

UNIVERSITÀ DEGLI STUDI DI CAMERINO

FACOLTÀ DI SCIENZE E TECNOLOGIE

Corso di Laurea in Informatica

Dipartimento di Matematica e Informatica



**PUBBLICAZIONE DI ATTI PUBBLICI SUL WEB
PROGETTAZIONE E IMPLEMENTAZIONE DI
UN'APPLICAZIONE WEB PER LA PUBBLICAZIONE DEGLI
ATTI PUBBLICI DI UN ENTE COMUNE**

Tesi di Laurea
In
Reti di Elaboratori

Laureando
Francesco Federico

Relatore
Prof. Fausto Marcantoni

ANNO ACCADEMICO 2007/2008

INDICE

INTRODUZIONE Pag. 3

CAPITOLO 1 Pag.

IL PROGETTO

1.1 Scopo della tesi..... »»

1.2 Normativa di riferimento..... »»

1.3 Le applicazioni Web – based..... »»

1.3.1 Contesto di esecuzione..... »»

1.3.2 Architettura..... »»

1.3.2.1 Livello di presentazione..... »»

1.3.2.3 Livello logico..... »»

1.3.2.3 Livello dati..... »»

1.3.2.4 Problematiche..... »»

CAPITOLO 2 Pag.

ACQUISIZIONE DEGLI STRUMENTI UTILI ALLA

REALIZZAZIONE DELL'APPLICAZIONE WEB

2.1 Il Server Web Apache..... »»

2.2 Il Database MySQL..... »»

2.3 Il linguaggio PHP..... »»

2.3.1 Le caratteristiche del linguaggio..... »»

2.3.2	Il PHP è un linguaggio di scripting.....	»»
2.3.3	È un linguaggio “HTML-embedded”	»»
2.3.4	Opera server-side.....	»»
2.4	I vantaggi del linguaggio.....	»»
2.5	L'interprete PHP.....	»»
2.6	Altre tecnologie acquisite: HTML e SQL.....	»»
2.6.1	Il linguaggio HTML.....	»»
2.6.2	Il linguaggio SQL.....	
CAPITOLO 3		Pag.
IL CONTESTO REALE		
3.1	Specifiche utenti.....	»»
3.2	Specifiche sicurezza.....	»»
3.3	La realizzazione.....	»»
CONCLUSIONI		Pag. 1
BIBLIOGRAFIA		Pag. 1

INTRODUZIONE

L'efficienza del web sia come mezzo di divulgazione che come mezzo di vendita di prodotti e servizi è ormai nota a tutti. La rivoluzionaria affermazione di questo nuovo strumento basato sull'elaborazione distribuita ha creato nuove opportunità.

L'applicazione, "Progettazione ed implementazione di un'applicazione web per la pubblicazione degli atti pubblici di un Ente Comune", realizzata nello studio di questa tesi, che ha assunto come caso reale il Comune di Crotona, rappresenta il mezzo con cui la Pubblica Amministrazione locale può comunicare con i cittadini e le imprese, in quanto permette di rendere pubblici atti, quali: delibere, determine, bandi, regolamenti, ecc. Al tempo stesso permette di ottemperare agli obblighi normativi di pubblicazione, interagendo con la collettività.

La pubblicazione degli atti che devono essere portati a conoscenza del pubblico è stata da sempre effettuata attraverso l'affissione agli albi pretori (una serie di bacheche collocate alle pareti dei corridoi, o nell'atrio del palazzo comunale), in modo da assicurare la visibilità (pubblicità legale) degli atti medesimi.

Questa modalità di pubblicazione impone necessariamente di doversi recare presso la sede del palazzo comunale per poter conoscere gli atti adottati dagli organi collegiali. L'innovazione

tecnologica cambia radicalmente con la pubblicazione attraverso il web, strumento prontamente accessibile, in modo semplice ed intuitivo, la quale garantisce una maggiore agevolezza nella reperibilità degli atti di proprio interesse.

L'applicazione realizzata risponde alle nuove esigenze di efficienza e trasparenza che gli enti locali intendono soddisfare, in coerenza con quanto richiesto dalle emergenti tendenze dell'e-government. L'adozione di questo nuovo metodo di gestione degli atti, li rende ancora più realmente pubblici e con almeno un duplice risultato e cioè, maggiore visibilità e abbattimento del costo indiretto dei cittadini per la consultazione.

La tesi è organizzata nel seguente modo:

Il primo capitolo offre una panoramica, sui sistemi informatici orientati alla gestione di applicazioni web, e sugli obiettivi prefissati da questo progetto, con un breve riferimento alla normativa che disciplina la pubblicazione degli atti di interesse pubblico sul web, nelle pubbliche amministrazioni.

Il secondo capitolo descrive la tecnologia e gli strumenti che sono stati acquisiti ed impiegati per la realizzazione dell'applicazione: Apache, MySQL, PHP, HTML, SQL, su piattaforma Microsoft.

Il terzo capitolo riporta l'analisi dei requisiti, svolta attraverso interviste effettuate presso la sede del Comune di Crotona agli impiegati addetti, ed in modo speciale ai responsabili dei settori che si occupano in particolar modo del trattamento degli atti.

Le specifiche sulla sicurezza, quindi la definizione delle credenziali per l'accesso all'area di back-office (Area Privata), da parte degli

operatori addetti alla pubblicazione e gestione dei documenti. Riporta la descrizione dell'analisi concettuale dei dati, del modello relazionale, della progettazione logica e fisica dei dati sviluppata parallelamente allo sviluppo delle funzioni, ed infine illustra le funzionalità dell'applicazione realizzata.

La tesi si conclude con la descrizione dei vantaggi che l'applicazione ha portato all'amministrazione locale e quest'ultima, in termini di servizio, ha portato agli utenti fruitori, i cittadini.

CAPITOLO 1

IL PROGETTO

1.1 SCOPO DELLA TESI

Obiettivo principale della tesi, è quello di definire e realizzare un sistema informatico, che permetta ad un Ente Comune, in questa tesi è stato assunto come caso reale di studio il Comune di Crotone, la pubblicazione degli atti di interesse pubblico sul web.

A tal fine è stata analizzata e studiata, con gli impiegati addetti e con i funzionari responsabili del servizio, la procedura di generazione, gestione e pubblicazione degli atti quali: delibere di consiglio, delibere di giunta, determinazioni, statuti e regolamenti, gare di appalto, modulistica. Si è cercato di far fronte, sia ai problemi legati alla organizzazione delle attività, quindi problemi relativi al trattamento del documento fino alla sua pubblicazione sul web, sia ai problemi inerenti la normativa che disciplina tale attività di pubblicazione degli atti. Ed infine si è cercato di rendere, ossia di offrire, agli utenti che ne facessero richiesta (fruitori quindi del servizio), la reperibilità degli atti di proprio interesse in modo

agevole e nella massima semplicità. L'applicazione è stata dotata di un'interfaccia di back-office (Area Privata) dove gli addetti possono inserire tutti i dati necessari alla pubblicazione degli atti, immediatamente visibili sull'interfaccia di front-office, raggiungibile all'indirizzo:

<http://www.sitoistituzionale/pubblicazioneatti/index.php>

L'applicazione intende collocarsi come “servizio aggiunto” all'interno del sito istituzionale dell'ente, perciò i destinatari di questo servizio saranno tutti i cittadini e le imprese che collaborano o intendono collaborare con l'ente.

L'applicazione oltre a voler garantire maggiore trasparenza, intende anche offrire all'amministrazione locale un abbattimento dei costi diretti: la pubblicazione degli atti che devono essere portati a conoscenza del pubblico è stata da sempre effettuata attraverso l'affissione agli albi pretori¹, in modo da assicurare la visibilità (pubblicità legale) in ottemperanza alle disposizioni di legge. Questa nuova metodologia di gestione prevedendo l'acquisizione dei documenti direttamente in formato digitale per la pubblicazione sul web, consente agli uffici preposti di ridurre il volume di carta² prodotta con un conseguente risparmio economico, che si riflette naturalmente anche sugli utenti che intendono consultare gli atti (i cosiddetti costi indiretti), perché non dovranno più necessariamente recarsi al palazzo comunale, ma potranno consultare il materiale di

¹ Gli albi pretori sono situati in delle bacheche collocate nell'atrio del palazzo comunale o nei corridoi.

² Ad esempio: le stampe per la modulistica verranno ridotte di almeno il 60% prima che il servizio abbia raggiunto il massimo livello di divulgazione.

loro interesse, direttamente da casa propria o da una qualsiasi postazione che disponga di un accesso ad internet.

1.2 NORMATIVA DI RIFERIMENTO

Il Capo V del Codice dell'Amministrazione Digitale (L. 82/2005) regola nel dettaglio le caratteristiche (art. 53) ed i contenuti (art. 54) dei siti web delle pubbliche amministrazioni, attribuendo così a tali strumenti di comunicazione un ruolo indubbiamente stabile e di rilievo nei processi organizzativi delle p.a. L'art. 54 del Codice dell'Amministrazione Digitale (CAD) elenca poi, al primo comma, le tipologie di dati pubblici che devono essere presenti sui siti delle amministrazioni, sia centrali che regionali e locali.

La direttiva P.C.M. 30 Maggio 2002, infatti, con una certa lungimiranza indicava già come “massa critica” di informazioni da pubblicare sui siti web istituzionali i dati relativi alla struttura organizzativa dell'ente, all'articolazione degli uffici, ecc., precisando però che “...Questo...non deve rappresentare il contenuto principale del sito, che deve focalizzato sull'erogazione dei servizi all'utente”. Il quarto comma dell'art. 54, impone alle pubbliche amministrazioni di garantire la conformità e la corrispondenza delle informazioni contenute sui siti rispetto a quelle contenute nei provvedimenti amministrativi originali. L'art. 1, comma 3, del decreto legislativo n. 360/1998 prevede l'obbligo di pubblicazione su un apposito sito web delle delibere dei comuni relative all'addizionale comunale Irpef con decorrenza dell'efficacia delle stesse delibere dalla data di pubblicazione.

L'art. 24 della legge n. 340/2001 ha disposto che a decorrere dal 1° gennaio 2001 le pubbliche amministrazioni sono tenute a pubblicare tutti i bandi e gli avvisi di gara su uno o più siti Internet, individuati con apposito decreto del Presidente del Consiglio. Tale pubblicazione, seppure limitatamente ai bandi ed avvisi di gara di importo inferiore a quello di applicazione della disciplina comunitaria, ha sostituito ogni altra forma di pubblicazione prevista da norma di legge o di regolamento.

Viste le varie disposizioni di legge, che regolamentano la pubblicazione sul web degli atti prodotti dalla pubblica amministrazione, il Comune di Crotona, per venire incontro alla necessità di disciplinare ed organizzare la pubblicazione dei dati sul web, ha redatto un apposito regolamento. In particolare tenendo conto delle direttive del Decreto legislativo marzo 2005 n. 82, Codice dell'Amministrazione Digitale (CAD) già citato, e del provvedimento del Garante del 19 aprile 2007, ha indicato le linee guida per il corretto utilizzo degli strumenti informatici e stabilito cosa e come pubblicare le informazioni sul web. Di seguito si riportano solo alcuni punti del suddetto regolamento, che riguardano dati inerenti allo sviluppo dell'applicazione web realizzata nello studio della tesi.

Articolo II Contenuto del sito istituzionale.

Il Comune di Crotona rende disponibile l'accesso telematico a dati, documenti e procedimenti mediante l'uso di tecnologie dell'informazione e della comunicazione. Restano salvi i limiti previsti dalle leggi e dai regolamenti in materia di protezione dei

dati personali. Il Comune di Crotone, a tal fine, rende disponibile sul web, rispettando i principi di accessibilità, usabilità, ecc., completezza dell'informazione, chiarezza di linguaggio, semplicità di consultazione ed interoperabilità.

Tra i dati pubblici da pubblicare necessariamente troviamo: l'elenco delle tipologie di procedimento svolte da ciascun ufficio (delibere, determine, ecc.); l'elenco di tutti i bandi di gara e di concorso; l'elenco della documentazione richiesta per i singoli provvedimenti, i moduli e i formulari validi ad ogni effetto di legge.

I dati pubblici sono fruibili in rete, sul web, gratuitamente e senza necessità di autenticazione informatica.

Le informazioni devono essere conformi e corrispondenti alle informazioni contenute nei provvedimenti amministrativi originali.

Articolo III La diffusione dei dati su internet.

Tutti documenti pubblicati devono essere in formato aperto³.

Tutti gli atti dell'amministrazione comunale sono pubblici, ad eccezione di quelli che siano considerati "riservati" per espressa indicazione di legge, oppure per effetto di una dichiarazione del sindaco che ne vieti l'esibizione poiché la loro diffusione può pregiudicare il diritto della riservatezza di persone, gruppi o imprese e quindi possono essere diffusi tramite il web.

E' consentita la pubblicazione delle deliberazioni consultabile rispettando il principio di pertinenza e non eccedenza o, se i dati

³ pdf, rtf, txt, o altri formati accessibili con piattaforme open source.

sono sensibili o giudiziari, di indispensabilità, rispetto alle finalità perseguite con i singoli atti.

E' consentita la pubblicazione, da parte dell'ente, di notizie a sfondo giornalistico.

E' consentita la pubblicazione di esiti concorsuali, in particolare le graduatorie dei vincitori di concorsi, rispettando il principio di pertinenza e non eccedenza nel trattamento dei dati.

Articolo IV Profilo organizzativo.

Il Centro Elaborazione Dati (CED) è responsabile della strumentazione necessaria per il funzionamento del sito.

Ogni settore ha l'obbligo di gestire autonomamente, seguendo le indicazioni del presente regolamento, i dati da pubblicare sul web.

I dirigenti di settore conferiscono la nomina di addetto all'aggiornamento delle pubblicazioni ad almeno due unità che, dopo un'opportuna formazione possono provvedere alla pubblicazione e all'aggiornamento dei dati della propria struttura.

La pubblicazione dei bandi di gara ed avvisi pubblici è a cura dell'ufficio competente del servizio. Tutti gli atti amministrativi⁴ vengono pubblicati sul web dall'ufficio preposto alla stampa degli stessi. Le news, comunicati stampa ed eventuali forum vengono gestiti direttamente dall'ufficio comunicazioni e/o dagli addetti degli uffici di staff e agli organi istituzionali.

