

Indice

Indice	1
Introduzione.....	3
Finalità	4
Scelte di implementazione	5
SCORM.....	6
Lo SCORM nell'e-Learning	6
Compatibilità dei LO con lo SCORM	6
Prima versione.....	8
Software di base.....	8
Server.....	8
Realizzazione	9
Conclusioni	12
Seconda versione.....	14
Obbiettivi	14
Software base.....	14
Vantaggi di Moodle	16
Struttura di moodle	17
Un introduzione della struttura principale di moodle	17
Tipi di utenti	18
I plugin più importanti	19

Temi.....	22
Struttura di un tema	22
File e cartelle.....	23
CSS	24
Layout.....	24
Personalizzazione della piattaforma	27
Plugin utilizzati	27
Blocchi	27
Altri plugin.....	28
Tema scelto.....	28
Selezione finale.....	32
Personalizzazioni del tema	34
Operazioni base sui css.....	35
Creazione del tasto Back	37
Eliminazione dello switch delle colonne.....	38
Creazione di una pagina di login	41
Modifica tasto My Home.....	43
Modifica pulsante Calendario	44
Risoluzione problemi	46
Considerazioni finali.....	47
Sitografia.....	48

Introduzione

Durante il tirocinio ho collaborato con una società di consulenza, Modo Network S.r.l., che opera nel settore del retail. Il progetto iniziale prevedeva la realizzazione di una web application per sondaggi aziendali e condivisione del materiale, la prima idea fu quella di sviluppare il software da zero, ma successivamente sono state trovate molte alternative opensource valide ed ampiamente testate, quindi ci si è orientati alla personalizzazione di quest'ultime. Ma durante l'utilizzo di questi software e la stesura delle linee guida del progetto le features utili per una azienda sono cresciute molto di numero e si è dato molto peso ad alcune nuove possibilità che si presentarono, su tutte quella di poter formare i dipendenti di una azienda tramite l'e-learning. Proprio per questo fine è nato egrigori (www.egrigori.it), un progetto dell'azienda che si propone come creatore di piattaforme per l'apprendimento e di rielaborare i vecchi corsi in aula per renderli fruibili via web.

Questo elaborato si propone di sintetizzare le scelte e le motivazioni che hanno portato alla realizzazione del software per la formazione che attualmente egrigori propone sul mercato, presentandone il percorso evolutivo, motivando ed argomentando le varie scelte prese in considerazione e descrivendo brevemente gli strumenti ed i software utilizzati. Non intende essere una relazione completa del tirocinio, sarebbe stato inutile fare un mero elenco di tutte le modifiche apportate al codice originale, ne una documentazione sulla personalizzazione di un'installazione di Moodle (il software open source utilizzato), dato che esiste già sia ed è esaustiva. Non sono state descritte nemmeno le operazioni di amministrazione che hanno permesso di settare e popolare la piattaforma, sono relative alla manutenzione e amministrazione, non allo sviluppo. In conclusione è un esempio pratico di come operare per ottenere una piattaforma che garantisca tutte le funzionalità per l'insegnamento online e come farla su misura per le proprie esigenze.

Finalità

Realizzare una piattaforma completamente web based per:

- Autenticazione degli utenti
- Distribuzione del materiale informativo dall'azienda ai propri dipendenti
- Condivisione del materiale tra i dipendenti
- Erogazione dei corsi di formazione via web
- Possibilità per l'azienda di effettuare sondaggi
- Possibilità per i dipendenti di contattarsi tra loro e di condividere l'esperienza
- Completa tracciabilità del percorso del singolo dipendente all'interno della piattaforma

Il tutto deve inoltre essere clusterizzabile o definibile per ogni singolo dipendente. Inoltre deve essere accessibile anche da dispositivi mobile e ampliabile in caso di nuove features.

Scelte di implementazione

L'applicativo rientra nella categoria dei Learning Management System(LMS) da wikipedia:

Un learning management system (LMS) è la piattaforma applicativa (o insieme di programmi) che permette l'erogazione dei corsi in modalità e-learning al fine di contribuire a realizzare le finalità previste dal progetto educativo dell'istituzione proponente. Il learning management system presidia la distribuzione dei corsi on-line, l'iscrizione degli studenti, il tracciamento delle attività on-line.

Per la realizzazione sono sorte subito 2 alternative:

- Implementazione completa
- Personalizzazione di un software esistente

L'implementazione del software senza nessuna base è sicuramente molto onerosa, comporta una conoscenza dei sistemi di sicurezza e del trattamento dei dati personali non indifferente. La conoscenza completa del codice ne consentirebbe una più facile modificabilità, ma il periodo di prova e di correzione degli errori sarebbe molto lungo e oneroso. La poca esperienza nello sviluppo di tali applicativi non garantirebbe la qualità del prodotto finale. Inoltre bisognerebbe gestirne gli aggiornamenti e la manutenzione. In sostanza il grado di rischio sarebbe molto alto e le risorse da investire numerose.

La seconda opzione sembra molto più vantaggiosa, esistono numerosi applicative che realizzano tutte le funzionalità richieste da un LMS. Il problema sorge nella personalizzazione e nel tipo di licenza che possiede. Inoltre è fondamentale una valutazione dell'affidabilità, della documentazione esistente e, per i software opensource, dell'attività e professionalità della community che segue il progetto.

Contro la prima ipotesi non c'erano solo i tempi ristretti di realizzazione, il budget limitato e l'inesperienza ma a mio avviso anche l'idea di fondo sbagliata. Durante la mia breve esperienza lavorativa ho analizzato numerosi preventivi relativi ad applicativi e siti web, tutti includono nell'offerta la realizzazione di CMS o complementi a costi molto elevati e con canoni annui considerevoli quando la rete offre soluzioni open source

aggiornate e performanti. I software presenti offrono molte più funzionalità rispetto a quelle richieste e richiedono solo personalizzazioni. La loro affidabilità è testata e gli aggiornamenti dei componenti sono costanti.

Indipendentemente dalla scelta la piattaforma dovrà essere pienamente SCORM compatibile, dato che è il protocollo di tracciamento degli elementi formativi utilizzato dai tool più diffusi. Qui sotto descriverò brevemente in cosa consiste.

SCORM

Lo SCORM – "Shareable Content Object Reference Model" (Modello di Riferimento per gli Oggetti di Contenuto Condivisibili) è tecnicamente un "modello virtuale" (reference model), cioè una raccolta di specifiche tecniche che consente, primariamente, lo scambio di contenuti digitali in maniera indipendente dalla piattaforma.

Al momento attuale le ultime specifiche dello standard sono relative alla versione 1.3 (detto anche SCORM 2004) anche se il più utilizzato rimane ancora lo scorm 1.2. La specifica SCORM 2004 ha subito diverse revisioni e miglioramenti. La revisione più recente è la quarta (4th edition).

Lo SCORM nell'e-Learning

Lo SCORM definisce, nell'e-Learning, le specifiche relative al riutilizzo, tracciamento e catalogazione degli oggetti didattici (learning object), i "mattoni elementari" con i quali vengono strutturati i corsi. La piattaforma di e-learning ha solo il compito di dialogare con l'oggetto, interpretando i messaggi che gli vengono passati. Ciò è possibile in quanto SCORM definisce al suo interno le caratteristiche che dovrebbero essere supportate dal LMS. La compatibilità della piattaforma si rende necessaria solamente per "capire la lingua" dell'oggetto e, se necessario, per potergli rispondere.

Compatibilità dei LO con lo SCORM

Per essere compatibile con lo standard SCORM, ogni Learning Object (LO) deve avere le seguenti caratteristiche:

- Essere catalogabile attraverso dei metadati (campi descrittivi predefiniti) in modo da poter essere indicizzato e ricercato all'interno dell'LMS. I campi descrittivi

richiesti sono molti, non tutti obbligatori. Viene ad esempio richiesto l'autore, la versione, la data dell'ultima modifica fino ad arrivare ai vari livelli di aggregazione tra i vari oggetti. Il tutto viene archiviato nella sezione in un file chiamato `imsmanifest.xml`.

- Poter dialogare con l'LMS in cui è incluso, passandogli dei dati utili al tracciamento dell'attività del discente, ad esempio il tempo passato all'interno di una certa lezione, i risultati conseguiti in un test e i vincoli previsti per passare all'oggetto successivo. Il dialogo avviene attraverso dei dati che passano dal LO all'LMS e dall'LMS al LO. Il linguaggio con cui si comunica è il Javascript che viene interpretato da una API (Application programming interface) che funge da ponte tra i dati che i due elementi (LMS e LO) si trasmettono.
- Essere riusabile: l'oggetto deve essere trasportabile su qualsiasi piattaforma compatibile senza perdere di funzionalità. Questo principio è alla base dello standard in quanto, rispettando le direttive di costruzione, l'oggetto e la piattaforma non devono essere modificati per attivare le funzionalità di tracciamento e catalogazione.
- Un materiale didattico SCORM è un file con estensione `.zip`, oppure `.pif`, che contiene all'interno diverse sezioni relative alla struttura, alla descrizione con metadati ed al suo funzionamento all'interno di un LMS. Lo SCORM quindi non specifica un formato di file che possa rappresentare l'oggetto didattico: qualsiasi formato può essere incluso in un pacchetto SCORM, a seconda del fatto che l'oggetto sia preposto a comunicare con un LMS o ad essere un oggetto di un supporto che non comunica con la piattaforma di e-learning. Se questo oggetto è programmato per comunicare con la piattaforma prende il nome di SCO, se invece è un oggetto di supporto prende il nome di ASSET.
- Il linguaggio con cui l'oggetto SCO comunica con la piattaforma di e-Learning è il Javascript; tra i formati più comuni per costruire SCO possiamo citare l'HTML, Flash, Java o altri formati più chiusi o meno diffusi.

Prima versione

Software di base

Il primo software ad essere stato utilizzato come base è stato Docebo Community Edition 4.0.4 della Docebo S.r.l.. Docebo(in origine SpaghettiLearning) è un LMS sviluppato da un gruppo di programmatori italiani. Fino a Docebo 4.0.4 il codice è stato rilasciato in due versioni Enterprise e Community. La versione community, rilasciata con licenza GPL, sperava di creare una comunità di programmatori italiani che ne seguisse lo sviluppo e procedesse nell'ampliamento del codice, il che sembra aver riscosso poco successo. Dalla successiva versione rilasciata agli inizi di quest'anno Docebo ha mantenuto aggiornata solo la versione commerciale eliminando la parte community dal suo sito, lo sviluppo del seguito open source è ora seguito da e-learning community (<http://www.elearningcommunity.net>).

Docebo è scritto in PHP, HTML e JavaScript ed utilizza numerosissime volte le librerie Yui(<http://yuilibrary.com/>) tutte le informazioni sono salvate in un database MySql e implementa correttamente lo standard SCORM.

Server

Il server che ospitava la piattaforma era un VPS Aruba con OS CentOS. Il software installato necessario al funzionamento di Docebo è:

- Server software: Apache/2.2.3 (CentOS)
- PHP Version: 5.3.10
 - Safe mode: OFF
 - register_global: OFF
 - upload_max_fsize: 128M
 - post_max_size: 128M
- Versione Client Mysql: : 5.5.23
- Supporto mime_content_type(): ON

- Ldap: ON

fondamentale per il funzionamento il disabilitare dalle opzioni di PHP(il file php.ini) la safe_mode e il settaggio dei limiti all'upload di file.

Realizzazione

La versione community rispondeva alla totalità delle richieste iniziali, anche se peccava notevolmente di accessibilità, le pagine contenevano molte informazioni inutili e i comandi fondamentali erano poco evidenti. La piattaforma si presentava molto fredda e poco user friendly come si può osservare dalle immagini qui sotto.



