



UNIVERSITÀ DI PISA

Dipartimento di Filologia, Letteratura e Linguistica

Corso di Laurea in Informatica Umanistica

**Sistema informativo per la gestione di pratiche
assicurative**

Candidato:

Federica Raffaelli

Relatore:

Prof. Vittore Casarosa

Anno accademico 2016/2017

Sommario

1 Introduzione	3
1.1 Il mondo assicurativo e il ruolo del perito.....	3
1.2 Come venivano catalogate le pratiche prima.....	4
1.3 Cosa significa gestire le pratiche assicurative oggi.....	6
2 Analisi generale	8
2.1 Design dell'applicazione	8
3 Analisi delle funzionalità	9
3.1 Introduzione – che cos'è un'applicazione web	9
3.2 Studio Colaiacomo - perizie assicurative: un'applicazione web.....	10
3.3 Navigazione dell'applicazione	10
3.3.1 Home page.....	11
3.3.2 Inserimento sinistro	11
3.3.3 Cerca sinistro.....	14
3.3.4 Cerca collaboratore e cerca agenzia	15
3.3.5 Inserimento.....	17
4 Progettazione di una base di dati	19
4.1 Definizione di database	19
4.1.1 Modello relazionale.....	20
4.4.2 Modello entità-relazione	21
4.2 Fasi di progettazione di un database	22
4.3 Utenti di un database.....	22
4.4 Realizzazione del database.....	23
5 Strumenti di sviluppo	25
5.1 Linguaggi lato client.....	25
5.1.1 HTML.....	25
5.1.2 CSS.....	26
5.1.3 JavaScript	26
5.1.4 jQuery.....	27

5.2 Linguaggi lato server	27
5.2.1 PHP.....	28
5.2.2 SQL.....	29
6 Conclusioni	31
6.1 Sviluppi futuri	31
7 Ringraziamenti.....	32
8 Bibliografia	33
9 Sitografia.....	34

1 Introduzione

Questa tesi nasce come proseguimento delle mie attività di tirocinio, che ho svolto presso lo Studio Peritale Colaiacomo di La Spezia, il quale opera nel settore assicurativo, svolgendo, su mandato delle Compagnie assicurative di cui è fiduciario, attività di accertamento e quantificazione dei danni relativi ai sinistri denunciati dai contraenti di assicurazioni vale a dire coloro che sottoscrivono polizze a garanzia di eventi come incendio, furto, infortunio, ecc.

1.1 Il mondo assicurativo e il ruolo del perito

L'assicurazione, intesa come necessità di provvedere alla propria sicurezza, proteggendosi dall'imprevedibilità degli eventi (rischi), è presente sin dalle origini dell'uomo. E' stato difatti possibile riscontrare come questo stesso bisogno, nel corso della storia, si sia sviluppato e come l'uomo poi sia stato in grado di fronteggiarlo. Ad esempio, i popoli antichi sopperivano a questa necessità frazionando il rischio in più persone in modo che ciascuna di esse se ne prendesse a carico solo una piccola parte. Ma l'assicurazione vera e propria ha origini solamente nel XIII secolo con l'importanza assunta dal commercio marittimo. Infatti l'aumento del traffico per mare costrinse gli armatori ed i mercanti a porsi al riparo dai rischi della navigazione. Nei secoli successivi, nei vari paesi d'Europa, l'assicurazione marittima si sviluppò sempre più, tanto che nacquero poi altre forme di assicurazione.

In tempi moderni l'assicurazione ha lo scopo di garantire al soggetto assicurato, ossia colui che ha stipulato regolare contratto di assicurazione (polizza), il risarcimento da effetti dannosi avvenuti in seguito al verificarsi di un evento imprevisto (sinistro).

Quindi in caso di sinistro colui che è titolare di una polizza assicurativa deve inoltrare alla propria agenzia, con la quale ha stipulato regolare contratto di assicurazione, avviso di sinistro.

A questo punto l'agenzia inoltrerà l'avviso di sinistro al liquidatore competente, ovvero la figura professionale che all'interno delle Compagnie di assicurazioni è preposta alla gestione della pratica, il quale, al termine dell'iter istruttorio, deciderà se accogliere o meno la richiesta di risarcimento.

Il liquidatore, in prima battuta, esaminerà l'avviso di sinistro, al fine di verificare se lo stesso è completo in tutti i suoi elementi, se è tempestivo e, soprattutto, se la fattispecie oggetto di denuncia rientra nella copertura assicurativa.

Esaurito questo primo passaggio, il liquidatore, se riterrà soddisfatti i requisiti di cui sopra, darà corso all'attività istruttoria vera e propria volta all'accertamento e liquidazione dei danni subiti da assicurati e terzi.

A svolgere detta attività non è il liquidatore, il quale è sprovvisto di competenze specifiche, bensì altri soggetti, studi peritali, geometri, ingegneri, architetti, medici legali etc., ai quali le Compagnie assicuratrici, e nello specifico i liquidatori, si affidano per avere pareri tecnici.

Ed è in questo momento che entra in gioco il ruolo del perito assicurativo che consiste, per l'appunto, nel raccogliere i dati tecnici e nell'elaborarli per conto del liquidatore. Il perito, dunque, svolgerà una indagine tecnica, redigendo una relazione con la quale si pronuncerà sulla compatibilità dei danni rispetto alla dinamica denunciata e sulla relativa stima.

Il rapporto tra liquidatore e perito comporta il conferimento a quest'ultimo di numerosi incarichi.

Proprio perché il perito assicurativo è chiamato ad operare in tale contesto, lo stesso negli ultimi anni ha dovuto adeguare il proprio modo di lavorare alle esigenze, di volta in volta, rappresentate dalle assicurazioni.

Queste, infatti, nel corso degli ultimi decenni ed in parallelo con una progressiva informatizzazione del lavoro, hanno attuato un processo di accentramento delle proprie strutture.

Tale processo non ha interessato le agenzie, le quali continuano ad operare a livello locale col compito di stipulare polizze, bensì la figura del liquidatore, la cui presenza sul territorio, in origine, era quasi capillare.

Era il liquidatore, presente di regola in ogni provincia, a trattare i sinistri avvenuti in tale ambito territoriale e, quindi, a conferire personalmente l'incarico al perito di svolgere gli accertamenti propedeutici all'accoglimento oppure al rigetto della richiesta di risarcimento.

Il perito, pertanto, aveva la possibilità di relazionarsi direttamente col liquidatore, dal quale riceveva l'incarico e al quale consegnava la propria relazione.

Prevalendo il rapporto personale, il perito era abituato ad avere come punto di riferimento il nominativo del liquidatore, piuttosto che il numero della pratica.

La progressiva informatizzazione del lavoro, a cui sopra si faceva cenno, ha spinto le compagnie assicurative a centralizzare l'attività, dividendo il territorio in macro aree e creando dei centri di liquidazione (dentro ai quali sono confluiti i singoli liquidatori) incaricati di gestire e trattare i relativi sinistri.

Tale processo di accentramento è stato condotto sfruttando le nuove tecnologie e creando delle banche dati alle quali sia i liquidatori che i professionisti convenzionati, tra i quali appunto i periti, posso accedere per recuperare i dati relativi ad un determinato sinistro e condividere gli accertamenti tecnici eseguiti.

In tale contesto, la figura professionale del perito si è dovuta evolvere.

Ad oggi, quindi, il ruolo del perito deve essere conciliato con l'avanzare delle tecnologie in quanto le stesse compagnie assicurative si avvalgono di portali automatizzati nei quali vengono inseriti i dati del sinistro ed anche i relativi documenti.

1.2 Come venivano catalogate le pratiche prima

Prima di procedere a descrivere le fasi di cui si compone il processo di gestione delle pratiche assicurative, per poi comprendere al meglio quali siano le necessità che hanno portato alla realizzazione di un nuovo sistema di archiviazione, è doveroso illustrare quale fosse il metodo di catalogazione che lo Studio Colaiacomo utilizzava.

Inizialmente come sistema di archiviazione di dati veniva utilizzato il programma Excel. I professionisti dello Studio, una volta che ricevevano, dai sopracitati portali delle compagnie, nuovi incarichi, erano tenuti a memorizzare i dati più significativi che rappresentavano gli incarichi stessi (quali numero di riferimento interno, numero di sinistro, nominativo dell'assicurato, ecc) in appositi fogli Excel, suddivisi in base alla compagnia e al centro liquidazione danni. Questi stessi fogli di lavoro rappresentavano pertanto l'unico metodo di catalogazione e ritrovamento delle pratiche, attraverso l'uso di filtri, di cui disponeva lo studio (vedi Fig. 1).

	A	B	C	D	E
	RIF - RM	COMPAGNIA	Numero sinistro	CONTRAENTE/CONTROPARTE	Luogo sinistro
3				CLD-LU-REALE 2015	
4					
5	CLC-15-01	REALE MUTUA	2015/15747	BIRINDELLI MARIA IRENE	ISOLA D'ELBA
6	CLC-15-02	REALE MUTUA	2015/39230	COND MANIERO	QUARRATA
7	CLC-15-03	REALE MUTUA	2015/96035	BERTI DORIANO	ISOLA D'ELBA
8	CLC-15-04	REALE MUTUA	2015/98468	SERINI ALESSANDRA	F. D.MARMII
9	CLC-15-05	REALE MUTUA	2015/106433	COMPAGNIA ITALIANA ALIMENTARI	PRATO
10	CLC-15-06	REALE MUTUA	2015/109216	CONGR.FIGLIE DI GESU'	MARINA DI CARRARA
11	CLC-15-07	REALE MUTUA	2015/109234	CONGR.FIGLIE DI GESU'	MASSA
12	CLC-15-08	REALE MUTUA	2015/98350	VEZZONI ALESSANDRO	F. D.MARMII
13	CLC-15-09	REALE MUTUA	2015/98369	VEZZONI ALESSANDRO	F. D.MARMII
14	CLC-15-10	REALE MUTUA	2015/110924	STILE LIBERO SRL	FIRENZE
15	CLC-15-11	REALE MUTUA	2015/114584	AUTOLINEE TOSCANA NORD SRL	BENNABIO
16	CLC-15-12	REALE MUTUA	2015/154503	FAVILLA CRISTIANA	SAN CONCORDIO
17	CLC-15-13	REALE MUTUA	2015/154465	VANITA' SNC	CAMPI BISENZIO

Figura 1. Esempio di tabella Excel in cui venivano archiviati gli incarichi

Nonostante Excel, che nasce come semplice foglio elettronico destinato a svolgere calcoli più o meno complessi, con il passare del tempo si sia evoluto, condensando in se stesso tante e diverse funzionalità, tra le quali anche quella di database, e venendo particolarmente apprezzato per via della sua semplicità da coloro che non possiedono capacità sufficienti a utilizzare software più specifici, lo stesso non poteva più essere in grado di soddisfare le esigenze imminenti a cui lo Studio andava incontro. Difatti, quando il database è limitato a livello di quantità di dati, è possibile gestirlo attraverso un semplice foglio Excel ma quando il numero delle pratiche da gestire, e quindi da archiviare, aumenta in maniera significativa è necessario ricorrere ad un database professionale che possa consentire illimitato inserimento di pratiche, dati e filtri che possano creare un metodo di archiviazione completo, fruibile da tutti gli utenti dello Studio.

