Università degli Studi di Roma "La Sapienza"

Facoltà di Lettere e Filosofia

Anno accademico 2001-2002

Tesi di laurea

La Bibliografia Elettronica dei Trovatori in Rete. Tecnica di interfaccia e moduli di ricerca

Candidato: Luca De Nigro

Relatore: Prof. Stefano Asperti

Correlatore: Prof. Pasquale Stoppelli

1 INTRODUZIONE 4
1.1 Premessa 4
1.2 La BEdT online 6

2 LA BASE DATI BEDT 11
2.1 Definizione della base dati 12
2.2 Tabelle 14
2.3 Funzionalità della base dati 15

3 TECNOLOGIE INFORMATICHE 18
3.1 Le reti telematiche 18
3.2 I servizi di base della rete 19
3.3 Il Modello ipertestuale 20
3.4 Tipologie di accesso ai dati – architetture di base 21
3.4.1 accesso via reti telematiche dedicate (pre-web) 24
3.4.1.1 tipologia terminal-server 24
3.4.1.2 tipologia file-server 25
3.4.1.3 tipologia client-server 25
3.4.2 accesso via World Wide Web 27
3.4.2.1 file server via web 27
3.4.2.2 client-server via web 28
3.5 Architettura dell'applicazione web-based 29
3.6 L'ambiente di sviluppo 29
3.6.1 ASP – caratteristiche 33
3.6.2 ASP.NET – caratteristiche 34
3.6.3 CFM – caratteristiche 34
3.6.4 PHP – caratteristiche 35
3.6.5 JSP – caratteristiche 35
3.7 Struttura-tipo di un'applicazione ASP.NET 36

4 L'APPLICAZIONE WEB BEDT 38
4.1 Architettura di sistema 38
4.1.1 Requisiti minimi di sistema per la visualizzazione 39
4.1.2 Software applicativo utilizzato per la generazione del codice 40
4.2 Struttura dell'applicazione 40
4.2.1 Impostazione dell'interfaccia a menu 40
4.2.2 Descrizione dei comandi 42
4.3 Dettagli di programmazione di alcune pagine-tipo 46
4.3.1 File: indice_pc.aspx 46
4.3.2 File: ind_canz_pc.aspx 52
4.4 Problemi rilevati durante la creazione del codice 63
4.5 Elementi di cornice non implementati 64
4.6 Considerazioni sulla possibilità di integrare documenti originali nella base di dati 65

5 BIBLIOGRAFIA 66
5.1 Risorse online 68
5.1.1 Risorse in Rete dedicate alla filologia romanza e in particolare alle ricerche filologiche online 68
5.1.2 Risorse in Rete dedicate alle tecnologie di sviluppo per il web .. 72

6 INDICE DELLE FIGURE 75

7 IL CD-ROM 76

1 INTRODUZIONE

L'oggetto di questa tesi è l'analisi delle possibili modalità di diffusione attraverso le reti telematiche di una base di dati di tipo umanistico e la conseguente implementazione di un modello di consultazione tecnicamente funzionante. L'applicazione realizzata, i materiali necessari al suo funzionamento, la documentazione e le istruzioni per l'installazione sono contenuti nel cd-rom allegato.

1.1 Premessa

L'enorme diffusione, a livello mondiale, di elaboratori in grado di memorizzare grandi quantità di documenti, la disponibilità di supporti con i quali distribuire a basso costo le informazioni (*cd-rom*), l'avvento delle reti di trasmissione dati ad alta velocità e la disponibilità di programmi per la ricerca delle informazioni pongono certamente in crisi, oggi, le strutture tradizionali di diffusione del sapere: stampa, case editrici, biblioteche, centri di documentazione.

Ciononostante, riguardo alle possibili evoluzioni future, formulare ipotesi realistiche è ancora arduo e, in fondo, prematuro. Certamente i settori indicati stanno subendo una profonda trasformazione, ma su almeno un punto non sembrano porsi difficoltà: la direzione intrapresa consente e incoraggia le diverse "funzionalizzazioni" dei supporti tradizionali e di quelli informatici.

In un contesto, quello filologico, in cui l'attenzione per il testo scritto, l'oggetto libro e l'esperienza della scrittura "solida" (contrapposta all'evanescenza del testo visualizzato sullo schermo del computer) sono l'essenza stessa della materia, lo strumento informatico può trovare una naturale collocazione in tutti quegli aspetti della ricerca che hanno come scopo la razionalizzazione e la velocizzazione della documentazione e il reperimento di informazioni "strutturate", fino a diventare uno strumento di indagine a sé stante in grado di fornire prospettive originali.

E' questo il caso della *Bibliografia Elettronica dei Trovatori (BEdT).* Una base di dati, di prossima pubblicazione, sviluppata dal Prof. Stefano Asperti, di questa Università, con la quale si è inteso dare risposta all'esigenza di uniformare e rendere maggiormente efficienti, grazie all'informatica, gli strumenti-base di lavoro del filologo romanzo: i repertori A. Pillet – H. Carstens *Bibliographie der Troubadours*, Halle a.S. 1933 e I. Frank, *Répertoire métrique de la poésie des troubadours*, Paris 1953-57[1].

Tali repertori cartacei sono, inevitabilmente, non coordinati tra loro. Tuttavia, essendo strutturati sulla base di principi di classificazione comuni, quali identificazione del testo e dell'autore secondo precise e puntuali norme di catalogazione, è stato possibile procedere all'integrazione reciproca delle informazioni, attraverso l'uso di un sistema di archiviazione dati informatico (*database*); integrazione che ha reso possibile, come risultato immediatamente evidente, una rapida consultazione dei contenuti ed ha gettato le basi per la costruzione di un indice analitico elettronico di tutti i canzonieri contenenti testi trobadorici aperto a qualsiasi integrazione futura.

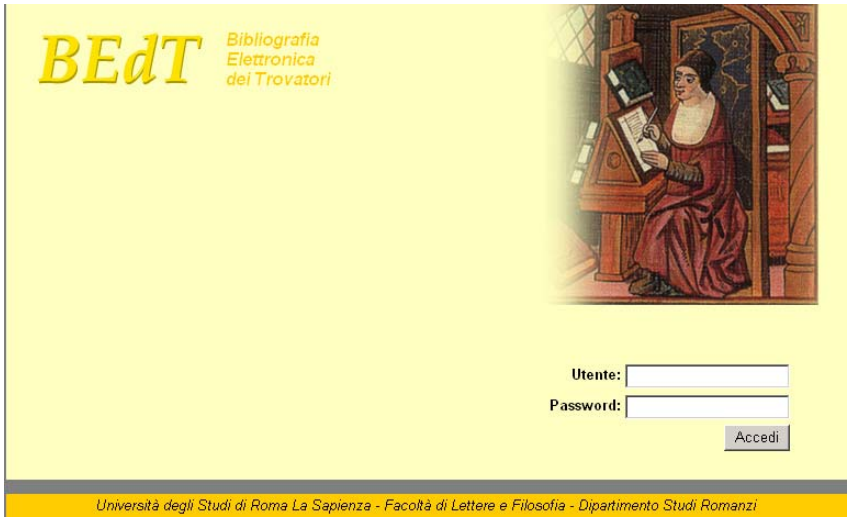


Figura 1 – Pagina principale dell'applicazione BEdT

[1] La Base di dati BEdT Asperti (in formato Microsoft Access 2000) è disponibile attualmente solo in versione di prova (nel cd-rom allegato). Per tutte le indicazioni sulla filosofia progettuale e strutturale cfr. Asperti, *BEdT Istruzioni per l'uso* (*MANUALE.doc*), p. 4 e ss., anch'esso reperibile all'interno del cd-rom allegato.

1.2 La BEdT online

Pur garantendo Internet la più ampia diffusione del progetto BEdT, unita alla possibilità di aggiornamento in tempo reale - al di fuori della portata di qualsiasi supporto di altra natura - le notevoli limitazioni in termini di potenza, velocità di elaborazione, complessità di gestione tipiche della struttura ipertestuale della Rete costringono ad una riorganizzazione del materiale, se non dal punto di vista strutturale e contenutistico, dal punto di vista della logica dell'applicazione e delle modalità di presentazione e interrogazione dell'archivio. Da ciò dipenderà in buona misura l'efficacia di ogni operazione futura sui dati ed, in ultima analisi, la fortuna del sistema[2].
Problema fondamentale nel trasferimento in Rete è, di conseguenza, la scelta di una tecnologia efficiente in relazione all'entità delle indagini che dovranno essere svolte sulla base dati anche, e forse soprattutto, rispetto al "sistema Internet", non dimenticando che un tale progetto può richiedere notevoli caratteristiche di flessibilità operativa, per proporsi come un'interfaccia multifunzionale e in grado di adattarsi facilmente a tutti i nuovi contenuti che potranno aggiungersi[3].

Dopo aver analizzato la struttura interna della base dati sono state individuate le principali tecniche di programmazione di contenuti dinamici su

---

[2] E' noto infatti che la consultazione di qualsiasi risorsa è pesantemente condizionata dall'effettiva velocità di accesso alle pagine in Rete, parametro estremamente variabile e legato principalmente alla tipologia di connessione utilizzata dall'utente (modem, isdn, adsl, rete via cavo, fibra ottica, satellite) e quindi non direttamente controllabile.

[3] La sempre maggiore diffusione di Internet nelle scuole e nei centri di studio e ricerca ha incoraggiato la scelta di posizionamento sul web della base dati. Il progetto si affianca, così, in maniera originale, alla già numerosa serie di risorse online dedicate alla filologia romanza e alle ricerche sui testi in generale. Per una rassegna delle principali, si veda il relativo capitolo più avanti.

Internet, per evidenziare la metodologia più adatta ed impostare un progetto operativo funzionante e immediatamente pubblicabile[4].

Nella sua forma esteriore il programma realizzato, e qui proposto, si presenta come un'applicazione Internet di tipo *client-server* in cui il *server* - il computer remoto su cui risiedono la logica dell'applicazione e l'insieme dei dati di archivio - interagisce con il *client*, il computer dell'utente, attraverso la Rete, fornendo a distanza gli elementi dell'interfaccia grafica di fruizione e le risposte alle interrogazioni, in un continuo interscambio di informazioni.
Non si tratta quindi di un semplice "sito web-vetrina" basato sulle tecnologie di ipertesto, come è norma diffusa su Internet, bensì di una struttura dinamica in grado di *autoaggiornarsi sia in presenza di richieste dell'utente sia in presenza di modifiche introdotte dall'amministratore al livello di origine dei dati*.
La struttura base dell'applicazione prevede una suddivisione primaria basata sui parametri di ricerca ricavabili direttamente dall'organizzazione dei dati dei repertori Pillet-Carstens e Frank attraverso richiami incrociati:

- AUTORI: es. "PC 001" (indice di catalogazione del repertorio) oppure "nome dell'autore"
- TESTI: es. "PC 001,001" oppure "F 001,001" oppure "incipit testo"
- FONTI manoscritte: es. "sigla canzoniere" oppure "segnatura"

In seconda battuta viene offerto un ventaglio di proposte di indagini sui dati bio-bibliografici fondamentali dei testi, degli autori e dei canzonieri.

---

[4] Non si dimentichi, infatti, il problema del notevole grado di complicazione tecnologica che comporta una scelta simile. Di fatto viene attivato un processo apparentemente estraneo alle problematiche tipicamente filologiche, sebbene si possa ragionevolmente affermare che queste nuove modalità di "interrogare" un testo possono dar luogo a una vera e propria "metodologia" di ricerca filologica non alternativa alla tradizionale, bensì propriamente complementare.

In tal modo si "costruisce" un vero e proprio "enunciato dinamico di ricerca", facilitato oltretutto dalla sua costituzione in forma di "menu gerarchico" in tutto e per tutto simile a quello standard del sistema operativo Windows, di chiara lettura e interpretazione per gli utenti di qualsiasi esperienza.

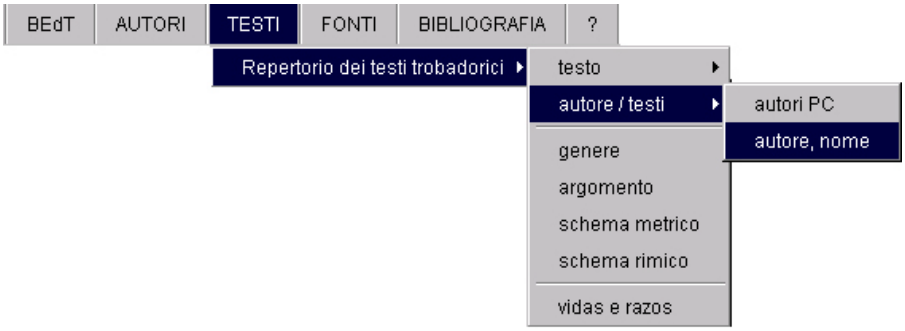


Figura 2 – Menu di comando dell'applicazione. Scelta di un percorso di ricerca.
Es.: rapporto di tutti i testi di un dato trovatore, indicato per nome

Dopo aver individuato un percorso generico, tramite menu, l'informazione sul contesto della ricerca viene inviata al server, il quale risponde impostando un indice dinamico relativo al percorso scelto.

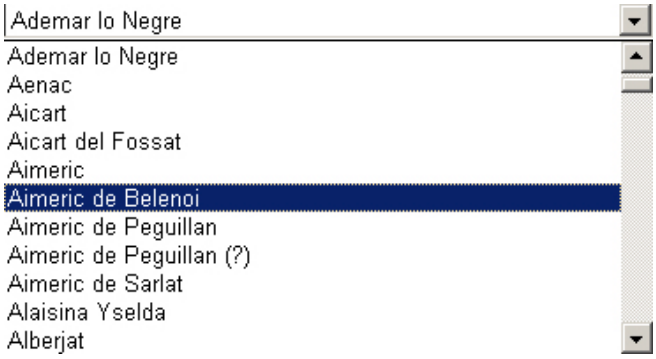


Figura 3 – Esempio di indice dinamico. Estrazione dal DB di tutti i nomi dei trovatori

L'indicazione della scelta da parte dell'operatore completa quindi l'enunciato di ricerca ponendo il server nella condizione di individuare direttamente una "scheda" informativa di riepilogo oppure un "rapporto", in formato tabellare, con possibilità di dettaglio per ogni elemento (riga) della tabella.

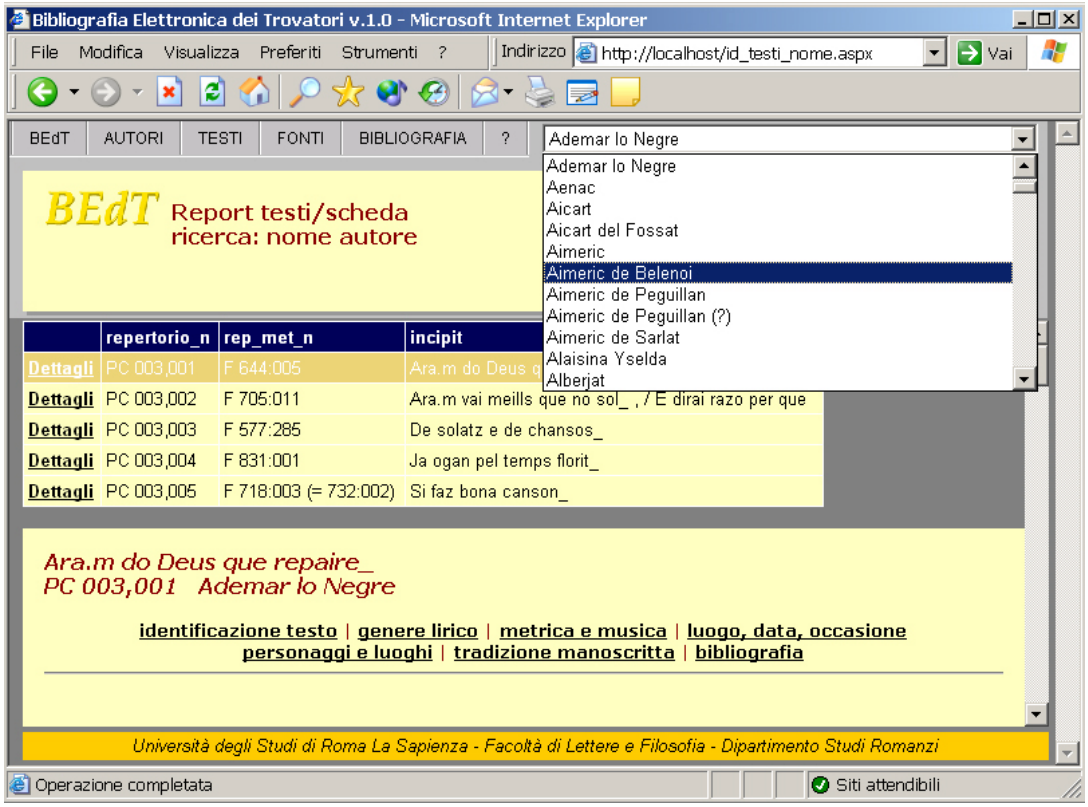


Figura 4 – Aspetto completo della pagina di ricerca. Tramite l'indicazione del nome di un trovatore si accede al rapporto di tutti i testi attribuiti con possibilità di visualizzare i dettagli per ogni testo in un'apposita scheda

La struttura così creata mira a guidare passo passo l'utente verso il raggiungimento del risultato, indipendentemente dalla sua competenza strettamente informatica, dando cioè il modo di concentrarsi sul dato prettamente filologico, lasciando al programma il compito di reagire dinamicamente, modificando i puntamenti in relazione a qualsiasi nuova configurazione.

La logica dell'applicazione, allo stato attuale, non prevede ricerche di tipo *semantico*, non essendo previsti moduli di ricerca avanzata nello specifico del *testo.*

E' stato privilegiato, in questo stadio progettuale, un'efficace modello di *ostensione* del dato. Nonostante ciò non mancano interessanti prospettive individuabili già ad una prima visione dei dati così disposti. Si vedano, a titolo di esempio, i moduli relativi ai rapporti sui testi basati sulla ricerca per elementi strutturali quali schema metrico, rimico, genere ed argomento,

capaci di evidenziare, anche a colpo d'occhio, sia un'estrema variabilità delle tipologie del corpus trobadorico che numerose suggestioni e associazioni potenzialmente inedite.

Si sottolinea il fatto che una tale impostazione di interfaccia è comunque aperta a una successiva implementazione di ulteriori moduli di ricerca, completamente liberi, basati per esempio su campi testuali ed operatori logici booleani, mentre è di fatto già pronta ad accogliere, senza sostanziali modifiche, un numero indefinito di nuovi repertori contestualizzati alle diverse realtà della ricerca in campo filologico.

2 LA BASE DATI BEDT

La base di dati BEdT Asperti si presenta come un'applicazione database di Microsoft Access 2000, sviluppata a partire da una prima versione in Access 2.0 poi sostituita dalla versione '97, nell'ambito del sistema operativo Microsoft Windows 95, poi 98, ora XP Pro[5].

