

*Università degli Studi di Padova*

*Facoltà Di Ingegneria*

*Dipartimento di Tecnica e Gestione dei Sistemi Industriali*

*Tesi di Laurea di Primo Livello*

SELEZIONE DEL SOFTWARE PER UN PORTALE DI  
DIPARTIMENTO UNIVERSITARIO:  
CASO DI STUDIO CON JOOMLA!

Relatore: Ing. Reggiani Monica

Laureando: Bonato Jacopo

ANNO ACCADEMICO 2010-2011



# INDICE

CAPITOLO 1 Introduzione .....	1
CAPITOLO 2 Descrizione del problema e dello stato dell'arte .....	3
1.Descrizione del problema .....	3
1.1.Scopo del progetto .....	3
1.1.1.Introduzione .....	3
1.1.2.La veste grafica .....	3
1.1.3.I contenuti .....	4
1.1.4.La qualità delle informazioni .....	5
1.1.5.L'accessibilità .....	5
1.2.La nostra scelta .....	5
2.Cos'è un CMS .....	6
2.1.Introduzione .....	6
2.2.Storia .....	6
2.3.Utilizzo .....	7
3. CMS vs Macromedia Dreamweaver .....	7
3.1.La scelta .....	7
3.1.1.Vantaggi dei CMS .....	7
3.1.2.Limiti dei CMS .....	9
3.1.3.Cos'è Macromedia Dreamweaver .....	9
3.1.4.Vantaggi di Macromedia Dreamweaver: .....	10
3.1.5.Svantaggi di Macromedia Dreamweaver: .....	10
3.1.6.Conclusioni confronto .....	10
4.Accessibilità .....	11
4.1.L'accessibilità nel Web .....	11
4.1.1.Requisiti fondamentali .....	11
4.1.2.CMS accessibili .....	12
4.2.Standard e Leggi .....	12
4.2.1.In Italia .....	12
4.2.2.Nel resto del mondo .....	13
4.3.W3C .....	13
4.3.1.Introduzione .....	13
4.3.2.Membri .....	13
4.3.3.Obiettivi del W3C .....	14
4.3.4.Iniziative del W3C .....	14
4.3.5.La situazione in Italia .....	15

5.L'usabilità di un sito web .....	15
5.1.Introduzione .....	15
5.2.Il parere degli esperti circa l'usabilità .....	15
5.2.1.Verso l'usabilità .....	15
5.2.2.Il parere contrario .....	16
5.2.3.Conclusione dibattito .....	17
6.Progettare un buon sito .....	17
6.1.Introduzione .....	17
6.2.I contenuti .....	18
6.3.Lo stile .....	18
6.4.Gli espedienti visivi .....	18
6.5.Conclusioni .....	19
7.Le Licenze .....	19
7.1.Introduzione .....	19
7.2.L'open source .....	20
7.2.1.Vantaggi dell'open source .....	20
7.2.2.Svantaggi dell'open source .....	21
7.2.3.La nostra situazione .....	21
7.3.Il mondo delle licenze .....	22
7.3.1.Introduzione .....	22
7.3.2.Tipi di licenza .....	22
7.3.3.La GNU General Public License .....	22
7.3.4.La GNU Lesser General Public License .....	23
7.3.5.La CPL Common Public License .....	24
8.PHP .....	24
8.1.Introduzione .....	24
8.2.Caratteristiche .....	25
8.3.Diffusione .....	25
CAPITOLO 3 Descrizione della metodologia utilizzata .....	27
1.Il processo di software selection .....	27
2.Prima fase: ricerca di potenziali soluzioni .....	27
3.Seconda fase: stabilire un livello di informazione minimo .....	28
4.Terza fase: creazione della long list .....	29
5.Quarta fase: casi di studio e requisiti .....	30
5.1.Casi d'uso .....	30
5.2.Analisi dei requisiti .....	31
5.2.1.Difficoltà nell'analisi .....	32

5.2.2.Classificazione dei requisiti .....	33
5.2.3.Requisiti funzionali .....	34
5.2.4.Requisiti non funzionali .....	35
6.Quinta e sesta fase: dalla long alla short list .....	36
CAPITOLO 4 Il processo di software selection .....	39
1.Casi d'uso .....	39
2.Requisiti .....	43
2.1.Requisiti funzionali: capacità .....	43
2.2.Requisiti funzionali: sicurezza .....	44
2.3.Requisiti non funzionali: usabilità .....	45
2.3.1.Usabilità: fattori umani .....	45
2.3.2.Usabilità: help e documentazione .....	45
2.4.Requisiti non funzionali: affidabilità .....	46
2.5.Requisiti non funzionali: sostenibilità .....	47
2.5.1.Sostenibilità: adattabilità .....	47
2.5.2.Sostenibilità: manutenibilità .....	47
2.5.3.Sostenibilità: internazionalizzazione .....	47
2.5.4.Configurabilità .....	48
2.6.Requisiti non funzionali: implementazione .....	48
2.6.1.Implementazione: limitazione sulle risorse .....	48
2.6.2.Implementazione: linguaggi e strumenti.....	48
2.6.3.Implementazione: requisiti di interfaccia .....	49
2.6.4.Implementazione: requisiti operativi .....	49
2.6.5.Implementazione: requisiti fisici .....	49
3.Presentazione dei CMS .....	49
3.1.CMS Made Simple .....	49
3.2.dotCMS .....	50
3.3.Drupal .....	50
3.4.e107 .....	50
3.5.Joomla! .....	50
3.6.Mambo .....	51
3.7.MODx .....	51
3.8.mojoPortal .....	51
3.9.ONEcms .....	51
3.10.OpenCMS .....	52
3.11.PHP-Fusion .....	52
3.12.PHP-Nuke .....	52

3.13.Plone .....	53
3.14.SilverStripe .....	53
3.15.TikiWiki CMS .....	53
3.16.TYPO3 .....	54
3.17.TYPOlight .....	54
3.18.XOOPS .....	54
4.Long List .....	55
4.1.Tabella comparativa .....	55
4.2.Classifica .....	57
5.Short List .....	58
5.1.Considerazioni .....	58
CAPITOLO 5 Realizzazione di un portale universitario con Joomla!.....	61
1.Cos'è Joomla! .....	61
1.1.WAMP Server .....	62
2.Storia di Joomla! .....	62
3.Installazione di Joomla! .....	62
3.1.Installazione di WAMP Server .....	62
3.2.Installazione di Joomla! .....	65
4.Gestione dei contenuti .....	67
4.1.Creazione di una seziona/categoria .....	69
4.2.Creazione di un articolo .....	70
4.3.I moduli .....	71
4.4.Creazione e gestione dei menù .....	72
4.5.I componenti .....	76
4.6.Le estensioni .....	76
4.7.Creazione e gestione degli eventi .....	77
5.Gestione degli utenti .....	81
5.1.Gruppi di utenti .....	82
6.Supporto multilingua .....	83
7.Realizzazione del portale del Dipartimento di Tecnica e Gestione dei Sistemi Industriali .....	85
8.Backup e Ripristino .....	87
8.1.Backup del database .....	88
8.2.Backup del sito .....	88
8.3.Restore.....	89
8.4.Backup tramite Akeeba Backup.....	90
9.Rispetto dei requisiti .....	91

CAPITOLO 6 Conclusioni .....	94
BIBLIOGRAFIA .....	96





# CAPITOLO 1

## Introduzione

Con questo elaborato si vuole presentare un progetto di collaborazione tra studenti di Ingegneria Gestionale del Dipartimento di Tecnica e Gestione dei Sistemi Industriali dell'Università degli Studi di Padova. Il progetto nasce dalla necessità di migliorare l'attuale portale web del dipartimento perché, se confrontato con ciò che realtà simili al DTG offrono su Internet, è evidente una certa carenza nell'aspetto, nell'organizzazione della struttura e nei contenuti.

Il nuovo portale dovrà essere in grado di fornire in modo chiaro ed esauriente tutte le informazioni di cui necessitano sia gli studenti e i docenti, ma anche i potenziali futuri iscritti e le aziende interessate alle attività di ricerca svolte nelle strutture del dipartimento. I contenuti dovranno essere meglio classificati per rendere la ricerca più intuitiva e meno dispersiva. Per fare ciò è necessario creare sezioni ad hoc per indirizzare ciascuna categoria di utenza verso le informazioni di cui ha bisogno.

La prima parte del progetto si è svolta in team con un'attività di *software selection*, per individuare un numero ristretto di prodotti adatti alla realizzazione del portale. Tale selezione è avvenuta applicando un processo rigoroso che si articola in diverse fasi: lo studio della situazione attuale, l'analisi delle esigenze che permettono di migliorarla e la traduzione di queste esigenze in requisiti. Si è anche studiato una mappa del sito ottimale che si adattasse al meglio a una realtà universitaria.

La seconda parte, svolta individualmente, ha visto la concretizzazione della fase precedente. A ciascun membro del team è stato assegnato un software con l'obiettivo di realizzare il sito progettato, in modo da testare sul campo la propria piattaforma.

La comparazione finale delle esperienze incontrate ha permesso di ultimare il processo di selezione software, individuando il prodotto migliore.

La struttura di questa tesi riprende i passi compiuti nell'affrontare il progetto. Nei prossimi capitoli sono descritti in modo dettagliato e approfondito tutti i passaggi elencati in precedenza.

Il capitolo 2 tratta del problema affrontato, degli obiettivi del progetto, di quale tipologia di software si è voluto adottare e il motivo di questa scelta. Si è quindi concentrata l'attenzione sulle licenze software, con uno sguardo particolare verso il mondo open source. In relazione ai limiti dell'attuale portale, sono trattati anche temi quali l'usabilità di un sito web e la legislazione in favore dell'accessibilità dei portali per gli utenti diversamente abili.

Nel capitolo 3 è descritta con un approccio teorico la metodologia seguita nella fase di studio iniziale e nel processo di software selection, analizzando in modo dettagliato ciascuno step affrontato. Si parte con la ricerca delle possibili soluzioni e la definizione delle proprietà di base che ciascun software candidato deve possedere. Successivamente si passa alla creazione della long list e la fissazione dei requisiti, attraverso l'analisi dei casi d'uso tipici a cui il software andrà in contro. È inoltre descritta la tecnica FURPS+ utilizzata per la raccolta e la classificazione dei criteri fissati. Il capitolo si conclude con la descrizione dei passaggi che portano a definire la short list.

Nel capitolo 4 sono riepilogate le fasi operative della prima parte del progetto: sono riportati i casi d'uso a cui il portale dovrà far fronte, sono elencati e descritti i requisiti fissati per la selezione dei CMS e vengono presentate le piattaforme scelte per essere inserite nella long list. Si prosegue con il report dei punteggi attribuiti nella long list e l'individuazione dei candidati per la short list. La lista definitiva fa seguito ad alcune considerazioni sull'effettiva adottabilità di ciascun candidato.

Il capitolo 5 verte sull'esperienza personale di ciascun componente del team. In questo elaborato è analizzata la realizzazione di un portale di dipartimento universitario con l'ausilio di Plone, un CMS open source. Sono presentate le caratteristiche di base del software, di alcune sue particolarità e delle funzionalità che hanno permesso di realizzare la struttura del sito precedentemente progettata.

A conclusione del lavoro sono state stilate alcune conclusioni comparative tra i vari CMS, evidenziando quali fossero le debolezze o i punti di forza di ciascun pacchetto software rispetto ai concorrenti. A seguito di queste si è potuto individuare la piattaforma che più si adatta all'obiettivo fissato.

# CAPITOLO 2

## Descrizione del problema e dello stato dell'arte

Nel capitolo che si andrà tra breve a trattare saranno descritte le ragioni che hanno portato allo sviluppo di questo lavoro e i concetti chiave per interpretarne fino in fondo il carattere.

Verranno presentate, in un primo momento, le cause, legate ai problemi incontrati durante la navigazione all'interno del sito attuale del Dipartimento di Tecnica e Gestione dei Sistemi Industriali. Problemi quali la grafica, i contenuti, la qualità delle informazioni e l'accessibilità.

In secondo luogo verrà chiarito il significato di CMS, parola cardine dell'analisi, e la scelta verso tale categoria di software. Per infine permettere una visione più generale del problema, saranno enunciate tre questioni fondamentali, l'accessibilità di un sito, l'usabilità dello stesso e il mondo delle licenze.

Il capitolo si concluderà con una breve descrizione del linguaggio di programmazione più usato tra i CMS esaminati, cioè PHP.

### 1.Descrizione del problema

#### 1.1.Scopo del progetto

##### *1.1.1.Introduzione*

Gli studenti della facoltà di Ingegneria presso l'Università degli Studi di Padova, sede di Vicenza, hanno l'esigenza quotidiana di navigare all'interno del sito del Dipartimento per avere tutte le informazioni necessarie per il loro cammino di studi. Questo servizio è di fondamentale importanza anche per altre numerose tipologie di utenti, quali docenti, ricercatori, dottorandi, personale tecnico amministrativo, aziende esterne e in modo particolare nuovi probabili futuri studenti. È da sottolineare proprio quest'ultima categoria in quanto soprattutto grazie ad essa ci si è posti il seguente quesito: è migliorabile il sito del Dipartimento di Tecnica e Gestione dei Sistemi Industriali, sede di Vicenza, esistente?

##### *1.1.2.La veste grafica*

Tale sito ha circa un anno di vita e continua ad essere sottoposto a miglioramenti da parte dei tecnici informatici del Dipartimento. Il loro lavoro, in base al tempo che gli possono dedicare, è di buona qualità se ci si basa sugli aspetti ai cui hanno voluto dare importanza. Ma, dopo uno studio accurato, si è giunti alla conclusione che ciò che si

voleva avesse maggiore rilievo, non coincideva con i risultati di tale analisi. L'aspetto grafico è scarsamente curato e quindi il primo impatto che si ha con la home page del Dipartimento non è molto accattivante. Questo non è che una sottigliezza se ad affacciarsi è uno studente già iscritto, ma potrebbe essere un punto penalizzante per un ragazzo delle medie superiori, il quale immerso nelle difficoltà dell'importante scelta per il prossimo cammino accademico, accostandosi al sito cerca quel qualcosa in più per accertare la bontà della sua decisione. Tale conferma, che è certamente data dalla serietà del sito internet, può essere aumentata da una veste grafica invitante grazie, ad esempio ad un numero maggiore di foto, le quali possono mettere in risalto la grande qualità degli edifici e dei servizi di cui dispone la sede di Vicenza dell'Università degli Studi di Padova.

Ciò che è stato appena descritto non è certamente la questione di maggior rilevanza messa in luce dall'indagine fatta, ma come si è detto in precedenza uno dei motivi che ha dato il via a tale studio.

### *1.1.3.1 contenuti*

Il vero problema risulta essere la non divisione dei contenuti tra utenti esterni e interni. Ora come ora i vari settori si mischiano l'un con l'altro a scapito della chiarezza e linearità dei vari temi proposti. Ciò comporta un rallentamento nella velocità di navigazione all'interno del sito stesso. Per futuri studenti, società/enti interessati delle attività svolte o semplicemente utenti curiosi l'accesso ai contenuti non è ottimizzato, cioè si incontrano difficoltà nel reperire le informazioni desiderate. Questo fatto non è dato certamente dalla mancanza di competenza degli autori del sito attuale, ma da un carente studio di quale potesse essere la struttura di base del sito stesso per rendere ottimale la navigazione al suo interno da parte di tutti, probabilmente per una mancanza effettiva di tempo disponibile. Ad esempio, aprendo la pagina iniziale, non c'è chiarezza su chi siano i destinatari degli avvisi. In particolare si può fare riferimento alle news, dove non è spiegato se siano rivolte ai frequentanti dei corsi di gestionale, di meccanica o di mecatronica oppure se siano rivolte a tutti. Vengono soprattutto mescolate informazioni tecniche, date dei ricevimenti e altre notizie strettamente didattiche per gli studenti in corso a contenuti informativi sul Dipartimento di Tecnica e Gestione dei Sistemi Industriali rivolte invece a tutt'altri destinatari, che per lo più sono al loro primo accesso al sito. Nella pagina di apertura sono evidenziate le offerte formative, le quali sono ben distinte per i vari corsi di laurea, ma essendo immerse tra tutti gli altri tipi di informazione perdono di valore.

Manca uno spazio dedicato alle FAQ, cioè a tutte quelle domande che sorgono con più frequenza ad un possibile visitatore, soprattutto in quegli utenti che sono alla ricerca di

informazioni per le prime volte, notizie che potrebbero essere anche solo di carattere orientativo e che l'utente stesso ha l'esigenza di reperire velocemente.

#### *1.1.4. La qualità delle informazioni*

Si nota una mancanza di omogeneità e completezza nella descrizioni di alcuni settori. E' molto difficile avere informazioni utili sugli insegnamenti e su chi siano i rispettivi docenti, quasi da essere considerati due concetti completamente distaccati. Il materiale dei vari corsi è presente e facilmente raggiungibile, ma potendo uno stesso corso essere svolto da diversi docenti che possono fornire diverso materiale didattico, le distinzioni a riguardo non sono così chiare. Discorso simile si può fare in ambito ricerca, dove non c'è uno spazio comune per capire fino in fondo come siano divisi i vari gruppi, quale sia la loro struttura organizzativa, quali argomenti trattino e chi siano i ricercatori. Tali notizie potrebbero essere utili anche ad aziende esterne, particolare di non poca importanza vista la grande presenza di imprese nel territorio e il rapporto che esse hanno con l'università, legame che è di interesse comune diventi sempre più forte.

#### *1.1.5. L'accessibilità*

Un discorso a parte merita la questione dell'accessibilità che verrà in seguito approfondita. Esistono alcuni standard ed alcune linee guida per definire l'accessibilità. Le linee guida internazionalmente più diffuse per quanto concerne il web si basano sulla certificazione W3C. In Italia le norme riguardanti le caratteristiche che deve avere il sito internet di una Pubblica Amministrazione fanno capo alla Legge Stanca. Essendo però molto rigida su talune questioni quali per esempio l'ipovisibilità, tale normativa viene rispettata da un numero estremamente esiguo di amministrazioni, ed oltretutto non è ancora del tutto operativa. Quindi per l'analisi in questione la certificazione W3C sarà il punto di riferimento.

#### 1.2. La nostra scelta

Dopo tutte le considerazioni fatte e appena descritte, si è pensato ad una nuova possibile realizzazione del sito del Dipartimento considerando che oggi un sito Internet serve ad informare, risolvere i problemi degli utenti e cercare di attirare nuove opportunità. Valutando il panorama globale delle offerte per la realizzazione di siti, si è cercato lo strumento più idoneo per soddisfare le nostre pretese. Dopo una prima scrematura dovuta alle risorse disponibili, le conoscenze informatiche possedute e il budget a disposizione, la scelta è ricaduta sull'utilizzo dei CMS, Content Management System. Dallo studio di un ampio numero di Sistemi di Gestione dei Contenuti è stato

possibile individuare la migliore alternativa per la risoluzione del problema. Tutta l'analisi è riportata nei capitoli e paragrafi seguenti.

## **2.Cos'è un CMS**

### 2.1.Introduzione

Content Management System (CMS in acronimo), letteralmente “sistema di gestione dei contenuti”, è una categoria di sistemi software per organizzare e facilitare la creazione collaborativa di documenti e altri contenuti. Un Content Management System è uno strumento che permette di ottenere il controllo sulla creazione e sulla distribuzione di informazione, un sistema di gestione dei contenuti che consente di creare, modificare e gestire le informazioni offerte attraverso un'interfaccia semplice ed intuitiva, facilitando così il compito di gestione del portale informativo.

Un CMS si installa su un server web per la gestione automatizzata di siti web tramite pannello di controllo protetto da password. L'utilità di questi sistemi non sarebbe a priori limitata alla gestione di siti web (particolarmente se di grandi dimensioni e frequentemente aggiornati); tuttavia proprio questo è attualmente il loro utilizzo più proficuo e diffuso.

I CMS possono essere programmati in vari linguaggi tra cui più comunemente in PHP e ASP; il tipo di linguaggio adoperato è indifferente a livello di funzionalità. Solitamente i CMS in PHP vengono utilizzati per piattaforme Linux, i CMS in ASP vengono utilizzati su piattaforme Windows.

### 2.2.Storia

I Content Management System sono nati negli Stati Uniti e sono stati inizialmente sviluppati da alcune organizzazioni che producevano notevoli quantità di pubblicazioni, per il loro uso interno. Nel 1995 la CNET rese pubblici gli studi e i prodotti sviluppati internamente, distribuendoli con l'etichetta *Vignette*. La compagnia cominciò a mettere a disposizione il proprio software come sistema di gestione dei contenuti via web. Per la prima volta un utente poteva creare il proprio sito direttamente dal web, usando l'interfaccia CNET.

Nel 1998, la Pencom Web Works, una compagnia di consulenza aziendale, introdusse il server di trasformazione dati (DTS) *Metaphoria*, che permetteva agli sviluppatori Java di scrivere applicazioni che si potevano collegare ai contenuti e permettevano di distribuire tali contenuti su canali diversi. Il prodotto non ebbe successo, ma il concetto che era stato introdotto costituì le basi di ciò che è diventato il CMS odierno.

### 2.3.Utilizzo

Nonostante i CMS non siano stati concepiti per il Web, oggi il loro utilizzo più diffuso è rivolto alla gestione di siti web, soprattutto se di grandi dimensioni e con necessità di frequente aggiornamento.

Una delle applicazioni più utili infatti si ha nella gestione dei portali, dove vengono impiegati i CMS come strumento di pubblicazione flessibile e multi-utente. I CMS sono solitamente orientati alla gestione di contenuti testuali (news, articoli, etc.), immagini, liste di discussione, forum, materiale scaricabile. A volte danno la possibilità di gestire anche più versioni dello stesso sito (ad esempio, HTML o WAP). Inoltre può essere modificata anche la struttura stessa delle pagine in numero ed organizzazione.

I CMS consentono di definire utenti, gruppi e diritti, in modo da poter permettere un distribuzione del lavoro tra più persone. Per esempio, è possibile definire una classe di utenti abilitati esclusivamente all'inserimento delle novità, mentre si può riservare la scrittura degli articoli ad un altro gruppo, e limitare tutti gli altri alla sola consultazione.

Tecnicamente il CMS è un'applicazione lato server, divisa in due parti: il back end, ovvero la sezione di amministrazione, che si occupa di organizzare e supervisionare la produzione del contenuto, e il front end, cioè la sezione dell'applicazione con cui l'utente è a contatto e che usa per realizzare fisicamente modifiche, aggiornamenti ed inserimenti.

## **3. CMS vs Macromedia Dreamweaver**

### 3.1.La scelta

La scelta di accostarsi alla categoria dei CMS è nata dalle condizioni del Dipartimento universitario preso in esame. È necessario infatti precisare che esistono in commercio numerosi programmi per la realizzazione di pagine web: Macromedia Dreamweaver, Microsoft FrontPage, HomeSite, SciTe, Nvu, Kompozer,etc. In particolar modo Macromedia Dreamweaver è uno dei più affermati, ma soprattutto tra i più completi sia per chi si accosta come programmatore più o meno esperto sia come webmaster. Nonostante ciò i vantaggi che tra poco saranno chiariti dei CMS hanno propenso in modo deciso verso di essi.

#### *3.1.1.Vantaggi dei CMS*

- Un CMS permette di costruire e aggiornare un sito dinamico, anche molto grande, senza necessità di scrivere una riga di HTML e senza conoscere linguaggi di programmazione lato server (come PHP) o progettare un apposito database. L'aspetto può essere personalizzato scegliendo un foglio di stile CSS appositamente progettato per un determinato CMS.

- Esistono CMS personalizzati, cioè appositamente progettati per un tipo preciso di contenuti (un'enciclopedia on-line, un blog, un forum etc.) e CMS generici, che tendono ad essere più flessibili per consentire la pubblicazione di diversi tipi di contenuti.
- Un CMS mantiene separato il codice di programmazione, la grafica e i contenuti. Questo significa che su ognuno di questi tre elementi possono intervenire persone diverse con competenze specifiche:
  - il grafico web realizza l'interfaccia grafica del sito;
  - il programmatore informatico configura e personalizza la struttura del sistema integrando il materiale grafico e realizzando eventuali template;
  - l'azienda o l'ente committente provvede a gestire i contenuti.
- Con un CMS viene tenuta distinta l'attività di sviluppo del sito dall'attività di mantenimento del sito. La prima fase è a cura della società informatica che realizza il progetto (il contenitore con tutti gli automatismi necessari). La seconda fase è a cura dell'organizzazione committente che inserisce e aggiorna le informazioni del sito (il contenuto).
- Nei CMS è possibile definire dei modelli grafici di pagina web che possono poi essere richiamati al momento della creazione di nuovi contenuti. Questo permette di avere un varietà grafica all'interno del sito, scegliendo fra alcuni template predefiniti, quello che meglio si adatta ai contenuti da pubblicare.
- L'aggiornamento delle informazioni contenute nel sito viene effettuato tramite browser web. Qualsiasi utente autorizzato può contribuire, in collegamento via Internet, alla definizione dei contenuti. Questo consente a persone dislocate in sedi diverse, anche lontane fra di loro, di partecipare all'aggiornamento del sito.
- Nei CMS sono integrati degli editor visuali che consentono di scrivere i contenuti della pagina o della sezione di pagina con le funzionalità standard di un semplice word processor: inserimento di testi, modifica del font e della dimensione del carattere, inserimento di elenchi puntati, inserimento di foto o immagini, inserimento di tabelle, scelta del colore dello sfondo, scelta della posizione di testi e foto.
- La gestione multilingua del sito, solitamente laboriosa e complessa, viene effettuata in modo semplice utilizzando un CMS.
- Quasi tutti i CMS dispongono di moduli aggiuntivi che possono essere facilmente integrati al sistema base come ad esempio forum, newsletter, sondaggio, e altro. Sono anche interfacciabili database esterni, cataloghi di prodotti, sistemi e-commerce e in generale funzionalità web evolute.



### 3.1.2.Limiti dei CMS

- Un CMS è tanto più efficiente quanto più è specializzato.
- Molti piccoli portali fanno ricorso a CMS distribuiti (scritti da altri e messi a disposizione gratuitamente o a pagamento) di tipo generico; per quanto un CMS possa essere flessibile, un sito basato su questa struttura in genere presenta un aspetto poco personalizzato. I contenuti, inoltre, saranno sempre ancorati a quanto previsto da chi ha progettato il CMS e non alle esigenze di chi pubblica il sito.
- Problemi di gestione possono derivare dal fatto che chi pubblica o gestisce il sito può usare il CMS per intervenire sui contenuti e sull'aspetto, ma generalmente non è in grado di intervenire direttamente (o far intervenire) sulla struttura del CMS stesso; questo è un limite strettamente connesso al vantaggio primario dei CMS: pubblicare un portale senza doverne progettare la struttura o senza possedere le conoscenze tecniche (o le risorse finanziarie) per uno sviluppo personalizzato. Questi problemi sono risolvibili utilizzando software open source: la possibilità di accedere al codice sorgente del prodotto permette di personalizzare il software sulla base delle proprie esigenze a patto di non avere necessità di apportare modifiche al prodotto adottato. Anche in questo caso, vanno messi in conto i costi per lo sviluppo di moduli personalizzati o funzioni particolari a meno di non possedere in proprio o nella propria struttura aziendale le conoscenze tecniche per intervenire nel codice sorgente.
- I portali di una certa importanza non fanno mai ricorso a CMS distribuiti bensì usano programmi e database progettati su misura, ovvero CMS personalizzati e dunque necessariamente specializzati; in questo modo la struttura e la presentazione vengono realizzate tenendo presenti i contenuti che il sito dovrà ospitare e potranno essere modificati in seguito a nuove esigenze.

### 3.1.3.Cos'è Macromedia Dreamweaver

Macromedia Dreamweaver è un programma per la realizzazione di pagine web. Prodotto da Macromedia (dal 2005 società di proprietà di Adobe Systems) è tra i programmi più utilizzati per realizzare siti web. Il programma dà un'interfaccia visuale (per questo viene definito *editor visuale* o WYSIWYG); con esso si possono fare pagine web, scrivendo semplicemente del testo, inserendo immagini e collegamenti ipertestuali, senza dover conoscere il linguaggio HTML. Macromedia Dreamweaver è utile anche per i programmatori che preferiscono scrivere autonomamente il codice; il programma aiuta l'utente dando suggerimenti mentre vengono scritti i tag o attributi (ad

esempio, scrivendo le prime due lettere di un tag o attributo viene proposta automaticamente una lista di tag o attributi che è possibile scegliere, una funzione molto simile all' IntelliSense di Microsoft). Macromedia Dreamweaver presenta anche molte funzioni utili per i webmaster, permettendo di lavorare con codice server-side (PHP, Perl) e con codici Javascript, X/HTML e CSS.

