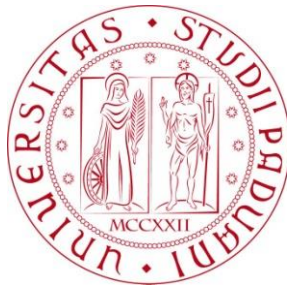


Università degli Studi di Padova  
Dipartimento di Scienze Statistiche  
Corso di Laurea Triennale in  
Statistica e Tecnologie Informatiche



RELAZIONE FINALE

## **Analisi dei sistemi CMS per la costruzione di siti web**

**Using Content Management System to create web sites**

Relatore Prof. Ombretta Gaggi  
Dipartimento di Matematica

Laureanda: Giada Signorin  
Matricola N° 1010253

Anno Accademico 2012 / 2013



*Un sincero ringraziamento alla mia famiglia che mi ha permesso di proseguire gli studi e mi ha sempre sostenuto; alla docente Ombretta Gaggi che mi ha seguito nella stesura di questo elaborato con grande disponibilità e ai miei compagni universitari, che hanno reso questi tre anni indimenticabili.*



# Indice

<b>Capitolo 1: INTRODUZIONE</b> .....	9
1.1 OBIETTIVI E MODALITÀ DELLO STAGE .....	10
1.2 DESCRIZIONE DELL'AZIENDA.....	10
1.3 RIASSUNTO DEL LAVORO SVOLTO EFFETTIVO .....	11
1.4 STRUMENTI UTILIZZATI.....	12
<b>Capitolo 2: I CONTENT MANAGEMENT SYSTEM</b> .....	13
Premessa: siti web static e dinamici .....	13
2.1 CARATTERISTICHE DI UN CMS .....	13
2.1.1 Gestione dei contenuti .....	15
2.1.2 Più utenti con un solo CMS .....	16
2.1.2.1 Amministratore .....	16
2.1.2.2 Editore .....	17
2.1.2.3 Autore.....	17
2.1.2.4 Collaboratore .....	17
2.1.3 I template grafici .....	17
2.1.4 Gli standard .....	18
2.1.5 SEO friendly .....	18
2.1.6 Strumenti aggiuntivi: i Plugin .....	18
2.2 LA STORIA .....	19
2.3 I PRINCIPALI CMS .....	20
2.3.1 Come scegliere un CMS .....	23
2.4 COME FUNZIONA UN CMS .....	23
2.4.1 Creare un sito in locale .....	24
2.4.2 Pubblicare un sito.....	28
2.4.3 Backup di un sito.....	29
2.5 I PROTOCOLLI.....	30
2.5.1 TCP/IP.....	30

2.5.2 HTTP.....	32
2.5.3 FTP.....	33
2.6 I LINGUAGGI UTILIZZATI .....	34
2.6.1 HTML .....	34
2.6.2 CSS.....	37
2.6.3 PHP.....	41
2.6.3.1 La gestione dei database .....	42

## **Capitolo 3: CREARE UN SITO WEB CON WORDPRESS**.....43

3.1 Prima fase: INSTALLAZIONE DI WORDPRESS .....	44
3.1.1 Configurazione del Virtual Host .....	44
3.1.2 Creazione del database .....	45
3.1.3 Installazione .....	46
3.1.4 Struttura del back-end di un sito Wordpress .....	48
3.2 Seconda fase: INSERIMENTO DEI CONTENUTI E PERSONALIZZAZIONE DELLA GRAFICA DEL SITO .....	49
3.2.1 Personalizzazione della grafica.....	49
3.2.1.1 Come realizzare un tema figlio .....	51
3.2.2 Inserimento dei contenuti.....	53
3.2.2.1 Creazione di una pagina.....	53
3.2.2.2 Creazione dell’Homepage: lo Slideshow.....	56
3.2.2.3 Creazione di un articolo.....	58
3.2.2.4 Utilizzo di “Catalog”.....	61
3.2.2.5 Gestione delle Sidebar e dei Widget.....	65
3.2.2.6 Il menu.....	69
3.2.3 Problemi legati al tema.....	70
3.2.3.1 Duplicazione di un Widget.....	71
3.2.3.2 Cambiare una parola a livello di file.....	72
3.3 Terza fase: PUBBLICAZIONE DEL SITO.....	74
3.3.1 Il database.....	74
3.3.2 Trasferimento file tramite FTP.....	74

3.4 ALCUNI PLUGIN UTILIZZATI .....	75
3.4.1 WPML .....	75
3.4.2 Magic Fields .....	76
3.4.3 Contact Form 7 .....	77
3.4.4 NextGen Gallery .....	77
3.4.5 WooCommerce .....	78
<b>Capitolo 4: CONCLUSIONI</b> .....	<b>81</b>
Bibliografia .....	83





# Capitolo 1

## Introduzione

Con il presente lavoro si intende presentare l'esperienza fatta durante lo stage svolto nel periodo dal 22 aprile 2013 al 20 agosto 2013 presso Websonica S.r.l., azienda specializzata nella creazione di siti web e di Web Marketing, spiegando passo passo uno dei progetti da me sviluppati.

Lo scopo dello stage, concordato con il tutor aziendale Massimo Sterlino, consiste nello sviluppo di siti web tramite l'utilizzo di uno dei CMS in uso presso l'azienda: Wordpress.

Per il primo periodo, mi è stato necessario consultare e studiare manuali inerenti al linguaggio PHP, in quanto nella mia carriera studentesca non l'avevo mai utilizzato; ciò non ha costituito un grosso ostacolo, poiché ho potuto fare spesso riferimento alle mie conoscenze di altri linguaggi quali C, C# e Java.

Inoltre, ho dedicato molto tempo a consultare documentazioni riguardanti il CMS che poi avrei utilizzato. Ciò che ho appreso da queste attività va ad affiancare la formazione che il mio tutor mi forniva.

Indispensabili sono state le mie conoscenze riguardanti il web, apprese nel corso di "Sistemi Informativi" seguito nel terzo anno universitario, di "Reti di calcolatori" seguito nel secondo anno, e di programmazione e di gestione dei database che ho potuto maturare nei corsi di "Sistemi di elaborazione" e "Basi di dati" del primo anno.

Nell'arco delle ore di stage svolto ho avuto una stretta collaborazione con Alessia Nastasio e Andrea Rampazzo, che rappresentano la parte creativa dell'azienda e che si occupano della creazione del layout grafico e di tutta l'estetica che sta alla base di un sito; con Alberto Riolfo che si occupa del marketing e con Roberto Biasiolo e Michele Zamuner che si occupano invece della programmazione.

Oltre alla creazione di siti con Wordpress, ho avuto modo di realizzare newsletter e landing page, scrivendo in codice html pagine che fossero identiche alla grafica fornitami da un file psd (creato con Photoshop); grazie a questa esperienza ho dovuto far fronte ai problemi legati alle diverse visualizzazioni nei browser e ho acquisito una maggior capacità di utilizzo dei CSS (Cascading Style Sheets), prestando particolare attenzione ai layout flessibili rispetto a quelli fissi.

## 1.1 OBIETTIVI E MODALITÀ DELLO STAGE

Gli obiettivi dello stage sono:

- Sviluppo di soluzioni software su CMS o Custom finalizzate all'affinamento delle tecniche necessarie alla costituzione di una Web Area in grado di far fronte alle progettualità che il mercato richiede;
- Webmarketing.

Il tirocinante è stato costantemente seguito dal tutor aziendale nello svolgimento dei suoi compiti al fine di ampliare le sue conoscenze tecniche e di consolidare quelle già acquisite nel corso degli studi.

## 1.2 DESCRIZIONE DELL'AZIENDA

Fino a pochi anni fa, Internet ha rappresentato uno strumento che solamente un gruppo ristretto di persone erano in grado di sfruttare.

Negli ultimi tempi però, grazie allo sviluppo delle moderne tecnologie e alla crescita del numero e tipologia di servizi che offre, esso è diventato una realtà a cui ogni giorno migliaia e migliaia di persone fanno ricorso: basti pensare a ciò che porta la comodità di possedere un indirizzo email o alle innumerevoli informazioni a cui una persona può accedere semplicemente effettuando una veloce ricerca nel web.

Un aspetto molto importante che ha contribuito a promuovere la diffusione dell'utilizzo di Internet è la facilità nell'accesso e del suo utilizzo, fatto che lo rende effettivamente alla

portata di tutti; non servono specifiche competenze o strumenti altamente tecnologici, è sufficiente possedere un qualsiasi computer, smartphone o tablet in grado di connettersi al web. Ciò ha permesso la creazione di un ampio canale di comunicazione e scambio di informazioni in tempo reale in tutto il mondo.

Questo strumento ha contribuito a rivoluzionare non solo le realtà individuali delle singole persone, ma ha portato grandi cambiamenti anche nel modo di lavorare e di farsi conoscere delle aziende, che hanno sentito fin da subito la necessità di sfruttare questo strumento, data la sua ampia diffusione.

Avere un proprio sito web che ogni utente possa facilmente accedere e consultare è diventato ormai di primaria importanza per il successo di qualsiasi impresa; bisogna dunque provvedere sin dall'inizio alla creazione di un sito facilmente reperibile, utilizzando strumenti di Web Marketing.

Di tutto ciò si occupa Websonica S.r.l., azienda che da 10 anni si muove nel mondo di Internet e che offre ai suoi Clienti un servizio comprensivo di Comunicazione, Marketing e Software al fine di realizzare soluzioni create appositamente per un certo tipo di business dal quale trarne i migliori benefici.

Il suo staff si divide in due sezioni diverse, cui competono specifici compiti: la prima si occupa di Comunicazione e Marketing, dedicata alla ricerca di soluzioni per ogni cliente con cui creare una certa immagine che si coordini alla sua azienda, e promuovendola sui Social Network con apposite strategie di Web Marketing; la seconda si occupa invece della parte di programmazione, utilizzando diversi software e linguaggi di programmazione per realizzare al meglio il sito vero e proprio o Web Application consultabili da qualsiasi dispositivo mobile.

## **1.3 RIASSUNTO DEL LAVORO SVOLTO EFFETTIVO**

Durante il periodo di stage svolto, il mio compito è consistito nel compiere diverse mansioni.

In particolare:

- creazione di landing page;
- creazione di siti web;
- modifica di siti web;

- backup di siti web, svolti ripetutamente durante la fase di creazione e di pubblicazione online;
- pubblicazione di siti web, effettuati tramite FTP;
- realizzazione di newsletter, create da zero, in HTML.

## 1.4 STRUMENTI UTILIZZATI

Gli strumenti da me utilizzati durante il periodo di stage mi sono stati forniti dall'azienda, predisposti dal tutor aziendale ed erano costituiti da un computer comprensivo del software necessario già installato.

Nel computer, con piattaforma Windows 7, era presente:

- WAMP Server, indispensabile per il funzionamento di siti realizzati in locale con i CMS;
- FileZilla, utilizzato come client FTP con il quale pubblicare i siti web nei domini acquistati dai clienti;
- Heidi SQL, software in grado di gestire i database MySQL;
- Mail aziendale su dominio gmail e Skype, per la comunicazione interna.
- GitHub, software con il quale sincronizzare un sito in remoto con quello in locale;
- NetBeans, ambiente di sviluppo multi-linguaggio in cui creare/modificare le pagine in linguaggio HTML e PHP;
- Firebug, estensione di Mozilla Firefox che permette di visualizzare come è fatta una pagina web e che consente la modifica del CSS e del codice HTML permettendo di vedere in tempo reale gli effetti che certe modifiche hanno sul debug della pagina.

# Capitolo 2

## I Content Management System

### **Premessa: siti web statici e dinamici**

Per spiegare quali sono le funzionalità di un CMS (Content Management System) e le sue principali caratteristiche, occorre innanzitutto effettuare una distinzione tra due tipologie diverse di siti web: i siti statici e i siti dinamici.

I siti web statici prevedono un'interazione sostanzialmente unilaterale: l'utente può visualizzare immagini e dati di un sito ma non modificarne lo stato, elaborare contenuti e informazioni.

I siti web dinamici, invece, consentono una maggiore interazione con l'utente, poiché le informazioni sono aggiornate continuamente e le pagine vengono generate in base al tipo di richiesta effettuata. Per la loro natura, essi sono più complessi da gestire.

Proprio al fine di gestire grandi quantità di dati, i siti dinamici sono per necessità collegati a database, che dovranno essere continuamente aggiornati in concomitanza al sito.

Queste complesse procedure sono oggi semplificate grazie all'utilizzo di appositi software, i CMS, in grado di automatizzare la creazione delle pagine web evitando l'utilizzo diretto del codice particolarmente sofisticato che sta alla loro base, velocizzando inoltre le operazioni di modifica.

### 2.1 CARATTERISTICHE DI UN CMS

Un Content Management System è un'applicazione lato server che si appoggia su un database preesistente per lo stoccaggio dei contenuti.

Esso permette lo sviluppo di un sito senza la necessità di particolari conoscenze, poiché l'interfaccia nasconde il codice di programmazione che sta alla base, permettendo la creazione di siti personalizzati anche da parte di chi ha solamente una minima conoscenza in materia.

I contenuti vengono infatti redatti tramite un editor visuale molto intuitivo che permette una formattazione grafica della scrittura e l'inserimento di contenuti aggiuntivi mediante URL o caricamento di immagini; si tratta di un editor di tipo WYSIWYG ("What You See Is What You Get" e cioè "Ciò che vedi è quello che ottieni"), dove ciò che appare in fase di scrittura sarà uguale al risultato finale.

Un CMS è suddiviso sostanzialmente in due parti:

- una sezione di amministrazione o gestione (**back-end**), che serve ad organizzare e supervisionare la produzione dei contenuti; l'amministratore del CMS infatti gestisce dal proprio terminale, tramite un pannello di interfaccia e controllo a cui accede mediante username e password, i contenuti da inserire o modificare;
- una sezione applicativa (**front-end**), che l'utente web usa per fruire dei contenuti e delle applicazioni del sito; essa è la parte pubblica del sito.

Le modifiche al sito possono essere fatte esclusivamente da Back-end, al quale si accede generalmente dalla pagina [www.nomedelpropriosito/administration/](http://www.nomedelpropriosito/administration/).

I CMS possono essere realizzati in vari linguaggi web (tra cui, più comunemente, ASP, PHP, Microsoft .NET); il tipo di linguaggio adoperato è indifferente a livello di funzionalità. Esistono inoltre CMS specializzati, cioè appositamente progettati per un tipo preciso di contenuti (un'enciclopedia on-line, un blog, un forum, una rivista, ecc.) e CMS generici, che tendono ad essere più flessibili per consentire la pubblicazione di diversi tipi di contenuti.

Le caratteristiche che contraddistinguono un CMS da un normale editor di siti web e che lo rendono preferibile ad un sito realizzato da zero sono numerose; le più importanti sono legate alla gestione dei contenuti e degli utenti, alla facile gestione della grafica e per il fatto che queste applicazioni generano pagine HTML rispettando gli standard web (vedi sezione 2.4.1).

### **2.1.1 Gestione dei contenuti**

Letteralmente, l'acronimo CMS significa "Sistema di Gestione di Contenuti".

Questo nome mette in risalto quella che evidentemente è la sua caratteristica principale: dopo aver deciso la grafica con la quale si vuole realizzare il sito (aspetto che può essere facilitato dal ricorso all'utilizzo di uno dei numerosi template grafici creati appositamente per i CMS) ci si deve preoccupare esclusivamente dell'inserimento dei contenuti.

I problemi di formattazione del testo, di posizionamento delle immagini o di come disporre degli articoli all'interno di una pagina, sono tutti gestiti dal CMS.

Tutto ciò assume particolare importanza se si applica, ad esempio, alla creazione di un sito dove vengono pubblicate informazioni che per necessità devono essere continuamente aggiornate in tempo reale (come nel caso della creazione di siti di giornali online) oppure un sito di e-Commerce che offre particolari promozioni in base al periodo dell'anno in cui ci si trova e che deve aggiornare la gamma di prodotti che offre, in base alle richieste dei clienti e ai loro bisogni continuamente in mutazione.

Se non esistessero i CMS, ogni qualvolta si dovessero modificare le informazioni fornite da un sito, si dovrebbero creare o modificare i file sorgente, un lavoro esclusivo per coloro che conoscono l'HTML (a differenza di quanto accade con i content editor forniti dall'interfaccia dei CMS che si occupano del solo contenuto).

Oltre a questo, vi sarebbe la necessità di ricreare ogni volta la grafica opportuna, adattando i cambiamenti portati con la formattazione generale del sito modificato.

È facile intuire come tutto ciò non sia un passaggio immediato, ma richieda personale altamente qualificato oltre a grandi quantità di tempo da dedicare al complesso lavoro legato alle continue modifiche; inoltre, la propensione a commettere errori sarebbe molto più elevata e, data l'importanza che rivestirebbe il lavoro della persona dedicata a quello specifico compito, i costi aziendali subirebbero un notevole incremento a causa della sua retribuzione.

Tramite l'utilizzo dei CMS tutto questo non è più necessario: basta infatti accedere al back-end del proprio sito, andare nella sezione dedicata all'inserimento di un nuovo articolo o nella pagina da modificare e inserire testi ed eventuali immagini negli appositi spazi; la pubblicazione sarà fatta in modo immediato, dopo aver dato la conferma delle modifiche riportate. Non saranno così possibili errori di posizionamento immagini, di formattazione o codifica.

## **2.1.2 Più utenti con un solo CMS**

Un'altra caratteristica che contraddistingue un CMS da un normale editor di siti web riguarda la gestione degli utenti. Un CMS prevede una serie di funzionalità che definiscono le risorse alle quali una specifica categoria di utente potrà accedere, gestendo migliaia di utenti con il minimo sforzo.

Ciò significa che quando si intraprende la creazione di un sito, si possono definire gli utenti e la loro classe di appartenenza, in modo tale che determinati utenti possano accedere esclusivamente a specifiche aree.

Ciò costituisce un aspetto di particolare importanza: si pensi alle conseguenze che l'accesso a determinate informazioni o aree di gestione di un sito adatte alla sola consultazione e modifica da parte di personale altamente formato e competente possa avere se effettuato da parte di un individuo non competente che ingenuamente le modifica; può compromettere l'intero lavoro frutto di giorni e giorni di applicazione.

Grazie ai CMS, questi problemi sono risolti, poiché sono necessari specifici permessi per l'accesso a determinate sezioni.

In generale, le categorie previste dai CMS sono le seguenti:

- Amministratore
- Editore
- Autore
- Collaboratore.

### **2.1.2.1 AMMINISTRATORE**

L'amministratore è l'utente che ha un accesso di tipo completo e che ha il pieno controllo sulla gestione del sito: egli può cambiare i ruoli degli altri utenti, attivare o disattivare i plugin, intervenire sui contenuti di pagine, post e categorie, cambiare il layout del sito e modificare il codice sorgente; in poche parole, è la figura che gerarchicamente si pone al di sopra delle altre.

In un singolo sito la figura di amministratore può essere attribuita ad un solo utente o più; in quest'ultimo caso, però, occorre prestare particolare attenzione, poiché ogni utente che ricopre questo ruolo ha la possibilità di intervenire e modificare ogni parte del sito, compresa la zona riservata all'attribuzione delle varie classi ai relativi utenti; potrebbe dunque capitare



che, nel caso di controversie o disguidi, un amministratore si ritrovi declassato da altri amministratori, non potendo quindi più intervenire direttamente per ripristinare il suo ruolo di primaria importanza.

#### 2.1.2.2 EDITORE

L'editore ha i permessi per inserire, modificare e cancellare le pagine, gli articoli, le categorie, i tag; può intervenire sulle immagini e tutti i media in generale e occuparsi della gestione dei link.

#### 2.1.2.3 AUTORE

L'autore ha il permesso di modifica e cancellazione degli articoli e di inserimento di immagini e file.

#### 2.1.2.4 COLLABORATORE

Il collaboratore può creare e modificare solo i suoi articoli ma non può pubblicarli; quando viene creato, un articolo può essere pubblicato da un amministratore, un editore o un autore prima della revisione. L'articolo pubblicato non può essere più modificato dal collaboratore se non viene riportato in stato di revisione.

Il collaboratore non è abilitato all'inserimento di immagini o file.

### 2.1.3 I Template grafici

I CMS offrono la possibilità di installare dei template grafici, cioè grafiche già pronte, disponibili a migliaia sul web in modo gratuito o a pagamento.

L'utilizzo di questi template è di rilevante importanza in quanto con un minimo sforzo si dà al proprio sito una grafica di alta qualità e, nel momento in cui la si volesse rinnovare completamente, sarà sufficiente scaricare un template diverso ed installarlo.

Nel caso in cui uno di questi template già fatti si avvicinasse a quello che si vuole ottenere come risultato finale, vi è la possibilità di creare un tema figlio a partire da quello originale, modificando solo gli aspetti che si vogliono cambiare, dando luogo così ad una grafica personalizzata e preservando la qualità del template di partenza.

#### **2.1.4 Gli standard**

Chiunque abbia avuto a che fare con la creazione di siti web è a conoscenza del fatto che vi sono degli standard da seguire.

