

UNIVERSITÀ DEGLI STUDI DI PADOVA
DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE
CORSO DI LAUREA TRIENNALE IN INGEGNERIA INFORMATICA

Progettazione ed implementazione
di un Web server a basso costo
per una biblioteca digitale
personalizzata

Laureando: Federico Lancerin

Relatore: Prof. Giorgio Maria Di Nunzio

24 Settembre 2012

Anno Accademico 2011/2012

Indice

- 1 Introduzione
- 2 Assemblaggio e setup del server
 - 2.1 Scelta componenti
 - 2.2 Assemblaggio
 - 2.3 Installazione Sistema Operativo
- 3 Creazione del DataBase
 - 3.1 Analisi dei requisiti
 - 3.2 Progettazione concettuale
 - 3.3 Progettazione logica
 - 3.4 Codice SQL
 - 3.5 Software utilizzato
- 4 Setup del web server
 - 4.1 Progettazione e realizzazione dell'interfaccia grafica per la biblioteca digitale
 - 4.2 Collegamento web server – database
 - 4.3 Registrazione dominio
- 5 Conclusioni

Capitolo 1

Introduzione

Una biblioteca digitale è una biblioteca in cui una collezione di materiali (che possono essere testi ma anche audio e video) è mantenuta in formato digitale. La biblioteca ha il compito di gestire e preservare nel tempo i suoi contenuti, mettendo a disposizione dei suoi utenti alcune funzionalità prestabilite sui materiali mantenuti.

Generalmente queste funzionalità sono accessibili attraverso computer, che si possono trovare sia nello stesso luogo della biblioteca sia a distanza, una biblioteca digitale è consultabile da remoto attraverso la rete.

L'obiettivo è l'installazione di un server Web e la progettazione di una biblioteca digitale che offra un servizio di consultazione online degli indici dei titoli presenti nella biblioteca fisica di casa. La biblioteca consiste di centinaia di volumi dislocati in un numero consistente di mensole, scaffali, armadi. Il servizio deve permettere ad un utente di fare delle ricerche per venire a conoscenza della presenza o meno di alcuni libri, nello specifico si richiede di poter effettuare ricerche per titolo, per nome o per cognome autore. Il servizio deve essere fornito da un server casalingo da acquistare ed assemblare che dovrà rispettare delle caratteristiche quali:

- basso consumo energetico
- minimo rumore acustico
- rientrare in un budget limitato
- ridondanza dei dati
- possibilità di espansione / personalizzazione
- possibile aggiunta di servizi in futuro

Dopo un'analisi del problema la soluzione che si propone è quella di assemblare un computer scegliendo i componenti offerti dal mercato che più rientrano nell'ottica di bassi costi, bassi consumi e possibilità di espansione futura. Una volta assemblato il computer si è scelto di utilizzare solamente software libero per l'ovvia convenienza in termini di costi e personalizzazione.

Capitolo 2

Assemblaggio e setup del server

2.1 Scelta componenti

La ricerca dei componenti è stata fatta online considerando review e comparazioni di questi noti siti specializzati in hardware per PC assemblati a cui seguono le conclusioni a cui hanno portato :

- <http://www.silentpcreview.com>
- <http://www.notebookcheck.net/>
- <http://forum.xbmc.org>
- <http://www.techradar.com>
- <http://www.anandtech.com>

La configurazione desiderata deve quindi avere le seguenti caratteristiche :