Articolo V Controllo delle attività.

Per quanto stabilito dal CAD, le pubbliche amministrazioni garantiscono che le informazioni contenute sul web siano conformi

⁴ Delibere e determine.

e corrispondenti alle informazioni contenute nei provvedimenti amministrativi originali dei quali si fornisce comunicazione tramite il web e pertanto la pubblicazione telematica produce effetti di pubblicità legale nei casi e nei modi espressamente previsti dall'ordinamento. L'amministrazione nomina, con atto successivo l'incarico di Editore al dipendente che avrà l'onere di controllare e verificare di quanto disposto nel presente regolamento ed avrà la responsabilità di pubblicare il lavoro prodotto dai vari uffici.

1.3 LE APPLICAZIONI WEB - BASED

Applicazione Web, ovvero Web-application, è l'espressione che in generale viene impiegata per indicare tutte le applicazioni distribuite web-based. Più precisamente impiegata in ambito engineering, dove con il termine webapp si descrive un'applicazione accessibile via web per mezzo di una network, come ad esempio una Intranet o attraverso la rete Internet.

Questo modello applicativo è divenuto popolare alla fine degli anni novanta, in considerazione della possibilità per un client generico, di accedere a funzioni applicative, utilizzando come terminale normali web browser. Le funzioni svolte da questo tipo di software possono andare dalla semplice ricerca e fruizione di un file in una directory, a delle applicazioni estremamente sofisticate che eseguono in tempo reale, vendite, tenuta del magazzino, registrazione di dati finanziari, ecc.

Anche le tecnologie che sono alle spalle di questi software sono estremamente varie: dal semplice script fino ad un complicato programma in linguaggio di alto livello che interagisce con diverse risorse dati localizzate su sistemi ed in modi diversi. A causa di tali caratteristiche le applicazioni sono estremamente disomogenee, e non permettono alcuna tassonomia o ricerca. L'unica analisi che può essere realizzata, e che vedremo più avanti, è uno studio del tutto generale a riguardo della struttura interna del software.

Le applicazioni Web si pongono come valida alternativa alle tradizionali applicazioni Client/Server per vari motivi:

- facilità di distribuzione e aggiornamento: un'applicazione Web si trova interamente sul server, per cui la pubblicazione sul server coincide con la distribuzione e l'aggiornamento effettuato sul server ed è automaticamente reso disponibile a tutti gli utenti;
- accesso multiplatforma: l'accesso all'applicazione è indipendente dall'hardware e dal sistema operativo utilizzato dagli utenti;
- riduzione del costo di gestione: l'uso di Internet come infrastruttura per un'applicazione Web riduce notevolmente sia i costi di connettività che i costi di gestione dei client;
- scalabilità: un'applicazione Web ben progettata può crescere insieme alle esigenze dell'azienda senza particolari problemi.

1.3.1 CONTESTO DI ESECUZIONE

Un elemento molto importante nella progettazione di applicazione Web è il contesto in cui verrà eseguita l'applicazione: Internet, Intranet o Extranet.

Se il contesto di applicazione è Internet, come nel nostro caso, occorrerà prendere in considerazione la compatibilità con diversi tipi e versione di browser, sarà necessario tenere conto delle velocità di accesso alle risorse e di conseguenza alla riduzione delle dimensioni del materiale da scaricare e all'ottimizzazione dei tempi di risposta da parte del server. Nel caso di trasferimento di documenti e informazioni riservate sarà opportuno prevedere l'uso di protocolli sicuri. Nel caso di Intranet vengono meno le considerazioni relative alla compatibilità con i browser e alla velocità di accesso alle risorse, per cui si ha una maggiore libertà nello sviluppo dell'applicazione.

Se il contesto di esecuzione è Extranet, cioè una Intranet accessibile totalmente o in parte anche dall'esterno, è necessario riconsiderare la parte accessibile dall'esterno in base a considerazioni fatte sul contesto Internet. In ogni caso una particolare cura dovrà essere posta nel controllo delle autorizzazioni di accesso all'applicazione da parte degli utenti esterni.

1.3.2 ARCHITETTURA

I componenti fondamentali di un'applicazione Web sono analoghi per certi versi a quelli di una tradizionale applicazione client/server. Una tipica applicazione client/server è costituita da un client che implementa l'interfaccia utente con alcune funzionalità di elaborazione e di comunicazione, e da un server che fornisce una serie di servizi come la gestione e l'accesso ai dati di un database. Nell'ambito Web l'interazione tra client e server è un po' più articolata, per consentire l'integrazione di componenti di varia natura.

Un'applicazione Web si basa su elementi software standard indipendenti dalle caratteristiche della particolare applicazione e dalla piattaforma software e hardware su cui viene eseguita.

Per sua natura, una webapp può presentarsi con diverse strutture ed organizzazioni logiche, poiché di fatto racchiude in se, un modello tecnico ed una filosofia di sviluppo, allo stesso tempo. Tuttavia, sul piano dell'informatica teorica è possibile riconoscere una strutturazione tipica su tre livelli logico-funzionali (applicazioni Three-Tier) ma che possono essere distribuiti anche su più livelli (applicazioni Multi-Tier).

Nella figura seguente è mostrata l'architettura tipica di un'applicazione Web Three-Tier, che può essere scomposta in tre livelli: presentazione, logico e dati .

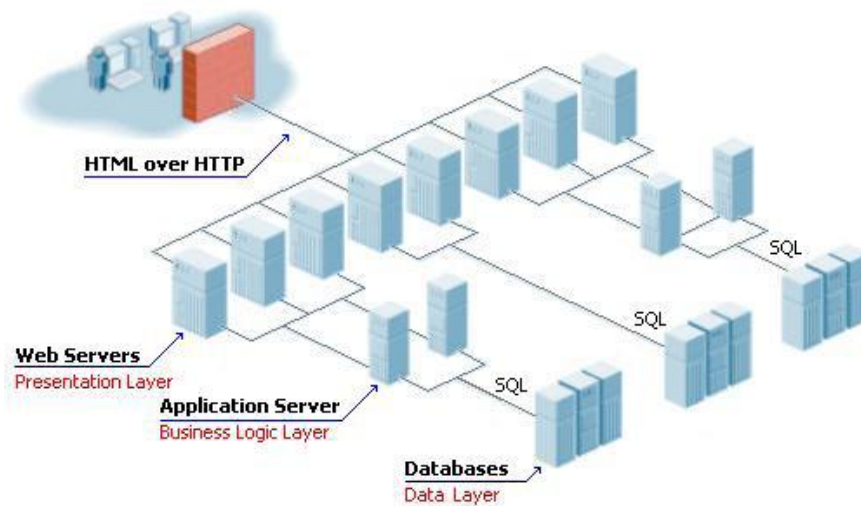


Figura 1 Applicazione Web Three-Tier

- il livello più esterno, di presentazione, rappresenta l'interfaccia utente dell'applicazione, e si occupa di acquisire dati e visualizzare risultati tramite un linguaggio interpretabile dal browser (o da un generico client), tipicamente html oppure xml;
- il livello logico, quello intermedio è il vero nucleo dell'applicazione. In esso si concentra l'implementazione degli algoritmi, sia quelli funzionali all'offerta del servizio, sia quelli accessori, come per esempio il controllo di sicurezza. Capita, perciò, che questo strato sia talmente sottile da poterlo inglobare nel *presentation layer* oppure, al contrario, sia così complesso da dover essere diviso in

diverse sezioni cooperanti fra loro che si occupano di compiti molto specifici.

- L'ultimo livello è il livello dati, non meno importante, rappresenta l'insieme dei servizi offerti da applicazioni indipendenti dal Web, come ad esempio un gestore di database, un sistema di gestione di posta elettronica, ecc.

Non sempre i livelli logici di un'applicazione Web corrispondono a locazioni fisiche sulla rete. Si va dal caso in cui tutti e tre i livelli risiedono sulla stessa macchina a varie altre distribuzioni, fino alla corrispondenza di ciascun livello con una macchina fisica. Di seguito verranno analizzati i livelli architetturali di una tipica applicazione Web e le varie modalità di progettazione.

1.3.2.1 LIVELLO DI PRESENTAZIONE

Il livello di presentazione costituisce l'interfaccia utente dell'applicazione Web e corrisponde a quello che nelle applicazioni client/server standard è il client. Esso è costituito da vari componenti combinati tra loro: browser, documenti HTML, applet Java, controlli ActiveX, ecc. La capacità di utilizzo di questi elementi da parte della piattaforma client è uno dei problemi principali nella realizzazione di questo livello. Le soluzioni vanno dalla scelta di sfruttare al massimo le capacità elaborative del client, alla realizzazione di un livello di presentazione universale. In genere l'identificazione del contesto di esecuzione dell'applicazione contribuisce in larga parte a delineare le soluzioni

da adottare nella realizzazione del livello di presentazione. Un altro elemento da prendere in considerazione è il livello di complessità dell'interfaccia utente dell'applicazione. Il linguaggio HTML, nato per la distribuzione di documenti in ambienti distribuiti, non consente di progettare interfacce utenti molto avanzate (drag-and-drop, tree-view control, tabbed control, ecc.). Una soluzione è quella di prevedere un insieme di pagine HTML standard, arricchite da applet Java.

1.3.2.2 LIVELLO LOGICO

Il livello logico, quello intermedio di un'applicazione Web contiene il core applicativo, ovvero il motore elaborativo dell'applicazione. Esso è in grado di soddisfare le richieste di dati e di elaborazione del client. Le modalità di realizzazione del livello intermedio dipendono spesso dalle caratteristiche e dalle tecnologie supportate dal server Web e/o da componenti installati sul server applicativo. In ogni caso la funzionalità fondamentale del server Web su cui si basa l'intera applicazione è il supporto di elaborazioni. In base alla tipologia di applicazione da sviluppare è opportuno prevedere funzionalità particolari, come ad esempio:

- gestione delle transazioni: gestione affidabile del flusso dati;
- gestione della sicurezza e della riservatezza: gestione della sicurezza nell'accesso all'applicazione e la riservatezza nella trasmissione di informazioni; la gestione della sicurezza è spesso realizzata tramite l'autenticazione

(username e password) dell'utente, mentre la riservatezza si basa su protocolli ad hoc, come SSL (Secure Socket Layer) o PCT (Private Communications Technology).

Il livello intermedio di un'applicazione Web può essere costituito da un insieme di script (codice scritto in un qualche linguaggio di sviluppo dinamico⁵ lato-server), componenti e programmi interagenti tra di loro e con il server Web tramite le seguenti tecnologie:

- Common Gateway Interface (CGI): consente l'attivazione di un programma o di uno script su richiesta del client (portabile su qualsiasi piattaforma).
- Internet Server Application Programming Interface (ISAPI): consente l'esecuzione di una libreria dinamica (DLL) all'interno dello spazio di memoria del server Web (funziona solo su piattaforma Windows).
- Active Server Pages (ASP): consente l'interpretazione di script nell'ambiente del server Web e la creazione in modo dinamico di documenti Web (funziona solo su piattaforma e server Web Windows).
- Hypertext Preprocessor (PHP): consente l'interpretazione di script nell'ambiente del server Web e la creazione in modo

⁵ Linguaggi dinamici come PHP, ASP, ASP.NET, ecc. Già a partire dal primo Congresso Web tenutosi a Ginevra nel 1994 è iniziata ad affermarsi l'esigenza di rendere il web dinamico. L'HTML di per se è statico, permette al massimo di inserire in una pagina un'immagine, una colonna sonora oppure uno spezzone video, ma non è capace di generare dinamicamente il contenuto di una pagina. Infatti non possiamo definirlo un linguaggio di programmazione, ma solo di formattazione dei contenuti, permette cioè, di controllare soltanto come un testo comparirà nella finestra del browser.

dinamico di documenti Web (ma funziona sia su piattaforma UNIX-Like che Microsoft).

- Java Servlet: consente di eseguire classi Java su richiesta del client (portabile su qualsiasi piattaforma)

1.3.2.3 LIVELLO DATI

Il livello dati fornisce servizi non direttamente disponibili tramite il Server Web. Questi servizi sono generalmente forniti da applicazioni indipendenti dall'ambiente Web e spesso costituiscono le applicazioni preesistenti in un'azienda. Tipici esempi di applicazioni presenti a questo livello sono: server dati (DBMS), server di mail, ecc.

In genere è opportuno prevedere dei componenti dell'architettura dell'applicazione che fungono da connettori tra il livello intermedio e il livello dati. Infatti, utilizzando dei connettori per l'interazione con applicazioni esterne non standard si facilita la manutenibilità nel caso in cui queste vengano modificate o sostituite (es. ODBC, OLEDB).

1.3.2.4 PROBLEMATICHE

La realizzazione di applicazioni Web presenta alcune problematiche sconosciute alle applicazioni tradizionali. Queste problematiche derivano dalla natura stessa del Web, pensato originariamente, non per la realizzazione di applicazioni ma per la distribuzione di documenti. Un classico problema è quello del cosiddetto *mantenimento dello stato dell'applicazione*. Il modello di interazione di base del Web è privo del concetto di connessione, per cui l'esecuzione di un'applicazione Web consiste in una serie di interazioni disconnesse. Lo sviluppatore deve utilizzare informazioni codificate nella richiesta HTTP e nelle risorse condivise durante la sequenza di interazioni per stabilire la continuità dell'applicazione. La progettazione di un'applicazione Web deve tenere conto anche dell'ampiezza di banda di trasmissione disponibile e del carico di lavoro stimato per il server. Dalle considerazioni sull'ampiezza di banda derivano le scelte relative alla quantità di dati da inviare ad un client in risposta ad una richiesta. Ciò può coinvolgere anche l'interfaccia utente dell'applicazione stessa, portando in certi casi a rinunciare ad una veste grafica accattivante pur di fornire tempi di risposta accettabili.

