Università degli studi
di Modena e Reggio Emilia
Facoltà di Scienze Matematiche, Fisiche e Naturali
Corso di Laurea in Informatica

Utilizzo di ASP.Net e SQL Server 2008 per lo Sviluppo di Pagine Web Dinamiche

Signi Ferdinand
Tesi di Laurea
Anno Accademico 2009/2010

Relatore
Prof. Riccardo Martoglia

Ringraziamenti

Desidero ringraziare il Prof.Riccardo Martoglia per l'infinita disponibilità e assistenza.
Un ringraziamento speciale va alla mia famiglia ed ai miei amici che mi hanno sostenuto e sopportato.
Lode al Signore per la sua infinita bontà verso di me in questo periodo.
Grazie ai miei fratelli Cristiani per le loro preghiere e sostenio.
Grazie a tutti coloro che mi hanno dato una mano affinché sia realizzato questa tesi.

Parole chiave

ASP.NET
SQL Server 2008
Pagina Dinamica
Database
Web

Indice

Elenco delle figure **7**

Elenco delle tabelle **9**

Introduzione **11**

I Tecnologie Web **12**

1 La Tecnologia ASP **13**
1.1 Introduzione 13
1.2 Elaborazione di una richiesta ASP 13
1.3 Linguaggi di scripting 14
1.4 Oggetti ASP 14
1.4.1 L'oggetto Server 15
1.4.2 L'oggetto Application 15
1.4.3 L'oggetto Session 17
1.4.4 L'oggetto Request 17
1.4.5 L'oggetto Response 18
1.4.6 L'oggetto ASPError 18

2 La Tecnologia ASP.NET **21**
2.1 Introduzione 21
2.2 Modello d'elaborazione degli applicazioni ASP.NET 21
2.3 L'approccio componente di ASP.NET 22
2.3.1 Componenti di gestione dell'interfaccia grafica 23
2.3.2 Componenti per la geszione del contenuto 25
2.3.3 Componenti per la gestione della sicurezza 26

3 Interfacciamento di una base di dati **27**
3.1 Introduzione 27
3.2 Interfacciamento con ASP3.0 27

3.2.1 Connessione al database: l'oggetto CONNECTION . . 28
3.2.2 Ottenere Record : l'oggetto RECORDSET 29
3.2.3 Modificare i dati: l'oggetto COMMAND 32
3.3 ASP.Net e i fonti di dati:ADO.Net 33
3.3.1 DOT NET Data Providers 34
3.3.2 Collegamento e manipolazione di una fonte di dati . . 35
3.3.3 ACCESSO ai dati : la classe DataReader 37
3.3.4 La Classe DataAdapter 37
3.3.5 L'oggetto DataSet 38

II Reingegnerizzazione del sito web di ISGroup mediante ASP.NET e SQL Server 2008 40

4 Progetto 41
4.1 Il sito web di ISGroup 41
4.1.1 Analisi script ASP 42
4.1.2 Schema E-R del sito 43
4.1.3 Schema logico del sito 44
4.1.4 Relazioni tra le tabelle 44
4.2 Progetto del sito 45

5 Implementazione 47
5.1 Ambiente di sviluppo 47
5.2 Area grafica del sito 47
5.2.1 Il blocco d'intestazione 48
5.2.2 Il blocco pagina 49
5.2.3 Il blocco pie di pagina 49
5.3 Script aspx delle pagine 51
5.3.1 Script della pagina Project.aspx 51
5.3.2 Script della pagina Thesis.aspx 55
5.3.3 Script della pagina Publications.aspx 60

Conclusione 65

III Appendici 66

Appendici 67
5.4 Il Codice aspx 67
5.4.1 MasterPage.master 67
5.4.2 Default.css 67

5.4.3 Default.skin 67
5.4.4 Projects.aspx 67
5.4.5 These.aspx 67
5.4.6 Publications 67

Elenco delle figure

1.1 Elaborazione di una richiesta 14
1.2 Gli oggetti ASP e communicazione fra di loro 15

2.1 Compilazione ed esecuzione di una applicazione ASP.NET . . 22
2.2 Generazione del codice HTML adatto ad ogni browser 23
2.3 Modello a componenti di ASP.NET 24

3.1 Collegamento mediante OLE DB Provider per ODBC 29
3.2 Creazione di un record set 30
3.3 Creazione di un record set con active connection 31
3.4 Utilizzo del metodo Recordset.Open 31
3.5 Esempio di creazione di un'oggetto Command 32
3.6 Utillizzo di CommandText 32
3.7 Utillizzo di ActiveConnection e Execute() 33
3.8 Architettura di ADO.NET 34
3.9 Utilizzo della classe SqlConnection 36
3.10 Utilizzo della classe SqlCommand 37
3.11 Creazione e utilizzo di un DataReader 38
3.12 Creazione e utilizzo di un DataSet 38

4.1 Schema E-R del sito 43
4.2 Schema delle relazioni tra le tabelle 45
4.3 Schema del sito 46

5.1 Codice dell'intestazione della pagina master 48
5.2 Codice dell'intestazione 48
5.3 Codice del blocco pagina 49
5.4 Codice del blocco pie di pagina 49
5.5 Codice della pagina Default.aspx 50
5.6 Codice behind della pagina Default.aspx 50
5.7 Pagina Default.aspx nel browser 50
5.8 Collegamento alla fonte di dati 52

5.9 Data Binding e presentazione nella DataList 52
5.10 Selezione di un topic 52
5.11 Detagli sul topic selezionato 53
5.12 Creazione della Datasource e collegamento alla DropDownList 53
5.13 Codice di presentazione dei dati estratti 53
5.14 Codice della parte relativa ai progetti Europei 54
5.15 Risultato ottenuto 54
5.16 Schema de Proposed Theses 55
5.17 Codice collegamento alla base di dati 56
5.18 Codice presentazione dei risultati 56
5.19 Selezione di un autore 57
5.20 Informazioni sulla tesi dell'autore selezionato 57
5.21 Codice del collegamento alla base di dati 58
5.22 Codice presentazione dei risultati 58
5.23 Selezione di un autore 59
5.24 Informazioni sulla tesi dell'autore selezionato 59
5.25 Parte dello script di costruzione della Web form 60
5.26 Web form dovuto alla ricerca 60
5.27 Codice parziale della Stored Procedure 61
5.28 Data Source costruitta con la stored procedure 61
5.29 Scelta dell'autore 62
5.30 Scelta del topic 63
5.31 Resultati ottenuti 63
5.32 Risultati per tutte le pubblicazioni 64

Elenco delle tabelle

1.1 Metodi dell'oggetto Server 16
1.2 Metodi dell'oggetto Application 16
1.3 Proprietà dell'oggetto Application 16
1.4 Collezioni dell'oggetto Application 16
1.5 Metodi dell'oggetto Session 17
1.6 Proprieta dell'oggetto Session 17
1.7 Metodi dell'oggetto Response 18
1.8 Proprietà dell'oggetto Response 19
1.9 Proprietà dell'oggetto ASPError 19

Introduzione

Questa tesi nasce dal mio desiderio ardente di lavorare nei campi dei basi di dati per i quali nutrisco una vera passione in genere e in particolare sul loro interfacciamento con un linguaggio di programmazione di alto livello. Pure tra i soggetti proposti dal mio relatore , ho scelto questo perchè era l'occazione per me di soddisfare la mia passione ma soppratutto imparare questa tecnologia che offre il Framework .NET di Microsoft e che sta crescendo ogni giorno. Infatti , il sito web di *ISGroup(Information Systems Group)* è stato svilupato per permettere agli insegnanti del Dipartimento dell'Ingegneria dell'Informazione(DII) che lavorano nel dominio dei sistemi dell'informazione di publicare e diffondere le loro ricerche. Questa piattaforma è stata sviluppata nel 2003 con *ASP(Active Server Pages)* una delle tecnologie proposta dal Microsoft per lo sviluppo delle pagine Web dinamiche e si appogia su una base di dati Microsoft Access 2000.

Il modello d'elaborazione di ASP è un modello interpretato e di conseguenza lento dovuto al fatto che ogni riga viene interpretato prima dell'esecuzione. Lo stesso, il DBMS Access non gestisce in maniera ottimizzata l'accesso multipla e concorrente che caratterizza un sito web.

In questa tesi presentiamo brevemente le tecnologie ASP e ASP.NET e proponiamo una possibile migrazione del sito verso il Framework .NET utilizzando il DBMS SQL Server 2008. Mettiamo in opere questa soluzione attraverso le pagine dinamiche del sito.

Questa tesi è strutturalmente organizzata in cinque capitoli, suddivisi in due parti. Nella **prima** parte, in *capitolo 1* studieremo la tecnologia ASP. In *capitolo 2* presentiamo la nuova tecnologia ASP.Net e le confronteremo affinchè sia giustificata la nostra scelta della seconda parte. In *capitolo 3* presentiamo l'interfacciamento dei basi di dati usando ADO e ADO.NET. Nella **seconda** parte e precisamente nel *capitolo 4* si svolge l'attività progettuale del sito web del gruppo *ISGroup*. In *capitolo 5* presentiamo come abbiamo sfruttato ASP.Net e SQL Server 2008 per sviluppare le pagine dinamiche del sito web del gruppo *ISGroup*. La conclusione presenta in breve il lavoro svolto e gli sviluppi futuri.

Parte I

Tecnologie Web

Capitolo 1

La Tecnologia ASP

1.1 Introduzione

L'ASP, acronimo di **Active Server Pages (Pagine Server Attive)**, è una tecnologia (lato server) di scripting introdotta da Microsoft nel 1996 che permette di creare pagine Web dinamiche. Queste pagine, oltre a contenere puro codice HTML, contengono anche degli script che verranno interpretati dal server per generare il codice da inviare al browser dell'utente. Per poter eseguire pagine ASP è necessario installare un componente di Windows, chiamato *IIS (Internet Information Services)*, il quale contiene una libreria di funzioni chiamata ASP.DLL che consente al Web Server di interpretare il codice ASP e fornire in output una pagina HTML.

1.2 Elaborazione di una richiesta ASP

L'elaborazione di una richiesta ASP si svolge secondo il modello di seguito rappresentato in Figura 1.1. Quando il cliente desidera accedere ad una pagina dinamica realizzata con la tecnologia ASP, essa si collega al server web facendo la sua domanda. Il Web Server che riconosce dall'estenzione *.asp* che si tratta di un file ASP lo legge riga doppo riga e trasmette al suo motore che lo esegue e manda gli output al server. Alla fine il Web Server manda i risultati al browser utente. Uno script ASP è quindi interpretati dal server e non ci sono modifiche da fare sui browser utenti. Inoltre gli utenti non possono accedere al codice sorgente che sono composti di HTML e degli altri linguaggi di scripting.

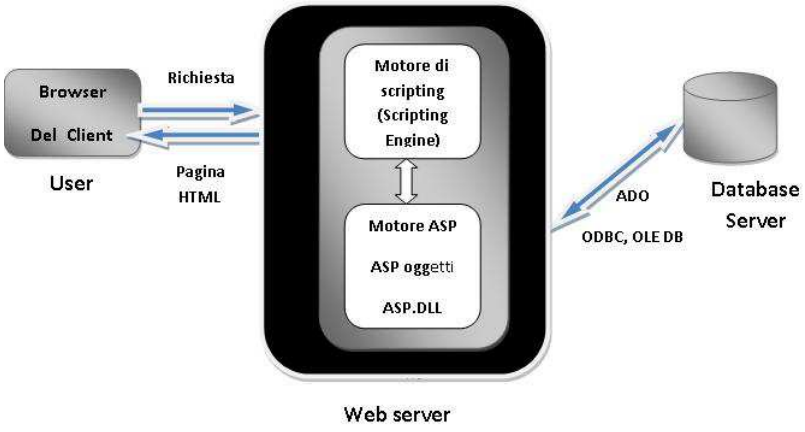

Figura 1.1: Elaborazione di una richiesta

1.3 Linguaggi di scripting

ASP supporta in modo nativo due linguaggi di scripting, VBScript e JScript, per lo sviluppo delle pagine dinamiche. Le sezioni di scripting sono dichiarate all'interno dei tag <SCRIPT> e </SCRIPT> usando la clausola ***language*** oppure all'interno dei simboli <%Codice dello script%>.

Esempio: <SCRIPT language=VBScript|Jscript runat=server src='url'> codice dello script </SCRIPT>

Il tag ***language*** é mandatorio e definisce il linguaggio dovendo essere utilizzato durante l'interpretazione. Il tag opzionale ***runat*** definisce dove lo script sarà interpretato. Quando ***runat***=*client*, il codice sarà mandato al cliente senza interpretazione. È quindi importantissimo impostarlo al server se vogliamo che l'interpretazione sia fatta dal server. Il tag opzionale *src* ci permette di chiamare una porzione di codice situata all'indirizzo definito dall'url. La seconda dichiarazione é una maniera simplificata di definire il blocco di scripting. Sarebbe necessario utilizzare la direttiva <%@ LANGUAGE %> all 'inizio del file per spostare il linguaggio dello script usato.

1.4 Oggetti ASP

Una Caratteristica fondamentale del linguaggio ASP è l'utilizzo di un modello di programmazione ad oggetti permettendogli di aumentare il livello di astrazione delle pagine sfruttando l'ereditarietà, l'incapsulazione e le altre tecniche che offrono questo tipo di progettazione. Gli oggetti ASP costituiscono l'essenziale del motore di scripting e sono composti da tre tipi di

entità: le collezioni, le proprietà e i metodi. Lato server si trovano gli oggetti ***SERVER*** , ***APPLICATION*** e ***SESSION*** . La comunicazione fra server e l'utente si realizza tramite gli oggetti ***REQUEST*** e ***RESPONSE*** come illustrato nella Figura 1.2. Quando il browser utente manda una richiesta al server ad una certa porta una sessione è creata all'interno dell'applicazione identificata da questo numero di porta. L'applicazione si esegue e può fare anche una chiamata alla base di dati tramite ADO doppo di che la risposta è mandata al cliente attraverso l'oggetto ***RESPONSE***.

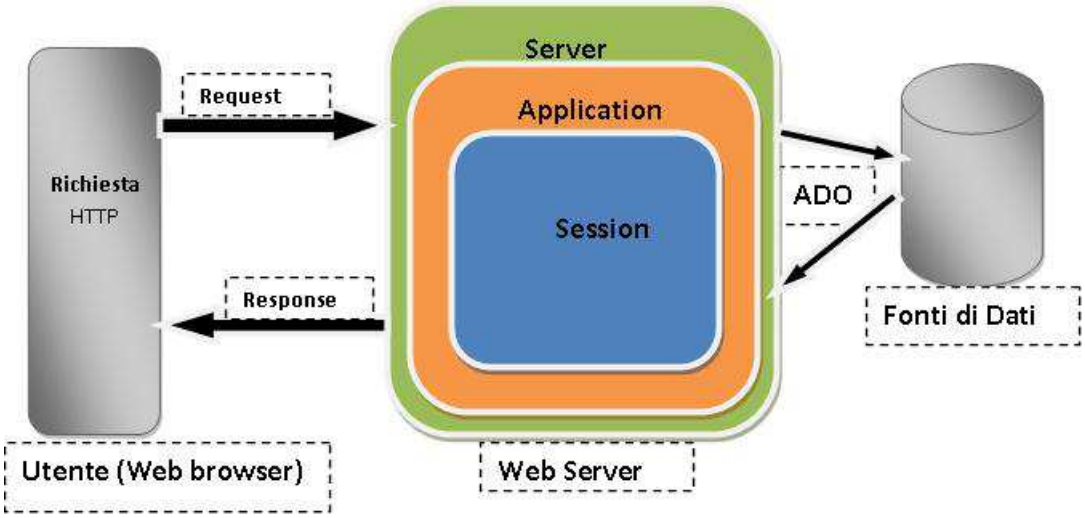

Figura 1.2: Gli oggetti ASP e communicazione fra di loro

1.4.1 L'oggetto Server

L'oggetto ***Server*** permette l'uso dei componenti Activex lato server nelle applicazioni web. Con questo oggetto è possibile istanziare i componenti scritti in Visual basic, Visual C++ oppure gli altri ambiente di sviluppo permettendo la creazione dei componenti COM. I principali metodi di questo oggetto sono presentati nella tabella 1.1.

La principale proprietà di questo oggetto é *ScriptTimeout* che Imposta o restituisce il numero massimo in secondi per completare uno script.

1.4.2 L'oggetto Application

L'oggetto Application permette di condividere le informazioni fra utenti della stessa applicazione. I suoi metodi, proprietà, eventi e collezioni sono elencati nelle tabelle 1.2, 1.3, e 1.4 rispettivamente.

Nome	Descrizione
CreateObject(Type_Objet)	Istanzia un oggetto Server
HTMLEncode(car)	Applica la codifica HTML ad la stringa car
MapPath(path)	Mappa un percorso logico in un percorso fisico
Execute(path)	Esegue il file .asp avendo il path specificato
GetLastError()	Restituisce un oggetto che descrive l'errore
Transfer(path)	Trasferisce l'informazione fra pagine ASP
URLEncode(string)	Applica la codifica URL ad una stringa

Tabella 1.1: Metodi dell'oggetto Server

Nome	Descrizione
Lock	Impedisce di modificare le proprietà dell'applicazione
UnLock	Autorizza di a modificare le proprietà dell'applicazione

Tabella 1.2: Metodi dell'oggetto Application

Nome	Descrizione
Contents.count	Numero degli elementi nella collezione
Contents.item(key)	Elemento identificato dall'argomento key
StaticObjects.count	Numero degli elementi nella collezione
StaticObjects.item(key)	Elemento di StaticObjects identifico da key

Tabella 1.3: Proprietà dell'oggetto Application

Nome	Descrizione
Contents	Insieme degli elementi aggiunti all'applicazione
StaticObjects	Oggetti aggunti attraverso il tag <object>

Tabella 1.4: Collezioni dell'oggetto Application

Nome	Descrizione
Abandon	Termina una sessione e distrugge tutte le variabili
Contents.Remove(key)	Elimina un dato elemento della collezione
Contents.RemoveAll()	Elimina tutti gli elementi della collezione

Tabella 1.5: Metodi dell'oggetto Session

Nome	Descrizione
CodePage	Specifica il set di caratteri utilizzato
Contents.count	Restituisce il numero degli elementi nella collezione
Contents.item(key)	Restituisce l'elemento identificato da key
LCID	Imposta l'internazione di una sessione
StaticObjects.count SessionID	Restituisce un identificativo per ogni utente
StaticObjects.count	Restituisce il numero degli elementi nella collezione
StaticObjects.item(key)	Restituisce l'elemento identificato dall'argomento key
Timeout	Restituisce/imposta il timeout per l'oggetto sessione

Tabella 1.6: Proprieta dell'oggetto Session

1.4.3 L'oggetto Session

Una sessione rappresenta un periodo di tempo nella quale un utente sta navigando fra un ensieme di pagine web ASP. L'oggetto ***SESSION*** permette di definire le variabili, costanti,e procedure permettendo di memorizzare le informazioni durante questa sessione. L'elenco dei suoi metodi è presentato nella tabella 1.5.