- **Cpu e Gpu integrati in un unico microchip.**
In particolare ci si riferisce alle ultime APU (Accelerated Processing Units) che sono veri e propri SoC (System On a Chip) ovvero sistemi che integrano processore centrale, processore grafico e vari controller in un unico pezzo di silicio. Essendo centralizzata l'unità elaborativa la sorgente di calore da dissipare è solamente una e scegliendo un dissipatore adeguato è possibile avere una dissipazione passiva ovvero che non prevede l'uso di ventole che a fronte di un minimo consumo di corrente sono la maggior fonte di rumore.
- **Alimentatore ad alta efficienza e ventola silenziosa**
Una delle caratteristiche principali del sistema è che deve assorbire la minor potenza possibile dalla rete elettrica, questo nei sistemi a basso consumo è raggiungibile solamente con un alimentatore adeguato, che mantiene un buon rendimento (ovvero un buon rapporto potenza elettrica utile / potenza elettrica assorbita).
- **2 Hard Disk Drive da 2.5" identici in RAID1**
Un requisito del nostro sistema è la ridondanza dei dati, ovvero la tolleranza ai guasti meccanici che affliggono gli Hard Disk Drive (HDD). Dato che non è possibile prevedere quando un HDD si guasterà ogni produttore dichiara un valore in ore chiamato MTBF, che sta per Mean Time Between Failures ed indica il tempo medio di uso di un HDD prima che questo si rompa e causi la perdita dei dati che conteneva. Niente vieta all'HDD di rompersi molto prima del raggiungimento del MTBF, così una delle pratiche utilizzate nella maggior parte dei server è quella di avere più HDD che sono la copia identica l'uno dell'altro in modo che se anche uno si guasta, i dati rimangono reperibili dagli altri o dall'altro. Sostituendo poi il drive corrotto è possibile tornare ad una situazione in cui i dati sono al riparo da eventuali guasti meccanici degli HDD. Questo sistema è chiamato RAID1 (che sta per Redundant Array of Independant Disks) ed è stato il primo modo di utilizzare la ridondanza per diminuire il rischio di perdita di dati. La scelta dei dischi da 2.5" invece di quelli standard da 3.5" è stata fatta per una questione di consumi, che con i dischi di diametro minore sono più che dimezzati.

Alcuni pezzi sono stati riutilizzati da computer assemblati in precedenza, per i rimanenti necessari sono state compiute ricerche sui vari noti negozi online che seguono :

- <http://www.e-key.it/>
- <http://www.prokoo.com/>
- <http://www.planethardware.it/>
- <http://www.hardware-planet.it/>
- <http://www.neths.it/>
- <http://www.drako.it>
- <http://www.amazon.it/>

La lista definitiva dei pezzi utilizzati con link alle schede prodotto è la seguente:

Scheda madre	Asus E35M1-M incl. AMD E-350 - Scheda Prodotto ¹
Memoria	G.Skill DDR3-1333 Kit (F3-10600CL8D-4GBHK) 4GB - Scheda Prodotto ²
Alimentatore	Seasonic S12II-380 - 380 Watt - Scheda Prodotto ³
Case	Silverstone La Scala SST-LC17B - Scheda Prodotto ⁴
Hard Drive	2 x Western Digital WD3200BEVT da 320GB -
Scheda Wi-Fi	D-Link DWL-G510
Unità Ottica	Masterizzatore DVD+/-RW LG

1 – Link : http://www.asus.com/Motherboards/AMD_CPU_on_Board/E35M1M/

2 – Link : <http://www.gskill.com/products.php?index=77>

3 – Link : <http://www.seasonicusa.com/S12II.htm>

4 – Link : <http://www.silverstonetek.com/product.php?pid=82&area=en>

2.2 Assemblaggio

Nell'ordine sono stati effettuati i seguenti passi:

- Posizionamento della scheda madre nel case vuoto
- Fissaggio dei due hard disk in un bay da 3.5" tramite un riduttore 2 x 2.5" - 1 x 3.5"
- Fissaggio del drive ottico nel bay da 5.25"
- Posizionamento sulla scheda madre della scheda Wi-Fi nello slot PCI Express 1x
- Posizionamento sulla scheda madre dei 2 banchi di RAM per un totale di 4 GB
- Fissaggio dell'alimentatore e routing dei cavi in modo da non ostacolare l'unico flusso d'aria forzato che sarà presente nel sistema (da sinistra verso destra nella foto)
- Collegamento delle periferiche SATA alla scheda madre

Il risultato dell'assemblaggio:



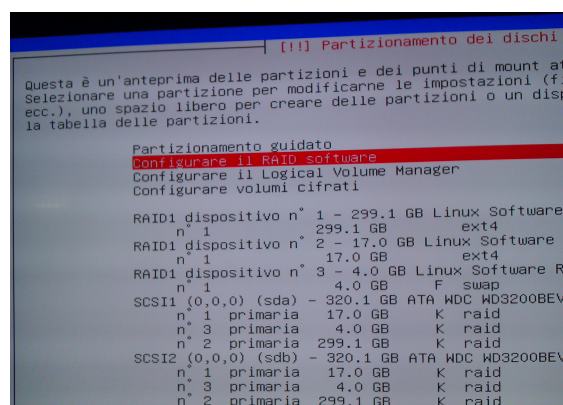
2.3 Installazione Sistema Operativo

Come sistema operativo è stato scelto Debian “squeeze” ovvero la versione attualmente stabile di [Debian](#), la 6.0.5, nella sua versione a 64 bit.



