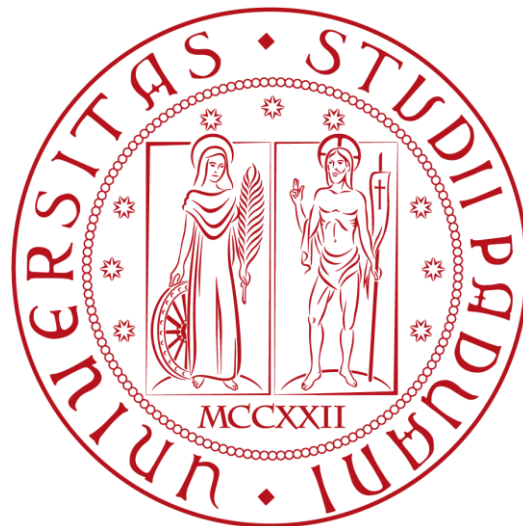


UNIVERSITÀ DEGLI STUDI DI PADOVA

Facoltà di Ingegneria dell'Informazione

Corso di laurea in Ingegneria delle Telecomunicazioni



CREAZIONE E GESTIONE DI SITI INTERNET

Relatore: **Prof. Sergio Congiu**

Tesi di laurea di **Samuele Canova**

Matricola: **579119**

Anno Accademico 2009/2010

Indice

1. INTRODUZIONE	6
1.1 OBIETTIVI TIROCINIO	7
1.2 DESCRIZIONE DELL'AZIENDA	8
1.3 RIASSUNTO LAVORO SVOLTO	11
1.4 STRUMENTI UTILIZZATI	12
2. ASPETTI TEORICI E TECNOLOGICI	13
2.1 ARCHITETTURA DELL'INFORMAZIONE E WEB USABILITY	14
2.2 SISTEMI INFORMATIVI E DATABASE	19
2.3 PROGRAMMAZIONE LATO SERVER	25
2.4 LINGUAGGI LATO SERVER PIÙ UTILIZZATI	28
2.5 EMAIL MARKETING	32
3. RELAZIONE TIROCINIO	39
3.1 TECNOLOGIA LATO SERVER E SOFTWARE	41
3.1.1 ASP.NET	41
3.1.2 SQL SERVER 2005	46
3.1.3 VISUAL STUDIO 2005	50
3.1.4 INTERNET INFORMATION SERVICES (IIS) VERSIONE 6.0	51

3.2 ORGANIZZAZIONE AZIENDALE	55
3.3 REPORT SITI CREATI	57
3.4 REPORT MODIFICHE EFFETTUATE SU OGNI SITO	69
3.5 REPORT EMAIL CREATE	75
3.6 REPORT BACKUP EFFETTUATI	79
4. CONCLUSIONI	81

Elenco delle figure

Figura 2.1: fasi principali per realizzare un sistema informativo	19
Figura 2.2: modello di sviluppo iterativo ed evolutivo di un sistema informativo	20
Figura 2.3: attività principale di trasformazione dell'informazione in un sistema informativo ..	20
Figura 2.4: sistema informativo e sistema informatico	21
Figura 2.5: modello relazionale	22
Figura 2.6: modello gerarchico	22
Figura 2.7: modello reticolare	23
Figura 2.8: modello a oggetti	23
Figura 2.9: livelli di architettura standardizzata per DBMS	24
Figura 2.10: programmazione lato client	25
Figura 2.11: programmazione lato server.	27
Figura 2.12: numero di caselle di posta elettronica abitualmente utilizzato	32
Figura 2.13: numero di mailing list	33
Figura 2.14: i sei passi di una campagna di email marketing	36
Figura 3.1: panoramica del lavoro svolto	40
Figura 3.2: lavori svolti per ogni fiera	41

Figura 3.3: processo di esecuzione delle Master Page	45
Figura 3.4: logo di SQL Server 2005	47
Figura 3.5: logo di Visual Studio 2005	51
Figura 3.6: processo IIS	52
Figura 3.7: interfaccia di amministrazione di IIS	53
Figura 3.8: proprietà delle configurazioni di un sito web in IIS	54
Figura 3.9: procedura di richieste per le attività di Multimedia Tre	56
Figura 3.10: sito Arezzo Sposi in Fiera	57
Figura 3.11: sito Ferrara Balloons Festival	57
Figura 3.12: sito Fiere Sposi	58
Figura 3.13: sito Fantasy&Hobby Genova	58
Figura 3.14: sito MOAC	58
Figura 3.15: sito Ferrara Sposi in Fiera	58
Figura 3.16: sito Multimedia Tre	59
Figura 3.17: sito Fiera Incontro	59
Figura 3.18: sito Venezia Sposi in Fiera	59
Figura 3.19: menù classico dei siti	60
Figura 3.20: suddivisione per fiera delle richieste di modifiche	69
Figura 3.21: slot machine del gioco per vincere il biglietto omaggio	73
Figura 3.22: suddivisione per fiera di email richieste	76

Elenco delle tabelle

Tabella 1.1: manifestazioni fieristiche attualmente organizzate da Multimedia Tre	9
Tabella 2.1: principali problemi dell'usabilità del web	16
Tabella 3.1: informazioni relative all'edizione di una fiera	61
Tabella 3.2: informazioni degli espositori memorizzati nel database	62
Tabella 3.3: informazioni dei visitatori perfetto memorizzati nel database che si scaricano la guida per il matrimonio	65
Tabella 3.4: descrizione arredi e accessori	66
Tabella 3.5: arredi e accessori associati alla fiera	66
Tabella 3.6: descrizione della foto gallery	67
Tabella 3.7: informazioni dei visitatori memorizzati nel database che si scaricano il biglietto ridotto o giocano per vincere il biglietto omaggio	71
Tabella 3.8: conteggio dei tentativi di vincita del biglietto omaggio	73
Tabella 3.9: descrizione di convegni e programmi	75

Capitolo 1

INTRODUZIONE

Con il presente lavoro si intende esporre il tirocinio svolto dal 3 maggio 2010 al 16 luglio 2010 presso Multimedia Tre, azienda specializzata nell'organizzazione di manifestazioni fieristiche in tutta Italia. La struttura, composta da uno staff di 30 esperti suddivisi nelle varie aree aziendali, consente all'azienda di organizzare tutte le manifestazioni, curandone ogni singolo aspetto dalla progettazione alla realizzazione. Ogni manifestazione fieristica è seguita da un project manager e uno o più commerciali a seconda della fiera. Il project manager ha poi a disposizione per lo sviluppo del prodotto, oltre ai commerciali, una serie di servizi interni all'azienda come creatività e grafica, comunicazione e marketing, ufficio stampa e pubbliche relazioni, segreteria, logistica, sviluppo software e internet. Inoltre si avvale di servizi esterni di ricerca in linea con il prodotto fieristico organizzato come allestitore, service audio-video, agenzie di spettacolo.

Lo scopo del tirocinio concordato con il tutor aziendale Luca Bugno, direttore di Multimedia Tre, è stato dedicato a uno dei servizi interni all'azienda che viene messo a disposizione del project manager: lo sviluppo del software e di applicazioni internet. Indispensabile per compiere tale servizio, è stata la familiarità con la programmazione appresa nei corsi di "Fondamenti di Informatica 1" e "Fondamenti di Informatica 2 e architettura degli elaboratori" e poi implementata nella realizzazione di specifici software per il web.

Nell'arco delle 250 ore si è avuta una stretta collaborazione con l'art director graphics Valentina Giora, responsabile che cura il layout grafico di ogni cosa inerente alla manifestazione fieristica: volantini, fax, biglietti, email, cartelli pubblicitari, piantine guida, siti internet, newsletter, loghi, banner, cataloghi.

Si è voluto iniziare il presente lavoro descrivendo gli obiettivi preposti, l'organizzazione aziendale, i compiti effettuati e gli strumenti utilizzati. Poi, viene proposto un inquadramento di tipo teorico relativo alle conoscenze ritenute essenziali, offrendo una panoramica generale dell'usabilità del web, dei sistemi informativi, dei database, della programmazione lato server e dell'email marketing. In particolare, per svolgere il tirocinio, sono state analizzate le tecnologie utilizzate nell'ambiente lavorativo. Il passo successivo è stato quello di approfondire i lavori svolti facendo particolare attenzione a problemi e difficoltà incontrate.

1.1 OBIETTIVI TIROCINIO

Gli obiettivi del tirocinio sono:

- creare siti internet: inserire i contenuti consegnati dal project manager in ogni pagina del sito web adeguandoli al layout grafico concordati con l'art director graphics. Alcune pagine potranno essere dinamiche, in quanto dei contenuti sono collegati a database; altre avranno dei form, interfacce di applicazioni che permettono di inviare email e memorizzare informazioni utili nelle banche dati. Come tecnologia di sviluppo di software per il web si utilizzerà ASP.NET e come Relational Database Management System (RDBMS) si utilizzerà Microsoft SQL Server.
- gestire siti internet: modificare, aggiornare e inserire contenuti nei siti online, facendo particolare attenzione al layout grafico.
- contribuire all'email marketing: creare semplici file HTML (e altri formati) con impostazioni specifiche. Questi verranno successivamente inviati a un certo numero di email presenti negli archivi dell'azienda, facendo particolare attenzione alla gestione della privacy con una procedura di cancellazione adeguata. Per particolari email si controllerà la provenienza da certe fonti, attraverso procedure di monitoraggio.
- fare backup siti internet: eseguire dei backup di siti internet di manifestazioni fieristiche concluse in un server locale dell'azienda, permettendo la visione di questi esclusivamente ai collaboratori di Multimedia Tre.

1.2 DESCRIZIONE DELL'AZIENDA

Dal 1992 Multimedia Tre si occupa di organizzare manifestazioni fieristiche nel territorio nazionale. Le fiere sono considerate come uno strumento di comunicazione dotato di un vantaggio importantissimo anche nei confronti delle comunicazioni online: l'interazione personale. Consente di avere un rapporto diretto con il mercato incontrando contemporaneamente clienti, fornitori e competitor.

L'azienda è suddivisa in vari settori che sono: creatività e grafica, comunicazione e marketing, ufficio stampa e pubbliche relazioni, segreteria, logistica, sviluppo software e internet. Ogni fiera è gestita da un project manager e uno o più commerciali che hanno il compito di seguire ogni passo del processo aziendale. L'obiettivo principale è quello di massimizzare i risultati ottenibili dagli espositori, in termini di raggiungimento del target ricercato, e dai visitatori, in termini di soddisfazione delle aspettative.

Le manifestazioni fieristiche in Italia organizzate nel passato nel corso dell'attività di Multimedia Tre hanno avuto esperienze molto significative, come ad esempio:

- **Crescere per Competere:** ciclo di otto manifestazioni sui temi del lavoro, dell'orientamento e della formazione organizzate per conto del Ministero del Lavoro negli anni 1998 e 1999 a Torino, Parma, Padova, Viterbo, Foggia, Cosenza, Palermo e Cagliari. Circa 1200 espositori hanno partecipato alle otto tappe di Crescere per Competere divisi nelle seguenti aree: area aziende, area istituzionale, area formazione, area orientamento, area studi universitari. Circa 172.000 visitatori, tra cui diplomandi, diplomati, universitari e laureati dai 18 ai 25 anni, si sono recati a questa manifestazione che Multimedia Tre ancora organizza con Incontro a Padova, Vicenza, Pordenone, Bologna e Roma.
- **COM - marketing, comunicazione, eventi:** manifestazione realizzata dal 1999 al 2001 per conto della Fiera di Vicenza sui temi del marketing e della comunicazione di impresa. OmniCom Expo è il nome con cui questa esperienza continua ancor oggi e viene realizzata a Milano dopo una tappa romana nel 2009.

- Big Blu - salone della nautica: manifestazione sulla nautica progettata e realizzata per tre anni dal 2007 al 2009 da Multimedia Tre. Circa 800 espositori e 100.000 visitatori hanno partecipato ad ogni edizione. Tale manifestazione, di cui Multimedia Tre era in parte proprietaria, è stata ceduta al socio di maggioranza Fiera Roma.

Attualmente Multimedia Tre organizza in Italia circa 20 eventi di merceologie diverse destinati sia ad un target business che consumer. Le città che ospitano le manifestazioni fieristiche sono Arezzo, Ferrara, Genova, Milano, Pordenone, Roma, Sanremo, Trieste, Venezia, Verona e Vicenza. Senza tener conto di alcune fiere ancora con data da stabilire, le manifestazioni in corso sono mostrate nella tabella 1.1.

Descrizione	Dove	Quando
	MOAC - Mostra mercato dell'artigianato www.moacsanremo.it	Sanremo 20 – 29 agosto
	Ferrara Balloons Festival - Il festival delle mongolfiere www.ferrarafestival.it	Ferrara 17 – 26 settembre
	Habitat Fiera - Salone dell'arredamento www.habitatfieraferrara.it	Ferrara 23 ottobre 1 novembre
	Sposi in Fiera - Salone per il matrimonio www.ferrarasposinfiera.it	Ferrara 23 ottobre 1 novembre
	Sposi in Fiera - Salone per il matrimonio www.arezzosposinfiera.it	Arezzo 23 ottobre 1 novembre
	Fiera Incontro - Orientamento, formazione e lavoro www.fieraincontro.it	Pordenone 4 – 5 novembre

Descrizione		Dove	Quando
 FIERA DELLA FANTASIA E DELLA CREATIVITA'	Fantasy&Hobby - Fiera della fantasia e della creatività www.fantasyehobby.it	Genova	5 – 7 novembre
 FIERA DELLA FANTASIA E DELLA CREATIVITA'	Fantasy&Hobby - Fiera della fantasia e della creatività www.fantasyehobby.it	Milano	26 – 28 novembre
 LA FIERA SUL LAVORO, LA FORMAZIONE, L'ORIENTAMENTO	Fiera Incontro - Orientamento, formazione e lavoro www.fieraincontro.it	Vicenza	2 – 3 dicembre
 La fiera per il Matrimonio	Sposi in Fiera - Salone per il matrimonio www.veronasposinfiera.it	Verona	27 – 31 gennaio
 La fiera per il Matrimonio	Sposi in Fiera - Salone per il matrimonio www.veneziasposinfiera.it	Venezia	5 – 6 febbraio
	Liberamente Fiera - Salone del tempo libero www.liberamentefiera.it	Ferrara	26 – 27 febbraio
 FIERA DELLA FANTASIA E DELLA CREATIVITA'	Fantasy&Hobby - Fiera della fantasia e della creatività www.fantasyehobbyferrara.it	Ferrara	26 – 27 febbraio
 giardinaggio - orticoltura - arredo esterno	Pollice Verde in Fiera - Salone del giardinaggio www.polliceverdefiera.it	Ferrara	26 – 27 febbraio

Descrizione		Dove	Quando
	Storia in Fiera - Salone delle rievocazioni storiche www.storiainfiera.it	Ferrara	5 – 6 marzo
	Liberamente Fiera - Salone del tempo libero www.liberamentefiera.it	Trieste	26 – 27 marzo
	OmniCom Expo - Idee e prodotti per il marketing www.omnicomexpo.com	Milano	2 – 4 maggio
	DM expo - Direct & relationship marketing www.dmexpo.it	Milano	2 – 4 maggio

Tabella 1.1: manifestazioni fieristiche attualmente organizzate da Multimedia Tre

1.3 RIASSUNTO LAVORO SVOLTO EFFETTIVO

Nell'arco delle 250 ore a disposizione è stato possibile compiere ogni obiettivo concordato con il tutor aziendale. In particolare:

- siti internet creati: www.ferrarasposinfiera.it - www.ferrarafestival.it - www.fieresposi.it - www.multimediatre.com - www.fantasyhobby.it - www.arezosposinfiera.it - www.moacsanremo.it - www.fieraincontro.it - www.veneziasposinfiera.it;
- siti internet gestiti: www.moacsanremo.it - www.omnicomexpo.com - www.dmexpo.it - www.habitatfieraferrara.it - www.ferrarafestival.it - www.fantasyhobby.it - www.fieraincontro.it - www.arezosposinfiera.it - www.multimediatre.com - www.veneziasposinfiera.it - www.ferrarasposinfiera.it;

- email create per il marketing: 6 per Ferrara Festival Balloons, 2 per Fantasy&Hobby Genova, 9 per Habitat Fiera Ferrara, 2 per Fiera Incontro, 2 per Multimedia Tre, 7 per MOAC, 3 per OmniCom Expo, 2 per Arezzo Sposi in Fiera, 3 per Ferrara Sposi in Fiera, 1 per Milano Sposi in Fiera, 3 per Venezia Sposi in Fiera;
- backup fatti: Habitat Fiera Ferrara 2009, Ferrara Sposi in Fiera 2009, Venezia Sposi in Fiera 2009, Verona Sposi in Fiera 2010, Fantasy&Hobby Genova edizione marzo 2010, Fiera Incontro 2009-2010, Fantasy&Hobby Casa Natale Ferrara 2009, Fantasy&Hobby Ferrara 2010, Pollice Verde 2010, Liberamente Ferrara 2010, Liberamente Trieste 2010, Storia in Fiera 2010, Multimedia Tre.

Il lavoro è stato svolto in totale autonomia e le decisioni strategiche sono state studiate assieme al tutor aziendale. Inoltre, le relazioni che si sono create con la segreteria, con i diversi project managers e commerciali e soprattutto con la responsabile della grafica, sono state molto significative per la realizzazione dei vari compiti.

1.4 STRUMENTI UTILIZZATI

Gli strumenti utilizzati durante il tirocinio sono stati predisposti dal tutor aziendale e prevedevano un computer con tutto il software necessario già installato. Le tecnologie adottate sono state stabilite in precedenza dell'inizio del tirocinio, secondo la scelta fatta da Multimedia Tre per lo sviluppo di applicazioni dinamiche per il web e per la gestione di database. In particolare si è scelto di utilizzare ASP.NET come tecnologia per il web e SQL server come soluzione per l'analisi e la gestione dei dati.

Nel computer utilizzato con piattaforma di Windows 2000 Professional, l'ambiente installato prevedeva Microsoft Visual Studio 2005 Professional Edition come ambiente di sviluppo del software; SQL Server 2005 Management Studio per l'accesso, la configurazione, la gestione, l'amministrazione e lo sviluppo di tutti i componenti di SQL Server; FileZilla Client come client FTP e Outlook Express come client di posta elettronica. Inoltre, è stato installato il server web di Microsoft, Internet Information Services (IIS) versione 6.0 per poter testare e controllare il codice prima di metterlo online.

Capitolo 2

ASPETTI TEORICI E TECNOLOGICI

In questo capitolo si vogliono focalizzare gli aspetti teorici necessari per poter compiere gli obiettivi del tirocinio. Poiché i compiti principali assegnati riguardano la creazione e la gestione di siti internet, di seguito si descrive come un web site deve essere organizzato affinché sia accessibile e come l'usabilità del web sia fondamentale per venire incontro ai bisogni degli utenti finali.

Successivamente si parla dei sistemi informativi e come questi vengano collegati ai database, collezione di dati che rappresentano informazioni di interesse. Ogni sito web creato e/o modificato durante il tirocinio, utilizza una base di dati per memorizzare dati, visualizzare informazioni e permettere la gestione di particolari notizie da rendere note ai visitatori.

Non poteva mancare una panoramica sulla programmazione lato server, in cui le elaborazioni delle informazioni contenute nei database, vengono fatte dal server, in base alle richieste dell'utente e successivamente restituitegli.

Un quadro generale si è disegnato per parlare dei linguaggi lato server utilizzati, facendo un approfondimento sulle tecnologie lato server più diffuse come CGI, JSP, PHP, API proprietarie, CFML e ASP.

Infine, si descrive l'email marketing, un tipo di marketing diretto che usa la posta elettronica come mezzo per comunicare messaggi a potenziali clienti e visitatori. Si sono descritte anche le caratteristiche che i file HTML del messaggio email, creati durante il tirocinio, dovevano avere affinché il messaggio sia compatibile e leggibile.

