

UNIVERSITÀ DEGLI STUDI DI BOLOGNA

FACOLTÀ DI INGEGNERIA
Corso di Laurea in Ingegneria Informatica

Tecnologie Microsoft per Sistemi Server Side

Relazione di tirocinio di:
Jonathan Cristoforetti

Responsabile Accademico:
Chiar.mo Prof. Ing. Antonio Corradi

Responsabile aziendale:
Ing. Michele Callegari

Anno Accademico 2001-2002

INDICE

INTRODUZIONE.....	2
CAPITOLO 1 L'AZIENDA.....	4
CAPITOLO 2 ACTIVE SERVER PAGES (ASP)	6
2.1 TECNOLOGIE WEB E ASP	6
2.2 STRUTTURA DI UNA PAGINA ASP.....	6
2.3 OGGETTI BUILT-IN E I COMPONENTI COM	7
2.4 ASP E DATABASE	7
2.5 L'OGGETTO RECORDSET	8
CAPITOLO 3 IL PRIMO PERIODO.....	12
3.1 GLI STRUMENTI E L'AMBIENTE DI SVILUPPO.....	12
3.2 IL SITO AFFITTITURISTICI.IT.....	13
3.3 ENDURANCE GESTIONALE.....	14
CAPITOLO 4 IL MODULO BANNER.....	16
4.1 LE SPECIFICHE.....	16
4.2 IL MODULO ADROTATOR.....	17
4.3 L'APPLET DYNAMIC BILLBOARD.....	18
4.4 STRUTTURA DEL DATABASE.....	19
4.5 LA STRUTTURA DELLE PAGINE.....	20
4.6 IL LOG DI CLICK E IMPRESSION.....	23
CAPITOLO 5 UPLOAD DIIMMAGINI PER IL SITO MINARDI.....	26
5.1 REQUISITI.....	26
5.2 LA SCELTA DI UTILIZZARE ASP.NET	26
5.3 UN PRIMO PROTOTIPO.....	27
5.4 IL PROBLEMA DEL TEMPO.....	27
5.5 LA SOLUZIONE.....	28
CAPITOLO 6 IL MODULO SONDAGGIO	30
6.1 LE SPECIFICHE.....	30
6.2 LA STRUTTURA DEL DATABASE	30
6.3 LA STRUTTURA DELLE PAGINE.....	31
CAPITOLO 7 RICERCA E SPERIMENTAZIONI CON ASP.NET	34
7.1 MICROSOFT .NET	34
7.2 NOVITÀ IN ASP.NET	35
7.3 L'ACCESSO AI DATI IN ASP.NET: ADO.NET	35
7.4 LA VISUALIZZAZIONE DEI DATI: DATAGRID, DATALIST E REPEATER.....	37
7.5 SPECIALIZZAZIONE DELL'CONTROLLO DATALIST: IL CONTROLLO DATALIST NAV.....	38
7.6 DATALIST E DATAGRID INNESTATE.....	39
CONCLUSIONE.....	42

Introduzione

Una delle maggiori critiche poste al sistema universitario italiano è quella di fornire un'istruzione sì d'alto livello, ma di tipo prettamente teorico. Il risultato è che i laureati italiani si ritrovano con un'ottima preparazione, ma spesso in difficoltà al momento di mettere in pratica le conoscenze acquisite.

Al momento di decidere il mio piano di studio per il terzo anno, vedendo tra le altre varie la possibilità di svolgere un'attività pratica presso un'azienda, ho subito pensato che fosse un'ottima possibilità per ovviare a questa "mancanza" dell'istruzione universitaria.

Il tirocinio è stato svolto presso l'azienda Endurance di Bologna per un periodo di circa tre mesi. Durante lo svolgimento si è lavorato allo sviluppo di pagine web basate su tecnologia ASP e ASP.NET di Microsoft. Durante tutto lo svolgimento del tirocinio mi ha seguito l'ing. Michele Callegari.

Capitolo 1

L'Azienda

Endurance è un'azienda di medio-piccole dimensioni che ha sede a Bologna. L'attività principale consiste nello sviluppo di siti e applicazioni web, ma l'azienda si occupa anche di formazione, progettazione di reti aziendali, sviluppo di soluzioni per l'e-commerce e per l'e-government.



I progetti realizzati sono basati principalmente su tecnologie Microsoft: ASP per lo scripting server-side, IIS come web-server e MS-SQL Server come RDBMS. Sono tuttavia utilizzati anche sistemi Open Source e Unix come PHP, MySQL, Oracle.

Tra i principali clienti dall'azienda vi sono il Team di F1 Minardi (www.minardi.it), Marzocchi Group (www.marzocchi.com), Borse.it (www.borse.it) e il gruppo Telecomunicazioni del DEIS (www-tlc.deis.unibo.it).

Il primo contatto con l'azienda è avvenuto intorno alla metà di Febbraio, prima telefonicamente, poi tramite un colloquio in azienda. Durante il primo incontro si è fatta la conoscenza dell'Ing. Michele Callegari che mi ha illustrato gli ambiti d'interesse dell'azienda e le eventuali attività che si sarebbero svolte.

Capitolo 2

Active Server Pages (ASP)

2.1 Tecnologie web e ASP

Il modello base di un sistema web è applica il modello client/server: il server web si limita ad inviare su richiesta dei client delle pagine html statiche. Nel corso degli anni questo modello si è evoluto al fine di ottenere una rappresentazione più dinamica delle informazioni e una maggiore interattività per l'utente. Si sono quindi via via diffuse tecnologie quali lo scripting client-side, le CGI (Common Gateway Interface) e infine le Server Pages. Queste ultime permettono al web server di restituire al client delle pagine create dinamicamente in base alle richieste del cliente e allo stato della sua interazione col sistema.

ASP (Active Server Pages) è una tecnologia Microsoft, sviluppata sulla falsariga di PHP, che implementa l'architettura web delle Server Pages. Il codice delle pagine ASP può essere sviluppato in due linguaggi di scripting: VBScript e JScript. Il primo è una semplificazione del Visual Basic ed è sicuramente il più diffuso grazie alla sua semplicità.

2.2 Struttura di una pagina ASP

La struttura tipica di una pagina ASP è costituita di quattro tipi di elementi:

1. **Direttive per l'interprete:** sono dei comandi tramite i quali la pagina ASP comunica all'interprete delle informazioni necessarie alla corretta esecuzione.
2. **Comandi SSI (Server-Side Include):** sono comandi che gestiscono l'inclusione, lato server, di risorse esterne alla pagina.
3. **Codice di scripting (Scriptlet):** è la parte attiva della pagina, contiene il codice scritto dall'autore che deve essere interpretato dal server.

4. **Codice statico:** è la parte di pagina che richiede semplicemente di essere trasferita al client.

2.3 Oggetti built-in e i componenti COM

Gli oggetti built-in sono degli oggetti istanziati automaticamente dal server e messi a disposizione all'esecuzione di ciascuna pagina. Ognuno di questi fornisce delle funzionalità ben precise che permettono al linguaggio di scripting di interagire con l'ambiente d'esecuzione. In totale sono cinque:

1. **Application:** consente la creazione e la gestione delle variabili con scope di applicazione, vale a dire accessibili e condivise da ogni pagina appartenente alla stessa applicazione web.
2. **Request:** permette di ricevere gli input dal client, in particolare i parametri passati in GET o in POST e i cookie.
3. **Response:** consente di inviare l'output al client e di gestire alcune impostazioni ad esso legate.
4. **Server:** permette di interagire con il web server, ottenere informazioni sull'URL corrente, creare oggetti ActiveX.
5. **Session:** consente la creazione e la gestione dello stato legato ad ogni client.

Per estendere le funzionalità fornite dall'ambiente ASP, è possibile creare, avvalendosi di un linguaggio di programmazione adatto (C++ in primis), dei componenti COM personalizzati. Microsoft stessa fornisce dei componenti COM assieme alla piattaforma IIS/ASP per la gestione dei file-system, per l'invio di mail, per l'accesso a database eccetera. È ovviamente possibile affidarsi a componenti sviluppati da altri: sono molto diffusi componenti per l'elaborazione delle immagini e per l'upload avanzato di file.

Istanziare un oggetto COM è molto semplice:

```
Set objFSO = Server.CreateObject("Scripting.FileSystemObject")
```

In seguito è possibile richiamare i metodi dell'oggetto tramite la notazione puntata, ad esempio:

```
objFSO1.DeleteFile(Server.MapPath("/") & "testo.txt")
```

2.4 ASP e Database

Con ASP è molto semplice interfacciarsi e comunicare con una qualsiasi base di dati che rispetti lo standard ODBC (Open DataBase Connectivity). L'accesso ai dati avviene tramite il componente ActiveX Data Object (ADO), che fornisce una serie di oggetti grazie ai quali connettersi al database, eseguire

delle query ed elaborare i risultati. Gli oggetti principali sono l'oggetto Connection e l'oggetto RecordSet.

Il metodo più immediato per richiamare dei dati da database è il seguente:

1. Creare un oggetto di tipo Connection:

```
set conn=Server.CreateObject("ADODB.Connection")
```

2. Specificare la stringa di connessione:

```
Conn.ConnectionString = "driver={SQL  
Server};server=sviluppo;uid=sa;pwd=endntadm;database=Jonathan;p  
rovider=sqloledb"
```

3. Aprire la connessione:

```
conn.Open()
```

4. Si esegue la query, ottenendo un oggetto RecordSet contenente il risultato:

```
rs=conn.Execute("SELECT * FROM proprieta)
```

5. Si utilizza l'oggetto RecordSet per elaborare i dati:

```
while not rs.EOF  
    Response.write("<p>Nome: " & rs("Nome") & "<br>")  
    Response.write("Provincia: " & rs("Provincia"))  
    rs.MoveNext()  
wend
```

6. Una volta finito di utilizzare il RecordSet, questo deve essere chiuso:

```
rs.close()
```

7. Stesso discorso per la connessione:

```
conn.close()
```

Naturalmente il comando Execute può essere utilizzato anche per query di modifica o di eliminazione dei dati.

L'oggetto Connection ha molte altre proprietà e metodi utili per gestire situazioni particolari o per utilizzare funzionalità più avanzate, come ad esempio le transazioni.