Prima schermata di un'installazione Debian 6.0.5

Ora con Debian se si vuole è possibile configurare gli HDD già durante l'installazione per essere utilizzati in RAID, è sufficiente selezionare “configurazione manuale” quando viene chiesto di scegliere la modalità di configurazione dei dischi, poi bisogna creare nei due dischi delle partizioni della stessa capacità e nello stesso ordine per ciascuno dei seguenti elementi: la “root” che è la partizione dove risiederà il sistema operativo, la “home” che è la partizione dove saranno immagazzinati i dati degli utenti (che conviene tenere separati dalla home perchè in caso si voglia cambiare OS non è necessario fare backup dei dati utente per non perderli e la partizione di “swap” che serve al sistema operativo per salvare temporaneamente dati su cui qualche processo sta lavorando quando la memoria centrale (RAM) è completamente utilizzata. Una volta create le partizioni le si marca con un flag “usa come partizione RAID” e si seleziona la voce “configurare il RAID software”. Verrà dunque chiesto che tipo di RAID si vuole realizzare, quali partizioni utilizzare a due a due e come si desidera usarle (root, home, swap). Il risultato ottenuto è il seguente :



Schema delle partizioni correttamente configurate per un setup dell'OS su RAID1

Una volta terminato il setup dei dischi comincerà la vera e propria installazione che richiederà solo una minima interazione con l'utente per configurare il sistema di gestione dei pacchetti e per l'installazione del bootloader. Al termine dell'installazione il sistema verrà riavviato e durante il caricamento dell'OS dovremmo essere in grado di identificare la presenza dell'utility “mdadm” (multiple device administrator) che carica le 3 partizioni in RAID che sono chiamate md1, md2 e md3.

```
Loading, please wait...
mdadm: /dev/md/2 has been started with 2 drives.
mdadm: /dev/md/1 has been started with 2 drives.
mdadm: /dev/md/3 has been started with 2 drives.
INIT: version 2.88 booting
Using makefile-style concurrent boot in runlevel S.
Starting the hotplug events dispatcher: udevd.
Synthesizing the initial hotplug events...done.
Waiting for /dev to be fully populated...[ 7.367381
address 0xbafe00 already in use
done.
Setting parameters of disc: (none)_
-
```

caricamento delle unità multi disco all'avvio del sistema operativo

Una volta avviato il sistema è possibile controllare lo stato di salute degli array RAID con un semplice comando da terminale “cat /proc/mdstat” che restituisce questa schermata :

```
fedez@winterfell:~$ cat /proc/mdstat
Personalities : [raid1]
md3 : active (auto-read-only) raid1 sda3[0] sdb3[1]
      3878900 blocks super 1.2 [2/2] [UU]

md1 : active raid1 sda2[2] sdb2[1]
      292088696 blocks super 1.2 [2/2] [UU]

md2 : active raid1 sda1[0] sdb1[1]
      16598968 blocks super 1.2 [2/2] [UU]

unused devices: <none>
fedez@winterfell:~$
```

risultato del comando di controllo degli array

Se a fianco di ciascuna partizione multi device troviamo tra parentesi quadre tutte “U” significa che l'array è sano e tutti i dischi sono funzionanti, se invece uno dei due dischi fosse guasto troveremmo un underscore “_” al posto della sua “U”. Se ad esempio il primo disco (/dev/sda) fosse guasto troveremmo il simbolo [_U] a fianco di ciascun array. A quel punto si dovrà procedere alla sostituzione del primo HDD con uno identico e il sistema penserà a ripristinare lo stato di salute del RAID1 copiando lo schema delle partizioni sul nuovo disco e tutti i dati contenuti nel disco sano. Questa è una fase delicata perchè se dovesse succedere qualcosa al disco sano in questo momento allora avremmo perso comunque tutti i dati. Per minimizzare questo rischio a volte si creano RAID1 con 3 dischi in modo da avere della ridondanza anche durante la fase di ripristino del RAID. Se si desidera un livello maggiore di ridondanza è consigliabile rivolgersi ad altri tipi di RAID in quanto un RAID1 a 3 dischi ha un'efficienza in termini di spazio di archiviazione utile / spazio di archiviazione utilizzato solamente del 33.3%.

Capitolo 3

Creazione del Database

Un Database è la rappresentazione digitale di un mini-mondo di interesse. Nella fase che precede la sua creazione è necessario avere una chiara descrizione della realtà da rappresentare identificandone i requisiti e i vincoli. Una volta messi insieme i dettagli della realtà di interesse si possono riassumere le caratteristiche principali in alcune “frasi” che saranno utili quando si passerà alla fase progettuale.