È opportuno quindi definire nello specifico quali siano i limiti e gli svantaggi dell'utilizzo di Excel rispetto alla creazione di un vero e proprio database.

Un *database* può essere definito come una collezione di dati, correlati tra loro, gestiti da un elaboratore elettronico. Comprende inoltre, una raccolta di definizioni che descrivono la struttura dei dati, le restrizioni sui valori ammissibili dei dati (vincoli di integrità), le relazioni esistenti tra gli

insiemi e alcune operazioni eseguibili sui dati. In parole più semplici, è un insieme di oggetti, chiamate tabelle, suddivise a loro volta in campi e record, che sono in relazione tra loro.

È proprio nel fatto che esiste rapporto di relazione tra le tabelle che si individua la differenza principale tra database ed Excel e conseguentemente gli svantaggi che apporta l'uso di questo stesso programma come metodo di archiviazione.

A fronte del costante incremento degli incarichi affidati, il precedente mezzo di catalogazione, vale a dire l'inserimento dei dati più significativi di una pratica (numero di riferimento interno, numero di sinistro, nominativo dell'assicurato, ecc) all'interno di un semplice foglio Excel, rischiava di generare confusione, causando, potenzialmente, anche lo smarrimento delle pratiche.

Col precedente metodo lo Studio peritale, dopo aver inserito gli estremi della pratica nel foglio Excel, doveva comunque creare delle cartelle - suddivise per Compagnia, tipologia sinistro, area geografica e tipologia di indagine eseguita – in cui inserire, mediante digitalizzazione, i dati dei singoli incarichi.

La moltiplicazione delle cartelle di archiviazione, unitamente all'impossibilità di incrociare i dati, oltre a rallentare il lavoro, aumentava il pericolo di smarrire le pratiche stesse, esponendo lo Studio peritale al rischio di non evadere l'incarico, con conseguente perdita di credibilità nei confronti della Compagnia mandataria.

In Excel manca infatti la possibilità di mettere in relazione le tabelle, queste rimangono oggetti separati sui quali non è possibile effettuare *query*, ovvero interrogazione sui dati. I database, invece, prevedono, oltre alle tabelle e alle relazioni, un insieme di strumenti per inserire, modificare, cancellare e cercare, stampare, analizzare i dati, visualizzare i dati su schermo e qualsiasi altra operazione sia necessaria alla vita dell'archivio. Il tutto deve essere costruito attraverso un programma per costruire e gestire una base di dati, che permette cioè di creare tabelle, definire le relazioni, costruire tutti gli strumenti per inserire, cercare, analizzare, modificare ed eliminare i dati; questi programmi prendono il nome di DBMS (*Database Management System*).

Pertanto, quando si parla di Excel il termine database è inappropriato, in quanto quello che si può creare col citato programma è in realtà un elenco, ovvero una tabella che rappresenta un oggetto separato da tutti gli altri.

1.3 Cosa significa gestire le pratiche assicurative oggi

Nell'ambito della gestione degli incarichi affidati ai periti, suddetti portali rappresentano quindi una soluzione tecnologica in grado di favorire le relazioni tra il perito, il liquidatore e le agenzie assicurative. Difatti sono stati creati come sistema di gestione integrata dedicato a tutti coloro che sono coinvolti nella gestione di un sinistro assicurativo, proponendo un metodo organizzativo in grado di ridurre tempi e costi di un sinistro attraverso l'organizzazione delle informazioni che sono rese accessibili a tutti gli utenti, i quali sono localizzati su tutto il territorio. Gli utenti autorizzati ad accedere al portale sono: gli studi peritali (periti stessi e collaboratori), liquidatori, rappresentanti degli uffici fiduciari ed agenzie di competenza. I portali assicurativi quindi giocano un ruolo fondamentale nella semplificazione del complesso processo di affidamento degli incarichi da parte dei periti.

In base al ramo, ovvero alla tipologia del sinistro (incendio, evento atmosferico, globale fabbricati, responsabilità civile, ecc.), all'area geografica in cui si è verificato e alle competenze del perito gli incarichi vengono affidati dalle agenzie ai diversi periti attraverso, come accennato, l'inserimento di documenti digitali. Tramite l'utilizzo del portale tutti gli utenti possono monitorare lo stato delle pratiche affidate; lo stesso perito può costantemente informare sia le agenzie che i liquidatori sullo stato di avanzamento della pratica mediante interlocutorie che servono ad attribuire un importo preventivo del danno, poiché, in base all'importo stimato, gli incarichi possono essere revocati ed affidati ad altri periti le cui competenze, maturate nel corso del tempo, sono più adeguate.

Pertanto il perito, non potendo più avere interazione dirette con il liquidatore e dovendo gestire gli incarichi ricevuti mediante l'accesso al sopraccitato portale, necessita di un sistema informativo, una sorta di archivio digitale interno al proprio studio, che gli consenta di archiviare i dati identificativi degli incarichi ricevuti.

La creazione di un archivio digitale apporta numerosi vantaggi. Innanzitutto è possibile migliorare la stessa produttività e condizione del lavoro in quanto la ricerca delle informazioni è resa più semplice, veloce ed intuitiva grazie anche al fatto che l'organizzazione dei dati, catalogandoli, garantisce notevoli risparmi di tempo. Di conseguenza si assicura una rapida rintracciabilità delle informazioni senza dover perdere tempo per cercare le pratiche all'interno degli archivi cartacei.

Il perito quindi in assenza di un metodo di archiviazione valido ed efficace, rischierebbe di non riuscire ad evadere gli incarichi affidategli nei tempi prestabiliti.

Dunque, in linea con l'evoluzione dell'informatica, e soprattutto dopo un'attenta analisi delle problematiche e dell'esigenze che comporta l'attività di uno studio peritale, è stato deciso di sviluppare un progetto che, attraverso l'archiviazione delle informazioni relative agli incarichi ricevuti, fosse in grado di facilitare il perito nella raccolta e ricerca dei dati.

2 Analisi generale

L'azienda committente è rappresentata dallo Studio peritale Colaiacomo che esercita la propria attività nell'ambito assicurativo. Quotidianamente lo Studio deve gestire numerosi incarichi relativi a sinistri assicurativi che gli vengono affidati dalle Compagnie assicuratrici. Per tali incarichi deve procedere, dopo aver eseguito un opportuno sopralluogo, alla stesura di una relazione di perizia in cui quantifica il danno che verrà in seguito liquidato da un'altra figura professionale operante nell'ambito delle assicurazioni (v. cap. 1).

Dall'esigenza quindi di tenere sempre traccia degli incarichi in gestione e dalla necessità di tutelare e conservare le informazioni più significative relative ai suddetti incarichi, è stata concepita l'idea di creare questo sistema informativo, basato essenzialmente sui principi della consultazione e della ricerca.

In base ai bisogni dello Studio sono state infatti create diverse funzionalità:

- Funzionalità di inserimento: esiste nell'applicazione creata la possibilità che un utente inserisca tutti i dati relativi ai nuovi incarichi affidati in modo che gli stessi vengano archiviati;
- Funzionalità di ricerca: sono state create diverse pagine dinamiche in cui viene data la possibilità all'utente di ricercare le informazioni relative agli incarichi che lo stesso ha in gestione, ma non solo: è possibile difatti ricercare informazioni riguardanti anche i collaboratori (professionisti come ingegneri, geometri, architetti, ecc) che lavorano per conto dello Studio, ed anche relative alle Agenzie locali con cui spesso lo Studio deve interagire;
- Funzionalità di modifica: l'utente può modificare alcuni dati degli incarichi affidategli.

L'analisi specifica delle caratteristiche e di queste stesse funzionalità dell'applicazione verranno in seguito spiegate più nel dettaglio (v. cap. 3).

È comunque opportuno esaminare quale scelte, a livello di visualizzazione e di stile, siano state adottate durante la realizzazione dell'interfaccia dell'applicazione.

2.1 Design dell'applicazione

Innanzitutto in tutte le pagine dell'applicazione è stato deciso di inserire il logo aziendale in quanto esso rappresenta un aspetto fondamentale per la definizione della grafica delle pagine web ed inoltre è stato realizzato un ampio menu suddiviso per parole chiave che garantiscono l'accesso alle varie macro sezioni dell'intera applicazione.

Anche la scelta dei colori assume molta importanza nella progetto di *web-design*: le scelte cromatiche sono state basate dalle richieste del committente e dettate anche dal logo scelto. I colori che prevalgono nella realizzazione del progetto sono nero, bianco e grigio, in quanto questi stessi colori sono neutri e sinonimo di sobrietà, eleganza e serietà, valori che rispecchiano i pilastri su cui appunto si basa la filosofia dell'azienda committente.

Altresì l'aspetto tipografico non è marginale, nel caso specifico, è stato scelto l'utilizzo di un *font* standard per garantire, in primo luogo, la leggibilità dei contenuti e rendere quindi allo stesso tempo

gradevole la lettura, e comunque per rimanere in tema con le scelte stiliste di eleganza e semplicità adottate anche per gli altri aspetti strutturali.