#### *3.1.4. Vantaggi di Macromedia Dreamweaver:*

- molto potente nel complesso;
- facile da usare per un esperto;
- ha un editor di codice molto comodo in quanto supporta diversi linguaggi (HTML, PHP, ASP, Javascript, CSS, etc..) e ha anche una sorta di IntelliSense;
- possibilità di gestire *snippets* (pezzetti) di codice più usato in un archivio e richiamarli all' occorrenza;
- potente client FTP integrato;
- editor visuale WYSIWYG molto avanzato;
- controlli incrociati sul singolo progetto web per rilevare errori nel codice e nei collegamenti;
- strumenti di validazione del codice integrati.

#### *3.1.5. Svantaggi di Macromedia Dreamweaver:*

- programma professionale per creazioni di alto livello, non adatto a non professionisti;
- non sempre l'editor WYSIWYG genera un codice conforme allo standard W3C;
- non molto reattivo in versione per Mac OS X;
- licenza proprietaria e costo elevato.

#### *3.1.6. Conclusioni confronto*

Nonostante Macromedia Dreamweaver sia attualmente uno dei migliori editor web in commercio, la scelta è ricaduta sui CMS per tre importanti motivi. Per prima cosa si è considerato la situazione d'uso del sito in esame. Il sito [www.gest.unipd.it](http://www.gest.unipd.it) viene gestito dai tecnici presenti in Dipartimento, ma ad esso si affacciano molteplici figure: docenti, ricercatori, dottorandi, personale tecnico amministrativo. Quindi è di fondamentale importanza che tutte queste tipologie di utenti si possano accostare allo strumento senza eccessiva difficoltà. La gestione del contenuto del sito deve essere resa disponibile anche a delle persone che non hanno un background tecnico, non sanno come è composta una pagina HTML né come è strutturato un sito Internet. E non

vogliono saperlo. Vogliono solo aggiungere contenuto al sito Web. Per tale ragione, come evidenziato nei punti suddetti, un CMS è senza dubbio più adatto a tale requisito in quanto Macromedia Dreamweaver è più efficiente nelle mani di personale altamente esperto. È necessario ricorrere ad un software che possa creare un framework intorno al sito, in modo che, definita una struttura portante ed un design attraente, le informazioni inserite nel sito vengano strutturate in modo automatico, coerente ed intuitivo. L'obiettivo della proposta è, infatti, la progettazione di un sistema di comunicazione via web, in grado di poter essere facilmente modificato da persone non esperte di programmazione o web development, con cui sia possibile gestire efficacemente una gran mole di informazioni di diverse tipologie.

In secondo luogo si è tenuto conto della questione costo. Macromedia Dreamweaver attualmente non è gratuito, anzi presenta un costo relativamente elevato, mentre esiste un ampio panorama di CMS gratuiti.

Infine è stato analizzato il problema dell'accessibilità. Secondo normative italiane in vigore i siti web, in particolar modo quelli della Pubblica Amministrazione, quale è il Dipartimento di Tecnica e Gestione dei Sistemi Industriali, devono sottostare alla Legge Stanca. Come già detto però una esigua parte di siti mette in pratica quanto standardizzato e quindi è necessario soffermarsi, non tanto sulla Legge Stanca, ma sulla certificazione W3C. Detto ciò dalle considerazioni fatte nei paragrafi precedenti, Macromedia Dreamweaver non sempre è conforme a quanto stilato dal W3C mentre esistono vari CMS conformi e taluni in sviluppo verso le indicazioni della Legge Stanca.

## **4. Accessibilità**

L'accessibilità, in informatica, è la capacità di un dispositivo, di un servizio o di una risorsa d'essere fruibile con facilità da una qualsiasi categoria d'utente.

Il termine è comunemente associato alla possibilità anche per persone con ridotta o impedita capacità sensoriale, motoria, o psichica (ovvero affette da disabilità sia temporanea, sia stabile), di fruire dei sistemi informatici e delle risorse software a disposizione. Il termine ha trovato largo uso anche nel settore di Internet col medesimo significato.

### 4.1. L'accessibilità nel Web

#### *4.1.1. Requisiti fondamentali*

Un sito web accessibile facilita l'accesso ad individui con ogni tipo di disabilità, ma anche ad individui non affetti da patologie. Più nello specifico:

- utilizza un codice semanticamente corretto, logico e validato secondo i parametri del W3C;

- utilizza testi chiari, fluenti e facilmente comprensibili;
- utilizza testo alternativo per ogni tipo di contenuto multimediale;
- sfrutta titoli e link che siano sensati anche al di fuori del loro contesto (evitando, ad esempio, link su locuzioni come "clicca qui");
- ha una disposizione coerente e lineare dei contenuti e dell'interfaccia grafica.

Inoltre dovrebbe essere compatibile col maggior numero di browser e configurazioni software e utilizzare colori standard e ad alto contrasto fra di loro.

L'applicazione corretta dei criteri di accessibilità deve permettere la lettura delle pagine web da parte di software detti screenreader, specifici per ipovedenti o non vedenti.

#### *4.1.2.CMS accessibili*

I criteri di accessibilità hanno importanza anche nella realizzazione dei Sistemi di Gestione dei Contenuti (CMS), ossia di quei sistemi web che permettono la creazione e la pubblicazione di documenti da parte di più autori che, molto spesso, dispongono di una vera e propria struttura editoriale.

In questi casi, è opportuno non solo rendere il sito conforme agli standard, ma anche:

- aiutare gli autori a pubblicare documenti che non compromettano l'accessibilità;
- fornire soluzioni CMS accessibili direttamente agli autori con disabilità.

Esistono già alcuni CMS (sia commerciali che open source) che promettono il rispetto della normativa per l'accessibilità.

## 4.2.Standard e Leggi

Esistono alcuni standard ed alcune linee guida per definire l'accessibilità. Le linee guida internazionalmente più diffuse per quanto concerne il web sono le WCAG, *Web Content Accessibility Guidelines* della WAI, *Web Accessibility Initiative* (sezione del World Wide Web Consortium).

### *4.2.1.In Italia*

In Italia per le nuove realizzazioni e le modifiche apportate dalla Pubblica amministrazione al proprio o ai propri siti web si deve tenere conto (pena nullità dei contratti stipulati) della Legge Stanca -"Disposizioni per favorire l'accesso dei soggetti disabili agli strumenti informatici" (Legge 4 del 9 gennaio 2004, pubblicata nella Gazzetta Ufficiale il 17 gennaio 2004), resa operativa col decreto attuativo di fine 2005. Il medesimo obbligo è in carico, come specificato nell'art. 2 della legge, *agli enti pubblici economici, alle aziende private concessionarie di servizi pubblici, alle aziende municipalizzate regionali, agli enti di assistenza e di riabilitazione pubblici, alle aziende*

*di trasporto e di telecomunicazione a prevalente partecipazione di capitale pubblico e alle aziende appaltatrici di servizi informatici. L'obbligo della applicazione della legge sussiste esclusivamente per i siti pubblici (o di interesse pubblico) mentre, sempre nell'ambito pubblico, le disposizioni di legge non si applicano ai sistemi informatici destinati ad essere fruiti da gruppi di utenti dei quali, per disposizione di legge, non possono fare parte persone disabili.*

#### *4.2.2. Nel resto del mondo*

Negli Stati Uniti è in vigore una legge, "section 508", seguita come linea guida in tutto il mondo. Altre leggi sull'accessibilità sono state varate in varie nazioni. Una lista è consultabile sul sito del W3C.

### 4.3.W3C

#### *4.3.1. Introduzione*

Il World Wide Web Consortium (W3C) non è un organismo di standardizzazione, come invece lo sono l'ISO, ITU-T o la Commissione Elettrotecnica Internazionale (IEC). Il suo compito è invitare i produttori a seguire le proprie Raccomandazioni (direttive realizzate dal W3C). A differenza degli organismi di standardizzazione, il W3C non ha un programma di certificazione e molti dei suoi standard di fatto non definiscono formalmente i livelli di conformità. Di conseguenza molte Raccomandazioni sono implementate solo parzialmente.

#### *4.3.2. Membri*

Attualmente il W3C comprende oltre 400 membri. Tra questi:

- aziende informatiche del calibro di (ordine alfabetico): Adobe, Apple, Cisco Systems, Google, IBM, Intel, Microsoft, Oracle, Siemens Sony e Sun Microsystems;
- compagnie telefoniche come Ericsson, Nokia, NTTDoCoMo;
- società di grandi dimensioni appartenenti ai più svariati settori, ma strategicamente interessate alla crescita del Web: American Express, Agfa-Gevaert N. V., Boeing, ChevronTexaco;
- associazioni come la Mozilla Foundation e The Open Group;
- università e istituzioni per la ricerca.

L'importanza dei suoi membri fa del W3C un organismo di grande autorevolezza.

#### 4.3.3. *Obiettivi del W3C*

Il motto del W3C è "Leading the Web to Its Full Potential".

- Il Web ormai non è più uno strumento per «appassionati», ma è diventato parte integrante della vita comune dell'essere umano. Attualmente esistono diversi tipi di apparecchi (come cellulari e PDA) che accedono ad internet. Ciò è possibile solo grazie ad un «comune linguaggio di comunicazione» (nello specifico protocollo di comunicazione e linguaggio in cui è scritto il file) tra server, PC ed altri dispositivi. Il W3C si occupa di aggiornare e creare queste specifiche.
- Il Web ha un potenziale praticamente illimitato, ed apre nuove strade ai portatori di handicap, anche gravi. Il W3C cerca di studiare i modi per rendere quanto più agevole l'accesso al web da parte di queste persone diversamente abili.
- Il Web è unico perché è libero. Chiunque può creare un documento HTML e metterlo gratuitamente online. Il W3C cerca di evitare che interessi di qualsiasi genere possano porre un freno a questa assoluta libertà.

#### 4.3.4. *Iniziative del W3C*

Considerando nel loro complesso le specifiche, gli strumenti e le attività promozionali sviluppati nell'ambito del W3C si deve sottolineare che essi compongono un disegno complessivo di grande portata e lungimiranza, basato su una lucida consapevolezza dell'importanza che il Web riveste per il mondo attuale.

Molte delle iniziative del Consorzio hanno avuto pieno successo. Di fatto ha determinato vari standard ampiamente adottati, la cui autorevolezza ha dato ordine e organicità alla crescita del Web dagli ultimi anni '90 ad oggi. Occorre dire che alcune Raccomandazioni sono state definite dopo dibattiti molto accesi che hanno visto lo scontro di grandi interessi industriali contrapposti. In questo senso il Consorzio si può considerare un ambiente che favorisce il raggiungimento di compromessi molto utili per la crescita aperta del Web.

Altre iniziative invece non hanno ottenuto le adesioni sperate e molti standard avanzati non si sono imposti.

Il W3C inoltre mette a disposizione online all'indirizzo <http://validator.w3.org> un software per la validazione dei siti web; in ogni report di questa ricerca è possibile consultare il responso del W3C sul sito preso in esame.

Esiste poi un servizio offerto gratuitamente, Bobby. È molto utile, in appoggio a quello ufficiale del W3C, perché riporta la schermata del sito analizzato mettendo in evidenza

gli errori e suggerendo possibili modifiche. Anche in questo caso, nelle schede delle singole analisi è possibile visualizzare il report di Bobby.

#### *4.3.5. La situazione in Italia*

Importanti Istituti di ricerca hanno affrontato in questi anni analisi di vario tipo sui siti della Pubblica Amministrazione italiana. Questi studi dimostrano come l'accessibilità sia diventato un requisito importante per la Pubblica Amministrazione, ma che occorre ancora lavorare molto per una piena e totale soluzione del problema soprattutto in ambito locale dove i ritardi risultano essere maggiori.

## **5. L'usabilità di un sito web**

### 5.1. Introduzione

Il tema dell'usabilità è un aspetto imprescindibile nello sviluppo di un sito web. Il tema dell'accessibilità è cosa ben diversa dal punto di vista concettuale pur presentando aspetti comuni. Secondo la definizione data dalla norma ISO 9241, l'usabilità è il "grado in cui un prodotto può essere usato da particolari utenti per raggiungere certi obiettivi con efficacia, efficienza e soddisfazione in uno specifico contesto d'uso". Quindi per progettare e sviluppare un sito usabile si dovrebbe rispondere alle seguenti fondamentali domande: a cosa serve un determinato sito web? Chi lo userà e cosa si aspetterà di trovarci?

La realizzazione di un efficace sito, nasce dalla sua progettazione. La costruzione di un sito ottimizzato è un insieme di fasi essenziali: accurata scelta delle keyword da inserire, particolare attenzione nella scelta del dominio, definizione della veste grafica in funzione della propria immagine, studio dell'intera struttura del sito, segnalazione sui principali motori di ricerca e un costante aggiornamento.

Ma non è da sottovalutare il discorso dell'usabilità, raccogliere e selezionare contenuti interessanti per i visitatori e renderli facilmente raggiungibili semplificando la navigazione congiuntamente alla scelta delle giuste tecnologie per rendere dinamico un sito, sono passaggi fondamentali per rendere efficace la pubblicità sul Web.

### 5.2. Il parere degli esperti circa l'usabilità

#### *5.2.1. Verso l'usabilità*

Daniele Ziggiotto nell'introduzione del suo corso sull'usabilità afferma: "L'utente tipico mostra, nell'uso del Web, impazienza e desiderio di soddisfazione immediata. Se non viene messo nelle condizioni di interpretarne con facilità il contenuto, l'utente abbandona rapidamente il sito, e difficilmente vi fa ritorno. L'usabilità diventa quindi il primo obiettivo di un sito web e vede nel processo di valutazione il momento

fondamentale della progettazione centrata sull'utente...". Questo significa che per valutare la buona riuscita di un sito web bisogna considerare il contenuto del sito, il servizio offerto, la capacità di fornire facilmente l'informazione che l'utente richiede, o ancora la struttura del sito stesso, che deve rendere agevole la navigazione. Michele Visciola approfondisce quest'aspetto legando il concetto dell'usabilità ai bisogni dell'utente: "Un sito web è usabile quando soddisfa i bisogni informativi dell'utente finale che lo sta visitando e interrogando, fornendogli facilità di accesso e navigabilità e consentendo un adeguato livello di comprensione dei contenuti. Un sito web deve cercare di soddisfare al meglio uno o più bisogni mettendo l'utente in posizione dominante e preferenziale".

Non si può che concludere questa breve rassegna di contributi sul tema dell'usabilità con le parole di Jakob Nielsen, guru dell'usabilità, che sostiene che i siti web debbano essere chiari e coerenti, debbano permettere una navigazione semplice ed efficace, debbano mantenere quello che promettono evitando ogni tipo di ambiguità e di ridondanza del messaggio poiché "l'obiettivo principale della maggioranza dei progetti web dovrebbe essere il rendere semplice per le persone (o per i clienti) lo svolgimento di attività che hanno un qualche fine pratico" (J. Nielsen, *Web Usability*, Apogeo).

### *5.2.2. Il parere contrario*

Franco "Bifo" Berardi nell'articolo "Dissociare il webdesign dall'usabilità" (dall'esplicativo sottotitolo Il rischio che la Rete diventi un supermercato in cui la bellezza e l'intelligenza siano vietate) mette però in guardia da alcuni rischi insiti in questo discorso sottolineando che un'esasperazione di questo concetto possa appiattire la ricerca di nuovi linguaggi espressivi che continuamente la rete offre; egli afferma che sul tema dell'usabilità "si gioca una partita decisiva per il futuro della rete e in generale delle tecnologie di comunicazione. Quel che è in gioco, infatti, è il problema della produzione di significato e delle modalità di interpretazione sociale. L'idea che si afferma attraverso la definizione di criteri di web-usability è quella dell'automatizzazione dei processi di interpretazione e di navigazione. Dietro la semplificazione della navigazione si intravede la trasformazione della rete in una sorta di percorso prestabilito che segue strade precostituite verso destinazioni che poi sono facilmente intuibili: comprare, comprare, comprare. Making things easy (facilitare le cose) è il principio guida per la trasformazione della rete in un sistema di potere economico e politico rigido, automatico, inevitabile. Se riduciamo Internet a un sistema pavloviano di domande prevedibili e di risposte precostituite, la rete diverrà un congegno di produzione e distribuzione di merce e di potere". Bifo conclude questo "manifesto della creatività" con una domanda che chiunque, navigatore o sviluppatore



di siti web, il disabile o il Nielsen di turno, dovrebbe porsi: "Come possiamo allora parlare della bellezza, dell'estetica, della polisemanticità del messaggio che circola attraverso la rete?". In un altro intervento Bifo rincara la dose affermando che: "la rete tende a diventare un ambiente primario di comunicazione, e dunque tende a modellare il linguaggio sociale. Vogliamo ridurre il linguaggio a mera usabilità? Vogliamo progressivamente espungere dalla comunicazione ogni ironia, ogni ambiguità, ogni polisemanticità? Ma quello è il linguaggio del lager, magari un lager che si chiama e-commerce". Maurizio Boscarol condivide in un suo articolo questa lettura e afferma che "semplificare la scrittura vuol dire anche semplificare il pensiero e semplificare il pensiero significa imboccare in modo del tutto inoffensivo (offendere, ossia colpire, stimolare) il nostro lettore".

### *5.2.3. Conclusione dibattito*

La lunga citazione di Bifo dovrebbe chiarire bene i termini di una questione che diventerà sempre più centrale nel dibattito sullo sviluppo del Web. Quale sarà il futuro del Web in ambito usabilità?

Franco-Saverio Caccavella, nella conclusione di un suo articolo mette in guardia sottolineando che "l'usabilità, spaziando dalla grafica alla scrittura, dall'architettura alle scelte tecnologiche, è un concetto che sembra posto lì come un macigno ad indicarci la strada della semplicità e della facilità. Costruire il futuro linguaggio del Web su queste basi così minimaliste può essere una iattura di cui preoccuparci. Il minimo sforzo produce quasi sempre minimi risultati".

## **6. Progettare un buon sito**

### 6.1. Introduzione

Come suddetto, la realizzazione di un sito che conta necessita di precise linee guida. Prima di tutto bisogna sottolineare il seguente fatto: gli utenti del Web non leggono, ma scorrono il testo alla ricerca di frasi o parole che attirino la loro attenzione, questo perché leggere sul monitor rispetto che su carta è molto più difficoltoso.

Scrivere per il Web diventa così un mestiere in parte diverso dallo scrivere su carta, anche se è basato sulle stesse qualità di fondo, sul Web vi sono però ulteriori criteri di scrittura e composizione che vanno seguiti per facilitare la lettura, così come vi sono errori da non fare assolutamente.

L'impatto iniziale è molto importante per aggiudicarsi la fidelizzazione del visitatore. Sono tre gli ambiti d'azione: come disporre i contenuti, quale stile utilizzare, e infine quali espedienti visivi facilitano la lettura su monitor.

### 6.2.I contenuti

Il Web è il regno della concretezza e della concisione. I contenuti devono essere attinenti a ciò che l'utente sta cercando. È indispensabile promuovere esclusivamente quelle parole chiave che rispecchiano gli effettivi contenuti del sito. E' buona norma poi non accomunare troppe argomentazioni ed essere esaurienti in quelle ritenute fondamentali. Chi giudica negativamente, difficilmente lo farà una seconda volta.

Secondo Grice, studioso della comprensione dei testi sarebbe opportuno:

- scrivere la quantità di informazione necessaria: non di più, non di meno;
- scrivere di ciò di cui si hanno prove;
- essere pertinente: dire quello che è rilevante e coerente con l'argomento;
- essere chiaro.

### 6.3.Lo stile

A causa soprattutto delle difficoltà nella lettura, il Web vuole una scrittura pratica, concisa. E' meglio evitare di essere troppo letterari, eleganti, involuti. È bene togliere tutte le parole superflue, è meglio essere oggettivi, concisi e precisi. Si dovrebbe evitare però di rendere il testo freddo e noioso; in una parola, si dovrebbe trasmettere un certo entusiasmo, ma al tempo stesso rimanere professionali.

### 6.4.Gli espedienti visivi

La grafica e' molto importante allo scopo di dare un aspetto curato e professionale. Per poter al meglio trasmettere i contenuti è necessario creare per il visitatore, un habitat gradevole e soprattutto percorsi di navigazione semplici e lineari. Prima di ogni cosa, la completa visualizzazione della pagina deve avvenire entro e non oltre 5/6 secondi, un lasso di tempo superiore potrebbe provocare un abbandono immediato da parte del visitatore. E' statisticamente provato che la pazienza del navigatore è molto precaria durante l'attesa della visualizzazione della pagina, pertanto va bene l'ottima grafica, ma occhio alle dimensioni. L'ottimizzazione delle immagini è dunque fondamentale.

In secondo luogo, visto che la lettura si fonda su due processi, il riconoscimento visivo e l'estrazione del significato, è necessario porre attenzione anche a quest'ultimo punto. Il secondo processo viene ostacolato da una cattiva percezione. Per migliorare la percezione visiva del testo si possono usare molteplici espedienti visivi: spezzare i periodi andando frequentemente a capo, evidenziare le parole chiave, non utilizzare una colonna di testo troppo larga, utilizzare dove possibile elenchi numerati o con puntatore, scegliere colori che assicurino un buon contrasto, utilizzare un carattere tipografico standard, etc.

A questi suggerimenti va aggiunta una considerazione più progettuale. Nelle prime pagine non ci devono assolutamente essere testi lunghi. Solo lanci di notizie, civette, ben organizzate sulla pagina, con buoni titoli e sommari. Non è solo un problema di scrittura, ma di grafica.

Nelle pagine di destinazione, i testi possono (e a volte devono) essere più lunghi. Se un utente è interessato e si sono seguiti i suggerimenti, farà lo sforzo, oppure salverà la pagina e la leggerà con calma, magari dopo averla stampata.

## 6.5.Conclusioni

Contenuti coerenti ed esaurienti, pagine snelle e una semplice navigazione, danno al sito le chiavi vincenti di un ottimo posizionamento.

Queste linee guida non sono affatto semplici da seguire. Spesso è necessario un laborioso processo di riscrittura e di adattamento del testo, soprattutto se lungo. Comunque il tempo passato a curare la progettazione del sito, in particolar modo i contenuti, è sempre tempo ben speso.

## **7.Le Licenze**

### 7.1.Introduzione

Appurata la necessità di utilizzare un CMS, la scelta di quale utilizzare, come si potrà osservare in seguito, è un'operazione lunga e difficoltosa, dato il gran numero di CMS in circolazione. Innanzitutto è necessario chiarire la distinzione tra CMS Open Source e CMS Closed Source. La differenza non risiede soltanto nel prezzo, generalmente vale l'equazione "open source = gratis" anche se le due cose sono assai diverse, ma anche nel livello di supporto che potrà essere necessario in fasi più avanzate del deployment dell'applicazione. Mentre per quanto riguarda un CMS Closed Source, proprietario, il supporto viene tipicamente fornito e garantito dall'azienda produttrice, optando per una soluzione open source esistono due alternative: appoggiarsi alla comunità online, e quindi ai forum dedicati, oppure cercare una ditta che fornisca supporto a pagamento per il CMS scelto. Ovviamente sia l'efficacia del supporto delle community sia il numero di aziende che forniscono servizi di supporto tecnico per un particolare prodotto sono direttamente proporzionali alla diffusione dello stesso: più un prodotto è diffuso, più è possibile trovare persone in grado di aiutare in caso di necessità.

In quest'ottica è necessario utilizzare un CMS molto diffuso, efficiente e affidabile.

## 7.2.L'open source

In informatica, open source (termine inglese che significa sorgente aperta) indica un software rilasciato con un tipo di licenza per la quale il codice sorgente è lasciato alla disponibilità di eventuali sviluppatori, in modo che con la collaborazione (in genere libera e spontanea) il prodotto finale possa raggiungere una complessità maggiore di quanto potrebbe ottenere un singolo gruppo di programmazione. Il termine "Open Source" è stato coniato in sostituzione a free software, al fine di evitare l'equivoco dovuto al doppio significato di free nella lingua inglese, visto che spesso veniva interpretato come "gratuito" invece che come "libero". L'open source ha ovviamente tratto grande beneficio da internet. Alla filosofia del movimento Open Source si ispira il movimento Open Content: in questo caso ad essere liberamente disponibile non è il codice sorgente di un programma ma contenuti editoriali quali testi, immagini, video e musica.

### *7.2.1. Vantaggi dell'open source*

Secondo i sostenitori dell'open source, esso presenta numerosi vantaggi rispetto al software proprietario:

- per la sua natura aperta, è solitamente gratuito (salvo rare eccezioni);
- essendo possibile modificare liberamente il software, è possibile personalizzarlo ed adattarlo alle proprie esigenze;
- il codice sorgente è sottoposto ad una revisione da parte di moltissime persone, pertanto è più difficile che contenga buchi e malfunzionamenti. In ogni caso, è sempre possibile per chiunque tenere un indice pubblico dei problemi, in modo che gli utenti li conoscano;
- se viene scoperto un baco o una falla di sicurezza, la sua correzione di solito è molto rapida;
- essendo il sorgente liberamente consultabile, non è possibile inserire intenzionalmente nel software backdoor, cavalli di Troia o spyware senza che questi vengano prontamente scoperti ed eliminati, come invece è accaduto per alcuni software commerciali;
- non esistendo standard proprietari, le cui specifiche sono normalmente segrete, è molto più facile costruire software interoperabile;
- permettere a chiunque di modificare i sorgenti garantisce che ogni nuova funzionalità o copertura di un baco possa essere proposta da chiunque e immediatamente applicata dagli sviluppatori. Questo permette di avere già a

disposizione un software che rispetta le esigenze di chi ha richiesto le modifiche;

- collaborando con sviluppatori volontari e utilizzando il lavoro della comunità, anche le piccole e medie imprese sono in grado di sviluppare e vendere prodotti di alta qualità, senza dover ampliare il loro organico.

### *7.2.2.Svantaggi dell'open source*

Secondo alcuni il software open source avrebbe delle limitazioni e degli svantaggi rispetto al software proprietario:

- alcune tipologie di software, soprattutto di nicchia, non sarebbero disponibili come software open source;
- lo sviluppo del software open source non segue i tradizionali principi di ingegneria del software; ciò porta a risultati incoerenti e ad una mancanza di uniformità e consistenza; nonostante il codice sorgente sia liberamente disponibile, non tutti sono in grado di apportarvi modifiche, inoltre l'usabilità spesso non è ottimale per un utente inesperto;
- alcuni software moderni hanno raggiunto livelli di complessità tale da richiedere il supporto di un'azienda che impieghi molte persone che lavorano a tempo pieno al progetto;
- mancando un sistema formale di revisione, il software open source sarebbe inadatto ad applicazioni mission-critical;
- anche se spesso è disponibile molta documentazione, a volte (ma qualcuno sostiene che accada spesso) è incompleta o non aggiornata.

### *7.2.3.La nostra situazione*

Un'enorme differenza tra le licenze open source e quelle commerciali risiede nella disponibilità del codice sorgente. L'opportunità di poter modificare radicalmente un software per correggere problemi o aggiungere features è uno dei "cavalli di battaglia" delle licenze open source. Per quanto riguarda il progetto che si sta analizzando, il Dipartimento universitario di Ingegneria a Vicenza, l'obiettivo non è sviluppare il CMS che si deciderà di usare, quindi una licenza open source non è un motivo discriminante nella scelta da compiere, lo è invece il costo del pacchetto software. Il risparmio è una questione chiave nella decisione. È fondamentale avere un software gratuito ma non per forza open source.

## 7.3. Il mondo delle licenze

### *7.3.1. Introduzione*

La licenza in ambito informatico è il contratto che può accompagnare un prodotto software. Tale contratto specifica le modalità con cui l'utente può usare tale prodotto, garantendo dei diritti ed imponendo obblighi. È imposta da chi detiene il copyright sul prodotto software; la sua validità dipende dalla presenza del diritto d'autore, questo perché solo chi detiene il copyright ha il diritto di far rispettare in ogni sede la licenza stessa. In vari casi l'autore può rilasciare un prodotto software con più licenze differenti, lasciando l'utente scegliere il tipo che preferisce; in altri casi l'autore può lasciare all'utente la libertà di scegliere la versione di una licenza che preferisce.

L'accettazione della licenza può avvenire in diversi modi. È molto importante leggere le licenze prima di usare il programma perché se non si rispetta la licenza si compiono delle azioni illegali e decadendo la licenza non si ha più diritto né all'uso del software né a qualsiasi azione di rivalsa contro chi ci ha fornito o chi ha prodotto tale software.

