

ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

II Facoltà di Ingegneria

Corso di INGEGNERIA INFORMATICA

Laurea in SISTEMI DISTRIBUITI

Configurazione di un portale XWiki per la visualizzazione su dispositivi mobili: il caso di APICe

Candidato

Andrea Buscarini

Relatore

Prof. Andrea Omicini

Correlatore

Nazzareno Pompei

Anno Accademico 2011/2012 - Sessione II

A Giulia, la mia parte migliore,
che spesso mi sopporta e allo stesso tempo mi fa agitare,
ma senza di lei le giornate non avrebbero lo stesso sapore;
alla mia famiglia che mi ha dato tante possibilità,
a volte negandosele, e ancora piú valori;
agli amici che mi sono stati sempre vicini in questo viaggio,
e che senza dubbio lo saranno per il resto della mia vita.

Indice

Introduzione al mondo mobile	7
1 La tecnologia XWiki e i dispositivi mobile: lo stato dell'arte	11
1 Il mondo mobile	11
2 La piattaforma XWiki	16
2.1 Architettura di XWiki	18
2 Responsive Web Design	23
1 Architettura delle informazioni	25
2 Metodi implementativi del Responsive Web Design	28
3 Adattamento a dispositivi mobile del portale XWiki: identificazione del problema e possibili soluzioni	31
1 Analisi delle possibilità di progetto per il caso di XWiki	32
2 Motivazioni inerenti alla scelta presa	34
4 Adattare la Skin di un portale XWiki	37
5 Caso di studio: Skin per APICe	43
1 Analisi del portale APICe	43
1.1 Analisi del codice che genera la skin del portale APICe.	46
2 Meccanismo di autoloading e riferimenti alla versione mobile . . .	47
3 Header	49
4 Content	57
5 Footer	59
Conclusioni	63

Bibliografia	67
1 Sitografia	67
Ringraziamenti	69

Introduzione al mondo mobile

*Il Web è ben lungi dall'essere fatto,
è solo in una fase farraginoso di costruzione.*

- Tim Berners-Lee, 1999 -

Era il lontano 1969 quando aldilà dell'oceano i primi computer venivano collegati tramite un'infrastruttura embrionale che verrà poi chiamata Internet. Dal 1969 ad oggi è evidente e sotto gli occhi di tutti a quali enormi avanzamenti tecnologici si sia giunti e in che modo abbiano reso Internet quello che è oggi; basti pensare allo sviluppo che hanno subito i browser o semplicemente all'ampliarsi delle possibilità e funzionalità che la rete ha maturato. Pur avendo ampliato le vedute della tecnologia nel contesto di Internet, l'attuale obiettivo rimane fedele a quello posto al momento della sua creazione e progettazione: la condivisione di informazioni. Internet infatti, è stato progettato per sopperire alla mancanza di canali di comunicazione che permettessero lo scambio di informazioni tra gli utenti di una comunità, in particolare quella scientifica; ancora oggi tale funzione continua ad essere quella principale del Web, anche se ottenuta con metodi e mezzi tecnologici diversi. Posto questo come obiettivo, risulta facile comprendere come una tecnologia, nata anch'essa con il fine ultimo di condividere informazioni, abbia preso piede nella comunità di Internet ed oggi sia largamente utilizzata: sto parlando della tecnologia Wiki. Semplice è fornire un esempio eloquente di quanto quest'ultima sia sfruttata ed utile citando Wikipedia.com, il portale implementato tramite la tecnologia Wiki che diffonde nella rete una enciclopedia creata dalla comunità web.

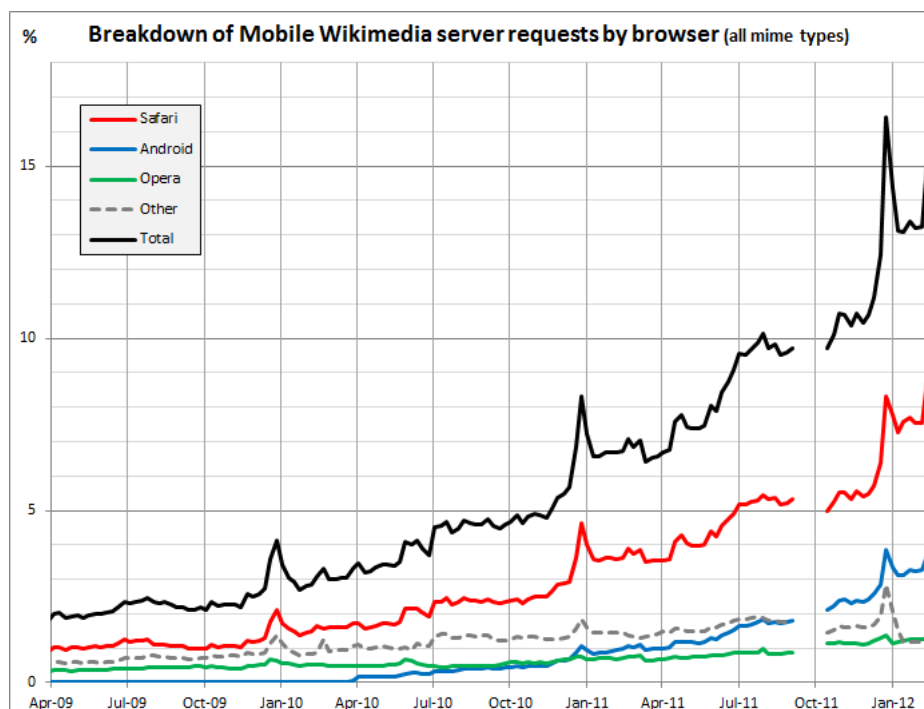


Figura 1: Wikipedia tramite mobile.

Osservando il panorama del web odierno e la sua comunità facile notare come il trend abbia negli ultimi anni invaso il mercato tecnologico delle comunicazioni, ovvero i dispositivi mobile. Essi sono oggi un elemento presente nella vita molti utenti il cui numero cresce sempre più, e ora forniscono la possibilità di accedere ad Internet e di utilizzarlo intensamente. Questa funzionalità unita alla diffusione di mobile devices di ultima generazione, ha generato il problema di integrare i contenuti e le informazioni presenti in rete, alla fruibilità su questi dispositivi i quali presentano caratteristiche hardware diverse dai normali computer con cui abitualmente accedevamo ad Internet.

Considero la tecnologia wiki uno strumento al passo con l'innovazione web odierna, soprattutto per quanto riguarda la possibilità data agli utenti web di poter non solo disporre dei contenuti, ma anche poter interagire con questi per ampliarli con esperienza e conoscenza propria. Proprio per questa capacità di rendere la comunità web più partecipe di ciò che essa stessa andrà

ad utilizzare, oltre all'aumento del numero di utenti che sfruttano portali XWiki in Internet come contenitore delle loro conoscenze, nasce l'intenzione di rendere il più adatta possibile la tecnologia Wiki con i dispositivi mobili; fornendo quindi, la possibilità di poter gestire le informazioni che queste pagine hanno insite in loro nel miglior modo possibile, anche senza essere nel proprio studio o davanti al proprio computer, ma semplicemente gestendole ovunque ed in qualunque situazione ci si trovi grazie ad i mobile devices.

Per rendere possibile questa idea ho progettato un modo per visualizzare un portale XWiki tramite dispositivi mobili, cercando la miglior soluzione possibile al fine di mostrare i contenuti del portale, facendo selezione tra quelli maggiormente utilizzati e creando una disposizione di questi tale che fosse semplice sfruttarli, attuando scelte di progettuali tipiche per i dispositivi che stiamo analizzando.

Capitolo 1

La tecnologia XWiki e i dispositivi mobile: lo stato dell'arte

In questo capitolo si parlerà dello stato attuale della tecnologia per quanto riguarda tutto il mondo dei dispositivi mobili e della tecnologia che viene implementata per creare un portale XWiki.

1 Il mondo mobile

La solitudine, oggi, è un cellulare che non squilla mai.

- Giovanni Soriano -

I dispositivi mobile spediti solamente nel terzo quadrimestre del 2012 sono circa 181 milioni di unità; numeri come questo descrivono in maniera esauritiva il fenomeno e possono far riflettere sull'importanza che il mercato del mobile possiede per quanto riguarda le industrie dell'Information Technology.

Top Six Smartphone Mobile Operating Systems, Shipments, and Market Share, Q3 2012 (Preliminary)
(Units in Millions)

Operating System	3Q12 Shipment Volumes	3Q12 Market Share	3Q11 Shipment Volumes	3Q11 Market Share	Year-Over- Year Change
Android	136.0	75.0%	71.0	57.5%	91.5%
iOS	26.9	14.9%	17.1	13.8%	57.3%
BlackBerry	7.7	4.3%	11.8	9.5%	-34.7%
Symbian	4.1	2.3%	18.1	14.6%	-77.3%
Windows Phone 7/ Windows Mobile	3.6	2.0%	1.5	1.2%	140.0%
Linux	2.8	1.5%	4.1	3.3%	-31.7%
Others	0.0	0.0%	0.1	0.1%	-100.0%
Totals	181.1	100.0%	123.7	100.0%	46.4%

Figura 1.1: Differenze tra SO spediti nel terzo trimestre del 2011 e 2012.

Un ulteriore dato indicativo ci è fornito da Juniper Research che afferma che, il numero degli utenti che utilizzeranno smartphone e tablet personali all'interno dei luoghi di lavoro sarà più che raddoppiato entro il 2014.

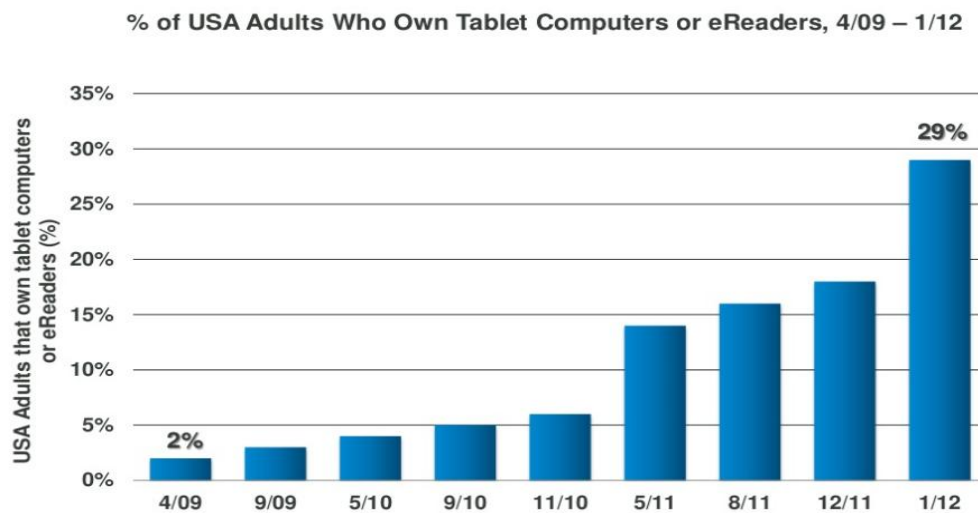


Figura 1.2: Aumento dei possessori di tablets e eReaders tral il 2009 e il 2012.

I principali motivi che hanno portato allo sviluppo dei dispositivi mobile sono:

- la possibilità di trasportare questi devices ovunque e facilmente diventando quindi il primo strumento tecnologico disponibile nel caso ne servisse l'utilizzo;
- la facilità d'uso: i mobile devices sono estremamente semplici ed immediati nell'utilizzo permettendo anche ai consumers meno esperti di averne benefici;
- multifunzionalità: questi dispositivi svolgono non solo le funzionalità basilari per cui sono stati creati, ad esempio per smarthphone quella di telefonare, ma hanno una potenza di calcolo e componenti hardware sufficienti per permettere al dispositivo di eseguire un insieme di funzionalità che vanno dal browsing di Internet allo scatto di fotografie, piuttosto che editare fogli di calcolo o fruire di contenuti multimediali.

L'interconnessione alla rete Internet, che nel corso degli anni è diventata sempre più veloce e l'estenuante diffusione dei dispositivi hanno fatto il resto; ora un dispositivo mobile è ciò che ci permette, nella maniera più semplice, di fruire ovunque della rete e di tutte le possibilità che questa ci offre, oltre ad unire tutti i devices in una sorte di comunità mobile fornendo contenuti utili all'utilizzo sia personale che lavorativo.

Anche i dispositivi mobile sono cambiati col passare del tempo, stando al passo con l'evoluzione tecnologica che ha accompagnato lo sviluppo dei PC; tali dispositivi sono passati da semplici organizer portatili, a veri e propri terminali ricchi di funzionalità, potenza di calcolo e connettività. Quest'ultima caratteristica è quella che ne ha maggiormente allargato il consumo rendendoli estremamente versatili sia per applicazioni di tipo personale che privato.