Tuttavia, all'ottimizzazione dell'uso dei mezzi trasmissivi può contribuire, e in certi casi essere determinante, un'attenta

ripartizione del carico di elaborazione tra client e server. Infatti, consentire al client di effettuare alcune elaborazioni senza coinvolgere il server, come ad esempio la convalida dei dati inseriti in un form, permette di ridurre il traffico sulla rete, sia di snellire il carico di lavoro del server. Inoltre, la presenza di applicazioni esterne particolarmente complesse sulla stessa macchina del server Web può rendere meno efficiente l'applicazione Web sottraendole risorse preziose. In questi casi è opportuno distribuire il carico di lavoro su macchine diverse prevedendo, ad esempio, una macchina diversa per un DBMS o per un'applicazione particolarmente esosa di risorse.

Possiamo concludere dicendo che non sempre i criteri tipici della progettazione di applicazioni client/server possono essere applicati alla progettazione di applicazioni Web. Infatti, nella maggior parte dei casi la natura stessa del Web e delle problematiche che esso presenta, porta a basare le proprie scelte progettuali sulla propria esperienza o sulla esperienza acquisita da altri, in mancanza di modelli di riferimento consolidati.

CAPITOLO 2

ACQUISIZIONE DEGLI STRUMENTI UTILI ALLA REALIZZAZIONE DELL'APPLICAZIONE WEB

Abbiamo già detto che l'applicazione web non è un unico blocco a se stante, ma piuttosto un sistema complesso, in cui un numero anche elevato di componenti eterogenei interagiscono e collaborano, al fine di fornire un servizio o un prodotto agli utenti che ne fanno richiesta. Definiamo in questo ambito l'acronimo W.A.M.P., una piattaforma o ambiente, la quale ha permesso di implementare un'applicazione web scritta in PHP su una macchina Windows based. Con queste quattro lettere si intende una particolare combinazione di software di sistema che sarà d'ora in poi la piattaforma di riferimento per questa tesi.

Questa combinazione è costituita da:

- **Windows:** un sistema operativo che oggi è ampiamente utilizzato; L'applicazione web è stata realizzata prendendo in considerazione come caso reale di

studio il Comune di Crotona, che utilizza tutti i sistemi informatici su piattaforma windows.

- **Apache:** un server web che gode di una amplissima diffusione presso gli ISP di tutto il mondo, sulla quale girano la maggior parte dei siti sulla rete;
- **MySQL:** un RDBMS (relational database management system), che sta acquisendo tutte le caratteristiche che sono proprie delle basi di dati blasonate. Inoltre è utilizzato anche negli ambienti più eterogenei.
- **PHP:** un linguaggio di scripting per il web, che gode di ampia notorietà, date le proprie doti di flessibilità e versatilità nello sviluppo di applicazioni per gli ambienti di rete.

La piattaforma W.A.M.P. è stata realizzata attraverso l'installazione di un tool, WM Server Tools 3.3, su una macchina windows based. Il tool raccoglie appunto: Apache - PHP - Perl - PHPMyAdmin - MySQL e WMServer SMTP (per testare l'invio delle e-mail in locale).

E' uno strumento molto discreto, infatti al suo avvio si limita ad inserire un'icona sulla barra delle applicazioni, dove è possibile visualizzare l'avvio corretto di Apache e MySql tramite due spie di colore arancio (grigio = spento; rosso = errore). Con un "popup menu" disponibile direttamente sull'icona è possibile eseguire le seguenti impostazioni:

- Setting: Cambiare la lingua e impostare le opzioni di avvio;

- Menu Editor: Gestire, tramite l'apposito editor, i collegamenti che sono presenti sul menu e i corrispondenti sottomenu, crearne nuovi, nonché cambiare le icone;
- Check for Update: Controllare eventuali aggiornamenti.
- Add-ons: Controllare ed aggiungere nuove funzioni. (Di default è installato WMServer SMTP che consente di testare l'invio di una e-mail in locale e WMLogs Reset) .
- WWW: Accedere direttamente all'homepage del sito, al settings del phpmyadmin e alle cartelle contenenti il tutto.
- Utility: Configurare i file my.ini e php.ini, i servizi di windows e altro.
- Start Apache: Avviare Apache;
- Stop Apache: Terminare Apache;
- Start MySql: Avviare MySql;
- Stop MySql - Terminare MySql.

Gli strumenti che abbiamo a disposizione sono resi ancora più funzionali dalla presenza di PHPMyAdmin, una comodissima interfaccia web per il database, che permette di fare operazioni dal browser senza scrivere una riga di SQL, e dalla possibilità di installare anche più versioni del linguaggio PHP, del database MySQL e del web server Apache, nel caso in cui ovviamente ne avessimo bisogno, per testare o usare applicativi “particolari”. Direi che la combinazione W.A.M.P. è la soluzione definitiva per lavorare e testare applicazioni di grandissima portata. Il tutto nella più assoluta semplicità e sicurezza. Ultima nota di merito: La piattaforma W.A.M.P. può essere assemblata con licenza GPL

tranne naturalmente il sistema operativo Windows. Non vi sono dubbi che l'ambiente lato-server in cui PHP dà il meglio di sé sia quello Unix-like. Resta il fatto però che Windows è il sistema operativo più utilizzato, soprattutto nel contesto reale di riferimento, il Comune di Crotona. Buona parte del successo di PHP deriva dalla particolarità che non solo si tratta di linguaggio quasi perfettamente multiplatforma, ma che è anche possibile ricreare in Windows un ambiente di esecuzione molto simile a quello dove gli script, una volta testati in locale, andranno a girare dopo essere stati caricati sul server in hosting.

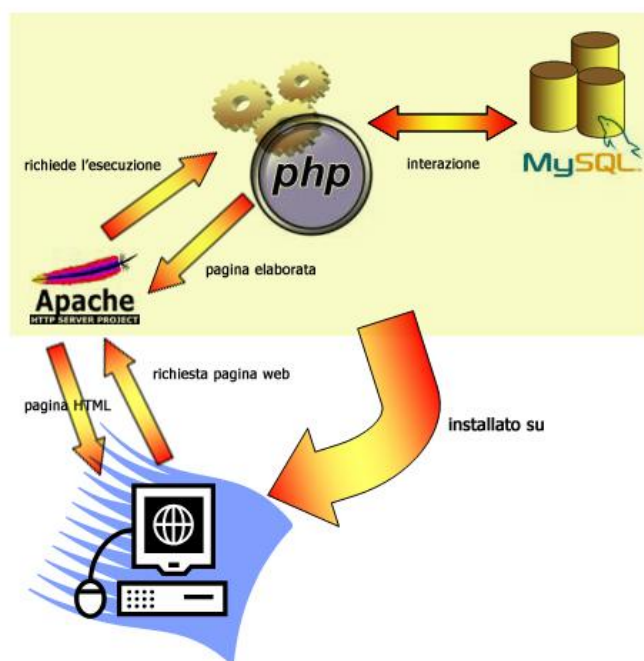


Figura 2 (Visione d'insieme dell'ambiente di esecuzione W.A.M.P.)

Abbiamo detto che W.A.M.P. è un acronimo, una semplice sigla che vuole significare un paradigma di programmazione molto diffuso, ovvero la pacchettizzazione in un unico tool di installazione dei tre software prima menzionati, naturalmente per ambiente Windows, e che brevemente andiamo a descrivere.

Tutti i programmi sono installati dentro a questa path: C:\WM, e la pathname dove ci metteremo tutti i files visibili dall'esterno sarà c:\WM\WWW. Si crea così una sorta di *jail*, di gabbia dentro la quale girerà il nostro web server.

2.1 IL SERVER WEB APACHE

Prima di andare a descrivere il server web Apache, impiegato in questa applicazione, vediamo quali sono i web server attualmente più diffusi nel mondo:

- Apache, sviluppato dalla Apache Software Foundation, è disponibile sia per piattaforme Microsoft che per sistemi Unix.
- Internet Information Server, prodotto da Microsoft per i propri sistemi operativi.
- Personal Web Server, prodotto anch'esso da Microsoft, accompagna il popolare editor visuale FrontPage.
- Xitami, prodotto dalla iMatix è un altro Web server Open Source disponibile sia per Unix che per ambienti Windows.

- OmniHTTPd, prodotto da Omicron Technologies Corporation è un web server con licenza gratuita per usi non commerciali. Esiste solo in versione per Windows.

C'è da dire innanzitutto, che un web server è un programma che si occupa di ascoltare un canale di comunicazione per intercettare una richiesta da servire, secondo lo schema di figura.

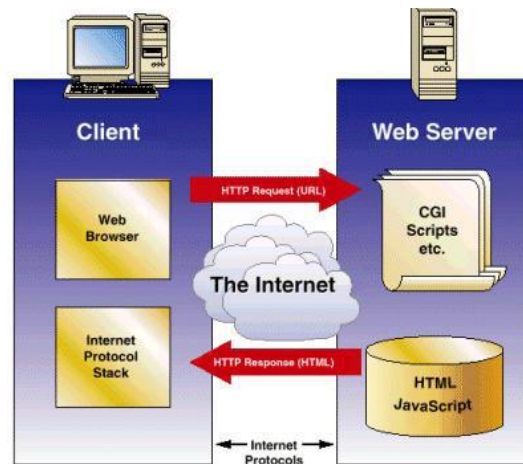


Figura 3 Semplice schematizzazione dell'interazione client-web server.

Il client, utilizzando un browser, lancia una messaggio di richiesta HTTP, contenente la URL, attraverso il collegamento di rete al web server. Questo, catturata la richiesta, risponde, sempre attraverso il protocollo HTTP, con una pagina HTML con il contenuto informativo desiderato dal client.

Apache HTTP Server, o più comunemente Apache è il nome della piattaforma server web modulare più diffusa⁶. Sviluppata dalla Apache Software Foundation è in grado di operare dai sistemi operativi UNIX-Linux a Microsoft.

Apache è un software che realizza le funzioni di trasporto delle informazioni, di internetwork e di collegamento, ha il vantaggio di offrire anche funzioni di controllo per la sicurezza come quelli che compie un proxy. E' un web server stabile e sicuro, supporta l'integrazione con i principali RDBMS e con diversi linguaggi di programmazione, inoltre è facilmente espandibile grazie alla gran quantità di moduli dalle funzioni più disparate.

Operativamente, è composto da un demone, in ambiente UNIX, o da un servizio, in ambiente Microsoft, che sulla base delle impostazioni contenute nel file di configurazione *httpd.conf* permette l'accesso a uno o più siti, gestendo varie caratteristiche di sicurezza e potendo ospitare diverse estensioni per pagine attive (o dinamiche), come PHP.

Il web server Apache presenta un'architettura modulare, quindi ad ogni richiesta del client, vengono svolte funzioni specifiche da ogni modulo di cui è composto, come unità indipendenti. Ciascun modulo si occupa di una funzionalità, ed il controllo è gestito dal core.

⁶ Il grande successo di diffusione di questo software è l'indicatore più chiaro della qualità e dell'affidabilità di questo prodotto: secondo un'indagine Netcraft del 2005, su 75 milioni di siti web, circa 52 milioni utilizzavano Apache, ad ottobre 2006 il numero è salito a 60 milioni.

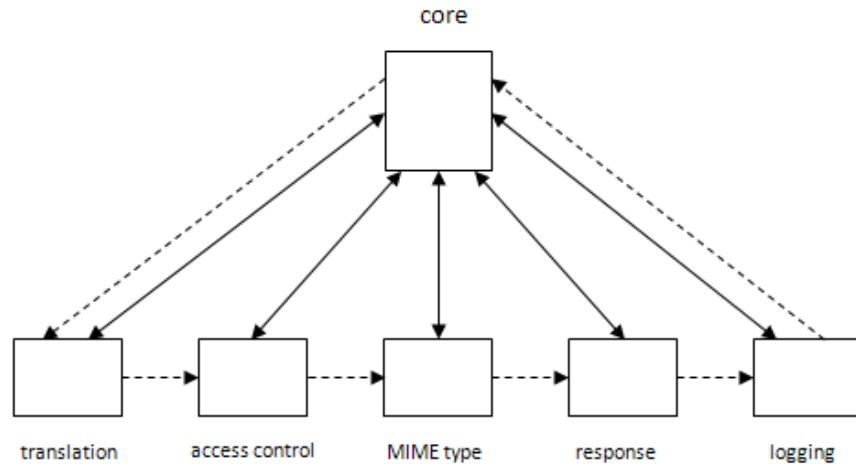


Figura 4 architettura modulare del web server Apache.

Nella figura 3, la linea continua rappresenta il flusso dei dati reale, e la linea tratteggiata il flusso dei dati astratto che forma la pipeline.

I moduli:

- Core: programma principale composto da un ciclo sequenziale di chiamate ai moduli;
- Translation: traduce la richiesta del client;
- Access control: controlla eventuali richieste dannose;
- MIME type: verifica il tipo di contenuto;
- Response: invia la risposta al client e attiva eventuali procedure;
- Logging: tiene traccia di tutto ciò che è stato fatto.

Il core suddivide la richiesta ai vari moduli in modo sequenziale, usando i parametri di uscita di un modulo come parametri di

accesso per l'altro, creando così l'illusione di una comunicazione orizzontale fra i moduli (pipeline software). Sopra il ciclo core c'è un ulteriore ciclo di polling svolto da un demone che interroga continuamente le linee logiche da cui possono pervenire messaggi di richiesta.

2.2 IL DATABASE MYSQL

Un Database (traducibile in italiano come “base di dati”) non è un altro che un insieme di dati logicamente correlati fra loro.

I Data Base Management System (DBMS) sono invece i prodotti software in grado di gestire i database; le loro caratteristiche sono:

- capacità di gestire grandi quantità di dati;
- condivisione dei dati fra più utenti e applicazioni;
- utilizzo di sistemi di protezione e autorizzazione per l'accesso ai dati stessi;

Possiamo identificare diversi tipi di database, in base alla loro struttura logica:

- database gerarchici
- database reticolari
- database relazionali
- database ad oggetti

Il modello gerarchico, basato su strutture ad albero nelle quali ogni dato che non sia a livello radice ha uno e un solo padre, è quello che ha conosciuto il maggior utilizzo fino agli anni '80.

Il modello reticolare deriva da quello gerarchico, rispetto al quale supera la rigidità della struttura ad albero nell'interdipendenza dei dati, ma la cui complessità ne ha impedito una larga diffusione.

Il modello relazionale organizza i dati in tabelle, basandosi sulle relazioni fra essi. Il modello ad oggetti infine, il più recente, estende i concetti del modello relazionale adattandoli alla programmazione ad oggetti.