### *7.3.2. Tipi di licenza*

Le licenze utilizzate per la distribuzione dei CMS sono licenze per il software libero: GNU General Public License (GPL), GNU Lesser General Public License (LGPL) e Common Public License (CPL). Sono tutte approvate sia dall'Open Source Initiative (OSI) che dalla Free Software Foundation (FSF). I CMS presi in esame in maniera approfondita nella seguente relazione sono tutti software open source gratuiti. Inoltre sono tutti rilasciati sotto i termini della GPL o della LGPL, tranne uno che viene concesso sotto licenza CPL.

### *7.3.3. La GNU General Public License*

La GNU (General Public License) è una licenza per software libero. Viene spesso indicata con l'acronimo GNU GPL o semplicemente GPL.

La versione 1.0 è stata scritta nel 1989 da Richard Stallman e Eben Moglen, per distribuire i programmi creati dal Progetto GNU. Stallman, programmatore del MIT (Massachusetts Institute of Technology), rifiutatosi di lavorare per una società privata, aveva fondato nel 1985 la Free Software Foundation (FSF), un'organizzazione senza fini di lucro per lo sviluppo e la distribuzione di software libero. In particolare l'obiettivo era lo sviluppo di un sistema operativo completo distribuito con una licenza permissiva, con tutti gli strumenti necessari altrettanto liberi. Si trattava del progetto GNU. Il termine era nato come acronimo ricorsivo con lo scopo di collegarsi e contemporaneamente distinguersi dal sistema allora in uso UNIX, ovvero "GNU's Not UNIX".

*“L’obiettivo principale di GNU era essere software libero. Anche se GNU non avesse avuto alcun vantaggio tecnico su UNIX, avrebbe avuto sia un vantaggio sociale, permettendo agli utenti di cooperare, sia un vantaggio etico, rispettando la loro libertà.”*

Era in arrivo così la GNU General Public License (GPL), il preambolo del cui manifesto comincia con:

*“Le licenze per la maggioranza dei programmi hanno lo scopo di togliere all’utente la libertà di dividerlo e di modificarlo. Al contrario, la GPL è intesa a garantire la libertà di condividere e modificare il free software, al fine di assicurare che i programmi siano “liberi” per tutti i loro utenti.”*

Nel 1991 la versione 1.0 del 1989 viene aggiornata alla versione 2.0. Il 29 giugno 2007 la Free Software Foundation ha rilasciato la versione 3 della licenza. La GNU GPL è basata su una licenza simile usata per le prime versioni di GNU Emacs. Contrapponendosi alle licenze per software proprietario, la GNU GPL permette all'utente libertà di utilizzo, copia, modifica e distribuzione; a partire dalla sua creazione è diventata una delle licenze per software libero più usate.

#### *7.3.4. La GNU Lesser General Public License*

La GNU Lesser General Public License (abbreviata in GNU LGPL o solo LGPL) è una licenza creata dalla Free Software Foundation, studiata come compromesso tra la GNU General Public License e altre licenze non-copyleft come ad esempio le cosiddette Licenza BSD, la Licenza X11 e la Licenza Apache. Il suo scopo è di essere appetibile per aziende e progetti che sviluppano software proprietario, tutelando al tempo stesso la comunità del Software Libero da abusi da parte delle stesse. Il testo originale è disponibile all'URL <http://www.gnu.org/copyleft/lesser.html>.

Stabilisce il copyleft (obbligo di rendere disponibile a ogni acquirente il codice sorgente, incluse tutte le modifiche eventualmente effettuate al prodotto) sul singolo file di codice sorgente, ma non sull'intero software (ciò che avveniva con la GPL). Questo comporta che, per esempio, del software rilasciato sotto licenza LGPL può essere incluso liberamente in un'applicazione licenziata sotto condizioni proprietarie, a patto che le modifiche apportate al codice sorgente del software stesso vengano rese pubbliche; tutti gli altri file dell'applicazione possono essere rilasciati con licenza proprietaria e senza codice sorgente. È utile quando si sviluppano librerie, in modo da rendere compatibili programmi liberi e non. Un tempo era infatti chiamata "GNU Library General Public License".

### 7.3.5. La CPL Common Public License

La CPL (Common Public License) è una licenza per software open source pubblicata da IBM.

Gli scopi dichiarati della CPL sono quelli di incoraggiare e promuovere lo sviluppo open source, pur mantenendo la possibilità di riutilizzare i contenuti sotto un'altra licenza.

La CPL ha alcune condizioni che la rendono simile alla GNU General Public License, ma vi sono alcune differenze chiave. Una differenza chiave è la possibilità di ri-licenziare: si può compilare un *Programma* rilasciato sotto CPL senza effettuare modifiche e rilasciarlo sotto un'altra licenza, anche proprietaria. Un'altra differenza è una clausola relativa ai brevetti: per evitare che *collaboratori (contributors)* malevoli possano introdurre del codice che infrange i loro brevetti per poi potersi rivalere dei diritti (*royalty*), la CPL prevede che ogni *collaboratore* garantisca a tutti i destinatari (*recipients*) del programma una licenza libera da *royalty*. Quest'ultima clausola rende la CPL incompatibile con la GPL, sebbene sia possibile che una futura versione della GPL possa adottare una simile, compatibile, clausola.

Per quanto riguarda il codice sorgente la CPL prevede (come la GPL) che esso sia sempre reso disponibile.

## 8.PHP

Tra i Content Management System, più diffusi e usati in sistemi che richiedono un carico computazionale di un certo livello, il linguaggio di programmazione più adottato è PHP. Al momento sul mercato i sistemi che lo utilizzano sono più maturi, più seguiti e supportati. Molti dei CMS analizzati in questa relazione appartengono a questa categoria, è per tale motivo che è necessario sottolineare le caratteristiche di tale mezzo.

### 8.1.Introduzione

PHP è un linguaggio di scripting interpretato, con licenza open source, originariamente concepito per la realizzazione di pagine web dinamiche. Attualmente è utilizzato principalmente per sviluppare applicazioni web lato server ma può essere usato anche per scrivere script a linea di comando o applicazioni stand-alone con interfaccia grafica.

Il suo nome è un acronimo ricorsivo che sta per PHP: Hypertext Preprocessor (PHP: preprocessore di ipertesti).



## 8.2.Caratteristiche

PHP riprende per molti versi la sintassi del C, come peraltro fanno molti linguaggi moderni, e del Perl. È un linguaggio a tipizzazione debole e dalla versione 5 migliora il supporto al paradigma di programmazione ad oggetti. Certi costrutti derivati dal C, come gli operatori fra bit e la gestione di stringhe come array, permettono in alcuni casi di agire a basso livello; tuttavia è fondamentalemente un linguaggio di alto livello, caratteristica questa rafforzata dalla esistenza delle sue moltissime API, oltre 3000 funzioni del nucleo base.

PHP è in grado di interfacciarsi a innumerevoli database, tra cui MySQL, PostgreSQL, Oracle, Firebird, IBM DB2, Microsoft SQL Server, solo per citarne alcuni, e supporta numerose tecnologie, come XML, SOAP, IMAP, FTP, CORBA. Si integra anche con altri linguaggi/piattaforme quali Java e .NET e si può dire che esista un wrapper per ogni libreria esistente, come CURL, GD, Gettext, GMP, Ming, OpenSSL ed altro.

PHP fornisce un'API specifica per interagire con Apache, nonostante funzioni naturalmente con numerosi server web. È anche ottimamente integrato con il database MySQL, per il quale possiede più di una API. Per questo motivo esiste un'enorme quantità di script e librerie in PHP, disponibili liberamente su Internet. La versione 5, comunque, integra al suo interno un piccolo database embedded, SQLite.

Dispone di un archivio chiamato PEAR che mette a disposizione un framework di librerie riusabili per lo sviluppo di applicazioni PHP e di PECL che raccoglie tutte le estensioni conosciute scritte in C.

## 8.3.Diffusione

La popolarità del linguaggio PHP è in costante crescita grazie alla sua semplicità: nel Giugno 2001, ha superato il milione di siti che lo utilizzano. Nell'ottobre 2002, più del 45% dei server Apache usavano PHP. Nel gennaio 2005 è stato insignito del titolo di "Programming Language of 2004" dal TIOBE Programming Community Index, classifica che valuta la popolarità dei linguaggi di programmazione sulla base di informazioni raccolte dai motori di ricerca.

Nel 2005 la configurazione LAMP (Linux, Apache, MySQL, PHP) ha superato il 50% del totale dei server sulla rete mondiale.



# CAPITOLO 3

## Descrizione della metodologia utilizzata

Nei capitoli precedenti è stato analizzato il problema in senso lato ed è stato appreso lo stato dell'arte, concludendo che per raggiungere gli scopi prefissati, ovvero sviluppare un portale universitario, si necessita di un CMS. Di questa tipologia di software open source ne esiste un quantitativo consistente, bisogna perciò applicare una procedura di *software selection* per individuare quello più adatto alle esigenze del contesto di analisi.

### 1. Il processo di software selection

Il processo di software selection consiste di passaggi allo stesso tempo fondamentali e complicati da formalizzare, in quanto basati su una valutazione a sua volta complicata dalla scarsità e, a volte imprecisione, delle informazioni a disposizione. È quindi evidente come il buon senso e l'esperienza dettino le linee guida al fine di eseguire una selezione il più possibile basata su dati di fatto e, pertanto, il più possibile scientifica.

Detto questo, l'approccio seguito si articola nelle seguenti fasi:

- sondare preliminarmente il mercato alla ricerca di soluzioni potenziali;
- stabilire un livello d'informazione minimo considerando le funzionalità richieste al software;
- creazione long list;
- innalzare il livello informativo scendendo maggiormente nei dettagli dei requisiti (funzionali, non funzionali, legali e tecnici). In supporto alla definizione dei requisiti si utilizzano i casi studio;
- applicare modalità di selezione e *ranking*, utilizzando i requisiti individuati nella fase precedente, per arrivare alla *short list*;
- Si potranno quindi studiare in dettaglio i CMS della short list, per ognuno dei quali si testeranno potenzialità e funzionalità.

Il metodo di creazione di una long list e successivamente di una short list per l'attività di software selection si compone delle sei fasi precedentemente indicate, specificate nei paragrafi successivi.

### 2. Prima fase: ricerca di potenziali soluzioni

La principale fonte da cui sono stati attinti i candidati CMS adatti allo studio in oggetto è stata la rete. In Internet, infatti, sono numerosi i siti e i blog dedicati alla trattazione dei Content Management System. Una lista di alcuni tra i più utilizzati software CMS in uso

è reperibile anche dal sito di Wikipedia, [www.wikipedia.it](http://www.wikipedia.it). La rete è ricca di informazioni al riguardo, anche per i software meno conosciuti; in diversi siti web si possono trovare utili consigli, recensioni, screenshot, documentazione, plug-in per tutti i vari CMS, oltre ovviamente al sito web ufficiale del software. Per i più famosi e utilizzati è anche presente una community completamente in lingua italiana, dove si possono ricevere informazioni e scambiare confronti in modo totalmente gratuito.

Può tornare utile farsi un'idea preliminare di quella che sia l'evoluzione nel tempo di questi software, il loro utilizzo a livello mondiale, un'intuizione di ciò che si definisce popolarità del software. Il sito [www.google.it](http://www.google.it) rende possibile tutto questo mostrando, per un certo numero di software selezionati, la loro popolarità nel tempo, il loro utilizzo nelle varie zone geografiche, le indicazioni di quanto la community del CMS sia attiva per risolvere bug ed errori, e diversi altri indici.

Per il lavoro di software selection in questione è stato di fondamentale importanza il sito web [www.cmsmatrix.com](http://www.cmsmatrix.com). In questo sito è raccolta la quasi totalità dei software CMS in circolazione, che possono essere confrontati gli uni con gli altri su requisiti più o meno importanti. Tutti i requisiti vengono suddivisi in macroaree, tra cui requisiti di sistema, sicurezza, supporto, facilità d'uso, performance, gestione, applicazioni integrate. Ad esempio tra i requisiti di sistema si possono trovare informazioni sui linguaggi di programmazione, sistemi operativi, web server, database, ecc supportati dal CMS esaminato. Le tabelle di comparazione non si limitano a confermare o meno la presenza di un certo requisito nel programma, ma specificano se tale requisito è integrato in forma limitata o in una versione a pagamento nel software.

Oltre all'ausilio offerto da cmsmatrix si vuole ribadire la presenza in molti altri siti web di moltissime informazioni sui CMS in circolazione, che sono tornate utili per la selezione dei software nella long list e per l'individuazione di requisiti importanti che sarebbero potuti passare inosservati.

### **3.Seconda fase: stabilire un livello di informazione minimo**

Analizzando il mercato e le fonti dove attingere possibili CMS, è stato notato che ne esiste un'immensa varietà, e per di più continuamente in crescita. Scovare dei requisiti e sottoporre tutti i CMS disponibili all'analisi è una procedura alquanto dispendiosa sia in termini di tempistica, sia per quanto riguarda l'utilizzo di risorse. Si è quindi preferito delineare un insieme di caratteristiche, detto anche livello informativo minimo, che dovevano appartenere al CMS in modo da apportare una prima scrematura. In accordo con quanto scritto poc'anzi, il livello informativo minimo deve essere tale da permettere una selezione contemporaneamente veloce ed accurata, per quanto possibile.

L'obiettivo è di scartare sin da subito le soluzioni che non presentino neppure il set minimo di funzionalità.

Il livello informativo minimo nel caso in esame consiste in:

- popolarità: sono stati scelti i CMS più popolari e meglio recensiti nella rete;
- costo: sono stati considerati solo CMS gratuiti open source;
- multilingua: i software analizzati dovevano avere un supporto multilingua;
- accessibilità: sono stati selezionati i software che avessero almeno una certificazione W3C.

Per quanto concerne il criterio del costo, questo non è stato un parametro che ha fornito notevole scrematura, fortunatamente, perché la maggior parte dei CMS in circolazione sono a licenza gratuita e open source. È stato invece di grande aiuto il criterio della popolarità: in diversi siti, tra cui quello di google precedentemente nominato, sono presenti indici di popolarità, di attività della community e di utilizzo; tutto questo ha permesso una scrematura iniziale che ha evitato inutili perdite di tempo nello studio di software poco utilizzati (quindi con una probabilità maggiore di presenza di errori). Se un CMS è poco utilizzato, è un ragionamento che vale per il caso di studio in questione ma in generale per ogni software di tipo open source, aumenta vertiginosamente il rischio che esso contenga dei errori, dei bug non corretti; viceversa più il software è utilizzato e maggiore è la probabilità che la comunità di sviluppo sia attiva e corregga in un lasso di tempo decisamente ragionevole tutti gli errori denunciati dagli utenti che usufruiscono del prodotto. È per questo che quello della popolarità è stato il criterio a cui è stato attribuito maggior peso.

#### **4.Terza fase: creazione della long list**

Seguendo quanto individuato nella fase precedente, si è passati alla stesura della long list. Nel processo di software selection la long list usualmente è composta da un numero di elementi che va da quindici ad una trentina al massimo. Si è ritenuto opportuno limitare la lista ad un insieme di diciotto software, un numero sufficiente per cogliere le varie tipologie risolutive disponibili sul mercato, ma non troppo elevato per smarrirsi nella ricerca del software idoneo agli scopi dello studio.

## **5.Quarta fase: casi di studio e requisiti**

La quarta fase consiste nella determinazione dei requisiti da utilizzare per la selezione vera e propria del software. Per meglio svolgere questa fase si sono precedentemente individuati alcuni casi d'uso, nella fattispecie per il portale universitario. Tale analisi è utile in questo contesto perché per ottenere ciò che si vuole bisogna prima capire cosa si vuole.

### 5.1.Casi d'uso

I casi studio si possono identificare informalmente come storie scritte, ampiamente utilizzate per scoprire e registrare i requisiti. Essi influenzano molti aspetti di un progetto. Più specificatamente i casi studio sono documenti di testo, non diagrammi, e la modellazione dei casi d'uso è innanzitutto un atto di scrittura di testi, non di disegno di diagrammi. Tali strumenti sono il modo più semplice e familiare per apprendere il contesto di lavoro sia dalla visuale del progettista, sia dal punto di vista del cliente, al fine che questi possano contribuire nella fase di definizione e revisione dei requisiti in modo tale da diminuire il rischio di fallire. Un altro valore dei casi d'uso è che mettono in risalto gli obiettivi e il punto di vista dell'utente, i casi d'uso costituiscono una risposta alle domande "chi utilizza il sistema?", "quali sono gli scenari tipici?", "quali gli obiettivi?".

A tal punto è utile esporre alcune definizioni: un attore è qualcosa o qualcuno dotato di comportamento, come una persona, un sistema informatico o un'organizzazione; uno scenario è una sequenza di azioni e interazioni tra il sistema e alcuni attori. Uno scenario descrive una particolare storia nell'uso del sistema, o un percorso attraverso il caso d'uso. Più dettagliatamente poi si possono identificare tre tipi di attore:

- attore primario: raggiunge degli obiettivi utente utilizzando il sistema sotto analisi. Nella fattispecie sono attori primari gli studenti, i professori, il personale tecnico amministrativo, etc.;
- attore di supporto: offre un servizio al sistema in analisi. I sistemisti del dipartimento sono ad esempio attori di supporto;
- attore fuori scena: ha un interesse nel comportamento del caso d'uso, per esempio lo sono enti governativi o gli amministratori del portale stesso.

Template standard per i casi studio:

<b>Sezione del caso d'uso</b>	<b>Commento</b>
<b>Nome del caso d'uso</b>	Inizia con un verbo
<b>Portata</b>	Il sistema che si sta progettando
<b>Livello</b>	Obiettivo utente o sottofunzione
<b>Attore primario</b>	Chiede al sistema di fornirgli un servizio
<b>Parti interessate e interessi</b>	A chi interessa questo caso d'uso e che cosa desidera
<b>Pre-condizioni</b>	Che cosa deve essere vero all'inizio e vale la pena di dire al lettore
<b>Garanzie di successo</b>	Che cosa deve essere vero se il caso d'uso viene completato con successo, e vale la pena dire al lettore.
<b>Scenario principale di successo</b>	Uno scenario comune di attraversamento del caso d'uso, di successo e incondizionato.
<b>Estensioni</b>	Scenari alternativi, di successo e di fallimento.
<b>Requisiti speciali</b>	Requisiti non funzionali correlati.
<b>Elenco della varianti tecnologiche dei dati</b>	Varianti nei metodi I/O e nel formato dei dati.
<b>Frequenza di ripetizione</b>	Frequenza prevista di esecuzione del caso d'uso.
<b>Varie</b>	Altri aspetti, come per esempio i problemi aperti.

### 5.2. Analisi dei requisiti

La fase più critica in un processo di software selection è costituita dalla raccolta e analisi dei requisiti ed ha lo scopo di stabilire *che cosa* il sistema in questione deve fare. L'analisi dei requisiti parte normalmente dalle esigenze degli utenti, venendo poi integrata da quelle che sono le funzionalità disponibili dei software CMS. Questo è un passaggio di elevata criticità: talvolta è alquanto complicato tradurre telematicamente il linguaggio naturale degli utenti. Inoltre tipicamente l'attività di analisi dei requisiti viene accompagnata da un'altra di analisi dei costi e benefici, che fornisce un giudizio sulla fattibilità del progetto.

### 5.2.1. Difficoltà nell'analisi

Individuare i requisiti appropriati e che permettono una selezione ottimale, come già detto, è una fase alquanto delicata perché un eventuale errore in tale contesto potrebbe implicare la necessità di riprendere il lavoro di selezione dagli albori. Stilare una lista di requisiti, non è possibile a priori, ma è un processo ricorsivo, in quanto spesso durante lo svolgimento del software selection ci si ritrova a modificare e rivedere i requisiti. Infatti sovente non è possibile comprendere fin dall'inizio tutte le caratteristiche che deve avere un software, ma strada facendo ci si imbatte in problematiche che fanno emergere aspetti che l'applicazione dovrebbe possedere.

Nonostante le diversificate precauzioni che si prendono per evitare di commettere errori in tal fase solo il 26% dei progetti di software selection e di sviluppo software terminano con successo. Tale dato esprime quindi che il 74% dei progetti falliscono, tuttavia in realtà neanche quel 26% è totalmente esente da errori. Di questo quantitativo di insuccessi il 56% è dovuta ad una scelta inopportuna dei requisiti.

Le problematiche che i requisiti possono presentare sono le più svariate, si possono comunque individuare dei macro difetti che spesso alcuni dei parametri di valutazione possono presentare:

- ambiguità: requisiti interpretabili in modo differente:
  - *Esempio 1*: specificare un tempo senza fornire il riferimento al fuso orario (in un applicazione che gestisce chiamate intercontinentali);
  - *Esempio 2*: significato di "appropriato visualizzatore":
    - *interpretazione utente*: visualizzatore specifico per ogni tipo di documento;
    - *interpretazione sviluppatore*: generico visualizzatore di testo che mostri il contenuto del documento;
- incompletezza: i requisiti non includono la descrizione di tutte le caratteristiche richieste;
- inconsistenza: conflitti o contraddizioni nella descrizione delle caratteristiche del sistema.

Alcuni dei motivi che possono indurre a compiere errori nella stesura dei requisiti sono:

- assenza di linguaggio comune tra gli attori;
- requisiti spesso poco chiari;
- impossibilità di considerare tutti i requisiti e di produrre un lavoro completo.

Errare nell'individuazione dei requisiti oltre a comportare problematiche organizzative e rendere il lavoro meno lineare, hanno un ingente peso economico. Infatti il costo degli



errori fatti nei requisiti è molto alto, per cui la validazione è molto importante. Mettere a posto un errore nei requisiti dopo aver distribuito il sistema può costare fino a 100 volte di più il costo di mettere a posto un errore in fase di implementazione.

Quindi una volta individuati i requisiti vanno verificati e analizzati molto attentamente, tale fase è detta verificabilità dei requisiti. I requisiti vanno scritti in modo che possano essere oggettivamente verificati nel prodotto finale.

Esempio: *“Il sistema dovrebbe essere facile da usare da parte di operatori esperti ed organizzato in modo tale che gli errori degli utenti siano minimizzati”*. Questo requisito è vago: che significa “minimizzare gli errori”? Occorre quantificare il tasso degli errori. – *“Gli operatori esperti dovrebbero poter controllare le funzioni di sistema dopo due ore di formazione. Dopo tale formazione, il numero medio di errori degli operatori esperti non dovrebbe superare i due al giorno”*

Durante la fase di verifica dei requisiti ci pongono alcune domande che corrispondono ad altrettanti aspetti che devono avere i requisiti.

- Validità: Il sistema fornisce le funzioni che soddisfano i bisogni del cliente?
- Consistenza: Ci sono conflitti tra i requisiti?
- Completezza: Sono incluse tutte le funzioni richieste dal cliente?
- Realisticità: I requisiti sono soddisfacibili dato il budget disponibile e la tecnologia corrente?

Oltre alla verificabilità dei requisiti, solitamente si svolge anche una revisioni di questi. Le revisioni dovrebbero essere fatte regolarmente durante il periodo di definizione dei requisiti. Alle revisioni partecipano sia persone del cliente che del contraente e possono essere formali (strutturate) o informali.

Durante le revisioni si svolgono i seguenti controlli:

- Verificabilità: Il requisito è realisticamente controllabile?
- Comprensibilità: Il requisito si capisce correttamente?
- Tracciabilità: L'origine del requisito è definita correttamente?
- Adattabilità: Il requisito può essere modificato senza influenzare pesantemente gli altri requisiti?

### *5.2.2. Classificazione dei requisiti*

A tal punto è opportuno però comprendere esattamente cosa sia un requisito e quali tipologie ne esistono riferendosi alla selezione di un software.

Occorre anzitutto capire chi siano gli utenti e quali siano le categorie di interesse ed i bisogni degli stessi; è dopo questa fase che si può stilare una prima versione dei

requisiti. Un requisito è una capacità o una condizione a cui il sistema deve essere conforme: una capacità che il sistema deve possedere (come ad esempio essere capace di gestire certe informazioni o di fornire certe funzionalità), una condizione che deve essere verificata affinché sia possibile considerare risolto il problema. Detto in altro modo un requisito è una capacità del software richiesta dall'utente per risolvere un problema che consente di raggiungere un obiettivo, ovvero una capacità del software che deve essere soddisfatta o posseduta da un sistema o una componente di sistema per soddisfare un contratto, uno standard o una specifica.

È da notare che non tutti i requisiti hanno la stessa importanza, per cui bisognerebbe definire delle classi di priorità: ci sono dei requisiti obbligatori, dei requisiti desiderabili ed altri opzionali. Una prima suddivisione dei requisiti, secondo la classificazione *FURPS+*, consiste nel distinguere i requisiti tra funzionali e non funzionali; in una buona specifica ogni requisito deve poter essere verificato dal gruppo di test (altrimenti non è un requisito). I requisiti funzionali esprimono le funzionalità che il sistema deve fornire, mentre tutti gli altri requisiti vengono detti non funzionali, e non riguardano le specifiche funzioni del sistema, quanto piuttosto le proprietà (qualità) del sistema nel suo complesso.

Il modello *FURPS+* classifica i requisiti come segue:

- funzionali: caratteristiche funzionali, capacità;
- usabilità: fattori umani, documentazione;
- affidabilità: gestione dei fallimenti, ripristino;
- prestazioni: tempo di risposta, throughput, disponibilità;
- sostenibilità (supportabilità): adattabilità, manutenibilità.

Il + nella sigla *FURPS+* indica requisiti secondari:

- di implementazione;
- di interfaccia;
- operativi;
- fisici;
- legali.

### *5.2.3. Requisiti funzionali*

I *requisiti funzionali* esprimono un'azione che il sistema dovrebbe eseguire, definiscono sia lo stimolo che la risposta, quindi sia l'input che l'output, e vengono descritti tramite i casi d'uso; descrivono le interazioni tra il sistema e il suo ambiente indipendentemente dalla sua implementazione (l'ambiente include l'utente e ogni altro sistema esterno). Un requisito funzionale è per esempio la possibilità di implementare una granularità dei

privilegi all'interno del proprio CMS. Sarebbe opportuno che i requisiti funzionali siano completi, cioè indichino tutti i servizi richiesti dagli utenti, e coerenti, cioè non debbano avere definizioni contraddittorie. In realtà, come già affermato precedentemente, è molto difficile ottenere requisiti completi, per questo spesso si ricorre ad un processo iterativo per la definizione degli stessi.

Il tempo impiegato per l'analisi dei requisiti da parte del gruppo di lavoro è stato considerevole vista la criticità e l'importanza di tale passaggio; dopo un periodo di studio e collaborazione con il personale tecnico del dipartimento si è giunti alla determinazione di una serie di requisiti funzionali per il software CMS, suddivisi per capacità (applicazioni integrate) e sicurezza. I requisiti di capacità sono stati determinati cercando di individuare quegli strumenti, quelle applicazioni integrate, considerate indispensabili per un software CMS che debba realizzare un portale universitario. I requisiti funzionali che riguardano la sicurezza invece sono stati selezionati di modo che il sistema possa essere ripristinato in qualsiasi momento qualora si verifichi o si commetta un errore, con l'obiettivo di sapere sempre quale utente esegue le operazioni nell'amministrazione e di modo che ogni utente possa agire entro determinati confini all'interno del portale.

Tra i requisiti funzionali quello che necessita di maggiore attenzione è il motore di ricerca, quale applicazione integrata del software. Sarebbe opportuno che il CMS metta a disposizione un buon strumento di search engine per le ricerche all'interno delle pagine web, di modo che per trovare le informazioni necessarie non si debba spendere troppo tempo a visitare le pagine web.

#### *5.2.4. Requisiti non funzionali*

La complessità di un sistema software è determinata in parte dalle sue funzionalità e in parte da dei vincoli globali che incidono sullo sviluppo del sistema; ad esempio, fattori come i costi, prestazioni, robustezza, portabilità, alle volte gravano pesantemente sulla bontà del prodotto finale: trascurarli può portare a sviluppare un prodotto inutile. Tali vincoli sono spesso chiamati *requisiti non funzionali* e giocano un ruolo centrale durante tutto il processo di sviluppo. Essi descrivono gli aspetti del sistema che non sono direttamente legati al comportamento (funzionalità) del sistema stesso; includono una grande varietà di richieste che si riferiscono a diversi aspetti, dall'usabilità alle performance. Non possono essere descritti attraverso i casi d'uso e vengono suddivisi in varie classi per aumentarne la comprensione e la tracciabilità. Riassumendo in una frase si può dire che i requisiti non-funzionali descrivono vincoli sui servizi offerti dal sistema, e sullo stesso processo di sviluppo. Al fine di selezionare un software CMS

adatto alla creazione di un portale universitario sono stati suddivisi i requisiti non funzionali secondo usabilità, affidabilità, sostenibilità e implementazione.