Esistono sostanzialmente cinque tipologie di dispositivi mobili, ognuna dotata di caratteristiche particolari da tenere in considerazione quando si progettano applicazioni:

- Pocket PC: sono i così detti palmari. Tra le principali caratteristiche di questi dispositivi troviamo il display di tipo touch screen, la compat-

tezza, una discreta possibilità di espansione e una discreta potenza di calcolo. I Pocket PC si dividono a loro volta in due famiglie:

- Standard (non dotati di funzionalità telefoniche)
 - Phone Edition: hanno tutte le caratteristiche dei Pocket PC standard con l'aggiunta delle funzionalità telefoniche.
- Cellulari muniti di browser per la navigazione in Rete.
 - Smartphone: sono dispositivi prettamente telefonici (sia nella forma sia nella modalità di input). Generalmente sono meno potenti dei Pocket PC. Data la tipologia ed il target di questi dispositivi, anche la loro espandibilità attraverso device esterni è limitata.
 - Tablet PC: sono dispositivi molto simili ai normali Notebook ma dotati di touch screen e di funzionalità di riconoscimento della scrittura.
 - UMPC (Ultra Mobile PC): sono una via di mezzo tra il Pocket PC e il Notebook. Più grandi e potenti di un Pocket PC ma più piccoli e meno potenti di un Notebook, sono più orientati verso il mercato consumer che business.



Figura 1.3: Evoluzione dei dispositivi

Le prime due tipologie di dispositivi sono ormai poco diffuse in quanto obsolete. Cellulari e PocketPC infatti, sono stati soppiantati dai moderni smart-

phone che ne integrano le funzionalità, all'interno di un unico dispositivo.

Lo sviluppo software è andato di pari passo con quella hardware; il particolare mercato ha subito nel corso degli anni una netta e distintiva separazione tra i due sistemi operativi che seppur diversi tra loro, sono riusciti a capire ciò che i consumers chiedevano in materia di connettività e social activities.

Questi sistemi sono iOS di Apple e Android di Google, che dettano legge sul mercato software essendo installati nella quasi totalità dei dispositivi in circolazione. Precedentemente a questi sistemi il mercato dei primi cellulari intelligenti era suddiviso tra i sistemi operativi Symbian (Nokia), Blackberry OS (RIM) e Palm OS (Palm).

Attualmente oltre a quelli citati ovviamente, esistono tanti altri OS mobili come per esempio WebOS (di HP), che ha preso il posto di PalmOS, e Mee-go, una piattaforma open-sorce sviluppata da Nokia e Intel, ed il più noto Windows Phone di Microsoft.

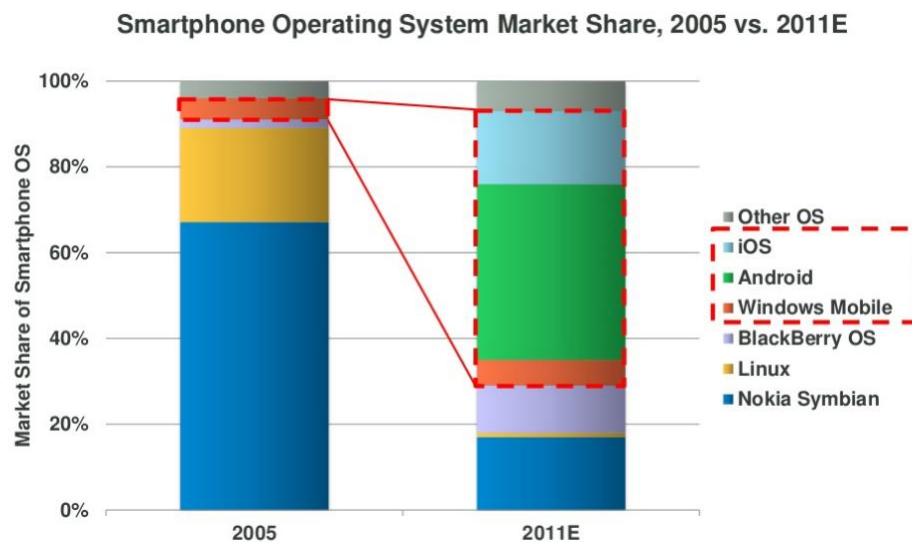


Figura 1.4: Evoluzione dei dispositivi

2 La piattaforma XWiki

*Internet è la più grande biblioteca del mondo.
È solo che tutti i libri sono sparsi sul pavimento..
- John Allen Paulos -*

Una pagina Wiki si può definire come uno spazio in cui i suoi utilizzatori creano e modificano i contenuti, arricchendoli con le proprie conoscenze personali, in modo da ampliare la conoscenza di un determinato argomento e condividerla liberamente in rete. Si tratta quindi di uno strumento che prende i lati positivi di una conoscenza condivisa tra una comunità di utenti e rende disponibili i contenuti modificabili e ampliabili in maniera aperta.

Le pagine Wiki sono considerate uno dei modi più semplici ed affidabili per diffondere e ampliare la conoscenza nella rete, ciò è confermato dall'enorme rilevanza che ha la più famosa piattaforma Wiki nel web: Wikipedia. Questa enciclopedia, fondata nel 2001, ha visto accrescere i propri contenuti grazie all'esperienza ed alla conoscenza personale della comunità web che ha interagito con la piattaforma creando, per la sola sezione del sito italiano, 981711 pagine.

Andando nel dettaglio, la piattaforma Wiki consente la scrittura dei contenuti tramite linguaggi di markup, in particolare, il linguaggio utilizzato viene chiamato wikitext e si differenzia in maniera netta dai più famosi linguaggi di markup come l'HTML in quanto non ha le funzionalità che quest'ultimo offre e non permette l'iterazione diretta con javascript o CSS; sarà il motore wiki a tradurre le informazioni in una pagina html, leggibile e interpretabile da ogni browser.

Sono quindi evidenti i limiti presenti nelle pagine Wiki, che non concedono la possibilità di ampliare la qualità dei contenuti presentati e di sfruttare le potenzialità di linguaggi di programmazione di più alto livello essendo negata l'iterazione di questi con la piattaforma. Per sopperire ai limiti del motore Wiki, nasce la piattaforma XWiki.

XWiki è una piattaforma web open source sviluppata sfruttando il linguaggio Java, all'interno di un motore wiki simile a quello utilizzato per wikipedia, il cui principio fondamentale è quello di creare web applications, orientate per lo più alla condivisione di informazioni tramite la collaborazione di una comunità di utenti, dove ognuno può contribuire alla creazione e presentazione di contenuti, proprio come le pagine wiki, concedendo però, la possibilità di interagire con paradigmi e linguaggi di programmazione diversi che consentono di ampliare le possibilità di modellazione dei contenuti.

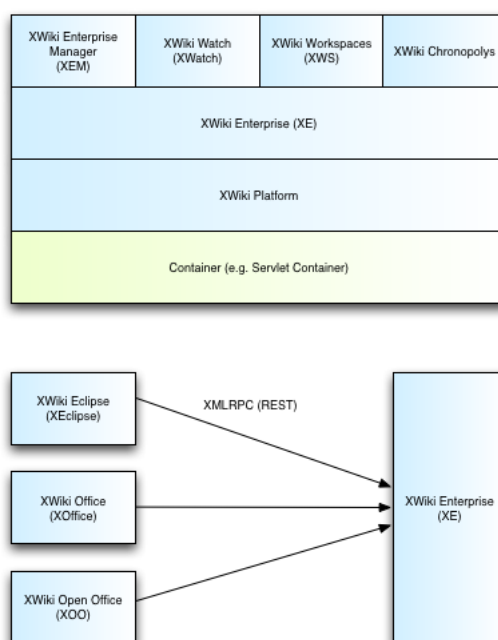


Figura 1.5: Ecosistema XWiki.

Si evince quindi che non è un semplice motore wiki in quanto fornisce molteplici possibilità e possiede caratteristiche interessanti che consentono alla piattaforma XWiki di creare applicazioni funzionali per diversi ambiti, anche quello aziendale. Esso è strutturato in modo che gli utilizzatori possano crearsi i loro space privati per collaborazioni, o pubblicazioni di documentazione di applicazioni, sia open source che non. In tutti questi campi l'architettura XWiki può essere una soluzione ideale, sia per semplicità di installazione e configurazione, sia che per la facilità d'uso dell'interfaccia di amministrazione dello space ed inoltre per la facilità di gestire una quantità di pagine non

banale, ciò è possibile grazie alle funzionalità estese di XWiki; rispetto alle classiche pagine wiki, che consentono di utilizzare strutture dati indicizzate e ricercabili all'interno della pagina.

La vera forza di XWiki sta però nella sua realizzazione; essa permette di creare pagine dinamiche, grazie all'utilizzo di script che generano codice html implementati con linguaggi di scripting come Velocity e Groovy, che permettono di estendere le funzioni base. Possiede inoltre API versatili che unite alla possibilità di editare script, consentono di realizzare vere e proprie applicazioni all'interno di una pagine web. È infatti possibile creare pagine che richiama contenuti, o che ne generano di nuovi oppure ne modificano dinamicamente gli attributi, piuttosto che spedire email. Sono proprio queste funzionalità a fare di XWiki una piattaforma che può interagire in maniera efficiente con gli ambiti più disparati, rispondendo ad una varietà enorme di necessità degli utilizzatori. Un ulteriore vantaggio di questa piattaforma risiede nell'ottenere tutto ciò che ho elencato lavorando sempre ad alto livello, senza toccare il core, ma scrivendo codice tramite un comune browser; ciò rende XWiki uno strumento facilmente utilizzabile anche da tutti quegli utenti che non hanno ferrate basi informatiche.

XWiki è inoltre compatibile con tutti i principali servlet container, grazie al fatto che è completamente implementato in Java, e con questi vengono estese le applicazioni hostate dai web server.

2.1 Architettura di XWiki

Il sistema XWiki è in realtà composto da più componenti ma ne andremo ad analizzare in particolare uno: XWiki Enterprise. Esso possiede le features più evolute che stanno al passo con le applicazioni del Web 2.0. XWiki Enterprise infatti permette di controllare e implementare blog, rights management, pdf export, skinning, supporto per advanced forms e scripting avanzato.

L'architettura di XWiki fonda le proprie radici maggiormente su strutture open sources, come:

- DBCP per la gestione del pooling delle connessioni al database;

- JRCS per il versioning degli attachment;
- Hibernate per la gestione del database: middleware che gestisce le richieste alla base di dati e la relativa memorizzazione;
- Struts per la gestione degli URL scheme;
- Velocity per la gestione delle viste e per la programmazione in generale delle pagine;
- Radeox come motore per il rendering;
- OSCache per la gestione della cache e dei documenti;
- OSUser e Security Filter per la gestione dell'autenticazione e per le autorizzazioni.

La piattaforma XWiki offre APIs e UI comuni; nel service APIs sono compresi il Core, i plugins e i moduli, in UI sono compresi skins, templates e applications.

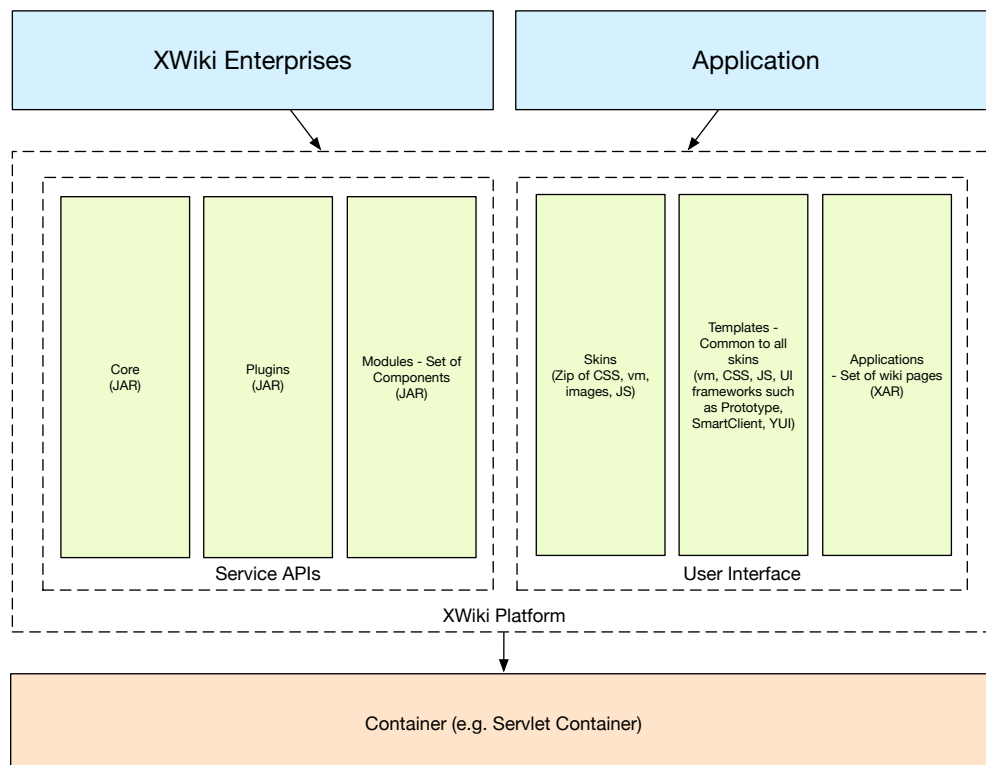


Figura 1.6: Architettura XWiki.