2.5 L'oggetto RecordSet

Le funzionalità offerte dall'oggetto RecordSet sono ben più vaste di quanto visto nel paragrafo precedente. Per accedere a queste funzionalità aggiuntive è necessario istanziare un oggetto RecordSet isolatamente e impostare le proprietà desiderate:

```
set rs = Server.CreateObject("ADODB.Recordset")  
rs.ActiveConnection = "StringaDiConnessione"  
rs.Source = "SELECT * FROM dbo.tblBanner "  
rs.CursorType = 0  
rs.CursorLocation = 2  
rs.LockType = 3  
rs.Open()  
...
```

```
response.write(rs("Nome"))
...
rs.close()
```

Si può notare come non sia necessario creare e aprire preventivamente una connessione tramite un oggetto `Connection`, ma basti impostare la stringa di connessione nella proprietà `ActiveConnection` del `ResultSet`. Nel momento in cui sarà richiamato il metodo `Open`, verrà automaticamente aperta la connessione necessaria. Nel caso servisse un oggetto `Connection` con particolari proprietà è possibile crearlo anticipatamente e assegnarlo a `ActiveConnection`.

Il secondo particolare interessante è la proprietà `CursorType`. Ad ogni `RecordSet` è associato un cursore che permette di muoversi attraverso i risultati della query. Esistono quattro tipi di cursore:

- **adOpenForwardOnly** (0): è un cursore simile al cursore statico, ma permette di scorrere i record solo in avanti.
- **adOpenKeySet** (1): è un cursore bi-direzionale, permette di accorgersi di modifiche e cancellazioni effettuate sui record da altri utenti mentre il `RecordSet` è aperto.
- **adOpenDynamic** (2): è un cursore bi-direzionale, permette di accorgersi di inserimenti di nuovi record effettuati da altri utenti.
- **adOpenStatic** (3): è un cursore bi-direzionale, non permette di riscontrare alcuna modifica effettuata ai record.

I cursori con funzionalità più avanzate sono i più comodi da utilizzare ma offrono performance peggiori rispetto ai più semplici; occorre quindi utilizzarli soltanto quando sono veramente necessari.

La proprietà `CursorLocation` indica la locazione del cursore; può assumere tre valori:

- **adUseNone** (1): obsoleto, indica di non usare un cursore.
- **adUseServer** (2): default, indice di usare un cursore server-side.
- **adUseClient** (3): indica di usare un cursore client-side.

Utilizzando cursori client-side è possibile che siano disponibili funzionalità più avanzate rispetto ai cursori di default.

Un'altra impostazione interessante del `RecordSet` è la proprietà `Lock` che controlla i diritti e le modalità di accesso ai dati. Ci sono quattro possibilità:

- **adLockReadOnly** (1): non consente alcuna modifica ai dati, è la modalità predefinita
- **adLockPessimistic** (2): permette l'aggiornamento dei dati, verificando la correttezza ad ogni modifica (Pessimistico).

- **adLockOptimistic** (3): permette l'aggiornamento dei dati, verificando la correttezza soltanto quando viene invocato il metodo Update() (Ottimistico).
- **adLockBatchOptimistic** (4): aggiornamento batch ottimistico.

L'oggetto RecordSet fornisce altre funzionalità, ad esempio per gestire la paginazione dei record o per il salvataggio dei RecordSet sui file.

Capitolo 3

Il primo periodo

I primi giorni di tirocinio sono stati spesi per prendere confidenza con l'ambiente aziendale, con gli strumenti di sviluppo utilizzati e per acquisire, operativamente, delle conoscenze su ASP.

3.1 Gli strumenti e l'ambiente di sviluppo

Per lo sviluppo, l'azienda utilizza un server unico, chiamato "Sviluppo", sul quale risiedono sia IIS, che SQL Server. Ogni sito sviluppato è "mappato" su una porta diversa del web server, in modo da poter lavorare su ognuno di essi come se si trovasse su un server dedicato. Per evitare di dover ricordare tutti i numeri di porta associati ai vari siti, è presente, sulla porta 80 di default, una pagina di gestione dei vari progetti:

 Elenco dei siti Oggi è il 24/6/2002 Utente:										
(mostra dimensione delle celle) (mostra siti terminati)										
Num.	Sito Locale	Sito Remoto	IP	Provider	Porta	Gestione	Dimensione	Data Creaz.	Data Public.	Stato
1	A-bici	www.a-bis.com	195.110.	Mc-Link	155			03/05/2002		Realizza
2	AffitturisticiLit	www.affitturistici.it	195.110.	Mc-Link	87	gestione			04/04/2001	Manu
3	Albercati	www.albercati.com	195.110.	Mc-Link	102			05/03/2001	19/09/2001	In attesa d
4	Annovi_eld			Self-hosted	123	gestione/menus_tabelle_fs.htm		12/11/2001		Sito
5	Ariele	www.pescocostanzini.it	195.110.	Mc-Link	145			25/03/2002		Realizza
6	ATRCC	www.atrec.it	151.39.1	Non definito	138			13/02/2002		Realizza
7	Borse.it	www.borse.it	195.110.	Deda	/					Manu
8	BUR	www.hub.unibo.it	137.204.	Non definito	151			30/04/2002		Realizza
9	Bussola	www.cour.it/bussola	62.129.	Peppercom	158			11/06/2002		Realizza
10	BuyER	www.buyeremilioromagne.it	195.110.	Mc-Link	140			05/03/2002		In attesa d
11	CAAB	www.caab.it	195.110.	Mc-Link	161			19/06/2002		Realizza
12	Careers	www.cour.it/careers	62.129.	Peppercom	92	gestione		09/05/2001	03/07/2001	Manu
13	Cour	www.cour.it	62.129.	Peppercom	92	gestione			23/11/2000	Manu
14	CRMaxudi		62.129.	Peppercom	104			22/05/2001	19/06/2001	Manu

Gli strumenti di sviluppo che sono stati usati principalmente sono Macromedia Dreamweaver Ultradev 4, per lo sviluppo del codice Asp, ed Enterprise Manager per la gestione dei vari database e per la creazione delle query per l'accesso ai dati.

Dreamweaver Ultradev è uno strumento molto potente che permette di creare in modo semplice pagine web dinamiche con accessi ai dati. In particolare è possibile creare, in modo automatico, pagine di visualizzazione, di

inserimento e di modifica di record, semplicemente specificando la base di dati sorgente e la tabella su cui lavorare. Per lo sviluppo di pagine un minimo complesse o non standard è comunque necessario mettere mano al codice, ma fornisce comunque un aiuto non indifferente.

Enterprise Manager invece permette di connettersi ad un server SQL MS, e di gestire tutti i database in esso contenuti. È possibile gestire in visuale ogni aspetto di un database. Molto utile la possibilità di creare e visualizzare dei diagrammi di struttura del database, utili al fine di prendere confidenza con le basi di dati con cui si deve lavorare. Lo strumento permette inoltre di creare query di media complessità in maniera completamente visuale.

3.2 Il sito *AffittiTuristici.it*

Durante il primo periodo si è eseguita un'opera di reverse-engineering su dei siti già esistenti, in modo da prendere confidenza con l'ambiente ASP. In particolare il primo sito su cui si è lavorato è stato *AffittiTuristici.it*. Si tratta di un portale turistico per chi cerca un alloggio per le vacanze che permette agli utenti di comunicare direttamente con i proprietari degli immobili senza intermediazioni.



L'aspetto forse più interessante del sito è la gestione delle varie lingue; sono infatti presenti altre due versioni del sito: una tedesca (www.fewoitalien.net) e una inglese (www.italianhousing.com). Per evitare di

dover scrivere tre versioni diverse di ogni pagina è stato utilizzato il seguente sistema: i testi presenti sono riportati in tutte tre le lingue e, invece che essere messi direttamente in output, vengono passati ad un'apposita funzione "traduci" che, a seconda del sito in cui ci si trova, restituisce il testo da inviare al cliente. Per capire in che sito ci si trova si utilizza una variabile Application, definita nel file global.asa. In questo modo è possibile personalizzare le pagine a seconda della lingua. Ad esempio viene utilizzata per visualizzare banner diversi su ciascun sito.

Dopo aver preso confidenza con la struttura del sito e con il database, mi è stato chiesto di sviluppare la mia prima pagina. Si tratta di una schermata che visualizza, per ogni regione e per ogni provincia, il numero di proprietà presenti, raggruppandole in modo opportuno. Pur essendo abbastanza semplice, mi ha permesso di mettere in pratica le conoscenze acquisite inizialmente.



Nei giorni successivi si è effettuata della manutenzione ad altre pagine di questo e di altri siti. Tra le altre si sono realizzate una pagina per l'invio mail, tramite il componente CDONTS fornito dalla MS, per il sito Marzocchi Extranet.

3.3 Endurance Gestionale

Infine si è analizzato il database dell'applicativo Endurance Gestionale. Questo software, sviluppato internamente all'azienda, permette la gestione di tutti le informazioni legate all'azienda: dipendenti, clienti, progetti, banche. Attraverso questo gestionale ogni dipendente registra le ore di lavoro dedicate ad unna determinata commessa. Ad esempio, nel mio caso le ore sono state registrate sotto la commessa "Endurance", progetto "Ricerca e Sviluppo":

Data	Commessa	Progetto	Descrizione	Durata
20/3/2002	Endurance	Ricerca & Sviluppo	Presa di confidenza con gli strumenti di sviluppo: Enterprise Manager e Macromedia Dreamweaver Ultradev	6h 30m

In questo modo è possibile tener conto precisamente delle ore di lavoro dedicate ad un determinato progetto, in modo da poter determinare il costo da addebitare al cliente. Il software permette inoltre di gestire molti altri aspetti come i preventivi rilasciati, le fatture emesse, gli abbonamenti effettuati ai diversi provider, i domini registrati, i vari pagamenti e altro ancora.

Capitolo 4

Il modulo Banner

Dopo un primo periodo di manutenzione e di reverse-engineering di codice sviluppato principalmente da altri, mi è stato assegnato un primo progetto da sviluppare da zero. Si tratta di un sistema di gestione delle campagne banner da utilizzare all'interno dei vari siti sviluppati dall'azienda.

4.1 Le specifiche

Il modulo Banner Rotator deve permettere di gestire facilmente diverse campagne di banner, mantenendo per ognuna diversi tipi di statistiche.