2.1 ARCHITETTURA DELL'INFORMAZIONE E WEB USABILITY

I siti web possiedono una architettura con la quale si interagisce e sono dotati di una struttura logica che permette di trovare le risposte e portare a termine gli obiettivi. Se non sono dotati di tale struttura logica, è difficile navigare all'interno del sito. Quando si parla di usabilità si pensa quasi sempre alla percezione del layout, o, per dirla più semplicemente, all'aspetto grafico. Quando si parla di accessibilità, si pensa invece ad aspetti tecnici, come tempi di caricamento, diversificazione della presentazione dei contenuti per tutti i tipi di supporto tecnologico. L'accessibilità è un requisito indispensabile perché tutti i visitatori devono poter accedere alle informazioni di pubblica utilità. Se i siti, pur essendo costruiti con le tecniche più aggiornate, sono inaccessibili e con tante informazioni da rendere note, diventa impossibile per l'utente arrivare al contenuto al quale è interessato. Ordinare l'organizzazione di un web site e contrastare gli effetti negativi dell'accumulo di informazioni è possibile utilizzando l'architettura dell'informazione: combinazione di organizzazione, categorizzazione e schemi di navigazione all'interno di un sistema di informazioni. Una buona architettura delle informazioni deve cercare di:

- costruire schemi e sistemi di navigazione fissi con elenchi, categorie e sottocategorie in cui fare rientrare ogni elemento. Ha un metodo per organizzare gli schemi e non si affida al caso.
- identificare parole chiavi e associare metadati ad ogni contenuto del sito. Quando due pagine sono associate a uno stesso termine di un vocabolario controllato, allora potranno essere collegate tra loro tramite un link, anche se non appartengono alla stessa categoria di navigazione.
- prevedere metadati aperti. Il sistema deve accettare continuamente parole, nuovi termini correlando tra loro le notizie associate a questi termini.
- tenere conto dei comportamenti dei visitatori, cercando di capire i legami più stretti che i visitatori trovano tra pagine diverse.




Visto il grande numero di siti che si può scegliere, gli utenti hanno sempre meno pazienza per i siti difficili da usare. L'usabilità diventa molto importante per non perdere visitatori e possibili clienti.

L'usabilità è quindi un indicatore di qualità che dice quanto una determinata cosa è semplice da usare.











L'insegnamento fondamentale del web è quello di avere meno di due minuti per dire quello che si vuole dire: ogni pagina deve giustificare la propria esistenza e il tempo che richiede. Se non lo si fa con chiarezza, gli utenti vanno su un altro sito. Elenchiamo di seguito alcuni consigli utili per rendere un sito web il più possibile usabile dagli utenti.














- Non cercare di obbligare gli utenti a entrare dalla homepage perché le pagine interne costituiscono il 60% delle pagine visitate per prime in un sito.
- Dire agli utenti dove sono arrivati e come raggiungere altre parti del sito.
- Ogni pagina del sito deve sempre contenere nome o logo aziendale, link diretto all'homepage e box di ricerca.
- Orientare gli utenti rispetto al resto del sito, includendo anche link ad altre risorse collegati alla posizione attuale senza riempire eccessivamente la pagina con link a tutte le aree del sito.
- L'homepage deve comunicare, entro i trenta secondi che i nuovi visitatori spendono sulla pagina, su che sito sono arrivati, quali sono i servizi e vantaggi che possono trovare, quali sono i prodotti o sviluppi recenti dell'azienda e dare la possibilità di raggiungere la sezione del sito di loro interesse.
- Nelle pagine interne del sito, mettere i link importanti all'interno del contenuto dove è maggiore la probabilità che vengano notati visto che gli utenti dedicano circa 30 secondi a ciascuna pagina interna.
- Progettare pagine brevi perché la maggior parte degli utenti non fa scorrere il contenuto della pagina e se lo fa, lo fa raramente: ogni pagina lunga più di 2-3 schermate rischia di non essere visualizzata completamente.
- Per il testo usare font comuni e una dimensione maggiore o uguale a 10 punti di colore nero su sfondo bianco, cercando di evitare gli sfondi affollati, testo scorrevole, il maiuscolo (che rende più lenta la lettura) e il testo grafico.

I principali problemi che esistono, possono essere suddivisi in 3 gradi di impatto sull'usabilità:

1.  Grave: non può essere trascurato, soprattutto dai designer
2.  Medio: astenersi da errori di questo genere è importante per un designer ma non prioritario
3.  Minore: ricordare questi errori cercando di evitare di generarli, anche se non influenza il successo del sito

La tabella 2.1 cerca di descrivere i principali problemi, analizzandoli. Cerca inoltre di trovare la soluzione da applicare, qualora non sia scontata dalla descrizione del problema.

Impatto	Problema
	Link che non cambiano colore dopo essere stati visitati: possono creare confusione perché gli utenti non capiscono dove si trovano e quali sono le diverse scelte a loro disposizione.
	Pulsante "Indietro" che non funziona: nei browser è prevista la presenza del pulsante "Indietro" che è statisticamente la seconda funzionalità più usata nella navigazione web. Viene considerato come l'ancora di salvezza degli utenti per qualsiasi cosa succeda. Se tolto, la maggior parte degli utenti si sente in trappola e abbandonata.
	Aprire nuove finestre del browser: riduce la capacità dell'utente di tornare sui propri passi, copre la finestra dove l'utente sta lavorando e viola le aspettative dell'utente. Eccezione va fatta per i documenti PDF (Portable Document Format) e file simili.
	Finestre pop-up: agli utenti recano molto disturbo e se compaiono improvvisamente, si discostano da ciò che l'utente sta cercando. Vengono chiusi immediatamente.
	Parti dell'interfaccia che sembrano pubblicità: l'obiettivo dell'utente è cercare informazioni e ignorano qualsiasi altra cosa che possa disturbarli.
	Violazioni delle convenzioni del web: gli utenti applicano al sito le aspettative di funzionamento che hanno visto in altri siti e si confondono se si comporta in modo diverso.
	Contenuti vaghi e sensazionalismo gratuito: gli utenti sono molto impazienti e dedicano poco tempo a ogni pagina. E' fondamentale indicare rapidamente cosa si offre e i vantaggi.
	Contenuti densi e testo non scandibile: i grandi blocchi di testo tendono ad allontanare gli utenti web anche con una sola vista. I testi dovrebbero essere quindi brevi e sintetici.
	Tempi di caricamento lunghi: anche chi ha banda larga, può scaricare solo una certa quantità di dati al secondo. Perciò è buono preoccuparsi di quanto pesa una pagina.
	Frame: cercare di evitarli è buono perché è difficile gestire la stampa e la possibilità di utilizzare il pulsante "Indietro".

Impatto	Problema
	Flash: cercare di evitarli è buono perché fanno perdere tempo e utilizzarli per ravvivare una pagina è sconsigliato perché invece di attirare gli utenti, li allontana.
	Ricerca sul sito con risultati poco rilevanti: dopo la navigazione, la ricerca è il modo più diffuso con cui le persone trovano ciò che desiderano in un sito.
	Multimedia e video: è necessario progettare multimedia adatta al mondo online, dove gli utenti sono impazienti. Il video non dovrebbe superare il minuto per evitare di ridurre l'attenzione.
	Layout congelati: è meglio utilizzare un layout che si adatta alla larghezza della finestra in modo da evitare lo scorrimento orizzontale, odiato dagli utenti.
	Incompatibilità fra piattaforme: i siti internet devono funzionare con browser e computer diversi in modo da evitare che su certe piattaforme il sito non si visualizzi correttamente.
	Cliccabilità incerta: i link che non danno alcuna indicazione di cliccabilità rischiano di non rendere mai accessibili le informazioni perché non raggiunte. Le convenzioni che gli utenti hanno ormai assimilato, prevedono che i link abbiano un testo colorato e sottolineato e gli oggetti grafici un aspetto tridimensionale che assomiglino a dei pulsanti.
	Scorrimento: scorrere tutta la pagina è da evitare perché l'utente decide dove andare basandosi sulla prima schermata e abbandona la pagina senza scorrere tutto il suo contenuto.
	Registrazione: non è un'operazione particolarmente accettata dagli utenti e i siti che obbligano la registrazione perdono potenziali visitatori.
	URL complessi: causano problemi di usabilità e di posizionamento nei motori di ricerca. Ogni pagina dovrebbe avere un URL compreso tra i 20 e i 50 caratteri.
	Menu a tendina e menu gerarchici: più complesso è il menù e più difficile è usarlo perché un elemento dinamico crea confusione agli utenti.
	Plug-in e tecnologie d'avanguardia: essendo migliorati e oggi di facile aggiornamento, diventa più facile utilizzare tecnologie di ultima generazione. Devono essere però disponibili da almeno due anni.
	Interfacce utente 3D: sono troppo difficili da usare e il risultato non vale la fatica perché le immagini tridimensionali vengono visualizzate su una superficie bidimensionale.
	Design barocchi: evitare di avere pagine con troppe cose in movimento, lampeggianti e con grafica inutile. E' meglio comunicare informazioni e sostanza.
	Pagine di benvenuto: devono sparire per non dare l'impressione che il sito tenga più alla propria immagine che all'utilità e per non sprecare tempo ai potenziali clienti.
	Grafica in movimento e testo che scorre: viene considerato dagli utenti come cose inutili e quindi vengono ignorate.
	Controlli di interfaccia fai-da-te: ogni controllo dell'interfaccia dell'utente, come barre di scorrimento o caselle di spunta, devono essere come quelli standard.







Impatto	Problema
	Non rivelare chi è all'origine delle informazioni: fiducia e credibilità sono importanti nel web. Gli utenti devono sapere chi c'è dietro alle informazioni che ci sono nel sito.
	Parole inventate: gli utenti non sanno cosa significano e si allontanano. Inoltre rendono vano il posizionamento nei motori di ricerca.
	Contenuti datati: il problema esiste quando le informazioni datate presente in archivi vengono offerte agli utenti come nuove.
	Siti internamente inconsistenti: evitare che ogni pagina abbia una grafica diversa dalle altre. Danno l'idea che appartengono ad aziende diverse.
	Richieste premature di informazioni personali: prima di rivolgere domande personali, il sito deve costruire una certa credibilità presso l'utente.
	Siti multipli: le informazioni di un'azienda devono appartenere tutte ad un sito web in modo da rafforzarne i contenuti e non su siti con diversi indirizzi.

Tabella 2.1: principali problemi dell'usabilità del web

Alla fine bisogna ricordare che il web viene utilizzato non perché è bello (salvo poche eccezioni) ma perché è utile. Un sito internet se è visivamente gradevole è un bene ma la qualità del contenuto è la cosa migliore da avere.

La regola fondamentale dell'usabilità vale perciò sia per chi scrive i contenuti e sia per il design, ovvero gli utenti preferiscono la chiarezza alla confusione. Il segreto sta proprio nel progettare i siti pensando all'esigenza degli utenti e non al gusto personale.

2.2 SISTEMI INFORMATIVI E DATABASE

Si definisce sistema informativo un insieme di elementi, interconnessi tra di loro, che hanno lo scopo di gestire e organizzare le informazioni necessarie per perseguire gli obiettivi dell'organizzazione. Come mostra la figura 2.1, le fasi principali per la realizzazione dei sistemi informativi sono quattro.

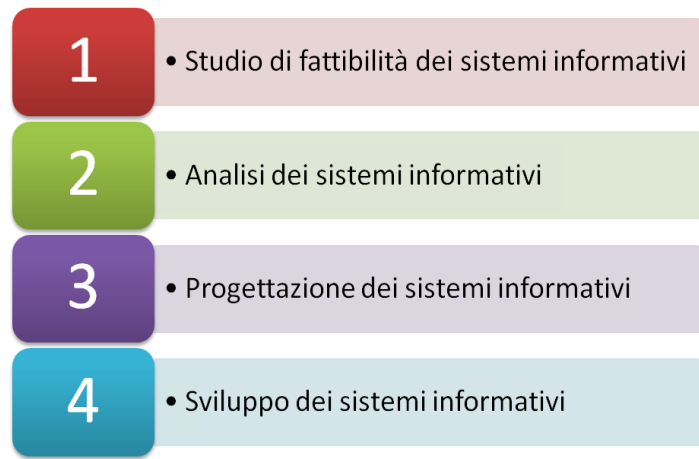


Figura 2.1: fasi principali per realizzare un sistema informativo

In particolare, le quattro fasi prevedono:

1. Studio di fattibilità dei sistemi informativi: valutazione preliminare dei costi utili per un sistema informativo. L'obiettivo è quello di stabilire se è possibile creare il sistema informativo, analizzando le possibili alternative, le scelte ragionevoli e le risorse finanziarie ed umane per la sua realizzazione.
2. Analisi dei sistemi informativi: specificare le caratteristiche di qualità che il sistema informativo dovrà soddisfare, facendo particolare attenzione al problema e alla realtà applicativa, cioè capire come verrà utilizzato e quali saranno le sue funzionalità.
3. Progettazione dei sistemi informativi: soluzione concettuale che soddisfi i requisiti trovati nella fase precedente, definendo come raggiungerli attraverso un'opportuna architettura hardware e software e la relativa implementazione. In questa fase si caratterizza la

capacità del sistema informativo di evolvere in modo affidabile e semplice a seguito di cambiamenti nei requisiti che potrebbero sorgere durante la fase dei manutenzione.

4. Sviluppo dei sistemi informativi: si utilizza un modello di sviluppo iterativo ed evolutivo, come mostra la figura 2.2.

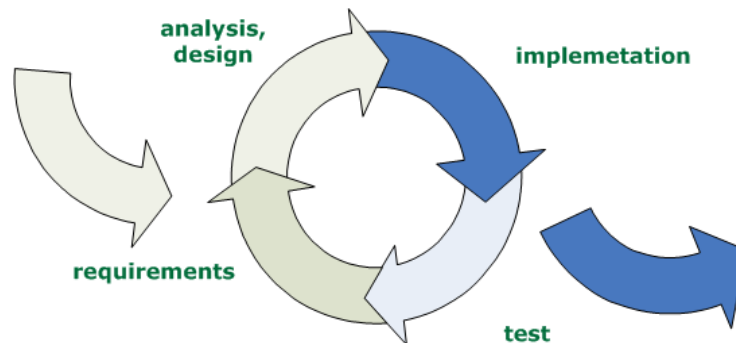


Figura 2.2: modello di sviluppo iterativo ed evolutivo di un sistema informativo

Per ciascuna iterazione si prevede le attività di analisi, progettazione, implementazione software e test. Attraverso molteplici iterazioni, il processo iterativo si basa sul raffinamento e ampliamento del software e del sistema informativo, prendendo ogni volta in considerazione il feedback degli utenti del sistema. Il sistema informativo cresce facendo evolvere il sistema e le specifiche del progetto grazie all'adattamento e al feedback.

L'informazione rappresenta la materia prima che viene trasformata dai sistemi informativi attraverso tre attività principali (in figura 2.3): input, elaborazione e output. Il feedback ha lo scopo di migliorare l'input.



Figura 2.3: attività principale di trasformazione dell'informazione in un sistema informativo

Le informazioni vengono rappresentate e scambiate secondo le tecniche naturali tipiche delle attività umane più semplici. Sono state individuate opportune forme di organizzazione e codifica delle informazioni man mano che i sistemi informativi si sono automatizzati, in modo tale da creare i sistemi informatici, composti da calcolatori, dalle reti informatiche, dalle procedure per la memorizzazione e la trasmissione elettronica delle informazioni. Quindi, la parte automatizzata di un sistema informativo è denominata sistema informatico e fa parte del sistema informativo (figura 2.4).

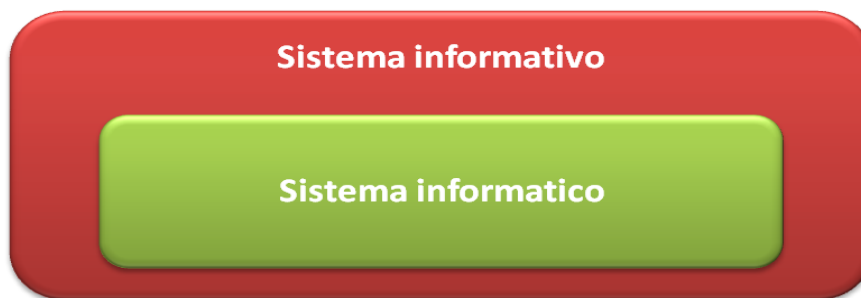


Figura 2.4: sistema informativo e sistema informatico

Per ragioni tecnologiche e alla semplicità dei meccanismi di gestione, le informazioni vengono rappresentate per mezzo di dati che hanno bisogno di essere interpretati per fornire informazioni. I dati da soli non hanno alcun significato ma, una volta interpretati e correlati opportunamente, forniscono informazioni.

Una delle componenti del processo di sviluppo di un sistema informativo è la progettazione di database. Un database, o base di dati, è una collezione di dati utilizzata per rappresentare le informazioni di interesse per un sistema informativo. Un database gode della scalabilità, in quanto riesce a mantenere intatte le proprie prestazioni all'aumentare della quantità di dati, delle ricerche eseguite, delle tabelle e delle funzionalità. In termini più precisi, un database è una collezione di dati gestita da un DBMS (Data Base Management System) che è un sistema software in grado di manipolare collezioni di dati che siano:

- grandi: quantità di dati elevata, in generale molto maggiori della memoria centrale disponibile;
- condivise: applicazioni e utenti devono poter accedere a dati comuni, riducendo così la ridondanza dei dati e conseguentemente la possibilità di inconsistenze;

- persistenti: tempo di vita non limitato a singole esecuzioni di programmi che le utilizzano;
- affidabili: conservare il contenuto dei database in modo intatto in caso di malfunzionamento hardware e software;
- privati: ciascun utente, riconosciuto e con l'autorizzazione per interagire con il DBMS, svolge solo determinate azioni sui dati;
- efficienti: capacità di svolgere le operazioni con un tempo e spazio accettabile per l'utente;
- efficaci: capacità di rendere produttive in ogni senso le attività dei suoi utenti.

Per organizzare i dati di interesse e descriverne la struttura, i DBMS utilizzano dei modelli logici che riflettono una particolare organizzazione. Sono stati definiti quattro tipi di modelli:

- modello relazionale: permette di definire tipi per mezzo del costruttore relazione, che consente di organizzare i dati in insiemi di record a struttura fissa. La relazione è rappresentata attraverso una tabella (in figura 2.5) dove le righe sono i record e le colonne corrispondono ai campi. Definito agli inizi degli anni Settanta e affermatosi negli anni Ottanta, attualmente è il più diffuso.

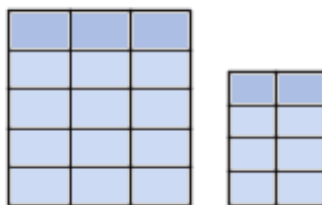


Figura 2.5: modello relazionale

- modello gerarchico: è basato sull'uso di strutture ad albero, nella quale i nodi rappresentano i record e gli archi rappresentano le associazioni (figura 2.6).

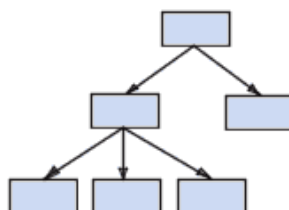


Figura 2.6: modello gerarchico

Definito agli inizi degli anni Settanta, è oggi ampiamente utilizzato. I limiti che presenta il modello gerarchico sono dovuti al fatto che la struttura dei dati creata è rigida e non riesce ad ovviare la ridondanza dei dati.

- modello reticolare: è basato sull'uso di grafi (figura 2.7). E' stato sviluppato dopo il modello gerarchico cercando di risolvere il limite della ridondanza basandosi su due concetti: record e set. Il record è una registrazione logica composta da più campi. Il set è il modo di rappresentare le relazioni esistenti tra i record che formano il database.

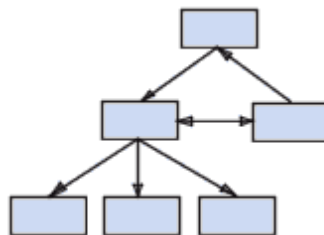


Figura 2.7: modello reticolare

- modello a oggetti: evoluzione del modello relazionale, estende al database il paradigma di programmazione a oggetti (figura 2.8).