La scelta primaria di utilizzare un'applicazione di database, d'ora in poi definito anche "DB", nell'ambito del gruppo di lavoro universitario, è stata dettata, in prima istanza, dalla necessità di inserire i dati in possesso in una struttura solida, facilmente elaborabile e potenzialmente espandibile a piacimento.

A questo proposito, bisognerà ricordare che tutte le applicazioni che accedono ai dati fanno normalmente uso, per funzionare, del solo *File System*, un modulo presente in tutti i sistemi operativi e finalizzato alla gestione dei file; questa impostazione reca con sé il notevole svantaggio di vincolare i dati all'applicazione che li ha creati attraverso un legame a doppia dipendenza[6].

Nel caso in cui debba operare con flessibilità sui dati, invece, il sistema informativo si deve affidare ad uno strumento software alternativo chiamato *Database Management System* (DBMS) che rappresenta, in prima approssimazione, un vero e proprio strato di "comunicazione" tra le applicazioni e i dati[7].

Attraverso il DBMS è possibile effettuare sul DB operazioni di *definizione*, *manipolazione*, *interrogazione*.

---

[5] Vedi il relativo file *BEdT2k01.mdb* nel cd-rom allegato. All'apertura del database verrà visualizzato il *Pannello Comandi* generale che dà accesso alle varie maschere. Per ogni maschera è disponibile la consultazione del relativo foglio dati basato sulle tabelle.

Sull'uso e sulle proprietà del database Microsoft Access, cfr. Davis 1999 e Cassel-Palmer 1999.

[6] Cfr. Naccarato 1999, p. 3.

[7] *Ibidem,.* p. 4, fig. 1.1.

Nel DBMS è inoltre incluso un linguaggio per le operazioni sopradescritte. Uno standard de facto è il linguaggio SQL (*Structured Query Language*), comune a tutti i pacchetti commerciali di DBMS e che verrà ampiamente utilizzato per le operazioni di estrazione dei dati[8].

L'utilizzo di un DBMS fa sì che i dati *non dipendano più dalle applicazioni* e che queste possano essere *modificate*, addirittura *sostituite*, lasciando invariata la base di dati.
Nel nostro caso Microsoft Access verrà utilizzato, in qualità di DBMS, per quanto riguarda l'aggiornamento in locale, da parte del gruppo di lavoro incaricato di alimentare la sorgente dati, mentre la vera e propria applicazione pubblica di elaborazione sarà rappresentata dall'interfaccia web di presentazione delle informazioni e dai moduli interattivi di ricerca.

2.1 Definizione della base dati

La BEdT è una base di dati "relazionale" ovvero segue il principio teorico matematico della *corrispondenza d'insiemi*[9].
Sul piano pratico la *relazione* pone in corrispondenza tra loro alcuni dati; ad esempio costituiscono una relazione i dati relativi ad una classe di elementi:

- "nome autore", "numero di repertorio", "luogo di origine", "luogo di attività"

oppure

---

[8] Per una panoramica sul linguaggio SQL, con la descrizione dettagliata di tutti i comandi, cfr. Kline 2000. Vedi anche le sezioni relative a SQL in Davis 1999, pp. 405-450 e Cassel 1999, pp. 389-417. Sull'integrazione di SQL con moduli dinamici per Internet, cfr. Payne 2001, pp. 225-248 e, nello specifico dell'accesso ai dati con Microsoft .NET, MacKenzie-Sharkey 2002, pp. 259-271.

[9] Sui rapporti tra basi di dati e applicazioni umanistiche - archivi di biblioteche, centri di documentazione, etc - soprattutto a livello di impostazione strutturale, cfr. Fadini-Savy 1999, pp. 112-115 e par. ss.

- "incipit testo", "nome autore", "genere", "numero di repertorio", "testimoni"

Queste relazioni sono rappresentate all'interno di *tabelle*, nelle quali ciascuna colonna rappresenta un attributo (detto anche *campo*) e ciascuna riga una ennupla di corrispondenti valori.
Di conseguenza una base di dati relazionale è un insieme di relazioni (cioè di tabelle).
Nell'ambito della tabella un insieme di attributi è detto *chiave* se identifica univocamente una riga della tabella.

Microsoft Access - [TROVATORI_PROVENZALI : Tabella]

File Modifica Visualizza Inserisci Formato Record Strumenti Finestra ?

rep_n_aut	repe	BdT_au	autore	sigla	graf	epoca	gene
PC 000	PC	000	sigla che indica i testi adespoti	-			@
PC 001	PC	001	Ademar	Adem			3
PC 002	PC	002	Ademar Jordan	AdJord			4
PC 003	PC	003	Ademar lo Negre	AdNegre			4
PC 004	PC	004	Ademar de Poitiers	AdPoit			3
PC 005	PC	005	Ademar de Rocaficha	AdRocaf			@
PC 006	PC	006	Aenac	Aenac			3
PC 006a	PC	006a	Aicart	Aic			6
PC 007	PC	007	Aicart del Fossat	AicFoss			6
PC 008	PC	008	Aimeric	Aim			4
PC 009	PC	009	Aimeric de Belenoi	AimBel		prima me	4
PC 010	PC	010	Aimeric de Pegulhan	AimPeg		fine XII s	4

Figura 5 – Esempio di tabella di BEdT con l'indicazione della chiave primaria fornita dal numero di repertorio Pillet

Spesso all'interno delle relazioni si stabilisce più di una chiave univoca, ne viene scelta allora una che viene detta *primaria*[10].
In BEdT la maggior parte delle volte la chiave primaria è associata al campo definito dal *numero di repertorio,* facendo leva sulle modalità di catalogazione comune dei due repertori Pillet-Carstens e Frank, ora concatenati:

- "PC 001" autore numero 1 del corpus Pillet BdT (riconosciuto anche in Frank)

[10] Sull'uso di chiavi univoche "naturali" nell'impostazione di basi di dati, in previsione di una efficiente indicizzazione, cfr. Cassel-Palmer 1999, pp. 97-99.

- "PC 001,001" testo numero 1 dell'autore numero 1 del corpus BdT

A partire da questa informazione di base è costruita l'intera impalcatura di BEdT che, rispettando fedelmente la schedatura originale di Pillet-Carstens, fatte salve alcune integrazioni risultato di recenti scoperte e alcune correzioni, può giovarsi, nativamente, di una *chiave primaria univoca* d'eccezionale valore pratico[11].

2.2 Tabelle

Le tabelle principali in cui sono organizzati i dati sono:

TROVATORI PROVENZALI
REPERTORIO TESTI TROBADORICI
VIDAS E RAZOS
INDICI CANZONIERI
TAVOLE ANTICHE
BIBLIOGRAFIA

Di seguito una breve descrizione del contenuto delle tabelle[12].
- La tabella "Repertorio testi trobadorici" costituisce il cuore del sistema e rappresenta lo strumento fondamentale di ricerca. I dati inseriti sono fondamentalmente quelli previsti nella BdT di Pillet-Carstens con aggiunte e correzioni e con l'integrazione strutturale dei dati derivanti dal repertorio metrico di Frank.
- La tabella "Indici canzonieri" contiene indici analitici descrittivi e formalizzati collegabili attraverso il numero BdT del singolo testo (idem per la tabella "Tavole antiche").

---

[11] Vedi anche *MANUALE.doc,* p. 11 per l'impostazione di base e p. 19 per il riepilogo delle integrazioni rispetto alla BdT di Pillet e Carstens.
[12] *Ibidem,* pp. 32-47.

- La tabella "Vidas e Razos" contiene il censimento dei materiali in prosa. Rispetta in massima parte le sigle di inventario Boutière-Schutz[13].
- La tabella "Trovatori provenzali" contiene una breve scheda biografica degli autori indicando i dati essenziali circa i luoghi e i tempi in cui operarono e l'estensione del loro corpus di opere. Si segnala, rispetto ai repertori classici, la presenza del dato relativo alle "generazioni" di trovatori[14].
- La tabella "Bibliografia" contiene i dati relativi alle varie tipologie di edizione dei testi e alle informazioni correlate.

In allegato è riportata la descrizione tecnica puntuale di ogni tabella con il suo contenuto informativo strutturale campo per campo: definizioni dei campi, tipologie, grandezze, particolarità[15]. Si rimanda invece alle pagine del manuale d'uso Asperti, sempre in allegato, per una discussione esaustiva sulle scelte progettuali fondamentali e sull'impostazione metodologica.

2.3 Funzionalità della base dati

La base di dati dispone di una interfaccia minimale costituita da un pannello comandi principale che dà accesso alle maschere di consultazione (essendo comunque sempre possibile la visualizzazione del foglio dati della tabella relativa).

---

[13] Si fa riferimento alla raccolta *Biographies des Troubadours* di J. Boutière e A.H. Schutz, Paris 1973. Vedi anche *MANUALE.doc*, p. 41.

[14] *Ibidem,* pp. 45-46.

[15] Vedi nel cd-rom allegato il file *analisi tabelle.rtf*, il quale riporta la documentazione tecnica relativa alla struttura di ogni tabella, come estrapolato dalla base dati attraverso la funzione "analizzatore" di Microsoft Access.

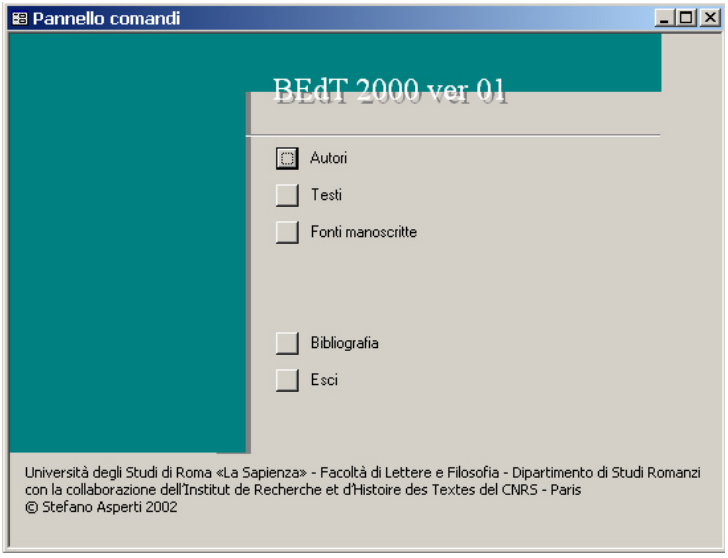


Figura 6 – Il Pannello Comandi principale della base dati

La struttura delle maschere rispecchia in modo fedele la struttura delle tabelle eliminando la visualizzazione dei soli dati eventualmente ridondanti[16].

Figura 7 – Esempio di maschera di visualizzazione dati "Indici Generali dei Canzonieri"

L'attuale concezione delle maschere, pur consentendo la restituzione di tutti i dati relativi ai singoli record delle tabelle, non consente funzioni elaborate di ricerca ed estrazione dei dati ma solo una visualizzazione sequenziale dei record.

---

[16] Vedi *BEdT2k01.mdb,* direttamente dal *Pannello Comandi*, o selezionando la sezione "maschere" nell'interfaccia di gestione del DB.

I possibili criteri di ricerca applicabili alla base dati sono legati, di conseguenza, alla comprensione da parte dell'operatore dei meccanismi logici e tecnici di interrogazione a basso livello fornita dall'applicazione Access. Vale a dire le seguenti modalità principali:

- Ordinamento rapido, la possibilità di ordinare secondo principi alfanumerici i record delle tabelle a partire dalla selezione di un campo.
- Ricerca di testo in un campo, dopo aver selezionato il campo all'interno del foglio dati della tabella.
- Selezione di dati all'interno di una tabella attraverso le funzioni standard di *Filtro o filtro avanzato*.
- Generazione di una Query, ovvero un comando logico di selezione che attraverso un appropriato linguaggio (SQL, *Structured Query Language*) consente interrogazioni e la generazione di nuove tabelle con i risultati della ricerca.

Va da sé che i sistemi di interrogazione tipici di un DBMS come Access sono ampiamente sufficienti per qualsivoglia tipologia di ricerca. Per contro, necessitano della piena competenza tecnica sia per quanto riguarda il loro corretto utilizzo sia per l'efficacia della loro impostazione metodologica in relazione agli scopi per cui possono essere usati.
Di qui la necessità di uniformare il livello di approccio alla base di dati e garantire la confrontabilità dei risultati tra diverse possibili ricerche.

3 TECNOLOGIE INFORMATICHE

3.1 Le reti telematiche

Al giorno d'oggi le reti telematiche costituiscono l'infrastruttura fondamentale per la trasmissione delle informazioni nelle sue varie forme: dati, suoni, immagini, attraverso una varietà di mezzi: l'etere, il telefono, i cavi, le fibre ottiche, i collegamenti satellitari.

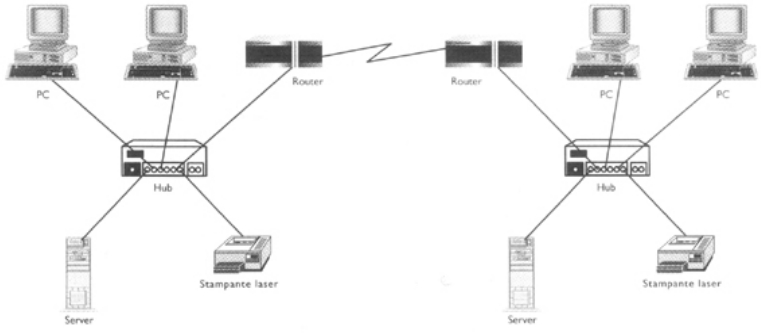


Figura 8 – Esempio di sistema di rete (fonte: Norton-Kearns 1999)

Attraverso le reti telematiche è possibile condividere risorse di qualsiasi genere: programmi, file di dati, informazioni, messaggi. E non solo. Procedure più complesse consentono oggi la realizzazione di sistemi informatici distribuiti cooperanti nell'esecuzione di un compito comune di elaborazione.

Ma le reti esistono in quanto esiste un percorso fisico del segnale tra postazioni fisse ed è per questo motivo che si hanno differenti gradi di estensione. Le reti possono essere limitate ad un ufficio (LAN, *Local Area Network*), estendersi ad un comprensorio (*Campus Network*), oppure svilupparsi in un'ampia area geografica (WAN, *Wide Area Network*), ma *tutte*

possono essere a loro volta collegate tra loro in modo da formare una *Rete di Reti* di cui la più nota, naturalmente, è la *rete Internet*[17].

3.2 I servizi di base della rete

I servizi normalmente disponibili sulla rete sono il più delle volte predisposti per sistema o grazie a organizzazioni fornitrici, e sono[18]:

- *Log-In remoto*: una qualsiasi stazione di lavoro può essere configurata temporaneamente come *terminale* di un server e, se ha un'autorizzazione, opera con i programmi del server (*TelNet*). E' questo uno dei primi usi pubblici della rete, sfruttato per permettere l'accesso a banche dati telematiche, centri di documentazione, biblioteche, etc.
- *Trasferimento di file*: la stazione di lavoro entra in collegamento con un server remoto e viene consentito l'invio e la ricezione di documenti (FTP, *File Transfer Protocol*).
- Uso di software di rete condiviso con altri utenti (*application sharing*).
- *Accesso a file di rete* per le normali operazioni di scrittura, modifica, cancellazione ed inoltre condivisione di documenti (NFS, *Network File System*). Normalmente disponibile per le operazioni di un ristretto numero di utenti.
- *Posta elettronica* (*e-mail*): lo scambio di messaggi tra gli utenti di tutti i calcolatori connessi in rete.
- *World Wide web*: un libero servizio di distribuzione in rete di informazioni, di qualsiasi tipo, dalle notizie ai giochi, dalla documentazione alla ricerca, dalle comunicazioni della Pubblica

---

[17] Si veda uno dei manuali più completi sulle Reti, Norton-Kearns 1999; per una trattazione specifica sul protocollo di comunicazione per reti TCP/IP, Leiden-Wilensky 1999. Per un approccio mirato alla comunicazione e ai nuovi media vedi anche Grossi 1999, pp. 36-98.

[18] Vedi nota precedente, ma anche Fadini-Savy 1999, pp. 44-46.

Amministrazione alle pagine private, caratterizzato dall'interfaccia completamente grafica inserita nell'ambito dei sistemi operativi a finestre. Insieme alla posta elettronica è stato il servizio di maggior successo e diffusione e il motore principale dello sviluppo di Internet[19].

3.3 Il Modello ipertestuale

Il concetto di ipertesto, così tipico di Internet, è nato quando, grazie all'impiego del computer, è divenuto possibile *ricreare* e *programmare* su un testo ex novo la pratica *non sequenziale* della lettura tipica della fruizione dei libri di studio, dei saggi, delle enciclopedie; nel complesso una buona parte, se non la maggiore, delle esplicazioni della pratica della scrittura. Fino a farne addirittura un nuovo modello di espressione artistica[20].

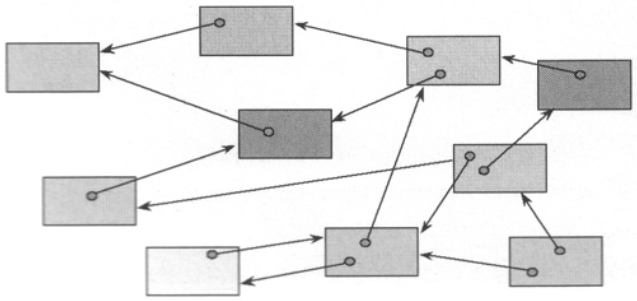


Figura 9 – Rete di collegamenti tra le varie pagine di un ipertesto (fonte: Maiocchi 2000)

[19] Attualmente World Wide Web è praticamente sinonimo di Internet. Di seguito si intenderà conservare tale concezione, sebbene non sia superfluo ricordare che si tratta di due aspetti della medesima realtà tecnologica purtuttavia *differenti*. E' la straordinaria facilità di fruizione del WWW che ne ha decretato l'essere la definitiva *modalità di interfacciamento* alla Rete a dispetto di tutte le altre tecnologie pure antecedenti ed ancora alternative.

[20] Cfr. Maiocchi 2000, pp. 9-16; Toschi 2001, pp. 98-99; Fadini-Savy 1999, pp. 225-235. Per una introduzione di tipo tecnico, vedi anche Boschin 1995, pp. 5-7 e ss.

In una struttura ipertestuale i legami tra i diversi elementi sono realizzati attraverso collegamenti (*link*) attivabili tramite la semplice pressione di un tasto, in corrispondenza di determinate parole o immagini, dando accesso immediatamente all'interno del nuovo ambiente di fruizione e stabilendo volta per volta percorsi diversi[21].

La naturale evoluzione del concetto di ipertesto è l'attuale proliferazione di siti Internet, dalle caratteristiche più disparate, dotati di brani di testo, immagini, immagini in movimento, suoni.

La visualizzazione della struttura ipertestuale è consentita nei computer dall'utilizzo di un particolare programma chiamato *browser*, ovvero "sfogliatore", in grado di comunicare con il server remoto da cui dipende l'invio delle pagine richieste ad ogni scelta dell'utente[22].