L'ambiente di lavoro dei CMS può essere visualizzato con qualsiasi browser e il sito è sviluppato secondo gli standard del W3C (World Wide Web Consortium).

I siti web realizzati con queste applicazioni sono inoltre conformi alle linee guida sull'accessibilità dettate dalla Legge Stanca, che illustra le disposizioni per favorire l'accesso dei soggetti disabili agli strumenti informatici.

Questa normativa definisce il concetto di accessibilità come "la capacità dei sistemi informatici, nelle forme e nei limiti consentiti dalle conoscenze tecnologiche, di erogare servizi e fornire informazioni fruibili, senza discriminazioni, anche da parte di coloro che a causa di disabilità necessitano di tecnologie assistive o configurazioni particolari".

#### **2.1.5 SEO friendly**

Per quanto riguarda il SEO (Search Engine Optimization), la maggior parte dei CMS prevede un sistema di creazione delle pagine del sito in modo da ottimizzare il loro ritrovamento da parte dei motori di ricerca, ricorrendo all'utilizzo di tag e all'url rewriting, che permette di creare URL delle singole pagine e sezioni senza caratteri incomprensibili e di facile memorizzazione per l'utente.

Inoltre, viene generata automaticamente la mappa del sito con tutti i collegamenti e le sezioni all'interno delle pagine e tra le pagine, organizzando il tutto in maniera gerarchica.

#### **2.1.6 Strumenti aggiuntivi: i Plugin**

Un plugin è un programma non autonomo che, interagendo con un altro programma, ne amplia le funzionalità.

I CMS hanno la capacità di supportare questo software aggiuntivo e ciò è molto importante, in quanto rende possibile l'ampliamento e la personalizzazione delle sue funzioni, in maniera relativamente semplice e veloce.

I plugin in circolazione risolvono oramai la maggior parte dei problemi legati alla personalizzazione di un sito, e il loro utilizzo è molto semplice: i CMS offrono, infatti, un'apposita interfaccia per la loro installazione e aggiornamento.

Nel caso in cui il sito richieda qualcosa di più specifico, vi è la possibilità di creare un proprio plugin e di integrarlo al software del CMS utilizzato, personalizzando ulteriormente il proprio sito secondo specifiche richieste.

## 2.2 LA STORIA

I CMS sono nati negli Stati Uniti per opera di alcune organizzazioni che producevano notevoli quantità di pubblicazioni. Inizialmente essi vennero sviluppati per uso interno aziendale.

Nel novembre del 1995 Ross Garber e Neil Webber fondarono la Vignette Corporation con lo scopo di creare qualcosa che rendesse più facili e personalizzabili le pubblicazioni sul web. Nell'anno successivo, Vignette fornì il proprio software a CNET, azienda che tutt'oggi vanta il posto di leader nel settore tecnologico informatico e che si occupa della pubblicazione di articoli, recensioni, file multimediali e analisi sempre aggiornati, che lo utilizzò per la gestione del suo sito [news.cnet.com](http://news.cnet.com).

Successivamente, nel 1998 la compagnia di consulenza aziendale Pencom Web Works introdusse il server di trasformazione dei dati (DTS - Data Transformation Services) Metaphoria, che permetteva agli sviluppatori Java di scrivere applicazioni che si potessero collegare ai contenuti e distribuire tali contenuti su canali diversi.

Anche se il prodotto non ebbe successo, esso introdusse il concetto che costituì le basi di ciò che è diventato il CMS odierno.

La prima piattaforma utilizzata prendeva il nome di story server e fu il punto di partenza da cui si basarono i programmatori per creare i CMS attuali.

## 2.3 I PRINCIPALI CMS



Creato da Matt Mullenweg e rilasciato per la prima volta il 27 Maggio 2003, Wordpress è ideale per la realizzazione di siti di dimensioni contenute (poche centinaia di pagine o meno), con un'organizzazione relativamente semplice; è perfetto per i blog.

È il sistema più semplice da installare, utilizzare ed aggiornare.

I template grafici disponibili sono infiniti, sia gratuiti che a pagamento, e lo stesso vale per i plugin.

Nel campo del blogging, le potenzialità di questo CMS sono indiscutibili, ma non è adatto per creare e-Commerce o portali con molte pagine in quanto non è in grado di gestire siti di grandi dimensioni (non "scala" altrettanto bene come altri CMS in circolazione).

Uno dei grandi vantaggi dell'utilizzo di Wordpress è che essendo utilizzato in grande scala, la maggior parte delle soluzioni ai problemi, sia banali che complessi, la si può trovare nel forum ufficiale, che si estende ai numerosi sviluppatori di temi e plugin, che in base alle segnalazioni di bug riescono a migliorarne il software



Joomla nasce nel settembre 2005, tutelato da una licenza open source GNU GPL, versione 2.0.

Come Wordpress, anche Joomla può contare su un buon supporto in rete e di un elevato numero di plugin e template in circolazione, ma soprattutto ha una capacità di gestione degli utenti davvero efficace.

Joomla è infatti la soluzione ideale per siti con molti utenti e contenuti da gestire, che presentano un'elevata complessità di gestione, come i portali.

È però poco intuitivo e complesso per gli utenti meno esperti, per questo ne è sconsigliato l'utilizzo per siti di dimensioni contenute: in quel caso il miglior CMS resta Wordpress.

Uno dei limiti di Joomla è che ogni singolo contenuto è in genere associato ad una sola pagina: il sistema diventa quindi più semplice da capire ma può essere pesante da aggiornare: inoltre vi è la

limitazione di associare una sola categoria per articolo, il che può rendere necessaria la duplicazione di contenuti.



Uno dei CMS più utilizzati quando si ha a che fare con la gestione un e-Commerce è Magento, nato il 31 Marzo 2008 con l'obiettivo di soddisfare le esigenze di chi vuole creare uno shop online.

È perfetto per progetti di grandi dimensioni e per chi ha necessità specifiche in quanto si ha un controllo completo della piattaforma e le possibilità di personalizzazione sono svariate.

Prevede un'ottimizzazione SEO con controllo dei meta dati, di landing page e di sitemap per Google ed ha una massima attenzione per le modalità di pagamento.

Comprende una vasta gamma di funzioni per e-Commerce, ma necessita di hosting particolari con risorse hardware significative, ed è molto complesso nella gestione di template e del pannello di amministrazione.



Creato nel 2001 e tutelato da licenza GNU GPL come Joomla, Drupal è un'ottima scelta per i siti più complessi.

Supporta una vasta gamma di strutture di sito e grazie ai componenti aggiuntivi è possibile definire regole dettagliate per personalizzare alcuni tipi di contenuti. Il pacchetto base di Drupal non prevede un editor di tipo WYSIWYG, ma lo si può tranquillamente installare, rendendo più semplice la gestione dei contenuti e degli aggiornamenti per gli amministratori. Le numerosi opzioni e impostazioni che rendono tale CMS flessibile e potente, lo rendono però molto complesso, e capire cosa offre il sistema e come configurarlo è più difficile rispetto a Wordpress e Joomla.



Plone è un sistema potente e robusto, adatto per soddisfare richieste molto complesse. Viene utilizzato per i siti di importanti giornali e aziende, e grazie all'efficiente struttura degli strumenti di amministrazione dei contenuti, è di facile utilizzo anche per categorie di amministratori "non tecnici".

Le funzionalità di Plone sono alla pari o addirittura superiori rispetto agli altri CMS, tranne per quanto riguarda il supporto di community.

Il principale difetto di questo CMS sta nell'installazione e configurazione: Plone richiede infatti un hosting con determinate caratteristiche, e non un tipico ambiente Linux / Apache / MySQL / php come accade invece per gli altri CMS.

L'integrazione di nuovi temi grafici e di plugin aggiuntivi è tutt'altro che banale, e capire come funziona Plone richiede una formazione specifica anche per i tecnici esperti; inoltre a

differenza degli altri sistemi dove viene utilizzato il linguaggio php, è scritto in Python e questo rende più difficile trovare qualcuno che intervenga sul codice per personalizzarlo.

### **2.3.1 Come scegliere un CMS**

Quando ci si trova di fronte alla scelta di un CMS per la gestione di un progetto web occorre valutare numerosi parametri; tra i più significativi vi sono la facilità di utilizzo, la possibilità di personalizzazione dell'interfaccia grafica, la possibilità di integrare strumenti per arricchire il sito, la compatibilità con i sistemi server disponibili.

Prima di scegliere il CMS è comunque importante capire quale sarà il suo uso, ovvero quale architettura si andrà a gestire; dati i numerosi vantaggi e svantaggi che ogni diverso CMS offre, non ce n'è quindi uno migliore di altri, ma la scelta dovrà essere effettuata in base a quali funzionalità si vogliono sfruttare e quali aspetti del sito potenziare, ponendo in primo piano certi aspetti piuttosto che altri.

Il CMS da utilizzare sarà dunque quello che più si avvicina alle esigenze evidenziate, in base alla tipologia di sito web che si ha intenzione di gestire.

## **2.4 COME FUNZIONA UN CMS**

Un CMS è un'applicazione "server-side": ciò significa che per funzionare, deve essere installato su un server web.

Questa necessità deriva dal fatto che certe operazioni devono essere compiute lato server in quanto richiedono l'accesso a informazioni o funzionalità non disponibili lato client, o richiedono misure di sicurezza che altrimenti non sarebbero affidabili.

Dato un ambiente client/server in un'architettura di protocolli TCP/IP (Transmission Control Protocol / Internet Protocol) e HTTP (HyperText Transfer Protocol), programmare lato server significa creare applicazioni che andranno eseguite nel server, accettando richieste dal client; una volta elaborate le informazioni richieste, le restituisce al client come pagine html.

La programmazione lato server si differenzia da quella lato client in quanto il linguaggio utilizzato lato client viene interpretato direttamente dal browser; oggi si sceglie spesso questa soluzione in quanto è più facilmente gestibile.

Vi sono diversi linguaggi con i quali programmare lato server; i CMS sono scritti per la maggior parte in PHP.

Essendo un'applicazione programmata lato server, il contenuto generato dal CMS non esiste fisicamente ma è il risultato di una combinazione tra il codice e i dati provenienti dal database.

Per far funzionare un sito creato con CMS si ha dunque bisogno di:

- spazio su un server con le caratteristiche appropriate;
- un database.

Quando si vuole realizzare un sito, è bene svilupparlo in locale sul proprio computer e pubblicarlo online in un secondo momento, per evitare che qualcuno facendo una ricerca nel web lo trovi e lo visualizzi incompleto; se invece il sito è già stato realizzato e si trova online, ci sono due modi per procedere a fronte di eventuali modifiche.

Nel caso in cui fossero semplici, come nascondere una pagina, correggere un errore di ortografia o cambiare un'immagine, allora la si può fare modificando il sito da beck-end direttamente online.

Se invece le modifiche da apportare fossero più complesse, come installare un plugin, modificare i template o eseguire degli aggiornamenti, allora è di buon auspicio creare una copia del sito sul proprio computer, eseguire tutte le modifiche richieste e testare la funzionalità del sito prima di renderlo pubblico.

## **2.4.1 CREARE UN SITO IN LOCALE**

Come detto in precedenza, un sito creato con CMS ha bisogno di essere installato su un server per funzionare, quindi salvarlo nel disco di un normale computer non basta: è necessario scaricare un apposito programma, il WAMPServer.



Il WAMPServer formato da un pacchetto di software di sviluppo web / database che implementa la piattaforma WAMP che a sua volta è implementazione adattata a Windows della piattaforma AMP (LAMP è quella adattata per GNU/LINUX), che prende il nome dalle iniziali dei componenti software con cui è realizzata:

- **A**pache , il web server;
- **M**ySQL , il DBMS (Database Management System) con SQLite e relativi tool grafici (come PhpMyAdmin);
- **P**hp , il linguaggio di scripting.

Grazie a questo ambiente di sviluppo si è in grado di far funzionare pagine dinamiche in locale, e inoltre vengono installati gli applicativi PHPMyAdmin e SQLiteManager che aiutano nella creazione e gestione del database collegato.

Nella mia esperienza di lavoro svolta durante lo stage, come applicativo per la gestione dei database ho utilizzato un programma a parte, HeidiSQL.

Nato per la gestione di database MySQL, HeidiSQL consente la connessione a più database contemporaneamente e questo dà la possibilità di creare / modificare tabelle, utilizzare query e trasferire dati da un server all'altro in maniera diretta.

Grazie alla sua interfaccia grafica, il lavoro di gestione dei database è stato semplificato e consente di applicare query o funzioni al database in modo immediato; anche i privilegi di amministrazione possono essere gestiti in modo semplice.

Una volta eseguita l'installazione del WampServer, sarà possibile creare e visualizzare tutti i siti dinamici che si vorranno realizzare, ma si dovranno seguire dei semplici passi.

### **Primo passo: il database.**

Per funzionare, un sito costruito a partire da un CMS ha bisogno di un database nel quale memorizzare i dati.

Si crea dunque un database che in fase di installazione sarà associato a quel particolare sito.

## Secondo passo: configurazione del server Apache.

Se il database è indispensabile, di pari importanza è la configurazione del server Apache. Apache supporta due tipi di domini virtuali:

- IP based – dove ogni dominio è associato ad un differente IP;
- name based – con il quale avere più domini associati allo stesso IP (con http 1.1).

In entrambi i casi, viene usato il container VirtualHost nel quale sono definite le direttive che stanno alla base della configurazione di un server web.

Data la scarsità di indirizzi IP disponibili, la metodologia maggiormente utilizzata è la seconda; se si sceglie questa soluzione, bisogna innanzitutto andare nella cartella di sistema:

```
C : \ WINDOWS \ system32 \ drivers \ etc \
```

e aggiungere una semplice riga al file “hosts”, formata dall’indirizzo IP del proprio host locale, indicato con 127.0.0.1 o semplicemente con “localhost”, e il nome del sito che si va a realizzare, per esempio:

```
127.0.0.1    www.nomedelsito.com
```

(nel caso in cui il sito si trovasse in un’altra macchina, per poterlo visualizzare basterà modificare questa riga mettendo al posto di “127.0.0.1” l’indirizzo IP del computer nel quale risiede fisicamente il sito).

Per configurare il VirtualHosts, basta invece modificare il file “httpd-vhosts.conf” che si può trovare nella directory:

<Dove è stato installato il server WAMP> \ bin \ apache \ Apache2.2.17 \ conf \ extra  
aggiungendo le seguenti righe:

```
<VirtualHost *:80>

    ServerAdmin webmaster@esempio.org
    ServerName www.nomedelsito.com

    ServerAlias www.nomedelsito.it
    DocumentRoot "C:\wamp\www\nomedelsito"
    #File di log
```

```
    ErrorLog
    "C:\wamp\www\log\www.nomedelsito.com\error.log"
    CustomLog
    "C:\wamp\www\log\www.nomedelsito.com\access.log"
</VirtualHost>
```

Dove:

- ServerAdmin – indica la mail del creatore del sito web;
- ServerName – è il nome del sito web al quale farà riferimento il server Apache per individuare la risorsa in questione;
- ServerAlias – questo campo va a sostegno del serverName, in quanto anch'esso sarà preso a riferimento dal server Apache per indicare il sito. Grazie a questo campo aggiuntivo il virtualHost risponderà sia per "www.nomedelsito.com" (ServerName) che per "www.nomedelsito.it" (ServerAlias);
- DocumentRoot – indica la directory in cui è stato salvato il sito (nell'esempio riportato sopra, "nomedelsito" corrisponde alla cartella che contiene tutti i file che compongono il codice sorgente);
- ErrorLog – indica in quale file (specificandone la directory) verranno scritti gli errori di log, e cioè tutti gli errori legati al dominio del virtualhost (in questo caso, di "www.ilmiosito.com");
- CustomLog – indica in quale file (specificandone la directory) verranno scritte le azioni degli utenti.

## Terzo passo: installazione del CMS

Dopo aver creato il database e aver configurato il virtual host, è il momento di installare il CMS.

Il software dovrà essere scaricato dal corrispondente sito ufficiale all'interno della cartella indicata nel campo "DocumentRoot" del virtualHost (nel nostro esempio, nella cartella "nomedelsito" che si trova nella directory C:\wamp\www).

Dopo aver fatto questa operazione, si riavvia il WampServer (se tutte le operazioni sono state fatte nel modo corretto, l'icona diventerà verde) e si scrive l'indirizzo del sito nella barra riservata all'URL di un qualsiasi browser.

A questo punto, se tutto è andato a buon fine, saranno visualizzate direttamente nel browser le istruzioni di installazione del CMS, in cui verranno impostati username e password dell'admin del sito e il collegamento al database.

Una volta completati i vari step, sarà possibile visualizzare sul proprio computer il front-end del proprio sito o accedere al back-end tramite le credenziali inserite in fase di installazione per cominciare a modificare o creare il sito.

## **2.4.2 PUBBLICARE UN SITO**

Una volta che il sito è stato realizzato, si passa alla fase di pubblicazione, uno dei momenti più delicati di tutto il lavoro.

Per prima cosa, bisogna chiedere al cliente l'approvazione ufficiale di quanto è stato fatto e una volta ricevuto l'ok si concorda la data e l'ora in cui il sito verrà pubblicato, richiedendo gli accessi al server e al database che il cliente ha acquistato per il sito. Per pubblicare un sito online è infatti necessario usufruire di un servizio di hosting che consiste nella possibilità di allocare un sito su un server web, rendendolo così accessibile dalla rete Internet e dai suoi utenti.

Nel caso in cui il sito sia un e-Commerce, è necessario che il cliente abbia avuto l'approvazione del modulo COM6BIS o che siano trascorsi 30 giorni dalla comunicazione per esercitare il tacito assenso.

Se tutto è pronto, si passa alla pubblicazione vera e propria: innanzitutto, se il cliente aveva già un sito web, si esegue il backup integrale della precedente situazione online.

Successivamente, si trasferiscono tutti i file del nuovo sito tramite FTP e si fa un'importazione del database che era stato utilizzato per la realizzazione del sito.

Per copiare i file tramite FTP (File Transport Protocol) è sufficiente l'utilizzo di uno dei tanti software che funzionino da client FTP, e cioè che utilizzano tale protocollo per la spedizione dei dati (ad esempio, FileZilla).

Per funzionare, questo tipo di software ha bisogno di varie informazioni per configurare l'hosting:

- Un nome per l'host del sito (www.ilmiosito.com)
- Selezionare che il server è di tipo FTP
- Username e Password per il login dell'utente (per poter pubblicare il sito, si ha bisogno di avere accesso alla root: ecco perché tali accessi devono essere forniti dal cliente).

Per quanto riguarda il database, la sua importazione è fondamentale: tutti i contenuti inseriti infatti sono memorizzati al suo interno, e se non si compisse questa operazione, il sito pubblicato sarebbe vuoto.

Una volta che tutti i file saranno stati copiati e il database importato, bisogna modificare il file di configurazione del CMS con i nuovi accessi al database e da quel momento sarà visibile da qualsiasi utente.

Importante: per poter visualizzare la versione del sito online dal computer nel quale era stato creato, è indispensabile togliere l'host che era stato impostato in fase di installazione del CMS nel file "hosts" e per fare ciò è sufficiente mettere a commento la riga che configurava il sito appena pubblicato; nell'esempio, diventa:

```
#127.0.0.1 www.ilmiosito.com
```

Nel caso in cui il cliente volesse effettuare dei test per cambiare il proprio sito, sarà sufficiente togliere il cancelletto, riavviare il WampServer e richiamando il sito nel browser si visualizzerà quello salvato in locale.

## 2.4.3 BACKUP DI UN SITO

È fondamentale, in fase di creazione del sito, eseguire numerosi backup.

Questa operazione è necessaria per prevenire che l'installazione di un plugin o un aggiornamento rovinino il frutto di giorni e giorni di lavoro ma anche per tutelarne la

sicurezza: avere una versione funzionante del sito può permetterlo di ripristinarlo in caso di malfunzionamenti o attacchi hacker.

Per eseguire correttamente il backup di un sito creato con l'ausilio di un CMS basta copiare l'intero contenuto della cartella situata nel server remoto sul nostro pc e fare un dump del database.

Il dump è un riepilogo della struttura delle tabelle del database e dei relativi dati, ed è formato da una lista di dichiarazioni SQL. È necessario in fase di backup in quanto è una specie di copia del database, e grazie ad esso si possono ripristinare i contenuti del caso di perdita di dati.

## **2.5 I PROTOCOLLI: TCP/IP, HTTP E FTP**

### **2.5.1 TCP/IP**

Il modello TCP/IP è l'unione di due protocolli: il TCP (Transmission Control Protocol) e l'IP (Internet Protocol). È il protocollo di rete per eccellenza: esso detta le regole per l'instradamento dei pacchetti (i dati da trasmettere vengono suddivisi in pacchetti che viaggiano autonomamente) all'interno della rete, permettendo che le informazioni giungano da un mittente ad un destinatario attraversando migliaia di chilometri e centinaia di sistemi intermedi (router) senza che l'utente se ne accorga.