Riepilogando, i punti cardine sui quali ho focalizzato l'attenzione nella realizzazione dell'applicazione, sono stati:

- Creazione di una grafica semplice e gradevole;
- Facile navigazione che permettesse all'utente di trovare in modo semplice ed intuitivo le informazioni di cui aveva bisogno;
- Pagine web leggere per velocizzare i tempi di caricamento del browser.

3 Analisi delle funzionalità e progettazione del database

3.1 Introduzione – che cos'è un'applicazione web

Per applicazioni web si intendono tutte quelle applicazioni che, invece di risiedere fisicamente sui computer degli utenti, risiedono su un'unica macchina (*server* remoto) che può essere collocata in qualsiasi parte del mondo.

In pratica un'applicazione web è un programma per il cui funzionamento non è necessaria l'installazione di alcun software ma è accessibile semplicemente tramite un *browser* web (Internet Explorer, Mozilla Firefox, ecc.), che svolge il ruolo di *client*. Appare quindi chiaro come il meccanismo sul quale si basano le applicazioni web è di tipo *client-server*: il *client* riceve un comando dall'utente e provvede quindi a collegarsi al *server* che, a sua volta, elabora la richiesta e provvede a restituire i dati che verranno poi mostrati all'utente dal *client*.

Alcuni esempi pratici di applicazioni web sono: motori di ricerca come Google, piattaforme *social network* quali Facebook e Twitter, siti di *e-commerce* e molti altri ancora.

Le applicazioni web, la cui crescita e diffusione è iniziata attorno alla fine degli anni novanta con il contemporaneo avanzamento del web dinamico, rappresentano quindi una soluzione innovativa e potente grazie alla loro facilità d'uso ed anche ai numerosi vantaggi che si ottengono prediligendo il loro utilizzo a discapito invece di un'applicazione installata sul proprio computer.

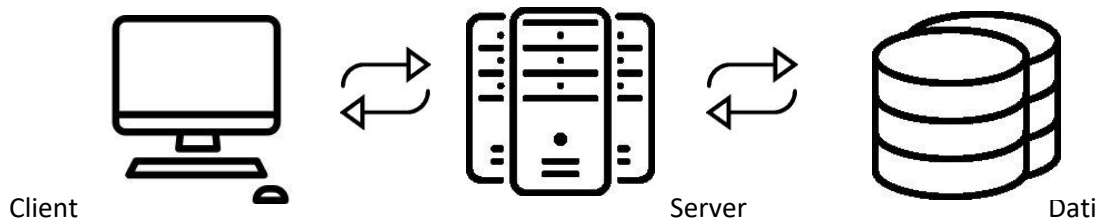


Figura 2: Architettura client-server

Di seguito alcuni dei principali vantaggi:

- **Ridotta complessità:** le applicazioni web non richiedono l'installazione di alcun *software* specifico sul proprio pc.
- **Accesso multi-piattaforma:** le applicazioni web sono compatibili con tutti i computer, indipendentemente dal sistema operativo; per accedervi è sufficiente una connessione a Internet e sono inoltre fruibili da qualsiasi dispositivo (non dipendono dall'*hardware* che si utilizza).
- **Gestione ottimizzata:** installazione, distribuzione e aggiornamento efficienti e veloci poiché vengono svolti dal server che ospita l'applicazione.

- **Accessi multipli consentiti:** sono facilmente accessibili da più utenti contemporaneamente e da qualsiasi postazione.
- **Sicurezza dei dati:** essendo gli stessi dati salvati su un *server* remoto le operazioni di *backup* vengono eseguiti dai sistemi online in modo continuo e affidabile.
- **Scalabilità:** è facile creare funzionalità aggiuntive grazie alla semplicità degli aggiornamenti.
- **Riduzione dei costi:** l'uso di Internet come infrastruttura per un'applicazione web riduce notevolmente sia i costi di connettività che i costi di gestione.

Tuttavia esistono ancora degli svantaggi legati alle applicazioni web:

- **Necessità di una connessione Internet:** se da un lato, uno dei vantaggi delle applicazioni web è proprio il fatto che per funzionare non necessitano l'installazione di alcun tipo di software specifico, dall'altro rappresenta un limite in quanto in mancanza di connessione Internet le stesse non possono funzionare.

3.2 Studio Colaiacomo - perizie assicurative: un'applicazione web

L'applicazione web che viene sviluppata in questo progetto è stata pensata per la gestione di pratiche assicurative.

Più precisamente, lo scopo principale è quello di implementare un sistema che sia in grado di raccogliere e archiviare tutti i dati significativi relativi agli incarichi affidati allo Studio peritale Colaiacomo.

Assumendo come cardini principali della realizzazione del progetto i concetti di semplicità e trasparenza, è stata creata un'interfaccia web attraverso la quale ogni singolo utente dello Studio ha la possibilità di visualizzare, inserire, modificare e ricercare le informazioni degli incarichi che lo Studio stesso ha in gestione in accordo con la necessità di creare un metodo di archiviazione che rendesse, in parte, l'attività del perito più celere e immediata.

Le funzionalità richieste sono:

- **Inserimento** di nuovi incarichi e di nuovi dati inerenti ai sinistri assicurativi;
- **Ricerca e visualizzazione** delle pratiche che lo Studio ha in gestione e dei collaboratori e agenzie con cui lo Studio lavora;
- **Modifica** dei dati relativi ad un determinato incarico.

Esempio. Un dipendente dello Studio riceve una telefonata da un cliente, il quale, fornendo il numero di sinistro della propria pratica, vuole ricevere informazioni riguardanti il perito collaboratore che si occuperà della suddetta pratica. Il dipendente verifica, tramite una semplice funzionalità di ricerca, i dati relativi al sinistro indicato dal cliente in cui comparirà anche il nominativo del collaboratore. In seguito sarà possibile, tramite la funzione di ricerca collaboratore, individuare i dati (tra cui recapito telefonico) inerenti allo stesso.

3.3 Navigazione dell'applicazione

Ogni pagina dell'applicazione web è formata da un menu che si trova in posizione fissa centrale.

Ogni elemento del menu è costituito da pagine *HTML e PHP*:

Home	Inserisci sinistro	Cerca sinistro	Cerca collaboratore	Cerca agenzia	Inserimento
------	--------------------	----------------	---------------------	---------------	-------------

Figura 2.1: Menu di navigazione del sito

Di seguito si esaminano nello specifico le sopracitate pagine web.

3.3.1 Home

La *home page* (Fig. 2.2), come suggerisce il nome stesso, è la pagina principale dell'applicazione web, ed è inoltre la prima pagina che si apre quando si digita l'indirizzo web nell'URL senza specificare una pagina in particolare.

Il suo scopo è quello di far capire, agli utenti che la navigano, quali siano i contenuti e i servizi offerti dall'applicazione web stessa.



Figura 2.2: Screenshot della *home page* del sito

3.3.2 Inserisci sinistro

La pagina *Inserisci sinistro* (Fig. 2.3) consente all'utente, tramite la compilazione di un *form*, di effettuare l'inserimento di ogni nuovo incarico che viene affidato allo Studio peritale dalla Compagnia mandataria; l'operazione avviene tramite la memorizzazione dei dati nel database. I campi da inserire sono:

- Numero di sinistro: sequenza di caratteri che identifica in modo univoco le pratiche;
- Polizza: sequenza di caratteri alfanumerici che identifica il contratto sottoscritto dall'assicurato;
- Ramo: indica la tipologia del danno trattato, può essere ad esempio rc (responsabilità civile), rca (responsabilità civile auto), evento atmosferico, re (ramo elementare), ecc.

- CId: è il centro liquidazione danni che si occuperà, in seguito alla definizione della pratica da parte del perito, del vero e proprio indennizzo del sinistro;
- Compagnia: è l'istituto finanziario che opera offrendo prodotti assicurativi. Questo campo è facilmente compilabile tramite l'uso di un menu a tendina;
- Agenzia: rappresenta a livello territoriale la Compagnia assicurativa, il suo ruolo è quello di far stipulare contratti di assicurazione. È intermediario tra contraenti, perito e Compagnia stessa.
- Luogo del sinistro: luogo in cui è avvenuto il sinistro oggetto della pratica in gestione dallo Studio peritale;
- Data del sinistro: data in cui è avvenuto il sinistro;
- Data di affidamento: data in cui la pratica è stata consegnata per la definizione della stessa allo studio peritale da parte della Compagnia tramite l'utilizzo dei portali automatizzati di cui si faceva cenno nel capitolo precedente;
- Collaboratore: nominativo del perito a cui viene affidato il compito di seguire le varie fasi di definizione del sinistro.

Studio Colaiacomo
PERIZIE ASSICURATIVE

Home Inserisci sinistro Cerca sinistro Cerca collaboratore Cerca agenzia Inserimento

Inserisci il sinistro

Numero di sinistro

Polizza Assicurato

Ramo

CId

Compagnia Seleziona una compagnia ▼

Agenzia

Luogo del sinistro

Data del sinistro

Data affidamento del sinistro

Collaboratore

Figura 2.3: Screenshot della pagina *Inserisci sinistro*

Una volta che l'utente ha inserito correttamente tutti i dati quest'ultimi vengono salvati nel database.

Qualora la nuova pratica affidata allo Studio fosse relativa ad una polizza non ancora registrata nel database, l'applicazione prevede che il programma, al termine delle operazioni di inserimento dati da parte dell'utente, invii allo stesso un messaggio di errore per avvisarlo che il numero di polizza non risulta registrato nel database e, pertanto, occorre procedere a tale adempimento affinché l'inserimento abbia successo.

Nello specifico, l'utente viene rimandato, attraverso un link, ad un'altra pagina in cui potrà inserire, prima di procedere al vero e proprio inserimento del sinistro, la polizza, la quale verrà a sua volta salvata nel database.

Lo stesso procedimento avviene per quanto riguarda i campi *Ramo*, *Cld*, *Agenzia* e *Collaboratore*.

