

Simulazione seconda prova Esame di Stato

Sito Web - Gestione Asl: alternanza scuola lavoro

(Parte seconda)

Il sistema informativo deve essere realizzato in un computer centrale (il server di database) al quale possono accedere i diversi terminali o PC (i computer client) installati presso gli uffici dell'intranet dell'istituto. Tutti i servizi del sistema informativo devono essere resi disponibili nel sito Web dell'intranet dell'istituto, in modo che il personale della scuola possa eseguire, in tempo reale, l'aggiornamento dei periodi di tirocinio e le operazioni sui dati disponibili.

Competenze verificate

- Progettare la struttura logica e fisica dei file di un sito Web
- Codificare interrogazioni SQL su un database in pagine ASP
- Costruire applicazioni Web interattive mediante pagine ASP

Volumi di riferimento

Teoria e laboratorio

ICT Siti Web e soluzioni per l'e-commerce
 Modulo 1 Progetto di siti Web - UD 3 Ambienti software autore
 Modulo 2 Soluzioni per l'e-commerce – UD 2 Sviluppo di applicazioni Web

Riflessioni per il docente

Il metodo di lavoro proposto ripercorre i passi per il progetto e la realizzazione di una soluzione per l'e-commerce, basata sulle pagine ASP il cui script è stato sviluppato con il linguaggio VBScript.

Il sito Web è stato realizzato in FrontPage, per fornire la possibilità a tutti gli allievi di collaudare il sistema proposto. In allegato a questa prova è quindi possibile effettuare il download dell'intero sito Web, memorizzato nella cartella *asl*. Il sito Web è completo e funzionante per cui può essere collaudato pubblicandolo su un server Web di Internet oppure dell'intranet del laboratorio del proprio Istituto. In alternativa, il collaudo può essere realizzato su un qualsiasi computer stand-alone in cui è stato installato il programma di sistema che lo trasforma in un server Web. Esempi di programmi per creare un server Web sono: IIS (*Internet Information Services*) in un computer Windows NT/2000/XP, PWS (*Personal Web Server*) in un sistema Windows 95/98/Me e Apache in un sistema Linux. Le pagine ASP del sito accedono a un file di database di Access, per cui il server di distribuzione deve anche disporre dei driver per questo RDBMS.

Per collaudare il sito Web possiamo seguire le tre fasi successive.

1. Pubblicare la cartella *asl* effettuando l'upload su un server Web con una connessione HTTP (se il server Web dispone delle estensioni di FrontPage) oppure FTP. Nel caso di un server Web realizzato con un computer stand-alone, la pubblicazione può essere realizzata creando un nuovo sito Web vuoto di nome *http://NomeComputer/asl* oppure *http://localhost/asl* e importando (in FrontPage: comando *Importa...* del menu *File*) tutto il contenuto della cartella *asl* (file e singole cartelle) in quella del nuovo sito Web.

Dopo aver svolto una delle operazioni precedenti, la home page del sito Web avrà come URL il seguente:

<http://NomeServerWeb/asl/default.htm>

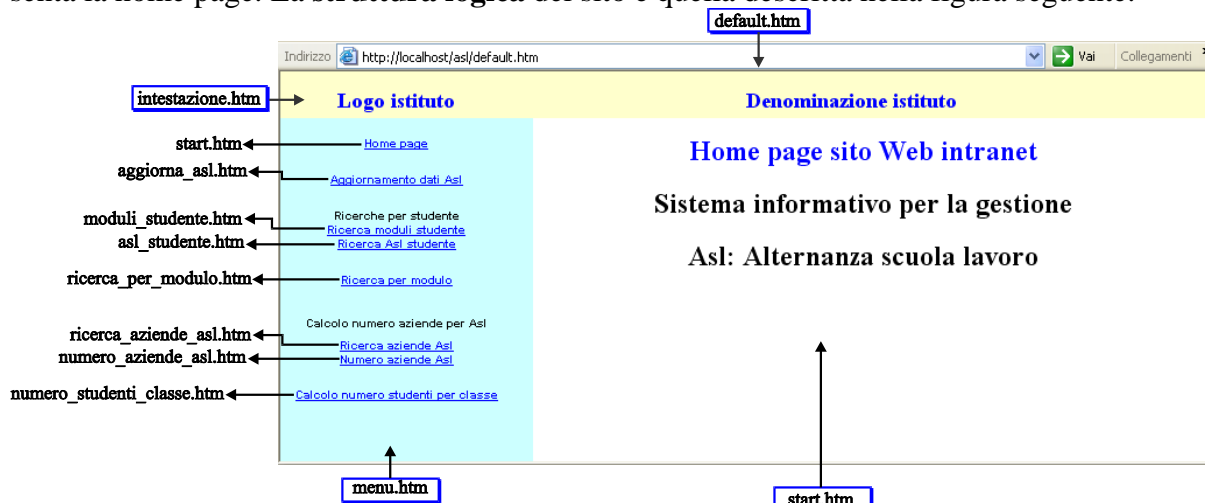
2. Aggiungere alla cartella del sito Web con il database i permessi di scrittura necessari per permettere di effettuare operazioni di aggiornamento dei dati.

Ad esempio, se si pubblica un sito su un server Web che dispone delle estensioni di FrontPage, è sufficiente accettare la creazione guidata della connessione al file di database proposta dal server. In questo caso, le estensioni di FrontPage memorizzano il file di database in una cartella di nome *fpdb* che dispone automaticamente dei permessi sia di lettura sia di scrittura sul database stesso. In alternativa, per aggiungere il permesso di scrittura, si deve accedere (se si ha accesso al server di distribuzione) alla cartella del sito Web con il file di database e modificare manualmente i permessi mediante le funzioni locali del file system del sistema operativo.

Il sito Web è stato realizzato in FrontPage 2003 ed è stato collaudato in un computer Windows (di nome *pc*) sia 2000 sia XP con installato il servizio IIS.

Creazione del sito Web

Il sito è basato su una struttura a frame definita mediante il frameset *default.htm*, che rappresenta la home page. La **struttura logica** del sito è quella descritta nella figura seguente.



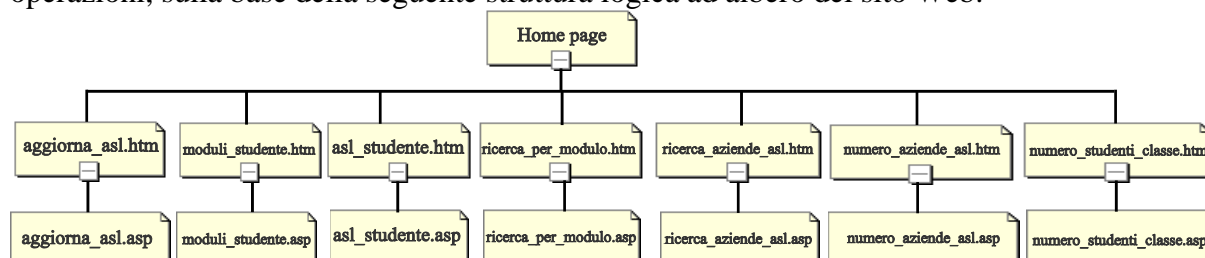
Il frameset è formato da tre frame con le seguenti funzioni:

- il frame superiore (*intestazione.htm*) è fisso e contiene il logo e la denominazione dell'istituto;
- il frame inferiore a sinistra (*menu.htm*) contiene il menu i cui hyperlink richiamano le operazioni del sistema informativo;
- la cornice inferiore a destra visualizza i risultati ottenuti dall'esecuzione delle operazioni.