Il protocollo IP definisce come devono essere organizzati tali pacchetti, mentre il protocollo TCP assicura che tutti giungano realmente a destinazione.

Il TCP/IP è basato sulla struttura Client-Server del tipo Peer-to-Peer: un sistema che implementa questo protocollo può cioè funzionare da Client e da Server indifferentemente e addirittura cambiare la propria identità in fase di elaborazione.

Ha una connessione di tipo full duplex a commutazione di pacchetto: non esiste un'autorità centralizzata nella rete, ma un qualunque nodo è collegato ad un qualsiasi altro, trovandosi quindi allo stesso livello gerarchico. Ciascun nodo può creare, elaborare e trasmettere informazioni.

Per poter applicare questo protocollo a tutti i terminali, indipendentemente dal sistema operativo, il protocollo TCP/IP è stato scomposto in più moduli.

Ogni modulo svolge un determinato compito, e l'ordine in cui essi vengono svolti è ben preciso, con un sistema stratificato (ragione per cui si parla di modello a livelli).

I dati che transitano sulla rete attraversano più livelli di protocolli, in modo tale che i pacchetti siano trattati livello dopo livello, ognuno dei quali aggiunge al pacchetto un elemento di informazione (chiamato intestazione) e lo passa al livello successivo.

Il TCP/IP ha avuto una grande diffusione nella connessione delle reti grazie alla sua semplicità.

È molto simile all'architettura del modello ISO-OSI (costituito da 7 livelli, elaborato dall'Organizzazione Internazionale di Standardizzazione ISO, per standardizzare la comunicazione tra computer), come si può vedere in figura 2.1.

### Modello ISO-OSI

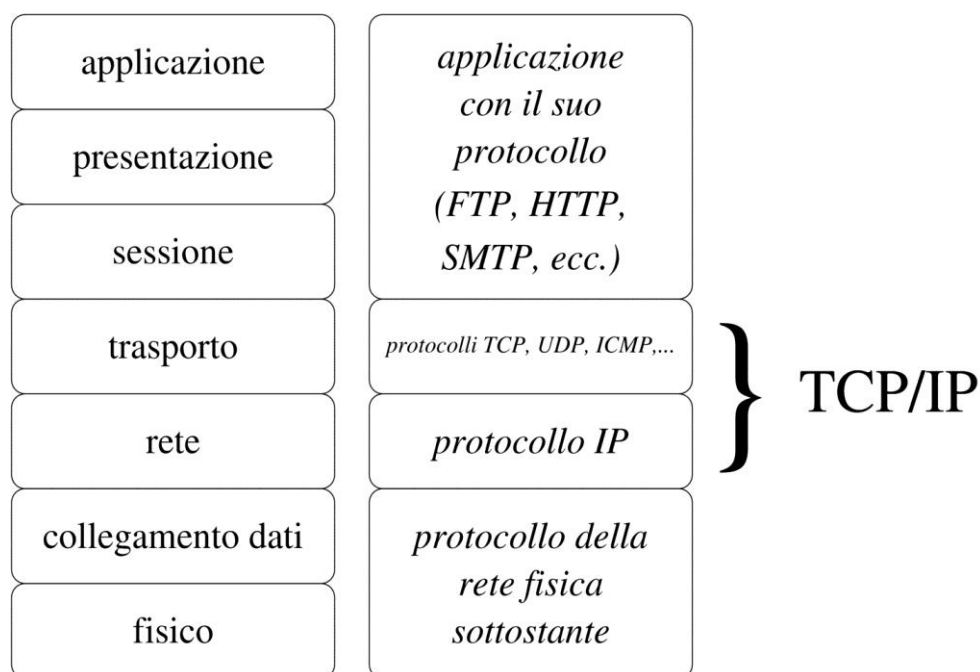


Figura 1: I modelli ISO/OSI e TCP/IP

IP e TCP sono i protocolli più rappresentativi presenti rispettivamente ai corrispondenti livelli 3 e 4 del modello ISO-OSI; il livello applicativo è sempre presente ma, in alcuni casi, può innestarsi direttamente sul protocollo TCP, saltando alcuni livelli intermedi.

Il modello, inoltre, non specifica i due livelli più bassi e questo sta a significare che il protocollo IP si può innestare su qualunque livello sottostante, quindi su qualsiasi LAN.

Questo porta l'architettura TCP/IP ad essere alla base dell'interconnessione di reti e si parla di internetwork, cioè quando reti LAN e WAN sono collegate tra loro.

## 2.5.2 HTTP

L' HyperText Transfer Protocol (HTTP) è un protocollo "stateless" (senza memoria) che permette sia la ricerca che il recupero dell'informazione in maniera veloce. La scelta di un protocollo di questo tipo è stata necessaria affinché fosse possibile passare velocemente da un server ad un altro attraverso i link ipertestuali, ma questa caratteristica non fornisce dati sulla sessione di navigazione, e talvolta costringe gli sviluppatori di contenuti web ad utilizzare metodi basati sui cookie.

La prima versione di questo protocollo risale agli anni ottanta e costituiva, assieme al linguaggio HTML e agli URI, il nucleo di base del Word Wide Web (WWW) creato al CERN di Ginevra.

La prima versione effettivamente disponibile del protocollo fu la HTTP 1.0, implementata nel 1991, ma ben presto si notarono diversi limiti, in particolare

- l'impossibilità di salvare più siti web sullo stesso server (risolto tramite virtual host);
- il mancato riutilizzo delle connessioni disponibili;
- l'insufficienza dei meccanismi di sicurezza.

Il protocollo venne quindi revisionato ed esteso alla versione HTTP 1.1 (e presentato come RFC (Request For Comments, un insieme di documenti cui la comunità di internet fa riferimento che descrivono, specificano e aiutano all'implementazione della maggior parte delle norme, standard e protocolli legati a internet).

È gestito da un software (server HTTP) che risiede negli hosts che intendono essere fornitori di informazioni. Per accedere alle informazioni fornite, è invece necessario un software lato client (il browser), in grado di interpretare le informazioni ricevute dal server.

Le specifiche del protocollo sono definite dal W3C e generalmente un server HTTP resta in ascolto sulla porta 80, usando il protocollo TCP a livello di trasporto.

L'HTTP funziona su un meccanismo richiesta/risposta: il client effettua una richiesta e il server restituisce una risposta; vi sono quindi messaggi di richiesta e messaggi di risposta.

È un protocollo di livello 7 del modello ISO-OSI (livello applicativo), e differisce dagli altri protocolli in quanto le connessioni vengono chiuse una volta che una certa richiesta è stata



soddisfatta, poiché avendo una struttura ipertestuale, diminuisce il numero di connessioni attive, aumentando l'efficienza sia sul client che sul server.

Un'evoluzione di questo protocollo è costituita dall'HTTPS (HyperText Transfer Protocol over Secure Socket Layer), nel quale è stato inserito un protocollo di crittografia asimmetrica per garantire trasferimenti di dati in modo riservato, in modo da impedire intercettazioni dei contenuti da terzi.

### **2.5.3 FTP**

L' FTP, File Transfer Protocol, è un protocollo di livello 7 del modello ISO-OSI (livello applicativo) che si occupa del trasferimento di dati tra host, basato sul protocollo TCP).

I suoi obiettivi principali, descritti nella sua RFC ufficiale (RFC959), sono:

- promuovere la condivisione di file;
- utilizzare computer remoti;
- risolvere in maniera trasparente l'incompatibilità tra differenti sistemi di memorizzazione di file tra gli host;
- trasferire dati in modo efficiente ed affidabile.

La creazione di questo protocollo risale al 1971, anno in cui era stato elaborato un sistema di trasferimento file tra i terminali del MIT (Massachusetts Institute of Technology). Numerosi RFC hanno apportato in seguito dei miglioramenti al protocollo di base, ma le più grandi innovazioni sono state effettuate nel luglio 1973.

Il protocollo FTP si basa su un modello client-server, in cui un terminale invia degli ordini (il client) e un altro aspetta delle richieste per effettuare delle azioni (il server).

Durante una connessione FTP, si aprono due canali di trasmissione: uno per i comandi (canale di controllo) e l'altro per i dati.

Tramite FTP è possibile copiare file da un computer remoto al proprio (operazione di download) oppure dal proprio computer ad uno remoto (operazione di upload); quest'ultima, per andare a buon fine, necessita di un'autenticazione mediante username e password per ragioni di sicurezza, poiché si suppone che l'utente disponga di un accesso personale a quel server.

## 2.6 I LINGUAGGI UTILIZZATI: HTML, CSS, PHP

Un sito web è costituito da tre parti:

- Struttura
- Presentazione
- Comportamento.

La struttura è formata dai contenuti testuali e dalla loro organizzazione, ed è implementata in linguaggio HTML (HyperText Markup Language).

La presentazione è gestita dai CSS (Cascading Style Sheets) e riguarda come vengono presentati visivamente gli elementi grafici e di impaginazione.

La parte scritta con codici di script come il PHP gestiscono la dinamicità del sito grazie alla loro capacità di interfacciarsi ad un database e ne definiscono il comportamento.

### 2.6.1 HTML

L' HTML appartiene alla famiglia dei linguaggi SGML (Standard Generalized Markup Language) e la sua sintassi è stabilita dal World Wide Web Consortium (W3C).

Acronimo di Hypertext Markup Language, l'HTML non è un linguaggio di programmazione ma un linguaggio di markup per la costruzione di ipertesti (per ipertesto si intende, approssimativamente, un insieme di unità di informazione connesse tra loro attraverso una serie di associazioni messe a disposizione dell'utente).

Ciò significa che l'HTML non è utilizzabile per costruire programmi applicativi in quanto è privo di tutti i costrutti propri della programmazione che consentono di prendere delle decisioni (costrutto IF – ELSE) o di compiere delle interazioni (costrutto FOR), ma permette di definire in modo preciso come disporre gli elementi all'interno di una pagina attraverso i marcatori, chiamati "tag".

Un file contenente tag HTML è formato esclusivamente da caratteri alfanumerici e privo di qualsiasi caratteristica grafica e per passare dal codice HTML alla visualizzazione a video della pagina è necessaria un'operazione di rendering, cioè di un lavoro svolto dal browser di trasformazione del codice HTML in pagina web.

È considerato un linguaggio non WYSIWYG (What You See Is What You Get) e questo perché non si può sapere con precisione cosa accadrà quando la pagina sarà visualizzata

in un altro browser, in quanto HTML è stato concepito per definire il contenuto logico e non l'aspetto finale del documento, ed è per questo che è indispensabile per ogni programmatore testare il proprio codice in browser differenti.

Un file di tipo HTML è così formato:

```
<!DOCTYPE html>

<html lang="it">
  <head>
    <title>La mia prima pagina Web</title>
  </head>

  <body>
    <!-- Questo è un commento e il codice non verrà
    interpretato dal browser -->
  </body>
</html>
```

Il doctype serve per dichiarare il tipo di documento; in questo caso, si tratta di un documento scritto in HTML5 (ultima versione disponibile).

Per indicare che il contenuto del documento è scritto in una lingua diversa dall'inglese, basta inserire l'attributo "lang" all'interno del tag <html>; in questo caso, lang="it" indica che il documento è scritto in italiano.

Nella sezione <head> contenente l'intestazione vengono inserite informazioni sul titolo che verrà visualizzato nel browser (utilizzando il tag <title>) e altro codice che deve essere eseguito prima del caricamento della pagina.

La sezione <body> contiene invece il corpo del documento, ed è qui che avviene la composizione fisica della pagina.

Nel 1989 Tim Berners-Lee e Robert Cailliau, ricercatori del CERN di Ginevra (laboratorio europeo per la Fisica dell'Alta Energia), crearono il World Wide Web per costruire un sistema che permettesse la condivisione di documentazioni scientifiche mediante un concetto di ipertesto indipendentemente dalla piattaforma utilizzata, e definirono standard e protocolli per scambiare tali documenti sul web, progettando il linguaggio HTML e il protocollo HTTP.

Inizialmente, gli standard e i protocolli supportavano esclusivamente la gestione di pagine statiche, ma nel corso degli anni l'HTML ha subito diverse revisioni conseguentemente allo sviluppo di internet e alla creazione di strumenti progettati per la gestione di pagine web dinamiche.

Nel Dicembre del 1997 il W3C pubblica una Candidate Recommendation in favore di HTML 4.01 e nel 1999 viene reso pubblico.

Nel 1998 il W3C (organismo internazionale fondato nel 1994, composto da università e aziende private, con lo scopo di guidare lo sviluppo del web e di definire gli standard) costituì il Web Standards Project (WaSP), con l'obiettivo di incoraggiare Netscape, Microsoft e le altre case produttrici di browser a supportare pienamente gli standard.



Figura 2: Logo WaSP

Tale progetto aveva lo scopo di:

- ridurre il costo e la complessità dello sviluppo di pagine web,
- di far produrre codice valido agli editor;
- di incrementare l'accessibilità.

Una delle iniziative più importanti in campo di accessibilità è la WAI, Web Accessibilità Iniziative, definita dal W3C con lo scopo di rendere il web accessibile universalmente.

Attualmente, l'ultima versione di HTML è la 5, ma non è ancora stata proclamata standard: ha ricevuto una Candidate Recommendation nel Dicembre 2012 e dovrebbe essere reso pubblico nel 2014.

Nonostante ciò, i web designer hanno già cominciato ad utilizzarla, con la speranza che la versione precedente (HTML 4) sia presto abbandonata.

Tra le novità introdotte nella versione 5 vi sono:

- Miglioramento della separazione tra la struttura (markup), caratteristiche di stile (CSS) e contenuti;
- Introduzione di elementi specifici per il controllo di contenuti multimediali: tag <video> e <audio>;
- Estensione a tutti i tag di una serie di attributi, in particolare quelli finalizzati all'accessibilità;

- Sostituzione del lungo e complesso “doctype” con un semplice <!DOCTYPE html>;
- Introduzione della geolocalizzazione in risposta alla forte diffusione di sistemi operativi mobili (i più diffusi: Android e iOS).

## 2.6.2 CSS

Il linguaggio CSS (Cascading Style Sheet, foglio di stile a cascata) è un documento di formato testuale che definisce il layout di visualizzazione di una pagina HTML, con caratteristiche di stile riguardante i font, i colori, gli allineamenti e le spaziature.

Avere tutte le caratteristiche di visualizzazione in un unico file permette di separare la struttura dalla presentazione e facilita la gestione della grafica: infatti, se si decide di cambiare il colore di sfondo in un sito formato da centinaia di pagine, modificando il valore dell’attributo una sola volta lo si cambia in tutto il sito, e ciò limita gli errori (se non esistesse un foglio di stile, si dovrebbe cambiare l’attributo che definisce il colore di background in ogni pagina, e ciò implica la possibilità di saltarne qualcuna, incombendo in errore).

La caratteristica principale di questo linguaggio è data dal fatto che l’informazione sulla presentazione di un documento viene gestita a livelli e si trasmette da un livello a quello inferiori, e da qui il nome “a cascata”; inoltre, vi è la possibilità di richiamare o invocare parti di codice già scritto ogni qualvolta risulta necessario.

Le regole di composizione di un file CSS sono state definite dal W3C a partire dal 1996 e sono contenute in un insieme di direttive (Recommendations).

Per collegare il file CSS ad una pagina HTML è sufficiente inserire l’informazione nell’intestazione della pagina, e lo si può fare in due modi diversi.

Utilizzando il tag <style>:

```
<html>
<head>
  <title>Esempio</title>
  <style type="text/css">
    @import url(style.css);
  </style>
</head>
```

Utilizzando il tag <link>:

```
<html>
<head>
  <title>Esempio</title>
  <link rel="stylesheet" type="text/css"
href="style.css"/>
</head>
```

Il codice CSS è strutturato sotto forma di una o più regole, espresse come coppia *proprietà : valore*, e gli elementi sono specificati tramite uno o più selettori.

Le regole vengono applicate agli elementi interessati direttamente dal browser, durante la fase di rendering.

La sintassi di scrittura della regola è la seguente:

```
selettore {
proprietà1 : valore1;
proprietà2 : valore2;
}
```

I selettori possono essere di diversa natura:

1. selettori di tipo (fanno riferimento ai tag HTML);
2. classi (la regola è applicata a tutti gli elementi della pagina che presentano la proprietà `class="nome_classe"`; un elemento può appartenere a più classi);
3. identificatori (o ID; un elemento può averne al massimo 1 e deve essere unico);
4. pseudo-classi (selezionano elementi in base alle loro proprietà);
5. pseudo-elementi (identificano una parte di un elemento);
6. selettore di discendenza, figlio o fratello (selezionano gli elementi che si trovano in una particolare condizione di discendenza nella struttura);
7. selettore di attributi (seleziona elementi in base ai loro attributi).

```

/* applicare una certa formattazione: */
/* ad un paragrafo */
p {
[...]
}
(1)

/* agli elementi che appartengono alla classe "nome_classe" */
.nome_classe {
[...]
}
(2)

/* all'elemento con ID="nome_ID" */
#nome_ID {
[...]
}
(3)

/* al nodo di tipo p che è il primo figlio di un elemento
padre */
p:first-child {
[...]
}
(4)

/* alla prima linea di un paragrafo */
p:first-line {
[...]
}
(5)

/* ai <p> che contengono <span> - DISCENDENZA */
p span {
[...]
}
(6)

/* ai <div> che contengono direttamente <p> (FIGLIO diretto)
*/
div > p {
[...]
}
(6)

```

```

/* al primo elemento successivo ad un altro con cui condivide
lo stesso padre. Es: il primo <p> fratello di <h1> */
h1 + p {
[...]
}

```

(6)

```

/* agli elementi con lo stesso attributo. Es: stesso titolo:
*/
a[title=Esempio]{
[...]
}

```

(7)

Un'ulteriore caratteristica dei file CSS è che vi è la possibilità di applicarli solo sui dispositivi specificati; questo aspetto è importante in quanto il layout di visualizzazione su un pc non può essere lo stesso che viene utilizzato su un cellulare.

La sintassi da utilizzare per collegare un certo file CSS in base al media utilizzato è la seguente:

```

<link rel="stylesheet" type="text/css"
href="style_tipo_media.css" media="tipo_media">

```

Dove tipo\_media può essere:

- screen (desktop)
- handheld (PDA e smartphone)
- print (stampanti)
- braille (per dispositivi braille)
- embossed (per stampanti braille)
- speech o aural (sintetizzatori vocali, screen reader)
- tty (telescriventi)
- tv (televisioni)
- all (tutti i dispositivi).



Spesso utilizzare l'attributo handheld per identificare tutti i cellulari non è sufficiente, e quindi si utilizza un ulteriore attributo: la dimensione.

Un altro modo per gestire la visualizzazione di una pagina è, infatti, in base alle dimensioni della finestra del browser; in questo caso si fa utilizzo di una dimensione minima e massima entro le quali si andrà a richiamare uno specifico foglio di stile.

Esempio per individuare I cellulari:

```
<link rel="stylesheet" type="text/css"
href="style_tipo_media.css"
media="handheld, screen and (max-width:480px) ;
only screen and (max-device-width:480px) ">
```

### 2.6.3 PHP

Con l'acronimo PHP (PHP Hypertext Preprocessor) si indica un linguaggio open source lato server completo di scripting (linguaggio di programmazione interpretato), sofisticato e flessibile, che può girare su qualsiasi web server, su qualsiasi sistema operativo e soprattutto di interagire con i principali tipi di database.

Si può utilizzare per creare una semplice homepage dinamica fino al grande portale o ad un sito di e-Commerce.

Il codice PHP viene elaborato sul server e come risultato produce codice HTML che viene inviato al browser dell'utente che aveva fatto una certa richiesta.

PHP nasce nel 1994 ad opera del danese Rasmus Lerdorf come una serie di macro che permetteva una facilitazione ai programmatori nel gestire le pagine personali; per questo, il significato originario dell'acronimo era Personal Home Page.

Oggi PHP permette di raccogliere dati, generare contenuti dinamici interagendo con database e tanti altri servizi, utilizzando i principali protocolli, tra cui:

- IMAP (Internet Message Access Protocol, protocollo di comunicazione per la ricezione di e-mail)

- SNMP (Simple Network Management Protocol, protocollo di rete che consente la configurazione, gestione e supervisione dei nodi appartenenti ad una rete)
- NNTP (Network News Transport Protocol, protocollo di livello applicativo usato principalmente nei software sviluppati per i server basato su stringhe di testo mandate su canali che rispettano il protocollo di trasporto TCP – Transmission Control Protocol)
- POP3 (Post Office Protocol 3, il protocollo più utilizzato per il download della posta elettronica).

Tra le applicazioni principali si ricorda l'autenticazione degli utenti, la gestione di template e di file XML (eXtensible Markup Language, un linguaggio di markup che permette di definire delle regole di composizione con le quali controllare il significato degli elementi contenuti in un documento), la creazione dinamica di immagini.

Inoltre, questo linguaggio è caratterizzato da una velocità di sviluppo notevole e dalla capacità di semplificare azioni quali interrogazioni di database o utilizzo di socket TCP, a confronto degli altri linguaggi.

