

Web Content Manager potente e flessibile: **Drupal**



Drupal è uno dei CMS Open Source più noti ed utilizzati. Recentemente è stato scelto anche per il sito della Casa Bianca... vi spieghiamo perché!



Andrea Franceschini

a.franceschini@oltrelinux.com

Studiante di Ingegneria Informatica a Padova, appassionato di informatica a tutto tondo, in particolare di musica, grafica ed elaborazione di immagini e suoni. Programma in C/C++, PHP, varie ed eventuali, fotografa e disegna a tempo perso

Tra i contenuti dei siti ed il web server che li ospita, c'è spesso un sottile strato di applicazioni che serve a facilitare la gestione dei contenuti ed a separarli dalla logica di presentazione. Una volta i siti web si creavano con niente più che semplici file in formato HTML, qualche contenuto multimediale e, se eravamo fortunati, anche fogli di stile (CSS). Questo tipo di contenuti aveva il fondamentale problema di essere totalmente statico, mentre in alcuni casi è utile, o addirittura indispensabile, presentare contenuti dinamici.

Per questo sono stati introdotti software lato server che permettessero da un lato di gestire più comodamente i contenuti, dall'altro di fornire ai client pagine dinamiche. Ad oggi la maggioranza dei siti web che troviamo è dinamica. Un esempio su tutti? I blog: home page dinamica, contenente link agli ultimi post ed a quelli più commentati, agli ultimi commenti, etc.

Nei casi delle applicazioni più semplici non è raro che chi sviluppa tecnicamente il sito web ricorra a piccoli script realizzati ad hoc, ma sempre più spesso si sceglie di fare affidamento a soluzioni già pronte ed ampiamente testate, dedicando il tempo ad altri servizi a valore aggiunto, ad esempio l'integrazione con i software aziendali (CRM, anagrafiche, etc) dei clienti. Per fortuna, le soluzioni di questo tipo nel mondo Open Source non mancano di certo: avrete probabilmente sentito parlare di Joomla, Wordpress o Drupal. Quest'ultimo è proprio il software su cui ci concentreremo in questo articolo.

Un po' di storia

Come tante delle componenti che costituiscono l'infrastruttura di Internet, anche Drupal è nato all'interno di un'Università. Lo sviluppo è infatti iniziato nel 2000 a cura di Dries Buytaert, uno studente che voleva creare un sito con cui comunicare con i suoi amici: doveva inizialmente servire a scambiarsi informazioni, notizie e lo stato della piccola rete wireless che il suo gruppo di amici aveva instaurato tra le aule dell'Università di Antwerp e il loro dormitorio, dal momento che l'Università faceva pagare l'accesso ad Internet.

Dopo la laurea di Dries, il gruppo decise di rendere disponibile via Internet il software che avevano sviluppato, in modo da rimanere in contatto più facilmente: nel Gennaio 2001 nasce Drupal.

Di acqua sotto i ponti ne è passata parecchia e Drupal oggi è molto diverso da quelle prime versioni: attorno al progetto si è formata un'enorme comunità che ha contribuito alla crescita vertiginosa ed alla sempre più ampia adozione di Drupal come piattaforma per una vasta gamma di siti, dai più semplici blog personali a grandi portali di informazione o siti aziendali. La lista completa è su <http://buytaert.net/tag/drupal-sites>.

Installazione

Nonostante Drupal sia fornito come pacchetto da alcune delle maggiori distribuzioni Linux, è generalmente preferibile installarlo manualmente. Le dipendenze prevedono i soliti PHP e MySQL, che potremo installare tramite il sistema di packaging della nostra distribuzione (su Ubuntu vi basterà un `apt-get install lamp-server^`). Un'alternativa è disporre di un sito web già registrato con associato un servizio di hosting di tipo LAMP, ma vi consigliamo comunque di avere un'installazione



locale su cui effettuare tutti i test e le prove.

Supponiamo di avere già il nostro stack LAMP. La document root del web server Apache nel nostro caso è `/var/www/`. Assicuriamoci di avere i permessi per operare al suo interno con l'utente che utilizziamo abitualmente per sviluppare (riquadro 1): in questa directory andremo a mettere i file che saranno elaborati e serviti al browser da Apache.

La seconda cosa di cui dobbiamo occuparci è creare un database vuoto che Drupal riempirà a seconda delle sue necessità. Possiamo farlo attraverso un'interfaccia per la gestione di MySQL, ma dato che sarà probabilmente l'unica operazione che andremo ad eseguire direttamente sul database, possiamo anche avviare un terminale, entrare nella shell interattiva di MySQL con il comando:

```
$ mysql -u root -p
```

ed inserire la password (che dovremmo aver configurato al momento dell'installazione di MySQL). Una volta all'interno del client, eseguiamo:

```
mysql> CREATE DATABASE drupal;
Query OK, 1 row affected (0.09 sec)
```

per creare un database che ospiterà le tabelle del nostro sito. Quindi creiamo un utente non privilegiato per operare sul database. Dalla shell di MySQL eseguiamo:

```
mysql> GRANT ALL ON drupal.* TO 'drupal_user'@'localhost'
> IDENTIFIED BY 'password'
Query OK, 0 row affected (0.00 sec)
```

per creare l'utente `drupal_user` che può connettersi solamente da `localhost` usando `password` come credenziale di accesso (ovviamente ciascuno imposta la password che preferisce) e che ha il permesso di lavorare solo sul database `drupal` che abbiamo appena creato. Infine usciamo con il comando `exit`.