L'oggetto *Session* utilizza le collezioni *StaticObjects* e *Contents*.

1.4.4 L'oggetto Request

L'oggetto request riceve le informazioni mandate dal browser tramite una richiesta http. Le informazioni possono provenire da :

- Un modulo
- I cookies
- I QueryStrings nell'URL(Uniform Ressource Locator)
- Le variabili dell'ambiente HTTP
- Certificati utenti per le transazioni sicure on line

Nome	Descrizione
AddHeader(nome, valore)	Aggiunge un valore nell'intestazione HTTP
AppendToLog(input)	Aggiunge un testo per una richiesta nel giornale
BinaryWrite(Testo)	Manda un testo al browser senza converzione
Clear()	Cancella tutto il contenuto del cash
End()	Interrompe il trattamento ASP e riinvia il risultato
Flush()	Manda subito il contenuto del cash
Redirect(URL)	Ordina al browser di collegarsi a un'altro URL
Write(Testo)	Scrive una variabile string sulla pagina

Tabella 1.7: Metodi dell'oggetto Response

L'accesso ad una variabile dell'oggetto Request si fa referendo l'oggetto prima, poi la collezione e infine la variabile voluta(REQUEST[Collection](Variable)). Il metodo *BinaryRead()* è utilizzato per leggere i dati mandati dal cliente tramite una richiesta POST. La proprietà *TotalBytes* indica la quantità di bytes mandati dal cliente nel corpo del messagio della richiesta. Le Collezioni usati dall'oggetto ***Request*** sono:

- *ClientCertificate*: Valori contenendo i campi del certificato dell'utente
- *Cookies*: Valori dei cookies inviati dal browser
- *Form* : Valori degli elementi di un modulo inviati dal browser
- *QueryString*: Valori delle variabili di una catena di richiesta http
- *ServerVariables* : Valori delle variabili dell'ambiente e HTTP

1.4.5 L'oggetto Response

L'oggetto Response effettua il trattamento e rimanda i risultati al cliente. Le tabelle 1.7 e 1.8 presentano i suoi metodi, e le sue proprietà.

1.4.6 L'oggetto ASPError

L'oggetto ASPError contiene le informazioni sugli errori nello script ASP. Questo oggetto é stato creato specialmente per gestire gli errori lato server. La tabella 1.9 presente diverse proprietà di quest'oggetto.

In questo capitolo abbiamo dato una descrizione breve dei principali oggetti della tecnologia ASP. Il capitolo seguente presenta una tecnologia

Nome	Descrizione
Buffer[True|False]	Indica se le pagine debano essere in cash
CacheControl(Private|Public)	Indica dove mettere i file generati da ASP nella cash
Charset(nom_jeu)	Aggiunge il nome del Charset all'intesta del contenuto
ContentType[Type_Contenu]	Indica il valore dell'intesta HTTP
Expires Nombre_minutes	Indica la scadenza di una pagina nel cach
ExpiresAbsolute	Indica la data e l'ora di scadenza di una pagina
IsClientConnected()	Fornisce le informazioni sulla validità della conessione
Status	Indica il valore della linea di statuto HTTP

Tabella 1.8: Proprietà dell'oggetto Response

Nome	Descrizione
ASPCode	Codice dell'errore generato da IIS
Number	Numero dell'errore COM standard
Source	Codice sorgente responsabile dell'errore
Category	Indica il codice responsabile dell'errore
Line	Numero della linea avendo generato l'errore
Column	Numero della colonna avendo generato l'errore
Description	Descrizione dell'errore
ASPDescription	Descrizione dettagliata dell'errore ASP

Tabella 1.9: Proprietà dell'oggetto ASPError

nata dopo la versione ASP 3.x e basata sul Framework .NET di Microsoft e migliorando quella presentata precedentemente.

Capitolo 2

La Tecnologia ASP.NET

2.1 Introduzione

La tecnologia ASP.NET è nata dopo la versione ASP 3.x ed è basata sul Framework .NET di Microsoft. ASP.Net è un approccio componente per lo sviluppo di applicazioni web. Prende in conto le capacità rispettive dei diversi browser. Dot Net è una piatta forma completa per sviluppare distribuire e gestire applicazioni Web, Windows, Mobile e Server. Il Framework .Net propone un modello di programmazione unificato. Il Framework .NET è gratuito ed integrato a Windows. É istallato di default su Windows Server 2003 ed è istallabile gratuitamente attraverso Windows .Net su Windows XP e Windows 2000 secondo le condizioni di service pack.

2.2 Modello d'elaborazione degli applicazioni ASP.NET

Le applicazioni ASP.NET sono eseguite all'interno del CLR [1] che fornisce servizi molto importanti che consentono al Framework .Net di fare magie. Il CLR è un livello tra l'applicazione e il sistema operativo sul quale si esegue. Il CLR semplifica la progettazione delle applicazioni e riduce la quantità di codice che gli sviluppatori devono scrivere perché offre una varietà di servizi d'esecuzione che comprendono la gestione della memoria, gestione dei thread, e di default la gestione degli errori. Il vantaggio principale del CLR è che fornisce in modo trasparente i servizi di esecuzione a tutte le applicazioni, indipendentemente dal linguaggio di programmazione usato e senza alcun sforzo supplementare da parte dello sviluppatore. Il CLR è anche respons-

[1] Common Language Runtime

abile della compilazione del codice prima della sua esecuzione. Invece di produrre una rappresentazione binaria del codice come i compilatori tradizionali, il compilatore produce un codice in un linguaggio comune al Framework .NET : il Microsoft Intermediate Language(MSIL) spesso chiamato Intermediate Language(IL). Quando il codice viene eseguito per la prima volta, il CLR richiama un compilatore speciale chiamato Just In Time (JIT), che trasforma l'IL in istruzioni eseguibili e ottimizzate per il tipo e il modello del processore del sistema. Poiché tutti .NET hanno la stessa rappresentazione compilata, hanno tutti caratteristiche e prestazioni simili. Questo significa che un programma scritto in Visual Basic. NET avrà la stessa prestazione di quella scritta in Visual C + +, . NET e vice versa. La Figura 2.1presenta graficamente il processo di compilazione e d'esecuzione di una applicazione ASP.NET.

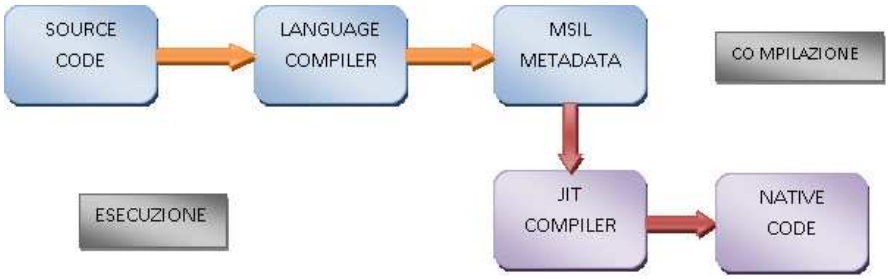

Figura 2.1: Compilazione ed esecuzione di una applicazione ASP.NET

Ogni pagina ASPX é quindi compilata attraverso il Framework .Net per generare un codice HTML standard appropriato per ogni tipo di browser del mercato. Questo processo rapresentato dalla Figura 2.2 permette di ottimizzare e velocizzare le applicazioni Web basate sul Framework .Net.

Per poter facillitare la progettazione delle applicazioni, ASP.NET permette di costruire pagine Web dinamiche utilizzando un approccio per componenti.

2.3 L'approccio componente di ASP.NET

ASP.Net permette di costruire pagine Web dinamiche sfruttando l'architettura presentata nella Figura 2.3. I controlli permettono di gestire la sicurezza, l'utenza e l'interfacciamento dei fonti di dati. In breve questi controlli permettono la gestione del contenuto. La presentazione è gestita dalla pagina Master(che erediterano le pagine figli), i temi(che includono i fogli di stili, imagini,etc) e skins chi definiscono lo scheletro del sito. L'internazional-

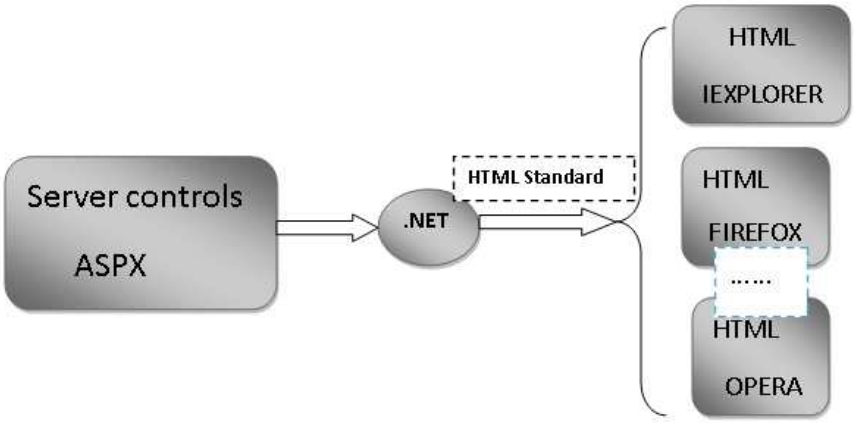

Figura 2.2: Generazione del codice HTML adatto ad ogni browser

isazione permette di costruire pagine che si eseguono nel linguaggio dell'utente. ASP.NET separa il codice della presentazione grafica utilizzando la tecnica di ***CODE BEHIND***. Questo permette ai grafici di occuparsi dell'aspetto grafico mentre i programmatori si occupano del contenuto scrivendo il codice VB.Net, C Sharp, Perl.Net etc.

2.3.1 Componenti di gestione dell'interfaccia grafica

Temi e Skins

I temi combinano tutti gli elementi che definiscono l'interfaccia grafica. L'interfaccia grafica è gestita dai fogli di stili CSS [2], il file *.skin* chiamato *appearence control file* e la mappa grafico del sito chiamato *web.sitemap* che fornisce per ogni pagina il nome associato e le immagini utilizzate nel sito. I componenti presenti nel file *.skin* possono essere:

1. il componente <asp:SiteMapPath> dà il percorso della pagina corrente

2. il componente <asp:Menu> permette di visualizzare il menu orizzontale

[2]Cascading Style Sheet

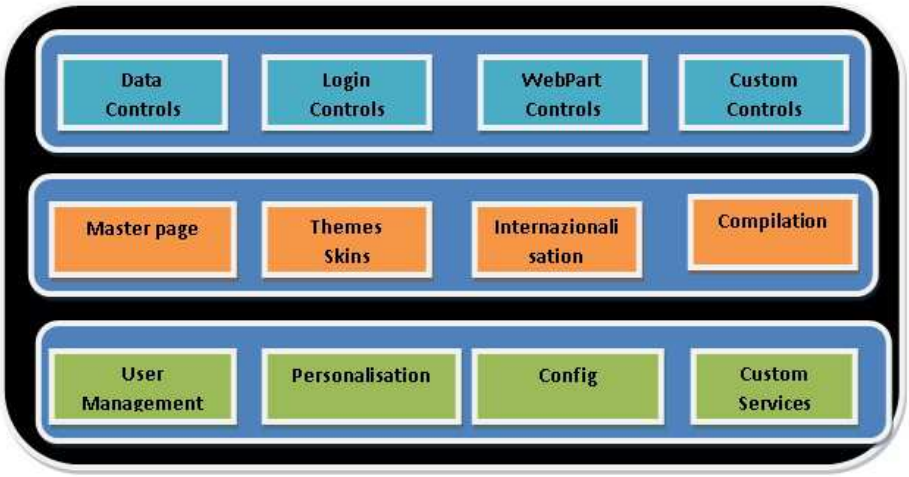

Figura 2.3: Modello a componenti di ASP.NET

3. il componente <asp:TreeView> permette di visualizzare il menu sulla sbarra di destra o sinistra

L'internazionalisazione

L'Internazionalizzazione è il processo di progettazione e d'integrazione di supporto di una determinata lingua o cultura in un'applicazione. La piatta forma Microsoft ASP.Net fornisce il supporto esteso per internazionalizzare le applicazioni standard, integrando lo standard appogiato sul Framework Microsoft AJAX. Sfrutando l'internazionalizatione, gli sviluppatori possono creare applicazioni che si eseguono nella lingua dell'utente. Progettare componenti adattabili alla lingua del cliente migliora la produttività e riduce la quantità di lavoro necessario per adattare un componente di operare a livello globale.

La pagina Master

Ci sono diverse tecniche per la condivisione del formato grafico tra più pagine (comprese le UserControl). Il concetto di pagina Master è una novità di ASP.NET 2.0 ed è stato progettato specificamente per semplificare il factoring del Codice per quanto riguarda la rappresentazione grafica delle pagine. L'obiettivo essendo di migliorare il disign, l'elaborazione, l'evoluzione e la manutenzione delle pagine. Quindi la pagina Master ragruppa tutte le parte

delle pagine che non cambiano. Essa prevede una parte modificabile [3] incapsulata all'interno dei tag <asp:ContentPlaceHolder> e <asp:ContentPlaceHolder> permette anche ai programatori di cambiare alcune funzionalità tramite i suoi metodi *public*.

2.3.2 Componenti per la geszione del contenuto

Il contenuto è gestito dai componenti d'accesso e di presentazioni dei dati.

Componenti d'accesso ai fonti di dati

ASP.Net offre una moltiplicatà di componente d'accesso ai fonti di dati tra le quali:

- il componente <asp:SqlDataSource> che permette di collegarsi ad una fonte di dati SQL Server.
- il componente <asp:SiteMapDataSource> che permette di collegarsi ad una fonte di dati utilizzato per la mappa del sito.
- il componente <asp:XmlDataSource> che permette di collegarsi ad una fonte di dati nel formato XML.
- il componente <asp:ObjectDataSource> che permette di collegarsi ad un oggetto che incapsula la logica mestiere.

Componenti di presentazioni dei dati

Dopo l'esecuzione di una dichiarazione SQL i risultati[4] possono essere presentati usando i conponenti seguenti:

1. il componente <asp:GridView> presenta i dati sotto forma di tabelle
2. il componente <asp:DataList> presenta i dati secondo il bisogno del programmatore usando il tag <ItemTemplate>
3. il componente <asp:DetailsView> presenta i dati a partire di quelli scelti nel GridView attraverso il databinding[5]
4. il componente <asp:ListView> presenta i dati sotto forma di una lista

[3] il contenuto di ogni pagina
[4] insieme di record
[5] collegamento dei dati

2.3.3 Componenti per la gestione della sicurezza

ASP.Net offre una multitudine di controlli server per la creazione di utente, l'impostazione dei suei dati, la connessione al sito web, la modifica della password etc. La creazione di une utente si fa attraverso il controllo server <asp:CreateUserWisard>. L'impostazione dei dati di connessione si svolge usando i controlli <asp:LoginName> e <asp:ChangePassWord> mentre la recuperazione della password persa si fa tramite il controllo <asp:PasswordRecorvery>.

In questo capitolo, abbiamo presentato in breve la tecnologia ASP.NET. Precisamente, abbiamo presentato la sua architettura e i suoi componenti importanti. ASP.Net è una tecnologia potente che separa la presentazione della logica, è event driven e compilato. ASP invece gruppa la presentazione e la logica, il suo codice è sequenziale e interpretato. Il capitolo che segue presenterà l'interfacciamento dei basi di dati usando ASP e ASP.Net.

Capitolo 3

Interfacciamento di una base di dati

3.1 Introduzione

L'interfacciamento dei basi di dati è sicuramente l'aspetto piu importante dei linguaggi di programmazione e soppratutto nei linguaggi di programmazione web dinamici. In questo capitolo presentiamo l'interfacciamento da parte di ASP prima e poi da parte di ASP.net. Infine confrontiamo i due metodi per fare la scelta migliore per il proseguimento di questa tesi.

3.2 Interfacciamento con ASP3.0

Un server Web con ASP in esecuzione si connette ai database per mezzo della tecnologia ADO [1] che consente di accedere ai dati senza fatica in modo coerente. In oltre ADO permette di lavorare con molti tipi diversi di fonti e non solo con i dati nei database. Ogni oggetto nel modello ADO rappresenta un elemento particolare che viene usato quando un programma lavora con i dati e non presenta alcuna interfaccia grafica all'utente. ADO si usa esclusivamente nel codice. La parte più onerosa connessa al lavoro da svolgere con ADO di solito è configurare le proprietà di ogni oggetto. Poi si chiamano i metodi degli oggetti per chiedere alla fonte di recuperare i dati desiderati, aggiungere nuovi dati e cancellare o modificare dati esistenti. Si segue un processo a quattro stadi quando si usa ADO per lavorare con i basi si dati:

1. La concessione al database

[1] Activex Data Objects

2. La definizione dei dati desiderati
3. La manipolazione dei dati
4. La visualizzazione dei dati

Per poter lavorare con ADO bisogna conoscere o la tecnologia OLE DB oppure ODBC. Infatti ODBC [2]permette a database di produttori diversi di lavorare insieme. E un interprete grazie al quale tutti I database sembrano uguali l'uno all'altro e anche agli occhi dei programmi che devono interagire con essi indipendentemente dal lavoro interno ad ogni database. Il vantaggio insito nell'uso di ODBC è che occorre imparare solo una metodologia per lavorare con molti database diversi. In pratica, tutti i produttori di database forniscono software (OBDC driver) da installare che permette ai loro database di interfacciarsi con ODBC.

OLE DB [3] è una tecnologia di nuova generazione che migliora ODBC. Per quanto potente, ODBC ha un limite significativo: opera solo con I database. Oltre ai database, OLE DB permette di lavorare con le altre fonti di dati (sistemi di email ; sistemi di indicizzazione di documenti sul disco rigido) nello stesso modo in cui si lavora con i database standard. OLE DB rappresenta la prima passo verso l'implementazione di Universal Data, modello in cui tutti i dati possono essere gestiti nello stesso modo independentemente dalla fonte. OLE DB interagisce con i dati attraverso uno strato di software chiamato OLEDB Provider che è simile a un driver ODBC poiche fornisce un meccanismo di conversione delle strutture di dati per aderire ad uno standard comune ampiamente supportato. I produttori di fonte di dati forniscono gli OLE DB provider driver ODBC.