Figura 1 Pagina originale di login

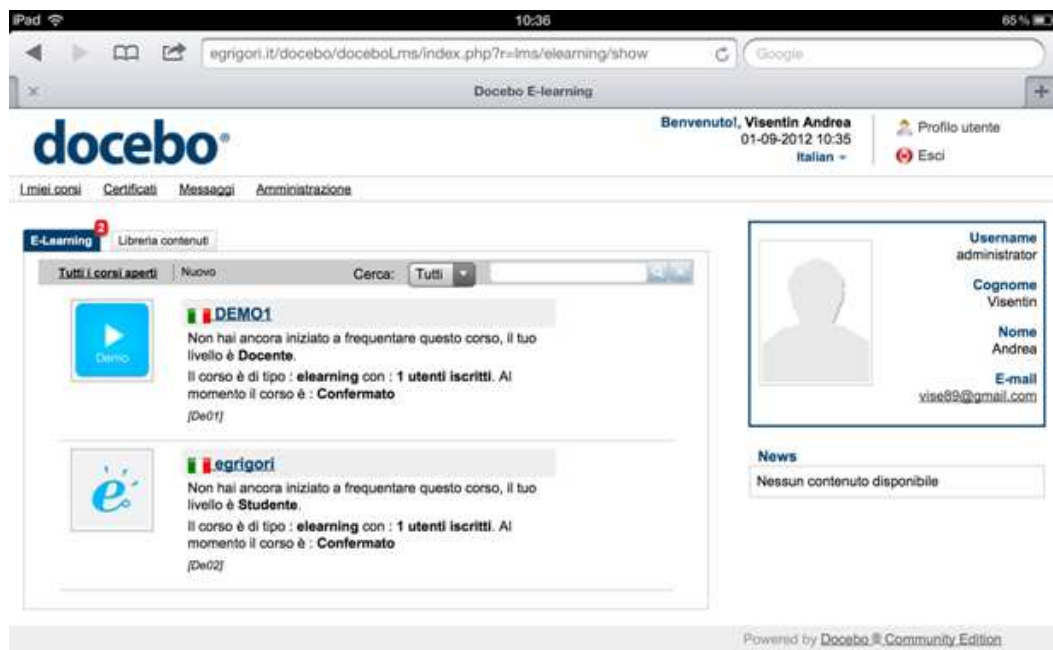


Figura 2 Homepage

Per risolvere il problema abbiamo lavorato sulla sforbiciatura delle informazioni e delle possibilità per l'utente, senza limitarne le potenzialità. L'obiettivo era rendere il percorso alle singole informazioni il più rapido e semplice possibile, anche perché spesso la popolazione aziendale è molto eterogenea dal punto di vista dell'alfabetizzazione informatica. Quindi ho eliminato il menu sopra l'elenco dei corsi e nascosto tutte le informazioni superflue sulla descrizione in home dei corsi, inoltre ho reso cliccabile anche il logo del corso e non solo il nome. Il problema di queste modifiche è che non è stato possibile effettuarle all'interno dei file del tema o del layout, ma ho dovuto farle nel core applicativo; il che è sbagliato dato che in caso di aggiornamenti o correzioni di bug queste modifiche vengono perse irrimediabilmente, un grosso difetto della modularità di Docebo.

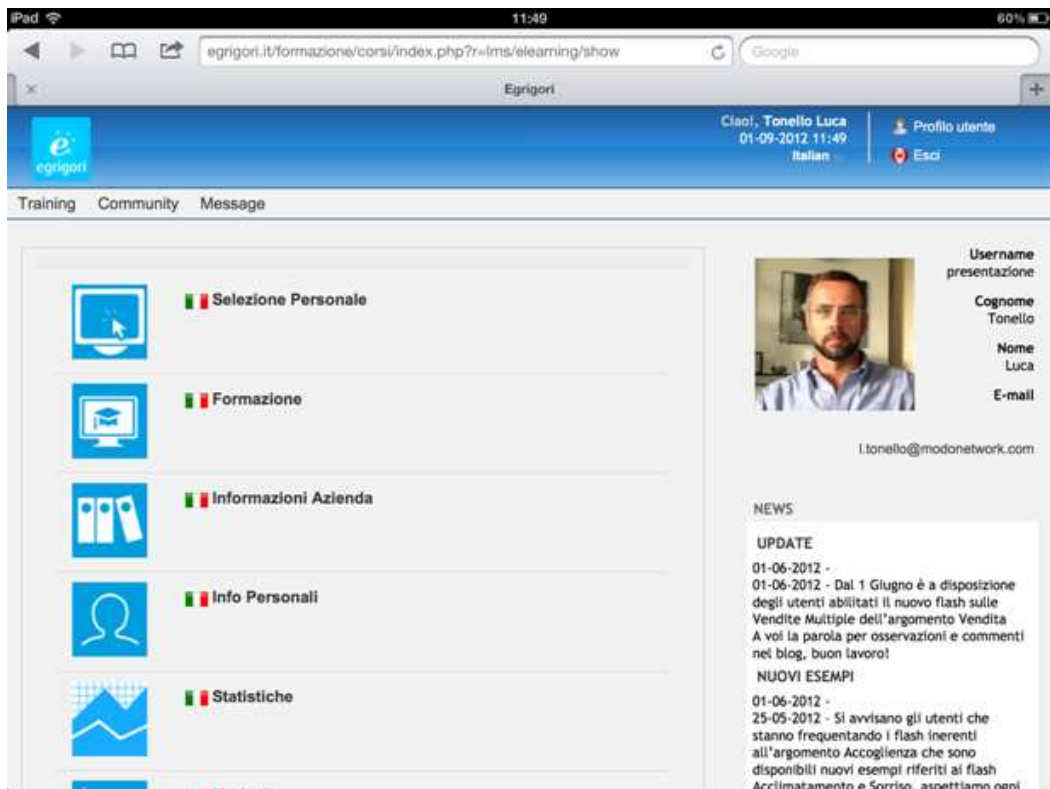


Figura 3 Homepage semplificata

Il passo successivo era rendere la piattaforma graficamente più accattivante, e la collaborazione con un web designer ha aiutato molto. Sono state evidenziate le aree chiave ed aumentata la dimensione di pulsanti ed informazioni. La grafica della pagina di login è stata completamente cambiata per rendere più accogliente l'ingresso dell'utente nella piattaforma.

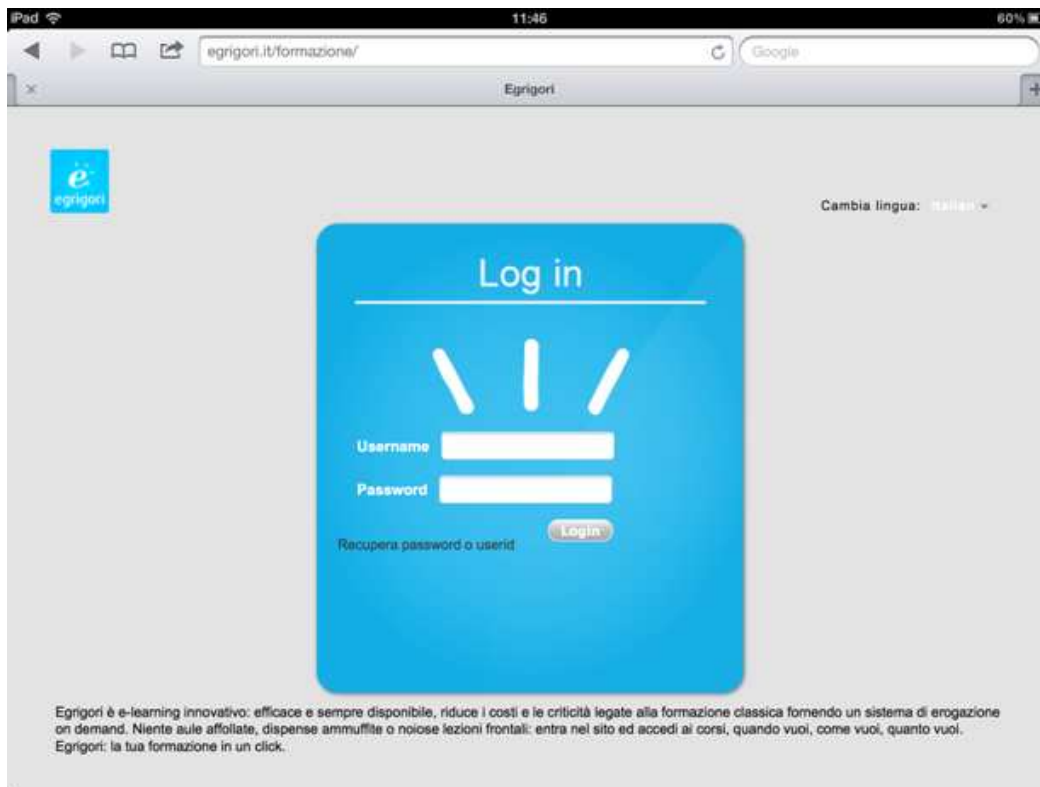


Figura 4 Nuova pagina di login

Ho lavorato anche all'integrazione della piattaforma con un CMS in questo caso erano disponibili moduli configurabili che ne garantivano lo scambio di utenti con Drupal. La componente funzionava solo unidirezionalmente e sono state necessarie molte modifiche per correggere l'iscrizione degli utenti ai corsi.

Conclusioni

Nonostante l'ottimo risultato ottenuto abbiamo deciso di non proseguire oltre nell'utilizzo e nel miglioramento della piattaforma Docebo per i seguenti motivi:

- La mancanza di una documentazione decente: nella composizione di una pagina interagiscono anche centinaia di file e l'unico modo per rintracciare l'origine è partire da un frammento di codice html di output e rintracciare il singolo file che lo genera facendo il cammino inverso. Oppure cercando il tag della traduzione in tutti i file della piattaforma e valutando dove viene usato. Per effettuare queste operazioni ho sempre usato Agent Ransack un software di ricerca, utilizza query personalizzabili e cerca all'interno di file e cartelle. Il processo risulta spesso molto lungo ed la registrazione delle modifiche complicata.

- La struttura del template mal organizzata: la struttura prevede che il template agisca quasi totalmente sulla parte grafica, non toccando minimamente il layout delle pagine, in questo modo qualsiasi modifica al layout deve essere fatta direttamente sul codice PHP che la genera, rendendo rischiosa la modifica. Distribuendo in troppi file del core di Docebo la modifica del layout è difficilmente tracciabile e trasferire in altre installazioni, inoltre eventuali aggiornamenti (anche di sicurezza) rischiano di sovrascrivere quei file e di perdere le modifiche fatte.
- La mancanza completa di una versione mobile e l'eccessiva complessità nel crearla dato che la piattaforma non supporta temi differenti in caso di dispositivi differenti e che, come detto sopra, il layout è composto da centinaia di file e bisognerebbe modificarli tutti.
- La mancanza di moduli e componenti che diano features aggiuntive.
- I grossi problemi riscontrati utilizzando le api.
- Dal 2012 Docebo ha definitivamente abbandonato il suo progetto community chiudendo forum, quindi successivi aggiornamenti non saranno più disponibili. Una community sta provando a mantenere in vita il ramo open source ma con risultati limitati al momento.

Seconda versione

Obbiettivi

I nuovi dispositivi mobili stanno incrementando notevolmente l'informatizzazione della nostra vita quotidiana, e stiamo assistendo ad un'evoluzione di questi dispositivi molto particolare. Quindi la nuova versione deve prevedere una versione facilmente accessibile da dispositivi mobili.

Mentre i primi smartphone erano telefoni che tentavano di assomigliare ad un computer, si è osservata un'inversione di tendenza dato che i computer cercano di evolvere le proprie interfacce e applicativi sempre più similmente ai dispositivi mobili (basti osservare l'interfaccia del nuovo Windows 8 oppure i chromebook). L'obiettivo principale della seconda versione era di creare una piattaforma il più possibile cross-device, e possibilmente (cosa non esistente attualmente) che la versione mobile sia quasi identica alla versione completa, in modo da dare all'utente una sensazione di uniformità e ridurre il problema dell'adattamento alle interfacce distinte.

Questa linea guida porta ad una selezione accurata delle funzionalità che è necessario mantenere e delle informazioni da visualizzare, per non limitare l'operato di amministratori e docenti è possibile assegnare a loro temi diversi rispetto a quello assegnato agli studenti.

Software base

La seconda versione della piattaforma è un'installazione di un altro software open source utilizzato anche dalla nostra facoltà: Moodle.