I requisiti di usabilità sono stati individuati per indicare bontà di apprendimento dell'utente, ovvero quanto facilmente un utente impara ad usare il sistema e a capire il suo funzionamento. Vengono a loro volta suddivisi in requisiti che riguardano i fattori umani e altri che si riferiscono ad assistenza e documentazione. Per i primi è stata presa in considerazione la sola certificazione W3C, di cui si discuterà nel capitolo successivo, mentre per i secondi si è andati alla ricerca di requisiti che potessero fornire un indice di bontà sul supporto e sull'assistenza disponibili per un dato software. I requisiti di affidabilità sono utili per indicare la capacità del sistema o di una componente del sistema a fornire una determinata funzione richiesta sotto certe condizioni e per un definito periodo di tempo. Quelli di sostenibilità vengono individuati per rendere semplici le modifiche dopo il deployment, e si dividono in adattabilità, manutenibilità, internazionalizzazione.

I requisiti di configurabilità per un software che deve costruire e gestire un portale universitario sono stati rilevati in configurazioni di setup iniziale, skin e template. Si chiarisce che per configurabilità si intende la capacità del software di essere impostato nel suo comportamento, senza richiederne modifiche al codice sorgente. Il concetto di configurabilità è diretta conseguenza del principio di separazione, ovvero separare il meccanismo dalla politica, e dal principio DRY (don't repeat yourself), ovvero ciascuna informazione deve essere dichiarata una sola volta nel software.

Per requisiti di implementazione si intendono invece tutti quei vincoli sull'implementazione del sistema, incluso l'uso di tool specifici, linguaggi di programmazione o piattaforme hardware. Per il caso di studio in questione vengono differenziati fra limitazione sulle risorse, linguaggi e strumenti, requisiti di interfaccia, requisiti operativi, requisiti fisici.

## **6.Quinta e sesta fase: dalla long alla short list**

Per passare dalla long alla short list si è scelto di dare un diverso peso ai vari requisiti, seguendo una scala da 1 a 4, a seconda dell'importanza che la funzionalità assumeva nel progetto. Si sono poi analizzati i vari software singolarmente, e per ogni requisito si è dato un punteggio, se la funzionalità era presente in toto, il punteggio era uguale a 4, parzialmente presente, il punteggio corrispondente era 3, e se il requisito era assente il punteggio decretato era 0. Si sono moltiplicati poi i vari pesi dei requisiti con il punteggio associato agli stessi per lo specifico software, la somma dei vari prodotti eseguiti corrisponde al punteggio totale raggiunto dal CMS. Si può a tal punto stabilire

una classifica in ordine decrescente, dalla quale si sono estratti i componenti della short list. Nella fattispecie sono stati considerati i primi sei software.

Il passo successivo è stato svolto da diverse individualità, ognuno si è occupato di analizzare potenzialità e funzionalità di un software della short list. I vari CMS sono stati installati, e testati provando a sviluppare il portale universitario. Ogni sviluppatore del portale così ha dovuto avvicinarsi alla community del CMS da lui studiato, riscontrando quali siano le difficoltà che esso poteva far affrontare ed si è dovuto quindi cercare delle strade risolutive. In tale fase del lavoro, si è preferito tralasciare i contenuti concentrandosi sul far sì che l'applicazione web fosse il più possibile navigabile e fosse altrettanto semplice reperire delle informazioni. Per rendere i lavori più confrontabili si è scelto di partire dalla medesima struttura per l'applicazione web. Più specificatamente si è stilata una site map, alla quale tutti i lavori dovevano attenersi. Infatti il fine dello studio non era essere una gara competitiva che doveva decretare chi fosse stato a sviluppare il portale migliore, ma, come già specificato, il fine dello studio è decretare quale sia il CMS più idoneo per creare un portale universitario. La trattazione scritta di questa parte dell'analisi è ritrovabile sui rispettivi elaborati.



# CAPITOLO 4

## Il processo di software selection

In questo capitolo verranno descritti i casi d'uso individuati per il portale universitario, nonché i requisiti funzionali e non funzionali assunti come metro di confronto delle piattaforme software. Saranno poi presentate le soluzioni individuate come candidate per essere inserite nella long list. Per ciascuna di esse verrà data una breve descrizione e, infine, verranno complessivamente valutate attraverso una tabella illustrativa delle loro funzionalità, al fine di decidere quali saranno inserite nella short list.

### 1.Casi d'uso

Vengono riportati in seguito alcuni casi d'uso ideati per il portale. Gli scenari e gli attori per un'applicazione web, in realtà, sono i più svariati e le motivazioni o gli scopi con cui essi si apprestano ad utilizzarla possono essere molti e alquanto diversificati. Considerando quanto detto si è quindi pensato ad alcuni esempi, che non ricoprono l'intera panoramica di utilizzo del portale, ma possono essere comunque utili a comprendere le necessità che un portale deve soddisfare:

<b>Caso d'uso UC1</b>	Utilizzo portale universitario da un utente interno allo stesso.
<b>Portata</b>	Applicazione web per servizi per studenti, professori e personale tecnico appartenente alla facoltà.
<b>Livello</b>	Obiettivo utente.
<b>Attore primario</b>	Studenti, professore e personale tecnico-amministrativo.
<b>Parti interessate e interessi</b>	<ul style="list-style-type: none"><li>• <u>Studente</u>: vuole reperire informazioni inerenti al suo corso di studio e riguardanti attività extra (ad esempio borse di studio, etc.)</li><li>• <u>Professore</u>: inserire avvisi per gli studenti e accedere alla proprio spazio di e-learning.</li><li>• <u>Personale tecnico amministrativo</u>: inserire modulistica, avvisi e scadenze sia per professori che per studenti.</li></ul>
<b>Pre-condizioni</b>	Gli utenti se vogliono inserire o modificare materiale nel portale devono essere identificati e autenticati.

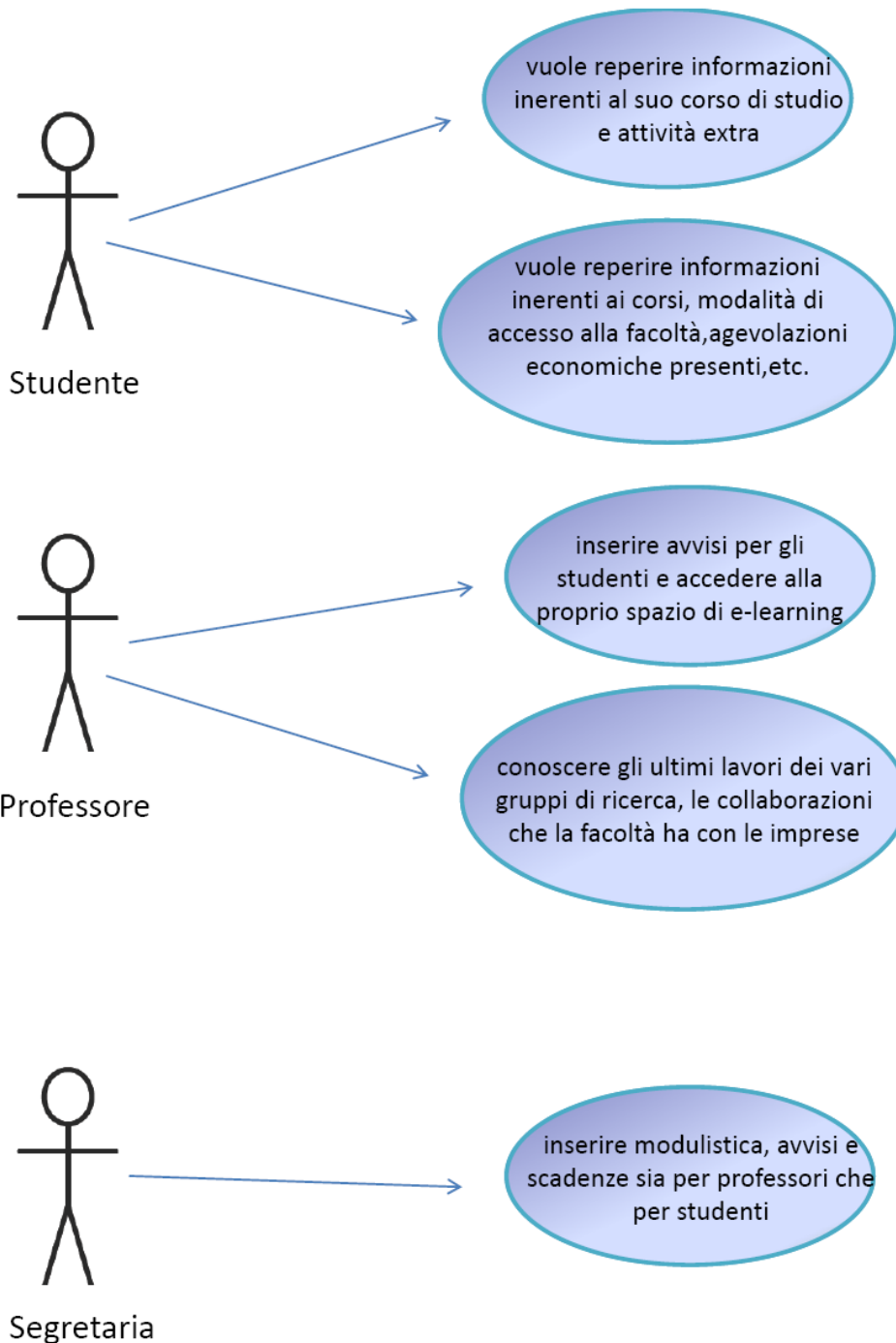
<b>Garanzie di successo</b>	Il successo del caso d'uso non consiste nella sola prosecuzione dell'obiettivo ma anche nelle modalità con cui essa avviene, in quanto queste devono essere agevoli e i percorsi per raggiungere le informazioni non troppo articolati e complessi.
<b>Scenario principale di successo</b>	<ul style="list-style-type: none"> <li>• Lo studente vuole controllare le aule in cui si terranno le lezioni del giorno successivo, e conoscere la loro ubicazione esatta.</li> <li>• Si collega al sito della facoltà, va sul link dell'orario delle lezioni, nel quale visualizza il nominativo dell'aula.</li> <li>• Utilizzando il link, mappe degli edifici è poi possibile conoscere l'indirizzo e il percorso esatto per raggiungere l'aula.</li> </ul> <ol style="list-style-type: none"> <li>1. La segretaria vuole avvisare che la chiusura per le immatricolazioni sarà il giovedì venturo.</li> <li>2. Si collega al sito della facoltà, si deve autenticare.</li> <li>3. Utilizzando il link inserisci nuova news, può inserire l'avviso, questo collegamento lo possono utilizzare solo il personale amministrativo e i professori.</li> </ol>
<b>Frequenza di ripetizione</b>	Frequenza quasi ininterrotta, con picchi di utilizzo durante il giorno e distensioni di accessi al portale nella notte.

<b>Caso d'uso UC2</b>	Utilizzo portale universitario da un utente esterno allo stesso.
<b>Portata</b>	Applicazione web per servizi per futuri studenti e professori che vogliono reperire informazioni su corsi, gruppi di ricerca, etc. della facoltà.
<b>Livello</b>	Obiettivo utente.
<b>Attore primario</b>	Studenti, professore e utenti esterni in generale.



<p><b>Parti interessate e interessi</b></p>	<ul style="list-style-type: none"> <li>• <u>Studente futuro</u>: vuole reperire informazione inerenti ai corsi, modalità di accesso alla facoltà, agevolazioni economiche presenti, etc.</li> <li>• <u>Professore</u>: conoscere gli ultimi lavori dei vari gruppi di ricerca, le collaborazioni che la facoltà ha con le imprese.</li> </ul>
<p><b>Garanzie di successo</b></p>	<p>L'utente deve essere invogliato nel continuare la navigazione nel sito, ciò è possibile se la grafica è piacevole e le informazioni da lui ricercate facilmente individuabili.</p>
<p><b>Scenario principale di successo</b></p>	<ol style="list-style-type: none"> <li>1. Uno studente di un'altra facoltà vuole frequentare i corsi dalla laurea magistrale presso il DTG, quindi vuole conoscere i parametri di ammissione.</li> <li>2. Entra nel portale.</li> <li>3. Utilizzando il link accesso alla facoltà, scarica il file contenente tutte le normative da lui cercate.</li> </ol>
<p><b>Estensioni</b></p>	<ul style="list-style-type: none"> <li>• Un professore inglese vuole reperire informazioni sugli'ultimi lavori del gruppo di ricerca economico.</li> <li>• Accede al portale.</li> <li>• Seleziona la lingua inglese</li> <li>• Sul link gruppi di ricerca, si collega a quello che gli interessa.</li> <li>• Cerca informazioni sugli'elaborati del gruppo.</li> <li>• Nota che il link del gruppo di ricerca non presenta delle pagine tradotte.</li> <li>• Segnale all'amministratore il suo dissenso e la sua volontà di reperire talune informazioni in inglese.</li> </ul>
<p><b>Frequenza di ripetizione</b></p>	<p>Frequenza imprevedibile a priori. Si possono alternare momenti in cui l'utenza è considerevole e altri in cui gli accessi al portale sono in un quantitativo esiguo.</p>

Di seguito viene riportata la rappresentazione UML dei casi d'uso, che consente di illustrare i nomi e gli attori, nonché le relazioni tra gli stessi.



## 2.Requisiti

### 2.1.Requisiti funzionali: capacità

Questi requisiti sono stati determinati cercando di individuare le applicazioni e gli strumenti necessari in un software CMS che debba essere utilizzato per la realizzazione di un portale universitario. Si compongono di:

- **Calendario e Gestione Eventi:** indica la presenza di un'applicazione che possa gestire appuntamenti ed eventi, dando la possibilità agli utenti di tenere traccia delle attività di loro interesse;
- **Content Scheduling:** è l'applicazione che permette di rendere visibili a partire da un certo istante i contenuti precedentemente creati o di eliminarli al raggiungimento della data di scadenza desiderata;
- **Content Syndication** (Supporto RSS): dà la possibilità di diffondere notizie tramite sistema RSS per rendere più rapida la loro consultazione agli utenti.
- **Database Reports:** indica la possibilità della piattaforma di generare report dal database. Questa caratteristica può essere sfruttata, ad esempio, nella visualizzazione delle prenotazioni di aule e laboratori;
- **E-mail Form:** prevede la possibilità di creare una pagina "Contattaci" per l'invio di e-mail da browser tramite apposito form;
- **FAQ management:** è un'applicazione che permette la raccolta e l'organizzazione delle risposte alle domande poste più frequentemente;
- **File Distribution:** è il sistema di distribuzione di file e documenti con integrata la gestione dei privilegi per permettere o meno agli utenti di ricevere/scaricare tali files;
- **Gestione Photogallery e Videogallery:** consente di integrare un sistema di hosting e visualizzazione di immagini e video, che permetta anche la creazione automatica di anteprime e slide show dei file caricati;
- **Mass Upload:** è un sistema che gestisce l'upload di file e immagini centralizzato per poter usufruire eventualmente più volte delle stesse risorse nella creazione dei contenuti senza creare duplicati;
- **Motore di Ricerca:** integra un sistema di indicizzazione e ricerca dei contenuti tramite parole chiave;
- **MyPage:** dà la possibilità di mettere a disposizione di ciascun utente una dashboard personalizzabile in cui sono sintetizzate informazioni personali e altri contenuti di interesse;
- **Newsletter:** è il sistema che permette agli utenti di iscriversi/disiscriversi a liste di distribuzione di e-mail informative e di notifica;

- **Site Map:** è un sistema di generazione automatica di un diagramma ad albero della strutturazione del portale per permettere un miglior orientamento dell'utente nella navigazione;
- **Web-Editor (WYSIWYG):** consente la creazione e l'inserimento dei contenuti attraverso un text-editor che permetta l'impaginazione e la formattazione del testo senza ricorrere a linguaggi di programmazione come HTML, CSS e XML;

## 2.2.Requisiti funzionali: sicurezza

I requisiti funzionali che riguardano la sicurezza sono stati selezionati di modo da avere a disposizione tutti gli strumenti necessari per evitare l'intrusione di utenti malintenzionati, di definire i confini di operatività degli utenti accreditati e di monitorare le loro azioni. Tali requisiti vengono classificati in:

- **Audit Trail** (tracciabilità delle operazioni) : è un meccanismo di controllo di un sistema che permette, per i dati imputati e successivamente processati dal sistema, la tracciabilità a ritroso al dato originale. Permette al sistema di tenere traccia di chi ha eseguito degli aggiornamenti, delle aggiunte, delle cancellazioni;
- **Captcha:** funzione che fornisce la possibilità di eseguire un test composto da una o più domande e risposte per determinare se l'utente sia un umano (e non un computer o, più precisamente, un bot). Un test captcha tipicamente utilizzato è quello in cui si richiede ad un utente di scrivere quali siano le lettere o numeri presenti in una sequenza di lettere o numeri che appaiono distorti o offuscati sullo schermo;



- **Login History:** tale funzione permette al sistema di tenere traccia degli utenti che hanno avuto accesso, e della data il cui è avvenuto il login. Vengono spesso memorizzate informazioni come il browser da cui l'utente si collega, l'indirizzo IP, i collegamenti falliti, ecc;
- **Privilegi granulari:** possibilità di diversificare i privilegi degli utenti. La granularità nella scelta dei privilegi è necessaria per definire con precisione cosa ogni utente o applicazione può fare. Una bassa granularità indica che gli utenti hanno tutti gli stessi privilegi, quindi per esempio non è possibile assegnare ad ogni persona esattamente la risorsa che deve aggiornare.

- **Problem Notification:** questa funzione fornisce al sistema un meccanismo per avvisare gli amministratori (tramite e-mail, messaggi istantanei, ecc) quando viene rilevato un problema;
- **Session management:** il sistema fornisce agli amministratori delle risorse che permettono di verificare informazioni sugli utenti loggati in quel momento, su cosa stiano facendo e consentono agli amministratori di scollegare gli utenti;
- **Verifica indirizzo e-mail:** funzione che permette di inviare all'utente un codice di attivazione alla mail indicata con l'obiettivo di verificare se tale indirizzo immesso corrisponde ad un indirizzo valido.

### 2.3.Requisiti non funzionali: usabilità

Tali requisiti sono stati individuati per indicare bontà di apprendimento dell'utente, ovvero quanto facilmente un utente impara ad usare il sistema e a capire il suo funzionamento. Vengono a loro volta suddivisi in requisiti che riguardano i fattori umani e altri che si riferiscono ad assistenza e documentazione.

#### *2.3.1.Usabilità: fattori umani*

Sotto questa categoria è stato individuato un solo requisito di accessibilità, caratterizzato da criticità alta.

- **Accessibilità** (Certificazione W3C): Un sito certificato W3C rispetta gli standard proposti dal World Wide Web Consortium. Un sito realizzato secondo tali principi si considera utile allo sviluppo del web orientato all'interoperabilità e all'accessibilità. Alcuni tra i vantaggi di tale certificazione sono: accessibilità ai contenuti del sito anche da utenti disabili (accessibilità WAI di primo livello, livello A, utile per permettere a persone disabili ad accedere in maniera chiara e semplice ai contenuti del sito), usabilità cross browser (possibilità di visualizzare il sito in maniera corretta con l'ausilio del software preferito), maggiore visibilità nei motori di ricerca, minore dimensione delle pagine e caricamento più veloce, compatibile con le diverse piattaforme. Risulta perciò molto penalizzato un software CMS che non dispone di tale certificazione.

#### *2.3.2.Usabilità: help e documentazione*

Questo insieme di requisiti è stato individuato affinché si possa avere un indice di bontà del supporto e assistenza disponibili per un dato software. I requisiti di help e documentazione vengono catalogati secondo:

- **Commercial Manuals:** indica la disponibilità di un servizio a pagamento che fornisca materiale di supporto disponibile in modalità cartacea.

- **Developer Community:** indica, per un determinato portale, la presenza di una community di sviluppo libera (non a fini di lucro). Essa favorisce un ambiente di sviluppo per il portale, dando la possibilità di tenerlo sempre aggiornato ed implementare nuove funzioni;
- **On-line Help:** indica la presenza di un sistema di guida on-line integrato nel sistema;
- **Professional Hosting:** indica la possibilità di poter acquistare un servizio hosting a pagamento professionale. Si definisce hosting un servizio che consiste nell'allocare su un server web le pagine di un sito web, rendendolo così accessibile dalla rete Internet . Tale "server web", definito "host", è connesso alla rete Internet in modalità idonea a garantire l'accesso alle pagine del sito mediante browser, con identificazione dei contenuti tramite dominio ed indirizzo IP;
- **Professional Services:** indica la presenza di organizzazioni di servizi professionali a pagamento per fornire o personalizzare i servizi amministrativi per il portale;
- **Public Forum:** indica la presenza di un forum pubblico gratuito dedicato alla piattaforma;
- **Support & Training:** indica la possibilità di poter contattare a pagamento un'organizzazione commerciale che abbia uno staff qualificato che fornisca supporto e assistenza;
- **Third-party Developers:** indica la presenza di sviluppatori che creano eventuali plug-in per il sistema anche su richiesta.

#### 2.4.Requisiti non funzionali: affidabilità

Tale requisito è utile per indicare la capacità del sistema o di una componente del sistema a fornire una determinata funzione richiesta sotto certe condizioni e per un definito periodo di tempo. Come requisito di affidabilità è stato individuato il

- **Versioning:** funzione utile a realizzare un prototipo di sistema per la modellazione degli schemi e la gestione delle versioni. Questo prototipo dovrà supportare la gestione delle history degli schemi concettuali ed il backporting assistito delle nuove modifiche alla versione corrente. In questo modo sarà quindi possibile evitare l'obsolescenza del sito.

## 2.5. Requisiti non funzionali: sostenibilità

Tali requisiti vengono individuati per rendere semplici le modifiche dopo il deployment. Includono l'adattabilità, cioè l'abilità di cambiare il sistema per trattare concetti addizionali del dominio di applicazione, la manutenibilità, cioè l'abilità di cambiare il sistema per trattare nuove tecnologie e per far fronte a difetti, l'internazionalizzazione e la configurabilità.

### *2.5.1. Sostenibilità: adattabilità*

Questo genere di requisiti sono stati selezionati con l'intento di permettere all'utente di cambiare interfaccia, anche se solo a livello superficiale, secondo i propri gusti, oppure per permettere la modificazione dell'interfaccia da parte del sistema in risposta al comportamento dell'utente. I requisiti di adattabilità vengono suddivisi in:

**Advertising Management:** indica la possibilità di inserire annunci pubblicitari e/o altre funzioni simili per facilitare lo spazio dedicato ad aziende che vogliono mostrarsi tramite il sito;

**UI levels:** indica la possibilità di differenziazione delle interfacce amministrative a seconda del grado di competenza dell'utente.

### *2.5.2. Sostenibilità: manutenibilità*

La manutenibilità esprime un importante requisito di progetto di un sistema che definisce la sua capacità di essere facilmente ripristinato qualora sia necessario realizzare un intervento di manutenzione. La manutenibilità rappresenta cioè il requisito indispensabile del sistema per ottimizzare l'implementazione delle attività manutentive. In tal merito è stato selezionato il solo requisito di **Help desk** con sistema di ticketing, usato per la segnalazione di errore, ad esempio quelli di codice.

### *2.5.3. Sostenibilità: internazionalizzazione*

L'internazionalizzazione è un requisito volto al processo di adattamento del prodotto, pensato e progettato per un mercato o un ambiente definito, ad altri mercati o ambienti, in modo particolare altre nazioni e culture. Per il caso di studio in questione tale requisito è stato suddiviso in:

- **Localizzazione interfaccia:** offre la possibilità di scelta per l'amministratore della lingua con cui operare;
- **Supporto multilingue:** possibilità di visualizzazione del sito nella lingua desiderata dall'utente.

#### 2.5.4. Configurabilità

La configurabilità è la capacità del software di essere impostato nel suo comportamento, senza richiederne modifiche al codice sorgente. Il concetto di configurabilità è diretta conseguenza del principio di separazione, ovvero separare il meccanismo dalla politica, e dal principio DRY (don't repeat yourself), ovvero ciascuna informazione deve essere dichiarata una sola volta nel software. I requisiti di configurabilità per un software che deve costruire e gestire un portale universitario sono stati rilevati in:

- **Site Setup:** indica la possibilità da parte del sistema di una creazione di default di aree destinate ai contenuti, stili e altre impostazioni durante la fase di inizializzazione o installazione del sito;
- **Themes & Skins:** indica la possibilità di personalizzare l'interfaccia grafica e la gestione della struttura del sito.

#### 2.6. Requisiti non funzionali: implementazione

Per requisiti di implementazione si intendono tutti quei vincoli sull'implementazione del sistema, incluso l'uso di tool specifici, linguaggi di programmazione o piattaforme hardware. Per il caso di studio in questione vengono differenziati fra limitazione sulle risorse, linguaggi e strumenti, requisiti di interfaccia, requisiti operativi, requisiti fisici.

##### 2.6.1. Implementazione: limitazione sulle risorse

L'unico aspetto tenuto in considerazione riguarda quali Sistemi operativi sono supportati dal software CMS in esame.

##### 2.6.2. Implementazione: linguaggi e strumenti

Appartengono a questa categoria tutti quelle indicazioni che riguardano linguaggi di programmazione, web server, application server, ecc con i quali il CMS è in grado di lavorare e supporta. Sono stati suddivisi in:

- **Linguaggio di programmazione:** sta ad indicare il linguaggio di programmazione con il quale è implementato il CMS e/o con il quale è possibile svilupparlo;
- **Database:** indica il DBMS (data base management system) che il CNS utilizza per salvare contenuti e impostazioni;
- **Web server:** indica la piattaforma di gestione database, ovvero il software con il quale il CMS è compatibile, per poter essere eseguito.



### *2.6.3.Implementazione: requisiti di interfaccia*

I requisiti di interfaccia rappresentano i vincoli imposti da necessità di interfacciamento verso sistemi esterni. Si considera il solo parametro **LDAP Authentication** (Autenticazione LDAP - Lightweight Directory Access Protocol), ovvero si verificherà se il CMS ha la possibilità di supportare il protocollo LDAP per dialogare con database e directory.

### *2.6.4.Implementazione: requisiti operativi*

Sono requisiti che riguardano la gestione del sistema nel contesto applicativo. Vengono individuati in:

- **Online administration:** indica la possibilità di gestire completamente il sistema tramite un web browser (quindi da un qualsiasi terminale) senza dover dipendere da un software installato su un determinato pc;
- **Web Statistics:** indica la disponibilità di accumulare statistiche e visionare report relativi al sistema (numero di utenti on-line, numero di visite di una determinata pagina, ecc.).

### *2.6.5.Implementazione: requisiti fisici*

I requisiti fisici riguardano:

- **Licenza:** il tipo di licenza sotto la quale è distribuito il CMS;
- **Costo:** indica il costo approssimativo del CMS. È deciso da chi lo distribuisce ed è regolamentato dalla licenza con la quale è certificato.

## **3.Presentazione dei CMS**

### 3.1.CMS Made Simple

CMS Made Simple è un CMS Open Source che permette di creare e gestire interi portali Web in modo molto semplice.

La documentazione, disponibile anche in lingua italiana, è molto aggiornata grazie alla numerosa community. Come linguaggio questo CMS fa uso di PHP e si appoggia su database MySQL o PostgreSQL. Può essere reso operativo su Windows, Mac OS X, Linux e qualsiasi piattaforma che supporti PHP e almeno un web server tra Apache, IIS o LightTDP. La distribuzione è gratuita e sotto licenza GPL.

Contatti:

- Sito web ufficiale: <http://www.cmsmadesimple.org>
- Community Italiana [http://wiki.cmsmadesimple.org/index.php/Main\\_Page/it](http://wiki.cmsmadesimple.org/index.php/Main_Page/it)

### 3.2.dotCMS

dotCMS è un CMS open source. È un'applicazione web basata su J2EE, che si appoggia su web server Tomcat. Si tratta di un buon prodotto che può contare su una comunità di utenti in continua crescita.

Giunto alla versione 1.6, disponibile per il download, il prodotto viene distribuito secondo la dual license strategy: può essere utilizzato gratuitamente sotto licenza GPL 2.0, obbligando però l'utilizzatore a distribuire eventuali applicazioni sviluppate per esso, o con licenza commerciale se non si vuole essere vincolati a tale distribuzione.

Contatti:

- Sito web ufficiale <http://www.dotcms.org>

### 3.3.Drupal

Drupal è un CMS modulare scritto in linguaggio PHP e distribuito sotto licenza GNU GPL. Drupal funziona su diversi sistemi operativi, tra cui Windows, Mac OS X, Linux e qualsiasi piattaforma che supporti i web server Apache o IIS e PHP. Drupal utilizza un database per memorizzare i contenuti, e necessita dunque di un software come MySQL e PostgreSQL, che sono gli unici DBMS al momento supportati.