#### 2.6.3.1 LA GESTIONE DEI DATABASE

Come detto in precedenza, il linguaggio PHP interagisce con i database; nel mio caso, utilizzando un server web di tipo AMP, per quanto riguarda la gestione dei database si utilizza MySQL.

MySQL è un Relational database management system (RDBMS), un software utilizzato per la gestione di basi di dati relazionali, che consente di gestire e organizzare grandi quantità di dati.

È possibile interagire con MySQL utilizzando il linguaggio SQL (Structured Query Language), un linguaggio standardizzato per la gestione dei dati che formano un database relazionale.

# Capitolo 3

## Creare un sito web con Wordpress

Wordpress è un noto CMS la cui funzione principale è quella di creare blog. Negli ultimi anni ha subito molti aggiornamenti che hanno aggiunto molte funzioni e strutture tali da avere maggior controllo sul proprio sito, in particolare nell'ultimo anno.

Grazie all'aiuto di migliaia di sviluppatori, sono state introdotte modifiche molto potenti, rendendolo un CMS adatto a creare qualsiasi tipologia di sito web; ciò nonostante, per quanto riguarda tipologie specifiche quali e-Commerce o siti di grandi dimensioni, restano di prima scelta altri CMS, come Magento (per l'e-Commerce) e Joomla (per i portali).

In questa sezione, spiegherò come si deve procedere per realizzare un sito avvalendosi di questo CMS, facendo riferimento a uno di quelli più semplici da me creati durante il periodo di stage: [www.msservice.info](http://www.msservice.info).

Il sito è stato commissionato da MS Service, azienda che si occupa della distribuzione di volantini, pubblicità in cassetta e altro, che necessitava di un nuovo sito web da sostituire a quello che già aveva online.

Websonica ha stilato un preventivo che prevedeva la realizzazione di una piattaforma marketing su base Wordpress per supportare tutte le attività di registrazione, partecipazione ed engaged dell'utente nel metodo più semplice possibile.

Dopo aver stilato un layout grafico apposito, le sezioni previste da implementare sono:

- Home con Slider;
- sezione offerte speciali con l'inserimento di tutte le offerte presenti;
- integrazione su tutte le pagine, form "richiedi preventivo";

- inserimento registrazione newsletter;
- contatti con Mappa;
- form contatti di invio mail;
- registrazione a Google Maps;
- database di salvataggio dei contatti integrato.

Operativamente, si effettuano 3 fasi:

1. installazione di Wordpress (in locale);
2. inserimento dei contenuti e personalizzazione del template, in base alla grafica elaborata;
3. pubblicazione del sito (dopo la conferma del cliente).

## 3.1 Prima fase: INSTALLAZIONE DI WORDPRESS

Per prima cosa, si procede scaricando l'ultima versione disponibile di Wordpress (al momento, la 3.6; è prevista anche la versione in italiano) dal sito <http://wordpress.org/download/> .

Si scompatta il file appena scaricato nella directory nella quale si vuole salvare il codice sorgente (per esempio in C:\wamp\www) e si rinomina la cartella con il nome del sito che si va a creare, in questo caso "msservice".

Per procedere all'installazione del sito in locale, quindi utilizzando il WampServer, è necessario configurare un opportuno Virtual Host e creare il database.

### 3.1.1 CONFIGURAZIONE DEL VIRTUAL HOST

Per configurare il Virtual Host, si crea un collegamento tra la macchina che si sta utilizzando e il dominio del sito che si va a creare nel file "hosts", presente in C:\Windows\System32\drivers\etc , inserendo la riga:

```
127.0.0.1    www.msservice.info
```

(al posto di 127.0.0.1 si poteva scrivere "localhost"; entrambi i metodi indicano il computer corrente). Questo file non ha estensione in quanto è un file di sistema, e per poterlo

modificare lo si apre utilizzando un qualsiasi editor di testo in modalità amministratore; se non si esegue questa procedura, il file non può essere modificato.

Se si vuole fare in modo che il cliente riesca a vedere il sito prima della sua pubblicazione, senza che si rechi fisicamente nell'ufficio in cui esso è elaborato, dovrà modificare questo file mettendo al posto di 127.0.0.1 l'indirizzo IP del computer sul quale il sito è memorizzato.

A questo punto, si aggiunge, all'interno del file "httpd-vhosts.conf", che si trova in C:\wamp\bin\apache\Apache2.2.17\conf\extra\, il codice seguente:

```
<VirtualHost *:80>

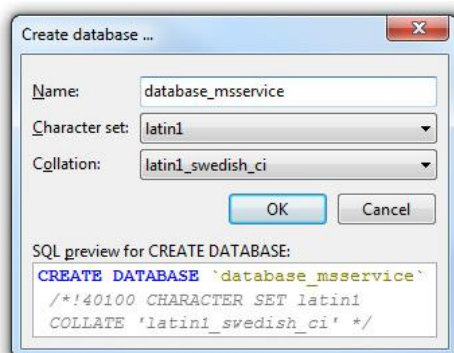
    ServerAdmin la_mia_mail@dominio.it //mail del webmaster
    ServerName www.msservice.info //nome del sito
    DocumentRoot "C:\wamp\www\msservice" //dove si trova il sito

    #File di log, dove verranno stampati gli errori e le azioni degli
    utenti
    ErrorLog "logs/msserviceerror.it.log"
    CustomLog "logs/msservice.it.log"

</VirtualHost>
```

### 3.1.2 CREAZIONE DEL DATABASE

Per la gestione dei database il software da me utilizzato è HeidiSQL: una volta avviato, si crea il database "database\_msservice" nel localhost, selezionando la voce Create Database (figura 3).



NOTA: per potersi connettere al localhost è necessario avviare il server Wamp e aspettare che la sua icona (nei collegamenti, in basso a destra) diventi verde.

Figura 3: Creazione del database.

Per collegare il database al sito web è ora necessario rinominare il file “wp-config-sample.php” in “wp-config.php”, aprirlo con un editor di testo (per esempio Notepad) e modificarlo in questo modo:

```
/** The name of the database for WordPress */
define('DB_NAME', 'database_msservice');

/** MySQL database username */
define('DB_USER', 'username_ms');

/** MySQL database password */
define('DB_PASSWORD', '123456789');

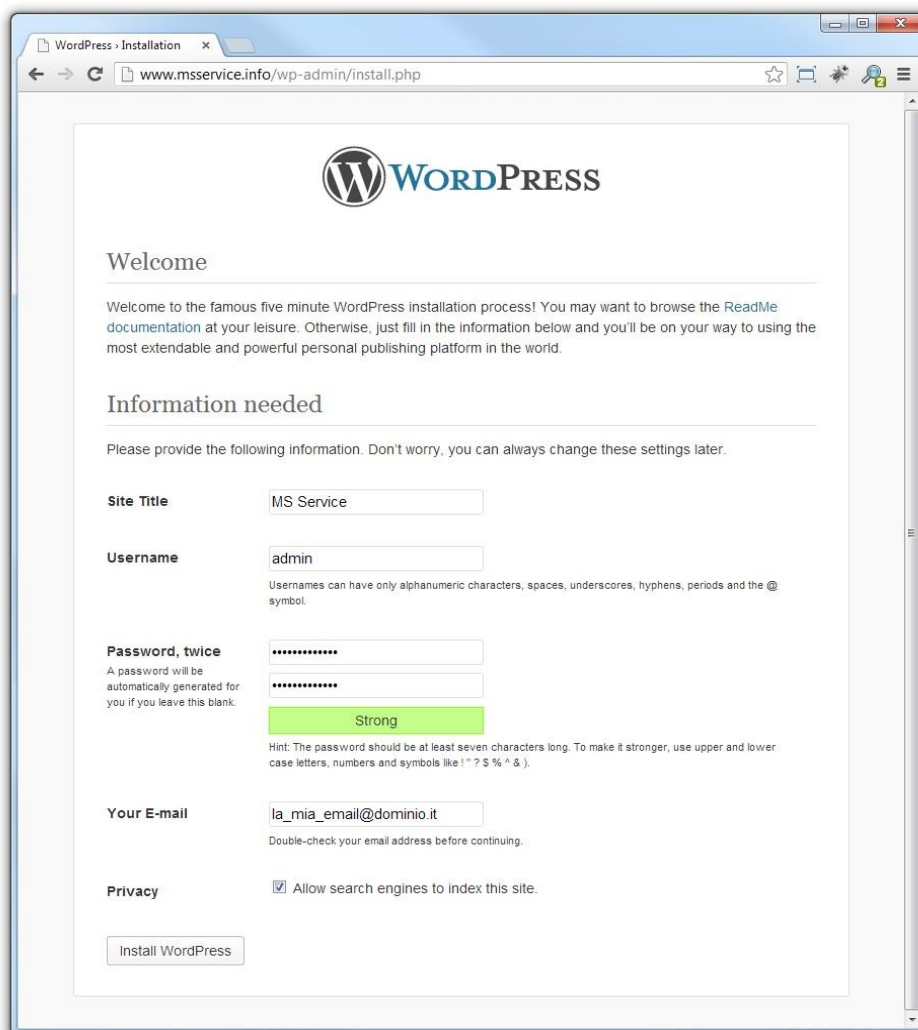
/** MySQL hostname */
define('DB_HOST', 'localhost');
```

Dove al posto di username\_ms e '123456789' vanno inseriti username e password dell'utente che ha gli accessi di tipo root (e cioè quelli del localhost; nel codice sopra riportato tali informazioni sono state inventate per motivi di sicurezza).

### 3.1.3 INSTALLAZIONE

A questo punto si dispone di tutti gli elementi necessari all'installazione: una volta riavviato il WampServer si inserisce il dominio del sito e cioè www.msservice.info all'interno di un qualsiasi browser, si compila il form di installazione e si clicca su “Install Wordpress” (figura 4).

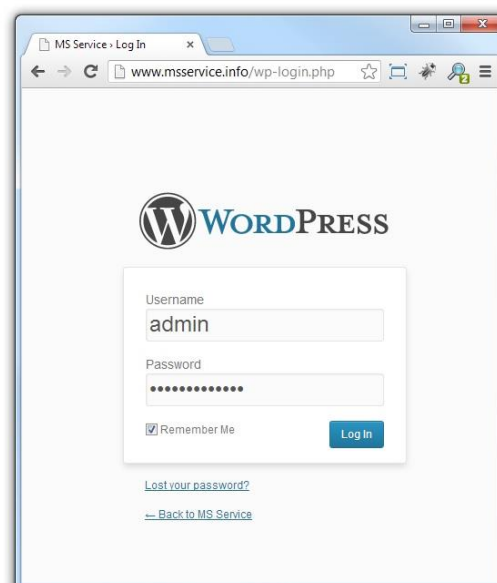
Se tutto va a buon fine, si ottiene una schermata come in figura (5) e si può effettuare il login (figura 6), con le credenziali inserite in precedenza (figura 4).



**Figura 4:**  
*Installazione di  
WordPress.*



**Figura 5:** *Schermata di installazione avvenuta.*



**Figura 6:** *Fase di login.*

In questo modo, si accede al back-end del sito, dal quale si possono inserire i contenuti, modificare le impostazioni e i template, creare pagine, ecc. (figura 7)

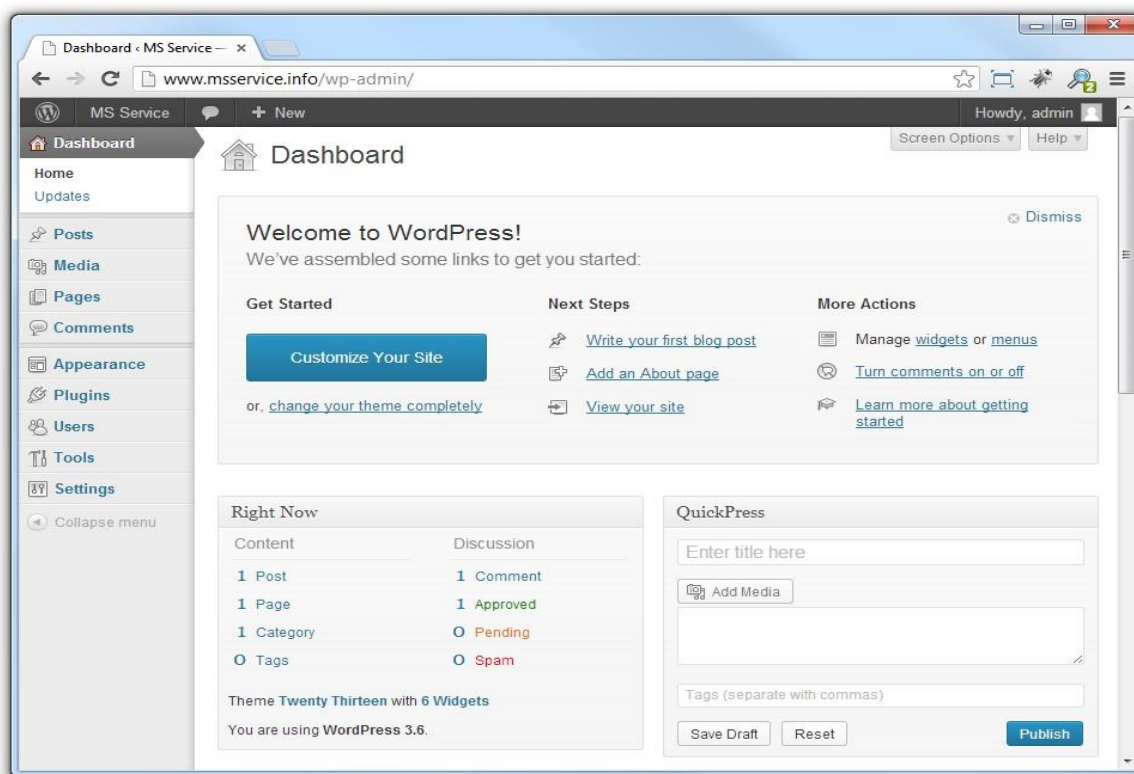


Figura 7: Il Back-end.

Ogni qualvolta si volesse accedere a questa sezione, basta inserire l'URL `www.msservice.info/wp-admin/` all'interno del browser, ed effettuare il login.

Prima di continuare, sarà opportuno esplorare com'è strutturato il back-end del sito.

### 3.1.4 STRUTTURA DEL BACKEND DI UN SITO WORDPRESS

Quando si accede al back-end di un sito, la pagina visualizzata di default è la Dashboard (figura 7).

Essa costituisce la bacheca generale del sito, nella quale vengono riportate le informazioni di riepilogo: il numero di pagine, di post, di commenti inseriti, di widget e di plugin utilizzati.

Sulla sinistra è presente il menu di navigazione, che consente di muoversi all'interno delle varie sezioni del back-end. Vi è la possibilità di aggiungerne di nuove installando plugin o



utilizzando temi che prevedono pannelli aggiuntivi; per la versione standard essi sono (fra parentesi il corrispettivo nome in italiano, qualora fosse diverso):

- Post (*articoli*), nel quale creare e gestire articoli, con la possibilità di raggrupparli in categorie. Essi rappresentano il cuore del nostro sito (o blog);
- Media, dove caricare e gestire i file immagine, video pdf e zip, che si potranno utilizzare all'interno delle pagine o degli articoli come normali Link;
- Pages (*pagine*), in cui creare le pagine del sito. Il form di creazione è simile a quello degli articoli, ma a differenza di questi ultimi le pagine sono tendenzialmente più statiche e rappresentano la parte più statica del sito, come ad esempio le pagine "Chi siamo" o "La storia dell'azienda";
- Comments (*commenti*), area dedicata alla gestione dei commenti, in cui si possono approvare, cancellare o segnalare come spam;
- Appearance (*aspetto*), che consente la personalizzazione della grafica di visualizzazione, offrendo la possibilità di installare temi, utilizzare widget o creare dei menu;
- Plugins, in cui vengono gestite le installazioni dei plugin, estensioni di codice utilizzate per aggiungere funzionalità che di default non erano presenti;
- Users (*utenti*), dove gestire degli utenti attribuendo loro i permessi adeguati (cioè associando una particolare categoria di utenza) in base alle loro caratteristiche;
- Settings (*impostazioni*), nella quale gestire la configurazione del sito sotto diversi punti di vista, dal titolo del blog alla formattazione di data e ora.

## 3.2 Seconda fase: INSERIMENTO DEI CONTENUTI E PERSONALIZZAZIONE DELLA GRAFICA DEL SITO

### 3.2.1 PERSONALIZZAZIONE DELLA GRAFICA

Per prima cosa si installa il template destinato a designare la grafica del sito; per MS Service è stato scelto il tema Nimble, scaricabile dal sito <http://themeforest.net/>.

Si va quindi nella sezione Appearance > Theme > Install Themes e si fa un upload del tema, selezionando il file zip che lo contiene e che deve essere precedentemente scaricato; si torna poi nella scheda Manage Themes e si attiva il tema appena installato.

È possibile cambiare tema in qualsiasi momento si desideri, e per farlo è sufficiente tornare in questa sezione del back-end e attivarne uno diverso; con un semplice passaggio si è in grado di rivoluzionare completamente la grafica un sito, e questa è una delle caratteristiche di Wordpress (e dei CMS in generale) più importanti, che facilitano e velocizzano un lavoro che altrimenti avrebbe richiesto svariate ore di lavoro.

Salterà subito all'occhio che, una volta installato il tema, la barra di Navigazione subirà dei cambiamenti (figura 8): Nimble, infatti, prevede di default delle componenti aggiuntive:

- OneByOneSlider
- Slideshow
- Benefits
- Catalog
- Partners, Clients, etc.
- Photo Album
- Portfolio
- Team
- Testimonials
- Video

oltre, ovviamente, al "Theme Options" che contiene tutte le caratteristiche (personalizzabili) del tema.

Sarà a cura del creatore del sito scegliere quali di queste componenti utilizzare, in base alla struttura che il sito finale dovrà avere.



**Figura 8:** Barra di navigazione

Un'ulteriore operazione di modifica alla visualizzazione del sito dovrà essere fatta sia parallelamente che dopo l'inserimento dei contenuti, per la visualizzazione di informazioni aggiuntive laddove il tema non lo preveda, la sistemazione della formattazione di particolari frammenti di testo, la dimensioni di determinate immagini o il colore di certi elementi.

Queste modifiche dovranno essere effettuate facendo ricorso ai CSS. All'interno della cartella relativa al tema utilizzato è presente un file chiamato "style.css" con il quale il tema specifica i colori, la formattazione e il layout di default. Modificando questo file si otterrebbe il risultato desiderato ma sarebbe poco efficiente; infatti, nel momento in cui vi fosse una nuova versione del tema, non si potrebbero fare aggiornamenti o tutte le modifiche andrebbero perse.

Per risolvere questo problema, si deve creare un nuovo tema che erediti le caratteristiche di quello installato (Nimble) al quale applicare le modifiche, in modo tale che in caso di aggiornamenti non vi sia nessuna perdita di informazioni. Tra le sue molteplici caratteristiche, Wordpress supporta anche questa funzionalità, ed il tema che si crea a partire da uno già esistente prende il nome di Tema Figlio.

### 3.2.1.1 COME REALIZZARE UN TEMA FIGLIO

Per tema figlio si intende quel tema che viene creato a partire da uno già esistente, qualora vi fossero delle modifiche da apportare riguardo la visualizzazione delle pagine del sito, che andrebbero effettuate nei file che costituiscono i template o nei file CSS.

La necessità di effettuare modifiche di questo tipo in un tema figlio e non nel tema originale è legata all'aggiornamento dei plugin o del tema; se, infatti, si modifica il file del tema padre, dopo un aggiornamento, queste modifiche sarebbero sovrascritte e andrebbero rifatte.

Creare un tema figlio è molto semplice. Per prima cosa bisogna andare nel codice sorgente del sito e creare una cartella in: `msservice\wp-content\themes\` (solitamente la si chiama "NomeTemaPadre-child", quindi in questo caso "dt-nimble-child").

All'interno di questa cartella, si crea un file chiamato "style.css", necessario per la creazione del tema figlio, nel quale vengono inserite diverse informazioni che lo riguardano, come il nome, una descrizione, l'autore e il suo sito web, che verranno utilizzate qualora fosse necessario (ad esempio, alcuni temi inseriscono nel footer la scritta "powered by NomeAutore", in cui viene automaticamente inserito il link del sito web).

All'interno di questo file, il codice da inserire è il seguente:

```
/*
Theme Name: Nimble Child Theme
Theme URI: http://dream-theme.com/
Description: Child Theme of Nimble
Version: 1.0.5
Author: Webmaster
Author URI: http://www.sitoDelWebmaster.it/
Tags: dream-theme, nimble, responsive, adaptive, mobile,
tablet, corporate, portfolio, creative, gallery, photo-albums,
custom-backgrounds, custom-layouts, slider, slideshow,
background-builder, layout-builder, page-templates,
shortcodes, customizable, left-sidebar, right-sidebar, multi-
columns, fixed-width, custom-colors, custom-header, custom-
menu, editor-style, featured-images, full-width-template,
sticky-post, theme-options, translation-ready, password-
protected, more
Template: dt-nimble
*/
@import url(../dt-nimble/style.css);
```

“Template: dt-nimble” indica la cartella in cui è contenuto il tema padre, dal quale il nostro tema figlio eredita i template; “@import url(../dt-nimble/style.css);” serve, invece, per importare il foglio di stile del tema padre, in modo tale che nel tema figlio si inseriscano solamente le modifiche.

