di default di XAMPP, quindi in una sottodirectory della directory htdocs della directory di installazione di XAMPP.

Sintassi di base PHP

Un vantaggio di PHP è la possibilità di spostarsi tra HTML e i comandi che fanno parte del linguaggio di programmazione PHP. Funziona in questo modo: le sezioni dello script comprese tra un tag `<?php` e `?>` saranno interpretate dal motore di PHP.

Ad esempio, provate a scrivere un file `provaPhp1.php`, a copiarlo nella dir `www` o `htdocs` dell'installazione di php e a connettervi utilizzando l'indirizzo `127.0.0.1/provaPhp1.php`

```
<?php  
echo "Ciao, ";  
?>  
mamma.
```

Quando lo script viene eseguito dal server Web, questa crea una pagina Web che semplicemente stampa "Ciao, mamma". Il comando `echo` di PHP gestisce la prima parte della riga. Naturalmente PHP può fare delle cose più complesse.

Sintassi di base PHP

Come ogni altro linguaggio di può lavorare con le variabili e prendere delle decisioni. Vediamo un esempio più complesso. Codificatolo nel file provaPhp2.php, copiatelo nella dir dell'installazione di php e connettetevi utilizzando l'indirizzo localhost/provaPhp2.php

```
<?php  
echo "Ciao, Mamma";  
$var=date("H");  
if ($var <= 11){ echo "Buongiorno"; }  
elseif ($var > 11 and $var < 18 ){  
    echo "Buon Pomeriggio"; }  
    else{echo "Buona Sera"; }  
?>
```

In questo script, dopo aver stampato il saluto, c'è la vera programmazione. È stata usata la funzione integrata date di PHP per estrarre l'ora del giorno in un formato a 24 ore. Il valore è immediatamente assegnato alla variabile chiamata \$var, Poi viene eseguita una scelta e viene stampato il testo appropriato a seconda dell'ora del giorno. Nell'istruzione if, le parentesi graffe contengono i comandi da eseguire a seconda della condizione, e la condizione stessa è racchiusa tra parentesi.

Le funzioni date() e echo, sono solo 2 delle centinaia di funzioni integrate in PHP

Lo script dbconnect.php

Questo script è il seguente:

```
<?php  
if (!($mylink = mysqli_connect('localhost','root',' ','guestbook'))  
{  
print "<h3>could not connect to database</h3>\n";  
exit;  
}  
?>
```

La istruzione **mysqli_connect**, ha sostituito `mysql_connect`, indica a MySQL che il server Web (l'entità che esegue lo script) si trova sulla macchina locale, ha un nome utente uguale a `root` e non ha una password. e viene selezionato il database `guestbook`

Lo script sign.php

Prima di analizzare nel dettaglio lo script che esegue l'operazione di inserimento dati nel DB bisogna sapere qualcosa di più sulle variabili in PHP. Prima abbiamo visto come creare una variabile all'interno di uno script PHP, ma dato che siamo in un ambiente client/server, bisogna ottenere dei dati per le variabili provenienti dal client (il browser Web) e portarle in PHP. Solitamente questo viene fatto con i moduli HTML.

Nel seguito vedremo una spiegazione di base dei moduli HTML. Per ora, ci interessa solo sapere che ogni elemento del modulo ha un nome, e, quando il modulo viene inviato, i nomi di questi elementi del modulo diventano disponibili come variabili nello script PHP a cui il modulo è stato inviato. Vediamo un esempio.

Con il modulo che segue, quando questo viene inviato, le variabili \$surname e \$submit saranno disponibili nello script PHP in myscript.php. Il valore di \$surname conterrà quello che l'utente ha inserito nel campo di testo. Il valore di \$submit sarà la stringa di testo "submit".

```
<form action="myscript.php">  
<input type="text" name="surname">  
<input type="submit" name="submit">
```