3.1 Analisi dei requisiti

Frase per Libro:

Per ogni Libro, identificato univocamente da un ID, rappresentiamo il titolo, il genere l'autore e la posizione in cui si trova. Inoltre si vuole che sia tenuta traccia degli spostamenti dei libri da una posizione all'altra in modo che se non si riesce a trovarne uno fisicamente si sa dov'era in precedenza e si può controllare se si trova ancora lì.

Frase per Autore:

Per ogni Autore, identificato univocamente da un ID, rappresentiamo il nome, il cognome e quali e quanti libri ha scritto. Deve essere possibile effettuare ricerche di libri per nome o per cognome dell'autore.

Frase per Posizione:

Per ogni Posizione, identificata univocamente da un ID, rappresentiamo la stanza in cui si trova nella casa e una breve ma precisa descrizione di dove si trova perché possono esserci molteplici posizioni all'interno della stessa stanza. Rappresentiamo inoltre la capacità di ciascuna posizione cioè la quantità di Libri che possono essere immagazzinati.

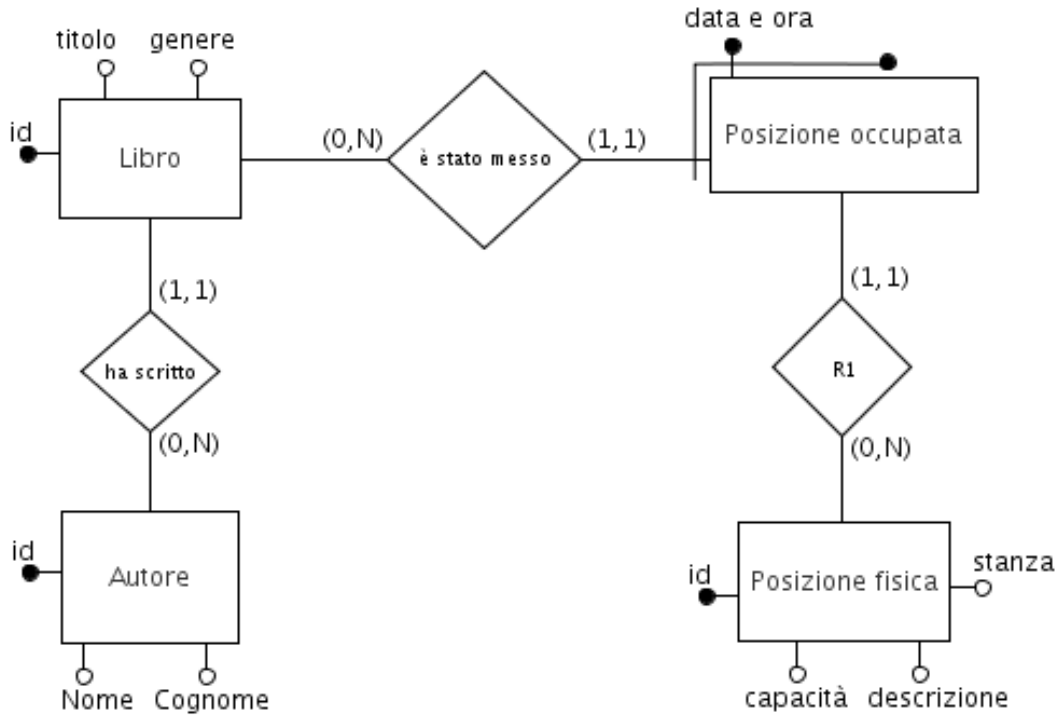
Operazioni sulla base di Dati

Operazione	Tipo	Frequenza
Inserimento Libro	Modifica	5/mese
Inserimento Autore	Modifica	5/mese
Riposizionamento Libro	Modifica	20/mese
Lista libri	Interrogazione	40/mese
Lista Autori	Interrogazione	10/mese

3.2 Progettazione concettuale

Modello Concettuale Entità-Associazione (E-R)

Schema concettuale prodotto per la rappresentazione della realtà di interesse :