Figura 5 Logo di Moodle

Moodle (acronimo di Modular Object-Oriented Dynamic Learning Environment) è una piattaforma web open source per l'e-learning, (Learning Management System), progettata

per permettere agli insegnanti e agli educatori di creare e gestire corsi on-line con ampie possibilità di interazione tra studente e docente. Attualmente viene usato da oltre 66 mila siti di formazione in 215 nazioni per un totale di 6 milioni di corsi e 59 milioni di utenti. Moodle è scritto in PHP e JavaScript. E' rilasciato con licenza open(GNU). Il copyright è di proprietà dei singoli programmatori che lo hanno sviluppato, non è di una singola entità, il progetto è comunque gestito dalla Moodle Pty Ltd, con sede in Australia, di proprietà del fondatore di Moodle Martin Dougiamas.

La community di Moodle è particolarmente numerosa ed attiva. Fornisce numerosissimi aggiornamenti, moduli aggiuntivi e template. L'archivio completo è disponibile nel sito del progetto, ed ad ogni componente sono assegnati un piccolo forum ed un bugtracker.

Come la maggior parte dei gestori di contenuti, Moodle è strutturato con un application core corredato di numerosi plugin che ne implementano le specifiche funzionalità. E' progettato per essere largamente estendibile e personalizzabile senza modificare le librerie del core(a differenza di Docebo), evitando problemi successivamente l'aggiornamento ad una nuova versione. Quindi tutte le estensioni e personalizzazioni vanno eseguite utilizzando l'architettura a plugin implementata.

I plugin di Moodle sono divisi in categorie con librerie associate differenti. Ad esempio un plugin per l'autenticazione e un blocco che visualizza il calendario comunicheranno con il core utilizzando diverse API, specifiche per le necessità del plugin. Esistono alcune funzionalità che restano comunque trasversali a dispetto della categoria, ad esempio quelle di installazione, update, permessi e configurazione.

L'installazione base di Moodle non include solamente il core ma anche numerosi plugin di ogni tipo considerati standard, in modo che la piattaforma possa funzionare per scopi generici già dopo l'installazione. Successivamente è possibile cancellarli, installarne di nuovi o modificare i presenti adattando la propria installazione alle specifiche richieste. La maggior parte dei plugin si trovano direttamente nella directory presente nel sito ufficiale di Moodle, divisi per categoria con i voti degli utenti ed un piccolo topic/bugtracker.

Vantaggi di Moodle

Affidabilità

Rispetto agli altri LMS open Moodle è utilizzato in maniera molto maggiore, specialmente in ambiente universitario. Il che testimonia la sua affidabilità, inoltre l'utilizzo così massiccio permette di scovare rapidamente bug e problematiche.

Modularità

Moodle è basato su un core centrale su cui si appoggiano numerosi plugin per fornirgli le funzionalità richieste. In questo modo ogni installazione di Moodle si presenta come un puzzle da comporre con i pezzi necessari al cliente.

Community

La community di programmatori che segue il progetto è molto attiva, e periodicamente vengono rilasciate nuove versioni con i relativi script di aggiornamento ed i bug riscontrati vengono risolti molto rapidamente.

Directory

Grazie ad una community molto attiva ed al sistema modulare sul sito di Moodle è presente una directory contenete centinaia di plugin utilissimi e completamente gratuiti.

Documentazione

Un altro punto di svolta rispetto all'utilizzo di Docebo è la documentazione. All'interno del sito esistono centinaia di articoli che aiutano sia gli amministratori alla gestione di una piattaforma di Moodle sia per gli sviluppatori che intendono migliorarlo o creare nuovi plugin. La documentazione per gli sviluppatori è particolarmente dettagliata e non solo ti guida nella creazione di plugin ma: orienta lo stile di programmazione in modo da uniformarlo per rendere più leggibile il tuo codice da parte di altri programmatori, impone vincoli di sicurezza, consiglia come gestire la sintassi dell'XHTML, regola l'utilizzo dei JavaScript, da indicazioni sull'utilizzo degli IDE e molto altro ancora. Può essere considerato a tutti gli effetti un manuale sulle regole da mantenere per programmare in gruppo. Oltre alle spiegazioni tecniche esistono molti tutorial che seguono il programmatore nello sviluppo di alcuni plugin di base.

Struttura di moodle

Un'installazione di Moodle necessita di un web server capace di eseguire PHP, di un database SQL(MySQL, PostgreSQL, Microsoft SQL Server, or Oracle) e di una cartella per salvare i files uploadati o generati. Queste tre componenti possono essere presenti tutte all'interno di un singolo server(come nel nostro caso) oppure distribuite per bilanciare il carico di lavoro in macchine differenti: web server, database cluster e file-server.

La versione installata su un'istanza del server virtuale associata al dominio egrigori.net è la 2.2. L'istanza possiede tutti i requisiti hardware e software al funzionamento. La root è /var/www/vhosts/egrigori.net/httpdocs/moodle mentre il percorso della cartella di file storage è /var/www/vhosts/egrigori.net/private/moodledata.

Considerando che moodle è composto, senza considerare file temporanei, immagini, corsi o altro inserito durante il funzionamento, di 9 182 file e di 2 732 cartelle, quindi analizzerò solo quelle che ho trattato maggiormente.

Un'introduzione della struttura principale di moodle

Il core di Moodle contiene tutte le infrastrutture necessarie per la costruzione di un LMS. Implementa i componenti chiave di cui si baseranno tutti i plugin. Includono:

- **Corsi ed attività:** un corso di Moodle è una sequenza di attività e risorse raggruppate in sezioni. I corsi a loro volta sono sviluppati in una struttura gerarchica tramite le categorie.
- **Utenti:** utenti, profili, pagine personali di Moodle.
- **Gestione di gruppi e cluster**
- **Iscrizione e controllo degli accessi**
- **Visibilità degli elementi, permessi e ruoli.**
- **Completamento di corsi ed attività**
- **Navigazione , impostazioni e configurazione**
- **JavaScript library:** Moodle adotta le Yahoo User Interface library. C'è anche un sistema molto utile per il caricamento più rapido(tramite combinazione) dei JavaScript aggiuntivi che ogni pagina richiede.
- **Installazione e upgrade**
- **Log, tracciamento e statistiche**

Tipi di utenti

Amministratori

Gli amministratori possiedono numerosi privilegi, tra cui:

- Cambiare le impostazioni
- Creare corsi
- Accedere a qualsiasi corso
- Modificare temi e lingua
- Modificare gli utenti
- Accedere a statistiche ed informazioni personali

Inoltre possono eseguire indirettamente codice PHP. Le loro funzioni possono essere parzialmente limitate tramite il file config.php.

Gli amministratori devono essere persone che conoscono bene il software e di cui ci si possa fidare.

Docenti

I docenti possono inserire contenuti all'interno dei corsi, iscrivere studenti ed insegnare. Necessitano dei privilegi per:

- Caricare files e testi html
- Creare e gestire le attività
- Valutare gli studenti
- Avere accesso ai voti degli studenti e ad altre informazioni personali

Il caricamento di file contenenti JavaScript, flash o altri script è considerato un rischio per la sicurezza, ma non può essere eliminato dato che i docenti devono inserire attività che prevedono l'utilizzo di quest'ultimi, ad esempio gli SCORM.

Gli utenti con i diritti di docente devono essere assegnati con estrema cautela, dato che rappresentano una falla nel sistema di sicurezza di Moodle, perché la limitazione eccessiva dei loro privilegi ridurrebbe di molto la qualità dell'insegnamento.

Studenti

Gli studenti sono gli utenti che parteciperanno ai corsi offerti dalla piattaforma, i loro limitati privilegi riducono al minimo la possibilità che si rivelino pericolosi per il sistema. Hanno la possibilità di:

- Postare articoli contenenti immagini ed allegati
- Partecipare ad i corsi assegnati
- Caricare file binari

I file caricati devono essere gestiti con la massima attenzione, inoltre bisogna filtrare il codice inserito nei post e negli interventi in modo da eliminare qualsiasi tag html o script pericoloso.

Ospiti

In alcuni siti è possibile lasciare l'accesso libero ad alcune sezioni anche agli utenti non registrati nella piattaforma. Questi utenti vengono classificati come ospiti e non hanno nessuna autorizzazione a caricare file, inserire testi o inviare messaggi agli altri utenti. Logicamente tutto questo serve al fine di garantire la sicurezza degli studenti e dei loro dati.

I plugin più importanti

Fisicamente un plugin di Moodle è una semplice cartella contenente script PHP (e CSS, JavaScript e altre risorse se necessario). Il core di Moodle comunica con i plugin cercando uno speciale script, solitamente definito nel file `lib.php` all'interno del plugin. Ogni plugin contiene all'interno le traduzioni di se stesso in tutte le lingue supportate.

Esistono numerosissimi tipi di plugin diversi (più di 30), qui di seguito citerò solo quelli di maggior importanza o che ho utilizzato maggiormente.

Per avere l'elenco aggiornato di tutti i plugin è sufficiente chiamare la funzione

`get_plugin_types()` definita in `lib/moodlelib.php`.

Attività e risorse

Attività e risorse sono le componenti più fondamentali per costruire un corso e il principale strumento per l'apprendimento e l'insegnamento. Alcuni esempi di attività

sono: forum, wiki, quiz e sondaggi. Alcuni esempi di risorse sono: pagine, link, SCORM(lezioni) e documenti.

Le attività sono i plugin più sostanziosi in termini di codice, ad esempio un forum o una wiki potrebbero essere un progetto di software a se stante. Sia le attività che le risorse sono presenti nella cartella 'mod'.

Blocchi

I blocchi sono piccole componenti che possono essere inserite all'interno delle pagine della piattaforma. Solitamente vengono inseriti ai lati ma non è vincolante, ogni tema definisce la posizione in cui vengono inseriti i blocchi. I blocchi vengono solitamente usati per visualizzare maggiori informazioni all'interno di una pagina o garantire altre funzionalità. La visibilità dei blocchi può essere impostata a discrezione degli amministratori.

I blocchi vengono salvati nella cartella 'blocks'.

Temi

Il layout e la grafica di tutte le pagine di un'installazione di Moodle, un corso o la pagina del calendario per esempio, possono essere cambiati semplicemente agendo su questi plugin. I temi sono il sistema standard per interagire sulla grafica e sul layout di Moodle, senza intaccare contenuti e funzionalità. E' il plugin su cui ho lavorato maggiormente e tratterò più dettagliatamente nella sezione dedicata.

I temi sono collocati nella cartella 'theme' presente in root, il sito ufficiale di Moodle contiene un database con numerosissimi temi.

Pacchetti lingua

Moodle è una piattaforma multilingua. In una stessa installazione possono convivere più lingue in contemporanea. I plugin lingua si scaricano normalmente dal pannello amministrativo di Moodle, ma possono essere scaricati dalla directory standard presente sul sito ed installati come gli altri plugin.

Gli amministratori possono inoltre modificare le stringhe di un linguaggio direttamente dal pannello amministrativo.

Formato del corso

Gestisce il sistema in cui le attività di un corso vengono presentate agli studenti. Questi plugin si trovano nella cartella 'course/format'.

Autenticazione

Controllano come l'utente si logga. Moodle può autogestirsi gli utenti ed il sistema di login, oppure appoggiarsi ad un altro database. In alternativa esistono numerosi plugin che utilizzano il single-sign-on, cioè autorizzano l'autenticazione utilizzando tuoi account già esistenti, ad esempio quello di Facebook o Google.

I plugin di autenticazione risiedono nella cartella 'auth'.

Repository plugins

Gestiscono il sistema per caricare files all'interno di Moodle, sia caricandolo dal proprio hard disk, sia reperendoli dal web, ad esempio tramite Dropbox o Google Docs.

Questi plugin si trovano nella cartella 'repository'.

Temì

Un tema di Moodle permette di cambiare completamente grafica e layout di un sito Moodle. I temi possono essere assegnati a: l'intero sito, una categoria, un corso o ad uno specifico utente. Alcuni temi possono essere sviluppati per specifici dispositivi, ad esempio telefoni o tablet. Di base Moodle prevede di assegnare un tema ad ognuno di queste 4 categorie di dispositivi: default, legacy, mobile, tablet.