Lo script sign.php

```
<html><head></head>
<?php include("dbconnect.php"); ?>//collega a script dbconnect per connessione a DB
<body><h2>Sign my Guest Book!!!</h2>
<form method="post" action="create_entry.php">
<label>Name:<input type="text" size="40" name="name" placeholder="nome"></label><br>
<label>Location:<input type="text" size="40" name="location" placeholder="località"></label><br>
<label>Email:<input type="email" name="email" autocomplete="on" placeholder="email@domain.ext
"></label><br>
<label>Home Page URL:<input type="url" name="url" autocomplete="on" placeholder=
"http://mywebsite.com"></label><br>
<label>Comments:<textarea name="comments" cols="40" rows="4" wrap="virtual" > </textarea>
</label> <br>
<label>data:<input type="datetime" name="mydatetime"></label><br>
<input type="submit" name="submit" value="Sign!">
<input type="reset" name="reset" value="Start Over">
</form> </body></html>
```

Quando l'utente completa questo modulo e lo invia (pulsante submit), le informazioni sono inviate a create_entry.php. Col pulsante reset il modulo viene resettato.

Lo script create_entry.php

Lo script create_entry.php anzitutto richiama lo script db_connect.php che esegue la connessione al db, quindi assegna i valori delle variabili provenienti dalla form html alle corrispondenti variabili da utilizzare per inserire i dati nel db, quindi, dopo aver verificato che il pulsante digitato è "Sign", svolge una query sul db (mysqli_query) inserendo i dati nella tabella opportuna, quindi ringrazia

Nel caso in cui la condizione non sia verificata riesegue lo script sign.php.

Lo script create_entry.php

```
<html><head></head><?php include("dbconnect.php");
$submit = $_POST['submit'];
$location = $_POST['location'];
$name = $_POST['name'];
$url = $_POST['url'];
$email = $_POST['email'];
$comments = $_POST['comments'];
$mydatetime=$_POST['mydatetime'];
if ($submit == "Sign!") {
$query = "insert into guestbook (name,location,email,url,comments) values ('$name', '$location', '$email',
'$url', '$comments')";
mysqli_query($mylink,$query); // $mylink è la variabile di connessione del file dbconnect.php
?>
<body><h2>Thanks!!</h2>
<h2><a href="view.php">View My Guest Book!!!</a></h2>
<?php
}
else{ include("sign.php");}
?></form> </body></html>
```

Lo script view.php

Questo script ha lo scopo di visualizzare tutte le informazioni di ciascun ospite del DB, usando semplici istruzioni di stampa di ciascuno degli attributi di ciascuna riga della tabella.

La query iniziale stampa una serie di righe che vengono scandite tramite l'istruzione `mysql_fetch_array`.

Lo script view.php

```
<html><head></head><?php include("dbconnect.php"); ?>
<body><h2>View My Guest Book!!</h2>
<?php
$result = mysqli_query($mylink,"select * from guestbook");
if ($result) {
    while ($row = mysqli_fetch_array($result)) {
        print "<b>Name:</b>"; print $row["name"]; print "<br>\n";
        print "<b>Location:</b>"; print $row["location"]; print "<br>\n";
        print "<b>Email:</b>"; print $row["email"]; print "<br>\n";
        print "<b>URL:</b>"; print $row["url"]; print "<br>\n";
        print "<b>Data:</b>"; print $row["mydatetime"]; print "<br>\n";
        print "<b>Comments:</b>"; print $row["comments"]; print "<br>\n";
        print "<br>\n<br>\n";
    }
    mysqli_free_result($result);
}
?>
<h2><a href="sign.php">Sign My Guest Book!!</a></h2> </body></html>
```

Esercitazione 1

Costruire una nuova applicazione, da salvare nella directory `guestbook-e1` che abbia le stesse caratteristiche della applicazione iniziale di esempio, in più aggiunga 2 query:

- Una query che visualizzi in uscita (su interfaccia web) il numero attuale di ospiti presenti nel db
- Una query che visualizzi il numero di ospiti per i quali la location è "Cagliari".

Esercitazione 2

Costruire una nuova applicazione, da salvare nella directory `guestbook-e2` che, a partire dalle caratteristiche della applicazione iniziale di esempio, aggiunga:

- La possibilità di associare a ciascun utente un tipo predefinito, da scegliere fra:
 - Guest
 - Staff
 - Owner
- Una query che visualizzi il numero di ospiti suddivisi per categoria.
- Una query che visualizzi solo il numero di ospiti di una specifica categoria (dando la possibilità di scegliere con una opzione)
- Una query che visualizzi il numero di ospiti per i quali la location è "Cagliari" suddivisi per categoria.

Specifica: Usare una nuova tabella per codificare le 3 tipologie