Nel momento in cui si volesse modificare un particolare template, ad esempio quello che definisce il layout delle pagine, sarà sufficiente copiare il file “page.php” dal tema padre alla cartella “dt-nimble-child” e modificarlo.

Per attivare il tema figlio è sufficiente andare nel back-end del sito ed attivarlo (se tutto è andato a buon fine, il tema figlio si troverà in automatico tra gli altri temi installati, in Appearance > Themes, figura 9).

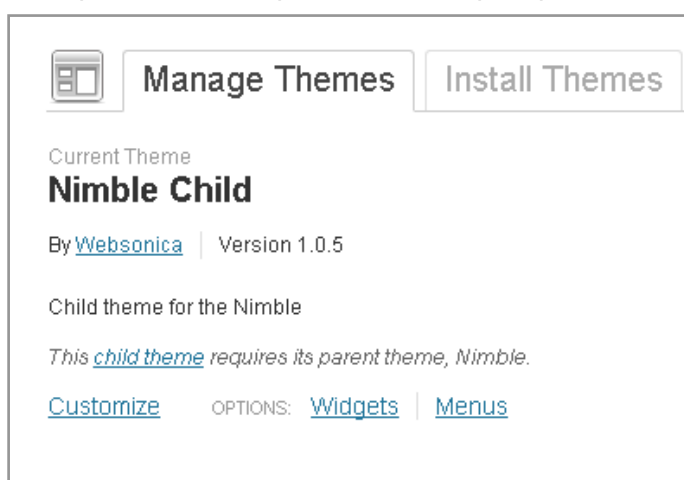


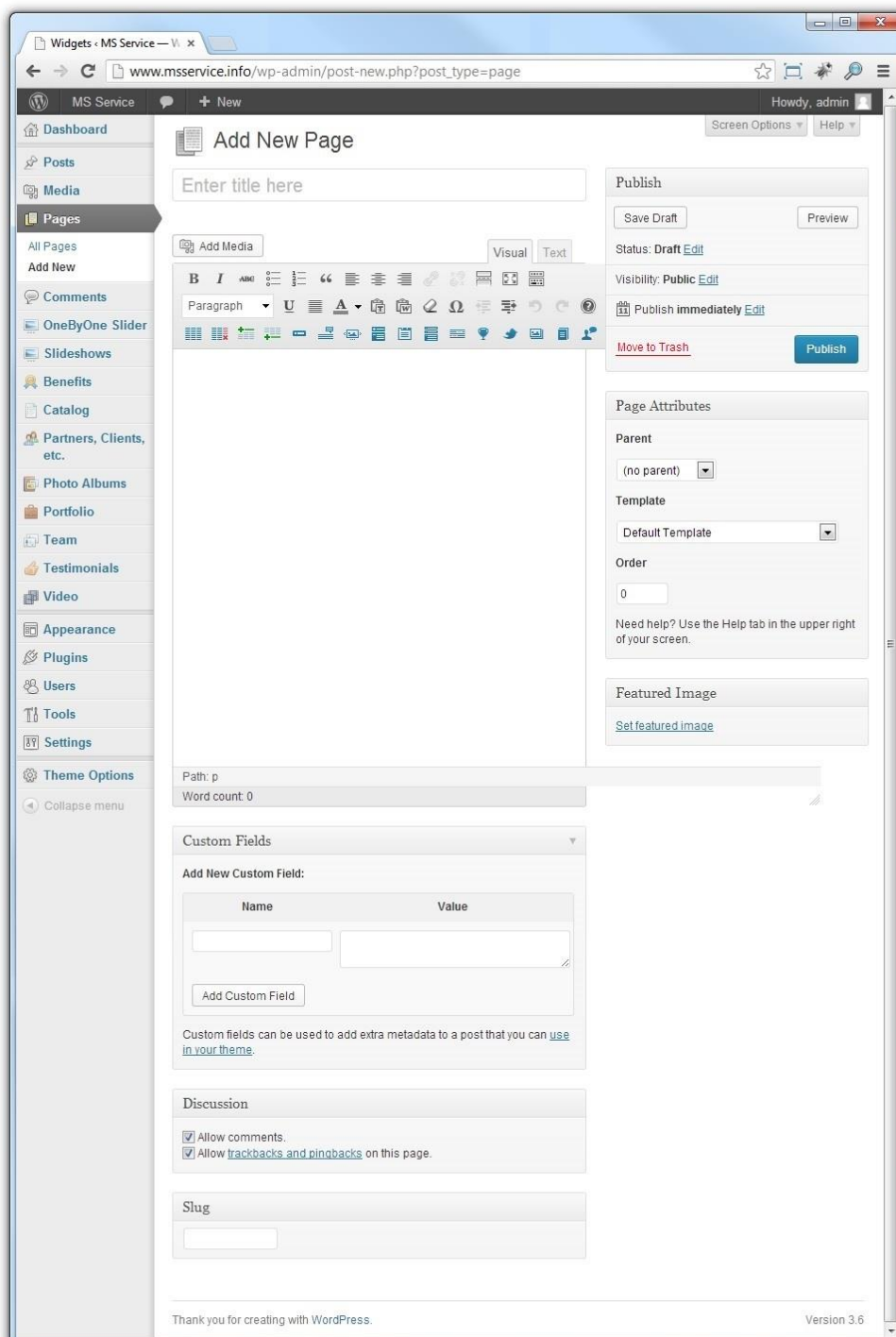
Figura 9: Selezione del tema figlio.

## 3.2.2 INSERIMENTO DEI CONTENUTI

Accedendo al back-end, si ha la possibilità di creare pagine, articoli, photogallery e tutti gli altri elementi che si prevedono per un sito.

### 3.2.2.1 CREAZIONE DI UNA PAGINA

Per creare una pagina si accede a Pages > Add New (figura 10).



**Figura 10:**  
*Creazione di una pagina.*

Per poter visualizzare le sezioni aggiuntive in fondo alla pagina, si devono spuntare tutte le opzioni che si trovano nella scheda “Screen Options”, in alto a destra.

Questa sezione è divisa in diverse parti:

1. Titolo;
2. Form di inserimento dei contenuti della pagina;
3. Publish (*Pubblicazione*);
4. Page Attributes (*Attributi della pagina*);
5. Foto in evidenza;
6. Opzioni aggiuntive, in base al tema/plugin installati.

1. La sezione del titolo, come si può facilmente intuire, andrà a contenere il titolo della pagina. Esso sarà visualizzato o meno a seconda del tema utilizzato.

2. Per quanto riguarda il form di inserimento dei contenuti della pagina, esso si suddivide in due parti: Visual (*Visuale*) e Text (*Testo*), nei quali si possono inserire (oltre al testo) altri elementi, come le immagini, cliccando sul bottone “Add Media” situato a sinistra, appena sopra il form. In questo modo, inserire foto o altri media è molto semplice: si apre infatti una finestra apposita nella quale selezionare l’immagine da inserire direttamente dalla Media Library (costituita da tutti i file multimediali caricati fino a quel momento, che si può gestire dalla sezione “Media” del menu di navigazione) oppure mediante upload direttamente dal computer (se si segue questa procedura, il file caricato sarà automaticamente salvato nella Media Library).

All’interno della scheda Text si inserisce il codice HTML, mentre nella Visual è possibile vedere un’anteprima di quello che sarà poi visualizzato dal browser, in tempo reale: essa è di fatto un editor di testo WYSIWYG.

Questa duplice maniera con cui inserire i contenuti è importante in quanto non si devono avere necessariamente particolari competenze per formattare il testo, in quanto nella sezione Visual appare un menu (in alto) costituito da funzionalità che consentono di creare elenchi puntati, allineare o cambiare il colore del testo, e altre formattazioni tipiche degli editor di testo. Oltre alla formattazione del testo, in questa sezione è possibile gestire i media, visualizzando un’anteprima degli effetti che si applicano, come ad esempio il ridimensionamento o l’allineamento.

**3.** Nell'area di Pubblicazione si gestisce di fatto lo stato e la visibilità della pagina: se è in fase di progettazione (*Draft*) o se è in attesa di revisione (*Pending Review*), e se la sua visibilità è pubblica (*Public*), protetta da password (*Password protected*) o privata (*Private*).

Questa ultima caratteristica consente di nascondere provvisoriamente delle pagine o di limitarne l'accesso ai soli utenti autorizzati facilitando notevolmente la gestione del sito, ed è una delle tante funzionalità che rende preferibile l'utilizzo di un CMS rispetto alla programmazione di un sito da zero.

Sempre in questa parte, vi è inoltre la possibilità di modificare la data di pubblicazione della pagina e il pulsante con il quale pubblicarla (o aggiornarla).

**4.** Per gestire la gerarchia delle pagine o impostare un particolare template di visualizzazione, si utilizza invece la sezione degli attributi della pagina.

Qui infatti è possibile indicare se la pagina ha un padre (campo *Parent*) selezionandolo tra quelle già create del sito, se deve essere visualizzata in un modo particolare, quindi utilizzando un particolare template (campo *Template*), per esempio se è una pagina di blog o se sono presenti sidebar, e modificare l'ordine di visualizzazione delle pagine (campo *Order*).

**5.** La *Featured Image* non è altro che l'immagine in evidenza; il suo caricamento è facilissimo, basta cliccare su "Set featured image" e selezionare l'immagine direttamente dal computer.

È possibile cambiare questa immagine in qualsiasi momento e tutte le volte che si ritiene necessario.

**6.** Tra le opzioni aggiuntive, riportate sotto il form di inserimento dei contenuti, la voce *Discussion* è una delle più modificate. Essa infatti setta per default entrambi i campi di cui è formata, che consentono agli utenti di commentare (*Allow comments*) e di inserire l'opzione di *trackback* e *pingback* (*Allow trackbacks and pingbacks on this page*) nella pagina, ma nella maggior parte dei casi le pagine non devono essere commentabili, perciò vanno deselezionati questi due campi.

Il *trackback* è un meccanismo che permette all'autore di un post di ricevere una notifica quando qualcuno cita un suo articolo; il *pingback* è un link che un utente attiva verso un altro blog.

Dopo aver pubblicato la pagina, nell'area di pubblicazione comparirà il pulsante "Preview Changes" con il quale sarà possibile vedere un'anteprima della pagina.

È utile visualizzarla prima di salvarla quando si fanno delle modifiche perché in caso di errore sarà sufficiente uscire dalla pagina (ovviamente senza salvare) o aggiornarla e si ritornerà alla versione precedente.

### 3.2.2.2 CREAZIONE DELL'HOMEPAGE: LO SLIDESHOW

Per il sito in questione è stato concordato con il cliente l'inserimento uno slideshow nell'Homepage.

Il tema Nimble comprende per default la sezione OneByOne Slider, uno strumento con il quale creare le singole slide che verranno poi visualizzate in uno slideshow in ordine di creazione (a partire dalla più recente) nelle pagine desiderate.

Procediamo quindi alla creazione di una slide, accedendo a OneByOne Slider > Add New OneByOne Slide (figura 11).

Il titolo di questo elemento non verrà visualizzato nel sito, per cui è bene nominare le singole slide in modo da identificarle in base al loro contenuto per poterle trovare con facilità in un secondo momento.

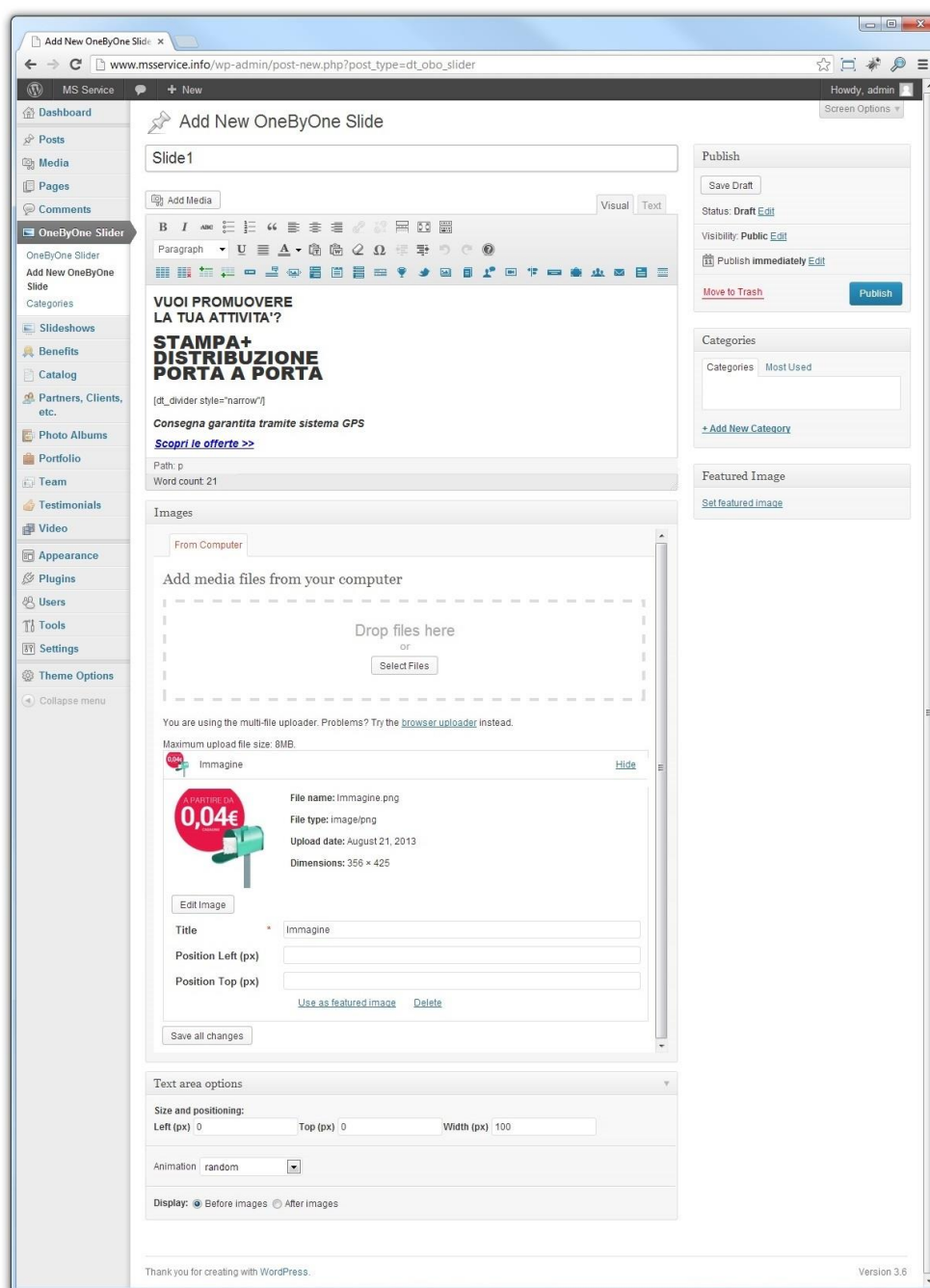
Due sono gli elementi che si possono inserire: testo e immagini. Il testo va inserito nel solito form dedicato all'inserimento dei contenuti, per le immagini vi è invece una sezione apposita, "Image".

In questa parte è possibile caricare una o più immagini direttamente dal computer e, ad upload terminato, consente di aggiungere alcuni attributi: il titolo, utile per l'accessibilità, e la distanza (in pixel) da sinistra e dall'alto, definendo così la posizione desiderata.

Anche la posizione del testo è personalizzabile, in "Text area options", nel quale è possibile inoltre selezionare la transazione desiderata per questa specifica slide.

Si noti che è presente la sezione "Categories": questo perché vi è la possibilità di aggiungere delle categorie e attribuirle alle slide, in modo tale da creare dei gruppi in cui dividerle.





**Figura 11:**  
 Creazione di una slide utilizzando "OneByOne Slider".

Per inserire lo slideshow all'interno di una pagina è sufficiente modificare il template della pagina stessa, selezionando la voce "OneByOne Slider (full-width)", "OneByOne Slider with sidebar" o "OneByOne Slider with blog" in base alla pagina in questione.

Nel nostro caso, trattandosi dell'homepage, il template da selezionare è il primo (full-width).

Modificando il template, tra le opzioni aggiuntive della pagina comparirà quella dedicata alla visualizzazione dello slideshow (Display slideshow), nel quale è possibile indicare se le slide da visualizzare sono tutte o solo alcune; in quest'ultimo caso, le slide si selezionano in base alla categoria di appartenenza.

Il risultato lo si può vedere in figura 12.

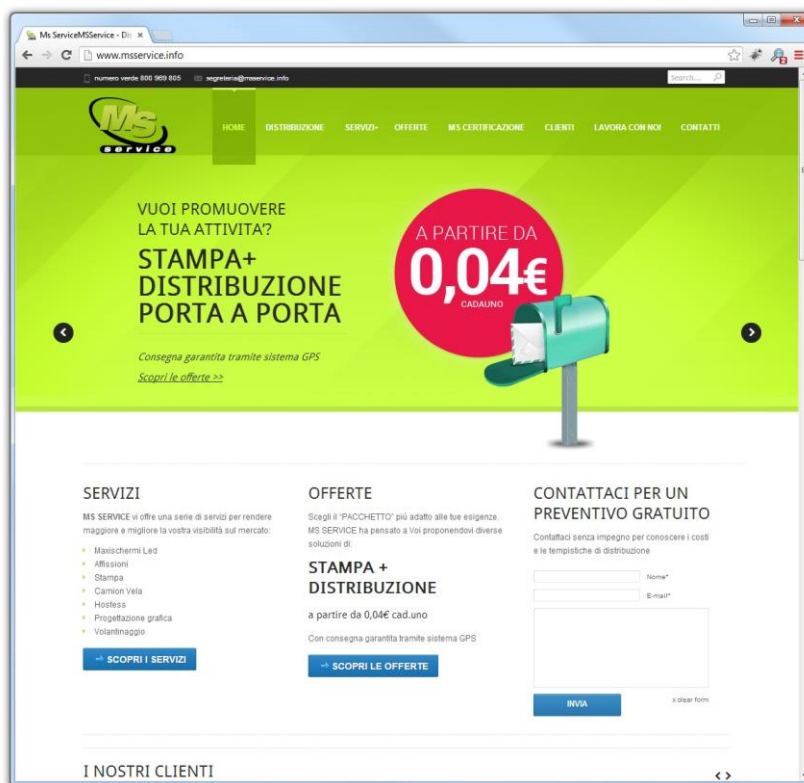


Figura 12: Homepage.

### 3.2.2.3 CREAZIONE DI UN ARTICOLO

La creazione di un articolo o post (figura 15) è molto simile a quella di una pagina (figura 10): titolo, form di inserimento dei contenuti, la sezione di pubblicazione, di discussione e la featured image funzionano esattamente allo stesso modo.

Le aree che invece sono specifiche dei post sono:

1. Excerpt;
2. Author;
3. Format;
4. Categories;
5. Tags;
6. Post options;
7. Custom Background.

1. L'Excerpt è sostanzialmente un riassunto del contenuto del post, che viene visualizzato nell'anteprima. Nell'area dedicata all'inserimento di questo testo è possibile inserire anche tag HTML, con i quali definirne la formattazione.

2. Author indica, come si può intuire dal nome della sezione, l'autore del post.

3. I post possono essere di diverso formato: Immagine o Standard.

Se si desidera postare un'immagine, allora il formato da adottare è "Image", nel quale è possibile mettere una descrizione. Se invece il post è prevalentemente costituito da testo, allora il formato corretto è quello standard.

Il modo in cui essi verranno visualizzati dipende esclusivamente dal template installato; nelle figure 13 e 14 sono riportate le anteprime dei post rispettivamente di tipo standard (nel quale si mette in risalto la possibilità di formattare l'excerpt) e di tipo immagine, utilizzando il tema Nimble.



Figura 13: Anteprima di un post di tipo Standard.



Figura 14: Anteprima di un post di tipo Immagine.

4. Ad ogni post è possibile associare una o più categorie, inseribili dall'utente in Post > Categories, e questa è la sezione dedicata per farlo.

Le categorie sono un ottimo strumento per organizzare i post in base ai loro contenuti tematici, e indicano in genere dei macroargomenti dal significato più ampio, come ad esempio "Tempo Libero", "Musica" o "Sport".

5. I tag sono delle etichette che, inserite in un post, mettono in risalto i principali argomenti trattati o, più semplicemente, evidenziano le parole chiave del testo.

Si differenziano dalle categorie in quanto quest'ultime sono gerarchizzabili (posso cioè creare delle sottocategorie), mentre i tag no.