A questo punto dirigiamoci sul sito di Drupal e scarichiamo l'ultima versione del Core, da <http://drupal.org/download>. Esistono numerosi profili, che sono l'equivalente delle distribuzioni per Linux: raccolte personalizzate di plugin pacchettizzati insieme, per raggiungere più velocemente specifici scopi. Per esempio esistono profili pensati per creare siti di notizie, e-commerce, portali e quant'altro. Dato che lo scopo di questo articolo è illustrare i concetti alla base di Drupal, ignoreremo tali offerte e partiremo direttamente dal Core, a cui



R1 - Modificare i permessi

Generalmente le directory che stanno al di fuori della nostra home appartengono ad un utente diverso dal nostro e ad un gruppo in cui potremmo non essere inseriti. Tuttavia sappiamo bene che lavorare come `root` non è mai una buona idea e quindi avremo bisogno di impostare la *Document Root* in modo da potervi accedere in lettura e scrittura usando il nostro utente normale, ma in modo che anche l'utente non privilegiato con cui è in esecuzione Apache possa lavorarci tranquillamente. Quindi, per prima cosa scopriamo quale sia l'utente di Apache con il comando

```
$ ps aux | egrep -i "apache|httpd"
```

che ci restituirà una folta lista di processi, il primo dei quali dovrebbe essere in esecuzione come `root` (l'utente si vede nella prima colonna dell'output) mentre gli altri sono eseguiti, nel mio caso, dall'utente `apache`. Scopriamo a che gruppo è associato l'utente `apache`:

```
$ groups apache
```

che in genere dovrebbe restituire solo il gruppo `apache` (`www-data` nel caso di sistemi Debian). Se andiamo a vedere a chi appartiene la *Document Root*, scopriremo che appartiene proprio all'utente `apache` e al gruppo `apache` (o `www-data`). La soluzione più efficace è aggiungere il nostro utente al gruppo `apache` (o `www-data`):

```
# usermod -a -G apache miutente
```

A questo punto siamo pronti per lavorare.

aggiungeremo qualche modulo che riteniamo particolarmente importante.

Una volta scaricato il Core, che al momento della scrittura è la versione 6.20, andiamo a scompattarlo all'interno della document root di Apache:

```
$ tar xzf drupal-6.19.tar.gz -C /var/www/
```

A questo punto troveremo tutto in `/var/www/drupal-6.20/`. C'è ancora una cosa da fare prima di poter avviare l'installazione, cioè creare un file di configurazione su cui Apache (e quindi Drupal), possa scrivere. Posizioniamoci in `drupal-6.20/sites/default` e creiamo il file con i permessi corretti:

```
$ touch settings.php
```

```
$ chmod 660 settings.php
```

cioè concediamo il permesso di lettura e scrittura all'utente proprie-



tario ed agli utenti che stanno nel gruppo a cui appartiene il file e nessun permesso a tutti gli altri utenti. Facciamo di nuovo riferimento al riquadro 1 per impostare correttamente l'utente e il gruppo proprietari di questo file, usando il comando `chown`, mentre nel riquadro 2 è riportata la procedura per supportare più di un sito con una sola installazione di Drupal.

Finalmente possiamo avviare l'installazione visitando l'URL

<http://localhost/drupal-6.20> (figura 1). La procedura ci propone l'installazione in inglese, oppure ci suggerisce un modo per procedere usando un'altra lingua come predefinita. È sempre possibile installare un'altra lingua in un secondo momento, ma scegliamo di installare subito l'italiano: dirigiamoci all'URL <http://www.drupalitalia.org/download> e scarichiamo la traduzione italiana per la nostra versione. Il pacchetto disponibile al momento della scrittura fa riferimento alla 6.17, ma questo è abbastanza normale: le traduzioni sono sempre un po' indietro rispetto alle release ufficiali e tra la 6.17 e la 6.20 non c'è molta differenza, quindi possiamo usare tranquillamente la traduzione per la versione più vecchia. Una volta scaricata, scompattiamola all'interno della directory di Drupal, quindi con il comando:

```
$ tar xzf drupal-6.17-it-translations.tgz \
-C /var/www/drupal-6.20
```

e ricarichiamo la pagina di installazione: questa volta ci verranno proposte due lingue. Scegliamo l'italiano e facciamo click sul bottone *Select language*. A questo punto la procedura ci chiederà le informazioni sul database che conterrà tutti i dati relativi al nostro sito. Ricordate il nome che abbiamo scelto qualche riga più su? Bene, è quello che dovremo inserire nel campo *Nome database*. Ci verranno inoltre chieste le credenziali con cui collegarsi al database. Infine è possibile impostare alcune opzioni avanzate, ma di norma è bene lasciare perdere se non si sa cosa si sta facendo: andranno bene come sono per la maggioranza degli utenti. Facendo click sul pulsante *Salva e continua* avvieremo finalmente la procedura automatica di installazione e configurazione, al termine della quale ci verrà ricordato di rimuovere i permessi di scrittura al file `settings.php`, cosa che facciamo col comando:

```
$ chmod a-w settings.php
```

La procedura ora ci chiede qualche informazione sul sito che stiamo creando, sull'utente amministratore e riguardo alcune utili impostazioni del server. Scegliamo un nome ed inseriamolo nell'apposito campo. Impostiamo anche un indirizzo e-mail che verrà usato come mittente nei messaggi che il sito potrebbe dover inviare ai suoi utenti. Di seguito scegliamo un nome utente per l'amministratore, inseriamo il suo indirizzo email e scegliamo una password. Infine le impostazioni del server dovrebbero essere già corrette, quindi possiamo sorvolare e fare click sul pulsante *Salva e continua*. Dopo pochi istanti Drupal ci informerà che tutto è pronto e che possiamo visitare il nostro nuovo sito all'URL <http://localhost/drupal-6.20> (figura 2).