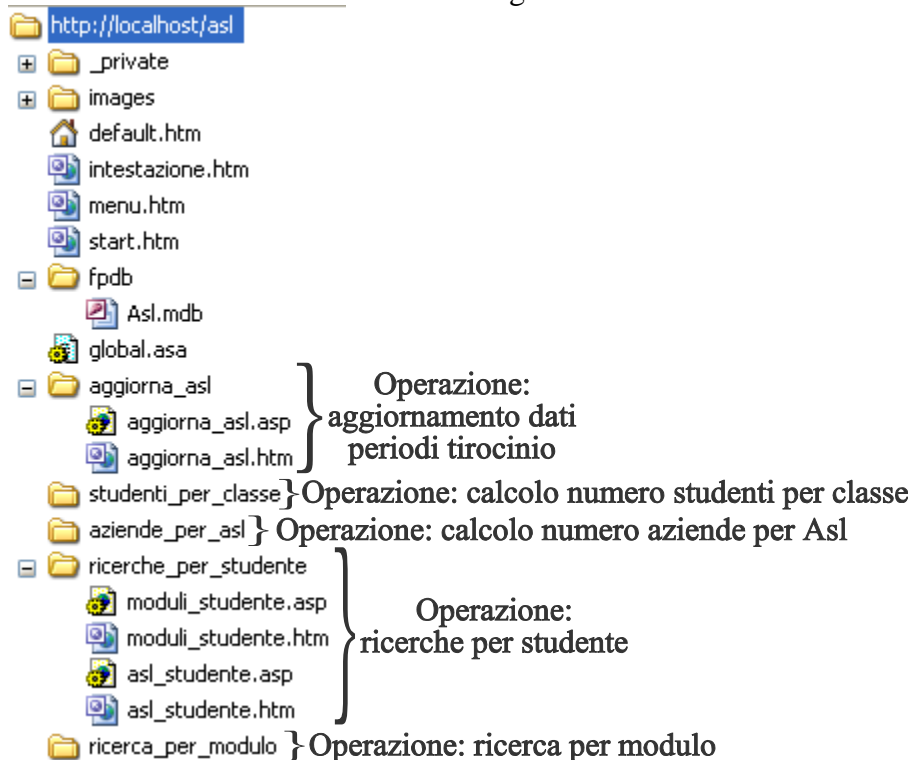
Il codice sorgente HTML del frameset che descrive la struttura in frame della pagina *default.htm* è quindi il seguente.

```
<FRAMESET ROWS="16%,*" FRAMESPACING="0" BORDER="0" FRAMEBORDER="0" >
  <FRAME NAME="intestazione" SCROLLING="NO" NORESIZE TARGET="sommario"
    SRC="sommario.htm">
  <FRAMESET COLS="24%,*" >
    <FRAME NAME="sommario" TARGET="principale" SRC="menu.htm"
      SCROLLING="AUTO" NORESIZE>
    <FRAME NAME="principale" SRC="start.htm" SCROLLING="AUTO" >
  </FRAMESET>
</FRAMESET>
```

Il frame *menu.htm* contiene una barra di hyperlink che richiamano le pagine per l'avvio delle operazioni, sulla base della seguente struttura logica ad albero del sito Web.



La **struttura fisica** del sito Web è illustrata nella figura successiva.



Per realizzare l'applicazione distribuita client-server si è utilizzata la tecnologia software ASP (*Active Server Pages*), utilizzando come linguaggio di programmazione di script il VBScript. Nel seguito, per semplicità, realizzeremo soltanto l'operazione di aggiornamento dei dati dei periodi Asl e il primo servizio del sistema informativo.

Progetto e realizzazione della base di dati

Il database è quello realizzato nel RDBMS Access nella prima parte dell'esercitazione. Per rendere disponibile il database, dobbiamo quindi pubblicare il file *Asl.mdb*, ad esempio, nella sottocartella *fpdb* del sito Web *asl*.

Impostazione della connessione del sito Web al database

Per l'accesso a una base di dati, il VBScript utilizza la tecnologia orientata agli oggetti ADO e le interfacce software basate sugli standard OLE DB e ODBC. Per collegare una pagina ASP con il database dobbiamo definire una nuova connessione logica, ad esempio di nome *ConnessioneDB*, usando le seguenti istruzioni VBScript.

```
Dim ConnessioneDB
Set ConnessioneDB = Server.CreateObject("ADODB.Connection")
```

Nel modello ADO, la connessione *ConnessioneDB* è un oggetto della classe *Connection*. Se seguiamo lo standard OLE DB, l'oggetto *ConnessioneDB* deve essere collegato al file fisico di database mediante il metodo *Open* (della classe *Connection*) e una stringa di connessione, con una sintassi del tipo che segue.

```
ConnessioneDB.Open StringaConnessione
```