Oltre ad essere altamente descrittivi, i tag sono molto comodi perché possono essere utilizzati come una seconda barra di navigazione, utilizzando appositi widget che visualizzano i tag più utilizzati, variando la dimensione dei caratteri in base al numero di post ai quali sono associati, una specie di menu di popolarità delle parole chiave. Il loro utilizzo è a discrezione dell'utente, che può decidere liberamente quali parole utilizzare, tenendo però conto che l'ideale sarebbe di inserire 3-4 tag per post, ognuno dei quali formato da non più di due parole. La gestione dei tag è possibile in Post > Tags.

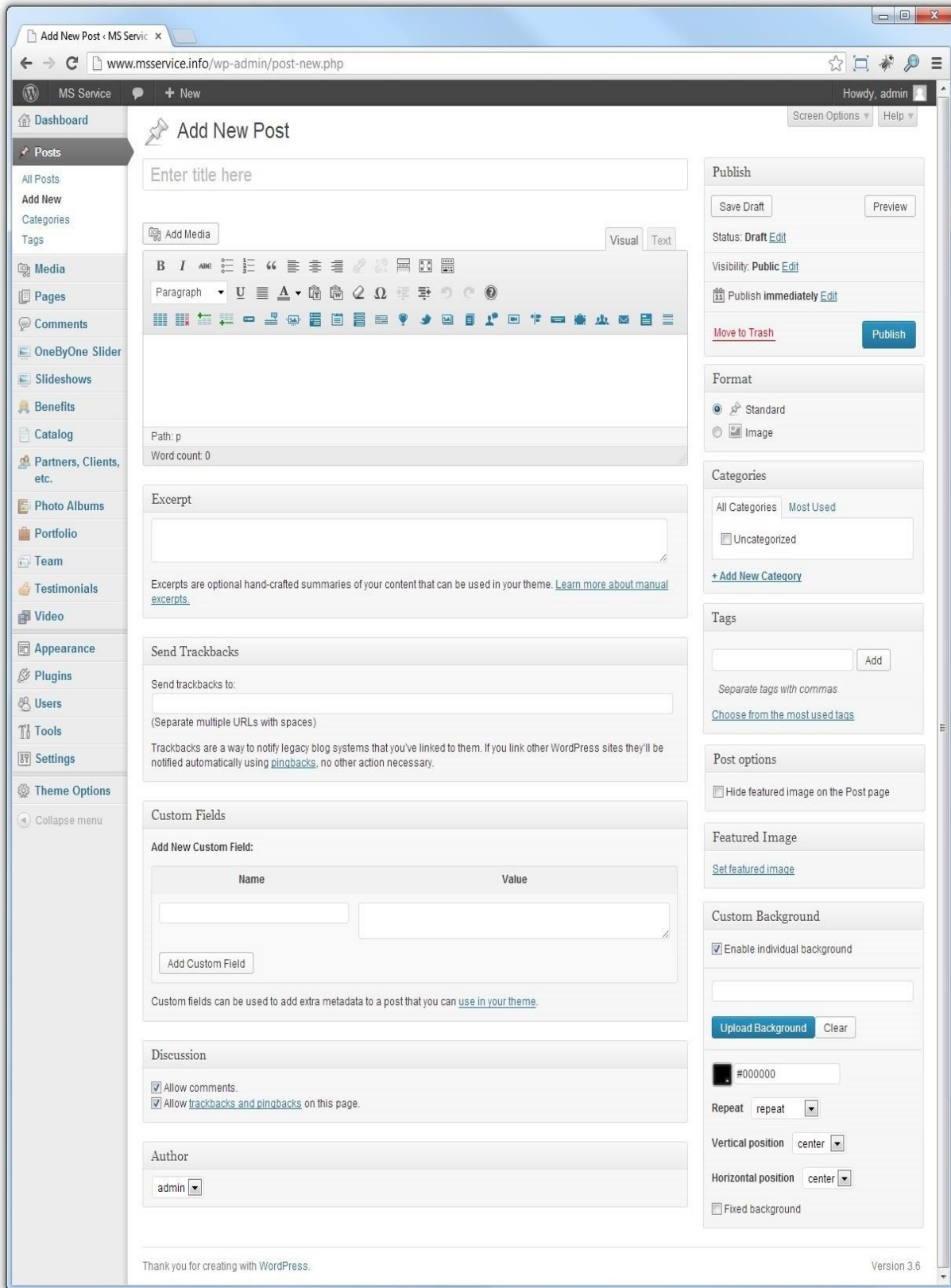


Figura 15: Creazione di un post.

6. In questa sezione è possibile indicare se, all'interno del post vero e proprio, la Featured Image sarà visualizzata o meno.

7. La sezione "Custom Background" consente di personalizzare lo sfondo di ogni post, e ciò è possibile selezionando l'attributo "Enable individual background". Lo sfondo può essere costituito da un'immagine (ripetuta o meno) o da un certo colore, come si può vedere dalla figura 15.

#### 3.2.2.4 UTILIZZO DI "CATALOG"

Il sito di MS Service prevedeva una pagina in cui fossero visualizzati i servizi che l'azienda offre.

Lo strumento più indicato per implementare quella parte è "Catalog", proprio perché, come suggerisce il nome, è concepito per la creazione di cataloghi con i prodotti e i servizi che si offrono.

Siamo avvantaggiati dal fatto che questa opzione è già presente di default nel tema che abbiamo scelto, quindi possiamo utilizzarlo senza dover ricorrere all'utilizzo di particolari plugin.

Sostanzialmente, il form di creazione di un articolo del catalogo è molto simile ad un post: esso è formato da un editor di testo per inserire i contenuti, un'area di pubblicazione, di gestione delle categorie e di discussione, prevede una feature image, l'excerpt e la possibilità di personalizzazione dello sfondo di ogni pagina contenente un singolo articolo (figura 16).

Come attributi aggiuntivi esso presenta:

1. Advanced options;
2. Catalog Item Options;
3. Catalog Item Pictures (for slideshow);
4. Related Catalog Items.

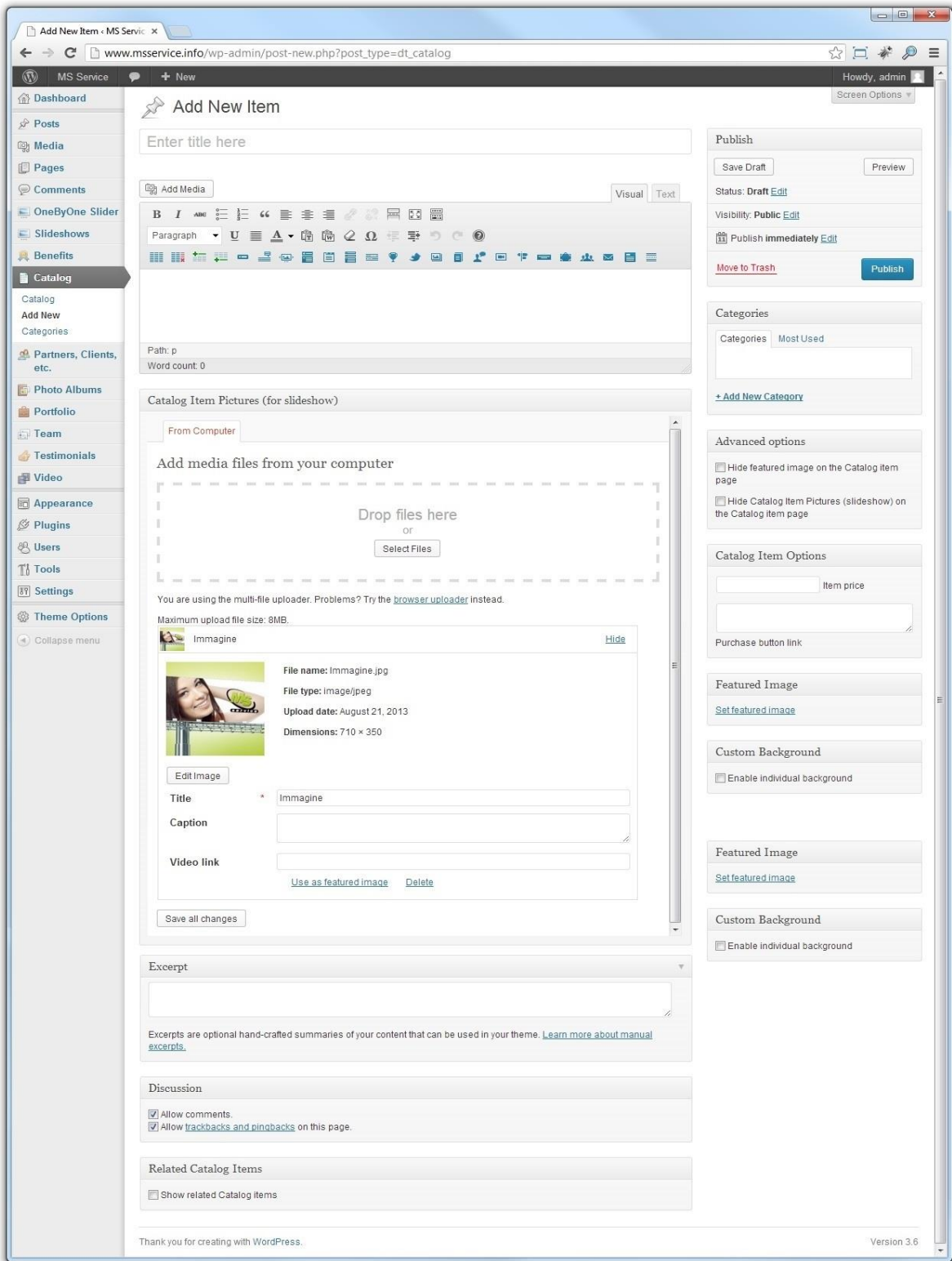


Figura 16: Creazione di un elemento del catalogo.

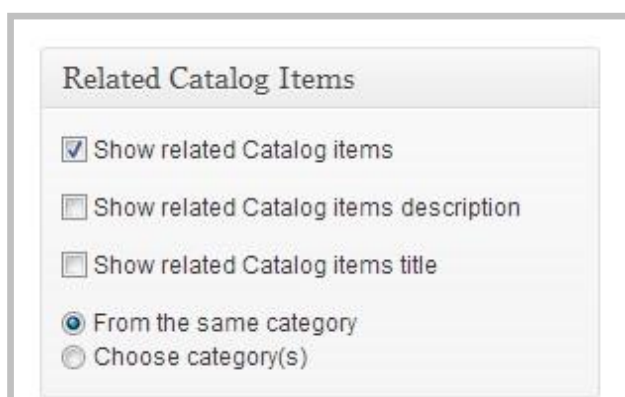
1. Tra le opzioni avanzate vi sono due attributi da poter selezionare, che modificano la metodologia di visualizzazione della pagina relativa al prodotto inserito nel catalogo: con il primo si nasconde la Featured Image (“Hide featured image on the Catalog item page”); il secondo permette invece di nascondere le immagini (punto 3) relative a quell’articolo (“Hide Catalog Item Pictures (slideshow) on the Catalog item page”).

2. Tra le opzioni relative all’articolo del catalogo (Catalog Item Options) vi è la possibilità di indicarne il prezzo (Item Price) e il link al quale si sarà reindirizzati qualora si cliccasse sul bottone relativo all’acquisto di quello specifico articolo.

3. Nella pagina dedicata ad un articolo, è previsto uno slideshow, che si può creare all’interno della sezione “Catalog Item Pictures (for slideshow)”, nel quale si esegue l’upload delle varie immagini che lo compongono.

4. In un catalogo, molti possono essere i prodotti / servizi correlati. La sezione “Related Catalog Items” risolve questo problema, con la possibilità di inserire per ogni articolo del catalogo con quali altri è correlato. Questa funzione è particolarmente importante perché aiuta l’utente a trovare una certa informazione, specialmente se non sta cercando una cosa specifica ma vuole esaminare le varie opzioni all’interno di un certa categoria di articoli.

Questi collegamenti possono essere effettuati in due diversi modi (come si può vedere in figura 17): o si visualizzano gli articoli appartenenti alla stessa categoria di quello corrente che si sta creando/modificando, oppure tutte le categorie in generale.



Related Catalog Items

- Show related Catalog items
- Show related Catalog items description
- Show related Catalog items title
- From the same category
- Choose category(s)

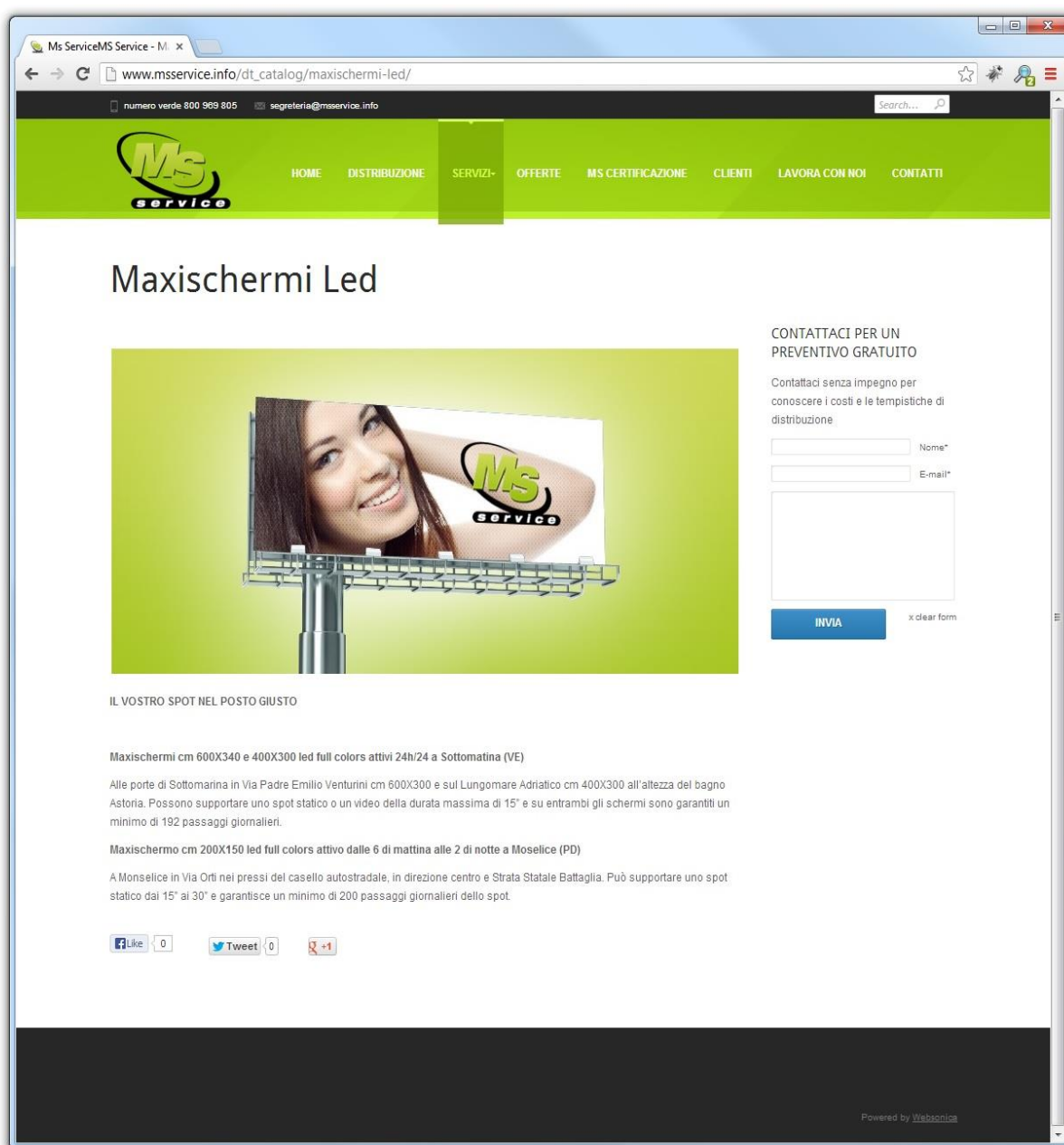
**Figura 17:** Creazione degli elementi correlati.



Per questo sito, l'implementazione dei servizi contenuti nel catalogo non prevedeva molti campi, ma semplicemente un'immagine e del testo.

Per quanto riguarda il template utilizzato, ne è stato scelto uno in cui si prevedeva una sidebar a destra, nella quale inserire il form di contatto(figura 18).

L'anteprima dei vari servizi è illustrata in figura 19.



The screenshot shows a web browser window displaying the website 'www.msservice.info/dt\_catalog/maxischermi-led/'. The page has a green header with the 'MS service' logo and a navigation menu with items: HOME, DISTRIBUZIONE, SERVIZI- (highlighted), OFFERTE, MS CERTIFICAZIONE, CLIENTI, LAVORA CON NOI, and CONTATTI. The main content area is titled 'Maxischermi Led' and features a large image of a woman smiling on a billboard. To the right of the image is a contact form titled 'CONTATTACI PER UN PREVENTIVO GRATUITO' with fields for 'Nome\*' and 'E-mail\*', a large text area, and an 'INVIA' button. Below the image, there is text describing the service: 'IL VOSTRO SPOT NEL POSTO GIUSTO' and details about two locations: Sottomarina (VE) and Moselice (PD). At the bottom, there are social media sharing buttons for Facebook Like, Twitter Tweet, and a plus sign for more options. The footer of the page indicates 'Powered by Websonica'.

Figura 18: Pagina del servizio "Maxischermi Led"



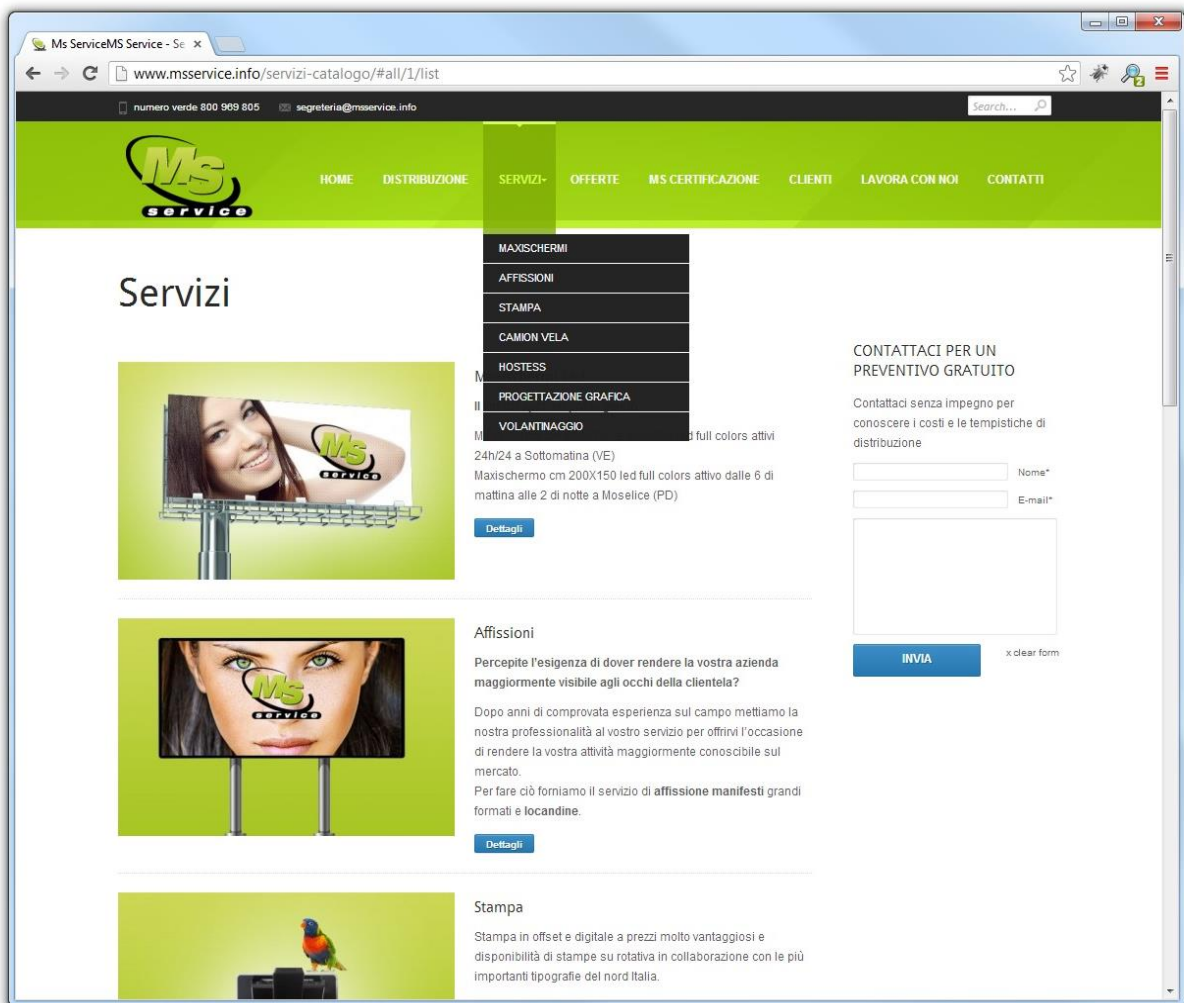


Figura 19: Pagina dei servizi, in cui si vedono le anteprime degli elementi del catalogo.