I temi non definiscono solo la grafica, ma anche il layout. Possono riscrivere la struttura e la grafica del sito completamente, oppure ereditare un tema già esistente ed ampliarlo, in questo modo se il tema non definisce il comportamento per un singolo componente o pagina verrà ereditato quello del genitore. Viene quasi sempre utilizzata quest'ultima opzione e tutti i temi più recenti estendono il tema base. Solitamente si cerca di partire da un tema esistente il più possibile vicino alle proprie necessità ed o estenderlo oppure modificarlo, è quello che è stato fatto pure in questo caso.

Moodle possiede un sistema per migliorare notevolmente le prestazioni dei temi. I file CSS vengono combinati in un singolo file ottimizzato, e sia i file CSS che JavaScript vengono minimizzati per assicurare che non ci siano sprechi di traffico o connessione. I file vengono massivamente salvati in cache, ma anche versionati, in modo che gli utenti non siano mai obbligati a pulire la loro caches.

Struttura di un tema

I componenti più importanti ad tenere in considerazione nello sviluppo sono:

- config.php, questo file è obbligatorio in ogni tema. Definisce configurazione e tutti i requisiti per far funzionare il tema in Moodle. Includono i temi ereditati, file, localizzazione e opzioni.
- I layout: all'interno del config.php esiste una definizione che associa ogni tipo di pagina di Moodle al file di layout che verrà utilizzato per visualizzarla. Questo file contiene l'HTML e il minimo PHP necessario per la visualizzazione degli elementi.
- Il tema base: non è pensato per l'utilizzo come tema di una piattaforma. E' il layout più semplice possibile ed include solo i CSS essenziali per rendere leggibile le pagine di Moodle. Il suo obiettivo è di non includere nessuna regola non necessaria. E' il tema perfetto da ereditare quando si vuol disegnarne uno da capo, dato che ci sono pochissimi colori, margini e padding da sovrascrivere.

File e cartelle

La cartella contenente un tema deve essere posizionata all'interno del percorso '/theme'. I file e le cartelle più importanti sono:

Cartella	File	Descrizione
	/config.php	Contiene tutte le configurazioni e le definizioni di un tema
	/lib.php	Contiene le classi e le funzioni utilizzate dal tema
	/settings.php	Contiene le configurazioni personalizzate del tema. Queste impostazioni permettono all'amministratore di cambiare alcune impostazioni basilari(ad esempio logo o colore di sfondo) direttamente dal pannello amministrativo
/javascript/		Contiene tutti i file JavaScript
/lang/		Contiene la traduzione, nelle varie lingue supportate, di tutte le stringhe del tema
/layout/		Contiene tutti i file di layout
/pix/		Contiene le immagini utilizzate sia dal codice PHP che dai file CSS.
/style/		Contiene tutti i file CSS

Tutti gli attributi devono essere aggiunti a \$THEME. Che è l'oggetto che gestisce la configurazione del tema, viene creato da Moodle con le impostazioni di default e man mano che vengono assegnate viene sovrascritto.

`$THEME->name`

Semplicemente il nome del tema creato.

`$THEME->parents`

Definisce i temi che verranno estesi.

`$THEME->sheets`

Un array che elenca tutti i fogli di stile, è sufficiente inserire il nome(dato che Moodle presuppone che l'estensione sia .css e che siano contenuti nella cartella style)

`$THEME->layouts`

Definisce il file .php che ogni layout utilizzato dalle componenti utilizzerà.

CSS

Tutti i file css di un tema vanno posizinati in

`\theme\themename\style*.css`

E' utile notare l'ordine con cui i CSS vengono trovati e inclusi crea una gerarchia. Questo ordine permette che le regole, all'interno dei fogli di stile del tema, abbiano la priorità rispetto ad altre identiche introdotte precedentemente. In questo modo possono estendere le definizioni precedenti, ma garantiscono anche che le regole CSS del tema abbiano l'ultima parola.

Esistono molti altri posti in cui è possibile trovare i file css e modificare o aggiungere nuove regole, ad esempio ogni plugin ha un foglio di stile a se stante, ma è sbagliato agire su questi, un web designer dovrebbe operare solo sulla cartella del tema a cui sta lavorando.

In questa relazione non tratterò le modifiche fatte in collaborazione con il grafico sulle regole del css, dato che non le ritengo fondamentali per il risultato finale.

Layout

Ogni pagina generata da Moodle crea una istanza di un oggetto \$Page. In ogni pagina viene chiamato il metodo dell'oggetto Page set_pagelayout che richiede come parametro una stringa. La stringa sarà il nome del layout da cercare nel file di configurazione del tema. I layout utilizzati di standard da Moodle sono:

Layout	Purpose
--------	---------

base	Il layout maggiormente retrocompatibile, funziona senza blocchi e viene usato di default.
standard	Layout standard con i blocchi, raccomandato per la maggioranza di pagine con informazioni generali.
course	La pagina principale di un corso.
coursecategory	Usata per navigare tra le categorie dei corsi.
incourse	Questo layout viene utilizzato quando si apre un elemento all'interno di un corso, ad esempio i pacchetti SCORM.
frontpage	La home page.
admin	Layout che viene visualizzato nelle sezioni di amministrazione del sito.
mydashboard	My dashboard page.
mypublic	La pagina pubblica personale.
login	The login page.
popup	Le pagine che appaiono come pop-up; senza navigazione, blocchi o intestazione.
embedded	Pagine che richiedono più spazio possibile, come iframe.
maintenance	Usata durante installazione e aggiornamenti. Non deve contenere assolutamente i blocchi e nemmeno link ad altre parti del sito.
print	Viene utilizzata quando una pagina deve essere stampata. No blocchi e struttura minima, si lascia importanza solo al contenuto.

In più ogni utente può crearne di propri a seconda delle necessità. Ogni tema può definire tutti i layout che vuole, ed ereditando il base automaticamente definisce anche tutti gli standard, logicamente si possono anche sovrascrivere le definizioni dei genitori. Se un tema è completamente nuovo deve ridefinire tutti i layout.

I layout vengono ridefiniti nel file config.php all'interno di \$THEME->layout.

La prima cosa che Moodle guarda è il nome del layout, in questo caso è 'standard'. Ci sono anche altre opzioni che possono essere settate su un layout:

- theme: è il tema in cui è contenuto il file di layout, in modo da poter utilizzare file da altri temi. Opzionale
- file: è il nome del file di layout che si vuole utilizzare. Richiesto
- region: sono le regioni in cui possono essere posizionati i blocchi all'interno di un tema. Richiesto
- defaultregion: la posizione di default di regions. Opzionale

Personalizzazione della piattaforma

L'installazione base di Moodle garantisce funzionalità molto maggiori di quelle richieste dal progetto. Anzi, l'utilizzo di questo LMS ha permesso di ampliare l'offerta commerciale di numerose opzioni inizialmente non pensate. Quindi la maggior parte del lavoro è stata finalizzata al miglioramento dell'aspetto grafico e dell'accessibilità, quindi ho operato quasi esclusivamente ad un plugin di tipo tema.

Plugin utilizzati

Blocchi

I blocchi utilizzati in questa installazione sono (facilmente visualizzabili al percorso 'amministrazione del sito → plugin → blocchi → gestione'):

- File personali (componente standard): un utente può caricare e conservare all'interno i propri file personali, è in sviluppo una modifica che permette agli amministratori di caricare i file tramite un'interfaccia all'utente desiderato, in modo da rendergli disponibili alcuni documenti ad esempio le buste paga.
- Barra progressi (installato, tradotto e modificato): visualizza lo stato di completamento di tutte le attività di un corso, è fondamentale inserire nelle impostazioni delle singole attività la condizione di completamento
- Attività recenti (standard): all'interno di un corso visualizza le ultime attività inserite
- Utenti online(standard): visualizza gli utenti loggati nella piattaforma
- Navigazione(standard): funge da breadcrumbs, cioè serve a navigare all'interno della piattaforma.
- Menu Principale(standard): visualizzato in homepage serve a raccogliere le voci più utilizzate dall'utente o le più importanti, nel nostro caso contiene le news del sito, il forum, i materiali che condivide l'azienda e i sondaggi
- Ricerca nel forum(installato): permette di effettuare una ricerca all'interno del forum, utile per trovare rapidamente se un utente ha avuto lo stesso problema e risolverlo

- Messaggi(standard): visualizza se ci sono nuovi messaggi per l'utente loggato e da la possibilità di accederci rapidamente
- Prossimi eventi (standard): permette la visualizzazione dei prossimi eventi del calendario personale dell'utente loggato
- HTML (standard): in questo blocco l'amministratore può inserire liberamente codice html, nella homepage è stato utilizzato per inserire le convenzioni

Altri plugin

Ho inserito nuove tipi di domande per i quiz interni a Moodle ma non sono stati utilizzati dato l'interattività dei quiz che è possibile ottenere tramite i pacchetti SCORM.

Un nuovo formato di corsi, il Course Grid

(http://moodle.org/plugins/view.php?plugin=format_grid). Questo plugin di tipo “formato corsi” visualizza le singole sezioni di un corso come un'immagine che se cliccata apre un pop-up che visualizza le singole attività all'interno. La visualizzazione è molto più accattivante e intuitiva rispetto a quelle standard, ma inizialmente il pop-up dava problemi con la versione mobile, ora risolti.

Tema scelto

Realizzare un tema senza nessuna base sarebbe stato molto oneroso e probabilmente anche inutile dato che La selezione dei temi da cui partire per la realizzazione, tutti installati e testati per vederne le potenzialità ed i vantaggi, è stata:

Aardvark

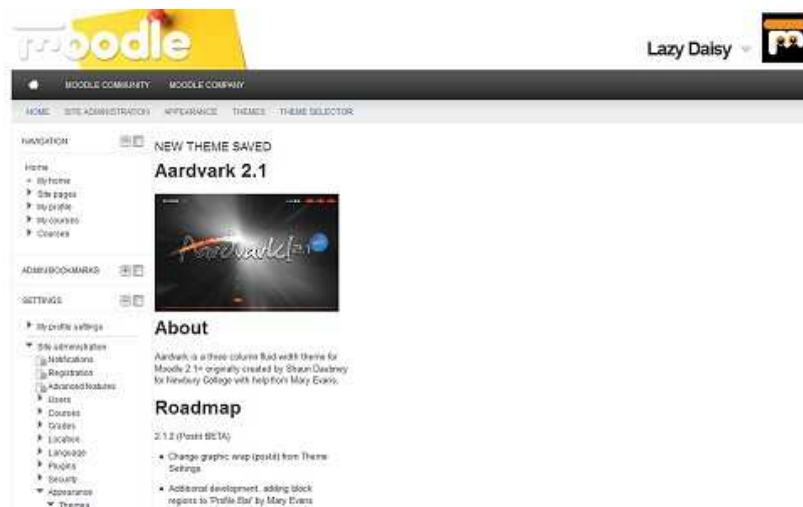
(http://moodle.org/plugins/view.php?plugin=theme_aardvark)



Un tema molto accattivante, con un gran uso delle librerie jQuery. Una grafica moderna ed alcune chicche come il menu utente in alto apribile in cui è possibile inserire alcuni blocchi.

Aardvark Post-it

(http://moodle.org/plugins/view.php?plugin=theme_aardvark_postit)



Stessi vantaggi del precedente, è solo una versione con una grafica più istituzionale.