Per ogni campagna deve essere possibile specificare le seguenti informazioni:

- Nome
- Data di inizio
- Data di fine
- Impressions previste (numero previsto di visualizzazioni del banner)
- Altezza e larghezza delle immagini
- Stato (Attiva o meno)

Le campagne possono essere di due tipi:

- Basate sul componente MSWC.AdRotator
- Basate sull'applet Java DynamicBillBoard

Le campagne AdRotator si basano su un componente fornito dalla Microsoft.

Per ogni campagna AdRotator deve essere creato automaticamente via ASP il file di configurazione (banner<IDCampagna>.txt) che specifica tutte le informazioni necessario al funzionamento e deve essere posizionato all'interno di un'apposita directory ("/schedules").

Nella pagina di gestione di ciascuna campagna, di entrambi i tipi, deve essere visualizzato il codice da inserire dove si desidera che i banner compaiano.

L'upload delle immagini da usare come banner deve avvenire tramite pagina ASP. Una pagine di gestione immagini deve consentire di cancellare le immagini che non sono più utilizzate (o copiate per errore).

Le statistiche relative a banner e campagne si basano sulla memorizzazione su DB di Impression e Click.

Per ogni impression/click devono essere memorizzati:

- Nome del file
- Data e ora
- URL di destinazione
- IDCampagna

Per ogni campagna si deve poter visualizzare impression, click e clickThrough globali, relativi ad un particolare mese e/o relativi ad un particolare banner. Deve essere inoltre possibile raggruppare le statistiche per giorno del mese o per fascia oraria.

Al fine di risparmiare spazio sul database, deve essere possibile, una volta che la campagna risulti terminata, eliminare i dati relativi ai singoli click/impression e mantenere soltanto le informazioni statistiche.

4.2 Il modulo AdRotator

Il modulo AdRotator è un componente fornito direttamente dalla Microsoft per la visualizzazione di una serie di immagini specificati in un determinato file di configurazione.

Esempio file di configurazione:

```
REDIRECT redirect.asp
WIDTH 468
HEIGHT 60
BORDER 0
*
http://www.yoursite.com/images/banner_468_60/banner001.gif
http://www.microsoft.com
Go to the Microsoft Home Page
20
http://www.yoursite.com/images/banner_468_60/banner002.gif
http://www.oracle.com
Learn About Oracle Database Products
10
```

Come si può notare, il file di configurazione è suddiviso in due sezioni, una di configurazione globale, e una che contiene le configurazioni dei singoli banner.

Globalmente vengono specificati: la pagina che effettuerà la redirezione, la larghezza e l'altezza delle immagini, e il bordo desiderato. Per ogni banner invece vanno specificati: l'immagine, l'URL di destinazione, il testo alternativo

e un valore numerico intero. Quest'ultimo è chiamato peso, e influisce sulla frequenza di visualizzazione del banner relativo. Ad esempio, nel file sopra riportato, il primo banner ha il doppio di probabilità di comparire rispetto al secondo.

Nelle pagine che sfruttano il modulo occorre inserire del codice simile al seguente nel punto dove si vuole che appaia l'immagine:

```
<%  
Dim objAdrot  
'--- Create the component  
Set objAdrot = Server.CreateObject("MSWC.AdRotator")  
  
'--- Get the ad from the rotator schedule file and write out the information  
Response.Write ( objAdrot.GetAdvertisement( "/file_di_configurazione.txt" ) )  
  
'--- Destroy the object  
Set objAdrot = nothing  
>%
```

Dopo aver istanziato un oggetto AdRotator, si richiama il metodo GetAdvertisement passando come parametro il nome del file di configurazione e si fa il response.write del risultato.

Infine è necessario creare una pagina che effettui il redirect. La più semplice possibile è la seguente:

```
<%  
Response.Redirect ( Request.QueryString("URL" ) )  
>%
```

Nel nostro caso invece la pagina di redirezione sarà più complicata per gestire le statistiche.

4.3 L'applet *Dynamic Billboard*

L'applet Dynamic Billboard visualizza anch'essa una serie di banner. La differenza rispetto ad AdRotator sta nel fatto che non viene scelta una singola immagine ma vengono visualizzate tutte ciclicamente, passando da una all'altra grazie a degli effetti di transizione.

La configurazione dei vari banner non è contenuta in un file dedicato ma è costituita semplicemente dai parametri dell'applet. Per visualizzare i banner in una pagina occorre quindi includere, nel punto desiderato, del codice simile al seguente:

```
<applet code="DynamicBillboard" width="XXX" height="XXX">  
<param name="delay" value="20000">  
<param name="billboards" value="X">  
<param name="bill0" value="XXX.gif,http://X.X.X/XXX.html">  
<param name="bill1" value="XXX.gif,http://X.X.X/XXX.html">  
<param name="bill2" value="XXX.gif,http://X.X.X/XXX.html">  
<param name="bill3" value="XXX.gif,http://X.X.X/XXX.html">  
<param name="transitions"
```

```

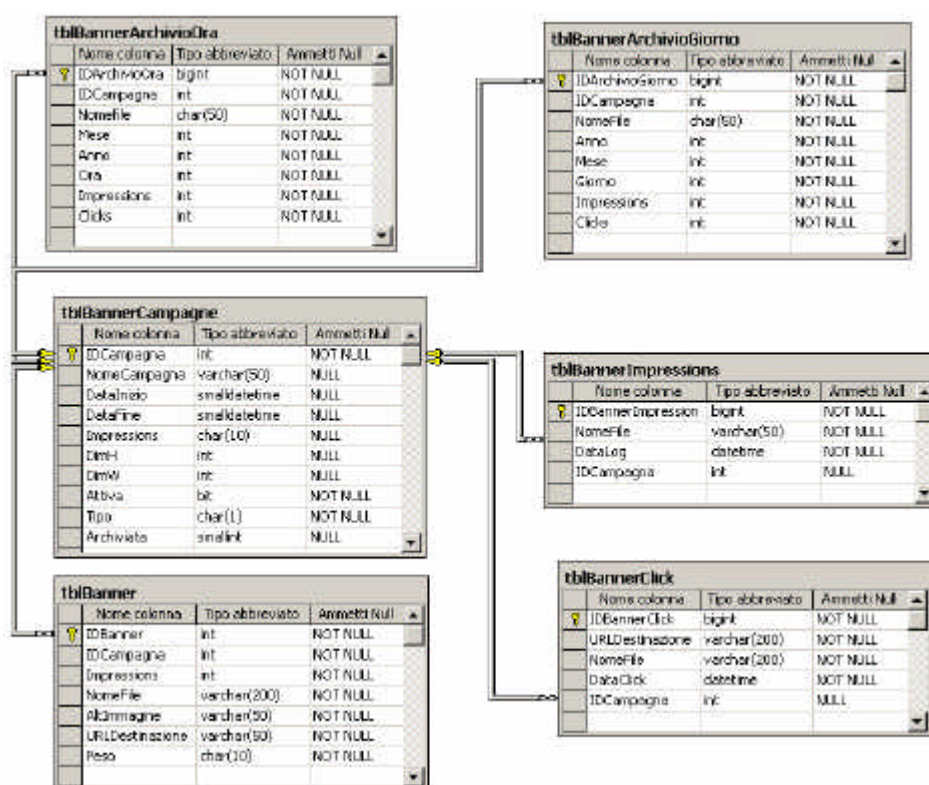
value="4,ColumnTransition,FadeTransition,TearTransition,SmashTransition">
<param name="bgcolor" value="#FFFFFF">
<param name="target" value="_self">
</applet>

```

I parametri sono abbastanza auto-esplicativi. *Delay* rappresenta il tempo in millisecondi per cui ogni immagine viene visualizzata, *billboards* è il numero di immagini, *transitions* contiene il numero e i nomi delle classi di transizione da utilizzare.

4.4 Struttura del database

Il primo passo nella realizzazione del componente è stato quello di realizzare la struttura del database di supporto. Inizialmente l'archiviazione delle campagne non era prevista e quindi la struttura creata era la seguente:



Le tabelle *tblBannerCampagne* e *tblBanner* contengono le informazioni di configurazione delle varie campagne e dei relativi banner. Le tabelle *tblBannerClick* e *tblBannerImpressions* contengono invece informazioni sulle impression e sui click relativi ad un determinato banner di una determinata

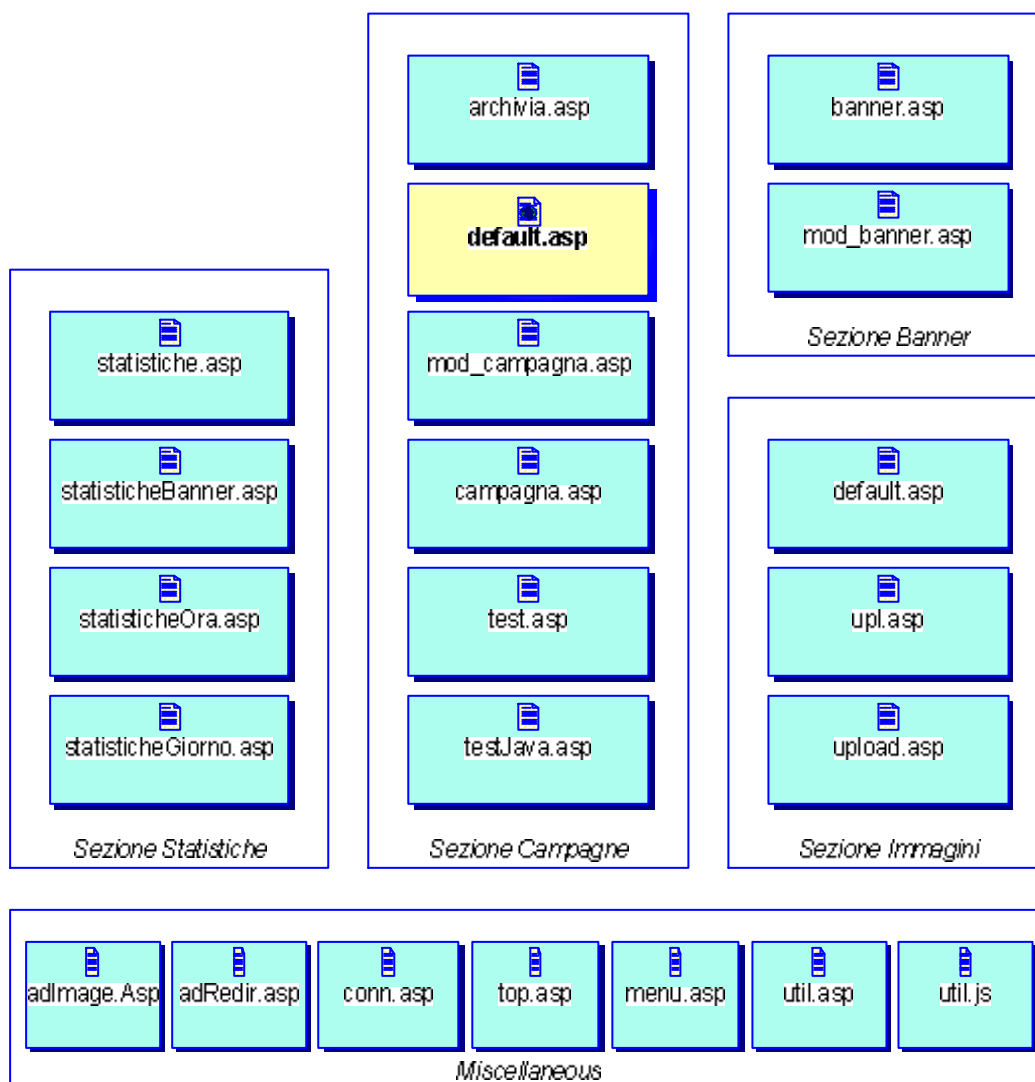
campagna. Le ultime due tabelle sono le tabelle dove vengono archiviate le informazioni statistiche delle campagne terminate.