3.2.1 Connessione al database: l'oggetto CONNECTION

Creare una connessione a una fonte di dati è la prima cosa che si deve fare quando si usa ADO. Ci si connette a una fonte di dati mediante l'oggetto Connection. Per creare un oggetto Connection, si usa la funzione *createObject(). CreateObject()* richiede un parametro, una stringa che indica l'oggetto da creare. Una stringa di connessione fornisce ad un oggetto connection le informazioni di cui ha bisogno per il collegemento ad una fonte di dati. Una stringa di connessione è composta da una serie di copie *nome=valore*. Ogni

[2]Open DataBase Connectivity
[3]Object Linking and Embeding DataBases

copia nome=valore è separata dalla sucessiva per mezzo di un punto e virgola. Queste copie *nome=valore* indicano all'oggetto Connection la fonte di dati da usare, che tipo di OLE DB Provider utilizzare , nonche ID utente e password necessari per accedere alla fonte di dati, qualora quest'ultima richiedesse tali informazioni. Il listato della Figura 3.1 che segue mostra una pagina ASP che contiene collegamento alla fonte di dati ISGroupDB mediante un OLE DB Provider per ODBC. ISGroupDB è il nome della fonte di dati ODBC relativa ad Information Systems Group.

```
<SCRIPT language=vbscript runat=Server>
  Dim conn ' Variabile che fa riferimento all'oggetto di connessione
  Dim strConn' Variabile che fa riferimento ad una stringa
    'Creazione dell'oggetto
  Set conn=CreateObject ("ADODB.Connection")
    'Creazione di una di connessione
  strConn = "Provider=MSDASQL; Data Source=ISGroupDB;"
  StrConn=strConn & "User=Username; Password=User Password"
   'Assegnazione della stringa alla proprieta stringa di connessione
  conn.ConnectionString=strConn
    'Apertura della connessione
  conn.Open
    'Chiusura della connessione
  conn.Close
</SCRIPT>
```

Figura 3.1: Collegamento mediante OLE DB Provider per ODBC

3.2.2 Ottenere Record : l'oggetto RECORDSET

Si usa l'oggetto Recordset per memorizzare I dati restituiti dal database definito nell'oggetto Connection. Un recordset è una collezione di record. Si puo pensare a un record come una riga di dati da un foglio elettronico di excel dove le colonne sono chiamate campi e le righe Record. Si recuperano applicando un'istruzione SQL alla fonte di dati a cui è collegato l'oggetto connection. Il database elabora l'istruzione SQL restituisce i dati conformi alle specifiche dell'istruzione SQL. Si passa l'istruzione SQL come parametro del metodo Open dell'oggetto Recordset. Sono cinque le fasi da compiere per estrare dati da una fonte di dati e visualizzarli in una pagina ASP. Tali fasi sono :

- Creare un oggetto Recordset usando *CreateObject()*.
- Creare un'istruzione SQL che definisca i dati da recuperare.

- Definire un' oggetto Connection per l'oggetto Recordset da usare come connessione attiva.
- Usare il metodo *open ()* dell'oggetto Recordset, passando l'istruzione SQL, per popolare il Recordset con i dati provenienti dalla fonte di dati.
- Examinare Recorset per creare il codice HTML che formatta la visualizzazione dei dati.

Creazione di un'oggetto Recorset

Si crea un oggetto Recorset usando il metodo CreateObject(). Un Recordset è un oggetto distinto all'interno della libreria di oggetti ADO. Il codice in Figura 3.2 crea un oggetto Recordset e lo assegna la variabile resultSet.

```
<SCRIPT language=vbscript runat=Server>
    Dim resultSet 'variabile per un oggetto Recordset
    'Creazione di un oggetto Recordset
    set resultSet=CreateObject("ADODB.Recordset")

</SCRIPT>
```

Figura 3.2: Creazione di un record set

Usare Recordset.ActiveConnection

Un oggetto Recordset deve incapsulare un oggetto Connection aperto per conoscere il database con cui lavorare. Si usa ActiveConnection per associare un oggetto connection a un oggetto Recordset. Il codice della Figura 3.3 crea un oggetto Connection e un oggetto Recordset. Poi assegna l'oggetto Connection alla proprietà ActiveConnection dell'oggetto Recordset.

Usare Recordset.Open

Una volta in possesso di un oggetto Connection impostato in modo da fornire una connessione attiva a recordset, si puo eseguire un'istruzione SQL nei confronti del database usando il metodo Open dell'oggetto Recordset. Il metodo open richiede come argomento l'istruzione SQL che descrive I dati da recuperare. Il seguente codice della Figura 3.4 ne presenta un esempio.

```
<SCRIPT language=vbscript runat=Server>
    Dim resultSet 'variabile per un oggetto Recordset
    Dim conn 'variabile per l'oggetto connection
    set conn= createObject("ADODB.Connection")
    'Creazione di un oggetto Recordset
    set resultSet=createObject("ADODB.Recordset")
    'Associa l'oggetto Connection all'oggetto Recordset
    resulSet.ActiveConnection=conn
</SCRIPT>
```

Figura 3.3: Creazione di un record set con active connection

```
<SCRIPT language=vbscript runat=Server>
    Dim resultSet 'variabile per un oggetto Recordset
    Dim conn 'variabile per l'oggetto connection
    Dim SQL
    set conn= CreateObject("ADODB.Connection")
    'Creazione di un oggetto Recordset
    set resultSet=createObject("ADODB.Recordset")
    'istruzione sql da eseguire
    SQL= "SELECT TITOLO FROM TOPICOFINTEREST"
    resultSet.Open SQL
    'oppure per la versione condensata
    resultSet.Open=SQL,conn
</SCRIPT>
```

Figura 3.4: Utilizzo del metodo Recordset.Open

3.2.3 Modificare i dati: l'oggetto COMMAND

Si usa l'oggetto Command di ADO per aggiungere dati ad una fonte di dati ODBC. L'oggetto Command rappresenta un'istruzione che si vuole eseguire nei confronti di un database. Si usa l'oggetto Command per aggiungere nuovi dati, modificare e cancellare dati esistenti. Si usa l'oggetto command unitamente alle istruzioni SQL che si creano per descrivere la funtionalita che si vuole eseguire. Una volta definita l'istruzioni SQL, si assegna il testo dell'istruzione alla proprietà Command Textdell'oggetto Command. Poi si chiama il metodo Execute dell'oggetto Command.

Creazione di un'oggetto Command

Si crea un oggetto Command usando il metodo *CreateObject()*. Il codice della Figura 3.5 che segue crea un oggetto Command e lo assegna la variabile cmd.

```
<SCRIPT language=vbscript runat=Server>
    Dim cmd 'variabile per un oggetto Command
    'Creazione di un oggetto Command
    set cmd=CreateObject("ADODB.Command")
 </SCRIPT>
```

Figura 3.5: Esempio di creazione di un'oggetto Command

Usare Command.CommandText

La modifica dei dati nel database si fa assegnando l'istruzione SQL alla proprietà CommandText come segue in Figura 3.6.

```
<SCRIPT language=vbscript runat=Server>
    Dim cmd 'variabile per un oggetto Command
    Dim SQL
    'Creazione di un oggetto Command
    set cmd=CreateObject ("ADODB.Command")
    SQL =INSERT INTO AUTORI (Id, Nome, Search)
    SQL = SQL & VALUES (27, 'G.Cabri')
   cmd.CommandText= SQL
</SCRIPT>
```

Figura 3.6: Utillizzo di CommandText

Usare la proprieta Command.ActiveConnection e il metodo Command.Execute

Prima di poter chiamare il metodo Execute dell'oggetto Command, si deve assegnare un oggetto connection come connessione attiva per l'oggetto command. Quest'ultimo ha una proprietà ActiveConnection proprio come l'oggetto Recorset. Una volta sistemato l'oggetto Command assegnando valori alle proprietà CommandText e ActiveConnection, si è pronti per chiamare il metodo Execute. Il listato in Figura 3.7 presenta l'uso e della proprietà Command.Active Connection e del metodo Execute.

```
<SCRIPT language=vbscript runat=Server>
    Dim cmd 'variabile per un oggetto Command
    Dim SQL'Creazione di un oggetto Command
    Dim conn 'variabile per l'oggetto connection
    'Creazione di un oggetto Command
    set cmd=CreateObject("ADODB.Command")
    set conn=CreateObject("ADODB.Connection")
     'si definisce qua la stringa di connessione
     'e l'appertura della cosi detta connessione
    SQL =INSERT INTO AUTORI (Id, Nome, Search)
    SQL = SQL&VALUES(27, 'G.Cabri')
    cmd.CommandText= SQL
   'Associa l'oggetto Connection all'oggetto Command
    cmd.ActiveConnection=conn
   'execuzione dell'istruzione SQL
    cmd.Execute
 </SCRIPT>
```

Figura 3.7: Utillizzo di ActiveConnection e Execute()

3.3 ASP.Net e i fonti di dati:ADO.Net

La piatta forma .NET fornisce un insieme di classe permettendo l'accesso ai dati. Quest'insieme di classe è ragruppato sul nome ADO.NET. ADO.NET Object Model è composta da due componenti fondamentali : DataSet e .NET Data Provider. DataSet è scollegato alla fonte di dati e non si occupa di dove vengono i dati gestiti. .NET Data Provider invece si occupa della connessione e dell'esecuzione delle dichiarazioni SQL. La seguente Figura 3.8 presenta l'architettura di ADO.NET.

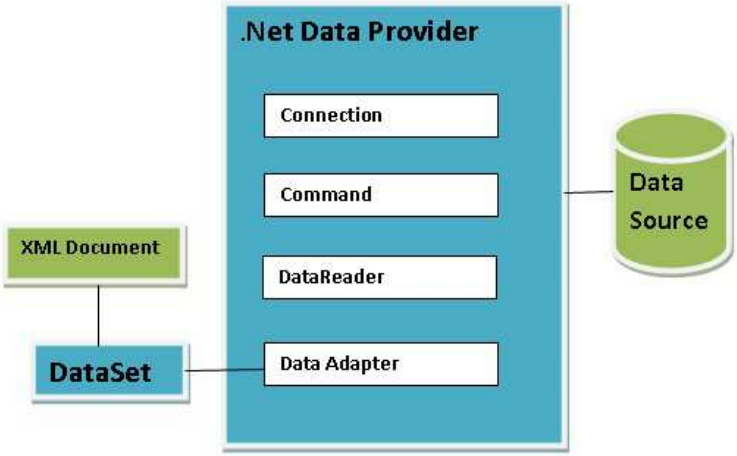

Figura 3.8: Architettura di ADO.NET

3.3.1 DOT NET Data Providers

DOT NET Data Providers forniscono meccanismi efficaci e veloci per accedere alle fonti di dati, eseguire dichiarazioni e recuperare i risultati. Sono in tre Data provider Offerti dal DOT NET Framework :

- SQL Server Data Provider
- OLE DB Data Provider
- ODBC Data Provider

The SQL Server Data Provider

SQL SERVER Data Provider è stato ideato da Microsoft per il DBMS SQL Server versione 7.0 e piu. Utilizza un protocollo proprietario per comunicare direttamente con SQL Server senza strati aggiuntivi come OLE DB oppure ODBC. L'uso di questo protocollo proprietario velocizza l'accesso e il trasferimento dei dati nelle applicazioni dove si utilizza il DBMS SQL Server(a partire della versione 7.0). Per le applicazioni che girano sul DBMS SQL Server (version anteriori fino a 6.5) si consiglia l'uso di OLE DB Data Provider. Le classi per il SQL Server Data Provider si trovano nel namespace *System.Data.SqlClient*.

The OLE DB Data Provider

OLE DB Data Provider è usato per fonti di dati chi sopportano le inetrfaccie OLE DB. Questo provider utilizza Native OLE DB attraverso l'interoperabilità con l'interfaccia COM per accedere e eseguire dichiarazioni SQL. Per utilizzare OLE DB Data Provider, bisognerebbe avere compatibili OLE DB Provider. Sono disponibili i seguenti OLE DB Provider compatibili con ADO.Net.

1. SQLOLEDB - Microsoft OLE DB Provider for SQL Server

2. MSDAORA - Microsoft OLE DB Provider for Oracle

3. Microsoft.Jet.OLEDB.4.0 - OLE DB Provider for Microsoft Jet

OLED DB Data Provider non prende in considerazione le interfacce OLE DB 2.5 ad esempio quelle necessarie per Microsoft OLE DB Provider per Exchange e Microsoft OLE DB Provider per Internet Publishing. OLE DB Data Provider è indicato per applicazioni che usano SQL Server 6.5 or earlier, Oracle oppure Microsoft Access. Le classi OLE DB Data Provider si trovano nel namespace System.Data.OleDB. ADO.NET espone un modello comune per . NET Data Provider che permette di scrivere il codice una volta e utilizzarlo insieme agli altri Data Provider. Questo è ottenuto attraverso l'uso di ADO.NET Data Provider Interface. Ci sono quatro interfacce per questo motivo :

- L'interfaccia IDbConnection per SqlConnection e OleDbConnection

- L'interfaccia IDbCommand per SqlCommand e OleDbCommand

- L'interfaccia IDataReader per SqlDataReader e OleDbDataReader

- L'interfaccia IDbDataAdapter per SqlDataAdapter e OleDbDataAdapter

3.3.2 Collegamento e manipolazione di una fonte di dati

Ogni ADO.NET Data Provider contiene un insieme di classe ragruppato nel NAMESPACE System.Data permettendo di gestire in modo sicuro la connessione, l'esecuzione delle dichiarazioni, e la restituzione dei risultati.

Collegarsi ad una fonte di dati: la classe Connection

La classe Connection eredita, dall'interfaccia IDbConnection. La classe Connection viene utilizzata per aprire una connessione alla fonte di dati in cui i comandi saranno eseguiti. E utilizzato per collegare l'applicazione alla sorgente di dati.

La classe SqlConnection La classe SqlConnection è utilizzata per collegarsi alle fonti di dati Microsoft SQL Server 7.0 e piu. Il listing della figura 3.9 che segue crea una nouova connessione a partire dalla catena csISGROUP che è la connection string per collegarsi alla fonte di dati SQL Server 2008 isgrouppro DB.mdf che si trova nella cartella DataDirectory.

```
Dim csISGROUP AS String
csISGROUP="Data
source=.\SQLEXPRESS;AttachDbFilename=|DataDirectory|"
+ "\isgrouppro DB.mdf;Integrated Security=True; User Instance=True"
SqlConnection con = new SqlConnection (csISGROUP);
```

Figura 3.9: Utilizzo della classe SqlConnection

La classe OleDbConnection La classe OleDbConnection è utilizzata per collegarsi alle fonti di dati tramite OleDb Data Provider.É possibile Memorizzare le stringhe di connessione nel file di configurazione web.config che ci permette di mantenerli in un solo luogo. Questa tecnica rende facile la modifica e il cambiamento di stringhe di connessione. Inoltre, poiché il file di configurazione è un XMLformattedfile di testo, la stringa di connessione può essere cambiata senza aver bisogno della ricompilazione dell'applicazione.

Esecuzione dei dichiarazioni SQL

Le classi *OleDbCommand* and *SqlCommand* sono usati per eseguire dichiarazioni su una fonte di dati. Una volta una connessione stabilita, un comando può essere utilizzato per eseguire un'istruzione T-SQL o una stored procedure, basato sul valore assegnato alla proprietà *CommandType*. I costruttori delle classi *SqlCommand* e *OleDbCommand* prendono un'istruzione SQL o una stored procedura ed un oggetto di connessione come argomenti.

Il listato della Figura 3.10 presenta la creazione di un oggetto SqlCommand a partire di un file di configurazione dell 'applicazione.

```
SqlConnection con = new SqlConnection(

ConfigurationSettings.AppSettings["constring"]
);
SqlCommand cmd = new
SqlCommand("SELECT * FROM Autori", con);
```

Figura 3.10: Utilizzo della classe SqlCommand

3.3.3 ACCESSO ai dati : la classe DataReader

ADO.Net fornisce la classe DataReader equivalente al Recordset di ADO in modo connesso e permettendo l'accesso veloce in modo read only e foward only ai dati. I classi DataReader devono implementare le interfacce IDataReader e IDataRecord. L'interfacia IDataReader fornisce metodi e proprietà permettendo di attraversare il set di risultati e di interrogare il DataReader. Mentre l'interfaccia IDataRecord fornisce metodi e proprietà per l'accesso ai dati nel risultato DataReader. I DataReader piu usati sono SqlDataReader e OleDbDataReader che si trovano nel namespace *System.Data.SqlClient* e *System.Data.OleDb* rispettivamente.

Creazione e utilizzo di un DataReader

Per accedere ad un DataReader, dobbiamo prima di tutto avere un oggetto Command. Il metodo ExecuteReader esegue il SQL o una stored procedura che l'oggetto Command rappresenta e restituisce un oggetto DataReader. Il listato della Figura 3.11 presenta un esempio di creazione e d'utilizzo di un DataReader.

3.3.4 La Classe DataAdapter

ADO.NET fornisce un altro modo per recuperare i dati da un database attraverso l'oggetto DataAdapter. L'oggetto DataAdapter è strettamente legato con la classe DataSet per creare in memoria rappresentazione dei dati. La principale differenza tra un DataReader e un DataAdapter si trova nel modo in cui gestire la connessione con il database. L'oggetto DataReader recupera le informazioni da un database aprendo una connessione. l'esecuzione del comando SQL, passando per il record recuperati e, infine,chiudere la connessione quando non sono necessari altri interventi. L'oggetto DataAdapter utilizza la connessione alla fonte di dati per il tempo che è assolutamente necessario riempire l'oggetto DataSet, rilasciando tutte le varie risorse dopo aver terminato l'operazione. La classe DataAdapter implementa il IDataAdapter

```
csISGROUP="Data
source=.\SQLEXPRESS;AttachDbFilename=|DataDirectory|"
+ "\isgroupproDB.mdf;Integrated Security=True;User Instance=True"

SqlDbConnection conn = new SqlDbConnection(csISGROUP);
SqlDbCommand command = new SqlDbCommand ( "select * from
AUTORI", conn );
conn.Open ();
SqlDbDataReader reader = command.ExecuteReader ();
while (reader.Read ())
{
// code in here
}
reader.Close ();
conn.Close ();
```

Figura 3.11: Creazione e utilizzo di un DataReader

interfaccia, che definisce le funzioni che un DataAdapter deve sopportare. La classe DataAdapter fornisce i blocchi di base per implementare un meccanismo di accesso ai dati disconnessi per riempire l'oggetto DataSet e per aggiornare la sorgente di dati. Il listato seguente presenta l'utilizzo di un oggetto DataAdapter per creare e per riempire un oggetto DataSet.

```
csISGROUP="Data source=.\SQLEXPRESS;AttachDbFilename=|DataDirectory|"
+ "\isgroupproDB.mdf;Integrated Security=True;User Instance=True"

SqlDataAdapter da = new SqlDataAdapter("SELECT id, nome, search FROM
Autori", csISGROUP);
DataSet ds = new DataSet("Authors");
da.Fill(ds, "author");
```

Figura 3.12: Creazione e utilizzo di un DataSet

3.3.5 L'oggetto DataSet

Il DataSet è il fulcro di uno scollegato, l'applicazione data-driven, ma è una rappresentazione in memoriadi una serie completa di dati, incluse le tabelle, relazioni e vincoli. Il DataSet non mantiene una connessione ad una fonte di dati, permettendo la gestione veramente scollegata dei dati. I dati in un DataSet possono essere accessi, manipolati, aggiornati o cancellati, e quindi compatibile con l'origine dei dati originali. Il DataSet utilizza eXtensible Markup Language (XML) per la trasmissione e la persistenza dei dati.