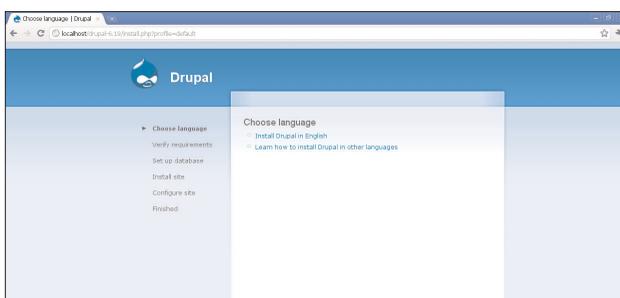
i INFO R2 – Install multi-site

Una delle cose migliori di Drupal è la sua capacità di gestire più siti con un'unica installazione del Core, semplicemente disponendo di apposite directory e opportuni file di configurazione per ciascun sito. Le directory vanno create all'interno di `sites/` presente nella root dell'installazione. Questa directory deve avere il nome corrispondente all'URL che si vuole considerare come base del sito. Se per esempio vogliamo creare un nuovo sito Drupal all'URL <http://www.example.com/sito/>, allora dovremo creare la directory `sites/www.example.com.sito/` contenente un opportuno file `settings.php`. Naturalmente potevamo anche creare una directory `sites/example.com.sito/`, infatti Drupal ha un algoritmo di ricerca che divide in token il nome del dominio ed il percorso di base e li combina a scalare per cercare un file di configurazione. Nel caso accedessimo a <http://www.sub.example.com/sito/>, Drupal effettuerebbe la seguente ricerca:

```
sites/www.sub.example.com.sito/settings.php
sites/sub.example.com.sito/settings.php
sites/example.com.sito/settings.php
sites/www.sub.example.com/settings.php
sites/sub.example.com/settings.php
sites/example.com/settings.php
```

terminando al primo file di configurazione che incontra, oppure a `sites/default/settings.php`

che, se non viene trovato, produce un errore. Con questo meccanismo possiamo, per esempio, gestire più domini di terzo livello, ciascuno con un suo database e con sue impostazioni personalizzate.



Pagina di installazione di Drupal

1



R3 - Drupal 7

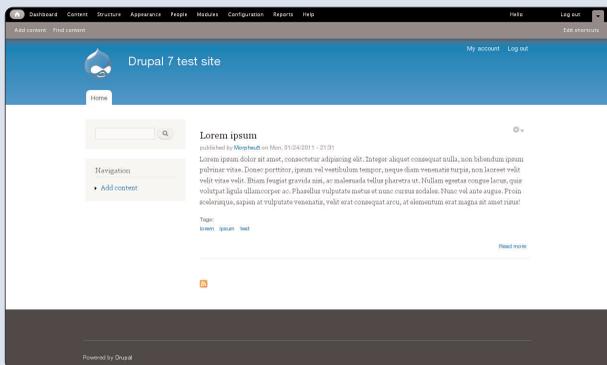
Il 5 gennaio 2011 è stato rilasciato Drupal 7, una major release enormemente attesa dalla comunità, alle prese con un web in continua evoluzione, che promette una grande quantità di migliorie e riorganizzazioni di cui si cominciava a sentire la necessità.

Nonostante l'uscita, abbiamo deciso di trattare la release 6.20 per diversi motivi: è quella attualmente utilizzata in produzione, la più stabile e diffusa e contiene comunque tutti i concetti di base che servono a lavorare anche con la versione 7.0, che ha comunque ancora bisogno di qualche rifinitura, vedremo più avanti i dettagli.

E' comunque importante andare a vedere le novità della nuova release, per capire dove sta andando il CMS open più usato al mondo.

