

Creare applicazioni web con WordPress

Questo libro ha lo scopo di aiutarvi a creare con WordPress qualsiasi cosa: siti web, temi, plugin, servizi web e applicazioni web. Abbiamo scelto di concentrarci sulle applicazioni web perché potete considerarle come “super siti” web, che utilizzano tutte le tecniche che tratteremo.

Molti sono convinti che WordPress non sia abbastanza potente o che non sia concepito per la creazione di applicazioni web. Parliamone: costruiamo applicazioni web con WordPress da molti anni e sappiamo che è possibile utilizzarlo per creare applicazioni anche molto scalabili.

In questo capitolo, iniziamo definendo che cosa sono le applicazioni web e spieghiamo perché WordPress è un ottimo ambiente di sviluppo. Descriviamo anche alcune situazioni in cui l'utilizzo di WordPress *non ci sembra* il modo migliore per creare un'applicazione web.

Che cos'è un sito web?

Sapete bene che cos'è un sito web: è un insieme di una o più pagine web, contenenti informazioni di varia natura, cui si accede tramite un browser web.

Che cos'è un'applicazione?

Ci piace la definizione di Wikipedia: “Il termine *applicazione*, in informatica, individua un software installato o una serie di software in fase di esecuzione su un computer, con lo scopo e il risultato di ren-

In questo capitolo

- **Che cos'è un sito web?**
- **Che cos'è un'applicazione?**
- **Che cos'è un'applicazione web?**
- **Perché usare WordPress?**
- **Quando non è il caso di usare WordPress**
- **WordPress come framework di applicazioni**
- **Anatomia di un'applicazione WordPress**

dere possibile una o più funzionalità, servizi o strumenti utili e selezionabili *on demand* dall'utente tramite interfaccia utente, spesso attraverso un'elaborazione a partire da un input fornito dall'utente interagendo con esso”.

Che cos'è un'applicazione web?

Un'applicazione web è semplicemente un'applicazione *eseguita dentro un browser web*. Tenete presente che in alcune applicazioni web, la tecnologia del browser è nascosta. Per esempio, ciò accade quando integrate un'applicazione web in un'applicazione nativa Android o iOS, quando eseguite un sito web come se fosse un'applicazione di Google Chrome o quando eseguite un'applicazione con Adobe AIR. Tuttavia, all'interno di queste applicazioni, esiste ancora quel sottosistema che esegue il codice HTML, CSS e JavaScript. Potete anche considerare un'applicazione web come *un sito web, con in più altre cose di tipo applicativo*. Non esiste un confine netto, dove un sito web diventa automaticamente un'applicazione web. È una di quelle situazioni in cui si valuta caso per caso. Quello che possiamo fare è spiegarvi alcune funzionalità tipiche di un'applicazione web, fornirvi alcuni esempi e poi provare a formulare una definizione abbreviata, in modo da ricordarci su quello di cui stiamo parlando quando nel libro usiamo il termine *applicazione web*.

NOTA

Nelle pagine di questo libro faremo riferimento a SchoolPress. SchoolPress è un'applicazione web che stiamo costruendo per aiutare le scuole e gli educatori, e il cui scopo è quello di gestire gli studenti e i loro curriculum. Tutti gli esempi di codice sono orientati a funzionalità che potrebbero esistere in SchoolPress. Parleremo meglio dei concetti generali su cui si basa SchoolPress più avanti in questo stesso capitolo.

Funzionalità di un'applicazione web

Di seguito riportiamo alcune funzionalità associate, in genere, alle applicazioni web e alle applicazioni in generale. Più funzionalità sono presenti in un sito web, più è appropriato considerarlo un'applicazione web.

NOTA

Molte delle idee presentate in questa parte del capitolo sono influenzate dai seguenti post: *What is a Web Application?* di Dominique Hazaël-Massieux e *What is a Web Application?* di Bob Baxley.

Elementi interattivi

Una tipica interazione con un sito web prevede il caricamento delle pagine, un po' di scrolling e un clic su dei collegamenti ipertestuali. Anche le applicazioni web possono avere collegamenti ipertestuali e scrolling, ma in un'applicazione si tende a usare altri metodi di navigazione.

I siti web contenenti moduli offrono esperienze “a transazione”. Un esempio potrebbe essere un modulo di ricontatto su un sito web o un modulo di richieste in una pagina di un sito web aziendale. I moduli consentono agli utenti di interagire con un sito utilizzando qualcosa di più di un clic.

Le applicazioni web sono dotate di elementi di interfacciamento con l'utente più interattivi, che comprendono barre degli strumenti, elementi *drag-and-drop*, editor di testo avanzati e barre di scorrimento.

Compiti anziché contenuti

Le applicazioni web sono progettate per aiutare l'utente a svolgere compiti specifici. Gli utilizzatori di Google Maps ottengono indicazioni stradali. Gli utilizzatori di Gmail scrivono messaggi di posta elettronica. Gli utilizzatori di Trello gestiscono elenchi. E gli utilizzatori di SchoolPress commentano le lezioni impartite in aula.

Alcune applicazioni si focalizzano sui contenuti. Una sessione d'uso tipica di un'applicazione Facebook o Twitter comporta un tempo di lettura pari a circa il 90%. Tuttavia, le applicazioni presentano una modalità di navigazione dei contenuti diversa dalla tipica esperienza di navigazione nel Web.

Account di accesso

Un account consente all'applicazione web di salvare informazioni sui suoi utenti. Queste informazioni vengono utilizzate per semplificare le principali attività di impostazione dell'applicazione e per garantire un'esperienza persistente. Una volta eseguito l'accesso, gli utenti di SchoolPress possono vedere quali discussioni non sono state lette. Hanno anche un nome-utente, che identifica le loro attività all'interno dell'applicazione.

Le applicazioni web possono anche avere più livelli di utenti. SchoolPress consente agli amministratori di controllare il funzionamento interno dell'applicazione, agli insegnanti di creare le lezioni e agli studenti di partecipare alle discussioni riguardanti le lezioni.

Funzionalità del dispositivo

Le applicazioni web in esecuzione su un telefono possono accedere alla fotocamera, alla rubrica, alla memoria interna e alle informazioni di geolocalizzazione. Le applicazioni web in esecuzione su un computer desktop possono accedere a una webcam o a un disco rigido locale. La stessa applicazione web potrebbe rispondere in modo diverso a seconda del dispositivo impiegato per accedervi. Le applicazioni web si adattano alle diverse dimensioni, risoluzioni e funzionalità dello schermo.

Funzionalità offline

Quando possibile, è bene far funzionare offline le applicazioni web. Ovviamente, l'interattività di Internet è esattamente ciò che definisce la qualifica “web” del termine “applicazioni web”, ma un sito funzionante anche quando il treno è in galleria avrà più l'aspetto di un'applicazione.