Nel nostro progetto, con pagine ASP in sottocartelle della radice (*root*) del sito Web, la stringa di connessione per il RDBMS Access assume quindi una forma del tipo presentato nel seguito.

```
StringaConnessione = "DRIVER={Microsoft Access Driver (*.mdb)}; " &_
    "DBQ=" & Server.MapPath("../fpdb/DBAsl.mdb")
```

Il metodo **MapPath** dell'oggetto predefinito *Server* usato con la sintassi seguente, restituisce il percorso fisico del server Web dove è effettivamente memorizzato il file con il database.

```
Server.MapPath("URL Relativo File")
```

Utilizzando *Server.MapPath* nella stringa di connessione è quindi possibile rendere indipendente la connessione logica a un file di database dalla sua effettiva posizione sulle memorie di massa del server Web di distribuzione.

Nell'URL relativo per il file di database si sono usati i simboli ".." per individuare la radice del sito Web da una sua sottocartella. Le nuove versioni del software di sistema (quale IIS 6), che realizzano il servizio WWW, non supportano i simboli ".." che sono sostituiti da uno slash "/" per indicare la radice del sito. Nel caso di questi server Web, la stringa di connessione diventa la seguente.

```
StringaConnessione = "DRIVER={Microsoft Access Driver (*.mdb)};" &_  
"DBQ=" & Server.MapPath("/fpdb/DBAs1.mdb")
```

Nell'applicazione Web, in fase di realizzazione, useremo una delle stringhe di connessione precedenti (sulla base della versione del software di sistema del server), in modo che il sito possa essere pubblicato su qualsiasi server Web di distribuzione senza effettuare alcuna modifica al codice sorgente dello script.

Dichiarazione della stringa di connessione al database come variabile globale

La stringa di connessione può essere dichiarata come variabile globale in modo che sia visibile durante una sessione HTTP di un utente all'applicazione Web. Per questo obiettivo è sufficiente creare un file di testo di nome *global.asa*, nella radice del sito Web, e dichiarare la variabile *StringaConnessione* nel corpo dell'evento *Session_OnStart()* con la seguente sintassi.

```
global.asa
```

```
<SCRIPT LANGUAGE="VBScript" RUNAT="SERVER">  
Sub Session_OnStart()  
    Dim StringaConnessione  
    Session("StringaConnessione")="DRIVER={Microsoft Access Driver (*.mdb)};" &_  
        "DBQ=" & Server.MapPath("../fpdb/DBAs1.mdb")  
End Sub  
</SCRIPT>
```

Il file *global.asa* può essere già presente nel sito Web; in questo caso, è sufficiente modificare il contenuto aggiungendo, nella procedura di evento *Session_OnStart()*, la dichiarazione della variabile globale *StringaConnessione*.

Una variabile globale è visibile in tutte le pagine del sito Web relative a una sessione, utilizzando l'identificatore: *Session("VariabileGlobale")*.

Nel nostro esempio, dopo avere dichiarato la variabile globale *StringaConnessione*, per creare la connessione con il file di database possiamo impiegare, in tutte le pagine *.asp* di accesso ai dati, la seguente istruzione:

```
ConnessioneDB.Open Session("StringaConnessione")
```

L'uso di una variabile globale può essere utile nel caso di applicazioni Web con molte pagine di accesso ai dati, infatti, nel caso di modifica del percorso del file di database sul server Web, è sufficiente riscrivere la stringa di connessione una sola volta, nel file *global.asa*, e non in tutte le pagine *.asp* che impiegano la connessione stessa.

Sviluppo dell'applicazione client-server distribuita

Nel seguito realizzeremo l'operazione di aggiornamento dei dati dei periodi Asl e il primo servizio del sistema informativo. Gli algoritmi realizzati nelle altre operazioni sono analoghi a quelli che descriveremo.