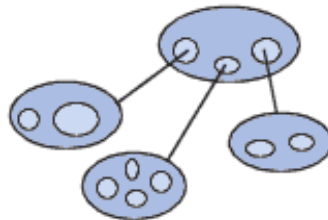


Figura 2.8: modello a oggetti

Nei database esiste una parte sostanzialmente invariante nel tempo, detta schema della base dei dati che è costituita dalle caratteristiche dei dati, e una parte variabile nel tempo, detta istanza della base dei dati che è costituita dai valori effettivi. Lo schema è quindi la componente intensionale del database e l'istanza è la componente estensionale.

Un architettura standardizzata per DMBS prevede di essere articolata in tre livelli in modo tale da garantire l'indipendenza dei dati, principale proprietà dei DBMS. L'indipendenza permette ad utenti e programmi, che utilizzano il database, di interagire a un elevato livello di astrazione che

non dipende dalle caratteristiche con cui la base di dati è stata realizzata e costruita. I tre livelli di architettura, in figura 2.9, sono:

1. schema logico: descrizione del database attraverso il modello logico adottato
2. schema interno: rappresentazione dello schema logico attraverso strutture fisiche di memorizzazione
3. schema esterno: descrizione di una porzione del database di interesse attraverso un modello logico, anche diverso da quello utilizzato per lo schema logico

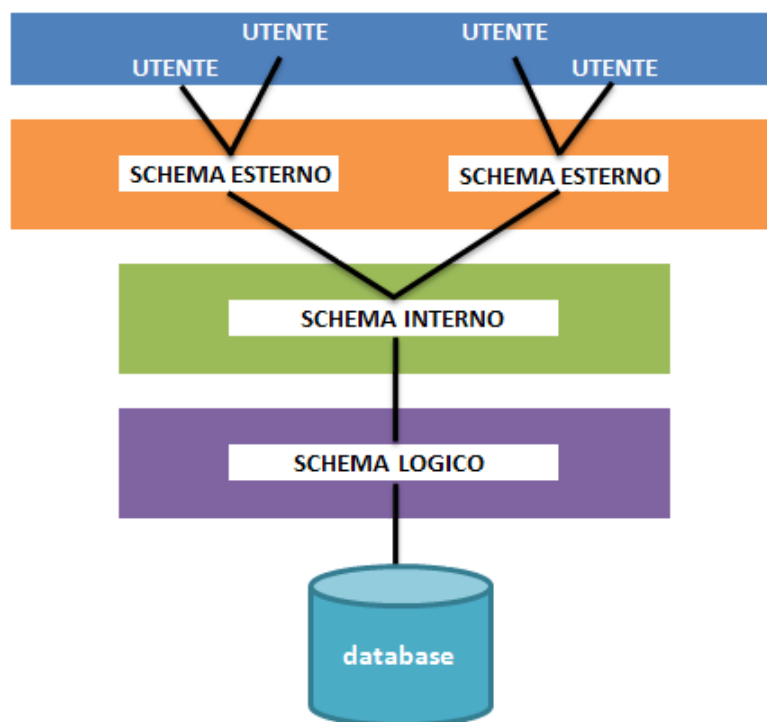


Figura 2.9: livelli di architettura standardizzata per DBMS

I DBMS open source più diffusi sono MySQL, Firebird SQL, InterBase, Ingres, PostgreSQL e quelli commerciali sono IBM Microsoft SQL Server, DB2, Sybase e Oracle.

2.3 PROGRAMMAZIONE LATO SERVER

Dato un ambiente client/server in un'architettura di protocolli TCP/IP (Transmission Control Protocol/Internet Protocol) e HTTP (Hypertext Transfer Protocol), per programmazione lato server si intende lo sviluppo di applicazioni che andranno in esecuzione prevalentemente sul server, accettando richieste del client. Una volta che il server ha elaborato i risultati, fornisce al client le pagine HTML. HTML (HyperText Markup Language) è considerato comunemente un linguaggio di programmazione non WYSIWYG (What You See Is What You Get: quello che vedi è quello che ottieni). Significa che per quanto un programmatore provi il codice HTML nel suo browser, non può sapere con precisione cosa accadrà quando sarà visualizzato in un altro browser. L'unica possibilità che lo sviluppatore ha a disposizione è di testare il proprio codice nella maggior quantità possibile di situazioni.

La programmazione lato server si differenzia dalla programmazione lato client in quanto il linguaggio lato client viene interpretato dal browser dell'utente che si collega, mentre quello lato server viene interpretato dal server e poi restituito in formato HTML al browser. Questa diversità comporta molti vantaggi per la programmazione lato server. Due di questi sono la compatibilità universale dei browser e la dinamicità delle pagine.

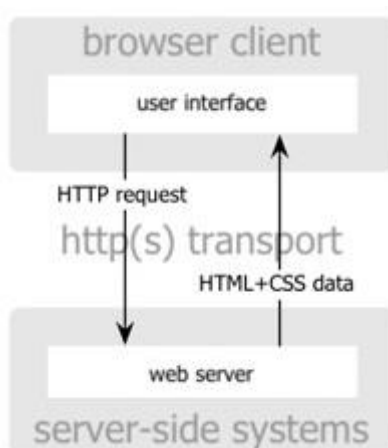


Figura 2.10: programmazione lato client

La programmazione lato utente è un metodo attraverso il quale l'utente richiede al server, attraverso un programma definito browser una determinata pagina web; questa pagina viene fisicamente trasferita dal server al computer dell'utente e grazie al browser viene poi interpretata ed i risultati visualizzati sullo schermo. In definitiva tutti i processi d'elaborazione, la logica, ed il codice vengono eseguiti sul computer in locale, che effettua la richiesta al server, come mostra la figura 2.10.

I principali linguaggi e tecnologie che possono essere utilizzati dal lato client sono:

- JavaScript: linguaggio di programmazione interpretato e leggero, creato dalla Netscape. E' il linguaggio di scripting client side più diffuso ed utilizzato in Internet. Lo svantaggio principale è che uno script può comportarsi in modo differente a seconda del browser in cui viene eseguito. Le funzioni principali di JavaScript sono il controllo dell'aspetto, del contenuto del documento e del browser. Inoltre permette l'interazione con l'utente con i moduli e permette di leggere e scrivere sui cookies (stringhe di testo memorizzate sul computer dell'utente che permettono di mantenere informazioni ottenute nelle connessioni precedenti).
- Java: linguaggio creato da Sun e concepito specificatamente per un utilizzo in ambienti di rete. L'architettura prevede infatti un interprete Java dal lato client (JVM Java Virtual Machine) presente in ogni browser che è in grado di rendere l'architettura indipendente da ogni piattaforma.
- VbScript: linguaggio di scripting adottato dalla Microsoft ed è compatibile solo con Internet Explorer (principale svantaggio) e con Visual Basic. Permette di controllare il browser e altre applicazioni. Inoltre può essere impiegato per controllare oggetti OLE (Object Linking and Embedding) e applet Java contenuti in una pagina web.
- Active X: tecnologia Microsoft che permette di creare controlli in Visual C++ o Visual Basic. E' presente dalle versioni superiori di Internet Explorer 3. Gli altri browser possono visualizzare controlli Active X solo utilizzando appositi plugin (principale svantaggio).

- Flash: plugin con relativo applicativo di sviluppo prodotto da Macromedia, che sta avendo una notevole diffusione. L'applicativo di sviluppo è potente e permette di generare pagine molto leggere, grazie all'utilizzo estensivo di grafica vettoriale.

La programmazione lato server invece prevede il solito meccanismo di chiamata ma i processi, la logica, ed il codice vengono eseguiti nel server. All'utente arriva un file (di solito pagina HTML) contenente la risposta, ovvero le elaborazioni fatte dal server in base alle richieste dell'utente.

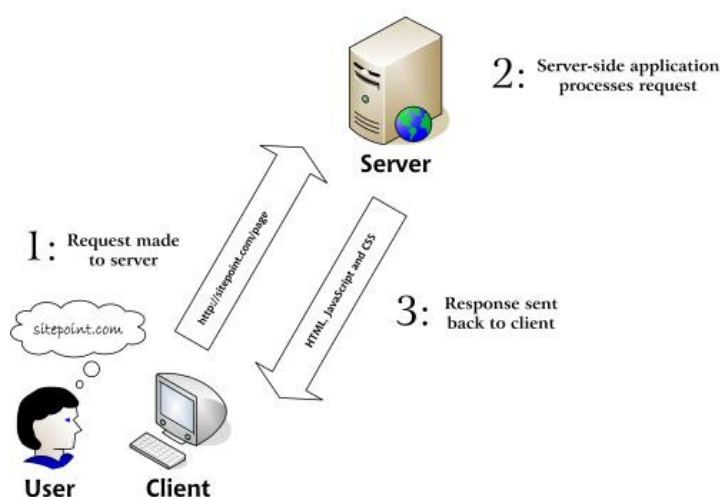


Figura 2.11: programmazione lato server

Un programma scritto con questo tipo di linguaggi viene sempre elaborato sul server, come mostrato nella figura 2.11. Inoltre il programma non viene mai reso disponibile all'utente, il quale può visualizzare solo il risultato del programma. Questo concetto è molto importante in quanto sta alla base della sicurezza e dell'affidabilità offerti dalla programmazione lato server.

Per far eseguire i programmi lato server, ogni web server dovrà essere opportunamente configurato. Quindi ogni web server deve far riferimento alla root virtuale: particolare directory sul disco fisico del web server. L'utente, quando si collegherà al server, non si collegherà alla directory root del server ma ad una sottodirectory fisica del disco che avrà la stessa funzione della root virtuale. All'interno della root virtuale, saranno contenute tutte le pagine HTML accessibili dall'esterno.

Il web server, affinché possa integrarsi con il linguaggio lato server, deve essere configurato adeguatamente per:

- poter riconoscere l'estensione dei file con il programma da eseguire;
- sapere la posizione dell'interprete da mandare in esecuzione per interpretare i linguaggi lato server;
- indicare la cartella del server contenente il file lato server da eseguire.

Ogni linguaggio lato server dovrà inoltre interagire con il server SQL (Structured Query Language) e in particolare avere la possibilità di interagire tramite primitive con particolari DBMS installati sul server.

2.4 LINGUAGGI LATO SERVER PIÙ UTILIZZATI

Le principali e più utilizzate tecnologie lato server sono: Common Gateway Interface (CGI), Microsoft Active Server Pages (ASP), Java Server Pages (JSP), Pre Hypertext Processor (PHP), Application Program Interface (API) proprietarie, Cold Fusion Markup Language (CFML), Server Side JavaScript, FileMaker Pro, Lasso, JavaScript for Databases, Oracle Application Server, Tango. Di seguito descriviamo alcuni di questi appena citati.

CGI

Con l'acronimo CGI (Common Gateway Interface, in italiano Interfaccia Comune di Uscita dal server) si indica una tecnologia lato server che riguarda l'interazione tra web server e differenti fonti di dati come applicativi e database presenti nel web server. E' la prima forma di elaborazione lato server implementata nel 1993 quando il World Wide Web era agli esordi, inizialmente per l'utilizzo con le mailing list. Un programma CGI può essere scritto praticamente in qualsiasi linguaggio di programmazione (ad esempio C, C++ e Perl).

Il principio su cui si basano gli script CGI è il classico meccanismo input-output per la formulazione delle richieste, l'interpretazione di queste ultime e la soddisfazione delle stesse tramite una risposta adeguata, cioè divisa in tre parti:

1. input: utilizzando il proprio browser, l'utente formula una richiesta (link, compilazione form)
2. elaborazione: il web server raccoglie la richiesta effettuata dall'utente e ricerca l'output in grado di soddisfarla
3. output: il web server invia al browser del client la risorsa che soddisfa la richiesta

I comportamenti generati dagli script CGI sono automatizzati, cioè non richiedono l'intervento umano per portare a termine il compito per cui sono stati creati. Possono generare anche comportamenti non richiesti direttamente dall'utente, ma derivanti da azioni predefinite in sede di sviluppo.

JSP

Con l'acronimo JSP (Java Server Pages) si indica una tecnologia lato server basata su Java, il linguaggio creato da Sun. I vantaggi sono ereditati da Java, che garantisce la metodologia object oriented (predisposizione al riutilizzo del codice) e la quasi totale portabilità multiplatforma. Lo svantaggio di JSP è essenzialmente nella lentezza di sviluppo, legata alla complessità della programmazione, in quanto si basa su un insieme di speciali tag e una sintassi specifica. Differisce dalle altre tecnologie non per i contenuti dinamici che si possono produrre ma per l'architettura interna del software che costituisce l'applicazione web e, di conseguenza, sui tempi di sviluppo, la portabilità, la modificabilità, le prestazioni, e altri aspetti di qualità del software. Nel contesto della piattaforma Java, la tecnologia JSP è correlata con quella dei servlet (applicazione Java in esecuzione su una JVM residente sul server). Al momento della richiesta di una pagina JSP, viene eseguito un servlet, che viene creato automaticamente. Nel web server esiste un motore JSP che esegue le operazioni di page translation (creazione del sorgente Java del server corrispondente alla pagina JSP) e di page compilation (creazione del file .class del servlet). Il servlet, una volta creato e avviato, rimane in memoria per servire le richieste successive.

PHP

Con l'acronimo PHP (Pre Hypertext Processor) si indica una tecnologia lato server che utilizza un linguaggio open source completo di scripting, sofisticato e flessibile, che può girare su qualsiasi web server, su qualsiasi sistema operativo e soprattutto di interagire con i principali tipi di database. L'elaborazione di codice PHP sul server produce codice HTML da inviare al browser dell'utente che ne fa richiesta. PHP nasce nel 1994 ad opera di Rasmus Lerdorf come una serie di macro, la cui funzione era quella di facilitare ai programmatori l'amministrazione delle homepage personali(inizialmente PHP appunto Personal Home Page). Oggi PHP permette di raccogliere dati, generare contenuti dinamici, interagire con i cookie e con altri servizi utilizzando i principali protocolli (IMAP Internet Message Access Protocol, SNMP Simple Network Management Protocol, NNTP Network News Transport Protocol, POP3 Post Office Protocol 3). Tra le applicazioni principali si ricorda l'autenticazione degli utenti, la gestione di template, la gestione di file XML (eXtensible Markup Language), la creazione dinamica di immagini e di documenti PDF utilizzando apposite librerie. Inoltre PHP è caratterizzato da una velocità di sviluppo veramente notevole. Azioni come interrogazioni di database o utilizzo di socket TCP possono essere gestite in modo molto semplice, almeno in confronto agli altri linguaggi. Tutti questi motivi, rende PHP uno dei linguaggi più utilizzati in tutto il mondo per la programmazione server side. I problemi principali si trovano al mancato adempimento delle best practice nella programmazione e da vulnerabilità presenti in codice scritto in versioni vecchie di PHP.

API proprietarie

Con l'acronimo API (Application program Interface) si indica un insieme di procedure, protocolli e strumenti per realizzare un'applicazione software. Si dice proprietarie perché Microsoft e Netscape hanno sviluppato entrambe le proprie API per permettere agli sviluppatori di scrivere applicazioni dal lato server come delle librerie, probabilmente per rispondere all'inefficienza di CGI. Queste librerie vengono caricate nello stesso processo del web server e sono in grado di rispondere a richieste multiple senza generare nuovi processi. Lo svantaggio principale delle API proprietarie è che può essere utilizzato solamente nella specifica piattaforma in cui è stato creato e il trasferimento ad un'altra risulterebbe particolarmente difficile.

Inoltre, visto che sono eseguite nello stesso processo del server, l'intero server potrebbe cadere se si verifica una violazione di accesso.

CFML

Con l'acronimo CFML (ColdFusion Markup Language) si indica un linguaggio di markup disegnato nel 1995 da Allaire specificatamente per scrivere applicazioni per il web e le intranet. Include un server e una serie di tool di sviluppo per integrare database e pagine web. I documenti creati con Cold Fusion includono tag scritti in questo linguaggio che semplificano l'interazione con i database ed evitano l'utilizzo di linguaggi più complessi. Lo svantaggio principale è la disponibilità su poche piattaforme (Windows, Solaris, Linux e HP/Ux). Il vantaggio principale invece è la semplicità alla connessione ai database.

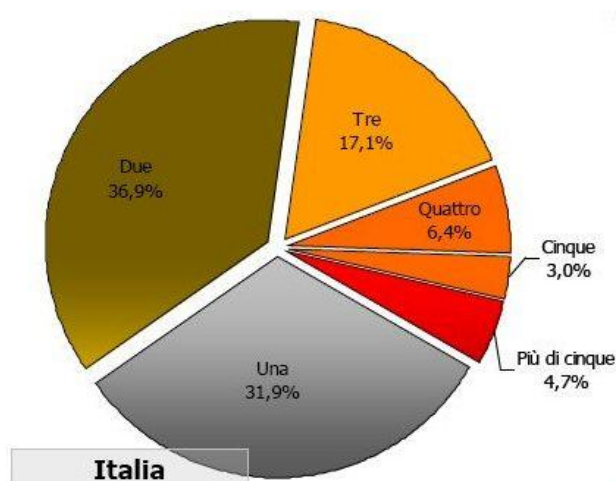
ASP

Con l'acronimo ASP (Active Server Pages) si indica una tecnologia lato server ideata da Microsoft. Una pagina ASP contiene testo, codice HTML e comandi di script, scritti utilizzando VBScript e JScript visto che ASP non è un linguaggio di scripting in sé, ma un ambiente che processa gli script incorporati nelle pagine web. È possibile interfacciare le pagine ASP con qualsiasi tipo di database che abbia un driver ODBC (Open Database Connectivity) per la connessione ai DBMS. Funziona ufficialmente solo sul web server Microsoft Internet Information Services (IIS). Il vantaggio principale è la semplicità e la facilità di comprensione della sintassi di programmazione. Lo svantaggio principale è che l'interprete ASP presenta dei limiti, soprattutto sulle prestazioni. ASP è stato ufficialmente abbandonato in favore di ASP.NET (descritto nel paragrafo 3.1.1) che consente di creare applicazioni web su piattaforma Microsoft che possono sfruttare anche funzionalità avanzate.

2.5 EMAIL MARKETING

L'email è diventata il principale mezzo di comunicazione interaziendale e interpersonale. Non si intende negare l'importanza della comunicazione verbale ma sottolineare che c'è un flusso di informazioni sempre più importante recapitato attraverso l'email.

Con le seguenti informazioni si vuole sottolineare come sia importante utilizzare l'email nella comunicazione aziendale. Una ricerca, condotta da Human Highway, mostra in figura 2.12 il numero di caselle di posta elettronica utilizzate abitualmente.

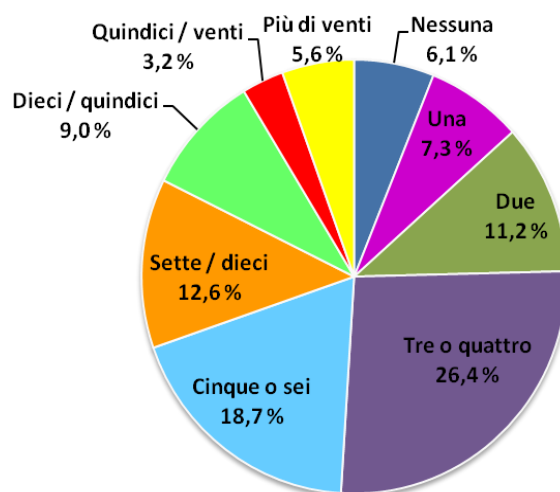


Email Marketing Consumer Report 2009 - Italia

Figura 2.12: numero di caselle di posta elettronica abitualmente utilizzato

Il totale è di 51 milioni di caselle di posta utilizzate dall'utenza Internet Italiana. La media è di 2,30 caselle per utente. Circa il 55% dell'utenza Internet possiede e utilizza regolarmente tra due e tre caselle di posta elettronica.