Con Gmail potete scrivere messaggi anche offline. Evernote vi consente di redigere note offline e poi di sincronizzarle su Internet quando viene ripristinata la connettività.

Interazioni

Le applicazioni web possono anche collegare insieme più applicazioni web. Un'applicazione web può utilizzare vari servizi web e API per il push e il pull dei dati. Potreste avere un'applicazione web che estrae da Twitter e Foursquare delle informazioni basate sulla posizione (longitudine e latitudine) e poi pubblicarle su Google Maps.

Applicazioni mobile

Da qualche tempo, le applicazioni web, e in particolare quelle per dispositivi mobili, hanno letteralmente preso il volo. Sulla maggior parte dei siti web, le visite da dispositivi mobili hanno ormai superato quelle da computer desktop, come fonte di traffico (fonte: Perficient, Inc.).

Anche solo nel 2012, l'applicazione web per antonomasia somigliava a Basecamp, un software di gestione dei progetti, cui si poteva accedere tramite il browser web dal desktop. Nel 2019, l'applicazione web per eccellenza somiglia più a Twitter, un'applicazione di comunicazione accessibile tramite il proprio telefono, iOS o Android.

Poiché nella maggior parte dei casi, la maggior parte dei propri utenti farà accesso ai vostri siti web e alle vostre applicazioni da un dispositivo mobile, nella progettazione e sviluppo di applicazioni web dobbiamo cercare di favorire una mentalità *mobile first*. Nel Capitolo 16 vedremo come far funzionare nativamente le applicazioni WordPress su dispositivi mobili. Nel Capitolo 4 analizzeremo le basi della progettazione reattiva e della visualizzazione dei siti web su display di qualsiasi formato.

Progressive Web App

Le *Progressive Web App* (PWA) sono siti web che sfruttano le moderne funzionalità dei browser per funzionare come app native per Android, iOS o il desktop. In particolare, quei siti web che per funzionare in modalità offline utilizzano *service worker* dispongono di un *file manifest dell'applicazione web* per definire l'app per il sistema operativo e soddisfano alcuni altri requisiti, in modo che possano essere avviati come applicazioni, direttamente dal browser.

Le PWA sono state promosse dal team di Google Chrome, ma ora sono supportate da iOS e dai browser web più recenti. È in fase di sviluppo un plugin per supportare le funzionalità principali delle PWA nel nucleo interno di WordPress. Potete usare questo plugin per trasformare il vostro sito WordPress in una PWA, e questo è opportuno, ma in realtà programmare una PWA è più una questione di mentalità che di semplice conversione. Come le funzionalità da applicazione web che abbiamo appena descritto, il sito PWA di Google offre una checklist delle funzionalità previste per la maggior parte delle PWA, incluse le seguenti funzionalità di base.

- Il sito opera via HTTPS.
- Le pagine funzionano su tablet e dispositivi mobili.
- Tutti gli URL delle applicazioni vengono caricati e poi usati offline.
- Vengono forniti metadati per la schermata *Aggiungi a Home*.
- Il primo caricamento è veloce anche in 3G.

- Il sito è inter-browser.
- Le transizioni fra pagine non sembrano dipendere dai tempi di reazione della Rete.
- Ogni pagina ha un proprio URL.

Oltre alle funzionalità di base, esiste una checklist di elementi per le PWA “esemplari”, che riguarda l’*esperienza utente* (UX) e le prestazioni. Lo strumento Lighthouse di Google fornisce test e rapporti automatizzati per valutare il soddisfacimento di questi criteri da parte delle PWA. Anche applicazioni completamente native o create per browser possono trarre vantaggio da alcuni dei suggerimenti presenti nella checklist PWA e nei report Lighthouse.

Perché usare WordPress?

Non esiste un unico linguaggio di programmazione o strumento software adatto per tutte le esigenze. Esploreremo in seguito i motivi per cui potreste non voler usare WordPress; ma per ora esaminiamo alcune situazioni in cui l’uso di WordPress per creare un’applicazione web rappresenta una buona scelta.

State già utilizzando WordPress

Se state già utilizzando WordPress per il vostro sito, potreste aver solo bisogno di un semplice plugin in aggiunta alle sue attuali funzionalità. WordPress offre ottimi plugin per l’e-commerce (WooCommerce), i forum (bbPress), i siti per associazioni (Paid Memberships Pro), le funzionalità da social network (BuddyPress) e la *gamification* (BadgeOS). Realizzare un’applicazione basata su un sito WordPress esistente vi farà risparmiare tempo e semplificherà la vita ai vostri utenti. Quindi, se la vostra applicazione è abbastanza semplice, potete creare un plugin personalizzato sul vostro sito WordPress per programmare la funzionalità della vostra applicazione web.

Se il vostro sito funziona bene con WordPress, non date ascolto a chi vi dice di aggiornarlo o di aggiungere determinate funzionalità. Probabilmente non è vero. Non avete bisogno di buttare via tutto il lavoro già svolto in WordPress e quelle che seguono *sono* ottime ragioni per restare in WordPress.

La gestione dei contenuti è semplice

Sviluppato inizialmente come piattaforma per blog, WordPress si è evoluto, nel corso degli anni e con l’introduzione dei *Custom Post Type* (CPT) nella versione 3.0, fino a diventare un sistema di gestione dei contenuti (CMS – *Content Management System*) perfettamente funzionale. Qualsiasi pagina o post può essere modificato dagli amministratori dalla *dashboard*, cui è possibile accedere tramite il browser web. Scopriremo come usare i CPT nel Capitolo 5.

WordPress semplifica l’aggiunta e la modifica di contenuti tramite un editor WYSIWYG (*What You See Is What You Get*), quindi non dovrete ricorrere ai web designer ogni volta che dovrete apportare una semplice modifica al sito. Potete anche creare menu personalizzati ed elementi di navigazione senza modificare in alcun modo il codice.

Se la vostra applicazione web si concentra su frammenti di contenuti (per esempio, la nostra applicazione SchoolPress gestisce compiti assegnati e discussioni), l'API Custom Post Types per WordPress (descritta nel Capitolo 5) semplifica la configurazione e la gestione rapida di questi mini-contenuti personalizzati.

Perfino le applicazioni più orientate alle attività avranno in genere poche pagine per le informazioni, la documentazione e le vendite. WordPress vi permetterà di gestire appieno la vostra applicazione e tutti i suoi contenuti.

La gestione degli utenti è facile e sicura

WordPress offre tutto il necessario per contemplare nel vostro sito sia gli utenti amministrativi sia gli utenti finali.