Operazione: aggiornamento dei dati dei periodi di tirocinio Asl

Per realizzare questa operazione, possiamo progettare il form *aggiorna_asl.htm*, che richiama la pagina *aggiorna_asl.asp*, che esegue le operazioni di aggiornamento sui dati.

La pagina HTML *aggiorna_asl.htm*, richiamata nel menu del sito Web con l'hyperlink [Aggiornamento dati Asl](#), contiene (nel suo BODY) un modulo (vedi figura e codice HTML seguenti) per la raccolta dei parametri per l'esecuzione dell'operazione di aggiornamento.

```
<FORM ACTION="aggiorna_asl.asp" METHOD="POST">
  <FIELDSET><LEGEND><B>Aggiornamento dati periodi di tirocinio Asl</B></LEGEND>
  Codice azienda <INPUT TYPE="TEXT" NAME="txtAzienda" SIZE="7" VALUE="A-">
  Codice studente <INPUT TYPE="TEXT" NAME="txtStucente" SIZE="7" VALUE="S-">
  Codice docente <INPUT TYPE="TEXT" NAME="txtDocente" SIZE="7" VALUE="D-"><BR>
  Data iniziale (mm/gg/aaaa) <INPUT TYPE="TEXT" NAME="txtDataIniziale" SIZE="10">
  Numero di giorni <INPUT TYPE="TEXT" NAME="txtGiorni" SIZE="2" VALUE="0">
  <BR><BR><INPUT TYPE="SUBMIT" VALUE="Invia" NAME="cmdInvia">
  <INPUT TYPE="RESET" VALUE="Reimposta" NAME="cmdReimposta">
</FIELDSET>
</FORM>
```

Il form precedente per l'input dei parametri del servizio

```
<FORM ACTION="aggiorna_asl.asp" METHOD="POST">
```

richiama la pagina *aggiorna_asl.asp*, il cui script (riportato nel seguito) esegue l'operazione di inserimento di una nuova riga nella tabella *Asl* del database.

```
aggiorna_asl/aggiorna_asl.asp
```

```
<%@ Language=VBScript %>
<% Response.Buffer=True %>
<%Dim ConnessioneDB, rsTabella, QueryINSERT
Set ConnessioneDB = Server.CreateObject("ADODB.Connection")
ConnessioneDB.Open Session("StringaConnessione")

Dim Azienda, Studente, Docente, DataIniziale, Giorni
Azienda = Request.Form("txtAzienda")
Studente = Request.Form("txtStucente")
Docente = Request.Form("txtDocente")
DataIniziale = Request.Form("txtDataIniziale")
Giorni = Request.Form("txtGiorni")

QueryINSERT="INSERT INTO Asl(Azienda, Studente, Docente, " &_
              "DataIniziale, Giorni) " &_
              " VALUES ('" & Azienda & "', '" & Studente & "', '" &_
              Docente & "', '" & DataIniziale & "', '" & Giorni & ');"

Dim RecordInseriti
On Error Resume Next
ConnessioneDB.Execute QueryINSERT, RecordInseriti
%>
```

```

<HTML>
  <HEAD><TITLE>Aggiornamento dati Asl</TITLE></HEAD>
<BODY>
  <% If RecordInseriti=1 Then
    Response.Write "<H3>Operazione effettuata in modo corretto.</H3>"
  Else
    Response.Write "<H3>Errore nell'esecuzione dell'operazione.</H3>"
    Response.Write "Verificare se i tutti i codici digitati sono corretti."
  End If
  %>
</BODY>
<%
  ConnessioneDB.Close
  Set ConnessioneDB = Nothing
  Session.Abandon
%>

```

Operazione: ricerca moduli studente

Per realizzare questa operazione, possiamo progettare la pagina HTML *moduli_studente.htm*, richiamata nel menu del sito Web con l'hyperlink [Ricerca moduli studente](#), che contiene (nel suo BODY) un modulo (vedi figura e codice HTML successivi) per la raccolta dei parametri per l'esecuzione del servizio.