Alcune delle email che vengono ricevute sono spedite non da persone conosciute, ma da aziende. La figura 2.13 mostra il numero di mailing list alle quali si risulta iscritti.



Email Marketing Consumer Report 2009 - Italia

Figura 2.13: numero di mailing list

Il totale è di 140 milioni di destinatari di email, ovviamente duplicati. La media è di 6,4 mailing list per utente. La quasi totalità degli utenti Internet Italiani dichiara di essere iscritta ad almeno una mailing list.

L'email marketing è una forma di direct marketing che utilizza l'email come mezzo di invio dei messaggi pubblicitari, raccogliendo ogni forma di comunicazione strutturata via email. Dunque non contiene solo i messaggi di direct marketing o promozionali, ma tutte le comunicazioni via email che intercorrono tra una organizzazione e i propri interlocutori. È importante considerare che l'email è un tipo di comunicazione interpersonale e, anche se la si utilizza per lavoro, non può e non deve essere assimilata a una lettera commerciale. I messaggi spediti arrivano direttamente nella casella di posta elettronica del destinatario che la percepisce in modo del tutto differente da quello che avviene con la casella della posta cartacea, del fax o del telefono. Ecco perché l'email è stata fatta oggetto di un'attenzione considerevole dal Garante e dalla normativa sulla privacy, a differenza di quanto succede con la posta tradizionale, il fax e il telefono. Il Garante ha infatti imposto per l'email norme molto più restrittive per quanto riguarda la possibilità di inviare

messaggi pubblicitari rispetto a quanto ha legiferato per gli altri mezzi tradizionali. Attualmente le regole da seguire per l'email marketing, sono quelle contenute nei seguenti testi normativi:

- Codice del Consumo (D. lgs. 2005 n. 206)
- Codice in materia di dati personali (D. lgg. 2003 n. 196)
- Decreto Legislativo 2/2/2002 n. 24 sul commercio elettronico.

Semplificando al massimo, si riassume il tutto così: non è possibile inviare messaggi di posta elettronica a contenuto commerciale a potenziali clienti se non si è ottenuto il loro preventivo consenso espresso a tale attività. Per inviare comunicazioni di carattere commerciale, occorre quindi fare una differenza su l'Informativa sulla Privacy e sul Consenso. Quando si inizia a trattare un dato personale (ogni informazione che sia riferibile ad una persona fisica o giuridica, come l'indirizzo email), occorre mandare l'informativa sul trattamento dei dati personale, sufficientemente inviandola: non è necessario che il soggetto proprietario dei dati esprima un qualche consenso o la rispedisca firmata all'azienda mittente. Per poter inviare email di carattere commerciale, occorre invece ottenere il consenso che deve essere documentabile per iscritto. Quindi per il consenso è necessario che il soggetto dia una esplicita autorizzazione al ricevimento di messaggi promozionali.

Sotto il termine email marketing rientrano differenti tipi di comunicazione via email. Ciò che le accomuna non è tanto il loro obiettivo quanto la modalità con cui sono composte e inviate. Quando un generico invio di email è realizzato con l'intento di fornire un reale servizio a un utente che sappiamo interessato, allora si parla di email marketing. Si possono raggruppare le email in due categorie:

1. email promozionali: hanno un contenuto commerciale. Possono essere di ausilio alle azioni di marketing per fidelizzare la clientela (indurre al cliente di restare fedele ai prodotti dell'azienda cercando di favorire e stimolare questo legame), per acquisire nuovi clienti e per rafforzare il branding (azioni di marketing che si prefiggono lo scopo di rafforzare la notorietà della marca e quindi l'apertura di una comunicazione via email data dalla riconoscibilità del mittente).

Alcuni esempi di email promozionali sono: email di presentazione, invio di cataloghi prodotto, invio di coupon, email per il lancio di prodotti o servizi, newsletter, comunicati stampa, inviti ad eventi, sondaggi.

2. email di servizio: sono impiegate generalmente nell'operatività quotidiana. Hanno un tasso di apertura e di lettura molto elevato e sono autorizzate implicitamente dal rapporto in essere con il destinatario.

Alcuni esempi di email di servizio sono: email di benvenuto, conferme di iscrizione, email per assenza di prodotto, email di avviso, email di segnalazione dello stato dell'ordine, remember email, assistenza tecnica, auto responder.

Un fattore molto importante da tenere in considerazione quando si utilizza l'email per comunicare, è il tempo di arrivo a destinazione della posta elettronica. È istantaneo, soprattutto se viene comparato con la posta cartacea: un'email arriva in secondi o minuti. L'email ha rappresentato negli ultimi anni un mezzo per la comunicazione commerciale innovativo per almeno tre aspetti: tempi, misurabilità e costi. L'email marketing è caratterizzato da tempi di preparazione simili a quelli dell'invio cartaceo tradizionale ma con costi di invio molto più bassi e una possibilità di misurazione della resa totale che consente di misurare il ROI (Return of Investment) di una campagna indirizzando così le proprie risorse verso il gruppo di potenziali clienti che garantiscono il massimo rendimento. I costi di un invio di email promozionale con un sistema di email marketing professionale dipendono dal software scelto e dalle funzionalità offerte. Le misure fondamentali per una campagna di email marketing sono il numero di email inviate, numero di email consegnate, numero di email non giunte a destinazione suddivise per motivo, numero di email HTML aperte e numero di link cliccati. Alcuni sistemi permettono inoltre di misurare anche i tempi di apertura di una email per capire se chi ha aperto l'email si è soffermato qualche secondo, il tempo sufficiente affinché venisse aperta nell'anteprima del client di posta.

I sei passi di una campagna di email marketing sono rappresentati in figura 2.14.



Figura 2.14: i sei passi di una campagna di email marketing

Di seguito si trova una breve descrizione delle caratteristiche che deve avere un'applicazione di email marketing:

1. **Inserimento – acquisizione degli indirizzi dei destinatari:** lo strumento commerciale scelto deve essere flessibile nell'inserimento degli indirizzi di email e delle caratteristiche dei destinatari, permettendo di inserirli tramite interfaccia di amministrazione, tramite form e mediante importazioni di file (CSV Comma Separated Values, XLS, XML, WAB Windows Address Book e i principali database).
2. **Profilazione dei destinatari:** lo strumento deve poter gestire una serie di parametri o attributi aggiuntivi per identificare e caratterizzare al meglio ogni contatto in modo flessibile.
3. **Composizione del messaggio:** lo strumento deve gestire la composizione e la spedizione di email in formati diversi, come HTML, text o multipart. Occorre che sia presente un editor HTML per impaginare con facilità il messaggio e per poter utilizzare i parametri della profilazione e le informazioni riguardanti l'anagrafica del destinatario. Deve consentire un

facile inserimento di link tracciati, vale a dire link per cui in seguito si sia in grado di capire chi li ha cliccati e quante volte.

4. Test del messaggio: provare il messaggio composto inviandolo prima a uno sottolista di destinatari campione.
5. Invio del messaggio: scegliere quando inviare il messaggio (invio immediato o differito) e se inviarlo a tutti i destinatari assieme o suddividere l'invio una volta che si è composta l'email e selezionati i destinatari.
6. Verifica dei risultati ottenuti: analizzare le statistiche per valutare il risultato ottenuto dalla campagna.

Infine, si vuole approfondire brevemente i requisiti indispensabili che il messaggio email in formato HTML deve avere, in quanto è stato utilizzato nel corso del tirocinio. Non vanno inclusi script o funzionalità particolare come JavaScript, ActiveX, Flash, Video, Embedding, Include e Form. Tutto questo non tanto per problemi di antispam ma semplicemente perché non funzioneranno nella gran parte dei client. Riguardo ai CSS (Cascading Style Sheets), essendo spesso usati dagli spammer per mascherare parole chiave, è sempre più difficile utilizzarli nei messaggi email. Se l'uso dei CSS è proprio indispensabile, è da suggerire la definizione degli stili direttamente in linea all'interno del codice, non dichiararli quindi all'inizio né richiamare CSS dall'esterno. Questo perché alcuni client li ignorano completamente, altri rimuovono completamente tutto quello che si trova all'interno del tag <HEAD>. Si consiglia infine di non colorare i bordi, le tabelle e le immagini. Per avere un messaggio compatibile e leggibile, è necessario osservare le seguenti caratteristiche:

- Il peso del file non deve essere superiore a 50 KB;
- Utilizzare un formato non superiore a 600x600 pixels (l'ideale è 560 pixel), in modo da consentire all'utente la visualizzazione del messaggio nella prima schermata visibile;
- Non usare lettere accentate, ma usare gli apostrofi al loro posto;
- Non scrivere testo di colore bianco o comunque dello stesso colore dello sfondo;
- Verificare la corretta sintassi HTML in modo che non ci siano errori nel codice;

- Le scritte font e size è bene abbiano un valore assoluto e non relativo;
- Definire i colori con il codice esadecimale e non con stringhe;
- Evitare se possibile l'uso di simboli speciali come < > & €;
- Usare il title="descrizione" nei tag delle immagini e dei link per avere una breve legenda in sovrapposizione al passaggio del mouse;
- Preferire il tag
 al tag <P> che genera uno spazio eccessivo o viene addirittura rimosso;
- Usare se possibile indirizzi brevi per i link per evitare che vengano troncati;
- Non usare Word per scrivere il proprio messaggio;
- Creare tutto il contenuto all'interno di una tabella a larghezza fissa;
- Assicurarsi che tutti i link funzionino all'interno dell'email e che prevedano il tag target="_blank" per aprire il collegamento in una pagina esterna e no in un ristretto frame di un sistema webmail;
- Le immagini devono essere nel formato jpg o gif evitando immagini con dimensione superiore a 50 KB;
- Le immagini molto grandi andrebbero divise in due o più parti e poi composte usando una tabella;
- Non usare immagini come sfondo di pagina e di tabella;
- Evitare se possibile le immagini con una mappa di link e dare alle immagini un nome rappresentativo e non generico;
- Aggiungere sempre un testo alternativo ALT alle immagini per agevolare la lettura a chi legge il messaggio senza scaricare le immagini.

Capitolo 3

RELAZIONE TIROCINIO

In questo capitolo si descrivono gli strumenti utilizzati durante il tirocinio e cosa si è realizzato praticamente. In particolare si parla della tecnologia di sviluppo utilizzata e i vari software necessari per creare e gestire i vari siti web. Successivamente si fa una panoramica sull'organizzazione aziendale trovata per compiere le richieste che i project manager e commerciali facevano. Infine si analizzano i vari siti creati, le modifiche effettuate, l'email realizzate e i backup eseguiti.

I primi giorni sono stati di formazione per comprendere come l'ambiente era organizzato. Inoltre è stato necessario spendere del tempo per conoscere e documentarsi sugli strumenti e tecnologie che si sono utilizzati.

Grazie a un lavoro svolto quotidianamente durante le 250 ore del tirocinio, è stato possibile analizzare i lavori svolti. Questa attività prevedeva di scrivere su un banale foglio elettronico i vari compiti realizzati. Inoltre si associava per ogni descrizione la tipologia del compito e a quale fiera apparteneva. E' così possibile mostrare una panoramica generale delle varie richieste eseguite, mostrando con la figura 3.1 qual è il lavoro che è stato svolto.

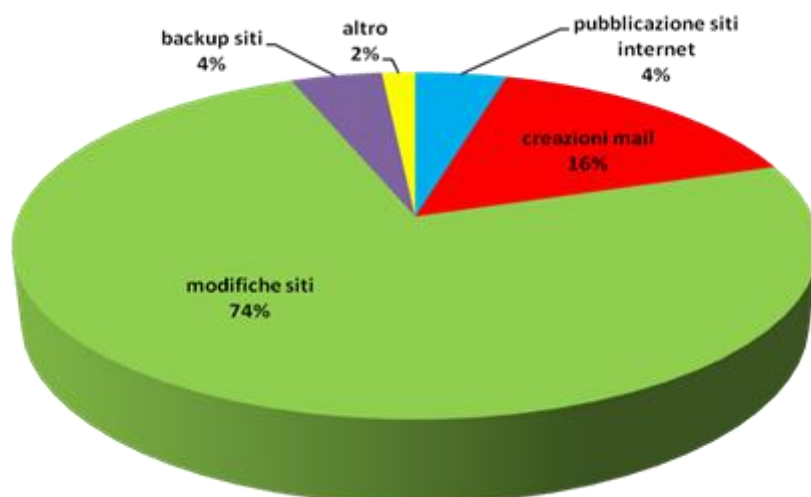


Figura 3.1: panoramica del lavoro svolto

Come mostra la figura 3.1, si sono svolte molte attività di modifiche. Questo perché quando si pubblicava un nuovo sito internet, molte pagine non avevano ancora i contenuti e molti testi erano ancora incompleti. Per questo motivo si sono svolti molti aggiornamenti alle pagine web. Per il 2% si sono svolte altre attività, come riunioni o discussioni per decidere varie procedure e il layout grafico.

E' inoltre interessante mostrare in percentuale i lavori svolti per ogni fiera (in figura 3.2).

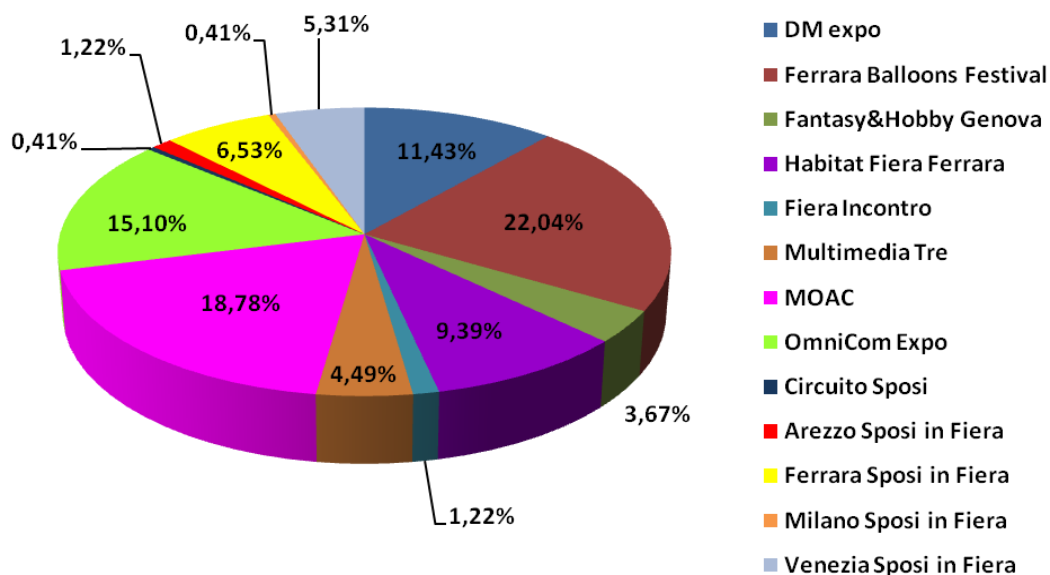


Figura 3.2: lavori svolti per ogni fiera

Si è lavorato principalmente per il sito di Ferrara Balloons Festival, MOAC, OmniCom Expo, DM expo, Habitat Fiera Ferrara, Ferrara Sposi in Fiera e Venezia Sposi in Fiera.

3.1 TECNOLOGIA LATO SERVER E SOFTWARE

In questa sezione si descrivono gli strumenti principali utilizzati durante il tirocinio per poter compiere le mansioni affidate. Tutte queste informazioni indispensabili sono state di grande utilità per capire e per poi eseguire tutti gli obiettivi preposti e concordati con il tutor aziendale. In particolare si descrive:

- ASP.NET, la tecnologia utilizzata per creare e/o modificare pagine web dei siti internet in possesso di Multimedia Tre. Si farà una panoramica delle funzionalità di ASP.NET e di come funziona il ciclo di vita di una pagina ASP.NET. Inoltre si parlerà del grande vantaggio offerto dalle Master Page e delle caratteristiche del file web.config per configurare un'applicazione web.
- SQL server, la soluzione utilizzata per l'analisi e la gestione dei dati. Si farà una panoramica del DBMS, analizzando i tipi di dati, le tabelle, i vincoli per controllare i dati e le istruzioni SQL necessarie per manipolare i dati presenti nel database.
- Visual Studio, l'ambiente di sviluppo utilizzato per scrivere il codice delle pagine web.
- Internet Information Services (IIS), il server web utilizzato per testare il codice creato prima di pubblicarlo online. Si descriverà come sono state impostate le configurazioni per far funzionare i siti web in locale, ovvero nel computer utilizzato.

3.1.1 ASP.NET

ASP.NET è un ambiente di sviluppo web che include i servizi necessari per la generazione di applicazioni web. Le applicazioni possono essere codificate in un linguaggio compatibile con

Common Language Runtime (CLR), ad esempio Microsoft Visual Basic, C#, JScript .NET e J#. Durante il tirocinio si è scelto di utilizzare C# che è un linguaggio di programmazione che deriva dai linguaggi C e C++ ma più moderno, semplice e interamente a oggetti. Le pagine ASP.NET sono quindi completamente orientate a oggetti e al loro interno è possibile operare su elementi HTML utilizzando proprietà, metodi ed eventi.

ASP.NET è integrato in .NET Framework che presenta due componenti principali: CLR e la libreria di classi .NET Framework. Il primo rappresenta la base di .NET Framework in quanto gestisce il codice in fase di esecuzione, fornendo servizi di base come la gestione della memoria, gestione di thread e servizi remoti, attivando una rigida indipendenza dei tipi, garantendo protezione ed efficienza. Il secondo è un insieme completo orientato agli oggetti di tipi riutilizzabili che possono essere impiegati nello sviluppo delle applicazioni. ASP.NET è quindi l'ambiente che consente agli sviluppatori di utilizzare .NET Framework per applicazioni basate sul web visto l'architettura completa di cui è dotato.

La piattaforma ASP.NET è stata progettata per supportare l'editor HTML WYSIWYG e altri strumenti di programmazione, come Visual Studio descritto nel paragrafo 3.1.3. Per creare un'applicazione ASP.NET, si utilizza il Web Form, pagine web basate su form con i controlli server ASP.NET.

ASP.NET ha inoltre semplificato significativamente le funzionalità di accesso ai database. Ad esempio, è molto più semplice ed immediato generare una pagina contenente una lista di dati prelevati da un database attraverso la vasta disponibilità di controlli, classi e strumenti di sviluppo, come ad esempio il DataList o GridView.

Di seguito sono descritti le fasi del ciclo di vita di una pagina ASP.NET, che sono otto:

1. **Richiesta della pagina.** ASP.NET determina se la pagina deve essere analizzata e compilata oppure se è possibile inviare in risposta una versione della pagina memorizzata nella cache senza eseguirla.
2. **Avvio.** Vengono impostate le proprietà della pagina, quali Request (oggetto HttpRequest con informazioni sulla richiesta HTTP corrente) e Response (oggetto HttpResponse che consente di inviare i dati di una risposta HTTP a un client e fornire informazioni su tale

risposta). In questa fase, la pagina determina anche se la richiesta è un postback o una nuova richiesta e imposta la proprietà `IsPostBack`.

3. **Inizializzazione della pagina.** Sono disponibili i controlli sulla pagina e viene impostata la proprietà `UniqueID` (identificatore per il contenitore dei nomi del controllo server) di ciascun controllo. Inoltre, viene eventualmente applicato alla pagina un tema.
4. **Caricamento.** Se la richiesta corrente è un postback, le proprietà dei controlli vengono caricate con le informazioni recuperate dallo stato di visualizzazione e dallo stato di controllo.
5. **Convalida.** Viene chiamato il metodo `Validate` di tutti i controlli di convalida, il quale imposta la proprietà `IsValid` dei singoli controlli di convalida e della pagina.
6. **Gestione di eventi postback.** Se la richiesta è un postback, viene chiamato qualsiasi gestore eventi.
7. **Rendering.** La pagina chiama il metodo `Render` (invia il contenuto del controllo server a un oggetto `HtmlTextWriter` fornito, che scrive il contenuto di cui si effettuerà il rendering sul client) per ogni controllo, fornendo un writer di testo che scrive l'output in `OutputStream` della proprietà `Response` della pagina.
8. **Scaricamento.** Vengono scaricate le proprietà della pagina, ad esempio `Response` e `Request`, e vengono eseguite le operazioni di pulizia.