Si può notare come in queste due tabelle non sia riportato semplicemente l'IDBanner, ma vengano riportati il nome del file, l'ID della campagna e l'URL di destinazione. Ciò è dovuto ad alcune scelte effettuate per limitare il numero di query al database per ogni impression e/o click. Per ottenere l'ID relativo ad un banner sarebbe infatti necessario effettuare una query apposita ad ogni visualizzazione dell'immagine e ad ogni click.

In seguito è stata aggiunta una vista che permette di semplificare le query eseguite dalle varie pagina ASP.

4.5 La struttura delle pagine

Le varie pagine che costituiscono il modulo sono state organizzate in quattro sezioni: campagna, banner, immagini e statistiche.



Tra le pagine della sezione campagna, che è il cuore del sistema, è evidenziata in giallo la pagina iniziale, default.asp. Tramite questa è possibile visualizzare le campagne presenti, selezionarne una per la modifica o inserirne una nuova.

The screenshot shows the 'BANNER MANAGER' interface. On the left, there are links for 'Gestione Campagne' and 'Gestione Immagini'. The main area contains a table of campaigns with columns for 'Nome Campagna', 'Data di inizio', and 'Data di fine'. A legend indicates status colors: blue for 'Scaduta', grey for 'Disattiva', and red for 'Disattiva e Scaduta'. A dropdown menu is open, showing options like 'non scadute', 'attive', 'attive e non scadute', 'scadute', 'disattive', and 'disattive e scadute'. The table lists campaigns such as 'ADFinal', 'Test', 'Java Test', 'TestArchivio', and 'ProvaDisatt'.

Nome Campagna	Data di inizio	Data di fine	Stato
ADFinal	2/5/2002	4/5/2004	Scaduta
Test	4/9/2002	5/7/2002	Disattiva e Scaduta
Java Test	5/5/2002	5/5/2002	Disattiva
TestArchivio	5/9/2001	5/9/2002	Disattiva
ProvaDisatt	5/10/2002	5/10/2003	Disattiva

Il colore delle varie righe permette di capire in che stato si trovi la campagna mentre un menu a tendina permette di selezionare quali campagne visualizzare.

Facendo click sul nome di una campagna si accede alla relativa pagina di gestione, dove è possibile visualizzare ed eventualmente modificare qualsiasi impostazione legata ad essa.

The screenshot shows the management page for the 'ADFinal' campaign. It includes a table with campaign details, a table of banner statistics, and a code editor for banner code.

Nome campagna	ADFinal
Data inizio	2/5/2002
Data fine	4/5/2004
Impressioni previste	3000
Dimensioni Banner	
Altezza	60 pixel
Larghezza	468 pixel
Stato attuale	Attiva
Archiviata	No
Tipo di campagna	Ad Rotator

[Modifica dati della campagna](#)
[Statistiche della campagna](#)
[Disattiva questa campagna](#)
[Test della campagna](#)

Banner attivi per questa campagna:

Nome Banner	All. Immagine	URL di destinazione	Peso	Impressioni previste	Impressioni	Click	ClickThrough
borse_forum_full.gif	Borse Forum	http://www.borse.it/forum	20	1000	17	5	29.41%
calcolata_full.gif	Borse Forum	http://www.borse.it	20	2000	27	15	40.54%
Totale				3000	54	20	27.04%

[Inserisci nuovo banner](#)

Codice da inserire nel punto dove comparirà il banner:

```


Set ad = Server.CreateObject("MSIUC.AdRotator")
ad.TargetFrame="body"
Response.Write ad.GetAdvertisement("banner34.txt")
    
```

Nella parte in alto sono riportate le informazioni generali relative alla campagna. Subito sotto sono presenti una serie di link che permettono, nell'ordine, di modificare i dati della campagna, visualizzare le statistiche, attivare/disattivare la campagna e infine aprire una pagina di test in una finestra pop-up. Inoltre se la campagna è finita, è presente un link alla pagina archivia.asp che si occupa di effettuare l'archiviazione.

Nella parte centrale sono invece riportati i banner associati alla campagna corrente; per ognuno sono riportate le informazioni di base e i relativi impression, click e clickthrough.

Infine nella parte inferiore viene riportato in una text area il codice da inserire nel punto dove si desidera che il banner compaia, differenziato a seconda che si tratti di una campagna AdRotator o basata sull'applet Java. Tale codice è calcolato dinamicamente in quanto è possibile cambiare tipo di campagna in ogni momento.

Facendo click sul nome di un banner si accede alla relativa pagina di gestione, dove vengono visualizzate le relative informazioni e statistiche. Viene naturalmente anche visualizzata l'immagine relativa al banner e viene presentato un link che porta alla pagina di modifica.

BANNER MANAGER	
Gestione Campagne	Nome file
Gestione Immagini	borse_forum_full.gif
	
AltImmagine:	Borse Forum
URL di destinazione:	http://www.borse.it/forum
Peso:	20
Impressions predefinite:	1000
Impressions:	17
Clicks:	5
ClickThrough:	29.41%
	Modifica il banner

In tale pagina si può modificare qualsiasi informazione relativa ad un banner. In particolare per cambiare immagine viene presentato un menù a tendina tramite il quale scegliere fra le immagini disponibili sul server. Se si desidera inserire una nuova immagine è possibile accedere alla pagina di upload.

Per la gestione delle immagini è comunque presente anche una sezione dedicata, tramite la quale è possibile visualizzare la lista delle immagini presenti sul server e delle campagne che ne fanno uso. Nel caso un immagine non sia utilizzata da nessuna campagna viene data la possibilità all'utente di eliminarla.

Nome File	Campagne che lo utilizzano	Operazioni
borse_forum_full.gif	ADFinal	
	Test	
borse_mail_full.gif	Test	
drinks_baby_full.gif	Inutilizzato	Elimina
hot-ex_full.jpg	JavaTest	
	TestArchivia	
hot-ex_full2.jpg	JavaTest	
natsabe_full.gif	ADFinal	
non_gener_bott.gif	Inutilizzato	Elimina
wine_full.gif	Inutilizzato	Elimina

Le pagine della sezione statistiche si occupano invece di mostrare all'utente le informazioni desiderate riguardo all'andamento della campagna. È possibile visualizzare informazioni generali su click, impression e clickthrough della campagna o dei singoli banner, su tutto l'arco temporale o relativamente ad un singolo mese. Inoltre sono fornite statistiche raggruppate per giorno del mese o per ora del giorno.

Naturalmente le statistiche vengono visualizzate sia per le campagne in corso che per le campagne archiviate, accedendo alle corrispondenti tabelle del database per ottenere le informazioni corrette.

4.6 Il log di click e impression.

Uno degli scopi principali del modulo è quello di fornire delle statistiche dettagliate riguardo alle impression e ai click. È stato quindi necessario trovare un sistema per memorizzare i dati necessari sul database.

Per effettuare il log dei click si è utilizzata la pagina di redirectione (adRedir.asp) necessaria al funzionamento del componente adRotator. Quando un utente clicca su un banner, il componente non fa altro che passare i parametri del banner (impostati nel file di schedule) alla pagina adRedir.asp. Questa si connette al database registrando le informazioni relative al banner e poi effettua la redirectione. Questa tecnica viene sfruttata anche con l'applet Dynamic Billboard, impostando come URL di destinazione la pagina adRedir.asp e passando tramite GET i parametri necessari alla redirectione.

Risolto il problema dei click rimane il quello delle impression: com'è possibile registrare su database ogni volta che un'immagine viene visualizzata su schermo?

Il problema è stato risolto creando una pagina (adImage.asp) che, ricevuti in ingresso il nome del file e le informazioni relative alla campagna e al banner, effettua la registrazione dell'impression sul database, imposta il mime/type corretto ed effettua una scrittura binaria sull'output del file richiesto.

Impostando questa pagina, invece che direttamente le immagini, nelle configurazioni di adRotator e dell'applet Java è stato possibile effettuare anche i log delle impressioni.

Capitolo 5

Upload di immagini per il sito Minardi

5.1 Requisiti

L'obiettivo è sviluppare una pagina che consenta ai gestori del sito Minardi di effettuare l'upload di immagini sul server web.

La pagina deve consentire l'upload di più immagini contemporaneamente (massimo dieci) e deve permettere di specificare per ognuna una descrizione in italiano, una descrizione in inglese e delle parole chiave per la ricerca.

Queste informazioni devono essere memorizzate in un'apposita tabella presente nel database Minardi e verranno poi utilizzate nel sito per fornire agli utenti un sistema di ricerca delle immagini.