I database di tipo relazionale sono, attualmente, di gran lunga i più diffusi. Il modello relazionale è stato introdotto nel 1970 dal britannico Edgar F. Codd col documento “A Relational Model of Data for Large Shared Data Banks” (modello di dati relazionale per grandi banche dati condivise). Questo modello si basa sulle relazioni fra i dati, i quali vengono presentati in forma tabulare, cioè come un insieme di tabelle ciascuna composta da righe e colonne. Possiamo affermare che:

- ogni tabella contiene i dati relativi ad una entità;
- le colonne della tabella rappresentano i campi, ovvero le proprietà o attributi dell'entità;
- le righe della tabella esprimono le ricorrenze dell'entità.

Insieme al modello relazionale è stato introdotto il linguaggio SQL (Structured Query Language), che consente di operare sui dati tramite frasi che contengono parole chiave prese dal linguaggio corrente (ovviamente della lingua inglese!).

Del linguaggio SQL sono stati pubblicati tre standard, l'ultimo dei quali risale al 1999. Naturalmente, visto l'ampio successo dei database relazionali, sono molti gli RDBMS presenti sul mercato:

tra i principali, citiamo IBM DB2, Oracle, Microsoft SQL Server, Sybase, Filemaker Pro, Microsoft Access, Informix, PostgreSQL, SQLite, e naturalmente MySql sviluppato dalla compagnia svedese MySQL AB, che è il database utilizzato nella nostra applicazione. Alcuni di questi sono software proprietari, mentre altri fanno parte della categoria open source: questi ultimi, fra quelli citati, sono MySql, PostgreSQL e SQLite.

Ovviamente ogni software ha le sue caratteristiche, e la grande diffusione dei DB relazionali portò l'inventore del modello a definire, nel 1985, una serie di regole ("le 12 regole di Codd"), alle quali un DBMS dovrebbe attenersi per potersi considerare Relazionale. In effetti una base di dati è definita da un insieme di regole che specificano il modo in cui lo stream informativo debba essere registrato. Esso si occupa, da un lato di gestire la struttura e la coerenza dei file su disco, e dall'altro di rispondere alle interrogazioni che l'utente o un'applicazione gli pongono secondo un linguaggio di interrogazione standard (SQL, Standard Query Language).

Interpretando rigidamente queste regole, sono ben pochi i sistemi che potrebbero propriamente fregiarsi di questo titolo. Anche a MySql, soprattutto nelle versioni non recentissime, mancano alcune funzionalità come le viste e l'integrità referenziale, a causa delle quali si può sostenere che non sia del tutto corretto considerare MySql come un RDBMS; qualcuno infatti preferisce definirlo "un DBMS con supporto SQL". In sostanza, quello che già era il più diffuso database open source diventa un vero e proprio DBMS di

livello enterprise. MySQL è in grado di gestire database in diversi formati. I principali sono:

- myISAM, implementazione dei B-alberi, è il formato nativo ed include molti tool di ottimizzazione e backup;
- innoDB, supporta le transazioni e le foreign keys;
- BerkeleyDB, con supporto per le transazioni ma poco flessibile con basse prestazioni.

Il database MySQL è formato da un certo numero di programmi. Fra questi, il principale è naturalmente mysqld, cioè il server vero e proprio. Tutti i programmi MySQL si lanciano di linea di comando digitando il nome del programma seguito da eventuali opzioni e altri argomenti. Gli argomenti che iniziano con un dash (-) sono relativi ad opzioni; gli altri forniscono informazioni addizionali al programma.

Le opzioni possono essere specificate in diversi modi:

- sulla linea di comando di seguito al nome del programma
- in un file di opzioni che il programma legge all'avvio
- nelle variabili di ambiente

Questi tre elementi vengono letti nell'ordine inverso a quello che abbiamo appena esposto, e quelli letti per ultimi prevalgono sui precedenti. Quindi la riga di comando può essere utilizzata per modificare impostazioni di default presenti ad esempio sui file di opzioni.

2.3 IL LINGUAGGIO PHP

Il PHP è l'acronimo di (*Hypertext Preprocessor*), attualmente è uno dei linguaggi di programmazione per il web più diffusi su internet. Lo scopo del linguaggio è quello di consentire agli sviluppatori web di realizzare in modo veloce pagine dinamiche. Per pagine dinamiche si intendono pagine il cui contenuto viene, almeno in parte, generato nel momento in cui le stesse vengono richieste al web server. Un esempio classico di pagina dinamica è fornito dai ben noti motori di ricerca: i risultati che ci vengono restituiti a seguito di una interrogazione non sono pagine web statiche, bensì documenti generati “su misura”, sulla base della nostra richiesta. Pertanto il PHP è un vero e proprio linguaggio di programmazione; è importante quindi rendersi conto della differenza con l'HTML, che non è un linguaggio di programmazione, ma un linguaggio di descrizione e formattazione del testo. Dalla documentazione ufficiale il PHP viene definito come un «linguaggio script dal lato server immerso nel codice HTML». In seguito vedremo nel dettaglio questa caratteristica di PHP chiamata “HTML embedded”. Il codice PHP inserito tra i marcatori HTML viene interpretato dall'elaboratore server e non dal navigatore del client. Il server elabora tramite l'interprete PHP gli script i quali generano le pagine HTML che vengono fornite al client tramite il protocollo HTTP. Dunque il PHP permette di rendere dinamiche le pagine HTML dal lato server. La figura seguente mostra i passaggi eseguiti dal server per rispondere alla richiesta del client.

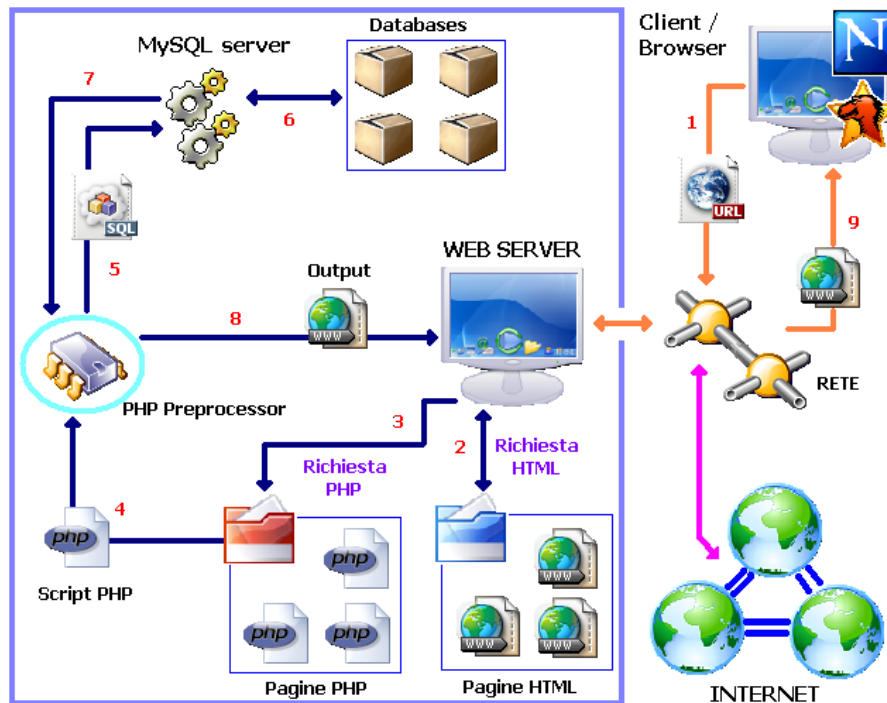


Figura 5 (Flusso dati di un web server con php)

All'arrivo della richiesta (1) il server esegue un controllo sull'URL e decide se prelevare una pagina HTML statica (2) oppure se eseguire uno script per la generazione di una pagina dinamica (3). In questa ultima eventualità il codice viene passato al PHP preprocessor che lo esegue, con eventuali richieste al server SQL (5, 6, 7) ed ai file systems, e genera in output una pagina statica (8). La pagina finale (prelevata dal file system o generata dallo script) viene restituita in risposta al client (9).

2.3.1 LE CARATTERISTICHE DEL LINGUAGGIO

La definizione “ufficiale” del linguaggio PHP, tratta dal manuale in lingua inglese su www.php.net, chiarisce le caratteristiche peculiari di questo linguaggio, e precisamente:

- Il PHP è un linguaggio di scripting;
- Il PHP è un linguaggio HTML embedded;
- Il PHP opera server-side, cioè lato server.

Vediamo, uno per volta, il significato di questi punti.

2.3.2 IL PHP È UN LINGUAGGIO DI SCRIPTING

I programmi scritti in linguaggio PHP, denominati brevemente *script*, vengono eseguiti tramite un apposito software, “l'interprete” PHP. Quest' ultimo si occupa di leggere il codice PHP e, *interpretandone* le istruzioni, esegue le operazioni corrispondenti (ad esempio la lettura di un file o un calcolo aritmetico). Dunque il PHP è quello che tecnicamente si definisce un *linguaggio interpretato* ed in questo esso si differenzia da altri linguaggi di programmazione, come ad esempio C++ e Java, il cui codice sorgente, per poter essere eseguito, deve prima essere compilato (tradotto cioè in codice macchina).

2.3.3 È UN LINGUAGGIO "HTML-EMBEDDED"

Questa caratteristica si riferisce al fatto che il codice PHP è *immerso* nell'HTML; gli script sono inseriti, in altre parole, nelle pagine HTML in cui devono produrre i loro effetti. Il web server riconosce le pagine PHP, distinguendole da quelle "statiche", sulla base dell'estensione, che non sarà la solita .htm o .html ma piuttosto .php, .php3 o simile; quando il server riconosce una estensione associata a PHP passa il testimone all'interprete, lasciando che sia quest'ultimo ad occuparsene.

2.3.4 OPERA SERVER-SIDE

Il PHP opera lato server. Ciò vuol dire che tutta l'elaborazione di uno script avviene sul server, *prima* che questi spedisca la pagina al browser (il "client"). Di conseguenza, chi accede ad una pagina PHP non ha la possibilità di leggere le istruzioni in essa contenute: essendo state processate, ciò che il client vedrà sarà il *risultato* dell'elaborazione; per riassumere: il client vedrà *cosa* fa lo script ma non *come* lo fa.

L'esempio sopra citato del motore di ricerca è immediato e familiare, tuttavia la necessità di generare automaticamente il contenuto di pagine web si presenta in molte altre situazioni. Si pensi ad un qualsiasi sito internet che fornisce in tempo reale (o quasi reale) gli indicatori dell'andamento del mercato azionario, o i risultati della giornata sportiva, o la situazione del traffico automobilistico, etc.; se non si vuole andare troppo lontano si pensi

al caso più concreto dell'azienda che ha bisogno di pubblicare su web il proprio catalogo di prodotti o servizi; nel nostro caso: sono esigenze non troppo dissimili.

Nel caso specifico di questa tesi la necessità di generare automaticamente delle pagine web è legata alla possibilità di interrogare il database contenente gli atti pubblici (determine, delibere, gare, ecc.) pubblicati dall'ente Comune di Crotona. In questa situazione occorre quindi avere a disposizione uno strumento che ci consenta appunto la comunicazione con un database rendendo il più possibile agevole la manutenzione dello stesso.

Semplice e potente, il linguaggio PHP rappresenta la soluzione ideale ad una vastissima gamma di esigenze, come indiscutibilmente dimostrato, del resto, dalla sua crescente popolarità e dal sempre maggiore numero di applicazioni internet/intranet basate su di esso.

2.4 I VANTAGGI DEL LINGUAGGIO

Il PHP offre una varietà di funzionalità e vantaggi e di seguito ne sono elencati i principali:

- l'interprete PHP (o processore di ipertesto PHP) può essere integrato facilmente nel web server Apache come modulo ed è piuttosto leggero, in confronto ad altri tools;
- è disponibile gratuitamente (i dettagli della licenza si trovano sul sito, oppure nel file LICENSE della distribuzione in sorgente);
- portabilità: utilizzando piccoli accorgimenti PHP è in grado di funzionare su diverse piattaforme (Windows32, Unix, Mac OS X, AS/400...) ed è in grado di appoggiarsi indifferentemente a diversi web server (Apache, Microsoft IIS, Xitami, OpenHTTPd, ecc...) sfruttando l'interfaccia CGI (Common Gateway Interface).
- è già fornito di tutte le librerie per accesso ai database più comuni e di tutte le funzioni per la gestione dei cookies, delle sessioni di navigazione, per l'utilizzo di protocolli di rete, per la generazione dinamica delle immagini e di tutto quello di cui abbiamo bisogno per programmare pagine web dinamiche anche complesse.
- facilita il debugging: fa uscire gli errori direttamente sulle pagine web e/o nell'errorlog, indicando chiaramente il tipo di errore, il percorso completo dello script che ha generato l'errore e il numero di riga su cui si è fermato il parser.

- può essere utilizzato anche come linguaggio di scripting al di fuori del web, ad es. per produrre script destinati ad essere chiamati dalla shell. Basta compilarne la versione CGI.

PHP è la tecnologia alternativa ad ASP (Active Server Pages) di Microsoft con la differenza che mettere su un server web con Linux, Apache, MySQL (o PostgreSQL) e PHP non costa nulla in termini di software. In questa tesi però si è dovuto impiantare Apache, MySQL e PHP su sistema Windows in quanto tutte le apparecchiature del Comune di Crotone girano su piattaforma Microsoft.

2.5 L'INTERPRETE PHP

PHP viene distribuito sia sotto forma di codice sorgente, da compilare, sia in formato binario precompilato; le versioni binarie vengono comunque rese disponibili sia per la piattaforma Microsoft Windows, che per quella Linux in formato rpm. L'installazione dell'interprete PHP può avvenire secondo due approcci alternativi, che differiscono tra loro per la modalità di interazione con il server web. Il primo approccio consiste nell'installazione di PHP sotto forma di applicazione CGI; il secondo nell'installazione sotto forma di modulo del server web. Nell'installazione di tipo CGI, l'interprete PHP verrà invocato dal server web come qualsiasi altra applicazione in grado di comunicare con esso attraverso la

Common Gateway Interface. L'interprete PHP, dunque, sarà un programma *esterno* al server web.