Conlusione: In questo capitolo abbiamo presento l'interfacciamento dei basi di dati con la tecnologia ASP e ASP.NET. ADO.NET ottimizza le applicazioni Web rispetto a ADO perche integra il supporto nativo di SQL Server ed è basato sul Framework NET che utilizza un linguagio compilato. Inoltre ADO.NET consente anche l'utilizzo dei DataSet (offline) che possono memorizzare più tabelle in memoria mentre ADO era permanance connesso al database ed è stato solo un recordset.

Parte II

Reingegnerizzazione del sito web di ISGroup mediante ASP.NET e SQL Server 2008

Capitolo 4

Progetto

La progettazione del software è passata attraverso una seria di fasi. Prima di tutto, presenteremo il sito Web di ISGroup con le diverse pagine dinamiche

4.1 Il sito web di ISGroup

L'ISGroup, letteralmente Information Systems Group, è un gruppo di ricercatori che ha sede presso il Dipartimento di Ingegneria dell'Informazione (DII) di Modena. Si occupa di design e sviluppo di nuovi sistemi, algoritmi e strutture dati per l'accesso e l'amministrazione dell'Informazione. Il sito del gruppo, visitabile al seguente link http://www.isgroup.unimo.it/, è composto da tre sezioni principali: Projects, Publications e Thesis. Nella prima sezione vengono descritti in dettaglio i Topic Of Interest studiati e i progetti a cui il gruppo partecipa o ha partecipato; nella sezione Publications vengono elencate, raggruppate per anno o topic, le pubblicazioni scientifiche realizzate dai vari membri del gruppo ed infine nell'ultima sono elencate le tesi proposte e una serie di tesi pubblicate. Il sito gira su un server Internet Information Services e mette a disposizione una serie di funzionalità realizzate mediante script ASP lato server e script JavaScript lato client. Si appoggia inoltre a un database pub.mdb, creato mediante MS Access, che permette di gestire autori, pubblicazioni, progetti e prototipi. Attraverso gli script in ASP è possibile reperire le informazioni sulle pubblicazioni, sulle tesi e sui progetti interrogando opportunatamente il database. Inoltre la sezione Publications consente anche di effettuare ricerche per autore e topic e di ordinare i risultati per topic o anno di pubblicazione. Mediante gli script in JavaScript, invece, vengono creati una serie di effetti grafici, filtri anti-spam e controlli per i campi dei form.

4.1.1 Analisi script ASP

Per poter realizzare alcune parti del progetto, sono state studiate una serie di funzionalità già presenti nel sito originale. In particolar modo sono stati analizzati gli script relativi alle sezioni delle pubblicazioni (publ.asp), delle tesi (thesis.asp) e dei progetti (projects.asp). Per poter effettuare migliorie alla base di dati e a vari script, sono state analizzate in dettaglio le funzionalità già presenti nel sito.

Publ

Mediante questo script viene realizzata la pagina delle pubblicazioni scientifiche (http://www.isgroup.unimo.it/publ.asp) che comprende un motore di ricerca attraverso il quale è possibile ricercare le pubblicazioni in base all'autore e al topic ed, inoltre, ordinare i risultati in base all'anno o al topic. Attraverso opportune query, vengono selezionate le opzioni che andranno poi a comporre i menù a tendina degli autori e del topic generati mediante appositi tag HTML. Da notare che gli autori visualizzati sono solo quelli che presentano nell'attributo search 'Y'. In un primo momento per ogni pubblicazione vengono visualizzati solo i seguenti dati: Titolo, Autori e Abbreviazione. Mediante una funzione JavaScript, cliccando sopra il riquadro della pubblicazione saranno visualizzate ulteriori informazioni tra le quali l'Abstract e il BibTeX; questi ultimi potranno essere letti solo selezionandoli.

Thesis

Tale script realizza la sezione Thesis (http://www.isgroup.unimo.it/thesis.asp). Nella parte Published Thesis, come nella sezione delle pubblicazioni, vengono visualizzati, inizialmente, gli autori e il relativo titolo della tesi. Solo cliccando sull'autore sarà possibile visualizzare altre informazioni. I dati visualizzati sono ottenuti recuperando dalla tabella 'Tesi' gli opportuni campi.

Projects

Questo script (http://www.isgroup.unimo.it/projects.asp), è stato realizzato come i precedenti. Vengono visualizzati i dati sia dei Topic Of Interest che dei progetti realizzati dall'ISGroup, recuperati rispettivamente dalle realitve tabelle mediante opportune query.

4.1.2 Schema E-R del sito

Il schema E-R del sito è stato utilizzato senza modifica perchè non è stata aggiunta alcuna funzionalità durante lo svolgimento di questa tesi. Per questo motivo, il schema presentato nella figura 4.1 è stato preso nella tesi in [1].

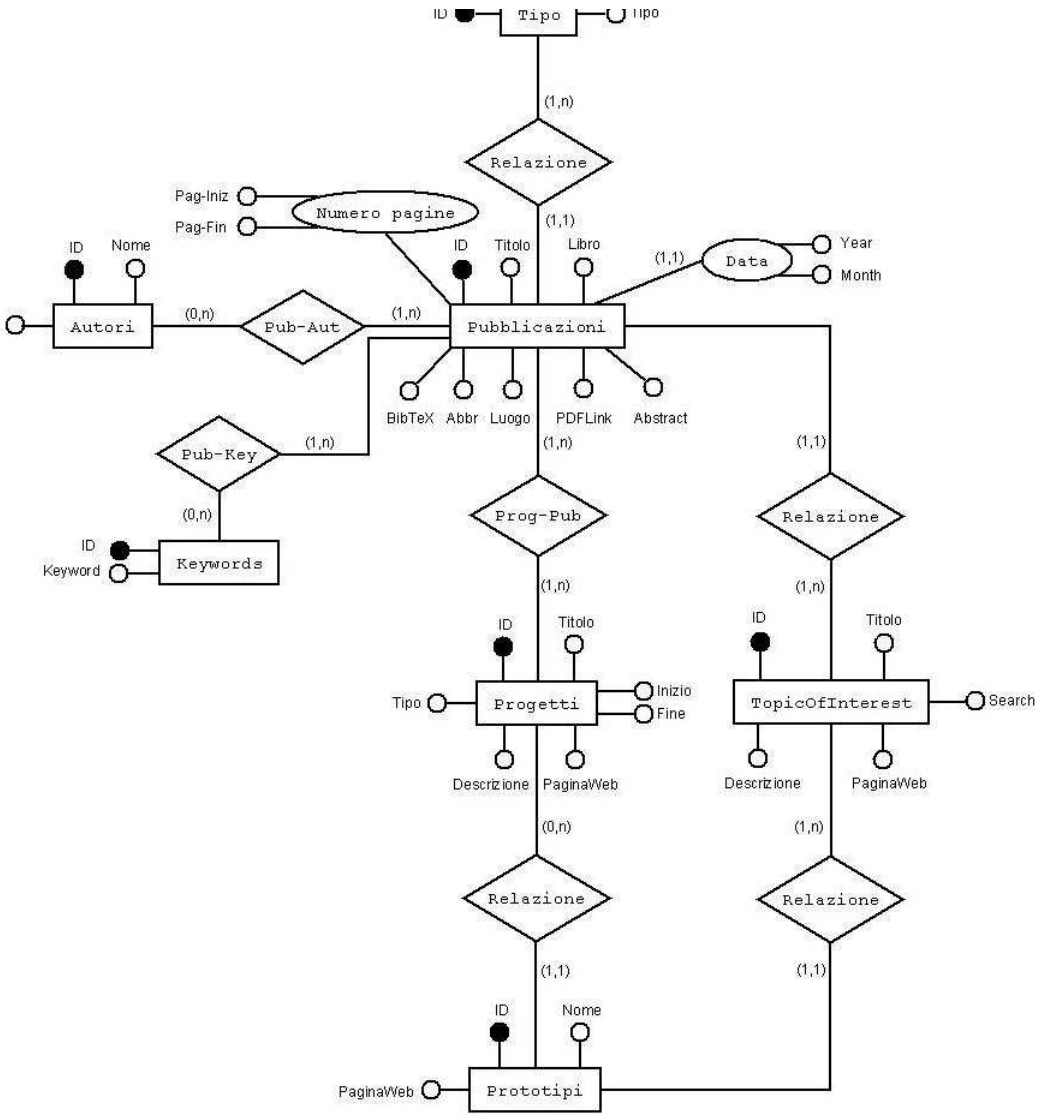

Figura 4.1: Schema E-R del sito

4.1.3 Schema logico del sito

La traduzione di entità e associazioni da schema E-R a progetto logico si ottiene:

```
AUTORI ( ID, Nome, Search )

PUB-AUT ( ID, Pub, Autore )
  AK: Pub, Autore
  FK: Pub REFERENCES Pubblicazioni
  FK: Autore REFERENCES Autori

PUBBLICAZIONI ( ID, Titolo, Libro, Abbr, Pag-Iniz, Pag-Fin, Luogo,
  Year,Month, Tipo, Topic, PDFLink, Abstract, BibTeX )
  FK: Tipo REFERENCES Tipo
  FK: Topic REFERENCES TopicOfInterest

KEYWORDS ( ID, Keyword )
PUB-KEY ( ID, Pub, Keyword )
  AK: Pub, Keyword
  FK: Pub REFERENCES Pubblicazioni
  FK: Keyword REFERENCES Keywords

TIPO ( ID, Tipo )
TOPICOFINTEREST ( ID, Titolo, Descrizione, PaginaWeb, Search )
PROG-PUB ( ID, Prog, Pub )
  AK: Prog, Pub
  FK: Prog REFERENCES Progetti
  FK: Pub REFERENCES Pubblicazioni

PROGETTI ( ID, Titolo, Descrizione, PaginaWeb, Tipo, Inizio, Fine )

PROTOTIPI ( ID, Nome, PaginaWeb, Topic. Progetto )
  FK: Topic REFERENCES TopicOfInterest
  FK: Progetto REFERENCES Progetti
```

4.1.4 Relazioni tra le tabelle

Dal seguente schema, realizzato mediante Access 2007, è possibile vedere in dettaglio le tabelle del database e le loro colonne, ma anche le relazioni che

le interessano.

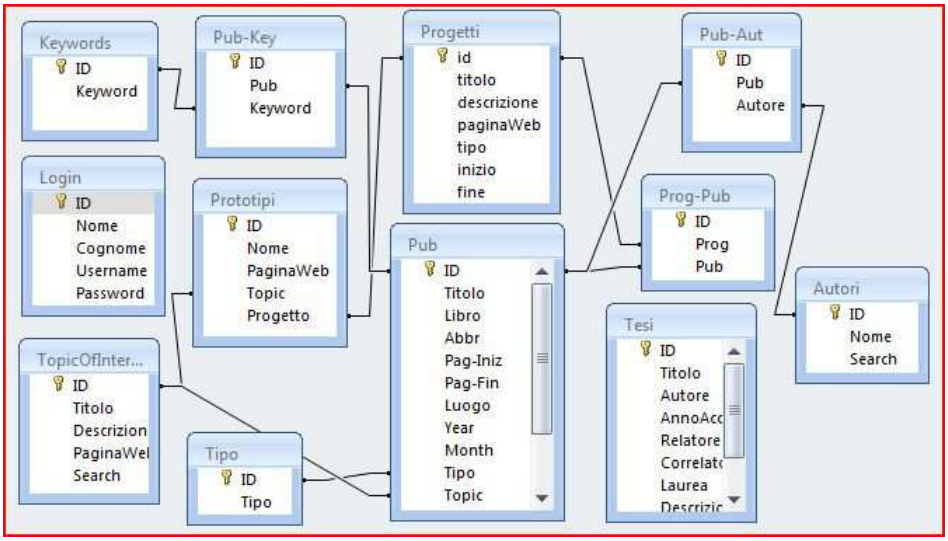

Figura 4.2: Schema delle relazioni tra le tabelle

4.2 Progetto del sito

Il sito web è stato progettato appogiandosi sul concetto d'ereditarietà. Infatti tutte le pagine ereditano della pagina master che definisce lo schemo grafico comune. La figura 4.3 presente lo schema del sito. La pagina master definisce tutte le funtionalità grafiche del sito lasciando il contenuto specifico ad ogni pagina.

Precisamente, questa pagina definisce tramite i tag e i fogli *CSS*:

- l'area di navigazione principale
- l'area di navigazione secondaria
- la pagina d'intestazione
- Il logo
- il banner pubblicitari
- il pie di pagina
- l'area dei contenuti contestuali basato sulla pagina corrente

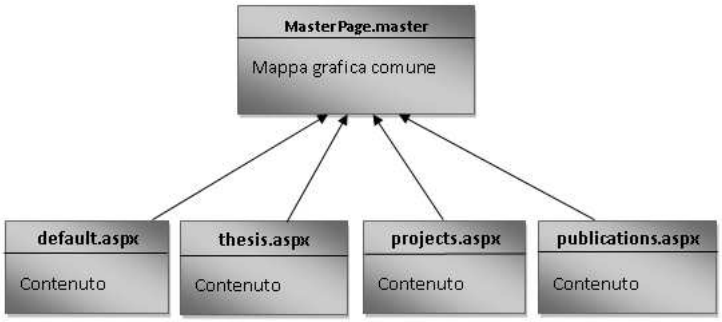

Figura 4.3: Schema del sito

Le pagine Default.aspx, Thesis.aspx, Projects.aspx e Publications.aspx ereditano degli elementi elencati precedentemente ma ognuno definisce il suo contenuto collegandosi(pagine Publications.aspx, Thesis.aspx, Projects.aspx) sia alle tabelle o sia ad una vista della fonte di dati. Nel capitolo implementazione presenteremo dettagliatamente il contenuto di ogni pagina.

Capitolo 5

Implementazione

In questo capitolo viene analizzato il progetto dal punto di vista implementativo. Viene inizialmente descritto in dettaglio l'ambiente di sviluppo e poi l'area grafica ed infine gli script ASPX lato server che interessano la maggior parte del software.

5.1 Ambiente di sviluppo

Per poter realizzare questo sito, abbiamo utilizzato Visual Web developper 2008 Express Edition SP1 un ambiente ideato da Microsoft per lo sviluppo delle applicazioni web dinamiche che si appogiano sul Framework .NET. Visual Web Developper è un software libero che permette a chiunque, dal principiante allo sviluppatore professionista,di creare siti Web ASP.NET. Intergra nativamente una base di dati SQL Server 2008 Express SP1. Per poter migrare i dati da Access 2000 a SQL Server 2008 abbiamo utilizzato SQL Server 2008 Migration Assistant for Access(SSMA).

5.2 Area grafica del sito

L'area grafica comprende il foglio di stilo Default.css, il file di gestione dell'aspetto Default.skin e il file PublicationsBanniere.xml per la gestione delle immagini per la pubblicità. La pagina MasterPage.master la cui intestazione è definita nella figura 5.1 è suddivisa in tre principali blocchi:

1. il blocco per l'intestazione incapsulato nei tag et

2. il blocco per la pagina incapsulato nei tag et

3. il blocco per il pie di pagina incapsulato nei tag et

```
<%@ Master Language="VB" CodeFile="MasterPage.master.vb" Inherits="MasterPage"
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.o
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title></title>
    <link  href=  "Default.css" rel="stylesheet" type="text/css"/>
</head>
```

Figura 5.1: Codice dell'intestazione della pagina master

L'intestazione presenta il linguaggio utilizzato, il nome del file di codice behind associato, il foglio di stilo usato.

5.2.1 Il blocco d'intestazione

Questo blocco contiene tre sotto blocchi ed è rapresentato dalla figura 5.2 . Il blocco per il menù che fa riferimento alla home page del sito ISGroup, a quella del dipartimento di ingegneria e alla home page della facoltà. Il blocco contenendo le foto dei funding members del gruppo. Questo blocco utilizza il controllo <asp:AdRotator >. Il blocco informa l'utenza sul camino seguito dalla radice fino alla pagina corrùente.Questo blocco utilizza il controllo <asp:SiteMapPath>

```
<div class="header">
      <div class="menua"><!-- Zone : menu en haut de la page -->
        <a href=http://www.isgroup.unimo.it>ISGroup Home</a>|<a href=http://
      </div>
      <div class="rotator"> <!--anneau publicitaire  pubblicita-->
       <asp:AdRotator   ID="AdRotator1" runat="server" />
      </div>
      <div class="nav"> <!-- Zone : bara di navigazione affiggendo il camino
          <asp:SiteMapPath ID="SiteMapPath1" runat="server">
              <PathSeparatorTemplate>
                  <asp:Image ID="Image1" runat="server"
                      ImageUrl="~/App_Themes/Default/Images/bullet-2.gif" />
              </PathSeparatorTemplate>
          </asp:SiteMapPath>
       </div>
</div>
```

Figura 5.2: Codice dell'intestazione

5.2.2 Il blocco pagina

Questo blocco contiene i blocchi per il menù di destra, e per il contenuto delle pagine figli. La figura 5.3 presenta il codice del blocco pagina.

```
<div class="page"><!-- Zone : Zone de navigation à gauche de la page  -->
    <div id="sidebar"> <!--Zone : Espace membre  -->
        <h1>Planing of the site</h1>
        <div id="liensEspaceMembre"></div><hr />
           <h1>Activities of our group</h1><!--Zone : Services des annonce
                <!--h2>Consult publications, tesis and projects</h2-->
                 <asp:SiteMapDataSource ID="SiteMapDataSource1"  ShowStart
                 <asp:TreeView ID="TreeView1" runat="server"
                   DataSourceID="SiteMapDataSource1">
                 </asp:TreeView>
    </div>
    <div id="content">><!-- Zona : Zona avendo il contenuto dei pagini figli
      <asp:ContentPlaceHolder ID="Main" runat="server">
      </asp:ContentPlaceHolder>
    </div>
</div>
```

Figura 5.3: Codice del blocco pagina

5.2.3 Il blocco pie di pagina

Questo blocco gestisce il contenuto di ogni pie di pagina del sito.La figura 5.4 presenta il codice del blocco pagina. Le figure 5.5 e 5.6 rappresentano il

```
<!-- Zone : Pied di pagina-->
<div class="footer">
    <asp:Menu ID="Menu1" runat="server" DataSourceID="SiteMapDataSource1">
    </asp:Menu>
    <asp:Literal ID="ltlPiedDePage" runat="server"
     Text="ISGroup Information Systems Group |Paolo Tiberio|Federica Mandreol
</div>
```

Figura 5.4: Codice del blocco pie di pagina

codice e il codice behind della pagina Default.aspx che eredita dalla pagina MasterPage.master. Questa pagina nell'attributo MasterPageFile specifica il file della pagina padre. Nel suo contenuto specifica l'ID del controllo ContentPlaceHolder della pagina padre. La figura 5.7 rappresenta lo schema della pagina Default.aspx.

```
<%@ Page MasterPageFile="~/MasterPage.master" Language="VB" Title="ISGroup-UNIM

   <asp:Content ID="content1" ContentPlaceHolderID="Main" runat="server">
       <asp:Label ID="Label1" runat="server" Text="Label"></asp:Label>

       </asp:Content>
```

Figura 5.5: Codice della pagina Default.aspx

```
Partial Class _Default
    Inherits System.Web.UI.Page

    Protected Sub Page_Load(ByVal sender As Object, ByVal e As System.EventArgs)
 Handles Me.Load
        Label1.Text = "Welcome to the ISGROUP's Web site"
        Master.PiedDePage = "Thanks to send your suggestions to <a href=martogl
    End Sub
End Class
```

Figura 5.6: Codice behind della pagina Default.aspx

Figura 5.7: Pagina Default.aspx nel browser

5.3 Script aspx delle pagine

5.3.1 Script della pagina Project.aspx

Come visto nella sessione precedente, questa pagina eredita della pagina MasterPage.master. La pagina è suddivisa in quatro:

1. Reasearch interests and projects che presenta i domini di ricerca e i progetti svolti dal gruppo

2. Past reasearch interests che presenta i domini di ricerca passati del gruppo

3. National and European funded project che presenta i progetti europeani di interesse dal gruppo

4. Past projects che presenta i progetti passati del gruppo

Reasearch

Per costruire questa parte abbiamo sfruttato l'ereditarietà per quanto riguarda la parte grafica della pagina e definito un collegamento alla sorgente di dati tramite la stringa di connessione *csISGROUP*. Il collegamento con la sorgente di dati chiamato *dsTopicOfInterest* è realizzato attraverso il parametro *SelectCommand* che seleziona tutti i campi della tabella *TopicOfInterest*. Questa sorgente di dati è utilizzata in una *DropDownList* per affiggere tutti i titoli dei domini di ricerca. Una altra sorgente di dati *dsTopicOfInterest1* è definita per fare il collegamento tra le informazioni scelte nella tabella *TopicOfInterest* e quelle del controllo *DropDownList*. La presentazione dei risultati è fatta attraverso il controllo *DataList* mediante la proprietà ItemTemplate. Le fugure 5.8, figura 5.9 e figura 5.10 presentano successivamente il codice delle due sorgenti di dati poi il collegamento realizzato ed infine la pagina ottenuta. La pagina ottenuta è presentata nella figura 5.13.