Bbyte Fusion Mod

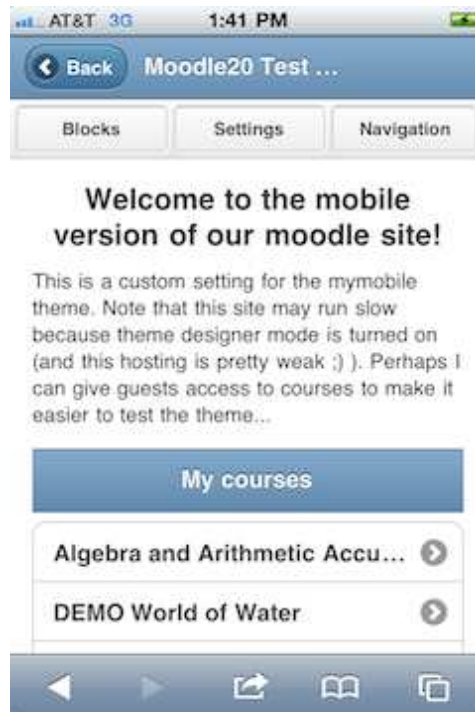
(http://moodle.org/plugins/view.php?plugin=theme_bbytefusionmod)



Tema della bananabyte, uno studio brasiliano specializzato nella creazione di soluzioni e template per l'e-learning. La grafica tende a far assomigliare la piattaforma ad un semplice sito di informazioni più che a un LMS

My Mobile

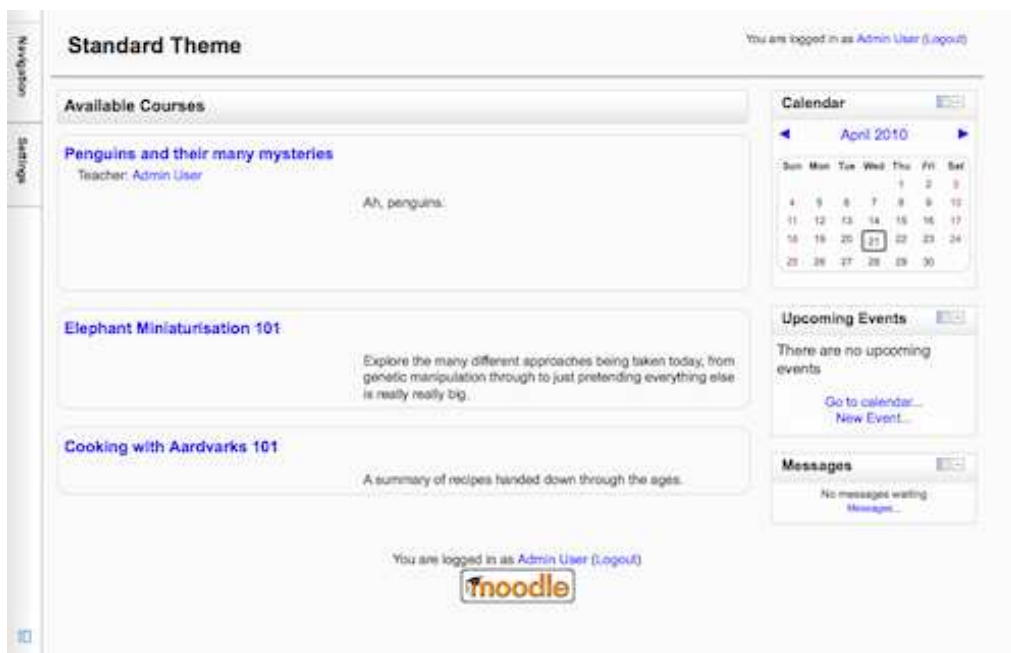
(installazione standard di Moodle)



Un tema disegnato specificamente per i dispositivi mobili, in particolare per iPhone e dispositivi android. La grafica riprende i menu dell'iOS, essenziale e pratica. I contenuti vengono limitati in ogni pagina. In caso di tablet come dispositivo viene visualizzato un pulsante che permette di passare da un layout a colonna singola ad uno a colonna doppia. Le funzionalità di modifica dei blocchi e di aggiunta dei materiali da parte di docenti e amministratori I pulsanti arrotondati e le sfumature non sono immagini, ma vengono generati tramite l'utilizzo delle proprietà del CSS3. Al cambiamento di pagina viene eseguita una transizione molto accattivante. L'intero tema sfrutta le librerie jQuery mobile ed è compatibile con tutti i browser più recenti.

Standard

(installazione standard di Moodle)



Il tema classico di Moodle, Molto essenziale e dai colori neutri. Sicuramente un'ottima base se qualcuno vuole scrivere un tema su misura dato che i CSS da sovrascrivere sono pochi, sicuramente non sufficiente se si vuole trovarne uno quasi pronto su cui fare solo minime modifiche.

Rocket

(http://moodle.org/plugins/view.php?plugin=theme_rocket)

La semplicità e la mancanza di bordature gli danno una linea moderna, solo che la



mancanza di interazione rendono piatto il layout.

Selezione finale

La scelta finale è stata quasi sorprendente, l'idea iniziale di mantenere un'interfaccia grafica più simile possibile indipendentemente dal dispositivo che la sta utilizzando è all'aumentarne la semplicità d'utilizzo ha portato alla scelta del tema MyMobile.

Nonostante sia un tema pensato per dispositivi mobili funziona perfettamente anche su pc (con qualche piccolo adattamento). Gli altri temi cambiavano graficamente ed introducevano piccole interazioni che li differenziavano, ma alla fine il risultato non variava molto. Inoltre essendo molto neutro come colori ed immagini è possibile riadattare il tema MyMobile a seconda del cliente molto rapidamente.

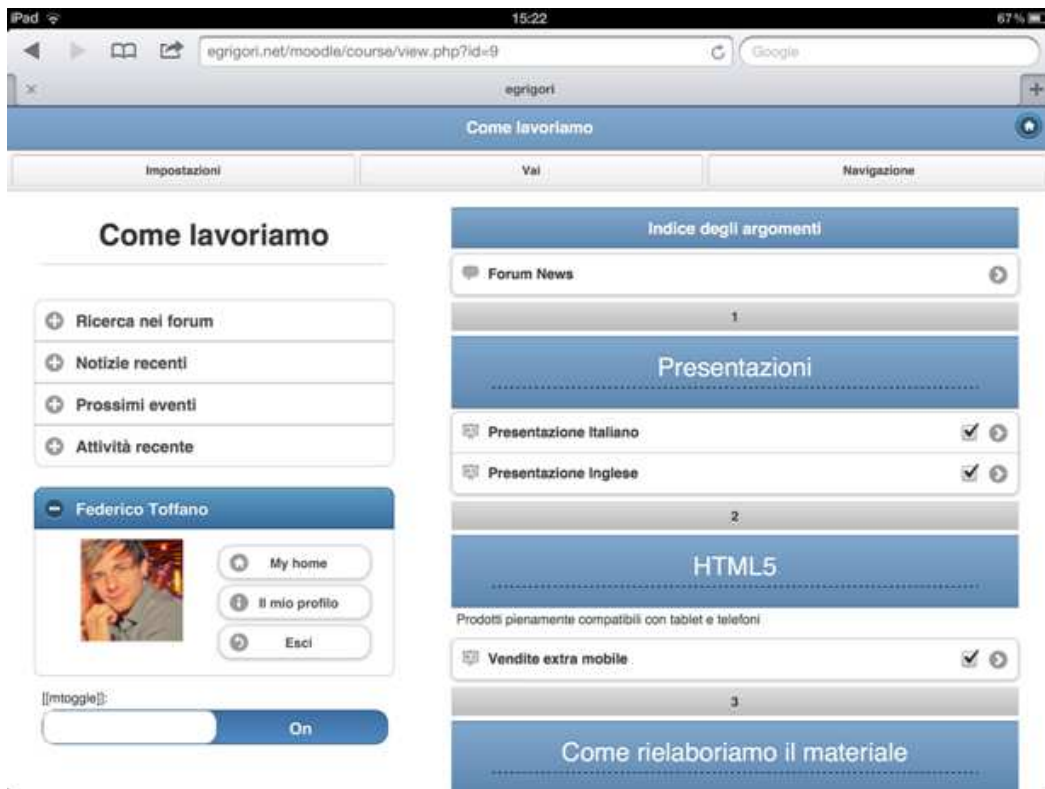


Figura 6 Homepage my mobile

Anche se la scelta di questo tema comporta due grandi problemi:

- Il tema non è compatibile con versioni di Internet Explorer precedenti alla 8, mentre funziona perfettamente con gli altri browser più utilizzati. Questo per un'incompatibilità tra alcune regole CSS3 utilizzate che i vecchi browser Microsoft. Le alternative sono o il ridirigere l'utente ad una pagina contenente una guida che lo accompagni nell'installazione o nell'aggiornamento del suo browser, oppure l'assegnazione agli utenti che non ne dispongono di un template secondario. Fortunatamente le versioni precedenti di Explorer, famose per aver complicato notevolmente la vita alla categoria dei web designer, stanno via via andando in disuso. Come si può evincere dal grafico tratto da StatCounter (<http://gs.statcounter.com/>) meno del 15% dei browser utilizzati in Italia fanno parte di questa categoria ed il loro numero è in continua discesa, basti pensare che solo a gennaio erano utilizzati da un quarto degli internauti (circa il 24 %).

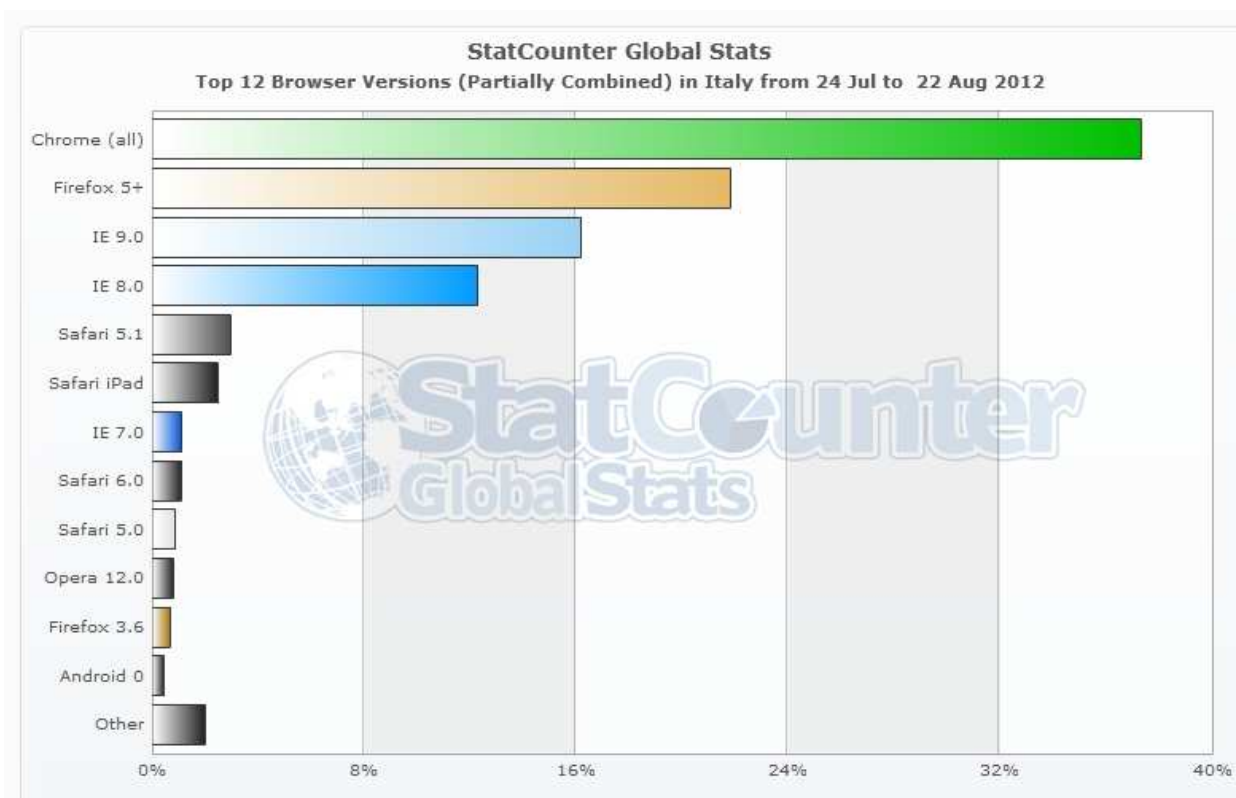


Figura 7 Statistiche utilizzo browser Italia

- I docenti e gli amministratori non possono utilizzare questo tema dato che limita troppo le funzionalità di inserimento attività e modifiche delle impostazioni di blocchi ed oggetti formativi. A loro sarà necessario assegnare un tema dedicato, questo non risulta essere un problema, dato che i docenti e gli amministratori risultano essere tutti interni alla nostra azienda.