Questo tipo di fruizione è resa tecnicamente possibile dal particolare tipo di linguaggio di programmazione usato per la pagine ipertestuali su Internet: l'HTML, *HyperTextual Mark-up Language,* il linguaggio di base per la visualizzazione dei contenuti e l'invio di richieste tramite il World Wide Web[23].

3.4 Tipologie di accesso ai dati – architetture di base

Come abbiamo avuto modo di vedere, attraverso la connessione in Rete esiste la concreta possibilità di sfogliare pagine dal contenuto più vario predisposte su altri computer remoti e legate tra di loro da collegamenti ipertestuali. Si può allora affermare che il WWW è sostanzialmente un'infrastruttura per la distribuzione a calcolatori-utente di documenti mantenuti su calcolatori-server?

---

[21] Il meccanismo di indirizzamento delle pagine su Internet si basa sugli URL (*Uniform Resource Locators*), cfr. Boschin 1995, p. 12.

[22] Cfr. White 1999, guida per lo sviluppatore in ambiente Internet Explorer.

[23] Per una trattazione sistematica su HTML, vedi Boschin 1995; a proposito delle ultime evoluzioni di HTML Dinamico, vedi anche Isaacs 1997 e Livingston-Brown 1999.

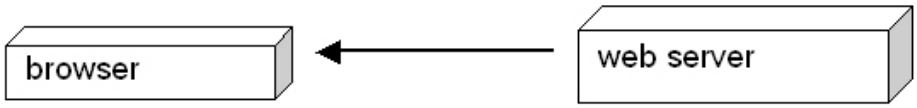


Figura 10 – Rapporto tra calcolatori-server e calcolatori-utente in configurazione non dinamica

Certamente ora non più. I limiti applicativi di questa impostazione sarebbero evidenti. In particolare nelle situazioni in cui si volesse rendere accessibili, attraverso Internet, documenti che, per molte e ovvie ragioni (dimensioni e sicurezza, per esempio), non possono essere veicolati facilmente attraverso i canali di rete.

Ma proprio per queste ragioni, per molto tempo, il WWW è rimasto estraneo al mondo delle basi di dati che pure adottano, da un punto di vista tecnico, architetture non così lontane da quelle delle reti di computer[24].

Fino a pochi anni fa per rendere accessibile un DB su rete sarebbe stato necessario ricorrere ad un DBMS molto sofisticato e dedicato (e quindi estremamente costoso) oppure accettare un considerevole aumento della complessità tecnologica del sistema (con altri evidenti problemi di costi).

L'enorme sviluppo attuale dei siti web e soprattutto le potenzialità del commercio elettronico hanno infine portato gli sviluppatori e le software-house a ricercare soluzioni valide e praticabili per mettere in comunicazione questi due mondi così vicini eppure così poco simbiotici.

Le moderne applicazioni software che consentono l'accesso dinamico ai dati sono sistemi composti da tre unità funzionali fondamentali:

- La componente di interfaccia utente (**UI**) che controlla l'*input* che l'utente fornisce e gli presenta l'*output* prodotto.
- La componente di logica applicativa (**AL**) che implementa gli specifici algoritmi di manipolazione dei dati che caratterizzano l'applicazione.
- La componente di gestione dei dati persistenti (**DB**), che si occupa della memorizzazione dei dati e del loro reperimento efficiente[25].

---

[24] Cfr. Mari 2001 (da cui traggono spunto le illustrazioni di questo capitolo), pp. xii-xiii.

[25] *Ibidem,* p. 1.

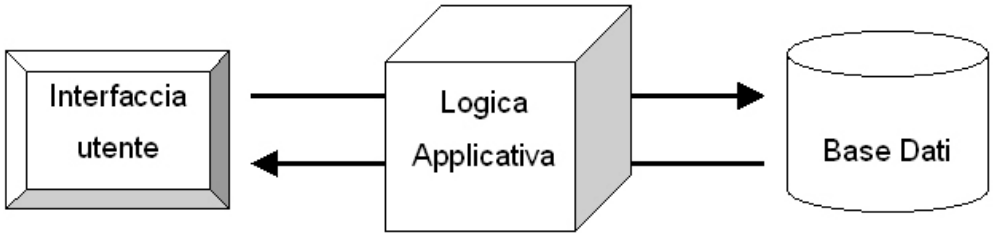


Figura 11 – Flusso di informazioni tra componenti

Questo tipo di architettura è alla base di qualsiasi tipo di programma informatico in cui sia posta la necessità di interazione con una base dati, sebbene la struttura esteriore del sistema appaia monolitica.
Nel caso di interazione via rete si dovrà tener conto, di conseguenza, dei sistemi tecnologici necessari per *gestire la connessione tra unità distanti tra loro* e spesso non concepite per lavorare in perfetta simbiosi.
In sostanza, il problema si incentra ora sulla condivisione dei dati del DB da parte di tutti gli utenti collegati, ognuno con il *proprio* calcolatore (di qualsiasi tipo) connesso alla *stessa* rete.
Il requisito fondamentale è che almeno la componente DB risieda su un calcolatore, con funzioni di server, in grado di gestire l'accesso concorrente di più utenti agli stessi dati.
I vantaggi di questa impostazione sono molteplici:

- *Prestazioni*: l'utente potrebbe aver bisogno solo di una piccola parte dei dati eventualmente selezionati.
- *Controllo di accesso*: utenti non autorizzati potrebbero non avere diritto di accesso al DB o almeno a parte di esso.
- Gestione dell'*integrità dei dati* e *aggiornamento* in tempo reale[26].

Per contro gli svantaggi risiedono nella *dipendenza dell'utente dall'infrastruttura di rete:* quando il server che contiene il DB è inattivo, nessun utente può più collegarsi e utilizzare le informazioni.

---

[26] *Ibidem,* pp. 13-14.

3.4.1 accesso via reti telematiche dedicate (pre-web)

Data dunque la struttura generale del sistema informatico di rete si possono identificare tre tipologie architetturali, distinte in base ai compiti affidati rispettivamente al server e al client attraverso una rete qualsiasi:

- Tipologia terminal-server
- Tipologia file-server
- Tipologia client-server

3.4.1.1 tipologia terminal-server

Il programma viene eseguito interamente sul server. Il sistema operativo gestisce i calcolatori collegati come semplici terminali che forniscono l'input e ricevono l'output sulla base delle informazioni richieste.
Si tratta di una variante di un programma *stand-alone* (ovvero un'applicazione singola che gira su una macchina singola). La novità nella struttura è la gestione degli accessi concorrenti. Dal lato del terminale non è praticamente richiesta alcuna potenza di calcolo se non quella necessaria per gestire la grafica visualizzata sullo schermo[27].

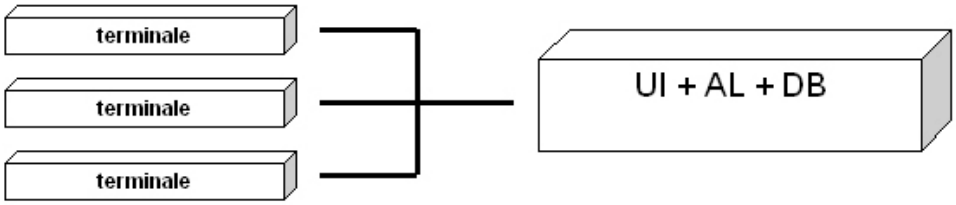


Figura 12 – Gestione dei calcolatori-utente come semplici terminali

[27] *Ibidem,* p. 15.

3.4.1.2 tipologia file-server

La logica dell'applicazione viene fisicamente separata tra calcolatore-utente e calcolatore-server. Le componenti UI e AL costituiscono un programma client installato sui calcolatori-utente mentre la componente DB rimane sul calcolatore-server. In questo modo viene solo separata la logica dai dati attraverso la rete.

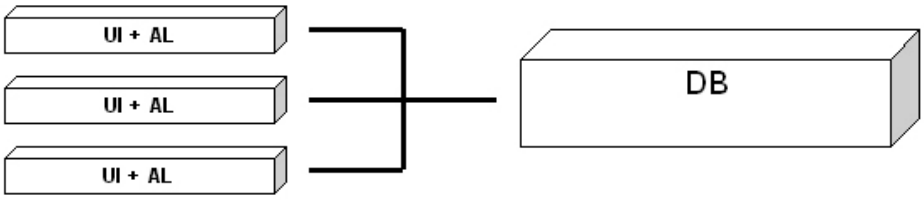


Figura 13 – Separazione dell'applicazione in due parti, collegate tramite rete

Questa tipologia comporta una serie di svantaggi operativi di cui i più evidenti sono: la difficoltà di aggiornare il programma client su tutti i calcolatori degli utenti e la gestione della sicurezza del DB che viene contattato dai programmi remoti senza controlli.
Normalmente si impiega questo sistema, in maniera efficace, solo all'interno di piccole reti private in cui ogni utente accede a dati comuni residenti sul proprio server[28].

3.4.1.3 tipologia client-server

E' opportuna una premessa: le tipologie sopradescritte prevedono in tutti i casi la presenza della componente DB all'interno del server, ma, una volta che si è scelto di non includere il DB nel programma dei calcolatori-utente,

---

[28] *Ibidem,* pp. 15-16.

l'*interfaccia di gestione originale del DB* (nel nostro caso il programma Access) *diventa inutilizzabile* in quanto non in grado di gestire convenientemente la connessione remota. Di conseguenza dovrà essere sviluppata una nuova interfaccia di comunicazione tra client e server, avendo cura di mantenere la compatibilità tra l'applicazione che si utilizza e la struttura profonda del DBMS.
La scelta di abbandonare il programma di gestione del DB è ulteriormente motivata dal fatto che la tecnologia che estrae i dati dal DB è basata su un componente software definito come una libreria di oggetti, *ActiveX Data Objects* (ADO), che funge da interfaccia di programmazione con il motore di ricerca interno del DB: *Microsoft Jet Database Engine,* motore in grado di operare anche su vari altri DB e in configurazione server-side[29].

A questo punto è possibile gestire una nuova applicazione basata sull'elaborazione concorrente tra server e client, entrambi dotati della componente AL, definita AL1 e AL2.
Al confronto con le precedenti, la soluzione client-server è l'unica che prevede una vera e propria elaborazione distribuita[30].

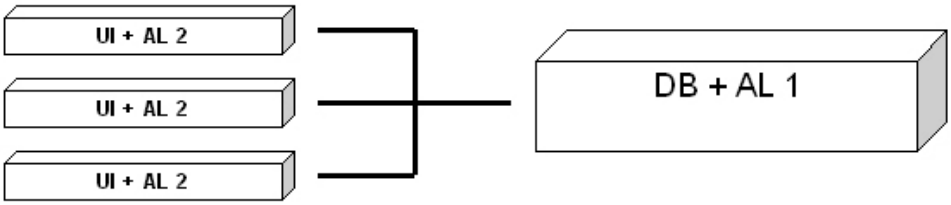


Figura 14 – Separazione della componente AL in due parti distribuite tra calcolatore-server e calcolatore-utente

[29] ADO è un modello di accesso ai dati creato in base a criteri di scalabilità e dinamicità, basato su XML. Fornisce un'interfaccia per tutte le origini dati compatibili OLE DB e consente l'accesso, la modifica e l'aggiornamento. Tra le fonti dati possibili, il motore interno di database Microsoft Jet Database Engine; cfr. Payne 2001, pp. 297-300.
[30] Cfr. Mari 2001, pp. 21-23.

Riepilogando:

- lato client: solo interfaccia utente, componente AL attiva solo a richiesta
- lato server: DB, nessuna interfaccia, componente AL sempre attiva in attesa di richieste (*background*)
- vincoli: lato-client/lato-server: linguaggio di programmazione, strato fisico della comunicazione e protocollo di comunicazione

3.4.2 accesso via World Wide Web

Le problematiche sopracitate si riferiscono naturalmente ad architetture tradizionali di rete. Negli ultimi anni si è diffusa in maniera straordinaria un'altra architettura di rete - adatta sia per reti locali che per Internet - che risolve in parte tutte le questioni sopra elencate e permette soluzioni anche ibride: il *World Wide Web*.
La caratteristica più importante del WWW è il suo protocollo di comunicazione denominato *HyperText Transfer Protocol* (HTTP), che consente al massimo grado applicazioni client-server e file-server complete[31].

3.4.2.1 file server via web

E' la configurazione tipica della maggior parte dei siti Internet. Normalmente l'utente esegue, attraverso un link ipertestuale inserito in una pagina, la richiesta di visualizzazione di una nuova pagina, seguendo percorsi in parte stabiliti a priori dall'amministratore del sistema e in parte realizzati sulla base di preferenze personali; il server web, a questo punto, non fa altro che inviare il file richiesto tramite il protocollo http; il browser dell'utente si occupa poi di

---

[31] Per una panoramica sulla suite di protocolli TCP/IP, tra cui HTTP, cfr. Leiden-Wilensky 1999, pp. 53-68.

riconoscere il file in arrivo e di attivare l'interprete dei comandi corretto per visualizzare i contenuti della pagina.
In questo caso *il browser web è assimilabile ad una interfaccia utente universale,* in quanto è in grado di visualizzare qualsiasi elemento in arrivo dal server, anche se i dati inviati avranno sempre una forma completamente statica e immodificabile[32].

3.4.2.2 client-server via web

Nella sua configurazione più avanzata, il server web si può trasformare da gestore di file in gestore di processi. Ricevendo una richiesta che contiene la specificazione di un file *eseguibile,* il server può ricercare il documento nel file system e attivarne l'esecuzione. A questo punto il programma, tecnicamente definito *back-end,* è in grado di lavorare parallelamente ai processi propri del server, producendo un qualche tipo di *output* (il risultato dell'elaborazione). L'output ottenuto viene così inviato dal server all'indirizzo dell'utente e visualizzato.
Questa soluzione, per quanto rivoluzionaria, comporta ancora taluni svantaggi in quanto, in qualche modo, ripercorre la stessa strada del sistema file-server. Ma, se il server, oltre all'informazione elaborata dal back-end, invia anche una *porzione di programma eseguibile sul calcolatore-utente,* allora l'elaborazione si snellisce realmente, il server può occuparsi più facilmente della gestione degli accessi concorrenti e il sistema può dirsi compiuto[33].

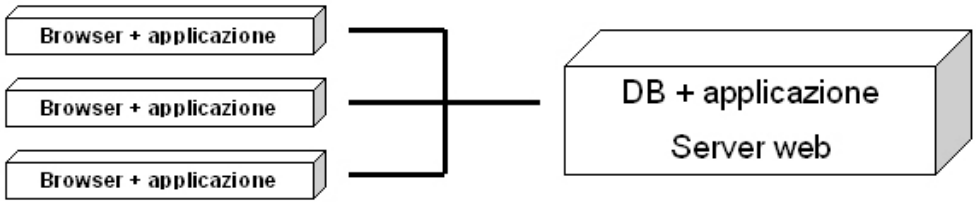


Figura 15 – Struttura finale dell'applicazione client-server via web

[32] Cfr. Mari 2001, pp. 27-34.
[33] *Ibidem,* pp. 35-43.

3.5 Architettura dell'applicazione web-based

Dalle considerazioni soprariportate si evince come la migliore configurazione possibile per un'applicazione distribuita attraverso la Rete sia, in fondo, il frutto di un compromesso tra i vari sistemi, ognuno dei quali è in grado di garantire le massime prestazioni solo in alcune occasioni particolari.
Nel caso della BEdT online il motore dell'applicazione AL1 dovrà pertanto risiedere interamente sul server, insieme alla componente DB, mentre la struttura dell'interfaccia gestionale, AL2, dovrà essere elaborata direttamente dal calcolatore-utente indipendentemente dalla piattaforma utilizzata[34].
Nei paragrafi che seguono verrà indicata la strada seguita nell'implementazione di questo modello di architettura web.

3.6 L'ambiente di sviluppo

Molti fattori intervengono nella scelta di un ambiente di sviluppo piuttosto che un altro. Di questi, alcuni ricadono nel campo dell'oggettività mentre altri, quasi fatalmente, fanno riferimento a fattori di preferenza personale, scelte di campo, background culturale oppure semplici questioni di forza maggiore. Nel nostro caso la richiesta minima, come abbiamo avuto modo di vedere, è stata la possibilità di gestire convenientemente un'applicazione di tipo client-server e non un semplice sito Internet (il quale non avrebbe dato luogo a particolari problemi di scelte tecniche, con la contropartita di limitare eccessivamente l'orizzonte delle potenzialità future).
La progettazione di un'applicazione web deve prevedere una molteplicità di aspetti tecnici, dalla scelta del sistema operativo all'ambiente software, dalla scelta della tecnologia del server al linguaggio di programmazione (sarebbe meglio dire: i tanti linguaggi di programmazione), per finire con tutti gli

[34] La parte di codice elaborata dal browser Internet viene realizzata attraverso linguaggi di programmazione completamente multipiattaforma, in modo da mantenere il principio dell'interfaccia utente universale.

innumerevoli elementi di corredo che concorrono alla quadratura dell'insieme.

E' un dato di fatto che attualmente non esiste un sistema veramente integrato di sviluppo, concretamente semplice, pienamente configurabile e malleabile, insomma valido per tutti gli scopi[35]. Sebbene sia vero che un tentativo è stato fatto in questa direzione con il pacchetto software *ColdFusion* di Macromedia, in grado di sviluppare un vero e proprio sistema di *content management* completo delle funzioni di server e amministrazione[36].

Perché allora non considerare la ricerca conclusa in partenza? Perché sviluppare un'applicazione web significa in primo luogo tener conto della compatibilità con tutte quelle entità con cui ci si deve confrontare, sotto il profilo tecnico, metodologico, e non ultimo di comunità.

Il pacchetto Macromedia, forte di una discreta diffusione ad alto livello, rappresenta pur sempre una nicchia di mercato troppo stringente, dati gli obblighi che comporta in termini di richieste server e di programmazione, rispetto al mondo in estremo fermento dello sviluppo web, proponendosi come un'alternativa troppo complessa per gli scopi prefissati.

Sul fronte della programmazione *Java-oriented* valgono sostanzialmente gli stessi argomenti. Il linguaggio e l'ambiente Java hanno subito profonde trasformazioni negli ultimi anni fino a proporsi potenzialmente come sistemi definitivi per un prossimo futuro, ma l'attuale diffusione di tali piattaforme è ancora molto legata al mondo professionale avanzato e di altissimo livello, anche grazie al felice matrimonio con sistemi operativi server *Unix* (soluzioni e-commerce, banche online), tanto da far ragionevolmente sospettare che il

---

[35] Agli inizi della programmazione web (solo qualche anno fa), era possibile usare solo la *Common Gateway Interface*, ovvero l'insieme di regole comuni sviluppate in convenzione tra i programmatori del CERN e della NCSA. Grazie a quelle specifiche e all'uso di alcuni particolari linguaggi espressamente sviluppati per il web, quale ad esempio il PERL (*Practical Extraction and Report Language*), la realizzazione di pagine dinamiche è diventata una realtà alla portata di tutti.

[36] Cfr. Uboldi 2003, pp. 18-21.

livello di complicazione (e di istruzione) necessario per sviluppare applicazioni con missioni certamente non critiche possa non rendere giustizia della potenza di base di tali sistemi[37].

