

Università degli Studi di Padova
Facoltà di Ingegneria

Corso di Laurea Triennale in Ingegneria Informatica

Tesi di laurea triennale

Una base di dati per la gestione di materiale didattico multimediale

Struttura di un'applicazione

Candidato:
Simone Zaminga
Matricola 491154

Relatore:
Nicola Orio

Anno Accademico 2011–2012

Spunta il giorno. Il tempo è davanti a voi. Un'alba. Questo giorno che ci sta davanti - voi dite - lo faremo noi! - Sì? Voi? E salutatemi tutte le tradizioni! Salutatemi tutti i costumi! Mettetevi a parlare! Ripeterete tutte le parole che si sono sempre dette! Credete di vivere? Rimasticate la vita dei morti!

— Luigi Pirandello - Enrico IV

Dedicato alla mia famiglia ed a tutti coloro che mi hanno sospinto a raggiungere questo traguardo.

INDICE

1	INTRODUZIONE	1
1.1	Quadro Generale	1
1.1.1	Docenti	1
1.1.2	Tecnici	2
1.1.3	Considerazioni personali	3
2	MODELLO DI UN PROGETTO	5
2.1	Introduzione alla progettazione	5
2.1.1	Ciclo di vita dei sistemi informativi	5
2.1.2	Metodologie di progettazione e basi di dati	7
2.1.3	Progettazione concettuale	7
2.1.4	Progettazione logica	9
2.1.5	Progettazione fisica	10
2.1.6	Accenno alle architetture web-oriented	10
3	PROGETTO	13
3.1	La progettazione concettuale	13
3.1.1	Raccolta dei requisiti	13
3.1.2	Schema Entità-Relazione	15
3.2	La progettazione logica	17
3.2.1	Ristrutturazione dello schema E-R	17
4	PROGRAMMAZIONE	25
4.1	Creazione del web server	25
4.1.1	Un po' di storia	25
4.1.2	Scelte operative	26
4.2	Software in esercizio	28
5	CONCLUSIONI	31
A	APPENDICE	33
A.1	Questionario	33
A.2	Schema relazionale	33
A.3	Listato SQL	34
	Bibliografia	41

ELENCO DELLE FIGURE

Figura 1	Schema Entità-Relazione	16
Figura 2	Schema Entità-Relazione: ristrutturato	22

ELENCO DELLE TABELLE

Tabella 1	Dizionario dei dati: entità	19
Tabella 2	Dizionario dei dati: relazioni	23

Forse è in cantina. Vado un attimo di sopra a controllare.

— *M. C. Escher*

RINGRAZIAMENTI

Mamma Sisiglia, papà, soruma, Mescia Sina, il ricordo di Mesciu Roccu, zia Mirella, zio Totò, il ricordo di zio Franco, zia Lucia, zia Teresa, zio Mario, zio Marco, zia Anna Maria, zio Antonio, zia Claudia, Amato Alessandro, Rosa, Francesco, Giulio, Alessio, Moira, Chiara, Matteo, Elena, ricordo di Alessandro Zaminga, il ricordo di zio 'ntonucciu, zia Graziella, zio Pippi e zia Lucia, il ricordo di zia Franca e zio Antonio, zia Cesarina, Katia, Paola, Alessandro, il ricordo di zio Gino, Davidissimo, Anna, Biondo, Domenico, Ciola, Francesco, Milanese, Luigina, Pilusu, Andrea, Dieguz, Luis, Lara, Tasso, Buba Maura, Ilaria, Farfalla, Diletta, Sonia, Victor, Banana, Raffaella, Buscio, Luca Leo, Presidente, Beбето, Gioia, Fausto, Luisa e Gabriele, 40, Santurizzu, Carlo, Lustro, Leo, Fabio, Federica, Luca e Filippa, Rocco e Tiffany, Antonio Di Noia, Bombea, Giacoma, Dario, Stefanuccio, Matteo, Nosellame, Pippo, Fabio, Lesley, Marsi, Diego Ceccarello, Paolo, Angela, Giacoma, Mirco Ballò, Samena, Khalil, il rock, il metal, il blues, il punk, Jimi Hendrix, Chuck Schuldiner, Barrett, Waters, Gilmour, Mason, Wright, Clapton, Harris, Murray, Smith, Di'Anno, Burr, Dickinson, McBrain, Beethoven, Mozart, Bach, Chopin, i colleghi di lavoro, il dottore e relatore Nicola Orio che ha curato la mia tesi in tutte le sue parti.

Padova, settembre 2012

S. Z.

SOMMARIO

Lo scopo di questo lavoro è fornire agli utenti come professori, studenti e tecnici che popolano l'università, uno strumento utile rispettivamente per la ricerca, lo studio e la catalogazione di materiale multimediale. Tale obiettivo si è reso necessario per la moltitudine di informazioni presenti presso il Dipartimento dei Beni Culturali. Informazioni collocate spesse volte su supporti ottici come DVD, ubicati in armadi e contenenti centinaia di file con riferimento ad audio, foto, video e documenti. Trovare ad esempio una determinata immagine, magari salvata con un solo codice identificativo, diventa un lavoro cospicuo in termini di tempo utilizzato e di energie. Creare una piattaforma invece, dove riversare tutte le informazioni d'interesse, affiancate da un titolo e da parole chiave utili per il reperimento unite ad un motore di ricerca, è quello che ci occorre per poter ottenere in modo più efficace ed efficiente tale traguardo.

Posto in questi termini, anche se funzionale, questo archivio informatico è molto restrittivo perché sarebbe adoperato dal solo personale tecnico. L'idea quindi di uno strumento universale che faccia confluire le diverse realtà ed aree del dipartimento è stato il passo successivo. Per cui l'applicazione è stata progettata per essere reperibile on-line, di conseguenza non più archivio privato ma web server ove i diversi professori possono collegarsi per reperire il proprio materiale. Ed è proprio questo il punto. Tramite tale applicazione sarà possibile ai docenti cercare dei file, dopo essersi autenticati, creare delle cartelle private, o condivise con docenti o studenti, inserendo all'interno magari la stessa lezione spiegata in mattinata, in modo che se qualche studente era assente, possa ritrovare quanto spiegato dal docente in un secondo momento.

Durante l'elaborazione del progetto, si è dovuto intervistare gli utilizzatori finali, in modo da poter analizzare i problemi tipici che riscontravano durante la formulazione ed il reperimento del loro materiale didattico, potendo così sondare le esigenze e le difficoltà per l'utilizzo del un nuovo strumento di lavoro. Ampio spazio lo si è dato anche alle necessità dei tecnici implementando delle interfacce che tengono nota del tipo di strumentazione e di varie informazioni più minute, per ogni file prodotto con riprese in esterna dallo stesso personale tecnico. Questo permetterà loro di agevolare il lavoro di catalogazione evitando di occupare spazio fisico per mezzo dei supporti ottici, centralizzando il tutto in un solo server ove l'unica peculiarità sarà quella di predisporre un solido strumento di sistema per il backup dei dati.

La scelta delle soluzioni adottate è avvenuta in collaborazione con il del dott. Orio, relatore della tesi, che mi ha permesso di realizzare una solida base di dati, suggerendomi utili osservazioni funzionali durante la programmazione in PHP.

L'esposizione del lavoro è articolata come segue:

NEL PRIMO CAPITOLO viene offerta una breve visione d'insieme del contesto in cui è nato lo sviluppo del progetto, vengono inoltre presentate le idee di fondo che lo caratterizzeranno.

NEL SECONDO CAPITOLO vengono introdotte le fasi del processo di elaborazione di un progetto ed i costrutti utilizzati per la creazione di schemi

Entità-Relazione.

NEL TERZO CAPITOLO vengono esposte tutte le fasi di realizzazione del progetto dalla progettazione concettuale sino alla progettazione logica che ha come prodotto la ristrutturazione dello schema Entità-Relazione.

NEL QUARTO CAPITOLO vengono descritte, sinteticamente, le scelte adottate durante la realizzazione del software in fase di programmazione. In ultima istanza, si espone il funzionamento del web server in esercizio.

NEL QUINTO CAPITOLO vengono riferite alcune considerazioni personali in chiusura del documento di tesi.

NELL'APPENDICE sono presenti documenti e listati prodotti durante l'elaborazione del progetto.

NELLA BIBLIOGRAFIA sono enunciate le fonti primarie di consultazione.

PREMESSA

Il seguente documento nasce innanzitutto come conclusione di un percorso universitario intrapreso anni fa. Naturalmente delle diverse discipline in cui si articola il Corso di Studi in Ingegneria Informatica è prevalso quello inerente all'insegnamento di Basi di Dati. D'altro canto, non poteva essere altrimenti visto che ricopro un ruolo di tecnico informatico nel Dipartimento di Beni Culturali dell'Università di Padova e dove la necessità di catalogare in modo sicuro ed ordinato il materiale didattico multimediale è di primaria importanza. Quanto esposto potrà essere utilizzato come strumento di lavoro per quanti la prima volta si avvicineranno al progetto per apportare modifiche o comprendere il funzionamento.

Nel laboratorio audio-visivi, sede del mio lavoro, è presente una cospicua mole di immagini, si contano oltre 30.000 unità. In modo del tutto naturale nasceva l'esigenza di poterle catalogare univocamente attraverso la creazione di un database.

A questo ci avevano già pensato i miei predecessori ma non si era giunti ancora ad una vera e propria realizzazione di alcun software. Riesumando la vecchia documentazione presente e lavorando alla prima bozza di progetto, si è ritenuto utile che tale database potesse diventare una vera e propria applicazione a tutti gli effetti per il corpo docente. Nello specifico, a lavori ultimati, si potrà attraverso un proprio spazio privato, caricare le immagini ricercate nel database creando un file PDF e/o PowerPoint. Questo strumento potrebbe avere riscontri vantaggiosi nella formulazione delle lezioni e come strumento di ricerca.

La recente approvazione della riforma della scuola, legge Gelmini, ha portato nell'anno 2011 ad una riorganizzazione dei Dipartimenti di Ateneo. Il Dipartimento di Storia delle arti visive e della musica, si accorpava con Archeologia e Storia del Cinema, facendo nascere il Dipartimento dei Beni Culturali: Archeologia, Storia dell'Arte, del Cinema e della Musica.

Il progetto per cui lavoravo era ad un punto di stallo. Avevo ben disposto degli schemi E-R (Entity-Relationship), ed anche creato un piccolo database in MySQL ma, mancava interamente un substrato forte: lo studio dell'analisi dei requisiti e tutta l'implementazione del web-server per l'accesso remoto all'applicazione di basi di dati. Inoltre, l'inesperienza diretta a tale richiesta minava già in partenza le basi per un risultato soddisfacente. Fortunatamente dovetti affrontare l'esame di Basi di Dati come ultima prova della carriera di studi universitari; allora forte del nuovo bagaglio di conoscenze acquisite non mi perdetti d'animo e cominciai a rivedere il lavoro svolto. Mi balenò l'idea di portare lo sviluppo dell'applicazione come elaborato di tesi, in modo da avere la supervisione ed i consigli del relatore per poi riuscire a produrre un risultato qualitativamente buono e che rispondesse alle esigenze cui sopra ho indicato.

Il confluire di questi eventi, esternamente dettato dalla dipartimentazione ed internamente dal connubio del progetto sia come elaborato di tesi che come finalità nel campo lavorativo, hanno portato nuova linfa per migliorare ulteriormente quanto si era prestabilito in partenza, come di seguito esplicherò. Parlando col dott. Orio, relatore della tesi, si è giunti alla conclusione che si poteva abbracciare tutte le nuove entità nel Dipartimento dei Beni Culturali: costruendo un progetto solido che non solo riguardasse le

immagini, come in principio auspicato bensì anche la musica, e il cinema, salvando sole poche cose della struttura iniziale ed reingegnerizzando la precedente attività.

1

INTRODUZIONE

1.1 QUADRO GENERALE

La legge Gelmini ha permesso alle diverse realtà dipartimentali di potersi unire sotto un'unica egida. Naturalmente ogni docente e personale incardinato ha portato con sé il proprio bagaglio organizzativo che ora si sta cercando di uniformare. Per la stesura di questi paragrafi mi attengo a quanto ho percepito dall'esperienza diretta avvenuta in ambito lavorativo dal contatto coi docenti e colleghi del Dipartimento di Storia delle arti visive e della musica. Facciamo ora un quadro generale del contesto in cui si è evoluto il progetto. Verranno esposte le modalità di organizzazione della didattica da parte dei docenti e dei tecnici che collaborano con essi.

1.1.1 Docenti

Ogni insegnamento viene impartito in apposite aule, di solito sempre gremite da una folta platea di universitari. I docenti hanno a disposizione diverse opportunità per divulgare il loro esercizio. Ovviamente microfoni, lavagna e gesso, se parliamo delle lezioni in modo più *classico*. Ma questo non basta e non è sufficiente, se dobbiamo rivolgerci a degli studenti finalizzati ad apprendere i fondamenti della storia dell'arte e della musica. Per cui se qualche anno fa, al massimo una decina, l'utilizzo degli stativi e dei proiettori di diapositive per riprodurre le immagini e/o dei mangianastri o lettori Compact Disk amplificati da casse, per quanto riguarda le lezioni di musica, erano apparecchiature comunemente utilizzate durante le ore d'istruzione, ora nell'era digitale molti di questi mezzi sono superati. Si ricorre all'utilizzo del PC portatile collegato ad un proiettore il quale trasmetterà su uno schermo i filmati, le immagini e le presentazioni. Per il suono invece vi è un piccolo armadietto multifunzione, posto di fianco alla cattedra, che permette l'esecuzione di brani o di filmati direttamente riprodotti da CD, DVD. Ovviamente l'armadietto è un piccolo concentratore che compie altre operazioni importanti per l'esposizione della lezione, come ad esempio: l'accensione del videoproiettore, la connessione tra proiettore e computer per mezzo di un cavo con uscita VGA, la regolazione dei valori di equalizzazione e dei volumi delle casse e dei microfoni, eccetera. Infine le aule sono munite di presa per il collegamento IP qualora fosse necessario navigare o svolgere operazioni in rete.

Come accennato poco prima all'interno di questo paragrafo, il materiale delle lezioni del corpo docente è in linea di massima prevalentemente costituito da presentazioni in PowerPoint eseguite col pacchetto software di Microsoft Office e proiettate in aula. Si tratta abitualmente di mettere insieme una carrellata di diapositive, dette anche slide, costituite da poche nozioni scritte che introducono o affiancano un'immagine. Certi docenti invece eliminano il testo e preferiscono le immagini sciolte. E' prassi comune, in quest'ultima eventualità, utilizzare programmi come Windows Media Player o IrfanView. Attualmente molti docenti stanno abbandonando l'ambiente desktop di Microsoft per convertirsi a quello Mac sicché alcuni pro-

grammi potrebbero non essere implementati nel nuovo sistema operativo, anche se c'è una tendenza ad uniformare il tutto fornendo una certa corralità degli strumenti di lavoro; tant'è vero che vi è una suite del pacchetto Office anche nel contesto di Apple.