In questa situazione, l'esecuzione di uno script PHP consiste in una invocazione dell'interprete, al quale viene passato il percorso del file da eseguire tramite la variabile d'ambiente `PATH_INFO`.

Questo tipo di installazione è il più semplice ma comporta alcuni svantaggi, sia in termini di prestazioni che in termini di funzionalità. Le prestazioni vengono influenzate negativamente dalla necessità di creare un nuovo processo ogni volta che viene invocato uno script PHP, con un considerevole aggravio del carico di lavoro della macchina server. Inoltre, alcune funzionalità del linguaggio, in particolare quelle che richiedono una interazione più profonda con il server web, sono disponibili solo quando PHP viene utilizzato come modulo. L'interprete PHP può essere integrato nel server web sotto forma di modulo statico o dinamico. Nel primo caso (modulo *statico*) sarà necessario effettuare la ricompilazione del server web ed il risultato ottenuto sarà un nuovo eseguibile, completo di supporto PHP. Nel secondo caso (modulo *dinamico*) l'interprete PHP verrà compilato come libreria a caricamento dinamico: DSO (*Dynamic Shared Object*) su piattaforme Unix e DLL (*Dynamic Link Library*) su piattaforme Windows. Notiamo che l'esecuzione di uno script PHP non richiede l'invocazione di un programma esterno (e quindi la creazione di un nuovo processo, come avveniva nell'ipotesi di installazione CGI) in quanto viene effettuata *all'interno* del server web.

L'installazione di PHP sotto forma di modulo del server web è quella che fornisce le prestazioni più elevate, peraltro senza rinunce in termini di funzionalità disponibili; tale approccio è stato quindi preferito nel contesto dell'applicativo reale.

2.6 ALTRE TECNOLOGIE ACQUISITE: HTML E SQL

Solo grazie al supporto di linguaggi dinamici (PHP, ASP e altri) è possibile creare pagine web interattive. Ma per la realizzazione di questa applicazione è stato necessario acquisire la conoscenza di altre tecnologie, quali: HTML e SQL.

2.6.1 IL LINGUAGGIO HTML

Con il diffondersi del World Wide Web (cui spesso ci si riferisce usando la nota sigla WWW), una delle più diffuse applicazioni disponibili sulla rete Internet, si è sviluppato anche uno speciale linguaggio denominato *Hypertext Markup Language* (HTML) utilizzato per descrivere il contenuto di una pagina Web, ma non per costruirla. Poiché HTML rientra nella categoria dei linguaggi a marcatori, esso è costituito da un insieme di istruzioni (dette marcatori, o tag), che servono a descrivere la struttura del documento. Ciò che si richiede quindi per scrivere un documento in HTML è la conoscenza della sintassi con cui i marcatori vengono introdotti all'interno del documento. Oggi esistono comunque dei

programmi che permettono di creare ed aggiornare pagine Web anche senza scrivere direttamente in linguaggio HTML. Alcuni di questi si usano come semplici programmi di videoscrittura e salvano i documenti direttamente in formato HTML. Fra gli editor HTML troviamo ad esempio Netscape Composer, incluso nel pacchetto di Netscape Communicator, Microsoft FrontPage e Macromedia Dreamweaver; anche con le ultime versioni di Microsoft Word è possibile creare una nuova pagina Web a partire da un documento di Word ed usare numerosi modelli che rappresentano i tipi più comuni di pagine Web. Se quindi viene data la possibilità di creare in via immediata documenti più o meno complessi da pubblicare nel Web anche all'utente non conoscitore della sintassi del linguaggio HTML, d'altro lato è comunque importante conoscere il linguaggio formale o perlomeno i suoi rudimenti, in quanto spesso gli editor consentono di visualizzare direttamente il codice HTML che sta dietro alla costruzione delle pagine Web e può quindi essere molto utile saper modificare direttamente il codice sorgente.

Due sono le caratteristiche fondamentali del linguaggio HTML, riflesse anche dallo stesso nome:

- HTML è un linguaggio ipertestuale che permette di stabilire collegamenti (*link*) all'interno di uno stesso documento oppure fra più documenti nello stesso sito Web o in siti diversi, utilizzando le URL (*Uniform Resource Locators*);
- HTML è un linguaggio a marcatori che incorpora, oltre all'informazione che si vuole comunicare attraverso la

pagina Web, particolari sequenze di caratteri (marcatori, *tag*) che descrivono la struttura e il comportamento del documento.

In un documento HTML troviamo quindi sia il testo (contenuto del documento) sia i marcatori (*tag*) che forniscono le istruzioni per la sua struttura. La stesura di un documento HTML richiede quindi la specifica di alcuni parametri per la formattazione del testo. Il documento minimo in HTML è composto da :

```
<HTML>
<HEAD>
<TITLE> Titolo del documento </TITLE>
</HEAD>
<BODY>
Contenuto del documento
</BODY>
</HTML>
```

I *tag* sono racchiusi tra parentesi angolari < > , controllano il modo in cui il documento viene strutturato e visualizzato da un browser.

In particolare:

- il *tag* <HTML> posto all'inizio del documento e il *tag* </HTML> posto alla fine indicano al browser che l'intero documento è scritto in HTML;
- il titolo del documento viene indicato dal *tag* <TITLE> che marca l'inizio del titolo e dal *tag* </TITLE> che marca la fine del titolo;

- i tag `<HEAD>` e `</HEAD>` definiscono l'intestazione del documento e servono come contenitore per altri tag di intestazione, quali ad esempio `<TITLE>`;
- i tag `<BODY>` e `</BODY>` delimitano l'inizio e la fine del contenuto della pagina Web.

Fra questi *tag* è compreso tutto il testo che compare nella pagina Web, inclusi i vari collegamenti. È possibile riassumere alcune caratteristiche dei *tag*:

- i nomi dei *tag* sono generalmente un'abbreviazione della funzione stessa svolta dai *tag* e possono essere scritti indifferenteemente con lettere maiuscole o minuscole; ad esempio `<TITLE>` e `<title>` rappresentano lo stesso *tag*; per una migliore leggibilità si scriveranno i nomi dei *tag* in maiuscolo.
- Affinché i *tag* non siano ignorati occorre che i caratteri che compongono il nome siano contigui e quindi occorre non inserire spazi nei *tag* o fra le parentesi angolari e il nome dei *tag*. Ad esempio i seguenti *tag* `< HEAD>`, `</ HEAD>` verrebbero ignorati.
- Spesso esiste un *tag* di apertura ed un *tag* di chiusura aventi lo stesso nome. Nel *tag* di chiusura il nome del marcatore è preceduto da "/" (esempio `</TITLE>`).

Per visualizzare una pagina Web scritta in HTML si utilizza un browser, cioè un programma che interpreta e visualizza pagine Web. Quando il browser visualizza il documento i *tag* non appaiono; invece il testo scritto fra il *tag* di apertura ed il *tag* di

chiusura seguirà le istruzioni del *tag* nella visualizzazione del documento con il browser.

2.6.2 IL LINGUAGGIO SQL

La sigla SQL, acronimo di *Structured Query Language* ovvero linguaggio di richiesta strutturato, è ormai diventata sinonimo di linguaggio standard per la definizione, l'interrogazione e l'aggiornamento di database relazionali. Nasce nel 1974 nei laboratori dell'IBM. Nasce come strumento per lavorare con database che seguano il modello relazionale. A quel tempo però si chiamava *Sequel*. Nel 1975 viene sviluppato un prototipo chiamato *Sequel-XRM*; con esso si eseguirono sperimentazioni che portarono, nel 1977, a una nuova versione del linguaggio, che inizialmente avrebbe dovuto chiamarsi *Sequel/2* ma che poi divenne, per motivi legali, SQL. Su di esso si sviluppò il prototipo *System R*, che venne utilizzato da IBM per usi interni e per alcuni suoi clienti. Ma, dato il suo successo, anche altre società iniziarono subito a sviluppare prodotti basati su SQL. Nel 1981 IBM iniziò a vendere alcuni prodotti relazionali e nel 1983 rilasciò DB2, il suo DBMS relazionale diffuso ancor oggi. SQL divenne subito lo standard industriale per i software che utilizzano il modello relazionale. L'ANSI lo adottò come standard fin dal 1986, senza apportare modifiche sostanziali alla versione inizialmente sviluppata da IBM. Nel 1987 la ISO fece lo stesso. Questa prima versione standard è denominata SQL/86. Negli anni successivi si

realizzarono altre versioni, che furono SQL/89, SQL/92 e SQL/2003. Tale processo di standardizzazione mirava alla creazione di un linguaggio che funzionasse su tutti i DBMS (Data Base Management Systems) relazionali, ma questo obiettivo non fu raggiunto. Infatti, i vari produttori implementarono il linguaggio con numerose variazioni e, in pratica, adottarono gli standard ad un livello non superiore al minimo, definito dall'Ansi come Entry Level.

Essendo un linguaggio dichiarativo, SQL non richiede la stesura di sequenze di operazioni (come ad es. i linguaggi imperativi), ma piuttosto di specificare le proprietà logiche delle informazioni ricercate. Esso si divide in tre sottoinsiemi:

- Data Definition Language (DDL), linguaggio di definizione dei dati, viene utilizzato per definire lo schema della base di dati. Le definizioni dello schema vengono memorizzate in una base di dati speciale, detta Dizionario dei dati (*Data Dictionary*). Il DDL opera sia a livello esterno, sia a livello logico, sia a livello interno; in particolare, le istruzioni che definiscono il livello esterno prendono talvolta il nome di *view definition* (definizione delle viste, cioè degli schemi dell'utente), mentre le definizioni che definiscono il livello interno prendono il nome di DCL (*device control language*, o linguaggio per il controllo dei dispositivi).
- Data Manipulation Language (DML), viene utilizzato per manipolare la base di dati, cioè per svolgere le seguenti funzioni: Estrarre informazioni dalla base di dati, cioè

formulare un'interrogazione comunque complessa sulla base di dati ed ottenere la risposta all'interrogazione; modificare il contenuto della base di dati, cioè svolgere operazioni di inserimento di nuovi dati, cancellazioni di dati preesistenti, e modifica di specifici valori di dati preesistenti. I comandi utilizzati sono: *SELECT* per la ricerca, *INSERT* per l'inserimento, *UPDATE* per l'aggiornamento e *DELETE* per la cancellazione.

- Data Control Language (DCL), linguaggio utilizzato per fornire o revocare agli utenti i permessi necessari per poter utilizzare i comandi DML e DDL, oltre agli stessi comandi DCL (che servono per poter a sua volta modificare i permessi su alcuni oggetti).

Una caratteristica importante del linguaggio è il suo grado di proceduralità: un linguaggio è procedurale se il programmatore deve specificare in modo esatto quale metodo di accesso ai dati dovrà essere utilizzato per ritrovare l'informazione; un linguaggio è non procedurale se l'utente deve solo indicare quale informazione dovrà essere ritrovata, e viceversa omettere di indicare il modo in cui l'informazione verrà ritrovata. In tal caso, il sistema determina in modo automatico il cammino di accesso ai dati.

CAPITOLO 3

IL CONTESTO REALE

3.1 LE SPECIFICHE DEGLI UTENTI

Nello sviluppo del progetto e durante le varie fasi di realizzazione, sono state analizzate le necessità che i funzionari e gli amministratori pubblici riportavano ai loro quesiti, allo scopo di risolvere i punti critici che quotidianamente ogni amministrazione locale incontra. Queste analisi hanno permesso lo sviluppo della nostra applicazione web per lo studio di questa tesi, caratterizzata da una totale autonomia di gestione, interfaccia semplificata ed immediatezza nelle fasi di consultazione. Come abbiamo già avuto modo di dire nel capitolo precedente, l'applicazione è stata realizzata su piattaforma Microsoft con l'impiego di strumenti, come PHP per la scrittura del codice, MySQL come sistema di base di dati e Apache come web server. Una caratteristica del PHP è la capacità di gestire direttamente i principali server di base di dati tramite connessioni dirette, senza l'utilizzo di strati intermedi quali, ODBC (anche se gestiti perfettamente) o JDBC; una volta installato l'interprete in modo corretto, si è in grado di connettersi a qualunque base di dati, nel nostro caso MySQL. Il PHP è in grado di gestire MySQL mediante il linguaggio di interrogazione

strutturato SQL, che tramite una sintassi standard permette di effettuare tutte le operazioni necessarie sui dati che saranno trattate più avanti. Una volta creata la base di dati e attivate le politiche di accesso, ciò di cui si necessita per operare sui dati è:

- L'indirizzo IP dell'elaboratore che ospita il DBMS MySQL;
- Nome della base di dati. Ogni base di dati viene identificata attraverso un nome;
- Username. Il nome dell'utente abilitato alla manipolazione della base di dati;
- Password. La parola d'accesso alla base di dati.

Appare chiaro che questi dati sono importanti e vanno tenuti segreti. Tutti i parametri necessari alla connessione sono stati concentrati in un unico file, tale file è incluso all'inizio di ogni pagina dalla quale si voglia effettuare una connessione alla base di dati. Il file che contiene i parametri alla connessione si chiama `ConnMysqlGen.inc.php` e contiene le seguenti righe:

```
$hostname_csqli = "localhost";  
$database_csqli = "krotone";  
$username_csqli = "root";  
$password_csqli = "";
```

E' stato dato questo nome al file per diversi motivi:

- "Conn", indica che si tratta di una connessione;
- "Mysql", che si sta utilizzando come DBMS MySQL;
- "Gen", che si tratta di una connessione generale o principale;

- “.inc”, che ci ricorda che questa pagina non verrà visualizzata direttamente, ma verrà inclusa all’interno di altre pagine.
- “.php”, è stato inserito per motivi di sicurezza. Se qualcuno cercherà di visualizzare questa con il browser, vedrà solo una pagina vuota. Il web server, infatti, grazie a questa estensione, prima di passare la pagina al browser, la farà elaborare dal modulo PHP. Visto che non è previsto nessun output, sul browser verrà visualizzata una pagina bianca.