Contatti:

- Sito web ufficiale <http://www.drupal.org>
- Sito italiano di riferimento <http://www.drupalitalia.org/>

### 3.4.e107

e107 è un CMS open source basato su PHP con supporto a database MySQL. È un prodotto relativamente giovane, nato nel 2002, ed è attualmente distribuito gratuitamente con licenza GNU GPL. È supportato da una comunità internazionale in continua espansione che sviluppa frequentemente nuovi plugin e skin.

L'ultima versione disponibile per il download è la v7.11.

Contatti:

- Sito web ufficiale <http://www.e107.org>
- Sito web ufficiale e107 Italia <http://www.e107italia.org>

### 3.5.Joomla!

Joomla! è un CMS realizzato completamente nel linguaggio PHP. È nato nel settembre 2005 come un fork del codice di Mambo; attualmente è in rapido sviluppo, sotto la guida di un gruppo di sviluppatori (per buona parte ex-sviluppatori di Mambo) riuniti nell'associazione no-profit Open Source Matters. Joomla! è pubblicato con licenza open source GNU GPL v.2, e l'ultima versione disponibile per il download è la 1.5.6.

Contatti:

- Sito web ufficiale <http://www.joomla.org>
- Sito web ufficiale Joomla! Italia <http://www.joomla.org>

### 3.6.Mambo

Mambo è uno dei CMS open source più popolari del web e può contare su un'ampia comunità di supporto. Si avvale della tecnologia PHP in appoggio a database MySQL.

Il CMS è formato dal core (il nucleo), e da un insieme di elementi aggiuntivi - denominati componenti, moduli, mambot e template - che ne permettono l'espansione. Data la relativa facilità di realizzazione di queste espansioni il CMS si contraddistingue per l'adattabilità ai contesti più diversi. Mambo è distribuito gratuitamente sotto licenza GNU/GPL v2 da oltre sette anni.

Contatti:

- Sito web ufficiale <http://mambo-foundation.org>
- Supporto ufficiale Mambo Italia <http://www.supportomambo.it>

### 3.7.MODx

MODx è un CMS open source sviluppato per pubblicare contenuti sul web e su intranet, sviluppato in PHP con supporto a database MySQL. MODx è il risultato di un fork di Etomite. La piena diversificazione dei contenuti, dell'interfaccia e delle funzioni è uno degli aspetti più forti e unici di MODx. Questo CMS è distribuito gratuitamente sotto licenza GPL.

Contatti:

- Sito web ufficiale: <http://www.modxcms.com>

### 3.8.mojoPortal

MojoPortal è un CMS open source e richiede ASP.NET 2.0. Se installato in Windows richiede il web server IIS, in Linux necessita di Apache. Necessita inoltre uno tra i seguenti database: MSSQL/SQLExpress, MySQL, PostgreSQL, Firebird Sql o SQLite. mojoPortal è rilasciato sotto licenza CPL.

Contatti:

- Sito web ufficiale: <http://www.mojoportal.com>

### 3.9.ONEcms

OneCMS è un CMS open source scritto in PHP che fa uso di un database MySQL. OneCMS, per come è stato scritto ed implementato, è comunemente usato per siti web

di giochi. OneCMS è nato nel 2004 da Charlie Page di Insane Visions. La versione 1.0 di OneCMS è stata rilasciata il 3 Giugno 2004, al prezzo di \$30. Quasi due anni dopo, il 15 Gennaio 2008, la versione 2.5 di OneCMS è stata rilasciata al pubblico.

Contatti:

- Sito web ufficiale: <http://www.insanevisions.com>

### 3.10.OpenCMS

OpenCMS è un CMS open source basato sulla piattaforma Java e tecnologia XML. È distribuito da Alkacon Software su licenza LGPL .

La storia di OpenCMS inizia attorno al 1999 con il suo predecessore MhtCms, che non fu rilasciato Open Source. La prima versione Open Source fu rilasciata al CeBIT Expo 2000. OpenCMS è un'applicazione web basata su JSP/Servlet e necessita quindi di un servlet container come Apache Tomcat. L'utilizzo necessita anche di un database relazionale come Mysql o PostgreSQL. L'ultima versione disponibile per il download è la 7.0.5.

Contatti:

- Sito web ufficiale <http://www.opencms.org>

### 3.11.PHP-Fusion

PHP-Fusion è un leggero CMS open source scritto e sviluppato da Nick Jones in PHP. Usa un database MySQL per immagazzinare il contenuto del sito e possiede un sistema di amministrazione semplice ma completo.

A partire dal 2006, PHP-Fusion supporta circa una dozzina di lingue oltre all'inglese. È rilasciato sotto i termini della GNU General Public License.

La versione più diffusa e stabile di questo CMS è la 6.01.15, ma dallo scorso agosto è stata rilasciata la prima release finale di PHP-Fusion 7.

Contatti:

- Sito web ufficiale <http://php-fusion.co.uk>
- Sito di supporto italiano <http://www.php-fusion.it>

### 3.12.PHP-Nuke

PHP-Nuke è un CMS sviluppato con tecnologia PHP e si appoggia ad un database MySql per l'immagazzinamento dei dati.

PHP-Nuke è rilasciato sotto licenza GNU GPL. Per ottenere l'ultima versione è necessario, da alcuni mesi, pagare 10 dollari, anche se rimangono gratuite tutte le versioni precedenti l'ultima. Il tipo di licenza adottato, tuttavia, consente a chi acquista l'applicazione di ridistribuirla, anche gratuitamente.

La sua diffusione avviene nell'anno del suo boom, il 2001, quando ormai giunto alla versione 4.4 comincia ad avere un numero sempre crescente di utilizzatori.

L'ultima versione disponibile per il download è la 8.1, ma presto sarà rilasciata la nuova release 9.0.

Contatti:

- Sito web ufficiale <http://phpnuke.org>

### 3.13.Plone

Plone è un CMS basato sul server per applicazioni web Zope e sul linguaggio di programmazione Python.

Plone è rilasciato sotto licenza GPL, si tratta quindi di un software completamente gratuito, che può essere scaricato e usato liberamente. Il progetto è nato nel 1999 grazie a Alan Runyan, Alexander Limi, e Vidar Andersen. La prima versione di questo content management system è stata rilasciata nel 2001. L'ultima versione disponibile è la 3.1.5.1 dello scorso agosto.

Contatti:

- Sito web ufficiale <http://www.plone.org>
- Community Plone in Italia <http://www.plone.it>

### 3.14.SilverStripe

SilverStripe è un CMS gratuito e open source per la creazione e l'amministrazione di siti web. Prima di SilverStripe 2.x, il CMS era distribuito a pagamento con una licenza proprietaria. Nello sviluppo di SilverStripe 2.x, iniziato alla fine del 2005, è stato compiuto un grande sforzo per usufruire dei benefici della programmazione orientata agli oggetti e delle nuove features di PHP5. Il 3 febbraio 2007, SilverStripe 2.0.0 è stato rilasciato al pubblico come software gratuito e open source

SilverStripe è ora disponibile sotto i termini della licenza BSD. L'ultima versione scaricabile è la 2.2.2 rilasciata a maggio 2008.

Contatti:

- Sito web ufficiale <http://www.silverstripe.com>

### 3.15.TikiWiki CMS

TikiWiki CMS/Groupware, originariamente e comunemente conosciuto come TikiWiki, è un CMS open source per lo sviluppo di siti web e portali su internet e intranet. TikiWiki può essere usato anche come una struttura wiki, un sistema di tracciamento dei bug, una piattaforma collaborativa, un blog o un forum internet.

La prima release stabile rilasciata è stata la versione 0.9 nell'Ottobre 2002. Ormai giunto alla versione 2.0, TikiWiki CMS/Groupware è disponibile sotto licenza LGPL.

Contatti:

- Sito web ufficiale <http://www.tikiwiki.org>

### 3.16.TYPO3

Typo3 è un CMS pensato per l'amministrazione di siti web. È scritto completamente in PHP, basato su una banca dati gestita con MySQL.

Typo3 può essere ampliato usando dei Plugin (chiamati Extensions). Queste estensioni consentono di usare molte applicazioni (per esempio: news, commercio elettronico, forum ecc.) senza doverle programmare. Il sistema è stato creato dal danese Kasper Skårhøj, che ha pubblicato la prima versione nel 1997. Typo3 è stato rilasciato sotto licenza GPL ed è un progetto open source che viene sviluppato continuamente. L'ultima versione di Typo3 è la 4.2 rilasciata ad aprile 2008. Lo sviluppo della versione 5.0 è già iniziato.

Contatti:

- Sito web ufficiale <http://www.typo3.com>
- Portale italiano <http://www.typo3.it>

### 3.17.TYPOlight

TYPOlight è un potente sistema di gestione di contenuti che rispetta i requisiti di accessibilità in modo totale (front end e back end) e che utilizza XHTML e CSS per generare pagine conformi a W3C/WAI. Sviluppato da Leo Feyer nel 2004, oggi viene rilasciato sotto licenza Lesser General Public License (LGPL). Si può installare su UNIX, Linux, Windows o Mac, necessità di un web server quali Apache (consigliato) o IIS e di un database come MySQL o Oracle. Giunto alla versione 2.6.

Contatti:

- Sito web ufficiale <http://typolight.org>
- Portale italiano <http://typolightcms.it>

### 3.18.XOOPS

XOOPS è un CMS che permette agli amministratori di creare facilmente siti web dinamici. È un potente strumento per lo sviluppo di piccole e grandi comunità virtuali. È scritto in PHP e utilizza database come MySQL.

XOOPS è l'acronimo di eXtensible Object Oriented Portal System (sistema per portali estensibili orientati agli oggetti). Anche se è nato come un sistema per portali, XOOPS

sta di fatto lasciando la strada dei Content Management System e può essere utilizzato come framework per piccoli, medi e grandi siti.

XOOPS è rilasciato sotto i termini della GNU General Public License (GPL) ed è libero di essere utilizzato, modificato e ridistribuito sotto licenza GPL.

Contatti:

- Sito web ufficiale <http://www.xoops.org>
- Supporto ufficiale XOOPS Italia <http://www.xoopsitalia.org>

## 4.Long List

### 4.1.Tabella comparativa

Di seguito è riportata la tabella comparativa nella quale sono stati raccolti i punteggi di ciascun CMS per ogni criterio fissato.

Si ricorda che il punteggio va da un minimo di 0 (zero) a un massimo di 2. Al fine dell'attribuzione del punteggio totale ciascuna valutazione è moltiplicata per il peso assegnato al requisito corrispondente. I pesi vanno da un minimo di 1 (minimamente importante) a un massimo di 5 (assolutamente importante).

	Peso [1-5]	CMS Made Simple	dotCMS	DRUPAL	e-107	JOOMLA	Mambo	MODx	mojoPortal	ONEcms	OpenCMS	Php-Fusion	Php-Nuke	Plone	SilverStripe	TikiWikiCMS	TYPO3	TYPOLight	XOOPS
<b>FUNZIONALITA'</b>																			
<b>CAPACITA'</b>																			
Calendario e Gestione eventi	3	0	2	2	0	2	0	0	0	0	0	0	0	2	2	0	2	2	2
Content scheduling	5	0	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
Content Syndacation	1	2	2	2	2	2	2	2	2	0	0	2	0	2	2	2	2	1	2
Database reports	4	0	2	0	2	2	2	2	0	0	0	1	0	2	2	1	2	0	2
E-mail form	1	2	2	2	2	2	2	2	2	0	2	2	0	2	2	2	2	2	2
FAQ Management	2	2	2	2	2	2	2	2	0	0	2	2	2	2	1	2	2	1	2
File Distribution	4	2	2	2	2	2	2	2	2	0	0	2	2	2	1	2	2	2	2
Motore di ricerca	5	2	2	2	2	2	2	2	2	2	2	2	0	2	2	2	2	2	2
My page / Dashboard	3	0	2	2	2	0	2	2	2	0	0	1	0	2	0	2	2	0	2
Newsletter	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
Photogallery & Videogallery	4	2	2	2	2	2	2	2	2	2	2	2	1	2	2	2	2	2	2
Site map	4	2	2	2	2	2	2	2	2	0	2	2	0	2	2	2	2	2	2
Upload File e Immagini	4	2	2	2	2	2	2	2	1	2	2	0	1	2	2	2	2	2	2
Web-editor (WYSIWYG)	5	2	2	2	2	2	2	2	2	2	2	2	1	2	2	1	2	2	2
<b>SICUREZZA</b>																			
Approvazione dei contenuti	3	0	2	2	2	2	2	0	2	0	2	2	0	2	1	2	2	1	2

Audit trail	5	2	2	2	2	0	0	2	0	0	2	0	0	2	2	2	2	2	1
Captcha	1	2	2	2	2	2	2	2	2	0	2	0	0	2	2	2	2	2	2
Login history	1	0	2	2	2	2	2	2	2	2	2	2	1	2	2	2	2	2	2
Privilegi granulari	5	0	2	2	2	2	2	2	2	0	2	2	0	2	2	2	2	2	2
Problem notification	1	0	2	0	2	0	0	0	2	2	2	0	0	2	0	0	1	1	1
Session Management	4	0	2	2	2	2	1	2	1	0	0	0	0	2	2	0	2	1	1
Verifica indirizzo e-mail	3	0	2	2	2	2	2	2	2	0	0	2	0	2	2	2	2	2	2
<b>USABILITA'</b>																			
<b>FATTORI UMANI</b>																			
Accessibilità (W3C)	5	0	0	2	2	2	1	1	1	0	1	0	0	2	2	0	2	1	1
<b>HELP E DOCUMENTAZIONE</b>																			
Commercial Manuals	4	0	2	2	0	2	2	2	0	0	2	0	0	2	2	0	2	0	2
Developer Community	4	2	2	2	2	2	2	2	2	0	2	2	1	2	2	0	2	2	2
On-line Help	5	2	0	2	2	2	2	2	2	2	2	1	0	2	2	2	2	2	1
Professional hosting	2	2	2	2	0	2	2	2	2	0	1	0	2	2	2	2	2	2	2
Professional services	2	2	2	2	2	2	2	2	2	0	2	0	0	2	2	2	2	2	2
Public Forum	4	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
Support & Training	3	0	2	2	0	2	2	2	2	0	2	1	0	2	2	1	2	2	2
Third-party Developers	2	2	2	2	2	2	2	2	0	0	2	2	1	2	2	1	2	2	2
<b>AFFIDABILITA'</b>																			
<b>RIPRISTINABILITA'</b>																			
Versioning e Undo	5	0	2	2	0	0	1	0	1	2	2	0	0	2	2	2	2	2	2
<b>SOSTENIBILITA'</b>																			
<b>ADATTABILITA'</b>																			
Advertising Management	2	2	2	2	2	2	2	2	0	2	0	0	2	2	1	2	2	2	2
UI levels	1	2	2	0	2	2	2	2	0	0	2	0	0	2	2	2	2	2	2
<b>MANUTENIBILITA'</b>																			
Help Desk - Ticketing	1	0	0	2	2	2	2	0	0	0	0	2	0	2	2	2	2	2	2
<b>INTERNAZIONALIZZAZIONE</b>																			
Localizzazione interfaccia	5	2	2	2	2	2	2	2	2	0	2	0	2	2	2	2	2	2	2
Supporto multilingue	5	0	2	2	2	2	2	2	2	0	2	2	2	2	2	2	2	2	2
<b>CONFIGURABILITA'</b>																			
Site setup	1	0	2	1	2	0	0	0	0	0	0	0	0	2	0	2	0	0	0
Themes & Skins	2	2	0	2	2	2	2	2	0	2	0	0	1	2	0	2	2	2	2
<b>IMPLEMENTAZIONE</b>																			
<b>LIMITAZIONE RISORSE</b>																			
Sistema operativo	1	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
<b>LINGUAGGI E STRUMENTI</b>																			
Database	5	2	2	2	2	2	2	2	2	1	1	1	2	2	2	2	2	2	1
Linguaggio di programmazione	5	2	1	2	2	2	2	2	1	2	1	2	2	2	2	2	2	1	1
Web server	5	2	2	2	2	2	2	2	2	1	2	2	2	2	2	2	2	2	2
<b>INTERFACCIA</b>																			



Autenticazione LDAP	5	0	2	2	2	2	2	0	2	0	1	0	2	2	2	2	2	0	2
<b>OPERATIVI</b>																			
Online Administration	5	2	2	2	2	2	2	2	2	2	2	2	2	1	2	2	2	2	2
Web Statistics	2	2	2	2	2	2	2	2	0	2	0	0	1	2	2	2	2	0	2
<b>FISCHE</b>																			
Costo	5	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
Licenze	5	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
<b>TOTALE</b>		200	291	309	288	292	288	279	241	146	240	194	153	320	299	264	321	263	290
	Peso [1-5]	CMS Made Simple	dotCMS	DRUPAL	e-107	JOOMLA	Mambo	MODx	mojoPortal	ONEcms	OpenCMS	Php-Fusion	Php-Nuke	Plone	SilverStripe	TikiWiki CMS	TYPO3	TYPOLight	XOOPS

Si procede ora con la stesura della classifica basata sul punteggio totale raggiunto da ciascuna piattaforma analizzata.

#### 4.2. Classifica

<b>CMS</b>	<b>Punteggio</b>
TYPO3	321
Plone	320
DRUPAL	309
SilverStripe	299
JOOMLA	292
dotCMS	291
XOOPS	290
e-107	288
Mambo	288
MODx	279
TikiWikiCMS/Goupware	264
TYPOLight	263
mojoPortal	241
OpenCMS	240
CMS Made Simple	200
Php-Fusion	194
Php-Nuke	153
ONEcms	146

Come si nota dalla tabella precedente, tra la nona e la decima posizione si ha un netto calo di punteggio. Si può così individuare una lista dei CMS candidati per entrare nella short list. Tra i 9 CMS candidati si dovrà operare un'opportuna selezione per raggiungere il numero fissato di 6 piattaforme.

## 5.Short List

A partire dalla long list si ottiene, attraverso una prima valutazione, una lista più ristretta di software che meglio rispondono alle esigenze del Dipartimento. Prima di inserire direttamente nella short list le soluzioni migliori, è opportuno integrare i risultati della tabella precedente con considerazioni sulla reale possibilità di utilizzo di ciascuna piattaforma, in modo da limitare ulteriormente il numero di software per ottenere una short list di 6 elementi. Solo a questo punto si può stilare la short list, dalla quale si andrà ad estrarre la soluzione che alla fine verrà utilizzata.

### 5.1.Considerazioni

<b>CMS</b>	<b>Punteggio</b>
TYPO3	321
Plone	320
DRUPAL	309
SilverStripe	299
JOOMLA	292
dotCMS	291
XOOPS	290
e-107	288
Mambo	288

La prima piattaforma in classifica è TYPO3. L'elevato punteggio attribuitogli sta ad indicarne la sua completezza e rispondenza ai requisiti previsti. Tuttavia dall'analisi dei feedback rilasciati dagli utilizzatori nelle varie community dedicate, risulta che i tempi di apprendimento, pur per un utilizzo di base del prodotto, sono molto lunghi e che la piattaforma è complessa a causa della ricchezza degli strumenti presenti. Queste caratteristiche si scontrano con le necessità primarie del Dipartimento, rendendo inadatto TYPO3.

Il secondo CMS non inseribile nella short list è Mambo. Tale scelta è dettata dalla presenza di Joomla! tra i candidati. Si ricorda infatti che Joomla! deriva da un fork del codice sorgente di Mambo e dalla sua nascita sviluppatori e utenti hanno spostato le loro attenzioni sempre più verso il nuovo progetto, rendendo Mambo meno appetibile soprattutto in prospettiva futura.

Tra i CMS rimanenti viene escluso dotCMS in quanto, da un'analisi comparativa tramite lo strumento Google Trends, risulta essere il meno popolare della lista. Questa caratteristica, insieme alla scarsità di community dedicate reperibili in rete, lo rende poco adatto nel caso sia necessario avere assistenza.

Dopo queste considerazioni, i CMS che entrano a far parte della Short List sono i seguenti:

- Plone
- DRUPAL
- SilverStripe
- JOOMLA
- XOOPS
- e-107

La fase successiva consisterà nell'analisi operativa di ciascuna piattaforma, per poter decretare quale sia la più adatta nella realizzazione di un portale di dipartimento universitario.



# CAPITOLO 5

## Realizzazione di un portale universitario con Joomla!

In questo capitolo verrà presentato Joomla! attraverso la sua storia e le sue particolarità. Si andrà poi a studiare la realizzazione di un portale di dipartimento universitario utilizzando questo software. Verranno analizzate le caratteristiche di Joomla! relativamente al progetto in esame, le problematiche incontrate e le soluzioni trovate sfruttando gli strumenti offerti dalla piattaforma.

### 1.Cos'è Joomla!

Joomla! è un software per la gestione dei contenuti (CMS, *Content Management System*) dedicato alla realizzazione di siti web. È realizzato interamente in linguaggio PHP ed è distribuito con licenza GNU GPL v.2. L'utente ha quindi libertà di utilizzo, copia, modifica e distribuzione, purché le modifiche apportate siano a loro volta rilasciate con le stesse modalità. Joomla! consente di creare, modificare e gestire i contenuti di un sito internet, di una rete intranet o di altre applicazioni per il web senza avere particolari conoscenze tecniche. Il tutto è contenuto in un pacchetto compresso. L'installazione richiede pochi minuti e un pacchetto software con una piattaforma di sviluppo web/database. L'amministrazione del CMS avviene tramite un'interfaccia web-based, perciò è possibile prendersi cura del proprio sito da qualsiasi postazione dotata di connettività internet utilizzando un semplice browser.

#### 1.1 WAMP Server

WAMP Server è un'implementazione -dedicata per Windows- della piattaforma AMP. È anch'esso rilasciato sotto licenza GNU GPL ed è disponibile in versione "installer" e .zip (che permette di avviare WAMP Server da chiavetta USB).

Il nome è un acronimo derivante dal nome dei componenti software di cui è costituita:

- Microsoft **W**indows: il sistema operativo prerequisito;
- **A**pache (v. 2.2.17): il Web Server;
- **M**ySQL (v. 5.3.4): il Database Management System;
- **P**HP (v. 5.1.53): il linguaggio di scrittura.

Sul mercato sono presenti diversi pacchetti, tra i quali EasyPHP, Filezilla. La scelta è ricaduta su WAMP Server (v. 2.1) per la facilità d'installazione (nessun problema di compatibilità con Win7-64 bit), la presenza dell'ultima versione di PHP e l'interfaccia intuitiva.

## 2. Storia di Joomla!

Joomla! nasce nel Settembre 2005 da una fork dal codice del CMS Mambo. Il suo rapido e continuo sviluppo (è attualmente in rilascio la versione 1.7) è sotto la guida di un gruppo di programmatori riuniti nell'associazione no-profit Open Source Matters, ma gode del contributo di un'ampia comunità internazionale, che ne è il vero cuore pulsante. Essa infatti implementa molti dei componenti (moduli, templates e plug in) che vengono poi distribuiti tramite i siti nazionali delle varie comunità sparse per il globo.

Sono frequenti gli eventi di richiamo per gli sviluppatori del CMS, sia a livello nazionale che mondiale, a supporto del continuo miglioramento del software. Per l'utenza che invece si pone come semplice utilizzatrice di Joomla! la comunità ha il suo luogo d'incontro nei forum on-line, dove vengono affrontati problemi, trovate soluzioni, e implementate nuove funzionalità.

## 3. Installazione di Joomla!

### 3.1. Installazione di WAMP Server

Il primo passo è l'installazione del pacchetto WAMP Server, il cui installer è scaricabile da <http://www.wampserver.com/en/download.php>. L'unica attenzione da porre è nella scelta del file, condizionata dalla versione di Windows presente nel pc in uso: 32 o 64 bit.

Se è attivo Skype, il consiglio è di chiudere momentaneamente l'applicativo, a causa di un possibile conflitto di porte (80 e 8080) utilizzate anche da WAMP Server. Chiarite le premesse, basta lanciare l'installazione con un doppio click;



Figura 5.1- pop up di installazione

L'installatore presenta i soliti passaggi necessari all'installazione di un software. Cliccare "Next" per proseguire, "Cancel" per tornare indietro e annullare l'operazione. Viene richiesto all'utente di scegliere in quale cartella installare il pacchetto e in seguito di creare un account di amministrazione tramite l'inserimento di un *Nome utente* e *Password*.

L'intera procedura richiede pochi minuti, al termine dei quali viene installato il database e configurati i file necessari.

Conclusa l'installazione e lanciata l'applicazione comparirà una piccola icona in basso a destra:

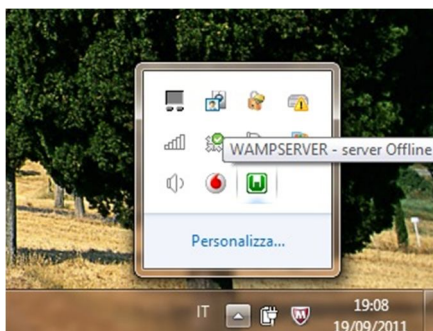


Figura 5.2 – icona WAMP Server

Con un click si aprirà un menù a tendina. Cliccando su "Start all services" e attendendo alcuni secondi, l'icona cambierà colore, dal viola al verde. Non sarà più necessario ripetere l'operazione ad ogni accensione del pc, perché l'avviamento è impostato di default.

Selezionando dal menù a tendina la voce "Localhost" si aprirà una finestra sul browser, come questa:

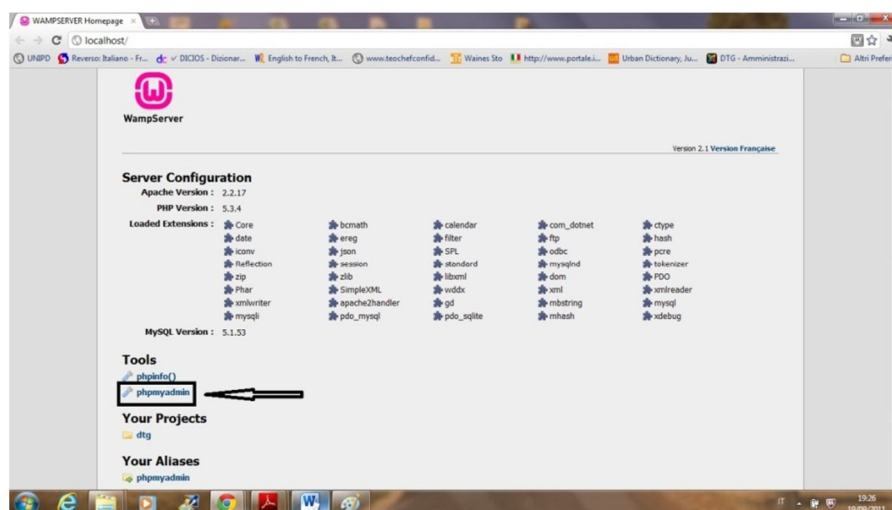


Figura 5.3 – Localhost

Cliccando su "phpMyAdmin" si possono andare ad impostare i parametri del database che supporterà il sito e le credenziali e autorizzazioni dell'utente che avrà la possibilità di modificarne i contenuti.