Uno dei principali obiettivi di Drupal 7 è migliorare la cosiddetta UX (*User Experience*), cioè tutti quegli aspetti che coinvolgono direttamente l'utente, dall'interfaccia grafica alle sequenze di operazioni necessarie per raggiungere un determinato scopo. Per esempio, una delle più grosse pecche rilevate dagli esperti di usabilità in Drupal 6 è il fatto che, per default, non adotta una separazione chiara tra le pagine accessibili ai visitatori del sito e quelle accessibili agli amministratori, impiegando addirittura lo stesso tema per entrambe e, sebbene sia possibile scegliere un tema di amministrazione differente, alcune pagine continuano ad essere visualizzate col tema "pubblico". In realtà questo è un sintomo della natura *orientata alla community* di Drupal che prevede, attraverso un complesso sistema di permessi, di concedere ad alcuni utenti non amministratori l'accesso a sezioni altrimenti riservate. Tuttavia, dal momento che è sempre bene chiarire nel modo più evidente possibile che alcune funzionalità possono avere un grosso impatto sul funzionamento del sito, Drupal 7 adotta una migliore separazione tra i due aspetti, anche attraverso la scelta di un tema di default diverso da quello pubblico per le pagine di amministrazione. Oltre a questo miglioramento *cosmetico*, l'intera area di amministrazione è stata riorganizzata, spostando alcune sezioni che comparivano in posti non del tutto corretti ed integrando nel core alcune funzionalità che in Drupal 6 erano fornite da moduli opzionali. Un altro aspetto molto importante è la migrazione verso la libreria PDO (PHP Data Objects), un'interfaccia astratta tra Drupal e il database di supporto in grado di consentire con una singola API l'accesso a molteplici diversi DBMS: MySQL, PostgreSQL, SQLite, Oracle e Microsoft SQL Server.

Tuttavia, nonostante i tanti miglioramenti introdotti da questa release,



Home page di default di un sito Drupal 7

per esempio un netto incremento delle prestazioni e PHP 5.2 come requisito minimo, il passaggio a Drupal 7 è probabilmente ancora prematuro. Si incontrano infatti alcune difficoltà con certe funzionalità e



Il nuovo tema di amministrazione: per default le schermate appaiono in un riquadro sovrapposto alla pagina.

moduli che forse possono non sembrare gravi, ma che rappresentano punti critici per siti e portali che, per esempio, devono essere disponibili in molteplici lingue e tenere conto delle molteplici differenze culturali nei vari aspetti della rappresentazione e fruizione dell'informazione. Molto lavoro e molte parole sono stati spesi per esempio nell'integrazione dei moduli per la traduzione dei contenuti, la localizzazione (i10n) e l'internazionalizzazione (i18n) e tuttavia, alla data del rilascio di Drupal 7, il modulo fondamentale per realizzare un vero portale internazionalizzato (il modulo i18n reperibile all'URL <http://drupal.org/projects/i18n>) era disponibile esclusivamente via GitHub e, al momento della scrittura di questo riquadro, è disponibile in versione 1.0-alpha1 con la rassicurazione da parte dell'autore che la versione stabile sarà rilasciata entro la fine di febbraio.

Indubbiamente il lavoro necessario a realizzare nel migliore dei modi queste, ed altre, funzionalità è enorme, e sarebbe ingeneroso non lodare gli sforzi degli sviluppatori, ma ugualmente non si può fare a meno di chiedersi se forse non sarebbe stato meglio aspettare qualche mese in più (il code freeze è arrivato nel settembre del 2009 ma i lavori sono in corso almeno dal 2007) per fornire una release 7.0 in grado di mettere gli utenti nelle condizioni di migrare i loro siti dalla versione 6 senza rinunciare a niente e godendo di tutte le innegabili migliorie presenti in Drupal 7.

Infine, una nota per quanto riguarda l'aggiornamento dalla versione 6 alla versione 7. La regola generale per questo tipo di aggiornamenti è che non si può fare sostituendo il software e aspettandosi che tutto funzioni. Per prima cosa, l'API è cambiata in molti punti, il che ci costringe ad aggiornare contestualmente tutti i moduli che già usavamo e che, con ogni probabilità, smetterebbero di funzionare in seguito all'aggiornamento. La seconda cosa che è cambiata è il motore che gestisce i temi, quindi dovremo fare attenzione ad usare temi compatibili con la versione 7, ma dei temi ci occuperemo in dettaglio nella prossima puntata.

In ogni caso la regola d'oro è sempre la stessa: fare un backup di tutto quello che può essere danneggiato, cioè il database e tutti i file che stiamo usando. Detto ciò siamo pronti per aprire la guida all'aggiornamento (<http://drupal.org/documentation/upgrade/6/7>) ed avventurarci nel processo, consci comunque del fatto che ci sarà da rimboccarsi le maniche e profondere copiose quantità di olio di gomito.



Primi passi

La pagina principale ci propone una serie di cose da fare una volta creato il sito, ma dato che dobbiamo prima prendere confidenza con Drupal, saltiamo direttamente al punto in cui si creano i contenuti, che vengono chiamati *node*, un termine volutamente generico che, come avremo modo di capire, sottolinea la flessibilità del sistema.

Ciascun nodo può avere diverse *rappresentazioni*. Quella più comune è la modalità web e, in prima istanza, potremmo pensare ai nodi come le pagine del nostro sito. Quindi cominciamo a fare click sulla voce *Crea contenuto* nel menu a sinistra. Drupal ci chiederà di scegliere di che tipo sarà il contenuto che andremo a creare. Nella nostra installazione le scelte possibili sono *pagina* e *storia* ed è presente sotto ciascun tipo una descrizione.

Supponiamo di voler creare la classica pagina *About* che contenga un po' di informazioni sul nostro sito: sceglieremo *pagina* e ci troveremo di fronte ai campi da riempire con i nostri contenuti: il *Titolo* (obbligatorio, segnalato da un asterisco rosso) in cui inseriremo, ad esempio *Informazioni* ed il *Corpo*, in cui andremo ad inserire le informazioni riguardanti il sito, separando i paragrafi con una riga vuota.