Oltre a controllare l'accesso ai contenuti, il sistema di ruoli e autorizzazioni di WordPress è estendibile e consente di controllare quali *azioni* sono disponibili per i vari gruppi di utenti. Per esempio, di default, gli utenti con il ruolo di collaboratore potrebbero *aggiungere* nuovi post, ma non *pubblicarli*. Allo stesso modo, potete creare nuovi ruoli e autorizzazioni per gestire chi ha accesso alla vostra funzionalità personalizzata.

Potete utilizzare plugin come Paid Memberships Pro per estendere la gestione base degli utenti e poter nominare membri di diversi livelli e controllare a quali contenuti hanno accesso. Per esempio, potete creare un livello per consentire ai membri paganti l'accesso ai contenuti "premium".

Plugin

Nel repository WordPress sono disponibili oltre 55.000 plugin gratuiti. In vari siti Internet sono disponibili molti altri plugin, sia gratuiti sia a pagamento. Quando vi verrà un'idea per estendere il vostro sito web, è molto probabile che esista già un plugin che vi farà risparmiare tempo e denaro.

Esiste una manciata di plugin indispensabili, che finiamo per usare su quasi tutti i siti e le applicazioni web che costruiamo.

Per la maggior parte dei siti web, vi consigliamo di memorizzare in cache l'output, per avere una navigazione più veloce, di utilizzare strumenti come Google Analytics per il monitoraggio dei visitatori, di creare delle *sitemap* e di modificare le impostazioni delle pagine per facilitare il lavoro dei motori di ricerca (SEO), oltre a una serie di altri compiti comuni.

Esistono molti plugin ben supportati per tutte queste funzioni. Nel libro vi suggeriremo i nostri.

La flessibilità è importante

WordPress è un framework completo, in grado di fare molte cose. Inoltre, WordPress si basa sulle tecnologie PHP, JavaScript e MySQL, quindi in un'applicazione WordPress può essere facilmente inserito tutto ciò che potete creare con PHP/MySQL (ovvero praticamente qualsiasi cosa).

WordPress e PHP/MySQL, in generale, non sono perfetti per ogni attività, ma sono adatti a una vasta gamma di compiti. Il fatto che la piattaforma possa crescere con la vostra

attività può permettervi di applicare e modificare le cose più velocemente. Per esempio, ecco una progressione tipica per un sito web WordPress di una startup.

1. Annunciate la vostra startup con un sito web di una sola pagina.
2. Aggiungete un modulo per raccogliere gli indirizzi e-mail.
3. Aggiungete un blog.
4. Vi concentrate sulla SEO e ottimizzate tutti i contenuti.
5. Pubblicate i post del blog anche su Twitter e Facebook.
6. Aggiungete un forum.
7. Utilizzate il plugin Paid Memberships Pro per prevedere degli accessi a pagamento.
8. Aggiungete moduli, strumenti e funzionalità applicative personalizzate per gli utenti paganti.
9. Aggiornate l'interfaccia utente utilizzando tecniche e framework JavaScript.
10. Modificate il sito e il server per ampliarne la scala.
11. Localizzate il sito/l'app per più paesi e lingue.
12. Aggiungete il supporto Progressive Web App.
13. Avviate i wrapper iOS e Android per l'applicazione.

La cosa bella di questo percorso è che a ogni passo usate sempre lo *stesso database di utenti* e la *stessa piattaforma di sviluppo*.

Aggiornamenti di sicurezza frequenti

Il fatto che WordPress sia utilizzato da milioni di siti lo rende un obiettivo per gli hacker, che cercano di violare la sua sicurezza. Alcuni di quegli attacchi, in passato, hanno avuto successo; tuttavia, gli sviluppatori di WordPress sono pronti ad affrontare le vulnerabilità e a rilasciare costanti aggiornamenti, per risolverle. È come avere milioni di persone che sottopongono a test e sistemano costantemente il vostro software, ed è esattamente quello che sta accadendo.

L'architettura di base di WordPress rende l'applicazione di questi aggiornamenti un'operazione rapida e indolore, che anche gli utenti web più inesperti possono svolgere. Se siete in grado di configurare WordPress e di aggiornarlo all'ultima versione, quando si rende disponibile, avrete fra le mani una piattaforma di sviluppo sicura, più di qualsiasi altro programma disponibile. Discuteremo della sicurezza in modo più dettagliato nel Capitolo 8.

Costo

WordPress è gratuito. PHP è gratuito. MySQL è gratuito. Molti plugin sono gratuiti. I server e l'hosting sono costosi, ma a seconda delle dimensioni della vostra applicazione web e del volume di traffico, la spesa può essere relativamente contenuta. Se avete bisogno di funzionalità personalizzate non offerte da alcun plugin esistente, potreste dover pagare uno sviluppatore per crearle. Se invece siete voi stessi sviluppatori, la cosa vi costerà solo del tempo.

Alcune delle critiche più comuni

Alcune voci estremamente critiche su WordPress potrebbero dire che non sia un buon framework per la creazione di applicazioni web o che non sia neppure un framework. Con tutto il rispetto per queste opinioni, vorremmo approfondire i motivi per cui non siamo d'accordo. Ecco alcune delle critiche più comuni.

“WordPress è solo per i blog”

Molti pensano che, poiché WordPress è nato per gestire blog, sia utile solo per i blog. Dichiarazioni come questa erano vere alcuni anni fa, ma da allora WordPress ha implementato notevoli funzionalità CMS, un aspetto che lo rende utile anche per realizzare altri tipi di siti, focalizzati sui contenuti. Oggi WordPress è il CMS più diffuso, con una quota di mercato superiore al 60% (W3Tech monitora regolarmente l'uso dei vari sistemi di gestione dei contenuti).

La Figura 1.1 mostra una slide della presentazione *State of WordPress* di Matt Mullenweg al WordCamp di San Francisco del 2013. La piramide capovolta a sinistra rappresenta un sito WordPress del 2006: la maggior parte del codice è dedicata all'applicazione di blogging e una parte più piccola al codice CMS e al codice della piattaforma che sorregge il tutto. La piramide a destra rappresenta lo stato attuale della piattaforma WordPress: la maggior parte del codice si trova nella piattaforma stessa, sovrastata da un livello CMS e dall'applicazione di blogging sopra di esso. Oggi WordPress è una piattaforma molto più stabile rispetto anche solo a pochi anni fa.

Potete utilizzare l'API Custom Post Types per adattare l'installazione di WordPress al supporto di altri tipi di contenuti oltre ai post o alle pagine di blog. Analizzeremo in dettaglio questo concetto nel Capitolo 5.