Una delle importanti caratteristiche introdotte da ASP.NET è la possibilità di utilizzare le Master Page per dare un aspetto uniforme al proprio sito evitando di riscrivere lo stesso codice in tutte le pagine. Con una singola pagina Master è possibile definire l'aspetto e il comportamento standard per tutte o per un gruppo di pagine del sito internet. Quando richieste, le pagine di contenuto vengono unite alla pagina Master per generare un output in cui sono congiunti il layout definito nella pagina Master e il contenuto della pagina di contenuto. Le Master Page sono costituite essenzialmente da due parti, la pagina Master vera e propria e una o più pagine di contenuto. La prima è un file ASP.NET con estensione `master`, ad esempio `MasterPage.master`, che presenta un layout predefinito in cui sono compresi elementi HTML, controlli server e testo statico.

La Master Page è identificata da una direttiva @ Master speciale, che sostituisce la direttiva @ Page utilizzata per le pagine aspx ordinarie. La direttiva è simile a quella scritta di seguito:

```
<%@ Master Language="C#" CodeFile="MasterPage.master.cs" Inherits="MasterPage" %>
```

I contenuti della Master Page possono essere gli elementi HTML di primo livello, quali html, head, form e controlli ContentPlaceHolder che sono aree per la visualizzazione di contenuti sostituibili, definiti nelle pagine di contenuto. La direttiva per definire un ContentPlaceHolder è simile a quella scritta di seguito:

```
<asp:contentplaceholder id="CphMain" runat="server" />
```

Le pagine di contenuto associate a una Master Page specifica viene stabilita nella direttiva @ Page della pagina di contenuto includendo un attributo MasterPageFile che fa riferimento alla pagina Master da utilizzare. Il contenuto di una pagina di contenuto viene creato aggiungendo controlli Content ed eseguendo il mapping di questi ultimi sui controlli ContentPlaceHolder nella pagina Master. Un esempio è il seguente:

```
<%@ Page Language="C#" MasterPageFile="~/MasterPage.master" Title="Titolo" %>  
<asp:Content ID="Content1" ContentPlaceHolderID=" CphMain " runat="Server">  
  Contenuto della pagina.  
</asp:Content>
```

In una pagina di contenuto tutti gli elementi non inclusi all'interno del controllo Content genereranno un errore. E' possibile creare più di un controlli ContentPlaceHolder.

Il processo di esecuzione delle Master Page vengono eseguite seguendo lo schema presente in figura 3.3.

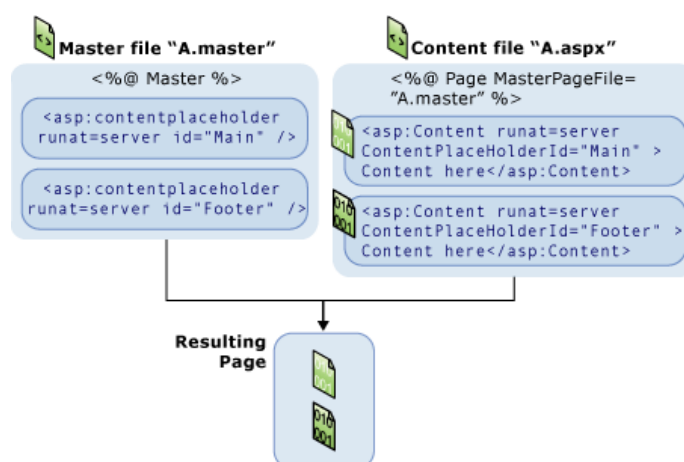


Figura 3.3: processo di esecuzione delle Master Page

In particolare:

1. L'utente richiede una pagina digitando l'URL (Uniform Resource Locator) della pagina di contenuto.
2. Una volta recuperata la pagina, la direttiva @ Page viene letta. Se la direttiva fa riferimento a una Master Page, viene letta anche quest'ultima. Entrambe le pagine vengono compilate, se richieste per la prima volta.
3. La Master Page con il contenuto aggiornato viene unita alla struttura dei controlli della pagina di contenuto.
4. Il contenuto di singoli controlli Content viene unito al controllo ContentPlaceHolder corrispondente nella Master Page.
5. Viene eseguito il rendering della pagina unita risultante nel browser.

Infine bisogna descrivere un meccanismo di configurazione previsto da ASP.NET per delle opzioni dell'applicazione web. Tutto questo è basato su un semplice file di testo chiamato web.config, con codifica XML. ASP.NET dunque non si affida ad IIS per la configurazione dell'applicazione ed è quindi più semplice modificare le impostazioni di base anche senza avere accesso fisicamente al server. Sono ammessi più file web.config per una sola applicazione, uno per ogni cartella. Ogni sotto cartella eredita le impostazioni della cartella padre.

E' diviso in due sezioni principali, ognuna con un compito preciso e nidificate dentro il tag <configuration>, che sono:

- <system.web>: contiene tutti gli elementi di configurazione predefiniti di ASP.NET, come la gestione dell'autenticazione in un'applicazione, definizione delle pagine di errore personalizzate, configurazione delle impostazioni internazionali e di lingua, configurazione dello stato della sessione, impostazioni di esecuzione HTTP, impostazioni dell'identità con cui viene eseguito ASP.NET e configurazione delle chiavi da utilizzare per la crittografia e la decrittografia dei dati dei cookie.
- <appSettings>: contiene eventuali costanti di configurazione introdotte dal programmatore e inerenti la specifica applicazione, non predefinite da ASP.NET. Vengono inserite dentro il tag <add key> e saranno lette a runtime dall'applicazione. Un'applicazione comune è quella di definire la stringa di connessione al database.

Per motivi di sicurezza, i file web.config non vengono mostrati all'utente. Viene ricevuto un messaggio di avviso che dice che la risorsa che si sta cercando di visualizzare è protetta.

3.1.2 SQL SERVER 2005

SQL Server è un DBMS di casa Microsoft, nato come prodotto leader del suo settore in ambiente Microsoft. E' un prodotto potente e affidabile per la gestione dei dati. Offre funzionalità avanzate, protezione dei dati e prestazioni elevate per client di applicazioni web leggere e archivi dati locali.

Durante il tirocinio si è utilizzato SQL Server 2005 (logo in figura 3.4) che supporta una piena integrazione con il .NET Framework attraverso il SQLCLR (Structured Query Language Common Language Runtime). L'ambiente CLR supporta oggetti di database procedurali come funzioni scritti nei linguaggi .NET (C# e VB.NET) non disponibili nel linguaggio T-SQL (Transact - Structured Query Language), utilizzato per eseguire operazioni con i dati del database. E' uno strumento completo che permette di operare anche in maniera visuale con il motore di database oltre che in sintassi T-SQL. Un database di SQL Server 2005 è costituito da un insieme di oggetti che contengono dati

(viste e tabelle) e da altri oggetti che legano tra loro questi dati (indici, stored procedure, funzioni definite dall'utente).



Figura 3.4: logo di SQL Server 2005

Un'istanza di SQL Server può supportare più database. Ogni database consente di archiviare dati correlati o non correlati di altri database. I tipi di dati che esistono nel database sono suddivisi nelle seguenti categorie:

1. dati numerici esatti. Possono essere bigint (dimensioni a 8 byte), int (dimensioni a 4 byte), smallint (dimensioni a 2 byte), tinyint (dimensioni a 1 byte), bit (valore 0 o 1), decimal (numeri con precisione e scala fisse), numeric (numeri con precisione e scala fisse), money (dimensioni a 8 byte), smallmoney (dimensioni a 4 byte).
2. stringhe di caratteri. Possono essere char (dati di tipo carattere a lunghezza fissa non UNICODE con una lunghezza di n byte, compreso tra 1 e 8.000), varchar (dati di tipo carattere a lunghezza variabile non UNICODE) e text (dati non UNICODE a lunghezza variabile con lunghezza massima di $2^{31}-1$ caratteri).
3. stringhe di testo UNICODE. Possono essere nchar (dati UNICODE a lunghezza fissa contenenti n caratteri, compreso tra 1 e 4.000), nvarchar (dati UNICODE a lunghezza variabile) e ntext (dati UNICODE a lunghezza variabile con lunghezza massima di $2^{30} - 1$ caratteri) .
4. numerici approssimati. Possono essere float (da $-1,79E+308$ a $-2,23E-308$, 0 e da $2,23E-308$ a $1,79E+308$) e real (da $-3,40E + 38$ a $-1,18E - 38$, 0 e da $1,18E - 38$ a $3,40E + 38$).
5. stringhe binarie. Possono essere binary (dati binari a lunghezza fissa con lunghezza di n byte, compreso tra 1 e 8.000), varbinary (dati binari a lunghezza variabile) e image (dati binari a lunghezza variabile da 0 a $2^{31}-1$ byte).
6. data e ora. Possono essere datetime (1 gennaio 1753 - 31 dicembre 9999) e smalldatetime (1 gennaio 1900 - 6 giugno 2079).

7. altri tipi di dati, come cursor, sql variant, table, timestamp, uniqueidentifier e xml.

Per contenere tutti i dati disponibili in un database si utilizzano le tabelle, dove i dati sono organizzati in righe e colonne. Ogni riga rappresenta un record univoco e ogni colonna rappresenta un campo all'interno del record. E' possibile definire fino a 1.024 colonne per tabella. I nomi di tabelle e colonne devono essere conformi alle regole per gli identificatori e devono essere univoci all'interno di ogni tabella, ma è possibile utilizzare lo stesso nome di colonna in tabelle diverse dello stesso database. Nelle colonne di una tabella si possono applicare dei vincoli che servono per controllare i dati. Ci sono diverse categorie, ad esempio:

- vincolo di chiave primaria. E' il più importante perché contrassegna l'unicità del record nella tabella. Può essere costituito da uno o più campi e ne deve esistere solo uno per tabella.
- vincolo di chiave esterna. E' un vincolo tra uno o più campi chiave di una tabella, con campi di un'altra tabella.
- vincoli predefiniti. Si creano quando SQL Server assegna un valore a un campo, come il `getdate()` che restituisce la data di creazione del record.
- vincolo non null. Proibisce l'inserimento di dati vuoti in un campo.
- vincolo check. Controlla che l'inserimento di un dato in un campo, appartenga ad un certo intervallo di valori.
- vincolo unique. Controlla che ogni voce che contiene è unica per tutta la tabella, evitando ridondanze di dati.
- vincoli identity. Un campo si autoincrementa il proprio valore per ogni nuovo record inserito.

Un altro oggetto utile nel database è la vista: tabella virtuale il cui contenuto è definito da una query, ovvero da una ricerca.

Durante lo sviluppo di un'applicazione web, le richieste per la manipolazioni e interrogazioni di dati sono fatte tramite istruzioni di INSERT, UPDATE, DELETE e SELECT scritte in linguaggio SQL.

E' un linguaggio specializzato perché lavora sulle tabelle di un database e non per creare grafici. Inoltre è un linguaggio non procedurale perché non si spiega al computer come deve fare ma cosa si vuole ottenere. Di seguito sono descritte le sintassi delle quattro istruzioni sopra citate che sono state utilizzate durante il tirocinio.

INSERT

La sintassi standard del comando INSERT è:

```
INSERT INTO {nome_tabella} [ (elenco_campi) ] VALUES (valore, ...)
```

L'*elenco_campi* contiene i nomi dei campi di una tabella separati da virgole e racchiuso tra parentesi tonde. Può essere anche omesso e in questo caso il comando agisce su tutte le colonne della tabella. Il *valore* contiene il dato vero e proprio da inserire nel record. Formulato in questa maniera, inserisce una sola riga nella tabella specificata con *nome_tabella*.

UPDATE

La sintassi standard del comando UPDATE è:

```
UPDATE {nome_tabella} SET nome_campo=nuovo_valore, .. [WHERE condizione]
```

Serve per modificare il contenuto di una tabella. Senza la clausola *WHERE*, il comando assegna il nuovo valore all'intera colonna indicata con *nome_campo*. Con la clausola *WHERE*, si rende selettivo il comando, impostando la *condizione* desiderata (ad esempio *nome campo = valore desiderato*).

DELETE

La sintassi standard del comando DELETE è:

```
DELETE [FROM] nome_tabella [WHERE condizione]
```

Serve per eliminare un intero record. Senza la clausola *WHERE*, il comando elimina tutti i record presenti nella tabella indicata con *nome_tabella*. Con la clausola *WHERE*, il comando elimina soltanto i record che verificano la *condizione* (ad esempio *nome campo = valore desiderato*).

SELECT

La sintassi standard del comando SELECT è:

```
SELECT [ALL | DISTINCT] elenco_di_selezione FROM nome_tabella
      [JOIN condizione_join] [WHERE condizione_selezione]
      [GROUP BY espressione_raggruppamento] [HAVING condizione_ricerca]
      [ORDER BY criterio_ordinamento [ASC | DESC] ]
```

Serve per interrogare il database. L' *elenco_di_selezione* è un elenco di colonne da estrarre dalla *nome_tabella* specificata nella clausola *FROM*. Deve contenere almeno un nome di colonne tra quelle presenti nella tabella. Se si vogliono estrarre tutte le colonne si mette il carattere asterisco. Se si utilizza il qualificatore *ALL*, si estraggono tutti gli elementi, come accade sempre. Se si utilizza il qualificatore *DISTINCT*, si specifica che la query deve trascurare gli eventuali valori ripetuti, presentando solo valori univoci. Con la clausola *WHERE* si specifica un criterio in base al quale selezionare il contenuto delle colonne indicate. Con la clausola *JOIN*, in alternativa alla clausola *WHERE*, si può stabilire una relazione tra tabelle. La clausola *GROUP BY* ha funzioni di raggruppamento e si deve segnalare sia in *elenco_di_selezione* e in *espressione_raggruppamento* la stessa colonna. Viene utilizzata con le funzioni di aggregazione come *AVG* che calcola il valore medio o *COUNT* che conta le righe. La clausola *HAVING* svolge correttamente il ruolo di *WHERE* in riferimento a un valore ottenuto mediante la clausola di raggruppamento. La clausola *ORDER BY* serve per ordinare diversamente le righe da come sono nella tabella di origine. Si indicano le colonne scrivendo i nomi su *criterio_ordinamento*. Le parole chiave *ASC* e *DESC* indicano il tipo di ordinamento che sono rispettivamente crescente (dal più piccolo al più grande) e decrescente (dal più grande al più piccolo).

3.1.3 VISUAL STUDIO 2005

Per realizzare applicazioni che si basano sul Framework .NET si utilizza un ambiente di sviluppo integrato chiamato Visual Studio. La Microsoft ha prodotto diverse versioni. Nel 2002 è stata rilasciata la prima versione chiamata Visual Studio .NET; nel 2003 è stato rilasciato Visual Studio

.NET 2003 e nel 2005 la prima versione di Visual Studio a non includere il suffisso .NET nel nome: Visual Studio 2005. Le ultime versioni sono Visual Studio 2008 e Visual Studio 2010. Ogni nuova versione aggiunge innovazioni e migliorie rispetto alle precedenti.

Durante il tirocinio si è utilizzato Visual Studio 2005 (logo in figura 3.5), dotato di buone caratteristiche di usabilità e potenza.



Figura 3.5: logo di Visual Studio 2005

Visual Studio supporta diversi tipi di linguaggio come C++, C#, J#, Visual Basic .NET e permette di realizzare applicazioni web, siti internet e servizi web. Mette a disposizione del programmatore dei mezzi che permettono di aumentare la produttività, come l'IntelliSense o un designer visuale delle forms. Inoltre, grazie a un debugger interno integrato in Visual Studio, è possibile correggere errori logici o sintattici senza compilare tutta l'applicazione.

3.1.4 INTERNET INFORMATION SERVICES (IIS) VERSIONE 6.0

Internet Information Services è un server web per sistemi operativi di Microsoft Windows. Un server web è un'applicazione in esecuzione su server che attende connessioni da parte di un client (browser) che si collegano attraverso la porta 80, come specificato dal protocollo TCP/IP. Il suo compito è quello di fornire informazioni codificate secondo uno specifico protocollo o linguaggio, come HTML, XML o XHTML (eXtensible HyperText Markup Language).

La versione 6.0 di IIS fa parte del sistema operativo Microsoft Windows 2003 Server e si trova sul cd di installazione del sistema operativo. Non può essere installata su nessun altro sistema operativo poiché c'è una strettissima integrazione tra IIS e il sistema operativo. Questa versione è stata progettata supportando ASP.NET e diventa quindi un web server più performante e affidabile.

Come mostra la figura 3.6, ogni richiesta web viene accettata dal processo HTTP.SYS che lavora in modalità Kernel Mode. Questo processo accetta e accoda tutte le richieste in arrivo e le passa al WAS (Web Administration Service) che è il gestore di Pool.

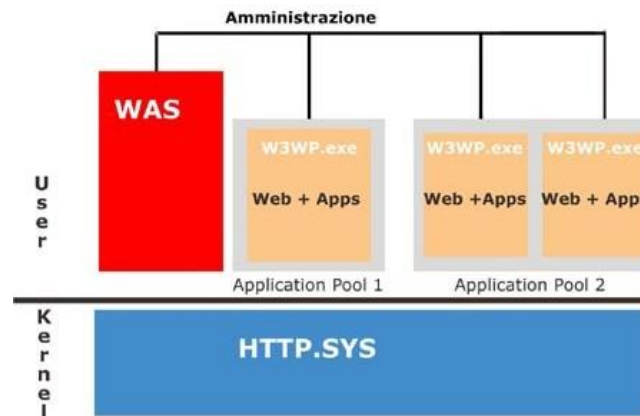


Figura 3.6: processo IIS

Il WAS ha il compito di controllare e monitorare lo stato delle applicazioni web e di riciclare i worker process (processi in cui girano le applicazioni). Ogni Pool di processi può essere costituito da uno o più worker process. La situazione ottimale, che è quella predefinita, si ha quando si associa ad ogni pool di applicazioni un solo sito web, in quanto un suo malfunzionamento non implicherebbe un degrado delle prestazioni anche degli altri nello stesso pool.

Nel computer messo a disposizione durante il tirocinio è stato installato IIS per poter lavorare autonomamente e per testare il codice scritto. Di seguito è spiegato come è stato possibile gestire i siti web tramite il server web.

Per poter usufruire delle funzionalità di IIS 6.0 e per impostare le configurazioni per far funzionare il sito web, si utilizza l'interfaccia di amministrazione di IIS che si trova all'interno del menu Programmi – Strumenti di amministrazione – Internet Information Services. La console visualizza un'interfaccia come quella mostrata in figura 3.7.

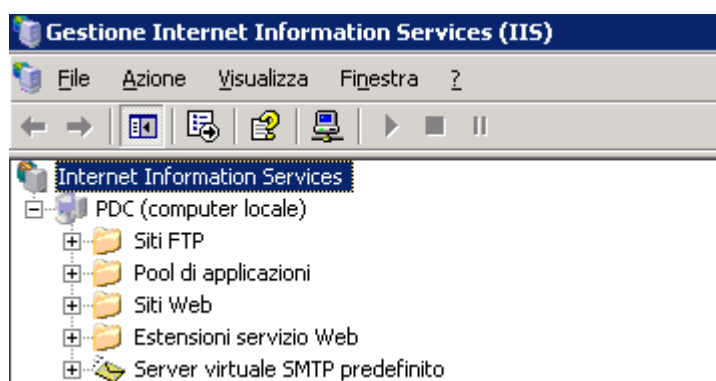


Figura 3.7: interfaccia di amministrazione di IIS

Nella colonna di sinistra si trova un'icona che rappresenta il server, dove è installato il servizio web. A questa icona sono collegate le funzionalità installate, ad esempio siti FTP (File Transfer Protocol), Pool di applicazioni, siti Web, Estensioni servizio Web, Server virtuale SMTP (Simple Mail Transfer Protocol) predefinito. Delle funzionalità offerte, è di interesse la gestione di siti internet che offre il servizio in grado di ospitare pagine web e di rispondere alle richieste del browser. Poiché i file di ogni sito internet creato si trovano su un disco fisso esterno, per creare nuovi siti, sono state utilizzate le directory virtuali, collegamenti a cartelle che risiedono fisicamente in un'altra parte del disco diversa da quella associata da IIS. Quindi è necessario cliccare con il tasto destro sul sito, selezionare Nuovo e poi Directory Virtuale. Si apre una finestra che chiede all'utente di indicare l'alias ossia il nome che la cartella avrà all'interno del sito web e che non necessariamente deve coincidere con il nome fisico. Successivamente è necessario selezionare il percorso fisico della cartella reale che verrà collegata all'alias. Infine si selezionano i permessi che devono essere associati alla cartella e che possono essere di lettura (file leggibili via web), di esecuzione script (come file .asp), di esecuzione (file eseguibili come .exe, .com, cgi e altri), di scrittura (all'interno della cartella) e di esplorazione (visualizzare i contenuti della cartella). Di solito sono stati dati tutti i permessi per dare il massimo controllo.