Past Research Interest and National and European funded Projects

L'estrazione e la presentazione dei dati relativi a questa parte è stato realizzato secondo il processo descritto precedentemente nella sezione 51. Infatti una connessione alla fonte di dati ci permette di trarre i progetti non scaduti e la presentazione si fà mediante una DataList. Di seguito presento prima il codice 5.15 relativo a questa parte e poi la pagina allegata.

```
<%@ Page Title="ISGroup-UNIMORE Projects" Language="VB" MasterPageFile="~/Maste
<asp:Content ID="Content1" ContentPlaceHolderID="Main" Runat="Server">
    <h1>            
    <hr>
    <center>
    <h2>At present our research interests are about the following topics:</h2>
    <h2>(chose a topic to access to more information about it) </h2>
    </center>
    <asp:SqlDataSource ID="dsTopicOfInterest" runat="server"
        ConnectionString="<%$ ConnectionStrings:csISGROUP %>"
        SelectCommand="SELECT * FROM [TopicOfInterest]"
        ProviderName="System.Data.SqlClient">
    </asp:SqlDataSource>
    <asp:DropDownList ID="DropDownList1" runat="server" AutoPostBack="True"
        DataSourceID="dsTopicOfInterest" DataTextField="Titolo" DataValueField="
    </asp:DropDownList>
```

Figura 5.8: Collegamento alla fonte di dati

```
    <asp:SqlDataSource ID="dsTopicOfInterest1" runat="server"
        ConnectionString="<%$ ConnectionStrings:csISGROUP %>"
        SelectCommand="SELECT * FROM [TopicOfInterest] WHERE ([ID] = @ID)">
        <SelectParameters>
            <asp:ControlParameter ControlID="DropDownList1" Name="ID"
                PropertyName="SelectedValue" Type="Int32" />
        </SelectParameters>
    </asp:SqlDataSource>
    <asp:DataList ID="DataList1" runat="server" DataKeyField="ID"
        DataSourceID="dsTopicOfInterest1" style="margin-left: 0px">
        <ItemTemplate>
  <table class=BorderedTable cellSpacing=1 cellPadding=1 width="99%" align=cent
            <tbody>
            <tr bgColor=#e1e1e1><td class=pubblfontbig  colSpan=2><b>
            <%# Eval("Titolo") %> </td> </tr>
            <tr bgColor=#f5f5f5>     <!-- Description -->
                <td class=pubblabstr width="100%" colspan=2>
```

Figura 5.9: Data Binding e presentazione nella DataList

Figura 5.10: Selezione di un topic

Figura 5.11: Detagli sul topic selezionato

```
    <h2> Past research interests   </h2>
 <asp:SqlDataSource ID="dsPastTopics" runat="server"
     ConnectionString="<%$ ConnectionStrings:csISGROUP %>"
     SelectCommand="SELECT [ID], [Titolo] FROM [TopicOfInterest] WHERE ([Search
     <SelectParameters>
         <asp:Parameter DefaultValue="N" Name="Search" Type="String" />
     </SelectParameters>
 </asp:SqlDataSource>
  <asp:DropDownList ID="DropDownList2" runat="server" AutoPostBack="True"
     DataSourceID="dsPastTopics" DataTextField="Titolo" DataValueField="ID">
</asp:DropDownList>
<asp:SqlDataSource ID="dsPastTopics1" runat="server"
     ConnectionString="<%$ ConnectionStrings:csISGROUP %>"
     SelectCommand="SELECT * FROM [TopicOfInterest] WHERE ([ID] = @ID)">
     <SelectParameters>
         <asp:ControlParameter ControlID="DropDownList2" Name="ID" |
             PropertyName="SelectedValue" Type="Int32" />
```

Figura 5.12: Creazione della Datasource e collegamento alla DropDownList

```
  </asp:SqlDataSource>
   <asp:DataList ID="DataList2" runat="server" DataKeyField="ID"
      DataSourceID="dsPastTopics1" style="margin-left: 0px">
      <ItemTemplate>
<table class=BorderedTable cellSpacing=1 cellPadding=1 width="99%" align=center
            <tbody>
            <tr bgColor=#e1e1e1><td class=pubblfontbig  colSpan=2><b>
            <%# Eval("Titolo") %>
                                    </td>
            </tr>
            <tr bgColor=#f5f5f5>     <!-- Description -->
               <td class=pubblabstr width="100%" colspan=2>
                   <br><div align=justify> <%# Eval("Descrizione") %> </div><b
               </td>
            </tr>
            <tr bgColor=#f5f5f5>
               <td class=pubblfont  width="50%">
```

Figura 5.13: Codice di presentazione dei dati estratti

```
<h2>We are participating in the following<b> national and european funded projec
<h2>(chose a topic and access to more information)</h2>
    <asp:SqlDataSource ID="dsProjects" runat="server"
        ConnectionString="<%$ ConnectionStrings:csISGROUP %>"

        SelectCommand="SELECT [id], [titolo] FROM [Progetti] WHERE (([inizio] &

        <SelectParameters>
            <asp:Parameter DefaultValue="2010" Name="inizio" Type="Int32" />
            <asp:Parameter DefaultValue="2010" Name="fine" Type="Int32" />
        </SelectParameters>
    </asp:SqlDataSource>
    <asp:DropDownList ID="DropDownList3" runat="server" AutoPostBack="True"
        DataSourceID="dsProjects" DataTextField="titolo" DataValueField="id">
    </asp:DropDownList>
    <asp:SqlDataSource ID="dsProjects1" runat="server"
        ConnectionString="<%$ ConnectionStrings:csISGROUP %>"
```

Figura 5.14: Codice della parte relativa ai progetti Europei

Past research interests

EXTRA: Example Based Machine Translation

EXTRA: Example Based Machine Translation

In collaboration with LOGOS Group, we study the field of EBMT (Example Based Machine Translation), in which approximate matching techniques are applied to sentences, also considering additional problems related to multi-linguism. Recently, we are also extending the syntactic similarity to the semantic field, by means of the study of techniques of disambiguation based on the use of WordNet.

Related Publications **Click here**

Related prototypes **Click here**

We are participating in the following **national and european funded projects:**
(chose a topic and access to more information)

Outdoor Video Protection: visione artificiale e tecnologie informatiche innovative p

Outdoor Video Protection: visione artificiale e tecnologie informatiche innovative per la sicurezza di ambienti di lavoro all'aperto
Progetto Regionale
2009-2010

Lo scopo principale del progetto è il monitoraggio di ampi spazi esterni, con particolare riferimento a luoghi di lavoro pericolosi quali i cantieri edili, al fine

Figura 5.15: Risultato ottenuto

5.3.2 Script della pagina Thesis.aspx

Questo script è suddiviso in tre parti.

Script della parte Proposed Thesis

Questa parte è stata realizzata attraverso tag html ed è raggiungibile tramite il link ipertessuale *Proposed Thesis*. La pagina della figura 5.16 presenta il risultato ottenuto.

Figura 5.16: Schema de Proposed Theses

Published Theses (Computer Engineering and computer science)

Queste parti sono state realizzate collegandosi alla tabella Tesi della base di dati e stampando il titolo, l'autore, il relatore, la descrizione del sogetto secondo il vincolo che la tesi puo' essere laurea in Ingegneria o Informatica. Nella DropDownList è possibile scegliere l'autore di una tesi e avere la descizione del topic, i nomi dei relatori e l'anno academico. É anche possibile stampare la tesi e la relativa presentazione. Il codice scritto per questa parte è presentato in figura 5.17 e 5.18. Un caso d'uso è presentato nelle figure 5.19 e 5.20.

La stessa metodologia è applicata per le tesi in computer science. Le figure 5.21, 5.22, 5.23, 5.24 presentano le fasi di completamento e d'uso per le tesi in computer science.

```
<asp:SqlDataSource ID="dsTesi" runat="server"
    ConnectionString="<%$ ConnectionStrings:csISGROUP %>"
    SelectCommand="SELECT * FROM [Tesi] WHERE ([Laurea] = @Laurea) ORDER BY [ID]
    <SelectParameters>
        <asp:Parameter DefaultValue="Ing" Name="Laurea" Type="String" />
    </SelectParameters>
</asp:SqlDataSource>
<asp:DropDownList ID="DropDownList1" runat="server" AutoPostBack="True"
    DataSourceID="dsTesi" DataTextField="Autore" DataValueField="ID">
</asp:DropDownList>
<asp:SqlDataSource ID="dsTesi1" runat="server"
    ConnectionString="<%$ ConnectionStrings:csISGROUP %>"
    SelectCommand="SELECT [Titolo], [Autore], [Relatore], [Descrizione], [PDFLi
    <SelectParameters>
        <asp:ControlParameter ControlID="DropDownList1" Name="ID"
            PropertyName="SelectedValue" Type="Int32" />
```

Figura 5.17: Codice collegamento alla base di dati

```
<asp:DataList ID="DataList1" runat="server" DataSourceID="dsTesi1">
    <ItemTemplate>

        <table class=BorderedTable cellSpacing=1 cellPadding=1 width="99%" alig
           <tbody>
           <tr bgColor=#e1e1e1><td class=pubblfontbig  colSpan=2><b><%# Eval("A
                                <br><%# Eval("Titolo") %>
                                <br>Anno <%# Eval("Annoacc") %></td>
           </tr>
           <tr bgColor=#f5f5f5>
              <td class=pubblabstr width="100%" colspan=2>
                  <br><div align=justify>
Eval("Descrizione") %></div><br>
              </td>
           </tr>
           <tr bgColor=#f5f5f5>
```

Figura 5.18: Codice presentazione dei risultati

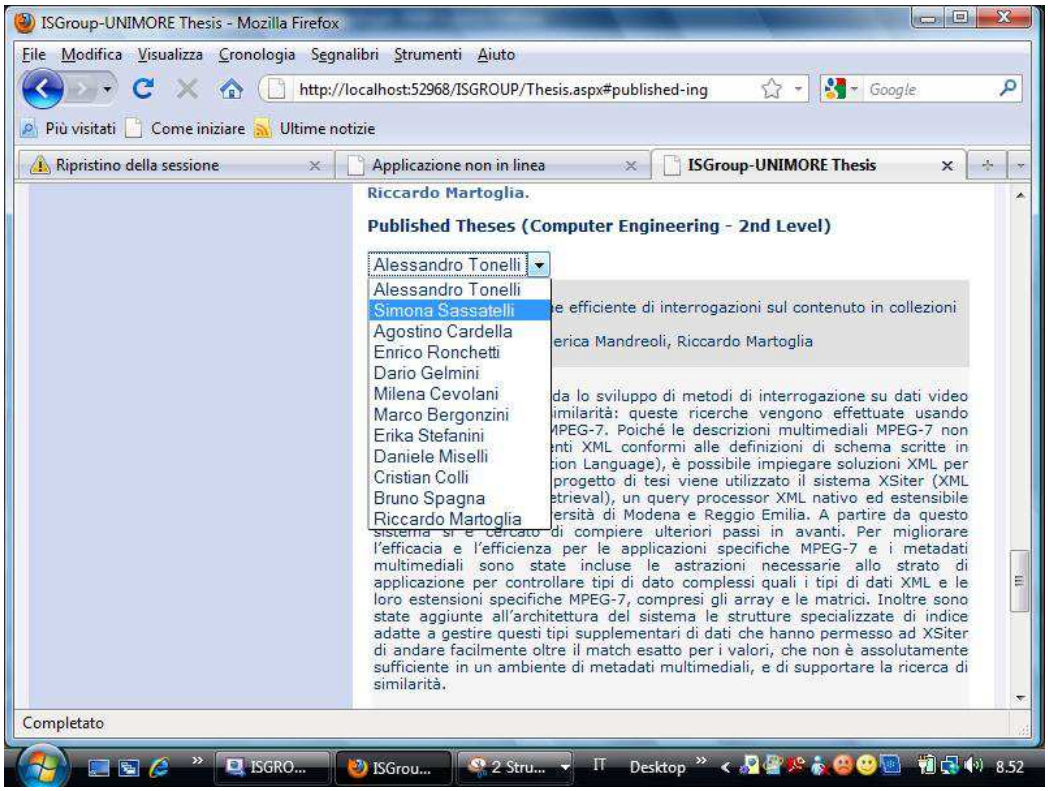

Figura 5.19: Selezione di un autore

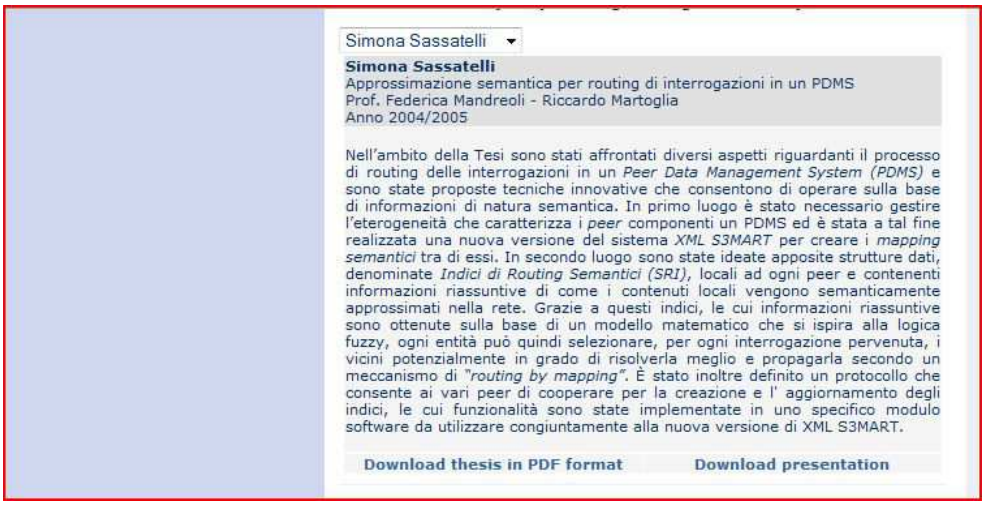

Figura 5.20: Informazioni sulla tesi dell'autore selezionato

```
<asp:SqlDataSource ID="dsTesInf" runat="server"
    ConnectionString="<%$ ConnectionStrings:csISGROUP %>"
    SelectCommand="SELECT [Autore], [ID] FROM [Tesi] WHERE ([Laurea] = @Laurea)
    <SelectParameters>
        <asp:Parameter DefaultValue="Inf" Name="Laurea" Type="String" />
    </SelectParameters>
</asp:SqlDataSource>
<asp:DropDownList ID="DropDownList2" runat="server" AutoPostBack="True"
    DataSourceID="dsTesInf" DataTextField="Autore" DataValueField="ID">
</asp:DropDownList>
<asp:SqlDataSource ID="dsTesInf1" runat="server"
    ConnectionString="<%$ ConnectionStrings:csISGROUP %>"
    SelectCommand="SELECT * FROM [Tesi] WHERE ([ID] = @ID)">
    <SelectParameters>
        <asp:ControlParameter ControlID="DropDownList2" Name="ID"
            PropertyName="SelectedValue" Type="Int32" />
```

Figura 5.21: Codice del collegamento alla base di dati

```
<asp:DataList ID="DataList2" runat="server" DataKeyField="ID"
    DataSourceID="dsTesInf1">
    <ItemTemplate>
       <table class=BorderedTable cellSpacing=1 cellPadding=1 width="99%" align=
           <tbody>
           <tr bgColor=#e1e1e1><td class=pubblfontbig  colSpan=2><b><%# Eval("Au
                                <br><%# Eval("Titolo") %>
                                <br>Anno <%# Eval("Annoacc") %></td>
           </tr>
           <tr bgColor=#f5f5f5>
              <td class=pubblabstr width="100%" colspan=2>
                 <br><div align=justify>
Eval("Descrizione") %></div><br>
              </td>
           </tr>
           <tr bgColor=#f5f5f5>
```