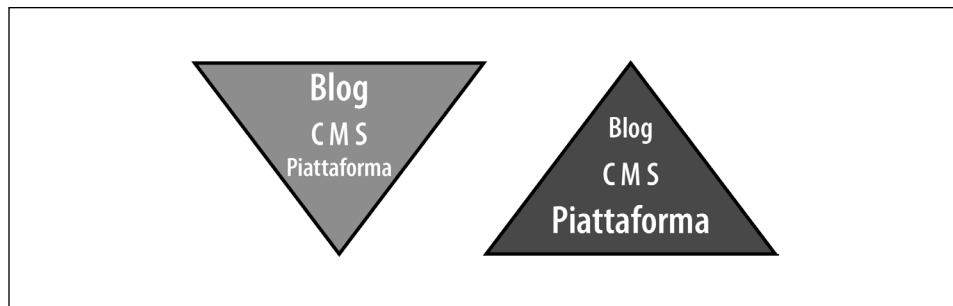


Figura 1.1 Diagrammi della presentazione *State of WordPress* del 2013 di Matt Mullenweg. WordPress non è mai stato più stabile di oggi.

“WordPress è adatto solo a siti di contenuti”

Affermazione simile a “è solo per blog”: qualcuno è convinto che WordPress sia solo per siti di contenuti.

Innanzitutto, anche se WordPress fosse adatto solo a siti e applicazioni basati su contenuti, rappresenterebbe già un gran numero di applicazioni. La schermata iniziale di un qualsiasi telefono probabilmente include molte applicazioni basate su contenuti, come Netflix, Twitter, Facebook, Reddit ed Evernote. Si tratta di applicazioni molto popolari e gestite

da giganti del settore. Ora, non stiamo suggerendo che queste applicazioni *debbano* essere eseguite su WordPress, ma che *potreste* creare un'applicazione analoga a queste, usando come framework proprio WordPress.

In secondo luogo, come esamineremo in dettaglio in questo libro, WordPress è un ottimo framework per la creazione di applicazioni web interattive. La funzionalità principale per utilizzare WordPress come framework è l'API Plugin, che consente di agganciarsi al funzionamento di WordPress e di modificarne il comportamento. Non solo potete utilizzare le migliaia di plugin disponibili nel repository di WordPress e su Internet, ma potete anche utilizzare l'API Plugin per scrivere nuovi plugin e fare in modo che WordPress faccia tutto ciò che è possibile fare in PHP/MySQL.

“WordPress non è adatto alle estensioni di scala”

Molti sostengono che un'installazione standard di WordPress in esecuzione su un host di fascia bassa operi in modo molto rallentato o che si arresti in modo anomalo sotto un carico pesante, e quindi giungono alla conclusione che WordPress sia adatto solo a piccoli sistemi, con piccoli carichi.

E quando abbiamo suggerito di creare un sito come Facebook con WordPress, qualcuno ha giustamente deriso l'idea.

NOTA

Se avete intenzione di creare un'applicazione “Facebook”, questo non è il libro che fa per voi. Ma prima chiedete al vostro direttore tecnico quanta parte del budget multimiliardario aziendale intende investire per la vostra applicazione e indicategli anche quanti e quali tecnici avete bisogno di sottrarre a Google e Amazon per creare la vostra piattaforma personalizzata.

In realtà, molti siti a elevato traffico funzionano in WordPress. WordPress.com adotta lo stesso software di base di ogni altro sito WordPress ed è uno dei siti web più trafficati al mondo.

A mano a mano che l'applicazione aumenterà di dimensioni, dovrete aggiornare e sostituire i suoi componenti per soddisfare tale ampliamento. I problemi di estensione di scala di WordPress sono gli stessi di qualsiasi altra applicazione: memorizzazione cache di pagine e dati, gestione più rapida delle chiamate al database e miglioramento delle prestazioni di rete. Grandi siti come WordPress.com, TechCrunch e i blog del *New York Times* sono stati realizzati in WordPress. Analogamente, si applica a WordPress anche la maggior parte delle lezioni apprese dal ridimensionamento delle applicazioni PHP/MySQL. Descriveremo in dettaglio l'estensione di scala delle applicazioni WordPress nel Capitolo 14.

“WordPress non è sicuro”

Come qualsiasi prodotto open source, quando si utilizza WordPress occorre scendere a compromessi in termini di sicurezza.

Da un lato, poiché WordPress è così popolare, sarà l'obiettivo degli hacker, che cercano continuamente nuovi varchi nella sua sicurezza. E poiché il codice è open source, questi varchi saranno più facili da scoprire.

D'altra parte, poiché WordPress è open source, non appena verrà scoperto un varco, lo saprete subito, e immediatamente qualcuno lo risolverà per voi.

Ci sentiamo più sicuri sapendo che siamo in molti a cercare di sfruttare WordPress e molti anche a cercare di rendere WordPress più sicuro contro quei varchi. Non crediamo nella “sicurezza basata sulla segretezza”, se non come misura aggiuntiva. Preferiremmo che i varchi nella sicurezza del nostro software vengano rivelati, invece di essere scoperti nel momento peggiore possibile.

Il Capitolo 8 analizza in dettaglio i problemi di sicurezza, incluso un elenco dei metodi migliori per rafforzare l'installazione di WordPress e per programmare in modo sicuro.

“I plugin di WordPress sono una schifezza”

L'API Plugin di WordPress e le migliaia di plugin che sono stati sviluppati sono l'ingrediente segreto e, a nostro avviso, la ragione principale per cui WordPress si è diffuso così tanto e ha riscosso questo successo come piattaforma per la realizzazione di siti web.

Alcuni diranno: “Certo, ha migliaia di plugin, ma *sono tutti* uno schifo”. OK, per alcuni di essi, questo è vero.

Ma ci sono molti plugin che sicuramente non fanno affatto “schifo”: tra questi, AppPresser, sviluppato fra gli altri da Brian Messenlehner. Se intendete usare WordPress per gestire dei contenuti scritti o il vostro sito di e-commerce, il plugin e la piattaforma AppPresser sono il modo più veloce per realizzare quei contenuti o per trasformarli in un'applicazione mobile.

Anche Paid Memberships Pro, sviluppato fra gli altri da Jason Coleman, non è esattamente una schifezza. L'uso di Paid Memberships Pro per gestire la fatturazione e la gestione dei membri vi consentirà di concentrare gli sforzi di sviluppo sulle funzionalità principali della vostra applicazione anziché su come dotare il vostro sito di un gateway a pagamento. Molti plugin fanno qualcosa di molto semplice (per esempio, nascondere la barra di amministrazione ai non amministratori), funzionano esattamente come pubblicizzato e non hanno davvero nemmeno la possibilità di essere una schifezza.

Temi e plugin presenti nel repository WordPress.org sono attentamente controllati da volontari, in termini sia di sicurezza sia di qualità del codice. Il processo di revisione dei temi di WordPress.org è notoriamente più rigoroso e completo rispetto ad altri. Il progetto Tide sta lavorando per aggiungere test automatici ai repository di plugin e temi, il che si traduce in plugin e aggiornamenti di qualità superiore, con un rilevamento più rapido dei problemi di compatibilità e sicurezza.