Una volta effettuato l'upload occorre ridimensionare le immagini in modo da ottenere tre dimensioni diverse: originale, media e thumbnail. Queste vanno poi memorizzate all'interno di tre directory diverse, una per ciascuna dimensione.

Il nome dei file deve essere lungo dieci caratteri e deve essere scelto casualmente in modo da essere (univoco occorre naturalmente verificare che effettivamente sia così) e deve essere lo stesso per tutte e tre le immagini.

Il sistema deve essere in grado di lavorare con immagini jpg, gif e png indifferentemente.

5.2 La scelta di utilizzare Asp.NET

Il problema più grande da risolvere per realizzare la pagina richiesta era quello dell'elaborazione delle immagini. ASP non fornisce alcun aiuto nella gestione delle immagini e l'unico sistema per effettuare il ridimensionamento è quello di utilizzare componenti ad hoc realizzati internamente o sviluppati da altri.

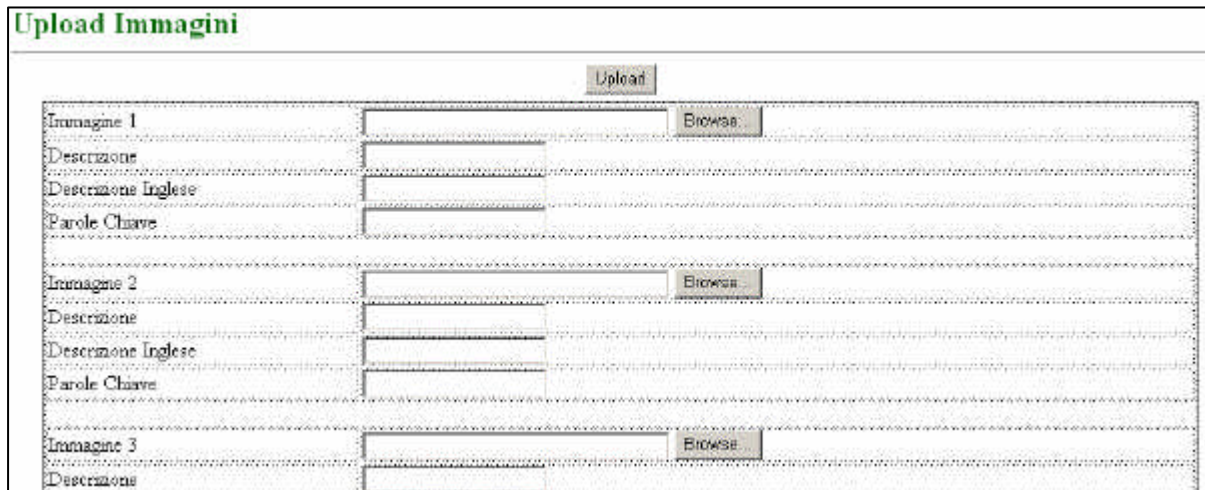
Inizialmente si sono svolte diverse ricerche su Internet per trovare un componente gratuito che fosse utile ai nostri scopi, ma la ricerca è stata vana. Tuttavia durante le ricerche è capitato di leggere delle guide che spiegavano con che facilità fosse possibile elaborare la immagini tramite ASP.NET.

Si è quindi deciso di utilizzare il nuovo ambiente Microsoft per lo sviluppo della pagina.

5.3 Un primo prototipo

Utilizzando ASP.NET è stato abbastanza semplice realizzare un prototipo che soddisfacesse le specifiche.

La pagina includeva un form di invio contenente dieci campi di invio file e dieci campi di descrizione e di parole chiave.



The screenshot shows a web form titled "Upload Immagini" in green text. At the top center, there is an "Upload" button. Below it, the form is organized into three distinct sections, each for uploading an image. Each section contains a file input field with a "Browse..." button to its right. Underneath each file input, there are three text input fields: "Descrizione", "Descrizione Inglese", and "Parole Chiave". The first section is labeled "Immagine 1", the second "Immagine 2", and the third "Immagine 3". The form is enclosed in a thin black border.

Alla fine dell'upload viene mostrata all'utente una pagina riassuntiva contenente informazioni sull'esito dei vari upload. Vengono mostrate le immagini thumbnail di ogni file inviato, oltre alle dimensioni in kb dell'immagine originale e al nome del file sul server.

5.4 Il problema del tempo

Durante i test in locale, la pagina funzionava in modo corretto: i file venivano inviati correttamente, il ridimensionamento funzionava e le immagini venivano salvate nelle apposite directory. Tuttavia, una volta montata sul server web, la pagina dava dei problemi di timeout se i file inviati erano troppo grandi.

Infatti effettuando l'upload di dieci immagini di 1Mb circa, anche tramite ADSL, il tempo necessario è superiore al tempo di time-out del server-web. Per risolvere il problema è stato necessario modificare il file machine.config sul server Minardi, per impostare il time-out a 60 minuti.

Pur avendo ottenuto una pagina funzionante, si era reso visibile un nuovo problema: come far capire all'utente che l'upload sta procedendo correttamente? È infatti normale che un utente medio, dopo un paio di minuti senza alcun riscontro visivo, si chieda se tutto sta procedendo correttamente e magari prema il pulsante di aggiornamento, interrompendo così l'upload.

Il problema è che durante il POST dei dati non è possibile interagire né tramite ASP né tramite Javascript con la pagina che sta caricando. Esistono svariati componenti che forniscono metodi alternativi per l'upload in modo da

visualizzare l'andamento dell'operazione agli utenti, ma purtroppo sono tutti a pagamento.

5.5 La soluzione

La soluzione da me realizzata, pur non visualizzando l'andamento reale dell'upload, fornisce un resoconto visivo all'utente, in modo che sappia almeno quale dei dieci file si sta inviando in questo momento e il risultato degli upload precedenti. Inoltre questa soluzione consente di risolvere in maniera più elegante il problema dei time-out del server.

Il primo passo è stato quello di modificare la pagina di upload, in modo che gestisse l'invio di un solo file. Poi è stata creata una pagina contenente dieci pagine di upload, ciascuna dentro ad un proprio frame inline. Infine si è creata una pagina di controllo, che comunica con le altre due per inviare i vari form.

L'idea è stata quella di dividere la pagina in due frame: uno di controllo e uno di upload. Il frame di controllo si occupa di gestire l'upload dei vari file comunicando con l'altro.

La struttura è la seguente:

Il diagramma illustra la struttura della pagina di upload, divisa in due frame principali: un frame di controllo (in rosso) e un frame di upload (in verde). Il frame di controllo (sinistra) contiene un campo "Upload status:", un pulsante "Invia" e un pulsante "Carne Upload". Il frame di upload (destra) è diviso in tre sezioni, ciascuna con i seguenti campi: "Nome File" (con un pulsante "Sfoglia..."), "Descrizione", "Descrizione Inglese" e "Parole Chiave".


La finestra è divisa in due frame principali: in quello di sinistra è contenuta la pagina di controllo (in rosso), in quello di destra la pagina contenitore (in verde). All'interno di questa sono racchiusi i dieci inline frame con le pagine di upload (in blu)

Il funzionamento è a grandi linee il seguente:

1. L'utente seleziona i vari file da inviare e compila i vari campi di testo.

2. L'utente preme il pulsante di upload nella pagina di controllo
3. Tramite Javascript la pagina di controllo esamina i vari campi delle pagine di upload e capisce quali form sono da inviare.
4. La pagina di controllo prende il primo/prossimo form da inviare e lo invia, comunicando alla pagina contenitore quale delle dieci pagine è in corso di upload.
5. La pagina contenitore evidenzia con dei colori intermittenti la pagina che è in corso di upload.
6. Una volta terminato l'invio, la pagina di upload mostra il resoconto e comunica alla pagina di controllo di passare al prossimo file.
7. La pagina di controllo comunica a quella contenitore che l'upload del file è terminato.
8. La pagina contenitore evidenzia con il colore verde la form che è stata inviata correttamente.
9. La pagina di controllo verifica se ci sono altri file da inviare. Se è così torna al punto 4.
10. Viene aperta una finestra alert per comunicare all'utente la fine degli upload.

Ad esempio durante un upload un utente vedrà una pagina del simile alla

Upload status:	
1 di 2 completati.	
<input type="button" value="Pulsante"/>	
<input type="button" value="Nuovo Upload"/>	
<div style="border: 2px solid green; padding: 5px;"> <p>Nome: d02mon809.jpg Nome sul server: HORHXFZUUN.jpg Tipo: image/jpeg Occupazione: 1250684 Descrizione: Prova1 Descrizione Inglese: Test1 Parole Chiave: Prova Test 1</p>  </div>	
<div style="border: 2px solid blue; padding: 5px;"> <p>Nome File: <input type="text" value="C:\Documents and Settings\christofretri\Docum..."/> <input type="button" value="Sfoglia..."/></p> <p>Descrizione: <input type="text" value="Prova2"/></p> <p>Descrizione Inglese: <input type="text" value="Test2"/></p> <p>Parole Chiave: <input type="text" value="Prova Test 2"/></p> </div>	

seguinte.

In verde è evidenziato l'upload terminato, mentre in blu (in realtà lampeggiante chiaro/scuro) è evidenziato l'upload in corso.

La comunicazione tra le varie pagine avviene tramite polling: una pagina controlla ogni n millisecondi un campo nascosto dell'altra in modo da capire quando vi è un cambiamento nello stato degli upload.

Nel caso si tenti di inviare un file che non ha un'estensione valida viene naturalmente visualizzato un messaggio di errore.

Capitolo 6

Il modulo sondaggio

L'ultimo progetto vero e proprio che mi è stato assegnato è il modulo sondaggio, anch'esso da utilizzare all'interno del sito Minardi.

6.1 Le specifiche

L'obiettivo è creare un modulo sondaggi per il sito web Minardi.

Deve essere organizzato in due parti principali: una sezione di gestione dei sondaggi, dove deve essere possibile creare e modificare i vari sondaggi e un file asp da includere all'interno delle pagine dove si vuole visualizzare il sondaggio.