Dizionario dei Dati

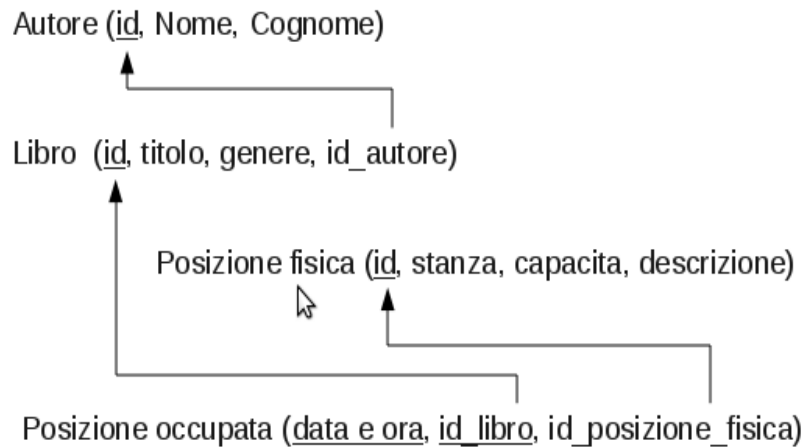
Entità	Descrizione	Attributi	Identificatore
Libro	È immagazzinato	id, titolo, genere	id
Autore	Scriva libri	id, Nome, Cognome	id
Posizione fisica	Posizioni valide all'interno della casa	id stanza, descrizione, capacità	id
Posizione occupata	Posizione occupata dal libro	data e ora	data e ora , Libro

Associazione	Attributi	Entità collegate
Ha scritto		Autore(0,N) Libro(1,1)
È stato messo		Libro (0,N) Posizione Occupata (1,1)
R1		Posizione Occupata (1,1) Posizione Fisica (0,N)

3.3 Progettazione logica

Modello Logico: Relazionale

Schema logico prodotto per la realtà di interesse :



Schema logico: regole di vincolo

- RV 1 Un Libro deve partecipare ad una ed una sola istanza di Autore
- RV 2 Una posizione occupata deve riferirsi solamente ad una posizione fisica
- RV 3 Una posizione occupata deve riferirsi ad una ed una sola istanza di Libro
- RV 4 Gli attributi di Libro devono essere non nulli
- RV 5 Gli attributi di Autore devono essere non nulli
- RV 6 Gli attributi di posizione fisica devono essere non nulli
- RV 7 Gli attributi di posizione occupata devono essere non_nulli

3.4 Codice SQL

Struttura

Con struttura si intende lo schema delle tabelle e dei loro riferimenti che costituisce la vera e propria implementazione del Database, una volta creata è subito possibile inserire valori nelle tabelle coerentemente con i vincoli definiti alla creazione di ciascuna. La struttura è definita in linguaggio SQL (da Structured Query Language) e si presenta così :

```
CREATE TABLE autore (  
id serial,  
nome varchar(64) NOT NULL,  
cognome varchar(64) NOT NULL,  
PRIMARY KEY(id)  
);
```

```
CREATE TABLE libro (  
id serial,  
titolo varchar(128) NOT NULL,  
genere varchar(64) NOT NULL DEFAULT 'sconosciuto',  
id_autore int NOT NULL,  
PRIMARY KEY(id),  
FOREIGN KEY(id_autore) REFERENCES Autore(id)  
);
```

```
CREATE TABLE posizione_fisica (  
id serial,  
stanza varchar(128) NOT NULL,  
descrizione varchar(256) NOT NULL,  
capacita int NOT NULL,  
PRIMARY KEY(id)  
);
```

```
CREATE TABLE posizione_occupata (  
id_libro int,  
data_e_ora timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,  
id_posizione_fisica int NOT NULL,  
PRIMARY KEY(data_e_ora, id_libro),  
FOREIGN KEY(id_libro) REFERENCES libro(id),  
FOREIGN KEY(id_posizione_fisica) REFERENCES posizione_fisica(id)  
);
```

Query

Con query si intende l'interrogazione di un database e serve a compiere diverse operazioni sia sulla struttura del database sia sui suoi contenuti. Con le query è possibile accedere ai contenuti del database formulando le interrogazioni sempre in linguaggio SQL. Seguono alcuni esempi relativi alla realtà di interesse con descrizione della richiesta da formulare, codice SQL della richiesta e risposta del database :