Personalizzazioni del tema

Ordinando le modifiche da fare sono:

- Grafiche: agendo sui css adattare la piattaforma ad i colori di egrigori e diminuire lo spazio bianco lasciato tra un argomento e l'altro in modo da cercare di visualizzare tutto in una singola schermata
- L'eliminazione dello switch che permette di passare dalla visualizzazione a due colonne a quella ad una sola, selezionando di default che layout utilizzare ad ogni situazione
- Nell'header è sempre presente un tasto home, ma spesso è più utile l'utilizzo di un tasto back(anche perchè in alcuni layout è stato deciso di eliminare la barra di navigazione per lasciare più spazio al contenuto)

- Il tasto My Home nell'area personale fa accedere ad una pagina ricca di informazioni di poco interesse per l'utente, è stato deciso di modificarla inserendo al suo posto un collegamento diretto al plugin che gestisce una piccola directory con i file personali. In questa pagina l'utente può trovare tutti i documenti che lo riguardano, tra cui CUD e buste paga
- Il link al calendario sottostante portava l'utente ad una pagina in cui erano visualizzati solo gli ultimi eventi e non la visione classica di un calendario, è stato necessario modificarlo per motivi di praticità, estetici e di ridondanza, in home è già presente un blocco ultimi eventi.
- Correzione di errori di funzionamento, ad esempio i link generati dal plugin calendario e dai blocchi prossimi eventi rimandavano in home ma non funzionavano.

Operazioni base sui css

Ho agito sui css per adattare la piattaforma ai colori istituzionali di egrigori (azzurro e bianco). Il problema è che la maggior parte sono gradienti di colore, quindi con l'aiuto del grafico abbiamo creato una piccola tavolozza di colori possibili e tonalità da utilizzare per le varie componenti. Successivamente le ho applicate alle varie classi CSS (non elenco le singole modifiche per non essere ripetitivo).

Oltre a questo ho lavorato per la limatura e il restringimento dei margini e dei padding in modo da cercare di visualizzare tutto il contenuto in una singola pagina ed evitare il ricorso alla barra di scorrimento.

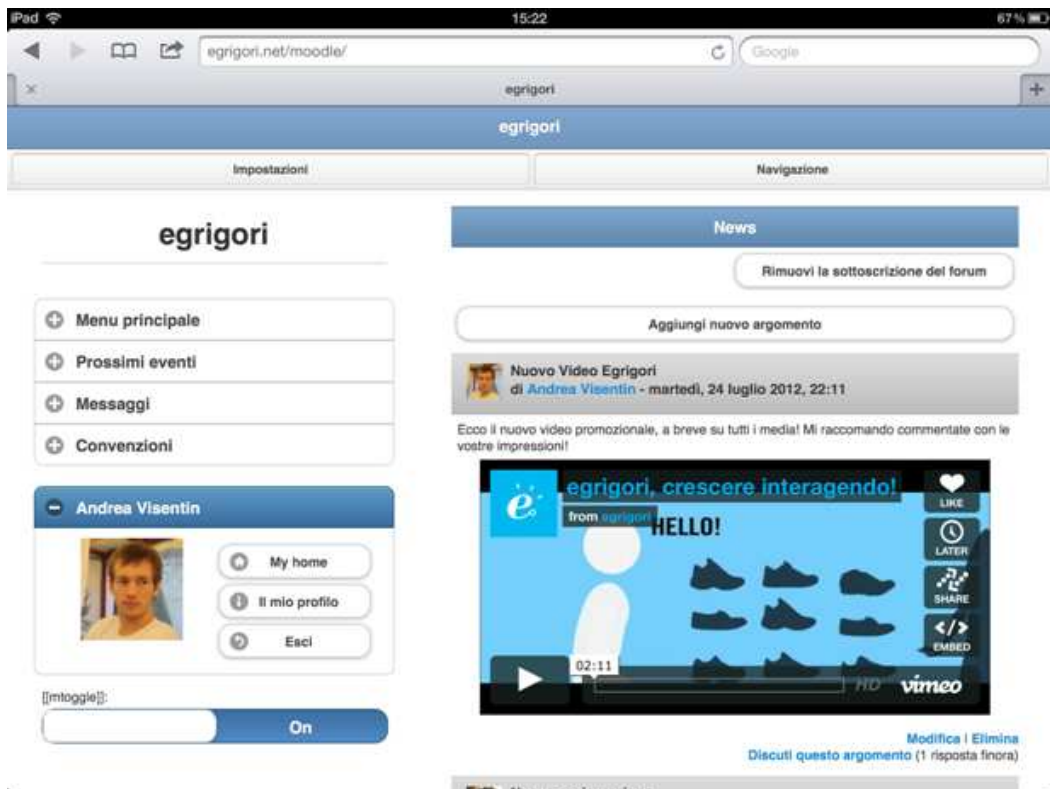


Figura 8 Colori e padding prima della modifica

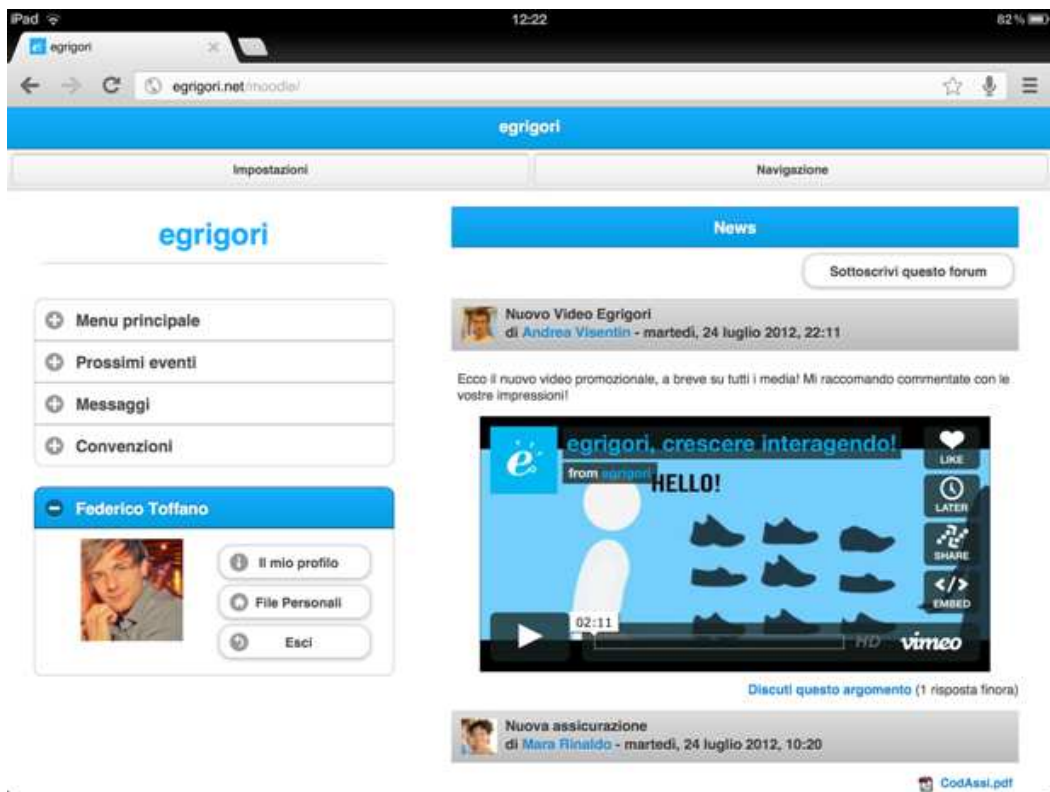


Figura 9 Homepage dopo la modifica

Creazione del tasto Back

Durante la navigazione risultava la presenza solo del tasto home, dato che spesso si voleva tornare solo al livello superiore, mentre la navigazione risultava scomoda ed è stata eliminata da alcune pagine.

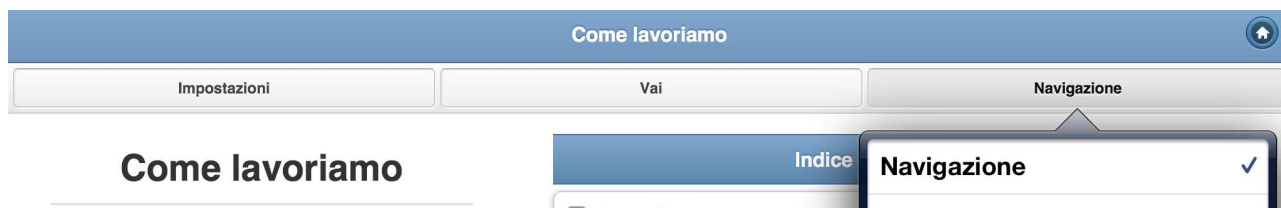


Figura 10 Header

La soluzione era semplice, la creazione di un tasto 'back'.

Per la posizione è stato deciso di metterlo in posizione diametralmente opposta rispetto al tasto 'home', di cui riprende la grafica mentre l'immagine è la stessa del tasto 'esci' presente nell'area personale. Per inserire il tasto in quella posizione è stato necessario definire nel file CSS principale(core.css) la classe un-btn-left in questo modo:

```
.ui-btn-left { position: absolute; left: 10px; top: .4em; }
```

Per il codice necessario all'esecuzione le alternative erano o utilizzare una funzione in PHP oppure una in JavaScript Per semplicità e leggibilità ho scelto la seconda dato che esiste una funzione predefinita che esegue la funzionalità richiesta. Quindi ho inserito il codice come attributo href di un tag a di HTML in questo modo.

```
<a class="ui-btn-left" data-icon="back" href="javascript:history.go(-1);" data-iconpos="notext" data-ajax="false">Back</a>
```

Questo tasto è stato inserito all'interno di tutti i file di layout che compongono il tema all'interno del div di header.



Figura 11 Footer con tasto back e home

Il blocco comprendente i due pulsanti viene visualizzato solo se si rispettano le condizioni dell'if seguente:

```
<?php if (isloggedin() && $mypagetype != 'site-index') { ?>
```

Quindi solo se l'utente è loggato e non è presente in home page.

Eliminazione dello switch delle colonne.

Visualizzando il sito tramite pc o tablet compare sotto al blocco personale un piccolo switch che trasforma il layout da 2 colonne a una sola.

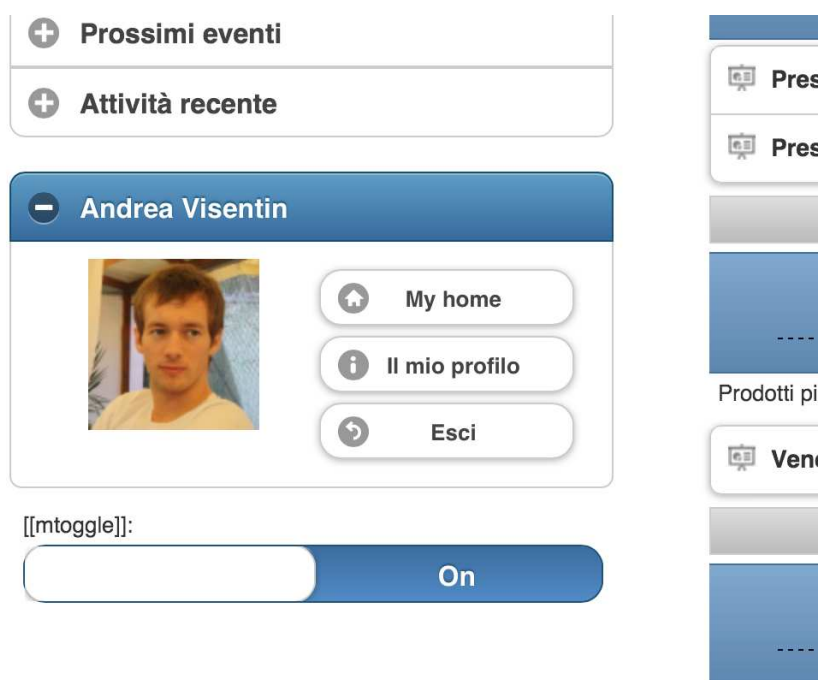


Figura 12 Profilo personale e switch

I problemi che sorgono sono due; la presenza di elementi futili distrae l'utente e può essere causa di problemi di utilizzo, solitamente ad ogni pagina si adatta solo un layout corretto. La seconda nello specifico è la più importante dato che la visione ad una colonna sola risulta un problema in pagine con contenuti molto corti e menu (ad esempio home page), mentre la colonna doppia toglie solo spazio ai contenuti quando devo visualizzare SCORM, altri elementi formativi, calendari o statistiche.