### 3.2.2.5 GESTIONE DELLE SIDEBAR E DEI WIDGET

Per quanto riguarda la visualizzazione delle pagine, di primaria importanza sono i widget e le sidebar, nelle quali essi vanno inseriti.

La gestione di queste componenti si effettua nella sezione Appearance > Widgets (figura 20).

Un **widget** è uno strumento con il quale siamo in grado di personalizzare il nostro sito. Ne esistono di moltissimi tipi ed ogni tema ne prevede alcuni di default.

Possono essere aggiunti mediante installazione di un plugin oppure creati mediante software che andrà inserito opportunamente nel codice sorgente (si ricorda che è bene modificare i file del codice sorgente esclusivamente all'interno del tema figlio).

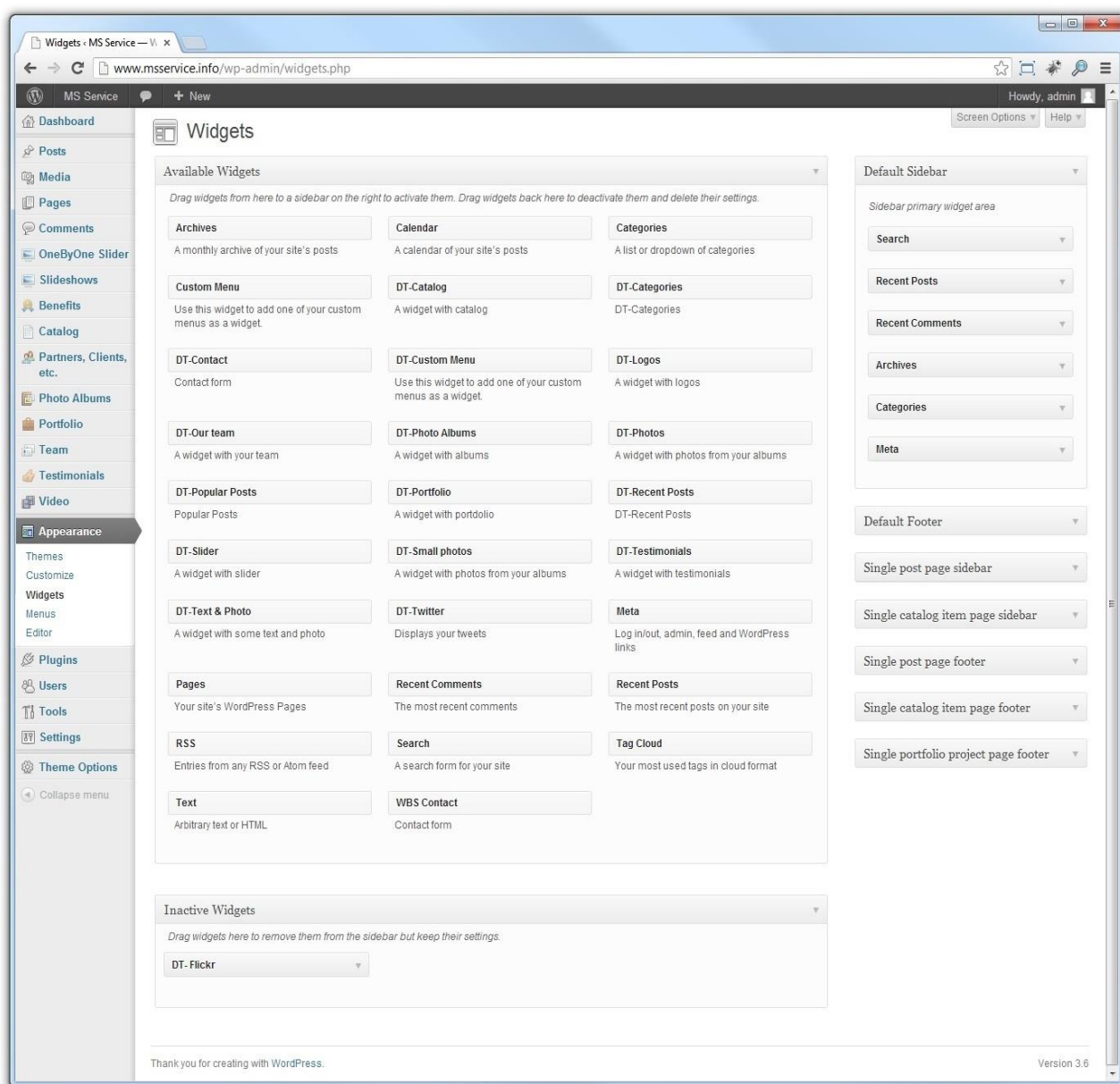


Figura 20: I Widget e le Sidebar.

Una **sidebar** non è altro che una specie di contenitore nel quale si inseriscono i widget che si vogliono visualizzare in una certa pagina.

Ovviamente, affinché le sidebar siano visibili all'interno di una pagina, non si devono utilizzare template di tipo "full-width", in quanto essi visualizzano il contenuto della pagina sfruttando la larghezza massima riservata al body, quindi senza sidebar.

Le sidebar all'interno di un sito sono molteplici, ognuna delle quali si riferisce ad un particolare template: per le pagine che costituiscono gli articoli del catalogo vi sarà associata la sidebar "Single catalog item page sidebar"; per il footer la sidebar "Default Footer" e così via.

È possibile creare nuove sidebar: alcuni temi prevedono questa opzione direttamente da back-end, altri invece no; nel secondo caso, o si crea direttamente con il codice (in fondo non è così complicato) oppure si ricorre all'installazione di uno dei tanti plugin che risolvono questo problema.

Un esempio di creazione di una sidebar direttamente dal codice è la seguente (questo codice è da inserire in "functions.php" all'interno del tema figlio):

```
if (function_exists('register_sidebar')) {  
  
    //sidebar Personalizzata  
  
    register_sidebar(array(  
        'name' => 'La Mia Slider,  
        'id' => 'sidebar_personalizzata'  
        'before_widget' => '<li class="widget widget-sidebar">',  
        'after_widget' => '</li>',  
        'before_title' => '<h3>',  
        'after_title' => '</h3>',  
    ));  
}
```

Per poter personalizzare le sidebar è sufficiente effettuare un'operazione di drag & drop dei widget all'interno della sidebar desiderata; per inserire il contenuto all'interno del widget è sufficiente espanderlo e inserire il testo nei campi appositi.

È possibile inserire più widget all'interno della stessa sidebar, ognuno dei quali (per la maggior parte dei casi) può essere presente più volte.

Nella creazione del sito di MS service, si sono utilizzate due differenti sidebar (figura 21):

- "Single catalog item page sidebar", visualizzata nelle pagine del catalogo;
- "Default footer".

Nella sidebar visualizzata nelle pagine del catalogo sono stati inseriti due widget: uno testuale, contenente una frase che invita gli utenti a contattare l'azienda per conoscere i dettagli dei servizi offerti, e l'altro è il "dt-contact", il form di contatti (il risultato lo si può vedere in figura 19).

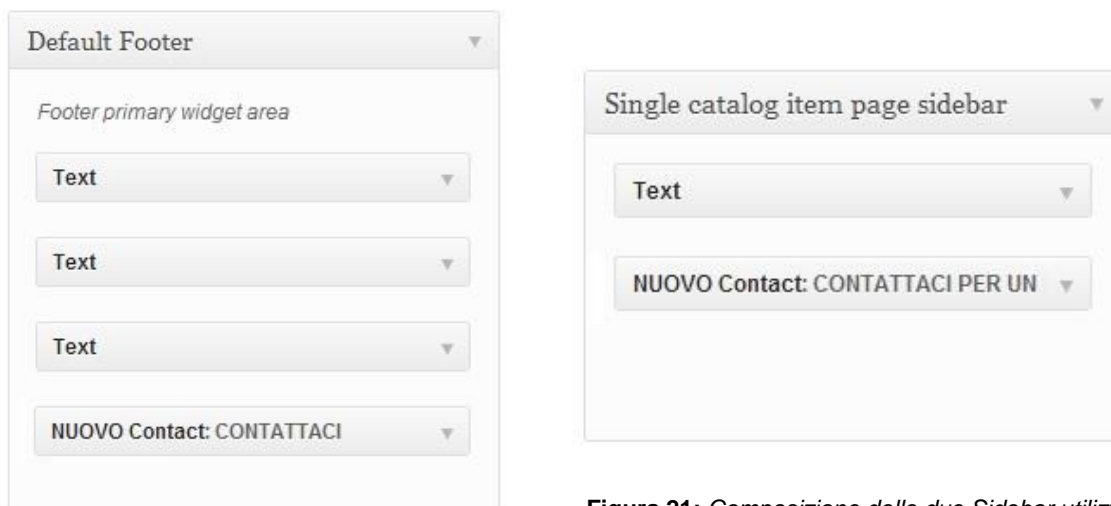


Figura 21: Composizione delle due Sidebar utilizzate.

Per quanto riguarda il footer, invece, è diviso in 4 parti: le prime 3 sono testuali, in cui sono inserite le informazioni di contatto e i link alle sezioni più importanti del sito; la 4 parte è invece occupata dal form di contatti (widget "dt-contact").

La figura 22 illustra il risultato.

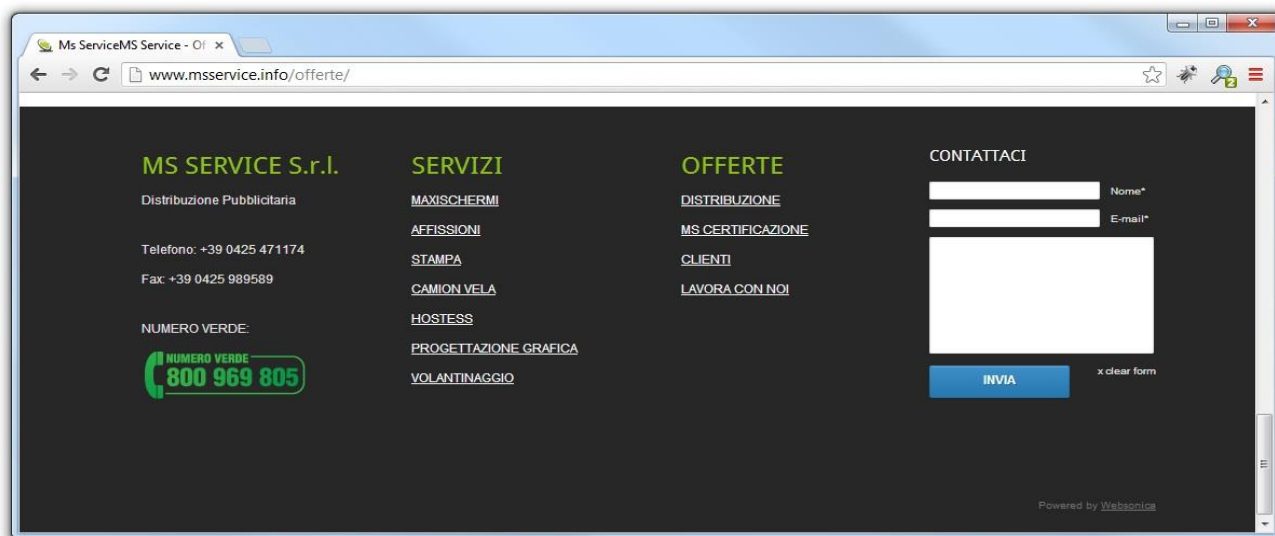


Figura 22: Il Footer.

### 3.2.2.6 IL MENU

Il menu è una delle parti fondamentali da implementare in quanto consente una facile navigazione all'interno del sito da parte dell'utente, oltre che ad un punto di riferimento in quanto presente in tutte le pagine/articoli/ecc., ed è bene scegliere con cura quali sezioni inserirvi.

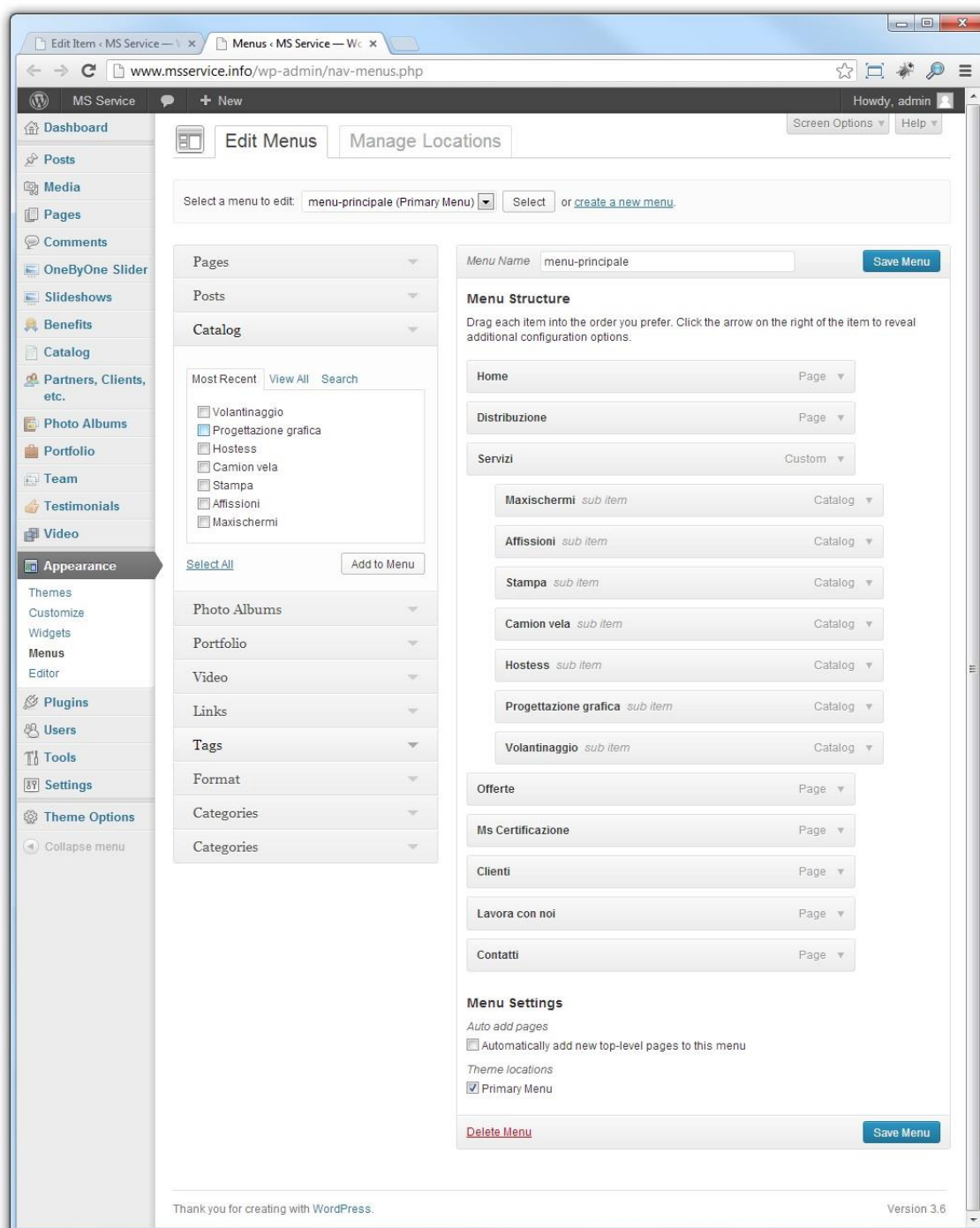


Figura 23: Creazione del Menu.

Wordpress prevede la possibilità di crearne diversi: uno principale che sarà inserito nella parte alta della pagina, sopra ai contenuti (è importante inserirlo in questo punto in quanto è la prima cosa che si vede all'apertura della pagina), e altri menu secondari da inserire nel Footer o nelle sidebar.

All'interno del menu è possibile inserire praticamente ogni elemento implementato nel sito, con la possibilità di creare delle gerarchie (sottomenu).

Per la sua creazione è sufficiente selezionare l'elemento desiderato dall'apposita area, sulla parte destra della pagina, e aggiungerlo; è possibile riordinare le voci secondo un ordine desiderato con un'operazione di drag & drop (come in figura 23).

Per ragioni di accessibilità è bene ricordare che ogni menu non deve essere troppo ampio e profondo (ampiezza massimo di 10, profondità di 4 o 5), ma ovviamente le richieste del cliente hanno la priorità, quindi se esso desidera un menu di ampiezza 20 è bene rendergli presente che la soluzione non è efficiente, ma va rispettata.

### **3.2.3 PROBLEMI LEGATI AL TEMA**

Quando si crea un sito può capitare che il tema installato o alcune opzioni a livello di codice sorgente debbano essere cambiate per soddisfare le richieste del cliente.

Nella realizzazione del sito di MS Service sono state necessarie due modifiche di questo tipo, per risolvere un problema legato ad un widget (contenente il form di contatti) e uno legato al template che regola la visualizzazione dei cataloghi.

Nel primo caso, la soluzione è stata quella di duplicare il widget in questione, personalizzandolo a dovere (il procedimento è stato spiegato nella sezione 3.2.3.1); nel secondo, è stato sufficiente modificare il template (sezione 3.2.3.2).

La scelta di agire direttamente nei file sorgente per operare queste modifiche è stata presa dopo un'attenta analisi delle componenti del sito da implementare e da ciò che offriva il tema.

Per prima cosa si è quindi inquadrata la natura delle modifiche da fare: traduzione di testo.

Le due modifiche, infatti, sono state necessarie in quanto i pulsanti presenti nelle due sezioni contenevano del testo in inglese, generato automaticamente dai template, mentre nel sito finale dovevano comparire in italiano.

A primo impatto si potrebbe pensare che la soluzione più corretta da prendere sia quella di installare un plugin che si occupi di traduzioni.

Purtroppo però non è così semplice: il miglior plugin che gestisce le traduzioni all'interno di un sito è WPML (vedi paragrafo 3.4.1), ma non è gratuito ed è un'assurdità chiedere al cliente di pagare una quota in più per un plugin che gestisce le traduzioni, quando il suo sito è previsto in una sola lingua.

Si può cercare un plugin di gestione delle traduzioni che sia gratuito, ma non è detto che preveda una sezione dedicata alla traduzione di stringhe contenute nei file dei template.

Trattandosi quindi di un sito implementato in una sola lingua, è ragionevole operare mediante modifica dei file sorgente, anche perché tutto sommato non è un procedimento complicato.

### 3.2.3.1 DUPLICAZIONE DI UN WIDGET

Il widget soggetto alle modifiche è quello (previsto del tema) del form di contatto, il "DT-contact". Tale modifica è stata necessaria per il fatto che nel bottone di invio del form, la scritta era "Submit", mentre il cliente desiderava ci fosse scritto "INVIA".

Per risolvere questo problema, il codice va inserito nel tema figlio, all'interno del file "functions.php".

```
/* CLONO IL WIDGET CONTACT FORM */

class NUOVO_contact_Widget extends WP_Widget {

    /* Widget setup */
    function __construct() {

        /* Widget settings. */
        $widget_ops = array( 'description' => _x( 'Contact form',
        'widget contact form', LANGUAGE_ZONE ) );

        /* Create the widget. */
        parent::__construct(nuovo-contact-widget,
            NUOVO . _x( ' Contact', 'widget contact form',
            LANGUAGE_ZONE ), $widget_ops);
    }
}
```

```

/* MODIFICO IL CLONE DEL WIDGET
(il codice di questa funzione è modificato a partire da quello
del widget originale)
*/

function widget( $args, $instance ) {

[ ... ]

<div class="but-wrap"><a href="#" class="button go_submit"
    title="<?php _ex( 'Invia', 'widget contact form',
    LANGUAGE_ZONE); ?>"><span><i class="submit"></i>
    <?php _ex( 'INVIA', 'widget contact form', LANGUAGE_ZONE
);
    ?></span></a>
</div>
[ ... ]
}
/* REGISTRAZIONE DEL NUOVO WIDGET */
function NUOVO_contact_register() {
    register_widget( 'NUOVO_contact_Widget' );
}

/* IL WIDGET VIENE CARICATO */
add_action( 'widgets_init', 'NUOVO_contact_register' );

```

### 3.2.3.2 CAMBIARE UNA PAROLA A LIVELLO DI FILE

Il tema Nimble per quanto riguarda la sezione del catalogo (utilizzato per implementare i servizi che l'azienda MS Service offre) prevede un apposito template con il quale definisce la modalità di visualizzazione di questa parte del sito.

Il problema sorge in quanto il bottone che manda al contenuto dell'articolo contiene la scritta "Details", mentre la grafica prevede ci sia scritto "Dettagli".