Perfino i plugin scadenti possono essere sistemati, riscritti o riutilizzati per funzionare meglio. A volte, è più facile riscrivere del tutto un plugin difettoso, invece di sistemarlo. Tuttavia, vi farà comunque comodo avere una base di partenza, per evitare di riscrivere tutto da soli.

Nessuno vi obbliga a utilizzare i plugin di WordPress senza esaminarli. Se state creando un'applicazione web impegnativa, è il caso di controllare il codice presente nei plugin utilizzati, modificarlo per soddisfare i vostri standard e procedere con lo sviluppo.

Quando non è il caso di usare WordPress

WordPress non è la soluzione ideale per ogni applicazione. Questa parte del capitolo descrive alcuni casi in cui è meglio *non* utilizzare WordPress per creare un'applicazione.

Se pensate di cedere in licenza o di vendere la tecnologia del vostro sito

WordPress utilizza la licenza GNU (General Public License), versione 2 (GPLv2), che impone restrizioni sulla distribuzione del software creato. Vale a dire, non potete limitare ciò che le persone faranno con il vostro software, una volta che lo venderete o lo distribuirete.

Questo è un argomento complicato, ma l'idea di fondo è che vendendo o regalando l'accesso alla vostra applicazione, non dovrete preoccuparvi della licenza GPLv2. Ma se intendete vendere o distribuire il codice sorgente dell'applicazione, al codice si applicherà la licenza GPLv2.

Per esempio, se ospitiamo SchoolPress sui nostri server e vendiamo l'account necessario per accedere all'applicazione, questa operazione non rientra nell'attività di distribuzione e la licenza GPLv2 non avrà alcun impatto sulla nostra attività.

Al contrario, se intendiamo consentire alle scuole di installare il software sui loro server, dovremo condividere con loro il codice sorgente. Ciò equivarrebbe a una distribuzione. I nostri clienti sarebbero così in grado di fornire legittimamente il nostro codice sorgente gratuitamente a chi vogliono, anche se inizialmente hanno pagato quel software. La licenza GPLv2, non ci consente di limitare ciò che gli utenti possono fare con il nostro codice sorgente, dopo averlo scaricato.

Se un'altra piattaforma può portarvi allo stesso risultato più velocemente

Se avete un team di sviluppatori Ruby, dovrete usare Ruby per creare la vostra applicazione web. Se sapete che esiste una piattaforma, un framework o un bundle che include l'80% delle funzionalità necessarie per la vostra applicazione web e WordPress non offre nulla di simile, probabilmente dovrete utilizzare quell'altra piattaforma.

Se la flessibilità non è importante

Una delle maggiori caratteristiche di un sito WordPress è la possibilità di modificare rapidamente parti del sito web per adattarlo alle vostre esigenze. Per esempio, se i "Mi piace" di Facebook smettono di indirizzare traffico, potete disinstallare il plugin Facebook Connect e installarne uno per Pinterest.

Generalmente, l'aggiornamento del tema o il cambio di plugin su un sito WordPress sarà più rapido rispetto allo sviluppo di nuove funzionalità partendo da zero su un'altra piattaforma. Tuttavia, per i casi in cui l'ottimizzazione e le prestazioni sono più importanti della possibilità di aggiornare rapidamente l'applicazione, probabilmente fareste bene a programmare un'applicazione nativa o a programmarla direttamente in PHP.

Se la vostra applicazione svolge *una sola operazione semplice*, probabilmente vorrete crearla a un livello inferiore. Per esempio, il server di licenze Paid Memberships Pro è fondamentalmente un singolo file JSON di informazioni aggiuntive, con un piccolo script che controlla le chiavi di licenza e fornisce i file compressi. Jason ha creato quel server di licenze direttamente in PHP, con un'ottima gestione cache. Il server delle licenze funziona su uno Sproplet DigitalOcean da 10 dollari al mese e serve oltre 80.000 siti con Paid Memberships Pro.

Allo stesso modo, se avete risorse “da Facebook”, potete permettervi di creare tutto a mano e utilizzare compilatori PHP-to-C personalizzati e componenti iOS nativi per rosicchiare qualche millisecondo dai tempi di caricamento del vostro sito web e delle relative applicazioni.

Se l'applicazione deve funzionare in tempo reale

Un potenziale svantaggio di WordPress, di cui parleremo in seguito, è la sua dipendenza dalla tipica architettura del server web. Nella configurazione tipica di WordPress, un utente visita un URL, che comunica con un server web (come Apache) via HTTP, avvia uno script PHP per generare la pagina e quindi restituisce l'intera pagina all'utente.

Potete migliorare le prestazioni di questa architettura utilizzando tecniche di caching e/o impostazioni ottimizzate del server. Potete rendere asincrono WordPress usando chiamate Ajax o accedendo al database con client alternativi. Tuttavia, se l'applicazione deve funzionare in tempo reale ed è completamente asincrona (per esempio è un'applicazione simile a una chat room o a un gioco multiplayer), pensateci due volte prima di utilizzare WordPress.

Molti sviluppatori WordPress, tra cui Matt Mullenweg, fondatore e leader spirituale di WordPress, conoscono bene questa limitazione. Sempre più funzionalità di WordPress vengono spostate in JavaScript, dove il calcolo può essere trasferito al browser, e possono essere utilizzate strutture come REACT per creare esperienze altamente interattive. Il nuovo editor Gutenberg, aggiunto in WordPress 5.0, è il miglior esempio di questa mossa ed è indicativo dell'evoluzione in corso, ma per ora dovrete affrontare un'ardua battaglia per cercare di far funzionare WordPress in modo asincrono con le stesse prestazioni di un'applicazione nativa o di qualcosa realizzato interamente in Node.js o con altre tecnologie specifiche per applicazioni in tempo reale.

WordPress come framework di applicazioni

I sistemi di gestione dei contenuti come WordPress, Drupal e Joomla spesso sono esclusi dalla discussione sui framework, ma in realtà WordPress (in particolare) è davvero eccezionale in quelle attività che dovrebbero proprio essere di competenza dei framework: creare rapidamente applicazioni.

In pochi minuti, potete configurare WordPress e ottenere un'applicazione completamente funzionale, dotata di registrazioni degli utenti, gestione delle sessioni, gestione dei contenuti e una dashboard per monitorare le attività del sito.

Le varie API, gli oggetti e le funzioni di supporto trattati in questo libro vi consentiranno di programmare più rapidamente applicazioni complesse, senza che dobbiate preoccuparvi della loro integrazione con i sistemi di livello inferiore.