Una volta completata questa operazione, scorriamo la pagina fino in

fondo, dove troveremo i pulsanti *Salva* e *Anteprima*. Se lo desideriamo, possiamo vedere un'anteprima di come verrà visualizzato il testo nel nostro sito, ma per questa volta fidiamoci e premiamo *Salva*. Dopo pochi istanti Drupal ci porterà automaticamente alla visualizzazione predefinita del nodo che abbiamo creato, che dovrebbe corrispondere all'URL <http://localhost/drupal-6.20/node/1> (da questo momento in poi, per praticità, ometterò la parte fissa dell'URL, ad esempio in questo caso avrei usato */node/1*). Se abbiamo fatto tutto correttamente, dovremmo trovarci in una situazione simile a quella in [figura 3](#): nel riquadro dedicato al nodo troviamo il titolo, il corpo e due pulsanti speciali, visibili solo agli utenti che hanno i permessi per modificare questo nodo. Facendo click su *Modifica* ci sposteremo su */node/1/edit* che, com'è intuibile, corrisponde alla pagina per modificare il nodo 1. Da qui, cliccando su *Mostra*, torneremo alla visualizzazione pubblica del nodo, cioè */node/1*.

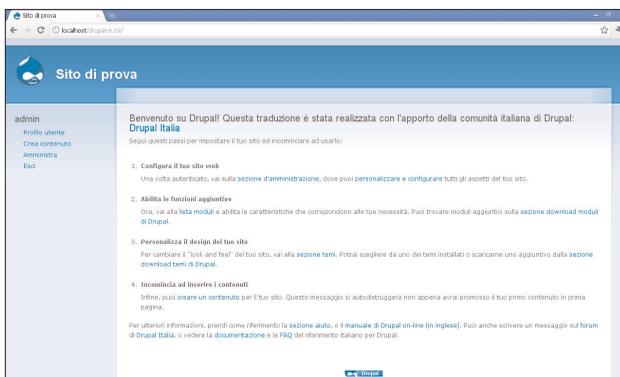
Facciamo un'altra prova: creiamo un nuovo contenuto, questa volta di tipo *storia*, inseriamo ciò che preferiamo nei campi *Titolo* e *Corpo* e salviamo. Drupal ci trasporterà automaticamente all'URL */node/2* e noteremo subito due differenze:

- sotto al titolo compaiono alcune informazioni, cioè la data e l'ora di pubblicazione ed il nome dell'utente che ha creato il contenuto.
- sotto al corpo del nodo, troviamo il link "Aggiungi un commento". Cliccandolo, Drupal ci presenterà un modulo per inserire il nostro commento, più o meno come succede in un blog.

C'è un'altra cosa da notare: se torniamo alla pagina principale (c'è un comodo link *Home* in alto a sinistra, oppure possiamo cliccare sul nome del sito) troveremo che la *storia* che abbiamo appena inserito ha rimpiazzato il messaggio di benvenuto, mentre la *pagina* non è visibile. Questo fatto ci suggerisce che i due tipi di contenuto sono più differenti di quanto ci sia sembrato inizialmente.

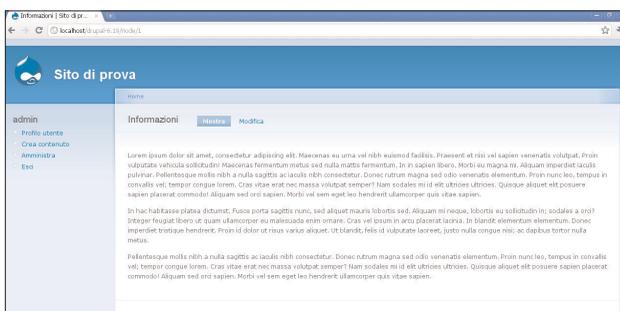
Se andiamo a modificare la *storia* all'URL */node/2/edit* e scorriamo verso il fondo della pagina, noteremo alcune voci che prima abbiamo ignorato: facendo click su *Opzioni di pubblicazione* si aprirà una tendina con tre caselle, tra cui notiamo che *Pubblicato* e *Promosso in prima pagina* sono spuntate. Se andiamo a vedere la stessa voce per la *pagina*, all'URL */node/1/edit*, noteremo che solo la voce *Pubblicato* è spuntata. Se spuntiamo *Promosso in prima pagina*, salviamo e torniamo alla pagina principale, troveremo anche il nodo *Informazioni* elencato (ma privo del link *Aggiungi un commento* che, volendo, potremmo abilitare in modo analogo a come abbiamo promosso il nodo in prima pagina).

Il concetto fondamentale è che possiamo creare diversi tipi di con-



La home del nostro nuovo sito basato su Drupal!

2



La nostra pagina di informazioni

3



tenuto ed assegnarvi diverse impostazioni predefinite, creando così, come in questo caso, due tipi di nodo adatti come pagine del sito (*pagina*) o come notizie (*storia*). La cosa è un po' più complessa di così, ma per il momento accontentiamoci di questo e diamo uno sguardo al resto.

Per avere una panoramica di tutti i nodi presenti nel nostro sito, possiamo fare click sulla voce *Amministra* del menu sulla sinistra, poi su *Gestione dei contenuti* ed infine su *Contenuto*, oppure andare direttamente all'URL `/admin/content/node`. La schermata ci presenta tutti i nodi a cui ci è permesso accedere in lettura, fornendoci un link per modificarli se disponiamo del permesso di modifica. E' disponibile anche un form per la ricerca e l'esecuzione di una stessa operazione su più nodi contemporaneamente.