The screenshot displays the 'Studio Colaiacomo' web application interface. At the top, the logo 'Studio Colaiacomo' is written in a cursive font, with 'PERIZIE ASSICURATIVE' underneath. Below the logo is a navigation menu with six items: 'Home', 'Inserisci sinistro', 'Cerca sinistro', 'Cerca collaboratore', 'Cerca agenzia', and 'Inserimento'. The 'Inserisci il sinistro' section contains the following fields and values:

- Numero di sinistro: 2016/8596
- Polizza Assicurato: 108/28/0015840
- Ramo: RE
- Cld: CLD LUCCA
- Compagnia: ITALIANA (dropdown menu)
- Agenzia: La Spezia Nazionale
- Luogo del sinistro: La Spezia
- Data del sinistro: 14-06-2016
- Data affidamento del sinistro: 21-06-2016
- Collaboratore: LUIGI COLAIACOMO

At the bottom of the form is an 'Inserisci' button. Below the form, the text 'INSERIMENTO RIUSCITO' is displayed, indicating a successful submission.

Figura 2.4: Esempio di inserimento sinistro completato

Studio Colaiacomo
PERIZIE ASSICURATIVE

Home Inserisci sinistro Cerca sinistro Cerca collaboratore Cerca agenzia Inserimento

Inserisci il sinistro

Numero di sinistro

Polizza Assicurato

Ramo

Cid

Compagnia

Agenzia

Luogo del sinistro

Data del sinistro

Data affidamento del sinistro

Collaboratore

POLIZZA DELL'ASSICURATO NON PRESENTE, CLICCA [QUI](#) PER ARCHIVIARE IL NUOVO ASSICURATO

Figura 2.5: Esempio di inserimento di sinistro non riuscito

Inoltre, nel caso in cui l'utente per errore provi ad inserire un sinistro che in realtà è già presente nel *database*, apparirà un messaggio in cui verrà per l'appunto specificato che il sinistro è già memorizzato nel database.

3.3.3 Cerca sinistro

La pagina *Cerca sinistro* (Fig. 2.6) permette invece di ricercare, tramite un *form*, tutte le informazioni più significative relative ad una pratica già presente nel database. La ricerca può avvenire tramite due criteri:

- Ricerca in base al numero di sinistro;
- Ricerca in base al nominativo dell'assicurato.

Studio Colaiacomo
PERIZIE ASSICURATIVE

Home Inserisci sinistro **Cerca sinistro** Cerca collaboratore Cerca agenzia Inserimento

Digita il numero di sinistro che stai cercando e troverai tutte le informazioni che ti interessano su quel sinistro

Numero di sinistro

Se non hai il numero di sinistro puoi effettuare una ricerca tramite il nominativo dell'assicurato

Nominativo dell'assicurato

Figura 2.6: Screenshot della pagina *Cerca sinistro*

Nella pagina che mostra il risultato di una ricerca del sinistro, se il sinistro viene trovato viene inoltre data la possibilità all'utente di modificare i dati relativi al sinistro stesso. La modifica avviene tramite un *form* già compilato con i dati del sinistro che è stato appena cercato, l'utente, in questo modo, potrà scegliere il campo da correggere.

3.3.4 Cerca collaboratore e cerca agenzia

La pagina *Cerca collaboratore* (Fig. 2.8) è dedicata alla ricerca delle informazioni inerenti ai collaboratori che lavorano per lo Studio. La ricerca avviene tramite un menu a tendina in cui l'utente sceglie il nominativo del collaboratore di cui vuole trovare tutti i dati utili.

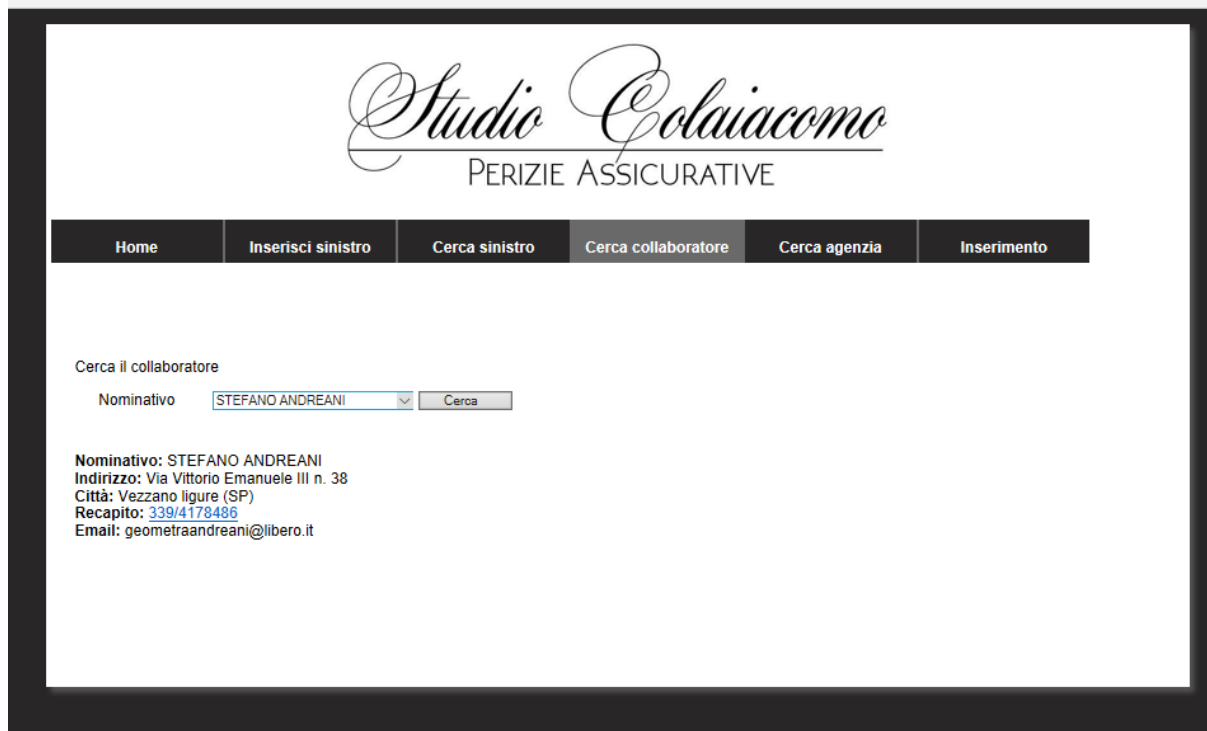


Figura 2.8: Screenshot della pagina *Cerca collaboratore*

Per quanto riguarda la pagina di ricerca Agenzia (Fig. 2.9), attraverso sempre l'utilizzo di un menu a tendina, l'utente dapprima seleziona la Compagnia di cui fa parte l'Agenzia che vuole ricerca, dopodiché ricerca l'Agenzia vera e propria.

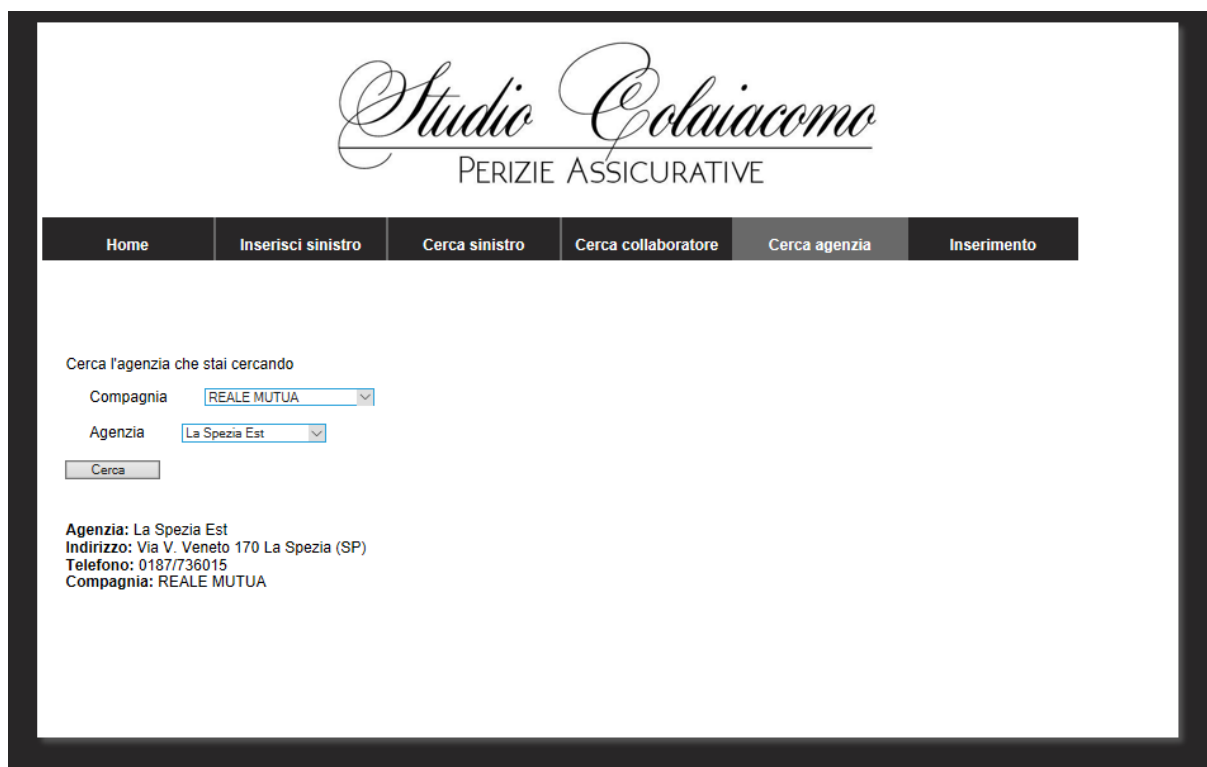


Figura 2.9: Screenshot della pagina *Cerca agenzia*

3.3.5 Inserimento

Alla voce del menu di navigazione *Inserimento* si nasconde un menu a tendina composto da diverse pagine *HTML* tra cui troviamo *Inserimento assicurato*, *Inserimento cld*, *Inserimento ramo*, *Inserimento collaboratore* ed infine *Inserimento agenzia*. Queste pagine, attraverso l'uso di un *form*, permettono all'utente di memorizzare nel database gli eventuali dati mancanti relativi ai nuovi sinistri in gestione allo Studio.