Servizio: ricerca dei moduli scelti da uno studente

```

<FORM ACTION="moduli_studente.asp" METHOD="POST">
  <FIELDSET><LEGEND><B>Servizio: ricerca dei moduli scelti da uno studente
  </B></LEGEND>
  Cognome studente <INPUT TYPE="TEXT" NAME="txtCognome" SIZE="20">
  Nome studente <INPUT TYPE="TEXT" NAME="txtNome" SIZE="20"><BR><BR>
  <INPUT TYPE="SUBMIT" VALUE="Invia" NAME="cmdInvia">
  <INPUT TYPE="RESET" VALUE="Reimposta" NAME="cmdReimposta">
</FIELDSET>
</FORM>

```

Il comando del form per l'input dei parametri del servizio

```
<FORM ACTION="moduli_studente.asp" METHOD="POST">
```

richiama la pagina *moduli_studente.asp*, il cui script (riportato nel seguito) esegue la query sul database e presenta una tabella con i risultati visualizzati in una pagina HTML dinamica di risposta. Usando il valore "POST" per l'attributo METHOD del tag doppio FORM, i dati del modulo sono inviati alla pagina *.asp* all'interno del corpo di un messaggio HTTP.

```
ricerche_per_studente/moduli_studente.asp
```

```

<%@ Language=VBScript %>
<% Response.Buffer=True %>

<HTML>
<HEAD><TITLE>Ricerca dei moduli scelti da uno studente </TITLE></HEAD>
<BODY>
<H3>Elenco dei moduli scelti da uno studente</H3>
<% Dim ConnessioneDB, rsTabella, Query, Cognome, Nome
  Set ConnessioneDB = Server.CreateObject("ADODB.Connection")
  ConnessioneDB.Open Session("StringaConnessione")

  Cognome = Request.Form("txtCognome")
  Nome = Request.Form("txtNome")

```


ricerche_per_studente/asl_studente.asp

```

<%@ Language=VBScript %>
<% Response.Buffer=True %>

<HTML>
<HEAD><TITLE>Ricerca dei periodi di tirocinio di uno studente </TITLE></HEAD>
<BODY>
<H3>Elenco dei periodi di tirocinio di uno studente</H3>
<% Dim ConnessioneDB, rsTabella, Query, Cognome, Nome
   Set ConnessioneDB = Server.CreateObject("ADODB.Connection")
   ConnessioneDB.Open Session("StringaConnessione")

   Cognome = Request.Form("txtCognome")
   Nome = Request.Form("txtNome")

   Query="SELECT Asl.* " &_
         " FROM Studenti INNER JOIN Asl ON Studenti.Codice = Asl.Studente" &_
         " WHERE Studenti.Cognome = '" & Cognome & "' AND " &_
         " Studenti.Nome = '" & Nome & "';"

   Set rsTabella = ConnessioneDB.Execute(Query)%>
<% If rsTabella.BOF=True And rsTabella.EOF=True Then %>
   <H4>Lo studente <%=Cognome%> e <%=Nome%> non ha periodi Asl </H4>
<% Else %>
   <TABLE BORDER="1">
   <TR><TH>Codice Asl</TH><TH>Codice azienda</TH><TH>Codice studente</TH>
   <TH>Codice docente</TH><TH>Data iniziale</TH><TH>Giorni</TH>
   </TR>
   <% Do until rsTabella.EOF=True %>
   <TR>
   <TD ALIGN="CENTER"><%=rsTabella.Fields("ID")%></TD>
   <TD ALIGN="CENTER"><%=rsTabella.Fields("Azienda")%></TD>
   <TD ALIGN="CENTER"><%=rsTabella.Fields("Studente")%></TD>
   <TD ALIGN="CENTER"><%=rsTabella.Fields("Docente")%></TD>
   <TD ALIGN="CENTER"><%=rsTabella.Fields("DataIniziale")%></TD>
   <TD ALIGN="CENTER"><%=rsTabella.Fields("Giorni")%></TD>
   </TR>
   <% rsTabella.MoveNext %>
   <% Loop %>
   </TABLE>
<% End If %>

<% rsTabella.Close : Set rsTabella = Nothing
   ConnessioneDB.Close : Set ConnessioneDB = Nothing
   Session.Abandon %>
</BODY>
</HTML>

```