Analizzando singolarmente i componenti che formano le services APIs, notiamo che l'XWiki Core è sin dalle origini del progetto un singolo .jar, suddiviso in diversi moduli distinti che implementano molte features, tra i quali:

- alcuni moduli fondamentali che contengono al loro interno l'effettiva implementazione delle classi che rappresentano il modello wiki (Document, Space, Class/Objects, Attachments per esempio);
- la gestione delle cache che recentemente è stata sostituita da un nuovo Cache Module;
- il syntax rendering, che coordina sia la vecchia versione, XWiki Syntax 1.0, che la nuova versione 2.0, in modo da poter gestire entrambi i formati e non obbligare gli utenti alla migrazione;
- localizations: ovvero la traduzione in vari linguaggi (è in fase di sviluppo una nuova versione del modulo);
- exports: modulo che consente di ottenere un rendering dei contenuti nei formati PDF, RTF, WAR;
- security: i servizi di autenticazione dell'utenza e gestione delle autenticazioni;
- user management.

I plugins si suddividono tra quelli .jar e quelli che sono situati nell'XWiki Core JAR. Svolgono funzioni come:

- Activity Stream che gestisce gli eventi immagazzinandoli in un database dedicato;
- Application Manager per amministrare le applicazioni;
- Calendario;
- Mail Sender;
- Scheduler dei job dei singoli Xobjects;
- Skin Extension, consiste nel modificare il layout e la visualizzazione del wiki;
- Tag;
- Watch List;
- Wiki Manager;
- gestione dei Captcha;

- Feed;
- File Upload;
- visualizzazione e creazione di grafici e di immagini tramite GraphViz e Image;
- coordinare le Query;
- Zip Explorer.

Tutti questi plugin sono facilmente scaricabili tramite lextensions wiki, sebbene il componente plugin sta per essere abbandonato in favore dei components, che definiscono un nuovo modo di eseguire funzionalità specifiche tramite componenti esterni alla piattaforma che devono essere installati.

I moduli offrono servizi in un dato dominio equivalenti a quelli che possono offrire i plugins. La differenza tra moduli e plugins risiede nell'architettura che li implementa, infatti i moduli sfruttano la nuova XWiki component-based architecture, quindi il loro sviluppo è component oriented; mentre i plugin, come dicevamo sono extension oriented. XWiki controlla la modularità tramite un manager indipendente chiamato lightweight component manager; tale scelta è stata attuata andando controcorrente; esistono infatti diversi Component Manager esterni alla piattaforma XWiki che si basano su Java come: Guice, OSGi o CDI. L'utilizzo di questi avrebbe reso più semplice l'utilizzo e meno oneroso lo sviluppo che sarebbe stato lasciato a terze parti. XWiki ha però scelto di essere indipendente dalle soluzioni esistenti, implementando semplici Component interface che possono poi essere collegate ad un qualsiasi component manager. Questa volontà è stata dettata, oltre che dall'indipendenza da strutture esterne, anche dalla mancanza di un vero e proprio standard definito in Java per il sistema component oriented.

Le applications sono la parte della piattaforma XWiki più vicina all'utente e alle sue esigenze applicative, infatti, come già citato, all'interno delle pagine XWiki è possibile inserire codice che, unito alla possibilità di combinare i contenuti di più pagine, possono portare ad ottenere vere e proprie applicazioni che risiedono all'interno del nostro wiki space. Una proprietà fondamentale delle applications è la loro riutilizzabilità: ovvero possono essere esportate in

altri XWiki. Il vero punto di forza della piattaforma XWiki sono le applicazioni che rendono quest'ultima più funzionale ed adattabile ad ogni ambiente rispetto alle semplici pagine wiki, anche perché sono capaci di rendere semplici pagine degli strumenti in grado di eseguire anche algoritmi complessi, piuttosto che sfruttare ed organizzare strutture dati articolate.

L'ultimo aspetto che voglio analizzare è la capacità di XWiki di estendersi, dovuta soprattutto alla comunità di sviluppatori che vi si trova dietro. I modi per interagire con la piattaforma ed estenderla sono molteplici:

- Scrivendo script nelle pagine dellXWiki;
- Scrivendo applicazioni, set di pagine wiki;
- Scrivendo plugins in Java;
- Scrivendo moduli, set di componenti, in Java;
- Progettando skin o rielaborando quelle esistenti,
- Estendendo l'esistente services APIs, controllando con attenzione ed in maniera preventiva la documentazione per verificare se siano stati forniti particolari spazi nel codice in cui è possibile estenderlo.

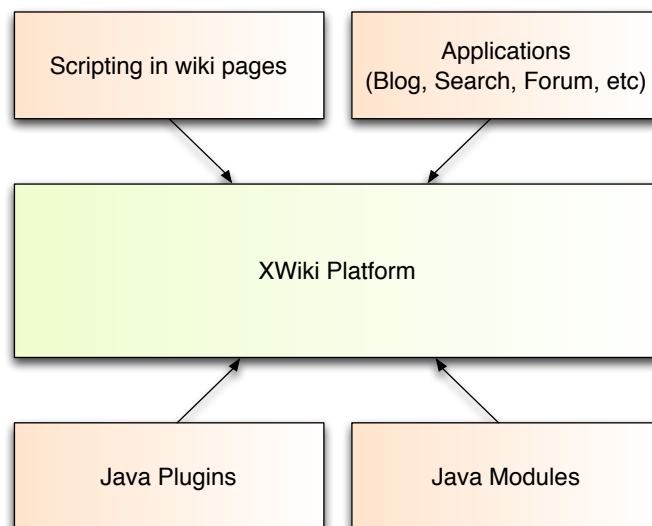


Figura 1.7: Piattaforme di estensione di XWiki.

Capitolo 2

Responsive Web Design

*The control which designers know in the print medium,
and often desire in the web medium, is simply a function of the limitation of the
printed page.*

*We should embrace the fact that the web doesn't have the same constraints, and
design for this flexibility.*

But first, we must accept the ebb and flow of things.

- John Allsopp -

L'estenuante distribuzione dei dispositivi mobile ha portato a riconsiderare come i contenuti web debbano adeguarsi ad essere fruibili anche per questi devices. Infatti è ormai consolidato il fatto che non si naviga più solo ed esclusivamente da Pc Desktop, ma anche su smartphone e tablet, o addirittura TV.

Il problema del rendering dei contenuti nasce dal fatto che i mobili devices hanno schermi differenti rispetto ai normali desktop, oltre ovviamente a risoluzioni diverse. Il 480 x 320 è una risoluzione ormai superata, le tendenze sono sempre crescenti, ora si parte da una risoluzione minima, presente negli smartphone, di 960 x 640 per quanto riguarda quelli usciti da qualche anno nel mercato e fino a 1280 x 720 per quelli di ultima generazione; si raggiungono risoluzioni dell'ordine di 2048 x 1536 per i tablet di ultima generazione. Ciò ha quindi dato motivo ai web designers di riconsiderare la progettazione dei contenuti in rete, rendendoli adatti anche a mobile devices.



Figura 2.1: Esempio di Responsive Web Design.

La filosofia di progettazione, oggi largamente sfruttata e considerata un'ottima soluzione si chiama Responsive Web Design. Questo modo di pensare definisce fondamentale la continua ricerca di flessibilità in fase di progettazione, sia per quanto riguarda l'adattamento del layout del web site per ogni tipo di dispositivo, sia per la scelta dei contenuti che saranno disponibili per ogni tipo di devices, identificando quindi, obiettivi ed uno stile di navigazione diverso in base a quale dispositivo utilizziamo.

Il Responsive Web Design è stato citato per la prima volta in un articolo di Ethan Marcotte del 2010, esso spiega le origini del nome che proviene da una disciplina emergente chiamata Responsive Architecture la si interroga su come debbano cambiare gli spazi fisici attraversati da persone. Il libro spiega come questo approccio architettonico sia formato da un sistema di loop multipli che formano un continuo e costruttivo scambio di informazioni. L'autore considera la mutuale influenza che hanno abitante e struttura e l'approccio architettonico che si ottiene, in più a quale effetto possa portare la relazione tra web e mobile device, indicando come maggiormente performante il trattare l'adattamento di un web site a più devices tramite una progettazio-

ne flessibile piuttosto che adattare progetti scollegati tra loro ad un numero sempre crescente di dispositivi. Le diverse progettazioni sono quindi sfaccettature della stessa esperienza.

Analizzando il *modus operandi* per la progettazione responsive si percepisce il bisogno di chiarire alcuni punti prima di iniziare a organizzare il lavoro, in particolare:

- **Identificare gli obiettivi:** perchè vogliamo sviluppare una versione mobile del nostro sito? La prima cosa da fare consiste, senza dubbio, nell'identificare le necessità comunicative che si desidera soddisfare offrendo i propri contenuti agli utenti in mobilità. Una volta identificato il nostro obiettivo potremo metterci al lavoro per perseguirlo.
- **Selezione dei contenuti:** una volta identificato l'obiettivo comunicativo è necessario fare un'attenta selezione dei contenuti da offrire scartando a priori quelli non pienamente compatibili o, comunque, inadatti a questo tipo di device.
- **Progettare il layout:** una volta selezionate le informazioni da offrire (il contenuto) è necessario concentrare le proprie energie nella progettazione/creazione di un contenitore adeguato. In tal senso potremmo optare per la progettazione di un duplice design: uno destinato agli schermi, piuttosto piccoli, degli smartphone ed uno per i display, un po' più grandi, dei tablet. Per fare ciò sarà sufficiente, in realtà, creare due differenti CSS in cui gli elementi della pagina (che sono comunque i medesimi sia per smartphone che per tablet) vengono stilizzati diversamente a seconda della grandezza del display.

1 Architettura delle informazioni

Nel contesto web per dispositivi mobili è fondamentale strutturare l'informazione nel modo più semplice possibile.

A differenza dei siti web adatti ad una visualizzazione sul monitor di un PC, dove l'utente ha almeno un'idea, una visione generale della struttura del sito, l'architettura informativa di un sito visualizzato sui dispositivi mobili invece

non è immediata proprio a causa delle limitate dimensioni dei loro schermi. Caricare le pagine di eccessive informazioni può generare un senso di frustrazione e disorientamento agli utenti.

Alcuni consigli, a tal proposito, ci vengono forniti dal World Wide Web Consortium (W3C) tramite "Mobile Web Best Practices 1.0" dove vengono date utili raccomandazioni riguardanti: il contenuto, le logiche di navigazione, il peso delle pagine, lo scrolling delle pagine, etc.

Per quanto riguarda il Page content si deve garantire che questo sia:

- Adeguato: il contenuto deve essere adatto all'uso in un contesto mobile;
- Chiaro: l'utilizzo di un linguaggio semplice dovrebbe essere una buona norma da seguire;
- Limitato: ossia deve essere rispondente alla richiesta ricevuta. L'esigenza di adattare il contenuto al contesto d'uso, limitandolo all'essenziale, è dettato in gran parte dai limiti tecnici e fisici del dispositivo (dimensioni ridotte, tempi di caricamento lunghi dovuti al tipo di connessioni mobili che sfruttano i mobile device).

Il W3C consiglia inoltre di porre attenzione a ciò che è presente nella navigations bar: bisogna infatti assicurarsi che il materiale che ha un ruolo centrale per il significato della pagina preceda quello che non lo ha.

Un altro punto focale per la progettazione mobile è il peso che ogni web page deve avere, questa grandezza deve essere appropriata in relazione alle limitate memorie di questi dispositivi. Pagine leggere rendono la navigazione molto più piacevole da parte dell'utente, che vedrà ridotti i tempi di attesa dovuti al caricamento delle pagine.