Figura 5.22: Codice presentazione dei risultati

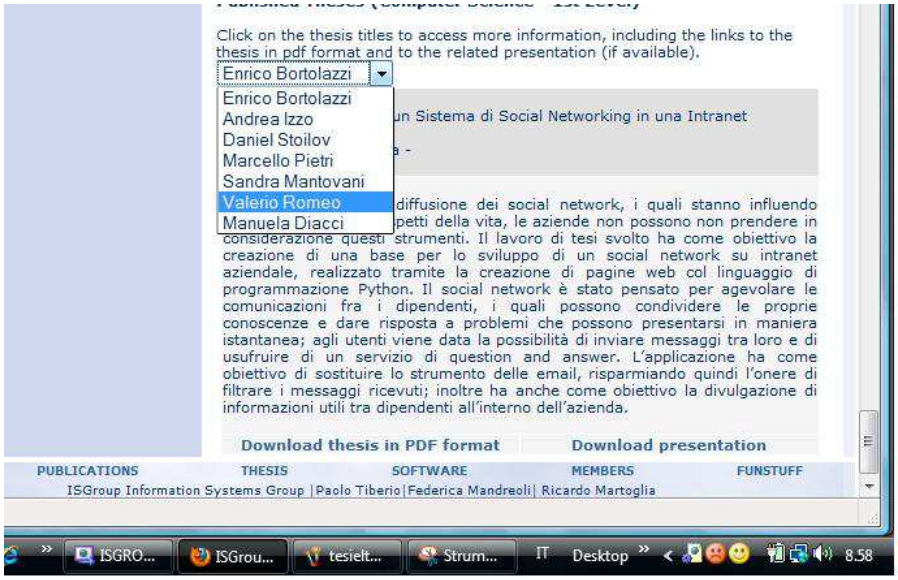

Figura 5.23: Selezione di un autore

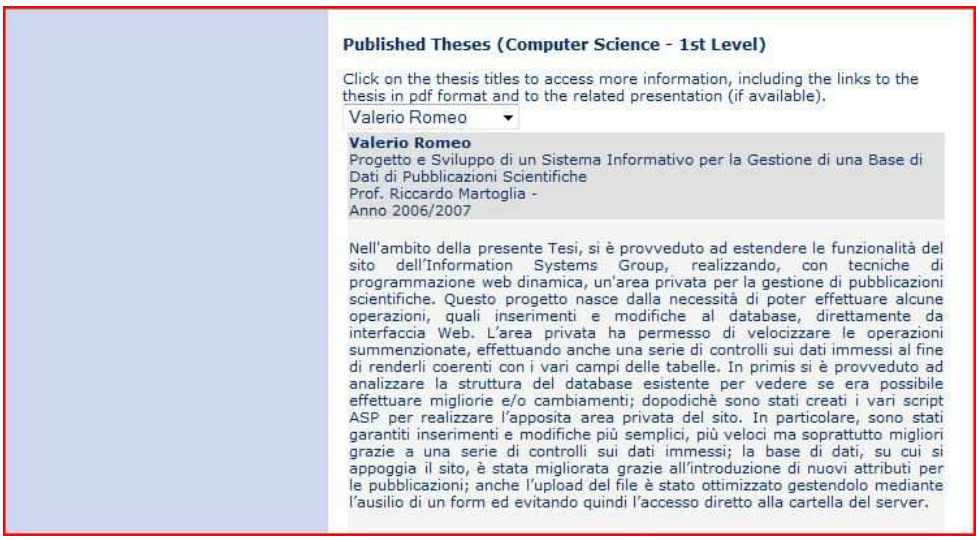

Figura 5.24: Informazioni sulla tesi dell'autore selezionato

5.3.3 Script della pagina Publications.aspx

Questo script è suddiviso in due parte principali. La prima parte dedicata ai criteri della ricerca ed la seconda dedicata ai risultati. La prima è costituitta da tre DropDownList *Authors*, *Topic* e *Group by* collegati ai fonti di dati *dsAutore* per selezionare gli autori, e *dsTopicOfInterest* per i domini di ricerche. La terza DropDownList ci permette di affiggere *Year* o *Topic* a seconda della scelta dell'utente. La figura 5.26 presenta lo script di costruzione della web form mentre la figura 5.32 presenta i risultati ottenuti.

```
<div>
 <asp:Label ID="authorLabel" runat="server" Text="Author:"></asp:Label><br>
 <asp:DropDownList ID="DropDownList1"  runat="server" DataSourceID="dsAutore" Va
     DataTextField="Nome" DataValueField="ID" OnSelectedIndexChanged="DropDownLi
     <asp:ListItem Value="0">All</asp:ListItem>
 </asp:DropDownList>
 <asp:RequiredFieldValidator ID="RequiredFieldValidator1" runat="server" Control
  <br>
 <asp:Label ID="topicLabel" runat="server" Text="Topic:"></asp:Label>
 <asp:DropDownList ID="DropDownList2" runat="server"   ValidationGroup="pouet"
     DataSourceID="dsTopicOfInterest" DataTextField="Titolo"   AppendDataBoundIt
     DataValueField="ID" OnSelectedIndexChanged="DropDownList2_SelectedIndexChan
     <asp:ListItem Value="0">All</asp:ListItem>
 </asp:DropDownList>
 <asp:RequiredFieldValidator runat="server" ControlToValidate="DropDownList2" In
```

Figura 5.25: Parte dello script di costruzione della Web form

Figura 5.26: Web form dovuto alla ricerca

La seconda parte è costituitta dal bottone search permettendo di avviare una ricerca ed una DataList per stampare i risultati. Una volta premuto il bottone *Search*, la procedura è eseguita ed i risultati mandati alla *DataList* per la stampa. La procdedura ci permette di eseguire dichiarazioni sql a seconda dei criteri scelti dall'utente. La figura 5.27 presente una parte del

codice della procedura *SearchPublications*. La figura 5.28 invece presenta il datasource collegata alla dataList e usando la procudura *SearchPublications*.

```
ALTER PROCEDURE SearchPublications
    @autore int,
    @topic varchar,
    @groupby varchar

    AS


     /* autore e topic */
IF @autore!=0 AND @topic!='0'
      IF @groupby='1' /* Order by Year*/
      BEGIN
      SELECT        Pub.ID, Pub.Year, Pub.Topic, [Pub-Aut].Autore, Pub.Titolo, Pub.L
                          Pub.FlashLink, Pub.[Pag-Iniz], Pub.[Pag-Fin]
    FROM            Pub INNER JOIN
                          [Pub-Aut] ON Pub.ID = [Pub-Aut].Pub
    WHERE        ([Pub-Aut].Autore = @autore) AND (Pub.Topic = @topic) ORDER BY Pub.
      END
      ELSE
```

Figura 5.27: Codice parziale della Stored Procedure

```
<asp:SqlDataSource ID="dsResearch" runat="server"
    ConnectionString="<%$ ConnectionStrings:csISGROUP %>"
    SelectCommand="SearchPublications" SelectCommandType="StoredProcedure">
    <SelectParameters>
        <asp:ControlParameter ControlID="DropDownList1" Name="autore"
            PropertyName="SelectedValue" Type="Int32" />
        <asp:ControlParameter ControlID="DropDownList2" Name="topic"
            PropertyName="SelectedValue" Type="String" />
        <asp:ControlParameter ControlID="DropDownList3" Name="groupby"
            PropertyName="SelectedValue" Type="String" />
    </SelectParameters>
</asp:SqlDataSource>
```

Figura 5.28: Data Source costruitta con la stored procedure

La ricerca degli informazioni sulle pubblicazioni puo essere svolto a seconda uno dei seguenti criteri:

- Tutte le pubblicazioni su un autore (Topic impostato a All)
- Tutte le pubblicazioni su un tema della ricerca(Author impostato a All)
- Tutte le pubblicazioni relativi a un autore e su un tema preciso della ricerca
- Tutte le pubblicazioni del gruppo(caso di default, Author e Tpoic impostati a All)

Le figure 5.29, 5.30 e 5.31 presentano la scelta di un autore, la scelta di un topic e infine i risultati ottenuti. La figura 5.32 invece presenta il caso di default.

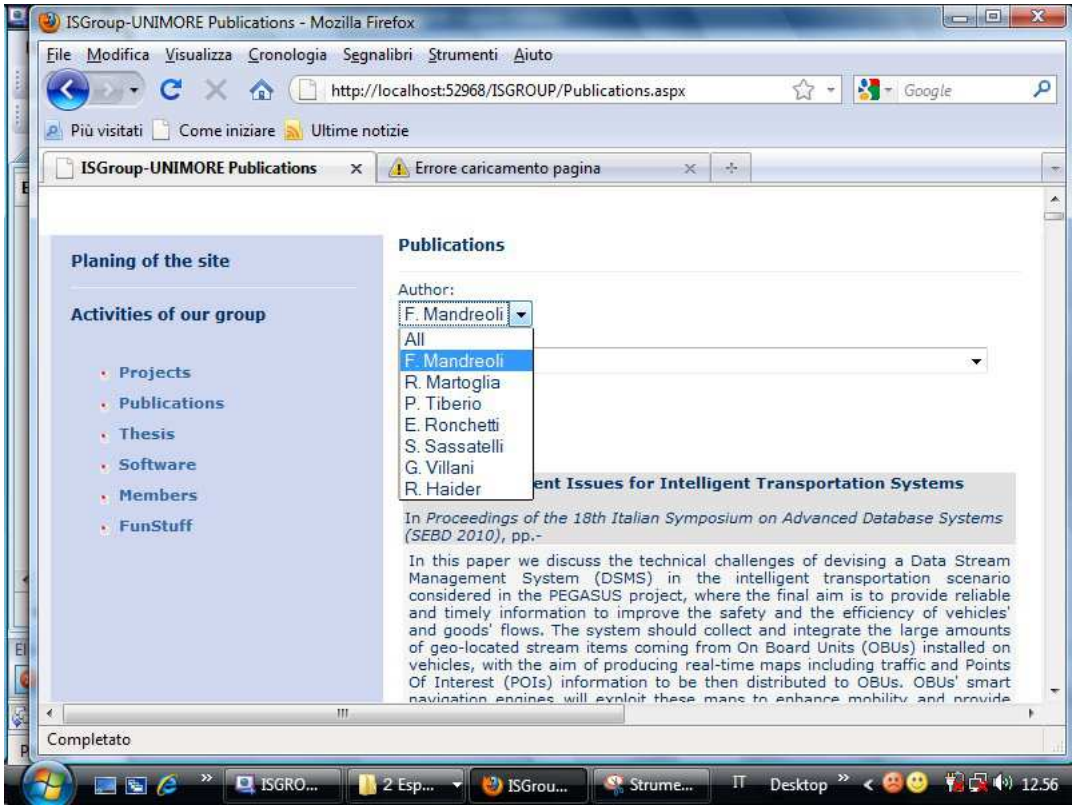

Figura 5.29: Scelta dell'autore

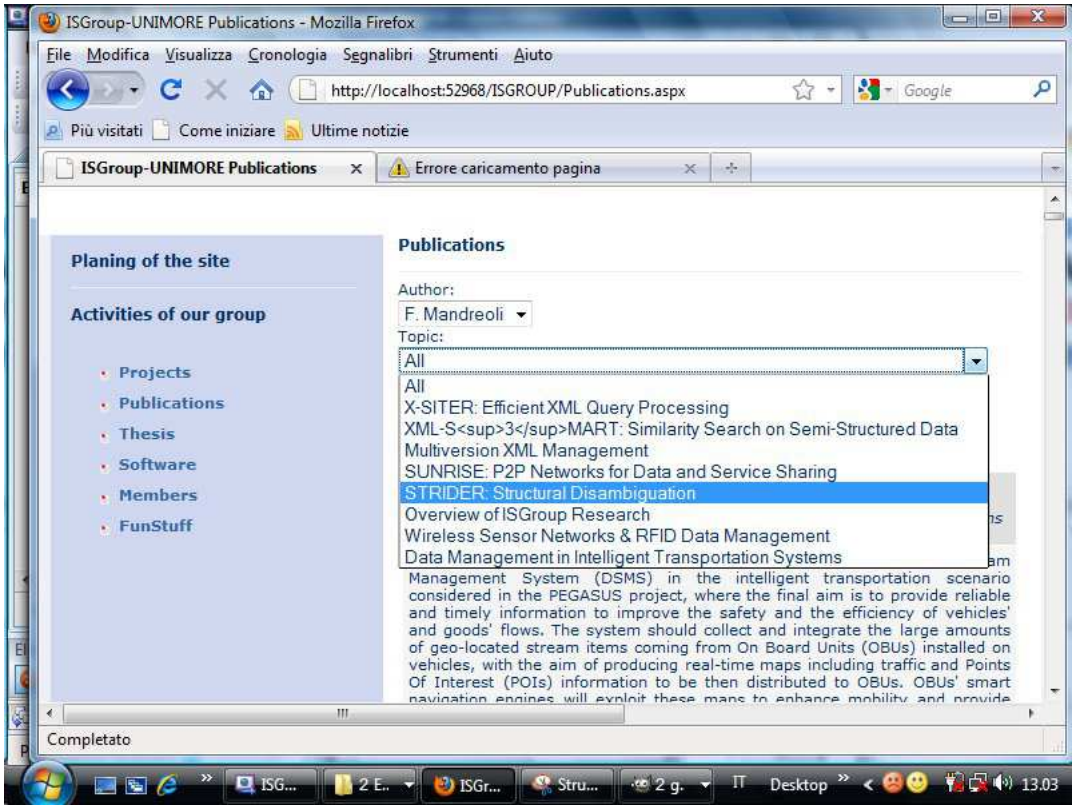

Figura 5.30: Scelta del topic

Figura 5.31: Resultati ottenuti

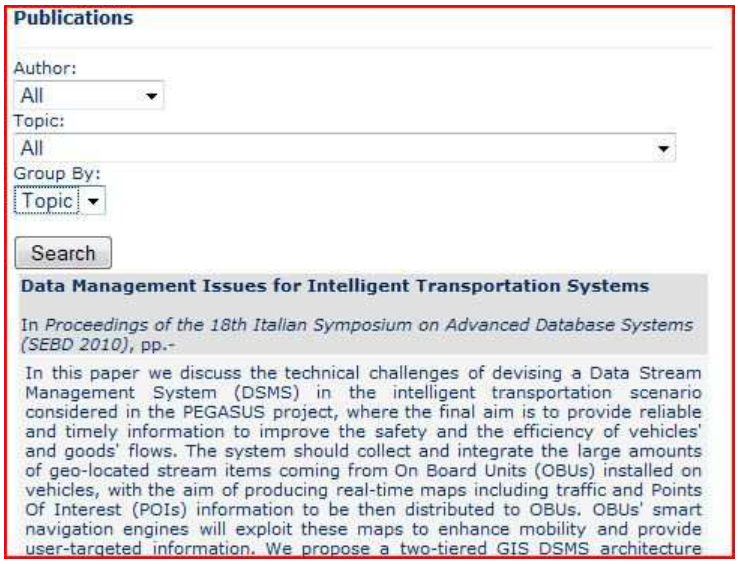

Figura 5.32: Risultati per tutte le pubblicazioni

Conclusione

Nell'ambito di questa tesi sono stati raggiunti diversi obiettivi:

- É stato fatto un confronto tra ASP e ASP.Net tecnologie Microsoft per capire quali sono le differenze dal punto di vista dell'elaborazione di una richiesta
- É stato fatto un confronto fra ADO e ADO.Net
- É stato utilizzato ASP.NET e SQL Server 2008 per sviluppare alcune pagine del sito Web del gruppo ISGroup.
- É stato mostrato che lo sviluppo di un sito Web va fatto meglio con ASP.Net accoppiato a SQL Server che ASP e Access

Per quanto riguarda gli sviluppi futuri si potrebbero migliorare i seguenti aspetti:

- Utilizzare AJAX per megliorare il lato Client e per aggiungere alcuni effeti grafici
- Utilizzare AJAX per velocizzare di più le pagine del sito aggiornando soltanto la parte della pagina dove c'è la necessità non l'intera pagina.