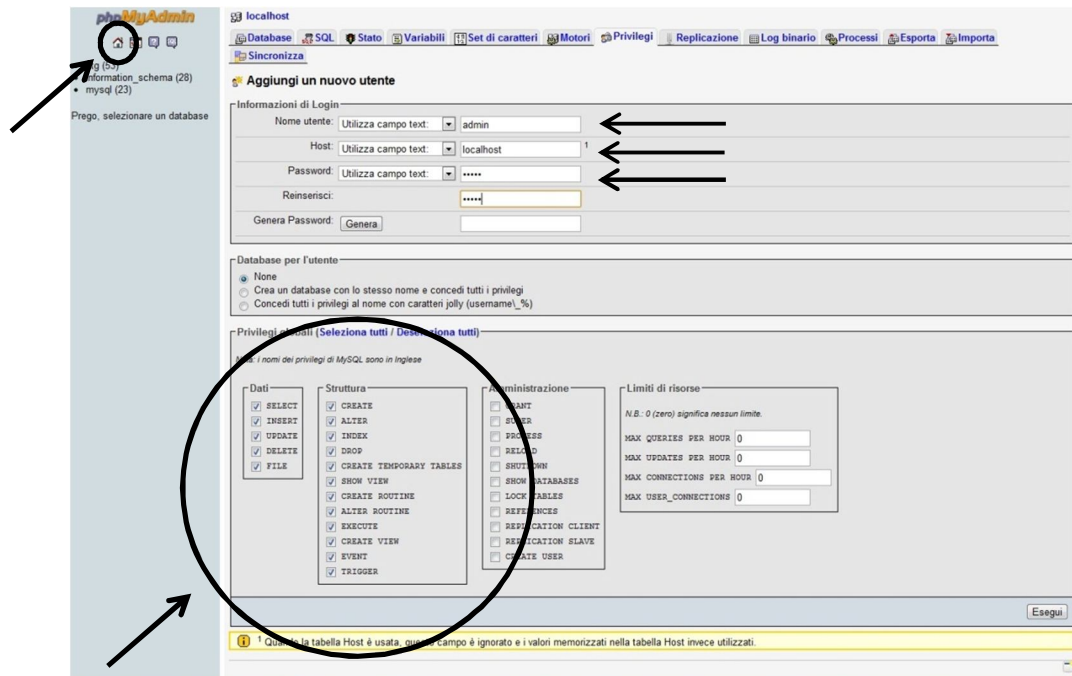


Figura 5.4 – phpmyadmin – Creazione nuovo utente

Cliccando su "Privilegi" e poi su "Aggiungi un nuovo utente" si carica una finestra come sopra. E' necessario inserire un *Nome utente* (settato a "admin"), l'*host* di riferimento ("localhost") e una *Password* (per semplicità è stata scelta "admin"). Vanno poi selezionati i privilegi globali, come in figura 5.4.

Ora bisogna tornare alla Home, cliccando sulla cassetta, in alto a destra, e compilare il campo *Crea un nuovo database*. Nel caso in questione è stato nominato "dtg" (v. figura 5.5).



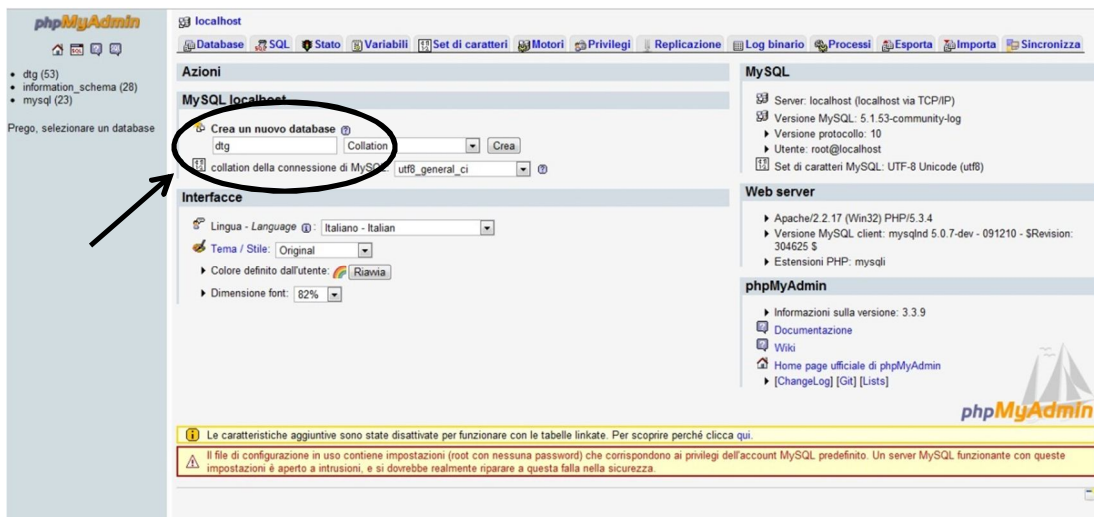


Figura 5.5 – Creazione database

Ora, cliccando sull’etichetta “Privilegi” bisogna selezionare il tasto di modifica dell’utente “admin”, come in figura 5.6

Utente	Host	Password	Privilegi globali 1	Grant	Azione
<input type="checkbox"/> Qualsiasi	%	-	USAGE	No	
<input type="checkbox"/> Qualsiasi	localhost	No	USAGE	No	
<input checked="" type="checkbox"/> admin	localhost	SI	SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, FILE, INDEX, ALTER, CREATE TEMPORARY TABLES, CREATE VIEW, EVENT, TRIGGER, SHOW VIEW, CREATE ROUTINE, ALTER ROUTINE, EXECUTE	No	
<input type="checkbox"/> root	127.0.0.1	No	ALL PRIVILEGES	SI	
<input type="checkbox"/> root	localhost	No	ALL PRIVILEGES	SI	

Figura 5.6 – Modifica privilegi utente

e alla voce “Aggiungi privilegi al seguente database” selezionare dal menù a tendina il database appena creato.

A questo punto è possibile passare all’installazione vera e propria di Joomla!.

### 3.2. Installazione di Joomla!

Il pacchetto è reperibile all’indirizzo <http://www.joomla.it/download.html> e si può scegliere la versione sulla quale si vuole costruire il proprio sito. In questo caso la scelta è ricaduta sulla versione 1.5.23 Stabile (in italiano).

Scaricato il pacchetto lo si decompone nella cartella C:\Programmi\wamp\www (riferimento standard), avendo l’accortezza di creare in quest’ultima una cartella dedicata al sito (ad esempio “sito dtg”).

Richiamando la pagina localhost dalla barra dell’indirizzo del browser o dal menù di WAMP Server (come visto sopra), alla voce “Your projects” risulterà presente la cartella appena creata (“sito dtg”). Cliccandovi sopra si aprirà la pagina di installazione di Joomla!. Verranno ora

visualizzati i controlli pre-installazione. Cliccando “avanti” due volte (la seconda è l'accettazione dei termini di licenza GNU GPL) si aprirà la finestra di configurazione del database.

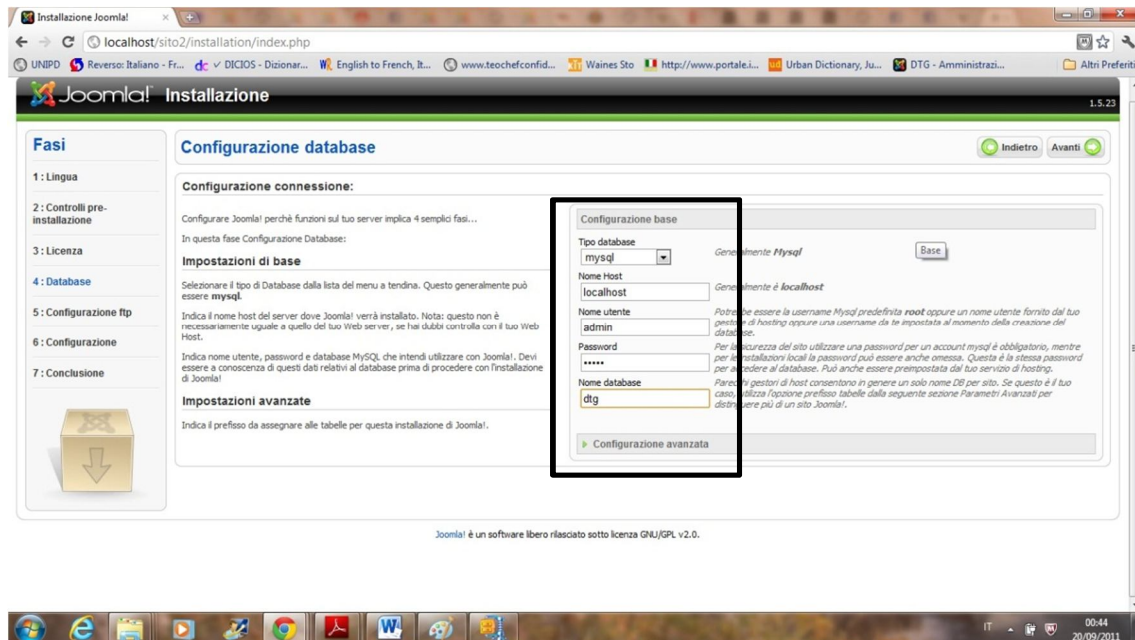


Figura 5.7 – Configurazione database

Vanno inseriti i dati che permettono a Joomla! di interfacciarsi con il database creato precedentemente.

Host name: localhost

Nome utente MySQL: admin

Password: admin

Nome database: dtg

Cliccando su “Avanti” e attendendo alcuni secondi si avrà conferma dell'esattezza dei dati inseriti. Il passi successivi saranno l'inserimento del nome del sito (nel caso “sito DTG”), di una e-mail dell'amministratore e di una password dell'amministratore (settata a “admin” per comodità). È opzionale (ma consigliato per i neofiti) installare i dati di esempio, che posso fungere da utile guida nel compiere i primi passi nella costruzione del sito.

L'installazione è ora conclusa. L'ultimo passaggio necessario comporta la cancellazione della cartella “installation” contenuta in (...) /www/dtg. Gli indirizzi che richiamano il sito, in locale sono:

front-end: <http://localhost/dtg>

amministrazione: <http://localhost/dtg/administrator>

## 4. Gestione dei contenuti

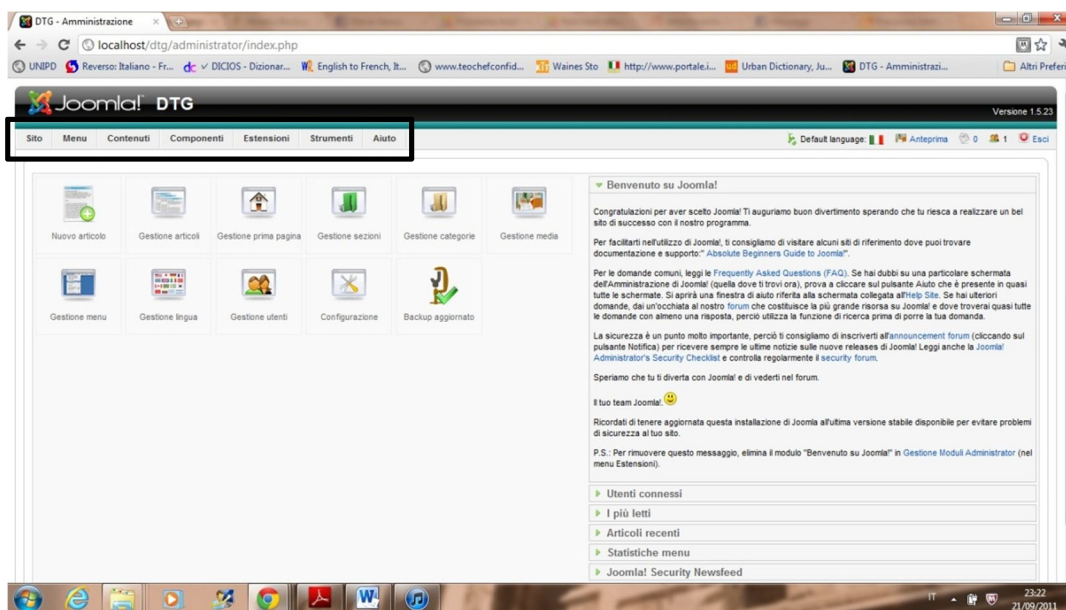


Figura 5.8 – Home page lato back-end

L'interfaccia di Joomla! (v. figura 5.8) sul back-end è molto semplice ed intuitiva. Le grandi icone a centro pagina sono quelle maggiormente utilizzate e rese quindi di facile accesso.

In alto c'è una barra con dei menù che richiamano la totalità delle voci di utilizzo nella costruzione del sito. All'apparenza può sembrare ridondante la presenza dei pulsanti e della barra dei menù, ma risulterà molto utile quando si andrà a lavorare. Nella barra si sottolineano in particolare le voci:

Menù: compare l'elenco dei menù creati e la possibilità di selezionarli per modificarne le caratteristiche o le voci di sottomenù.

Contenuti: in elenco compaiono *articoli*, *sezioni* e *categorie* oltre alla *gestione della prima pagina*.

Componenti: in questo menù vengono visualizzati i componenti preinstallati in Joomla! (banner, contatti, sondaggi, ecc) e quelli che l'utente può installare in seguito (ad es. componenti per il supporto multilingua, per il backup, gestione eventi, ecc.).

Estensioni: ne fanno parte *moduli*, *plugin* e *template*. Sono i tre "tipi" di assemblati sui quali è costruito un sito di questo CMS. In Joomla! vengono chiamate *estensioni*. Cliccando su una delle voci vengono visualizzati gli elenchi tabellari di tutti gli elementi di ogni tipo, con possibilità di modifica, cancellazione, pubblicazione, ecc., come per i contenuti. Di fondamentale importanza e di grande frequenza di utilizzo è la prima etichetta presente nel menù, che permette di installare/disinstallare le *estensioni* del sito. Queste compariranno come nuove

etichette nel menù "Componenti", come si vede in figura 5.9, o saranno rintracciabili in quello "Estensioni". Nel caso in esame sono state installate varie estensioni, che verranno descritte più dettagliatamente nel seguito, qui solo elencate:

- *Joom!fish* (v. 2.1.7) per il supporto multilingua del frontend del sito;
- *Attachments* (v. 2.2) per la gestione degli allegati agli articoli;
- *JCE Editor* (v. 2.0.12) per l'editor degli articoli;
- *EventList* (v. 1.0.1) per la gestione degli eventi;
- *Qivva Calendar*(v. 0.93) per la gestione del calendario eventi (integrato con EventList);
- *Akeeba Backup* (v. 3.3.3) per la gestione dei backup del sito e del database;
- *Xmap* per la creazione della sitemap;

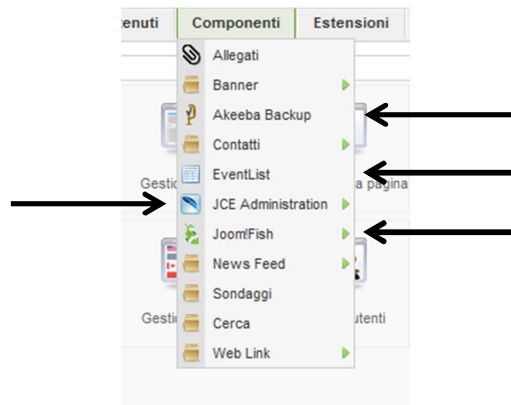


Figura 5.9 – menù "Componenti" aggiornato con le estensioni installate

I contenuti sono il fulcro di Joomla!. La loro configurazione ed elevata flessibilità permette di presentarli all'utente sotto forma di articolo, notizia, informazione, arricchita di immagini, video, foto, animazioni, tabelle, grafici, ecc.

La loro organizzazione risulta quindi fondamentale. Nel caso di un sito costruito con Joomla! fa riferimento a tre livelli gerarchici: *sezioni*, *categorie* e *articoli*. Quelli statici, di conseguenza, risultano non categorizzati né seguono logiche cronologiche particolari.

## 4.1. Creazione di una sezione/categoria

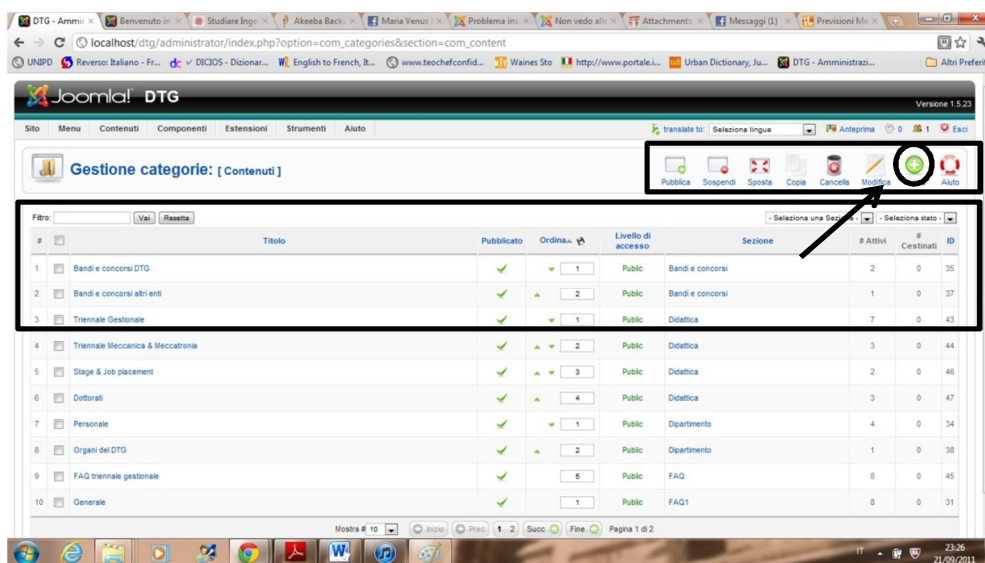


Figura 5.10 – screenshot “Gestione categorie”

Cliccando sulla rispettiva voce del pannello di controllo del backend (v. figura 5.9) (o in alternativa dal menù a tendina “Contenuti”) si accede all’elenco delle sezioni/categorie presenti. Vengono evidenziate le informazioni maggiormente utili, quali il titolo del contenuto, se è pubblicato o meno, il livello di accesso, ecc. La possibilità di utilizzare i filtri facilita la visualizzazione, nel caso siano presenti molti elementi. In alto c’è un pannello con i pulsanti di maggiore utilizzo per modificare, cancellare, pubblicare/sospendere i contenuti.

Cliccando su “Nuovo” si crea un nuovo contenuto. Si inseriscono i dati identificativi della nuova sezione/categoria. Nell’ordine i campi completabili sono:

- *Titolo* da assegnare alla sezione/categoria;
- *Alias della sezione* (se lasciata vuota viene completata in automatico da Joomla!);
- *Pubblicato si/no* che consente di pubblicare immediatamente o meno il contenuto;
- *Livello di accesso* per definire chi può vedere il contenuto;
- *Immagine* che si può associare alla descrizione;
- *Posizione* per posizionare l’immagine selezionata;
- *Descrizione* per descrivere il contenuto.

Nel caso delle categorie, a differenza della sezioni, è presente un menù a tendine nel quale va indicata a quale di queste ultime appartenga la categoria che si sta creando.

Dopo aver cliccato sull’icona “Salva”, la sezione/categoria viene creata e aggiunta all’elenco tabellare.

## 4.2. Creazione di un articolo

Il procedimento di inserimento di un articolo, sia esso statico o dinamico, è semplice e non differisce da quanto descritto sopra.

La differenza sostanziale tra articoli, categorie e sezioni consiste nel numero di opzioni configurabili, presenti nei tre menù a tendina posti a destra, raccolti in figura 5.10.

The image displays three screenshots of the Joomla! article creation interface, showing different parameter menus. The first screenshot shows the 'Parametri - Articolo' menu with fields for Author (Administrator), Alias Autore, Livello di accesso (Public), Data creazione (2011-09-21 22:13:42), Inizio pubblicazione (2011-09-21 22:13:42), and Termine pubblicazione (Mai). The second screenshot shows the 'Parametri - Avanzati' menu with various options like 'Mostra titolo', 'Titoli ipertestuali', 'Testo introduttivo', 'Nome sezione', 'Nome sezione ipertestuale', 'Nome categoria', 'Nome categoria ipertestuale', 'Valutazione articolo', 'Nome autore', 'Data e ora creazione', 'Data e ora modifiche', 'Icona PDF', 'Icona stampa', 'Icona e-mail', 'Lingua contenuto', 'Riferimenti chiave', and 'Testo alternativo Leggi tutto..'. The third screenshot shows the 'Informazioni Metadata' menu with fields for Descrizione, Parole chiave, Robot, and Author.

Figura 5.10 – menù Parametri articolo

Il primo elenco mostra i parametri base di configurazione, relativi all'autore e alla tempistica di pubblicazione dell'articolo. Nel caso di contenuti dinamici risulta fondamentale la possibilità di gestire un inizio e fine pubblicazione di un contenuto.

Il secondo entra in dettaglio sulla modalità di presentazione all'utente, in particolare sulla quantità di informazioni aggiuntive che possono essere visualizzate.

Il terzo elenco si occupa dei parametri utilizzati dal motore di ricerca interno/esterno e dell'indicizzazione delle pagine del sito. Un corretto utilizzo di questi settaggi crea un sito snello e con un alto grado di indicizzazione.

Va sottolineato che ognuna di queste impostazioni può essere modificata per ogni singolo articolo, il che permette un alto grado di customizzazione del contenuto, in base al tipo di articolo e al numero di informazioni aggiuntive che è possibile notificare all'utente. Nel caso si volesse creare un contenuto statico la sola condizione necessaria è settare il menù a tendina relativo alla sezione di appartenenza dell'articolo a "nessuna".

Nel caso in studio è stato installato l'editor JCE (v 2.0.12), con alcune funzionalità in più rispetto a quello installato di default in Joomla!: esso permette di scrivere in modalità

WYSIWIG (What You See Is What You Get) e con una comoda gestione dei link ipertestuali, dei pagebreak, delle tabelle, degli stili CSS ecc.

Tra le possibilità che JCE offre c'è l'upload di file o immagini dal proprio computer al server nelle fasi di inserimento dell'elemento nella pagina. Non si ha quindi la necessità di caricare precedentemente tutto il materiale o di dover abbandonare l'editor per caricare gli elementi di cui si necessita.

È stato inoltre installata un'estensione che rende comoda e di ottimo impatto visivo la gestione degli allegati: Attachments (v 2.2). Tramite un componente e tre plugins permette di caricare allegati di svariate estensioni (.doc, .xls, .pdf, ecc.) attraverso un pulsante posizionato alla base dell'editor che richiama un pop up nel quale è necessario inserire pochi parametri, come si vede in figura 5.10. L'estensione applica di default un'icona accanto al file, e rende visibile gli allegati nell'articolo di appartenenza, nella posizione desiderata.

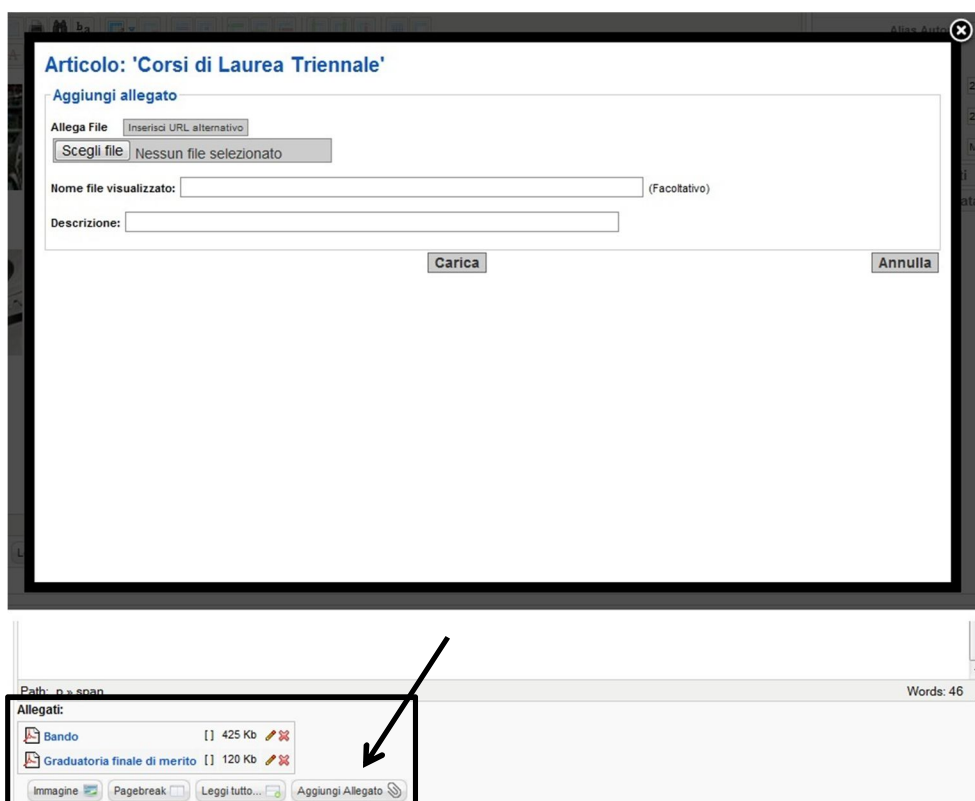


Figura 5.11 – pop up attachments e aspetto allegati

#### 4.3.1 moduli

I moduli sono genericamente delle aree nelle quali vengono visualizzati contenuti testuali, immagini o risultati di elaborazioni di alcuni componenti installati. Contribuiscono largamente alla personalizzazione del sito, tanto che la loro diversa gestione su di uno stesso template, conferisce la sensazione di una totale diversità.

Il loro posizionamento e il grado di visibilità sono infatti completamente configurabili dall'amministratore, che può decidere di visualizzarli solo in alcune aree del sito a discapito di altre.

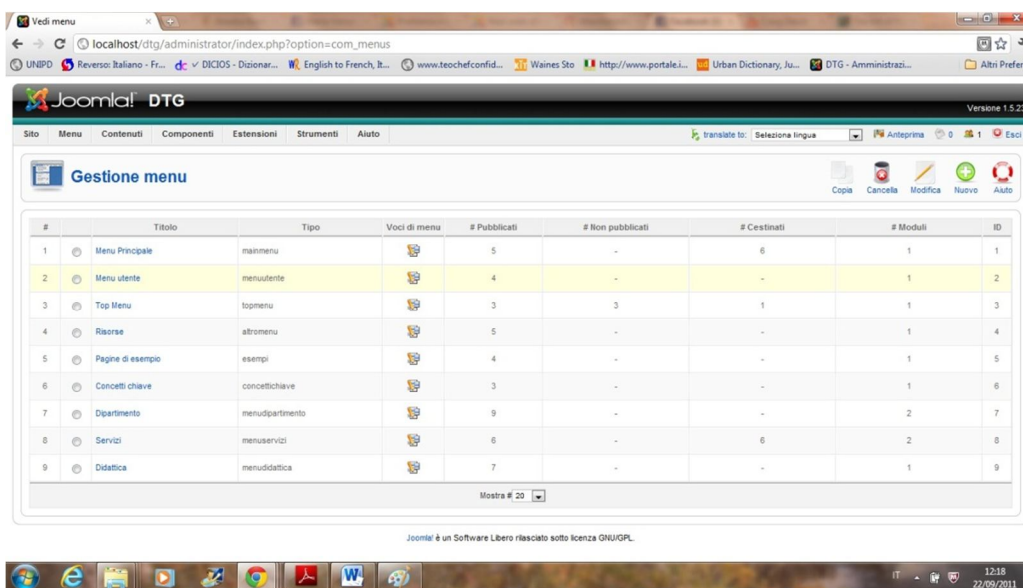
Ogni modulo dispone di un proprio pannello di controllo tramite il quale è possibile assegnare una grande quantità di parametri, a conferma della loro estrema flessibilità di utilizzo.

Verrà ora affrontato uno dei moduli più utilizzati in un sito, i *Menù*.

#### 4.4. Creazione e gestione dei menù

La creazione dei menù in Joomla! è necessariamente un passo successivo alla creazione dei contenuti, dato che ogni voce del menù deve puntare ad uno di essi, sia esso interno al sito o esterno (link web). Può sembrare un aspetto vincolante, ma ciò invece permette una gestione razionale degli stessi.

Dal pannello di controllo sulla mainpage del backend o dalla barra dei menù sovrastante si accede alla "Gestione menù", che presenta un elenco tabellare dei menù presenti e dà alcune informazioni aggiuntive, quali il titolo, il tipo, il pulsante per accedere ai sottomenù di ognuno, il numero di voci pubblicate, non pubblicate e cestinate.












#	Titolo	Tipo	Voci di menu	# Pubblicati	# Non pubblicati	# Cestinati	# Moduli	ID
1	Menu Principale	mainmenu		5	-	6	1	1
2	Menu utente	menuutente		4	-	-	1	2
3	Top Menu	topmenu		3	3	1	1	3
4	Risorse	altromenu		5	-	-	1	4
5	Pagine di esempio	esempi		4	-	-	1	5
6	Concetti chiave	concettichave		3	-	-	1	6
7	Dipartimento	menudipartimento		9	-	-	2	7
8	Servizi	menuservizi		6	-	6	2	8
9	Didattica	menudidattica		7	-	-	1	9

Figura 5. 12 – Gestione menù

Anche qui l'interfaccia è simile a quella degli altri contenuti. Cliccando su "Nuovo" si apre la pagina di creazione del menù.



**Dettagli menu: [ Nuovo ]**

Tipo menu:  ⓘ

Titolo:  ⓘ

Descrizione:  ⓘ

Titolo modulo:  ⓘ

**Figura 5. 113 – Nuovo menù**

L'unico campo opzionale è la descrizione. I pulsanti di informazione accanto aiutano nella compilazione corretta dei campi. Fondamentale è configurare in modo univoco il *Tipo menù* (in seguito si vedrà perché). Cliccando su "Salva" il menù viene aggiunto in lista. Ora è stato creato il "contenitore". Il passo successivo sarà la creazione delle voci del menù, che andranno a puntare ai contenuti. Prima però l'attenzione v'è rivolta alla creazione del *Modulo* che permetterà di visualizzare il menù sul frontend.