La Figura 1.2 mostra il triangolo destro della presentazione *State of WordPress* del 2013 di Mullenweg, che raffigura una piattaforma WordPress stabile, con il livello CMS incorporato nella parte superiore e un'applicazione di blogging costruita sul livello CMS. La realtà è che la maggior parte dell'attuale base di *WordPress Codex* supporta la piattaforma applicativa sottostante. Potete considerare una versione di WordPress come un framework di applicazioni, in bundle con un'applicazione di blogging di esempio.

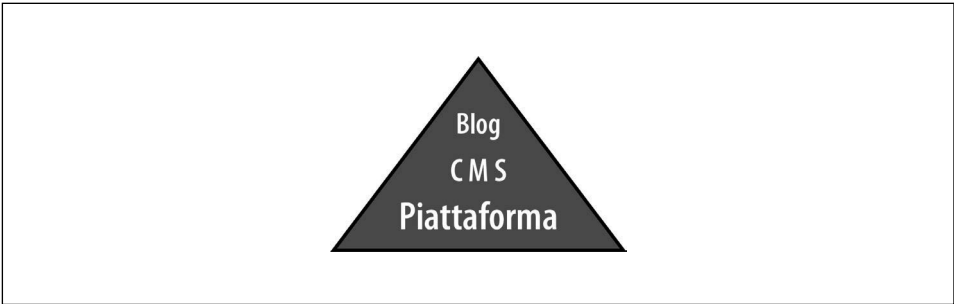


Figura 1.2 La piattaforma WordPress.

WordPress contro i framework Model-View-Controller

Model-View-Controller (MVC) è un modello di progettazione comunemente utilizzato da molti framework di sviluppo software. I principali vantaggi dell'utilizzo di un'architettura MVC sono il riutilizzo del codice e la separazione degli ambiti (SoC – *Separation of Concerns*). WordPress non utilizza un'architettura MVC, ma a suo modo incoraggia sia il riutilizzo del codice sia la separazione degli ambiti.

Spiegheremo brevemente l'architettura MVC e come si riferisce a un processo di sviluppo in WordPress. Se conoscete i framework basati su MVC, questa parte del capitolo dovrebbe farvi capire come affrontare lo sviluppo di WordPress in modo simile.

La Figura 1.3 descrive una tipica applicazione basata sul modello MVC. L'utente finale utilizza un *controller*, che manipola lo stato e i dati dell'applicazione tramite un *modello*, il quale a sua volta aggiorna una *vista*, mostrata all'utente. Per esempio, in un'applicazione di blogging, un utente potrebbe guardare la pagina dei post recenti (che è una *vista*). L'utente farebbe anche clic sul titolo di un post, che lo porterebbe a un nuovo URL (che è un *controller*) il quale carica i dati del post (in un *modello*) e visualizza il singolo post (un'altra *vista*).

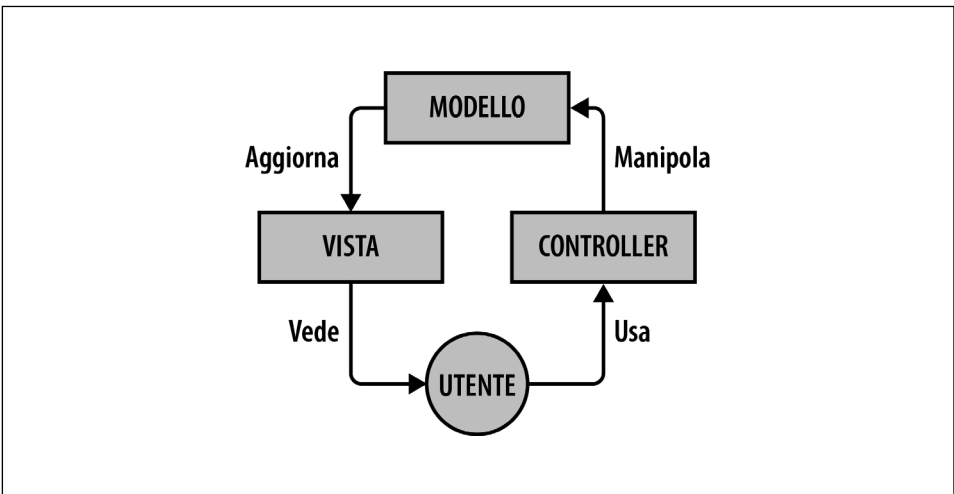


Figura 1.3 Come funziona l'architettura MVC.

L'architettura MVC supporta il riutilizzo del codice sorgente, consentendo l'interazione tra modelli, viste e controller. Per esempio, sia la vista dei post recenti sia la vista dei singoli post potrebbero utilizzare lo stesso modello di visualizzazione dei dati dei post. Gli stessi modelli potrebbero essere utilizzati nel front-end, per visualizzare i post, e nel back-end, per modificarli. L'architettura MVC supporta la SoC consentendo ai designer di focalizzare la loro attenzione sulle viste, mentre i programmatori focalizzano la loro attenzione sui modelli.

Potreste provare a usare un'architettura MVC in WordPress. Esistono numerosi progetti dedicati; tuttavia, pensiamo che provare ad applicare l'architettura MVC a WordPress potrebbe causare problemi, a meno che il core di WordPress non supporti ufficialmente MVC. Sugeriamo, invece, di seguire il "WordPress Way", come indicato in questo libro. Tuttavia, se siete interessati, il plugin WP MVC è in fase di sviluppo attivo e aiuta a utilizzare un framework MVC per creare plugin WordPress. Se non volete o non avete bisogno di usare appieno l'architettura MVC, esistono diversi modi per mappare un processo MVC su WordPress.

Modelli = plugin

In un framework MVC, il codice sorgente che memorizza le strutture dati e la logica operativa è contenuto nei modelli. È qui che i programmatori trascorreranno la maggior parte del loro tempo.

In WordPress, i plugin sono il posto giusto per memorizzare le nuove strutture di dati, le logiche operative complesse e le definizioni dei tipi di post personalizzati.

Questo confronto fra modelli si declina in vari modi. Innanzitutto, molti plugin aggiungono funzionalità "da vista" e contengono elementi progettuali: prendete per esempio un qualsiasi plugin che aggiunge un widget da utilizzare nelle pagine. In secondo luogo, anche i moduli e gli altri componenti progettuali utilizzati nella dashboard di WordPress sono generalmente gestiti tramite plugin.

Un modo per rendere più chiara la SoC quando si aggiungono componenti "tipo vista" ai plugin di WordPress consiste nel creare una cartella di *modelli* o di *pagine* e inserirvi il codice sorgente front-end. È pratica comune consentire ai modelli di sovrascrivere il modello utilizzato dal plugin. Per esempio, quando utilizzate WordPress con il plugin Paid Memberships Pro, potete collocare una cartella `paid-memberships-pro/pages` nel tema attivo, che prevarrà sui template di default delle pagine. Parleremo di questa tecnica di overriding dei modelli di plugin nel Capitolo 4.