1) Mostrare titolo, genere, nome e cognome autore, data e ora dell'ultimo riposizionamento , stanza e posizione :

```
SELECT l.titolo, l.genere, a.nome || ' ' || a.cognome AS autore , pf.stanza, pf.descrizione, tmp2.deo
FROM ((libro AS l INNER JOIN (SELECT tmp1.id_libro, tmp1.deo, po1.id_posizione_fisica AS idpf
FROM posizione_occupata AS po1 INNER JOIN (SELECT po.id_libro, MAX(po.data_e_ora) AS deo
FROM posizione_occupata AS po GROUP BY po.id_libro) AS tmp1 ON (po1.id_libro, po1.data_e_ora) =
(tmp1.id_libro, tmp1.deo)) AS tmp2 ON l.id = tmp2.id_libro) INNER JOIN autore AS a ON l.id_autore =
a.id) INNER JOIN posizione_fisica AS pf ON tmp2.idpf = pf.id;
```

titolo	genere	autore	stanza	descrizione	deo
Eragon	Fantasy	Christopher Paolini	camera fede	mensola sopra il letto	2012-09-10 15:41:19.729867
Eldest	Fantasy	Christopher Paolini	camera fede	mensola sopra il letto	2012-09-10 15:41:19.729867
Brisingr	Fantasy	Christopher Paolini	camera fede	mensola sopra il letto	2012-09-12 15:26:23.046116
Inheritance	Fantasy	Christopher Paolini	camera fede	mensola sopra il letto	2012-09-12 15:26:26.789903
Il signore degli anelli	Fantasy	J.R.R. Tolkien	camera fede	mensola sopra il letto	2012-09-12 15:31:24.122828
Le cronache del ghiaccio e del fuoco Vol I	Fantasy	George R.R. Martin	camera fede	mensola sopra il letto	2012-09-12 15:39:40.895464
Tempesta di spade	Fantasy	George R.R. Martin	camera fede	mensola sopra il letto	2012-09-12 15:39:45.071772
I fiumi della guerra	Fantasy	George R.R. Martin	camera fede	mensola sopra il letto	2012-09-12 15:39:49.095667
Il portale delle tenebre	Fantasy	George R.R. Martin	camera fede	mensola sopra il letto	2012-09-12 15:39:52.999731
Il dominio della regina	Fantasy	George R.R. Martin	camera fede	mensola sopra il letto	2012-09-12 15:39:59.160019
L'ombra della profezia	Fantasy	George R.R. Martin	camera fede	mensola sopra il letto	2012-09-12 15:40:03.997195

2) Mostrare lo storico delle posizioni occupate da un determinato libro identificato dal titolo :

```
SELECT titolo, data_e_ora, stanza, descrizione
FROM (posizione_occupata INNER JOIN libro ON posizione_occupata.id_libro = libro.id) INNER JOIN
posizione_fisica ON posizione_occupata.id_posizione_fisica = posizione_fisica.id WHERE libro.titolo =
'Inheritance';
```

titolo	data_e_ora	stanza	descrizione
Inheritance	2012-09-12 15:26:26.789903	camera fede	mensola sopra il letto
Inheritance	2012-09-15 16:14:28.948132	camera andrea e mirella	scaffale lato ovest
Inheritance	2012-09-15 16:15:44.794095	camera fede	mensola sopra il letto

(3 righe)

libreria=# █

3) Mostrare la situazione di riempimento dei vari posti in cui sono immagazzinati i libri :

```
SELECT pf.stanza, pf.descrizione, count(tmp1.id_libro) AS "libri presenti", pf.capacita
FROM posizione_fisica AS pf LEFT JOIN (SELECT tmp0.id_libro, po1.id_posizione_fisica AS idpf
FROM (SELECT po.id_libro, MAX(po.data_e_ora) AS deo
FROM posizione_occupata AS po GROUP BY po.id_libro) AS tmp0 INNER JOIN posizione_occupata
AS po1 ON (tmp0.deo, tmp0.id_libro) = (po1.data_e_ora, po1.id_libro)) AS tmp1 ON pf.id = tmp1.idpf
GROUP BY tmp1.idpf, pf.stanza, pf.descrizione, pf.capacita;
```

stanza	descrizione	libri presenti	capacita
corridoio	scaffali bianchi	0	400
salotto	mobile lato nord	0	250
camera fede	mensola sopra la scrivania	0	40
camera sofia	scaffale testiera del letto	0	50
camera sofia	mensola sopra la scrivania	0	20
camera fede	mensola sopra il letto	11	40
camera andrea e mirella	scaffale lato ovest	0	150

(7 righe)

3.5 Software utilizzato

Come sistema di gestione dei Database (o DBMS da DataBase Management System) è stato scelto PostgreSQL grazie al fatto che si tratta di software libero e quindi in linea con le scelte compiute finora e anche grazie all'esperienza pregressa avuta con questo DBMS che si è rivelato facile da configurare ed utilizzare.

In un sistema Debian è stato sufficiente chiamare da terminale il gestore di pacchetti e con un semplice comando `sudo apt-get install postgresql-8.4` il DBMS desiderato è stato installato.