Tassonomia

Tassonomia è una parola inusuale, ma cattura molto bene l'essenza dell'implementazione di Drupal: la tassonomia è la scienza che si occupa dei modi di classificare oggetti ed è proprio in questo senso che Drupal la intende. Supponiamo per esempio che nel nostro sito vogliamo occuparci di software e di hardware e, in particolare, di schede video e schede audio, oltre naturalmente ad ogni altro pezzo di ferraglia su cui riusciremo a mettere le mani.

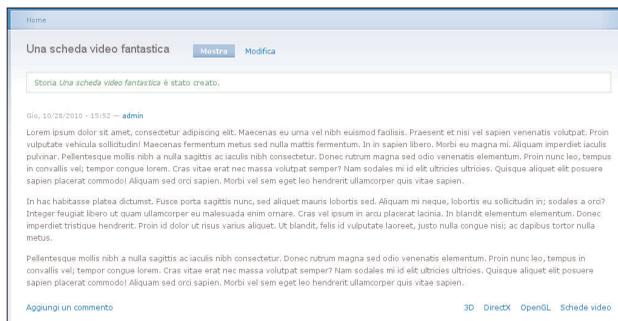
Facciamo click sulla voce *Amministra*, quindi su *Gestione dei contenuti* e infine su *Taxonomy* (per qualche motivo, nel language pack che ho installato questa voce è stata lasciata in inglese), oppure andiamo all'URL `/admin/content/taxonomy`. Notiamo subito che non è definito alcun vocabolario, quindi ne andremo a creare uno facendo click su *Aggiungi vocabolario*. Nella pagina che comparirà inseriremo *Sezione* come *Nome del vocabolario*, con una descrizione a piacere. Il *Testo di aiuto* comparirà agli utenti che sceglieranno i termini all'atto di creare nuovi nodi. Scorrendo la pagina, nella tendina *Tipi di contenuto* abilitiamo solo *Storia*: questo farà sì che i contenuti di questo tipo siano associabili ad una voce di questo vocabolario e lasciamo tutti gli altri campi così come sono. Facciamo infine click su *Salva* e verremo riportati alla pagina che elenca i vocabolari.

A questo punto facciamo click su *Aggiungi termini* sulla riga del vocabolario che abbiamo appena creato ed inseriamo *Software* nel campo *Nome del termine* insieme ad una descrizione a piacere. Ignoriamo per il momento le opzioni avanzate e facciamo click su *Salva*. Drupal ci comunicherà che il termine è stato salvato e ci riporterà di nuovo alla schermata per inserire un altro termine: molto comodo. Approfittiamo per aggiungere il termine *Hardware* con relativa descrizione.



Elenco dei termini all'interno del vocabolario

4



La pagina appena creata

5



Attivare il CCK nella lista dei moduli disponibili

6

A questo punto dobbiamo inserire i due sotto-termini *Schede video* e *Schede audio*: procediamo come già visto per quelli precedenti, però questa volta sotto *Opzioni avanzate* dovremo specificare nella casella *Padri* la voce *Hardware*.

Una volta terminata questa operazione, facciamo click su *Elenco* in cima alla pagina per accedere all'elenco dei termini che abbiamo appena creato (figura 4): ci verrà mostrata anche la struttura gerarchica dei termini, che potremo cambiare semplicemente trascinando le varie voci (facendo click sulla croce a sinistra di ciascun termine). Anche l'ordine di visualizzazione dei termini viene stabilito trascinandoli nella posizione corretta (di default è l'ordine alfabetico). Una volta che abbiamo preso confidenza con la lista dei termini,



riportiamo tutto com'era, oppure semplicemente cambiamo pagina senza salvare le modifiche.

Torniamo per un momento a `/admin/content/taxonomy` (o facciamo click sulla voce *Taxonomy* nel menu a sinistra) e creiamo un nuovo vocabolario: chiamiamolo *Tag*, associamolo al tipo di contenuto *Storia* e nella scheda *Impostazioni* spuntiamo l'omonima voce *Tag*. Questo rende dinamico il vocabolario, così si potrà scegliere più di un termine per ogni contenuto e se ne potranno creare di nuovi, se necessario. Facciamo click su *Salva* e verremo riportati alla lista dei vocabolari che, a sua volta, è ordinabile trascinando i vocabolari. Assicurandoci che il vocabolario *Sezioni* sia il primo della lista.

Creiamo un nuovo contenuto

A questo punto siamo pronti a creare un nuovo contenuto. Facciamo click su *Crea contenuto* dal menu a sinistra e quindi su *Storia* (notiamo l'URL `/node/add/story`) e creiamo il nostro contenuto a piacimento. Nella scheda *Vocabolari* vediamo che sono comparsi i nostri due vocabolari. Dal vocabolario *Sezione* scegliamo per esempio *Schede video* e nella casella di testo *Tag* inseriamo qualche termine separato da virgole, per esempio io ho inserito *3D, OpenGL, DirectX*.