A questo punto non rimane che entrare a pieno titolo in quella che può essere considerata una vera e propria *querelle* internazionale su base informatica. Si fa riferimento naturalmente alla guerra più o meno dichiarata tra i sistemi *Windows-based* e i sistemi *Linux-based*. Come è noto, il più potente gruppo informatico del mondo, Microsoft, è osteggiato *in primis* dai sostenitori del sistema cosiddetto *Open Source*, la grande comunità di sviluppatori che difende il principio della libera circolazione del codice originale[38] delle applicazioni, per servire alla causa comune del miglioramento della tecnologia senza costrizioni.

Da entrambi i "fronti" vengono proposte tecnicamente valide e completamente alternative: la tecnologia *Active Server Pages* (ASP)[39] di Microsoft e *HyperText PreProcessor* (PHP), basato principalmente su sistemi Linux[40].

Lungi dal voler esprimere una preferenza partigiana o dal voler cedere alle lusinghe dei *sistemi user-friendly,* bisognerà fare appello a considerazioni di buon senso applicato serenamente e oggettivamente al caso in questione.

ASP è una tecnologia Microsoft ben collaudata; ne è stata implementata ora una nuova versione (ASP.NET) completamente rivista, basata sulle specifiche del *NET Framework*, l'ambiente di sviluppo per tutte le nuove applicazioni in rete targate Microsoft. ASP trae le massime performance dal connubio con applicazioni quali i sistemi di database Access e soprattutto SQL Server (server di database professionale), potendo contare soprattutto

---

[37] Sulla programmazione web Java, cfr. Uboldi 2002, pp. 24-27 e Pelliccia 2002, pp. 68-71.

[38] Tecnicamente definito *codice sorgente*.

[39] Per ASP, cfr. Mitchell-Atkinsons 2000; per ASP.NET, cfr. Payne 2001.

[40] Il PHP può essere implementato anche su macchine Windows, perdendo però qualcosa in termini di funzionalità. Sull'architettura web basata su PHP, cfr. Greenspan-Bulger 2000; Meloni 2000; Mannelli 2003.

su una piattaforma diffusa in maniera più che capillare in tutto il mondo informatizzato.

PHP, dal canto suo, è in continua ascesa tra gli sviluppatori, sia per progetti amatoriali che per progetti di ampissimo respiro.

La sua filosofia aperta rappresenta, infatti, il più alto grado di libertà dai vincoli che si possa sperare in Rete, e la sua comunità di programmatori è attenta e attiva nella diffusione del sapere informatico. Ciononostante l'accettazione del suo dettato comporta l'adozione sistematica di tutti gli elementi dell'applicazione in standard *Open Source*, costringendo ad un cambio di rotta sostanziale e potenzialmente traumatico[41].

Volendo fare chiarezza nell'inevitabile confusione di linguaggi, tecnologie, piattaforme diventa allora indispensabile una certa dose di pragmatismo, sempre legato ad una prospettiva di assoluto rigore tecnico.

Di conseguenza, si considereranno dati prioritari quegli elementi individuati come maggiormente stringenti in relazione al risultato da ottenere: i fattori di compatibilità e di background culturale, le problematiche sulla sicurezza, l'espansibilità e la diffusione sul mercato; elementi che hanno fatto propendere per la scelta di un'architettura basata su tecnologia Microsoft, e, in particolare, ASP.NET.

Le garanzie ottenibili da questa scelta sono molteplici e, almeno le seguenti, di assoluto rilievo:

1) *Possibilità di mantenere tutti gli standard di lavorazione attualmente utilizzati*. Il gruppo di lavoro che alimenta la base di dati opera già da anni, con competenza, su un'applicazione MS Access, mantenendo l'aggiornamento delle varie versioni su sistema operativo Windows. Si aggiunge che le dimensioni della base dati non sono tali, né

---

[41] Non si prendono in considerazione eventuali passaggi interpiattaforma che esporrebbero a possibili errori e problematiche in parte anche poco approfondite nella letteratura. Si aggiungerà poi che, in parte, è la stessa filosofia di base del sistema che ne fissa alcune limitazioni di una certa rilevanza: in particolare, l'effettiva stabilità dell'intero sistema sarebbe garantita solo da alcune particolari configurazioni relative tra linguaggio, server e database, ancora ampiamente in fase di sperimentazione. Cfr. Greenspan-Bulger 2000.

probabilmente lo diventeranno presto, da richiedere nemmeno l'utilizzo forzato di DBMS professionali di più elevate capacità (anche rispetto agli stessi prodotti Microsoft per grandi gestioni)[42].

2) *Possibilità di elaborare l'applicazione web come se fosse un'applicazione stand-alone*, eliminando le problematiche relative alla trasmissione su Internet, con particolare riferimento alla visualizzazione dell'interfaccia, di fatto nativamente multipiattaforma e pienamente operativa con qualsiasi tipo di sistema di elaboratore con connessione alla Rete e browser.
3) *Salvaguardia dell'archivio e dell'applicazione sotto il profilo della sicurezza e dei diritti di copyright*. Il codice realizzato è interamente compilato all'interno del server, almeno per quanto riguarda la logica dell'applicazione; ciò impedisce la visualizzazione dei processi di esecuzione e l'accesso non protetto ai dati sensibili dell'archivio.

Svantaggi considerati: l'integrazione più o meno forzata nel mondo Windows, la richiesta di elaboratori con potenze di calcolo elevate (comunque alla portata degli attuali standard) e possibili lievitazioni dei costi (attualmente pressoché nulli) in occasione di un eventuale passaggio alle soluzioni top-level Microsoft.

In ogni caso, per maggiore chiarezza, passiamo in rassegna brevemente le caratteristiche generali dei principali ambienti di sviluppo per creare applicazioni dinamiche su Internet.

3.6.1 ASP – caratteristiche

---

[42] Il DBMS professionale Microsoft utilizzato normalmente per le applicazioni Internet che richiedono notevoli volumi di elaborazione è Microsoft SQL Server 2000. Nel caso in questione si è scelto di non procedere comunque al passaggio da Access a SQL Server, in quanto il volume di traffico stimato (10 utenti concorrenti in lettura/scrittura) e la grandezza del database (15 MegaByte circa) non danno luogo a preoccupazioni di sorta riguardo al corretto funzionamento della base di dati su Internet. Cfr. Sinclair 2000, pp. 31-40.

- Tecnologia per la creazione di pagine web dinamiche (gratuita).
- Supporto del modello client-server con prevalenza del lato server.
- Uso dei molti linguaggi di scripting (non compilati) per la programmazione del back-end (lato server) integrati dagli oggetti specifici ASP (gratuiti).
- Integrazione con qualsiasi database.
- Pieno supporto di tutto il software Microsoft e anche di altre software-house.
- Server Microsoft Windows NT/2000 (a pagamento)[43].

3.6.2 ASP.NET – caratteristiche

- Tecnologia per la creazione di pagine web dinamiche (gratuita).
- Supporto del modello client-server con prevalenza del lato-server.
- Uso di più di venti diversi linguaggi di programmazione (compilati) per la realizzazione del back-end (lato server) integrati dagli oggetti specifici ASP.NET (gratuiti).
- Integrazione con qualsiasi database.
- Pieno supporto di tutto il software Microsoft e anche di altre software-house.
- E' completamente indipendente da Microsoft ASP e fa parte del progetto globale Microsoft di rivoluzione della Rete denominato .NET.
- Server Microsoft Windows NT/2000 (a pagamento)[44].

3.6.3 CFM – caratteristiche

- Tecnologia per la creazione di pagine web dinamiche (sviluppata dalla software-house Macromedia) (a pagamento).

---

[43] Cfr. Mitchell-Atkinsons 2000, p. 4 e ss.

[44] Cfr. Payne 2001, pp. 3-23.

- Pacchetto integrato dotato di tutte le funzioni per gestire il server web e la programmazione lato server tramite un linguaggio proprietario a marcatori (*tag*) chiamato CFML.
- Integrazione con qualsiasi database[45].

3.6.4 PHP – caratteristiche

- Tecnologia di sviluppo di pagine dinamiche *Open Source* basata su un linguaggio di scripting lato server ottimamente integrato con altri strumenti *Open Source* come server web (Apache) e database (MySql) (tutti gratuiti).
- Pienamente integrato con i sistemi Unix-Linux. Compatibile con sistemi Windows e anche MacOSX[46].

3.6.5 JSP – caratteristiche

- Piattaforma Java di sviluppo di contenuti dinamici, programmabile con il linguaggio JSP e tramite blocchi di codice Java chiamati Servlet (gratuita).
- Basato strettamente sul linguaggio C++.
- Tecnologia multipiattaforma (teoricamente può funzionare su qualsiasi sistema operativo-utente a finestre).
- Integrazione con qualsiasi database.
- Server web Tomcat, non compatibile con Windows[47].

---

[45] Cfr. Uboldi 2003, pp. 18-21.

[46] Cfr. Meloni 2000, pp. ix-xii; Greenspan-Bulger 2000, pp. xxv-xxxii; Mannelli 2003, pp. 7-10; Pasqua 2002, pp. 12-15; Pasqua 2003, pp. 12-14; Perri 2002, pp. 21-23.

[47] Cfr. Pelliccia 2002, pp. 68-71; Uboldi 2002, pp. 24-27.

3.7 Struttura-tipo di un'applicazione ASP.NET

Un'applicazione ASP.NET consta di una serie di pagine di codice residenti sul server web e variamente legate fra di loro.
Le pagine sono in tutto e per tutto dei comuni documenti di testo tranne per l'estensione ".aspx" che correda il nome di ogni file[48].
Il codice contenuto in ogni documento è rigidamente separato in due parti.
Una parte contiene il codice della logica applicativa (AL), scritto in uno dei molteplici linguaggi consentiti dal sistema (Visual Basic .Net, C#, JScript, etc); tale parte viene messa in esecuzione, dopo essere stata compilata, *solo sul server*.
Il risultato dell'elaborazione viene integrato, poi, nella seconda parte del codice, scritta in linguaggio html con moduli tipici ASP.NET, per "montare" la pagina definitiva da inviare, tramite web, all'utente che ne ha fatto richiesta e ha attivato tutto il meccanismo[49].

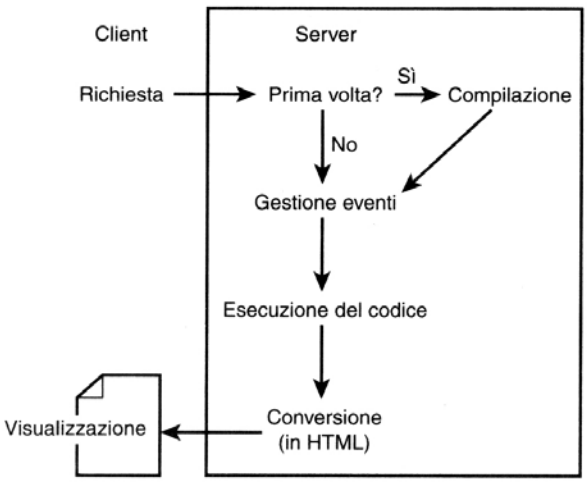


Figura 16 – Flusso di esecuzione di una pagina ASP.NET, dalla richiesta alla visualizzazione (fonte: Payne 2001)

---

[48] Vedi listati codice, reperibili all'interno del cd-rom nella cartella "listati". Sono convertiti in file di testo ".txt" per permetterne la lettura con qualsiasi editor di testo.
[49] Cfr. Payne 2001, pp. 3-11 e pp. 33-34.

In alcuni casi particolari è possibile inserire codice eseguibile anche all'interno della pagina html inviata. Esso verrà eseguito *solo all'interno del computer remoto*.

A questo punto il documento HTML, corredato di tutti i suoi elementi, viene visualizzato nel browser dell'utente il quale non ha modo di avvedersi del meccanismo di realizzazione della pagina e può sperimentarne il solo risultato esteriore.

Questo meccanismo di base si presta a molteplici applicazioni. Tipicamente viene usato per gestire i contenuti dinamici delle pagine in cui il risultato visualizzato non può essere fissato stabilmente nel tempo, in quanto frutto di elaborazioni in tempo reale di una quantità di dati indefinibile a priori.

L'intero meccanismo di gestione è reso funzionale dall'infrastruttura di Rete e dalla programmazione del server web.

Tutte le applicazioni ASP.NET operano all'interno del *NET Framework*, la piattaforma su cui poggia tutta la tecnologia NET, un insieme di oggetti e specifiche di Microsoft per la creazione di applicazioni.

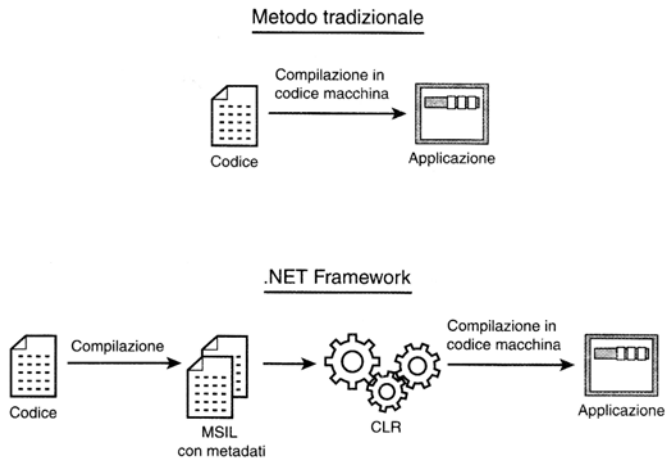


Figura 17 – Differenza tra il framework delle applicazioni tradizionali e .NET Framework (fonte: Payne 2001)

Il codice contenuto nelle pagine viene eseguito all'interno dell'ambiente CRL (*Common Language Runtime*) del Framework, attraverso la compilazione del codice originale in un linguaggio intermedio MSIL (*Microsoft Intermediate Language*), formato da metadati. Grazie a questa innovazione, all'interno del Framework è possibile utilizzare praticamente qualsiasi linguaggio di programmazione in quanto la conversione finale, in linguaggio macchina, è operata esclusivamente dal CLR di sistema, permettendo allo sviluppatore di muoversi con la massima libertà[50].

4 L'APPLICAZIONE WEB BEDT

4.1 Architettura di sistema

Il funzionamento della pagina web dinamica ASP.NET si basa sull'uso concorrente di una serie di tecnologie integrate e specializzate.
In concreto, per l'elaborazione e la verifica del codice lato server:

- Web server: software che gestisce il computer remoto che elabora e invia i dati attraverso la rete
- .NET Framework: ambiente di elaborazione del codice sorgente
- ASP.NET: tecnologia di creazione di pagine dinamiche
- Visual Basic .NET: linguaggio di programmazione lato server
- ADO.NET: insieme di oggetti e classi per comunicazione con database
- DBMS: database

[50] Cfr. Barbieri 2002, pp. 10-14; MacKenzie-Sharkey 2002, pp. 183-200; vedi anche il sito ufficiale Microsoft per documentazione .NET oppure i materiali disponibili dopo l'installazione del pacchetto .NET Framework.

e, per l'elaborazione e la verifica del codice lato client:

- HTML: linguaggio di formattazione di pagine web
- DynamicHTML: linguaggio di programmazione di pagine web con estensioni dinamiche
- JavaScript: linguaggio di programmazione scripting per la gestione di automatismi lato client (menu a tendine)

Tra i due elementi deve esistere lo strato di comunicazione fornito dalla rete Internet con il suo protocollo di comunicazione HTTP. Al fine di verificare il funzionamento in locale dell'applicazione (cioè senza avere a disposizione Internet), è necessario configurare un elaboratore per fornire una simulazione di tutti gli elementi cooperanti, come all'interno di una rete geografica[51].

4.1.1 Requisiti minimi di sistema per la visualizzazione

Per la consultazione in locale dell'applicazione sono necessari i seguenti elementi:

- Sistema operativo Windows 2000 o XP/XP Pro
- NET Framework ver. 1.0.3705.0[52]
- ASP.NET Premium 1.0 (estensione non obbligatoria)
- Server web Internet Information Service IIS 5.0 oppure
- Web Matrix 0.5, tool di sviluppo di pagine web con server integrato[53]
- Internet Explorer 5.0, browser web

Per la consultazione tramite Internet:

- Internet Explorer 5.0, browser web (la versione attuale di BEdT online è testata solo su browser IE)

---

[51] Vedi il file *istruzioni.htm* nel cd-rom allegato.

[52] Disponibile per l'installazione nel cd-rom allegato.

[53] Disponibile per l'installazione nel cd-rom allegato.

4.1.2 Software applicativo utilizzato per la generazione del codice

- sistema operativo Windows XP Pro[54]
- tool di sviluppo pagine ASP.NET: Microsoft Web Matrix Project ver. 0.5[55]
- ambiente di elaborazione: NET Framework ver. 1.0.3705.0
- applicativo database: Microsoft Access 2000
- gestore del menu: Pop Menu Creator Pro, Anoxy Software

4.2 Struttura dell'applicazione

4.2.1 Impostazione dell'interfaccia a menu

Una base di dati può essere interpretata come un modello di realtà: l'individuazione di una tipologia di interfaccia per rappresentare tale realtà è una scelta condizionata da un lato dalla realtà medesima (l'insieme dei dati) e dall'altro dai problemi connessi con la gestione tecnica.

> Come modello concettuale dei dati si usa il cosiddetto modello Entità/Relazioni secondo il quale la realtà è costituita da entità e da relazioni fra queste, rappresentabili in un grafo (…).
> Per entità si intende generalmente un qualsiasi elemento della realtà che può essere individuato: un libro, un autore (…).
> Un insieme di entità è il concetto astratto associato all'insieme dei libri, degli autori e così via (…).

---

[54] Cfr. Davis-Lewis 2000 e Mueller 2002, per i problemi derivanti dall'utilizzo del server Internet Information Service 5.0, necessario per testare le pagine in modalità operativa.

[55] Cfr. Cangiano 2002, pp. 16-20.

Il collegamento tra le entità avviene attraverso relazioni, nel senso matematico del termine; sul piano semantico, la relazione si può esprimere attraverso un gruppo verbale[56]:

nel nostro caso:

- un autore *scrive* un testo,
- il testo *appartiene* ad un genere… *è conservato* in un dato manoscritto, e così via…

In sostanza, il modello concettuale che è stato implementato per la costruzione della base dati viene riutilizzato anche nella struttura dell'interfaccia a menu, con l'intenzione esplicita di riproporre il modello di realtà generato dall'analisi originale dei dati:

- *un autore* – quale autore?
- *ha scritto qualcosa* – che cosa?

oppure

- *un testo* – quale testo?
- *chi lo ha scritto*? – attribuzione.
- *dove è conservato*? – canzonieri.