A questo punto abbiamo tutti i dati necessari per la connessione al database. Questa la possiamo realizzare attraverso la funzione `mysql_connect`;

```
$csql = mysql_connect($hostname_csql, $username_csql, $password_csql) or die("errore di connessione");
```

La funzione `mysql_connect` richiede diversi parametri, di cui generalmente si utilizzano i primi tre, che sono proprio quelli che abbiamo appena definito. Se la connessione ha buon fine ci restituisce un identificatore alla connessione che noi memorizziamo in `$csql`. Questa variabile la utilizzeremo ogni volta che vorremo fare un’operazione sul database. Se la connessione non dovesse andare a buon fine (per esempio se uno dei parametri fosse sbagliato) verrebbe restituito `FALSE`. Quindi abbiamo aggiunto di seguito al `mysql_connect()` “or die(‘errore di connessione’)”; grazie a queste righe se la connessione dovesse fallire, otterremo il messaggio di errore e l’interruzione dell’esecuzione del programma.

Subito dopo le specifiche sulla sicurezza, vedremo nel dettaglio la realizzazione delle due interfacce che costituiscono l'applicazione, quella di front-office e quella di back-office

3.2 SPECIFICHE SICUREZZA

L'applicazione è costituita da due interfacce, una cosiddetta di front-office, visibile a tutti gli utenti, dove è possibile reperire gli atti di proprio interesse; ed una cosiddetta di back-office, dove l'accesso è definito da una figura particolare detta amministratore del sistema, che abbia una adeguata conoscenza informatica, soprattutto nella gestione delle basi di dati e una opportuna conoscenza delle funzionalità dell'applicazione in questione. Questo ruolo di amministratore di sistema è stato attribuito dall'Amministrazione locale a tutto l'organico dell'ufficio C.E.D. (Centro elaborazione dati). Questi sono coloro che provvederanno, su richiesta da parte dei dirigenti, di creare nuovi profili d'accesso all'area privata (back-office) o modificare quelli esistenti.

I dirigenti di settore, dopo aver conferito la nomina di addetto alla pubblicazione e all'aggiornamento degli atti sul web (il regolamento interno che disciplina ed organizza la pubblicazione degli atti sul web, prevede almeno due unità lavorative per ogni settore), comunicheranno all'ufficio C.E.D. i nominativi di cui bisogna creare il profilo. I profili d'accesso sono stati così definiti:

- Addetto alla pubblicazione degli atti, quali: delibere di consiglio, delibere di giunta, determine (che saranno per lo più gestiti dalle due unità preposte dall'ufficio segreteria generale e dattilografia), poi la modulistica, le gare di appalto (che saranno gestite da due unità dell'ufficio gare e contratti), statuti e regolamenti (che saranno gestiti da due unità appartenenti alla direzione generale e alla segreteria generale). Questo profilo d'accesso è identificato dalla sigla WRT. Questi non sono abilitati alla pubblicazione delle notizie o dei comunicati stampa.
- Addetti alla pubblicazione delle notizie e dei comunicati stampa, sono identificati dalla sigla RDR (redattori), ha questo servizio è abilitato tutto l'organico dell'ufficio staff, comunicazioni e segreteria degli organi istituzionali. Questi non potranno pubblicare gli atti sopra indicati.
- Addetti alla pubblicazione di atti e notizie o comunicati stampa. Appartengono a tale profilo i funzionari responsabili del proprio settore e sono identificati dalla sigla WRD. Questi però, non potranno accedere alla notizie pubblicate da operatori RDR o RWR che vedremo di seguito.
- Addetti alla pubblicazione, sia di notizie o comunicati stampa che di atti. Anche per questo profilo gli operatori sono funzionari responsabili di settore. Sono identificati dalla sigla RWR. Questi non possono accedere agli atti pubblicati da operatori WRT o WRD.

Questo per mantenere le competenze principali per le pubblicazioni in due gruppi distinti, sostanzialmente RDR e WRT. Ad esempio: l'accesso alla pubblicazione di una notizia per un operatore WRD è una cosa straordinaria, quindi si dovrà limitare alla sola pubblicazione senza poter modificare e/o cancellare le pubblicazioni effettuate dagli operatori appartenenti agli uffici competenti, RDR o RWR. Naturalmente l'operatore amministratore del sistema, identificato con la sigla ADM, potrà accedere liberamente sia alla definizione della struttura della base di dati, che ai dati in essa contenuti. Comunque, un file di log terrà traccia di tutte le transazioni che saranno effettuate sulla base di dati, quindi, in ogni momento l'amministratore del sistema potrà verificare l'operatore e l'indirizzo IP da dove è stata effettuata l'operazione.

3.3 LA REALIZZAZIONE

Il menu di navigazione nella pagina principale (*index.php*) è stato studiato per poter essere gestito in maniera semplice dall'amministratore del sistema e nel contempo fruito in maniera intuitiva dall'utente. Tutti i contenuti sono stati suddivisi in sezioni o aree per una corretta ed intuitiva visualizzazione. A ciascuna sezione sono state associate delle pagine o moduli dinamici, comprensivi o meno di sottomenu sviluppati all'interno delle pagine dell'applicazione.

Abbiamo tre sezioni o aree:

- 1) Sezione Documenti;
- 2) Area Privata;
- 3) Elenco News.

La “Sezione Documenti” è quell’area di front-office dedicata alla visualizzazione degli atti comunali di interesse pubblico ed è costituita da:

- Delibere di Consiglio;
- Delibere di Giunta;
- Determine;
- Modulistica;
- Gare di Appalto;
- Gare Scadute;
- Statuti e Regolamenti;
- Ricerca Documenti.

Ciascuno di questi atti è messo a disposizione dell’utente che avrà possibilità di scaricare la documentazione nei vari formati stabiliti dall’ente Comune (.doc, .rtf, .txt, .pdf, .zip, .xls, .pps, .ppt), o di visualizzarne i contenuti direttamente sulla pagina richiesta. Vediamo nel particolare cosa visualizzano i links appena elencati. Per quanto riguarda “Delibere di Consiglio, Delibere di Giunta e Determine”, questi estrarranno l’elenco dei documenti relativi a ciascuno, pubblicati nell’arco dell’anno corrente. Lo script PHP che genera questa visualizzazione, dapprima determina la data di sistema, poi prepara un’interrogazione in SQL per il database, impostando appunto come criterio di estrazione, la ricerca dei

documenti pubblicati nell'arco dell'anno corrente. Vediamo la composizione della query SQL:

```
SELECT * FROM DOCUMENTI WHERE (coda='$coda')
AND (data_indizione > ='$data_partenza' AND
    data_indizione < ='$data_finale')
ORDER BY data_indizione DESC, numero DESC;
```

Dove \$coda è una variabile che contiene la tipologia del documento richiesto e cioè, delibera di consiglio, delibera di giunta o determine, mentre la \$data_partenza e la \$data_finale sono le due variabili che contengono rispettivamente la data di inizio anno corrente (es. 01-01-2009) e di fine anno corrente (es. 31-12-2009). Tutto questo per rendere più agevole la consultazione dei documenti (evitando un elenco troppo lungo a video di documenti che renderebbe la consultazione sicuramente meno agevole)⁷ e creando allo stesso un tempo una sorta di archivio virtuale degli atti, consultabile attraverso il modulo “Ricerca Documenti” che descriveremo più avanti.

L'elenco dei documenti visualizzati a video è composto da una tabella che è costituita dalle seguenti colonne o campi:

- Settore: indica il settore di competenza nella quale è stato prodotto l'atto;

⁷ Nell'arco dell'anno 2008 sono stati pubblicati dal Comune di Crotone all'incirca: 120 delibere di consiglio, 500 delibere di giunta, e 1500 determine.

- Numero: indica il numero che insieme all'anno identifica l'atto stesso;
- Oggetto: indica l'oggetto o un estratto dell'oggetto dell'atto in questione;
- Indizione: indica la data di indizione, che nella maggior parte dei casi coincide con la data di pubblicazione dell'atto;
- Scadenza: indica la data di durata della pubblicazione dell'atto. E' proprio attraverso l'uso di questo campo data, che è stata creata la modalità di visualizzazione dei soli documenti prodotti nell'arco di un solo anno, quello corrente, che abbiamo descritto sopra;
- Download: da la possibilità all'utente di scaricare il testo completo del documento in uno dei formati che abbiamo specificato sopra, oppure la semplice visualizzazione del documento nella pagina.

Il link "Modulistica" visualizza l'elenco della modulistica in formato digitale, naturalmente PDF, che l'ente a messo a disposizione degli utenti. Questo a comportato un notevole risparmio economico in termini di volume di carta acquistata e/o costi di tipografia, se pensiamo al fatto che il Comune di Crotone al 31/12/2008 contava circa un centinaio di tipologie di modelli diversi. A differenza delle delibere e determine, l'elenco non sarà il risultato di un'estrazione condizionata dei documenti, ma l'elenco conterrà tutti i modelli pubblicati dall'ente, visto il loro esiguo numero. La composizione della query SQL sarà:

```
SELECT * FROM DOCUMENTI WHERE coda='$coda'  
ORDER BY data_indizione DESC;
```

Dove \$coda è la variabile che contiene la tipologia del documento richiesto, nel nostro caso conterrà “modulistica”.

L’elenco visualizzato è composto da una tabella costituita per lo più dalle stesse colonne della tabella che abbiamo descritto sopra, tranne la colonna “numero” che diventa “tipologia”. L’intestazione della tabella per questa colonna è stata parametrizzata attraverso uno script PHP.

Il link “Gare di Appalto” visualizza l’elenco delle gare e dei bandi di gara che non sono ancora scaduti. Anche qui, uno script PHP determinerà la data di sistema e poi preparerà un’interrogazione SQL per il database, la quale estrarrà tutti i documenti, quindi le gare e i bandi cui la data di scadenza è maggiore o uguale alla data di sistema. Vediamo la composizione della query:

```
SELECT * FROM DOCUMENTI WHERE (coda='$coda')  
AND (data_scadenza>='$data_sistema')  
ORDER BY data_indizione DESC, numero DESC;
```

La variabile \$data_sistema contiene la data di sistema e la variabile \$coda contiene sempre la tipologia del documento richiesto, in questo caso “gare di appalto”. L’elenco che verrà visualizzato ha la stessa costituzione della tabella costituita per la modulistica, anche qui l’intestazione della colonna è stata parametrizzata con una variabile.

Invece il link “Gare Scadute” visualizza l’elenco delle gare e dei bandi di gara scaduti. Anche qui, uno script PHP determinerà la data di sistema e poi preparerà un’interrogazione SQL per il database, la quale estrarrà questa volta, tutti i documenti, quindi le gare e i bandi, cui la data di scadenza, è minore della data di sistema. La composizione della query sarà quindi:

```
SELECT * FROM DOCUMENTI WHERE (coda='$coda')
AND (data_scadenza< '$data_sistema')
ORDER BY data_indizione DESC, numero DESC;
```

La variabile \$data_sistema contiene la data di sistema e la variabile \$coda contiene sempre la tipologia del documento richiesto, in questo caso sarà ancora una volta “gare di appalto”. L’elenco che verrà visualizzato ha la stessa costituzione della tabella visualizzata per le gare di appalto.

Il link “Statuti e Regolamenti” visualizza l’elenco completo degli statuti e dei regolamenti pubblicati dal Comune di Crotona con la stessa metodologia usata per la modulistica. Anche qui l’estrazione non è condizionata, ma visualizzerà la lista completa⁸ dei documenti di tipo statuto o regolamento, visto l’esiguo numero. La composizione della query SQL sarà perfettamente uguale a quella della modulistica:

⁸ Al 31/12/2008 il Comune di Crotona contava all’incirca 100 documenti tra statuti e regolamenti.

```
SELECT * FROM DOCUMENTI WHERE coda='§coda'  
ORDER BY data_indizione DESC;
```

Infatti ciò che determina la differenza è il valore della variabile §coda, che in questo caso sarà uguale a “statuti e regolamenti”.

La “Sezione Documenti” si conclude con il modulo “Ricerca Documenti”, una pagina che permette all’utente di effettuare delle ricerche specifiche di documenti all’interno del database, attraverso l’inserimento di alcuni parametri che stabiliscono i criteri di ricerca, che sono:

- La tipologia del documento cercato (che è un campo obbligatorio);
- Un intervallo di data entro la quale il documento cercato è stato pubblicato (che è anch’esso campo obbligatorio);
- Una parte di testo (una stringa) contenuta nell’oggetto del documento che si sta cercando;
- Il settore di competenza che ha prodotto il documento;
- Infine, un intervallo di numerazione entro la quale si trova, o rientra, il documento cercato.

La selettività del risultato dipenderà naturalmente dall’impostazione di questi parametri di ricerca.

La “Sezione News” visualizza l’elenco delle notizie pubblicate dall’ufficio comunicazioni del Comune di Crotone, ordinate per data (alla più recente), in modo tale da avere in testa all’elenco la notizia più recente. In fase di pubblicazione, quindi nell’area privata, e ad opera di un operatore autorizzato alla pubblicazione, si

decide la lunghezza del testo (cosiddetto strillo) che la notizia avrà. Il sistema in fase di pubblicazione, in automatico, provvederà ad aggiungere in coda al testo il link (continua....) che ci permetterà di visualizzare il testo completo della notizia nello stesso frame dell'elenco.

La sezione dell'”Area Privata” rappresenta l'interfaccia di back-office, dove gli addetti provvederanno alla pubblicazione degli atti, che saranno immediatamente disponibili sull'interfaccia di front-office che abbiamo appena illustrato.

Prima di illustrare l'interfaccia di back-office, analizziamo la progettazione delle funzioni e della base di dati.

Una base di dati è un sistema informativo che deve memorizzare molti dati (testi, numeri, date, ecc.) e permette alcune operazioni fondamentali su di essi, che si possono raggruppare su tre livelli:

- 1) Amministrazione (*DB Administrator*): Progettazione, creazione e distribuzione delle strutture dei dati o dell'intera base di dati;
- 2) Transazione (*Transaction*): Inserimento, aggiornamento, cancellazione dei singoli dati o documenti.
- 3) Interrogazione (*Query*): Consultazione (lettura o download) dei dati o dei documenti.