Il discorso è leggermente differente per le discipline di Storia della musica. Oltre alle consuete presentazioni composte da testo ed immagini, possono essere riprodotti all'interno dello stesso file di PowerPoint anche dei filmati o più semplicemente dei brani audio. Delle volte anche le proiezioni dei file PDF (Portable Document Format) possono trovare spazio nella didattica i quali ripropongono le trascrizioni antiche di partiture musicali affiancate dall'esecuzione ed ascolto in parallelo del brano proposto. La trascrizione in chiave moderna, delle partiture antiche è normalmente lasciato al programma Finale che permette a lavoro terminato di creare i file PDF.

Il supporto del CD per la portabilità di alcuni brani originali come quello del DVD per visualizzare dei filmati è in leggero declino. Prassi comune è quella di caricarsi preventivamente i file sui computer portatili o su chiavette USB, vista la capacità sempre maggiore di memoria, rispettivamente degli hard disk e delle memorie a stato solido, successivamente di eseguirli direttamente dal portatile. Questa soluzione è molto vantaggiosa perché permette di ridurre ai minimi termini l'utilizzo di supporti esterni accedendo alla risorsa in pochi clic.

La principale risorsa per il personale docente come abbiamo potuto constatare sono i file. Essi possono giungere da diverse fonti:

- **web:** per quelle opere di dominio pubblico dove non è necessario conoscere i particolari ma solo l'opera d'arte in sé;
- **biblioteche:** in base ad accordi o ad abbonamenti intercorsi tra l'università e l'ente è permesso l'accesso e l'utilizzo di banche dati per fini didattici ;
- **musei:** anche qui si utilizzano degli accordi o degli abbonamenti coi musei per poter usufruire del materiale posto nelle loro banche dati;
- **riprese esterne:** i tecnici di laboratorio fotografico effettuano delle riprese fotografiche esterne per dei progetti di ricerca commissionati dai docenti in accordo con l'ente che detiene l'opera. Di solito possono essere riprodotti dei manoscritti, libri miniati, affreschi, elementi architettonici, spartiti, eccetera ;
- **riprese da fonti edite:** possono essere eseguite in laboratorio delle inquadrature fotografiche da alcuni libri, manoscritti, vecchie foto, eccetera;
- **registrazioni:** vengono alcune volte eseguiti dei riversamenti, con conseguente conversione, in file di alcuni long-playing o musicassette.

1.1.2 Tecnici

Di supporto al corpo docente, come accennato poco sopra, vi sono i tecnici di laboratorio fotografico. Il sottoscritto invece fornisce un aiuto mirato più al campo informatico: hardware, software e periferiche per cui strettamente correlato all'utilizzo del PC.

Il ruolo dei tecnici di laboratorio fotografico è ben diverso. Innanzitutto per le riprese esterne sono così organizzati:

1. il docente presenta un progetto di ricerca autorizzato, chiedendo la partecipazione dei tecnici per la produzione del materiale fotografico su un determinato soggetto da riprendere;
2. fase di pianificazione del lavoro e di strategia composta, se necessario, da un sopralluogo da parte dei tecnici per valutare il soggetto e le modalità di posizionamento della apparecchiature fotografiche per eseguire al meglio gli scatti. Viene così prodotta, un'immagine originale;
3. in laboratorio si esegue la fase di post-produzione. L'immagine viene ridimensionata ed elaborata per risaltarne le qualità. L'ottimizzazione della stessa avviene su richiesta esplicita del docente, il quale interviene, consigliando al tecnico, di risaltare eventualmente alcuni dettagli o nell'ordine della dimensione e risoluzione. Ad esempio perché le immagini possano rientrare negli standard di alcune basi di dati per poter essere caricate, vedi IPSA;
4. fase di consegna delle immagini su supporti magnetici, dopo aver firmato una delibera di consegna e di copyright del prodotto. Per lavorazioni già riproposte e meno importanti la trasmissione dei file può avvenire anche per mail.

La sequenza di operazioni, appena descritte, può avvenire anche nell'ambito di una registrazione audio di una conferenza o qualsiasi altra manifestazione ove è richiesta la nostra presenza. Naturalmente il prodotto non sarà più un'immagine bensì un file sonoro, equalizzato e pulito dai rumori. Questo genere di richieste sono molto rare, per questo non è stato definito un protocollo di produzione.

Se ritenuto necessario dal docente, proprio perché incentrate nel programma del corso, i file prodotti dai tecnici vengono caricati dagli stessi nelle postazioni informatiche poste nella biblioteca come strumento di consultazione e didattica rivolto agli studenti. Ovviamente le limitate condizioni di accesso, grazie al sostegno di determinate protezioni e limitazioni software inibiscono l'uso di molte funzioni del terminale evitando il salvataggio su qualsiasi supporto, presente o remoto che sia, da parte dello studente. In questa maniera viene garantita la riservatezza e l'utilizzo improprio dei dati.

Un'ulteriore sinergia tra i soggetti avviene per la creazione di presentazioni in PowerPoint, la selezione e scelta di immagini provenienti da diverse fonti, il supporto durante le lezioni nel caso che qualche malfunzionamento impedisca la corretta esecuzione della lezione ancorché, come annunciato in precedenza, la produzione in laboratorio di immagini prelevate da testi o materiale già edito in generale. Anche per quest'ultima attività viene consegnato un CD, o DVD, con annessa delibera controfirmata dall'insegnante, il quale è responsabile dell'utilizzo del materiale concesso.

Infine, per concludere il quadro generale dei rapporti tra docenti e personale tecnico bisogna aggiungere che questi non sono limitati solo all'istruzione che d'altronde non è l'unico fine dell'università; anche nella ricerca questo legame continua. Il supporto tecnico ai convegni di studio e nei seminari è di primaria importanza per il buon esito dell'evento.

1.1.3 Considerazioni personali

Per quanto mi riguarda son davvero entusiata di poter partecipare attivamente oltreché essere immerso in questa realtà. Certo il progetto che vado

a sviluppare non è dei più semplici ed è forse un po' ambizioso essermene assunto l'onere, ma personalmente le sfide mi appassionano. Il background culturale, grazie al percorso di studi intrapreso in ingegneria informatica, permette di poter affrontare al meglio la progettazione e la realizzazione di questa aspirazione.

Il duplice fine è quello di dare uno strumento valido sia per la didattica che per la ricerca. In realtà l'uniformità dei requisiti raccolti a livello di utente fa prevalere la prima soluzione, ovvero quella orientata all'insegnamento. Poter permettere ai docenti ed agli studenti di avere un ulteriore punto in comune dove reperire delle informazioni: riservandomi di spiegare nel proseguo del testo, le differenti caratteristiche dell'architettura implementata, le decisioni prese durante la progettazione, ed i risultati ottenuti. Mi attengo ad esporre tutte queste caratteristiche con chiarezza e semplicità, in modo da introdurre il lettore a beneficiare delle informazioni pratiche e guidarlo dall'idea iniziale del progetto sino alla realizzazione finale. Mi auguro che questo sia solo un punto di partenza e non di arrivo. Una volta che il prototipo sarà testato dagli utilizzatori per un determinato periodo di tempo, potrò verificare le impressioni sull'utilizzo dell'applicazione. Se intorno ad esso ci sarà interesse, lo sviluppo si arricchirà nei mesi a venire di nuove funzionalità prese dai suggerimenti degli utenti, venendo incontro alle loro esigenze. Insistendo su questo punto, nell'ottica idilliaca di successo, altri tecnici o ingegneri potrebbero avere interesse di partecipare al progetto; proprio per questo il mio obiettivo è di lasciare loro in mano una tesi che spieghi tecnicamente e umanamente la genesi del progetto sotto ogni profilo, in modo da non trovarsi disorientati nel contribuire al suo sviluppo ulteriore. Per ora restiamo con i piedi per terra; iniziamo piuttosto a far sul serio dal prossimo capitolo calandoci nella realtà dell'implementazione progettuale.

2

MODELLO DI UN PROGETTO

2.1 INTRODUZIONE ALLA PROGETTAZIONE

2.1.1 Ciclo di vita dei sistemi informativi

Per restare legati ai modelli accademici dell'insegnamento del corso di Basi di Dati conseguito durante la formazione universitaria suddividiamo le fasi del processo di elaborazione del progetto in: *progettazione concettuale*, *progettazione logica* e *progettazione fisica*. Partiamo dando una breve introduzione dei sistemi informativi e di come si progetta una base di dati. La progettazione di una base di dati costituisce solo una delle componenti di sistema informativo eterogeneo per cui il processo di sviluppo va inteso come quello, a tutti gli effetti del ciclo di vita dei sistemi informativi. Il ciclo di vita di un sistema informativo comprende, in generale, le seguenti attività:

- **Studi di fattibilità.** Serve a definire, in maniera precisa e per quanto sia intuibile, le priorità e le valutazioni delle componenti che formeranno il sistema. La nostra progettazione mira innanzitutto a creare un supporto ai docenti e facilitarli per creare il materiale necessario per le lezioni ed eventuale ricerca personale comunque connessa a fini didattici, come più volte è ribadito nel presente documento di tesi. La cosa forse non ancora esplicitata è che l'applicazione sarà una base di dati web oriented. Questo vuol dire che gli utenti che adopereranno il software dovranno interfacciarsi con un browser il quale farà delle richieste al server, quest'ultimo fornirà delle risposte e permetterà tutte le operazioni concesse dal progetto. Nel nostro caso si è cercato di fare una stima sommaria dell'apparato informatico presente in Dipartimento dei Beni Culturali, soprattutto sotto il profilo logistico e strutturale. In base ai sistemi presenti e da quel ch'è emerso dai discorsi intercorsi col collega di laboratorio, si è stimato che per la futura applicazione dovrà necessariamente essere acquistato un server dove poterla installare. La scelta sicuramente s'indirizzerà verso i modelli di server *rack*; chiamati così perché adottano misure standard e si posizionano in determinati armadietti studiati per contenere tutta l'apparecchiatura informatica per un data center. Tant'è vero che si sfrutterà senz'altro questa inclinazione, suggerita dal fatto che nei sotterranei del Palazzo Liviano è già presente una sala server dell'ex Dipartimento di Archeologia; l'ambiente è climatizzato per rispondere a determinati dettami di temperatura ed umidità che garantiscono la massima efficienza di funzionamento.
- **Raccolta ed analisi dei requisiti.** Consiste nella individuazione e nello studio delle proprietà e delle funzionalità che il sistema informativo dovrà avere. Questa fase richiede un'interazione con gli utenti del sistema e produce una descrizione completa ed informale, dei dati coinvolti e delle operazioni su di essi. Darò indicazioni più accurate a riguardo proseguendo avanti nella stesura del documento, più precisamente nel paragrafo [3.1.1](#) a pagina [13](#) che riguarda la progettazione

concettuale verranno definiti i criteri per poter sfruttare al meglio i requisiti raccolti.

- **Progettazione.** Essa si divide complessivamente in progettazione dei dati e progettazione delle applicazioni. Nella prima si individua la struttura e l'organizzazione che i dati dovranno avere, nella seconda si definiscono le qualità peculiari dei programmi applicativi. Per la struttura dei dati si potrà vedere più avanti, continuando a leggere il testo, ch'essa prenderà corpo nella fase finale della progettazione logica e farà riferimento ad un *modello logico dei dati*. Mentre per i programmi applicativi si è pensato di utilizzare un RDBMS (Relational Database Management System) come MySQL, ovvero un sistema per la gestione di basi di dati relazionali, che non è altro che un software che implementa il linguaggio SQL (Structured Query Language) utilizzato per definire, interrogare ed apportare modifiche alla base di dati. L'implementazione in PHP (PHP: Hypertext Preprocessor) permetterà invece d'interrogare il server, quindi direttamente la base di dati per mezzo di pagine web dinamiche; questo sarà lo strumento principe degli utilizzatori finali.
- **Implementazione.** Si tratta della realizzazione del sistema informativo secondo la struttura e le caratteristiche definite durante la fase di progettazione. Viene costruita e popolata la base di dati e viene prodotto il codice dei programmi. Un software utile è stato phpMyAdmin ch'è un'applicazione scritta in linguaggio PHP che permette di amministrare la base di dati di MySQL per mezzo di un'interfaccia browser.
- **Validazione e collaudo.** Serve a verificare il corretto funzionamento e la qualità del sistema informativo. Verrà inizialmente testato un prototipo in laboratorio interagendo con esso e ponendolo sotto stress per vedere se adempie in modo esaustivo alla propria funzione: fase alfa. Successivamente, avendo dato risultati accettabili, un ulteriore prototipo di test verrà posto per un certo periodo ai fruitori definitivi che suggeriranno e daranno le loro impressioni: fase beta.
- **Funzionamento.** In questa fase il sistema informativo diventa operativo ed esegue i compiti per i quali era stato originariamente progettato.

E' da precisare che il processo non è quasi mai strettamente sequenziale in quanto spesso, durante l'esecuzione di una delle attività citate, bisogna rivedere decisioni prese nell'attività precedente. Quello che si ottiene è proprio un ciclo di operazioni.

La progettazione e costruzione di un tale progetto si sono potute realizzare grazie alle conoscenze apprese durante il corso triennale in ingegneria informatica. Se non avessi prestato attenzione a tutti i consigli esposti sui libri avrei sicuramente realizzato un'applicazione *zoppa* che non avrebbe risposto ai canoni qualitativi dell'ingegneria del software. La scelta di avvalersi del binomio MySQL-PHP è dovuto dalla facilità di utilizzo, dalla capacità di adattamento del codice prodotto oltreché dalla popolarità di questo connubio che offre degli standard di riuscita sempre elevati; ricordando ch'è un software open source per cui libero di poter essere utilizzato ed indipendente perché svincolato dalle grosse aziende software che invece ripiegano sul profitto.

2.1.2 Metodologie di progettazione e basi di dati

Un aspetto che vale la pena di precisare è che cosa si intende per *metodologia di progettazione* e quali sono le proprietà che una metodologia deve garantire. Una metodologia di progettazione consiste in:

- una *decomposizione* dell'intera attività di progetto in passi successivi dipendenti tra loro,
- una serie di *strategie* da seguire nei vari passi e alcuni *criteri* per la scelta in caso di alternative,
- alcuni *modelli di riferimento* per la descrizione dei dati di ingresso e uscita delle varie fasi.

Le proprietà che una metodologia deve garantire sono principalmente:

- la *generalità* rispetto alle applicazioni e ai sistemi in gioco,
- la *qualità del prodotto* in termini di correttezza, completezza ed efficienza rispetto alle risorse impiegate,
- la *facilità d'uso* sia delle strategie che dei modelli di riferimento.

Nell'ambito delle basi di dati, si è consolidata negli anni una metodologia di progetto che ha dato prova di soddisfare pienamente le proprietà descritte. Tale metodologia è articolata in tre fasi principali da effettuare in cascata e si fonda su un principio dell'ingegneria semplice ma molto efficace: separare in maniera netta le decisioni relative a cosa rappresentare in una base di dati (prima fase), da quelle relative a come farlo (seconda e terza fase).