E' stato poi necessario loggare come utente proprietario del database ovvero come "postgres" con i seguenti comandi `sudo su` e `su postgres` per poter creare gli utenti necessari alla creazione e all'interrogazione del database. Sono stati creati due utenti, uno con privilegi di amministratore dei database con il mio nome utente ed uno senza privilegi a cui poi si garantiranno i permessi per collegarsi e fare interrogazioni al database desiderato. Il comando per creare gli utenti, una volta loggati come utente "postgres" è `createuser "nome_utente"` e a questo comando seguiranno le richieste sui privilegi che l'utente appena creato deve avere. Una volta aggiunto come amministratore di database il mio utente ho potuto creare il database "libreria con il comando `createdb libreria` per poi accedervi con il comando `psql libreria` e creare la struttura del database inserendo il codice SQL mostrato in precedenza.

Una volta creato il database si è scelto di dare all'utente "www" i permessi necessari ad accedere al database e alla sua interrogazione ma non alla sua modifica per eliminare il pericolo che un utente attraverso il web possa modificare il contenuto o la struttura del database. La concessione di questi permessi è stata data con i comandi

```
GRANT connect ON database libreria TO www
```

e

```
GRANT select ON table libro, autore, posizione_occupata, posizione_fisica TO www .
```


Capitolo 4

Setup del web server

Il software che è stato scelto per servire le pagine web è il web-server apache2 che come postgresQL è un software libero e già nelle repository ufficiali di Debian quindi per l'installazione è stato necessario solamente chiamare nuovamente il gestore dei pacchetti Debian con il comando `apt-get install apache2`. Il web server non ha avuto bisogno di essere impostato per funzionare e già subito dopo l'installazione era attivo e stava servendo sulla porta 80 una pagina di benvenuto situata nella cartella `/var/www`. Sostituendo il file `index.html` con il nostro appena creato ecco che la pagina servita è diventata la pagina con l'interfaccia utente per l'interrogazione del database. Ma per farla funzionare sono serviti un altro paio di passi.

4.1 Progettazione e realizzazione dell'interfaccia grafica per la biblioteca digitale :

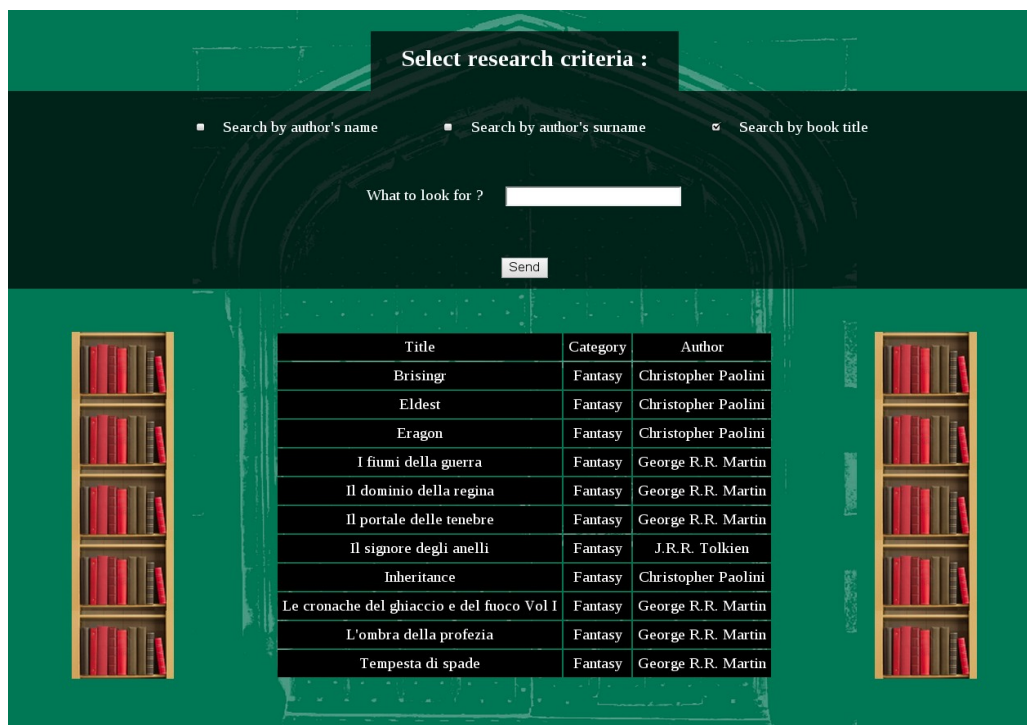
Le pagine HTML che costituiscono l'interfaccia web sono state realizzate utilizzando un semplice editor di testo che si trova preinstallato nella configurazione standard di Debian ed è chiamato “gedit”.