Parte III

Appendici

Appendici

5.4 Il Codice aspx

5.4.1 MasterPage.master

5.4.2 Default.css

5.4.3 Default.skin

5.4.4 Projects.aspx

5.4.5 These.aspx

5.4.6 Publications

Codice della stored procedure

Codice della pagina Publications.aspx

```
<%@ Master Language="VB" CodeFile="MasterPage.master.vb" Inherits="MasterPage"
%>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title></title>
    <link  href=  "Default.css" rel="stylesheet" type="text/css"/>
</head>
<body>
    <form id="form1" runat="server">
       <!-- Zone : En tête -->
      <div class="header">
            <div class="menua"><!-- Zone : menu en haut de la page -->
              <a href=http://www.isgroup.unimo.it>ISGroup Home</a>|<a
href=http://www.dii.unimo.it>DII Home</a>|<a
href=http://www.ing.unimore.it>Faculty Home</a>  |<a
href=http://www.unimore.it>University Home</a>
            </div>
            <div class="rotator"> <!--anneau publicitaire  pubblicita-->
             <asp:AdRotator   ID="AdRotator1" runat="server" />
            </div>
            <div class="nav"> <!-- Zone : bara di navigazione affiggendo il
camino della pagina corrente -->
                <asp:SiteMapPath ID="SiteMapPath1" runat="server">
                    <PathSeparatorTemplate>
                        <asp:Image ID="Image1" runat="server"
                        ImageUrl="~/App_Themes/Default/Images/bullet-2.gif" />
                    </PathSeparatorTemplate>
                </asp:SiteMapPath>
              </div>
      </div>

     <!-- Zone : Centro -->
    <div class="page"><!-- Zone : Zone de navigation à gauche de la page  -->
       <div id="sidebar"> <!--Zone : Espace membre  -->
                <h1>Planing of the site</h1>
              <div id="liensEspaceMembre"></div><hr />
                <h1>Activities of our group</h1><!--Zone : Services des
annonces proposés par le site  -->
                  <!--h2>Consult publications, tesis and projects</h2-->
                   <asp:SiteMapDataSource ID="SiteMapDataSource1"
ShowStartingNode="false" runat="server" />
                   <asp:TreeView ID="TreeView1" runat="server"
                     DataSourceID="SiteMapDataSource1">
                   </asp:TreeView>
        </div>
        <div id="content"><!-- Zona : Zona avendo il contenuto dei pagini figli
nel un controllo <asp :contentplaceholder> -->
          <asp:ContentPlaceHolder ID="Main" runat="server">
          </asp:ContentPlaceHolder>
        </div>
    </div>
        <!-- Zone : Pied di pagina-->
    <div class="footer">
        <asp:Menu ID="Menu1" runat="server" DataSourceID="SiteMapDataSource1">
        </asp:Menu>
        <asp:Literal ID="ltlPiedDePage" runat="server"
                   Text="ISGroup Information Systems Group |Paolo
            Tiberio|Federica Mandreoli| Ricardo Martoglia"></asp:Literal>
    </div>
    </form>
```

```
body {
        background-color: #ffffff;
        margin: 0;
        padding:0;
        text-align: left;
        font-family: verdana;
        font-size: 11px;
    font-weight: normal;
        color: #003063;
}

/* Zone EN-TETE*/
.header
{
        background-image: url(Images/topbar.gif);
        background-repeat: no-repeat;
    position: relative;
    width: 760px;
    height: 120px;
    margin-left: auto;
    margin-right: auto;
    font-weight: normal;
}
.rotator{
        position: absolute;
        right: 30px;
        top: 17px;
}
.nav
{
        width: 95%;
    position: absolute;
    margin: 0;

    margin-top: 100px;
    padding-left:35px;
    color: #ffffff;
    font-size: 10px;

    background-color:#c3c3c3;
}
.menua
{
    position: absolute;
    right:15px;
    top: 0px;
    color:#ffffff;
    font-size: 10px;
}
/* Zona PIE Di PAGINA */
.footer
{
    background-image: url(images/footer.gif);
    margin-left: auto;
    margin-right: auto;
    margin-top:0.4em;
    width: 760px;
    text-align: center;
   font-size: 9px;
    vertical-align: bottom;
}
/* Zona CENTRALE  PAGINA */
.page
{    margin-top:30px;
    margin-left: auto;
    margin-right: auto;
    text-align: left;
    background-image: url(images/body-repeat.gif);
    background-repeat: repeat-y;
    position: relative;
    width: 712px;
    padding: 0px 24px;
}
#sidebar {
        float: left;
        width: 214px;
        height: 100%;
}

#content {
        margin-left: 245px;
        min-height:400px;
}
```

```
#liensEspaceMembre
{
    margin-left: 2em;
    margin-top: 1.0em;
}
.gridcontent
{
    margin-top: 1.0em;
    width:460px;
    border-right: #7b9ebd 1px solid;
    border-top: #7b9ebd 1px solid;
    border-left: #7b9ebd 1px solid;
    border-bottom: #7b9ebd 1px solid;
}
.menu
{
    margin-bottom: 0.3em;
    text-transform: uppercase;
     vertical-align:baseline;
}
.treeview
{
    margin-left: 2em;
    margin-top: 1.0em;
    margin-bottom: 1.12em;
    line-height: 2.1em;
}
.headercell
{
    color: #003063;
    font-weight:bold;
    vertical-align:text-top;
    text-align:right;
    width:150px;
}

/* BARRE DE FRACTIONNEMENT */
hr {
       border: 0;
       border-top: 1px solid #efeff7;
       height: 1px;
}

/* TITRE */
h1,h2,h3
{
    margin: 0;
       font-family: verdana;
       font-size: 11px;
}
h1
{      margin-top: 1.0em;
       margin-bottom: 1.12em;
    font-size: 12px;
    font-weight: bold;
}
h2
{
    font-weight: normal;
}
h3
{
    font-weight: bold;
}

/* LIENS HYPERTEXTE */
a {
       text-decoration: none;
       color: #31659C;
       font-weight:bold;
}

a:visited {
       color: #31659C;
       font-weight:bold ;
}
```

```
a:hover {
        text-decoration: underline;
        color: #31659C;
        font-weight:bold;
}

a:active {
        color: #31659C;
        font-weight:bold;
}

/* LISTE */

ul {
        list-style-type: none;
        list-style-image: url(images/bullet-2.gif);
        list-style-position: outside;
        line-height: 2.1em;
        vertical-align: middle;
        margin-left: 1.25em;
}
.link {
        list-style-image: url(images/bullet-1.gif);
}
/* FORMAT DE LA DESCRIPTION D'UN SUJET */
#description
{
    margin-top: 2.0em;
    background-color: #efeff7;
    width:450px;
    border-bottom: #738294 thin solid;
    display:block;
}
#title
{
    text-align:left;
}
```

```
<asp:AdRotator runat="server"
AdvertisementFile="PublicationsBanniere.xml"
BorderColor="#C0C0FF" BorderStyle="Solid" BorderWidth="1px"/>
<asp:DropDownList ForeColor="#003063" runat="server"
></asp:DropDownList>
<asp:Label runat="server" BackColor="transparent" width="140px"
height="15px" align="right"></asp:Label>
<asp:TextBox runat="server" BackColor="transparent"
valign="middle" border="1px solid #929292" width="140px"
height="15px" ForeColor="#003063"></asp:TextBox>
<asp:GridView SkinId="tableMaitre" CssClass="gridcontent"
runat="server" GridLines="none">
  <SelectedRowStyle BackColor="#EFEFF7" />
</asp:GridView>
<asp:GridView SkinId="tableDetails" CssClass="gridcontent"
runat="server"  GridLines="none" >
  <RowStyle BackColor="#efeff7" ></RowStyle>
</asp:GridView>
<asp:DetailsView runat="server" FieldHeaderStyle-
HorizontalAlign="Right" BorderStyle="None" BorderWidth="0"
CellPadding="5"  Width="460px" >
  <FieldHeaderStyle CssClass="headercell" ></FieldHeaderStyle>
</asp:DetailsView>
<asp:TreeView CssClass="treeview"
                        runat="server"
                        ExpandDepth="0"
                        ExpandImageUrl="Images/signeplus.gif"

CollapseImageUrl="Images/signemoins.gif"
                        NoExpandImageUrl="Images/bullet-1.gif">
</asp:TreeView>
<asp:Menu CssClass="menu"  StaticPopOutImageUrl="Images/bullet-
3.gif"
          orientation="horizontal" runat="server" >
         <StaticMenuItemStyle ItemSpacing="6em"/>
         </asp:Menu>
<asp:SiteMapPath runat="server">
    <PathSeparatorStyle Font-Bold="True" ForeColor="White" />
    <CurrentNodeStyle ForeColor="#003063" />
    <NodeStyle Font-Bold="True" ForeColor="White" />
    <RootNodeStyle Font-Bold="True" ForeColor="White" />
</asp:SiteMapPath>
```

```
<%@ Page Title="ISGroup-UNIMORE Projects" Language="VB"
MasterPageFile="~/MasterPage.master" AutoEventWireup="false"
CodeFile="Projects.aspx.vb" Inherits="Projects" %>
<asp:Content ID="Content1" ContentPlaceHolderID="Main" Runat="Server">

<h1>            &
nbsp;           Research
Interests & Projects</h1>
    <hr>
    <center>
    <h2>At present our research interests are about the following
topics:</h2>
    <h2>(chose a topic to access to more information about it) </h2>
    </center>
    <asp:SqlDataSource ID="dsTopicOfInterest" runat="server"
        ConnectionString="<%$ ConnectionStrings:csISGROUP %>"
        SelectCommand="SELECT * FROM [TopicOfInterest]"
        ProviderName="System.Data.SqlClient">
    </asp:SqlDataSource>
    <asp:DropDownList ID="DropDownList1" runat="server" AutoPostBack="True"
        DataSourceID="dsTopicOfInterest" DataTextField="Titolo"
DataValueField="ID">
    </asp:DropDownList>
     <asp:SqlDataSource ID="dsTopicOfInterest1" runat="server"
        ConnectionString="<%$ ConnectionStrings:csISGROUP %>"
        SelectCommand="SELECT * FROM [TopicOfInterest] WHERE ([ID] = @ID)">
        <SelectParameters>
            <asp:ControlParameter ControlID="DropDownList1" Name="ID"
                PropertyName="SelectedValue" Type="Int32" />
        </SelectParameters>
    </asp:SqlDataSource>
    <asp:DataList ID="DataList1" runat="server" DataKeyField="ID"
        DataSourceID="dsTopicOfInterest1" style="margin-left: 0px">
        <ItemTemplate>
  <table class=BorderedTable cellSpacing=1 cellPadding=1 width="99%"
align=center>
                <tbody>
                <tr bgColor=#e1e1e1><td class=pubblfontbig  colSpan=2><b>
                <%# Eval("Titolo") %> </td> </tr>
                <tr bgColor=#f5f5f5>     <!-- Description -->
                   <td class=pubblabstr width="100%" colspan=2>
                       <br><div align=justify> <%# Eval("Descrizione") %>
</div><br></td></tr>
                <tr bgColor=#f5f5f5>
                   <td class=pubblfont  width="50%">
                      <div align=center>Related Publications<br>
                          <a href=>Click here</a><!-- Data binding with the
publication-->
                      </div>
                   </td>
                   <td class=pubblfont  width="50%">
                      <div align=center>Related prototypes<br>
                          <a href=>Click here</a><br><!-- Data binding with
the prototype-->
                        <br>
                      </div>
                   </td>
                </tr>
                </tbody>
            </table>
        </ItemTemplate>
    </asp:DataList>
        <br>
    <hr />
```

```
<h2> Past research interests   </h2>
    <asp:SqlDataSource ID="dsPastTopics" runat="server"
        ConnectionString="<%$ ConnectionStrings:csISGROUP %>"
        SelectCommand="SELECT [ID], [Titolo] FROM [TopicOfInterest]
WHERE ([Search] = @Search)">
        <SelectParameters>
            <asp:Parameter DefaultValue="N" Name="Search"
Type="String" />
        </SelectParameters>
    </asp:SqlDataSource>
     <asp:DropDownList ID="DropDownList2" runat="server"
AutoPostBack="True"
        DataSourceID="dsPastTopics" DataTextField="Titolo"
DataValueField="ID">
   </asp:DropDownList>
   <asp:SqlDataSource ID="dsPastTopics1" runat="server"
        ConnectionString="<%$ ConnectionStrings:csISGROUP %>"
        SelectCommand="SELECT * FROM [TopicOfInterest] WHERE ([ID] =
@ID)">
        <SelectParameters>
            <asp:ControlParameter ControlID="DropDownList2" Name="ID"
                PropertyName="SelectedValue" Type="Int32" />
        </SelectParameters>
    </asp:SqlDataSource>
     <asp:DataList ID="DataList2" runat="server" DataKeyField="ID"
        DataSourceID="dsPastTopics1" style="margin-left: 0px">
        <ItemTemplate>
  <table class=BorderedTable cellSpacing=1 cellPadding=1 width="99%"
align=center>
               <tbody>
               <tr bgColor=#e1e1e1><td class=pubblfontbig
colSpan=2><b>
               <%# Eval("Titolo") %>
                                  </td>
               </tr>
               <tr bgColor=#f5f5f5>     <!-- Description -->
                  <td class=pubblabstr width="100%" colspan=2>
                      <br><div align=justify> <%# Eval("Descrizione")
%> </div><br>
                  </td>
               </tr>
               <tr bgColor=#f5f5f5>
                  <td class=pubblfont  width="50%">
                     <div align=center>Related Publications<br>
                         <a href=>Click here</a><!-- Data binding
with the publication-->
                     </div>
                  </td>
                  <td class=pubblfont  width="50%">
                     <div align=center>Related prototypes<br>

                         <a href=>Click here</a><br><!-- Data binding
with the prototype-->
                        <br>
                     </div>
                  </td>
               </tr>
               </tbody>
            </table>
```

```
<h2>We are participating in the following<b> national and european funded
projects</b>: </h2>
<h2>(chose a topic and access to more information)</h2>
    <asp:SqlDataSource ID="dsProjects" runat="server"
        ConnectionString="<%$ ConnectionStrings:csISGROUP %>"
        SelectCommand="SELECT [id], [titolo] FROM [Progetti] WHERE (([inizio]
&lt;= @inizio) AND ([fine] &gt;= @fine))">
        <SelectParameters>
            <asp:Parameter DefaultValue="2010" Name="inizio" Type="Int32" />
            <asp:Parameter DefaultValue="2010" Name="fine" Type="Int32" />
        </SelectParameters>
    </asp:SqlDataSource>
    <asp:DropDownList ID="DropDownList3" runat="server" AutoPostBack="True"
        DataSourceID="dsProjects" DataTextField="titolo" DataValueField="id">
    </asp:DropDownList>
    <asp:SqlDataSource ID="dsProjects1" runat="server"
        ConnectionString="<%$ ConnectionStrings:csISGROUP %>"
        SelectCommand="SELECT * FROM [Progetti] WHERE ([id] = @id)">
        <SelectParameters>
            <asp:ControlParameter ControlID="DropDownList3" Name="id"
                PropertyName="SelectedValue" Type="Int32" />
        </SelectParameters>
    </asp:SqlDataSource>
    <asp:DataList ID="DataList4" runat="server" DataKeyField="ID"
        DataSourceID="dsProjects1" style="margin-left: 0px">
        <ItemTemplate>
  <table class=BorderedTable cellSpacing=1 cellPadding=1 width="99%"
align=center>
                <tbody>
                <tr bgColor=#e1e1e1><td class=pubblfontbig  colSpan=3><b> <%#
Eval("Titolo") %></b> <br><i><%#Eval("Tipo")%></i>
<br><%#Eval("Inizio")%>-<%#Eval("Fine")%>
                 </td>
                </tr>
                <tr bgColor=#f5f5f5>     <!-- Description -->
                   <td class=pubblabstr width="100%" colspan=3>
                     <br><div align=justify> <%# Eval("Descrizione") %>
                     </div><br>
                   </td>
                </tr>
                <tr bgColor=#f5f5f5>
                   <td class=pubblfont  width="33%">
                      <div align=center>
            <a href="paginaWeb" target="_blank">Project Home Page</a>
                   </div>
                   </td>
                   <td class=pubblfont  width="33%">
                      <div align=center>Related Publications<br>
                        <a href=>Click here</a><br><!-- Data binding with the
publications-->
                      </div>
                   </td>
                   <td class=pubblfont  width="33%">
                      <div align=center>Related prototypes<br>
<a href=>Click here</a><br><!-- Data binding with the prototype-->

                     </div>
                   </td>
                </tr>
                </tbody>
            </table><br>

        </ItemTemplate>
    </asp:DataList>
     <br>
    <hr>
```

```
<h2>Past projects: </h2>
<h2>(chose a topic and access to more information)</h2>
    <asp:SqlDataSource ID="dsPastProject" runat="server"
        ConnectionString="<%$ ConnectionStrings:csISGROUP %>"
        SelectCommand="SELECT [id], [titolo] FROM [Progetti] WHERE ([fine] &lt;
@fine) ORDER BY [inizio] DESC">
        <SelectParameters>
            <asp:Parameter DefaultValue="2010" Name="fine" Type="Int32" />
        </SelectParameters>
    </asp:SqlDataSource>
    <asp:DropDownList ID="DropDownList4" runat="server" AutoPostBack="True"
        DataSourceID="dsPastProject" DataTextField="titolo"
DataValueField="id">
    </asp:DropDownList>
    <asp:SqlDataSource ID="dsPastProject1" runat="server"
        ConnectionString="<%$ ConnectionStrings:csISGROUP %>"
        SelectCommand="SELECT * FROM [Progetti] WHERE ([id] = @id)">
        <SelectParameters>
            <asp:ControlParameter ControlID="DropDownList4" Name="id"
                PropertyName="SelectedValue" Type="Int32" />
        </SelectParameters>
    </asp:SqlDataSource>
    <asp:DataList ID="DataList5" runat="server" DataKeyField="ID"
        DataSourceID="dsPastProject1" style="margin-left: 0px">
        <ItemTemplate>
  <table class=BorderedTable cellSpacing=1 cellPadding=1 width="99%"
align=center>
 <tbody>
 <tr bgColor=#ele1e1><td class=pubblfontbig  colSpan=3><b>
<%#Eval("Titolo")%></b> <br><i><%#Eval("Tipo")%></i>
        <br><%#Eval("Inizio")%>-<%#Eval("Fine")%>
                                    </td>
                </tr>
                <tr bgColor=#f5f5f5>     <!-- Description -->
                   <td class=pubblabstr width="100%" colspan=3>
                       <br><div align=justify> <%# Eval("Descrizione") %>
</div><br>
                   </td>
                </tr>
                <tr bgColor=#f5f5f5>
                   <td class=pubblfont  width="33%">
                      <div align=center>

            <a href="paginaWeb" target="_blank">Project Home Page</a>

                      </div>
                   </td>
                   <td class=pubblfont  width="33%">
                      <div align=center>Related Publications<br>
                        <a href=>Click here</a><br><!-- Data binding with the
publications-->
                      </div>
                   </td>
                   <td class=pubblfont  width="33%">
                      <div align=center>Related prototypes<br>
<a href=>Click here</a><br><!-- Data binding with the prototype-->

                     </div>
                   </td>
                </tr>
                </tbody>
            </table><br>

        </ItemTemplate>
    </asp:DataList>
</asp:Content>
```

```
<%@ Page Title="ISGroup-UNIMORE Thesis" Language="VB"
MasterPageFile="~/MasterPage.master" AutoEventWireup="false"