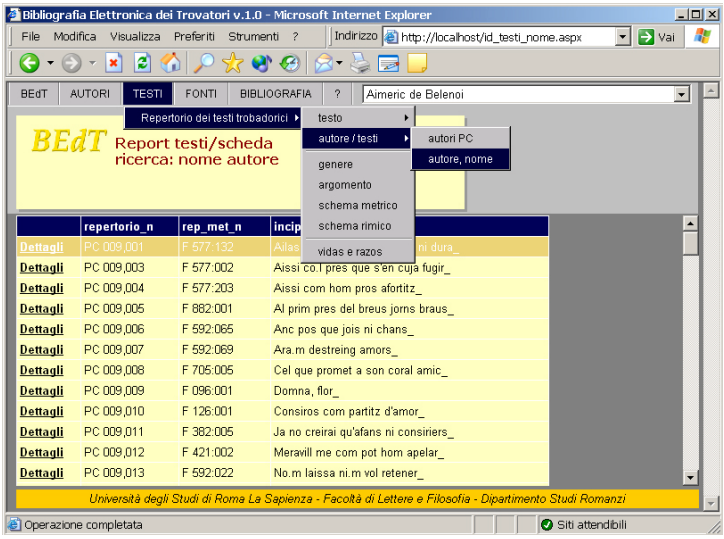


Figura 18 – Esempio di applicazione del modello all'interfaccia dell'applicazione

[56] Cfr. Fadini-Savy 1999, pp. 131-132; vedi anche Naccarato 1999, pp. 10-14.

4.2.2 Descrizione dei comandi

In base a queste considerazioni è stata implementata una struttura dell'applicazione ad albero, in cui a partire dalle scelte fondamentali di ambiente (AUTORI, TESTI, FONTI, BIBLIOGRAFIA) si snodano alcune possibili direzioni di ricerca. Fino ad arrivare all'elemento caratterizzante, ovvero l'elemento di filtro da applicare per ottenere il risultato in forma di scheda o di rapporto.
In sintesi la struttura è così concepita:

- **AUTORI**
  - **Trovatori provenzali**
    - *Pillet-Carstens* (ricerca dei dati biografici relativi ad un trovatore attraverso il numero di repertorio PC - il sistema rintraccia tutti i numeri di repertorio presenti all'interno dell'archivio e realizza un *indice dinamico*[57] all'interno del quale compiere la scelta - i dati sono poi visualizzati in forma di scheda)
    - *autore, nome* (ricerca dei dati biografici relativi ad un trovatore attraverso l'indicazione del nome - stessa modalità del precedente)
    - *luogo di origine* (rapporto delle schede di tutti i trovatori associabili tramite il luogo di origine, più o meno ipotetico)
    - *luogo di attività* (rapporto delle schede di tutti i trovatori associabili tramite il luogo di attività, più o meno ipotetico)
    - *autori / dettagli* (elenco sistematico di tutti i trovatori censiti, organizzato in ordine alfabetico con rimandi alle schede personali e dettagliate di ciascuno)

[57] Modalità da ritenersi valida per tutti gli indici dinamici.

- **TESTI**
  - **Repertorio dei testi trobadorici**
    - **testo**
      - *PC* (ricerca dei dati relativi a un testo, attraverso il numero completo di repertorio PC – il risultato è la scheda descrittiva del testo corredata di alcuni link interni a specifiche sezioni quali: *identificazione testo, genere lirico, metrica e musica, luogo, data e occasione, personaggi e luoghi, tradizione manoscritta, bibliografia*)[58]
      - *Frank* (ricerca dei dati relativi a un testo, attraverso il numero di repertorio Frank – stessa modalità del precedente)
      - *incipit* (ricerca dei dati relativi a un testo, attraverso l'indice dinamico degli incipit, ricostruito a partire dalle informazioni d'archivio – stessa modalità del precedente)
    - **autore / testi**
      - *autori PC* (rapporto di tutti i testi attribuiti a un trovatore, indicato tramite il numero di repertorio PC, con possibilità di visualizzazione della scheda relativa a ogni testo)
      - *autore, nome* (rapporto di tutti i testi attribuiti a un trovatore, indicato tramite il nome standard, con possibilità di visualizzazione della scheda relativa a ogni testo)
    - *genere* (rapporto, con possibilità di dettaglio, di tutti i testi associati attraverso l'indice dei generi lirici, ricostruito dinamicamente sulla base dei dati d'archivio)
    - *argomento* (rapporto, con possibilità di dettaglio, di tutti i testi associati attraverso l'indice dinamico degli argomenti)

[58] N.B.: valido per tutte le schede descrittive.

        - *schema metrico* (rapporto, con possibilità di dettaglio, di tutti i testi associati attraverso l'indice dinamico degli schemi metrici)
        - *schema rimico* (rapporto, con possibilità di dettaglio, di tutti i testi associati attraverso l'indice dinamico degli schemi rimici)
        - *vidas e razos* (rapporto, con possibilità di dettaglio, di tutte le vidas e razos attribuite a ogni trovatore, selezionato tramite l'indice dinamico dei nomi standard)

- **FONTI**
    - **Canzonieri**
        - *segnatura* (indice, con possibilità di dettaglio, di tutti i componimenti presenti in un canzoniere, selezionato tramite l'indice dinamico delle segnature)
        - *sigla* (indice, con possibilità di dettaglio, di tutti i componimenti presenti in un canzoniere, selezionato tramite l'indice dinamico delle sigle standard)
        - *rubrica* (rapporto, con possibilità di dettaglio, dei testi presenti nei canzonieri, a partire dall'indice dinamico delle voci di rubrica)
        - *testo PC / canzonieri* (rapporto, con possibilità di dettaglio, di tutti i testimoni di un testo, selezionato tramite l'indice dinamico dei numeri di repertorio PC)
        - *incipit / canzonieri* (rapporto, con possibilità di dettaglio, di tutti i testimoni di un testo, selezionato tramite l'indice dinamico degli incipit)
    - **Tavole antiche**
        - *segnatura* (indice, con possibilità di dettaglio, di tutti i componimenti presenti in una tavola, selezionata tramite l'indice dinamico delle segnature)
        - *sigla* (indice, con possibilità di dettaglio, di tutti i componimenti presenti in una tavola, selezionata tramite l'indice dinamico delle sigle standard)

        - *rubrica* (rapporto, con possibilità di dettaglio, dei testi presenti in tavole a partire dall'indice dinamico delle voci di rubrica)
        - *testo PC / tavole* (rapporto, con possibilità di dettaglio, di tutte le tavole testimoni di un testo, selezionato tramite l'indice dinamico dei numeri di repertorio PC)
        - *incipit / tavole* (rapporto, con possibilità di dettaglio, di tutte le tavole testimoni di un testo, selezionato tramite l'indice dinamico degli incipit)

- **BIBLIOGRAFIA** (schede relative alle risorse bibliografiche)
    - *chiave* (ricerca della scheda descrittiva di un testo critico attraverso l'indicazione della chiave standard di riconoscimento: es. Roncaglia 1949)
    - *autore* (rapporto, con possibilità di dettaglio, di tutti i testi critici censiti, con possibilità di dettaglio, attraverso l'indicazione del nome e cognome dell'autore)
    - *monografia* (ricerca della scheda descrittiva di un testo critico tramite l'indice dinamico dei titoli monografici)
    - *articolo* (ricerca della scheda descrittiva di un testo critico tramite l'indice dinamico dei titoli degli articoli)
    - *miscellanea* (rapporto, con possibilità di dettaglio, dei testi critici inseriti in raccolte tramite l'indice dinamico delle miscellanee)
    - *rivista* (rapporto, con possibilità di dettaglio, dei testi critici inseriti in raccolte tramite l'indice dinamico delle riviste)
    - *risorse online* (non ancora attivato)

4.3 Dettagli di programmazione di alcune pagine-tipo

4.3.1 File: indice_pc.aspx

La pagina presenta un menu a discesa con l'indice di tutti gli autori compresi nell'archivio tramite l'indice dinamico del numero di repertorio Pillet-Carstens. Una volta indicata una scelta, la pagina si autoaggiorna mostrando la scheda bio-bibliografica relativa all'autore selezionato con i dati inseriti nel DB.

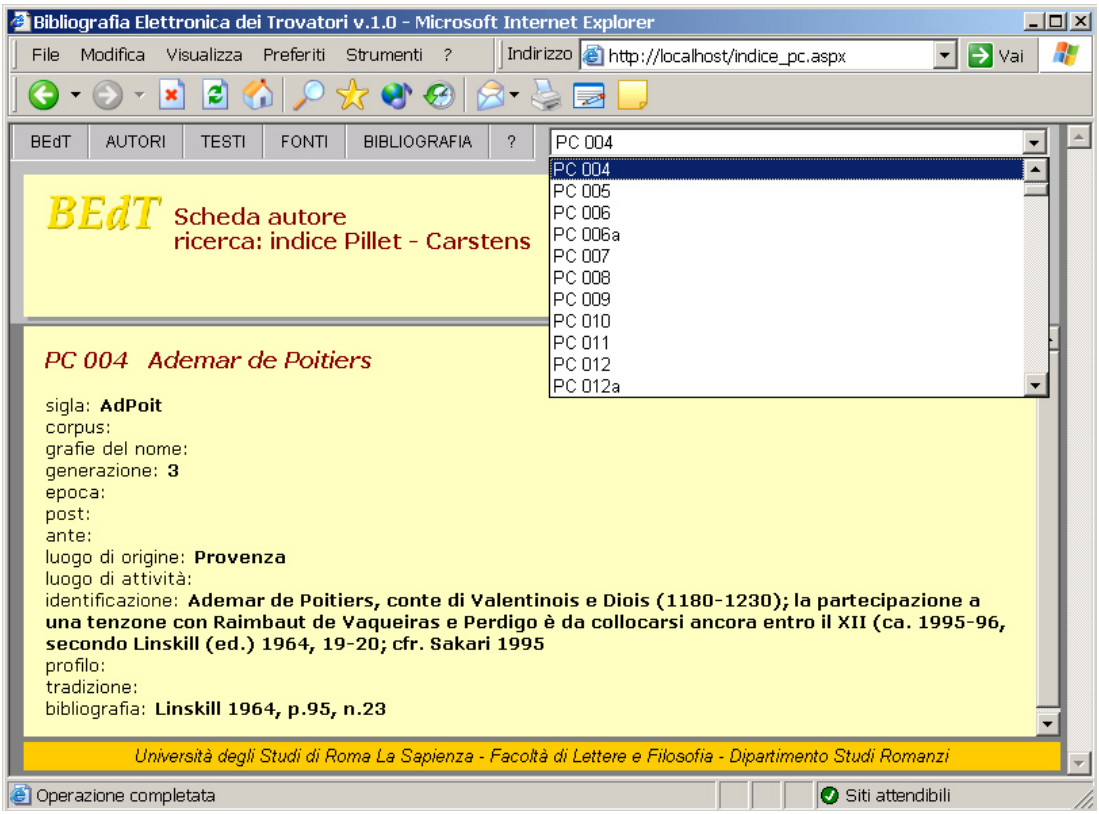


Figura 19 – La pagina indice_pc.aspx

Descrizione del codice:

Intestazione della pagina aspx, informa il server del tipo di linguaggio utilizzato e prepara una serie di elementi che saranno richiamati nel corso dell'elaborazione al completamento dell'aspetto grafico della pagina:

```
<%@ Page Language="VB" Debug="true" %>
<%@ Register TagPrefix="BEdTASP" TagName="title_tag" Src="title_tag.ascx" %>
<%@ Register TagPrefix="BEdTASP" TagName="script_menu" Src="script_menu.ascx" %>
<%@ Register TagPrefix="BEdTASP" TagName="body_tag" Src="body_tag.ascx" %>
<%@ Register TagPrefix="BEdTASP" TagName="menu_bar" Src="menu_bar.ascx" %>
<%@ Register TagPrefix="BEdTASP" TagName="piedepagina" Src="piedepagina.ascx" %>
```

importa i nomi di spazio necessari per l'interfacciamento al DB

```
<%@ import Namespace="System.Data" %>
<%@ import Namespace="System.Data.OleDb" %>
```

inizio del codice con l'indicazione dell'elaborazione lato server

```
<script runat="server">
```

subroutine che viene eseguita al caricamento della pagina e a ogni successivo refresh del sistema

```
  Sub Page_Load(Sender As Object, E As EventArgs)
        If Not Page.IsPostBack Then
```

dimensiona la variabile per la connessione al DB per rintracciare i dati da una specifica tabella

```
      Dim ConnectionString As String = "server=(local);database=pubs;trusted_connection=true"
      Dim CommandText As String = "select repertorio_n from TROVATORI_PROVENZALI"[59]
      Dim myConnection as new OleDbConnection _
       ("Provider=Microsoft.Jet.OLEDB.4.0;" & _
       "Data Source=C:\Inetpub\wwwroot\BEdT\database\BEdT2k01.mdb")
      Dim myCommand As New OleDbCommand(CommandText, myConnection)
         DropDownList1.DataTextField = "repertorio_n"
      myConnection.Open()
```

carica i dati rintracciati (tutti i numeri di repertorio PC) in un menu dinamico a discesa

```
     DropDownList1.DataSource =
myCommand.ExecuteReader(CommandBehavior.CloseConnection)
      DropDownList1.DataBind()
      DropDownList1.Items.Insert(0, "-- Seleziona un trovatore --")
      End If
   End Sub
```

[59] Istruzione SQL. Per le specifiche sul comando SELECT, vedi Kline 2000, pp. 148-166.

una volta selezionato un valore dal menu a discesa da parte dell'utente applica, in qualità di filtro, tale valore ad una seconda operazione di selezione sul DB

```
Sub ApplyFilter_Click(Sender As Object, E As EventArgs)
 Dim ConnectionString As String = "server=(local);database=pubs;trusted_connection=true"
 Dim CommandText As String
 Dim filterValue As String = DropDownList1.SelectedItem.Text.Replace("'", "''")
 CommandText = "select * from TROVATORI_PROVENZALI where repertorio_n = '" & filterValue
& "'"[60]
 Dim myConnection as new OleDbConnection _
        ("Provider=Microsoft.Jet.OLEDB.4.0;" & _
        "Data Source=C:\Inetpub\wwwroot\BEdT\database\BEdT2k01.mdb")
 Dim myCommand As New OleDbCommand(CommandText, myConnection)
 myConnection.Open()
```

carica un set di dati pronto ad essere visualizzato nella pagina web

```
    MyDataList.DataSource = myCommand.ExecuteReader(CommandBehavior.CloseConnection)
    MyDataList.DataBind()
 End Sub
```

fine dello script lato server

```
</script>
```

inizio dello script lato client[61]

```
<html>
<head>
```

carica il titolo della pagina, un blocco di codice JavaScript, le caratteristiche generali del corpo di pagina e il menu di comando prelevandoli da altri file sul server, in modo che qualsiasi modifica apportata ad essi si rifletta su tutte la pagine visualizzate

```
<BEdTASP:title_tag id="UserControl1" runat="server"></BEdTASP:title_tag>
<BEdTASP:script_menu id="UserControl2" runat="server"></BEdTASP:script_menu>
```

---

[60] Istruzione SQL (vedi nota 59).

[61] Questa sezione del codice viene eseguita sul computer dell'utente. Si tratta di operazioni che riguardano solo ed esclusivamente le modalità di visualizzazione della pagina e in alcun modo riguardano la logica dell'applicazione che rimane invisibile all'utente stesso.

```
</head>
<BEdTASP:body_tag id="UserControl3" runat="server"></BEdTASP:body_tag>
<BEdTASP:menu_bar id="UserControl4" runat="server"></BEdTASP:menu_bar>
```

carica gli elementi html necessari per visualizzare il menu opzioni utilizzato come filtro e li aggancia logicamente alla subroutine lato server

```
<form runat="server">
   <!-- LIVELLO SELECT -->
   <div id="select" style="Z-INDEX: 3; LEFT: 398px; WIDTH: 211px; POSITION: absolute; TOP: 2px;
HEIGHT: 25px">
   <asp:DropDownList id="DropDownList1" runat="server" width="370" AutoPostBack="true"
OnSelectedIndexChanged="ApplyFilter_Click"></asp:DropDownList>
     </div>
   <!-- FINE LIVELLO SELECT -->
```

apre una finestra con funzioni di scrolling per inserire testi esplicativi

```
   <!-- LIVELLO ISTRUZIONI -->
   <div id="spazioistr" style="Z-INDEX: 1; WIDTH: 483px; POSITION: absolute; TOP: 50px; HEIGHT:
90px">
      <div id="bordino" style="BORDER-RIGHT: #000000 1px; BORDER-TOP: #000000 1px; Z-INDEX:
2; LEFT: 3px; BORDER-LEFT: #000000 1px; WIDTH: 503px; BORDER-BOTTOM: #000000 1px;
POSITION: absolute; TOP: -10px; HEIGHT: 105px; BACKGROUND-COLOR: #999999; layer-
background-color: #999999">
      </div>
      <div id="testoistr" style="Z-INDEX: 5; LEFT: 7px; OVERFLOW: auto; WIDTH: 493px; POSITION:
absolute; TOP: -9px; HEIGHT: 97px">
         <div align="left"><img hspace="2" src="images/logo_bedt.jpg" width="100" align="left"
vspace="2" /><font face="Arial, Helvetica, sans-serif" size="2">
         <div style="font:12pt  verdana;color:darkred"><br><b>Scheda autore<br>ricerca:
nome</b></div>
         Spazio
            riservato alle istruzioni sul funzionamento della pagina, Spazio riservato alle istruzioni
            sul funzionamento della pagina, Spazio riservato alle istruzioni sul funzionamento
            della pagina, Spazio riservato alle istruzioni sul funzionamento della pagina,Spazio
            riservato alle istruzioni sul funzionamento della pagina alle istruzioni sul funzionamento
            della pagina, Spazio riservato alle istruzioni sul funzionamento della pagina, Spazio
            riservato alle istruzioni sul funzionamento della pagina, Spazio riservato alle istruzioni
            sul funzionamento della pagina,Spazio riservato alle istruzioni sul funzionamento
            della pagina</font>
         </div>
      </div>
```

```
    <br />
    <br />
    <div id="bianco" style="BORDER-RIGHT: 3px; BORDER-TOP: 3px; Z-INDEX: 4; LEFT: 0px;
BORDER-LEFT: 3px; WIDTH: 503px; BORDER-BOTTOM: 3px; POSITION: absolute; TOP: -13px;
HEIGHT: 105px; BACKGROUND-COLOR: #ffffc1; layer-background-color: #FFFFC1">
    </div>
  </div>
  <!-- FINE LIVELLO ISTRUZIONI -->
```

carica una tabella dinamica costituita da un modello grafico cui vengono associati di volta in volta i dati estratti dall'archivio posizionato in memoria dopo la selezione

```
 <!-- LIVELLO SCHEDE -->
  <div id="corpo" style="Z-INDEX: 2; LEFT: 10px; OVERFLOW: auto; WIDTH: 766px; POSITION:
absolute; TOP: 149px; HEIGHT: 303px">
    <ASP:DataList id="MyDataList" runat="server" RepeatDirection="Vertical" bgcolor="#FFFFC1"
RepeatColumns="1">
      <ItemTemplate>
        <div style="padding:15,15,15,15; font-size:10 pt;font-family:Verdana">
          <div style="font:12pt  verdana;color:darkred">
            <i><b>
<%#  DataBinder.Eval(Container.DataItem,  "rep_n_aut")%>     <%#
DataBinder.Eval(Container.DataItem, "autore")%> </b></i> <i><b></b></i>
          </div>
          <br />
```

esempio di collegamento dati: campo visualizzato "sigla", fonte dati "sigla"

```
          <b>sigla: </b> <%# DataBinder.Eval(Container.DataItem, "sigla")%>



          <br />
          <b>corpus: </b> <%# DataBinder.Eval(Container.DataItem, "corpus")%>
          <br />
          <b>grafie del nome: </b> <%# DataBinder.Eval(Container.DataItem,
"grafie_del_nome")%>
          <br />
          <b>generazione: </b> <%# DataBinder.Eval(Container.DataItem, "generazione")%>
          <br />
          <b>epoca: </b> <%# DataBinder.Eval(Container.DataItem, "epoca")%>
          <br />
          <b>post: </b> <%# DataBinder.Eval(Container.DataItem, "post")%>
          <br />
          <b>ante: </b> <%# DataBinder.Eval(Container.DataItem, "ante")%>
```

```
                <br />
                <b>luogo di origine: </b> <%# DataBinder.Eval(Container.DataItem,
"luogo_di_origine")%>
                <br />
                <b>luogo di attivit&agrave;: </b> <%# DataBinder.Eval(Container.DataItem,
"luogo_di_attività")%>
                <br />
                <b>identificazione: </b> <%# DataBinder.Eval(Container.DataItem, "identificazione")%>
                <br />
                <b>profilo: </b> <%# DataBinder.Eval(Container.DataItem, "profilo")%>
                <br />
                <b>tradizione: </b> <%# DataBinder.Eval(Container.DataItem, "tradizione")%>
                <br />
                <b>bibliografia: </b> <%# DataBinder.Eval(Container.DataItem, "bibliografia")%>
                <br />
            </div>
        </ItemTemplate>
    </ASP:DataList>
  </div>
   </form>
  <!-- FINE LIVELLO SCHEDE -->
```

carica il file che contiene il testo del piè di pagina

```
<BEdTASP:piedepagina id="UserControl5" runat="server"></BEdTASP:piedepagina>
```

fine del file di codice

```
</body>
</html>
```

4.3.2 File: ind_canz_pc.aspx

La pagina presenta un menu a discesa con l'indice di tutti i testi compresi nel repertorio, visualizzati per numero di PC, completo di indicazione del testo: es. "PC 001,001". Una volta indicata una scelta, la pagina si autoaggiorna mostrando una tabella con l'indice di tutte le occorrenze del testo selezionato all'interno degli indici di tutti i canzonieri censiti.