L'interfaccia doveva rispettare i principi di facilità d'uso, immediatezza e completezza. I servizi che doveva fornire per quanto riguarda la consultazione del database sono i seguenti :

- possibilità di visualizzare la lista completa degli autori ordinata per nome o per cognome
- possibilità di visualizzare la lista completa dei libri
- possibilità di cercare autori per nome o cognome o entrambi
- possibilità di cercare libri per titolo
- possibilità di visualizzare la lista di libri scritti da un autore cercando per nome/cognome autore

Questi risultati sono stati ottenuti mettendo a disposizione dell'utente un campo di testo in cui è possibile inserire parole o parti di parole da cercare nel database e 3 “checkbox” con cui è possibile selezionare le modalità di ricerca tra ricerca per nome autore, ricerca per cognome autore e ricerca per titolo del libro o una combinazione dei 3.

Ad esempio selezionando “ricerca per cognome autore” e per “titolo del libro” e inserendo ad esempio la parola “pao” (la parte iniziale del cognome “Paolini”) il risultato è la lista dei libri scritti da qualsiasi autore con le lettere “pao” nel cognome e dei libri in cui c'è la sillaba “pao” all'interno del titolo affiancati dai nomi e cognomi degli autori che li hanno scritti e dal luogo in cui si trovano nella libreria.



4.2 Collegamento web server - database

Una volta creata l'interfaccia utente è stato necessario far interagire la pagina web con il database sottostante.

Per fare questo è stato scelto un ulteriore strumento libero chiamato PHP (che sta per PHP: Hypertext Preprocessor) alla sua versione attuale, con il comando `apt-get install php5` è stato installato assieme anche al plugin per l'utilizzo di PHP con il web server apache.

A questo punto è stato sufficiente creare uno script in PHP che servisse da ponte tra le chiamate fatte da altri script PHP al database inserendo il nome del database con cui si vuole interagire, nel nostro caso "libreria" e il nome utente con cui si vuole accedere, ovvero "www".

Una volta istituito questo collegamento è stato possibile fare con PHP delle query al database presentando i risultati in una pagina web, che era proprio quello che si voleva ottenere.

4.3 Registrazione Dominio

L'ultimo passo per rendere il nostro sito web consultabile agevolmente dall'esterno è stato registrare l'indirizzo IP fisso di casa presso un qualsiasi provider DNS (da Domain Name System) che fornisca sottodomini gratuitamente, nel nostro caso è stato FreeDNS. E' bastato configurare il router di casa perchè reindirigesse tutte le richieste a lui fatte sulla porta 80 all'IP interno del computer che mantiene il web server perchè fosse possibile raggiungere la pagina web della libreria da qualsiasi browser andando alla pagina :

<http://lanboratory.strangled.net>

Capitolo 5

Conclusioni

Partendo dal bisogno di poter consultare la biblioteca di casa da remoto si è pensato di realizzare una biblioteca digitale che come servizio fornito avesse proprio la consultazione degli elenchi di libri presenti nella biblioteca fisica di casa, con la possibilità di effettuare ricerche di vario tipo sugli stessi.

Si è dunque passati alla progettazione del computer che avrebbe dovuto mantenere online la biblioteca digitale pensando ai requisiti che doveva avere per adattarsi alle esigenze di un ambiente familiare.

E' seguita la fase di reperimento dei materiali necessari all'assemblaggio e la fase di assemblaggio stessa, al termine della quale è stato installato il sistema operativo e il software necessario alla creazione del database e alla configurazione del Web server. E' stato poi necessario progettare il database che avrebbe mantenuto la vera e propria biblioteca digitale e l'interfaccia utente sotto forma di pagine HTML con la quale si intendeva accedere alla biblioteca. L'ultimo passo, necessario per rendere agevole l'accesso da remoto alla biblioteca digitale è stato la registrazione di un sottodominio presso un provider DNS che permette a chiunque sia connesso ad Internet di raggiungere l'interfaccia di accesso al database tramite qualsiasi browser su PC, tablet o smartphone. E' facile immaginare l'uso in futuro della stessa piattaforma per il mantenimento di altre biblioteche digitali che potrebbero contenere ad esempio la collezione delle ricette di famiglia o la collezione di album fotografici o qualsiasi collezione si voglia mantenere nel tempo e a cui si voglia poter accedere in qualsiasi momento da remoto.