Una volta creato il sito, è possibile cambiare le configurazioni impostate durante la creazione ed è possibile aggiungerne. Premendo con il tasto destro del mouse sul nome del sito e scegliendo Proprietà si apre una finestra con alcune schede, come mostrato in figura 3.8.

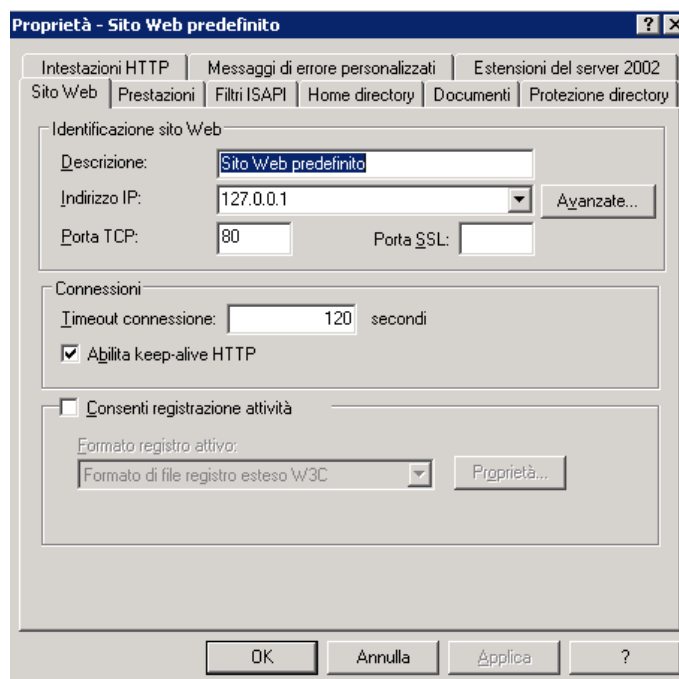


Figura 3.8: proprietà delle configurazioni di un sito web in IIS

Le configurazioni principali che si possono cambiare sul sito web sono raggruppate in schede, che sono:

1. Sito Web: permette di definire e modificare le impostazioni principali del sito.
2. Prestazioni: permette di limitare le risorse che il sito può utilizzare.
3. Filtri Isapi: permette di aggiungere dei filtri al server per gestire particolari tipologie di connessioni.
4. Home Directory: permette di definire le proprietà relative alla cartella dove risiede il sito web.
5. Documenti: permette di specificare il nome del documento che deve essere aperto quando l'utente digita l'url del sito. È possibile indicare più nomi per i documenti predefiniti.

Il sistema andrà nell'ordine indicato dal più in alto al più in basso, fino a che non trova nella cartella del sito un file con quel nome.

6. Protezione directory: permette di definire tutte le politiche di sicurezza che devono essere abilitate per rendere più o meno sicuro l'accesso al sito.
7. Intestazioni HTTP: permette di definire la cache delle pagine, il tipo di contenuto, le intestazioni e il rating di una pagina del sito.
8. Messaggi di errore personalizzati: permette di specificare se utilizzare i file HTML predefiniti per gli errori oppure utilizzarne di personalizzati.

Una volta configurato il sito, è possibile vederlo aprendo il browser e digitando nella barra degli indirizzi il seguente link:

<http://localhost/alias>

dove *alias* è quello specificato durante la creazione della directory virtuale.

3.2 ORGANIZZAZIONE AZIENDALE

Gli obiettivi del tirocinio sono stati concordati insieme al tutor aziendale, però le richieste di creare nuovi siti internet, di modificare pagine web già presenti online e di creare delle email per la comunicazione aziendale venivano fatte dai project manager e dai commerciali.

E' stata adottata una semplice procedura per poter organizzare tutte le richieste in modo da avere una visione generale del lavoro da svolgere e per fissare il tempo entro il quale eseguirle. Il tutto viene gestito dalla segreteria tramite invii di email che ha il compito di smistare le richieste all'ufficio e alle persone di competenze.

Nella figura 3.9 sono mostrati i vari passaggi di questa procedura.



Figura 3.9: procedura di richieste per le attività di Multimedia Tre

Tutto il giro di richieste e informazioni avviene tramite email per non perderne traccia e per evitare che non vengano eseguite perché comunicate a voce. Le richieste del project manager vengono inviate alla segreteria con la descrizione e la scadenza entro la quale ci si aspetta che venga eseguita. Una volta che è stata esaminata e trascritta nel calendario generale, la segreteria invia la richiesta all'ufficio di competenza. L'ufficio, una volta ricevuta la richiesta, la analizza e conferma la data di scadenza se gli è possibile, altrimenti avvisa la segreteria che non può essere svolto entro il tempo chiesto a causa di altre richieste che hanno una priorità più elevata. Una volta che la richiesta è stata eseguita dall'ufficio di competenza, viene comunicato alla segreteria, la quale informa il project manager.

Come sottolineato nell'introduzione, la collaborazione con la responsabile della grafica è stata molto importante, soprattutto per i nuovi siti internet e per modificare quelli già esistenti. Tutti i contenuti dovevano rispettare il layout grafico deciso dal project manager e dall'art director graphics.

I materiali ricevuti sono stati:

- immagini in formato JPEG (Joint Photographic Experts Group), PNG (Portable Network Graphics) e GIF (Graphics Interchange Format) da utilizzare nei siti web e nell'email;
- file SWF (formato compresso e non modificabile dei Flash) da utilizzare nei siti web;
- HTML di base per pagine speciali dei siti web o email;
- file PDF;
- file word con i testi per le pagine web e per le email.

3.3 REPORT SITI CREATI

I siti che sono stati creati durante il tirocinio sono stati nove. Le informazioni iniziali sono state date dalla responsabile della grafica, sotto forma di materiale grafico, e dal project manager o commerciali sotto forma di file di testo. Dopo una breve discussione a voce e una spiegazione su come impaginare il tutto, i siti creati sono stati:

- www.arezzosposinfiera.it



Figura 3.10: sito Arezzo Sposi in Fiera

- www.ferrarafestival.it



Figura 3.11: sito Ferrara Balloons Festival

- www.fieresposi.it



Figura 3.12: sito Fiere Sposi

- www.fantasyhobby.it



Figura 3.13: sito Fantasy&Hobby Genova

- www.moacsanremo.it



Figura 3.14: sito MOAC

- www.ferrarasposinfiera.it



Figura 3.15: sito Ferrara Sposi in Fiera

- www.multimediatre.com



Figura 3.16: sito Multimedia Tre

- www.fieraincontro.it



Figura 3.17: sito Fiera Incontro

- www.veneziasposinfiera.it



Figura 3.18: sito Venezia Sposi in Fiera

Il primo passo fatto è stato quello di cercare la parte più comune, dov'era possibile, per ogni pagina in modo da creare il Master Page. Con questa specie di template infatti, le modifiche da fare ad ogni pagina o ad esempio al menù, risultavano molto più semplici in quanto si cambiava

solo un file. Non sempre è stato possibile creare una sola Master Page visto la diversità tra certe tipologie di pagine. Sostanzialmente, nei siti creati durante il tirocinio, il problema si è rilevato soltanto tra l'home page e il resto delle pagine. I Master Page contengono tutto il contorno del sito, come il menù, il logo della fiera, i vari contatti, collaboratori e partner e altre informazioni che dovevano essere sempre presenti in ogni pagina, come descritto nella tabella 2.1. Il ContentPlaceHolder doveva essere posizionato in modo tale da permettere di inserire solo il contenuto della particolare pagina.

Come accennato, un contenuto da mettere all'interno del Master Page è il menù. Quasi ogni sito creato da Multimedia Tre contiene la stessa tipologia di menù, salvo per particolari manifestazioni che ne possono contenere anche di più. Ogni pagina ha la voce del menù con un colore tale da far capire all'utente in quale pagina e su quale area del menù si trova in modo da garantire una certa usabilità. Per fare questo si è utilizzato un semplice javascript che, a seconda della pagina web visitata, colora la scritta applicando uno stile adeguato. Il menù classico realizzato è mostrato in figura 3.19.

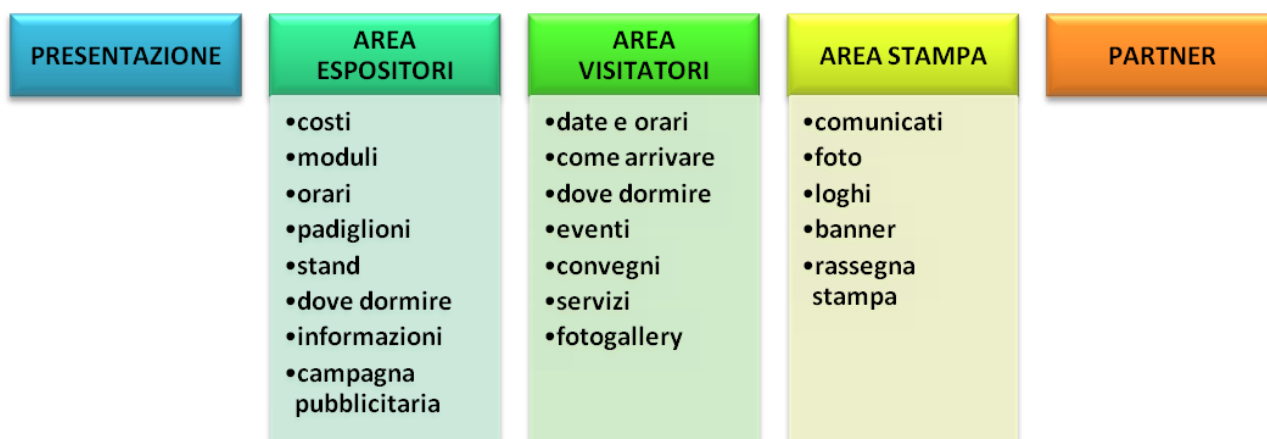


Figura 3.19: menù classico dei siti

Alla fine del paragrafo 3.1.1 si è parlato del file web.config. All'interno di questo file si è definito dentro a <connectionStrings>, che si trova dentro la sezione <appSettings>, l'elemento connectionStrings che specifica un insieme di stringhe di connessione al database, come coppie di nome e valore, da poi utilizzare nelle applicazioni. Un esempio è dato dalla seguente riga di codice:

```
<add name="nome connessione" connectionString="Data Source=server;
Initial Catalog=nome database; User ID=utente; Password=password accesso"/>
```


Inoltre, nel web.config sono stati inseriti sempre dentro a <appSettings> dei valori costanti all'interno del sito come coppia di nome/valore. Ad esempio, sono stati aggiunti il codice che identifica l'edizione della fiera e l'email principale del sito:

```
<add key="CodFieraEdizione" value="SPOFE10"/>
<add key="EmailSito" value="info@ferrarasposinfiera.it"/>
```

Il *CodFieraEdizione* è un valore molto importante e utilizzato in ogni form di registrazione per identificare la fiera (SPO sta per sposi), dove prende luogo (FE sta per Ferrara) e l'anno in cui si svolge (10 sta per 2010). Tutti questi valori sono memorizzate in una tabella del database online di Multimedia Tre. La tabella si chiama tb_web_FieraEdizione ed è così strutturata:

Nome del campo	Tipo di dato	Vincolo applicato
Codice	varchar(8)	non null, chiave primaria
Anno	char(4)	
DataInizio	datetime	
DataFine	datetime	
ComplessoFieristico	varchar(100)	
Dol	datetime	predefinito con getdate()

Tabella 3.1: informazioni relative all'edizione di una fiera

Il campo *Codice* è la chiave primaria ed è poi utilizzata in altre tabella come chiave esterna per identificare da quale fiera arriva.

Un'applicazione presente in ogni sito è il modulo di richiesta informazioni degli espositori. Il form generale richiede al potenziale espositore di inserire le generalità dell'azienda, come la ragione sociale, l'indirizzo, il CAP, la città, la provincia, il telefono, il fax, l'email e il sito internet. Inoltre viene chiesto chi è il referente, la tipologia dell'azienda, che prodotti espone, il settore di appartenenza, se ha già partecipato in passato alla manifestazione fieristica, se partecipa ad altre fiere dello stesso settore, la superficie richiesta per lo stand ed eventuali commenti. Una volta che ha compilato il form correttamente, viene inviata un'email con il riepilogo dell'informazioni dell'espositore al responsabile della manifestazione fieristica, il quale dovrà successivamente interessarsi e fornire altri dettagli della fiera.

Inoltre, tutti i dati dell'espositore vengono memorizzati nella tabella `tb_web_RichiestaInformazioni`, così strutturata:

Nome del campo	Tipo di dato	Vincolo applicato
Codice	int	non null, chiave primaria, identity
RagioneSociale	varchar(50)	non null
Indirizzo	varchar(50)	
Cap	char(5)	
Citta	varchar(50)	
Provincia	varchar(3)	chiave esterna
Telefono	varchar(15)	non null
Fax	varchar(15)	
Email	varchar(50)	
SitoInternet	varchar(100)	
Referente	varchar(30)	non null
TipologiaAzienda	varchar(100)	
Prodotti	text	
Settore	varchar(100)	
PartecipatoPassato	char(2)	
PartecipaFiereSettore	text	
SuperficieRichiesta	int	
Commenti	text	
CodiceFieraEdizione	varchar(8)	non null, chiave esterna
Dol	datetime	predefinito con <code>getdate()</code>
Origine	varchar(50)	

Tabella 3.2: informazioni degli espositori memorizzati nel database

Lo scopo del campo *Origine* è spiegato nel paragrafo 3.5. In questa tabella sono presenti due chiavi esterne: *CodiceFieraEdizione*, valore che identifica da quale sito si registra e *Provincia*. La provincia viene scelta tramite un `DropDownList` (controllo che consente all'utente di selezionare un singolo elemento da un elenco a discesa) che mostra tutte le province in Italia che sono memorizzate in una tabella presente nel database. Permette di evitare di avere molte province identiche scritte diversamente in modo da agevolare successive query di ricerca. Nel form ci sono dei campi obbligatori da inserire, come la ragione sociale, il referente e il numero di telefono. Tramite il controllo `RequiredFieldValidator` è possibile rendere obbligatorio un qualsiasi campo di input e non è possibile continuare se non si compila tale campo.

Un esempio è:

```
<asp:RequiredFieldValidator runat="server" ID="RfvRagioneSociale"
ControlToValidate="TxtRagioneSociale" ErrorMessage="Ragione sociale obbligatoria!" />
```

La proprietà *ID* è l'identificativo dell'oggetto, obbligatorio anche se non è utilizzato. La proprietà *ControlToValidate* rappresenta il valore da verificare, identificandolo attraverso l'ID della casella di testo. Se il valore da verificare è vuoto, viene mostrato un messaggio di errore contenuto nella proprietà *ErrorMessage*, invitando l'utente a compilare il campo per continuare con l'applicazione.

Un altro controllo applicato a tutti i form creati è il *RegularExpressionValidator* che permette di verificare una stringa attraverso delle espressioni regolari. Viene utilizzato su campi il cui valore deve avere una certa forma standard, come l'email, il numero di telefono o fax. Un esempio per l'email è:

```
<asp:RegularExpressionValidator ID="RevEmail" runat="server"
ValidationExpression="^[0-9a-zA-Z]([-.\w]*[0-9a-zA-Z])*
@([0-9a-zA-Z]([-.\w]*[0-9a-zA-Z]\.)) +[a-zA-Z]{2,9}$"
ErrorMessage="Email non valida!" ControlToValidate="TxtEmail" />
```

La proprietà *ValidationExpression* contiene l'espressione regolare di cui al controllo in oggetto.

Una volta che l'utente ha compilato il modulo correttamente, clicca sul bottone "Invia" generando così un evento che permette di inserire i dati nel database e di inviare l'email al project manager. Per poter eseguire l'istruzione di INSERT, è necessario prima aprire una connessione con il database dopo che si è creato un oggetto *SqlConnection* in cui è incluso tutto il contenuto di *ConnectionStringSection* per la configurazione predefinita dall'applicazione, come specificato nel *web.config*. La stringa dell'istruzione di INSERT è la seguente:

```
Insert Into tb_web_RichiestaInformazioni(RagioneSociale, Indirizzo, Cap, Citta,
Provincia, Telefono, Fax, Email, SitoInternet, Referente, TipologiaAzienda,
Settore, PartecipatoPassato, PartecipaFiereSettore, SuperficieRichiesta, CodiceFieraEdizione)
Values(@RagioneSociale, @Indirizzo, @Cap, @Citta, @Provincia, @Telefono, @Fax,
@email, @SitoInternet, @Referente, @TipologiaAzienda, @Settore, @PartecipatoPassato,
@PartecipaFiereSettore, @SuperficieRichiesta, @CodiceFieraEdizione)
```


Per poter eseguire le istruzioni SQL si utilizza un oggetto `SqlCommand`, inizializzandolo con il testo della query e un oggetto `SqlConnection`. Per evitare attacchi di SQL injection (tecnica dell'hacking mirata a colpire le applicazioni web che si appoggiano su un database di tipo SQL), i valori dei campi vengono passati come parametri. Per fare questo si utilizza su `Values` il valore `@nomecampo`. Vengono quindi identificati i tipi di dati e poi passati i valori. Ad esempio:

```
cmd.Parameters.Add("RagioneSociale", SqlDbType.VarChar, 50);  
cmd.Parameters["RagioneSociale"].Value = TxtRagioneSociale.Text;
```

dove `cmd` è l'oggetto `SqlCommand` e `TxtRagioneSociale` è il controllo della casella di testo. Successivamente viene eseguita la query chiamando il metodo `ExecuteNonQuery`.

Una volta inserito il record nella tabella `tb_web_RichiestaInformazioni`, si manda un'email di riepilogo al project manager. Per fare questo si utilizza la classe `SmtpClient` che consente alle applicazioni di inviare posta elettronica utilizzando il protocollo SMTP. In ASP.NET è necessario fornire alla classe `SmtpClient` le impostazioni relative al relay server, quali l'hostname e la porta alla quale connettersi. Successivamente si passa alla classe `Smtp` tramite il metodo `Send` le caratteristiche dell'email creata su `MailMessage`. Per specificare il mittente, il destinatario e il contenuto di un messaggio di posta elettronica, si utilizzano le proprietà associate alla classe `MailMessage`, creando così il messaggio con i valori inseriti nel form dall'espositore.

Un'altra applicazione presente in ogni sito degli Sposi è la possibilità per gli utenti di scaricarsi la guida perfetta per il matrimonio. Scopo di questa applicazione è di acquisire informazioni con il consenso dell'utente da poi utilizzare per l'email marketing. Per avere l'email corretta, viene specificato nel testo che l'utente riceverà un messaggio di posta elettronica con il link della guida da scaricarsi, evitando così di trovarsi email che non esistono. Quindi si crea un semplice form di registrazione che contiene il nome, cognome, la provincia, l'email e la data presunta di matrimonio.