La tabella fornisce alcune informazioni principali quali: *segnatura dei canzonieri*, *sigle dei canzonieri*, informazioni sulla *vecchia numerazione*, nome segnato in *rubrica*, *incipit* del testo, e *posizione* del testo nel canzoniere.

All'interno della tabella è presente un collegamento attivo che innesca un'ulteriore selezione su un testo specifico, provocando come risultato la visualizzazione della scheda completa del testo, arricchita di una serie di rimandi interni alle sezioni della scheda (es. *identificazione testo*, *decorazione*, *impaginazione*, *metrica e musica*, *marginalia e glosse*).

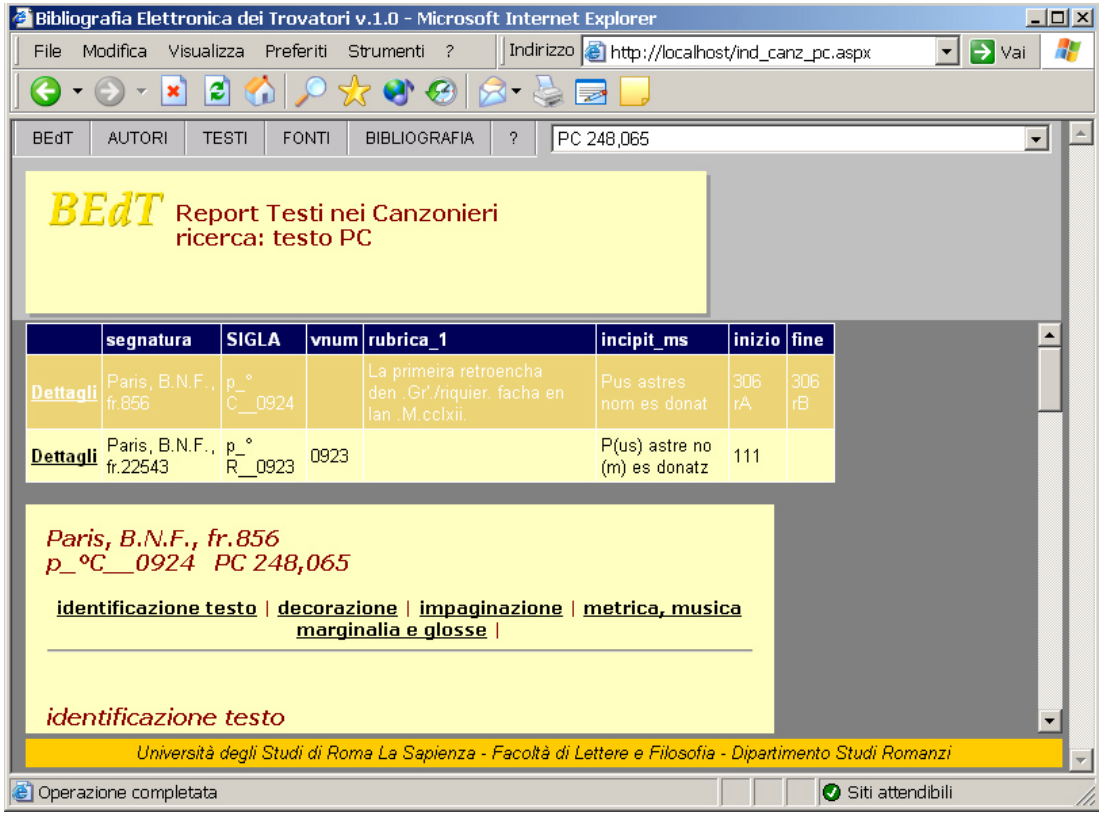


Figura 20 – La pagina ind_canz_pc.aspx

Descrizione del codice:

Intestazione della pagina aspx, informa il server del tipo di linguaggio utilizzato e prepara una serie di elementi che saranno richiamati nel corso dell'elaborazione:

```
<%@ Page Language="VB" Debug="true" %>
<%@ Register TagPrefix="BEdTASP" TagName="title_tag" Src="title_tag.ascx" %>
<%@ Register TagPrefix="BEdTASP" TagName="script_menu" Src="script_menu.ascx" %>
<%@ Register TagPrefix="BEdTASP" TagName="body_tag" Src="body_tag.ascx" %>
<%@ Register TagPrefix="BEdTASP" TagName="menu_bar" Src="menu_bar.ascx" %>
<%@ Register TagPrefix="BEdTASP" TagName="piedepagina" Src="piedepagina.ascx" %>
```

importa i nomi di spazio necessari per l'interfacciamento al DB

```
<%@ import Namespace="System.Data" %>
<%@ import Namespace="System.Data.OleDb" %>
```

inizio del codice con l'indicazione dell'elaborazione lato server

```
<script runat="server">
```

subroutine che viene eseguita al caricamento della pagina e a ogni successivo refresh del sistema

```
  Sub Page_Load(Sender As Object, E As EventArgs)
        If Not Page.IsPostBack Then
```

dimensiona la variabile per la connessione al DB per rintracciare i dati da una specifica tabella (in particolare seleziona tutte le prime occorrenze dei numeri di repertorio e crea un nuovo indice dinamico di numeri *unici*)

```
Dim ConnectionString As String = "server=(local);database=pubs;trusted_connection=true"
      Dim CommandText As String = "select distinct repertorio_n from INDICI_CANZONIERI"[62]
        Dim myConnection as new OleDbConnection _
       ("Provider=Microsoft.Jet.OLEDB.4.0;" & _
       "Data Source=C:\Inetpub\wwwroot\BEdT\database\BEdT2k01.mdb")
      Dim myCommand As New OleDbCommand(CommandText, myConnection)
      ' TODO: Update the DataTextField value
      DropDownList1.DataTextField = "repertorio_n"
        myConnection.Open()
```

[62] Istruzione SQL (vedi nota 59).

carica i dati rintracciati in un menu dinamico a discesa

```
DropDownList1.DataSource = myCommand.ExecuteReader(CommandBehavior.CloseConnection)
        DropDownList1.DataBind()
        DropDownList1.Items.Insert(0, "-- Indice Pillet - Carstens --")
      End If
     End Sub
```

crea una subroutine per impostare la tabella automatica e la successiva in cascata

```
Sub Pagina(Sender As Object, E As EventArgs)
          MasterGrid.SelectedIndex = 0
        BindMasterGrid()
        BindDetailGrid()
      End Sub
      Sub MasterGrid_Select(Sender As Object, E As EventArgs)
      BindDetailGrid()
   End Sub
      Sub MasterGrid_Page(Sender As Object, E As DataGridPageChangedEventArgs)
        If MasterGrid.SelectedIndex <> -1 Then
          MasterGrid.SelectedIndex = -1
        BindDetailGrid()
        End If

      MasterGrid.CurrentPageIndex = e.NewPageIndex
      BindMasterGrid()
      End Sub
      Sub BindMasterGrid()
          Dim ConnectionString As String = "server=(local);database=pubs;trusted_connection=true"
```

una volta selezionato un valore dal menu a discesa da parte dell'utente applica, in qualità di filtro, tale valore ad una seconda operazione di selezione sul DB e la aggancia alla tabella

```
  Dim filtro As string = DropDownList1.SelectedItem.Text.Replace("'", "''")
      Dim CommandText As String = "select segnatura, SIGLA, vecchia_numerazione as vnum,
rubrica_1, incipit_ms, inizio, fine from INDICI_CANZONIERI where repertorio_n = '" & filtro &
"'"[63]

        Dim myConnection as new OleDbConnection _
        ("Provider=Microsoft.Jet.OLEDB.4.0;" & _
```

[63] Istruzione SQL (vedi nota 59).

```
        "Data Source=C:\Inetpub\wwwroot\BEdT\database\BEdT2k01.mdb")
      Dim myCommand As New OleDbDataAdapter(CommandText, myConnection)
      Dim ds As New DataSet()
      myCommand.Fill(ds)
      MasterGrid.DataSource = ds
      MasterGrid.DataBind()
    End Sub
```

preleva il valore di filtro sulla tabella automaticamente e lo invia ad una successiva selezione sul DB. I dati così rintracciati verranno inviati alla scheda per la visualizzazione

```
Sub BindDetailGrid()
      If MasterGrid.SelectedIndex <> -1 Then
          Dim ConnectionString As String = "server=(local);database=pubs;trusted_connection=true"
         Dim filterValue As String = CStr(MasterGrid.DataKeys(MasterGrid.SelectedIndex)).Replace("",
"""")
         Dim CommandText As String = "select * from INDICI_CANZONIERI where sigla = '" &
filterValue & "'"[64]
         Dim myConnection As New OleDbConnection _
         ("Provider=Microsoft.Jet.OLEDB.4.0;" & _
         "Data Source=C:\Inetpub\wwwroot\BEdT\database\BEdT2k01.mdb")
         Dim myCommand As New OleDbCommand(CommandText, myConnection)
           myConnection.Open()
          MyDataList.DataSource =
myCommand.ExecuteReader(CommandBehavior.CloseConnection)
        End If
         MyDataList.DataBind()
       End Sub
```

fine dello script lato server

```
</script>
```

inizio dello script lato client

```
<html>
<head>
```

carica il titolo della pagina, un blocco di codice JavaScript, le caratteristiche generali del corpo di pagina e il menu di comando prelevandoli da altri file sul

[64] Istruzione SQL (vedi nota 59).

server in modo che qualsiasi modifica apportata ad essi si rifletta su tutte la pagine visualizzate

```
<BEdTASP:title_tag id="UserControl1" runat="server"></BEdTASP:title_tag>
<BEdTASP:script_menu id="UserControl2" runat="server"></BEdTASP:script_menu>
</head>
<BEdTASP:body_tag id="UserControl3" runat="server"></BEdTASP:body_tag>
<BEdTASP:menu_bar id="UserControl4" runat="server"></BEdTASP:menu_bar>
```

carica gli elementi html necessari per visualizzare il menu opzioni utilizzato come filtro e li aggancia logicamente alla subroutine lato server

```
<form runat="server">
   <!-- LIVELLO SELECT -->
   <div id="select" style="Z-INDEX: 3; LEFT: 398px; WIDTH: 211px; POSITION: absolute; TOP: 2px;
HEIGHT: 25px">
   <asp:DropDownList id="DropDownList1" runat="server" width="370" AutoPostBack="true"
OnSelectedIndexChanged="ApplyFilter_Click"></asp:DropDownList>
     </div>
   <!-- FINE LIVELLO SELECT -->
```

apre una finestra con funzioni di scrolling per inserire testi esplicativi

```
   <!-- LIVELLO ISTRUZIONI -->
   <div id="spazioistr" style="Z-INDEX: 1; WIDTH: 483px; POSITION: absolute; TOP: 50px; HEIGHT:
90px">
       <div id="bordino" style="BORDER-RIGHT: #000000 1px; BORDER-TOP: #000000 1px; Z-INDEX:
2; LEFT: 3px; BORDER-LEFT: #000000 1px; WIDTH: 503px; BORDER-BOTTOM: #000000 1px;
POSITION: absolute; TOP: -10px; HEIGHT: 105px; BACKGROUND-COLOR: #999999; layer-
background-color: #999999">
       </div>
       <div id="testoistr" style="Z-INDEX: 5; LEFT: 7px; OVERFLOW: auto; WIDTH: 493px; POSITION:
absolute; TOP: -9px; HEIGHT: 97px">
           <div align="left"><img hspace="2" src="images/logo_bedt.jpg" width="100" align="left"
vspace="2" /><font face="Arial, Helvetica, sans-serif" size="2">
           <div style="font:12pt  verdana;color:darkred"><br><b>Scheda autore<br>ricerca:
nome</b></div>
           Spazio
              riservato alle istruzioni sul funzionamento della pagina, Spazio riservato alle istruzioni
              sul funzionamento della pagina, Spazio riservato alle istruzioni sul funzionamento
              della pagina, Spazio riservato alle istruzioni sul funzionamento della pagina,Spazio
              riservato alle istruzioni sul funzionamento della pagina alle istruzioni sul funzionamento
              della pagina, Spazio riservato alle istruzioni sul funzionamento della pagina, Spazio
```

```
            riservato alle istruzioni sul funzionamento della pagina, Spazio riservato alle istruzioni
            sul funzionamento della pagina,Spazio riservato alle istruzioni sul funzionamento
            della pagina</font>
          </div>
        </div>
        <br />
        <br />
        <div id="bianco" style="BORDER-RIGHT: 3px; BORDER-TOP: 3px; Z-INDEX: 4; LEFT: 0px;
BORDER-LEFT: 3px; WIDTH: 503px; BORDER-BOTTOM: 3px; POSITION: absolute; TOP: -13px;
HEIGHT: 105px; BACKGROUND-COLOR: #ffffc1; layer-background-color: #FFFFC1">
        </div>
    </div>
    <!-- FINE LIVELLO ISTRUZIONI -->
```

carica una tabella dinamica costituita da un modello grafico cui vengono associati di volta in volta i dati estratti dall'archivio posizionato in memoria dopo la selezione

```
 <div id="corpo" style="Z-INDEX: 2; LEFT: 10px; OVERFLOW: auto; WIDTH: 766px; POSITION:
absolute; TOP: 149px; HEIGHT: 303px">
        <asp:datagrid id="MasterGrid" runat="server" DataKeyField="SIGLA"
OnSelectedIndexChanged="MasterGrid_Select" AllowPaging="False" PageSize="5"
OnPageIndexChanged="MasterGrid_Page" ForeColor="Black" BackColor="White" CellPadding="3"
GridLines="None" CellSpacing="1" width="80%">
            <HeaderStyle font-size="10pt" font-bold="True" forecolor="white"
backcolor="#000059"></HeaderStyle>
            <SelectedItemStyle forecolor="white" backcolor="#ECD577"></SelectedItemStyle>
            <PagerStyle horizontalalign="Right" backcolor="#C6C3C6" pagebuttoncount="30"
mode="NumericPages"></PagerStyle>
            <ItemStyle font-size="10pt" backcolor="#FFFFC1"></ItemStyle>
            <Columns>
              <asp:buttoncolumn text="Dettagli" commandname="Select" ItemStyle-Font-Bold="true"
ItemStyle-Font-Size="10pt"></asp:buttoncolumn>
            </Columns>
        </asp:datagrid>
        <hr color="gray" />
```

carica la scheda completa, di norma la prima della lista, altrimenti quella scelta dall'utente

```
<ASP:DataList id="MyDataList" runat="server" RepeatDirection="Vertical" bgcolor="#FFFFC1"
RepeatColumns="1">
            <ItemTemplate>
```

```
<a name="top">
  <div style="padding:15,15,15,15; font-size:10 pt;font-family:Verdana">
    <div style="font:12pt  verdana;color:darkred">
      <i><b><%# DataBinder.Eval(Container.DataItem, "segnatura")%>
      </b></i></b>
      <br />
      <i><b> <%# DataBinder.Eval(Container.DataItem, "SIGLA")%>   <%#
DataBinder.Eval(Container.DataItem, "repertorio_n")%> </b></i>
    </div>
    <div style="font:10pt  verdana;color:darkred">
      <br />
        <center>
        <b>
```