Viste = temi

In un framework MVC, il codice sorgente per visualizzare i dati per l'utente va scritto nelle viste. È qui che i designer e gli sviluppatori front-end trascorreranno la maggior parte del loro tempo.

In WordPress, i temi sono il posto giusto per archiviare il codice e la logica del modello. Ancora una volta, il confronto qui non è uno-a-uno, ma "viste = temi" è un buon punto di partenza.

Controller = caricatore di modelli

In un framework MVC, il codice sorgente per elaborare l'input dell'utente (sotto forma di URL o di dati `$_GET` o `$_POST`) e per decidere quali modelli e viste utilizzare per gestire una richiesta, viene collocato nei controller. Il codice del controller è generalmente curato da un programmatore e spesso viene creato una volta e poi dimenticato. La parte più importante della programmazione, in un'applicazione MVC avviene nei modelli e nelle viste. Ciononostante, i controller rappresentano una parte importante del funzionamento di un'applicazione.

In WordPress, tutte le richieste di pagine (a meno che stiano accedendo a un file `.html` in cache) vengono elaborate tramite il file `index.php` e poi da WordPress, secondo la *gerarchia dei template*. È il caricatore di modelli a determinare quale file, nel modello, deve essere utilizzato per visualizzare la pagina all'utente finale. Per esempio, se utilizzare `search.php` per mostrare i risultati della ricerca, `single.php` per mostrare un singolo post e così via.

Il comportamento di default può essere ulteriormente personalizzato tramite l'API `WP_Rewrite` (descritta nel Capitolo 7) e altri hook e filtri. Potete trovare ulteriori informazioni sulla gerarchia dei template nel *WordPress Theme Handbook*; affrontiamo in modo più approfondito la gerarchia dei template nel Capitolo 4.

Per una migliore comprensione di come funzionano i framework MVC, il framework PHP Yii offre un'ottima risorsa, che spiega in dettaglio come utilizzare al meglio la sua architettura MVC.

NOTA

Yii è un framework PHP basato sull'architettura MVC. Altri framework PHP, come Laravel, sono più popolari tra gli sviluppatori WordPress e la community PHP in generale, ma la documentazione relativa a MVC sul sito web Yii è particolarmente ben fatta.

Per ulteriori informazioni sullo sviluppo di applicazioni web utilizzando WordPress come un framework, vi basta proseguire nella lettura di questo libro.

Anatomia di un'applicazione WordPress

In questa parte del capitolo, descriviamo l'applicazione che abbiamo creato e che accompagna questo libro: *SchoolPress*. Tratteremo le funzionalità previste per *SchoolPress*, come funziona e chi lo utilizzerà e, cosa più importante per questo libro, come è stata creata in WordPress ogni parte dell'applicazione.

Non preoccupatevi se non comprenderete appieno alcuni dei termini seguenti. Nei prossimi capitoli esamineremo in modo più dettagliato tutto ciò che viene introdotto qui. Quando possibile, rimandiamo al capitolo corrispondente.

NOTA

Questo manuale non vuole essere un libro per ricreare l'applicazione *SchoolPress* o una guida dettagliata alla sua realizzazione. Semplicemente, quando la cosa ha senso, usiamo *SchoolPress* nei nostri esempi di codice in modo da rendere più concreto il contesto di ogni singolo esempio.

Che cos'è SchoolPress?

SchoolPress è un'applicazione web che consente agli insegnanti di interagire con facilità con i propri studenti, al di fuori dell'ambito della classe. Gli insegnanti possono creare lezioni e invitare i loro studenti. Ogni classe ha un forum per le discussioni mirate e anche un sistema più strutturato in cui gli insegnanti possono pubblicare *compiti assegnati* e fare in modo che gli studenti svolgano il proprio lavoro.

Potete trovare l'applicazione sul sito web di SchoolPress. Il codice sorgente dell'applicazione è disponibile nel repository GitHub di SchoolPress.

SchoolPress funziona su una rete Multisite WordPress

SchoolPress esegue una versione multisito di WordPress. Il sito principale ospita degli account gratuiti, dove gli insegnanti possono iscriversi e iniziare a gestire le loro lezioni. Contiene anche tutte le informazioni di marketing dei singoli siti scolastici della rete, inclusa la pagina per iscriversi con un abbonamento a pagamento.

Le scuole possono creare un sottodominio univoco, che ospiterà le lezioni per i loro insegnanti. Questa configurazione offre un controllo e dei report più accurati per tutte le classi nell'intera scuola. I dettagli sull'uso di una rete Multisite con WordPress sono esposti nel Capitolo 12.

Il modello operativo di SchoolPress

SchoolPress utilizza i plugin Paid Memberships Pro, PMPro Register Helper e PMPro Network per personalizzare il processo di registrazione e accettare pagamenti con carta di credito dalle scuole che si iscrivono.

Le scuole possono acquistare un sottodominio unico per la loro scuola versando una quota annuale. Nessun altro utente di SchoolPress paga per l'accesso. Quando gli amministratori della scuola si iscrivono, possono specificare il nome della scuola per il loro sottodominio `<nostrascuola>.schoolpress.me`. Per la scuola verrà creato un nuovo sito dedicato, dal quale gli amministratori hanno accesso a una versione molto lineare della dashboard di WordPress per il loro sito.

L'amministratore della scuola invita quindi gli insegnanti nel sistema. Anche gli insegnanti possono presentare un invito di partecipazione alla scuola, che deve essere approvato dall'amministratore della scuola. Gli insegnanti possono anche invitare gli studenti alle lezioni che creano. Gli studenti possono anche presentare un invito a un corso, che deve essere approvato dall'insegnante.

Gli insegnanti possono anche iscriversi gratuitamente per ospitare le loro lezioni su `schoolpress.me`. Le pagine ospitate su questo sottodominio possono prevedere annunci o altri schemi di monetizzazione. I dettagli sull'impostazione e-commerce con WordPress sono discussi nel Capitolo 15.

Livelli di iscrizione e ruoli utente

Agli insegnanti viene assegnato un livello di iscrizione *Teacher* (tramite Paid Memberships Pro) e un ruolo personalizzato chiamato "Teacher", che consente loro di creare e

modificare le lezioni, di moderare le discussioni nei forum delle loro classi e di creare e gestire i compiti assegnati per le loro lezioni.

Gli insegnanti non hanno accesso alla dashboard di WordPress. Creano e gestiscono le classi e i compiti assegnati attraverso moduli front-end creati appositamente.

Agli studenti viene assegnato in WordPress un livello di iscrizione *Student* e il ruolo di default “Subscriber”. Gli studenti hanno accesso unicamente per visualizzare e partecipare alle lezioni cui sono invitati dai loro insegnanti. I dettagli sui ruoli e le possibilità degli utenti sono spiegati nel Capitolo 6; il Capitolo 15 tratta l’uso dei livelli di iscrizione per controllare l’accesso.