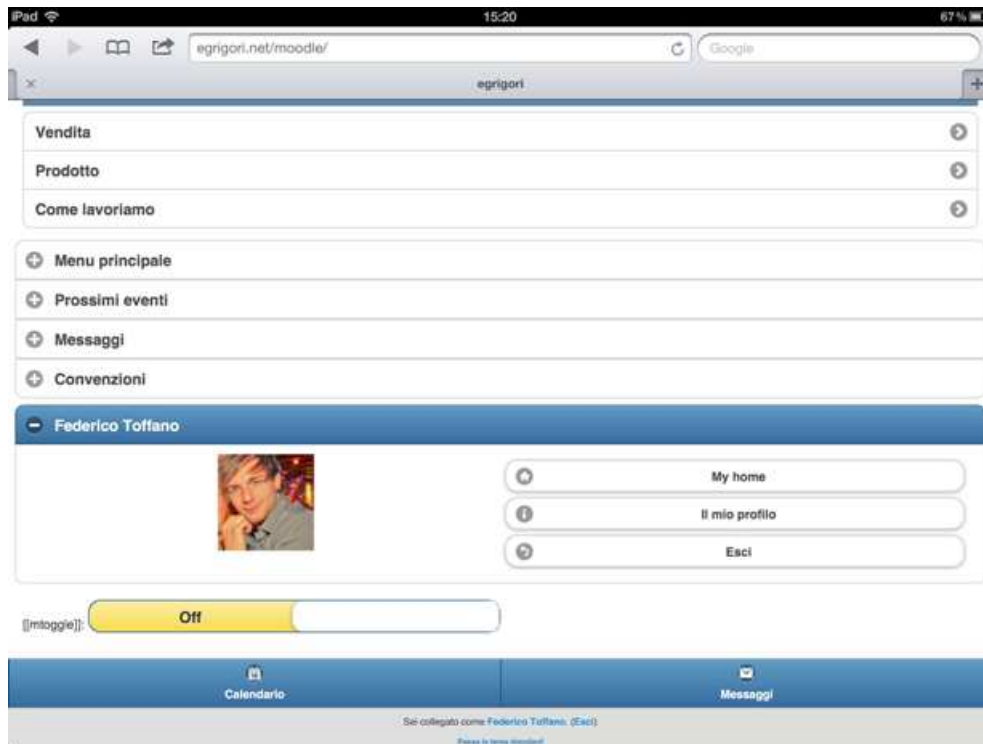


Figura 13 Homepage ad unica colonna



Figura 14 Elemento formativo a due colonne

Il blocco viene generato dal codice HTML:

```
<div data-role="fieldcontain" id="sliderdiv">
  <label for="slider"><?php p(get_string('mtoggle','theme_mobile'));
?>:</label>
```

```

        <select name="slider" class="slider" data-role="slider" data-track-
theme="b">
            <option value="on">On</option>
            <option value="off">Off</option>
        </select>
    </div>

```

e quando viene cliccato agisce sull'attributo 'class' del div principale del sito, assegnandogli la classe has-myblocks

```
<?php if ($hasmyblocks && $usercol) { echo 'has-myblocks'; } ?>
```

Quindi ho eliminato il codice che generava lo switch, duplicato il file di layout rinominandone uno in onecolum.php ed infine in uno assegnato la classe has-myblocks al div principale e all'altro no.

Inoltre dato che la visualizzazione ad una colonna sola viene utilizzato solo in pagine di contenuto e il blocco personale sotto risultava ridicolo e scomodo lo ho eliminato dal layout onecolum.

Per la selezione di layout che utilizzino una colonna sola ho scelto tutte le attività di un corso (riconducibili al layout 'incourse') e la pagina del calendario. La pagina del calendario sfrutta il layout standard, quindi sono dovuto entrare nel componente ed editarlo modificando il file view.php all'interno di 'calendar' sostituendo la riga

```
$PAGE->set_pagelayout('standard');
```

con

```
$PAGE->set_pagelayout('calendar');
```

anche se questo metodo non rispetta la modularità del tema è l'unico sistema per assegnare un layout specifico solo alle pagine del calendario.

Infine ho modificato il file config.php modificando l'array dei layout con questi due elementi:

```

        'calendar' => array(
            'file' => 'onecolum.php',
            'regions' => array(),
            'options' => array('nonavbar'=>true),
        'incourse' => array(
            'file' => 'onecolum.php',
            'regions' => array(),
            'options' => array('nonavbar'=>true),
        ),

```


L'accessibilità della piattaforma risulta molto migliorata e non è più necessario dover ogni volta operare con lo switch per visualizzare correttamente i contenuti di una pagina.

Come tocco finale ho eliminato la barra laterale contenente l'area personale dal layout ad una colonna sola, dato che sarebbe stato visualizzato sotto ed in maniera scorretta.

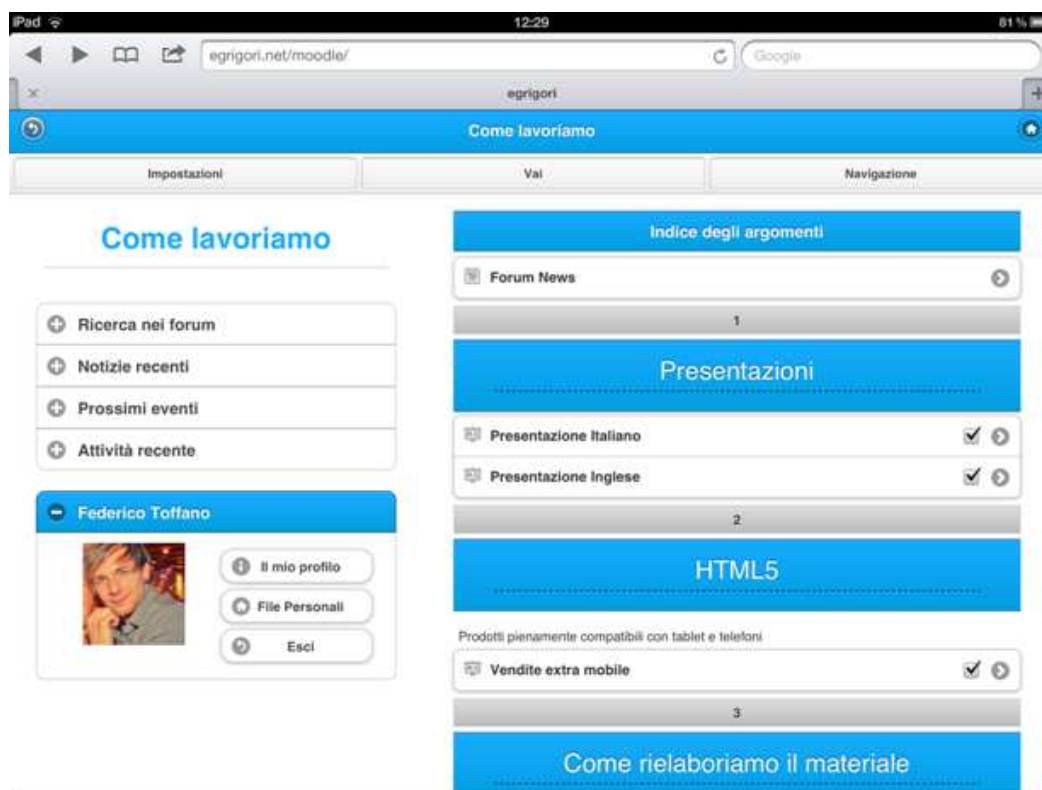


Figura 15 Home Page finale senza switch

Creazione di una pagina di login

La pagina di login iniziale prevedeva la singola colonna, ma il risultato lasciava molto a desiderare dato che erano presenti solo due aree per il testo ed un pulsante.

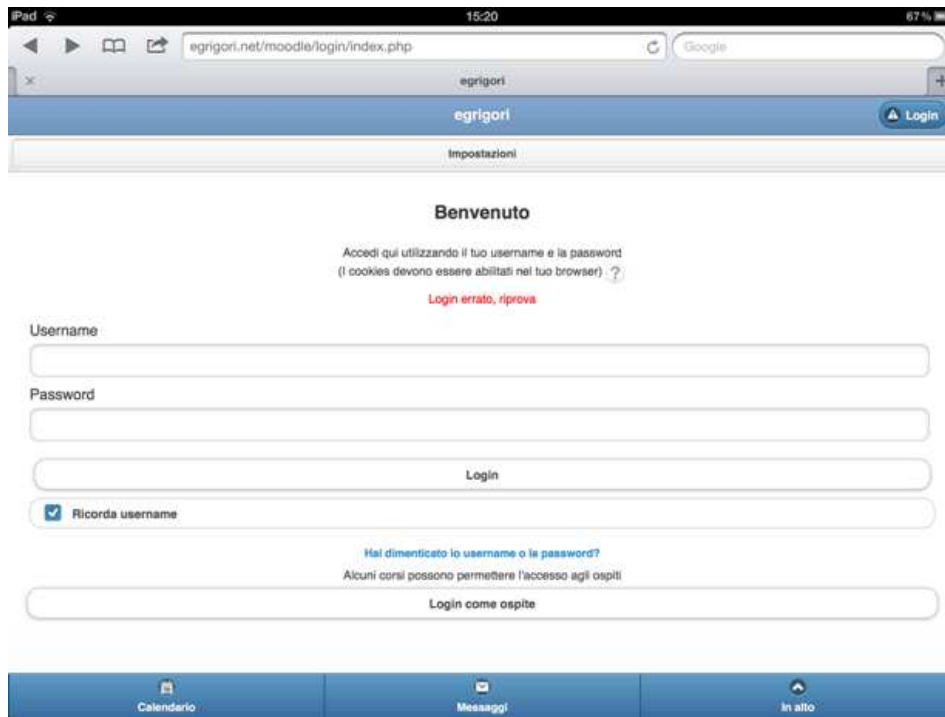


Figura 16 Pagina login originale

Anche dividerla a due colonne risultava un problema, dato che rimaneva un grande spazio vuoto a sinistra. Quindi si è deciso di riempire questo spazio con un'immagine contenente il logo di egrigori.