carica un menu di navigazione all'interno della scheda

```
        <a href="#id_testo">identificazione testo</a> </b>|<b>
        <a href="#id_decorazione">decorazione</a> </b>| <b>
        <a href="#id_impaginazione">impaginazione</a> </b>|<b>
        <a href="#id_metrica">metrica, musica</a>  <br />
        <a href="#id_marginalia">marginalia e glosse</a> </b>|
        </center>
    </div>
    <hr />
    <a name="id_testo">
    <br />
    <br />
    <div style="font:12pt  verdana;color:darkred"><b><i>identificazione testo</i></b>
    </div>
    <br />
    segnatura: <b><%# DataBinder.Eval(Container.DataItem, "segnatura")%></b>
    <br />
    corpus: <b><%# DataBinder.Eval(Container.DataItem, "corpus")%></b>
    <br />
    sigla: <b><%# DataBinder.Eval(Container.DataItem, "SIGLA")%></b>
    <br />
    canzoniere: <b><%# DataBinder.Eval(Container.DataItem, "canzoniere")%> </b>
    <br />
    n_:  <b><%# DataBinder.Eval(Container.DataItem, "n_")%></b>
    <br />
    vecchia numerazione:  <b><%# DataBinder.Eval(Container.DataItem,
"vecchia_numerazione")%></b>
    <br />
```

```
numerazione antica:  <b><%# DataBinder.Eval(Container.DataItem,
"numerazione_antica")%></b>
<br />
inizio:  <b><%# DataBinder.Eval(Container.DataItem, "inizio")%> </b>
<br />
fine:  <b><%# DataBinder.Eval(Container.DataItem, "fine")%> </b>
<br />
fascicolo:  <b><%# DataBinder.Eval(Container.DataItem, "fascicolo")%></b>
<br />
posizione nel fascicolo:  <b><%# DataBinder.Eval(Container.DataItem,
"posizione_nel_fascicolo")%></b>
<br>
sezione: <b><%# DataBinder.Eval(Container.DataItem, "sezione")%></b>
<br />
rubrica 1: <b><%# DataBinder.Eval(Container.DataItem, "rubrica_1")%></b>
<br />
tipo rubrica 1: <b><%# DataBinder.Eval(Container.DataItem, "tipo_rubrica_1")%></b>
<br />
copista rubrica 1: <b><%# DataBinder.Eval(Container.DataItem, "copista_rubr_1")%>
</b>
<br />
rubrica 2:  <b><%# DataBinder.Eval(Container.DataItem, "rubrica_2")%></b>
<br />
copista rubrica 2:  <b><%# DataBinder.Eval(Container.DataItem,
"copista_rubr_2")%></b>
<br />
note a rubrica:  <b><%# DataBinder.Eval(Container.DataItem, "note_a_rubrica")%></b>
<br />
rep_aut_ms:  <b><%# DataBinder.Eval(Container.DataItem, "rep_aut_ms")%> </b>
<br />
rep_2:  <b><%# DataBinder.Eval(Container.DataItem, "rep_2")%> </b>
<br />
rep aut standard:  <b><%# DataBinder.Eval(Container.DataItem,
"rep_aut_standard")%></b>
<br />
incipit ms:  <b><%# DataBinder.Eval(Container.DataItem, "incipit_ms")%></b>
<br>
repertorio n: <b><%# DataBinder.Eval(Container.DataItem, "repertorio_n")%></b>
<br />
copista del testo:  <b><%# DataBinder.Eval(Container.DataItem,
"copista_del_testo")%></b>
<br />
correzioni:  <b><%# DataBinder.Eval(Container.DataItem, "correzioni")%></b>
```

```
<br />
collazioni:  <b><%# DataBinder.Eval(Container.DataItem, "collazioni")%></b>
<br />
testo correzioni o collazioni:  <b><%# DataBinder.Eval(Container.DataItem,
"testo_correzioni_o_collazioni")%> </b>
<br />
posizione nella tradizione del testo: <b><%# DataBinder.Eval(Container.DataItem,
"posizione_nella_tradizione_del_testo")%></b>
<br />
edizione diplomatica: <b><%# DataBinder.Eval(Container.DataItem,
"edizione_diplomatica")%></b>
<br />
bibliografia: <b><%# DataBinder.Eval(Container.DataItem, "bibliografia")%> </b>
<br />
<br><b><a href="#top"><i><b>top >>></b></i></a> </b>
<hr />
<a name="id_decorazione">
<br />
<br />
<div style="font:12pt  verdana;color:darkred"><b><i>decorazione</i></b>
</div>
<br />
<br />
tipo di iniziale: <b><%# DataBinder.Eval(Container.DataItem, "tipo_di_iniziale")%></b>
<br />
miniatura: <b><%# DataBinder.Eval(Container.DataItem, "miniatura")%></b>
<br />
indicazioni per il miniatore: <b><%# DataBinder.Eval(Container.DataItem,
"indicazioni_per_il_miniatore")%> </b>
<br />
riproduzioni: <b><%# DataBinder.Eval(Container.DataItem, "riproduzioni")%></b>
<br />
<br><b><a href="#top"><i><b>top >>></b></i></a> </b>
<hr />
<a name="id_impaginazione">
<br />
<br />
<div style="font:12pt  verdana;color:darkred"><b><i>impaginazione</i></b>
</div>
<br />
<br />
numero strofe ms:  <b><%# DataBinder.Eval(Container.DataItem,
"numero_strofe_ms")%> </b>
```

```
<br />
numero tornadas ms:  <b><%# DataBinder.Eval(Container.DataItem,
"numero_tornadas_ms")%></b>
<br />
strofe di invio ms:  <b><%# DataBinder.Eval(Container.DataItem,
"strofe_di_invio_ms")%></b>
<br>
ordine delle strofe: <b><%# DataBinder.Eval(Container.DataItem,
"ordine_delle_strofe")%></b>
<br />
bianchi interni: <b><%# DataBinder.Eval(Container.DataItem, "bianchi_interni")%></b>
<br />
spazi finali: <b><%# DataBinder.Eval(Container.DataItem, "spazi_finali")%></b>
<br />
particolarità impaginazione: <b><%# DataBinder.Eval(Container.DataItem,
"particolarità_impaginazione")%> </b>
<br />
particolarità testo:  <b><%# DataBinder.Eval(Container.DataItem,
"particolarità_testo")%></b>
<br />
<br><b><a href="#top"><i><b>top >>></b></i></a> </b>
<hr />
<a name="id_metrica">
<br />
<br />
<div style="font:12pt  verdana;color:darkred"><b><i>metrica e musica</i></b>
</div>
<br />
<br />
notazione musicale:  <b><%# DataBinder.Eval(Container.DataItem,
"notazione_musicale")%></b>
<br />
identificazione melodia:  <b><%# DataBinder.Eval(Container.DataItem,
"identificazione_melodia")%></b>
<br />
note a melodia:  <b><%# DataBinder.Eval(Container.DataItem, "note_a_melodia")%>
</b>
<br><b><a href="#top"><i><b>top >>></b></i></a> </b>
<hr />
<a name="id_marginalia">
<br />
<br />
<div style="font:12pt  verdana;color:darkred"><b><i>marginalia e glosse</i></b>
```

```
            </div>
            <br />
            <br />
            marginalia:  <b><%# DataBinder.Eval(Container.DataItem, "marginalia")%> </b>
            <br />
            glosse:  <b><%# DataBinder.Eval(Container.DataItem, "glosse")%></b>
            <br />
            testi marginali:  <b><%# DataBinder.Eval(Container.DataItem, "testi_marginali")%></b>
            <br><br><b><a href="#top"><i><b>top >>></b></i></a> </b>
            <hr />
          </div>
        </ItemTemplate>
      </ASP:DataList>
    </div></form>
    <!-- FINE LIVELLO SCHEDE -->
```

carica il file che contiene il testo del piè di pagina

```
<BEdTASP:piedepagina id="UserControl5" runat="server"></BEdTASP:piedepagina>
```

fine del file di codice

```
</body>
</html>
```

4.4 Problemi rilevati durante la creazione del codice

La particolare tipologia dell'applicazione, soprattutto per la sua modalità di sviluppo dinamico delle pagine, ha dato modo di evidenziare alcuni problemi in fase di creazione delle stringhe di interrogazione del DB, soprattutto a livello di struttura della base di dati[65].

In particolare, è stata necessaria, a causa di una intolleranza del linguaggio di interrogazione SQL, la sistematica sostituzione dei nomi di campo o tabelle contenenti elementi di spazio con equivalenti nomi di campo o tabelle contenenti caratteri <_> in luogo degli spazi – es. *testi dei trovatori* → *testi_dei_trovatori* - (non è stata applicata la modifica inversa nella visualizzazione dei nomi di campo all'interno dell'interfaccia, per cui tutti i nomi di campo interessati conservano il carattere sostituito <_>).

Sempre relativamente ai problemi di alcuni caratteri speciali, si segnala la casuale presenza di caratteri apice <'> nelle stringhe di testo (nomi di trovatori, luoghi, titoli). Tali caratteri, essendo, purtroppo, anche fondamentali nella sintassi della stringa di interrogazione SQL sul DB, hanno generato una serie di errori in risposta alle ricerche. Per questo motivo è stata applicata una metodologia di sostituzione dinamica sistematica che, pur non danneggiando in alcun modo il dato originale, impedisce il continuo verificarsi di errori.

In secondo luogo, in alcuni casi, e soprattutto per i campi tematici (es. *argomento*, *genere*), si è resa necessaria una più efficace ridistribuzione dei dati su più campi: il primo (su cui viene eseguita la ricerca) sufficientemente generico e rispondente a una casistica intuitiva, il secondo approfondito ed esauriente.

---

[65] Per la costruzione del codice e la risoluzione dei problemi riscontrati, si è fatto costante riferimento ai manuali: Kline 2000; MacKenzie-Sharkey 2002; Payne 2001; Cassel-Palmer 1999; Isaacs 1997. Un apporto considerevole è stato apportato dalla consultazione delle risorse online su ASP.NET disponibili sul sito ufficiale Microsoft e sui siti dedicati agli sviluppatori web, indicati più avanti nelle pagine dedicate alle risorse online.

Rimane il problema dal punto di vista grafico della visualizzazione di alcuni campi troppo lunghi nei menu di selezione a discesa, a causa del loro contenuto eccezionalmente ampio rispetto alla media degli altri (es. alcuni schemi metrici, rimici o anche titoli di testi non possono essere visualizzati interamente).
Si aggiunge, a titolo di progetto di revisione futura, la necessità di imporre un ordinamento più stabile dei record, una visualizzazione più uniforme delle schede e una possibile gestione su finestre separate dei risultati delle varie elaborazioni.

4.5 Elementi di cornice non implementati

Pur essendo stati previsti per ragioni di completezza progettuale, i seguenti elementi non sono stati implementati:

- **BEdT**
  - *Il Progetto* (presentazione della base dati e dell'interfaccia)
  - *Aggiornamenti* (spazio per la pubblicazione di informazioni sugli aggiornamenti *in progress*)
  - *Area utenti* (spazio riservato alla registrazione degli utenti)
  - *Contatti* (spazio riservato alle comunicazioni degli utenti agli amministratori)
  - *Crediti* (spazio riservato alle informazioni di copyright)
- **?**
  - *Sommario e indice* (spazio riservato alla creazione di un manuale online per l'uso dell'applicazione)
  - *F.A.Q.* (spazio riservato alla risoluzione dei problemi più frequenti)
  - *Supporto tecnico* (spazio riservato al supporto tecnico, in caso di malfunzionamenti)
  - *Commenti e suggerimenti* (spazio riservato ai commenti degli utenti)
  - *Informazioni su BEdT* (Informazioni sulla versione dell'applicazione)

N.B. al momento solo la pagina di avvio possiede l'accesso riservato tramite indicazione di: Nome Utente e Password (bedt – bedt).

4.6 Considerazioni sulla possibilità di integrare documenti originali nella base di dati

Diversamente dal caso della base di dati BEdT attuale, dove l'informazione è in larga misura rigidamente strutturata, l'archiviazione e il reperimento di informazioni contenute in documenti di cui non sono noti dati caratteristici - a meno che non siano schedati anch'essi, ovviamente - introduce un fattore problematico in quanto entrano in gioco i concetti di *informazioni in senso semantico* oppure di *informazioni non strutturabili*[66].

Nel caso specifico dei manoscritti medioevali il problema si pone in quanto il documento originario non può essere memorizzato nell'elaboratore se non nelle forme della registrazione fotografica o della trascrizione (spesso già disponibile in edizione cartacea, l'edizione cosiddetta *diplomatica*), situazioni che richiederebbero nuove schedature sistematiche e uniformate dei dati fotografici e/o testuali.

Nello specifico, la struttura della BEdT si può agevolmente prestare all'apporto di tali nuovi classi di dati, sebbene possa esser necessario un ripensamento di alcuni suoi moduli, in funzione soprattutto di una eventuale implementazione di sezioni fotografiche documentali dei manoscritti e all'inserimento di trascrizioni complete, di edizioni critiche e raccolte di testi.

---

[66] La catalogazione di elementi non strutturati rappresenta un potenziale problema. Sull'argomento si veda la trattazione - più ampia in quanto coinvolge i problemi di catalogazione informatica per biblioteche e centri di documentazione - in Fadini-Savy 1999, pp. 167-179 e pp. 323-340.

5 BIBLIOGRAFIA[67]

chiave	autore	*monografia*	anno	edizione
Boschin 1995	Boschin, Roberto	*HTML*	1995	Apogeo
Cassel-Palmer 1999	Cassel, Paul; Palmer, Pamela	*Access 2000*	1999	Apogeo
Davis 1999	Davis, Mike P.	*Costruire applicazioni con Access*	1999	Apogeo
Davis-Lewis 2000	Davis, Peter; Lewis, Barry D.	*Windows 2000 Server*	2000	Apogeo
Downing-Covington 2000	Downing, Douglas A.; Covington,	*Dizionario di Internet e Computer*	2001	Il Sole 24Ore SPA
Fadini-Savy 1999	Fadini, Bruno; Savy, Carlo	*Informatica per le scienze umane*	1999	Franco Angeli
Greenspan-Bulger 2000	Greenspan, Jay; Bulger, Brad	*Sviluppare applicazioni per database con MySql/PHP*	2001	Apogeo
Grossi 1999	Grossi, Franco	*Tecnologie per l'informazione e la comunicazione*	1999	Franco Angeli
Isaacs 1997	Isaacs, Scott	*Inside Dynamic HTML*	1998	Mondadori Informatica
Kalata 2001	Kalata, Kathleen	*Programmazione Internet con VBScript e JavaScript*	2001	Apogeo
Kline 2000	Kline, Kevin e Daniel	*SQL guida di riferimento*	2001	Apogeo
Leiden-Wilensky 1999	Leiden, Candace; Wilensky,	*TCP/IP for Dummies*	1999	Apogeo
Livingston-Brown 1999	Livingston, Dan; Brown, Micah	*CSS e DHTML*	2000	Tecniche nuove
Lyons 2001	Lyons, Charles J.	*Progettare siti web. Dall'analisi all'implementazione*	2001	Tecniche nuove
MacKenzie-Sharkey 2002	MacKenzie, Duncan; Sharkey,	*Visual Basic .NET Guida completa*	2002	Apogeo
Maiocchi 2000	Maiocchi, Marco	*Ipertesti*	2000	Franco Angeli
Mannelli 2003	Mannelli, Piero	*PHP Collection. Progetti pratici per il tuo sito*	2003	Edizioni Master
Mari 2001	Mari, Luca	*Accesso a database via web*	2001	Apogeo
Meloni 2000	Meloni, Julie	*PHP per esempi*	2001	Jackson Libri
Mitchell-Atkinsons 2000	Mitchell, Scott; Atkinsons, James	*Active Server Pages 3.0*	2000	Apogeo
Mueller 2002	Mueller, John Paul	*Windows XP*	2002	Apogeo
Naccarato 1999	Naccarato, Giuseppe	*Visual Basic. La programmazione dei database*	1999	Apogeo
Norton-Kearns 1999	Norton, Peter; Kearns, David	*La Guida di Peter Norton a Le Reti*	2000	Apogeo

[67] Estratto in forma tabellare della Bibliografia. Per maggiore completezza di dati, vedi il file *Bibliografia_tesi.mdb* nel cd-rom allegato.

Payne 2001	Payne, Chris	*ASP.NET Guida completa*	2001	Apogeo
Sinclair 2000	Sinclair, Russel	*Da Access a SQL Server*	2001	Mondadori Informatica
Toschi 2001	Toschi, Luca	*Il linguaggio dei nuovi media*	2001	Apogeo
White 1999	White, Chuck	*Internet Explorer 5. Guida per lo sviluppatore*	2000	Tecniche nuove

BIBLIOGRAFIA: articoli

chiave	autore	*articolo*	rivista	supplemento	anno	n.	p.
Barbieri 2002	Barbieri, Lorenzo	*Microsoft .NET conviene davvero?*	ioProgrammo		nov 2002	63	10-14
Buono 2002	Buono, Luigi	*Utilizzare e programmare la barra dei menu*	ioProgrammo		dic 2002	64	72-74
Buono 2003	Buono, Luigi	*La tipologia d'interfaccia ed i menu standard*	ioProgrammo		gen 2003	65	70-74
Cangiano 2002	Cangiano, Antonio	*Alla scoperta di ASP.NET web Matrix*	ioProgrammo	Programmare il web	dic 2002	64	16-20
Naccarato 2002	Naccarato, Giuseppe	*Accesso ai dati con ADO.NET*	ioProgrammo		nov 2002	63	30-35
Naccarato 2003	Naccarato, Giuseppe	*SQL Server 2000 Microsoft Desktop Engine*	ioProgrammo		gen 2003	65	51-54
Pasqua 2002	Pasqua, Antonio	*PHP-Nuke 6.0 web Portal System*	ioProgrammo	Programmare il web	dic 2002	64	12-15
Pasqua 2003	Pasqua, Antonio	*PHP-Nuke 6.0 web Portal System (2)*	ioProgrammo	Programmare il web	gen 2003	65	12-14
Pelliccia 2002	Pelliccia, Carlo	*JDBC e l'accesso ai database*	ioProgrammo		dic 2002	64	68-71
Perri 2002	Perri, Antonio	*PHP & MySql*	ioProgrammo	Programmare il web	dic 2002	64	21-23
Uboldi 2002	Uboldi, Giuliano	*Programmazione Java Server Side*	ioProgrammo	Programmare il web	dic 2002	64	24-27
Uboldi 2003	Uboldi, Giuliano	*Introduzione a ColdFusion MX*	ioProgrammo	Programmare il web	feb 2003	66	18-21
Vergaini 2002	Vergaini, Marco	*Da ASP a ASP.NET*	ioProgrammo		ott 2002	62	27-30

5.1 Risorse online

Viene riportata di seguito la lista delle principali risorse online utilizzate, in supporto alla bibliografia cartacea, per la documentazione, la progettazione e la realizzazione dell'applicazione BEdT.

5.1.1 Risorse in Rete dedicate alla filologia romanza e in particolare alle ricerche filologiche online

http://www.sifr.it/
Sito ufficiale della Società Italiana di Filologia Romanza, fondata a Roma il 16 marzo 1990.

http://www.retimedievali.it
Iniziativa per gli studi medievistici, avviata nel 1998 (on line dal 2000) da un gruppo di studiosi appartenenti alle Università di Firenze, Napoli, Palermo, Trento e Venezia.

http://www.arnaut.it/
Rassegna di filologia occitana, con repertorio bibliografico aggiornato, a cura di Massimiliano De Conca.

http://www.spolia.it/
Rivista di filologia romanza curata da Teresa Nocita, con segnalazioni di ricerche in corso di svolgimento, resoconti da convegni, recensioni di articoli e libri, contributi inediti.
http://www.spolia.it/informatica/informatica.htm
Pagina dedicata alle risorse informatiche in rete dedicate al medioevo.

http://www.rialc.unina.it/
Repertorio informatizzato dell'antica letteratura catalana. A cura di Costanzo Di Girolamo (Università di Napoli).