Il percorso è il seguente: barra dei menù → "Estensioni" → "Gestione moduli". Cliccando su "Nuovo" compare un elenco di tipi di Modulo tra i quali scegliere (v. figura 5.14). Da notare la presenza anche di moduli relativi ad estensioni installate dall'utente.

**Modulo: [ Nuovo ]**

Sito | Amministrazione

Moduli

- Articoli correlati
- Banner
- Breadcrumb
- Cerca
- Chi è Online
- Contenuti archiviati
- Contenuti più letti
- Footer
- HTML personalizzato
- Immagine casuale
- JoomFish-Language Selection
- Latest Events
- Menu
- Notizie Flash
- Notizie recenti
- Qivva EventList Calendar module
- Sezioni
- Sondaggio
- Statistiche
- Syndication
- Visualizzazione feed
- Wrapper

Successivo Annulla Aiuto

**Figura 5. 14 – Nuovo Modulo**

Cliccando su "Successivo" si giunge alla pagina di configurazione del modulo (v. figura 5.15).

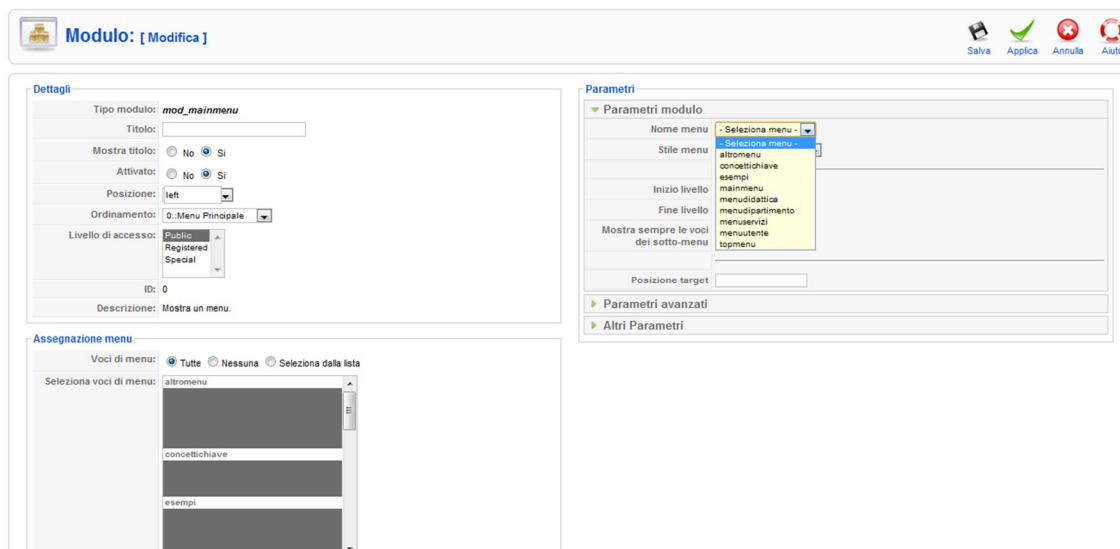


Figura 5. 15 – Configurazione Modulo menù

Alla voce “Dettagli” vanno configurati i seguenti campi:

- *Titolo* del menù;
- *Mostra Titolo si/no* se si vuole che il titolo venga visualizzato nel frontend;
- *Attivato si/no* se si vuole attivare subito il modulo o meno;
- *Posizione* per decidere in quale zona del template si vuole pubblicare il menù;
- *Ordinamento* rispetto agli altri moduli presenti in quella zona del template;
- *Livello di accesso* per decidere a quale tipo di utenza rendere visibile il menù.

Alla voce “Assegnazione menù” si può decidere in quale aree del sito pubblicare il menù; per default Joomla! lo pubblica sempre.

All’interno del riquadro “Parametri” troviamo il menù a tendina “Nome menù”: da questo bisogna selezionare il corrispettivo creato in precedenza. Cliccando su “Salva” si conclude il doppio step che porta alla visualizzazione di un menù nel frontend.

Ponendo l’attenzione nuovamente sulla lista dei menù alla voce “Gestione menù” e premendo il pulsante “Voci di menù” è possibile ora creare voci di sottomenù. La procedura è leggermente diversa. Cliccando su “Nuovo” si apre una pagina come in figura 5.16;

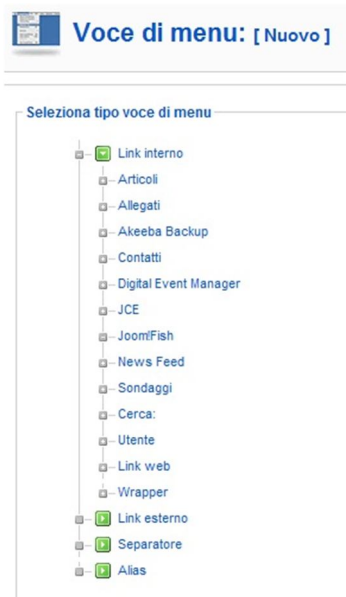


Figura 5. 16 – Tipo voce sottomenù

Joomla! offre la possibilità di puntare a un tipo di contenuto tanti quanti ne sono presenti nel sito. Si ritrovano in questo elenco, infatti, non solo i contenuti preinstallati, ma anche quelli installati dall'utente. Per alcuni tipi di contenuto è possibile scegliere fra varie modalità di presentazione: nel caso di un articolo, ad esempio, la voce di menù può puntare ad un singolo articolo, ad una sezione, ad una categoria e può visualizzarlo come una pagina di blog, come un elenco puntato, un elenco tabellare, ecc.

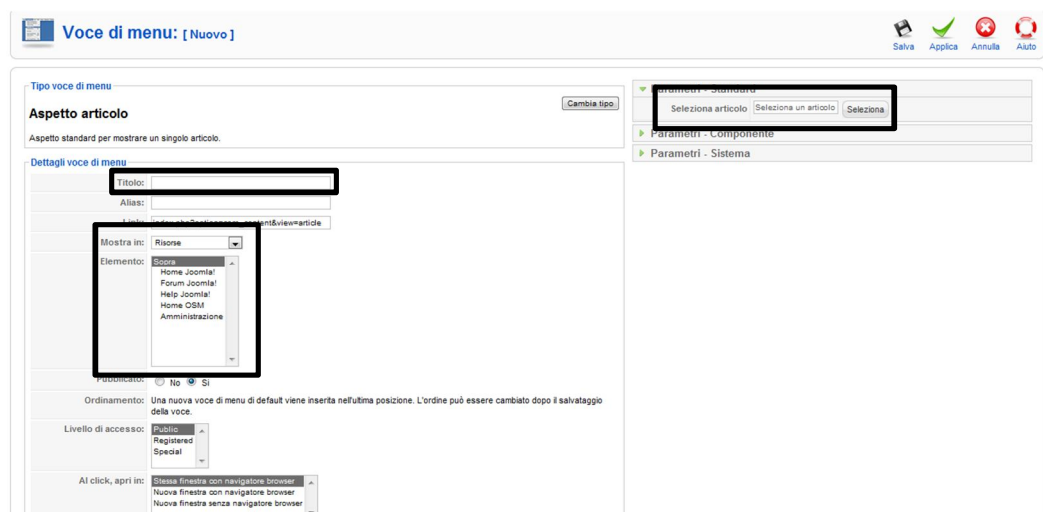


Figura 5. 17 – Aspetto articolo

Ipotizzando la scelta "Aspetto articolo" i parametri più importanti da settare sono:

- *Titolo* della voce menù;
- *Mostra in* per decidere in quale menù verrà visualizzata la voce;

- *Elemento* che identifica di quale grado gerarchico è la voce. Di default Joomla! evidenzia "Sopra", cioè il primo livello. Se si desidera che la voce sia un sottomenù bisogna evidenziare la voce "padre" relativa.
- *Seleziona articolo* : premendo su "Seleziona" si apre un pop up dal quale, tramite anche l'utilizzo di filtri di ricerca, è possibile selezionare l'articolo al quale punterà la voce di menù.

Cliccando su "Salva" si conclude l'operazione e la voce di menù risulterà presente nell'elenco.

#### 4.5.1 Componenti

I componenti sono l'unico tipo di estensione che ha un menù dedicato. Questo a causa della loro importanza nell'implementazione di un sito con Joomla!. Essi infatti sono delle applicazioni software vere e proprie che riescono a svolgere tutte quelle funzioni non incluse nell'installazione standard del CMS, come i forum, le applicazioni di e-commerce, i backup, la gestione degli eventi, ecc.

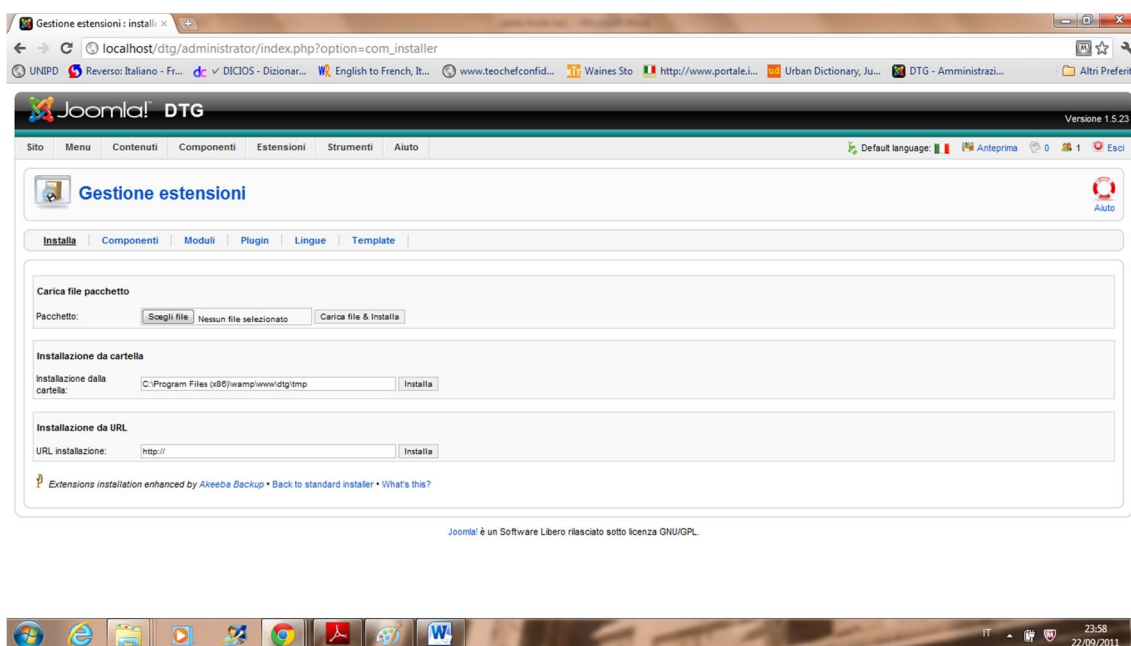
I componenti preinstallati in Joomla! sono:

- *Banner* per la gestione di banner di carattere pubblicitario, testuale o funzionale alla navigazione nel sito;
- *Contatti*;
- *News Feed* per la gestione di feed di provenienza esterna;
- *Sondaggi*;
- *Cerca* per la ricerca interna e la gestione delle relative statistiche;
- *Web Link* per raccogliere, categorizzare e gestire i link.

Oltre a questi ne sono stati installati altri, ad esempio per la gestione dei backup, degli eventi, ecc. Le potenzialità dei *Componenti* di Joomla! sono pressochè infinite, e ne contraddistinguono l'ampiezza di impiego nell'implementazione di siti di ogni natura, dal commerciale, all'istituzionale, allo sportivo, all'aziendale. E su questi elementi costituenti del CMS che la comunità di Joomla! è più attiva.

#### 4.6. Le Estensioni

Con il termine *Estensioni* i programmatori di Joomla! intendono la totalità degli elementi che compongono il CMS stesso. *Componenti, plugin, template, moduli, file di lingua*, tutti sono estensioni e l'installer implementato in Joomla! 1.5 è unico per tutte.



**Figura 5. 18 – Installazione Estensioni**

È necessario unicamente selezionare il pacchetto dell'add-on e Joomla! riconosce che tipo di software è e ne svolge l'installazione. L'origine dei dati, inoltre, può essere diversa dal pacchetto compresso sul pc. Joomla! 1.5 permette infatti, qualora non fosse reperibile il file compresso, l'installazione da directory. In alternativa, si può anche indicare un URL, anche esterno al sito, che punta al file compresso. Al resto pensa Joomla!.

L'unica attenzione da porre è se l'estensione è nativa per Joomla 1.5! o se è necessario abilitare la funzione *Legacy mode* per renderla compatibile.

#### 4.7. Creazione e gestione degli eventi

Joomla! non ha preinstallato un'estensione che si occupi della gestione degli eventi. È stata così installata una add-on tra le svariate reperibili su <http://extensions.joomla.org/>. È stata scelta EventList 1.0.1, in quanto di largo utilizzo nella community, solida nell'implementazione, ampiamente configurabile in base alle esigenze e dotata di plug-in aggiuntive quali la traduzione in italiano (Italian Translation Pack), il calendario (*Qivva Calendar*) e la possibilità di visualizzare gli ultimi eventi (Latest Events).

Ad installazione avvenuta, il primo passo conveniente è l'installazione dei dati di esempio, attraverso il pannello di controllo del componente e il pulsante relativo (v. figura 5.19), per iniziare da subito a sfruttare a pieno lo strumento.



Figura 5. 19 – Pulsante di EventList

È necessario ora creare una voce menù che punti a **Digital Event Manager → Layout Lista** per visualizzare gli eventi (v. figura 5.20).

Data	Titolo	Sede	Città	Type
31.10.2008	Alternative Night	Crash	Freiburg	Alternative
01.11.2008	Rock Punk	Crash	Freiburg	Rock
21.11.2008	Electroschock	Kamikaze	Freiburg im Breisgau	House
16.10.2009	Electronic Beats	Douala	Ravensburg	House
11.11.2009	DJ Night	Douala	Ravensburg	House
29.11.2009 - 30.11.2009	Balkan Beat	Kamikaze	Freiburg im Breisgau	Alternative

Figura 5. 20 – Dati di esempio

Verranno ora analizzate le diverse schede di configurazione presenti nel pannello di controllo principale di EventList (Componenti → EventList, cliccate “Configurazione”), che ne permettono un elevato grado di adattabilità ad ogni tipo di esigenza. Nella prima scheda si andrà ad implementare la *Configurazione di Base*.

The screenshot shows the 'Configurazione di base' interface with the following sections:

- Visualizza configurazione:** Includes options for 'Visualizza link pagina dettagli' (Titoli ipertestuali), 'Formato data' (%d.%m.%Y), 'Formato ora' (%H.%M), 'Nome' (h), and 'Memorizza IP' (No/Si).
- Gestione immagini:** Includes 'Dimensione max immagini' (100), 'Altezza max delle anteprime' (100), 'Larghezza max delle anteprime' (100), and 'Attiva anteprime' (No/Si).
- Gestione eventi:** Includes 'Gestisci eventi passati' (Non fare niente).
- Gestione mail:** Includes 'Mail ad ogni nuovo inserimento/modifica da parte degli utenti' (Disabilitato) and 'Mail all'utente dopo un inserimento' (Disabilitato).
- Opzione meta:** Includes 'Modifica parole chiave' (Titolo evento, Sede, Categoria, Data, Orario d'inizio) and 'Modifica descrizione' (Titolo evento, Sede, Categoria, Data, Orario d'inizio, Data di chiusura, Ora di chiusura).

Figura 5. 21 – Configurazione di Base

Le voci presenti sono:

- *Visualizza configurazione* dove vengono impostati i link all'articolo contenente i dettagli, l'eventuale IP di chi crea l'evento, data e ora e opzioni relative;

- *Gestione eventi* per decidere che fine fanno gli eventi scaduti. Di default vengono archiviati in una directory dalla quale possono essere richiamati;
- *Gestione mail* dal quale si può impostare l'invio/ricezione di mail all'amministratore nel momento in cui l'evento viene modificato;
- *Gestione immagini* per impostare le caratteristiche delle eventuali immagini (programma, locandine, ecc) relative all'evento. Per le anteprime è necessario il supporto GD2, la cui presenza/assenza è evidenziata da un avviso colorato;
- *Opzione Meta* dove si possono impostare le parole chiave e i Metadati da inserire nella pagina.

La seconda scheda di configurazione è *Controllo Utenti*.

The screenshot displays the 'Controllo Utenti' configuration interface, divided into several sections:

- Caricamento immagini:** Includes a dropdown menu for 'Caricamento immagini' set to 'Opzionale'.
- Descrizione:** Features a text input field for 'Lunghezza max della descrizione' with the value '1000'.
- AC - Eventi:** Contains four rows of settings:
  - Nuovi eventi dagli utenti:** A dropdown menu with options: '- disabled -', '- All Registered Users -', 'Lato pubblico', and '- Registered'. Below it are radio buttons for 'Ricorsivo' with 'No' and 'Si' (selected).
  - Autopubblicazione:** A dropdown menu with the same options as above. Below it are radio buttons for 'Ricorsivo' with 'No' and 'Si' (selected).
  - Modifica eventi dal frontend:** A dropdown menu with the same options as above. Below it are radio buttons for 'Ricorsivo' with 'No' and 'Si' (selected).
  - Consenti la modifica dei propri eventi:** Radio buttons for 'No' and 'Si' (selected).
- Registrazione:** Includes two dropdown menus for 'Consenti registrazioni agli eventi' and 'Consenti le cancellazioni dagli eventi', both set to 'Opzionale'.
- AC - Sedi:** Contains three rows of settings:
  - Nuove sedi dagli utenti:** A dropdown menu with options: '- disabled -', '- All Registered Users -', 'Lato pubblico', and '- Registered'. Below it are radio buttons for 'Ricorsivo' with 'No' and 'Si' (selected).
  - Autopubblicazione:** A dropdown menu with the same options as above. Below it are radio buttons for 'Ricorsivo' with 'No' and 'Si' (selected).
  - Modifica sedi dal frontend:** A dropdown menu with the same options as above. Below it are radio buttons for 'Ricorsivo' with 'No' and 'Si' (selected).
  - Consenti modifica delle proprie sedi:** Radio buttons for 'No' and 'Si' (selected).

**Figura 5. 21 – Controllo Utenti**

- *Caricamento immagini e Descrizione* per stabilire se l'upload di immagini è obbligatorio/facoltativo e la lunghezza massima della descrizione;
- *AC-Eventi* permette all'amministratore di configurare il livello di accesso degli utenti rispetto alla possibilità di creare/modificare eventi dal frontend;
- *Registrazione* è utile a decidere se gli utenti possono iscriversi/disisciversi all'evento;
- *AC-Sedi* per decidere se permettere agli utenti di creare/spostare le sedi degli eventi dal frontend.

La scheda *Dettagli* permette di configurare i dettagli dell'evento.

Figura 5.22 – Dettagli Evento

Le voci sono poche, e di grande facilità di configurazione, trattandosi di menù a tendina e on/off:

- *Eventi* dove decidere se visualizzare ora, descrizione e titolo dell'evento;
- *Sedi* per stabilire se mostrare l'indirizzo, descrizione, sito web ed eventualmente l'inserimento di una mappa, tramite Map24 o GoogleMaps (verrà richiesto un ID Map24 o un API Key Google e sarà necessario inserire il codice ISO per la Nazione: IT per Italia, ecc.);
- *Registrazione e commenti* dove si può impostare la configurazione ad altri plugin che permettono la visualizzazione dell'evento nel profilo utenti o la possibilità di pubblicare commenti.

La penultima scheda riguarda il *Layout* di pubblicazione degli eventi.

Figura 5.23 – Layout

Questa scheda è altrettanto intuitiva quanto la precedente e permette di configurare alcuni parametri nella visualizzazione di tipo "elenco tabellare" degli eventi: quali visualizzare o meno, la larghezza delle colonne e il titolo.

L'ultima scheda riguarda i *Parametri Globali*.





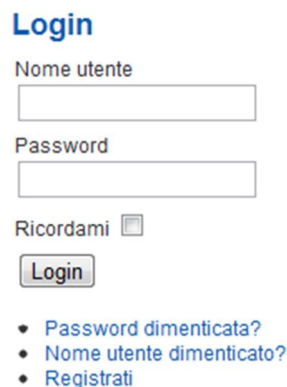
**Figura 5. 24 – Parametri globali**

In essa l'amministratore può decidere quanti eventi visualizzare di default nell'elenco, quante categorie, la presenza/assenza del filtro e le icone di stampa, stampa pdf, mail che sono ormai usuali negli articoli di Joomla!.

## 5. Gestione utenti

L'altro grado di flessibilità di questo CMS si riflette anche nella gestione degli utenti. Joomla! permette infatti a categorie diverse di utente di avere un accesso personalizzato sia ai contenuti posti nel frontend del sito sia alle funzioni di backend al cui utilizzo posso essere autorizzati dall'amministratore.

Il modulo di identificazione è preinstallato in Joomla! e visibile nel frontend. Esso permette ad un utente di effettuare il log in così come di registrarsi (v. figura 5.25).



**Figura 5. 25 – Login da frontend**

Sul lato backend l'amministratore ha ulteriori possibilità, che si ritrovano nel pannello di controllo Sito → Gestione utenti (v. figura 5.26)

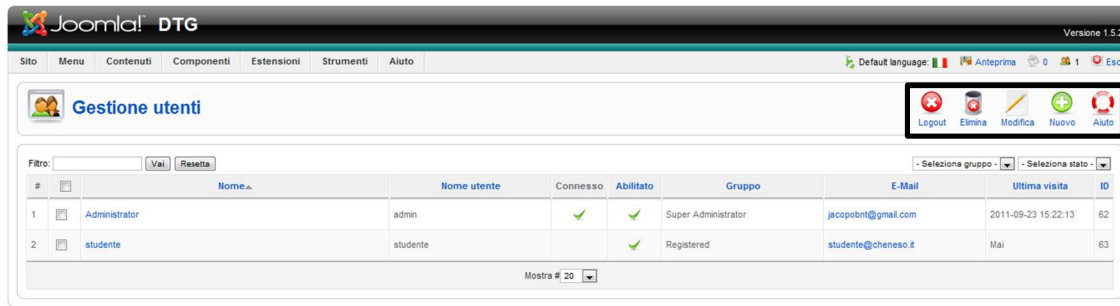


Figura 5. 26 – Gestione utenti

Le funzioni di cui gode l'amministratore sono in primis:

- *Logout* che sottintende la possibilità di chiudere la sessione dell'utente selezionato;
- *Eliminare* per cancellare gli utenti selezionati;
- *Modificare* per accedere al pannello di configurazione del singolo utente;
- *Nuovo* per creare un nuovo utente dal backend;
- *Aiuto* per richiamare una pagina della guida, in caso di necessità.

### 5.1. Gruppi di utenti

Joomla! prevede di default alcuni gradi di accessibilità per gli utenti, che possono quindi ottenere abilitazione a creare e modificare contenuti del sito, a visualizzare o meno alcuni di essi e ad accedere o meno ad alcune aree. La facilità di configurazione dei gruppi è notevole, e differisce per competenze relative al frontend e al backend. In genere la gerarchia dei privilegi è "a cipolla", cioè un utente di grado superiore ad un altro ne ha tutti i privilegi, più alcuni aggiuntivi.

Permessi per operare dal frontend:

- *Registered*: gli utenti registrati possono visitare aree normalmente non accessibili a visitatori che navigano in modalità anonima;
- *Autore*: oltre ad avere accesso a tutte le aree destinate agli utenti *Registered* può inviare notizie, link, ecc. tramite un'apposita voce di menù del *Menù utente*. I contenuti, prima della pubblicazione, sono sempre validati dall'amministratore;
- *Editor*: oltre ai privilegi dell'*Autore* può modificare gli articoli presenti in frontend, sempre previa autorizzazione dell'amministratore;
- *Publisher*: il privilegio più importante è il poter pubblicare articoli senza autorizzazione dell'amministratore.

Permessi per operare nel backend:

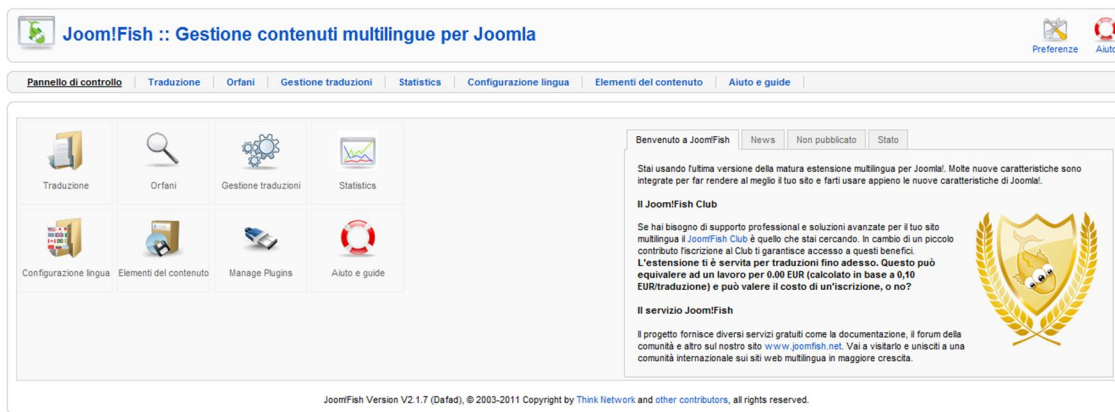
- *Manager*: consente solamente di operare su tutti i contenuti ed effettuare qualche semplice cambiamento di configurazione dei moduli. Può infatti:
  - Creare o modificare i menù;
  - Confermare l'iscrizione di nuovi utenti;
  - Gestire sezioni, categorie e contenuti;
  - Lavorare sui componenti presenti, senza poterli installare/disinstallare.
- *Administrator*: ha un ventaglio di privilegi relativamente ampio, ma gli è negato di:
  - Creare/modificare i Super-amministratori;
  - Apportare modifiche al menù di *Configurazione Globale*;
  - Inserire/modificare lingue e template;
  - Inviare mass mail agli altri utenti.
- *Super-Administrator*: Possiede tutti i permessi, inclusi ovviamente quelli dai quali era escluso l'*Administrator*.

## **6.Supporto multilingua**

Avere un sito in grado di essere consultabile in una o più lingue internazionali è ormai diventata una consuetudine. Per permettere questo e per gestire in modo ordinato e razionale la mole di contenuti ridondanti che viene a crearsi, Joom!fish risulta essere un'estensione ideale.

È uno strumento altamente intuitivo, facilmente configurabile e con una ricchezza di lingue supportate che lo rende leader tra le extensions di gestione della lingua. La comunità che lo sviluppa e ne implementa i moduli è infatti attivissima e dinamica ormai da anni.

L'installazione è semplice, e non differisce in nulla da quella di qualsiasi altro componente. A questa segue, per comodità, l'installazione del plugin della lingua riferita al backend, che permette di avere il pannello di controllo in italiano (v. figura 5.27) e l'installazione dei pacchetti relative alle lingue nelle quali si vuole visualizzare il sito.



**Figura 5. 27 – Pannello di controllo di Joom!fish**

Alla voce *Configurazione lingua* è possibile gestire le lingue installate, rendendo attive quelle di interesse.

Cliccando su *Traduzione* si accede ad una pagina che mostra (con la stessa interfaccia degli articoli), l'elenco dei contenuti del sito: articoli, banner, menù, ecc. (v. figura 5.28).

						Lingua:	Elementi del contenuto:
						English (United Kingdom)	Menus
						Filtro parola chiave:	Filtro menu: Tutti i menu
						Disponibilità traduzione: Qualsiasi	Resetta
<input type="checkbox"/>	Titolo	Lingua	Traduzione	Ultima Modifica	Stato	Pubblicato	
<input type="checkbox"/>	Amministrazione	English (United Kingdom)			●	●	
<input type="checkbox"/>	Aspetto dei contenuti	English (United Kingdom)			●	●	
<input type="checkbox"/>	Bandi e concorsi	English (United Kingdom)			●	●	
<input type="checkbox"/>	Biblioteca	English (United Kingdom)			●	●	
<input type="checkbox"/>	Blog categoria	English (United Kingdom)			●	●	
<input type="checkbox"/>	Blog sezione	English (United Kingdom)			●	●	
<input type="checkbox"/>	Calendario accademico	English (United Kingdom)	Academic schedule	Mercoledì 21 Settembre 2011 19:44	✓	✓	
<input type="checkbox"/>	Caratteristiche	English (United Kingdom)			●	●	
<input type="checkbox"/>	Collaborazioni	English (United Kingdom)			●	●	

**Figura 5. 28 – Elenco Traduzioni**

Tramite una serie di filtri ben strutturati si possono selezionare i contenuti di interesse. L'interfaccia rende visibile a colpo d'occhio (grazie ad una spia verde/rossa e ai campi "Traduzione" e "Ultima modifica") se i contenuti sono stati tradotti e quando.