Per far ciò ho duplicato il file di layout di standard ed eliminato il contenuto della barra laterale, sostituendolo con un semplice tag immagine

```

```

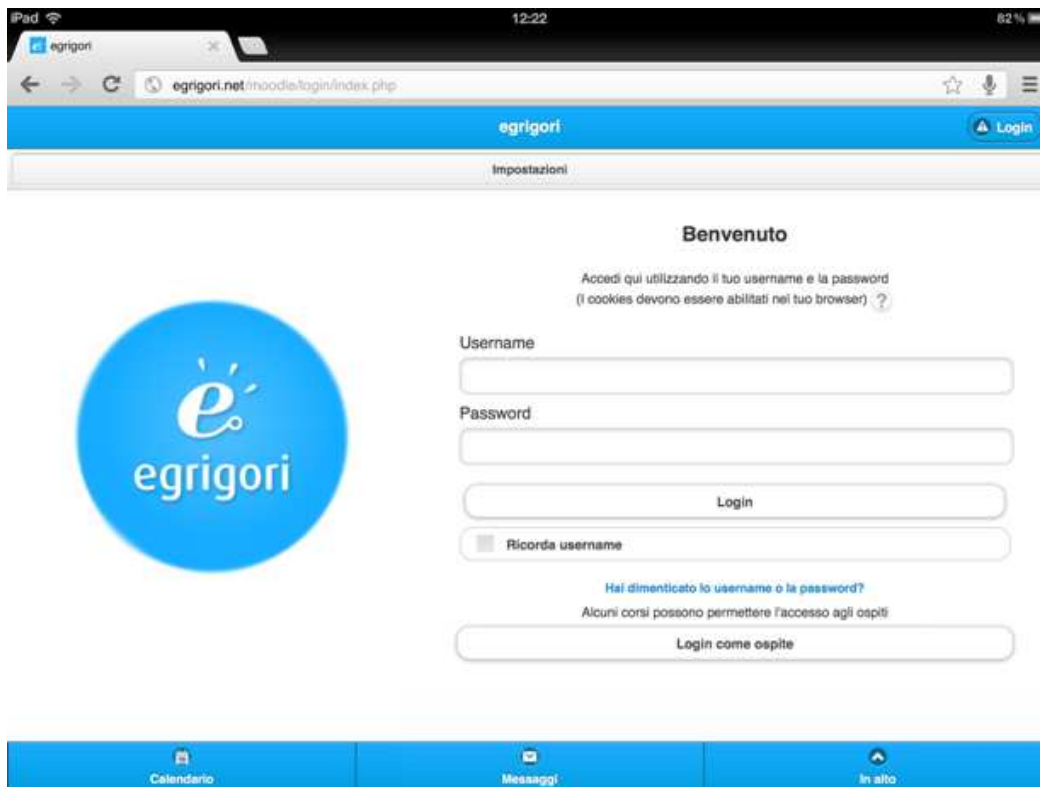


Figura 17 Nuova login

Modifica tasto My Home

Nell'area personale sono presenti 3 pulsanti:

- My Home: serve per visualizzare alcune informazioni relative ai corsi seguiti ed allo stato di completamento
- Il Mio Profilo: che contiene un'anagrafica dell'utente
- Esci: manda l'utente ad una pagina di logout

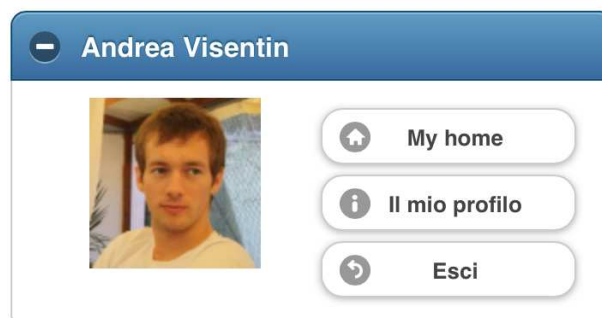


Figura 18 Area profilo personale

Il pulsante My Home è praticamente inutile, dato che sono informazioni ridondanti già presenti all'interno del profilo e dei singoli corsi. Per questo ho deciso di sostituirlo con

una pagina che utilizzi il plugin di repository per metterci i file personali. Ho modificato il pulsante

```
<a data-role="button" data-icon="home" href="<?php p($CFG->wwwroot) ?>/my/"><?php p(get_string('myhome')); ?></a>
```

Con

```
<a data-role="button" data-icon="home" data-ajax="false" href="<?php p($CFG->wwwroot) ?>/user/filesedit.php">File Personali</a>
```

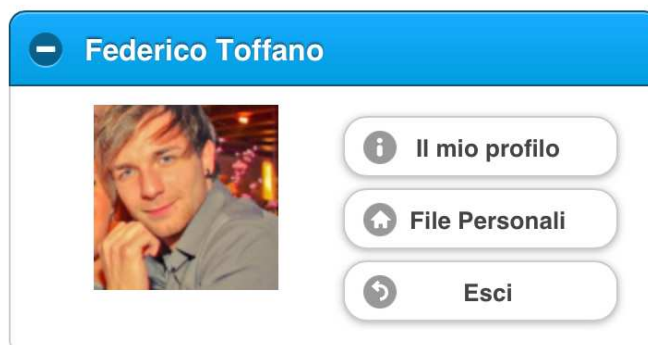


Figura 19 Nuova area personale

Se il cliente lo richiede è possibile personalizzare l'area in modo da utilizzarla per inserire buste paga e cud al suo interno.

Modifica pulsante Calendario

Nel footer sono presenti il pulsante Calendario e quello Messaggi. Il pulsante calendario manda l'utente in una schermata in cui è possibile vedere solo l'elenco dei prossimi eventi in ordine cronologico, non una visualizzazione classica.

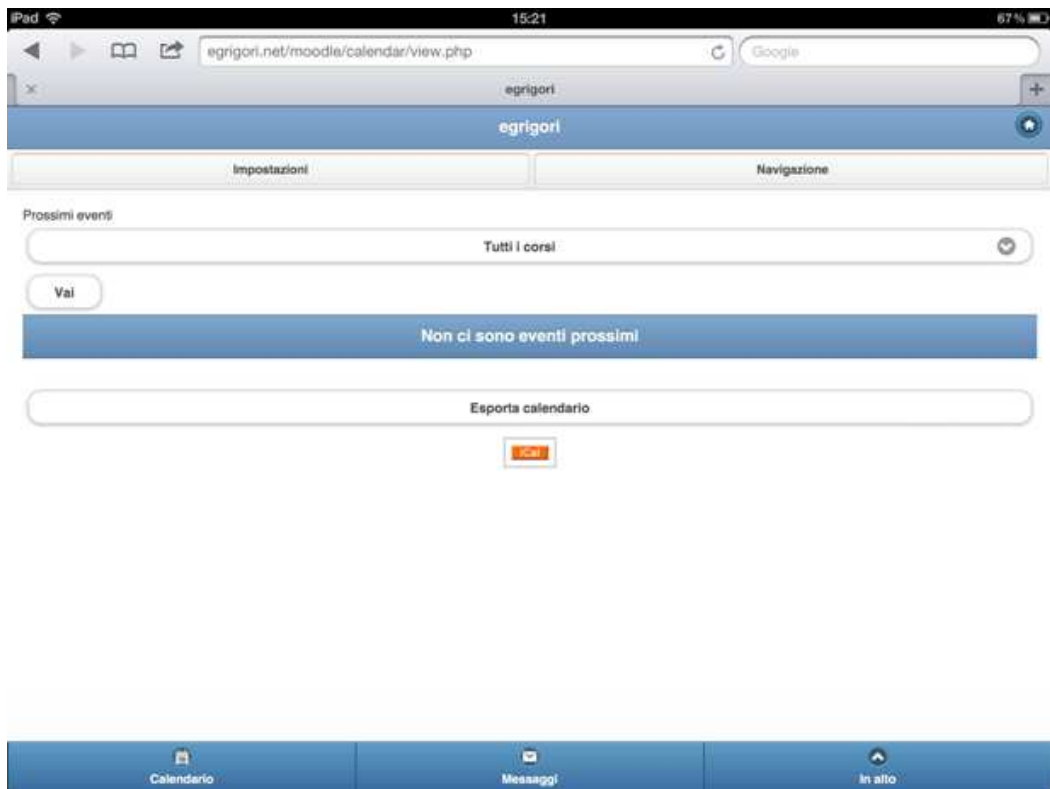


Figura 20 Calendario prima della modifica

Analizzando il comportamento del plugin calendario ho modificato l'url del pulsante aggiungendo il parametro view=month in modo da aver la visualizzazione desiderata.

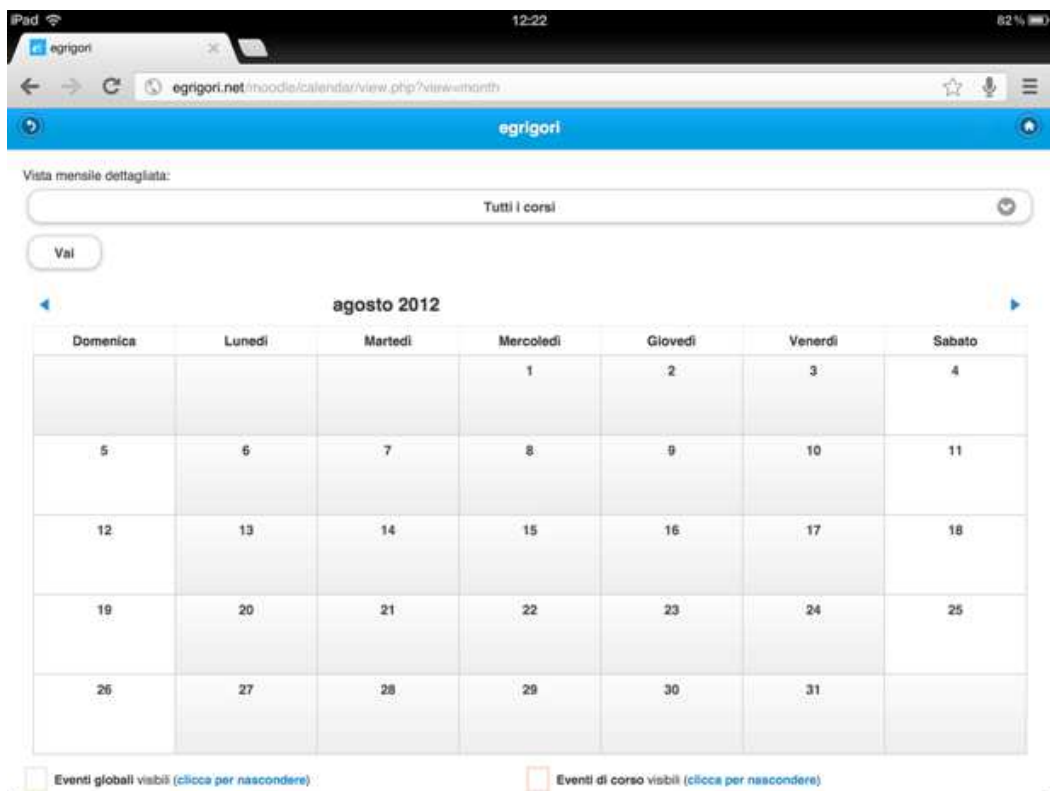


Figura 21 Nuovo calendario

Risoluzione problemi

Il template aveva alcuni problemi di funzionamento e nel caricamento delle pagine, nello specifico le librerie jQuery modificavano gli url ricevuti dal plugin parametro eliminando alcuni argomenti php partendo dal simbolo cancelletto. Per eliminare il problema ho dovuto agire nel componente ed inserire il parametro `data-ajax="false"` che obbliga il refresh della pagina.

Inoltre per motivi di sicurezza non era possibile inserire nei commenti e nei post tag HTML di tipo Object. Ho dovuto disattivarle per inserire i video nelle news in homepage.

Considerazioni finali

Il risultato finale è stato molto soddisfacente, la piattaforma riscuote apprezzamenti nelle varie presentazioni e ben presto inizieremo l'utilizzo vero e proprio.

Ma nell'implementazione e nello sviluppo ci sono ancora alcuni errori o piccole violazioni dello stile di programmazione che ho commesso e che dovrò correggere.

Ad esempio l'aggiunta di alcuni file di layout non è sbagliata, ma sarebbe molto più pulito (anche se meno chiaro) e più semplice da modificare se il file fosse unico e le modifiche venissero fatte tramite if in PHP.

Oppure il fatto che in alcune parti del template ho inserito del testo direttamente, senza aggiungere un elemento alla traduzione, questo potrebbe generare problemi quando sono presenti più lingue all'interno della piattaforma.

Ma sia moodle sia il template di base sono in continuo aggiornamento e sviluppo, quindi ad ogni nuova installazione e personalizzazione è conveniente partire dalla base più recente possibile, solo che avendo già effettuato e segnato tutte le modifiche da fare e conoscendo a fondo il software la maggior parte di queste sarà sufficiente trasportarle nella nuova versione, mentre le altre saranno molto rapide dato che so già dove agire e come modificare il codice.

Sitografia

Repository plugin di Moodle, <http://moodle.org/plugins/>

Moodle developer documentation, http://docs.moodle.org/dev/Developer_documentation

JQuery mobile documentation, <http://jquerymobile.com/test/index.html>

Docebo, <http://www.docebo.com/it>

Definizione SCORM da Wikipedia, <http://it.wikipedia.org/wiki/SCORM>