Le classi sono gruppi BuddyPress

Quando gli insegnanti creano delle “classi”, in realtà creano gruppi *BuddyPress* e invitano i loro studenti a quel gruppo. Grazie a BuddyPress otteniamo anche i forum di classe, la messaggistica interna e un bel modo per organizzare gli utenti.

I forum di discussione per le classi sono gestiti dal plugin bbPress. Viene generato un nuovo forum per ogni classe e BuddyPress gestisce ogni accesso ai forum. I dettagli su come sfruttare dei plugin di terze parti come BuddyPress e bbPress sono discussi nel Capitolo 3.

I compiti assegnati sono un CPT

I compiti assegnati sono un CPT (*Custom Post Type*) che impiega un modulo di presentazione front-end utilizzato dagli insegnanti per pubblicare nuovi compiti. I compiti assegnati sono esattamente come normali post di blog di WordPress, con un titolo, un contenuto e dei file allegati. L’insegnante che pubblica il compito assegnato è l’autore del post.

NOTA

WordPress ha tipi di post integrati, come i post e le pagine, e tassonomie integrate, come le categorie e i tag. Per SchoolPress, creiamo i nostri nuovi CPT e le nostre nuove tassonomie. Troverete ulteriori informazioni sulla creazione di Custom Post Type e tassonomie nel Capitolo 5.

Gli invii sono un sotto-CPT dei compiti assegnati

Gli studenti possono pubblicare commenti su un compito assegnato e possono anche scegliere di pubblicare la loro esposizione ufficiale per il compito tramite un altro modulo sul front-end.

Anche le esposizioni degli studenti, come i compiti assegnati dagli insegnanti, sono CPT. Le esposizioni sono collegate tramite il campo `post_parent` all’ID del relativo compito assegnato. Gli studenti possono pubblicare contenuti testuali e anche aggiungere uno o più allegati alla loro esposizione di risposta.

I semestri sono una tassonomia del CPT della classe

Nel CPT del gruppo/classe è impostata una tassonomia personalizzata denominata *Semester*. Gli amministratori delle scuole possono aggiungere ai loro siti nuovi semestri. Per esempio, potrebbero creare un semestre “autunno 2019” e gli insegnanti potrebbero far riferimento a questo semestre nella creazione delle loro lezioni. Gli studenti possono così sfogliare facilmente un elenco di tutte le lezioni dell’autunno 2019.

I dipartimenti sono una tassonomia del CPT della classe

Per il CPT del gruppo/classe viene inoltre impostata una tassonomia personalizzata denominata *Department*. È disponibile anche come elenco a discesa per gli insegnanti nella creazione delle loro lezioni e consente agli studenti di sfogliare l’elenco delle lezioni, suddivise per dipartimento.

SchoolPress ha un plugin personalizzato principale

Dietro le quinte, gli elementi personalizzati dell’applicazione SchoolPress sono controllati da un singolo plugin personalizzato, chiamato SchoolPress. Questo plugin principale include le definizioni per i vari CPT, le tassonomie e i ruoli utente. Inoltre include il codice sorgente per intervenire sui plugin di terze parti utilizzati da SchoolPress, come Paid Memberships Pro e BuddyPress.

Il plugin principale contiene anche le classi degli amministratori scolastici, degli insegnanti e degli studenti, che estendono la classe `WP_User`, e le classi per le classi (di scuola), i compiti assegnati e gli elaborati che includono la classe `WP_Post`. Queste classi (PHP) ci consentono di organizzare il codice a oggetti, il che aiuta a controllare il modo in cui collaborano le varie personalizzazioni e ogni futura estensione del nostro codice. Sono classi piacevoli da utilizzare e che permettono di scrivere il codice mostrato nell’Esempio 1.1.

Esempio 1.1 Possibili eventi di login di utenti.

```
if($class->isTeacher($current_user))
{
    //this is the teacher, show them teacher stuff
    //...
}
elseif($class->isStudent($current_user))
{
    //this is a student in the class, show them student stuff
    //...
}
elseif(is_user_logged_in())
{
    //not logged in, send them to the login form with a redirect back here
    wp_redirect(wp_login_url(get_permalink($class->ID)));
    exit;
}
else
```

```
{  
    //not a member of this class, redirect them to the invite page  
    wp_redirect($class->invite_url); exit;  
}
```

La creazione di plugin personalizzati è descritta nel Capitolo 3; l'estensione della classe `WP_User` nel Capitolo 6.

SchoolPress utilizza altri plugin personalizzati

Occasionalmente, ci capiterà di sviluppare codice sorgente per una particolare applicazione che però sarebbe utile anche in altri progetti. Se il codice può essere separato adeguatamente, in modo da poter essere eseguito anche al di fuori del contesto dell'applicazione corrente e del plugin principale, può essere integrato in un plugin personalizzato a parte. Un esempio potrebbe essere il plugin *Force First and Last Name as Display Name* che era un requisito per questo progetto. Non richiedeva l'esecuzione del codice sorgente del plugin principale e può essere utile anche per altri siti WordPress, del tutto estranei al contesto dell'applicazione SchoolPress. Perciò abbiamo creato un plugin separato per questa funzionalità e lo conserviamo nel repository *WordPress.org*, in modo che possa essere utile a tutti.

SchoolPress utilizza il tema Memberlite

Il sito principale `schoolpress.me` funziona su un tema figlio di Memberlite, personalizzato. Se un amministratore scolastico si iscrive a un sottodominio premium, può scegliere tra una varietà di temi figli di Memberlite; può anche modificare il colore, il tipo di carattere e il logo del tema, per adattarlo al marchio della scuola/classe. Inoltre, tutti i temi utilizzano un design reattivo (*responsive design*), che garantisce che il sito abbia un bell'aspetto sia su display mobili e tablet sia su desktop.

Il codice nel tema Memberlite è strettamente limitato alla programmazione relativa alla visualizzazione. Il codice del tema ovviamente include del codice HTML e CSS per il layout del sito, ma contiene anche alcune semplici logiche che si integrano con il plugin principale di SchoolPress (come il precedente codice ramificato). Tuttavia, ogni parte di codice sorgente che manipola i Custom Post Type o i ruoli utente o comporta molti calcoli viene delegata al plugin SchoolPress.

Ora che abbiamo stabilito che cos'è un'applicazione, che abbiamo analizzato il motivo per cui potreste volerne creare una con WordPress, che abbiamo presentato il "WordPress Way" per separare gli ambiti e che abbiamo descritto ad alto livello la nostra applicazione di esempio SchoolPress, analizziamo il core di WordPress: che cosa offre e come funziona.