Sono presenti tutti i pulsanti che permettono modifiche, pubblicazione/sospensione, eliminazione, ecc.

Attivando il modulo relativo si rendono visibili sul frontend le bandierine che permettono all'utente di visualizzare il sito (o sue aree) in una lingua piuttosto che in un'altra. Come per tutti i moduli, l'amministratore ha la facoltà di decidere in quali aree del sito pubblicare le bandiere e quindi permettere all'utente di visualizzarne i contenuti tradotti o meno.

## **7.Realizzazione del portale del Dipartimento di Tecnica e Gestione dei Sistemi Industriali**

Dopo un accurato studio, per poter ottimizzare l'usabilità del portale in via di costruzione, si è deciso che la struttura di base del sito del dipartimento di tecnica e gestione dei sistemi industriali dovesse essere la seguente:

*Struttura di base:*

- Dipartimento
  - Presentazione
  - Bandi e Concorsi
  - Seminari e Convegni
  - Mappe degli Edifici
  - Organi
  - Persone
    - Docenti
    - Collaboratori di ricerca
    - Dottorandi
    - Personale Tecnico e Amministrativo
  - Associazioni
  - Collaborazione
- Didattica
  - Corsi di Laurea
  - Insegnamenti
  - Calendario Accademico
  - Dottorati di Ricerca
  - Laboratori didattici
  - Post Lauream
  - Mobilità Internazionale
  - Job placament
  - Stage
- Ricerca
  - Gruppi di Ricerca
  - Laboratori di Ricerca
- Servizi
  - Biblioteca

- Prenotazione aule e laboratori
- Help e guide

*Elementi in primo piano:*

- Calendario
- Cerca
- Come Raggiungerci
- Contatti
- Eventi
- FAQ
- Link utili
- News e Avvisi
- Site map

*Area Studenti:*

- Appelli d'Esame
- Appelli di Laurea
- Iscrizione/Risultati Esami
- Orario delle Lezioni
- Prenotazione Tesi

Si tratta di una linea guida inizialmente concordata da seguire ed eventualmente migliorare in base alle opportunità offerte dal CMS con cui si opera.

Joomla! permette all'amministratore di implementare gran parte dei contenuti del sito grazie ai componenti preinstallati. Si sono rese comunque necessarie alcune extensions (cfr. par.4, pag.68) per gestire caratteristiche peculiari del sito in costruzione. Il primo passo è stata l'installazione di un modulo che gestisse comodamente il backup del sito. Lavorando in locale, la procedura risulta semplice anche da gestire manualmente, ma si è preferito testare una delle extension disponibili in rete. *Akeeba Backup* ha permesso di gestire questa occorrenza.

La costruzione del sito è partita dalla creazione delle sezioni e delle categorie necessarie, facendo riferimento alla struttura di base sopra elencata. In seguito sono stati creati gli articoli, grazie al *JCE Editor*, che ha reso semplice il collegamento tra i contenuti interni ed esterni,

L'inserimento delle immagini, video, ecc. È stata installata l'extension *Attachments* per allegare file ad alcuni contenuti che lo rendevano necessario.

La fase successiva ha riguardato l'implementazione dei menù e delle relative voci, cui sono state fatte corrispondere i contenuti. Il frontend del sito ha così iniziato a prendere forma.

Sono stati quindi installate le extensions che hanno permesso di implementare quelle funzioni necessarie per il sito del Dipartimento, e che non erano preinstallate nel CMS: *Joom!fish* per il supporto multilingua: sono stati tradotti i contenuti, le voci di menù, i moduli, ecc; *EventList* per la creazione e gestione degli eventi, e i relativi plug in, che hanno permesso di visualizzare nel frontend il calendario e gli ultimi eventi; infine *Xmap* per offrire all'utente la mappa del sito.

Si è passati quindi alla gestione grafica del frontend. In rete si possono trovare migliaia di template, più o meno conformi alle necessità e agli standard di accessibilità richiesti dal caso in studio. L'ampia flessibilità dello strumento ha permesso di dover ricorrere raramente alla configurazione dei fogli di stile CSS. Sono state necessarie solo alcune modifiche, relative al posizionamento del logo: procedure ben spiegate nei forum della community e nelle guide, e di facile gestione anche per l'utente neofita. Sono stati inseriti dei banner e si è lavorato sul posizionamento di tutti i contenuti, rendendo il sito il più possibile accessibile e chiaro all'utente.

## **8.Backup e ripristino**

Tempo ed energie spese per la costruzione di un sito possono essere vanificate con poco, soprattutto nel caso di siti dinamici, che presuppongono l'interazione con oggetti/soggetti esterni e lo scambio di dati di vario genere e grado. La possibilità di intrusioni esterne è quindi reale. A questa bisogna aggiungere i normali disservizi ed errori che possono verificarsi in qualunque momento, che possono portare alla completa distruzione del sito o alla sua inutilizzabilità.

Per questo motivo risulta non solo utile, ma necessario effettuare il backup dell'intero sito: periodicamente e anche prima alcune operazioni delicate o di entità rilevante che l'amministratore si appresta a compiere.

Il backup manuale prevede due attività da svolgere: il backup del database e il backup dei file del sito.

## 8.1.Backup del database

La copia del database è un'operazione che può essere svolta dall'applicazione *PhpMyAdmin* presente nel pacchetto WAMP Server e che tutti i provider rendono disponibile col servizio MySQL.

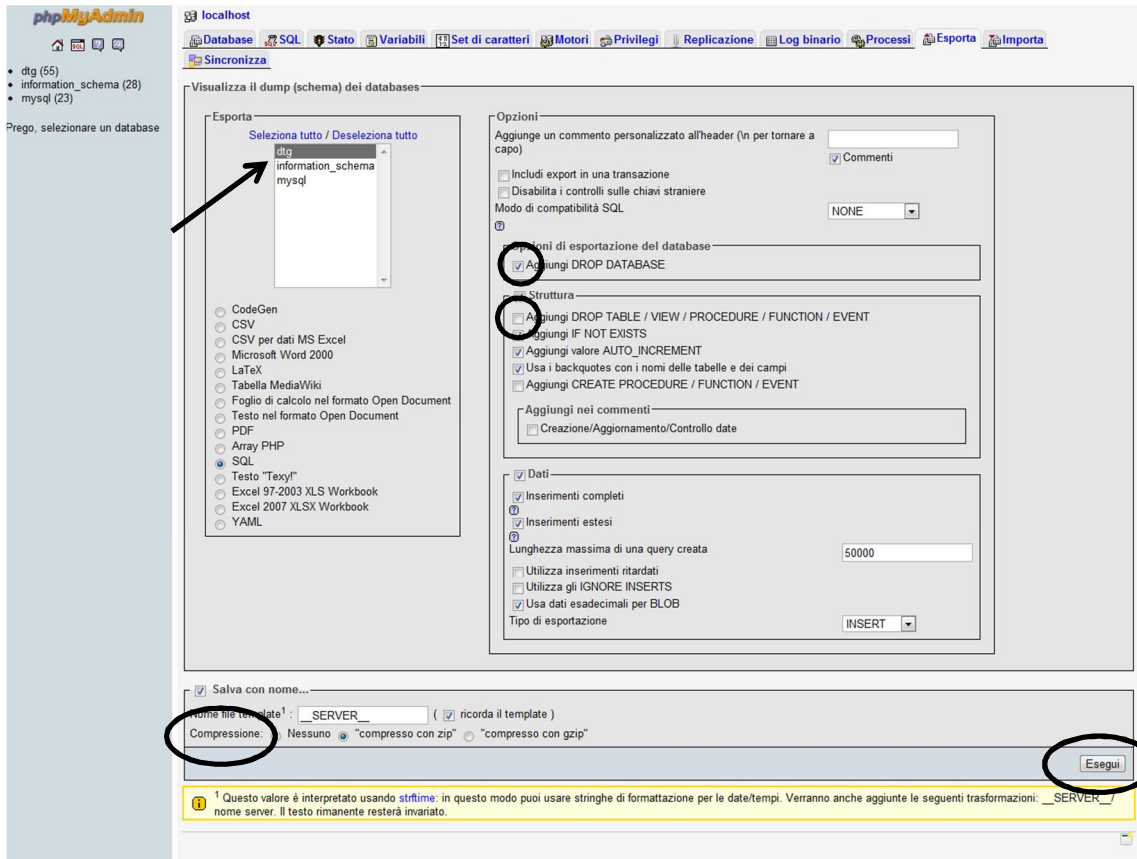


Figura 5. 29 – backup da PhpMyAdmin

Cliccando sull'etichetta "Esporta" si apre il pannello di configurazione del backup. Si seleziona il database che si vuole esportare (nel caso in studio "dtg"), si spuntano le caselle relative a "Aggiungi DROP DATABASE" e "Aggiungi DROP TABLE/ (...)" alla voce "Struttura". Si può scegliere se e in che modalità comprimere il file da esportare. Compilato il pannello, basta cliccare su "Esegui" e salvare il file sul proprio pc o su un'unità a scelta.

## 8.2.Backup del sito

Questa operazione altro non è che la copia fisica della cartella contenente i file del sito. All'interno della cartella C:\(...)\www si trova la cartella del sito (nel caso in studio "dtg"). La si copia ed è prassi inserirla in una cartella che abbia un riferimento della data e ora del backup, assieme al database esportato in precedenza.



### 8.3.Restore

Si tratta dell'operazione inversa, che permette di riportare il sito esattamente nelle condizioni in cui era al momento del backup. Si compone anche questo di due operazioni, relative alla copia fisica della cartella del sito e all'importazione del database tramite *PhpMyAdmin*. Quest'ultima operazione passa attraverso alcune brevi fasi: la creazione di un nuovo database; selezionato il nuovo database si va a cliccare sull'etichetta "Importa" dal pannello dello strumento e tramite il pulsante "Sfoggia" si seleziona il database che si intende ripristinare. Si clicca su "Esegui" e si attende che compaia un messaggio di conferma di buon esito dell'operazione.

La fase di copia della cartella del sito presuppone attenzione su due aspetti:

1. Nel caso dovessimo caricare su server i file, bisogna accertarsi se è necessario modificare il file *configuration.php* che è quello deputato a collegare il database ai file del sito. In particolare questo controllo e modifica sono necessari nel caso si cambi server. Le variabili da controllare/adequare sono le seguenti:

`var$host = 'nome dell'host MySQL. In generale localhost';`

`var$user = 'nome utente amministratore del database';`

`var$db = 'nome del database';`

`var$password = 'password di accesso al database';`

`var$log_path = 'percorso per trovare la cartella logs';`

`var$tmp_path = 'percorso per trovare la cartella tmp'.`

2. È necessario inoltre accertarsi che le cartelle che necessitano di permessi di scrittura lo siano effettivamente. Per verificarlo, dopo aver importato il database e aver effettuato l'accesso al backend, bisogna richiamare la voce *Info Sistema* dall'etichetta "Aiuto" sulla barra dei menù.

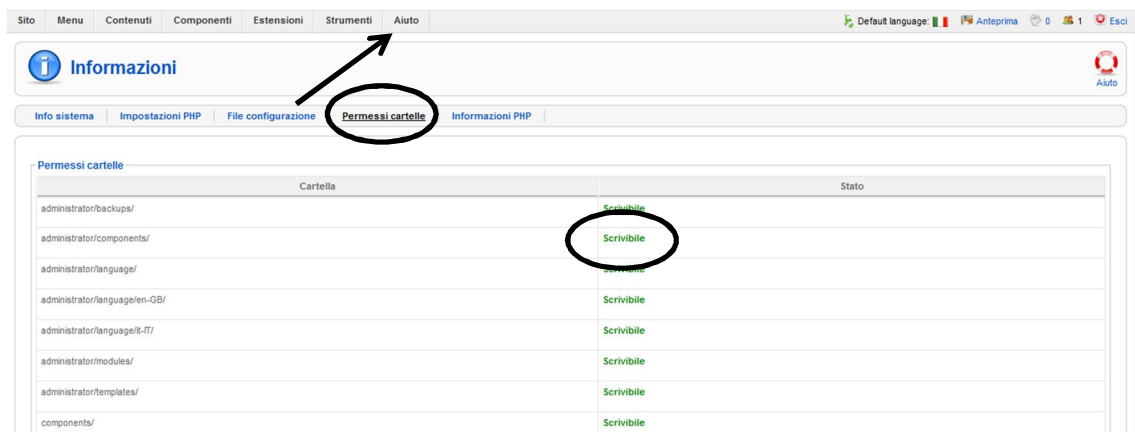


Figura 5. 30 – Info Sistema

Mediante un controllo della colonna "Stato" alla voce "Permessi cartelle" si può verificare la condizione attuale dei permessi di scrittura delle cartelle.

#### 8.4.Backup tramite *Akeeba Backup*

Per facilitare questa operazione di backup si è scelto di installare un'extension che permettesse un rapido e sicuro backup sia del database che dei file del sito.

Appena conclusa l'installazione di *Akeeba Backup* una procedura guidata permette di configurare le preferenze basilari per i backup. Sono tutti parametri che possono essere cambiati in ogni momento dall'amministratore, tramite un pannello di controllo intuitivo e ben fornito (v. figura 5.31).

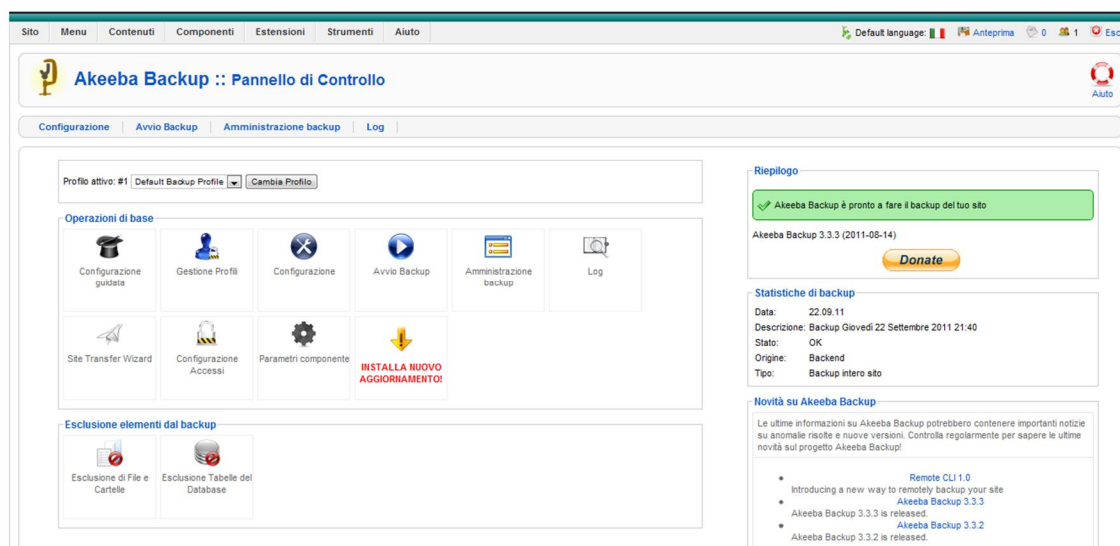


Figura 5. 31 – pannello di controllo di Akeeba Backup

Questo strumento offre alcune peculiarità che posso risultare utili:

- La possibilità di scegliere quali dati inserire nel backup: solo i file del sito, solo il database o entrambi;
- Uno storico dei backup, grazie al quale creare dei veri e propri punti di ripristino differenziati; Si può decidere che *Akeeba* effettui un backup ogni qualvolta viene richiamata la procedura di installazione di un'extension, così da prevenire ogni possibile problema durante questa operazione.
- La possibilità che lo strumento gestisca autonomamente i backup ad intervalli regolari di tempo;
- La possibilità di configurare gli accessi e dare credenziali diverse ad utenti diversi: Un *Super Administrator* potrebbe, ad esempio, avere poteri illimitati di

configurazione, mentre un *Administrator* potrebbe effettuare il backup senza però poterne configurare i parametri;

- Gli aggiornamenti software disponibili vengono visualizzati automaticamente sul pannello di controllo principale.

## **9. Rispetto dei requisiti**

Joomla! ha una fama ormai consolidata tra gli addetti ai lavori e non, e si è rivelato un CMS altamente fruibile, intuitivo, solido nella struttura e dinamico nello sviluppo. Il linguaggio PHP col quale è programmato e i pacchetti necessari al suo funzionamento sono i più diffusi e conosciuti. L'installazione è lineare e non richiede più di quindici minuti, nonostante, soprattutto per i principianti, sia necessaria una guida per le operazioni preliminari di configurazione del database.

Essendo uno dei CMS più diffusi, è anche uno dei più ricchi di template e estensioni rispondenti ai criteri di accessibilità vigenti oggi. Già il template preinstallato ha una validazione XHTML e CSS, è curato fin nei minimi dettagli ed è ricco di moduli e componenti che permettono la creazione della quasi totalità del sito, senza dover ricorrere a componenti esterni. Nella scoperta di tutte le potenzialità e della flessibilità di questo strumento, giocano un ruolo fondamentale i dati di esempio, vero sentiero guida per comprendere a fondo le infinite possibilità di configurazione di Joomla!. Ne accorciano indubbiamente la curva di apprendimento.

L'interfaccia è intuitiva e i pannelli di controllo sono organizzati in modo razionale, senza eccedere nella quantità di menù e opzioni: chi vuole farne uso le trova facilmente, ma l'utente principiante non rischia di inciamparci casualmente e può focalizzare l'attenzione sui parametri fondamentali. La stessa razionalità si trova nella struttura a sezioni e categorie, per la gerarchizzazione dei contenuti: semplice da comprendere e gestire, anche nel caso di una mole notevole di materiale. Neo di Joomla! risulta però l'impossibilità di scendere oltre in secondo grado gerarchico. Per ovviare a questo capita di dover creare molte sezioni e categorie e di non riuscire a creare una gerarchia ottimale.

Vero asso nella manica di Joomla! rimangono le estensioni, vere e proprie applicazioni software che possono rispondere alle esigenze più disparate. Il grado di integrazione è invidiabile, e lo stesso vale nel caso di migrazione da una vecchia release del CMS ad una più recente.

L'ormai avvenuta internazionalizzazione di Joomla! fa sì che si possano trovare in rete prodotti nella propria lingua o pacchetti di traduzione praticamente per ogni estensione. Lo stesso vale per le guide al CMS: sia il mercato editoriale che la rete rilasciano guide aggiornate per ogni release di Joomla!, disponibili in molte lingue, tra le quali l'italiano. La comunità italiana infatti, è una delle più attive in Europa e nel mondo, e permette agli utenti con diversi gradi di competenza di trovare informazioni utili e specifiche.

Giovano di questa diffusione i supporti multilingua e i relativi pacchetti di traduzione, sempre fedeli nelle traduzioni di tutti i componenti, e ben organizzati per ottimizzarne l'usabilità.

Joomla! si presenta quindi uno strumento "per tutti", con la predilezione per chi si affaccia per la prima volta alla costruzione di un sito, ma strizzando l'occhio anche all'utente esperto che ha aspettative elevate sul prodotto e sull'ampiezza di utilizzo.



## CAPITOLO 6

### Conclusioni

Ultimate le prove sul campo dei CMS è possibile stilare alcune considerazioni col fine di decretare quale sia la migliore piattaforma testata. Una nota positiva, al termine di questo lavoro, è stata la conferma della bontà dei requisiti fissati. I criteri scelti nel processo di software selection sono stati sufficienti per valutare le caratteristiche e le funzionalità del software, escludendo la necessità di fissarne di nuovi. Si è notata una certa discordanza tra alcuni dei punteggi assegnati e l'effettiva valutazione dei CMS. Questo aspetto era preventivato poiché l'attribuzione dei punteggi si è svolta sulla base dei dati pubblicati nei siti web ufficiali, delle notizie reperite in altri siti specializzati e dai feedback riportati dagli utilizzatori nei forum dedicati. Non sempre la qualità di questo tipo di informazioni è ottima, pertanto è prevedibile una certa incertezza nei dati raccolti, soprattutto nel campo del software dove è probabile che si manifestino problemi di compatibilità o alcuni bug nelle funzionalità software. Alla luce di ciò è possibile che non siano state inserite in short list alcune piattaforme che più di altre avrebbero meritato di essere sottoposte a un'analisi approfondita. In ogni caso l'obiettivo ultimo era di individuare il migliore CMS nel lotto di partenza e certamente l'obiettivo è stato comunque raggiunto. I CMS che si sono meglio comportati in questo studio sono Plone e Drupal. Con queste due piattaforme è stato possibile rispondere in modo più completo alle esigenze che sono sorte nella realizzazione del portale. Entrambi si sono dimostrati completi nelle funzionalità o facilmente personalizzabili con numerosi plug-in. Plone può essere considerato il CMS migliore tra i due per la maggiore integrazione del calendario e degli eventi e la gestione più efficace delle news. È necessario però non dimenticare che Plone sfrutta la web application Zope ed è sviluppato in Python. Questi due aspetti potrebbero costituire un leggero ostacolo nell'attrezzare un server web o un maggior costo se si opta per un servizio di hosting, rendendo preferibile Drupal basato sulla classica coppia Apache e PHP. Gli altri CMS testati si sono dimostrati carenti in molteplici aspetti. La gestione del supporto multilingua si è rivelato lacunoso o addirittura non implementabile a causa di problemi di compatibilità dei moduli aggiuntivi. Altri problemi sono sorti per la mancanza di un'interfaccia correttamente localizzata e della scarsità di supporto ottenuto nel tentativo di affrontare le difficoltà incontrate.

In particolare e107 ha sofferto la scarsità di compatibilità di alcuni addons e la presenza di problemi legati alla comprensione del codice javascript anche da parte dei più comuni browser di navigazione. Xoops non mette a disposizione fin da subito tutti gli strumenti necessari. Nella maggior parte dei casi è sufficiente ricorrere a plug-in aggiuntivi, ma alcuni di questi si sono rivelati non perfettamente funzionanti. Inoltre alcune operazioni come la personalizzazione dell'aspetto sono piuttosto macchinose e poco adatte per un utente alle prime armi. SilverStripe ha dimostrato qualche problema di giovinezza nonostante sulla carta fosse un ottimo prodotto: sono sorti numerosi problemi nell'installazione di alcuni moduli fondamentali per il progetto e certe operazioni non sono ancora disponibili direttamente dall'interfaccia di gestione, costringendo l'amministratore a modificare direttamente i file di configurazione. Se Zope non dovesse costituire un problema nell'attrezzare un server web, allora Plone sarà sicuramente in grado di soddisfare ogni esigenza nella realizzazione di un portale di dipartimento universitario.

## BIBLIOGRAFIA

R. Chimenti, 2008, Costruire siti dinamici con Joomla! 1.5: HOELPI –  
Settembre 2011

Akeeba Backup User's Guide - <https://www.akeebabackup.com/documentation/akeeba-backup-documentation.html>  
Agosto/Settembre 2011

Forum Joomla! Italia – <http://forum.joomla.it>  
Agosto/Settembre 2011

Joom!fish – <http://www.joomfish.net/>  
Settembre 2011

EventList - <http://www.schlu.net/>  
Settembre 2011

JCE Editor - <http://www.joomlacontenteditor.net/>  
Settembre 2011

“Sistemi CMS: il portale d'Ateneo” - R. Rascioni CAIM: Università degli Studi di Macerata –  
<http://webhouse.unimc.it/caim/index.php>  
Agosto/Settembre 2008

Ad2Sell, Un sito che conta [http://www.ad2sell.com/realizzazione\\_sito\\_ottimizzati\\_.art](http://www.ad2sell.com/realizzazione_sito_ottimizzati_.art)  
23 Agosto 2008

Analisi e progettazione del software, Luca Cabibbo – <http://www.dia.uniroma3.it/>  
Settembre 2008

Applicare UML e i pattern. Analisi e progettazione orientata agli oggetti, Larman C., editore  
Pearson Education Italia



Basic.HTML.it, guida Drupal – <http://basic.html.it/guide/leggi/146/guida-drupal/>  
Agosto/Settembre 2008

copesflavio.com, Copes Flavio, Tesi di laurea <http://www.copesflavio.com/blog/tesi-di-laurea/>  
03 Settembre 2008

Drupal – <http://www.drupal.org/>  
Luglio/Agosto/Settembre 2008

Drupal Italia – <http://www.drupalitalia.org/>  
Luglio/Agosto/Settembre 2008

Drupal Tutorial, Tutorial introduttivo all'uso di Drupal 4.7.4, K3K  
Luglio/Agosto/Settembre 2008

Drupalpa, Drupal per la pubblica amministrazione e accessibilità, Anastasis  
<http://drupalpa.anastasis.it/>

Luglio/Agosto/Settembre 2008

e107 Italia – <http://www.e107.org>

Luglio/Agosto/Settembre 2008

Guida Drupal multi sito, TuttoDrupal: installare, configurare e ottimizzare Drupal per le tue esigenze, Tiziana Ratano, promossa da kiwiage  
Luglio/Agosto/Settembre 2008

[http://spazioinwind.libero.it/gianluca\\_affinito/web\\_barriere/pa.htm](http://spazioinwind.libero.it/gianluca_affinito/web_barriere/pa.htm)

20 Agosto 2008

[http://spazioinwind.libero.it/gianluca\\_affinito/web\\_barriere/usabilita.htm](http://spazioinwind.libero.it/gianluca_affinito/web_barriere/usabilita.htm)

20 Agosto 2008

<http://www.cmsopensource.it/demo/portali-cms/e107.html>

Agosto 2008

[http://www.openfacile.it/e107\\_plugins/content/content.php?content](http://www.openfacile.it/e107_plugins/content/content.php?content).

15 Settembre 2008

italiafreeware.byethost4.com , Più di 150 moduli per Drupal

<http://italiafreeware.byethost4.com/?q=node/74>

Agosto 2008

Le Web-barriere, l'accessibilità dei siti della Pubblica Amministrazione, L'usabilità dei siti web

Le Web-barriere, l'accessibilità dei siti della Pubblica Amministrazione, Analisi dei siti della Pubblica Amministrazione centrale

Redomino Labs – <http://redomino.com/it/labs>

Agosto/Settembre 2008

Redomino Open Solutions – [www.redomino.com](http://www.redomino.com)

Agosto/Settembre 2008

Relazione di progetto, Damiano Callegari, Debora Gardin, Silvia Marini, Giuliano Pasqualotto,

Tesi di laurea 28 Agosto 2008 Requisiti e use cases, Maurizio Pizzonia –

<http://www.dia.uniroma3.it/>

Settembre 2008

Requisiti software, Sergio Di Martino – <http://people.na.infn.it/>

Settembre 2008

Scrivere e leggere i requisiti, Paolo Ciancarini – <http://www.cs.unibo.it/>

Settembre 2008

SEO Utilità, Come Ottimizzare un Sito Web – Dove agire e perché

<http://www.seoutilty.com/it/ottimizzazione-sito-web.htm>

20 Agosto 2008

Sito ufficiale Joomla! – <http://www.joomla.org>

Luglio/Agosto/Settembre 2008 – Settembre 2011

Sito ufficiale Joomla! Italia – <http://www.joomlaitalia.com>

Luglio/Agosto/Settembre 2008 – Settembre 2011

Sito ufficiale SilverStripe – <http://www.silverstripe.com>

Luglio/Agosto/Settembre 2008

Sito web ufficiale di Plone – <http://www.plone.org>

Agosto/Settembre 2008

Stefano Silvestrini, 2007, XOOPS creare siti dinamici con facilità “Tutti i segreti, dalla A alla Z, per creare siti dinamici ed ottenere ottimi risultati senza essere degli esperti!”, New York, NY: Stevensword

Usabile.it, usabilità, accessibilità e interaction-design per il web, Maurizio Boscarol, 2000, “Scrivere per il web” – <http://www.usabile.it/032000.htm>

23 Agosto 2008

Wikipedia – L’enciclopedia libera <http://it.wikipedia.org>

Giugno/Luglio/Agosto/Settembre 2008

Wikipedia – The free encyclopedia <http://en.wikipedia.org>

Luglio/Agosto/Settembre 2008

[www.ictv.it/](http://www.ictv.it/)

Settembre 2008

XOOPS Italia – <http://www.xoopsitalia.org/>

Luglio/Agosto/Settembre 2008

Xplaza – <http://www.xplaza.it/>

Luglio/Agosto/Settembre 2008