Prendendo in esame solamente alcuni dei punti delle "Mobile Web Best Practices 1.0" si evince già chiaramente che, un sito web per mobile non può essere una mera trasposizione del sito ottimizzato per PC. Notiamo infatti, che i siti ottimizzati per PC seguono degli schemi ben precisi, ovvero: una testata con un logo e un menu di navigazione principale, una colonna sinistra o destra

con un menu di navigazione secondario, una colonna per i contenuti, e un piè di pagina. Evidentemente, tale schema non può garantire una visualizzazione ottimale anche su dispositivi mobili a causa dei loro limiti fisici; quindi una progettazione basata sull'uso di un'impaginazione non troppo complessa, gestita, ad esempio, per mezzo di unità di misura relative e proporzionali, favorirà ulteriormente l'adattamento tentato dal browser, consentendo una navigazione tra i contenuti più adatta ad una fruizione su schermi piccoli come quelli di un cellulare, di uno smartphone o di un palmare. Pertanto, durante la fase progettuale è bene tener conto di questi fattori: un layout a due o più colonne superiore ai 320 pixel di larghezza, ad esempio, potrebbe generare un fastidioso scrolling orizzontale con quei browser che non sono in grado di adattare i contenuti alla risoluzione dello schermo.

Proprio lo scrolling è un altro aspetto che W3C analizza indicando che è bene limitare lo scrolling ad una sola direzione, a meno che lo scrolling secondario non possa essere evitato. È buona norma, pertanto, realizzare schemi di navigazione più semplici possibili:

- Preferire il layout ad una colonna;
- Mostrare solo le opzioni relative alla pagina visualizzata;
- Rendere evidenti gli strumenti di navigazione;
- Limitare lo scrolling verticale;

Ecco un esempio schematico di layout per mobile:

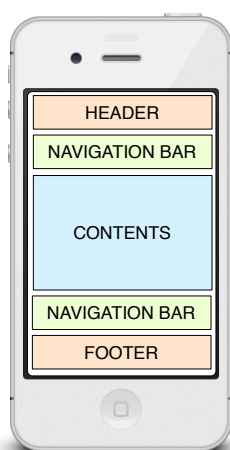


Figura 2.2: Organizzazione dei contenuti per il Responsive Web Design.

Al di là della testata che rappresenta la vera e propria intestazione del sito web, è da notare come il menu di navigazione sottostante la testata sia ripetuto anche in fondo alla pagina, subito dopo i contenuti centrali.

Ottimi ausili alla navigazione sono anche le cosiddette briciole di pane (breadcrumbs) che indicano la posizione gerarchica della pagina che si sta consultando.

2 Metodi implementativi del Responsive Web Design

La progettazione responsive offre vari metodi implementativi che è bene definire prima di iniziare a progettare effettivamente il web site. Tra le possibilità, quelle principali sono quelle di realizzare un nuovo sito web appositamente concepito per la visualizzazione su dispositivi mobili (sito da hoc) oppure optare per l'adattamento del sito normale alle nuove esigenze della mobilità (versione adattata).

Ad una prima analisi, creare una versione adattata del sito potrebbe sembrare la scelta più semplice, in realtà, nasconde una serie di insidie che la rendono la scelta sconsigliata; questo perché si presuppone che il sito originale sia realizzato con alcune perizie mediante le quali si separi nettamente contenuti e layout. Se tali premure non vengono considerate allora la portabilità del sito originario su dispositivi mobile diviene complessa. Questa scelta rimane però quella consigliata, anche da W3C, in quanto porta una serie di vantaggi, come ad esempio ridotti costi di gestione, basterà infatti aggiornare i contenuti utilizzati in un sito e questi verranno effettivamente aggiornati in tutte le versioni. Tuttavia vi sono alcuni punti critici che è bene considerare nel momento in cui si sceglie la strada che permette di creare una versione adattata del sito, in particolare, se l'adattamento alle versioni mobile avviene su un sito web precedentemente creato per navigazione da pc:

- La probabilità di effettuare errori è molto frequente, specie se si riadatta un sito già creato per la navigazione desktop, perchè taluni elementi

eccedono nella larghezza del display ad esempio;

- La fruizione dei contenuti su dispositivi mobili richiede semplicità e sintesi, mentre a volte i siti progettati per l'utilizzo sui comuni computer sono ricchissimi di informazioni che potrebbero rendere piuttosto lenta e difficoltosa la navigazione in mobilità;
- Alcuni elementi ricorrenti nelle comuni pagine web, come ad esempio animazioni in flash, non sono supportati da tutti i device mobili (ad esempio da iPhone e iPad); Alcuni elementi di interazione (come i form) sono difficilmente adattabili alla fruizione in mobilità (o meglio, lo sono ma solo attraverso un'attenta riprogettazione);
- Alcune funzionalità (come ad esempio il download di file) non sono universalmente supportate dai dispositivi mobili e potrebbero causare errori o malfunzionamenti poco gradevoli.

In maniera differente, la creazione di una versione ad hoc fornisce un controllo totale sull'aspetto del sito e sulle informazioni fruibili attraverso di esso. Realizzando una versione appositamente concepita per i dispositivi mobili si correranno minori rischi di commettere errori e sarà possibile adattare ogni cosa alle peculiarità del device di destinazione. Tra i metodi implementativi che sfruttano questa strada, hanno una rilevanza maggiore e per questo sono i metodi più utilizzati:

- Creazione di una Web App: possiamo definire una Web Application come una via di mezzo tra una App ed un sito web. In sostanza si tratta di pagine web evolute in grado di offrire funzionalità complesse simili a quelle fornite dalle comuni Applicazioni installate sul dispositivo. Ciò è possibile grazie ai linguaggi di scripting(sia lato client che lato server) che consentono di creare degli applicativi veri e propri (fruibili attraverso un browser) simili, anche dal punto di vista della interfaccia utente (UI), alle comuni App. Ciò è possibile grazie a tecnologie come AJAX che consentono di riprodurre in ambiente di Rete comportamenti tipici delle interfacce software, ad esempio gestendo aggiornamenti dei contenuti senza effettuare alcun refresh di pagina. Mentre una comune App è installata fisicamente sul dispositivo, una Web-App consiste, in una visione semplicistica, in un link verso un applicativo remoto. Una simile

soluzione comporta lo svantaggio di richiedere, ai fini di un corretto funzionamento, la disponibilità di un accesso alla Rete ma ha il vantaggio di non incidere minimamente sulle capacità di storage del dispositivo e di essere sostanzialmente indipendente dalle capacità di calcolo dello stesso; la velocità di una Web-App dipende, infatti, dalla bontà della connessione alla Rete e dalle performances del server remoto nell'offrire l'elaborazione richiesta dall'utente.

- Creazione di un sito copia a sè stante adatto ai dispositivi mobile: questa scelta è stata intrapresa per esempio da piattaforme come YouTube.com, la quale infatti reindirizza al sito www.m.youtube.com se si accede tramite mobile devices. Questa scelta consente al consumatore una navigazione del tutto simile a quella si che effettua da desktop, con la differenza che i contenuti saranno adattati al device con il quale naviga. Inoltre questa implementazione non costringe il consumer a fare il download di contenuti che dovranno poi essere memorizzati nel dispositivo e quindi occupare delle risorse, seppur in maniera minima, come avviene per le web-app.

Tra gli strumenti che ci aiutano nel compito di progettare versioni dedicate ha una rilevante importanza il linguaggio CSS3, in particolare la signature @media, la quale permette di impostare diversi layout in base alla dimensione dello schermo del dispositivo che richiede la pagina web. Oltre al CSS3, è presente anche l'XHTML MP ossia l'XHTML Mobile Profile; nasce dalle specifiche di XHTML fornite da W3C insieme alla necessita di creare un linguaggio che supporti le tecnologie mobili. XHTML MP rappresenta il formato figlio di XHTML e, pertanto, ne eredita le caratteristiche formali e la filosofia: codice più leggero, pulito da tutti gli elementi di formattazione e valorizzato semanticamente.

Capitolo 3

Adattamento a dispositivi mobile del portale XWiki: identificazione del problema e possibili soluzioni

L'importante è continuare a fare domande.

- Albert Einstein -

La capillare diffusione di smartphone e tablet rende indispensabile un adattamento delle comuni tecniche di web design al fine di realizzare contenuti web idonei ad essere visualizzati correttamente su questo tipo di dispositivi. In linea con ciò si hanno due principali soluzioni: le applicazioni, più comunemente chiamate app, ed il re-design dei siti web in una versione mobile.

Una mobile app è un'applicazione software progettata per mostrare su dispositivo mobile il web site a cui è riferita inoltre è scaricabile gratuitamente dalla rete. Una particolarità di queste app è che sono ben diverse rispetto a quelle native (per esempio quelle scaricabili da piattaforme dedicate come AppleStore o AndroidMarket), in quanto non vengono eseguite sul dispositivo, se non per qualche script di animazione grafica, lasciando la parte di iterazione tra dispositivo e rete al server, proprio come un normale web site; infatti una volta scaricate, continuano comunque ad utilizzare il browser an-

che se in una modalità di background.

Il re-design di web site consiste nel considerare, a design time, la fruibilità dei contenuti di un sito web anche attraverso dispositivi mobili; ciò consiste nel fare una versione dedicata per questi devices, che hanno schermi ridotti rispetto a normali desktop ed un funzionamento che ovviamente non sfrutta nè tastiera nè puntatore. Questo iter progettuale, seguendo le regole del responsive web design, adatta, in un unico progetto, i contenuti web originari a varie tipologie di dispositivi mobili.

La piattaforma XWiki ha il grande vantaggio di offrire un'ampia serie di funzioni; tale vantaggio potrebbe tramutarsi in complicazioni se volessimo rendere disponibile il sistema per dispositivi mobili. Infatti, come ampiamente descritto nei capitoli precedenti, i mobile devices si differenziano per tutta una serie di limitazioni dai comuni pc, soprattutto, per quanto riguarda dimensioni dello schermo e potenza di calcolo. Analiziamo ora come queste possibilità vengono implementate per la piattaforma XWiki.

1 Analisi delle possibilità di progetto per il caso di XWiki

La prima possibilità consiste nel creare una web app, per esempio implementandola con HTML5 per renderla portabile sulla maggior parte dei dispositivi. Creando un web app, si accede alle informazioni del portale XWiki tramite il database sul server dove il portale effettivamente è memorizzato, dopodichè si selezionano i contenuti da presentare tramite query. L'app sarà accessibile tramite redirect ad un altro URL oppure scaricabile dal sito per poter accedervi direttamente dal dispositivo, senza dover passare dal browser. Questa soluzione richiede la necessità di implementare delle interrogazioni al database ed una serie di accorgimenti per il server; indirizzando maggiormente questa scelta al non considerare a design-time situazioni che portano ad errori. Un altro aspetto preso in considerazione da questa soluzione consiste nel non sfruttare effettivamente le funzionalità che la piattaforma XWiki

fornisce; poiché dal portale, per accedere all'app, è necessario trasferirsi in un'altra struttura diversa da XWiki che contiene le informazioni richieste prendendole direttamente da un livello inferiore dell'architettura del sistema XWiki, il database, senza quindi sfruttare i vantaggi di cui il middleware del portale si fa forza, come la possibilità di nascondere il database stesso semplicemente accedendovi tramite plugin a livello superiore.

La seconda possibilità consiste nel cambiare e customizzare il layout del wiki adattandolo alla visualizzazione dei contenuti a dispositivi mobili. Ci sono due modi per customizzare lo skin di una piattaforma XWiki:

- modificare la skin inserendo un nuovo pacchetto skin che si attiva modificando l'URL dinamicamente tramite uno script velocity;
- utilizzare il plugin XWiki Skin Extension;

Il primo metodo consiste nell'inserire appositi script e template nel codice eseguito dalla piattaforma XWiki in modo da riconoscere il dispositivo da cui è partita la richiesta di visualizzazione dei contenuti del portale XWiki ed in base al tipo di dispositivo carichiamo un apposito skin. Questo è possibile perché tra le peculiarità di XWiki vi è il caricamento delle skin e di tutti i file inerenti allo style delle pagine XWiki; in particolare, XWiki segue uno stack di caricamento tra file di diversa natura che ha poi come risultato finale quello di utilizzare le modifiche sovrascritte per ultime. L'ordine utilizzato sarà il seguente:

- la piattaforma inizialmente effettua il load dei template, in particolare quelli inseriti nella cartella templates come `view.vm`, `footer.vm`, `header.vm` ed ulteriori template scritti in Velocity;
- il sistema darà priorità ai template idealmente inseriti nello stack di caricamento per ultimi; quindi, se si importa nel codice un template avente lo stesso nome di uno precedente importato, questo sarà quello che effettivamente verrà eseguito. Questa procedura consente quindi di importare ulteriori template o aggiungerne di nuovi, indirizzandoli dentro l'apposita cartella che contiene tutto il codice per il nuovo layout. Questa cartella sarà nella cartella Skins, comprendente tutti i layout possibili della piattaforma XWiki;

- aggiungendo funzionalità o modificando le esistenti sovrascrivendo il codice, editando la pagina della Skin utilizzata nel XWiki direttamente dal browser, nella sezione apposita Presentation contenuta nello spazio dedicato all'Administrator.

In maniera differente l'XWiki Skin Extension è un meccanismo che permette di modificare tutto il wiki oppure solamente alcune pagine, senza aver bisogno di cambiare il proprio skin templates o il suo stylesheet. Lo Skin Extension plugin fornisce la possibilità di inviare al browser file Javascript, CSS o Velocity extra che non fanno parte dell'attuale skin del wiki.

Il primo aspetto fondamentale consiste nel capire come viene definita una Skin Extension; questa infatti è implementata come un XWiki Objects, realizzabile come tutti gli oggetti dal proprio browser tramite la funzione edit che il portale fornisce. Ci sono poi due principali estensioni che attualmente sono supportate dalla piattaforma:

- JavaScript eXtensions: implementata dall'XWiki Objects di classe XWiki.JavaScriptExtension;
- StyleSheet eXtension: incarnata come un XWiki Objects di classe XWiki.StyleSheetExtension.

Utilizzando il primo metodo si ottengono, a mio avviso, risultati migliori in quanto, si ha la possibilità di modificare più contenuti ed attuare customizzazioni più estreme. I punti a sfavore di questa metodologia risiedono nella quantità e complessità del codice da editare rispetto all'altra opzione, in questo modo però si riescono a rispettare i consigli di progettazione del Responsive Web Design. La seconda opzione infatti permette di editare lo skin direttamente dal browser senza dover importare alcun file all'interno del portale, ciò va a scapito delle modifiche apportate che saranno limitate, soprattutto per quanto riguarda l'effettivo comportamento del sistema.

2 Motivazioni inerenti alla scelta presa

Analizzando le possibili soluzioni ho scelto di implementare quella che agisce sulle skin della piattaforma XWiki. La mia scelta è dovuta ad una serie di motivi, in particolare questi sono:

- L'utilizzo di questa soluzione permette di editare tutti contenuti del sito: in questo modo posso infatti fare il rendering su dispositivi mobili di tutto ciò che è presente nel sito, ed in più avere la possibilità di farlo in maniera dinamica; questa significa che i contenuti che sono modificati nel sito nella versione per desktop variano direttamente anche nella versione mobile proprio perché vado ad agire sulle tecniche di presentazione e non sui contenuti, senza implementare una nuova struttura dove memorizzare i contenuti, ma agendo solamente sulla skin.
- Questa soluzione segue le linee guida del Responsive Web Design, soprattutto per quanto riguarda i contenuti. Infatti scegliendo di adattare la piattaforma a dispositivi mobili si possono rendere disponibili tutti i contenuti o solo quelli più inerenti all'uso da mobile. I contenuti mostrati, possono appartenere anche a sezioni del sito completamente slegate semanticamente, ma avere in comune l'essere utili per un consumer mobile.
- La soluzione scelta è, secondo me, più performante perché non necessita di essere installata nel dispositivo e rende quindi la navigazione il più simile possibile a quella da normale pc. Un altro motivo per il quale considero questa scelta più performante risiede nella possibilità di avere a disposizione tutti i contenuti del sito se lo adatto a dispositivi mobili in modo da non limitare la navigazione dell'utente il quale ritroverà gli stessi percorsi di navigazione del sito nella versione normale.
- Adattando la piattaforma XWiki, senza ricrearla in un'altra architettura, si adattano anche tutte quelle funzionalità di controllo che sono presenti sul sito nella versione desktop. Un esempio rilevante è visibile con gli algoritmi di login, che saranno molto più facili da importare nella versione mobile con la scelta di adattare la piattaforma, oppure la funzionalità che permette di editare la pagina wiki tramite browser.
- Analizzando poi la possibilità di modificare i contenuti o le funzionalità della versione mobile in futuro, avendo scelto di adattare il portale XWiki, questo potrebbe mutare senza limitazioni ed in maniera meno onerosa, rendendo la soluzione considerata come la migliore, a parere mio.

Capitolo 4

Adattare la Skin di un portale XWiki

*Possiamo vedere solo poco davanti a noi,
ma possiamo vedere tante cose che bisogna fare.*

- Alan Touring -

In questo capitolo andremo ad osservare il funzionamento del plugin XWiki Skin Extension, in particolare quello relativo alla creazione di una Skin dedicata ai dispositivi mobili. Sarà importante notare come le potenzialità che si celano dietro la piattaforma XWiki, vengano svelate ogniqualvolta si implementi o si editi qualcosa su di essa.

Il plugin Skin Extension è un meccanismo ideato per personalizzare il layout della piattaforma, applicando la skin creata ad alcune pagine o particolari spaces o decidere di espandere la skin su tutto il portale. La modifica dello skin avviene senza agire su fogli di stile o template, ma tramite browser, dove si possono aggiungere file, come file CSS o Javascript. Lo Skin Extension Plugin è parte integrante della versione Enterprise di XWiki e permette di aggiungere XWiki Objects che andranno poi a modificare lo skin.

Gli XWiki Objects sono istanze di una classe che permette di definire con valori univoci le proprietà dell'oggetto. Le proprietà sono definite dalla classe Objects, ed ogni Objects può essere allegato ad una pagina; nello stesso modo ad una pagina si possono collegare più oggetti. Utilizzando gli XWiki Objec-

ts è possibile inserire, all'interno della pagina a cui è collegato, strutture di informazioni; questo aspetto differenzia la piattaforma XWiki dalla semplice Wiki, quest'ultima infatti non prevede che si possano aggiungere strutture dati complesse ma solamente informazioni.

Gli oggetti creati tramite lo Skin Extension possono essere di due tipi:

- JavaScript extension;
- StyleSheet extension;

Entrambi sono creati attraverso il browser, esattamente navigando nella pagina che vogliamo modificare ed editandola tramite il wiki editor. Ora tra le opzioni del menù Edit, selezioniamo Objects che ci permette di aggiungere un nuovo oggetto alla pagina e quindi possiamo scegliere il tipo di oggetto tra quelli elencati in precedenza.

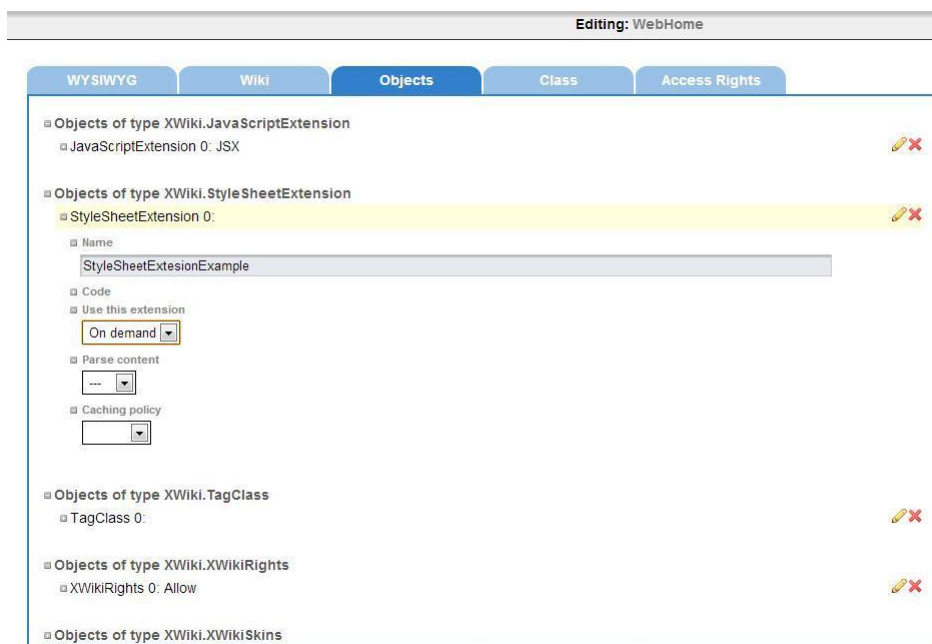


Figura 4.1: Editing dell'XWiki Objects.

Una volta che l'oggetto è stato creato ed risulta disponibile, si può cominciare ad implementare le effettive modifiche alla pagina digitando semplicemente il codice. Oltre al codice, vi sono campi che permettono la scelta tra diverse funzionalità utili per lo skin; tra queste vi è un campo che fornisce una

funzionalità interessante in quanto permette di definire quale sarà l'effettivo uso dell'estension: se "On demand" o "Always on this page" piuttosto che "Always on this wiki". Queste tre modalità rendono disponibile l'estension in particolari occasioni che dovranno essere specificate tramite alcune linee di codice Velocity, nel caso della clausola "On demand", oppure, di utilizzare l'estension solamente dopo aver accettato un warning che viene generato se si sceglie l'opzione "Always on this page", piuttosto che applicare l'estension su tutto il wiki scegliendo "Always on this wiki".

Creando un oggetto Skin Extension e settando ad esso l'opzione "On demand" ci si troverebbe ad avere una buona soluzione per creare uno skin per dispositivi mobile, infatti si potrebbe inserire uno script Velocity come controllo per azionare lo skin in modo che riconosca che la richiesta provenisse da dispositivo mobile; una volta riconosciuto il device, l'estension richiamerà l'oggetto appositamente creato in precedenza. In questo modo si ha la possibilità di creare più oggetti tali da rispondere alle richieste di dispositivi mobili diversi per dimensioni dello schermo o per altre caratteristiche.

Buona parte di queste azioni sono svolte in automatico da una particolare estensione che è disponibile gratuitamente sul sito dell'organizzazione XWiki, questa estensione che andrò ad analizzare è: XWiki Simple Mobile Skin. L'XWiki Simple Mobile Skin è uno skin per dispositivi mobile come smartphone e tablet. Il principio cardine su cui si basa questa estensione è quello di mostrare in view mode il contenuto della pagina usando la maggior parte dello spazio dello schermo disponibile per le informazioni contenute nella pagina. Come impostazione predefinita i pannelli e i tabs dei commenti sono nascosti, ma comunque accessibili usando links e dei drop down menù che permettono di mostrare tutte quelle funzionalità che nella pagina sono accessibili tramite un click.

Il Simple Skin Mobile è un prototipo, che viene però aggiornato frequentemente; in particolare grazie alle ultime release fornisce funzionalità come:

- l'edit wiki,
- algoritmi di login,

- riconoscimento di request di accesso al portale XWiki da parte di dispositivi quali iPhone, iPad, Android e IEMobile,
- edit tramite editor WYSIWYG, anche se in realtà questa funzione presenta alcuni bug che verranno corretti nel prossimo aggiornamento.

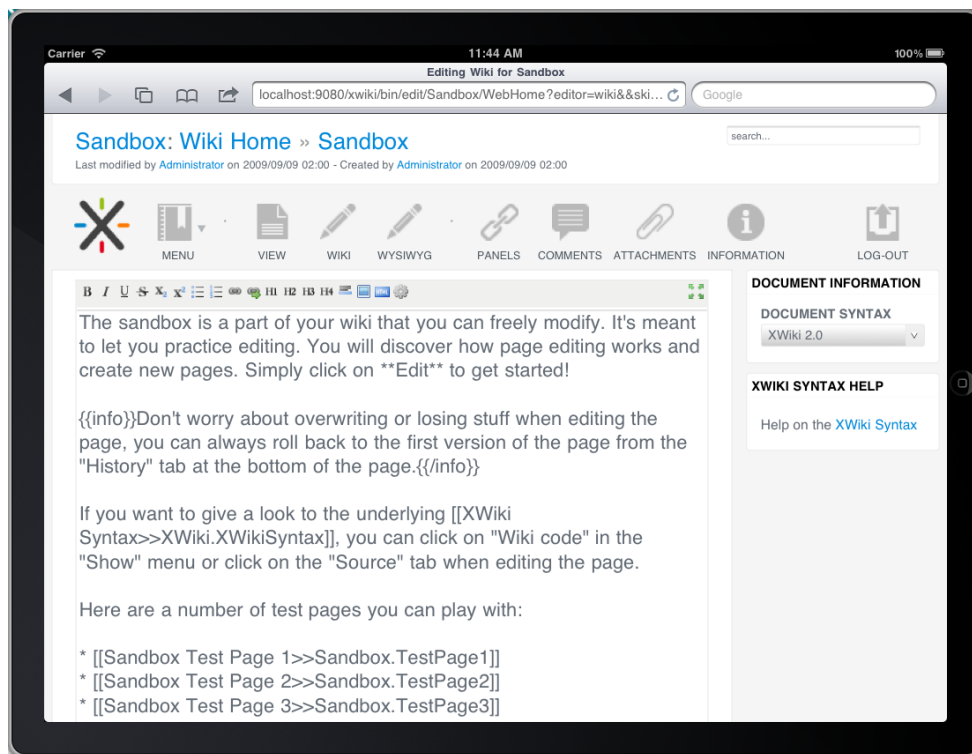


Figura 4.2: Edit dell'XWiki Simple Mobile Skin.