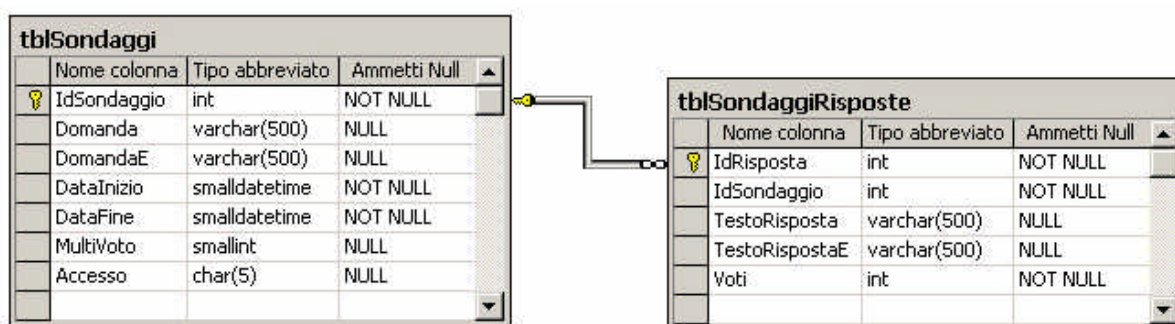
Ogni sondaggio deve essere disponibile sia in inglese che in italiano e prevede da 2 a n risposte possibili. Deve essere possibile scegliere quante volte ed eventualmente ogni quanto un utente può votare per ogni sondaggio.

Esistono cinque categorie di utenti nel sito Minardi: visitatori, utenti free, utenti silver, utenti gold e utenti gold family. Per ciascuno di queste deve essere possibile attivare o disattivare la visualizzazione di un sondaggio.

I risultati vanno memorizzati su database e deve essere creata una pagina che mostri i risultati sondaggi realizzati.

6.2 La struttura del database

La struttura del database è molto semplice. Bastano due tabelle: una per le informazioni relativa ad un sondaggio e un con le informazioni sulle risposte possibili ai vari sondaggi. In quest'ultima vengono memorizzati anche i voti relativi a ciascuna risposta.



6.3 La struttura delle pagine

La struttura delle pagine di gestione ricalca quella del modulo banner. La pagina principale mostra l'elenco dei sondaggi presenti offrendo la possibilità di selezionarne uno per la gestione o di accedere alla pagina di inserimento.

La pagina di gestione sondaggio mostra tutte le informazioni relative ad esso permettendo di modificarle a piacimento. Per la gestione del voto multiplo, è presente un menù a tendina che permette di scegliere ogni quanti giorni è possibile votare. La gestione degli accessi è resa possibile da una serie di checkbox con le quali specificare a quali utenti è rivolto il sondaggio.

[Ritorna alla pagina di gestione sondaggi](#)

Domanda:

Per quale motivo non ti sei ancora registrato?

DomandaE:

Why haven't you registered yet?

DataInizio:

7

6

2002

DataFine:

7

8

2002

Multivoto: Dopo

6

giorni

Accesso:

Tutti Seleziona
 Visitatori
 Utenti Free
 Silver
 Gold
 Gold Family

[Elimina questo sondaggio](#)

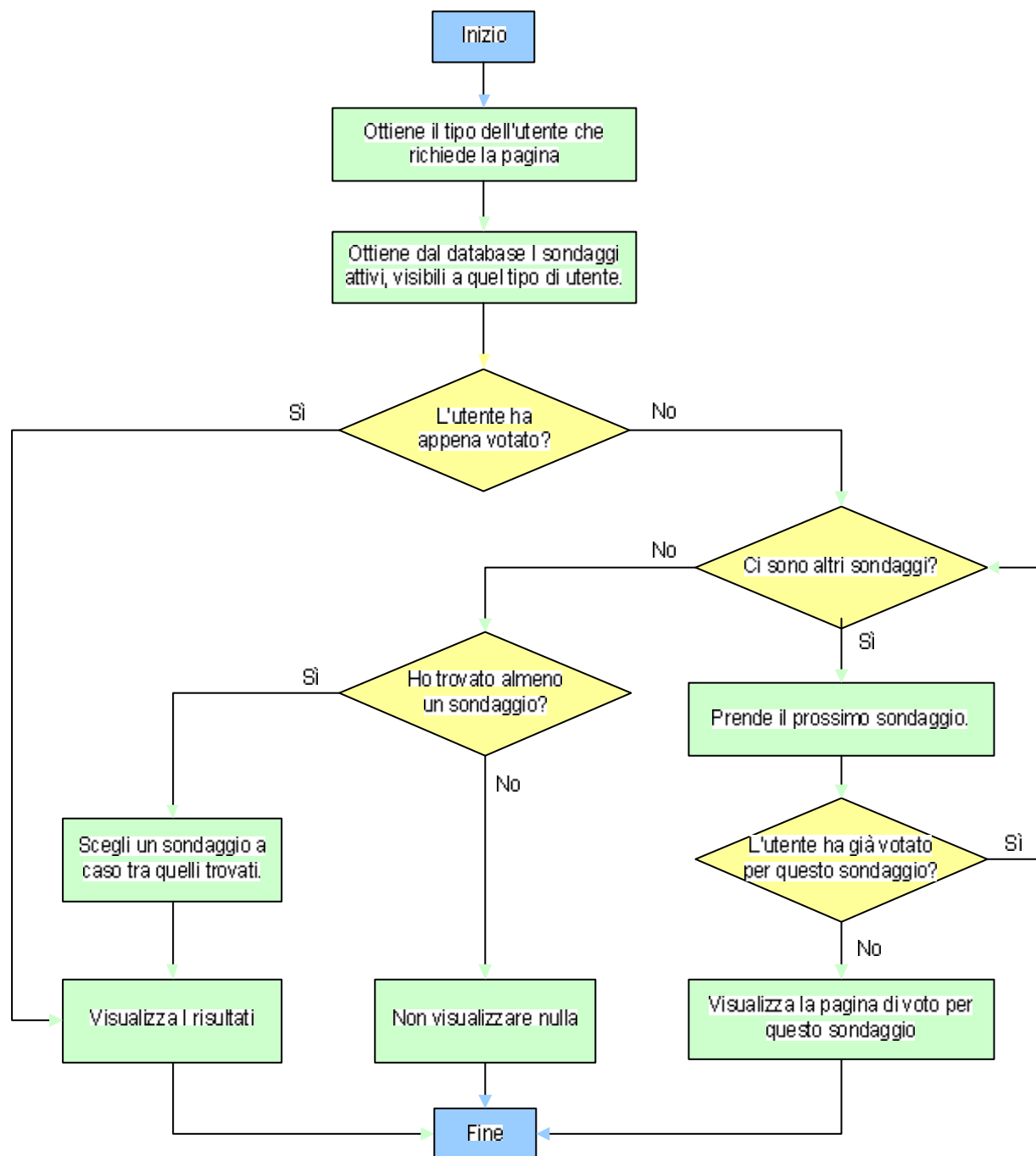
Risposte relative a questo sondaggio

Id Risposta	Testo Risposta	Answer Text	Numero Voti
2	Non mi va di darvi i miei dati	I don't want to give away my data	836
3	Non ho ancora avuto tempo	I haven't had time yet.	651
4	Non mi interessa	I'm not interested	333

[Aggiungi una risposta](#)

Nella parte inferiore sono riportate le risposte relative al sondaggio, con i relativi risultati, e un link alla pagina di inserimento di una nuova risposta. Facendo click sul testo di una risposta si accede alla pagina di modifica.

Il cuore del sistema banner è la pagina sondaggio.asp, che è la pagina che verrà inclusa nel punto dove si vuole che il sondaggio compaia. Questa pagina, all'invocazione, esegue le operazioni descritte nel seguente diagramma di flusso.

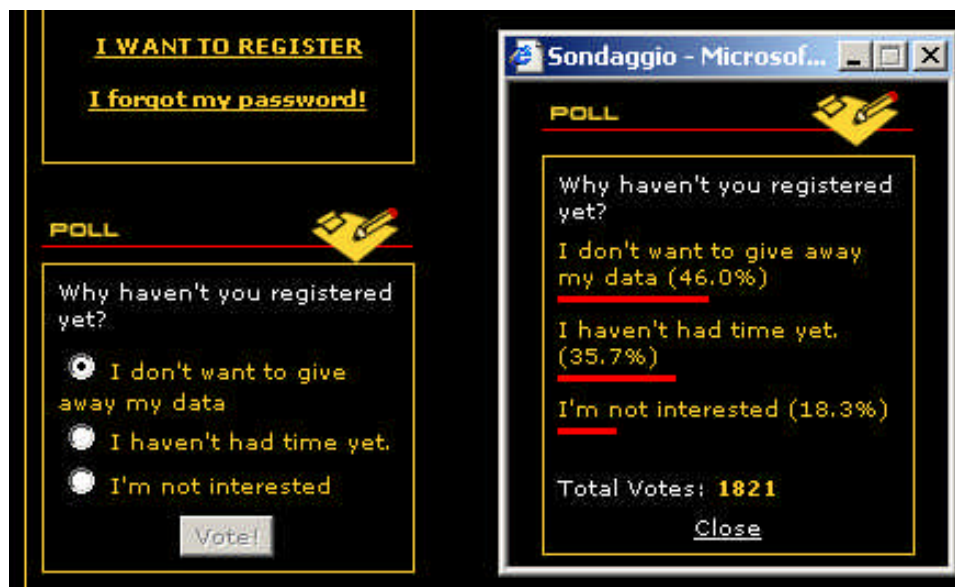


Come si può notare la pagina è in grado di gestire più di un sondaggio contemporaneamente ed è in grado di accorgersi se non è presente alcun sondaggio.

Per capire che tipo di utente sta richiedendo si sfrutta una variabile presente nell'oggetto session dal momento in cui si effettua il login: se la

variabile non è presente abbiamo a che fare con un visitatore, se invece è presente indica il livello dell'utente.

Per determinare se un utente ha votato o meno si utilizzano invece i cookie. Al momento del voto viene creato un cookie identificato dalla stringa "Voted<IDSondaggio>" e con un tempo di vita pari al numero di giorni dopo i quali l'utente ha diritto di votare nuovamente. Nel caso in cui non sia previsto il voto multiplo, il tempo di vita viene settato in base alla data di fine del sondaggio.