CodeFile="Thesis.aspx.vb" Inherits="Thesis" %>
<asp:Content ID="Content1" ContentPlaceHolderID="Main" Runat="Server">
<center>  <h1>Theses</h1>   </center>
      <hr>
      <ul>
        <li><strong><a href="#proposed">Proposed Theses</a> (in
Italian)</strong></li>
        <li><strong><a href="#published-ing">Published Theses (Computer
Engineering - 2nd Level)</a></strong></li>
        <li><strong><a href="#published-inf">Published Theses (Computer
Science - 1st Level)</a></strong></li>
      </ul>
      <hr>
      <a name="proposed"></a><h1>Proposed Theses</h1>
       Sono disponibli tesi per la <b>Laurea Specialistica</b> e la
<b>Laurea Triennale</b> riguardanti i seguenti argomenti di ricerca:
      <ul>
<li> <h1>Middleware database per la navigazione intelligente</h1>
     (nell'ambito del progetto <a href="Projects.aspx">PEGASUS</a>)
     <p align="justify"> Obiettivo del progetto PEGASUS (ProgEtto per la
Gestione della mobilità Attraverso Sistemi infotelematici per l'ambito
Urbano, per la Sicurezza di passeggeri, veicoli e merci) è la
realizzazione di una <b>piattaforma infotelematica integrata</b> per
migliorare la mobilità cittadina. Le aree urbane del Paese sono infatti
sempre più congestionate dal traffico, con conseguente aumento degli
incidenti, dei rallentamenti e dell'inquinamento atmosferico/acustico.
La piattaforma fornirà, in particolare, un servizio di <b>navigazione
intelligente</b> basato sulla raccolta ed elaborazione in <b>tempo
reale</b> dei dati provenienti da apposite <b>On Board Unit (OBU)</b>
che, installate sui veicoli, trasmetteranno informazioni sulla loro
velocità per consentire la rilevazione di eventuali rallentamenti o
ingorghi. Per la piattaforma dovrà essere sviluppato un <b>middleware
database</b> in grado di ricevere, memorizzare e gestire le informazioni
sul traffico in tempo reale e dotato di un innovativo algoritmo di
<b>ricerca</b> e <b>inserimento</b> di <b>eventi</b> (incidenti, code)
che alterano la normale viabilità.
     </p>
     In questo ambito sono disponibili tesi sui seguenti temi:
    <ul class="tesi">
    <li>Data-intensive DBMS temporali e spaziali
            <li>Tecniche per l'elaborazione di dati dinamici
    <li>Tecniche per l'indicizzazione e l'interrogazione di moving
object
    <li>Recommender system per la profilazione utente e la predizione
delle preferenze
    <li>Tecniche per la comunicazione vehicle-to-vehicle (V2V)
    </ul>
    <li> <h1>Accesso e gestione di dati RFID per il monitoraggio di ampi
spazi esterni</h1>
     (nell'ambito del progetto <a href="Projects.aspx">Outdoor Video
Protection: visione artificiale e tecnologie informatiche innovative per
la sicurezza di ambienti di lavoro all'aperto </a>)
     <p align="justify"> Lo scopo del progetto è il monitoraggio di ampi
spazi esterni, con particolare riferimento a luoghi di lavoro pericolosi
quali i cantieri edili, al fine di rilevare intrusi o situazioni non a
norma. Si tratta di ambienti molto complessi da controllare, solitamente
dislocati su aree molto estese i cui punti di accesso sono il più delle
volte non noti e non sorvegliabili. La soluzione proposta nell'ambito
del progetto prevede di superare questi limiti utilizzando
congiuntamente <b>telecamere</b> e <b>tecnologia RFID</b> in modo da
poter allo stesso tempo <b>localizzare</b> e <b>identificare</b> le
persone nell'area. Per utilizzare i dati RFID è tuttavia necessario
sviluppare tecniche appropriate che consentano di gestire <b>enormi
quantità</b> di dati, <b>ridonanti</b>, <b>poco affidabili</b> e
```

```
<b>temporali</b>.
     </p>
     In questo ambito sono disponibili tesi sui seguenti temi:
    <ul class="tesi">
    <li>Tecniche per la memorizzazione efficiente di dati RFID
    <li>Modelli probabilistici per gestire l'incertezza dei dati
    <li>Tecniche per l'esecuzione di query temporali su stream di dati
    <li>Modelli di reasoning per sensor fusion multimodale
    </ul>
      <li> <hl>Interrogazione flessibile di dati modellati a grafo</hl>
    (nell'ambito del progetto <a href="Projects.aspx">NeP4B - Networked Peers
for Business -</a>)
     <p align="justify">
Modelli di dati basati su strutture a <b>grafo</b> si stanno sempre più
diffondendo in molteplici aree di applicazione legate alle basi di dati, basti
pensare,  ad esempio, ai database biologici o ai dati del Web. In questi
contesti, si devono solitamente gestire <b>quantità enormi</b> di dati
<b>eterogeneei</b> per i quali è impensabile avere una perfetta conoscenza del
<b>vocabolario</b> utilizzato e delle modalità di organizzazione degli stessi.
Per questi motivi, una delle più importanti sfide della ricerca in questo
contesto è lo sviluppo di meccanismi di <b>interrogazione flessibili</b> che
consentano agli utenti di esprimere facilmente le loro richieste e recuperare
agevolmente i dati utili.
     </p>
    In questo ambito sono disponibili tesi sui seguenti temi:
    <ul class="tesi">
    <li>Tecniche di graph-matching non esatto
    <li>Tecniche di query processing su grafi
    <li>Tecniche di estrazione della semantica delle relazioni tra i nodi di un
grafo
    <li>Gestione dell'interfaccia con l'utente
    </ul></ul>
      Per informazioni rivolgersi alla <a href="federica.aspx"> Dott.ssa
Federica Mandreoli</a> o all'<a href="riccardo.aspx"> Ing. Riccardo
Martoglia</a>.
    <a name="published-ing"></a> <hl>Published Theses (Computer Engineering -
2nd Level)</hl>
         <asp:SqlDataSource ID="dsTesi" runat="server"
        ConnectionString="<%$ ConnectionStrings:csISGROUP %>"
        SelectCommand="SELECT * FROM [Tesi] WHERE ([Laurea] = @Laurea) ORDER BY
[ID] DESC">
        <SelectParameters>
            <asp:Parameter DefaultValue="Ing" Name="Laurea" Type="String" />
        </SelectParameters>
    </asp:SqlDataSource>
    <asp:DropDownList ID="DropDownList1" runat="server" AutoPostBack="True"
        DataSourceID="dsTesi" DataTextField="Autore" DataValueField="ID">
    </asp:DropDownList>
    <asp:SqlDataSource ID="dsTesi1" runat="server"
        ConnectionString="<%$ ConnectionStrings:csISGROUP %>"
        SelectCommand="SELECT [Titolo], [Autore], [Relatore], [Descrizione],
[PDFLink], [PPTLink], [Correlatore], [AnnoAcc] FROM [Tesi] WHERE ([ID] = @ID)">
        <SelectParameters>
            <asp:ControlParameter ControlID="DropDownList1" Name="ID"
                PropertyName="SelectedValue" Type="Int32" />
        </SelectParameters>
    </asp:SqlDataSource>
```

```
<asp:DataList ID="DataList1" runat="server" DataSourceID="dsTesi1">
        <ItemTemplate>
            <table class=BorderedTable cellSpacing=1 cellPadding=1 width="99%"
align=center>
                <tbody>
                <tr bgColor=#e1e1e1><td class=pubblfontbig  colSpan=2><b><%#
Eval("Autore") %> </b>
                                    <br><%# Eval("Titolo") %>
<br><%# Eval("Relatore") %> - <%# Eval("Correlatore") %>
                                    <br>Anno <%# Eval("Annoacc") %></td>
</tr>
                <tr bgColor=#f5f5f5>
                   <td class=pubblabstr width="100%" colspan=2>
                      <br><div align=justify>
<%# Eval("Descrizione") %></div><br>
                   </td>
                </tr>
                <tr bgColor=#f5f5f5>
                    <td class=pubblfont  width="50%">
                      <div align=center>
                            <a href="/files/<%# Eval("PDFLink")%>"><img
alt="Download thesis in PDF format"
src="C:\Users\SOTTANG.PAT\Desktop\ISGROUP\App_Themes\Default\Images\pdficon.gif"
border=0></img></a>
                      </div>
                   </td>
                   <td class=pubblfont  width="50%">
                      <div align=center>
                            <a href="/files/<%# Eval("PPTLink")%>"
target="_blank"><img alt="Download presentation"
src="C:\Users\SOTTANG.PAT\Desktop\ISGROUP\App_Themes\Default\Images\ppticon.gif"
border=0></img></A>
                      </div>
                   </td>
                   </tr>
                </tbody>
            </table>
        </ItemTemplate>
    </asp:DataList>
    <a name="published-inf"></a>
    <hr>
```

```
<h1>Published Theses (Computer Science - 1st Level)</h1>
      Click on the thesis titles to access more information, including the links
to the thesis in pdf format and to the related presentation (if available).
        <asp:SqlDataSource ID="dsTesInf" runat="server"
        ConnectionString="<%$ ConnectionStrings:csISGROUP %>"
         SelectCommand="SELECT [Autore], [ID] FROM [Tesi] WHERE ([Laurea] =
@Laurea) ORDER BY [ID] DESC">
        <SelectParameters>
        <asp:Parameter DefaultValue="Inf" Name="Laurea" Type="String" />
        </SelectParameters>
    </asp:SqlDataSource>
    <asp:DropDownList ID="DropDownList2" runat="server" AutoPostBack="True"
        DataSourceID="dsTesInf" DataTextField="Autore" DataValueField="ID">
    </asp:DropDownList>
    <asp:SqlDataSource ID="dsTesInf1" runat="server"
        ConnectionString="<%$ ConnectionStrings:csISGROUP %>"
        SelectCommand="SELECT * FROM [Tesi] WHERE ([ID] = @ID)">
        <SelectParameters>
            <asp:ControlParameter ControlID="DropDownList2" Name="ID"
                PropertyName="SelectedValue" Type="Int32" />
        </SelectParameters>
    </asp:SqlDataSource>
    <asp:DataList ID="DataList2" runat="server" DataKeyField="ID"
        DataSourceID="dsTesInf1">
        <ItemTemplate>
           <table class=BorderedTable cellSpacing=1 cellPadding=1 width="99%"
align=center>
                <tbody>
                <tr bgColor=#e1e1e1><td class=pubblfontbig  colSpan=2><b><%#
Eval("Autore") %> </b>
                                        <br><%# Eval("Titolo") %>
<br><%# Eval("Relatore") %> - <%# Eval("Correlatore") %>
<br>Anno <%# Eval("Annoacc") %></td></tr>
                <tr bgColor=#f5f5f5>
                   <td class=pubblabstr width="100%" colspan=2>
<br><div align=justify>
<%# Eval("Descrizione") %></div><br>
                   </td>
                </tr>
                <tr bgColor=#f5f5f5>
                    <td class=pubblfont  width="50%">
                      <div align=center>
                            <a href="/files/<%# Eval("PDFLink")%>"><img
alt="Download thesis in PDF format"
src="C:\Users\SOTTANG.PAT\Desktop\ISGROUP\App_Themes\Default\Images\pdficon.gif"
border=0></img></a>
                        </div>
                   </td>
                   <td class=pubblfont  width="50%">
                      <div align=center>
                          <a href="/files/<%# Eval("PPTLink")%>"
target="_blank"><img alt="Download presentation"
src="C:\Users\SOTTANG.PAT\Desktop\ISGROUP\App_Themes\Default\Images\ppticon.gif"
border=0></img></A>
                    </div>
                   </td>
                   </tr>
                </tbody>
            </table>
        </ItemTemplate>
    </asp:DataList>
</asp:Content>
```

```
ALTER PROCEDURE SearchPublications
      @autore int,
      @topic varchar,
      @groupby varchar
      AS
      /* autore e topic */
IF @autore!=0 AND @topic!='0'
  IF @groupby='1' /* Order by Year*/
      BEGIN
      SELECT Pub.ID, Pub.Year, Pub.Topic, [Pub-Aut].Autore, Pub.Titolo, Pub.Libro,
      Pub.Luogo, Pub.Month, Pub.Tipo, Pub.PDFLink, Pub.Abstract, Pub.BibTeX,
      Pub.Abbr,Pub.FlashLink, Pub.[Pag-Iniz], Pub.[Pag-Fin]
      FROM Pub INNER JOIN [Pub-Aut] ON Pub.ID = [Pub-Aut].Pub
      WHERE([Pub-Aut].Autore = @autore) AND (Pub.Topic = @topic) ORDER BY Pub.Year DESC
      END

   ELSE
       /*IF @groupby='2' */ /* Order by Topic*/

      BEGIN
      SELECT Pub.ID, Pub.Year, Pub.Topic, [Pub-Aut].Autore,
      Pub.Titolo,Pub.Libro,Pub.Luogo, Pub.Month, Pub.Tipo,Pub.PDFLink,
      Pub.Abstract,Pub.BibTeX,Pub.Abbr,Pub.FlashLink,Pub.[Pag-Iniz],Pub.[Pag-Fin]
      FROM  Pub INNER JOIN [Pub-Aut] ON Pub.ID = [Pub-Aut].Pub
      WHERE ([Pub-Aut].Autore = @autore)
            AND
           (Pub.Topic = @topic)
     ORDER BY Pub.Topic DESC,Pub.Year DESC
      END

        /*  Solo Topic*/





   /*  Solo Topic*/

   IF @autore=0 and @topic!='0'
       BEGIN
       SELECT * FROM Pub WHERE Pub.Topic=@topic ORDER BY Pub.Year DESC
       END

       /*  Solo Autore*/
   IF @autore!=0 AND @topic='0'
      IF @groupby='1'  /*  Order by Year*/
      BEGIN
        SELECT Pub.ID, Pub.Year, Pub.Topic, [Pub-Aut].Autore, Pub.Titolo,
          Pub.Libro,Pub.Luogo, Pub.Month, Pub.Tipo,
          Pub.PDFLink,Pub.Abstract,Pub.BibTeX,
          Pub.Abbr,Pub.FlashLink,Pub.[Pag-Iniz],Pub.[Pag-Fin]
      FROM Pub INNER JOIN [Pub-Aut] ON Pub.ID = [Pub-Aut].Pub
      WHERE [Pub-Aut].Autore = @autore)  ORDER BY Pub.Year DESC
      END

      ELSE /*   Order by Topic*/

      BEGIN
          SELECT Pub.ID, Pub.Year, Pub.Topic, [Pub-Aut].Autore, Pub.Titolo,
          Pub.Libro,Pub.Luogo, Pub.Month, Pub.Tipo, Pub.PDFLink, Pub.Abstract,
           Pub.BibTeX, Pub.Abbr, Pub.FlashLink, Pub.[Pag-Iniz], Pub.[Pag-Fin]
          FROM Pub INNER JOIN [Pub-Aut] ON Pub.ID = [Pub-Aut].Pub
          WHERE ([Pub-Aut].Autore = @autore)  ORDER BY Pub.Topic DESC,Pub.Year DESC
     END
   /*  nessun vincolo*/
   IF @autore = 0  AND @topic ='0'
    BEGIN
              SELECT * FROM Pub ORDER BY Pub.Year DESC

    END
```

```
<%@ Page Title="ISGroup-UNIMORE Publications" Language="VB"
MasterPageFile="~/MasterPage.master" AutoEventWireup="false"
CodeFile="Publications.aspx.vb" Inherits="Publications" %>

<asp:Content ID="Content1" ContentPlaceHolderID="Main" Runat="Server">
<form  id="form2" method="post" action="Publications.aspx" >
  <h1> Publications</h1>
<hr>
    <asp:SqlDataSource ID="dsAutore" runat="server"
        ConnectionString="<%$ ConnectionStrings:csISGROUP %>"
        SelectCommand="SELECT [ID], [Nome] FROM [Autori] WHERE ([Search] = @Search)">
        <SelectParameters>
    <asp:Parameter DefaultValue="Y" Name="Search" Type="String" />
        </SelectParameters>
    </asp:SqlDataSource>
    <asp:SqlDataSource ID="dsTopicOfInterest" runat="server"
     ConnectionString="<%$ ConnectionStrings:csISGROUP %>"
     SelectCommand="SELECT [ID], [Titolo] FROM [TopicOfInterest] WHERE ([Search] =
@Search)">
        <SelectParameters>
         <asp:Parameter DefaultValue="Y" Name="Search" Type="String" />
        </SelectParameters>
    </asp:SqlDataSource>
```

```
<div>
    <asp:Label ID="authorLabel" runat="server" Text="Author:"></asp:Label><br>
    <asp:DropDownList ID="DropDownList1"  runat="server" DataSourceID="dsAutore"
ValidationGroup="pouet" AppendDataBoundItems="true"
        DataTextField="Nome" DataValueField="ID"
OnSelectedIndexChanged="DropDownList1_SelectedIndexChanged"    >
        <asp:ListItem Value="0">All</asp:ListItem>
    </asp:DropDownList>
    <asp:RequiredFieldValidator ID="RequiredFieldValidator1" runat="server"
ControlToValidate="DropDownList1" InitialValue="0" Text="" Display="Dynamic"
ValidationGroup="pouet"></asp:RequiredFieldValidator>
     <br>
    <asp:Label ID="topicLabel" runat="server" Text="Topic:"></asp:Label>
    <asp:DropDownList ID="DropDownList2" runat="server"   ValidationGroup="pouet"
        DataSourceID="dsTopicOfInterest" DataTextField="Titolo"   AppendDataBoundItems="true"
        DataValueField="ID" OnSelectedIndexChanged="DropDownList2_SelectedIndexChanged" >
        <asp:ListItem Value="0">All</asp:ListItem>
    </asp:DropDownList>
    <asp:RequiredFieldValidator runat="server" ControlToValidate="DropDownList2"
InitialValue="0" Text="" Display="Dynamic" ValidationGroup="pouet"></asp:RequiredFieldValidator>

    <asp:SqlDataSource ID="dsResearch" runat="server"
        ConnectionString="<%$ ConnectionStrings:csISGROUP %>"
        SelectCommand="SearchPublications" SelectCommandType="StoredProcedure">
        <SelectParameters>
            <asp:ControlParameter ControlID="DropDownList1" Name="autore"
                PropertyName="SelectedValue" Type="Int32" />
            <asp:ControlParameter ControlID="DropDownList2" Name="topic"
                PropertyName="SelectedValue" Type="String" />
            <asp:ControlParameter ControlID="DropDownList3" Name="groupby"
                PropertyName="SelectedValue" Type="String" />
        </SelectParameters>
    </asp:SqlDataSource>
    <asp:Label ID="Label3" runat="server" Text="Group By:"></asp:Label><br />
    <asp:DropDownList ID="DropDownList3" runat="server"
OnSelectedIndexChanged="DropDownList3_SelectedIndexChanged" >
    <asp:ListItem Value="1"> Year</asp:ListItem>
    <asp:ListItem Value="2"> Topic</asp:ListItem></asp:DropDownList><br />
    <br />
```

```
<asp:Button ID="Button1" runat="server" Text="Search"  OnClick="Button1_Click"/>
        <asp:SqlDataSource ID="SqlDataSource1" runat="server"
            ConnectionString="<%$ ConnectionStrings:csISGROUP %>"
            SelectCommand="IdPublications" SelectCommandType="StoredProcedure">
            <SelectParameters>
                <asp:ControlParameter ControlID="DropDownList1" Name="autore"
                    PropertyName="SelectedValue" Type="Int32" />
                <asp:ControlParameter ControlID="DropDownList2" Name="topic"
                    PropertyName="SelectedValue" Type="String" />
                <asp:Parameter Direction="InputOutput" Name="pub" Type="Int32" />
            </SelectParameters>
        </asp:SqlDataSource>

    </div> <!-- fine de la procedura di ricerca e inizio della stampa dei risultati-->
    <asp:DataList ID="DataList1" DataSourceID="dsResearch" runat="server">
    <ItemTemplate>
    <TABLE class=BorderedTable cellSpacing=1 cellPadding=1 width="99%" align=center>
      <TBODY>
      <TR bgColor=#e1e1e1><TD class=pubblfontbig  colSpan=4><b>
 <%# Eval("Titolo")%></b> <br>
    <br><DIV class=pubblfont>In <i><%# Eval("Libro")%></i><%# Eval("Pag-Iniz")%>, pp.<%#
Eval("Pag-Iniz")%>-<%# Eval("Pag-Fin")%>
                                        </DIV></TD></TR>

                <TR bgColor=#f5f5f5>     <!-- Abstract -->
                   <TD class=pubblfont width="100%" colspan=3>

                      <DIV >
                         <TABLE>
                            <TR><TD class=pubblabstr width="100%">
                               <DIV align=justify>
 <%# Eval("Abstract")%> </DIV><br>
                            </TD></TR>
                         </TABLE>
                      </DIV>
                   </TD>
                </TR>
```

Bibliografia

[1] Valerio Romeo. *Progetto e Sviluppo di un Sistema Informativo per la Gestione di una Base di Dati di Pubblicazioni Scientifiche*. Tesi di Laurea, 2006-2007

[2] Bob Reselman. *Practical ASP3*. Jackson Libri, 2001

[3] Dave Mercer. *Guida a ASP.NET*. McGraw-Hill, 2002