Per installare l'XWiki Simple Mobile Skin si può importare nella sezione dedicata all'Administrator il package, se questo è nella versione XAR, se invece lo si scarica come file ZIP lo si deve scompattare nella cartella skins/ dove il codice del portale XWiki è memorizzato. Il risultato è il medesimo: verrà creata una cartella contenente tutti i template, file CSS ed immagini necessarie a creare la skin mobile. Per renderla funzionante, si dovrà settare come skin nella sezione apposita del menù dedicato all'Administrator la XWiki.MobileRedirectorSkin, fornita nel package della XWiki Simple Mobile Skin, questa, contiene il codice per riconoscere la request di un device mobile ed effettuare il redirect alla skin mobile. Nel caso di Skin customizzata, il codice deve essere esportato dalla XWiki.MobileRedirectorSkin ed inserito

direttamente nei template, precisamente in `view.vm`, presente nella cartella `templates/` dove il portale XWiki è memorizzato.

In ultima analisi, credo sia importante spendere qualche parola sul linguaggio Velocity, che verrà largamente usato nel caso studio. In particolare Velocity utilizza variabili e dichiarazioni che fanno riferimento a contenuti dinamici nel sito. Una variabile è un type di riferimento che può descrivere un oggetto definito in codice Java o pu prendere il proprio valore da una dichiarazione Velocity. Tutte le dichiarazione Velocity iniziano con il carattere `"#"` e contengono una direttiva. Quando un visitatore online richiede la pagina web, il Velocity Template Engine ricercherà nella pagina web tutti i caratteri `"#"`, tra tutti quelli trovati determinerà quali sono un mark per dichiarazioni VLT e quali non centrano niente con Velocity. La variabile in Velocity inizia con il carattere `"$"`; mentre le stringe sono racchiuse in virgolette che possono essere sia singole che doppie. Le virgolette singole assegneranno alla variabile a cui è associata la stringa il valore così comè; mentre le doppie permettono di usare i riferimenti Velocity e direttive per interpolare e sostituire la variabile contenuta nella stringa con il reale contenuto della variabile. Una volta che si è assegnato un valore ad una variabile, questa può essere utilizzata dovunque all'interno di un file HTML.

I metodi in Velocity sono definiti con codice Java e sono riferiti alla signature che inizia con il carattere `"$"` seguito da un identificatore VTL seguito a sua volta da un VTL Method Body. Il Method Body consiste in un identificatore VTL seguito da parentesi che al loro interno hanno paramentri da passare al metodo se richiesto. L'effettivo Method Body può essere implementato all'interno di un template o più frequentemente viene importato in package che contengono le definizioni di tutta una serie di metodi.

Per quanto riguarda gli operatori relazionali logici, le direttive condizionali e gli strumenti che permettono i loop il linguaggio Velocity è molto simile a Java, con la differenza che prima di ogni direttiva bisogna inserire il carattere `"#"`. Le ultime due operazioni che analizzo, in quanto molto importanti nel caso in questione, sono le Velocity Macro e lo script Include. Lo script Include utilizza una signature che come al solito è anticipata dal carattere

”#”, ma la particolarità sta in realtà in quello che prevede dopo; infatti la signature finisce con due parentesi tonde dove al loro interno si prevede il riferimento tra virgolette ad un template Velocity. Lo script Include ha la funzione di includere un template Velocity all’interno del codice, nell’esatto punto in cui lo script viene eseguito, ciò è fondamentale perché crea ordine nel codice e permette di cambiare il flusso dell’esecuzione oltre a rendere possibile il riutilizzo di un template. Le Velocity Macro invece danno la possibilità di definire un segmento di codice che potrà essere ripetuto all’interno di un template; questa direttiva è molto utile sia all’interno dei template Velocity, perché permette al programmatore di risparmiare tempo ed avere una migliore tabulazione del codice, anche nella modalità edit del portale XWiki accessibile tramite browser, questo perché all’interno di un template si possono implementare tutte le macro che si intendono utilizzare nel portale e rendere questo file locale, poi includere in tutte le pagine del portale che si vuole editare risparmiando tempo e riducendo la possibilità di commettere errori.

Capitolo 5

Caso di studio: Skin per APICe

*I nostri successi e i nostri fallimenti
sono tra loro inscindibili,
proprio come la materia e l'energia.
Se vengono separati, l'uomo muore.*
- Nikola Tesla -

Nel seguente capitolo si tratteranno le tematiche riguardanti la progettazione di una skin adatta a dispositivi mobili per il sito di APICe, partendo dal plugin Simple Mobile Skin fornita da xwiki.org e sviluppata da Ludovic Dubost ed Ecaterina Moraru. In particolare, le modifiche che apporteremo riguarderanno i quicklinks panel, header e footer, che sono già presenti nella skin. Tutte le modifiche verranno realizzate andando a modificare template Velocity e qualche tag HTML in modo da rendere questa estensione il più possibile adatta per dispositivi mobile.

1 Analisi del portale APICe

La prima analisi da effettuare è quella del portale XWiki di APICe, raggiungibile al sito <http://apice.unibo.it>, dove andremo ad analizzare in che modo vengono presentati i contenuti, considerando quali sono quelli più utili per i dispositivi mobili, come con questi si può interagire e come vengono mostrati, in termini di strutture utilizzate. Un altro aspetto che dobbiamo osservare è come avviene la navigazione nel sito per poter poi scegliere in che agire

nella versione mobile senza ottenere un gap eccessivo tra le due versioni di navigazione.

Analizzando il sito si può notare la presenza di una sidebar la quale raggruppa una serie di panels tra cui: personal spaces, product spaces, incoming events e running projects, ovvero tutto ciò che riguarda spazi personali dei componenti del laboratorio, spazi contenenti le informazioni inerenti ai prodotti sviluppati in APICe, gli eventi a cui APICe parteciperà ed i progetti in corso.

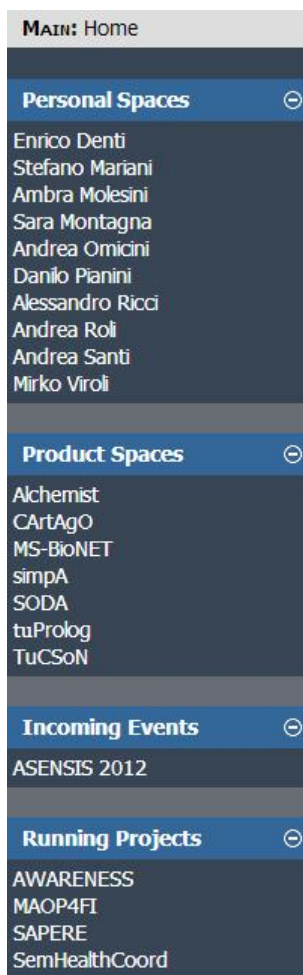


Figura 5.1: Side bar del portale APICe.

Un'altro elemento importante per la navigazione è la navigation bar posta sotto il logo di APICe: questo menù, raccoglie tutti gli spaces in cui il por-

tale è diviso, che quindi può essere considerato quindi un macro space. In particolare:

- Home: è il riferimento principale alla pagina iniziale del portale;
- People: contiene tutti gli spaces personali di professori, ricercatori, studenti e tecnici che collaborano in APICe;
- Publications: indicizza tutto ciò che viene pubblicato dai componenti del laboratorio, tutti i lavori che non sono ancora stati pubblicati ed i volumi curati;
- Talks: racchiude tutti i papers che vengono utilizzati in presentazioni, discorsi, seminari, nel contesto di lezioni, incontri o conferenze.
- Products: descrive tutti i prodotti che il laboratorio progetta e liberamente condivide;
- Projects: cataloga tutti i progetti di ricerca e industriali a cui APICe partecipa;
- Courses: descrive tutti i corsi che i membri di APICe svolgono;
- Theses: contiene tutte le tesi di ricerca svolte da studenti di Ingegneria dell'Alma Mater supervisionate dai membri di APICe;
- Events: informa sugli eventi a cui parteciperanno i membri del laboratorio.

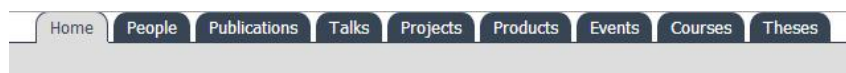


Figura 5.2: Navigation bar del portale APICe.

Un buon modo per identificare quali siano le informazioni che vogliamo riportare nella skin mobile è ricercare gli obbiettivi per cui un utente dovrebbe utilizzare il portale APICe tramite dispositivi mobili; ho quindi trovato le seguenti possibilità:

- cercare informazioni dei componenti del laboratorio di ricerca, che nella versione desktop risiedono negli spaces personali;
- informarsi su quali sono i progetti e prodotti su cui il laboratorio lavora;
- disporre delle pubblicazioni contenute nel portale;

- avere la possibilità di editare una pagina XWiki del portale in qualsiasi momento ed ovunque tramite l'utilizzo di un solo dispositivo mobile.

Considerando quindi questi obbiettivi organizzerò il layout in modo che abbia una navigation bar orizzontale posta nella parte superiore che abbia riferimenti agli spazi personali, allo spazio dei progetti e dei prodotti ed infine allo spazio delle pubblicazioni. Un altro elemento indispensabile è un'ulteriore navigation bar; questa volta, posta immediatamente prima del footer, che contenga pulsanti che permettano di mostrare i commenti alle pagine, gli attachments e diano la possibilità di editare la pagine tramite browser.

1.1 Analisi del codice che genera la skin del portale APICe.

Approfondisco ora in che modo i contenuti del portale vengono mostrati, esaminando in dettaglio quali template vengono richiamati e che azioni compiono. I template che devono essere eseguiti vengono compilati dal Template Engine, che si incarica di recuperare i dati selezionati dal template nel database del server, e restituisce in output il documento completo, nel nostro caso la pagina web del portale.

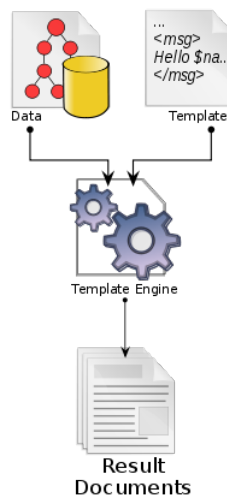


Figura 5.3: Template Engine di XWiki.

All'interno di `view.vm` vengono richiamati i template che caricano la struttura della skin ed i contenuti delle pagine. L'ordine con cui vengono richiamati è il seguente:

- `code.vm` fornisce la visualizzazione dei contenuti in modalità RAW, ovvero una versione scarna, nel caso ci sia una richiesta apposita;
- `startpage.vm` setta la visualizzazione dei contenuti in modalità VIEWER, ovvero quella che normalmente si vede se si accede al portale, e richiama i seguenti template:
 - `xwikivars.vm`:
 - * `frequentlyUsedDocs.vm`;
 - * `deprecatedVars.vm`.
 - `layoutvars.vm`;
 - `htmlreader.vm`:
 - * `stylesheets.vm`;
 - * `analytics.vm`;
 - * `javascripts.vm`.
 - `header.vm`:
 - * `global.vm`;
 - * `space.vm`.
 - `menuview.vm`;
- `contentview.vm`:
 - `titlevars.vm`.
- `docextra.vm`;
- `endpage.vm`:
 - `leftpanels.vm`;
 - `rightpanels.vm`;
 - `footer.vm`;
 - `htmlfooter.vm`.

Analizzando quindi questo schema gerarchico, da una prima analisi, si nota che il template che caricherà la skin mobile dovrà essere implementato come primo controllo nel template `view.vm`.