Per l'inserimento di questi dati non si è creata un'unica pagina *HTML*, ma è stata ritenuta, per più consona, la scelta di pagine *HTML* distinte in modo da non indurre l'utente nell'errore dell'inserimento del campo sbagliato



The screenshot shows a web page for 'Studio Colaiacomo PERIZIE ASSICURATIVE'. At the top, there is a navigation menu with buttons for 'Home', 'Inserisci sinistro', 'Cerca sinistro', 'Cerca collaboratore', 'Cerca agenzia', and 'Inserimento'. Below the menu, the page title is 'Inserisci una nuova compagnia assicurativa'. Underneath, there is a label 'Compagnia' followed by a text input field and an 'Inserisci' button.

Figura 2.10: Screenshot della pagina di *Inserimento compagnia*

Per la realizzazione di quanto finora esposto il primo passo necessario è stato la creazione di una base di dati che fosse in grado di contenere e gestire tutti i dati (v. cap. 4).

4 Progettazione di una base di dati

4.1 Definizione di database

Un database è un archivio di dati correlati tra loro e memorizzati su un supporto di memoria di massa. I database sono gestiti tramite software chiamati DBMS (*Database Management System*), i quali si occupano dell'aggiornamento e manutenzione dei dati e del controllo degli accessi, impedendo l'accesso ai dati a chi non possiede le autorizzazioni necessarie.

La diffusione dei database, avvenuta attorno alla fine degli anni sessanta, è aumentata sempre più soprattutto per sopperire ai limiti degli archivi tradizionali. Quest'ultimi infatti presentavano diverse problematiche, tra cui:

- **Ridondanza dei dati:** esisteva la possibilità che i dati venissero copiati più volte all'interno di uno stesso archivio e ciò comportava alcuni problemi tra cui l'**incongruenza** dei dati (i dati potevano non essere aggiornati in tutti gli archivi) e **inconsistenza** dei dati (i dati non erano più affidabili in quanto era possibile che per uno stesso dato esistessero valori differenti).
- **Difficoltà di accesso ai dati:** era possibile accedere ai dati solo attraverso programmi specifici, ed in caso di esigenze particolari era necessario sviluppare programmi *ad hoc*.
- **Accesso condiviso non consentito:** non era permesso accedere all'archivio a più utenti contemporaneamente.
- **Limitata sicurezza:** i meccanismi di sicurezza erano limitati ai servizi offerti dal *File System* che permetteva di gestirli solo a livello di file.

È chiaro quindi come la soluzione a questi limiti fosse la creazione di un sistema di gestione integrato degli archivi: raccogliendo i dati in unico grande contenitore, ossia il database, progettato in modo che i dati potessero essere condivisi in maniera semplice da applicazioni diverse e su piattaforme diverse. Si precisa comunque che ancora oggi, in casi in cui è necessario gestire una quantità molto ridotta di dati, la gestione mediante *File System* è possibile.

Sempre in ambito di database è opportuno definire il concetto di modello di dati, con il quale si intende un insieme di concetti utilizzati per organizzare i dati e descriverne la struttura in modo che essa risulti comprensibile ad un elaboratore.

I principali modelli di dati sono:

- **Modello gerarchico:** sviluppato attorno agli anni sessanta ed ha caratterizzato i primi DBMS. In questo caso i dati venivano rappresentati mediante una struttura ad albero.
- **Modello reticolare:** introdotto durante gli anni settanta, era basato sull'uso dei grafi.
- **Modello relazionale:** sviluppato nel 1970 da E.F. Codd, e introdotto poi nel decennio successivo; si basa sul concetto di relazione matematica in cui i dati sono strutturati in tabelle le cui righe (record o tuple) rappresentano gli "oggetti" mentre le colonne identificano le proprietà di tali oggetti.

- Modello ad oggetti: nasce negli anni ottanta come evoluzione del modello relazionale, estendendo alle basi di dati il paradigma della programmazione ad oggetti. Ad oggi non è ancora molto utilizzato.

I modelli sopra elencati sono stati definiti modelli logici in quanto le strutture utilizzate, pur essendo astratte, riflettono una particolare organizzazione (ad alberi, a grafi, a tabelle o ad oggetti), inoltre essi si preoccupano di descrivere la composizione e il formato dei dati. Esistono poi anche modelli di dati definiti modelli concettuali, quali ad esempio il modello entità-relazione, che permette di rappresentare i dati indipendentemente dal sistema, cercando di descrivere i concetti del mondo reale.

4.1.1 Modello relazionale

Il modello relazionale attualmente è il modello di dati più utilizzato per la produzione di DBMS in quanto fornisce sistemi semplici ed efficienti per rappresentare e manipolare dati. La rappresentazione dei dati è fondata unicamente sul concetto di relazione, intesa, non come relazione tra i dati ed il modello concettuale, ma come relazione algebrica.

Secondo la teoria degli insiemi, in algebra, per relazione si intende un sottoinsieme del prodotto cartesiano di una lista di valori, detti domini; gli elementi di una relazione invece sono chiamati tuple o record.

Le tabelle sono quindi la rappresentazione grafica della relazione algebrica, in cui ogni riga è una tupla (record) e le colonne rappresentano i campi a cui vengono dati dei nomi e sono definiti come gli attributi di ciascun record.

Nel modello relazionale viene inoltre introdotto il concetto di vincoli di integrità con il quale si indica tutto l'insieme dei valori assumibili dagli attributi di ogni tupla.

I principali vincoli di integrità sono quelli che specificano:

- Quali attributi possono assumere valore nullo: alcune volte può capitare di non essere in grado di specificare il valore di un attributo e da ciò nasce l'esigenza di ammettere nel dominio di definizione di un attributo il valore nullo.
- Quali attributi sono chiave: i vincoli di chiave nel modello relazionale sono i più importanti in quanto garantiscono l'univocità di ogni istanza di una relazione; esistono sostanzialmente tre tipologie di chiavi.

Superchiave è un insieme di attributi tale che in ogni istanza di relazione non possono esistere due tuple che coincidono su tutti gli attributi.

Superchiave minimale è una chiave per cui eliminato un suo attributo i restanti non formano più una superchiave.

Chiave primaria è una delle chiavi, e solitamente viene prediletta quella con il minor numero di attributi. Una chiave primaria non può mai assumere attributi con valori nulli poiché non garantirebbe più l'identificazione univoca di tutte le tuple della relazione.

- Quali attributi sono chiave esterna: le chiavi esterne sono fondamentali per la rappresentazione delle associazioni tra tabelle. Esse assumono il valore della chiave primaria di un'altra tabella a cui sono correlate. Il vincolo di chiave esterna può anche essere definito come vincolo di integrità referenziale o di *foreign key*.

Tramite l'algebra relazionale è possibile effettuare una serie di operazioni sulle tabelle i cui risultati sono altre tabelle. I principali operatori si possono suddividere in operatori primitivi ed operatori derivati. Tra gli operatori primitivi ci sono: *selezione* (seleziona un sottoinsieme di righe dalla tabella), *proiezione* (cancella colonne non desiderate dalla tabella, sostanzialmente elimina i duplicati), *prodotto* (consente di combinare due tabelle), *differenza* (considerando due tabelle T1 – T2, questo operatore restituisce le tuple presenti nella prima tabella ma non nella seconda), *unione* (considerando due tabelle T1 U T2, questo operatore restituisce le tuple presenti sia nella prima che nella seconda tabella) ed infine *ridenominazione* (viene usato per cambiare il tipo di una tabella). Per operatori derivati invece si intendono operatori derivati dai precedenti, tra questi ne fanno parte: *intersezione*, *join*, *natural join*, *external join* e *semijoin*.

Le tabelle nel modello relazionale possono essere collegate tramite tre tipi di relazione:

- Uno a molti (1:N): è una relazione in cui ad ogni elemento di una tabella si possono far corrispondere molti elementi di altre tabelle;
- Molti a molti (N:M): una relazione di questo tipo si ha quando a più record di una tabella corrispondono più record di un'altra tabella e viceversa;
- Uno a uno (1:1): è una relazione in cui a ciascun record di una tabella corrisponde uno ed un solo record di un'altra tabella.

4.1.2 Modello entità-relazione

Il modello entità-relazione (anche detto modello *entity-relationship*) è stato proposto nel 1976 da Chen ed è, come accennato in precedenza, il modello concettuale di dati. Descrive quindi la realtà modellata indipendentemente da come i dati verranno logicamente e fisicamente rappresentati.

Esso viene rappresentato tramite un formalismo grafico chiamato diagramma entità-relazione, o più semplicemente diagramma E-R.

I principali costrutti del modello sono:

- Entità: sono classi di oggetti con proprietà comuni ed esistenza autonoma. Ogni entità deve avere una chiave primaria ed un nome che le identifica univocamente. Nel diagramma viene rappresentato mediante rettangoli.
- Relazioni o associazioni: rappresentano i legami logici esistenti tra due o più entità, anch'esse hanno un nome che le identifica univocamente. Vengono rappresentate con dei rombi.
- Attributi: descrivono le proprietà elementari di entità e relazioni, possono essere pensati come gli attributi del modello relazionale. Vengono rappresentati con una linea unita ad un'ellisse al cui interno è specificato il nome dell'attributo stesso ed in caso di chiave primaria il nome viene sottolineato o cerchiato.

Le cardinalità delle relazioni (o associazioni) vengono specificate per ogni partecipazione di entità a una relazione e indicano il numero minimo e massimo di occorrenze di relazione a cui una occorrenza dell'entità può partecipare. Dicono quante volte, in una relazione tra entità, un'occorrenza di queste entità può essere legata a occorrenze delle altre entità coinvolte. Per definire la cardinalità si utilizzano tre valori: zero, uno e N (intero maggiore di uno).

È inoltre possibile definire cardinalità anche sugli attributi.

4.2 Fasi di progettazione di un database

La realizzazione di un database è un processo complesso in cui è necessario analizzare il contesto da rappresentare, per poi definirne la struttura, le caratteristiche e il contenuto.