2.1.3 Progettazione concettuale

Il suo scopo è quello di rappresentare le specifiche informali della realtà di interesse in termini di una descrizione formale e completa, ma indipendente dai criteri di rappresentazione utilizzati nei sistemi di gestione di basi di dati. Il prodotto di questa fase viene chiamato *schema concettuale* e fa riferimento a un *modello concettuale* dei dati. I modelli concettuali ci consentono di descrivere l'organizzazione dei dati a un alto livello di astrazione, senza tenere conto degli aspetti implementativi. In questa fase infatti, il progettista deve cercare di rappresentare il *contenuto informativo* della base di dati, senza preoccuparsi né delle modalità con le quali queste informazioni verranno codificate in un sistema reale, né dell'efficienza dei programmi che faranno uso di queste informazioni. Il prodotto finale della progettazione concettuale di una base di dati consiste nella realizzazione di uno schema Entità-Relazione in grado di descrivere al meglio le specifiche sui dati di un'applicazione.

Per poter avere una corretta interpretazione e lettura dello schema introduciamo una piccola parentesi sulle nozioni che lo compongono. Il modello Entità-Relazione è un modello concettuale dei dati, e come tale, è costituito da un raggruppamento di strutture, chiamati costrutti, utilizzati per rappresentare la realtà d'interesse svincolandoci dall'organizzazione finale che i dati assumeranno una volta posti nell'elaboratore. Questi costrutti si adoperano per la creazione degli schemi i quali descrivono l'organizzazione e la struttura delle occorrenze dei dati, ovvero, il valore che essi assumeranno al variare del tempo. Di seguito esplicheremo la loro rappresentazione e forma:

- **Entità.** Sono gli oggetti fondamentali dello schema e rappresentano delle classi di un oggetto con proprietà comuni ed esistenza *indipendente*. Le entità vengono raffigurate da un rettangolo.
- **Relazioni.** Costituiscono dei collegamenti logici che intercorrono tra due e più entità. Assumono, nello schema, la forma geometrica del rombo.
- **Attributi.** Descrivono delle proprietà elementari caratteristiche dell'entità o della relazione. Ad ogni attributo è associato ad un dato che fa riferimento ad un insieme, detto dominio, dell'entità (o della relazione), che contiene i valori attendibili per l'attributo. Nel nostro schema assumono la forma ovale.
- **Cardinalità e identificatori.** Costituiscono dei vincoli d'integrità sui costrutti visti in precedenza. Proprietà per cui ogni occorrenza di entità o di relazione devono adempiere per poter essere considerate *vere*.
- **Cardinalità delle relazioni.** Vengono specificate per ogni partecipazione di un'entità ad una relazione e stanno ad indicare il numero minimo e massimo di occorrenze di relazione a cui un'occorrenza dell'entità può partecipare. Dicono in altre parole quante volte, in una relazione tra entità, un'occorrenza di una di queste entità può essere legata ad occorrenze dell'altra entità coinvolte. Unico vincolo: la cardinalità minima dev'essere minore o uguale alla cardinalità massima.
 - per cardinalità minima, zero o uno si definisce rispettivamente, come una partecipazione opzionale e partecipazione obbligatoria;
 - per cardinalità massima, uno o molti (N) viene vista rispettivamente come una funzione che associa ad un'occorrenza dell'entità una sola dell'altra entità che partecipa alla relazione; nel secondo c'è invece un'associazione con un numero arbitrario d'occorrenze dell'altra entità.

Nello schema la cardinalità è rappresentata tra parentesi, una virgola separa il valore minimo dal massimo. È posizionata ai bordi delle entità che si connettono alle relazioni.

- **Identificatori dell'entità.** Permettono di identificare in maniera univoca le occorrenze delle entità. In molti casi, uno o più attributi di un'entità sono adeguati ad individuare un identificatore: si parla allora di identificatore interno (prende il nome anche di chiave). L'attributo chiave è usuale rappresentarlo nello schema sottolineandone il nome.
- **Generalizzazioni.** Rappresentano legami logici tra un'entità E, detta entità *genitore*, ed una o più entità dette entità *figlie*, in cui E generalizza e comprende tutte le sue discendenti logicamente attinenti. Nel nostro schema viene rappresentato da un triangolo con etichetta Gen che indica appunto la specializzazione delle entità *figlie* rispetto all'entità *genitore*.

La costruzione finale dello schema proviene da una serie di progressive fasi di raffinamenti e decisioni che vengono modellate in base alle necessità degli utenti che dovranno utilizzare la base di dati. Per individuare queste esigenze, prima di iniziare con la progettazione vera e propria, è doveroso

raccogliere ed analizzare i requisiti. Questa fase non è del tutto indipendente da quella della progettazione, ma possiamo affermare che procede per molti aspetti in modo parallelo. Possiamo infatti iniziare a costruire uno schema E-R quando non abbiamo ancora terminato di raccogliere e realizzare tutti i requisiti, per poi arricchirlo progressivamente man mano che le informazioni in nostro possesso aumentano.

Per ottenere uno schema qualitativamente corretto alcune attenzioni devono essere prestate durante la stesura. Innanzitutto deve poter utilizzare i costrutti messi a disposizione dal modello concettuale di riferimento. Bisogna fare attenzione a non incorrere in errori di natura sintattica o semantica. La prima riguarda un uso irregolare dei costrutti, la seconda invece intercorre quando si fa uso di costrutti non conformi alla definizione. Uno schema inoltre è completo se vengono rappresentati tutti i dati d'interesse e quando tutte le operazioni possono essere riportate realmente allo schema descritto. Inoltre è leggibile se i requisiti sono esposti in maniera non artificiosa bensì facilmente comprensibili e naturali. Infine bisogna fare attenzione che il tutto rappresenti una sintesi ben calibrata in ogni sua parte; che sia minimale, con concetti non ripetuti in diverse parti ma sintetizzando ed accorpando le entità dove possibile. Con questo voglio dire che in caso di necessarie ridondanze di concetti, essi vengano riportati sulla documentazione specificandone il motivo. La qualità di uno schema E-R solitamente si verifica per ispezione, partendo da un concetto del diagramma e parallelamente con la documentazione elaborata, si attesta che tutti i contenuti siano stati esposti e rappresentati.

2.1.4 Progettazione logica

Consiste nella traduzione dello schema concettuale definito nella fase precedente, nel modello di rappresentazione dei dati adottato dal sistema di base di dati a disposizione. Il prodotto di questa fase viene denominato *schema logico* della base di dati e fa riferimento ad un *modello logico dei dati*. Come accennato ad inizio paragrafo, la finalità della progettazione logica è quella di ottenere uno schema logico in grado di contenere, in maniera consona e funzionale, tutte le informazioni raccolte nello schema Entità-Relazione realizzato durante la fase di progettazione concettuale. Per poter ottenere questo obiettivo sarà opportuno prendere diverse decisioni che riguarderanno: la traduzione dello schema E-R, l'aspetto delle ridondanze e l'ottimizzazione degli indici di prestazione in base a delle operazioni previste.

La semplificazione dello schema E-R è indipendente dal modello logico adottato, ma necessaria per via dell'utilizzo di particolari costrutti, come ad esempio le generalizzazioni, che sono soggette a diverse possibilità di traduzione. In gergo ingegneristico questa fase prende il nome di **ristrutturazione dello schema Entità-Relazione** che oltretutto perfeziona e riorganizza tutta la struttura rendendo più agevole il passaggio alla fase successiva. In cascata seguirà la **traduzione verso un modello logico**, nel nostro caso utilizzeremo il modello relazionale che produrrà con le dovute modifiche e decisioni di realizzazione, la documentazione finale per l'attuazione della base di dati. Come noto, un modello logico ci consente di descrivere i dati secondo una rappresentazione ancora indipendente da dettagli fisici, ma concreta perché disponibile nei sistemi di gestione di base di dati.

2.1.5 Progettazione fisica

In questa fase lo schema logico viene completato con la specifica dei parametri fisici di memorizzazione dei dati (organizzazione dei file e degli indici). Il prodotto di questa fase viene denominato *schema fisico* dei dati. Tale modello dipende dallo specifico sistema di gestione di basi di dati scelto e si basa sui criteri di organizzazione fisica dei dati in quel sistema.

Vediamo ora in che maniera i requisiti della base di dati vengono utilizzati nelle varie fasi della progettazione. È bene qui fare una distinzione tra *specifiche sui dati*, che riguardano il contenuto della base di dati, e *specifiche sulle operazioni*, che riguardano l'uso che utenti e applicazioni fanno della base di dati. Nella progettazione concettuale si fa uso soprattutto delle specifiche sui dati mentre le specifiche sulle operazioni servono solo a verificare che lo schema concettuale sia completo, contenga cioè le informazioni necessarie per eseguire tutte le operazioni previste. Nella progettazione logica lo schema concettuale in ingresso riassume le specifiche sui dati, mentre le specifiche sulle operazioni si utilizzano insieme alle previsioni sul carico applicativo, per ottenere uno schema logico che renda tali operazioni eseguibili in maniera efficiente. Infine, nella progettazione fisica si fa uso dello schema logico e delle specifiche sulle operazioni per ottimizzare le prestazioni del sistema. In questa fase bisogna anche tenere conto delle caratteristiche del particolare sistema di gestione di basi di dati utilizzato.

L'esito finale della progettazione di una base di dati non è solo lo schema fisico, ma è composto anche dallo schema concettuale e dallo schema logico. Nel particolare, lo schema concettuale fornisce infatti una rappresentazione della base di dati di alto livello, che può essere molto utili a scopo documentativo, mentre lo schema logico fornisce una descrizione concreta del contenuto della base di dati che, prescindendo dagli aspetti implementativi, può essere utile come riferimento per le operazioni di interrogazione e aggiornamento. A partire dai requisiti rappresentati da documenti e moduli di vario genere, acquisiti anche attraverso l'interazione con gli utenti, viene costruito uno schema Entità-Relazione (rappresentato da un diagramma) che descrive a livello concettuale una base di dati. Questa rappresentazione viene poi tradotta in uno schema relazionale, costituito da una collezione di tabelle. Infine, i dati vengono descritti da un punto di vista fisico (tipo e dimensione dei campi) e vengono specificate strutture ausiliarie, come gli indici, per l'accesso efficiente ai dati.

2.1.6 Accenno alle architetture web-oriented

Per le applicazioni basate sui Web Service, in inglese Service-Oriented Architecture (SOA), si fa riferimento ad un sistema composto da un'architettura software adatta a supportare l'uso di servizi in ambito Web; una sorta di sistema distribuito con vari software indipendenti che devono collaborare per svolgere determinati compiti. Il tutto permette l'interoperabilità tra diversi sistemi così da consentire l'utilizzo delle singole applicazioni come componenti del processo di business e garantisce le richieste degli utenti in modo integrato e trasparente. Praticamente, queste applicazioni partecipano condividendo scambi di dati provenienti e realizzati da diverse altre applicazioni esistenti. Ovvero: ogni servizio mette a disposizione una propria funzionalità che unita a quelle disponibili dagli altri servizi, permettono di realizzare applicazioni di maggiore complessità.

Occorrono quindi delle regole che creino degli standard a livello applica-

tivo e comunicativo adoperando un unico protocollo che riesca a dialogare tra le chiamate ricevute e la componente applicativa, indipendentemente dal modello utilizzato per il trasporto dei dati; in modo che sia completo e sicuro.

L'architettura service-oriented è caratterizzata principalmente da questa prerogativa: la separazione del servizio dalla sua interfaccia, in un certo modo separa il *cosa* fare dal *come* farlo.

Vediamo ora alcune caratteristiche di una SOA:

- ogni servizio ha un suo ruolo ben definito; dev'essere autonomo dal contesto e dagli altri servizi.
- avere scarse probabilità di accoppiamento con altri servizi. Un'architettura così strutturata si verifica quando le sue componenti sono dipendenti tra loro in maniera limitata. Il sistema è più flessibile e modificabile.
- il servizio dev'essere agilmente ricercabile in base alla sua interfaccia garantendo un tempo di esecuzione rapido.
- composto da un'interfaccia ed autonomo dalla sua realizzazione. Il linguaggio di programmazione utilizzato per la creazione delle componenti utili a generare il servizio, è del tutto indipendente dalla piattaforma e dal sistema operativo in cui esegue la sua attività il servizio.
- una migliore reperibilità sulla rete attraverso la pubblicazione della sua interfaccia in un Service Directory o Service Registry. In modo che siano trasparenti le modalità d'accesso e i requisiti richiesti per aderire al servizio.
- utilizzare un'interfaccia implementata che garantisca poche funzioni e sicure. Questo permette di non avere programmi sofisticati di controllo che ne limitino l'interoperabilità tra i servizi. Questi ultimi infatti devono poter relazionare attraverso lo scambio di messaggi per mezzo di un formato standard: *Platform Neutral*. Il contenuto dei messaggi scambiati solitamente riguarda i risultati delle elaborazioni oppure informazioni utili al coordinamento tra i servizi.
- essere realizzato in modo da essere riutilizzabile in futuro. Proprio la separazione e libertà di ogni servizio dai restanti garantisce l'univocità di una determinata attività. Per l'appunto, nell'architettura SOA le applicazioni sono la conseguenza dell'elaborazione di più servizi. Servizi o applicazioni più elaborati rispetto a quelli fondamentali prendono il nome di *Service Orchestration*.

In un certo senso il nostro sistema effettua qualcosa del genere, ma del suo funzionamento se ne parlerà nella sezione 4.2 a pagina 28, quando si andrà a verificare il software in esercizio.

3 | PROGETTO

3.1 LA PROGETTAZIONE CONCETTUALE

3.1.1 Raccolta dei requisiti

I requisiti di un'applicazione provengono, nella maggior parte dei casi, da diverse fonti. Le principali sono:

- Gli *utenti della applicazione*. In questo caso le informazioni si acquisiscono mediante opportune interviste, anche ripetute, oppure attraverso una documentazione scritta che gli utenti possono aver predisposto appositamente per questo scopo.
- Tutta la *documentazione esistente* che ha qualche attinenza con il problema allo studio: moduli, regolamenti interni, procedure aziendali, normative. È richiesta in questo caso una attività di analisi, raccolta e revisione della documentazione trovata da parte del progettista della base di dati in comune accordo con gli utenti finali che assisteranno e supervisioneranno le decisioni.
- Eventuali *realizzazioni preesistenti*, ovvero applicazioni che si devono rimpiazzare o che devono interagire in qualche maniera con il sistema da realizzare.