E' evidente che questi tre livelli presentano importanti differenze anche a livello di sicurezza della base dei dati; il primo livello è il più delicato e andrà riservato a pochissime persone competenti, se non addirittura ad una. Il secondo è anche molto delicato, perché va a modificare la base di dati; quindi solo il personale autorizzato

potrà accedere a questo livello. L'ultimo livello invece è quello delle interrogazioni, può essere esteso all'intera utenza non comportando pericoli per l'integrità della base di dati; è però verosimile che in molti casi sia necessario distinguere ulteriori livelli di sicurezza; alcune informazioni possono essere riservate o segrete.

La teoria generale della base di dati si è progressivamente affinata negli ultimi cinquant'anni; realizzare un programma di base di dati che bene o male funzioni può essere relativamente facile, ma progettare una base di dati robusta e a prova di errori (detti anomalie) è molto, molto meno facile.

Come regola generale, vale quella di iniziare la progettazione al livello più alto e cioè, più lontano dalla macchina e più vicino al problema reale, e di lasciare per ultima la realizzazione fisica del progetto (scrittura del codice...). Si possono distinguere allora tre livelli nella progettazione della base di dati:

- Progettazione concettuale: è il livello più astratto e generale della base di dati; del tutto indipendente dalla macchina, individua tutti i dati, le relazioni tra i dati, i vincoli esistenti, ecc. Lo strumento più usato per la progettazione concettuale sono i cosiddetti Diagrammi Entità-Relazioni.
- Progettazioni logica: a questo livello si individuano le strutture dati necessarie per realizzare il progetto; i tre modelli logici più usati sono: il modello gerarchico, il modello reticolare e quello relazionale che è quello impiegato in questa tesi.

- Progettazione fisica: è il livello più vicino alla macchina ovvero al software utilizzato e deve tenere conto delle sue caratteristiche.

Esistono molti RDBMS per la gestione di basi di dati. In questa applicazione è stato utilizzato MySQL. La sua scelta non è stata casuale ma, in un certo senso, obbligata, visto il successo avuto negli ultimi anni da questo RDBMS in applicazioni Web. Inoltre vanta leggerezza e grandi prestazioni.

Prima di analizzare da vicino la sezione dell'area privata con le sue funzionalità, soffermiamoci sulla progettazione della base di dati.

Durante la fase di progettazione, detta analisi dei requisiti, si è provveduto alla raccolta ed allo studio delle informazioni e delle specifiche proprie della realtà di interesse, nel nostro caso gli atti comunali di interesse pubblico. La seconda fase dell'attività di progettazione consiste nell'utilizzo delle informazioni raccolte alla fine di pervenire alla costruzione di due schemi: il primo relativo alle funzioni, il secondo ai dati. La seconda fase di progettazione è divisa in due parti, che devono procedere parallelamente, alla costruzione di due schemi, relativi all'analisi delle funzioni e all'analisi dei dati.

La progettazione delle funzioni si è basata su quattro concetti principali:

- Le interfacce, che rappresentano gli utenti del sistema informatico, sono quindi l'origine e/o la destinazione dei dati;
- I processi che rappresentano le funzioni del sistema;

- I flussi dei dati, che descrivono sia i dati scambiati all'interno del sistema che quelli scambiati con le interfacce esterne;
- I depositi dei dati, che rappresentano i dati che devono essere memorizzati in maniera permanente all'interno del sistema informatico.

Inoltre nella fase di progettazione si è cercato di rispettare i seguenti requisiti:

- la completezza, che consiste nel non tralasciare nessuna funzione coinvolta nell'applicazione;
- la correttezza, legata ai principi di conservazione dell'informazione e di bilanciamento da seguire durante tutto il processo di costruzione; La conservazione dell'informazione esprime l'impossibilità di creazione o perdita di informazioni nei depositi di dati e l'uguaglianza tra l'informazione presente in ingresso ed in uscita in un dato processo; Il bilanciamento è legato al processo di scomposizione ed esprime l'uguaglianza dei flussi di dati del processo padre e quelli del processo figlio.
- Irridondanza, che prevede l'assenza di funzioni parzialmente o totalmente duplicate;
- Indipendenza funzionale, che prevede la massimizzazione delle connessioni tra le attività svolte all'interno di una funzione e la massimizzazione delle connessioni tra attività svolte da funzioni diverse.
- Infine, indipendenza da aspetti organizzativi e tecnologici.

Si tralascia l'illustrazione degli schemi relativi alla progettazione funzionale dell'applicazione, in quanto si ritiene materia più vicina ad uno studio specifico di basi di dati e non di una applicazione web come ha inteso realizzare questa tesi.

Parallelamente allo sviluppo delle funzioni, la seconda fase di progettazione ha previsto lo sviluppo e la realizzazione degli schemi relativi ai dati coinvolti nell'applicazione. Il modello adottato ai fini della progettazione concettuale dei dati è il modello ER (*Entità Relazioni*). La scelta di tale modello deriva dalle sue qualità di semplicità e potenza espressiva.

Le strutture di classificazione adottate dal modello ER sono:

- Le entità: corrispondono a classi di dati del mondo reale che hanno proprietà omogenee ai fini dell'applicazione;
- Gli attributi: rappresentano le proprietà elementari delle entità; ad ogni attributo è associato un insieme di valori, detto dominio, che rappresenta l'insieme di valori che l'attributo può assumere;
- Le relazioni: corrispondono a classi di fatti del mondo reale che sono significativi ai fini dell'applicazione; tali fatti mettono in relazione due o più entità;
- Le generalizzazioni: mettono in relazione un'insieme di entità, denominate entità figlie, con una entità detta entità padre che di esse è astrazione di generalizzazione. Tra l'entità padre e l'entità figlie è valido il principio di ereditarietà, cioè ogni proprietà dell'entità padre è anche proprietà dell'entità figlie.

- I sottoinsiemi: sono un caso particolare di generalizzazione, quella che sussiste tra l'entità padre e una sola entità figlia.
- I vincoli di integrità: sono rappresentati dalle cardinalità e dagli identificatori. La cardinalità minima (massima) di una entità definita in una relazione è il minimo (massimo) numero di volte che ogni occorrenza dell'entità può essere coinvolta in una occorrenza della relazione. Ad esempio, un identificatore dell'entità A è l'insieme degli attributi ed entità di uno schema che identificano univocamente ogni istanza di A.

Tralasciando anche qui l'illustrazione degli schemi concettuali dei dati, per i motivi già detti, si passa direttamente alla trattazione della progettazione logica dei dati e di quella fisica.

Lo scopo della progettazione logica dei dati è quello di ottenere, attraverso opportune trasformazioni del modello concettuale dei dati, una rappresentazione coerente col modello logico utilizzato dal sistema di gestione di basi di dati: il modello relazionale. Il modello relazionale adotta una rappresentazione tabellare; ogni tabella rappresenta appunto una relazione. Ad ogni tabella corrisponde un archivio diverso; le righe della tabella sono dette tuple, il formato delle quali è dato dalle colonne della stessa. Il modello relazionale diversamente dal modello entità relazione, non permette legami espliciti tra i dati, legami che devono essere quindi realizzati da programmi che utilizzano gli operatori relazionali. Procedendo alla traduzione, in base alle regole già espresse, si

ottiene per la nostra base di dati il seguente schema logico delle tabelle:

- Configurazioni (IdConfigurazioni, DimUploadDocumenti);
Il campo DimensioneUploadDocumenti conterrà il valore della dimensione massima che potranno avere i documenti da allegare alla pubblicazione di un atto, naturalmente nei formati consentiti.

- Documenti (Id_documenti, settore, oggetto, tipologia, data_indizione, data_scadenza, data_pubblicazione, coda, url, editore, numero);

La tabella documenti conterrà i dati inerenti agli atti che l'ente intende pubblicare sul web.

- News (Id_news, titolo, contenuto, data_pubblicazione, editore, visibile);

La tabella news conterrà i dati relativi alle notizie da pubblicare.

- Utenti (Usr_Id, usr_cognome, usr_nome, usr_username, usr_password, rol_code);

La tabella utenti conterrà i dati relativi agli operatori inseriti nel database che possono essere di cinque tipi in base ai cinque tipi di credenziali definiti nella tabella "Regole". Gli operatori sono:

- 1) Utenti ADM⁹ (administrator) che sono abilitati a tutto, quindi creazione, manipolazione e cancellazione delle

⁹ Appartengono a questo gruppo di operatori, almeno due impiegati dell'ufficio C.E.D. (Centro elaborazione dati).

tabelle e dei dati, inserimento di nuovi operatori, creazione e cancellazione di policy per attribuzione di credenziali agli operatori.

- 2) Utenti WRT¹⁰ (writer) che possono operare solo sulla tabella dei documenti, quindi creare nuove pubblicazioni e/o cancellare quelle esistenti.
- 3) Gli utenti RDR¹¹ (redattori) hanno più o meno le stesse funzionalità degli operatori WRT, con la differenza che possono solo operare sulla tabella delle news, quindi creare, modificare e/o cancellare le news.

Gli utenti WRD e RWR¹² possono pubblicare indipendentemente sia documenti che news, mantenendo comunque una distinzione e cioè: il sistema prevederà in automatico, in fase di pubblicazione, l'accesso all'operatore WRD solo ai documenti o alle news che sono stati inseriti nel database da altri operatori appartenenti allo stesso gruppo WRD o WRT (anche l'operatore WRT vedrà i documenti pubblicati da WRD) e non appartenenti ai gruppi RWR o RDR (anche gli operatori RDR vedranno le news pubblicate solo dagli operatori RWR). La stessa cosa, al contrario, vale anche per gli operatori RWR.

- Regole (Rol_Id, rol_code, rol_descrizione);

¹⁰ Appartengono a questo gruppo di operatori, almeno due impiegati per ogni settore.

¹¹ Appartengono a questo gruppo di operatori tutto l'ufficio comunicazioni.

¹² A questi due gruppi sono stati assegnati solo funzionari del Comune e non impiegati, attribuendogli maggiori credenziali. Alcuni documenti, come alcune news, possono essere pubblicate direttamente da questi operatori senza farli passare dagli operatori WRT o RDR.

La tabella regole contiene le policy che attribuiscono all'operatore le credenziali d'accesso per la pubblicazione degli atti ed eventualmente la manutenzione (nel caso si tratti di operatori ADM) di tutte le tabelle.

L'ultima fase consiste nella progettazione fisica, in cui tenendo conto del sistema di gestione di base si dati utilizzato, si descrive la base di dati come un insieme di registrazioni in memoria di massa. A questo livello viene descritta l'allocazione dei dati, cioè la loro distribuzione sui vari supporti che costituiscono la memoria di massa. La struttura è la seguente:

NOME TABELLA: CONFIGURAZIONI;

IDCONFIGURAZIONI	INT(4)	PRIMARY KEY
DIMENSIONEUPLOADDOCUMENTI	VARCHAR(15)	

NOME TABELLA: DOCUMENTI

ID_DOCUMENTI	INT(4)	PRIMARY KEY
SETTORE	VARCHAR(50)	
OGGETTO	VARCHAR(200)	
TIPOLOGIA	VARCHAR(50)	
DATA_INDIZIONE	DATETIME	
DATA_SCADENZA	DATETIME	
DATA_PUBBLICAZIONE'	DATETIME	
CODA	VARCHAR(50)	

URL	VARCHAR(150)	
EDITORE	INT(4)	
NUMERO	INT(4)	

NOME TABELLA: NEWS

ID_NEWS	INT(4)	PRIMARY KEY
TITOLO	VARCHAR(100)	
CONTENUTO	TEXT	
DATA_PUBBLICAZIONE	DATETIME	
EDITORE	INT(4)	
VISIBILE	TINYINT(1)	

NOME TABELLA: SYS_SEC_USERS

USR_ID	INT(4)	PRIMARY KEY
USR_SURNAME	VARCHAR(50)	
USR_FIRSTNAME	VARCHAR(50)	
USR_USERNAME	VARCHAR(60)	
USR_PASSWORD	VARCHAR(50)	
ROL_CODE	VARCHAR(3)	

NOME TABELLA: SYS_SEC_ROLES

ROL_ID	INT(4)	PRIMARY KEY
ROL_CODE	VARCHAR(3)	
ROL_DESCRIPTION	VARCHAR(20)	

Abbiamo detto che al più alto livello tra le operazioni su un database sono le operazioni di amministrazione. Sono queste le operazioni più delicate e quindi riservate a pochissime persone, a quelli che abbiamo definito sopra come ADM (*administrator*). Vediamo adesso da vicino l'interfaccia di back-office. Innanzitutto per accedere all'interfaccia di back-office bisogna avere le credenziali cioè, una username ed una password, che verranno naturalmente fornite dall'amministratore del sistema. La pagina principale dell'area privata contiene diverse funzioni:

- Crea ruoli: permette la creazione di ruoli cioè, inserisce un nuovo utente con l'attribuzione delle relative credenziali;
- Utenti: dove è possibile visualizzare l'elenco degli utenti abilitati, quindi le loro password, con la possibilità di effettuare anche modifiche ed eventuali cancellazioni.
- Log: visualizza un file in formato testo che contiene tutte quelle transazioni che hanno avuto effetto sulla base di dati, quindi pubblicazioni di atti, news, creazione di tabelle, cancellazioni e quant'altro.
- Pubblica, che permette la visualizzazione, pubblicazione e cancellazione degli atti, quali: delibere, determine, modulistica, gare, statuti e regolamenti.
- News: permette la visualizzazione, la pubblicazione, la modifica e/o cancellazione delle news (notizie).

- **Manutenzione:** prevede la manutenzione delle tabelle, quindi la creazione, l'eliminazione dell'intera tabella o lo svuotamento inteso come azzeramento dei dati in essa contenuti. Inoltre permette di impostare nella tabella configurazioni la dimensione massima dei file che potranno essere allegati ai documenti da pubblicare.

L'interfaccia si presenterà in due modalità diverse: se l'operatore è stato autenticato dal sistema come ADM, allora tutte le icone della pagina saranno abilitate, se invece l'operatore è stato autenticato come WRT nella pagina rimarrà abilitata solo l'icona "Pubblica" che gestisce la pubblicazione dei documenti, mentre tutte le altre verranno disabilitate, se invece l'operatore è RDR, rimarrà abilitata solo l'icona "News" e tutte le altre saranno disabilitate. Se gli operatori sono WRD o RWR, le icone che saranno abilitate sono sia "Pubblica" che "News".