A causa della sua complessità, nel corso degli anni, sono state proposte molte metodologie diverse ma quella che più si è consolidata è articolata principalmente in tre fasi da eseguire in sequenza. Tale metodologia si fonda su un principio dell'ingegneria in cui si afferma che è opportuna separare in maniera netta le decisioni relative a cosa rappresentare in un database (prima fase), da quelle relative a come farlo (seconda fase).

Le tre fasi in cui si articola il processo di realizzazione di un database sono:

- Progettazione concettuale;
- Progettazione logica;
- Progettazione fisica.

Prima di procedere alla descrizione più particolareggiata delle sopracitate fasi è bene chiarire che esiste una fase che le precede, chiamata analisi dei requisiti. Essa consiste in un processo dettagliato di individuazione e analisi delle richieste del committente. Si definiscono quali dati devono essere memorizzati e quali operazioni e applicazioni devono essere sviluppate per la gestione degli stessi.

La progettazione concettuale rappresenta il livello più alto della progettazione di un database, quello più vicino all'uomo e più lontano dalla macchina. Ha lo scopo di raffigurare la realtà di interesse, ma deve essere realizzata con strumenti e linguaggi comprensibili a tutti, non solo agli specialisti, e soprattutto deve essere indipendente dal sistema del database. Come suggerisce il nome stesso, durante questa fase viene utilizzato il modello concettuale dell'entità-relazione per poi produrre lo schema concettuale del database mediante un diagramma grafico. Tale diagramma viene fornito come punto di partenza per la fase successiva della progettazione logica.

La progettazione logica consiste nella traduzione dello schema concettuale dei dati in uno schema logico che rispecchi il modello dei dati adottato dal sistema di gestione del database e per fare ciò utilizza il modello logico relazionale. Il prodotto di questa fase è chiamato schema logico, il quale è ancora indipendente dal DBMS.

La progettazione fisica ha lo scopo di implementare lo schema logico definendo tutti gli aspetti fisici di memorizzazione e rappresentazione dei dati. Il prodotto di questa fase è denominato schema fisico e fa riferimento a un modello fisico dei dati, il quale dipende dal DBMS utilizzato. In questa fase si definiscono quali strutture fisiche utilizzare per l'implementazione della base di dati.

4.3 Utenti di un database

Gli utenti e amministratori di un database possono essere suddivisi nelle seguenti categorie:

- *Database Administrator*: è colui che gestisce gli accessi, predispone tabelle, indici, viste, ottimizza lo spazio disco ed effettua i *backup*;

- Programmatori: sono coloro che scrivono applicazioni che utilizzano i dati presenti nel *database*;
- Utente finale del database: è la figura di colui che successivamente utilizzerà le applicazioni create dai programmatori per interrogare il database;
- Sistemisti: verificano il corretto funzionamento dei sistemi hardware e software su cui gira il DBMS.

4.4 Realizzazione del database

Come già detto lo Studio Colaiacomo è uno studio peritale che si occupa dell'accertamento di sinistri assicurativi e che deve gestire numerosi incarichi che gli vengono affidati dalle Compagnie assicuratrici. Da tali necessità nasce la volontà di creare un sistema informativo in grado di raccogliere, gestire e rendere disponibili tutte le informazioni relative ai sopracitati incarichi.

Il nucleo centrale di questo sistema informativo è raffigurato da un database relazionale nel quale sono convogliate tutte le informazioni che, per l'appunto, fanno riferimento agli incarichi in gestione allo Studio.

Per la progettazione del database è stata impiegata la piattaforma XAMPP (v. cap. 5) che integra e utilizza MySQL quale RDBM, e sul quale sono state create tutte le strutture del database.

Ottenute le specifiche sopra indicate, sono state in seguito individuate diverse entità fondamentali, che sono:

- Sinistri: contiene tutti i dettagli più significativi relativi agli incarichi in gestione allo Studio. Le sue caratteristiche sono: *id, id_compagnia, id_assicurato, id_cld, id_ramo, id_collaboratore, id_agenzia, id_polizza, sinistro, luogo_sx, data_sx, data_affidamento, data_restituzione, data_accertamento, danno_accertato, danno_liquidato, data_incarico*;
- Agenzie: contiene le informazioni riguardanti per cui lo Studio lavora. Le sue caratteristiche sono: *id, filiale, id_compagnia, telefono, indirizzo*;
- Assicurati: raccoglie le informazioni relativi agli assicurati. Gli attributi individuati per questa entità sono: *id, id_compagnia, id_agenzia, nome*;
- Cld: essa rappresenta l'entità dei Centro Liquidazione Danni e le informazioni contenenti sono: *id, cld_nome, descrizione_cld*;
- Collaboratori: in essa sono contenute tutte le informazioni riguardanti le figure professionali che collaborano con lo Studio; le caratteristiche sono: *id, nominativo, telefono, indirizzo, città, regione, email*;
- Compagnie: contiene i dettagli relative alle compagnie assicuratrici, essi *sono id e nomeCompagnia*;
- Polizze: tale entità comprende i dati delle polizze di ciascun assicurato, pertanto le caratteristiche sono: *id, id-assicurato, id_compagnia, numero_polizza, nome_polizza*;
- Rami_elementari: contiene le informazioni relative ai rami elementari che caratterizzano ciascun sinistro. Le caratteristiche sono: *id, ramo e descrizione_ramo*.

Tra queste entità individuate esistono varie relazioni che permettono il collegamento tra di esse. Ad esempio l'entità Sinistri contiene tra le sue diverse caratteristiche moltissimi riferimenti ad altre caratteristiche di altre entità, come *id_compagnia* che rappresenta un rimando all'entità Compagnie.

Successivamente, dopo aver riconosciuto le varie entità, raffiguranti la realtà a cui ci si riferisce, si è passati alla fase di progettazione logica in cui ciascuna entità è stata trasformata in tabella, e le sue caratteristiche sono diventate le colonne (attributi) di queste stesse tabelle.

Una volta concretizzata la struttura relazionale del database in tabelle grazie a MySQL, è stato necessario procedere all'importazione dei dati preesistenti. Tale processo è un'operazione molto delicata in quanto deve assolutamente evitare qualsiasi perdita di informazioni.

Come è stato possibile notare confrontando le tabelle Excel che venivano utilizzate precedentemente dallo Studio come metodo di archiviazione (v. cap. 1), nel database da me creato ci sono alcune tabelle con caratteristiche molto simili. Pertanto è stato possibile, manipolando i dati in base alle nostre esigenze, utilizzare una procedura di importazione automatica che trasferisca i dati da Excel al nuovo sistema in MySQL. Per tale procedura, è necessario precedentemente trasformare le tabelle Excel in file in formato CSV (*Comma Separated Values*) in rappresentazione per l'appunto di queste stesse tabelle, dopodiché i dati in esse contenuti vengono inseriti nelle relative tabelle MySQL.

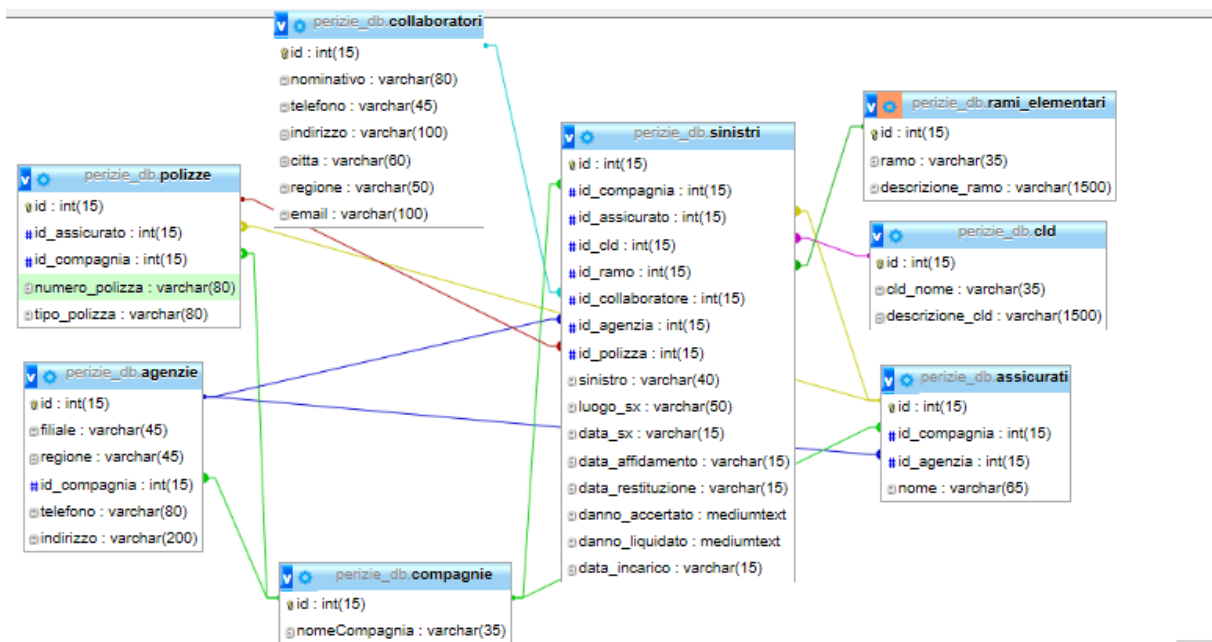


Figura 3: Struttura delle tabelle del database

5 Strumenti di sviluppo

Gli strumenti utilizzati per lo sviluppo del progetto sono stati messi a disposizione dall'azienda e comprendono linguaggi di programmazione, librerie e applicazioni.

5.1 Linguaggi lato client

I linguaggi lato *client* sono quei linguaggi di programmazione la cui elaborazione ed interpretazione vengono affidate al computer dell'utente, ovvero avvengono in locale. Da ciò ne deriva che le pagine che utilizzano la tecnologia *client* per essere visualizzate correttamente necessitano della presenza di software appositi che siano appunto in grado di interpretare le istruzioni che gli sono state inviate.