La visualizzazione delle barre dei risultati è ottenuta tramite l'utilizzo di un'immagine gif di dimensioni 6x1 pixel; i tre pixel superiori sono di colore rosso, i tre pixel inferiori trasparenti, in modo da fornire una spaziatura tra i risultati di una risposta e i successivi. La larghezza dell'immagine viene impostata dinamicamente in base ai risultati in modo da ottenere l'effetto desiderato. Per ottenere un effetto simile sarebbe stato possibile utilizzare anche una tabella, modificando la larghezza di una cella, ma tale soluzione, oltre ad essere esteticamente inferiore, non sarebbe stata visualizzata allo stesso modo su tutti i browser.

Infine la pagina di visualizzazione dei risultati dei vari sondaggi, non fa altro che riportare i risultati dei sondaggi accessibili all'utente connesso in quel momento. I risultati sono paginati, cinque sondaggi per pagina, tramite il supporto alla paginazione fornito dall'oggetto RecordSet.

Capitolo 7

Ricerca e sperimentazioni con ASP.NET

L'ultimo periodo del tirocinio è stato dedicato ad ASP.NET, sperimentando con il nuovo ambiente Microsoft. In particolare sono stati analizzati la nuova sintassi, i nuovi metodi di accesso a database e i nuovi controlli web per la visualizzazione dei dati.

7.1 Microsoft .NET

.NET è la nuova specifica Microsoft per lo sviluppo, la distribuzione e la gestione di soluzioni aziendali multi-tier. Per il suo funzionamento si basa sulla piattaforma di sviluppo .NET Framework.

.NET rappresenta un netto taglio con il passato da parte di Microsoft, che ha deciso di puntare in modo massiccio sul calcolo distribuito e in particolare sui Web Services e su XML.

Per molti aspetti risulta simile, e nasce probabilmente come risposta di Microsoft alla Sun, all'architettura J2EE: entrambe offrono un modello di componenti per lo sviluppo di applicazioni distribuite, una serie di API per lo sviluppo di interfacce grafiche e servizi per l'accesso ai dati. Tutte e due basano il loro funzionamento su una macchina virtuale: JVM per J2EE, CLR (Common Language Runtime) per .NET.

Tuttavia sono molte anche le differenze: la JVM è programmabile, ufficialmente, soltanto in Java, .NET invece supporta una grande varietà di linguaggi; d'altra parte CLR è disponibile soltanto per le piattaforme Microsoft mentre la JVM è disponibile per le principali piattaforme sul mercato.

Oltre a concorrere con J2EE, .NET si propone anche altri obiettivi, uno su tutti l'eliminazione de "l'inferno delle DLL". La soluzione proposta è basata sugli Assembly, evoluzioni delle attuali DLL. Rispetto a quest'ultime hanno il vantaggio di essere auto-descrittivi; infatti contengono un manifesto (manifest) che descrive il componente e tutte le relazioni con altri componenti e con il sistema.

7.2 Novità in ASP.NET

ASP.NET non è semplicemente una nuova versione di ASP: ASP.NET è stato riscritto a partire da zero e le differenze tra l'ultima e la precedente versione sono enormi.

Una novità molto importante è il concetto di Web Form; i Web Form estendono il concetto di form di Visual Basic al mondo ASP. Tramite l'ambiente di sviluppo Visual Studio .NET è possibile creare pagina in modo visuale e sfruttare i vantaggi legati all'utilizzo degli eventi.

Un'altra differenza fondamentale è che ASP.NET supporta i linguaggi compilati. Per programmare le varie pagine è ora possibile utilizzare un linguaggio a scelta tra Visual Basic .NET, C# e JScript, con la sola limitazione di utilizzare un unico linguaggio all'interno della singola pagina. Il vantaggio di avere del codice compilato rispetto ad ASP è evidente sotto il profilo delle prestazioni e della facilità di debug.

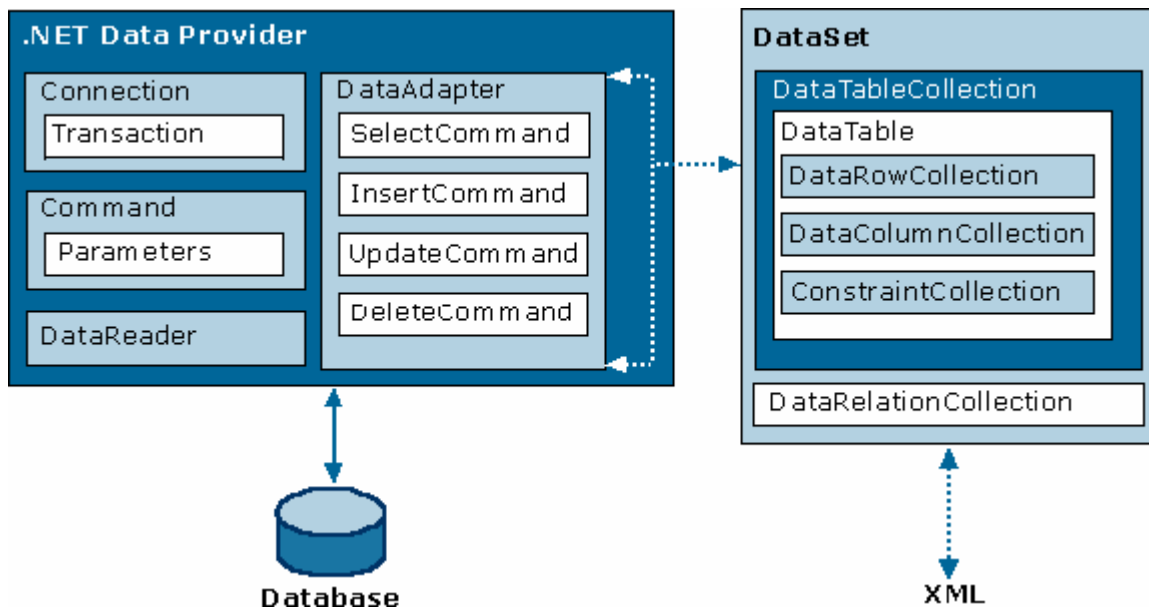
Con ASP.NET sono forniti una serie di Controlli Web utili allo sviluppo di pagine interattive. Esistono cinque famiglie di controlli: i controlli intrinseci che hanno un equivalente in HTML, i controlli list per la visualizzazione dei dati nelle pagine, i rich control che offrono sofisticate funzionalità di interfaccia, i controlli di validazione per controllare i dati forniti in input dall'utente i controlli mobile che incapsulano i tag WML per i dispositivi WAP.

Un'altra grande novità sono i metodi di accesso ai dati. ADO.NET rappresenta una netta evoluzione rispetto ad ADO e introduce i nuovi oggetti DataSet, DataReader e DataSet Adapter. Inoltre permette una migliore compatibilità cross-platform grazie all'utilizzo di XML come formato di interscambio dei dati. Infine ADO.NET è stato progettato per utilizzare la modalità disconnessa dei dati, garantendo migliori prestazioni grazie all'utilizzo di tabelle in memoria.

7.3 L'accesso ai dati in ASP.NET: ADO.NET

Pensando al grande successo riscontrato tra gli sviluppatori da ADO viene da chiedersi quale siano i motivi che hanno spinto la Microsoft a sviluppare ADO.NET. Il motivo principale è che nella versione attuale ADO non è integrato ancora pienamente in contesti Web e non ha nessun oggetto o tool di supporto per XML. Inoltre nelle connessioni a database tradizionali la connessione è in un certo modo persistente: deve rimanere aperta per tutta la durata dell'utilizzo dei dati da parte del client. Questo non è accettabile in ambiente Internet: un client potrebbe avere la necessità di connettersi al server, richiedere dei dati, disconnettersi e riconnettersi dopo un determinato periodo aspettandosi di trovare la risposta pronta.

Il seguente diagramma illustra la struttura dei componenti di ADO.NET.



Sulla sinistra è raffigurato il .NET Data Provider; questo componente implementa la connessione e la comunicazione con il database. Esistono due tipi di Data Provider in .NET, uno dedicato a Microsoft SQL Server e uno dedicato agli altri RDBMS che supportano lo standard OLE-DB. Un Data Provider è costituito da quattro tipi di oggetto:

1. **Connection:** permette la connessione al database.
2. **Command:** permette di inviare comandi SQL al database.
3. **DataReader:** permette di leggere i dati attraverso uno stream.
4. **DataAdapter:** insieme all'oggetto DataSet permette di memorizzare ed elaborare i dati ottenuti in modo disconnesso, mantenendoli in una struttura XML memory-resident.

In breve, le funzionalità principali di ADO.NET sono le seguenti:

1. Un oggetto connessione stabilisce la connessione fra la pagina web e il database.
2. Dopodiché è possibile inviare diversi comandi al database tramite un oggetto Command.
3. Per mostrare i dati, efficientemente e senza apportare modifiche, i risultati possono essere letti tramite un oggetto DataReader.
4. Per poter modificare i dati, i risultati vanno memorizzati all'interno di un oggetto DataSet, manipolandoli poi tramite un oggetto DataAdapter.

La novità più importante è l'oggetto DataSet che permette di mantenere in memoria una struttura dati disconnessa simile alla struttura relazionale del database. È possibile memorizzare più di una tabella all'interno del medesimo DataSet ed è possibile impostare i vincoli relazionali che esistono tra una tabella e l'altra. Attraverso un modello ad oggetti è poi semplice accedere ai dati che interessano.

7.4 La visualizzazione dei dati: DataGrid, DataList e Repeater.

Per quanto riguarda ASP.NET, le novità principali riguardanti i dati sono i tre controlli web per la visualizzazione: DataGrid, DataList e Repeater. Tutti e tre questi controlli permettono, agganciandosi ad una sorgente dati, di visualizzare in modo quasi automatico i dati ottenuti da database.

Il controllo DataGrid visualizza i dati in forma tabellare e fornisce una serie di funzionalità avanzate: gestisce in modo ottimale la paginazione (permettendo comunque di costruire un proprio meccanismo personalizzato) e gestisce l'input da parte dell'utente permettendogli di selezionare e modificare i dati all'interno della tabella. Lo stile della tabella è ridefinibile tramite l'uso di template. Sicuramente è il controllo web che sta riscuotendo il maggior successo tra gli sviluppatori. Di seguito è riportato un esempio di come possa essere sfruttato per visualizzare dati gerarchici.