Il mio lavoro è partito inizialmente dal secondo punto quindi da una *documentazione esistente* considerato che era già in progetto in Dipartimento di Storia delle arti visive e della musica, ormai da diversi anni, un sistema simile, ancora da realizzare. Analizzando gli incartamenti si evinceva che il sistema era predisposto e limitato solo alle immagini prodotte durante l'attività del laboratorio fotografico. La concezione originaria era proprio quella di una base di dati di immagini rivolta alla catalogazione ed archiviazione delle stesse, con alcune funzionalità da implementare come:

1. la marcatura digitale contro la violazione della legge sul diritto d'autore (watermark digitale);
2. la possibilità di produrre raggruppamenti ordinati di immagini in collezioni, con la possibilità della rappresentazione di confronti multipli tra immagini;
3. esportazione delle collezioni di immagini, con o senza descrizione di scheda di catalogo, nella forma di un insieme ordinato di file predisposti per la videopresentazione o già nella forma di videopresentazione eseguibile o nel formato MS-Powerpoint o in formato HTML (videopresentazione per didattica frontale e ricerca);
4. visualizzazione delle collezioni di immagini direttamente nella forma di videopresentazione tramite rete locale;

5. accesso in ricerca circoscritto alle collezioni d'immagini e sull'intero database dalle postazioni protette destinate agli studenti presso il Dipartimento (visualizzazione delle immagini mostrate a lezione; ricerca libera).

La rilettura dei documenti esistenti, mi ha permesso di rivedere il tutto sotto un'ottica diversa. Mantenendo fermi i criteri sopra esposti, si è ritenuto iniziare coerentemente con quanto prevede la progettazione concettuale per cui con delle interviste agli *utenti dell'applicazione*, attuandole dal mese di ottobre 2011 sino a marzo 2012.

Nel seguente elenco, posto in ordine cronologico dal più lontano al più recente, è riportato il personale strutturato nel Dipartimento di Beni Culturali, cui è stato sottoposto ad intervista:

- Giovanna Valenzano: Direttore del Dipartimento nonché professore ordinario in *Storia dell'arte medievale*;
- Michele Barollo: funzionario tecnico nel *Laboratorio audiovisivi*;
- Guido Bartorelli: ricercatore in *Storia dell'arte contemporanea*;
- Farah Polato: ricercatore in *Storia e Critica del cinema*;
- Antonio Lovato: professore associato in *Storia della musica medievale e rinascimentale*.

In appendice [A.1](#) a pagina [33](#) sono presenti le domande poste agli interlocutori. Ovviamente quest'ultime sono servite solamente come linee guida per intavolare poi un discorso più approfondito che man mano ha messo in luce dei particolari utili per la progettazione della base di dati.

Riassumiamo in poche parole le aspirazioni ed il contributo che hanno accumulato i diversi colloqui, come:

- la necessità di un proprio spazio personale e riservato dove poter scegliere, ricercare ed inserire i propri file a piacimento;
- tutti i docenti sono stati entusiasti del sistema a patto che esso sia allargato all'interazione e alla condivisione con gli studenti;
- i frequentanti potranno trovare le lezioni del docente accedendo con opportune credenziali solo da postazioni presenti nel Dipartimento dei Beni Culturali, preferibilmente quelle posizionate in biblioteca;
- verranno utilizzati dei *tag* per etichettare e ricercare le opere. I *tag* non sono altro che delle definizioni, parole semplici o composte, adoperate per selezionare un elemento d'interesse. Forse risulterà più semplice vederle come una *parola chiave* che permette di ricercare degli elementi comuni ad essa;
- si è riscontrato che raramente i docenti hanno bisogno di un'unica immagine (o file), bensì sono soliti a lavorare con un gruppo di file che abitualmente metteranno a confronto;
- possibilità di strumenti per poter modificare i file eseguendo delle azioni come: salvare un ingrandimento di una sezione, creare dei PDF o dei PowerPoint;

I tecnici di laboratorio fotografico, i quali si trovano in accordo per gran parte delle indicazioni sopra esposte, hanno comunque la priorità di compiere delle azioni sul sistema in progetto; pertanto anche se il loro accesso è proiettato con altre finalità, hanno espresso la necessità di:

- tabelle ed attributi aggiuntivi legate alla descrizione degli oggetti digitali, in modo che essi possano inserire i dati delle apparecchiature fotografiche utilizzate durante le riprese esterne o interne al laboratorio;
- accedere al sistema con delle interfacce semplici ed efficaci in modo da popolarlo rapidamente.

Per poter eseguire quanto sopra citato i fotografi avranno bisogno di far parte degli utenti che accedono al sistema, per questo essi disporranno di un'interfaccia differente rispetto a quella dei docenti o degli studenti. Questa distinzione di *vista esterna* è dovuta dai ruoli che ogni gruppo di utenti ricopre e permette a ciascuno di essi di beneficiare della base di dati per la posizione ricoperta, nascondendo il resto.

3.1.2 Schema Entità-Relazione

Il modello proposto a pagina 16 riassume gli aspetti salienti della base di dati in progetto. Prima di analizzare lo schema minuziosamente vale spendere due parole per introdurlo. Esso ha la peculiarità di rappresentare tutti i concetti protagonisti del sistema in un diagramma molto semplice ed intuitivo nell'interpretazione. Prendiamo in considerazione, ad esempio il concetto di *opera*. Essa sarà opportuno rappresentarla per mezzo del costrutto entità, visto che il ruolo ricoperto è significativo nello schema d'interesse, considerando che essa abbraccia varie discipline della storia dell'arte: pittura, scultura, cinema, musica, eccetera. Avrà dei legami con l'autore che l'avrà creata e con dei file che descrivono il tipo di sorgente con cui giunge a noi: immagine, audio, filmato, eccetera. Dal canto suo l'immagine potrà essere realizzata, poiché ripresa dal vero, dai tecnici o meno ecco perché si dirama in fonte diretta e in fonte indiretta. I docenti, particolarizzati per ruolo dalle generalizzazioni presenti, avranno l'esigenza di creare delle cartelle da condividere con gli studenti e preleveranno i file dal sistema per spostarli in una propria cartella, dopo aver predisposto le dovute modifiche ed opportunamente creato i PowerPoint e/o i PDF. Essi potranno, qualora lo ritenessero opportuno, anche descrivere per mezzo di parole chiave qualche opera, per essere più fruibile nella ricerca successiva. I protagonisti del sistema oltre ai docenti ed agli studenti saranno anche i tecnici che accederanno per popolare con dei file la base di dati oppure eseguire delle operazioni organizzative inerenti alla produzione fotografica. L'accesso degli utenti viene fatto per mezzo di un login e sfruttando le credenziali di uno username e di una password.

Verranno utilizzati i costrutti analizzati nel paragrafo relativo alla Progettazione concettuale esperessi nel paragrafo 2.1.3 a pagina 7 in abbinamento alle specifiche sopra descritte. Iniziando da un'entità fulcro e procedendo da questo concetto si è proseguito nella stesura dello schema a *macchia d'olio*, riferendosi alle specifiche, sino a toccare tutte le richieste più lontane. La nostra entità di partenza è stata OPERA intesa in modo molto generale

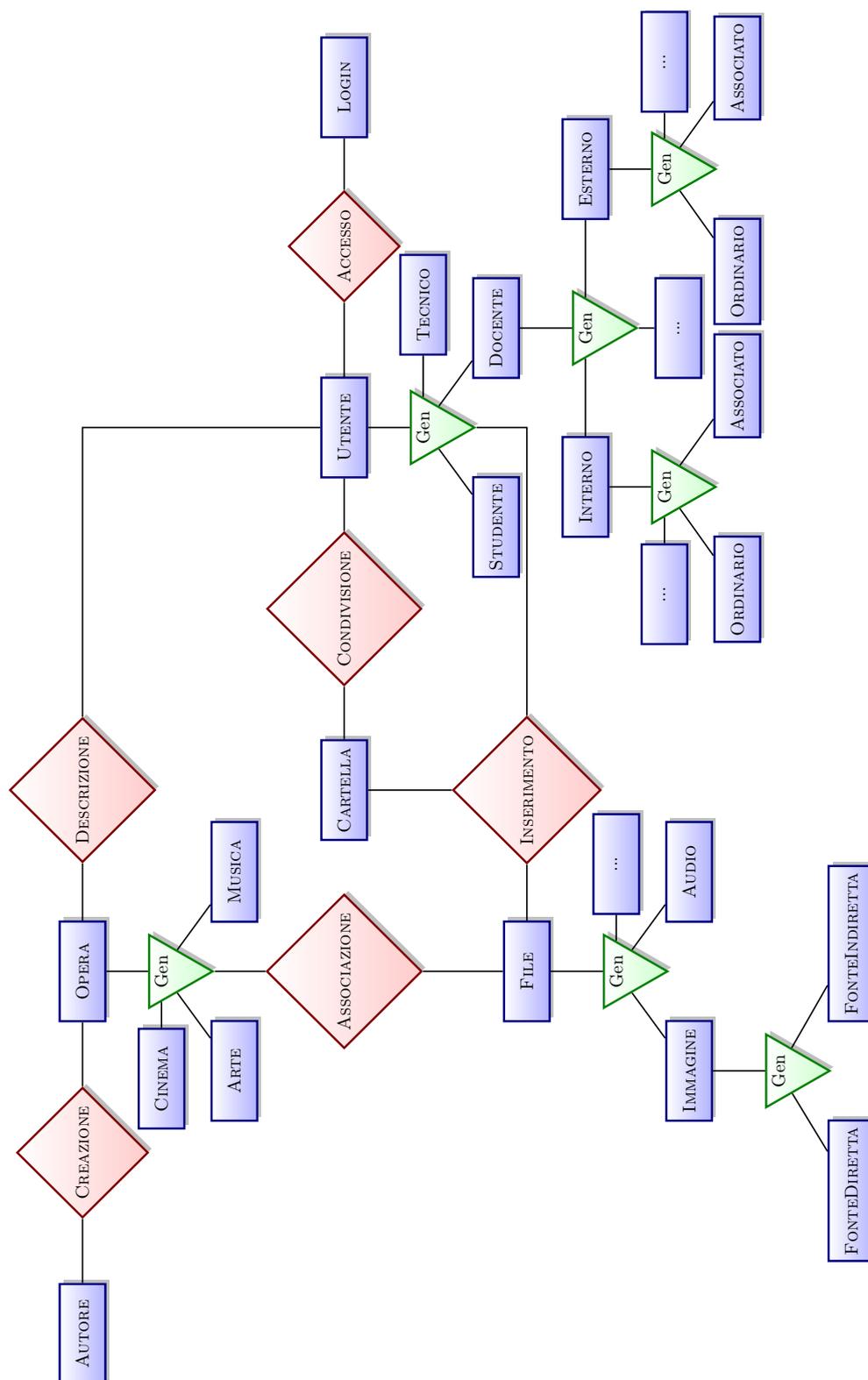


Figura 1.: Schema Entità-Relazione

come Opera d'arte. Ovviamente questo concetto risulta un po' generico e restrittivo perché le opere d'arte possono risiedere in diversi campi artistici ecco perché da quest'entità *genitore* si ripartono altre entità *figlie*: ARTE,

MUSICA e CINEMA ramificate dal triangolino verde con denominazione Gen in abbreviazione di generalizzazione. Per maggior chiarezza le entità che riportano dei puntini di sospensione stanno ad indicare, ogni qualvolta le s'incontrerà nel diagramma, la presenza di altre entità *figlie* non esprimibili per problemi di organizzazione dello spazio.

Ogni opera viene realizzata da un autore. L'entità AUTORE è legata logicamente all'entità OPERA per mezzo del costrutto relazione ch'è rappresentato da un rombo: nel nostro caso prenderà il nome di CREAZIONE. La relazione, o associazione, può unire due o più entità, questo se viene richiesto dai criteri di progettazione. Tornando a parlare dell'entità OPERA quest è associata all'entità FILE la quale si ripartisce in: IMMAGINE, AUDIO, VIDEO e DOCUMENTO, rispettivamente le diverse tipologie di file che i docenti manipolano. Vi è stata la necessità di scindere l'entità IMMAGINE in due entità *figlie*, per specializzare la provenienza dell'immagine, se acquisita direttamente fatto delle riprese fotografiche in esterna o se pervenute in un'altra maniera (depliant, manifesti, fotocopie, eccetera) e quindi di conseguenza: FONTE DIRETTA e FONTE INDIRETTA. In realtà questo si applicherebbe anche alle registrazioni, se dal vivo o prese da CD; comunque è una necessità che non è emersa dall'analisi dei requisiti.

Il file una volta inserito oppure ricercato nella base di dati dall'utente, dovrà essere posto in un proprio spazio personale per essere adoperato, ecco perché la presenza dell'associazione INSERIMENTO che unisce le tre entità: FILE, CARTELLA e UTENTE. L'utente potrà, a sua volta, essere d'accordo se far usufruire o meno agli altri utenti una parte delle proprie risorse; questo passaggio è raffigurato dall'entità UTENTE che si unisce alla CARTELLA per mezzo della relazione CONDIVISIONE. UTENTE raffigura una classe troppo generica per questo è stata specializzata in: STUDENTE, DOCENTE e TECNICO; tre figure protagoniste che con diverse utilità e mansioni usufruiranno dell'applicazione. L'entità DOCENTE racchiude molte realtà per questo vi è la dicotomia nel distinguerli in INTERNO ed ESTERNO rispettivamente se afferisce al Dipartimento dei Beni Culturali o meno. Le restanti generalizzazioni specificano la modalità di contratto che lega i docenti, interni o esterni che essi siano, al dipartimento; vengono distinti dalle entità come: ORDINARIO, ASSOCIATO, RICERCATORE o CONTRATTISTA.

LOGIN rappresenta l'accesso alla piattaforma da parte di un utente, infine, DESCRIZIONE è un'associazione che attraverso l'utilizzo di una parola chiave (o tag), inserita dall'utente, qualifica una determinata opera in modo che possa essere ricercata con più facilità nel database.

3.2 LA PROGETTAZIONE LOGICA

3.2.1 Ristrutturazione dello schema E-R

Utilizzando lo schema concettuale indicato a pagina 22, sono state avviate una serie di analisi e considerazioni per snellire e rendere più fluide le future richieste che dovranno essere poste nella base di dati. Naturalmente si è andata a modificare la struttura iniziale dello schema E-R, lasciando comunque lo spazio a nuove e future implementazioni che potranno aggiungersi nel corso del tempo, senza stravolgere la struttura portante della base di dati. Per la ristrutturazione dello schema E-R si è seguito il seguente criterio:

- per trasformare le generalizzazioni e farle figurare come legami tra entità e relazioni, si è scelta la soluzione per cui tutte le entità *figlie*

dovranno accorparsi nel *genitore*. In questo modo verranno introdotti degli attributi appositi che riguarderanno il diverso *tipo* di categoria a cui si farà riferimento;

- l'entità *UTENTE* si riduce ad assumere solo il ruolo dell'entità *DOCENTE* eliminando *TECNICO* e *STUDENTE*. La scelta non è tanto drastica perché in futuro sarà comunque possibile tornare a riproporre queste entità, ampliando il progetto con nuove funzionalità e compiti. Piuttosto ora permette innanzitutto di focalizzare il ruolo centrale del docente, utilizzatore principale del progetto, ma non esclusivo dato che i tecnici e gli studenti continueranno ad interagire con l'applicazione. Tenere traccia di un tecnico o di un amministratore attraverso delle interrogazioni rivolte all'interno della base di dati non ha alcun senso per la finalità stessa della realizzazione locale; sarebbe stato più lecito pre-supportarlo se queste figure interagissero da diverse sedi remote. In tal caso si potrebbe benissimo risalire all'autore di eventuali operazioni. I tecnici che lavorano ad un livello inferiore nella base di dati, potranno comunque accedere al sistema e fare i propri inserimenti e modifiche o più semplicemente manipolare i dati eseguendo delle statistiche .

La presenza dell'entità *STUDENTE* porta con se diverse complicazioni. Nello specifico basti pensare alla gestione di tutti gli accessi con l'immissione di un proprio username e password per decine e centinaia di studenti, potrebbe comportare un sovraccarico di attività di gestione ed anche di spazio in memoria nell'amministrarli, solo per verificare l'origine di un attacco informatico. L'ammissione al sistema da parte degli studenti avverrà quindi dalle postazioni informatiche collocate all'interno del circuito dipartimentale, per cui protette da proxy e limitate nelle azioni. Sarà poi compito del docente sulla scelta di condividere o meno, in base a criteri di riservatezza, la propria cartella di file con gli studenti o con i propri colleghi;

- conseguenza della scelta esposta nel punto precedente, è l'eliminazione della dell'entità *LOGIN*. I suoi attributi principali come utente e password riguarderanno solo il corpo docente; pertanto figureranno nella loro relativa nell'entità;
- a *DOCENTE* sono state collegate altre informazioni per mezzo delle entità *INSEGNAMENTO* e la conseguente *CORSODILAUREA*. Per via della cardinalità di partecipazione (0,1) in entrambi i lati di *INSEGNAMENTO*, quest'ultima accoglierà in forma tabellare, oltre ai dati propri anche il riferimento al docente ed al corso di laurea inerente. La precedente filosofia di pensiero è stata adottata anche per l'associazione *DESCRIZIONE* la quale, promossa ad entità, ci permetterà di risalire al docente che ha inserito un tag per riferirsi ad un'opera;
- La generalizzazione di *IMMAGINE* ha subito una *mutazione*. Dopo essere stata integrata nell'entità *FILE*, come esposto nel primo punto, per via della distinzione originaria tra *FONTEDIRETTA* e *FONTENDIRETTA*, è stato opportuno sottolineare solo la prima voce di queste due entità rinominandola in *DATIRIPRESA* e sganciandola dal contesto, creando così un'entità a se stante. Ad usufruire di questa ripartizione saranno esclusivamente i fotografi, colleghi del laboratorio in cui lavoro. Essi potranno utilizzare la base di dati per inserire le informazioni relative ad una ripresa esterna in comune accordo col docente e per questo finalizzata all'attività didattica del Dipartimento. Ovviamente

gli elementi per descrivere una fonte indiretta, come documenti, libri o letteratua grigia che sia, sono presenti nell'entità **IMMAGINE**;

- l'ultimo tassello della ristrutturazione dello schema Entità-Relazione riguarda la presenza della classe **COPYRIGHT**. Come suggerisce il nome, quest'ultima raccoglie gli enti che hanno la facoltà e il diritto di riproduzione sui file dell'opera.

Le piccole ellissi, ricordiamolo, sono gli attributi e rappresentano i descrittori delle entità e delle associazioni. Di questi si è tenuto conto di menzionare solo i più importanti includendo l'attributo chiave; i restanti attributi sono presenti in modo completo nel *dizionario dei dati*. Infatti la documentazione dei vari concetti rappresentati in uno schema è meglio ordinarla in due tabelle: la prima descrive le entità dello schema con il nome, una definizione libera in linguaggio informale, l'elenco di tutti gli attributi ed i possibili identificatori. L'altra tabella descrive le relazioni con il nome, una loro spiegazione non ufficiale, l'elenco degli attributi e l'elenco delle entità coinvolte insieme alla loro cardinalità di partecipazione. Entrambe sono presenti rispettivamente a pagina 19 con riferimento alla tabella che riguarda le entità ed a pagina 23 per la tabella relativa alle relazioni. Ribadiamo inoltre, per completezza che i numeri posti tra parentesi indicano la partecipazione di un'entità ad una relazione. In base a tale numero si avrà una traduzione al modello relazionale differente.

Tabella 1.: Dizionario dei dati: entità

Entità	Descrizione	Attributi	Identificatore
Autore	Autore che crea l'opera	ID, Nome, Cognome, NomePubblico, AnnoNascita, LuogoNascita, AnnoMorte, LuogoMorte, DateCerte	ID

Continua nella prossima pagina

Continua dalla pagina precedente

Entità	Descrizione	Attributi	Identificatore
Opera	Opera d'arte	ID, TipoOpera, Titolo, TitoloOriginale, TitoloInternazionale, Tecnica, Anno, Collocazione, AreaCompetenza, Altezza, Larghezza, Profondità, Durata, GenereFilm, AutoreFotografia, AutoreScenografia, FormaMusicale, Tonalità, IdentificativoOpera, PrimaEsecuzione, Organico, InformazioniLibretto, Incipit, IncipitModerno, DataCreazione, DataUltimaModifica	ID
File	File in cui è contenuta l'opera d'arte	ID, PathName, Nome, Estensione, AltezzaPunti, LarghezzaPunti, Durata, DimensioneByte, ProfonditaColore, SpazioColore, Campionamento_dpi, Campionamento_hertz, Campionamento_fps, Bit-Rate, CalibrazioneColore, Esecutore, CanaliAudio, SistemaCodifica, FonteEdita_Titolo, FonteEdita_ISBN, FonteEdita_Editore, FonteEdita_Curatore, FonteEdita_Collana, FonteEdita_NumInvBibl, FonteEdita_Proprietario, FonteAntica, Descrizione	ID

Continua nella prossima pagina

Continua dalla pagina precedente

Entità	Descrizione	Attributi	Identificatore
DatiRipresa	Informazioni sulla ripresa diretta	ID, DispositivoUtilizzato, Obiettivo, TipoIlluminazione, Filtro, ModalitàEsposimetrica, LetturaEsposimetrica, Tempo, Risoluzione, QualitàRipresa, QualitàComposizione	ID
Copyright	Ente proprietario che cura i diritti del file	File, TipoDiDiritto, EnteProprietario, Referente, Indirizzo, Telefono, Cellulare, Fax, Mail	File
Cartella	Spazio da utilizzare per condividere e gestire i propri file	ID, Nome, Descrizione, Docente, NumeroAccessi, NumeroFilePresenti, DataCreazione, DataUltimoAccesso, TipoAccesso	ID
Docente	Docente che utilizza il database	Matricola, Cognome, Nome, Qualifica, Login, Password, UltimoAccesso	Matricola
Descrizione	Descrizione di un'opera tramite tag	Docente, Opera, Tag	ID
Insegnamento	Materia d'insegnamento	Docente, CorsoDiLaurea, Codice, Nome	ID
CorsoDiLaurea	Corso di laurea legato all'insegnamento	Nome, Descrizione	Codice

Si conclude dalla pagina precedente

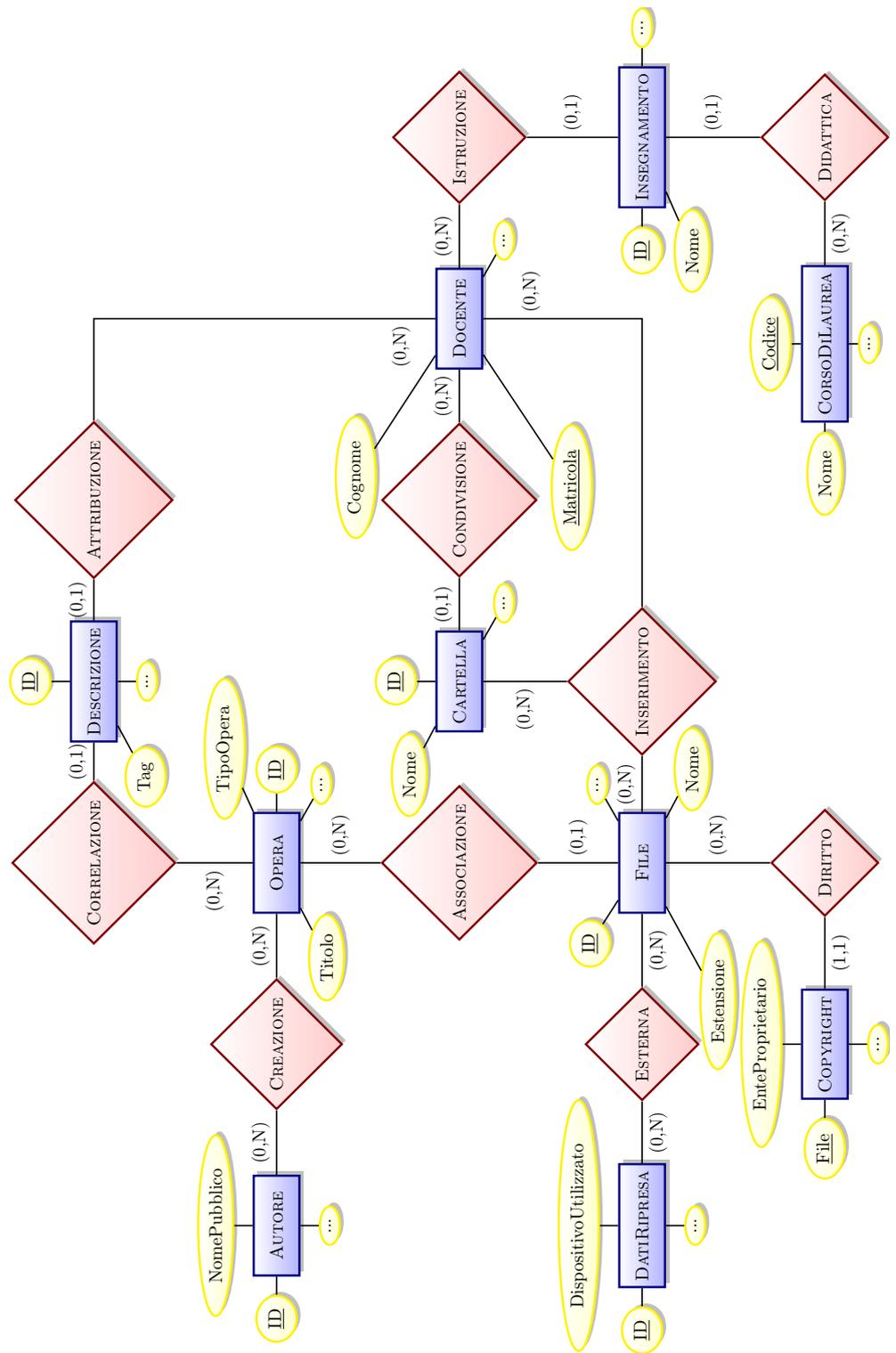


Figura 2.: Schema Entità-Relazione: ristrutturato

Tabella 2.: Dizionario dei dati: relazioni

Relazione	Descrizione	Entità Coinvolte	Attributi
Creazione	Associa un autore alla propria opera	Autore (0,N), Opera (0,N)	
Associazione	Associa un un'opera ad un file	Opera (0,N), File (0,1)	DataCreazione, DataUltima-Modifica, NumeroVisite
Esterna	Associa una ripresa esterna al file prodotto	File (0,N), DatiRipresa (0,N)	Data, Città, Luogo
Diritto	Associa un file all'ente proprietario con diritto di copyright	File (0,N), Copyright (0,1)	DataAttribuzione, DataScadenza
Inserimento	Associa il file inserito in una cartella da parte di un docente	Cartella (0,N), File (0,N), Docente (0,N)	Data
Condivisione	Associa un docente alla propria cartella	Docente (1,N), Cartella (0,1)	DataCreazione, DataUltimoAccesso
Attribuzione	Associa la descrizione di un file da parte del docente	Docente (0,N), Descrizione (0,1)	
Correlazione	Parola chiave per una determinata opera	Opera (0,N), Descrizione (0,1)	
Istruzione	Associa un docente alla propria attività d'insegnamento	Docente (1,1), Insegnamento (0,N)	
OffertaDidattica	Associa l'insegnamento ad un corso di laurea	Insegnamento (0,1), CorsoDiLaurea (0,N)	Semestre, Anno

4

PROGRAMMAZIONE

4.1 CREAZIONE DEL WEB SERVER

4.1.1 Un po' di storia

L'evoluzione della storia dei calcolatori è quella di creare sistemi centralizzati. Gli attuali servizi *cloud*, sulla nuvola, non sono altro che il susseguirsi di questo cambiamento che mantiene come punto fermo i server. I server non sono altro che particolari computer il cui compito è quello di soddisfare delle richieste che provengono dal web in particolar modo da un utente, chiamato *client*. I server possono essere impiegati per diverse modalità di utilizzo, come: Web server, file server, server di basi di dati, server di applicativi, ed altro ancora e le informazioni viaggiano da un capo all'altro della Terra grazie al protocollo di trasporto HTTP.

Il software utilizzato da piattaforma per offrire i servizi di comunicazione client-server è stato Apache. Questo programma affiancato ad altri due, rispettivamente PHP e MySQL, creano un'integrazione proficua tra i propri moduli che permette di poter essere operativi modificando solo alcune impostazioni dei file. Come anticipavamo nel capitolo 2.1.1 quando trattavamo del *Ciclo di vita dei sistemi operativi*, specialmente nei punti riguardanti la **Progettazione** e l'**Implementazione** si era discusso delle loro caratteristiche: ripercorriamole brevemente. PHP non è un vero e proprio programma, bensì un linguaggio di programmazione votato allo scripting, letteralmente *interpretato* il quale compare all'interno di altri linguaggi molto più elaborati per eseguire delle singole richieste specifiche. Utilizzato principalmente sul lato server, esso permette di generare dinamicamente delle informazioni in HTML. PHP si collega ad un server web il quale dopo aver concluso la creazione di HTML corretto lo spedisce al server web che a sua volta lo trasmette al client richiedente che lo visualizza. L'epiteto *dinamico* tante volte attribuito all'HTML creato da PHP è un falso, perché le pagine HTML sono comunque statiche; dinamiche sono le funzioni che ci permettono di ottenere i risultati richiesti interrogando la base di dati; per cui sarebbe più corretto dire che PHP è dinamico.

Il nostro RDBMS (Relational Database Management System) vale a dire il sistema di gestione di base di dati relazionale di riferimento è MySQL. L'iterazione col RDBMS normalmente avviene per mezzo di comandi testuali impartiti direttamente da riga di comando. È necessario per questo motivo adottare una piattaforma che implementi un'interfaccia grafica che renda più intuitiva la gestione: in gergo informatico si dice *user friendly*. In quest'ottica ci torna comodo utilizzare phpMyAdmin; programma scritto interamente in PHP che ci consente di amministrare la base di dati di MySQL. Questo software prevede la creazione da zero dell'intero database ed introduce diverse funzioni per implementarlo mantenendo fluidità e usabilità nelle iterazioni; inoltre è possibile coordinare per mezzo di relativi permessi l'accesso agli utenti che dovranno semplicemente consultarla o per i professionisti che dovranno ottemperare ad attività di manutenzione e controllo.

4.1.2 Scelte operative

Il documento finale della base di dati inteso come *Schema logico* altro non è che l'unione dello *Schema relazionale*, presente in appendice A.2 a pagina 33, con il Dizionario dei dati ¹, oltre che ad altra documentazione prodotta che completa e supporta la descrizione della base di dati; per esempio i vincoli di entità referenziale che sussistono tra le chiavi delle relazioni saranno presenti nel listato riguardante il codice SQL generato.

Partendo dallo *Schema logico* ed utilizzando da interfaccia grafica il software phpMyAdmin ho provveduto alla realizzazione in toto della base di dati. Il lavoro d'implementazione è stato abbastanza semplice. Infatti phpMyAdmin adotta un sistema d'interazione *user friendly*, intuitivo, che permette con poco sforzo ed utilizzando le dovute accortezze, di mettere insieme gli elementi principali come *tabelle* e *vincoli*, ed edificare la base di dati. Ovviamente lo stesso programma, a lavoro concluso, permette anche di popolare la base di dati. Per gli attributi delle relazioni, o tabelle, contenenti riferimenti numerici INT o alfa-numeric VARCHAR, abbiamo rappresentato il massimo esprimibile dalla applicazione; rispettivamente 11 cifre per il dato di tipo intero INT, il quale è un valore di default quando non viene specificato alcunché e 256 caratteri per il VARCHAR. Con le impostazioni appena descritte occupiamo solo 4 byte a campo per il VARCHAR dipende dalla lunghezza del testo. Ovviamente si poteva avere più accortezza ma attualmente le moderne tecnologie consentono e garantiscono la disponibilità di molto spazio in memoria, inoltre non si hanno limitazioni e pensieri al momento di dover inserire dei dati durante la compilazione dei form. In modo che risulti più chiaro a chi dovrà consultare ed impiegare la base di dati, ogni attributo che popola la relazione è stato affiancato da una descrizione opportuna. Questa permetterà di ottenere informazioni indispensabili sugli elementi che devono essere contenuti senza creare ambiguità nell'immissione.

Ovviamente sono stati rispettati tutti i vincoli del modello relazionale di riferimento, il software esegue già un controllo, per cui certe azioni sono vietate di default. Ricordiamo questi vincoli brevemente:

- *Vincoli di dominio*. Per ogni istanza di relazione, e per ogni tupla² che popola l'istanza, il valore assunto dalla tupla per un generico attributo deve essere un valore atomico del dominio di quell'attributo od il valore nullo.
- *Vincolo di chiave*. Se consideriamo due tuple distinte, in qualsiasi stato della relazione noi confrontiamo tra loro i valori degli attributi di chiave, essi non possono essere uguali; cioè contenere lo stesso valore per entrambe le tuple.
- *Vincolo d'integrità dell'entità*. Impone che nessun attributo di chiave primaria può assumere il valore nullo: null.
- *Vincolo di integrità referenziale*. Il vincolo è realizzabile solo quando due relazioni si legano. Pertanto è opportuno introdurre il concetto di chiave esterna, la quale è il collante che realizza il vincolo di integrità referenziale tra due relazioni: esempio R_1 e R_2 . Se noi prendiamo un insieme di attributi da R_1 e la facciamo riferire ad un insieme

¹ Per il Dizionario dei dati: entità vedi pag.19, relazioni pag. 23

² La tupla è un elenco ordinato di n valori. Ogni valore preso singolarmente riferisce ad un dominio, o al particolare valore null. Molto più grossolanamente potremo dire che un attributo dello schema relazionale corrisponde ad un dominio.

di attributi di chiave primaria con medesimo dominio in R_2 , allora otteniamo un vincolo di integrità referenziale ed R_1 viene detta chiave esterna. Per cui una tupla t_1 di R_1 fa riferimento alla tupla t_2 di R_2 . La cosa importante, forse passata in sordina durante la spiegazione, è che un ipotetico valore di chiave esterna vigente nella relazione R_1 può essere presente o meno, come attributo di chiave primaria nella relazione R_2 .

Passando oltre, la generazione del codice di programmazione in PHP è stata una delle cose un po' più difficoltose da eseguire. Per poter iniziare la stesura è stato opportuno immedesimarsi nell'utente finale, immaginandosi di essere di fronte all'applicazione e di utilizzarla compiendo delle operazioni. In questa fase embrionale mi è stato molto d'aiuto eseguire dei bozzetti su carta, *template*, in modo da avere delle linee guida in cui muovermi e agire. Dopo aver organizzato un po' le idee si è pensato di strutturare il lavoro in cartelle, come esporrò in seguito, non prima di ricordare che tutta la gerarchia fa capo ed è indirizzata in modo che, il web server Apache possa riconoscerla e trattare i file di PHP dal lato server. Questo vuol dire che in successione:

1. viene scritto il programma in modo testuale con un editor;
2. il file viene salvato con estensione .php;
3. si colloca il file nella cartella cui riferisce il server web;
4. dal browser si richiama il programma, quindi il file, digitando il proprio indirizzo di collocazione;
5. il server preleva la pagina chiamata e la spedisce al modulo PHP;
6. il modulo PHP esegue lo script contenuto nella pagina di programma e la traduce in una pagina web al volo;
7. il server invia la pagina web al browser il quale la interpreta e ne permette la visualizzazione.

Come scritto poco sopra, i file PHP sono stati organizzati in cartelle, mantenendo una logica dei nomi e delle disposizioni che ricoprono.

- `index.php`. Questo è un file collocato esternamente dalle directory che seguiranno. Il suo comportamento è più simile a quello di un regista; infatti, dopo aver aperto la connessione col server, è l'unico file ad essere visualizzato dal browser; questo perché si preoccupa di indirizzare alla visualizzazione delle altre pagine del sito in base alle richieste dell'utente.
- `/file`. La cartella contiene materialmente i file multimediali d'interesse per la base di dati; SQL punterà a questa cartella di riferimento per la popolazione della base di dati.
- `/images/service`. La sottocartella `/service` contenuta in `/images` riporta tutte le icone utilizzate nell'applicazione, mentre `/blob` potrebbe essere utilizzata per meglio organizzare il contenuto della cartella `/file` suddividendolo per appartenenza i diversi file di attribuzione ad un documento, piuttosto che ad un video, o ad un suono e così via.

- /config. Contiene le impostazioni per connettersi al web server php-MyAdmin; per cui password di connessione e la direttiva che indica il data base di utilizzo.
- /pages. All'interno di questa cartella sono presenti tutte le pagine del sito. Dalla home, a quella di gestione delle cartelle, a quella della ricerca e della condivisione.
- /templates. Contiene un unico file che gestisce la veste grafica del sito rendendolo tutto uniforme.

4.2 SOFTWARE IN ESERCIZIO

Passiamo ora ad introdurre alcuni aspetti del funzionamento dell'applicazione web server. Ogni utente abilitato all'accesso fornisce, nella pagina principale, il nome del Login e successivamente della password. Qualora non dovesse essere corretto uno dei due campi, e quindi non vi fosse corrispondenza con quanto salvato nella base di dati, l'applicazione interviene visualizzando un messaggio d'errore e quindi di controllare quanto inserito e di riproporre il login. Giusto per puntualizzare, le credenziali d'ingresso saranno fornite all'utente dai gestori sito tramite mail; inoltre quando avviene l'inserimento della password nell'apposito form, l'applicazione esegue una funzione di hashing, MD5 (Message Digest algorithm 5) algoritmo crittografico, che permuta i caratteri in modo che non vengano inviati al server in chiaro. In questo modo si evitano dispiacevoli disservizi come il furto delle password da parte di maleintenzionati o pirati informatici.

Una volta effettuato l'accesso alla piattaforma ci troviamo nella sua pagina principale, ovvero nella home. La pagina presenta in alto a sinistra quattro link: Home, Gestione Cartelle, Profilo e Logout. Il primo link è quello posto di default appena avviene l'accesso all'applicazione; qui si ha la possibilità, come spiegherò poco dopo, di coordinare tutte le azioni future di ricerca. Gestione delle cartelle permette di governare le proprie cartelle, il terzo Profilo rimanda ad una pagina dov'è possibile completare i dati personali del docente ed introdurre l'insegnamento praticato in riferimento al semestre ed al corso di laurea, infine Logout di facile comprensione permette di uscire dall'applicazione. Restiamo ancora nella home esponendo le ultime funzioni; vi è presente un messaggio indicante il riferimento all'ultima sessione d'accesso effettuato, in modo da tenere sott'occhio la possibilità che qualcuno abbia rubato le nostre credenziali entrando di conseguenza al sistema. Ma la peculiarità della home è un'altra, vi è all'interno un campo per poter eseguire delle ricerche sul data base. Inserendo il nome di un'opera, l'autore o una parola chiave presente tra i diversi *tag* si potrà effettuare una ricerca libera che setacci tutti i dati corrispondenti al nostro input; attenzione che l'inserimento della stringa deve contenere più di 3 caratteri, altrimenti non viene avviata nessuna ricerca. La risposta alla *query*, è questo il nome che viene utilizzato per indicare un'interrogazione rivolta alla base di dati, la si ottiene per mezzo di un indice di peso. In pratica l'obiettivo è che il sistema reperisca solamente i dati d'interesse per l'utente, minimizzando il numero di risultati irrilevanti. Il funzionamento è molto semplice: si prende la stringa, ch'è quella che noi digitiamo nel campo di ricerca, la si bonifica da quegli elementi che non sono determinanti ai fini della ricerca come articoli, preposizioni, avverbi e congiunzioni comprendente di punteggiatura e spazi vuoti. La funzione *PulisciQuery()* permette questo risultato.

Ripulita la stringa si otterrà un elenco di parole collocate in un array con la funzione *QueryToArray()*. A questo punto verrà creata una query per SQL con la funzione *CreaQueryRicerca()*, costituita da tutte le parole salvate in precedenza nell'array. Verranno quindi passate le variabili alla funzione in base al peso delle parole rilevate nel titolo dell'opera, nell'autore e nel tag ed il livello di precisione della ricerca. In base al livello di accuratezza si potranno trovare parole comprese in altre, esempio nel caso cercassi mastro o capomastro, o meno. In conclusione sommando il peso delle parole trovate durante la ricerca, disponiamo di una classifica disposta per ordine di importanza (dal più alto punteggio al più basso), che tiene conto della frequenza e della rilevanza della parola nel risultato. Il grado di rilevanza è stato così distribuito: 8 punti per il Titolo, 5 punti per il NomePubblico ed 1 per il Tag. Per cui se trovo 2 parole nel Titolo e 3 nel NomePubblico dell'autore otterrò un totale di 31 punti.

I risultati della ricerca saranno presenti sotto forma di piccole anteprime se si tratta di immagini o di filmati, relativamente di seguito espandibili e riproducibili e link se facenti parte a documenti piuttosto che suoni. Ogni file rintracciabile può essere salvato in una cartella; e qui ci riallacciamo al discorso di prima in riferimento all'amministrazione delle proprie cartelle e del loro contenuto. Quest'ultime vengono create in un'apposita pagina indirizzata dal link *Gestione cartelle*. Per cui quando andiamo a salvare un file il sistema ci propone i nomi delle directory dove i file debbano essere depositati.

Su ogni cartella è possibile abilitare dei permessi:

- *Privata*. È caratterizzata da un'icona raffigurante un lucchetto chiuso incorniciato da un riquadro di colore rosso. Di default il sistema, alla creazione di una cartella, attribuisce questa caratteristica. In questo stato la cartella che viene manipolata e gestita dal proprietario in modo privato, bloccato: non visibile al mondo esterno.
- *Condivisa*. Ritratta da un lucchetto chiuso avente per cornice un riquadro di colore blu. Permette la condivisione della cartella con altri docenti. Per ovviare a spiacevoli inconvenienti, solo il proprietario, creatore della cartella e della condivisione, potrà cancellare la directory o modificare i permessi della stessa. In questo stato d'adesione resta fermo il principio, già vigente nel permesso privato, in cui la cartella condivisa non è visibile all'esterno.
- *Pubblica*. L'icona questa volta è un lucchetto aperto racchiuso in un riquadro di colore verde. Chiaramente questa volta la cartella è condivisa con l'esterno, quindi è pubblica. Attenzione però, non cadiamo in facili intendimenti; per pubblica s'intende che la cartella è condivisa con gli studenti di un determinato corso di studi che utilizzano specifiche postazioni all'interno del Dipartimento dei Beni Culturali. Non che la cartella sia svincolata e quindi reperibile da qualsiasi connessione.

Se una cartella gode dei permessi pubblici o di quelli di condivisione con un altro docente, allora è possibile poter controllare e gestire, direttamente da una pagina del sito, tutte le proprie condivisioni e modificarle all'occorrenza.

Una cartella in realtà non occupa uno spazio fisico nella base di dati, ma sono solo delle query che ci permettono di proiettare il contenuto delle nostre cartelle. A questo punto, non è possibile cancellare uno o più file

dall'interno di una cartella, perché quest'azione a cascata si ripercuoterebbe su tutto il sistema. Quindi è possibile eliminare una cartella solo se non contiene dei file.

Faccio ora un breve accenno alla *Schema fisico* dei dati che in questa sede non viene implementato perché ci si è concentrati sulla realizzazione di un prototipo. Successivamente questa fase verrà analizzata consultando i risultati ottenuti durante l'utilizzo da parte degli utenti reali. Lo schema fisico è il prodotto della *progettazione fisica*, dopo aver ricevuto in ingresso lo schema logico dei dati. Si tratta di dover prendere in considerazione degli indici di riferimento sul carico di lavoro introdotto, e tutta una serie di analisi molto complesse come lo scambio di informazioni tra memoria principale e memoria secondaria, i costi della CPU in ordine di tempo, o il *clustering* operazione che associa degli indici di ricerca in un'opportuna collocazione del file system per ottenere più affidabilità nelle ricerche. Queste analisi non sono incluse nei fondamenti di un corso di sistemi di basi di dati, ciò non toglie che potranno essere motivo di interesse e studio in un prossimo futuro come ottimizzazione del progetto.

5

CONCLUSIONI

L'esperienza conclusa, alla quale mi accingo a mettere su carta le mie considerazioni finali, non è che il primo mattone nella possibile edificazione di un muro. Tanto infatti bisognerà lavorare per costruire un'un'applicazione matura e tante decisioni si dovranno prendere in merito. Restano ancora da implementare diverse pagine del sito, come quelle riguardo l'anagrafica dei docenti, e di tutte le restanti specifiche d'interesse per loro; non tralasciando riferimenti anche a quei dati come il diritto di copyright, le informazioni ricavate durante le riprese esterne e dell'attrezzatura utilizzata nello stesso contesto, che diversamente riguardano i tecnici di laboratorio. Per cui diverse viste si dovranno predisporre della stessa base di dati per i diversi utilizzatori. Il docente avrà il suo spazio nel quale sarà possibile gestire i file e condividerli con colleghi e studenti. Ai tecnici invece il compito di caricare i file nella base di dati, tenendo conto delle specifiche raccolte dalle uscite esterne (che saranno visibili solo a loro) e dalle descrizioni suggerite dai docenti.

Di seguito si dovrà migliorare il motore di ricerca, ora presente in modo scarso con un'unica selezione generica che raccoglie gli attributi di *Nome Pubblico* dell'autore, *Titolo* e *Tag* dell'opera. Per cui dovrà essere possibile in futuro reperire in modo più mirato le opere potendo scegliere ad esempio per tipo di file, area didattica di appartenenza ed altre specifiche inerenti e rappresentate nella base di dati. Quest'ultima essendo stata creata con criterio sarà difficile che siano apportate profonde modifiche. Attualmente, con la struttura presente è possibile solo creare cartelle di un solo livello di profondità, il passo successivo consentirà la predisposizione di diversi livelli all'interno della cartella; permettendo in questo modo la condivisione dei file non con tutti i docenti ma, volendo scegliere, solo con alcuni.

Attualmente il sistema è ancora in corso d'opera, ed incompleto in molte sue parti come sopra del resto descritto. Quando verrà creato un prototipo bisognerà farlo testare agli utenti finali. Nel periodo di rodaggio si raccoglieranno i pareri degli utilizzatori migliorando quegli aspetti in cui è comune qualche disagio. Volendo essere più puntigliosi un grosso sforzo si dovrà fare nel migliorare le prestazioni delle query e nell'algoritmo di ricerca, in modo da rendere tutto più fluido ed immediato. Nel caso che i test diano buoni risultati, affiancati dall'apprezzamento dei docenti e tecnici, ci saranno alte probabilità che il sistema venga adottato in pianta stabile. Per cui, l'applicazione che attualmente risiede su un computer locale, dovrà in futuro risiedere su un server. Quest'ultimo passo comporterà nuove problematiche, come la sicurezza informatica durante la consultazione dei dati, la localizzazione del server in luogo protetto da agenti fisici esterni e da maleintenzionati, per cui un ambiente climatizzato, asciutto e sicuro. Sarà necessario predisporre sistemi di backup ed il mirroring dei dischi in modo da mantenere rispettivamente copia dei dati e continuità dell'esercizio dell'applicazione.

Come potete leggere ci sarà tanto lavoro da fare; tenendo conto che una volta messo in funzione, si dovrà continuare ad ottimizzarlo con nuove funzioni ed interfacce sempre più intuitive, affiancandolo magari a nuovi

progetti esterni che puntano su di esso.

Concludo, ripetendo ancora il concetto che il presente documento di tesi vuole anche essere un punto di partenza per chi in futuro dovrà, magari, intervenire su questa base di dati relazionale. Mi auguro di aver espresso in modo chiaro le diversi componenti che hanno costituito il progetto, e di avervi condotto in maniera semplice, passo dopo passo, a comprendere come lo si è assemblato in un prodotto unico. Questo prodotto non sarebbe mai nato senza gli incontri col dott. Orio, il quale forte dell'esperienza di molti progetti prestigiosi è stato così paziente e di buon grado a curare alcune mie lacune dettate dalla mia inesperienza. Alla fine di tutto posso ritenermi fortunato, d'aver potuto lavorare con la sua collaborazione, e di aver avuto modo di approfondire le mie conoscenze nell'ambito della progettazione delle basi di dati. Tanti spunti infatti questa materia offre, come ad esempio la progettazione di schemi E-R, i linguaggi SQL, PHP e HTML, tutta la parte di gestione, sicurezza e manutenzione di un server, che nel perseguire ognuno di essi, specializzandosi, garantisce sicuramente uno sbocco lavorativo duraturo nel tempo, perché di queste figure professionali oggi giorno vi è necessità.

L'ultimo pensiero lo rivolgo all'Università di Padova ed al Dipartimento dei Beni Culturali per cui lavoro, nel quale il dott. Orio è anche presente, che mi hanno insieme permesso di poter compiere il tirocinio su un progetto da tempo predisposto ma mai realizzato. Il dott. Orio ha già manifestato la volontà di continuare questa nostra collaborazione ovviamente in modo del tutto informale, intendo svincolato dalle scadenze imposte da una tesi; fino al completamento dello scopo finale. Sicuramente continuerò ad annotare le cose degne di nota, ed implementare come un diario questa tesi, arricchendola di nuove informazioni dettate dalle mie esperienze.

A

APPENDICE

A.1 QUESTIONARIO

1. Che utilizzo fa delle immagini e/o dei file audio e/o filmati?
 - Vengono adoperate per fine didattico che come strumento di ricerca?
 - Quali sono le immagini e/o file audio e/o normalizzabili (di uso comune)?
 - Quelli utili per lei?
 - Quelli ricercabili?Quindi lei necessiterebbe di poterli:
 - Scaricare?
 - Visualizzare?
 - Che tipo di risoluzione gradirebbe?
 - Esige la possibilità di salvare un particolare ingrandito?
 - Potrebbe essere interessato a visualizzare un confronto tra immagini?
2. Secondo lei l'applicazione dovrebbe essere uno strumento di supporto alla ricerca o anche didattico rivolgendolo agli studenti?
3. Che tipo di dati (metadati) sono acquisibili ogni volta che adopera un file?
4. Potrebbe interessarla o essere utile un campo per descrivere un file presente nella base di dati?
5. Se fossero dei tag, secondo lei potrebbero tornare utili?
6. Ad ogni file presente nella base di dati sarà abbinata una scheda riassuntiva con la descrizione delle caratteristiche dell'opera; sarebbe per lei opportuno rendere pubblico quanto riportato, o deve rimanere in uno *spazio* privato?

A.2 SCHEMA RELAZIONALE

AUTORE(ID, Nome, Cognome, NomePubblico, AnnoNascita, LuogoNascita, AnnoMorte, LuogoMorte, DateCerte)

CREAZIONE(Autore, Opera)

OPERA(ID, TipoOpera, Titolo, Tecnica, Anno, Collocazione, AreaCompetenza, Altezza, Larghezza, Profondità, Durata, TitoloOriginale, TitoloInternazionale, GenereFilm, AutoreFotografia, AutoreScenografia, FormaMusicale, Tonalità, IdentificativoOpera, PrimaEsecuzione, Organico, InformazioniLibretto, Incipit, IncipitModerno, DataCreazione, DataUltimaModifica)

FILE(ID, Opera, PathName, Name, Estensione, AltezzaPunti, LarghezzaPunti, Durata, DimensioneByte, ProfonditaColore, SpazioColore, Campionamentodpi, CampionamentoHertz, CampionamentoFps, BitRate, CalibrazioneColore, FonteEditaTitolo, FonteEditaISBN, FonteEditaEditore, FonteEditaCuratore, FonteEditaCollana, FonteEditaNumInvBibl, FonteEditaProprietario, FonteAntica, Esecutore, CanaliAudio, SistemaCodifica, Descrizione, DataCreazione, DataUltimaModifica, NumeroVisite)

DATIRIPRESA(ID, DispositivoUtilizzato, Obiettivo, TipoIlluminazione, Filtro, ModalitàEsposimetrica, LetturaEsposimetrica, Tempo, Risoluzione, QualitaRipresa, QualitaComposizione)

ESTERNA(DatiRipresa, File, Data, Città, Luogo)

COPYRIGHT(File, TipoDiDiritto, EnteProprietario, Referente, Indirizzo, Telefono, Cellulare, Fax, Mail, DataAttribuzione, DataScadenza)

INSERIMENTO(File, Cartella, Docente, Data)

CARTELLA(ID, Nome, Descrizione, *Docente*, NumeroAccessi, NumeroFilePresenti, DataCreazione, DataUltimoAccesso, TipoAccesso)¹

DOCENTE(Matricola, Cognome, Nome, Qualifica, Login, Password, UltimoAccesso)

DESCRIZIONE(ID, *Docente*, *Opera*, Tag)

INSEGNAMENTO(ID, *Docente*, *CorsoDiLaurea*, Nome, Semestre, Anno)

CORSODILAUREA(Codice, Nome, Descrizione)

A.3 LISTATO SQL

```
CREATE TABLE IF NOT EXISTS 'Autore' (
  'ID' int(11) NOT NULL AUTO_INCREMENT COMMENT 'Chiave autoincrementante',
  'Nome' varchar(255) DEFAULT NULL COMMENT 'Nome dell''autore',
  'Cognome' varchar(255) DEFAULT NULL COMMENT 'Cognome dell''autore',
  'NomePubblico' varchar(255) DEFAULT NOT NULL COMMENT 'Nome pubblico col quale e' conosciuto autore',
  'AnnoNascita' int(5) DEFAULT NULL COMMENT 'Anno di nascita dell''autore',
  'LuogoNascita' varchar(255) DEFAULT NULL COMMENT 'Luogo di nascita dell''autore',
  'AnnoMorte' int(5) DEFAULT NULL COMMENT 'Anno di morte dell''autore',
  'LuogoMorte' varchar(255) DEFAULT NULL COMMENT 'Luogo in cui e' morto l''autore',
  'DateCerte' tinyint(1) NOT NULL COMMENT 'Flag per indicare se le date sono certe',
  PRIMARY KEY ('ID')
) ;

CREATE TABLE IF NOT EXISTS 'Cartella' (
```

¹ Si tenga presente che il testo *enfattizzato* di alcuni attributi indica la possibilità di trovarsi valori nulli.

```

'ID' int(11) NOT NULL AUTO_INCREMENT,
'Nome' varchar(255) NOT NULL COMMENT 'Nome della cartella',
'Descrizione' text COMMENT 'Descrizione del contenuto della cartella',
'Docente' varchar(10) DEFAULT NULL COMMENT 'Matricola del docente
proprietario della cartella',
'NumeroAccessi' int(7) DEFAULT NULL COMMENT 'Contatore del numero di
accessi alla cartella',
'NumeroFilePresenti' int(7) DEFAULT NULL COMMENT 'Contatore dei file
presenti nella cartella',
'DataCreazione' timestamp NULL DEFAULT NULL COMMENT 'Data di creazione
della cartella',
'DataUltimoAccesso' timestamp NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE
CURRENT_TIMESTAMP COMMENT 'Data dell''ultima modifica avvenuta
nella cartella',
'TipoAccesso' varchar(255) NOT NULL COMMENT 'Tipo di condivisione della
cartella: proprietario, pubblico, pubblico con altri docenti',
PRIMARY KEY ('ID'),
KEY 'Docente' ('Docente')
) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO_INCREMENT=52 ;

CREATE TABLE IF NOT EXISTS 'Copyright' (
'File' int(7) NOT NULL DEFAULT '0' COMMENT 'Con riferimento all''ID del
file',
'TipoDiDiritto' varchar(255) DEFAULT NULL COMMENT 'Tipo di diritto del
proprietario del file',
'EnteProprietario' varchar(255) DEFAULT NULL COMMENT 'Nome dell''ente
proprietaria dei diritti sul file',
'Referente' varchar(255) DEFAULT NULL COMMENT 'Nome e cognome della
persona fisica a cui si puo' fare riferimento',
'Indirizzo' varchar(255) DEFAULT NULL COMMENT 'Indirizzo dell''ente
proprietario del copyright',
'Telefono' varchar(255) DEFAULT NULL COMMENT 'Recapito telefonico dell
''ente',
'Cellulare' varchar(255) DEFAULT NULL COMMENT 'Cellulare di un
referente ',
'Fax' varchar(255) DEFAULT NULL COMMENT 'Numero di recapito fax',
'Mail' varchar(255) DEFAULT NULL COMMENT 'Indirizzo mail ',
'DataAttribuzione' date DEFAULT NULL COMMENT 'Data del periodo di
inizio del copyright',
'DataScadenza' date DEFAULT NULL COMMENT 'Data di scadenza del
copyright',
PRIMARY KEY ('File'),
KEY 'File' ('File')
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

CREATE TABLE IF NOT EXISTS 'DatiRipresa' (
'ID' int(11) NOT NULL AUTO_INCREMENT,
'DispositivoUtilizzato' varchar(255) DEFAULT NULL COMMENT 'Dispositivo
utilizzato per la ripresa, specificando marca e modello',
'Obiettivo' varchar(255) DEFAULT NULL COMMENT 'Obiettivo utilizzato per
la ripresa: marca, modello e lunghezza focale',
'TipoIlluminazione' varchar(255) DEFAULT NULL COMMENT 'Tipo d''
illuminazione utilizzata per la ripresa',
'Filtro' varchar(255) DEFAULT NULL COMMENT 'Sigla del filtro utilizzato
',
'ModalitaEsposimetrica' varchar(255) DEFAULT NULL COMMENT 'Misurazione
della luce ambiente compiuta dall''esposimetro',

```

```

'LetturaEsposimetrica' double DEFAULT NULL COMMENT 'Valore di diaframma
: + o - 1/3',
'Tempo' varchar(255) DEFAULT NULL COMMENT 'Tempo di apertura del
diaframma',
'Risoluzione' double DEFAULT NULL COMMENT 'Indicare il numero di Mega
pixel della macchina fotografica',
'QualitaRipresa' varchar(255) DEFAULT NULL COMMENT 'Indicare con una
sigla (o lettera) la qualita' della ripresa',
'QualitaComposizione' varchar(255) DEFAULT NULL COMMENT 'Indicare con
una sigla (o lettera) la qualita' della composizione',
PRIMARY KEY ('ID')
) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO_INCREMENT=1 ;

CREATE TABLE IF NOT EXISTS 'CorsoDiLaurea' (
'Codice' varchar(10) NOT NULL COMMENT 'Codice del Corso di Laurea',
'Nome' varchar(255) NOT NULL COMMENT 'Nome del Corso di Laurea',
'Descrizione' text COMMENT 'Descrizione sul Corso di Laurea',
PRIMARY KEY ('Codice')
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

CREATE TABLE IF NOT EXISTS 'Descrizione' (
'ID' int(10) NOT NULL AUTO_INCREMENT,
'Docente' varchar(10) DEFAULT NULL COMMENT 'Matricola del docente che
inserisce un tag sull''opera',
'Opera' int(11) NOT NULL COMMENT 'Con riferimento all''ID dell''opera',
'Tag' varchar(255) DEFAULT NULL COMMENT 'Parola utilizzata per
etichettare l''opera',
PRIMARY KEY ('ID'),
KEY 'Docente' ('Docente'),
KEY 'Opera' ('Opera')
) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO_INCREMENT=5 ;

CREATE TABLE IF NOT EXISTS 'File' (
'ID' int(11) NOT NULL AUTO_INCREMENT,
'Opera' int(11) DEFAULT NULL COMMENT 'Opera riferita al file',
'PathName' varchar(255) DEFAULT '/Tesi/file/' COMMENT 'Percorso di
localizzazione del file',
'Name' varchar(255) DEFAULT NULL COMMENT 'Descrittore del file: se
rappresenta un particolare, una breve scena filmata, ecc',
'Estensione' varchar(255) DEFAULT NULL COMMENT 'Tipo di estensione del
file: jpeg, tiff, avi, png, pdf, ecc.',
'AltezzaPunti' int(11) DEFAULT NULL COMMENT 'Numero di pixel dell''
altezza del file',
'LarghezzaPunti' int(11) DEFAULT NULL COMMENT 'Numero di pixel della
larghezza del file',
'Durata' int(11) DEFAULT NULL COMMENT 'Durata del file espressa in
secondi',
'DimensioneByte' int(11) DEFAULT NULL COMMENT 'Peso in byte del file',
'ProfonditaColore' int(11) DEFAULT NULL COMMENT 'Quantita' di bit
necessari per rappresentare il colore di un singolo pixel in un''
immagine',
'SpazioColore' varchar(255) DEFAULT NULL COMMENT 'Tipo di spazio colore
utilizzato per il file',
'Campionamento_dpi' int(11) DEFAULT NULL COMMENT 'Numero di pixel
presenti in un pollice: dpi',
'Campionamento_hertz' int(11) DEFAULT NULL COMMENT 'Misura espressa in
hertz del numero di volte al secondo in cui un segnale analogico
viene misurato e memorizzato in forma digitale',

```

```

'Campionamento_fps' int(11) DEFAULT NULL COMMENT 'Numero di fotogrammi
  trasmessi in un secondo, espressi in: fps',
'BitRate' int(11) DEFAULT NULL COMMENT 'Indica il numero di bit
  impiegati per descrivere l''immagine di un fotogramma: espressi in
  K (chilo) bit/sec',
'CalibrazioneColore' varchar(255) DEFAULT NULL COMMENT 'Bilanciamento
  del bianco',
'FonteEdita_Titolo' varchar(255) DEFAULT NULL COMMENT 'Titolo del libro
  o del periodico o di qualsiasi fonte edita',
'FonteEdita_ISBN' varchar(10) DEFAULT NULL COMMENT 'Codice ISBN del
  libro o del periodico',
'FonteEdita_Editore' varchar(255) DEFAULT NULL COMMENT 'Editore del
  libro o del periodico',
'FonteEdita_Curatore' varchar(255) DEFAULT NULL COMMENT 'Curatore del
  libro o del periodico',
'FonteEdita_Collana' int(11) DEFAULT NULL COMMENT 'Numero del volume
  del libro o del periodico (rivista)',
'FonteEdita_NumInvBibl' varchar(255) DEFAULT NULL COMMENT 'Numero di
  catalogazione nell''inventario della biblioteca del Dipartimento
  dei Beni Culturali',
'FonteEdita_Proprietario' varchar(255) DEFAULT NULL COMMENT 'Ente
  proprietario della fonte edita',
'FonteAntica' varchar(255) DEFAULT NULL COMMENT 'Riferimento alla prima
  testimonianza',
'Esecutore' varchar(255) DEFAULT NULL COMMENT 'Esecutore o Direttore d
  ''orchestra del brano',
'CanaliAudio' varchar(255) DEFAULT NULL COMMENT 'Tipo di registrazione
  audio: mono, stereo',
'SistemaCodifica' varchar(255) DEFAULT NULL COMMENT 'Sistema di
  codifica multicanale utilizzata: Dolby, AAC, ATRAC, MP3, Vorbis,
  WMA, ecc.',
'Descrizione' text COMMENT 'Descrizione dettagliata',
'DataCreazione' timestamp NULL DEFAULT NULL COMMENT 'Data di creazione
  del file',
'DataUltimaModifica' timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP ON
  UPDATE CURRENT_TIMESTAMP COMMENT 'Data e ora dell''ultima modifica
  del file',
'NumeroVisite' int(11) DEFAULT NULL COMMENT 'Contatore del numero di
  visite',
PRIMARY KEY ('ID'),
KEY 'Opera' ('Opera')
) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO_INCREMENT=5 ;

CREATE TABLE IF NOT EXISTS 'Docente' (
  'Matricola' varchar(10) NOT NULL COMMENT 'Numero di matricola del
  docente',
  'Cognome' varchar(255) NOT NULL COMMENT 'Cognome del docente',
  'Nome' varchar(255) NOT NULL COMMENT 'Nome del docente',
  'Qualifica' varchar(255) NOT NULL COMMENT 'Tipo di qualifica d''
  insegnamento: docente ordinario, assegnista, ricercatore, ecc.',
  'Login' varchar(255) NOT NULL COMMENT 'Username per logarsi al sistema
  ',
  'Password' varchar(255) NOT NULL COMMENT 'Password d''accesso al
  sistema',
  'UltimoAccesso' timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE
  CURRENT_TIMESTAMP,
  PRIMARY KEY ('Matricola'),
  UNIQUE KEY 'Login' ('Login')
)

```

```

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

CREATE TABLE IF NOT EXISTS 'Esterna' (
  'DatiRipresa' int(11) NOT NULL,
  'File' int(11) NOT NULL,
  'Data' date DEFAULT NULL COMMENT 'Data in cui e' stata ripresa l''opera
  ',
  'Citta' varchar(255) DEFAULT NULL COMMENT 'Citta' o paese dov''e' stata
  fatta la ripresa dell''opera',
  'Luogo' varchar(255) DEFAULT NULL COMMENT 'Nome del luogo o del
  complesso monumentale che custodiva l''opera ripresa',
  PRIMARY KEY ('DatiRipresa','File'),
  KEY 'File' ('File')
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

CREATE TABLE IF NOT EXISTS 'Insegnamento' (
  'ID' int(11) NOT NULL AUTO_INCREMENT COMMENT 'ID progressivo',
  'Docente' varchar(10) NOT NULL COMMENT 'Matricola del docente dell''
  insegnamento',
  'CorsoDiLaurea' varchar(10) NOT NULL COMMENT 'Codice del Corso di
  Laurea ',
  'Codice' varchar(255) NOT NULL COMMENT 'Codice dell''insegnamento',
  'Nome' varchar(255) NOT NULL COMMENT 'Nome dell''insegnamento',
  'Semestre' int(1) NOT NULL COMMENT 'Esprimere con un numero il semestre
  di collocazione dell''insegnamento',
  'Anno' year(4) NOT NULL COMMENT 'Anno dell''insegnamento',
  PRIMARY KEY ('ID'),
  KEY 'CorsoDiLaurea' ('CorsoDiLaurea'),
  KEY 'Docente' ('Docente')
) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO_INCREMENT=1 ;

CREATE TABLE IF NOT EXISTS 'Inserimento' (
  'File' int(11) NOT NULL COMMENT 'Con riferimento all''ID del file
  utilizzato',
  'Cartella' int(11) NOT NULL COMMENT 'Con riferimento all''ID della
  cartella',
  'Docente' varchar(10) NOT NULL COMMENT 'Con riferimento al numero di
  matricola del docente',
  'Data' timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_
  TIMESTAMP COMMENT 'Data e ora dell''ultima modifica effettuata ',
  PRIMARY KEY ('File','Cartella','Docente'),
  KEY 'Cartella' ('Cartella'),
  KEY 'Docente' ('Docente')
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

CREATE TABLE IF NOT EXISTS 'Opera' (
  'ID' int(11) NOT NULL AUTO_INCREMENT COMMENT 'Chiave autoincrementante
  ',
  'TipoOpera' enum('A','C','M','T') NOT NULL COMMENT 'Tipologia di opera
  (A=arte, C=cinema, M=musica, T=testo)',
  'Titolo' varchar(255) DEFAULT NULL COMMENT 'Titolo dell''opera',
  'Tecnica' varchar(255) DEFAULT NULL COMMENT 'Tecnica di realizzazione
  dell'' opera: dipinto, scultura, film, composizione musicale, ecc
  .',
  'Anno' int(5) DEFAULT NULL COMMENT 'Anno di creazione dell''opera',
  'Collocazione' varchar(255) DEFAULT NULL COMMENT 'Luogo o museo in cui
  e' visibile l''opera',

```

```

'AreaCompetenza' varchar(255) DEFAULT NULL COMMENT 'Area didattica di
appartenenza: Storia dell''Arte, Storia del Cinema, Storia della
Musica, ecc.',
'Altezza' double DEFAULT NULL COMMENT 'Altezza in centimetri dell''
opera',
'Larghezza' double DEFAULT NULL COMMENT 'Larghezza in centimetri dell''
opera',
'Profondita' double DEFAULT NULL COMMENT 'Profondita' in centimetri
dell''opera',
'Durata' int(11) DEFAULT NULL COMMENT 'Durata in secondi dell''opera',
'TitoloOriginale' varchar(255) DEFAULT NULL COMMENT 'Titolo originale
dell''opera',
'TitoloInternazionale' varchar(255) DEFAULT NULL COMMENT 'Titolo
internazionale per cui e' conosciuta l''opera',
'GenereFilm' varchar(255) DEFAULT NULL COMMENT 'Tema o caratteristica
ricorrente del film: avventura, azione, drammatico, horror,
musical, animazione, western, guerra, ecc.',
'AutoreFotografia' varchar(255) DEFAULT NULL COMMENT 'Direttore della
fotografia responsabile della fotografia cinematografica durante
la realizzazione di un film',
'AutoreScenografia' varchar(255) DEFAULT NULL COMMENT 'Ideatore di
elementi scenici in uno spettacolo cinematografico',
'FormaMusicale' varchar(255) DEFAULT NULL COMMENT 'Tipo di composizione
musicale: sinfonia, fuga, ballata, ecc. ',
'Tonalita' varchar(255) DEFAULT NULL COMMENT 'Tonalita' dell''opera
musicale',
'IdentificativoOpera' varchar(255) DEFAULT NULL COMMENT 'Numero dell''
opera composta',
'PrimaEsecuzione' varchar(255) DEFAULT NULL COMMENT 'Data e luogo della
prima rappresentazione dell''opera',
'Organico' varchar(255) CHARACTER SET utf32 DEFAULT NULL COMMENT '
Composizione e quantita' dell''organico orchestrale: ottavino,
flauto, archi, ecc.',
'InformazioniLibretto' varchar(255) DEFAULT NULL COMMENT 'Informazioni
libretto: autore, fonte letteraria',
'Incipit' varchar(255) DEFAULT NULL COMMENT 'Incipit testuale della
musica ',
'IncipitModerno' varchar(255) DEFAULT NULL COMMENT 'Incipit testuale in
trascrizione moderna ',
'DataCreazione' timestamp NULL DEFAULT NULL COMMENT 'Data e ora dell''
inserimento dei dati',
'DataUltimaModifica' timestamp NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE
CURRENT_TIMESTAMP COMMENT 'Data e ora dell''ultima modifica ai
dati inseriti',
PRIMARY KEY ('ID')
) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO_INCREMENT=11 ;

--
-- Constraints for table 'Cartella'
--
ALTER TABLE 'Cartella'
ADD CONSTRAINT 'Cartella_ibfk_2' FOREIGN KEY ('Docente') REFERENCES '
Docente' ('Matricola') ON DELETE CASCADE ON UPDATE CASCADE;

--
-- Constraints for table 'Copyright'
--
ALTER TABLE 'Copyright'

```

```

ADD CONSTRAINT 'Copyright_ibfk_2' FOREIGN KEY ('File') REFERENCES 'File'
  ('ID');

--
-- Constraints for table 'Creazione'
--
ALTER TABLE 'Creazione'
  ADD CONSTRAINT 'creazione_ibfk_1' FOREIGN KEY ('Autore') REFERENCES 'Autore' ('ID'),
  ADD CONSTRAINT 'creazione_ibfk_2' FOREIGN KEY ('Opera') REFERENCES 'Opera' ('ID');

--
-- Constraints for table 'Descrizione'
--
ALTER TABLE 'Descrizione'
  ADD CONSTRAINT 'Descrizione_ibfk_1' FOREIGN KEY ('Docente') REFERENCES 'Docente' ('Matricola'),
  ADD CONSTRAINT 'Descrizione_ibfk_2' FOREIGN KEY ('Opera') REFERENCES 'Opera' ('ID');

--
-- Constraints for table 'Esterna'
--
ALTER TABLE 'Esterna'
  ADD CONSTRAINT 'Esterna_ibfk_1' FOREIGN KEY ('DatiRipresa') REFERENCES 'DatiRipresa' ('ID'),
  ADD CONSTRAINT 'Esterna_ibfk_2' FOREIGN KEY ('File') REFERENCES 'File' ('ID');

--
-- Constraints for table 'File'
--
ALTER TABLE 'File'
  ADD CONSTRAINT 'File_ibfk_1' FOREIGN KEY ('Opera') REFERENCES 'Opera' ('ID');

--
-- Constraints for table 'Insegnamento'
--
ALTER TABLE 'Insegnamento'
  ADD CONSTRAINT 'Insegnamento_ibfk_3' FOREIGN KEY ('Docente') REFERENCES 'Docente' ('Matricola') ON DELETE CASCADE ON UPDATE CASCADE,
  ADD CONSTRAINT 'Insegnamento_ibfk_4' FOREIGN KEY ('CorsoDiLaurea') REFERENCES 'CorsoDiLaurea' ('Codice') ON DELETE CASCADE ON UPDATE CASCADE;

--
-- Constraints for table 'Inserimento'
--
ALTER TABLE 'Inserimento'
  ADD CONSTRAINT 'Inserimento_ibfk_1' FOREIGN KEY ('File') REFERENCES 'File' ('ID') ON DELETE CASCADE ON UPDATE CASCADE,
  ADD CONSTRAINT 'Inserimento_ibfk_2' FOREIGN KEY ('Cartella') REFERENCES 'Cartella' ('ID') ON DELETE CASCADE ON UPDATE CASCADE,
  ADD CONSTRAINT 'Inserimento_ibfk_3' FOREIGN KEY ('Docente') REFERENCES 'Docente' ('Matricola') ON DELETE CASCADE ON UPDATE CASCADE;

```

BIBLIOGRAFIA

- [1] P. Atzeni, S. Ceri, S. Paraboschi, R. Torlone. *Basi di dati - Modelli e linguaggi di interrogazione*. Seconda edizione, 2006 - McGraw-Hill
- [2] R. A. Elmasri, S. B. Navathe. *Sistemi di basi di dati - Fondamenti*. Quinta edizione, 2007 - Pearson
- [3] R. S. Pressman. *Principi di ingegneria del software - Fondamenti*. Quinta edizione, 2007 - McGraw-Hill
- [4] P. B. MacIntyre. *PHP - Le tecniche per scrivere il codice migliore*. 2010 - O'Reilly
- [5] G. Troiani. *CSS - Terza edizione*, 2011 - Apogeo

SITI WEB CONSULTATI

Per la realizzazione del progetto mi sono servito dell'aiuto della community e delle informazioni raccolte in:

1. www.html.it
2. www.guitex.org utilizzatori di \LaTeX .