I principali linguaggi client sono:

- *HTML*
- *CSS*
- *JavaScript*

5.1.1 HTML

L'HTML (*HyperText Markup Language*) non è un vero e proprio linguaggio di programmazione (in quanto non prevede alcuna definizione di variabili, strutture dati, funzioni, strutture di controllo), ma può essere definito più correttamente come un linguaggio di marcatura. È utilizzato per descrivere le modalità di impaginazione, formattazione e visualizzazione grafica del contenuto di pagine web; viene interpretato dal browser il quale lo elabora e genera la visualizzazione delle pagine richieste.

Lo scopo dell'HTML è quindi quello di creare e gestire la struttura di pagine ipertestuali, ovvero di pagine composte da più elementi quali testo, immagini video, ecc. Tale gestione avviene tramite l'utilizzo dei *tag* (marcatori) che servono per specificare i diversi ruoli delle componenti della pagina web, tant'è che il browser stesso mostrerà formattazioni predefinite in base ai *tag* utilizzati.

I *tag*, il cui ruolo è quello di contrassegnare gli elementi della pagina HTML, sono rappresentati da una sequenza di caratteri racchiusi tra due parentesi angolate (< >). Generalmente ogni elemento è delimitato da un *tag* di apertura e un *tag* di chiusura (rappresentato come il *tag* di apertura ma preceduto dal simbolo "/"), tuttavia esistono alcune eccezioni. Ci sono difatti elementi che non richiedono l'utilizzo del *tag* di chiusura, questi devono semplicemente essere chiusi usando il carattere "/" alla fine del *tag* stesso.

Ogni documento HTML inizia con l'indicazione della definizione del tipo di documento (*Document Type Definition*), la quale segnala al browser l'URL delle specifiche HTML utilizzate per il documento, indicando quali elementi è possibile utilizzare e a quale versione di HTML si fa riferimento. Successivamente la struttura che presenta il documento è quello di un albero annidato composto da sezioni e sottosezioni delimitate da *tag*. La struttura più esterna è racchiusa tra i *tag* <html> e </html>. All'interno di questa stessa struttura si definiscono due macro aree:

- *Header*: racchiusa tra i *tag* <head> e </head>, normalmente gli elementi contenuti in questa sezione non sono visualizzati dal browser. Alcuni dei principali sono: collegamenti a

file esterni, titolo della pagina (visualizzato nella finestra del principale del browser), metadata, ecc.

- *Body*: racchiusa tra i tag `<body>` e `</body>`, rappresenta la sezione del documento che viene visualizzata dal browser.

5.1.2 CSS

I CSS (*Cascading Style Sheets*), o fogli di stile, a differenza dell'HTML che si occupa di descrivere il contenuto delle pagine web, sono impiegati per la formattazione delle stesse pagine web. Comprende ad esempio, la definizione delle caratteristiche dei *font*, colori, bordi, margini e molto altro ancora. L'uso dei CSS presenta notevoli vantaggi:

- Permette di separare il contenuto (documento HTML) dalla visualizzazione (fogli di stile – CSS) rendendo il codice più leggibile e facilitando la sua manutenzione.
- Lo stesso tema stilistico può essere utilizzato per più pagine web attraverso l'uso di selettori quali, selettori di tipo, di classi e di identificatore.

Generalmente l'inserimento più corretto del codice CSS nelle pagine web dovrebbe essere effettuato mediante l'inserimento nel tag `<head>` della pagina HTML di un collegamento ad un figlio di stile esterno.

```
1 <html>
2 <head>
3   <title>Titolo pagina web</title>
4   <link rel="stylesheet" type="text/css" href="foglio_di_stile.css"/>
5 </head>
6 </html>
```

Figura 4: Esempio di inserimento di file CSS esterno

5.1.3 JavaScript

JavaScript è un linguaggio di *scripting* orientato agli oggetti. Una delle sue caratteristiche principali è quella di essere un linguaggio interpretato, ovvero il codice non viene compilato, ma interpretato da un interprete JavaScript interno al browser, questo significa che il codice viene eseguito sul *client* e non sul *server*.

Difatti nasce per rendere maggiormente interattive le pagine web. Con esso è possibile: accedere e modificare pagine HTML, reagire ad eventi generati dall'interazione fra utente e pagina (quali click del mouse, visita ad un link, ecc), validare i dati inseriti dall'utente, lavorare con i *cookie*.

Come per i fogli di stile CSS, è consigliabile collegare alla pagina HTML codice JavaScript presente in un file esterno (tipicamente con estensione .js). Questa tecnica permette di separare nettamente la struttura del documento e il codice. Per inserire un file JavaScript esterno ci si serve del tag `<script>`:

```

1 <html>
2 <head>
3   <title>Titolo pagina web</title>
4   <script type="text/javascript" src="file_javascript.js"></script>
5 </head>
6 </html>

```

Fig. 4.1: Esempio di inserimento di file JavaScript esterno

Il suo essere un linguaggio interpretato comporta alcuni vantaggi, tra questi vi è sicuramente il fatto che, anche in presenza di *script* particolarmente complessi, il *server* non viene sovraccaricato a causa delle richieste del *client*. Allo stesso tempo tuttavia questa sua caratteristica può avere aspetti negativi poiché il tempo di scaricamento, nel caso per l'appunto di *script* complessi, può diventare molto lungo. Un altro svantaggio di JavaScript si evidenzia quando si ha la necessità di accedere a informazioni contenute all'interno di un database remoto, in questo caso l'acquisizione deve essere affidata ad un altro linguaggio, il quale effettua esplicitamente la transizione e restituisce i risultati ad una o più variabili JavaScript. Si è ovviato a questa problematica con l'avvento della tecnica AJAX.

AJAX (*Asynchronous JavaScript and XML*) è una tecnologia scritta in JavaScript utilizzata per lo sviluppo di applicazioni web dinamiche dedicate allo scambio asincrono di informazioni tra *client* e *server*. La sua caratteristica di asincronia si riferisce al fatto che i dati sono richiesti al server on background senza interferire con l'operatività della pagina corrente. Dall'uso di questa tecnologia ne derivano diversi vantaggi tra cui: minor quantità di traffico scambiato tra *client* e *server* (non è più necessario attendere il ricaricamento di tutta la pagina ma solamente una sua parte), maggior velocità di interazione con l'utente ed inoltre c'è la possibilità di effettuare richieste simultanee.

5.1.4 jQuery

jQuery è una libreria di funzioni JavaScript. L'obiettivo principale del suo utilizzo è quello di rendere le pagine web visualizzabili dalla maggior parte dei browser; offre: servizi per la manipolazione dei fogli di stile, degli elementi HTML, per la creazioni di effetti o animazione ed anche metodi per l'uso di AJAX.

Per utilizzare la libreria jQuery è sufficiente scaricarla dal sito ufficiale e collegarla alla pagina HTML inserendola nel tag `<head>`:

```

1 <html>
2 <head>
3   <title>Titolo pagina web</title>
4   <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"></script>
5 </head>
6 </html>

```

Figura 4.2: Esempio di inserimento della libreria jQuery nella pagina HTML

5.2 Linguaggi lato server

I linguaggi lato *server* sono quei linguaggi di programmazione che vengono interpretati ed elaborati dal *server*, il quale mette a disposizione una serie di servizi utili all'acquisizione di informazioni e funzionalità non disponibili nella macchina dell'utente. Un importante aspetto di questa tipologia di

linguaggi è che non sono direttamente resi disponibili all'utente, il quale può solamente visualizzare il risultato del programma precedentemente elaborato dal server.

5.2.1 PHP

PHP (*Hypertext Preprocessor*) è un linguaggio di *scripting open source*, indicato per lo sviluppo di applicazioni web. Come per JavaScript può essere definito un linguaggio interpretato ma in questo caso l'interprete del PHP è il *server*.

Il PHP permette di interagire con diversi tipi di database, può raccogliere dati, generare contenuti dinamici, interagire con i *cookie* e con altri servizi.

Un altro aspetto fondamentale da sapere dei file PHP è che sono strutturati come documenti HTML ma contengono sezioni di codice delimitate dai tag `<?php` e `?>`, le quali vengono elaborate dall'interprete PHP che successivamente le sostituisce con il codice HTML corrispondente alla sua esecuzione, tutto ciò che invece è al di fuori dei delimitatori `<?php` e `?>` viene ignorato dall'interprete e passato direttamente al *client* nella sua versione inalterata.

Per lo sviluppo del progetto finale, quando si è trattato di programmare in linguaggio PHP è stato reso necessario l'utilizzo di diversi strumenti quali ad esempio XAMPP.

XAMPP (v. fig. 4.3) è un pacchetto software gratuito contenente APACHE HTTP Server, il database MySQL e tutti gli strumenti necessari per utilizzare i linguaggi di programmazione PHP e Perl. Il nome è un acronimo dei programmi che compongono questo pacchetto, mentre la lettera "X" sta per *cross-platform* ciò significa che è multiplatforma ovvero funziona sia su ambienti Linux sia su Mac che Windows.

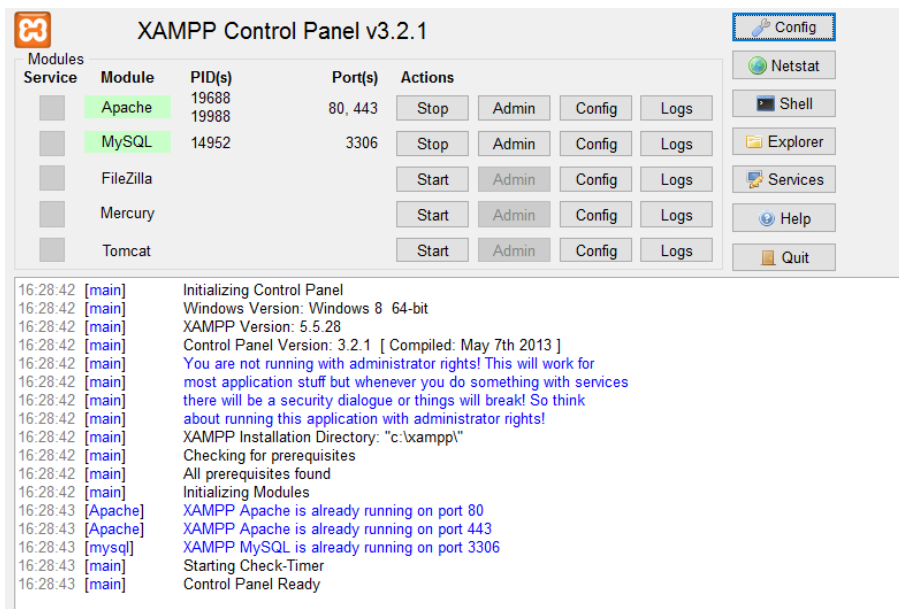


Figura 4.3: Pannello di controllo di XAMPP

APACHE HTTP Server, o più comunemente noto semplicemente come APACHE, è una piattaforma *server web*, disponibile per più sistemi operativi come Windows e Linux. Esso permette di trasformare il proprio computer in un *server* in grado di ospitare veri e propri siti internet. La funzione del *server web* è fondamentale in quanto è in grado di interpretare il codice PHP di pagine web dinamiche processando queste stesse istruzioni e mostrando poi all'utente il risultato finale. APACHE presenta un'architettura modulare, quindi ad ogni richiesta del client vengono svolte funzioni specifiche da ogni modulo di cui è composto, come unità indipendenti.

MySQL è un RDBMS (*Relational Database Management System*) ovvero un database relazionale *open source*. Per *database* relazionale si intende un insieme di tabelle, ognuna di esse composte da righe identificate per mezzo di un codice univoco denominato chiave primaria. Le tabelle non sono del tutto indipendenti tra loro ma relazionate da legami logici.

phpMyAdmin (v. fig.4.4) è un'applicazione scritta in PHP che permette di gestire un database MySQL mediante un'interfaccia grafica, che grazie alla sua semplicità permette di svolgere in modo molto intuitivo diverse operazioni. Queste comprendono: la creazione o l'eliminazione di un intero database, la gestione delle tabelle (creazione, modifica, eliminazione, e definizione di vincoli) e dei record in esse contenuti (inserimento, modifica, eliminazione e sviluppo di *query*).

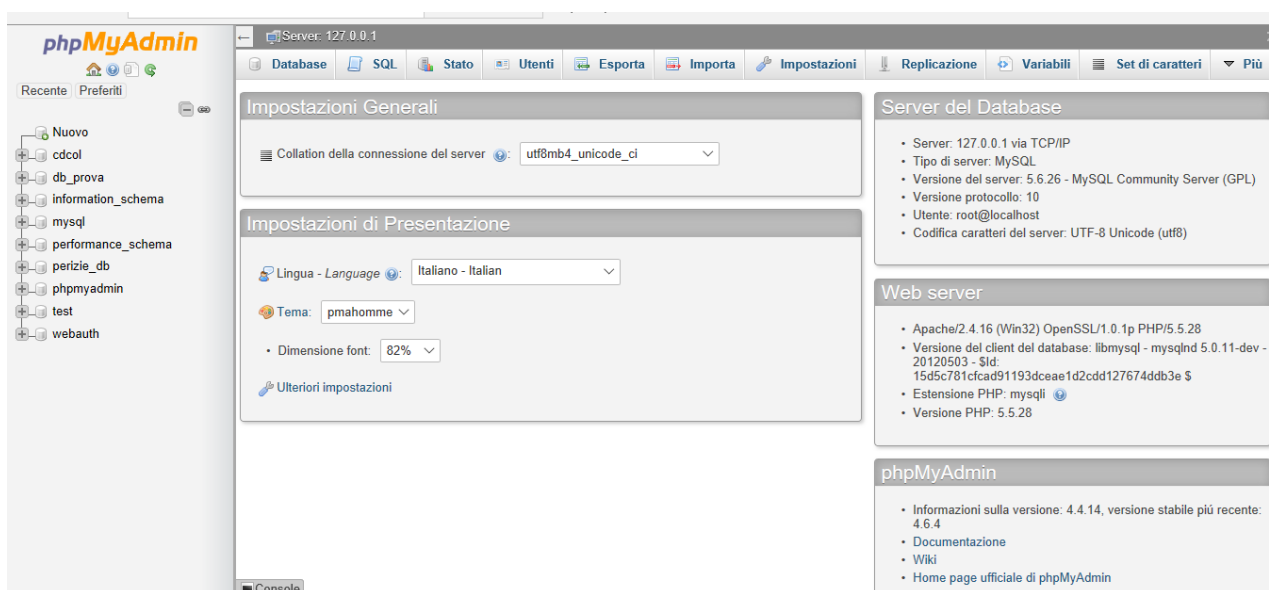


Figura 4.4: Interfaccia di phpMyadmin

5.2.2 SQL

SQL (Structured Query Language) è il linguaggio che permette di effettuare le operazioni per estrarre e manipolare i dati da un database. È lo standard tra i sistemi relazionali, viene infatti usato in tutti i prodotti DBMS come set di comandi per l'utente della base di dati.

All'interno di SQL fanno parte diversi tipi di istruzioni, tra cui:

- **DDL (*Data Definition Language*):** consente di descrivere la struttura logica del *database*, comprende tutti quei comandi per la creazione, modifica e cancellazione delle tabelle,

definizione degli indici, specifica inoltre i vincoli e integrità referenziali. Alcuni esempi di comandi sono: CREATE TABLE (crea una nuova tabella), ALTER TABLE (modifica la struttura di una tabella), DROP INDEX (elimina l'indice specificato), ed altri ancora.

- DML (*Data Manipulation Language*): permette di modificare i dati contenuti nel database, tramite le operazioni di inserimento, cancellazione e variazione, tra questi ne fanno parte: INSERT, DELETE, UPDATE.
- DQL (*Data Query Language*): è il linguaggio che permette di porre interrogazioni al database. Alcuni esempi: SELECT (consente di estrarre i dati in modo mirato), JOIN (come per il comando SELECT, permette di estrarre dati in modo mirato ma ciò avviene dall'unione di più tabelle)-
- DCL (*Data Control Language*): viene utilizzato per gestire il controllo degli accessi e i permessi degli utenti.

6 Conclusioni

Prima di introdurre l'applicazione che ho sviluppato, ho ritenuto opportuno descrivere l'attività di cui si occupa uno studio peritale al fine di far comprendere al lettore le problematiche che nel quotidiano possono incontrarsi nell'archiviazione delle pratiche.

In tale ottica, è stato necessario effettuare un raffronto tra le modalità di archiviazione utilizzate prima del mio intervento e quelle adottate successivamente, applicando il programma elaborato.

Si è proceduto quindi con una breve analisi generale delle necessità del committente, ed in seguito si è passati all'implementazione, analizzando nel dettaglio le funzionalità. In ultimo sono stati esaminati gli strumenti di sviluppo utilizzati per la realizzazione del progetto finale.

La tecnica illustrativa adottata ha permesso di porre in evidenza le peculiarità del programma ed i vantaggi che lo stesso garantisce all'utente.

6.1 Sviluppi futuri

Il progetto realizzato offre molti spunti per sviluppi futuri. Sarà infatti necessario completare l'applicazione anche con ulteriori funzionalità.

Nella prima versione dell'applicazione è stato deciso di focalizzare l'attenzione su altri aspetti, come il data base e l'archiviazione vera e propria dei dati, poiché quest'ultimi sono stati ritenuti più fondamentali per lo scopo dell'intero progetto, che era per l'appunto quello di creare un nuovo metodo di archiviazione più rapido ed efficace.

Tra i possibili sviluppi futuri c'è la necessità di creare la funzione di *login* per permettere, solamente agli utenti registrati, di accedere, e soprattutto modificare i dati. Questa funzione non è stata creata fin dall'inizio dello sviluppo del programma in quanto ad oggi l'applicazione non è ancora disponibile in rete.

L'altro rilevante sviluppo riguarda la possibilità di rendere disponibile all'utente che naviga l'applicazione, non solo l'opzione di ricerca i dati più significativi relativi agli incarichi che lo Studio ha in gestione, ma anche i documenti veri e propri.

Ed inoltre, tra le varie ipotesi di sviluppo, è emersa la possibilità di rendere un domani l'applicazione fruibile anche da smartphone.

7 Ringraziamenti

Vorrei dedicare quest'ultima pagina per ringraziare tutti coloro che in me hanno sempre creduto e con cui ho condiviso momenti importanti, soprattutto quelli più difficili.

Desidero ringraziare il Professor Casarosa, relatore di questa tesi, per la grande disponibilità e cortesia dimostratemi, e per l'aiuto fornitomi durante la stesura .

Un ringraziamento particolare lo dedico alla mia famiglia, perché mi è sempre stata accanto e non mi ha mai fatto mancare il suo sostegno e il suo aiuto in questi anni e so che per me ci sarà sempre. Senza di loro non sarei mai diventata la persona che sono oggi.

8 Bibliografia

Albano, Antonio, Giorgio Ghelli e Renzo Orsini. 2005. *Fondamenti di basi di dati*. Bologna, Zanichelli.

Atzeni, Paolo, Stefano Ceri, Piero Fraternali, Stefano Paraboschi e Riccardo Torrone. 2013. *Basi di dati: modelli e linguaggi di interrogazione*. Milano, McGraw-Hill.

Yank, Kevin. 2012. *Sviluppare applicazioni con PHP e MySQL*. 2012. Milano, Apogeo.

Guercio, Maria, Stefano Pigliapoco, Federico Valacchi. *Archivi e informatica*. 2010. Lucca, Civita editore.

9 Sitografia

HTML. <http://www.w3schools.com/html/> (Consultato il 7 settembre 2016)

CSS. <http://www.w3schools.com/css/> (Consultato il 7 settembre 2016)

JavaScript. <http://www.w3schools.com/js/> (Consultato il 7 settembre 2016)

PHP. <http://www.w3schools.com/php/> (Consultato il 12 settembre 2016)

SQL. <http://www.w3schools.com/sql/> (Consultato il 12 settembre 2016)

XAMPP. <https://www.apachefriends.org/it/index.html> (Consultato il 15 settembre 2016)

phpMyAdmin. <https://www.phpmyadmin.net/> (Consultato il 15 settembre 2016)