Il file in cui questo template è definito è chiamato "content-dt-catalog.php" e per poter eseguire la modifica richiesta è necessario copiarlo e modificarlo nel tema figlio come segue:



```

[ ... ]
<div class="<?php dt_portfolio_classes('2_col-
list','info');?>">
[ ... ]
<?php
    dt_the_content();
    dt_details_link_bis();
    dt_edit_link();
?>
</div>
[ ... ]
</div>

```

La scritta all'interno del bottone veniva infatti generate dalla funzione "dt-details-link ( )"; ho sostituito questa funzione con una chiamata "dt-details-link-bis ( )" da me definita all'interno del file "functions.php".

```

function dt_details_link_bis( $post_id = null, $class =
'button' ) {
    if( empty($post_id) ) {
        global $post;
        $post_id = $post->ID;
    }

    $url = get_permalink($post_id);

    if( $url ) {
        printf('<div class="but-wrap"><a href="%s"
class="%s"><span><i
class="more"></i>Dettagli</span></a></div>', $url,
$class);
    }
}

```

## 3.3 Terza fase: PUBBLICAZIONE DEL SITO

Per la fase di pubblicazione, dopo aver ricevuto la conferma dal cliente e gli accessi al database e al server nel quale risiederà il sito, due sono le operazioni da fare: copiare il database e trasferire i file sul server. Sono due operazioni molto semplici, ma devono essere svolte con particolare cura.

### 3.3.1 IL DATABASE

Per quanto riguarda il database, si deve fare il dump di quello che si è utilizzato per la realizzazione del sito fino a quel momento e successivamente lo si importa in quello di cui il cliente ha fornito gli accessi.

### 3.3.2 TRASFERIMENTO FILE TRAMITE FTP

Per prima cosa, è necessario modificare il file “wp-config.php” all’interno della cartella “msservice” che contiene l’intero sito.

In esso infatti erano state inserite le informazioni relative al database che veniva utilizzato in locale; ora tali parametri vanno cambiati, inserendo quelli che sono stati forniti dal cliente.

A questo punto si copia la cartella contenente il sito sul server, utilizzando un apposito software che funzioni come client FTP.

Ovviamente, se nel server era presente un sito che utilizzava lo stesso nome, è bene fare un backup della vecchia versione online prima di procedere all’upload di quello nuovo.

La pubblicazione di un sito può durare diverso tempo, e ciò dipende dalla quantità di file che lo compongono e dal dominio che il cliente ha scelto.

Per questo motivo, è bene procedere alla pubblicazione di un sito nella prima parte della giornata, in modo tale che a trasferimento di file compiuto, si possano correggere eventuali errori senza far passare troppo tempo (è fortemente sconsigliata per queste ragioni la pubblicazione di venerdì pomeriggio).

## 3.4 ALCUNI PLUGIN UTILIZZATI

Quando si ha a che fare con siti più complessi, indispensabile diventa l'utilizzo di funzionalità aggiuntive: i plugin.

Durante il periodo di stage effettuato, ho avuto modo di installare, utilizzare e modificare diversi plugin, ognuno dei quali compiva una precisa funzione.

A seguito presenterò quelli da me più utilizzati, dando un quadro generale dei problemi che essi risolvono, senza scendere troppo in dettaglio.

Installare un plugin è molto semplice: si accede alla sezione Plugins > Add New e si carica o mediante upload da computer (del file .zip, scaricato precedentemente) oppure si cerca direttamente da Wordpress e si installa.

Un plugin può essere attivato e disattivato in qualsiasi momento.

### 3.4.1 WPML

WPML (Wordpress MultiLingual Plugin) è il miglior plugin multilingua per Wordpress in circolazione e consente di creare il sito in tutte le lingue che si desidera, permettendo di gestire praticamente tutti i contenuti testuali, menu di navigazione, testi presenti all'interno del tema utilizzato, ecc.

Esso è scaricabile (a pagamento) dal sito <http://wpml.org/it/>.

Questo plugin è suddiviso in diverse parti, ognuna delle quali viene installata o meno, creando la combinazione più adeguata alle nostre esigenze.

**WPML Multilingual CMS** è il nucleo di questo plugin, e trasforma di fatto Wordpress in un sito multilingue. Questo modulo consente la duplicazione delle pagine, di articoli e dei menu nelle diverse lingue in cui esso è previsto.

Nel momento in cui si volesse ricorrere all'utilizzo di WPML, l'installazione di questo modulo è necessaria.

**WPML media Translation** è una delle componenti aggiuntive, che permette di organizzare la galleria di Wordpress in diverse lingue, in modo tale da scegliere quali immagini visualizzare per ogni specifica lingua.

**WPML String Translation** consente di tradurre i testi inseriti dal tema o generati dai plugin direttamente dal back-end del sito, senza dover utilizzare i file sorgente che permettono il confronto tra stringhe e, a seguito, di tradurle nella lingua desiderata (questi file sono generalmente di estensione .po, Portable Object Template, che poi vengono effettivamente utilizzati dall'applicazione finale tramite estrapolazione mediante codice PHP nei file di template).

**WPML Translation Management** è il modulo che aiuta gli amministratori del sito a gestire le traduzioni fatte da altri utenti con i quali si è in collaborazione.

Permette di inviare i contenuti ai collaboratori e seguire i progressi di traduzione del sito.

**WPML Translation Analytics** facilita la gestione delle traduzioni, in quanto questo lavoro può essere complicato, specialmente se si hanno un sacco di contenuti e si collabora con più traduttori.

Esso infatti permette di avere relazioni sintetiche sullo stato attuale della traduzione del sito, in modo tale da avere un quadro generale delle traduzioni nel tempo.

**WPML Sticky Links** mantiene aggiornati i collegamenti interni del sito. Esso è utile in particolare quando si cambia la struttura dei permalink, la gerarchia delle categorie ecc., in quanto esso tiene traccia automaticamente delle pagine collegate e aggiorna i link.

**WPML CMS Navigation** è il modulo che aggiunge elementi di navigazione al sito, che possono essere utilizzati durante la sua costruzione. Questo include breadcrumbs, menu a discesa (per cambiare lingua) e la navigazione della barra laterale.

### 3.4.2 MAGIC FIELDS

Durante la fase di creazione di un sito può succedere che il cliente richieda l'inserimento di un campo aggiuntivo non previsto dal tema utilizzato.

Per risolvere questo problema, si può ricorrere all'utilizzo del plugin "Magic Fields", che consente di creare nuovi campi che possono essere di diverso tipo, tra cui immagini, audio, slider, caselle di testo, selettori di data, caricamento di file, ecc., che possono essere posizionati ovunque si desideri, modificando il codice del template corrispondente, inserendo la funzione:

```
get ( ' nome_del_field ' ).
```

Ogni field viene creato in una particolare sezione del sito, di cui esso fa parte (pagine, articoli, immagini, ecc.) e sarà inserito automaticamente nel back-end tra i campi aggiuntivi dell'elemento corrispondente.

A volte diventa necessario inserire un campo aggiuntivo di questo tipo (per esempio l'upload di un file) per facilitare l'operazione di inserimento dei contenuti da parte di un utente esterno che non ha competenze necessarie per la programmazione.

### 3.4.3 CONTACT FORM 7

Questo plugin, noto anche con la sua abbreviazione CF7, permette la gestione di uno o più moduli di contatto, personalizzabili. È possibile, oltre a scegliere le componenti del modulo, definire a proprio piacimento il comportamento e il contenuto delle email.

Gli elementi che possono comporre il modulo sono di svariati tipi, i principali sono:

- Text field, semplici caselle di testo;
- Email, text field in cui viene fatto il controllo di validazione per le email;
- URL, text field in cui viene fatto il controllo di validazione per le URL;
- Telephone number, text field in cui viene fatto il controllo di validazione per i numeri di telefono;
- Number, in cui è possibile introdurre esclusivamente numeri;
- Date, con il quale inserire una data (ad esempio, data di nascita);
- Text area, genera un'area in cui l'utente può inserire il testo da inviare;
- Checkboxes, utilizzate qualora vi siano più opzioni possibili che l'utente può selezionare;
- Radio buttons, utilizzate nel caso in cui l'utente debba selezionare una sola tra le opzioni indicate;
- File upload, che prevede il caricamento di un file da parte dell'utente (per esempio, l'invio di un curriculum all'azienda);
- Submit button, il bottone che invia i contenuti inseriti nel form.

### 3.4.4 NEXTGEN GALLERY

Può accadere che il cliente richieda l'inserimento di una photo-gallery in un punto del sito in cui il template non la preveda.

Per risolvere questo problema, è sufficiente ricorrere all'installazione di uno dei tanti plugin che implementano una gallery: NextGen Gallery è uno di questi.

Con questo plugin, oltre alle gallerie di immagini, è prevista la creazione di slideshow in modo semplice ed efficace.

### 3.4.5 WOOCOMMERCE

L'e-Commerce (o commercio elettronico) è una delle soluzioni più ricercate dai clienti, che si riferisce all'insieme della comunicazione e della gestione di attività commerciali attraverso modalità elettroniche, con sistemi automatizzati di raccolta dati.

Esso è oggi una forma di commercio ampiamente sviluppata che riguarda le transazioni tramite le moderne tecnologie telematiche; coinvolge, oltre al tradizionale scambio di beni e servizi, anche una più vasta gamma di servizi virtuali (come l'e-banking) che, in molti casi, non sarebbe possibile promuovere nel mercato tradizionale.

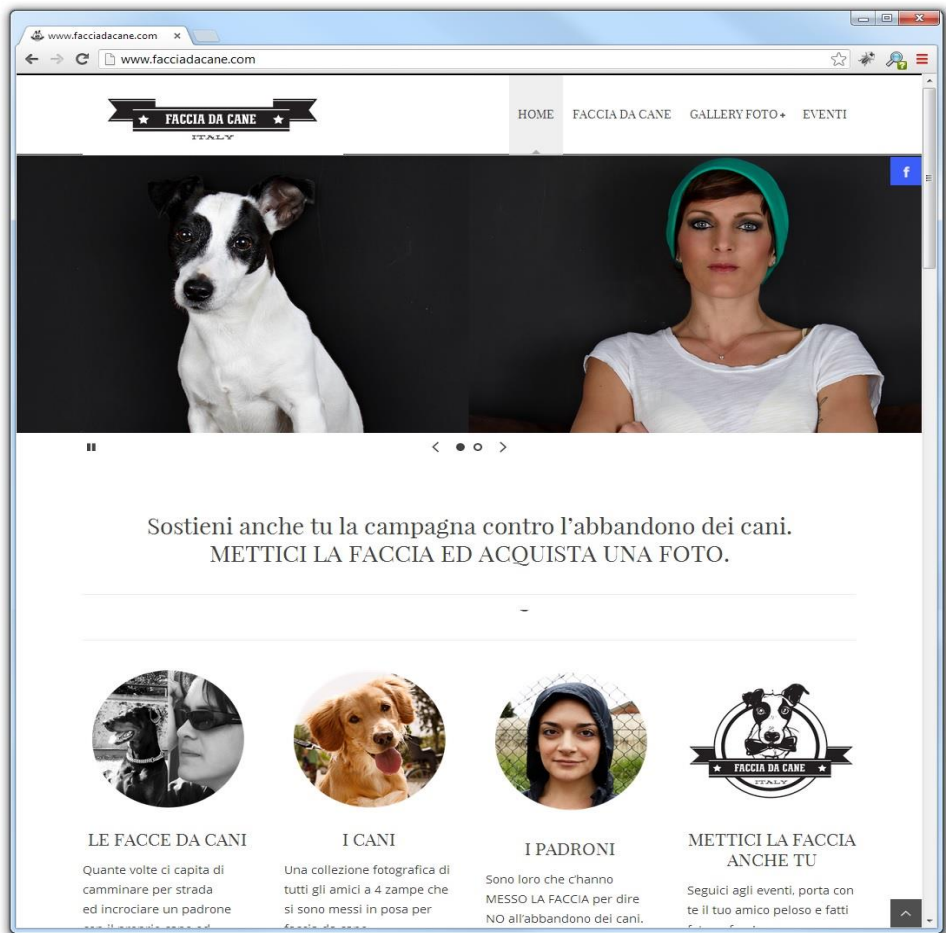
Realizzare un e-Commerce è tutt'altro che semplice: ecco perché utilizzare un plugin che lo implementi diventa la scelta più corretta da prendere.

Come detto in precedenza (sezione 2.2), Wordpress non è il CMS più adatto per gestire un e-Commerce, in quanto è stato concepito per la creazione di blog o per siti di ristrette dimensioni.

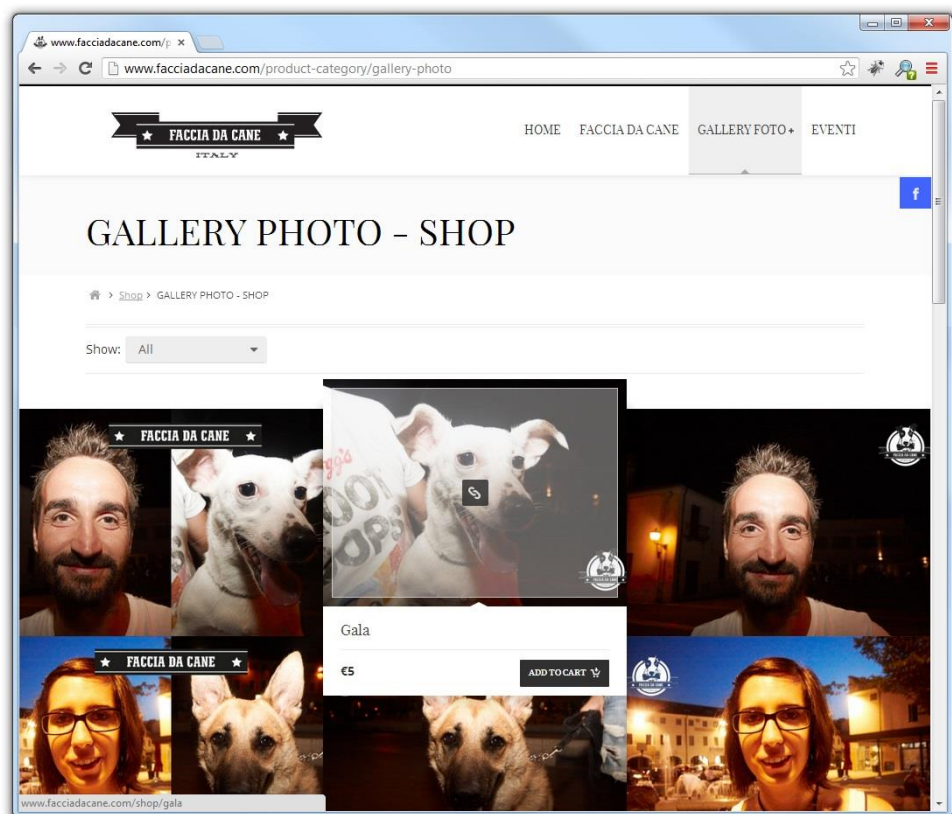
Ciò nonostante, se l'e-Commerce da implementare è molto semplice e con un numero contenuto di prodotti, è possibile farlo anche su questo CMS grazie al plugin chiamato "Woocommerce", di alta qualità e in continua crescita.

Durante il periodo di stage, uno dei siti con i quali ho avuto a che fare è stato [www.facciadacane.com](http://www.facciadacane.com) (figura 24), che utilizza questo plugin per implementare l'acquisto di immagini (figura 25).

Essendo una forma di e-Commerce molto semplice, è preferibile utilizzare Wordpress e questo plugin per la sua creazione, senza ricorrere a CMS più elaborati e complicati di cui si utilizzerebbe solo una minima parte.



**Figura 24:**  
Homepage di  
www.facciadacane.com



**Figura 25:**  
Photogallery in cui acquistare le immagini.

Woocommerce si sviluppa in due sezioni distinte, presenti nel menu di navigazione di Wordpress: Woocommerce e Products.

La sezione **Woocommerce** implementa:

- Orders (Ordini), che contiene l'elenco dei prodotti venduti con il relativo stato dell'ordine, con le informazioni di spedizione;
- Reports (Statistiche), dove sono riportati i grafici e alcuni dati sulle statistiche, come il numero totale di ordini, il numero totale delle vendite, ecc.;
- Coupons (Buoni Sconto), in cui vengono gestiti gli sconti con un sistema a Coupon;
- Settings (Impostazioni), nel quale si gestiscono i sistemi di pagamento abilitati e le impostazioni generali;
- System Status (Stato del sistema), in cui sono riportate le informazioni relative allo stato del sistema e i dati tecnici.

La sezione **Products**, invece, gestisce tutte le informazioni relative ai prodotti:

- Products (Prodotti), in cui vi è un elenco di tutti i prodotti inseriti;
- Add Product (Aggiungi Prodotto), sezione dedicata all'inserimento di un nuovo prodotto;
- Categories (Categorie), in cui definire eventuali categorie in cui suddividere i prodotti (come si fa per gli articoli);
- Tags, dedicato alla gestione dei Tag dei prodotti;
- Shipping Classes (Classi di Spedizione), dove vengono create le classi di spedizione per accomunare prodotti simili;
- Attributes (Attributi), in cui creare attributi extra da attribuire al prodotto, come taglia o colore.



# Capitolo 4

## Conclusioni

In questo lavoro ho riportato l'esperienza di stage da me effettuata presso Websonica S.r.L. al termine del mio percorso universitario.

Durante il periodo trascorso (circa 500 ore), ho avuto modo di acquisire le competenze che, assieme alle conoscenze personali maturate durante la mia carriera scolastica, mi hanno permesso di portare a termine i compiti a me assegnati che erano stati concordati inizialmente con il tutor.

Nella sezione teorica (cap. 2) ho presentato gli strumenti utilizzati, e il funzionamento e le caratteristiche generali dei CMS.

La seconda parte (cap. 3), invece, è stata dedicata a Wordpress, il CMS adottato dall'azienda che ho utilizzato, dando importanza al processo di creazione del sito [www.msservice.info](http://www.msservice.info).

I siti con i quali ho lavorato sono stati diversi, circa 20. Le operazioni che ho avuto modo di effettuare sono:

- creazione di siti web;
- inserimento dei contenuti;
- modifica e creazione di template dal codice sorgente;
- modifica dei CSS, prestando particolare attenzione al responsive;
- duplicazione di widget e creazione di sidebar da codice sorgente;
- modifiche ai plugin;
- inserimento delle traduzioni laddove il sito fosse multilingua;
- creazione di newsletter, scritte in linguaggio HTML;
- backup e pubblicazione di siti web.

Grazie a questa esperienza ho potuto constatare l'importanza dell'utilizzo di un CMS come Wordpress nella realizzazione di siti web, in particolare:

- facilità nella gestione del database;
- velocità di realizzazione di un sito;
- semplicità dell'inserimento dei contenuti, grazie alla dualità dell'interfaccia messa a disposizione da questo CMS, che permette la formattazione del testo sia nell'editor WYSIWYG (*Visuale*) per gli utenti meno esperti, sia tramite codice HTML (*Testo*);
- facilità con cui è possibile cambiare totalmente la grafica, mantenendo i contenuti (utilizzando i numerosi template grafici previsti per Wordpress);
- possibilità di personalizzare la grafica creando un tema figlio, che permette di effettuare le modifiche senza imbattersi nei problemi legati agli aggiornamenti del software;
- la facilità di inserire componenti aggiuntive in modo semplice, grazie ai plugin di cui Wordpress è ben fornito;
- la semplice metodologia di inserimento di immagini e di altri allegati, con la possibilità di avere un'area di gestione centralizzata di questi file, nella sezione "Media" del back-end.

In conclusione, posso affermare che è bene tenere in considerazione questi sistemi nel momento in cui si debba affrontare la creazione di un sito web, sia per la sua discreta facilità di utilizzo che riduce i tempi necessari per implementarlo, sia perché aiuta la gestione del sito, specialmente per quanto riguarda gli aspetti più delicati come il collegamento con il database.

# Bibliografia

- PHP, Apache e MySQL, sviluppo di siti web - E. Naramore, J. Gemer, Y. Le Scouamec, J. Stolz, M. K. Glass
- Informatica: sistemi operativi e reti per il sistema informativo aziendale - Agostino Lorenzi, Richelmo Giupponi
- Informatica: le basi di dati e il linguaggio sql access, Mysql, database in rete - Agostino Lorenzi, Enrico Cavalli
- Ado.net e asp.net - Michele De Benedittis

## Siti web consultati

- <http://it.forums.wordpress.com/>
- <http://bloggerfaidate.com/category/wordpress/>
- <http://bloggerfaidate.com/trackback-e-pingback-capire-differenza-per-gestire-meglio-blog/>
- <http://tutorial.altervista.org/wordpress/>
- <http://wordpress.org/>
- <http://wpml.org/it/>
- <http://www.marchettidesign.net/2013/04/creare-un-e-commerce-con-wordpress-e-woocommerce/>
- <http://www.blogjoomla.it/2009112760/cms-content-management-system.html>
- <http://it.wikipedia.org/>
- <http://www.mondositiweb.com/2008/12/cosa-e-un-cms-spiegato-in-modo-semplce/>
- <http://www.gianarb.it/blog/il-backend-di-wordpress/>
- <http://www.wptutorial.it/tag/wpml-plugin/>
- Legge Stanca [http://www.pubbliaccesso.gov.it/normative/legge\\_20040109\\_n4.htm](http://www.pubbliaccesso.gov.it/normative/legge_20040109_n4.htm)