Come dicevamo, i vocabolari di tipo tag sono speciali: ora vediamo che non solo è possibile creare nuovi termini al volo, ma se digitiamo un termine che è già presente nel vocabolario, Drupal se ne accorge e ce lo propone direttamente. Naturalmente la prima volta che usiamo il vocabolario, questo è vuoto, quindi per il momento nessun termine verrà recuperato automaticamente.

Una volta creato il contenuto, facciamo click su *Salva*. Verremo rediretti alla visualizzazione pubblica del nostro nodo (nel mio caso all'URL `/node/3`, figura 5). Notiamo in basso a destra l'elenco dei tag e delle sezioni a cui abbiamo associato il nostro nodo. Facendo click su uno di questi, Drupal ci porterà ad una pagina speciale, chiamata *pagina del termine*, in cui ci vengono elencati tutti i nodi associati a quel particolare termine.

Siamo già operativi: possiamo creare altri contenuti, associandoli ad altre sezioni e tag. È possibile anche associare uno stesso tag a contenuti associati a sezioni diverse, per cui possiamo parlare di un *software* o di una *scheda video* per l'*editing video*: quando visiteremo la pagina del termine editing video vedremo entrambi i contenuti nell'elenco.

Tipi di contenuto

Finora abbiamo usato i due tipi di contenuto che Drupal ci propone al termine dell'installazione, ma questo software è noto per essere uno dei più flessibili CMS del panorama Open Source: vediamo come

creare nuovi tipi di contenuto. Stabiliamo, ad esempio, di voler inserire nel nostro sito delle recensioni a cui associare un voto. Per farlo, dobbiamo ricorrere ad uno di quei moduli talmente essenziali che nessun utente o sviluppatore Drupal che intenda creare un sito di un certo livello si sognerebbe di escludere dalla propria installazione: il *Content Creation Kit* (CCK).

Andiamolo a scaricare dal sito <http://drupal.org/project/cck>, prelevando l'ultima versione stabile (al momento della stesura, 6.x-2.8). Il numero di versione dei moduli si compone della major version del Core (in questo caso la 6) e della versione vera e propria del modulo in questione (in questo caso la 2.8). Installare un modulo è piuttosto semplice: è sufficiente scompattarlo all'interno della directory dei moduli del nostro sito, oppure in quella generale (riquadro 2 per approfondire). Nel nostro caso scegliamo la prima opzione. Il nostro sito si trova in `/var/www/drupal-6.20/sites/default/`, quindi spostiamoci qui dentro e, dato che non è presente, creiamo la directory dove andremo a installare i moduli e scompattiamoci dentro CCK.

```
$ cd /var/www/drupal-6.20/sites/default
$ mkdir modules
$ tar xzf cck-6.x-2.8.tar.gz -C modules/
```

A questo punto dovremmo trovare `cck/` dentro `modules/`: siamo pronti ad attivare il modulo all'interno di Drupal. Torniamo alla schermata di amministrazione e clicchiamo su *Costruzione del sito* e poi su *Moduli*. Scorrendo la pagina ci troveremo di fronte ad una lunga lista di moduli, alcuni spuntati ed altri no. Cerchiamo la sezione CCK (figura 6) ed attiviamo i componenti *Content*, *Number*, *Option Widgets* e *Text*. Infine scorriamo la pagina fino in fondo e facciamo click sul pulsante *Salva configurazione*. Drupal procederà ad installare i moduli al suo interno e, dopo qualche istante, ci riproporrà la lista dei moduli per confermarci il successo dell'operazione. Il modulo che abbiamo scaricato non comprende tutte le traduzioni, però è possibile scaricarle e installarle separatamente, come spiegato nel riquadro 4.

CCK è un modulo estremamente potente e versatile che ci permette di arricchire i nostri tipi di contenuto con campi aggiuntivi. Per farlo, facciamo click su *Gestione dei contenuti* e poi su *Tipi di contenuto*. Qui troviamo elencati i due tipi con cui abbiamo già fatto conoscenza: è possibile aggiungere campi anche ai tipi forniti di default da Drupal, ma noi scegliamo di crearne uno del tutto nuovo facendo click su *Aggiungi tipo di contenuto* in cima alla pagina.

La prima schermata ci chiede di inserire le informazioni fondamentali del tipo di contenuto che stiamo creando: nel campo *Nome* inse-



riamo *Recensione* e in *Tipo* inseriamo una versione del nome ad uso interno. La convenzione vuole che si utilizzino nomi in minuscolo e in inglese, dato che faranno parte degli URL: la scelta è lasciata al gusto personale, io ho inserito *review*. Completiamo il tutto con una descrizione (anche se non è obbligatoria, è utile per guidare gli autori nella scelta del tipo di contenuto più appropriato), ignoriamo il resto delle opzioni e facciamo click su *Salva tipo di contenuto*. Abbiamo creato un tipo di contenuto abbastanza simile al tipo *Storia*, ma manca ancora l'aggiunta più importante. Facciamo click su *Amministra campi* sulla riga corrispondente al tipo *Recensione* e scorriamo la pagina in basso fino a trovare il riquadro *Aggiungi*. Nella riga corrispondente a *Nuovo campo* troviamo una serie di opzioni da configurare: in *Etichetta* inseriamo *Voto*, in *Nome del campo* dovremo scegliere un nome che Drupal userà internamente (valgono le stesse regole viste prima, io ho usato *vote*). Di seguito dovremo scegliere di che tipo sarà il nostro campo, per esempio potremmo decidere che i voti siano numeri da 1 a 5, quindi scegliamo *Intero* dal menu a tendina. Infine dobbiamo decidere con quale widget vogliamo operare la scelta del valore. Dato che abbiamo deciso di usare un insieme ben definito di valori, l'opzione *Caselline/pulsanti radio* potrebbe essere la scelta giusta. A questo

punto usiamo la croce a sinistra per trascinare il nostro nuovo campo tra i campi esistenti, per esempio tra *Corpo* e *Informazioni sull'autore* e facciamo click sul pulsante *Salva* a fondo pagina.