Testi della letteratura catalana medievale (secoli XI-XIV), in edizioni nuove o rivedute.

http://www.rialto.unina.it/
Repertorio informatizzato dell'antica letteratura trobadorica e occitana. A cura di Costanzo Di Girolamo.
Testi della letteratura occitana medievale (secoli XI-XIV), in edizioni nuove o rivedute.

http://www.eniware.it/trobvers/
A cura di Rocco Distilo (Università della Calabria).
Concordanza in linea dell'intero corpus dei trovatori. Accesso mediante una password, ottenibile abbonandosi alla rivista 'Critica del testo'.

http://www.silab.it/frox/200/
A cura di Francesco Bonomi.
Archivio della poesia italiana dalle Origini a Dante, dotato di un motore di ricerca (attualmente disattivato).
http://www.silab.it/frox/200/backlink.htm
Pagina di link in Rete.

http://sunsite.berkeley.edu/PhiloBiblon/
A cura di Charles B. Faulhaber (University of California, Berkeley).
Bibliografia degli antichi testi catalani (Biteca), castigliani (Beta) e galego-portoghesi (Bitagap).
http://sunsite.berkeley.edu/PhiloBiblon/phrl.html
Pagina dei link ad altre risorse.

http://airas.cirp.es/WXN/wxn/frames/meddb.html
A cura di Mercedes Brea (Universidade de Santiago de Compostela).

Base di dati della lirica profana galego-portoghese. Si accede mediante una password, richiedibile con un modulo in linea.

http://airas.cirp.es/WXN/wxn/frames/bilega.html
BILEGA. Base di dati che contiene informazioni bibliografiche su lavori di ricerca o divulgazione sulla lingua galega.

http://lluisvives.com/
Biblioteca Virtual Joan Lluís Vives, a cura dell'Universitat d'Alacant e dell'Institut Joan Lluís Vives, associazione di diciotto Università della Catalogna.
Contiene opere della letteratura catalana antica e moderna, in edizioni dichiarate.

http://www.csovi.fi.cnr.it/
A cura di Pietro G. Beltrami (Università di Pisa).
Banca dati testuale del *Tesoro della lingua italiana delle Origini* (testi anteriori al 1375), consultabile attraverso il motore di ricerca dell'ARTFL Project. Contiene il progetto OVI, *Opera del Vocabolario italiano*. Fa parte del consorzio ItalNet.

http://www.csovi.fi.cnr.it/gframe.htm
G.A.T.T.O. 3.3 - Programma per la Gestione degli Archivi Testuali del Tesoro delle Origini. Fa parte del progetto OVI.

http://www.lib.uchicago.edu/efts/ARTFL/databases/PROV/
ARTFL. The Project for American and French Research on the Treasury of the French Language. È un programma cooperativo per l'analisi e il trattamento informatico della lingua francese del Centro Nazionale delle Ricerca Scientifica francese (CNRS) e dei Dipartimenti di Umanistica, Scienze Sociali e Servizi sui Testi Elettronici dell'Università di Chicago.

http://ovisun198.ovi.cnr.it/italnet/

ItalNet è un consorzio internazionale che organizza e distribuisce basi di dati e altri materiali di ricerca pertinenti agli studi italianistici.

http://vocabolario.biblio.cribecu.sns.it/Vocabolario/html/_s_index2.html

Vocabolario online degli Accademici della Crusca. Si basa sul software TReSy.

http://www.cribecu.sns.it/analisi_testuale/settore_informatico/tresy/potenzialita.html

Sviluppato dal Centro Ricerche Informatiche per i Beni Culturali (Cribecu) della Scuola Normale Superiore di Pisa, TReSy (Text Retrieval System) è un motore di ricerca testuale in grado di gestire le informazioni che può contenere un testo (o più testi) conforme alle specifiche di SGML/XML.

http://www.tei-c.org/

Text Encoding Initiative (TEI) Consortium.

TEI, nato nel 1987, è uno standard internazionale e interdisciplinare per ricerche online su testi letterari e linguistici; utilizza un proprio schema di codifica appositamente realizzato e basato su XML.

http://cibit.humnet.unipi.it/

La *Biblioteca Italiana Telematica*.

La Biblioteca italiana telematica è una biblioteca digitale di testi (fonti primarie) dedicata alla tradizione culturale italiana dal Medioevo al Novecento gestita dal CIBIT – Centro Interuniversitario Biblioteca Italiana Telematica: http://cibit.humnet.unipi.it/CIBIT.HTM**.** Si presenta come un deposito di testi elettronici, di norma scaricabili in vari formati, e come uno strumento di consultazione e di ricerca in linea, assistito dal motore di ricerca DBT.

http://cibit.humnet.unipi.it/DBT.HTM

DBT è un sistema informatico per la gestione e l'interrogazione di archivi testuali, sviluppato da Eugenio Picchi presso l'Istituto di Linguistica Computazionale del Consiglio Nazionale delle Ricerche di Pisa.

http://www.aib.it/aib/lis/opac1.htm
Associazione Italiana Biblioteche. Il mondo delle biblioteche in rete. Pagina degli OPAC italiani (Online Public Access Catalog).

http://atilf.inalf.fr/tlf.htm
Il *Trésor de la langue française* è un dizionario del XIX e XX secolo in 16 volumi e 1 supplemento. Il *Trésor de la langue française informatisé* ne rappresenta la versione consultabile online tramite un motore di ricerca ad accesso libero. Sviluppato dal CNRS e dal Laboratoire ATILF (Analyse et Traitement Informatique de la Langue Française) http://www.inalf.fr/_ie/atilf.htm

5.1.2 Risorse in Rete dedicate alle tecnologie di sviluppo per il web

I siti Internet segnalati offrono la più ampia panoramica sui problemi e le tecnologie che hanno interessato questo progetto, ponendosi come risorsa bibliografica alternativa ma estremamente efficace alla tradizionale fonte stampata.

Microsoft
Il sito ufficiale Microsoft contiene tutte le informazioni relative alla tecnologia ASP.NET utilizzata, software di sviluppo, tutorial, articoli di supporto.

Microsoft Developer's Network
The ASP.NET Web Matrix Project
ASP.NET Tutorial
F.A.Q.
ASP.NET Forums
Introduzione a .NET

Microsoft NET e J2EE tecnologie a confronto

Macromedia

Nel sito ufficiale Macromedia viene presentata *ColdFusion*, una delle tecnologie di creazione dinamica delle pagine Internet prese in considerazione per la realizzazione dell'applicazione.

Homepage Macromedia

ColdFusion Support Forums

Latoserver.it

Sito web italiano dedicato agli sviluppatori e ai webmaster. Particolarmente specializzato nello sviluppo in ambiente Linux e Java.

Homepage LatoServer.it

ASPItalia.com

Sito web italiano dedicato allo sviluppo ASP.NET. Pubblica periodicamente articoli di programmazione, script di esempio e tutorial.

Serie di articoli base

Serie di articoli avanzati

webmasterpoint.org

Sito web italiano dedicato ai webmaster con sezioni appositamente dedicate ad ASP e ASP.NET.

Introduzione ad ASP.NET

Caratteristiche di ASP.NET

Previsioni

HTML.it

Uno dei siti web italiani di riferimento per gli sviluppatori. Presenta due ampie sezioni dedicate rispettivamente al PHP e all'ASP, con articoli, esempi e tutorial.

freePHP_it - Risorse sul linguaggio PHP per webmaster - Manuale PHP.htm

Articoli ASP_NET Freeasp_html_it.htm

Proteggere il DB

Altre risorse dedicate al mondo dello sviluppo in ambiente Microsoft .NET:

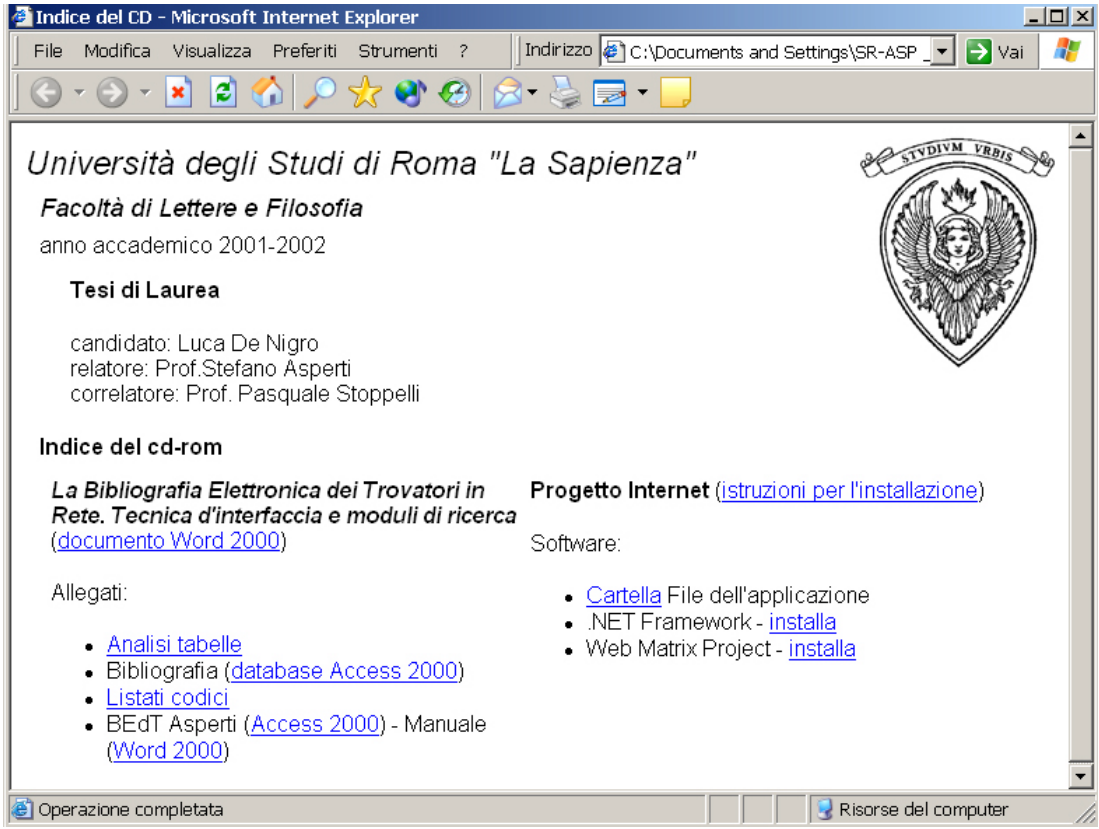
.NET 247	ASPWorkshops.com	MSDN
123aspx.com	c2i.fr (fr)	EraServer.NET
CodeGuru	Code Project	FreeDotNet.com
411ASP.NET Directory	CSharpFriends	FuzzySoftware.com
4GuysFromRolla.com	CShrp.net	GASP (fr)
Able Consulting, Inc.	DeepTraining.com	gasTIX.net
Alentus.com	Devpia	GotDotNet
angryCoder.com	DevASP.net	Innerhost.com
ASP 101	DeveloperFusion.com	MySantra.com
ASP Magazine (fr)	Developer.com	Orcsweb.com
ASP Today	DevelopersDex.com	PortalVB.com (sp)
ASP Watch	DevTrain (de)	Programmers Heaven
ASP Wire	DevX.com	QA.com/NET
ASP.NET FAQ	Dot NET Books	RegXLib.com
asp.net PRO	Dot NET Wire	scGlossary
ASP.NET.do (sp)	DotNET.be	Spinneret.co.uk
ASPAlliance	DotNet.Za.net	SuperExpert.com
ASPCN.com (cn)	DotNetBips.com	TripleASP.NET
ASPFree.com	DotNetDan.com	W3Coder.com
ASPItalia.com	DotNetExperts.com	Wintellect
ASPnedir?com (tr)	DotNetGerman.com (de)	WiseOwl.com
ASP.NET101.com	DotNetJunkies	WorldofDotNET.net
ASPObjects.com	MasterCSharp.com	XMLforASP.net
ASPSmith.com	DotNetSolutions	Yazgelistir.com (tr)

6 INDICE DELLE FIGURE

Figura 1 – Pagina principale dell'applicazione BEdT 5
Figura 2 – Menu di comando dell'applicazione. Scelta di un percorso di ricerca. 8
Figura 3 – Esempio di indice dinamico. Estrazione dal DB di tutti i nomi dei trovatori 8
Figura 4 – Aspetto completo della pagina di ricerca. Tramite l'indicazione del nome di un trovatore si accede al rapporto di tutti i testi attribuiti con possibilità di visualizzare i dettagli per ogni testo in un'apposita scheda 9
Figura 5 – Esempio di tabella di BEdT con l'indicazione della chiave primaria fornita dal numero di repertorio Pillet 13
Figura 6 – Il Pannello Comandi principale della base dati 16
Figura 7 – Esempio di maschera di visualizzazione dati "Indici Generali dei Canzonieri" 16
Figura 8 – Esempio di sistema di rete (fonte: Norton-Kearns 1999) 18
Figura 9 – Rete di collegamenti tra le varie pagine di un ipertesto (fonte: Maiocchi 2000)... 20
Figura 10 – Rapporto tra calcolatori-server e calcolatori-utente in configurazione non dinamica 22
Figura 11 – Flusso di informazioni tra componenti 23
Figura 12 – Gestione dei calcolatori-utente come semplici terminali 24
Figura 13 – Separazione dell'applicazione in due parti, collegate tramite rete 25
Figura 14 – Separazione della componente AL in due parti distribuite tra calcolatore-server e calcolatore-utente 26
Figura 15 – Struttura finale dell'applicazione client-server via web 28
Figura 16 – Flusso di esecuzione di una pagina ASP.NET, dalla richiesta alla visualizzazione (fonte: Payne 2001) 36
Figura 17 – Differenza tra il framework delle applicazioni tradizionali e .NET Framework (fonte: Payne 2001) 37
Figura 18 – Esempio di applicazione del modello all'interfaccia dell'applicazione 41
Figura 19 – La pagina indice_pc.aspx 46
Figura 20 – La pagina ind_canz_pc.aspx 52

7 IL CD-ROM

Il cd-rom allegato contiene tutta la documentazione, l'applicazione realizzata, i materiali necessari al suo funzionamento e le istruzioni dettagliate per l'installazione e la verifica delle sue parti. Il disco si avvia automaticamente, visualizzando una pagina di presentazione dei contenuti[68]:

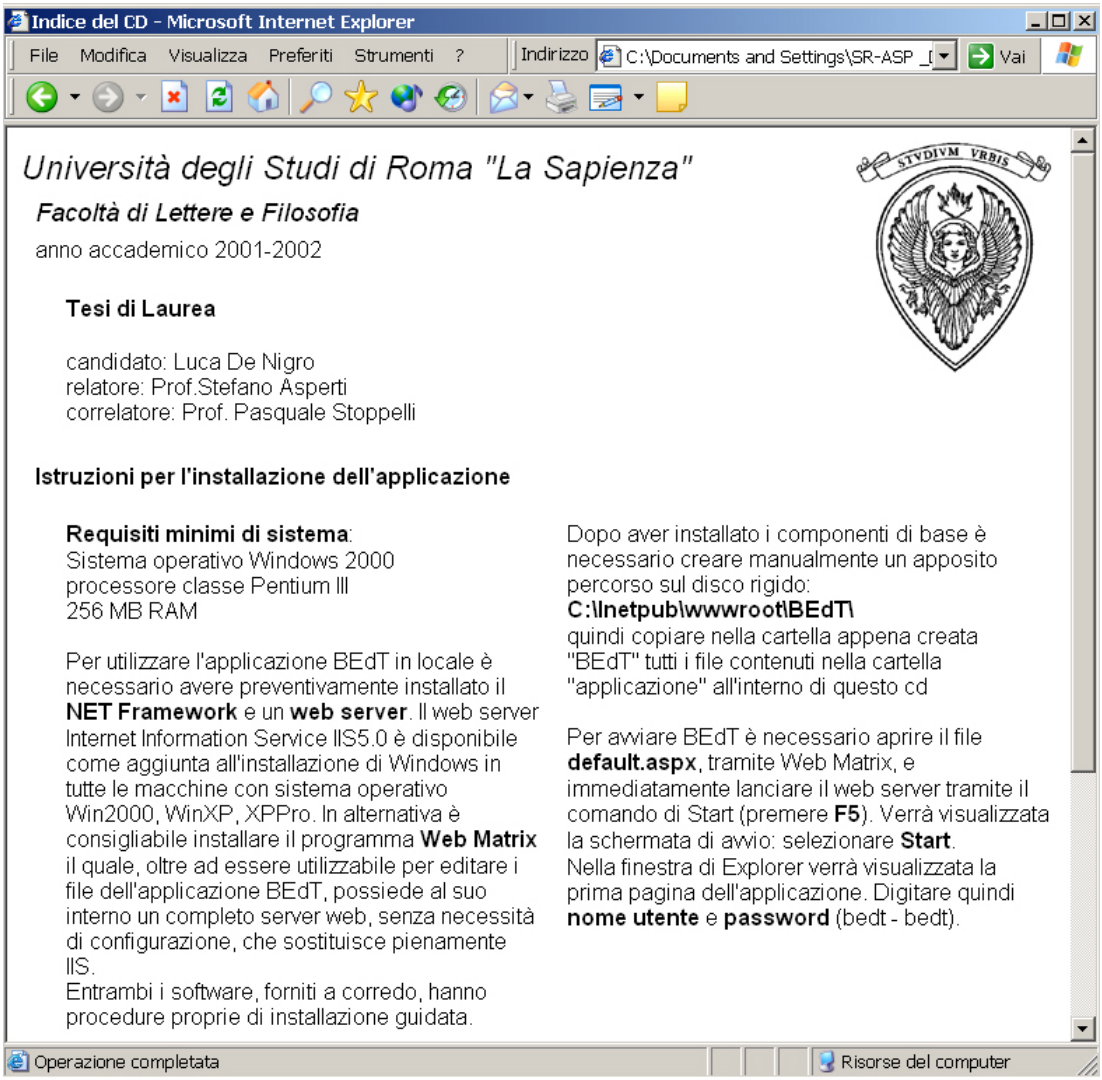


in dettaglio:

- Testo del presente elaborato in formato elettronico (documento word)
- Analisi dettagliata delle tabelle del database BEdT (documento word *analisi_tabelle.rtf*)
- Base di dati della Bibliografia (*bibliografia_tesi.mdb*)
- Listati completi del codice dell'applicazione (55 documenti con estensione ".txt" visualizzabili con qualsiasi editor di testo)

---

[68] Nel caso il disco non si avviasse automaticamente, a causa di particolari configurazioni del pc utilizzato per la prova, si dovrà richiamare (con un doppio click) il file *index.htm* che si trova all'interno del cd-rom.

- Base di dati BEdT Asperti (*BEdT2k01.mdb*)
- *BEdT Asperti. Istruzioni per l'uso* - versione di prova 0.8, gennaio 2003 (documento word *MANUALE.doc*)
- Cartella dei file dell'applicazione
- File di installazione di .NET Framework e Web Matrix Project
- Istruzioni per l'installazione dell'applicazione *BEdT online* (*istruzioni.htm*)

La pagina delle istruzioni guida nell'esecuzione di tutte le operazioni necessarie alla verifica dell'applicazione in locale, vale a dire su un elaboratore non connesso a Internet. In tal modo è possibile simulare la configurazione di una rete geografica con tutte le sue estensioni - dal server fino al terminale-utente - all'interno della struttura fisica e logica di una singola postazione di lavoro: