

4 Scelte tecniche

Esistono oggi professionalità altamente specializzate sia nella realizzazione grafica di *siti* Web che nel fornire servizi di “*server provider*”, si tratta tuttavia di prestazioni d’opera piuttosto costose sia per quanto riguarda la messa a punto di una prima versione del prodotto completo, che per i successivi inevitabili ammodernamenti. D'altro canto gli stessi professionisti hanno difficoltà a mantenersi aggiornati, per il rapido evolversi delle tecnologie che interessano questi settori.

Ciò che l’IVSLA si proponeva di realizzare, nel momento in cui ha deciso di risistemare la propria banca dati, era una struttura soggetta, per “costituzione”, a continue revisioni. Doveva infatti essere in grado di far fronte alle necessità di rinnovo dei contenuti (in termini sia di arricchimento dei settori presenti, sia di aggiunta di nuove categorie) e, conseguentemente, di adeguamento delle scelte espositive. Per realizzare un servizio efficiente bisognava quindi essere capaci di rispondere rapidamente al progressivo maturare di nuove esigenze organizzative, ma anche di accogliere le proposte innovative delle ditte produttrici di applicativi (software).

L'IVSLA ha perciò preferito investire risorse nel conseguimento di competenze autonome, che consentissero di creare e gestire il *sito* in modo indipendente, riservandosi di consultare specialisti esterni solo all’occorrenza. Ciò prospettava un avvio meno rapido del progetto, ma una libertà di gestione che in seguito avrebbe garantito elasticità e rapidità nell'aggiornamento, necessarie ad assicurare una buona stabilità al *sito*.

Anche la scelta di rivolgersi a prodotti *Open Source* è stata guidata, più che da motivazioni economiche (comunque tenute nella dovuta considerazione), dalle opportunità di autonomia e di flessibilità, nonché dalle garanzie di assistenza in *rete*, che essi sono in grado di offrire. L'ultima risoluzione adottata, nell'ottica di assicurare la massima indipendenza nell'amministrazione del *sito*, è stata uscire dalla rete GARR¹⁷, rinunciando alla connessione con la *rete* universitaria. Questa, infatti, imponeva vincoli alla gestione e manteneva inevitabilmente il *sito* dell'IVSLA in uno stato di subordinazione, poiché solo ad un amministratore autorizzato dall'ateneo è consentito l'accesso alle procedure di intervento ogni qual volta si verificano disfunzioni e disguidi di qualunque natura. Inoltre il GARR era in procinto di revocare la gratuità della connessione: veniva meno, così, l’unico consistente vantaggio che quell’opzione poteva offrire.

Una volta messo in atto e collaudato con successo l'apparato predisposto per la Banca Dati «Ambiente Venezia», l’IVSLA ha preso in carico anche l'amministrazione del *sito* Web associato al Progetto «TIDE», certo di poterne assicurare l'affidabilità dovuta.

¹⁷ Il [GARR](http://www.laterza.it/internet/leggi/internet2000/online/testo/29_testo.htm), (Gruppo Armonizzazione delle Reti di Ricerca), è un ente che fa capo al Ministero dell'Università e della Ricerca Scientifica e Tecnologica (MIUR), e che ha il compito di gestire e sviluppare la rete scientifica italiana. http://www.laterza.it/internet/leggi/internet2000/online/testo/29_testo.htm

4.1 Prodotti *Open Source*

Con la definizione *Open Source* viene individuata una categoria di prodotti software rispondenti a determinati requisiti, che non si limitano alla possibilità di accesso al codice sorgente. I termini di utilizzo e distribuzione di tali prodotti devono infatti rispettare una serie di criteri specifici, chiaramente fissati in un “manifesto” (si veda Appendice 2).

Si tratta di applicazioni liberamente elaborate da esperti ed appassionati, i quali non agiscono a fini commerciali, ma in quell’ottica di condivisione delle competenze e delle idee, che ha alimentato l’evoluzione di Internet e ne ha decretato il successo. Gli stessi principi ispiratori che animano la ricerca pura, dedita al progresso in quanto patrimonio della collettività, si protraggono nella filosofia di sviluppo di questi prodotti, resi disponibili senza vincoli (e generalmente scaricabili gratuitamente da Internet) sia nella forma compilata sia in quella comprendente il “sorgente”. Chiunque può apportare modifiche ed integrazioni, purché rispetti quegli stessi criteri di trasparenza e condivisione che gli hanno consentito il libero accesso al programma originale (<http://opensource.org/docs/definition.PHP>).

I software *Open Source* sono del tutto gratuiti nei casi in cui non vengano impiegati a scopo di lucro e non vengano inseriti in pacchetti complessi, che si intendono rivendere. Viceversa è sufficiente acquistare la licenza per poterne disporre pienamente.

La scelta di rivolgersi a prodotti di questo tipo è non solo economicamente vantaggiosa, ma presenta ulteriori risvolti pratici. Innanzi tutto il fatto di poter accedere ai codici sorgente consente di “personalizzare” le funzionalità presenti, adattandole alle proprie specifiche esigenze. Inoltre coloro i quali non sono particolarmente pratici con l’informatica, o necessitano di ausili particolari, possono avvalersi della consulenza offerta da una vasta comunità di programmatori che appoggiano il “progetto di diffusione del software libero” e forniscono assistenza gratuita attraverso gli strumenti telematici (quali *forum*, *newsgroup*, *newsletter*, *chat*, *mailing list* o altri dispositivi analoghi).

Questo modo di operare incentiva continuamente ad affinare e migliorare i programmi, cui ciascuno può contribuire attivamente apportando le proprie implementazioni (*patch*), scovando le lacune (*bug*), sviluppando varianti e diffondendo poi il risultato delle proprie elaborazioni. Tutto questo non necessariamente in vista di un guadagno economico, ma anche per la semplice soddisfazione di aver prodotto qualcosa di utile a molti o per il “vezzo” di esibire la propria bravura!

4.2 Configurazione della piattaforma *server*

La messa *online* della versione aggiornata della Banca Dati «Ambiente Venezia» e, successivamente, della Banca Dati «TIDE», hanno reso necessario l’impiego di un sistema capace di ovviare ai problemi di gestione derivanti sia dal sovraccarico (che riduce l’efficienza di collegamento e di “navigazione”), sia da atti di “pirateria informatica” (*spamming*, *denial of service*, *virus*), inconvenienti dai quali i *siti* devono essere adeguatamente tutelati.

La piattaforma predisposta per ospitare le banche dati è costituita da due unità HP NetServer, con 8 GB di Hard Disk ciascuna, in grado di supportare un’implementazione della memoria fino a 13 GB. Le macchine sono connesse *online* 24 ore su 24 attraverso quattro linee ADSL, provenienti da 3 diversi gestori (1 *linea* «Telecom» da 1,28 MB; 2 *linee* «Edisontel» da 640 KB; 1 *linea* «Punto d’Incontro» da 1,28 MB), a ciascuna delle quali è

associato un differente numero IP. La scelta di rivolgersi a più gestori è stata operata per garantire una maggiore stabilità perchè qualora uno di essi presentasse problemi interni alla propria struttura, la connettività al *sito* rimarrebbe assicurata attraverso gli altri.

Ciascuna macchina *server* ha installato un pacchetto software, prodotto della combinazione di 4 componenti:

- Sistema Operativo: **Linux** Debian GNU;
- Web *server*: **APACHE**;
- Relational DataBase Management System: **MySQL**;
- Hypertext Preprocessor: **PHP**.

Si tratta di due apparati gemelli, uno dei quali è operativo, mentre l'altro funge da "scorta", pronto ad entrare automaticamente in funzione in caso di guasto del suo complementare. Ciascuna macchina esegue periodicamente un *backup* automatico di tutti i dati in memoria. Entrambi questi "accorgimenti" sono diretti a garantire affidabilità e continuità al *sito*.

4.2.1 Linux

Linux¹⁸ è un sistema operativo UNIX disponibile per diverse piattaforme; come UNIX è costituito da due parti: i) la *shell*, che interpreta i comandi dell'utente e ne richiede l'esecuzione al *kernel*; ii) il *kernel*, che si prende carico della gestione delle risorse a livello operativo più basso. Presenta le caratteristiche di essere: i) *multitasking*, ovvero permette di eseguire più processi simultaneamente; ii) *multiuser*, ovvero più utenti possono collegarsi ed utilizzare il sistema allo stesso tempo (<http://www.polarnet.cnr.it/CORSI/LAMP/lamp.pdf>).

4.2.2 APACHE

Il Web *server* [APACHE](#) svolge la funzione di fornire l'accesso ai file contenuti nelle banche dati (siano essi immagini, tabelle di dati, file di testo, pagine HTML, ecc.) per i *client* che, servendosi ciascuno del proprio *browser* (*Netscape Navigator*, *Microsoft Internet Explorer*, *Opera*, *Mozilla*, per citare i più diffusi), ne inoltrano richiesta.

Esso opera attraverso la codifica di normali file ASCII contenenti il testo di un linguaggio di programmazione ipertestuale che permette di formattare testo, grafica e link all'interno di schermate visualizzabili dai *browser*.

Il protocollo che gestisce il trasferimento di questi file fra *client* e *server* Web è l'HTTP.

L'utente che "naviga" in *rete* non fa altro che trasferire file dall'hard disk del *server* al proprio computer, dove il *browser* decodifica le pagine HTML ricevute e le compone, richiedendo di volta in volta le immagini e gli altri file necessari per completarle. Il software Web *server* ha il compito di "ricevere" le richieste che gli giungono dai *browser* e di predisporre le "risposte", stimolando, se necessario, l'intervento del PHP affinché, inviato al RDBMS il comando di rintracciare all'interno dei datatabase i dati necessari a comporre, le restituisca poi adeguatamente strutturate (http://www.risorse.net/apache/il_web.asp).

¹⁸ Elaborato nel 1991 da Linus Benedict Torvalds, studente dell'università di Helsinki, in base alle regole della GNU Free Software Foundation, che ne hanno favorito la rapida diffusione e l'ulteriore sviluppo grazie alla collaborazione di numerosi appassionati informatici di tutto il mondo.

APACHE è un Web *server* a struttura modulare sviluppato dalla *APACHE Software Foundation* a partire da una parte di codice preesistente (appartenente alla NCSA - National Center for Supercomputing Application), al quale sono state apportate una serie di modifiche (*patch*) per migliorarne l'affidabilità e le caratteristiche; da qui il nome: "A PAtCHy sErver". Lo scopo era creare un Web server stabile, affidabile e veloce per piattaforme Unix.

Uscito in prima versione nel 1995 è ad oggi il più diffuso grazie alla grande quantità di moduli che supporta e alla facilità di configurazione.

4.2.3 MySQL

MySQL è un sistema di gestione di database relazionali che soddisfa i seguenti requisiti: i) è multiutente - consente per *default* di soddisfare fino a 101 connessioni simultanee, ma se ne possono aggiungere a piacimento; ii) assicura un'elevata sicurezza - è possibile assegnare differenti livelli di accesso al database rilasciando alle macchine remote diversi gradi di autorizzazione; iii) è flessibile - permette di modificare l'applicazione intervenendo sul codice sorgente e può essere eseguito su quasi tutte le piattaforme. Consente inoltre applicazioni di *server* a processi (*tread*) multipli: quando viene stabilita una connessione, il programma all'interno del *server* crea un *tread* che gestisce le richieste del *client* garantendo un'elevata velocità di esecuzione in quanto ad ogni utente connesso il sistema attribuisce un *tread* dedicato. Disponendo di numerose interfacce di programmazione (*Application Programming Interface* – API), può essere utilizzato praticamente in qualunque ambiente e può operare su supporti differenti, sia all'interno di architetture del tipo *client/server* sia come database autonomo.

MySQL risulta inoltre conforme agli standard emanati dall'*American National standard Institute* (ANSI).

Per la programmazione a livello di *server* è possibile impiegare diverse logiche, la cui efficienza è funzione delle mansioni che il sistema è destinato a svolgere. La parte più consistente dell'elaborazione può infatti essere assegnata: i) all'applicazione installata sul desktop dell'utente; ii) al server di database; iii) ad un livello (o gruppo di livelli) intermedio. Ciascuna di queste soluzioni risulterà più adeguata a seconda che si preveda un traffico più o meno intenso sul *sito* e a seconda delle dimensioni del database: se il *sito* è di proporzioni limitate e gestisce una quantità moderata di dati, il *server* non è soggetto a problemi di sovraccarico e perciò è in grado di gestire autonomamente la logica di programmazione. Viceversa, se il *server* è sottoposto ad un forte traffico e il servizio che espleta presuppone la necessità di effettuare numerosi calcoli è consigliato trasferire le manipolazioni sul *client*. Nel caso che molte macchine *client* debbano poter accedere simultaneamente ad un database tramite la stessa applicazione si può introdurre un livello intermedio atto a ricevere le richieste degli utilizzatori, gestire la logica di programmazione, accedere al database e restituire le risposte ai *client*. Quest'ultimo tipo di struttura risulta particolarmente adatta per le *Intranet* di grandi aziende.

Nel caso dell'IVSLA tutta la logica di elaborazione rimane in carico ai *server*. Le procedure generali di un'applicazione su Web supportata da una logica di questo tipo sono riassunte nei seguenti punti (Maslakowski, 2000):

- 1) un utente digita l'URL di una pagina del *sito* Web nel proprio *browser*;
- 2) il *browser* trova il *server* e richiede la pagina Web;

- 3) il *server* Web risponde alla richiesta e invia la pagina:
- 4) il *browser* riceve la pagina.

L'utente naviga nel *sito* ripetendo i punti da 1) a 4) finché non arriva ad una pagina che richiede un certo tipo di accesso ad un database (un *form* o una pagina generata in modo dinamico). A questo punto:

- 5) viene eseguito un programma o uno script, ottenendo i dati richiesti dal *server* di database;
- 6) la pagina viene composta e restituita all'utente.

4.2.4 PHP

[PHP](#) è l'acronimo ricorsivo di Hypertext Preprocessor. Si tratta di un linguaggio di programmazione adatto allo *scripting* dal lato *server* ed è solitamente impiegato nel contesto di una pagina HTML. Viene definito di tipo *HTML-embedded*, in quanto i comandi vengono inseriti nella sintassi delle pagine HTML e per essere eseguito non richiede apporti esterni.

Uno script PHP non richiede una complessa installazione né configurazioni particolarmente restrittive del Web *server*. Al contrario di una pagina HTML, non è inviato direttamente al *client*, ma è interpretato ed eseguito sul *server* prima che la pagina venga inviata al browser: il *client*, quindi, vedrà solo il risultato, cioè l'output del programma PHP. Una volta istruito il Web *server*, ogni *script* (o meglio, ogni pagina con all'interno il codice dello *script*) potrà essere eseguito in ogni directory esso si trovi (Vaccari, 2004).

Il codice di uno script PHP è in grado di svolgere numerose funzioni, come:

- verificare quale tipo di *browser* l'utente utilizza e generare il codice HTML opportuno;
- accedere ad informazioni, contenute in database, estraendole con *query* anche molto complesse;
- creare pagine dai contenuti dinamici;
- leggere e scrivere file sul disco;
- raccogliere dati da un *form* ed eseguire controlli di validazione;
- inviare e ricevere *cookie*;
- eseguire operazioni su server remoti;
- operare controlli di sicurezza e fornire solo ad alcuni utenti l'accesso a determinate risorse;

ed è in grado di compiere le proprie applicazioni indipendentemente dalla piattaforma hardware e software che le ospita (<http://www.polarnet.cnr.it/CORSI/LAMP/lamp.pdf>).

PHP è di norma impiegato in un contesto Web e necessita di un Web *server*, come APACHE. L'associazione PHP/APACHE è distribuita con l'installazione di *default* della maggior parte dei sistemi operativi UNIX liberi e commerciali (Vaccari, 2004).

PHP può integrarsi con APACHE in tre modalità: i) può essere inserito direttamente nell'eseguibile del *server*; ii) può essere installato come *Dynamic Shared Object* (DSO), ovvero come libreria in grado di dotare APACHE della capacità di interpretare ed eseguire codici PHP; iii) può esistere come vero e proprio programma eseguibile separato a cui è deputata la gestione del codice tramite le specifiche CGI (acronimo di *Common Gateway*

Interface, costituisce l'insieme di regole secondo le quali i programmi lato *server* inviano e ricevono informazioni). Nel nostro caso è stato installato come DSO.

PHP è stato progettato per integrarsi con la maggior parte dei DBMS presenti sul mercato (tra cui MySQL) e dispone di funzioni specifiche per ognuno di essi.

Le capacità di PHP non si limitano ad un output in HTML: è possibile infatti non solo generare facilmente qualsiasi testo, come XHTML, e qualsiasi altro file XML, ma anche immagini, files PDF e perfino filmati Flash *al volo*. PHP può autogenerare questi file, e salvarli nel file system, o eseguire un *printing* esterno o creare *cache* dal lato *server* per contenuti dinamici.

4.3 Configurazione delle postazioni

Il personale che si occupa a tempo pieno delle banche dati IVSLA opera su postazioni dotate di software commerciali comunemente diffusi:

- ❖ Sistema operativo → Microsoft Windows 2000 Professional
- ❖ Elaborazione testi → Microsoft Word 2000
- ❖ Elaborazione tabelle e fogli di calcolo → Microsoft Excel 2000
- ❖ Browser → Microsoft Internet Explorer
- ❖ Gestione database → Microsoft Access 2000
- ❖ Web editor → Macromedia Dreamweaver;
- ❖ GIS → Intergraph Geomedia Professional;
- ❖ Elaborazione immagini → Macromedia Fireworks, Macromedia FreeHand e IrfanView;
- ❖ Realizzazione Animazioni → Macromedia Flash;
- ❖ Riconoscimento ottico dei caratteri → OmniPage Pro;
- ❖ Compressione e pubblicazione Web di testo e cartografia → Adobe Acrobat.

La scelta di utilizzare programmi commerciali è stata condizionata dalla necessità di doversi relazionare con interlocutori esterni, con i quali è strettamente indispensabile poter “comunicare”. Questa esigenza si manifesta sia in fase di recupero della documentazione da trattare, che è necessario poter “leggere” ed elaborare agevolmente, sia nella produzione dei materiali da distribuire attraverso le banche dati, che devono presentare formati di uso ampiamente diffuso. Era inoltre indispensabile tutelarsi da problemi di conflitti ed incompatibilità tra software.

Si stanno comunque sperimentando soluzioni alternative, della famiglia degli *Open Souce*, come l'impiego di *Open Office*, in sostituzione di *Microsoft Office*, e di *Grass*, in sostituzione di *Geomedia Professional*. Inoltre vengono eseguite prove di collaudo utilizzando *browser* differenti (*Mozilla*, *Netscape*, *Opera*) da quello per il quale è stato ottimizzato il *sito* (*Explorer*), al fine di verificare la congruenza tra i criteri di navigazione e visualizzazione adottati da ognuno e poter così rendere il *sito* universalmente compatibile. A tale proposito facciamo rilevare che la banca dati «TIDE» è stata direttamente realizzata in conformità ai criteri suggeriti dal [World Wide Web Consortium](http://www.w3.org/) (W3C). Gli standard che il W3C propone costituiscono linee guida di riferimento a livello internazionale per gli operatori del Web che vogliono garantire ai propri elaborati il massimo grado di congruenza con la

svariata gamma di applicazioni evolutesi nell'ambito del WWW. La banca dati «TIDE» è il prodotto più recente, inoltre è nata in un contesto internazionale, pertanto era giocoforza che fosse immediatamente concepita come rispondente a tali requisiti. Ad ogni modo è tra i propositi a breve termine di uniformare agli standard anche le altre sezioni del *sito* IVSLA.