Partiamo da un operatore WRT, questi corrisponde a quelli del livello *transaction* e cioè, quelli che vanno a modificare i dati nel database ma senza modificare le strutture dei dati. Le procedure di transazione si dividono in tre grandi famiglie: procedure di inserimento (che consiste nell'aggiungere nuovi dati ad una o più tabelle), procedure di aggiornamento (che consiste nel modificare i dati) e procedure di cancellazione (che consiste nel rimuovere un elemento del database, sempre con riferimento ai dati).

L'unica differenza tra i due operatori WRT e RDR sta nel fatto che vanno ad operare su dati memorizzati in tabelle diverse (per gli

operatori WRT, la tabella di riferimento è documenti, mentre per gli operatori RDR è news).

La funzione “Pubblica” visualizza dapprima l’elenco di tutti i documenti pubblicati. In questa pagina l’operatore può eliminare documenti oppure fare nuove pubblicazioni. Se l’operatore sceglie di fare una nuova pubblicazione dovrà compilare i seguenti campi, senza tralasciarne alcuno:

- File: permette di selezionare il file da allegare alla pubblicazione in uno dei seguenti formati (doc, rtf, txt, pps, ppt, pdf, xls, zip); la dimensione dell’allegato non deve superare la dimensione massima raccomandata nel messaggio scritto in rosso sul lato sinistro della finestra, impostato naturalmente dall’amministratore del sistema.
- Settore: qui va descritto il settore competente nella quale è stato prodotto l’atto che va pubblicato, ad esempio: L’approvazione del verbale della conferenza dei servizi per il cambio di destinazione d’uso di un’area, è prodotto nel settore “Urbanistica”.
- Tipologia: qui va descritto la tipologia di documento, ad esempio: “Modulistica”, “Statuto”, “Gara di appalto”, ma potrebbe anche essere “Allegato ad una Determina” oppure “Verbale di una seduta di Consiglio”. Possiamo dire che è un campo più aperto (a testo libero) rispetto al prossimo campo che vedremo più avanti, “Associa”.

- Numero: questo campo ha una sua utilità solo se il documento pubblicato è identificato da un numero come ad esempio una delibera o una determina.
- Oggetto: questo campo riporta tutto o una parte dell'oggetto che richiama il documento inserito.
- Indizione: il valore di questo campo andrà a specificare¹³ la data in cui il documento avrà esecutività.
- Scadenza: questo valore indica, solo per alcune tipologie di documenti, la data in cui il documento non sarà più esecutivo¹⁴.
- Associa: infine questo campo indica, come abbiamo detto sopra, una tipologia di documento più ristretta, infatti deve essere selezionata tra quelle disponibili e non può essere scritta a campo libero.

In questa pagina non verranno visualizzati i campi “editore” e “data di pubblicazione” che saranno inseriti in automatico nella tabella documenti con i seguenti possibili valori: per editore “1” se l’operatore che pubblica è ADM, “2” se l’operatore che pubblica è WRT, “3” se l’operatore è RDR, “4” per l’operatore WRD e “5” per l’operatore RWR, la data di pubblicazione invece conterrà la data di sistema.

¹³ Diventa un campo estremamente importante per alcune tipologie di documenti, tipo gare e bandi di gara, mentre per gli altri documenti solitamente intende la data di pubblicazione.

¹⁴ Ad esempio, per le gare e i bandi di gara indica la data entro la quale il documento perde i suoi effetti, mentre per gli altri documenti come ad esempio delibere e determine indica la fine nell’anno nella quale il documento è stato pubblicato. In questo modo nell’anno nuovo il documento non sarà più direttamente visibile, ma l’utente potrà consultarlo attraverso l’apposito modulo “Ricerca Documenti”.

L'operatore RDR avrà invece nell'area privata la possibilità di accedere alla gestione delle news. Per quanto riguarda la visualizzazione dell'elenco delle news già pubblicate, vale lo stesso discorso che abbiamo già fatto prima per gli operatori WRT o WRD e cioè, che le news pubblicate da operatori WRD non verranno visualizzati nell'elenco delle news, in fase di pubblicazione, ma soltanto quelle pubblicate da un operatore RWR o RDR. L'operatore potrà visualizzare i documenti già pubblicati ed apportare modifiche su tutti i campi della tabella news, con la possibilità di vedere in anteprima come verrà visualizzata la news nell'homepage. Oppure potrà creare una nuova news inserendo i seguenti campi:

- Titolo: che sarà visualizzato come testata della notizia, racchiude quindi il contenuto della news.
- Contenuto: contiene il testo integrale della notizia.

Nella stesura del contenuto, l'operatore ha a disposizione alcuni strumenti per dare uno stile o una formattazione alla notizia:

il tasto (B) permette di scrivere in grassetto;

il tasto (I) permette di scrivere in corsivo;

il tasto (U) permette di scrivere sottolineato;

il tasto (URL) permette di inserire un *link*;

il tasto (Break) permette di troncare la notizia che sarà visualizzata nell'homepage (creerà una sorta di strillo) aggiungendo un link in coda allo strillo "(continua...)" che

darà la possibilità di visionare il testo completo nello stesso frame.

A questo punto le possibilità sono due: vedere in anteprima come verrà visualizzata la news nell'homepage e successivamente confermare l'inserimento, oppure confermare direttamente l'inserimento.

L'operatore ADM ha invece facoltà di operare senza restrizioni. Quindi potrà fare nuove pubblicazioni di documenti, cancellare quelli esistenti, anche se prodotti da altri operatori, pubblicare, cancellare e modificare le news, anche queste inserite da chiunque. In pratica può fare tutto sul database. Le operazioni specifiche di un operatore ADM sono:

- La creazione dei ruoli, cioè andare a creare degli utenti a cui saranno assegnate delle credenziali sulle operazioni da svolgere. Per creare un nuovo utente l'operatore ADM dovrà inserire alcuni dati: il nome dell'operatore, il cognome, la username, la password, ed infine il ruolo che stabilisce le credenziali d'accesso: ADM, WRT, RDR, WRD e RWR.
- Utenti: funzione che permette di gestire gli utenti creati, verificare le loro username e le password, modificare i loro dati e la loro abilitazione ed eventualmente eliminarli completamente dal database.
- Log: questa funzione permette solo all'operatore ADM di verificare tutte le transazioni che sono state fatte sulla base

di dati, quindi inserimenti, cancellazioni, modifiche, ecc. Il sistema provvederà a tenere traccia dei seguenti parametri: operatore, indirizzo IP della macchina dalla quale è stata effettuata transazione, la data e l'ora ed infine la descrizione della tipologia di operazione, vediamo una riga di esempio:

```
operatore: federico -  
ip macchina: 127.0.0.1 -  
data: 17/12/2008 12:33:42 -  
Operazione: Pubblicazione Documento -  
(Oggetto: Pubblicazione gara d''appalto.)
```

- **Manutenzione:** questa funzione permette all'operatore ADM di: in fase di start up creare le tabelle necessarie, e successivamente di svuotarle o eliminarle completamente dal database. La creazione delle tabelle avviene attraverso il linguaggio di definizione dei dati DDL. Come abbiamo visto nel capitolo precedente sono i comandi DDL a definire la struttura del database e quindi dei dati ivi contenuti. Vediamo nel dettaglio i comandi DDL che creano la struttura delle tabelle:

```
CREATE TABLE 'CONFIGURAZIONI '  
( 'IDCONFIGURAZIONI' INT(4) UNSIGNED NOT NULL  
AUTO_INCREMENT,  
'DIMENSIONEUPLOADDOCUMENTI' VARCHAR(50) NOT NULL  
DEFAULT' ',  
PRIMARY KEY ('IDCONFIGURAZIONI'))
```

```

TYPE=MYISAM AUTO_INCREMENT=1;
CREATE TABLE 'DOCUMENTI'
('ID_DOCUMENTI' INT(4) UNSIGNED NOT NULL
AUTO_INCREMENT,
'SETTORE' VARCHAR(50) NOT NULL DEFAULT ' ',
'OGGETTO' VARCHAR(200) NOT NULL DEFAULT ' ',
'TIPOLOGIA' VARCHAR(50) NOT NULL DEFAULT ' ',
'DATA_INDIZIONE' DATETIME DEFAULT NULL,
'DATA_SCADENZA' DATETIME DEFAULT NULL,
'DATA_PUBBLICAZIONE' DATETIME DEFAULT NULL,
'CODA' VARCHAR(50) NOT NULL DEFAULT ' ',
'URL' VARCHAR(150) NOT NULL DEFAULT ' ',
'EDITORE' INT(4) NOT NULL DEFAULT '0',
'NUMERO' INT(4) NOT NULL DEFAULT '0',
PRIMARY KEY ('ID_DOCUMENTI'))
TYPE=MYISAM AUTO_INCREMENT=1;

```

```

CREATE TABLE 'NEWS'
('ID_NEWS' INT(4) UNSIGNED NOT NULL AUTO_INCREMENT,
'TITOLO' VARCHAR(100) DEFAULT NULL,
'CONTENUTO' TEXT DEFAULT NULL,
'DATA_PUBBLICAZIONE' DATETIME DEFAULT NULL,
'EDITORE' INT(4) DEFAULT NULL,
'VISIBILE' TINYINT(1) DEFAULT NULL,
PRIMARY KEY ('ID_NEWS'))
TYPE=MYISAM AUTO_INCREMENT=1;

```


Per la tabella “Utenti” subito dopo i comandi DLL vengono eseguiti anche dei comandi DML per la definizione di un utente amministratore di default che avrà i seguenti parametri d’accesso:

username: ADMIN, Password: ADMIN; questo perché, nel momento in cui l’amministratore eliminerà o svuoterà le tabelle “Regole” o “Utenti” perderà anche le sue credenziali di accesso.

```
CREATE TABLE ‘SYS_SEC_USERS’
(‘USR_ID’ INT(4) UNSIGNED NOT NULL AUTO_INCREMENT,
‘USR_SURNAME’ VARCHAR(50) NOT NULL DEFAULT ‘ ’,
‘USR_FIRSTNAME’ VARCHAR(50) NOT NULL DEFAULT ‘ ’,
‘USR_USERNAME’ VARCHAR(60) NOT NULL DEFAULT ‘ ’,
‘USR_PASSWORD’ VARCHAR(50) NOT NULL DEFAULT ‘ ’,
‘ROL_CODE’ CHAR(3) NOT NULL DEFAULT ‘’),
PRIMARY KEY (‘USR_ID’))
TYPE=MYISAM AUTO_INCREMENT=1;

INSERT INTO ‘SYS_SEC_USERS’ VALUES (0, ‘ADMIN’,
‘ADMIN’, ‘ADMIN’, ‘ADMIN’, ‘ADM’);
```

Anche per la tabella “Regole” subito dopo i comandi DLL vengono eseguiti dei comandi DML per la definizione dei valori di default che sono:

```

CREATE TABLE `SYS_SEC_ROLES`
(`ROL_ID` INT(4) UNSIGNED NOT NULL AUTO_INCREMENT,
`ROL_CODE` CHAR(3) NOT NULL DEFAULT '',
`ROL_DESCRIPTION` VARCHAR(20) NOT NULL DEFAULT '',
PRIMARY KEY (`ROL_ID`))
TYPE=MyISAM AUTO_INCREMENT=1;

INSERT INTO `SYS_SEC_ROLES` VALUES (1, 'ADM', 'ADMIN');
INSERT INTO `SYS_SEC_ROLES` VALUES (2, 'WRT', 'WRITER');
INSERT INTO `SYS_SEC_ROLES` VALUES (3, 'RDR',
'REDATTORI');
INSERT INTO `SYS_SEC_ROLES` VALUES (4, 'WRD',
'WRT+RDR');
INSERT INTO `SYS_SEC_ROLES` VALUES (5, 'RWR',
'RDR+WRT');

```

Il modulo manutenzione prevede infine l'impostazione della dimensione massima del file che potrà essere allegato al documento da pubblicare.

CONCLUSIONI

Questa nuova metodologia di gestione degli atti pubblici ha mutato radicalmente la conduzione interna dell'iter, in quanto gli atti vengono acquisiti direttamente in formato elettronico dagli addetti al servizio, che sono incaricati di trasformarli in formato PDF per la successiva pubblicazione sul web. Questo fa sì che all'interno degli uffici vi sia un minore volume di carta prodotta, un abbattimento dei tempi di pubblicazione degli atti con un conseguente risparmio economico. Le caratteristiche innovative del servizio sono rappresentate dai seguenti elementi:

- Tutti gli atti vengono pubblicati in modo integrale (con la sola esclusione di quelli soggetti alla legge sulla privacy);

- Economie interne (gli uffici, tramite la pubblicazione on-line, verificano direttamente l'esecutività con evidenti risparmi sul lavoro e sui tempi);
- Economie esterne (gli utenti, fruitori del servizio, possono consultare gli atti senza doversi recare fisicamente presso l'Amministrazione locale, con risparmi evidenti sui tempi e sui costi di trasferimento);
- Gli utenti, attraverso il web, possono usufruire di una ricerca avanzata sugli atti pubblicati, rintracciando più agevolmente solo quelli di loro interesse.

L'impiego di questo sistema vuole essere, un valido strumento offerto ai cittadini per la consultazione on-line degli atti di pubblico interesse, e per l'Amministrazione locale un passo in avanti significativo verso la piena trasparenza dell'attività dell'Ente.

BIBLIOGRAFIA

- [1] P. Atzeni et al. *Basi di dati: concetti, linguaggi e architetture*. McGraw-Hill, seconda edizione 1999.
- [2] J. Greenspan, B. Bulger. *Sviluppare applicazioni per database con MySQL/PHP*. Coll. ApogeoPro 2001.
- [3] Mark Wandschneider. *Sviluppare applicazioni web con PHP e MySQL*. Apogeo 2006.
- [4] S.Hughes, A. Zmievski. *PHP Soluzioni professionali per lo sviluppatore*. Coll. ApogeoPro 2001.
- [5] *Zervaas Quentin*. *Sviluppare applicazione Web 2.0 con PHP*. Apogeo 2008.

RIFERIMENTI

- [6] <http://www.php.net/>
- [7] www.apache.com
- [8] www.mysql.com
- [9] <http://it2.php.net/manual/en/security.php>