Non è finita qui: Drupal ci presenterà un'ulteriore pagina di impostazioni per definire meglio il comportamento del nostro campo aggiuntivo. Possiamo inserire un testo di aiuto per guidare gli autori, ma la parte interessante viene più avanti: nella scheda *Impostazioni globali* spuntiamo la voce *Richiesto* così da rendere obbligatorio inserire un voto per una recensione, inoltre impostiamo il *Numero di valori* a 1, in modo che non ci sia la possibilità di aggiungere più di un campo voto. Scendendo ancora impostiamo i valori *minimo* e *massimo*, cioè 1 e 5 e, nel riquadro *Valori consentiti*, inseriamo i valori 1, 2, 3, 4 e 5 ciascuno su una riga: può sembrare una ripetizione, ma servono per costruire il widget. Infine facciamo click su *Salva le impostazioni del campo*.

Manca solo un altro dettaglio: abilitare sezioni e tag anche per il nostro nuovo tipo di contenuto. Andiamo su [/admin/content/taxonomy](#) (oppure facciamo click su *Amministra*, poi su *Gestione dei contenuti* e infine su *Taxonomy*) e, per ciascun vocabolario, facciamo click su *modifica vocabolario* e nel riquadro *Tipi di contenuto* mettiamo la spunta anche su *Recensione*.

Siamo finalmente pronti ad inserire una recensione. Andiamo all'URL [/node/add/review](#) (oppure facciamo click su *Crea contenuto* e quindi su *Recensione*) ed inseriamo il contenuto vero e proprio. Sotto al campo *Corpo* che conterrà il testo della recensione, ci sarà il nostro campo *Voto* con cinque caselle tra cui scegliere. Facciamo click su una di queste e salviamo il contenuto. Nella visualizzazione pubblica del nodo, in fondo alla pagina, noteremo che è comparsa la voce *Voto* seguita dal voto che abbiamo impostato (figura 7).

In conclusione

Drupal è un mondo vastissimo e con questo articolo abbiamo appena iniziato a scalfirne la superficie. La cosa migliore da fare è comunque procedere per piccoli passi ed essere sicuri di aver capito bene ogni cosa prima di procedere al passo successivo.

La flessibilità di questo software è tale da accomodare facilmente siti di tipo molto diverso tra loro. Il rovescio della medaglia è una curva di apprendimento piuttosto ripida, ma la documentazione è ben fatta e la community è molto estesa e ben disposta, per esempio possiamo fare una visita ai canali [#drupal](#) e [#drupal-italia](#) sul server IRC di Freenode.

Nel prossimo numero vedremo come personalizzare l'aspetto grafico del sito creando un tema ad-hoc!

i
R4 - Installare una traduzione

In Drupal è sempre possibile esportare un insieme di stringhe in formato PO (uno standard nel campo delle traduzioni per il software) ed importarlo una volta effettuata la traduzione. Tuttavia è molto probabile che moduli di grande diffusione, per esempio CCK, abbiano già una traduzione pronta.

Per installare una traduzione, il primo passo è procurarsela. In questo caso cerchiamo la traduzione italiana del modulo CCK andando all'URL <http://localize.drupal.org/translate/downloads?project=cck> e facendo click in corrispondenza della lingua desiderata e della versione utilizzata, nel nostro caso la 6.x-2.8. Una volta scaricato il file, andiamo nel nostro sito Drupal all'URL [/admin/build/translate/import](#), selezioniamo il file che abbiamo scaricato, selezioniamo la lingua in cui importare le traduzioni e facciamo click sul pulsante *Importa*. In pochi istanti la traduzione sarà installata e pronta all'uso.

Home
La mia prima recensione
Mappa Modifica

Ghi, 10/28/2010 - 18:17 - admin

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat. Duis autem vel eum irure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et justo odio dignissim qui blandit praesent luptatum zzril delenit augue duis dolore te feugiat nulla facilisis. Nam liber tempor cum soluta nobis eleifend option congue nihil imperdiet doming id quod mazim placerat facer possim assum. Typi non habent claritatem insitam; est usus legentis in liis qui facit eorum claritatem. Investigationes demonstraverunt lectores legere me lius quod ii legunt saepius. Claritas est etiam processus dynamicus, qui sequitur mutationem consuetudinum lectorum.

Mirum est notare quam littera gothica, quam nunc putamus parum claram, anteposuerit litterarum formas humanitatis per seacula quarta decima et quinta decima. Eodem modo typi, qui nunc nobis videntur parum clari, fiant solennes in futurum.

Voto:
4

Aggiungi un commento

Editing video
OpenGL
Software

La nostra recensione, con tanto di voto!