2 Meccanismo di autoloading e riferimenti alla versione mobile

Il codice che controlla la richiesta per accedere al portale, deve verificare che questa provenga da dispositivo mobile. Questo listato di codice viene

fornito direttamente dal Simple Mobile Skin plugin e io l'ho importato nel template `view.vm`, inoltre ho aggiunto un variabile, `$isskinmobile` che se settata a zero identifica che la skin mobile non è ancora stata caricata, mentre se ha valore pari a uno la skin è nella versione adattata. Questo meccanismo di flag ci permette di poter impostare all'interno delle pagine del portale XWiki quali contenuti mostrare nella versione normale e quali nella versione desktop, a condizione che il flag sia utilizzato nel template gerarchicamente di livello inferiore tra quelli che definiscono la pagina che vogliamo editare.

```
#set($mobile = $request.getCookie("xwikimobile").
    getValue())

## initialized the variable $isskinmobile that i
    have introduced
#set($isskinmobile = 0)

#if("$mobile"!="off")
    #set($ua = $request.getHeader("User-agent"))
    #if($ua.contains("iPhone")||$ua.contains("iPad")||
        $ua.contains("Android")||$ua.contains("IEMobile
        "))
        #if("$!request.skin"=="")
            $response.sendRedirect($doc.getUrl("view", "$!
                request.getQueryString()&skin=mobile"))

            ##set variablie that i have introduced when
                skin mobile is setted
            #set($isskinmobile = 1)
        #end
    #end
#end
#end
```

Listing 5.1: `autoload.vm`.

Ora sono andato a modificare anche la skin del portale XWiki nella versione desktop per aggiungere un riferimento nel footer che mi permettesse

di raggiungere la skin mobile; ho quindi aggiunto all'interno del template `footer.vm` il codice:

```
## link to the mobile version in the footer
tab
<span id="mobile"><span class="title">Go to:</
span> <a href="http://137.204.107.27/xwiki/
bin/view/Main/?skin=mobile">Mobile Version
</a>
</span>
```

Listing 5.2: codice che crea il riferimento al mobile version in `footer.vm`.

In seguito sono andato poi a modificare lo stile della skin; mi sono posizionato nella sezione Administrator del portale, poi nella sezione Presentation e successivamente Skin, alla fine ho cliccato il comando Customize e sono andato effettivamente a editare il wiki, in particolare, l'oggetto Default XWiki Skin, dove ho aggiunto nella barra del footer le specifiche CSS per visualizzare correttamente il link.



Figura 5.4: Link a mobile version.

3 Header

Lo header dovrà essere progettato in modo che contenga gli elementi utili alla navigazione, come raccomandato dal Responsive Web Design; infatti, conterrà un riferimento principale alla home, il puntatore alla pagina dove il visitatore si trova ed una navigation bar che abbia il riferimento alle sezioni importanti del portale. Buona parte di questi elementi sono forniti dal plugin Simple Mobile Skin che sfruttiamo, però presenta alcune imprecisioni alle mie intenzioni di progetto e che quindi andrò a modificare. Dividerò i pulsanti nel menu fornito dal plugin in due navigation bar, mettendone una visibile nell'header contenente solamente i riferimenti agli spazi People, Products,

Projects e Publications; l'altra nel footer.

I riferimenti a questi spazi mostreranno i panels principali delle pagine, in modo da avere una visione ottimale ed ordinata dei contenuti e offrendo la possibilità di vedere senza problemi il contenuto cercato. Sfrutterò una macro creata appositamente che imposta e visualizza un panel del portale come un elenco di link, questa macro è `#macro iosdisplaypanel()`. Di seguito il codice di questo elemento per creare il pulsante People, che sarà il medesimo per i pulsanti Product&Project e Publication:

```
##-----
## the People button
##-----
#macro(iosdisplaypanel $name)
  #set ($pobj = "")
  ## Since Velocity does not do any assignment if
  ## the RHS is null we need to reset the paneldoc
  ## as otherwise the previously set panel will be
  ## displayed. getDocument() returns null if the
  ## user doesn't have the right to view a document.
  #set ($paneldoc = "")
  #set ($paneldoc = $xwiki.getDocument($name))
  #if ($paneldoc != "")
    #set ($pobj = $paneldoc.getObject("Panels.
      PanelClass"))
    #if (!$pobj)
      ## discarded
    #else
      $!doc.display("content", "view", $pobj)
    #end
  #end
#end
#iosdisplaypanel("Panels.PersonalSpaceMenu")
```

Listing 5.3: codice che crea il riferimento pulsante People in `peoplepanel.vm`.

Dalla Simple Skin Mobile eredito il breadcrumbs e il form di ricerca; mentre ho deciso di aggiungere due pulsanti che permettono di scegliere in quale lingua devono essere tradotti i contenuti delle pagine del portale. Rimuovo dalla Simple Skin Mobile il link alle ultime modifiche, in quanto lo ritengo poco utile e dannoso in termini di layout. Non visualizzo inoltre tutti gli altri riferimenti, tranne edit wiki, attachments e comments che andranno nella successiva navigation bar, poiché non forniscono funzionalità utili alla navigazione da dispositivo mobile, come information. Di seguito riporto il codice finale di `header.vm`:

```
## make sure all links stay in this skin
$context.setLinksQueryString("skin=${xwiki.skin}")

#set($parents = $util.arrayList)
#set($discard = $parents.add($doc.fullName))

#macro(breadcrumb $doc $string $level)
  #set($parent = $doc.parent)
  #if(($parent != "") && ($level < 6) && (!$parents.
    contains($parent)))
    #set($discard = $parents.add($parent))
    #set($pdoc = $xwiki.getDocument($parent).
      getTranslatedDocument())
    #set($pdocurl = $pdoc.getURL("view"))
    #set($nstring = "<a href='$pdocurl'>$xwiki.
      getXMLEncoded(${pdoc.getRenderedTitle('plain
        /1.0'})})</a> <span class='separator'>&raquo;
      ;</span> $string")
    #set($level = $level + 1)
    #breadcrumb($pdoc $nstring $level)
  #else
    $string
  #end
#end
```

```

#if($context.getMode()==0) ## Visible only in a page
<div id="headerspace" class="layoutsection">
  <div id="navigation" class="ios">
    <div class="ios" id="breadcrumbs">
      <span class="breadcrumbspace"><a href="
        $spaceViewUrl">$xwiki.getXMLEncoded(${doc.
          space})</a>: </span>
#breadcrumb($doc "<a href='${doc.getURL('view')}'>
  $xwiki.getXMLEncoded($tdoc.getRenderedTitle('
    plain/1.0')) </a>" 0)
    </div>
    <div id="search" class="ios">
      <form action="$xwiki.getURL('Main.WebSearch')
        ">
        <input type="hidden" name="skin" value="${
          xwiki.skin}" />
        <label class="hidden" for="
          headerglobalsearchinput">$msg.get('
            panels.search.inputLabel')</label><
          input class="globalsearchinput withTip"
            id="headerglobalsearchinput" type="
              text" name="text" value="$msg.get('
                panels.search.inputText')" size="10"/>
        </form>
      </div>
    </div>
  </div>
</div>

<div id="clearfloats"><h1> <br> </h1></div>

  <div id="titleheader">

    <a class="logo" href="$!xwiki.getURL('Main.
      WebHome')" title="Home" rel="home"></a>

```

```
<div id="log" class="ios">
  #if (!$xcontext.action.startsWith('login'))
    #set ($fullurl = $xwiki.getRequestURL())
    ##$fullurl
    ##$xwiki.getURL("")
    #set($start = "http://137.204.107.27")
    #set( $testurl = $start+$doc.getURL("edit",
      "editor=wiki&$!languageparams"))
    ##$testurl
  #if ($isGuest)
    #set ($loginurl = $xwiki.getURL('XWiki.
      XWikiLogin', 'login', "xredirect=
      $escapetool.url($fullurl)"))
    <a class="action login" href="$!loginurl">$
      !msg.get('login')</a>
  #else
    #if ($xwiki.getRequestURL() == $testurl)
      #set ($fullurl = $start+$xwiki.getURL
        (""))
    #end
    #set ($logouturl = $xwiki.getURL('XWiki.
      XWikiLogout', 'logout', "xredirect=
      $escapetool.url($fullurl)"))
    <a class="action logout" href="$!logouturl
      ">$!msg.get('logout')</a>
  #end
#end
</div>
</span>
<div id="languages" class="ios">
  #if($xwiki.isMultiLingual())
    <span class="glink" id="headerlanguages">
```

```
#set($defaultLanguage = "${doc.
    getDefaultLanguage()}")
#if($defaultLanguage == '')
    #set($defaultLanguage = "${doc.
        getLanguage()}")
#end
#if($defaultLanguage == '')
    #set($defaultLanguage = "default")
#end
#if($defaultLanguage != "default")
    #if ($defaultLanguage == "en")
        <a href="${doc.getUrl("view
            ", "language=
            $defaultLanguage")" class
            ="language-default-en#if(
            $tdoc.realLanguage ==
            $defaultLanguage || (
            $defaultLanguage == '
            default' && $tdoc.
            realLanguage == ''))-
            language-current#end">en
        </a>
    #elseif ($defaultLanguage == "it")
        <a href="${doc.getUrl("view
            ", "language=
            $defaultLanguage")" class
            ="language-default-it#if(
            $tdoc.realLanguage ==
            $defaultLanguage || (
            $defaultLanguage == '
            default' && $tdoc.
            realLanguage == ''))-
            language-current#end">ita
        . </a>
```

```

        #end
    #end
    #set ($wikiSettingsLanguages = !$xwiki.
        getXWikiPreference('languages').trim().
        split('\s*[ ,| ]\s*'))
    #set ($hasLanguagesSet =
        $wikiSettingsLanguages.size() > 1 || "$!
        wikiSettingsLanguages.get(0)" != '')
    #foreach($lang in $doc.translationList)
        #if(!$hasLanguagesSet ||
            $wikiSettingsLanguages.contains($lang))
            #if ($lang == "en")
                <a href="$!doc.getUrl("view
                    ", "language=$lang")"
                    class="language-
                    translation-en#if($tdoc.
                    realLanguage == $lang)-
                    language-current#end">en
                </a>
            #elseif ($lang == "it")
                <a href="$!doc.getUrl("view
                    ", "language=$lang")"
                    class="language-
                    translation-it#if($tdoc.
                    realLanguage == $lang)-
                    language-current#end">ita
                    . </a>
            #end
        #end
    #end
#end
</div>

```

```
</div>

<div class="ios" id="menu">

## Edit Menu
## Compute the default edit mode
#set($editaction = $doc.getDefaultEditMode())

## section that group the horizontal buttons
<ul id="horizontalbuttons">
  ##People's button
  <li> <a class="action other" href="$doc.
    getURL("view", "viewer=peoplepanel")">
    People</a></li>

  ## modified the panel's button in Product &
    Project's button
  <li> <a class="action panelslist" href="$doc.
    getURL("view", "viewer=panels")">Product
    & Project</a></li>

  ## introduced the Publication's button
  <li> <a class="action testpanel" href="$doc.
    getURL("view", "viewer=publicationpanel")
    ">Publication</a></li>

  ## Tolgo l'icona di edit in quanto non
    voglio fornire la possibila di editare il
    wiki dalla
  ## mobile version
  ## <a class="action edit" href="$doc.getURL(
    $editaction, $!languageparams)">$msg.get
    ('core.menu.edit.wysiwyg')</a>
  ## <span class="separator">&#183;</span>
```

```
</ul>

<div class="clearfloats"></div>
</div>
</div>
#end
```

Listing 5.4: header.vm.

Il risultato finale sarà il seguente:



Figura 5.5: Header nella Mobile Skin.

4 Content

I contenuti del portale vengono adattati alla versione mobile tramite l'utilizzo del metodo `.getRenderedContent()`, la cui principale utilità consiste nel trasformare i contenuti con sintassi non HTML-compatible in maniera

che possano essere visualizzati correttamente, nel caso invece di informazioni implementate con sintassi compatibile con l'HTML non si presentano grossi problemi. Di seguito il codice del template `contentview.vm`, contenuto nella cartella mobile tra le skin disponibili, che si incarica di presentare i contenuti per dispositivi mobili:

```
### Display document contents in "view" mode.

## Compute the content to display

## Honor the syntax if it's passed in the request.
#set($outputSyntax = $xwiki.
    getAvailableRenderersSyntax($request.outputSyntax,
    $request.outputSyntaxVersion))
#if ($outputSyntax)
    ## If the passed syntax is not an HTML-compatible
    ## syntax we need to HTML-escape it so that it can
    ## be
    ## displayed fine in HTML (since at the point this
    ## vm file is called we're already inside an HTML
    ## page with
    ## panels on the side, header, etc).
    #set($syntaxType = $outputSyntax.type.toString()
    )
    #if (($syntaxType == "xhtml") || ($syntaxType == "
    html"))
        #set ($renderedContent = $tdoc.
            getRenderedContent($outputSyntax))
    #else
        ## Make sure to print correctly the result when
        ## it's not HTML
        #set ($renderedContent = "<pre>$escapetool.html(
            $tdoc.getRenderedContent($outputSyntax))</pre
            >")
    #end
```

```
#else
  #set ($renderedContent = $tdoc.getRenderedContent
        ())
#end

## Display the rendered content

<div id="xwikicontent">

<div id="document-title"><h1> <br> </h1></div>

$renderedContent

</div>

## Display the footnotes if the document is in 1.0
  syntax and if there are any defined

#set($footnotes = "#putFootnotes()")
$xwiki.renderText($footnotes, $doc)
```

Listing 5.5: contentview.vm.

5 Footer

Nel footer sono state apportate modifiche rilevanti, ho infatti inserito una seconda navigation bar che riprendesse i tab inerenti ai commenti ed agli attachments delle pagine wiki del portale. Questo menù ha tre pulsanti: Comments, Attachments, Wiki; che rispettivamente mostrano i commenti, i file collegati alla pagina e la funzione edit dal browser della pagina. La funzionalità di edit può essere sfruttata solamente se si è loggati come utenti del portale, quindi nel caso non si sia effettuata la procedura di login, cliccando il tasto si viene rimandati alla pagina in cui si effettua l'autenticazione. Di seguito mostro il codice che implementa il menù nel template `footer.vm`:

```
## vertical buttons to handle comments, attachment
and wiki

<div id="verticalbuttons">
    ## Comment button
    <a class="action comment" href="$doc.getURL
        ("view", "viewer=comments")">$msg.get('
        docextra.comments ')</a>

    ## Attachment button
    <a class="action attach" href="$doc.getURL("
        view", "viewer=attachments")">$msg.get('
        docextra.attachments ')</a>

    ## Wiki's button that allow to modify the
        wiki's pages through the browser
    #if (!$xcontext.action.startsWith('login'))
        #if ($isGuest)
            #set ($starturl = "http
                ://137.204.107.27")
            #set ($endurl = $doc.getURL
                ("edit", "editor=wiki&$!
                languageparams"))
            #set ($fullurl = $starturl+
                $endurl)
            #set ($loginurl = $xwiki.
                getURL('XWiki.XWikiLogin
                ', 'login', "xredirect=
                $escapetool.url($fullurl)
                "))
            <a class="action login" href
                ="$!loginurl">$!msg.get('
                core.menu.edit.wiki ')</a>
```

```

        #else
            <a class="action syntax"
                href="$doc.getURL("edit",
                    "editor=wiki&$!
                    languageparams")">$msg.
                    get('core.menu.edit.wiki
                    ')</a>
        #end
    #end
</div>

```

Listing 5.6: Parte di `footer.vm` che comprende la navigatio bar.

Nel footer è rimasta l'indicazione della versione del portale XWiki, ed ho aggiunto un riferimento alla home del portale nella versione normale. Di seguito il codice:

```

<div id="xwikilicence">$!copyright</div>
    #set ($version = $xwiki.getWebPreference("version
        "))
    #if ("${!version}" == "")
        #set ($version = $xwiki.getXWikiPreference("
            version"))
    #end
    #if ("${!version}" != "")
        #set ($version = $xwiki.parseContent($version))
    #else
        #set ($version = ${!xwiki.version})
    #end
    <div id="xwikiplatformversion">$version</div>
    $context.unsetLinksQueryString()
    <div id="mobile"><a href="$doc.getURL("view")"
        onclick="createCookie('xwikimobile','off',0);">
        Normal Version</a></div>

```

Listing 5.7: Parte di `footer.vm` che comprende l'endpage.

Il risultato è il seguente:

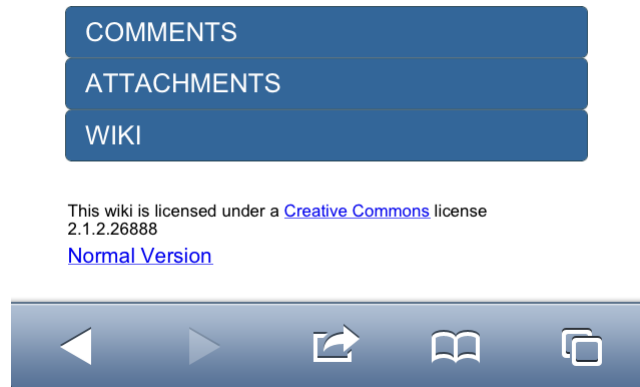


Figura 5.6: Footer nella Mobile Skin.

Conclusioni

È arrivato ora il momento di osservare se gli obiettivi che questa tesi si è posta inizialmente sono stati raggiunti e quali aspetti sono ancora da analizzare.

Per quanto riguarda la possibilità di adattare un portale XWiki ad essere visualizzato tramite dispositivi mobili posso affermare che questa opzione è implementabile, anche se presenta qualche problema di interazione con tutti i dispositivi e con qualche recente tecnica di scrittura delle pagine web. Questo, mi permette di dire che, considerando l'idea che gli sviluppatori di XWiki hanno sul loro progetto, ovvero renderlo uno strumento appetibile anche al mondo delle aziende, qualcosa ancora manca perché lo strumento XWiki sia al passo con la tecnologia vigente. Infatti se osserviamo la skin mobile che ho utilizzato nella mia tesi, è un progetto ancora embrionale che non fornisce ancora la possibilità di poter sfruttare tutti gli strumenti che attualmente sono disponibili per i web designer, oltre a non considerare una parte di dispositivi del mercato che non sono completamente portabili con la tecnologia attualmente sviluppata per questo portale. Se aggiungiamo a queste mancanze anche la rilevanza sempre maggiore nel mondo delle telecomunicazioni e l'aumento di market share del mondo mobile, la somma di queste problematiche dovrebbe portare gli sviluppatori a migliorare questa tecnologia rendendola sempre di più al passo con quelle nuove e con le tecniche di implementazione nel web.

Se analizziamo invece la rilevanza che può avere il portare siti web ad essere adatti alle richieste che provengono da dispositivi mobili, credo che questo sviluppo sia sempre più necessario, oltre per i già noti numeri sempre crescenti del mondo mobile, anche perché i dispositivi mobili potrebbero portare

un nuovo tipo di utente diverso da quelli già esistenti, che necessita di informazioni poste in un modo differente, che potrebbe essere anche disposto ad investire su queste tecnologie, fornendo quindi un nuovo tipo di consumatore e cliente all'interno di un mercato dinamico che può fornire enormi possibilità di business.

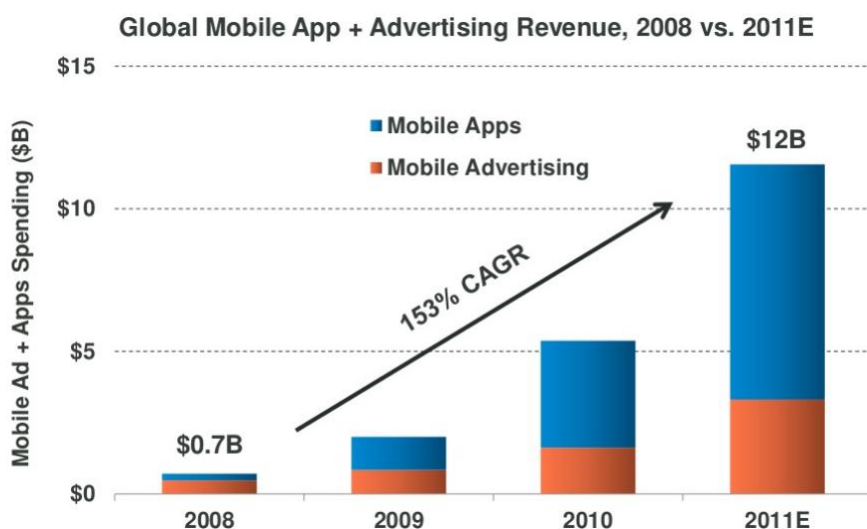


Figura 5.7: Aumento della pubblicità nei contenuti per dispositivi mobili.

Un modello attualmente preso in considerazione da molte aziende è il Bring Your Own Device, si tratta di concedere al dipendente l'utilizzo di dispositivo personali (telefono, smartphone, tablet) al posto di quelli tradizionalmente forniti dall'azienda, non solo per telefonare, ma anche per utilizzare la connettività ed accedere alle reti aziendali ed alle eventuali applicazioni messe a disposizione dei dipendenti. L'allargarsi di questo fenomeno avvalorava la mia idea che l'adattamento dei contenuti web per i dispositivi mobile è un passo che i web designer dovrebbero prendere in considerazione, per non perdere un numero in continuo aumento di consumers.

In ultima analisi considero quale architettura è preferibile utilizzare per adattare i contenuti web ai dispositivi mobile. La soluzione che ho mostrato nel caso di studio si basa su Single Page Application Architectures (SPAs), ovvero un architettura per siti web o web app che visualizza le informazioni su

un'unica pagina, fornendo all'utente una navigazione fluida. Questa architettura, se applicata a contenuti che vengono utilizzati tramite desktop, non è certo che dia benefici all'utente. Io credo che la SPAs sia invece fondamentale per la progettazione di web app o web site raggiungibili tramite dispositivi mobile, in quanto fornisce all'utente una navigazione fluida e ben organizzata, e al progettista un'architettura che disponga in maniera intelligente i contenuti tramite semplici strumenti.

1 Sitografia

- <http://www.xwiki.org>
- <http://extensions.xwiki.org/xwiki/bin/view/Extension/SimpleMobileSkin>
- <http://www.alistapart.com/articles/responsive-web-design/>
- <http://www.w3.org>

Ringraziamenti

Tra le pagine scritte questa é sicuramente la piú critica perché sará difficilissimo citare tutte le persone a cui devo almeno un semplice "Grazie".

Vorrei ringraziare il Professore Andrea Omicini per l'aiuto, il supporto, il tempo concessomi e la sua simpatia che mi ha accompagnato nel realizzare questa tesi insegnandomi il giusto modo di essere professionali senza prendersi troppo sul serio, l'Ing. Nazzareno Pompei perché senza il suo aiuto sarei ancora nei piú tetri meandri di XWiki ed il Dott. Stefano Mariani (mi scusi Ste, ma faccio ancora fatica a chiamarla Professore quindi si accontenti) per il supporto costante e per essere stato sempre disponibile.

Un ringraziamento lo devo a Giulia, la mia metà (ovviamente quella buona...), quella del "Come faresti senza di me?", ed in effetti a volte proprio non so come farei senza il suo eterno buon'umore e quel suo essere unica che colora le mie giornate e mi fa capire di essere fortunato ad aver trovato una persona speciale, che mi sprona e che mi sta accanto sempre. Ah dimenticavo...che sopporta anche i miei ritardi...a volte... :)

Un "Grazie" enorme spetta alla mia famiglia, la mia famiglia "allargata", i miei genitori che mi hanno dato la possibilità di essere qui in questo giorno particolare, che mi hanno insegnato quali sono i valori in cui é giusto credere e che mi hanno sempre permesso di seguire ogni mia aspirazione e anche di piú; a mia sorella Francesca che é per me il piú grande esempio di tenacia e abnegazione nell'inseguire i propri sogni e che insieme al suo compagno Marco mi hanno fatto il regalo migliore che potessi ricevere; ai miei zii che eternamente stanno dalla mia parte e che mi sono stati vicini quando ne ho

avuto bisogno senza che dovessi neanche bussare alla loro porta; a mio cugino Nicolás (mio fratello acquisito) che da sempre mi sostiene; ed infine ai miei nonni che sono la mia forza ed i primi ad incitarmi ad andare sempre avanti.

Per ultimi, ma non meno importanti, vorrei ringraziare tutti quegli amici veri che ho l'onore ed il privilegio di avere al mio fianco. Quelli con qui ho passato e continuerò a passare del tempo all'università: Zak (l'eterno compagno che più di una volta mi ha tolto dai problemi), Piero (chi prenderei in giro se non ci fossi? :)), Ste (anche se sei passato al lato oscuro della forza..rimani sempre un socio di risate infinite e colui che mi insegna sempre un modo diverso di vedere le cose), Rich, Campo (non sto neanche a citare una sua massima..). Un ringraziamento lo devo anche a tutti quegli amici che mi sopportano quando esco dall'università, che senza dire una parola capiscono di cosa ho bisogno, che non si lamentano mai se qualche volta non mi faccio sentire, che sono i protagonisti di mille storie vissute insieme e che non smettono mai di farmi sorridere: senza loro sarei senza dubbio una persona peggiore. Grazie a Edo, Corza, Franz, Fols, Versa, lo Zio Bizzo, Raffa, Nico, Carli, Sara, Gre, Marti, Die&Lillo&Davide (soci di aula studio) e tutto il resto della ciurma, per evitare di non dimenticare qualcuno.