Tutte queste informazioni vengono memorizzate in una tabella chiamata `tb_web_GuidaSposiScaricata` e così strutturata:

Nome del campo	Tipo di dato	Vincolo applicato
Codice	int	non null, chiave primaria, identity
Nome	varchar(50)	non null
Cognome	varchar(50)	non null
Provincia	varchar(2)	non null
Email	varchar(50)	non null
DataPresunta	varchar(50)	
CodiceFieraEdizione	varchar(8)	non null, chiave esterna
DaQualeMailArriva	varchar(10)	
Dol	datetime	predefinito con <code>getdate()</code>

Tabella 3.3: informazioni dei visitatori perfetto memorizzati nel database che si scaricano la guida per il matrimonio

Il campo *DaQualeMailArriva* è stato utilizzato per monitorare certi invii di email (vedi paragrafo 3.5). L'istruzione di INSERT, con i nomi dei campi, della tabella e dei parametri diversi, e l'invio dell'email sono analoghi a quelli effettuati per la richiesta delle informazioni degli espositori.

Molte informazioni presenti in un sito possono essere identiche per vari siti, come ad esempio gli stand, gli arredi e accessori e le foto. Per questo motivo in ogni sito c'è una directory virtuale utilizzata per condividere una cartella che contiene dei file comuni. Nel database ci sono quindi delle tabelle accessorie con la descrizione dell'oggetto e un'altra tabella che associa tramite chiavi esterne gli oggetti che appartengono a quella specifica fiera.

L'esempio che si mostra è quello degli arredi e accessori. La tabella accessoria è chiamata `tb_web_ArredoAccessorio` ed è così strutturata:

Nome del campo	Tipo di dato	Vincolo applicato
Codice	int	non null, chiave primaria, identity
CodiceVisualizzato	varchar(5)	non null
DescrizioneBreve	varchar(50)	
Descrizione	text	
Prezzo	decimal(18, 2)	
TipoCosto	varchar(50)	
Valuta	char(3)	
Immagine	varchar(100)	
DoI	datetime	predefinito con <code>getdate()</code>

Tabella 3.4: descrizione arredi e accessori

La tabella che associa l'arredo alla fiera invece si chiama `tb_web_Fiera_ArredoAccessorio` ed è così strutturata:

Nome del campo	Tipo di dato	Vincolo applicato
CodiceFieraEdizione	varchar(8)	non null, chiave primaria, chiave esterna
CodiceArredoAccessorio	int	non null, chiave primaria, chiave esterna
DoI	datetime	predefinito con <code>getdate()</code>

Tabella 3.5: arredi e accessori associati alla fiera

Per visualizzare tutti gli arredi e accessori (più di trenta per manifestazione) si utilizza un webcontrol chiamato GridView che possiede molte funzionalità utili. E' possibile inserire la paginazione, inserire colonne e agganciarci a qualsiasi fonte di dati, sfruttando la tabella `tb_web_Fiera_ArredoAccessorio`. Un esempio è:

```
<asp:GridView runat="server" ID="GvArredoAccessori" AllowPaging="true" PageSize="5"
DataSourceID="SdsArredoAccessori" AutoGenerateColumns="true">
<Columns>
    <asp:BoundField HeaderText="Codice" DataField="CodiceVisualizzato" />
    <asp:ImageField HeaderText="Immagine" DataImageUrlField="Immagine"
DataImageUrlFormatString="./mm3/images/ArrediAccessori/{0}" />
</Columns>
```

```

    <asp:BoundField HeaderText="Descrizione" DataField="Descrizione" />
    <asp:BoundField HeaderText="Prezzo" DataField="Prezzo" />
</Columns>
</asp:GridView>

```

dove *mm3* è la directory virtuale comune e il *DataField* è il nome del campo della tabella. Tra le proprietà del webcontrol è presente *DataSourceID* che si associa a *SqlDataSource*, la quale consente di accedere ai dati che si trovano nel database con una query di ricerca adeguata.

Un altro webcontrol utilizzato per le foto gallery è stato il *DataList*. E' possibile infatti definire un layout a tabella e specificare in modo automatico su quante colonne ripetere, tramite la proprietà *RepeatColumns*. Di seguito è mostrato un esempio.

```

<asp:DataList ID="Fotogallery" runat="server" RepeatLayout="Table"
RepeatColumns="3" RepeatDirection="Horizontal">
    <ItemTemplate>
        "
alt="foto" />
    </ItemTemplate>
</asp:DataList>

```

Per popolare questo webcontrol viene eseguita la query di *SELECT* fatta alla tabella *tb_web_Photosgallery* che contiene tutte le informazioni necessarie per la foto gallery, come il percorso del file dell'immagine piccola e grande, eventuale descrizione della foto e la fiera a cui è associata. La struttura della tabella è la seguente:

Nome del campo	Tipo di dato	Vincolo applicato
Codice	int	non null, chiave primaria, identity
Descrizione	varchar(200)	
FileNameSmall	varchar(50)	non null
FileNameBig	varchar(50)	
CodiceFieraEdizione	varchar(8)	non null, chiave esterna
Dol	datetime	predefinito con getdate()

Tabella 3.6: descrizione della foto gallery

I record che compongono la pagina che dobbiamo visualizzare vengono passati al webcontrol tramite l'oggetto DataSet passato al DataList. Il codice è il seguente:

```
string StrSql = "Select FileNameSmall From tb_web_Photogallery Where  
CodiceFieraEdizione = " + CodFieraEdizione + "";  
SqlDataAdapter SdaRisultato = new SqlDataAdapter(StrSql, cnn);  
DataSet querydataset = new DataSet();  
SdaRisultato.Fill(querydataset);  
Fotogallery.DataSource = querydataset;  
Fotogallery.DataBind();
```

dove la stringa *StrSql* contiene la query di SELECT, *SdaRisultato* è l'oggetto *SqlDataAdapter* che funge da ponte tra un oggetto *DataSet* e SQL Server per il recupero dei dati, *querydataset* rappresenta un set completo di dati che include le tabelle in cui ci sono i dati, *Fotogallery* è l'identificativo del *DataList* e *cnn* è l'oggetto *SqlConnection*. Nel metodo *Fill* viene utilizzato in modo implicito l'oggetto *DataReader* per restituire i nomi e i tipi delle colonne utilizzate per creare le tabelle nel *DataSet*, nonché i dati per compilare le righe delle tabelle nel *DataSet*. Il metodo *DataBind* associa l'origine dati al controllo *DataList*.

3.4 REPORT MODIFICHE EFFETTUATE SU OGNI SITO

I siti a cui sono stati modificati contenuti e inserite nuove informazioni o applicazioni, sono stati:

- www.omnicomexpo.com
- www.dmexpo.it
- www.habitatfieraferrara.it
- www.ferrarasposinfiera.it
- www.ferrarafestival.it
- www.moacsanremo.it
- www.fantasyhobby.it
- www.arezosposinfiera.it
- www.fieraincontro.it
- www.veneziasposinfiera.it
- www.multimediatre.com

Per i primi tre siti elencati, quelli delle fiere di OmniCom Expo, DM expo e Habitat Fiera Ferrara, si è dovuto studiare il layout grafico già presente per uniformare le modifiche e gli inserimenti con lo stile applicato e precedentemente deciso. I file di questi siti, sono stati utili per capire come poter gestire i siti creati nel corso del tirocinio e sono serviti da esempio.

Il grafico in figura 3.20, mostra la percentuale di richieste di modifiche fatte dai project manager e commerciali e successivamente realizzate per ogni sito.

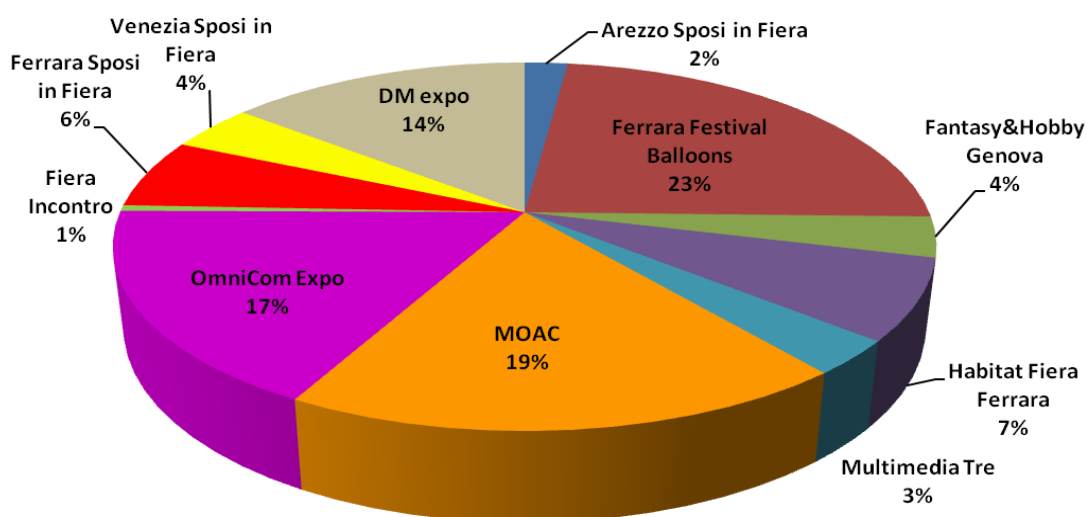


Figura 3.20: suddivisione per fiera delle richieste di modifiche

Le modifiche effettuate sono state di vario genere. I commerciali della fiera, non avendo certe informazioni al momento della pubblicazione del sito, passavano le informazioni sotto forma di richiesta da inserire nel sito alla segreteria segnalando le modifiche da effettuare. Tutte le tipologie di modifiche eseguite, sono state elencate di seguito:

- aggiungere nuovi articoli su rassegna stampa;
- aggiungere o modificare comunicati stampa con possibilità di scaricarli;
- aggiungere, sostituire o eliminare loghi o immagini tra patrocini, banner sponsor, partner o qualsiasi altro contenuto;
- aggiungere foto su fotogallery con possibilità di scaricarli (non sempre);
- aggiungere video su videogallery;
- aggiungere, sostituire o eliminare testi di pagine statiche;
- aggiungere, sostituire o eliminare voci del menù con creazione di nuove pagine laddove ce n'era bisogno;
- aggiungere, sostituire o eliminare file pdf (moduli per espositori o piantine guida);
- aggiungere form di registrazione per concorsi, biglietti ridotti o omaggio e moduli di informazioni per invio di email speciali;
- aggiungere pagine dinamiche con dati inseriti nel database e manipolabili da utenti perché di contenuto molto variabile (ad esempio programma della manifestazione, convegni, seminari, corsi e dimostrazioni);
- aggiungere banner della manifestazione sull'area download della stampa con file compressi;
- aggiungere banner in home page con ordine di visualizzazione casuale e che a rotazione ne mostra uno alla volta.

Le due applicazioni principali inserite durante il tirocinio sono state la possibilità di scaricarsi il biglietto ridotto e di vincere il biglietto omaggio. Lo scopo principale è quello di incitare i visitatori

a venire alla manifestazione fieristica con delle agevolazioni e nel frattempo, la possibilità per l'azienda di recuperare dati degli utenti da poi utilizzare per l'email marketing. Queste due applicazioni sono state inserite su Fantasy&Hobby Genova e Milano, Ferrara Sposi in Fiera e Habitat Fiera Ferrara. Per poter registrare i dati, dentro al database di Multimedia Tre si è creata una tabella che potesse contenere tutti i dati necessari per scaricare il biglietto ridotto e per vincere il biglietto omaggio, indipendentemente dal sito in modo da contenere tutti i dati all'interno di un'unica tabella. La tabella, con molte analogie alle tabelle precedentemente spiegate, si chiama `tb_web_BigliettoRidottoOmaggio` ed è così strutturata:

Nome del campo	Tipo di dato	Vincolo applicato
Codice	int	non null, chiave primaria, identity
Citta	varchar(50)	
Provincia	varchar(50)	
Email	varchar(50)	non null
Tipo	varchar(20)	non null
VintoOmaggio	bit	non null, predefinito false
CodiceFieraEdizione	varchar(8)	non null, chiave esterna
Dol	datetime	predefinito con <code>getdate()</code>

Tabella 3.7: informazioni dei visitatori memorizzati nel database che si scaricano il biglietto ridotto o giocano per vincere il biglietto omaggio

Il biglietto ridotto prevede un semplice form di registrazione in cui viene richiesta la città, la provincia e l'email. La città non è un campo obbligatorio perciò nella pagina web non c'è nessun `RequiredFieldValidator` e `RegularExpressionValidator`. La provincia e l'email invece sono campi obbligatori, quindi c'è il `RequiredFieldValidator` e per l'email c'è il `RegularExpressionValidator`. Una volta che l'utente compila i dati e sono scritti correttamente, i dati vengono memorizzati nel database, con la seguente query:

```
Insert Into tb_web_BigliettoRidottoOmaggio(Citta, Provincia, Email, Tipo, CodiceFieraEdizione)
Values(@Citta, @Provincia, @Email, @Tipo, @CodiceFieraEdizione)
```

Il campo `Codice` è chiave primaria e `identity`, quindi si auto incrementa da sola. I campi `Citta`, `Provincia` e `Email` contengono i dati inseriti dall'utente. Il campo `Tipo` è uguale alla stringa "ridotto". Il campo `VintoOmaggio` non ha nessun valore passato come parametro perché viene inserito a true solo se ho vinto il biglietto omaggio e quindi, per il biglietto ridotto è messo a false

per il vincolo impostato al campo. Il campo *CodiceFieraEdizione* ha il valore della variabile che identifica il sito (ad esempio HABFE10). L'ultimo campo *DoI* non ha nessun valore passato come parametro perché contiene l'informazione della data di creazione del record che viene presa dal vincolo impostato al campo con `getdate()`. Una volta che i dati sono stati inseriti correttamente e la query eseguita, viene visualizzato un messaggio con il link del file pdf del biglietto ridotto da scaricarsi e da stampare.

Il biglietto omaggio prevede un semplice form di registrazione come il biglietto ridotto. La tabella in cui vengono memorizzati i dati è la stessa. Il gioco prevede che ogni 30 utenti che provano a vincere il biglietto omaggio, ci sia un vincitore. Non è possibile per un utente che si presenta con la stessa email di giocare più di una volta al giorno. La regola del gioco è molto semplice e prevede una specie di slot machine con delle immagini che ruotano e che se ha vinto, mostra alla fine tre immagini uguali e se ha perso, ne mostra tre diverse. Per la rotazione delle immagini si sono utilizzati dei javascript che venivano attivati dopo la registrazione e che facevano ruotare le immagini per un certo tempo richiamando una funzione che cambiava le immagini in modo casuale con il metodo `setTimeout` (funzione javascript che serve per far partire dopo un certo tempo un'altra funzione). Il primo passo da fare è quello di controllare se l'utente ha già giocato nello stesso giorno. Per fare questo si esegue la query:

```
Select * From tb_web_BigliettoRidottoOmaggio Where CodiceFieraEdizione = "" + CodFieraEdizione + "" AND Email = "" + TxtEmail.Text + "" AND Tipo = 'omaggio' AND YEAR(DoI) = " + dataOggi.Year + " AND MONTH(DoI) = " + dataOggi.Month + " AND DAY(DoI) = " + dataOggi.Day
```

dove *dataOggi* è un dato `DateTime` inizializzato con il valore `DateTime.Now` che contiene la data presente nel server.

Per poter eseguire un'istruzione di `SELECT` si utilizza l'oggetto `SqlDataReader` che fornisce un metodo per la lettura di un flusso di righe di tipo `forward-only` da un database `SQL Server`. Per crearlo, è necessario chiamare il metodo `ExecuteReader` dell'oggetto `SqlCommand` a cui è stato passato la stringa della query e l'oggetto `SqlConnection`. Successivamente si applica all'oggetto `SqlDataReader` il metodo `Read` che ritorna `true` se sono presenti più righe e in caso contrario, `false`. Inoltre questo metodo sposta l'oggetto `SqlDataReader` al record successivo. Alla fine, per liberare l'oggetto `SqlConnection` si chiama il metodo `Close`.

Di seguito è mostrato il codice.

```
string StrSql = "query select";
SqlCommand cmd = new SqlCommand(StrSql, cnn);
SqlDataReader sdr = cmd.ExecuteReader();
if (!sdr.Read())
    //istruzioni
sdr.Close();
```

dove *cnn* è l'oggetto *SqlConnection*. Se la query legge almeno un valore, allora viene dato un messaggio che invita l'utente a giocare nei giorni successivi. Se invece è la prima volta che prova a giocare in quel giorno, viene mostrato a video una specie di slot machine (in figura 3.21) in cui può tentare la fortuna.



Figura 3.21: slot machine del gioco per vincere il biglietto omaggio

Per poter controllare la vincita, ci si aiuta con una tabella ausiliaria che mantiene il conteggio dei tentativi di vincita del biglietto omaggio. La tabella si chiama *tb_web_ConteggioBiglietto* ed è così strutturata:

Nome del campo	Tipo di dato	Vincolo applicato
Contatore	int	non null
CodiceFieraEdizione	varchar(8)	non null, chiave primaria, chiave esterna

Tabella 3.8: conteggio dei tentativi di vincita del biglietto omaggio

L'applicazione controlla per la propria fiera a che punto è arrivato il contatore. Si esegue perciò la seguente query:

```
Select Contatore From tb_web_ConteggioBiglietto  
Where CodiceFieraEdizione = '"+CodFieraEdizione+'"
```

Si stabilisce la vincita controllando il valore che si estrae con il numero di tentativi prestabiliti. Successivamente si incrementa di un'unità il valore del contatore se si ha perso, altrimenti lo si azzerà. Per fare questo, si utilizza l'istruzione SQL di UPDATE:

```
Update tb_web_ConteggioBiglietto Set Contatore="+n+"  
Where CodiceFieraEdizione="'+CodFieraEdizione+'"
```

dove *n* è la variabile intera con il nuovo valore da inserire nel database. Il passo successivo prevede di inserire nella tabella *tb_web_BigliettoRidottoOmaggio* i dati dell'utente aggiungendo alla query utilizzata per il biglietto ridotto il valore booleano per il campo *VintoOmaggio* con l'esito della giocata e mettendo la stringa "omaggio" nel campo *Tipo*. L'ultimo passo prevede di mostrare un messaggio con l'esito della vittoria dell'utente: se si è perso, vengono mostrate tre immagini diverse; se invece si è vinto, vengono mostrate tre immagini uguali e inviata un'email da stampare con il biglietto omaggio vinto con all'interno l'email indicata all'interno del messaggio per identificare il giocatore.

Un'altra tabella utilizzata su Habitat Fiera Ferrara e Ferrara Balloons Festival durante il tirocinio è stata la tabella *tb_web_Convegni*. E' stata utilizzata per le pagine dinamiche che avevano contenuto molto variabile, come il programma della manifestazione, convegni, seminari, corsi e dimostrazioni. Inoltre, per non creare una pagina statica con il contenuto della descrizione per ogni voce del programma, si inseriva nello stesso record anche la descrizione di questa e poi, tramite dei parametri che identificavano la voce del programma e che vengono passati a una pagina particolare, si poteva mostrare tutto il contenuto. Questo sistema ha permesso di diminuire la mole di lavoro e la possibilità di manipolare i dati, tramite opportune interfacce predisposte, direttamente dai commerciali di Multimedia Tre.

La tabella tb_web_Convegni, con molti campi a disposizione, è così strutturata:

Nome del campo	Tipo di dato	Vincolo applicato
Codice	int	non null, chiave primaria, identity
Tipo	varchar(50)	
Data	varchar(100)	
Ora	varchar(100)	
Padiglione	varchar(100)	
Titolo	varchar(500)	non null
Contenuto	text	
Link	varchar(200)	
ACuraDi	varchar(300)	
Sottotitolo	varchar(300)	
Pubblicato	bit	
CodiceFieraEdizione	varchar(8)	non null, chiave esterna
Dol	datetime	predefinito con getdate()

Tabella 3.9: descrizione di convegni e programmi

Con tutti questi campi, è possibile tramite opportune query, visualizzare nel miglior modo le informazioni e di gestire eventuali approfondimenti molto semplicemente. La visualizzazione di tutte le informazioni sono state studiate con la responsabile della grafica e il project manager.

3.5 REPORT EMAIL CREATE

Un altro obiettivo del tirocinio è stato quello di contribuire per l'email marketing, creando dei file HTML che rispettassero il più possibile le specifiche descritte alla fine del paragrafo 2.5. La grafica di alcune email sono state create dalla responsabile della grafica e successivamente revisionate con un codice HTML avente le caratteristiche corrette. La maggior parte dei file HTML richiedevano delle immagini. Queste venivano caricate dentro la sotto cartella "email" della cartella "images" presente in ogni sito e nel codice veniva scritto il percorso corretto per l'attributo src del tag , come ad esempio:

<http://www.veronasposinfiera.it/images/email/logo.jpeg>

L'email realizzate in totale sono state 40, di cui:

- 6 per Ferrara Festival Balloons
- 2 per Fantasy&Hobby Genova
- 9 per Habitat Fiera Ferrara
- 2 per Fiera Incontro
- 7 per MOAC
- 2 per Multimedia Tre
- 3 per OmniCom Expo
- 2 per Arezzo Sposi in Fiera
- 3 per Ferrara Sposi in Fiera
- 1 per Milano Sposi in Fiera
- 3 per Venezia Sposi in Fiera

Il grafico in figura 3.22 mostra le percentuali di email create per la specifica fiera.

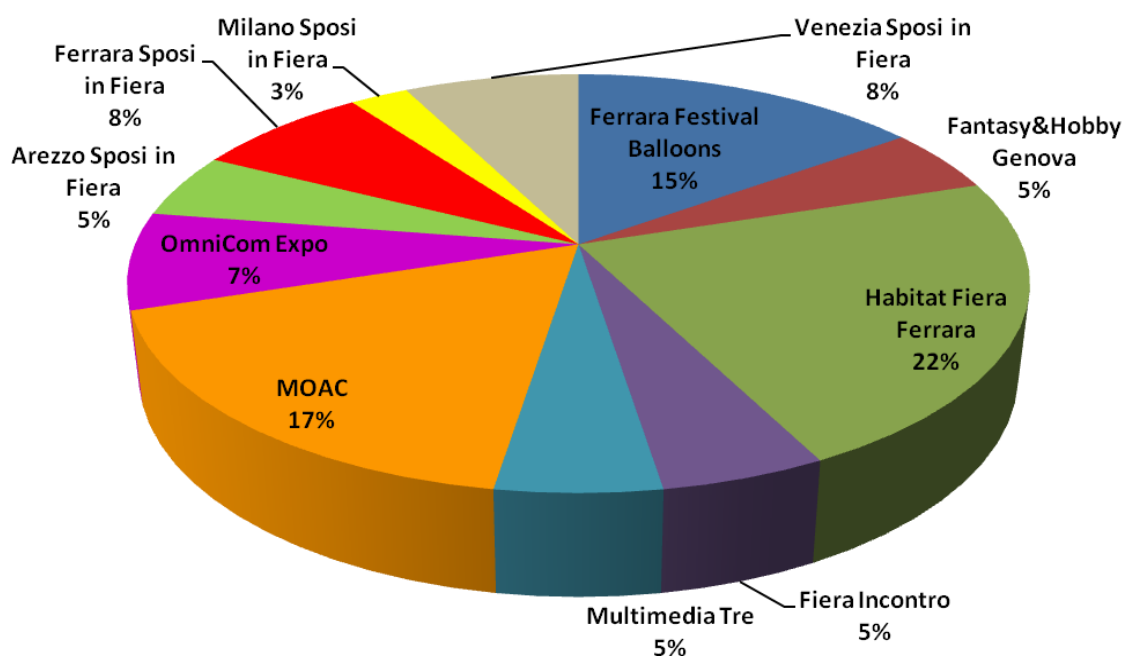


Figura 3.22: suddivisione per fiera di email richieste

L'email create sono state utilizzate per tre scopi:

- invio a tutte l'email presenti negli archivi dell'azienda;
- invio da parte di fornitori esterni;

- newsletter per invii specifici da parte dei commerciali dell'azienda.

L'email inviate a tutti i dati archiviati sono state mandate a potenziali visitatori ed espositori alle manifestazioni fieristiche. Ad esempio, i visitatori venivano invitati alle fiere informandoli delle date e gli orari con la possibilità di scaricarsi i biglietti omaggio o ridotti. Gli espositori invece venivano informati della possibilità di esporre alla fiera e dei vantaggi che potevano avere. Per questi ultimi, se desideravano ricevere informazioni sugli spazi espositivi potevano, attraverso un link specifico, richiedere ulteriori informazioni compilando un form. Per il project manager era importante capire quanti espositori si registravano al form per quell'invio specifico. Questo è stato possibile attraverso un monitoraggio avvenuto tramite il passaggio di parametri nei link. Particolare attenzione si deve dare alla privacy, sempre presente in ogni invio di email fatto da parte di Multimedia Tre, che dice:

Informativa per il trattamento dei dati personali legge 196/2003 (codice in materia di protezione dei dati personali). State ricevendo questo comunicato in virtù di precedenti rapporti o della vs. presenza su pubblici registri. Potete richiedere modifiche o cancellazione dei vs. dati in qualsiasi momento (art.7 d.l. 196/2003). Per richiedere la cancellazione dei vostri dati personali cliccate qui. Per richiedere modifiche dei vostri dati personali potete farlo via mail a privacy@multimediatre.com, oppure via fax al n. 0499830728, oppure oralmente (0499832150) a Multimedia Tre Srl che è titolare e responsabile del trattamento dei dati. Qualora questo messaggio fosse da voi ricevuto per errore, nel rispetto della legge sulla tutela dei dati personali, vogliate cortesemente darcene notizia e distruggere il messaggio ricevuto erroneamente.

La cancellazione dell'utente, come descritto nella privacy, poteva avvenire anche tramite un link. Se l'email veniva inviata a potenziali espositori, questa veniva monitorata segnalando a quale fiera apparteneva l'invio per poi cancellare nei prospect interni dell'azienda, creati per ogni manifestazione, i dati del potenziale espositore. Se l'email veniva inviata a potenziali visitatori, questa non veniva monitorata perché i dati non sono presenti nei prospect utilizzati dai commerciali ma in un'altra banca dati che possiede tutti i dati dei visitatori.

Un altro scopo per il quale sono state create l'email, è stato l'invio da parte di fornitori esterni. Questi possedevano dati di potenziali visitatori i quali, con l'email inviata, avevano la possibilità di scaricarsi i biglietti omaggio o ridotti o altri servizi. Non essendo inviate da Multimedia Tre, non

contenevano la privacy sopra citata ma quella delle fonti esterne. Anche qui per il project manager era importante capire quanti visitatori accedevano al servizio offerto registrandosi a form specifici. Questo è stato possibile attraverso un monitoraggio avvenuto tramite il link. Certi fornitori chiedevano sia l'email in formato HTML ma anche in formato PDF.

Le newsletter utilizzate da parte dei commerciali dell'azienda per invii a specifici espositori, venivano fatte direttamente dal commerciale una volta raggiunto un possibile accordo telefonicamente o via fax. Queste newsletter contenevano informazioni della manifestazione fieristica, come la descrizione, costi degli stand, date e orari, la campagna pubblicitaria. Poiché queste email venivano mandate a espositori che hanno dato il loro consenso ad essere informati, non c'è bisogno di privacy. Inoltre, per poter personalizzare l'email a loro piacimento, i commerciali richiedevano la newsletter generale in formato EML (messaggio di posta elettronica Outlook Express) e MSG (messaggio di posta elettronica Outlook, non Express).

Come è stato più volte citato in questo paragrafo, un'operazione importante da fare è il monitoraggio. Per poterlo fare è stato passato un parametro tramite il metodo di invio GET adeguato per l'uso. Ad esempio per la cancellazione di un espositore, il parametro assume il valore del codice dell'edizione della fiera, come "SPOVR11". Il parametro viene passato nel link indicato per la cancellazione, identificando con una chiave, nel seguente modo:

<http://www.miosito.it/Pagina.aspx?chiave=parametro>

Nel vettore Request.QueryString si indica la chiave utilizzata per recuperare il parametro con le parentesi quadre, ad esempio:

Request.QueryString["chiave"]

Se il parametro è diverso da null, allora è possibile recuperare il parametro con una variabile di tipo stringa e tale stringa è utilizzata per memorizzare i dati nel database. Il codice di esempio per recuperare il parametro è:

```
string parametro = (Request.QueryString["chiave"] == null) ? "" :  
Request.QueryString["chiave"].ToString();
```

Successivamente, prima di eseguire la query, si può controllare se la stringa ricavata è vuota, nulla oppure un numero, a seconda dell'applicazione da eseguire. Inoltre si può controllare se il parametro è proprio effettivamente quello passato dal link per evitare di danneggiare il database con codice non corretto.

3.6 REPORT BACKUP EFFETTUATI

L'ultimo obiettivo del tirocinio è stato quello di fare dei backup di siti internet di manifestazioni fieristiche concluse in un server locale dell'azienda, permettendo la visione di questi esclusivamente ai collaboratori di Multimedia Tre.

Per fare questo si è utilizzato un terminal server che esegue anche altre funzioni legate all'azienda. In questo terminal server si è creata una cartella condivisa e accessibile a tutti gli utenti legati ad esso in cui ci sono varie sotto cartelle, una per ogni backup effettuato. Una volta installato IIS, il compito da eseguire è stato quello di scaricarsi tramite FileZilla Client una copia del sito internet, copiarlo nella cartella adeguata nominata con il nome del sito e l'anno (ad esempio "www.veronasposinfiera.it - 2010"). Una volta effettuato il backup, si impostano le configurazioni di IIS per far funzionare il sito web creando una directory virtuale collegata alla cartella in cui è stato copiato tutto il contenuto, come descritto nel paragrafo 3.14. L'ultimo passo da eseguire è stato quello di aggiornare un file excel in comune a tutti i collaboratori di Multimedia Tre in cui si ricordano i link da utilizzare per vedere il sito di manifestazioni passate (ad esempio "http://192.168.1.10/veronasposinfiera_2010.it") e una breve descrizione dell'edizione della fiera.

I backup dei siti internet effettuati durante il tirocinio sono stati fatti principalmente perché sono stati creati i nuovi siti per le nuove edizioni. In totale, i backup fatti seguendo la procedura precedentemente esposta, sono stati 13 e riguardano le seguenti edizioni delle fiere:

- Habitat Fiera Ferrara edizione 2009
- Ferrara Sposi in Fiera edizione 2009
- Venezia Sposi in Fiera edizione 2009
- Fiera Incontro edizione 2009-2010
- Fantasy&Hobby Ferrara edizione 2010
- Pollice Verde edizione 2010

- Verona Sposi in Fiera edizione 2010
- Fantasy&Hobby Genova edizione marzo 2010
- Liberamente Trieste edizione 2010
- Storia in Fiera edizione 2010
- Liberamente Ferrara edizione 2010
- Fantasy&Hobby Casa Natale Ferrara edizione 2009
- Multimedia Tre

Durante il download di un sito, particolare attenzione si è dovuta dare alla directory virtuale, che viene utilizzata per condividere una cartella con dei file comuni (soprattutto immagini utili per gallerie, stand e accessori per gli arredi) accessibili a più siti. Quindi durante la procedura di backup bisogna stare attenti a scaricare file inerenti al sito e non file che riguardano siti di altre fiere, per evitare di sprecare spazio con file inutili.

Capitolo 4

CONCLUSIONI

Durante il tirocinio svolto è stato possibile eseguire ogni compito concordato inizialmente con il tutor aziendale. Sono stati creati nove siti e successivamente è stato possibile gestirli autonomamente. Inoltre, si è dato un contributo all'azienda creando delle applicazioni web utili e agevolando l'email marketing. Il risultato finale è stato superiore alle aspettative perché si sono conosciute e realizzate praticamente diverse cose. Tra le nozioni più importanti che si sono acquisite, ci sono la tecnologia ASP.NET con tutto il suo funzionamento e le varie istruzioni utili offerte. E' stato di notevole interesse anche studiare teoricamente come l'usabilità del web è un elemento fondamentale per l'utente. Si è cercato di rispettare il più possibile nei siti web gestiti i vari criteri esposti alla fine del paragrafo 2.1. Un'altra cosa che è stata di notevole interesse è stata la manipolazione dei dati sul database. Creare tabelle, aggiornare record, inserire dati tramite istruzione SQL, eseguire query di ricerca: tutti compiti che sono utili per Multimedia Tre e che hanno permesso di acquisire una professionalità che arricchisce la formazione. Interessante è stato l'ampio uso dell'email marketing e come sia importante rispettare la privacy degli utenti. Fondamentali sono state le relazioni che si sono create dentro all'azienda con i vari commerciali e con la responsabile della grafica per poter effettuare i compiti in totale indipendenza e per poter acquisire informazioni utili. I vari siti gestiti sono attualmente utilizzati e le applicazioni create sono ancora in uso per permettere a Multimedia Tre di compiere i loro scopi aziendali.

Bibliografia

Fabrizia Scorzoni, Informatica DataBase. Loescher, 2009.

Gruppo HTML, Database HTML.IT, <http://database.html.it>

Gruppo HTML, ASP.Net HTML.IT, <http://aspnet.html.it>

Gruppo HTML, Java HTML.IT, <http://java.html.it>

Gruppo HTML, PHP HTML.IT, <http://php.html.it>

Gruppo HTML, Programmazione HTML.IT, <http://programmazione.html.it>

Gruppo HTML, Server HTML.IT, <http://server.html.it>

Gruppo HTML, Web design HTML.IT, <http://webdesign.html.it> ai

Jakob Nielsen e Hoa Loranger, Web Usability 2.0 L'usabilità che conta. Apogeo, 2006.

Louis Rosenfeld e Peter Morville, Architettura dell'informazione per il world wide web. Hops Libri, 2002.

Marco Ferrero, SQL Pocket. Apogeo, 2004.

Mario de Ghetto, SQL Server 2008 Guida all'uso. Edizioni FAG Milano, marzo 2010.

Microsoft Corporation, MSDN, risorse per sviluppatori, abbonamenti MSDN, video tutorial e download, <http://msdn.microsoft.com>

Microsoft Corporation, Visual Studio .NET Procedure dettagliate. 2002.

Mr. Webmaster, Il network dei Webmaster e dei professionisti ICT, <http://www.mrwebmaster.it>

Nazzareno Gorni e Marco Maglio, E-mail Marketing Strategie e tecniche efficaci per fare business. Hoepli, 2009.

NetSorting, CardSorting, <http://www.cardsorting.info>

Paolo Atzeni, Stefano Ceri, Stefano Paraboschi e Riccardo Torlone, Basi di dati Modelli e linguaggi di interrogazione. McGraw-Hill, 2006.

Patrice Pelland, Visual C# 2008 Express Edition. Mondadori, 2008.

Roberto Ghislandi, Email Marketing. Alpha Test, giugno 2009.

Shinynews, Rivista di usability diritto e-business marketing & advertising e internet economy
<http://www.shinynews.it>

Webmasterpoint, Il punto di riferimento per il professionista ICT, <http://www.webmasterpoint.org>

Acronimi

- API Application Program Interface: insieme di procedure, protocolli e strumenti per realizzare un'applicazione software
- ASP Active Server Pages: tecnologia lato server ideata da Microsoft
- CFML Cold Fusion Markup Language: linguaggio di markup per scrivere applicazioni per il web e le intranet
- CGI Common Gateway Interface: tecnologia lato server che riguarda l'interazione tra web server e differenti fonti di dati come applicativi e database presenti nel web server
- CLR Common Language Runtime: la base di .NET Framework in quanto gestisce il codice in fase di esecuzione, fornendo servizi di base come la gestione della memoria, gestione di thread e servizi remoti, attivando una rigida indipendenza dei tipi, garantendo protezione ed efficienza
- CSS Cascading Style Sheets: fondamentale linguaggio standard del W3C per gestire tutto il layout di un sito Web
- CSV Comma Separated Values: formato di file basato su file di testo utilizzato per l'importazione ed esportazione di una tabella di dati
- DBMS Database Management System: sistema software progettato per consentire la creazione e manipolazione efficiente di database
- FTP File Transfer Protocol: protocollo per la trasmissione di dati tra host
- GIF Graphics Interchange Format: formato per immagini di tipo bitmap molto utilizzato nel web per le animazione e in secondo piano per le immagini fisse

HTML	HyperText Markup Language: linguaggio usato per controllare la struttura dei documenti ipertestuali disponibili nel web
HTTP	Hypertext Transfer Protocol: sistema per la trasmissione di informazioni sul web
IIS	Internet Information Services: server web per sistemi operativi di Microsoft Windows
IMAP	Internet Message Access Protocol: protocollo di comunicazione per la ricezione di e-mail
IP	Internet Protocol: protocollo di rete su cui si basa la rete Internet
JPEG	Joint Photographic Experts Group: primo standard internazionale di compressione per immagini a tono continuo, sia a livelli di grigio che a colori
JSP	Java Server Pages: tecnologia lato server basata su Java, il linguaggio creato da Sun
JVM	Java Virtual Machine: macchina virtuale che esegue i programmi scritti in byte code
NNTP	Network News Transport Protocol: protocollo usato dal servizio internet Usenet, la cui applicazione comune sono i cosiddetti Newsgroup
ODBC	Open Database Connectivity: API standard per la connessione ai DBMS
OLE	Object Linking and Embedding: tecnologia per la creazione di documenti composti sviluppata da Microsoft
PDF	Portable Document Format: formato di file basato su un linguaggio di descrizione di pagina per rappresentare documenti in modo indipendente dall'hardware e dal software
PHP	Pre Hypertext Processor: tecnologia lato server che utilizza un linguaggio open source completo di scripting, sofisticato e flessibile, che può girare su qualsiasi web server, su qualsiasi sistema operativo e soprattutto di interagire con i principali tipi di database
PNG	Portable Network Graphics: un formato di file per memorizzare immagini

POP3	Post Office Protocol 3: protocollo che ha il compito di permettere, mediante autenticazione, l'accesso ad un account di posta elettronica presente su di un host per scaricare le e-mail del relativo account
RDBMS	Relational Database Management System: indica un database management system basato sul modello relazionale
ROI	Return On Investment: indica la redditività e l'efficienza economica della gestione caratteristica a prescindere dalle fonti utilizzate
SMTP	Simple Mail Transfer Protocol: protocollo standard per la trasmissione via internet di e-mail
SNMP	Simple Network Management Protocol: protocollo che opera al livello 7 del modello OSI, consentendo la gestione e la supervisione di apparati collegati in una rete, rispetto a tutti quegli aspetti che richiedono azioni di tipo amministrativo
SQL	Structured Query Language: linguaggio di interrogazione per database progettato per leggere, modificare e gestire dati memorizzati in un sistema basato sul modello relazionale
SQLCLR	Structured Query Language Common Language Runtime: permette a SQL Server di supportare una piena integrazione con il .NET Framework
TCP	Transmission Control Protocol: protocollo di livello di trasporto della suite di protocolli Internet
T-SQL	Transact - Structured Query Language: estensione proprietaria del linguaggio SQL sviluppata da Microsoft e Sybase
URL	Uniform Resource Locator: sequenza di caratteri che identifica univocamente l'indirizzo di una risorsa in Internet
WAB	Windows Address Book: applicazione che ha un database locale ed un'interfaccia utente che permette agli utenti di avere una lista di contatti che può essere condivisa da vari programmi
WAS	Web Administration Service: gestore di Pool che ha il compito di controllare e monitorare lo stato delle applicazioni web e di riciclare i worker process
WYSIWYG	What You See Is What You Get: quello che vedi è quello che ottieni

- XHTML eXtensible HyperText Markup Language: linguaggio di marcatura che associa alcune proprietà dell'XML con le caratteristiche dell'HTML
- XML eXtensible Markup Language: un linguaggio marcatore che definisce un meccanismo sintattico che consente di estendere o controllare il significato di altri linguaggi marcatori