	Nome	Cognome	Telefono	In Affitto	Scadute	Riferimenti	
±	Emanuela	Fratini	0744272907	3	0		Edit
±	Goffredo	Scotto Di Carlo	081.8968341	1	0		Edit
±	Paola	Tomas	043967119	1	0		Edit
±	Roberto	Mezzone	0575 97384	2	0		Edit
±	Salvatore	Achenza	079 272200	1	0		Edit
±	Giuliana	Della Latta	0584 619739	2	0		Edit

	Nome	Cognome	Telefono	In Affitto	Scadute	Riferimenti	
-	Emanuela	Fratini	0744272907	3	0		Edit
				1		Villa Le casaline	Edit
				1		Casa Giovanna	Edit
				1		Casa Marianna	Edit
±	Goffredo	Scotto Di Carlo	081.8968341	1	0		Edit
±	Paola	Tomas	043967119	1	0		Edit
±	Roberto	Mezzone	0575 97384	2	0		Edit

Facendo click nel punto cerchiato in rosso vengono visualizzati i dati relativi alle proprietà di quel determinato utente. L'effetto è ottenuto tramite la costruzione dinamica in memoria di una tabella contenente, in una forma particolare, i dati relativi sia all'utente che a tutte le sue proprietà. Dopo aver agganciato il DataSet contenente la tabella al controllo DataGrid, viene

modificata la visibilità e il colore delle righe relative alle proprietà in modo da ottenere l'effetto desiderato.

L'interattività è ottenuta tramite la gestione degli eventi: al simbolo "+" è associato un evento che espande o collide le righe relative ad un utente.

Il controllo DataList è simile a DataGrid, ma permette di controllare in maniera più libera come i dati vengono visualizzati. La maggiore libertà si paga in termini di immediatezza, infatti è obbligatorio impostare manualmente, tramite l'utilizzo di template, la struttura visuale dei dati. Inoltre non vi è supporto alcuno alla paginazione e tutto è lasciato al programmatore. È ancora presente il supporto alla selezione delle righe e all'editing in-line.

L'oggetto Repeater infine fornisce il massimo grado di libertà, richiedendo però gli sforzi maggiori da parte del programmatore. È necessario specificare precisamente, sempre con l'ausilio di template, tutti i tag html che costituiscono il layout per output dei dati. Inoltre non è più fornito né il supporto alla paginazione né alla selezione e modifica dei dati.

7.5 Specializzazione dell'controllo DataList: il controllo DataListNav

Uno degli enormi vantaggi di ASP.NET rispetto al suo predecessore, è l'utilizzo di linguaggi di programmazione completamente ad oggetti: infatti anche Visual Basic .NET si è evoluto rispetto alla precedente versione e ora supporta tutti i costrutti base della programmazione ad oggetti, inclusa l'ereditarietà.

Sfruttando questa novità del linguaggio è possibile ereditare dai controlli web predefiniti in modo da aggiungere nuove funzionalità e personalizzarli a piacimento. Volendo provare questa nuova possibilità offerta dal sistema, si è deciso di ereditare dal controllo DataList al fine di aggiungervi il supporto alla paginazione.

Prima di tutto è stata creata una classe, Navigatore, che eredita dal componente Control generico. Questa permette, ridefinendo il metodo Render() della classe control, di scrivere in output il menù di navigazione.

Dopodiché è stata creata la classe DataListNav che contiene come membro privato, un oggetto Navigatore. Il risultato è raffigurato nell'immagine seguente.

Nome	Cognome	Email	Dettagli
a	b	drider@libero.it	Show details
A1	A1	a1@a1.com	Show details
AGOSTINO	TADDEI	sanremocasa@virgilio.it	Show details
agostino	conti	zconti@mail.omnitel.it	Show details
Aldina	damiani	ad71ad@hotmail.com	Show details
aldo	randazzo	mimmapo@i2000net.it	Show details
aldo	randazzo	aldo.rand@libero.it	Show details
alessandra	d'ercole	stdomus@libero.it	Show details
Alessandro	Delvai	alessandro.delvai@tin.it	Show details
alessandro	cornaro	adsyr@iol.it	Show details
alessandro	leggeri	ale969@tin.it	Show details
Alessia	Parisi	aleparisi@tiscalinet.it	Show details
Alessia	Mocci	sandalia.vacanze@tiscalinet.it	
alfredo	morselli	alfredo.morselli@tin.it	Show details
alfredo	gennaro d'agata	info@vulcanoconsult.it	Show details

1 2 3 4 5 6 ... 20

In basso è visibile il controllo navigatore che fornisce il menù di navigazione. È possibile inoltre notare l'utilizzo del supporto alla selezione per visualizzare la riga in giallo.

7.6 DataList e DataGrid innestate.

L'ultima sperimentazione con ASP.NET ha riguardato l'utilizzo combinato dei controlli DataList e DataGrid.

Si è cercato di riscrivere una pagina di gestione utenti del sito AffittiTuristici. Per ogni utente vanno visualizzate tutte le proprietà a lui associate, indicando, tramite l'ausilio di immagini colorate di verde e di rosso, su quali dei tre siti, è disponibile ciascuna proprietà.

L'effetto è ottenuto tramite l'innestamento di un controllo DataGrid all'interno di un DataList. La parte complicata consiste nell'agganciamento dei dati relativi alle proprietà di un determinato utente al componente DataGrid. Ciò è effettuato gestendo l'evento DataBound del controllo DataGrid. Questo evento viene scatenato dal controllo nel momento in cui sono stati agganciati i dati relativi ad una riga alla riga stessa. Nel gestore di questo evento viene effettuata una query al database, ottenendo le informazioni relative alle proprietà dell'utente.

Risulta inoltre abbastanza macchinoso accedere alla DataGrid interna, dato che è sempre necessario passare per l'oggetto padre e ottenere il riferimento tramite un determinato indice.

L'effetto finale è mostrato nell'immagine seguente.

Nome	Cognome	Email	Dettagli							
Emanuela	Fratini	0744272907	Show details	<table border="1"> <tr> <td>Villa Le casaline</td> <td>IT DE EN</td> </tr> <tr> <td>Casa Giovanna</td> <td>IT DE EN</td> </tr> <tr> <td>Casa Marianna</td> <td>IT DE EN</td> </tr> </table>	Villa Le casaline	IT DE EN	Casa Giovanna	IT DE EN	Casa Marianna	IT DE EN
Villa Le casaline	IT DE EN									
Casa Giovanna	IT DE EN									
Casa Marianna	IT DE EN									
Goffredo	Scotto Di Carlo	081.8968341	Show details	<table border="1"> <tr> <td>Tirreno residence</td> <td>IT DE EN</td> </tr> </table>	Tirreno residence	IT DE EN				
Tirreno residence	IT DE EN									
Paola	Tomas	043967119	Show details	<table border="1"> <tr> <td>Casa Paola</td> <td>IT DE EN</td> </tr> </table>	Casa Paola	IT DE EN				
Casa Paola	IT DE EN									
Roberto	Mezzone	0575 97384	Show details	<table border="1"> <tr> <td>Villa Diana/IL CASALE</td> <td>IT DE EN</td> </tr> <tr> <td>Villa Diana/IL RUSTICO</td> <td>IT DE EN</td> </tr> </table>	Villa Diana/IL CASALE	IT DE EN	Villa Diana/IL RUSTICO	IT DE EN		
Villa Diana/IL CASALE	IT DE EN									
Villa Diana/IL RUSTICO	IT DE EN									
Salvatore	Achenza	079 272200	Show details	<table border="1"> <tr> <td>Villa Costa Paradiso</td> <td>IT DE EN</td> </tr> </table>	Villa Costa Paradiso	IT DE EN				
Villa Costa Paradiso	IT DE EN									
Giuliana	Della Latta	0584 619739	Show details	<table border="1"> <tr> <td>Villa Giuliana</td> <td>IT DE EN</td> </tr> <tr> <td>Villa Speranza</td> <td>IT DE EN</td> </tr> </table>	Villa Giuliana	IT DE EN	Villa Speranza	IT DE EN		
Villa Giuliana	IT DE EN									
Villa Speranza	IT DE EN									
Nadia	Melidona	0165 53128	Show details	<table border="1"> <tr> <td>Lo Novesse B&B</td> <td>IT DE EN</td> </tr> </table>	Lo Novesse B&B	IT DE EN				
Lo Novesse B&B	IT DE EN									
Francesco	Maddanu	0338 2826999	Show details	<table border="1"> <tr> <td>Villa Saserra</td> <td>IT DE EN</td> </tr> </table>	Villa Saserra	IT DE EN				
Villa Saserra	IT DE EN									
MARCO	BISOGNI	096344667	Show details	<table border="1"> <tr> <td>Baia Safò</td> <td>IT DE EN</td> </tr> </table>	Baia Safò	IT DE EN				
Baia Safò	IT DE EN									
SILVIA	MORI	079671119	Show details	<table border="1"> <tr> <td>Appartamento Porto Rotondo</td> <td>IT DE EN</td> </tr> </table>	Appartamento Porto Rotondo	IT DE EN				
Appartamento Porto Rotondo	IT DE EN									

Conclusione

Effettuare quest'attività di tirocinio è stato sicuramente positivo, sia dal punto di vista delle conoscenze acquisite, sia dal punto di vista dell'esperienza di lavoro in azienda.

Sono state acquisite le conoscenze necessarie allo sviluppo di siti e applicazioni web basati su tecnologia ASP e su database Microsoft SQL Server.

Si è familiarizzato con strumenti di sviluppo professionali quali Macromedia Dreamweaver Ultradev, MS Enterprise Manager e Visual Studio .NET.

Infine si è realizzata un'attività di ricerca sul nuovo ambiente di sviluppo Microsoft .NET e in particolare sulle novità introdotte nell'ambito delle applicazioni web.

In conclusione si è trattato di un'esperienza molto positiva e molto formativa sotto tutti i punti di vista.

BIBLIOGRAFIA

“ASP Alliance”, <http://www.aspalliance.com>

“MSDN Library”, <http://msdn.microsoft.com/library>

A. Homer, D. Sussman e B. Francis: “*Professional Active Server Pages*”, Wrox Press Inc., Settembre 1999

Christian Graus “*Image Processing for Dummies with C# and GDI+ Part 1, 2, 3 and 4*”, <http://www.codeproject.com/cs/media/csharpgraphicfilters11.asp>