

*Università degli Studi di Modena e  
Reggio Emilia*

---

Facoltà di Ingegneria – Sede di Modena

Corso di Laurea in Ingegneria Informatica – *Nuovo Ordinamento*

**Progetto e realizzazione di una intranet  
aziendale con applicazioni web AJAX**

Relatore:  
Prof. Sonia Bergamaschi

Candidato:  
Orazio Valerio

**Parole chiave:**

*intranet*  
*applicazioni\_web*  
*AJAX*

# Indice

<b>1</b>	<b>Introduzione</b>	<b>1</b>
<b>2</b>	<b>Intranet e le applicazioni web</b>	<b>3</b>
2.1	Intranet . . . . .	3
2.1.1	Nascita e sviluppo . . . . .	4
2.1.2	Intranet per l'azienda . . . . .	5
2.1.3	Vantaggi . . . . .	6
2.2	Applicazioni Web . . . . .	8
2.2.1	Storia . . . . .	8
2.2.2	Architettura . . . . .	14
2.2.3	Vantaggi . . . . .	17
2.2.4	Contesto di esecuzione . . . . .	17
2.2.5	Problematiche . . . . .	18
<b>3</b>	<b>AJAX</b>	<b>20</b>
3.1	Nascita e sviluppo . . . . .	20
3.2	Funzionamento di AJAX . . . . .	21
3.2.1	AJAX fa la differenza . . . . .	22
3.3	Descrizione tecnica . . . . .	25
3.4	Considerazioni lato utente . . . . .	26
3.4.1	Vantaggi . . . . .	26
3.4.2	Svantaggi . . . . .	27
3.5	Considerazioni lato server . . . . .	28
3.5.1	Vantaggi . . . . .	28
3.5.2	Svantaggi . . . . .	28
3.6	Sicurezza . . . . .	30
3.6.1	Lato client . . . . .	30
3.6.2	Lato server . . . . .	31

---

<b>4</b>	<b>La Intranet aziendale</b>	<b>32</b>
4.1	Il contesto aziendale . . . . .	32
4.2	Le richieste di CNA . . . . .	33
4.3	La Intranet CNA . . . . .	34
4.3.1	Specifiche di progetto . . . . .	34
4.3.2	Configurazione CMS . . . . .	35
4.4	Sezione Rubrica . . . . .	35
4.4.1	Descrizione . . . . .	35
4.4.2	Progetto Database . . . . .	36
4.4.3	Funzioni . . . . .	37
4.5	Sezione Circolari . . . . .	37
4.5.1	Descrizione . . . . .	37
4.5.2	Progetto database . . . . .	39
4.5.3	Funzioni . . . . .	39
4.6	Sezione Applicazioni Internet . . . . .	40
4.6.1	Descrizione . . . . .	40
4.6.2	Progetto database . . . . .	41
<b>5</b>	<b>Programma di gestione password</b>	<b>43</b>
5.1	Funzionamento . . . . .	43
5.2	Linguaggi utilizzati . . . . .	45
5.3	Interfaccia grafica in XUL . . . . .	45
5.4	Metodi Javascript . . . . .	46
5.4.1	Leggi Pagina Corrente . . . . .	46
5.4.2	Individua Login . . . . .	47
5.4.3	Effettua Login . . . . .	48
5.4.4	Memorizza Login . . . . .	49
5.4.5	Richiesta dati al database con AJAX . . . . .	51
5.4.6	Richiesta e risposta AJAX . . . . .	51
5.4.7	File Php . . . . .	52
<b>6</b>	<b>Conclusioni</b>	<b>54</b>

# Elenco delle figure

2.1	Architettura Three-Tier di una Applicazione Web. . . . .	15
3.1	Il modello tradizionale delle applicazioni Web (alla sinistra) confrontato col modello AJAX (alla destra). . . . .	23
3.2	Modello di interazione sincrona di un'applicazione web (sopra) confrontato con il modello asincrono Ajax (sotto). . . . .	24
3.3	Carico del server con e senza AJAX. . . . .	29
3.4	Chiamate "a vuoto". . . . .	29
4.1	Schema E/R sezione Rubrica. . . . .	36
4.2	Funzione ricerca contatto. . . . .	37
4.3	Funzione inserisci nuovo contatto. . . . .	38
4.4	Schema E/R sezione Circolari. . . . .	39
4.5	Funzione ricerca circolari. . . . .	40
4.6	Funzione inserisci nuova circolare. . . . .	41
4.7	Sezione Applicazioni Internet. . . . .	42
4.8	Schema E/R sezione Applicazioni. . . . .	42
5.1	Barra delle password. . . . .	43
5.2	Esempio di uso del programma. . . . .	44
5.3	Richiesta dati al database con AJAX. . . . .	51

# Capitolo 1

## Introduzione

Il presente lavoro di tesi è frutto dell'attività di tirocinio da me svolta presso l'azienda CNA Servizi Modena.

I capitoli 2 e 3 di questo documento sviluppano i concetti teorici su cui si è basato il lavoro svolto, presentato poi nei capitoli 4 e 5.

Viene ora data una breve descrizione sul contenuto dei capitoli che compongono il documento.

### **Capitolo 2 - Intranet e le applicazioni web**

In questo capitolo si dà una definizione di reti intranet e si descrivono gli aspetti che hanno caratterizzato loro evoluzione dai primi anni della loro nascita ad oggi. Si descrive come le intranet si siano sviluppate soprattutto in ambito aziendale grazie, non solo alla capacità di fornire supporto alle attività, ma anche alla visione di esse come luogo di incontro e strumento per favorire la condivisione della conoscenza aziendale. Nel capitolo vengono inoltre trattate le applicazioni web in quanto sono esse a realizzare le funzionalità offerte da una intranet. Oltre alla loro architettura sono presentati i vantaggi che questo tipo di applicazioni hanno rispetto alle classiche applicazioni client/server.

Si evidenzia infine come l'intranet venga considerata il contesto ideale per lo sviluppo di applicazioni web. Nel caso di una intranet aziendale, infatti, si ha la conoscenza sia dei profili di utenza che dei supporti di rete presenti, pertanto vengono meno le considerazioni relative alla compatibilità con i browser e alla velocità di accesso alle risorse.

### **Capitolo 3 - AJAX**

In questo capitolo si parla principalmente di AJAX, una nuova tecnica per lo sviluppo di applicazioni web interattive e che di recente ha suscitato notevole interesse da parte

degli sviluppatori del web. L'intento di tale tecnica è quello di ottenere pagine web che rispondono in maniera più rapida, grazie allo scambio in background di richieste HTTP e senza che ci sia la necessità di ricaricare la pagina web ogni volta che l'utente effettua una modifica.

Nel capitolo si descrive il funzionamento di AJAX e i modi con cui può essere utilizzata per migliorare l'interattività, la velocità e l'usabilità di una pagina web. Si evidenziano, infine, i vantaggi e gli svantaggi derivanti dall'uso di questa tecnica, ponendo particolare attenzione alle problematiche relative alla sicurezza che, in contesti come Internet, non possono essere trascurate.

## **Capitolo 4 - La Intranet aziendale**

In questo capitolo viene presentata l'attività svolta durante il tirocinio aziendale. La descrizione si basa sulle fasi seguite per il progetto e la realizzazione della "Intranet CNA" ed in particolare delle sezioni in essa contenute. Le sezioni della intranet sono state costruite essenzialmente come applicazioni web e rese più dinamiche grazie all'uso della tecnica AJAX. Da qui si nota come tutti i concetti teorici, che sono stati descritti nei precedenti capitoli, siano stati messi in pratica con l'attività di tirocinio.

## **Capitolo 5 - Programma di gestione password**

In questo capitolo si presenta il programma di gestione password realizzato per la "Intranet CNA". Uno degli obiettivi previsti dal progetto richiedeva la realizzazione di un sistema "single sign-on" che permettesse agli utenti di accedere, non solo alle applicazioni web intranet, ma anche a quelle situate su domini esterni, ovvero applicazioni web Internet. La soluzione alla funzionalità appena descritta è stata possibile grazie alla realizzazione di un programma di gestione password installato nel browser web. Questo programma, denominato "Intranet Password Manager", realizzato appositamente per essere utilizzato nel contesto della intranet, ha le medesime funzionalità dei classici programmi di gestione password incorporati ormai in tutti i browser web. La caratteristica che differenzia però "Intranet Password Manager" è che la memorizzazione delle credenziali di accesso non è fatta sulla specifica macchina, ma eseguita sul database della intranet. Questa caratteristica permette all'utente di avere accesso al proprio archivio password da qualsiasi postazione connessa alla Intranet CNA. Infine si espone, anche con codice Javascript, le funzioni implementate comprese quelle sviluppate con la tecnica AJAX.

# Capitolo 2

## Intranet e le applicazioni web

### 2.1 Intranet

Una *Intranet* è una rete privata, costruita ed usata all'interno di un'impresa, con lo scopo principale di condividere informazioni aziendali e risorse informatiche tra tutti i dipendenti. Generalmente le Intranet sono basate sugli stessi protocolli di comunicazione utilizzati per la rete Internet. Una Intranet mantiene le funzionalità tipiche di un portale web ma si indirizza ad una utenza ben definita, interna ad una specifica organizzazione.

Gli attori coinvolti in un sistema Intranet possono essere:

- l'azienda al suo interno;
- l'azienda e la sua rete commerciale;
- l'azienda e i suoi clienti e/o fornitori;
- l'azienda e le sue filiali, i magazzini periferici, le consociate etc;
- un'associazione di aziende o di categoria e le sue affiliate.

Una rete Intranet consente quindi di comunicare informazioni di ogni genere ad un ristretto numero di utenti, garantendo un accesso riservato alle stesse.

Le tecnologie utilizzate su Internet (protocolli di comunicazione TCP/IP) e sul WWW (protocollo di trasmissione HTTP e linguaggio HTML) sono oggi lo standard per la comunicazione fra sistemi eterogenei e per lo scambio di informazioni multimediali. I protocolli di comunicazione di Internet sono già da tempo entrati a far parte delle reti private delle grandi organizzazioni, ma per parlare di Intranet devono entrare a far parte del sistema



anche le tecnologie WWW, che possono garantire la diffusione di informazioni multimediali.

Attualmente la concezione più comune di Intranet prevede un *Corporate Portal* come punto di ingresso ad applicazioni specifiche, quali:

- **Publishing:** pubblicazione, personalizzazione e visualizzazione dei contenuti sull'Intranet, realizzata con la comunicazione monodirezionale di contenuti verso il personale;
- **Document management:** supporto all'acquisizione ed alla gestione della conoscenza esplicita, con funzioni di archiviazione, indicizzazione, correlazione e ricerca;
- **Community:** supporti alla comunicazione e all'interazione tra utenti attraverso servizi interattivi (forum, mailing list, instant messaging, chat etc), finalizzati alla gestione della conoscenza implicita all'interno dell'azienda;
- **Collaborative work:** supporto alla collaborazione e al teamworking (ad esempio groupware, e-room, videoconferenze etc);
- **Legacy integration:** supporto all'accesso ai sistemi informativi aziendali, ai dati e alle procedure dei sistemi gestionali e di tutti gli altri applicativi in azienda;
- **Self Service:** funzionalità in grado di erogare servizi interattivi ai dipendenti, come e-learning, rubrica del personale, modulistica, help desk informatico etc.

Le modalità di impiego, utilizzo e gestione di queste funzionalità possono essere estremamente diverse per ogni azienda. Tanto che sotto il termine "Intranet" ricadono applicazioni difformi per obiettivi, funzionalità, attività supportate, tecnologie usate e legami con gli altri sistemi aziendali.

### 2.1.1 Nascita e sviluppo

Per capire come sono nate le prime reti Intranet si può considerare lo sviluppo dei *Sistemi Informativi Aziendali* a partire dagli anni '50. Negli anni '50 e '60 nascono i primi mainframe e alla fine degli anni '70 il binomio sistemi informativi ed elaboratore diviene un punto fisso di tutte le grandi e medie imprese. A partire dalla metà degli anni '80 sono stati introdotti sistemi di supporto alle decisioni per supportare processi come le fasi di acquisto in azienda. Si giunge ai primi anni '90 e nei nuovi modelli organizzativi aziendali (dalle organizzazioni gerarchiche pure a quelle reticolari) nascono strumenti

per supportare il lavoro collaborativo, sistemi di gestione della documentazione aziendale fino al Knowledge Management (metà degli anni '90) che consente di gestire e rendere disponibili i “corpi delle conoscenze” aziendali. Parallelamente allo sviluppo dei sistemi informativi in azienda nascono a partire dalla metà degli anni '90 le Intranet, esse sono state introdotte inizialmente come strumenti di gestione della comunicazione interna e repository di informazioni e documentazione aziendale. Successivamente le Intranet si sono integrate con tool di gestione della conoscenza in azienda e si è quindi puntato su strumenti di classificazione e archiviazione dei documenti o di creazione di community.

Ad oggi l'aspetto di un'Intranet varia da azienda ad azienda, prendendo forme anche completamente diverse a seconda degli obiettivi e delle teorie di management su cui il sistema viene costruito. A partire dalla sua nascita questo nuovo paradigma telematico ha impattato sulla mentalità, trasmettendo un cambiamento nella cultura aziendale e mettendo in discussione il sistema di gerarchie e abitudini consolidate. Intranet rappresenta la concezione della comunicazione in azienda, della trasparenza dei processi burocratici, della condivisione della conoscenza, della riduzione del lavoro inutile. Inoltre rafforza la capacità di controllo del centro aziendale sull'intera attività dell'impresa: le Intranet divengono così uno dei pilastri della ristrutturazione organizzativa, sia per facilitare altri cambiamenti e sia per migliorare la velocità e la flessibilità dell'azienda stessa.

Negli ultimi anni l'adozione di sistemi Intranet, partita dalle grandi aziende americane, si è diffusa anche in quelle italiane. Nell'immediato futuro il valore di una Intranet sarà sempre più associato alla capacità di disporre di informazioni e processi collegati al contesto dell'utente.

### **2.1.2 Intranet per l'azienda**

Ciò che motiva un'azienda a percorrere un progetto Intranet sono i benefici che in termini di produttività ed efficienza si possono trarre dal suo uso e dal fatto che molti processi sono automatizzati diminuendo, in questo modo, il carico di spesa e lavoro soprattutto per quanto concerne l'area dei servizi (accedere alla modulistica, gestione risorse umane ecc.). L'Intranet viene percepita in larga misura come strumento adatto a creare e mantenere un'identità aziendale e una conoscenza condivisa, a seconda delle singole esigenze l'Intranet sarà progettata enfatizzando maggiormente l'aspetto operativo o quello di community.

In termini generali, una Intranet aziendale è utile per distribuire economicamente e facilmente l'informazione risolvendo nel modo più efficace il problema della gestione, della distribuzione e dell'accesso dei flussi di informazioni.

## **Comunicazione interna**

Risorsa strategica dell'azienda per antonomasia, la comunicazione interna può essere notevolmente migliorata grazie ad una Intranet in quanto offre soluzioni più funzionali ed economiche all'esigenza di gestire i flussi di comunicazioni interni, attraverso l'utilizzo delle risorse di Internet che consentono di inviare informazioni ovunque in tempo reale.

Tutte le società devono gestire un flusso di informazioni all'interno della loro struttura, ad esempio la gestione degli ordini in arrivo da controllare e inoltrare al settore di competenza. Dopo aver gestito l'ordine, un pezzo viene preso dal magazzino, imballato e preparato per la spedizione. Chi è coinvolto in questo processo necessita costantemente di una visione completa del progresso dell'ordine, attraverso report e grafici. Le informazioni di questo tipo sono generalmente gestite dalle aziende in vario modo, a partire da report scritti a mano fino a sofisticati sistemi di controllo.

Altro problema da analizzare è la distribuzione di informazioni di supporto sempre aggiornate alla rete di vendita: listini prezzi, cataloghi prodotti, brochure commerciali etc. La diffusione di materiale cartaceo non sempre è efficace e tempestiva, spesso il materiale distribuito non è aggiornato e produzione e distribuzione presentano entrambi tempi lunghi e costi elevati. Mantenere le informazioni su un WWW server centrale è il modo migliore di garantire un costante aggiornamento del materiale, consentendone l'immediata fruizione da parte degli interessati, e permettendo di risparmiare i costi di stampa e spedizione.

Una Intranet aziendale inoltre è un sistema interno all'azienda, e come tale off-line e riservato. Si può tuttavia sfruttare la risorsa Internet per creare dei collegamenti esterni che consentano operazioni di marketing, comunicazioni e campagne di vendita via web. Ordini, feedback dei clienti, percorsi di ordini e servizi di supporto al consumatore possono essere organizzati e gestiti attraverso Internet e poi girati automaticamente ai rispettivi interessati o ai relativi software di gestione, in questo modo, si crea un flusso circolare di comunicazione.

Per ulteriori approfondimenti vedi [1].

### **2.1.3 Vantaggi**

I vantaggi di utilizzare una Intranet risiedono principalmente nell'utilizzare tecnologie sviluppate per connettere sistemi eterogenei con costi ridotti, nella relativa facilità di realizzazione, e con tecniche standard.

### **Indipendenza dall'hardware e salvaguardia degli investimenti fatti**

Una rete Intranet permette di evitare di porsi problemi quali la scelta dell'hardware o del sistema operativo, in quanto qualunque computer è in grado di collegarsi e scambiare informazioni con gli standard Internet.

E' inoltre indipendente dal sistema di comunicazione: si può utilizzare qualunque tipologia di rete per interconnettere i sistemi, purché utilizzi il protocollo TCP/IP di Internet. In pratica, una rete Intranet permette di avere un dialogo omogeneo fra sistemi eterogenei, problema che ha spesso frenato gli investimenti in tecnologia "in attesa di uno standard".

### **Basso costo di implementazione**

I costi che un'azienda con una rete interna deve sostenere per procurarsi i sistemi dedicati alle funzioni di pubblicazione e divulgazione delle informazioni sono limitati all'acquisto di personal computer di media potenza: non occorrono sistemi grandi e costosi, e la maggior parte del software è disponibile gratuitamente o a prezzi simbolici sulla rete Internet per qualsiasi piattaforma.

Si possono così utilizzare immediatamente le applicazioni sviluppate per Internet, che comprendono la posta elettronica, la pubblicazione di informazioni multimediali, l'interrogazione di basi di dati remote, la videoconferenza, la trasmissione di messaggi vocali. Inoltre il collegamento fra i vari nodi della rete Intranet fisicamente distanti fra di loro avviene su normali linee telefoniche o isdn ma sempre al costo di una telefonata urbana.

### **Facilità d'uso dell'interfaccia utente**

L'accesso e la consultazione delle informazioni da parte degli utenti sono governate da interfacce e modalità di approccio indipendenti dal tipo di computer utilizzato. Tale standardizzazione applicativa porta a evidenti vantaggi in termini di investimenti e di formazione del personale.

### **Espandibilità e modularità del sistema e utilizzo dei servizi Internet**

Un sistema Intranet consente di collegarsi alla rete mondiale Internet in modo molto semplice. Basta infatti connettere un punto della rete Intranet ad un fornitore di accesso per trovarsi collegati con tutto il mondo e rendere disponibili a tutto il mondo le informazioni presenti sulla rete Intranet, eventualmente filtrate.

Infatti, ogni giorno il panorama Internet si arricchisce di servizi che possono migliorare

le applicazioni di Intranet (e-mail, videoconferenza desktop, agende di gruppo, fax server, audioconferenza etc.)

### **Sicurezza e Riservatezza dati**

Una diffusione così ampia delle informazioni aziendali amplifica il problema della sicurezza e della riservatezza dei dati. Tali problematiche sono risolte proteggendo la rete Intranet con sistemi Firewall che permettono di limitare l'accesso dall'esterno ai computer dedicati alla pubblicazione e sistemi di codifica di accesso e crittografia che impediscono l'accesso alle zone riservate da parte di persone non autorizzate.

## **2.2 Applicazioni Web**

Una **applicazione web** è una applicazione software per Internet, a cui gli utenti accedono tramite un browser Web. Dal punto di vista del browser, l'interazione con una applicazione web è indistinguibile dall'accesso a un sito Web statico. Le pagine visualizzate dal browser, in questo caso, sono però generate dinamicamente dall'applicazione.

### **2.2.1 Storia**

All'inizio, tutte le pagine web erano statiche, gli utenti richiedevano una risorsa e il server gliela restituiva. Non c'erano elementi mobili o che comparivano a intermittenza. Per un gran numero di siti web queste caratteristiche erano perfette: le pagine web non erano altro che copie elettroniche del testo che era stato raccolto e distribuito. Nei primi giorni di vita del browser, la natura statica delle pagine web non rappresentava un problema, gli scienziati utilizzavano Internet per scambiarsi documenti sulle ricerche e le università inoltravano on-line le informazioni sui vari corsi. Le imprese non avevano ancora trovato un modo in cui sfruttare questo nuovo canale. In effetti, inizialmente, le home page aziendali contenevano essenzialmente le informazioni sui contatti e una breve documentazione.

Nei successivi anni il personal computer divenne un componente indispensabile per le attività commerciali e cominciava a comparire un pò ovunque, dalle università alle case. Gli utenti a quel punto, vista la diffusione e l'importanza dell'utilizzo, cominciarono a desiderare un'esperienza più dinamica del Web.

### **CGI**

La prima soluzione in grado di rendere più dinamico il Web fu *Common Gateway Interface* (**CGI**). CGI non si limitava al semplice recupero Web statico ma consentiva di creare

programmi che venissero eseguiti quando un utente presentava una richiesta. Ad esempio, se si desiderava visualizzare sul proprio sito web gli articoli in vendita, con uno script CGI era possibile accedere al proprio database dei prodotti e visualizzare i risultati. Utilizzando semplici moduli HTML e script CGI si poteva creare un'area da cui vendere i prodotti a chiunque avesse avuto un browser. Gli script CGI potevano essere compilati in numerosi linguaggi, da Perl a Visual Basic, e ciò ne rese possibile l'uso in una vasta gamma di set di capacità.

Tuttavia, CGI non era l'approccio più sicuro per creare pagine Web dinamiche. Con CGI, si consentiva letteralmente alle persone di eseguire un programma sul proprio sistema. Nella maggior parte dei casi ciò avrebbe potuto non presentare problemi, ma un utente malintenzionato poteva sfruttare questo script per fare eseguire al sistema operazioni indesiderate. Nonostante questo rischio potenziale, CGI è utilizzato ancora oggi.

## Applet

Era ormai chiaro che CGI poteva essere migliorato. Nel maggio 1995, John Gage di Sun and Andreessen (ora operante presso il Netscape Communications Corporation) annunciò la nascita del linguaggio di programmazione chiamato **Java**. Netscape Navigator offrì il supporto per questo nuovo linguaggio che era originariamente usato per i decoder (Set Top Box). Come accade spesso quando si verifica qualcosa di realmente rivoluzionario, Java e Internet si trovarono nel posto giusto al momento giusto e pochi mesi dopo il rilascio di questo linguaggio sul Web, migliaia di persone avevano già scaricato Java. Navigator di Netscape che supportava Java permise l'apertura di nuovi orizzonti per le pagine Web dinamiche: era iniziata l'era delle applet.

Le applet consentivano agli sviluppatori di scrivere piccole applicazioni che potevano essere incorporate in una pagina Web. Se utilizzavano un browser in grado di riconoscere Java, gli utenti potevano eseguire le applet nella JVM (Java Virtual Machine) del browser. Sebbene le applet potessero eseguire numerose operazioni, alcune avevano delle limitazioni: solitamente non potevano leggere o scrivere sul file system, non potevano caricare librerie native e avviare programmi sul client. Oltre a queste restrizioni, le applet erano eseguite con un modello di sicurezza sandbox che contribuiva a proteggere gli utenti dal codice dannoso.

Per molte persone, le applet furono il primo impatto con il linguaggio di programmazione Java e, per i tempi, erano un ottimo metodo per creare applicazioni Web dinamiche. Le applet consentivano di creare un thick client all'interno del proprio browser,

entro i vincoli di sicurezza della piattaforma. In quel periodo le applet erano abbondantemente sfruttate in alcune aree, tuttavia non riuscirono a conquistare completamente la comunità Web.

Un problema era noto agli sviluppatori dei thick client: era indispensabile utilizzare la versione Java adeguata al client. Dato che le applet erano eseguite nella periferica virtuale di un browser, gli sviluppatori dovevano accertarsi che sul client fosse installata la versione di Java adeguata. Sebbene non fosse insormontabile, questo problema compromise in modo decisivo la tecnologia applet. La diffusione subì un freno anche per la presenza di applet con script carenti che causavano complicazioni sui client e rendevano i clienti esitanti circa l'impiego di soluzioni basate sulle applet.

## JavaScript

All'incirca nello stesso periodo, Netscape creò un linguaggio di script chiamato alla fine **JavaScript**. (Era stato chiamato Mocha mentre era un prototipo e quindi ribattezzato LiveWire e LiveScript prima di essere denominato JavaScript.) JavaScript era progettato come metodo per facilitare lo sviluppo delle applet agli sviluppatori e ai programmatori Web che non conoscevano Java. (Chiaramente Microsoft aveva la propria risposta a JavaScript: un linguaggio di script detto **VBScript**.) Netscape assunse Brendan Eich per progettare e implementare il nuovo linguaggio ed egli ideò un linguaggio di script ad inserimento dinamico che si rivelò davvero indispensabile. Sebbene fosse stato aspramente criticato per la carenza degli strumenti di sviluppo, di messaggi di errore utili e di debugger, JavaScript era un mezzo potente per creare applicazioni Web dinamiche.

Originariamente, JavaScript era stato creato per agevolare gli sviluppatori nella modifica dinamica dei tag delle loro pagine, con lo scopo di arricchire l'esperienza dell'utente. Era divenuto evidente che si poteva considerare la pagina come un oggetto e ciò aveva dato luogo al modello DOM (Document Object Model). Inizialmente JavaScript e DOM erano strettamente connessi tra loro ma, con il tempo, si evolsero in costrutti separati. DOM era una rappresentazione completamente orientata agli oggetti della pagina che poteva essere modificata con un linguaggio script come JavaScript o VBScript.

Alla fine, il World Wide Web Consortium (W3C) utilizzò e sfruttò come standard DOM, mentre l'ECMA (European Computer Manufacturers Association) ratificò JavaScript come la specifica ECMAScript. Tutte le pagine e gli script compilati secondo questi standard devono avere lo stesso aspetto e lo stesso funzionamento in qualsiasi browser, conformi alle citate linee guida.

Nei primi anni di diffusione erano diversi i fattori avversi a JavaScript. Il supporto dei

browser era incostante (anche oggi lo stesso script potrebbe avere un funzionamento diverso a seconda del browser utilizzato) e i clienti potevano liberamente disattivare JavaScript (alcuni strumenti di violazione della sicurezza ben pubblicizzati avevano incoraggiato gli utenti a farlo). La difficoltà legata allo sviluppo di JavaScript provocò la riluttanza di numerosi sviluppatori a utilizzare spesso questo linguaggio, mentre altri semplicemente lo ignoravano, considerandolo un linguaggio-giocattolo per progettisti grafici. La maggior parte di loro si accontentava di creare semplici applicazioni basate sui moduli dopo aver fatto uno sforzo mentale considerevole nel tentativo di utilizzare, verificare e risolvere i complessi problemi di JavaScript.

### **Servlet, ASP e PHP**

Nell'era della connessione remota per l'operazione di download dell'intera base di codici per un'applet complessa poteva essere necessario più tempo di quanto gli utenti fossero disposti ad aspettare. Gli sviluppatori dovevano anche preoccuparsi della versione Java presente sul client e alcune periferiche virtuali erano abbastanza deludenti. Nella situazione ideale si presentavano solo pagine Web statiche, dopo tutto era questo lo scopo per cui era stato progettato Internet. Chiaramente le pagine statiche erano, appunto, statiche ma se fosse stato possibile generare dinamicamente il contenuto sul server e restituire poi del contenuto statico si sarebbe potuti arrivare lontano.

Entro un anno da quando Java era stato diffuso, Sun lanciò i servlet. Il codice Java non veniva più eseguito nei browser dei client come nel caso delle applet, bensì veniva eseguito in un server di applicazioni controllato dall'utente. Ciò consentiva agli sviluppatori di uniformare le applicazioni business esistenti e, in caso fosse stato necessario passare all'ultima versione Java, sarebbe stato sufficiente preoccuparsi solo del proprio server. La natura di tipo "scritto una volta ed eseguibile ovunque" di Java permetteva agli sviluppatori di selezionare i server di applicazioni e gli ambienti server migliori: questo era certamente un altro vantaggio della nuova tecnologia. I servlet erano anche un'alternativa agli script CGI.

I servlet rappresentavano quindi un decisivo passo in avanti. Fornivano l'accesso completo all'intero set di interfacce API (Application Programming Interface) di Java ed erano caratterizzati da un'intera libreria per la gestione http. Tuttavia i servlet non erano perfetti. La progettazione dell'interfaccia con i servlet poteva essere davvero difficoltosa. In un'interazione tipica dei server, si ottenevano alcune informazioni dall'utente, si eseguivano alcune operazioni di logica aziendale e poi, utilizzando ciò che costituiva le righe di stampa, si creava l'HTML che poteva essere visualizzato dall'utente.



Oltre ad essere inclini a causare errori e difficili da visualizzare, i servlet creavano una certa confusione sulle abilità necessarie per il loro impiego. In generale, le persone che scrivevano il codice su lato server erano sviluppatori software ben preparati nel settore degli algoritmi e compilatori, non i progettisti grafici che ideavano gli eleganti siti Web aziendali. Gli sviluppatori aziendali non si concentravano unicamente sulla scrittura della logica aziendale ma dovevano anche occuparsi della creazione di progetti coerenti. Era necessaria una separazione della logica per la presentazione da quella puramente aziendale. Ciò che serviva erano le pagine JSP (JavaServer Page).

Le pagine JSP erano, in un certo senso, la risposta alle pagine ASP (Active Server Pages) di Microsoft. Microsoft imparò dagli errori di Sun fatti con la specifica Servlet e creò le ASP per semplificare la creazione di pagine dinamiche. Microsoft aggiunse anche il supporto di uno strumento eccellente e la stretta integrazione con il proprio server Web. Le pagine JSP e ASP sono simili perché entrambe sono state progettate per separare l'elaborazione aziendale dal layout di presentazione. Esistono delle differenze tecniche (anche Sun ha imparato da Microsoft) ma entrambi i tipi di pagina consentono ai progettisti Web di concentrarsi sul layout mentre gli sviluppatori software possono dedicarsi completamente alla logica aziendale.

Chiaramente, Microsoft e Sun non vantano il monopolio delle soluzioni sul lato server. Esistono molte altre opzioni, da PHP a ColdFusion. Alcuni sviluppatori sono attratti da strumenti sorprendenti, altri cercano linguaggi più semplici. In conclusione si può però affermare che tutte queste soluzioni eseguono la stessa attività: generano dinamicamente il linguaggio HTML. Creando il contenuto sul server si risolve il problema della distribuzione; tuttavia l'esperienza vissuta dall'utente usando il puro linguaggio HTML è meno soddisfacente di quella fornita da un thick client o dalle applet. Nelle sezioni successive verranno illustrate molte altre soluzioni create nel tentativo di offrire all'utente un'esperienza sempre più completa.

## Flash

Microsoft e Sun non erano le uniche società interessate a risolvere il problema delle pagine Web dinamiche. Nell'estate del 1996 FutureWave rilasciò un prodotto chiamato FutureSplash Animator. Nato da un lettore di animazioni basate su Java, FutureWave vendette in poco tempo la propria società a Macromedia che mise in commercio il prodotto con il marchio **Flash**.

Flash consentiva ai progettisti di creare applicazioni straordinarie di natura considerevolmente dinamica. Le società potevano fornire applicazioni web molto interattive che

poco si differenziavano dalle simili di tipo thick client. A differenza di applet, servlet e script CGI, Flash non richiedeva particolari abilità di programmazione ed era facile apprendere il funzionamento. Nei giorni di sviluppo febbrile della fine degli anni 90 questa caratteristica rappresentava un indiscusso vantaggio, perché in quel periodo le aziende faticavano a trovare dipendenti che disponessero delle abilità richieste. Tuttavia, questa semplicità di utilizzo aveva un prezzo.

Come molte soluzioni, Flash richiedeva il software sul client. Il plug-in Shockwave Player richiesto era distribuito insieme a numerosi sistemi operativi e browser di uso comune, non era però universale. Nonostante fosse possibile scaricarlo gratuitamente, per il timore dei virus molti utenti si rifiutarono di installare il software. Le applicazioni Flash potevano anche richiedere una certa quantità di larghezza di banda di rete per essere eseguite in modo corretto e la mancanza di connessioni a banda larga su una vasta area limitò l'adozione di Flash (da qui deriva l'introduzione del famoso link skip intro). Mentre alcuni siti scelsero di avere più versioni della propria applicazione Web, a seconda delle diverse velocità di connessione, molte società non poterono giustificare i costi di sviluppo aggiuntivi necessari per supportare due o tre siti di catalogo.

Concludendo, la creazione delle applicazioni Flash richiedeva software proprietario e plug-in del browser. Se le applet potevano essere scritte con editor di testo e un kit di sviluppo Java (JDK) gratuito, l'utilizzo del toolkit Flash completo costava centinaia di dollari a sede. Sebbene non si trattasse di problematiche insormontabili, questi fattori rallentarono l'adozione su vasta scala di Flash per le applicazioni Web dinamiche.

## **DHTML**

Quando Microsoft e Netscape rilasciarono la versione 4 dei propri rispettivi browser, gli sviluppatori Web ebbero a disposizione una nuova opzione: l'HTML dinamico, il **DHTML**. Nonostante ciò che si potrebbe pensare, DHTML non era uno standard W3C ed era molto più che uno strumento di marketing. In realtà, era una combinazione tra HTML, Cascading Style Sheets (CSS), JavaScript e DOM. La combinazione di queste tecnologie consentiva agli sviluppatori di modificare al volo il contenuto e la struttura di una pagina Web.

Inizialmente DHTML riscosse un buon successo. Tuttavia, richiedeva versioni browser che non erano state largamente adottate. Anche se Internet Explorer e Netscape supportavano entrambi il linguaggio DHTML, le rispettive implementazioni erano diverse e implicavano che gli sviluppatori sapessero quale browser era utilizzato dai loro client. Spesso ciò comportava la creazione di molto codice per controllare il tipo e la versione

del browser, il tutto con un aumento dei costi di sviluppo. Alcuni esitarono nell'adottare questo approccio a causa della mancanza di uno standard ufficiale per DHTML. Tutto ciò però apriva uno spiraglio verso nuove possibilità per il futuro.

## AJAX

Una nuova tecnica per lo sviluppo di applicazioni web interattive e che di recente ha suscitato notevole interesse da parte degli sviluppatori del web è denominata AJAX. L'intento di tale tecnica è quello di ottenere pagine web che rispondono in maniera più rapida, grazie allo scambio in background di richieste HTTP e senza che ci sia la necessità di ricaricare la pagina web ogni volta che l'utente effettua una modifica.

Non si fa qui una trattazione dettagliata di AJAX in quanto oggetto del capitolo 3.

### 2.2.2 Architettura

I componenti fondamentali di un'applicazione Web sono analoghi per certi versi a quelli di una tradizionale applicazione client/server. Una tipica applicazione client/server è costituita da un client che implementa l'interfaccia utente con alcune funzionalità di elaborazione e di comunicazione e da un server che fornisce una serie di servizi come la gestione e l'accesso ai dati di un database.

Nell'ambito Web l'interazione tra client e server è un po' più articolata per consentire l'integrazione di componenti di varia natura. Un'applicazione Web si basa su elementi software standard indipendenti dalle caratteristiche della particolare applicazione e dalla piattaforma software e hardware su cui viene eseguita.

Un'applicazione Web, nella maggior parte dei casi, si sviluppa su tre livelli logico-funzionali (applicazioni Three-Tier) ma che possono essere distribuiti anche su più livelli (applicazioni Multi-Tier):

1. **livello di presentazione** - rappresenta l'interfaccia utente dell'applicazione e si occupa di acquisire dati e visualizzare risultati;
2. **livello intermedio** - si occupa delle elaborazioni dei dati in base alla cosiddetta business logic, cioè all'insieme delle regole per cui i dati sono considerati significativi e le loro relazioni consistenti, le elaborazioni del livello intermedio generano i risultati richiesti dall'utente;
3. **livello dati** - rappresenta l'insieme dei servizi offerti da applicazioni indipendenti

dal Web, come ad esempio un gestore di database, un sistema di gestione di posta elettronica, ecc.

Non sempre i livelli logici di un'applicazione Web corrispondono a locazioni fisiche sulla rete. Si va dal caso in cui tutti e tre i livelli risiedono sulla stessa macchina a varie altre distribuzioni fino alla corrispondenza di ciascun livello con una macchina fisica.

In figura 2.1 è mostrata l'architettura tipica di un'applicazione Web.

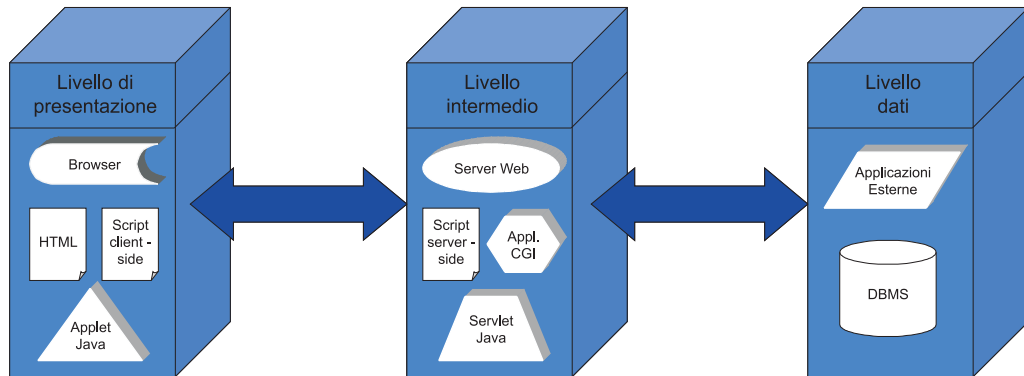


Figura 2.1: Architettura Three-Tier di una Applicazione Web.

Di seguito vengono analizzati i livelli architetturali di una tipica applicazione Web e le varie modalità di progettazione.

### Livello di presentazione

Il livello di presentazione costituisce l'interfaccia utente dell'applicazione Web e corrisponde a quello che nelle applicazioni client/server standard è il client. Esso è costituito da vari componenti combinati tra loro: browser, documenti HTML, applet Java, controlli ActiveX. La capacità di utilizzo di questi elementi da parte della piattaforma client è uno dei problemi principali nella realizzazione di questo livello. Le soluzioni vanno dalla scelta di sfruttare al massimo le capacità elaborative del client, alla realizzazione di un livello di presentazione universale. In genere l'identificazione del contesto di esecuzione dell'applicazione contribuisce in larga parte a delineare le soluzioni da adottare nella realizzazione del livello di presentazione.

Un altro elemento da prendere in considerazione è il livello di complessità dell'interfaccia utente dell'applicazione. Il linguaggio HTML, nato per la distribuzione di documenti in ambienti distribuiti, non consente di progettare interfacce utenti molto avanzate (drag-and-drop, tree-view control, tabbed control, etc.).

### Livello intermedio

Il livello intermedio di un'applicazione Web contiene la logica elaborativa dell'applicazione. Esso è in grado di soddisfare le richieste di dati e di elaborazione del client. Le modalità di realizzazione del livello intermedio dipendono spesso dalle caratteristiche e dalle tecnologie supportate dal server Web e/o da componenti installati sul server applicativo. In ogni caso la funzionalità fondamentale del server Web su cui si basa l'intera applicazione è il supporto di elaborazioni.

In base alla tipologia di applicazione da sviluppare è opportuno prevedere funzionalità particolari, come ad esempio:

- **gestione delle transazioni:** gestione affidabile del flusso dati;
- **gestione della sicurezza e della riservatezza:** gestione della sicurezza nell'accesso all'applicazione e la riservatezza nella trasmissione di informazioni; la gestione della sicurezza è spesso realizzata tramite l'autenticazione della password d'utente, mentre la riservatezza si basa su protocolli ad hoc, come SSL (Secure Socket Layer) o PCT (Private Communications Technology).

Il livello intermedio di un'applicazione Web può essere costituito da un insieme di script, componenti e programmi interagenti tra di loro e con il server Web tramite le seguenti tecnologie:

- **Common Gateway Interface (GCI):** consente l'attivazione di un programma o di uno script su richiesta del client (portabile su qualsiasi piattaforma).
- **Internet Server Application Programming (ISAPI):** consente l'esecuzione di una libreria dinamica (DLL) all'interno dello spazio di memoria del server Web (funziona solo su piattaforma Windows).
- **Active Server Pages (ASP):** consente l'interpretazione di script nell'ambiente del server Web e la creazione in modo dinamico di documenti Web (funziona solo su piattaforma e server Web Windows).
- **Java Servlet:** consente di eseguire classi Java su richiesta del client (portabile su qualsiasi piattaforma)

### Livello dati

Il livello dati fornisce servizi non direttamente disponibili tramite il Server Web. Questi servizi sono generalmente forniti da applicazioni indipendenti dall'ambiente Web e spes-

so costituiscono le applicazioni preesistenti in un'azienda. Tipici esempi di applicazioni presenti a questo livello sono:

- server dati (DBMS);
- server di mail;
- server di documentazione elettronica.

In genere è opportuno prevedere dei componenti dell'architettura dell'applicazione che fungono da connettori tra il livello intermedio e il livello dati. Infatti, utilizzando dei connettori per l'interazione con applicazioni esterne non standard si facilita la manutenibilità nel caso in cui queste vengano modificate o sostituite (es. ODBC, OLEDB). Per approfondimenti si veda [2].

### 2.2.3 Vantaggi

Le applicazioni Web si pongono come valida alternativa alle tradizionali applicazioni Client/Server per vari motivi:

- **facilità di distribuzione e aggiornamento:** un'applicazione Web si trova interamente sul server, per cui la pubblicazione sul server coincide con la distribuzione e l'aggiornamento effettuato sul server è automaticamente reso disponibile a tutti gli utenti;
- **accesso multiplatforma:** l'accesso all'applicazione è indipendente dall'hardware e dal sistema operativo utilizzato dagli utenti;
- **riduzione del costo di gestione:** l'uso di Internet come infrastruttura per un'applicazione Web riduce notevolmente sia i costi di connettività che i costi di gestione dei client;
- **scalabilità:** un'applicazione Web ben progettata può crescere insieme alle esigenze dell'azienda senza particolari problemi.

### 2.2.4 Contesto di esecuzione

Un elemento molto importante nella progettazione di applicazione Web è il contesto in cui verrà eseguita l'applicazione: Internet, Intranet o Extranet.

Se il contesto di applicazione è **Internet**, occorrerà prendere in considerazione la compatibilità con diversi tipi e versione di browser, sarà necessario tenere conto delle

velocità di accesso alle risorse e di conseguenza alla riduzione delle dimensioni del materiale da scaricare e all'ottimizzazione dei tempi di risposta da parte del server. Nel caso di trasferimento di documenti e informazioni riservate sarà opportuno prevedere l'uso di protocolli sicuri.

Nel caso di **Intranet** vengono meno le considerazioni relative alla compatibilità con i browser e alla velocità di accesso alle risorse, per cui si ha una maggiore libertà nello sviluppo dell'applicazione.

Se il contesto di esecuzione è **Extranet**, cioè una Intranet accessibile totalmente o in parte anche dall'esterno, è necessario riconsiderare la parte accessibile dall'esterno in base a considerazioni fatte sul contesto Internet. In ogni caso una particolare cura dovrà essere posta nel controllo delle autorizzazioni di accesso all'applicazione da parte degli utenti esterni.

### 2.2.5 Problematiche

La realizzazione di applicazioni Web presenta alcune problematiche sconosciute alle applicazioni tradizionali. Queste problematiche derivano dalla natura stessa del Web, pensato originariamente non per la realizzazione di applicazioni ma per la distribuzione di documenti. Un classico problema è quello del cosiddetto mantenimento dello stato dell'applicazione. Il modello di interazione di base del Web è privo del concetto di connessione, per cui l'esecuzione di un'applicazione Web consiste in una serie di interazioni disconnesse. Lo sviluppatore deve utilizzare informazioni codificate nella richiesta HTTP e nelle risorse condivise durante la sequenza di interazioni per stabilire la continuità dell'applicazione. La progettazione di un'applicazione Web deve tenere conto anche dell'ampiezza di banda di trasmissione disponibile e del carico di lavoro stimato per il server. Dalle considerazioni sull'ampiezza di banda derivano le scelte relative alla quantità di dati da inviare ad un client in risposta ad una richiesta, ciò può coinvolgere anche l'interfaccia utente dell'applicazione stessa, portando in certi casi a rinunciare ad una veste grafica accattivante pur di fornire tempi di risposta accettabili.

Tuttavia, all'ottimizzazione dell'uso dei mezzi trasmissivi può contribuire, e in certi casi essere determinante, un'attenta ripartizione del carico di elaborazione tra client e server. Infatti, consentire al client di effettuare alcune elaborazioni senza coinvolgere il server, come ad esempio la convalida dei dati inseriti in una form, permette di ridurre il traffico sulla rete sia di snellire il carico di lavoro del server. Inoltre, la presenza di applicazioni esterne particolarmente complesse sulla stessa macchina del server Web può rendere meno efficiente l'applicazione Web sottraendole risorse preziose. In questi casi è

opportuno distribuire il carico di lavoro su macchine diverse prevedendo, ad esempio, una macchina diversa per un DBMS o per un'applicazione particolarmente esosa di risorse.

Non sempre i criteri tipici della progettazione di applicazioni client/server possono essere applicati alla progettazione di applicazioni Web. Infatti, nella maggior parte dei casi la natura stessa del Web e delle problematiche che esso presenta porta a basare le proprie scelte progettuali sulla propria esperienza o sulla esperienza acquisita da altri, in mancanza di modelli di riferimento consolidati.



# Capitolo 3

## AJAX

La rete Internet cui siamo abituati oggi ha subito numerosi cambiamenti. Se all'inizio era solo un mezzo costituito da semplici browser di testo usati dagli scienziati per scambiarsi le ricerche, oggi Internet è il fulcro delle attività di commercio e informazione. Nel corso del tempo abbiamo assistito all'affermazione di numerose nuove tecnologie e approcci: dai primi browser grafici ai podcast. Attualmente Internet è la piattaforma principale per numerose applicazioni, ma nonostante la praticità garantita, pochi confonderebbero un'applicazione Web con la corrispondente applicazione desktop. I clienti oggi cercano un'applicazione sempre più completa e gli sviluppatori preferiscono evitare la distribuzione degli eseguibili a migliaia di workstation.

Le applicazioni Desktop hanno una completezza e una velocità di risposta elevata che risultano essere fuori dalla portata del Web. La semplicità che ha portato alla proliferazione del Web ha creato un divario tra le esperienze che si possono fornire mediante le applicazioni Web e le applicazioni Desktop. Fortunatamente quel divario oramai non è più enorme perchè oggi è disponibile un'altra opzione, un altro strumento per creare applicazioni basate sul browser veramente complete, oggi è disponibile AJAX.

### 3.1 Nascita e sviluppo

**AJAX**, acronimo di *Asynchronous JavaScript and XML*, è una tecnica di sviluppo web per creare applicazioni web interattive. Questo termine è stato enunciato per la prima volta da "Jesse James Garrett" di "Adaptive Path", il 18 Febbraio 2005, come titolo di un post all'interno del suo blog. Nel suo post intitolato "*AJAX A New Approach to Web Applications*"[3] Garrett tratta di come si stia colmando il distacco esistente da sempre tra applicazioni thick client o desktop e applicazioni thin client o Web. Chiaramente,

Google ha contribuito a elevare il profilo delle tecniche quando ha rilasciato Google Maps e Google Suggest nel Google Labs, inoltre sono stati pubblicati diversi articoli sul tema.

AJAX non è una nuova tecnologia né un'invenzione bensì un concetto utilizzato per sviluppare applicativi web avanzati. Il concetto è in parte espresso nell'acronimo scelto, un utilizzo asincrono di Javascript che attraverso l'interfacciamento con XML, può permettere ad un client di richiamare informazioni lato server in modo veloce e trasparente, allargando gli orizzonti delle rich internet applications[17]. Queste applicazioni fino a poco tempo fa erano legate principalmente alle tecnologie Adobe-Macromedia Flash o Java (con le applet). Entrambe purtroppo non sempre interpretabili dai client degli utenti e troppo spesso usate a sproposito con il solo scopo di stupire, discorso che spesso e purtroppo vale anche oggi.

In alternativa a queste tecniche di interazione client/server, quando nel 1996 venne introdotto l'iframe in Internet Explorer 3, molti sviluppatori sfruttarono quest'ultimo modificando l'attributo sorgente (src) della pagina racchiusa e simulando così un refresh trasparente di una parte di contenuti il che emulava, in modo abbastanza sporco, un'interazione asincrona. Nel 1998 Microsoft cominciò a sviluppare una tecnologia, chiamata "Remote Scripting", con lo scopo di creare una tecnica più elegante per richiamare contenuti differenti ed è in questo periodo, seppur con nome differente, che AJAX venne utilizzato per la prima volta, per poi evolversi in versioni più mature fino a diventare un oggetto vero e proprio, noto ora come "XMLHttpRequest".

Il motivo principale di tanto successo è che solo ultimamente il Remote Scripting ha suscitato lo stupore degli addetti ai lavori nel vedere cosa Google fosse riuscita a fare all'interno dei suoi applicativi senza necessità di Flash Player o Java Virtual Machine, mantenendo comunque la compatibilità con molteplici browser di utenti che per diversi motivi non potevano usufruire di Javascript.

Per ulteriori approfondimenti si può vedere [4][5].

## 3.2 Funzionamento di AJAX

AJAX, come già detto, non è una tecnologia ma è l'insieme di più tecnologie indipendenti che insieme danno vita a potenti prospettive. AJAX incorpora in se:

- Presentazione standard utilizzando XHTML e CSS;
- Layout dinamico con possibilità di interazione mediante il Document Object Model (DOM);

- Scambio e manipolazione dati mediante l'utilizzo di XML e XSLT;
- Recupero asincrono dei dati utilizzando XMLHttpRequest;
- Javascript che amalgama il tutto.

Le classiche applicazioni Web lavorano in questo modo: molte delle operazioni non sono altro che una richiesta HTTP inoltrata al web server. Il server fa l'elaborazione (raccolta dati, manipolazione numeri, dialogo con eventuali sistemi legacy) e ritorna una pagina HTML al client. Questo è un modello adottato sin dalle origini del Web e che lo rende ottimale per l'ipertesto ma non necessariamente per le applicazioni software.

L'approccio adottato da AJAX fa però nascere una miriade di soluzioni/quesiti tecnici che stravolgono i classici modelli del Web. Mentre il server elabora, l'utente cosa fa? Aspetta, e sarà costretto a farlo ad ogni richiesta. Naturalmente, se si progetta un'applicazione Web come un'applicazione software, questo non dovrebbe accadere. Una volta che è stata caricata l'interfaccia, perchè l'interazione dell'utente deve fermarsi ogni volta che l'applicazione ha bisogno di interagire con il server? Perchè l'utente deve vedere l'applicazione dialogare con il server così pesantemente?

### 3.2.1 AJAX fa la differenza

Un'applicazione AJAX elimina la natura del Web ad essere contraddistinta da momenti alternati di dialogo-risposta che fanno attendere l'utente, introducendo un intermediario (un motore AJAX) tra l'utente e il server. Aggiungere un livello all'applicazione potrebbe far pensare ad una risposta più lenta, ma è vero il contrario. Invece di caricare una classica pagina web, ad inizio sessione, il browser carica il motore AJAX scritto in Javascript e usualmente associato ad un frame invisibile. Questo motore è responsabile della comunicazione tra l'interfaccia utente e il server, mettendoli in comunicazione tra di loro. Il motore AJAX, indipendentemente dalla comunicazione con il server, permette all'interazione dell'utente con l'applicazione di essere asincrona. In questo modo l'utente non è mai succube della finestra del browser bianca e dell'iconcina indicante il caricamento, nell'attesa che il server esegua le operazioni

Ogni azione dell'utente che normalmente genera una richiesta HTTP prende la forma di una chiamata Javascript verso il motore AJAX. Ogni risposta che non richiede la chiamata diretta al server (come validare alcuni dati, modificare dati in memoria) è un lavoro compiuto dal motore AJAX. Se l'applicazione ha bisogno di interagire con il server, ad esempio per inviare dati, caricare un'interfaccia aggiuntiva o ricevere nuovi dati, il mo-

Il motore AJAX rende queste richieste asincrone, usualmente utilizzando XML, senza bloccare l'interazione tra utente e applicazione.

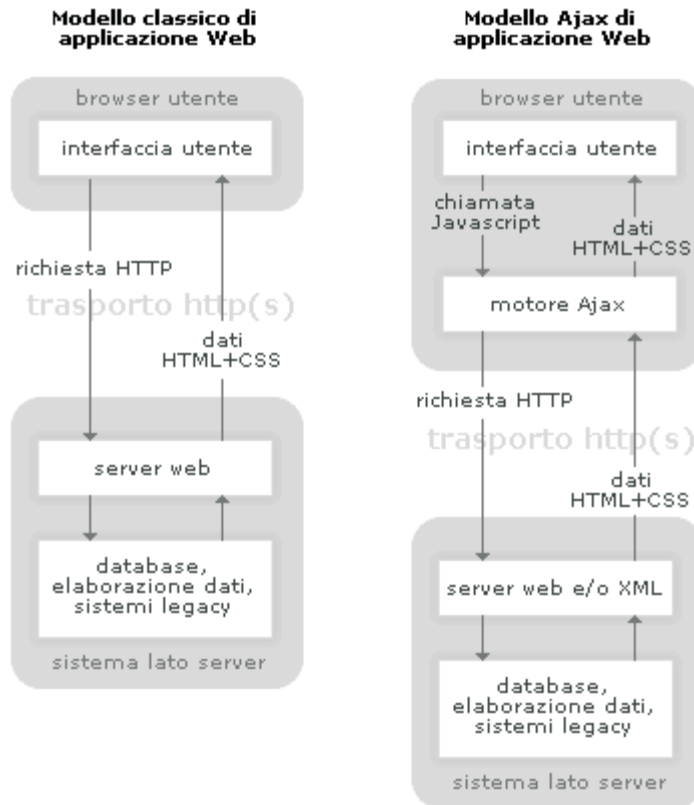


Figura 3.1: Il modello tradizionale delle applicazioni Web (alla sinistra) confrontato col modello AJAX (alla destra).

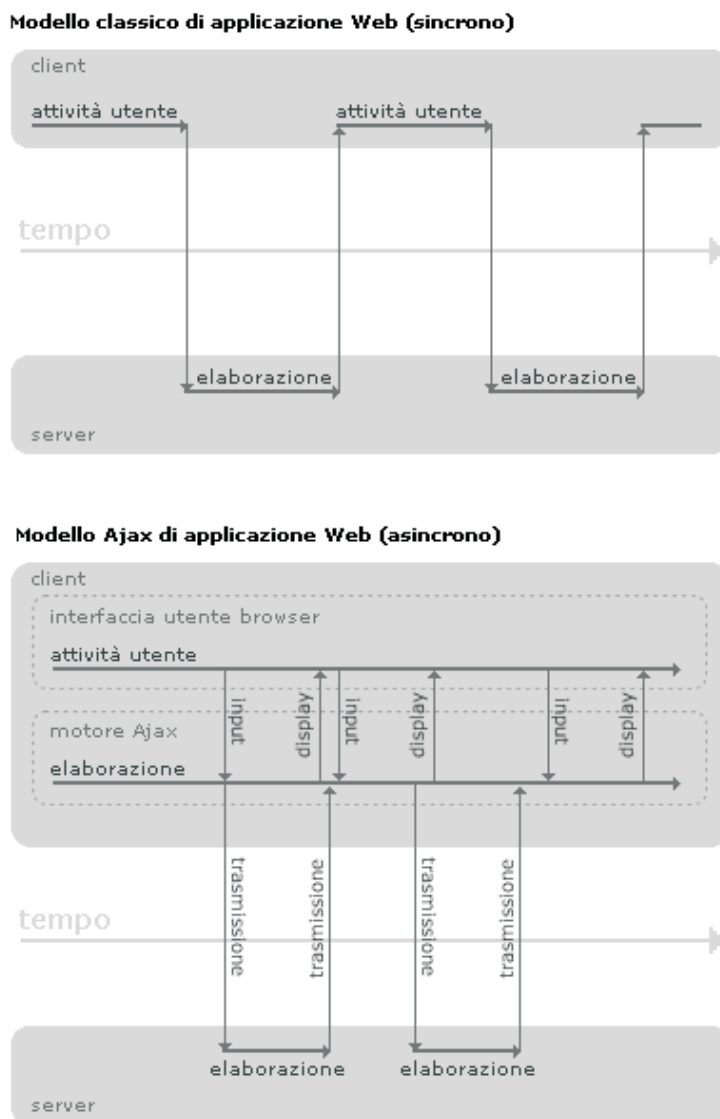


Figura 3.2: Modello di interazione sincrona di un'applicazione web (sopra) confrontato con il modello asincrono Ajax (sotto).

### 3.3 Descrizione tecnica

Dal punto di vista tecnico la caratteristica principale di AJAX è data dalla capacità che ha JavaScript di effettuare le seguenti due operazioni:

- Aggiornare parte di una pagina HTML senza che questa venga caricata nuovamente;
- Effettuare richieste tramite il protocollo HTTP.

Molto spesso si parla di AJAX anche quando soltanto una di queste due caratteristiche viene utilizzata.

La prima caratteristica evidenziata è quella di essere in grado di popolare una parte di una pagina con del testo definito in JavaScript. In particolare questa funzione può essere effettuata utilizzando il metodo `innerHTML` di un elemento o le specifiche DOM create appositamente per aggiungere, popolare o modificare elementi della pagina. Con il metodo `innerHTML`, non si fa altro che sovrascrivere il contenuto di un elemento del documento visualizzato e tutto quello che occorre è una funzione in grado di restituire l'elemento richiesto. Le funzioni più note per fare ciò sono `getElementById` e `getElementsByTagName` ed entrambe possono essere sfruttate per usare il metodo `innerHTML` oppure il DOM.

Entrando nei dettagli, `getElementById` è un metodo dell'oggetto `document` che cerca, nella pagina visualizzata, un tag HTML con un attributo assolutamente univoco (`id`). La semplice funzione `cambiaTesto()` mostrata nel listato 3.1 utilizza i due metodi appena descritti.

Listing 3.1: Funzione cambia

```
function cambiaTesto()
{
    var elemento = document.getElementById("ele");
    elemento.innerHTML = "Testo cambiato";
}
```

La seconda caratteristica di AJAX è realizzabile grazie al metodo `XMLHttpRequest`. A seconda del browser usato prende nomi differenti o viene richiamato in maniera differente. Nel caso di Internet Explorer, ad esempio, questo oggetto è restituito da un "ActiveXObject" mentre nei browser alternativi più diffusi (Mozilla, Safari, FireFox, Netscape, Opera ed altri) `XMLHttpRequest` è supportato nativamente, cosa che dovrebbe accadere anche per IE dalla versione 7.

Il metodo `XMLHttpRequest` permette di effettuare la richiesta di una risorsa (con HTTP) ad un server web in modo indipendente dal browser. Nella richiesta è possibile

inviare informazioni, ove opportuno, sotto forma di variabili di tipo GET o di tipo POST in maniera simile all'invio dati di un form. La richiesta è asincrona, il che significa che non bisogna necessariamente attendere che sia stata ultimata per effettuare altre operazioni, stravolgendo sotto diversi punti di vista il flusso dati tipico di una pagina web. Generalmente infatti il flusso è racchiuso in due passaggi alla volta, richiesta dell'utente (link, form o refresh) e risposta da parte del server per poi passare, eventualmente, alla nuova richiesta da parte dell'utente. Il listato 3.2 mostra l'uso della tecnica AJAX con il metodo XMLHttpRequest.

Listing 3.2: Richiesta HTTP con AJAX

```
// richiesta HTTP con AJAX
function AJAX(file, parametri)
{
    var XHR = new XMLHttpRequest();
    // funzione di risposta alla richiesta AJAX
    XHR.onreadystatechange = function()
    {
        if (XHR.readyState == 4 && XHR.status == 200)
        {
            // memorizza il testo di risposta del server
            var testoRisposta = XHR.responseText;
            // memorizza l'XML di risposta del server
            var xmlRisposta = XHR.responseXML;
        }
    };
    XHR.overrideMimeType("text/xml");
    XHR.open("POST", file, true);
    XHR.setRequestHeader("Content-Type", "application/x-www-form-urlencoded");
    XHR.send(parametri);
}
```

Altri esempi di uso dei metodi appena descritti possono essere reperiti nell' articolo [6].

## 3.4 Considerazioni lato utente

### 3.4.1 Vantaggi

L'impressione di avere a che fare con pagine apparentemente più interattive, dinamiche e professionali è praticamente immediata. L'effetto novità si aggiunge prepotentemente alla lista, stuzzicando la curiosità del navigatore. Questi aspetti apparentemente futili hanno permesso a tecnologie come Flash di affermarsi nel contesto web. Questo perchè l'interesse collettivo è la ricerca di informazioni, ma si preferisce navigare dove le informazioni sono presentate nel modo più semplice e veloce possibile.

La compatibilità praticamente totale con i browser più diffusi ed il supporto nativo, o

integrabile, dell'oggetto XMLHttpRequest contribuiscono ulteriormente allo sviluppo di applicazioni web dinamiche, allargando i campi di impiego per gli sviluppatori e coinvolgendo sempre di più i navigatori. Ottenere modifiche dinamiche e leggere sulle pagine, come i suggerimenti sui campi di ricerca, in grado di suggerire parole presenti e velocizzare l'inserimento, mappe, votazioni, statistiche, amministrazione, lezioni, chat, giochi e chi più ne ha più ne metta, ecco il motivo di tanto successo ed ecco perchè gli utenti possono trarre numerosi vantaggi da AJAX.

### 3.4.2 Svantaggi

Come già accennato in precedenza, gli utenti che necessitano di tecnologie assistive e coloro che non dispongono di connessioni veloci sono i più penalizzati. I primi perchè non possono, attualmente, godere del cambio parziale di contenuti e non saranno quindi in grado di sfruttarne le potenzialità. I secondi devono caricare, almeno al primo accesso, una mole di dati consistente rappresentata dai vari file JavaScript e potrebbero non essere in grado di sfruttare niente di quanto caricato. Si potrebbe non disporre di un browser aggiornato, oppure in situazioni di JavaScript disabilitato, di assenza dell'oggetto XMLHttpRequest. In questi casi si scaricano centinaia di bytes che non verranno mai usati a causa del controllo a posteriori utilizzato per conoscere la reale compatibilità con AJAX.

Questi aspetti puramente tecnici sono spesso aggirabili, grazie ad una stesura degradabile dell'applicativo da parte dello sviluppatore o grazie al caricamento dinamico del javascript proposto dopo aver verificato la compatibilità del browser. Ma ce n'è uno esclusivamente pratico, probabilmente il più noto nonché fastidioso: l'uso dei tasti "avanti" e "indietro" presenti nei browser. Qualsiasi interazione asincrona, dà comunque la sensazione di aver cambiato stato alla pagina e l'abitudine a tornare indietro, qualora il risultato non dovesse essere quello sperato, è intrinseca nel navigatore nonché lecita. Aprendo direttamente una pagina ricca di interazioni asincrone non sarà possibile, nemmeno volendo, cliccare sul tasto "indietro" del browser per tornare allo stato precedente, si verrà reindirizzati invece alla pagina precedente, il che può causare disorientamento. In questi casi, la consapevolezza di aver usato AJAX, qualora il navigatore sia preparato, non può certo aiutare, poichè anche l'aggiornamento della pagina non serve a tornare allo stato precedente ma solo allo stato iniziale. Oltre ad essere un vincolo di navigazione, questo problema è anche un vincolo per l'indicizzazione o la possibilità di segnalare ad altri la pagina visualizzata. Questo accade perchè il comando gestito da JavaScript non è portabile come un link, di conseguenza, se si vuole segnalare a qualcuno una pagina web visitata ed eventualmente modificata tramite AJAX non è possibile, se non dando delle



indicazioni precise sulle operazioni svolte una volta arrivati nella pagina in oggetto. La soluzione a questi problemi non è semplice ed i motivi sono diversi.

Una libreria nota e molto potente che sembra abbia risolto almeno in parte i problemi è quella di “backbase”[7], dove il caricamento incrementale del contenuto, attraverso la visualizzazione asincrona dello stesso, sembra essere la soluzione definitiva, gestendo sottogierarchie e le varie sezioni tramite una serie di informazioni appese dopo il carattere #, il quale permette solitamente di indicizzare sotto-contenuti di una pagina tramite l’uso di elementi con id univoco ma che può essere gestito senza problemi dal JavaScript proposto.

## 3.5 Considerazioni lato server

### 3.5.1 Vantaggi

È noto che usare fogli di stile esterni piuttosto che in linea ha l’effetto di decimare la banda divorata mensilmente dai navigatori, l’utilizzo di AJAX potrebbe fare altrimenti. Infatti con AJAX il server non ha necessità di trasferire tutta la pagina per ogni interazione, ma solo la porzione necessaria all’operazione richiesta. Questo rende sensibilmente più veloce l’interazione per l’utente e favorisce il risparmio di banda.

C’è una ricaduta anche in termini di calcoli da effettuare. Prendiamo il caso di un portale ricco di informazioni. Reperirle tutte per poter affrontare una interazione ‘tradizionale’ può richiedere uno stress relativo per diverse manciate di decimi di secondo (database, webservices etc.). Con AJAX le richieste sono puntuali ed il server può rispondere in modo più efficiente. Siti con migliaia di utenti simultanei non dovranno quindi operare su tutte le parti dell’applicativo, migliorando la possibilità di gestire mole di utenti elevata. Allo stesso tempo una parte dei calcoli può essere data in gestione al browser, così da sfruttare la potenza del PC-client e distribuire il carico su tutti, piuttosto che sul solo host.

Non bisogna ovviamente esagerare, pena il rallentamento del client qualora i calcoli dovessero essere troppi, va scremato minuziosamente l’utile dal superfluo e soprattutto non si dovrebbero creare applicativi così complessi da richiedere diversi secondi di attesa prima di poter essere utilizzati.

### 3.5.2 Svantaggi

Il principale rovescio della medaglia è causato dall’abuso di AJAX, gli sviluppatori meno accorti mettono a disposizione troppe interazioni su una sola pagina. Inoltre, se non si

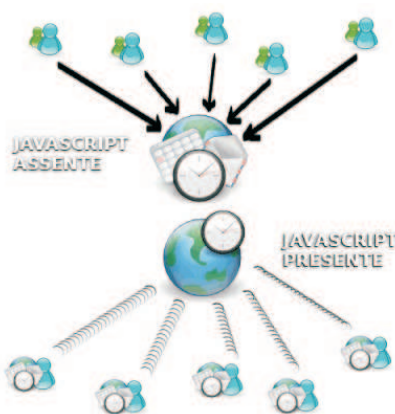


Figura 3.3: Carico del server con e senza AJAX.

controllano le operazioni sul client il server rischia di ritrovarsi sovraffollato di richieste probabilmente inutili.

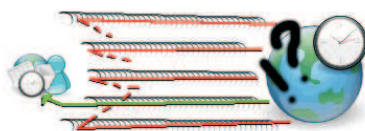


Figura 3.4: Chiamate “a vuoto”.

Altro esempio concreto è un suggest, miglior amico del navigatore impegnato a cercare informazioni e peggior nemico, se implementato male, dal server. Se per ogni lettera digitata si effettua una query al database il numero di operazioni si moltiplica in modo esponenziale ed il server rischia di rallentare o di intasarsi piuttosto che di essere alleggerito. La gestione dell'evento `onkeypress` (o in alternativa `onkeyup`) fatta ad hoc dovrebbe quindi valutare se una richiesta con quelle lettere sia già stata fatta ed in caso mostrarla grazie alla memorizzazione sul client dei risultati precedenti, oppure richiamare nuovamente il server solo dopo qualche istante e non immediatamente, eliminando l'operazione qualora l'utente dovesse cambiare idea in quel lasso di tempo, ad esempio nel caso abbia sbagliato a scrivere un carattere o abbia cancellato la parola da cercare. Salvo un archivio dati particolarmente contenuto, è altrettanto sconsigliabile restituire un riscontro dopo aver digitato la sola prima lettera come lo è dopo aver scritto dei caratteri inutili quali spazi, virgole, punti o altro. In questo caso solo dei rigorosi controlli client potrebbero salvaguardare il server ma allo stesso tempo questo dovrebbe impedire operazioni di ricerca inutili ed avere quindi a sua volta altrettanti controlli sul testo da cercare. Facile infatti risalire alle chiamate client e riproporle in altri modi, via socket o via form, evitare

quindi uno sperpero di risorse dovrà essere uno degli obiettivi principali dell'applicativo proposto.

Di esempi potenzialmente pesanti da gestire potrebbero essercene altri, come contatori di utenti on-line, chat rinfrescate ogni mezzo secondo, aggiornamenti di dati statistici troppo frequenti e l'unica soluzione, in questi casi, è trovare il giusto compromesso tra necessità effettiva della frequenza di aggiornamento e calcoli da effettuare per effettuarla.

## 3.6 Sicurezza

### 3.6.1 Lato client

Dalle considerazioni fatte in precedenza sui vantaggi portati da AJAX, soprattutto riguardo il server, verrebbe quasi spontaneo pensare ai client come a dei piccoli muletti ausiliari al quale delegare qualunque tipo di operazione. Questo non dovrebbe accadere, se non nella giusta misura, poichè mentre il server ha del codice compilato o interpretato, comunque non visibile ad occhi indiscreti, il JavaScript o i cookies sul client, non dovrebbero mai contenere informazioni sensibili quali password, numeri di carte di credito, informazioni riservate dalla legge sulla privacy o codici segreti. Allo stesso tempo bisogna essere molto cauti nell'inviare informazioni anche da parte del server, poichè potrebbe essere semplice replicare una richiesta asincrona da una pagina differente da quella proposta per tentare di leggere dati che si vorrebbero proteggere o non rendere pubblici.

La verifica del client è un'operazione complessa perchè la pagina di provenienza potrebbe non essere veramente quella riscontrata e l'unica soluzione per uno scambio dati asincrono sicuro è l'utilizzo di un'area protetta tramite SSL o altri sistemi dedicati. Soprattutto nelle aree di amministrazione sarebbe quindi consigliabile non inviare fin da subito tutto il codice di gestione interna ed aggiungerlo dinamicamente solo dopo aver effettuato il login.

Se si ha a che fare con un'area protetta che non prevede un utilizzo degradabile delle operazioni possibili, conoscere tutto il codice addetto a gestire le operazioni potrebbe aumentare in modo più o meno consistente la possibilità di ricevere attacchi esterni. Questo purtroppo accade molto spesso anche con aree di amministrazione in Flash, quando si lasciano tracce nei file SWF per risalire alle operazioni di modifica dei dati. È importante verificare sempre che le funzioni utilizzate lato client siano compatibili con tutti i browser previsti all'interno dell'area di amministrazione. Infatti, qualora una funzione non dovesse essere supportata a pieno ed il browser dovesse tentare lo stesso di utilizzarla si potrebbero aggiornare dati in modo errato o mostrare anomalie all'utente.

### 3.6.2 Lato server

Un errore purtroppo comune, e non recente, è quello di fare affidamento solo sul codice client. Questo accade spesso in applicativi Flash e l'unica fortuna di alcuni siti è non contenere dati interessanti per utenti maliziosi. L'errore è presumere che sia solo un certo client con un certo codice ad inoltrare le richieste e che tali richieste rispettino a pieno le regole imposte dallo sviluppatore.

Esistono diversi applicativi che non solo non verificano le autorizzazioni dell'utente ma danno per scontato che se il codice client ha la possibilità di inviare solo un numero, ad esempio un 'id', questo arrivi sicuramente sotto forma di un intero maggiore di zero. Leggendo il codice client, creandosi una semplice form html, ed inviando una richiesta tipo `id=0` or `id>0` si potrebbero spolverare decine di database attualmente online, archivi di news, utenti o altro ancora. Il mito da sfatare è quindi che l'uso di AJAX sia pericoloso poichè a renderlo tale può essere solo lo sviluppatore del sistema.

Non esistono controindicazioni specifiche per la sicurezza se non le solite, controllare sempre scrupolosamente che il dato in ingresso, GET o POST che sia, corrisponda al tipo di dato previsto, a prescindere che sia un'operazione da svolgere all'interno di un'area di amministrazione che all'interno di una qualunque pagina destinata agli utenti. Non è una problematica nuova o complessa, è solo la solita inerente le "SQL injections". Allo stesso tempo su un semplice form di contatti è assolutamente sbagliato fare affidamento ai soli controlli JavaScript. Quante volte è capitato di non poter inviare un messaggio in un'area contatti e disabilitando il JavaScript la stessa area permette di mandare il server in errore? Superfluo dire che uno sviluppatore esperto potrebbe sfruttare come meglio crede queste possibilità come via di entrata, come potrebbe sfruttare eventuali valutazioni del codice server o client.

Qualunque sia il prodotto proposto, questo dovrebbe essere prima sicuro senza JavaScript, da aggiungere solo al fine di alleggerire alcune operazioni. Si potrebbe pensare ad un controsenso, poichè se bisogna verificare tutto, dove stà il vantaggio per il server? Semplice, invece di ricevere tante richieste quanti sono gli eventuali errori, il client potrebbe aiutare l'utente a non sbagliare richiesta permettendo al server di controllarne ed elaborarne solo una presumibilmente corretta.

Quanto detto è rivolto soprattutto agli sviluppatori web, poichè in una intranet le possibilità che qualcuno faccia di proposito dei danni saranno sicuramente pochissime se non nulle. Si consiglia comunque una rigorosa cautela, la stessa che si usa per un applicativo senza JavaScript[8].

# Capitolo 4

## La Intranet aziendale

### 4.1 Il contesto aziendale

La **CNA** (*Confederazione Nazionale dell'Artigianato e della Piccole e Medie Imprese*) è il sistema generale nazionale ed unitario di rappresentanza delle imprese artigiane, dei loro imprenditori, delle piccole e medie imprese e delle relative forme associate. Il sistema CNA è costituito da 20 Federazioni CNA Regionali, 106 Associazioni Provinciali e tante Associazioni di Mestiere e/o Settore, il che significa l'intera copertura nazionale.

Ai suoi tanti associati la CNA offre una gamma di servizi davvero ampia:

- **contabilità, fisco e centri autorizzati di assistenza fiscale:** grazie al proprio software CNA personalizza, per ogni impresa, il piano dei conti e la riclassificazione di bilancio;
- **amministrazione del personale:** grazie ad un software fortemente innovativo, che permette la stampa di buste paga chiare e facilmente leggibili, anche ai meno esperti, la CNA è in grado di gestire gli oneri relativi al personale ai fini gestionali, ripartendoli per centri di costo;
- **consulenza a neo imprese, formazione aziendale e manageriale:** si occupa di progettazione di azioni formative per i diversi settori dell'artigianato e della piccola industria;
- **sportello donne imprenditrici:** CNA rivolge la sua attenzione al progetto impresa donna con un servizio specifico e specialistico per la consulenza e l'assistenza alle donne che intendono avviare nuove attività imprenditoriali;

- **qualità aziendale:** la qualità aziendale con applicazione delle normative UNI EN ISO 9000 - UNI EN 46000, la certificazione del prodotto e la qualifica dei processi speciali in conformità alle direttive UNI EN sono alcuni dei servizi che CNA propone alle imprese nel settore della qualità;
- **privacy informatica:** aggiornamento costante alle imprese sulla normativa relativa alla privacy attraverso seminari tecnici ed opuscoli informativi.

E tanti altri servizi consultabili sul sito [9].

## 4.2 Le richieste di CNA

Il sistema CNA è una struttura complessa, più di 70 sedi, una ventina di società di scopo, più di 800 addetti con bisogni formativi assai diversi tra di loro. Col tempo si sono create una miriade di archivi condivisi, procedure/prassi di interscambio di informazioni diverse tra di loro e non integrate. Un'altro importante problema è costituito dalla proliferazione delle password di accesso che merita un'analisi a sè stante: alcune figure arrivano, tra procedure interne e siti esterni, anche a 15-20 credenziali di autenticazione. Il piano di lavoro 2006 della CNA Servizi Modena, sulla base delle problematiche descritte, ha previsto nel suo progetto la realizzazione di una intranet aziendale.

Le specifiche di progetto richiedevano la realizzazione di diverse applicazioni web all'interno della intranet. Alcune di queste applicazioni, come quelle relative alla gestione di fatture e dei documenti di sede, erano già state realizzate in precedenza ed erano disponibili, mediante una procedura di autenticazione, sul sito pubblico di CNA Modena. Le specifiche hanno previsto il trasferimento di queste applicazioni web all'interno della rete intranet. I principali motivi che hanno portato a questa decisione sono dati dal fatto che alcune applicazioni web sono utilizzate solo all'interno dell'azienda ma anche perchè, in questo modo, si elimina il problema della procedura di identificazione, da effettuare ogni volta che si vuole accedere ad esse. Altre applicazioni, come la rubrica e la gestione circolari, sono invece state realizzate da zero ed inserite poi nel contesto Intranet.

Un altro degli obiettivi previsti nel progetto era quello di costruire, per ogni addetto, un "costrutto" tramite il quale poter accedere a tutte e sole le informazioni necessarie per lo svolgimento del proprio lavoro con il minor numero di password possibile. Questo meccanismo viene più propriamente detto sistema *single sign-on*[17].

## 4.3 La Intranet CNA

Dopo aver definito le caratteristiche principali di una intranet aziendale e analizzato le richieste dell'azienda si sono delineate le specifiche di progetto che hanno portato alla realizzazione della Intranet CNA.

Il CMS utilizzato per la realizzazione della intranet è denominato WebProfessional[10]. Questo CMS, realizzato dall'azienda CPSoft, è stato già utilizzato in precedenza dalla CNA Modena per la realizzazione di vari siti web di aziende ad essa collegate ma anche del proprio portale web. La scelta fatta dalla CNA Modena di non utilizzare un CMS open-source, ma uno a pagamento, nonostante in rete se ne trovino diversi, è giustificato dal fatto che l'azienda CPSoft fornisce un servizio di assistenza completo. Il servizio di assistenza fornito non riguarda esclusivamente la risoluzione di problemi riscontrati nell'utilizzo del software ma provvede anche all'installazione e configurazione del server web che ospita il CMS.

### 4.3.1 Specifiche di progetto

Il progetto ha previsto la realizzazione di una intranet denominata "Intranet CNA" all'interno dell'azienda CNA Modena. L'accesso alla intranet è consentito esclusivamente ai dipendenti dell'azienda CNA Modena i quali sono in possesso di username e password personali, la intranet pertanto prevede per l'accesso una procedura di autenticazione.

Le sezioni contenute nella intranet sono le seguenti:

1. **Documenti di sede**
2. **Fatture**
3. **Rubrica**
4. **Circolari**
5. **Applicazioni Internet**

Le prime due sezioni sono state trasferite dal sito pubblico CNA Modena alla Intranet CNA, pertanto non saranno descritte di seguito. Le restanti sezioni invece sono state realizzate secondo le specifiche previste.

Per quanto riguarda la sezione denominata "Applicazioni Internet" è stata progettata e realizzata in modo differente dalle altre sezioni. Questa sezione, infatti, deve permettere il meccanismo di *single sign-on* anche per le applicazioni web esterne alla Intranet. Per

svolgere la funzione prevista questa sezione deve essere utilizzata insieme al programma di gestione password, che sarà presentato nel successivo capitolo.

### 4.3.2 Configurazione CMS

Le prime operazioni effettuate hanno riguardato la configurazione generale del CMS e l'aspetto grafico del sito. Nella sezione Web-Power del CMS si sono definite le seguenti impostazioni:

- dati dell'azienda, che compaiono nella parte inferiore del sito (footer);
- font, dimensioni e stili dei caratteri;
- logo dell'azienda, che compare nella parte superiore del sito (header);
- immagine di sfondo;
- titolo;
- layout.

Le funzioni ed i contenuti sono invece stati definiti nella sezione del CMS denominata DB-Tools. La realizzazione delle sezioni della intranet è stata preceduta da una fase di progettazione della base di dati. Questa fase ha permesso di implementare tutte le funzioni previste nelle specifiche di progetto. Di seguito, per ogni sezione della intranet, è riportata la descrizione e il progetto della base di dati. Sono inoltre presentate immagini di esempio (screenshots) che mostrano alcune delle funzioni realizzate nelle singole sezioni.

## 4.4 Sezione Rubrica

### 4.4.1 Descrizione

La sezione "Rubrica" ha lo scopo di facilitare l'utente nella ricerca di numeri di telefono, di cellulare, di fax ed indirizzi e-mail dei dipendenti dell'azienda o delle aziende a questa collegate. Ogni contatto della rubrica ha i seguenti attributi:

- nome;
- telefono;
- cellulare;



- fax;
- e-mail;
- azienda;
- sede;
- settore (nel caso sia un dipendente).

L'accesso in lettura è consentito a tutti gli utenti i quali possono effettuare ricerche specificando vari attributi per trovare il contatto desiderato. La rubrica è tenuta sempre aggiornata dagli addetti alla rubrica che quindi hanno accesso anche in scrittura.

## 4.4.2 Progetto Database

### Schema E/R

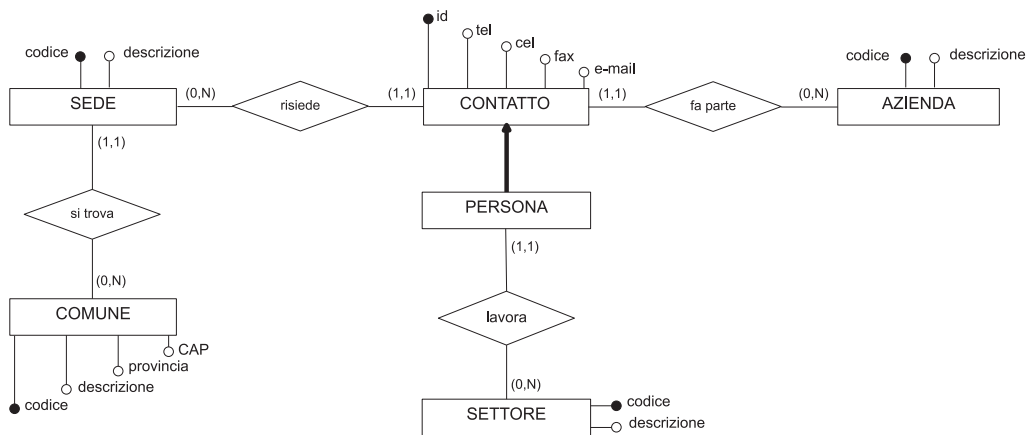


Figura 4.1: Schema E/R sezione Rubrica.

### Schema Relazionale

AZIENDA (CODICE, DESCRIZIONE)

SETTORE (CODICE, DESCRIZIONE)

COMUNE (CODICE, DESCRIZIONE, PROVINCIA, CAP)

SEDE (CODICE, DESCRIZIONE, COD\_COMUNE)

FK: COD\_COMUNE REFERENCES COMUNE

CONTATTO (ID, TEL, CEL, FAX, E-MAIL, COD\_SEDE, COD\_AZIENDA, COD\_SETTORE)

FK: COD\_SEDE REFERENCES SEDE

FK: COD\_AZIENDA REFERENCES AZIENDA

FK: COD\_SETTORE REFERENCES SETTORE

### 4.4.3 Funzioni

#### Ricerca contatto

La funzione “Ricerca contatto” permette di ricercare informazioni riguardanti numeri di telefono, fax e e-mail di tutti i dipendenti dell’azienda. La figura 4.2 mostra come sia possibile effettuare una ricerca in base a diversi criteri.

Figura 4.2: Funzione ricerca contatto.

#### Inserimento nuovo contatto

La funzione “Inserisci contatto” può essere eseguita solo dagli addetti alla rubrica. La funzione mostrata in figura 4.3 permette di inserire, modificare e cancellare contatti presenti nella rubrica.

## 4.5 Sezione Circolari

### 4.5.1 Descrizione

Nella sezione “Circolari” è visionabile l’archivio di tutte le circolari, ossia delle comunicazioni che l’azienda fa ai suoi dipendenti, è quindi una sezione accessibile a tutti gli

Figura 4.3: Funzione inserisci nuovo contatto.

utenti. La prima pagina della sezione mostra le ultime circolari inserite con le relative informazioni e i collegamenti ai rispettivi file. Ogni circolare ha i seguenti attributi:

- data di emissione;
- note/descrizione;
- numero protocollo;
- file allegato;
- categoria circolare;
- titolare paghe;
- sezione paghe.

L'utente può effettuare ricerche in base alla data, alla categoria, al titolare e alla sezione paghe. L'inserimento delle circolari può essere effettuato solamente dagli addetti alle circolari.

## 4.5.2 Progetto database

### Schema E/R

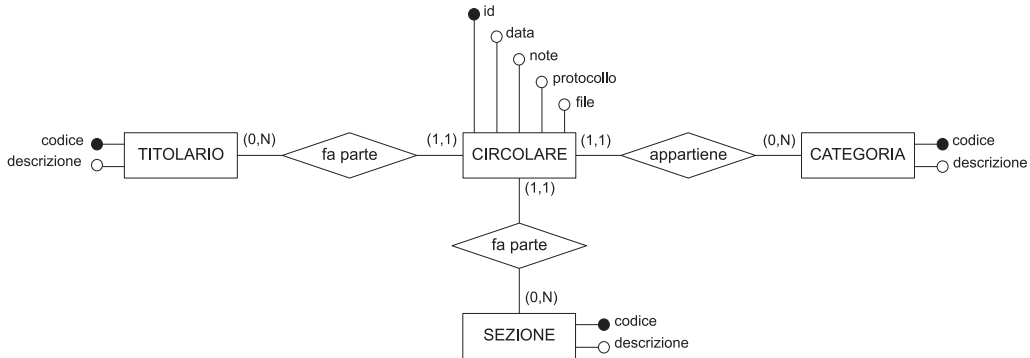


Figura 4.4: Schema E/R sezione Circolari.

### Schema Relazionale

CATEGORIA (CODICE, DESCRIZIONE)

TIPO (CODICE, DESCRIZIONE)

TITOLARIO (CODICE, DESCRIZIONE)

SEZIONE (CODICE, DESCRIZIONE)

CIRCOLARE (ID, DATA, NOTE, PROTOCOLLO, FILE,  
 COD\_CATEGORIA, COD\_TITOLARIO, COD\_SEZIONE)

**FK:** COD\_CATEGORIA REFERENCES CATEGORIA

**FK:** COD\_TITOLARIO REFERENCES TITOLARIO

**FK:** COD\_SEZIONE REFERENCES SEZIONE

## 4.5.3 Funzioni

### Ricerca circolari

La funzione “Ricerca circolari” permette all’utente di effettuare ricerche di circolari specificando diversi parametri. La figura 4.5 mostra l’interfaccia della funzione.

### Inserimento nuova circolare

La funzione “Inserisci circolare” può essere eseguita solo dagli addetti alle circolari. Questa funzione permette di inserire, modificare e cancellare le circolari. La figura 4.6 mostra l’interfaccia di questa funzione.

Figura 4.5: Funzione ricerca circolari.

## 4.6 Sezione Applicazioni Internet

Tutte le sezioni della *Intranet CNA*, viste finora, sono state concepite per essere utilizzate come applicazioni web per Intranet. La **Sezione Applicazioni Internet**, invece, è stata progettata e realizzata in modo differente. Questa sezione infatti, per l'utente Intranet, rappresenta un punto d'accesso alle applicazioni web Internet e quindi a siti web esterni. La creazione di questa sezione permette di estendere il concetto di *single sign-on* anche alle applicazioni web che non fanno parte della rete Intranet.

### 4.6.1 Descrizione

La sezione *Applicazioni Internet* consente agli utenti della *Intranet CNA* di accedere in modo rapido a tutte le applicazioni web Internet necessarie per lo svolgimento del proprio lavoro. Come è possibile notare dalla figura 4.7 la sezione è costituita da una pagina web contenente una lista di collegamenti a siti web esterni, questi collegamenti rappresentano le applicazioni Internet utilizzate da generico utente della *Intranet CNA*. Ogni utente ha la possibilità di personalizzare la propria lista di collegamenti, inserendo quelli preferiti, attraverso il menu di gestione presente nella sezione.

Per garantire un accesso di tipo *single sign-on* anche alle applicazioni web Internet, è necessario che l'utente utilizzi il programma "Intranet Password Manager". Questo programma, realizzato per l'uso Intranet, è stato installato su tutti i browser che si connettono alla rete *Intranet CNA* e si presenta come una toolbar per il browser Firefox. "Intranet

Figura 4.6: Funzione inserisci nuova circolare.

“Password Manager” permette di memorizzare le credenziali di accesso ai siti web e di effettuare il login in modo automatico, così come viene fatto dai classici programmi di gestione password incorporati nei browser web. La scrittura e la lettura delle credenziali di accesso viene eseguita sul database della Intranet ed è proprio questa caratteristica che differenzia “Intranet Password Manager” dai programmi di gestione password incorporati nei browser web. Con “Intranet Password Manager” le password sono memorizzate nel database della Intranet e quindi accessibili da tutti i computer collegati alla rete *Intranet CNA*.

## 4.6.2 Progetto database

Schema E/R

Schema Relazionale

UTENTI (USERINTRA, PASSINTRA) LOGIN (UTENTE, LINK, SITO, USERNAME, PASSWORD)

FK: UTENTE REFERENCES LOGIN



Figura 4.7: Sezione Applicazioni Internet.

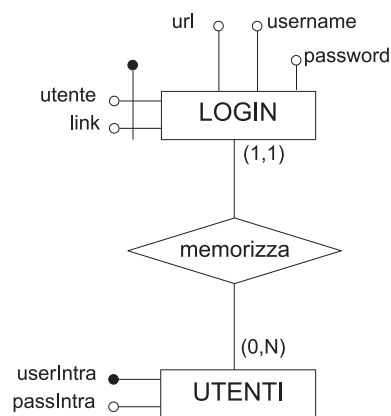


Figura 4.8: Schema E/R sezione Applicazioni.

# Capitolo 5

## Programma di gestione password

Nella sezione *Applicazioni Internet* si è data una breve descrizione del programma di gestione delle password denominato “Intranet Password Manager”. Di seguito si fornisce una descrizione del suo funzionamento, delle funzioni presenti e del modo con cui queste sono state implementate.

### 5.1 Funzionamento

Per utilizzare il programma “Intranet Password Manager” è necessario, ogni volta che si vuole accedere alla *Intranet CNA*, essere riconosciuti dal sistema eseguendo la procedura di autenticazione. Questa procedura, che permette lo sblocco del programma, consiste nell’inserimento all’interno dei campi di login delle credenziali di accesso alla Intranet. Dopo aver eseguito la procedura di autenticazione vengono attivati i due pulsanti “Effettua Login” e “Memorizza Login”. In figura 5.1 è mostrata l’interfaccia grafica del programma del programma.



Figura 5.1: Barra delle password.

**Memorizza Login:** questa funzione permette di memorizzare le credenziali di accesso a tutti i siti web che all’interno della pagina contengono dei campi di login. I passi da seguire per ottenere la memorizzazione delle password sono i seguenti:



1. collegarsi alla pagina web desiderata;
2. cliccare sul pulsante “Memorizza Login”;
3. inserire le credenziali di accesso e il nome del sito;
4. confermare l’inserimento/sovrascrittura.

**Effettua Login:** questa funzione permette di effettuare il login ai siti web che in precedenza sono stati memorizzati utilizzando la funzione “Memorizza Login”. I passi da seguire per ottenere l’autenticazione in una pagina di un sito web sono i seguenti:

1. collegarsi al sito web desiderato, preferibilmente attraverso la sezione “Applicazioni Internet”;
2. cliccare sul pulsante “Effettua Login”;
3. inviare i dati dal form della pagina web.

Un’immagine di esempio sull’utilizzo di queste funzioni sono mostrate nella figura 5.2. Naturalmente il programma gestisce anche gli eventuali errori commessi dall’utente, restituendo in output messaggi di avviso.

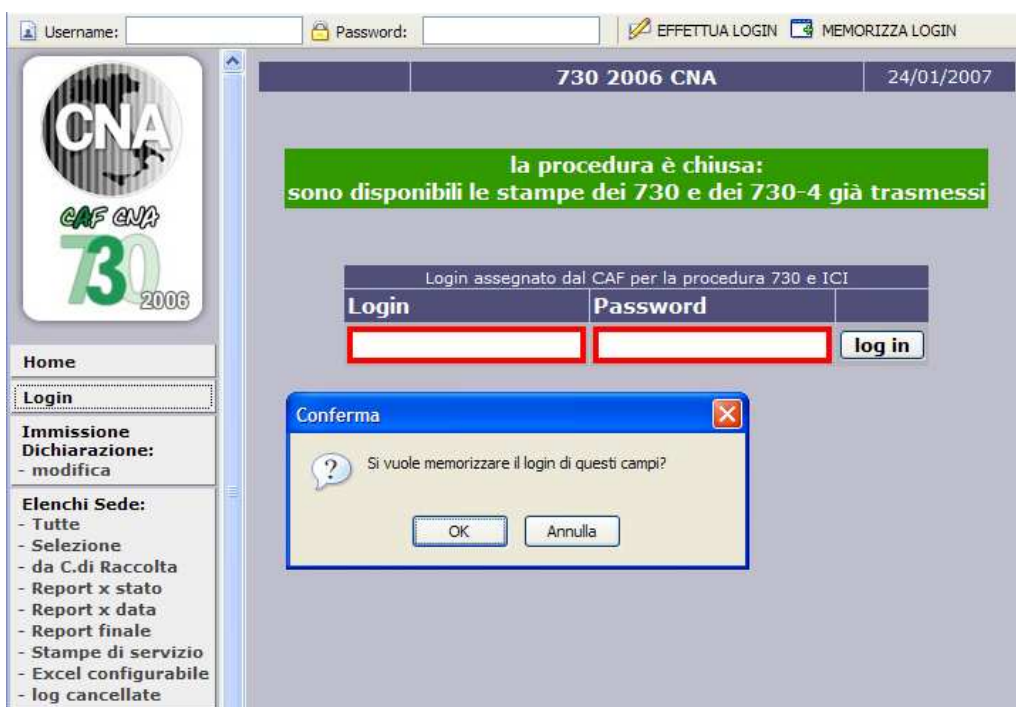


Figura 5.2: Esempio di uso del programma.

## 5.2 Linguaggi utilizzati

Il programma “Intranet Password Manager” è stato realizzato come “Extension” per il browser *Mozilla Firefox*. I linguaggi utilizzati per la realizzazione sono i seguenti:

- **XUL (XML User Interface Language)**: per descrivere l’interfaccia grafica dell’applicazione. Tutte le estensioni Firefox usano questo linguaggio[11];
- **RDF**: per specificare le caratteristiche funzionali e di installazione;
- **JavaScript**[12]: per implementare le funzioni e le richieste HTTP in AJAX;
- **DOM (Document Object Model)**[13]: usato all’interno del codice Javascript per gestire i file in formato XML;
- **Php**[14]: linguaggio server-side usato per le richieste dati al database MySQL[15].

## 5.3 Interfaccia grafica in XUL

Tutte le “Extension” del browser Firefox sono realizzate con una specifica XML denominata **XUL**, acronimo di *XML User Interface Language*. Il linguaggio XUL è utilizzato per creare le interfacce grafiche delle “Extension”, così come si fa con l’HTML per le pagine web. Le funzionalità di questi elementi sono implementate con il linguaggio JavaScript, per la precisione la versione 1.5, quindi la programmazione è analoga a quella delle pagine web. Nel listato 5.1 viene mostrato il file “interfaccia.xul” usato per la realizzazione dell’interfaccia del programma. Per approfondimenti sulla realizzazione di estensioni per Mozilla Firefox si rimanda al sito [16].

Listing 5.1: Interfaccia in linguaggio XUL

```
<?xml version="1.0"?>
<overlay id="passwordBarInterfaccia" xmlns="http://www.mozilla.org/keymaster/gatekeeper/there.is.only.xul">
  <script src="funzioni.js"/>
  <keyset id="mainKeyset">
    <key id="key_pbtoggle" keycode="VK_F1" modifiers="shift" command="cmd_toggleToolbar"/>
  </keyset>
  <commandset id="mainCommandSet">
    <command id="cmd_toggleToolbar" oncommand="goToggleToolbar('passwordbar','cmd_tooggleToolbar');"/>
  </commandset>
  <toolbarpalette id="BrowserToolbarPalette">
    <toolbaritem id="psbar">
```

```

<label value="Username: " image="img/username.jpg"/>
<textbox id="text_username"/>
<label value="Password: " image="img/password.jpg"/>
<textbox id="text_password"/>
<toolbarseparator/>
<toolbarbutton id="button_effettua" label="EFFETTUA LOGIN" image="img/effettua.jpg"
  oncommand="effettuaLogin();"/>
<toolbarbutton id="button_memorizza" label="MEMORIZZA LOGIN" image="img/memorizza.
  jpg" oncommand="memorizzaLogin();"/>
</toolbaritem>
</toolbarpalette>
<toolbox id="navigator-toolbox">
  <toolbar accesskey="R" hidden="false" chromeclass="toolbar" class="chromeclass -
    toolbar" customizable="false" id="passwordbar" mode="full" toolbarname="
    PasswordBar [shift+F1]" inherits="collapsed,hidden" persist="collapsed,hidden"
    defaultset="psbar"/>
</toolbox>
</overlay>

```

## 5.4 Metodi Javascript

Di seguito vengono descritti e presentati i listati dei metodi JavaScript implementati per realizzare le funzionalità del programma.

### 5.4.1 Leggi Pagina Corrente

Il metodo `leggiPaginaCorrente()` restituisce l'oggetto `document` della pagina che, nel momento dell'esecuzione, è visualizzata nel browser web. Questo metodo utilizza anche i metodi `getDocuments()` e `getContentWindow()` descritti nel listato 5.2.

Listing 5.2: `individuaLogin.js`

```

// restituisce l'oggetto document della pagina corrente
function leggiPaginaCorrente()
{
  var listaDocumenti = getDocuments(getContentWindow());
  return listaDocumenti[0];
}
// restituisce tutti gli oggetti document della pagina corrente
function getDocuments(frame)
{
  var documents = new Array();
  if(frame)
  {
    var frames = frame.frames;
    var framesLength = frames.length;
    // se il frame del documento esiste
    if(frame.document)
      documents.push(frame.document);
  }
}

```

```

// loop attraverso tutti i frame
for(var i = 0; i < framesLength; i++)
    documents = documents.concat(getDocuments(frames[i]));
}
return documents;
}
// ritorna il contenuto della finestra corrente
function getContentWindow()
{
    return window.top.getBrowser().browsers[window.top.getBrowser().mTabBox.selectedIndex].
        contentWindow;
}

```

## 5.4.2 Individua Login

Il metodo `individuaLogin()` richiede in ingresso una pagina web (ad esempio quella corrente) sulla quale effettua una scansione nella ricerca di campi di login. Se la pagina contiene dei campi di login allora questi vengono memorizzati nelle variabili `campoUsername` e `campoPassword`. Altrimenti se nella pagina non ci sono campi di login viene visualizzato un messaggio di errore.

Listing 5.3: `individuaLogin.js`

```

// variabili globali dove memorizzare i campi di login
var campoUsername = null;
var campoPassword = null;

// individua i campi login del sito
function individuaLogin(pagina)
{
    campoUsername = null;
    campoPassword = null;
    // il metodo getDocuments restituisce una lista di tutti gli oggetti document contenuti
    // nella pagina
    var listaDocumenti = getDocuments(pagina);
    var numeroDocumenti = listaDocumenti.length;
    var documento = null;
    var elemento = null;
    var listaElementi = null;
    var numeroElementi = null;

    // effettua una scansione di tutti gli oggetti documenti della pagina
    for(var i=0; i<numeroDocumenti && campoUsername==null; i++)
    {
        // oggetto document da analizzare
        documento = listaDocumenti[i];
        // si crea una lista contenente tutti gli elementi di tipo "input"
        listaElementi = documento.getElementsByTagName("input");
        numeroElementi = listaElementi.length;
        // scansione di tutti gli elementi di tipo "input"
        for(var j=0; j<numeroElementi; j++)

```

```

{
    // elemento "input" da analizzare"
    elemento = listaElementi[j];
    // se l'elemento ha come attributo "password"
    if(elemento.getAttribute("type").toLowerCase() == "password")
    {
        // trovato campo password
        campoPassword = elemento;
        // se l'elemento precedente è di tipo testo, allora è il campo username
        if (listaElementi[j-1].getAttribute("type").toLowerCase() == "text")
            campoUsername = listaElementi[j-1];
        // altrimenti sarà l'ultimo elemento con attributo "text" prima di quello "
        // password"
        else
        {
            for(var k=2; campoUsername == null && j >= k; k++)
                if (listaElementi[j-k].getAttribute("type").toLowerCase() == "text")
                    campoUsername = listaElementi[j-k];
        }
    }
}
// se non è stato trovato nessun campo username
if (campoUsername == null)
{
    alert("Il sito non contiene campi di login");
    return false;
}
// altrimenti evidenzia i campi di login
evidenziaLogin(true);
return true;
}

```

### 5.4.3 Effettua Login

Il metodo `effettuaLogin()` provvede a inserire le credenziali di accesso all'interno dei campi login della pagina corrente. Naturalmente l'azione dà esito positivo solo se le credenziali di accesso sono state in precedenza memorizzate, altrimenti sarà mostrato un messaggio di errore. I metodi `richiestaAJAX_effettua()` e `rispostaAJAX_effettua()` permettono di recuperare le informazioni relative alle credenziali di accesso dal database della Intranet.

Listing 5.4: `effettuaLogin.js`

```

// inserisce le credenziali di accesso nei campi di login
function effettuaLogin()
{
    // memorizza la pagina web che in quel momento è visualizzata nel browser
    var paginaCorrente = leggiPaginaCorrente();
    // legge URL della pagina corrente

```

```

var url = paginaCorrente.URL;
// è true se sono stati trovati i campi login con il metodo individualLogin
var presente = individualLogin();
// annulla l'operazione se non ci sono campi di login
if (presente == false)
    return;
// imposta i parametri e il file da passare al metodo AJAX
var parametri = "userIntra=" + userIntra + "&passIntra=" + passIntra + "&url=" + url;
var file = "http://intranetcnamo/passwordbar/effettua.php";
// richiesta con metodo AJAX
richiestaAJAX_effettua(file, parametri);
}

```

#### 5.4.4 Memorizza Login

Il metodo `memorizzaLogin()` permette di memorizzare le credenziali di accesso al sito della pagina corrente. Naturalmente questo è possibile solamente se nel sito sono presenti dei campi di login, altrimenti sarà mostrato un messaggio di errore. Se le credenziali di accesso a quel sito erano state già in precedenza memorizzate, il programma chiederà una eventuale sovrascrittura. I metodi `richiestaAJAX_memorizza()` e `rispostaAJAX_memorizza()` permettono di accedere al database della Intranet per salvare le credenziali di accesso.

Listing 5.5: `memorizzaLogin.js`

```

// memorizza la login del sito corrente
function memorizzaLogin()
{
    // si memorizza la pagina web visualizzata al momento nel browser
    var paginaCorrente = leggiPaginaCorrente();
    // individua i campi di login presenti nella pagina
    var presente = individualLogin(paginaCorrente);
    // annulla l'operazione se non ci sono campi di login
    if (presente == false)
        return;
    // richiesta di memorizzazione login
    var risposta = confirm("Si vuole memorizzare il login di questi campi?");
    // annulla l'operazine se la risposta è negativa
    if (risposta == false)
    {
        alert("L'operazione e' stata annullata");
        return;
    }
    // legge il valore del campo username
    var username = campoUsername.value;
    username = prompt("Username:", username);
    if (username == null || username == "")
    {
        if (username == null)
            alert("La procedura di memorizzazione login e' stata annullata");
        else

```

```
    alert("ATTENZIONE!\nNon e' stato inserito nessun username.\nRipetere la procedura
        cliccando su \"MEMORIZZA LOGIN\");
    return false;
}
// legge il valore del campo password
var password = campoPassword.value;
password = prompt("Password:", password);
if (password == null || password == "")
{
    if (password == null)
        alert("La procedura di memorizzazione login e' stata annullata");
    else
        alert("ATTENZIONE!\nNon e' stata inserita nessuna password.\nRipetere la procedura
            cliccando su \"MEMORIZZA LOGIN\");
    return false;
}
// legge l'URL della pagina corrente
var url = paginaCorrente.URL;
// legge il nome del sito
var nomeSito = paginaCorrente.title;
do
{
    nomeSito = prompt("Nome sito:", nomeSito);
    if (nomeSito == "")
        alert("ATTENZIONE! Inserire il nome del sito.");
}
while (nomeSito == "");
// toglie l'evidenziazione dei campi di login
evidenzialLogin(false);
// imposta i parametri e il file da passare al metodo AJAX
var parametri = "userIntra=" + userIntra + "&passIntra=" + passIntra + "&nomesito=" +
    nomeSito + "&url=" + url + "&username=" + username + "&password=" + password;
var file = "http://intranetctnamo/passwordbar/salva.php";
// richiesta AJAX
richiestaAJAX_memorizza(file, parametri, "&sovrascrivi=no");
}
```

### 5.4.5 Richiesta dati al database con AJAX

Per la richiesta dati al database della intranet è stata implementata una funzione che utilizza la tecnica AJAX. Il modo con cui questa richiesta viene gestita è descritta dai seguenti passi:

1. JavaScript chiama la funzione di richiesta AJAX specificando il nome del file PHP ed i relativi parametri;
2. la funzione di richiesta AJAX, attraverso l'oggetto XMLHttpRequest() effettua una richiesta HTTP al server secondo i parametri specificati dal chiamante e si pone in attesa di ricevere un file in formato XML;
3. il server, eseguendo il codice PHP, elabora la richiesta dati con una query al database;
4. il codice PHP ricevuta la risposta dal server del database restituisce in output un file in formato XML;
5. il file XML viene letto dal metodo di risposta AJAX;
6. Javascript mediante il metodo responseXML legge il file XML che può essere gestito con il DOM.

La figura 5.3 mostra in modo schematico il procedimento appena descritto.

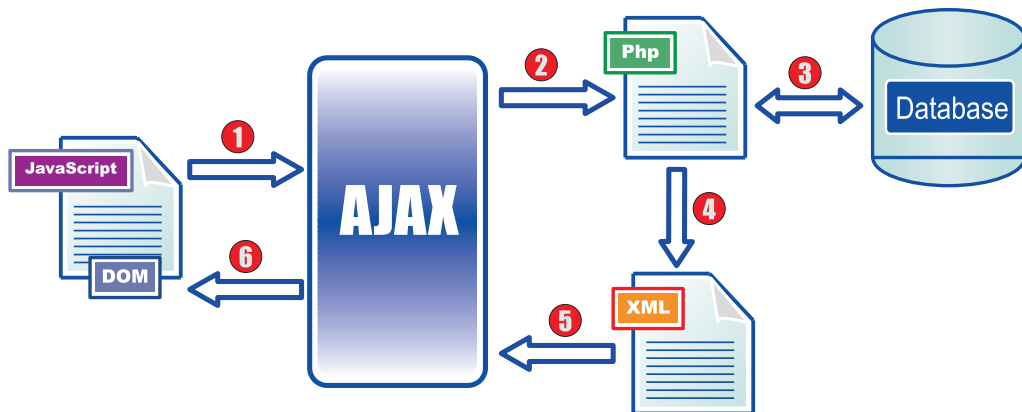


Figura 5.3: Richiesta dati al database con AJAX.

### 5.4.6 Richiesta e risposta AJAX

Il file  `richiestAJAX.js` , presentato nel listato 5.6, contiene sia la funzione di richiesta che quella di risposta AJAX. In questo caso la richiesta HTTP, gestita dal metodo  `XMLHttpRequest()` , è di tipo asincrono.



Listing 5.6: richiestaAJAX.js

```

// richiesta al server con AJAX
function richiestaAJAX(file, parametri)
{
    var XHR = new XMLHttpRequest();
    // funzione di risposta alla richiesta AJAX
    XHR.onreadystatechange = function()
    {
        if (XHR.readyState == 4 && XHR.status == 200)
        {
            // memorizza il testo di risposta del server
            var testoRisposta = XHR.responseText;
            // memorizza l'XML di risposta del server
            var xmlRisposta = XHR.responseXML;
        }
    };
    XHR.overrideMimeType("text/xml");
    XHR.open("POST", file, true);
    XHR.setRequestHeader("Content-Type", "application/x-www-form-urlencoded");
    XHR.send(parametri);
}

```

### 5.4.7 File Php

Il file Php (listato 5.7) viene utilizzato solamente per fare richiesta di lettura o scrittura al database della Intranet. Il documento ottenuto in uscita si presenta in formato XML, quindi gestibile dalla funzione AJAX.

Listing 5.7: effettua.php

```

<?php
// memorizza parametri passati con metodo POST nelle variabili
$userIntra = $_POST['userIntra'];
$passIntra = $_POST['passIntra'];
$url = $_POST['url'];

// query che selezione username e password di un sito
$query = "
    SELECT username, password
    FROM utenti, login
    WHERE userIntra = utente
    AND userIntra LIKE '$userIntra'
    AND passIntra LIKE '$passIntra'
    AND sito LIKE '$url'
";
// risposta in formato XML
echo "<?xml version=\"1.0\" ?>";
echo "<root>";
$result = mysql_query($query, $db);
if ($row = mysql_fetch_array($result))
{

```

```
    echo "<presente>si</presente>";
    echo "<username>$row[username]</username>";
    echo "<password>$row[password]</password>";
}
else
    echo "<presente>no</presente>";
echo "</root>";
?>
```

# Capitolo 6

## Conclusioni

Il progetto del tirocinio ha riguardato principalmente la realizzazione della Intranet aziendale CNA Modena.

L'avvento della Intranet nell'azienda è stato accolto interesse da parte di tutti i dipendenti. Già dai primi giorni di sperimentazione si è potuto constatare la notevole utilità di questo nuovo strumento, in particolare per le sezioni ad accesso pubblico come la sezione "Rubrica" e "Circolari". Con la sezione "Circolari", ad esempio, si è creata una sorta di bacheca virtuale nella quale tutti i dipendenti hanno la possibilità di visionare le ultime comunicazioni della direzione oppure di recuperare dall'archivio quelle precedenti. Questa sezione offre un miglioramento rispetto al passato dove il mezzo divulgativo principalmente utilizzato era costituito invece dalla posta elettronica.

Uno degli obiettivi del progetto era quello di realizzare un sistema di single sign-on, non solo per le applicazioni della Intranet, ma anche per le applicazioni Internet. Come descritto nel capitolo 5 la soluzione implementata consiste nel simultaneo utilizzo della sezione "Applicazioni Internet" e del programma di gestione password. È importante sottolineare che, al fine di giungere ad una soluzione, sono stati analizzati vari applicativi software, ma nessuno di questi possedeva le caratteristiche richieste. Si è deciso perciò di realizzare il programma "Intranet Password Manager" che, oltre a risolvere il problema del recupero password, si integra perfettamente con la Intranet aziendale.

Il programma "interagisce" con il server del database grazie all'uso della tecnica AJAX. Lo studio dedicato a tale tecnica, necessario per lo sviluppo del programma, mi ha permesso di apprezzarne le potenzialità offerte. Per questo motivo ho deciso di implementare il medesimo approccio anche nelle applicazioni web della Intranet, così da renderle più dinamiche.

Nel capitolo 3 si è visto come è possibile implementare AJAX per rendere gli applicativi web più dinamici ed interattivi, ma si è anche precisato che un uso scorretto di questa

tecnica può rappresentare una facile via d'entrata per utenti maliziosi. L'errore comune è quello di fare affidamento solo sul codice client, è importante invece effettuare sempre il controllo dei dati di ingresso sul lato server. Fortunatamente questi problemi riguardano soprattutto gli sviluppatori di applicazioni web per Internet, poichè in una Intranet le possibilità che qualcuno faccia di proposito dei danni saranno sicuramente pochissime se non nulle. Nonostante ciò, nello sviluppo delle applicazioni web dell'Intranet, è stata adottata una rigorosa cautela.

# Bibliografia

- [1] Alessandro Lucchini, Intranet: teoria e pratica, 2004 (Apogeo)
- [2] Omar Tomasoni e Luciano Brancati, Applicazioni web, 2003 (Tecniche nuove)
- [3] AJAX: A New Approach to Web Applications  
<http://www.adaptivepath.com/publications/essays/archives/000385.php>
- [4] Ryan Asleson e Nathaniel T. Shutta, AJAX la grande guida, 2006 (Mondadori)
- [5] Brett D. McLaughlin, Head Rush Ajax, 2006 (O'Reilly)
- [6] Tecniche per utilizzare AJAX:  
<http://antirez.com/articoli/spaghettaajax.html>
- [7] Backbase - Framework AJAX:  
<http://www.backbase.com>
- [8] Sicurezza in AJAX:  
<http://javascript.html.it/articoli/leggi/1927/ajax-basi-di-sicurezza>
- [9] CNA Servizi Modena:  
<http://www.mo.cna.it>
- [10] CMS per lo sviluppo di siti web:  
<http://www.webprofessional.it>
- [11] Guida di riferimento del linguaggio XUL:  
<http://www.xulplanet.com>
- [12] David Flanagan, JavaScript la guida II edizione, 2002 (O'Reilly)
- [13] W3C - World Wide Web Consortium:  
<http://www.w3.org>

[14] Guida del linguaggio Php:

<http://www.php.net>

[15] Database MySQL:

<http://www.mysql.com>

[16] Realizzare estensioni per Mozilla Firefox:

[http://developer.mozilla.org/en/docs/Building\\_an\\_Extension](http://developer.mozilla.org/en/docs/Building_an_Extension)

[17] Wikipedia:

<http://www.wikipedia.org>